

LEABHARLANN CHOLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH | TRINITY COLLEGE LIBRARY DUBLIN
Ollscoil Átha Cliath | The University of Dublin

Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

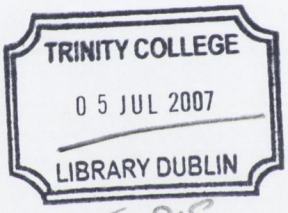
Optimum Stability in Control System Design

A dissertation submitted to the University of Dublin
for the degree of Doctor of Philosophy

Brian Cogan
Trinity College Dublin, November 2006

Department of Electronic and Electrical Engineering,
Trinity College Dublin





THOS S.
8104

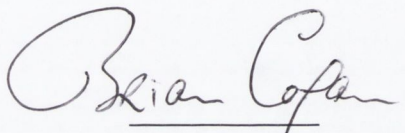
Dedication

This thesis is dedicated to my parents, Daniel and Elizabeth, to my wife Bernie, and to my daughter Hannah.

Declaration

I declare that I am the sole author of this thesis and that all the work presented in it, unless otherwise referenced, is my own. I also declare that this work has not been submitted, in whole or in part, to any other university or college for any degree or other qualification.

I authorize the library of the University of Dublin to copy or to lend this thesis.



Brian Cogan

November 2006

Summary

This thesis develops for the first time a general approach to the design of control systems that emphasizes optimum system stability as the primary design criterion. The design method is to select controller parameters that place the system's rightmost eigenvalue as far to the left as possible in the s-plane. When the system is operating at this point it is said to be in a state of optimum stability. This method is applied to the design of PI and PID controllers as more than 95% of controllers used in industrial applications are either PI or PID type (see references in section 6.1.1). The optimum stability design method is developed in the context of the root locus diagram, performance integrals, the Lyapunov matrix equation, and the Routh array and the equivalences between these characterizations are explored.

The procedures employed here do not maximise system stability - the purpose is to optimise system stability. By placing the rightmost eigenvalue as far to the left as possible we are minimizing the largest time constant of the system. We find, by using standard robustness measures, that the performance of controllers designed using these eigenvalue-assigning methods is better than the performance achieved using currently available design techniques.

New methods for calculating time-weighted performance integrals and performance sums are presented and several classical results are given new and simpler derivations. New methods are given for solving the continuous-time and discrete-time Lyapunov matrix equations using the Laplace and Z transforms respectively.

Root-locus-based optimum stability is used to design a PID controller for an unstable, non - minimum phase process, to design PI and PID controllers for multi-lag processes, and to design a second order system. The root locus approach is then extended to the design of PI controllers for time - delay processes. By using standard robustness measures, and by examining time responses, the performance of the designs based on optimum stability is seen to be better than the performance achieved using currently available design techniques. A relationship between root-locus-based optimum stability and an exponentially-weighted performance integral is also explored.

A method is developed to design PID controllers for multi-lag processes based on optimum stability using the Lyapunov matrix equation. This design method is important when there are more than two system parameters. In the case of a problem with two parameters, the Lyapunov method gives the same controller as the root locus method.

The idea that optimum stability may be at work in nature is explored in the context of a model of the human balance control system. The optimum stability criterion is used to select parameters for this model via the Nyquist diagram and the Lyapunov matrix equation.

Optimum stability is also interpreted in the context of the Routh array. It is shown that this technique produces the same controllers as the root-locus-based method. Controllers are designed for an unstable, non-minimum phase process, for multi -lag processes, and a study is made of the n -let polynomial.

Another controller design approach, based on the Maximum Power Transfer Theorem from standard linear AC circuit theory, is described. Controllers designed using this method are compared to controllers designed using optimum stability from root locus, among other criteria.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Professor Annraoi M. de Paor of U.C.D. for his help and encouragement during all stages of this research. It was a joy and a great privilege to carry out this work in collaboration with such a gifted and distinguished researcher.

I would also like to thank Dr. Anthony Quinn of T.C.D. for agreeing to supervise my work at T.C.D. and Professor Frank Boland of T.C.D. for his many helpful comments.

When I was having a problem with my "big sum" my daughter Hannah often advised me to "try something unexpected" or to "do the opposite of what you have tried already" - her advice was always very welcome.

Contents

List of Figures ..... **ix**

List of Tables ..... **xvi**

Chapter 1: Control System design methodologies **1**

1.1 Introduction *1*

1.2 Optimum stability from root locus *1*

1.3 Optimum stability using performance integrals *2*

1.4 Optimum stability using the Lyapunov matrix equation *2*

1.5 Optimum stability from the Routh-Hurwitz Criterion *4*

1.6 Optimum stability from the Nyquist diagram - optimizing gain and phase margins *4*

1.7 The maximum power transfer theorem - another setting for optimality *5*

1.8 Thesis organization *6*

1.9 Publications *7*

Chapter 2: A unified treatment of control system performance measures for continuous and discrete time system **8**

2.1 Introduction *8*

2.2 Kronecker product method for calculating performance integrals *9*

2.2.1 Condition for the existence of solutions for equation (2.2) 10

2.2.2 A new method for reducing the number of equations to be solved to a minimum 10

2.2.3 Constructing the matrix E 13

2.2.4 Constructing the matrix D 14

2.2.5 Notes on the matrices E and D 15

2.3 Example of calculating time weighted performance measures by using equation (2.4) *15*

2.4 Two methods for calculating exponentially weighted performance integrals *18*

2.4.1 Exponentially weighted performance integrals in terms of an infinite sum of matrices 18

2.4.2 Exponentially weighted performance integrals in terms of the Lyapunov matrix equation ... 19

2.5 Calculating discrete time system performance measures - a new derivation of Jury's procedure 20

2.5.1 A note on one difference between discrete time and continuous time systems 23

2.5.2 A note on solving embedded discrete time Lyapunov equations 23

2.5.3 A note on the condition for the existence of solutions for equation (2.11) 24

2.6 A new and simple method for computing the Barnett-Jury coefficients using a number triangle . *24*

2.7 A relationship between Barnett-Jury coefficients and Stirling Numbers of the Second Kind *31*

2.8 Proving a property of Barnett-Jury coefficients using Stirling Numbers of the Second Kind *32*

2.9 Further interesting and novel properties of the Barnett-Jury coefficients *33*

2.10 A method for calculating exponentially weighted performance sums *33*

2.11 Summary of results in this chapter *34*

2.12 Suggestions for future work *34*

Chapter 3: Using root locus based optimum stability to design a controller of minimum complexity for an unstable, non-minimum phase process **35**

3.1 Introduction *35*

3.2 A desirable controller structure *35*

3.3 The minimum order necessary for a controller *36*

3.4 H_∞ controllers can be fragile and of high order *37*

3.5 Root locus based design for a controller and a new procedure for calculating system parameters *37*

3.6 Discussion and conclusions *42*

Chapter 4: Root locus based optimum stability and a relationship with a class of exponentially weighted performance integrals **43**

4.1 A second order system *43*

4.1.1 Design for a second order system using root locus based optimum stability 43

4.1.2 Design for a second order system by minimising a performance integral 44

4.2 PI controller design for the unstable process $G(s) = 1/(s-1)$ *46*

4.2.1 Design for a PI controller for the process $G(s) = 1/(s-1)$ using root locus based optimum stability 46

4.2.2 Design for a PI controller for the process $G(s) = 1/(s-1)$ by minimising a performance integral 48

4.2.3 Comparison between the PI controller from optimum stability and a PI controller derived from the centroid of a stability region 50

4.3 Conclusions *52*

4.4 Suggestions for future work *52*

Chapter 5: PI and PID controller tuning using root locus based optimum stability **53**

5.1 Introduction *53*

5.2 PI controller design for multi-lag processes *53*

5.2.1 Ziegler-Nichols tuning for PI controller for $G(s) = 1/(s+1)^4$ 53

5.2.2 Fine-tuning of the Ziegler-Nichols PI controller for the process $G(s) = 1/(s+1)^4$ using root locus 54

5.2.3 PI controller for the process $G(s) = 1/(s+1)^4$ from root locus based optimum stability 55

5.2.4 Other methods for tuning PI controllers 57

5.2.5 Comparison of PI controllers for the process $G(s) = 1/(s+1)^4$ 57

5.2.6 Generalised root locus based optimum stability approach for designing a PI controller for the multi-lag process $G(s) = k_2/(s+b)^m$ 60

5.3 PID controller designs for multi-lag processes *60*

5.3.1 Ziegler-Nichols tuning for a PID controller for the process $G(s) = 1/(s+1)^4$ 60

5.3.2 Fine-tuning the Ziegler-Nichols PID controller for the process $G(s) = 1/(s+1)^4$ using root locus 61

5.3.3 Dynamics of Ziegler-Nichols and fine-tuned Ziegler-Nichols controllers 62

5.3.4 Discussion of the Ziegler-Nichols and fine-tuned Ziegler-Nichols controllers 63

5.3.5 Generalised root locus based optimum stability approach for designing a PID controller for the process $G(s) = k_2/(s+b)^m$ 63

5.3.6 PID controller for $G(s) = 1/(s+1)^4$ derived by various methods in the literature 66

5.3.7 Comparison of five PID controllers for the process $G(s) + 1/(s+1)^4$ 67

5.3.8 Discussion 68

5.3.9 PID controllers for the process $G(s) = 1/(s+1)^5$ designed by various methods 69

5.3.10 Comparison of six PID controllers for the process $G(s) = 1/(s+1)^5$ 70

5.3.11 Discussion 72

5.4 PID controller design for the double integrator process *72*

5.4.1 Performance integral for the system in Figure 5.25 75

5.5 Summary of results in this chapter *76*

5.6 Suggestions for future work *77*

Chapter 6: PI controller design for processes with time delay using the root locus method **78**

6.1 A procedure for designing PI controllers for general first-order lag plus time-delay process *78*
6.1.1 Introduction 78
6.1.2 Root locus diagrams for a PI controller and a general FOLPD process 79

6.2 Exploiting the triple eigenvalue *83*

6.3 Root locus based design for a PI controller for a general FOLPD process *84*
6.3.1 Comparing designs from optimum stability with designs from performance integrals 85
6.3.2 Comparing designs from optimum stability with designs from stability region methods 89

6.4 Discussion of PI controller designed using optimum stability for the FOLPD process *91*

6.5 The analytic root locus method for designing controllers for systems with time delay and a relationship with the Lambert W function *92*
6.5.1 Illustrative example of the analytic method from Palm [75] 92
6.5.2 Analytic method for designing PI controllers for an integrator with time delay 93
6.5.3 Evolution of the root locus for different values of F – illustration of the triple eigenvalue ... 94
6.5.4 An example of the gain equation 96
6.5.5 A relationship between the Bendrikov-Teodorchik equation and the Lambert W function ... 96

6.6 Calculating performance integrals for time delay systems *97*

6.7 Conclusions *99*

6.8 Suggestions for further work *99*

Chapter 7: PI and PID controller tuning by analogy with the Maximum Power Transfer Theorem of Circuit Theory **100**

7.1 Introduction *100*

7.2 Preliminary development of the theory *102*

7.3 Comparison of PI and PID controller designs for the process $G(s) = 1/(s+1)^3$ *based on maximum power transfer* *104*

7.4 Comparison of PID controller designs based on optimum stability, max power transfer, and centroid of stability region for the process $G(s) = 1/(s+1)^8$ *106*

7.5 Summary of results *109*

7.6 Suggestion for further work *109*

Chapter 8: Nyquist and Lyapunov based optimum stability in modeling the control of balance during quiet standing **110**

8.1 Introduction *110*

8.2 The characteristic equation of the model *111*

8.3 Selecting pairs of parameters to optimise phase margin on the Nyquist diagram *113*
8.3.1 Nyquist analysis with visual / somatosensory system only active 113
8.3.2 Nyquist analysis with vestibular system only active 114
8.3.3 Dynamics of the model when two pairs of operating points are combined 115

8.4 Selecting all four parameters simultaneously *118*
8.4.1 Optimum vector margin design 118
8.4.2 Optimum stability using the Lyapunov matrix equation 119

8.5 Summary of results in this chapter *122*

8.6 Suggestions for future work *122*

Chapter 9: Optimum stability from the Routh array and the Jeltsch-Fichera array 123

9.1 Introduction 123

9.2 Some properties of the Routh and Jeltsch-Fichera arrays 123
9.2.1 Constructing the Routh and Jeltsch-Fichera arrays 123
9.2.2 Stability from the Jeltsch-Fichera array 124

9.3 Optimum stability, the principal minors of a Hurwitz matrix, and the Jeltsch-Fichera array 125

9.4 Optimum stability, the Jeltsch-Fichera array, and the Routh array 126

9.5 The last n entries of all of the columns the Routh and Jeltsch-Fichera Arrays vanish at an optimum stability point 130

9.6 PID controller for an unstable, non-minimum phase process 133

9.7 Stability domain and optimum stability point for the n-let polynomial 134

9.8 Generalised Routh array based optimum stability approach for designing a PI controller for the process $G(s) = k_2/(s+b)^m$ 136

9.9 Summary of results in this chapter 136

9.10 Suggestions for future work 136

Chapter 10: Transform Methods for solving the Lyapunov matrix equations with applications 137

10.1 Introduction 137

10.2 Calculating time weighted performance integrals using the Laplace transform 137
10.2.1 Performance integrals with polynomial time weighting using the Laplace transform 137
10.2.2 Performance integrals with general time weighting 138

10.3 Analytic Solutions to the Lyapunov matrix equation 138

10.4 Using the Laplace transform to solve the continuous time Lyapunov matrix equation 139
10.4.1 Example 1 - Laplace transform solutions to the continuous time Lyapunov matrix equation 140
10.4.2 Extending Example 1 to infinitely many Lyapunov Matrix Equations 142
10.4.3 Example 2 - Laplace transform solutions to the continuous time Lyapunov matrix equation 142
10.4.4 Extending Example 2 to infinitely many Lyapunov Matrix Equations 143
10.4.5 Example 3 - Laplace transform solutions to the continuous time Lyapunov matrix equation 143
10.4.6 Discussion 144

10.5 General solution to the discrete time Lyapunov matrix equation using the $\mathcal{Z}$ transform 144

10.6 Calculating time weighted performance sums using $\mathcal{Z}$ transforms 146

10.7 Suggestions for future work 146

Chapter 11: Summary and Conclusions 147

11.1 Suggestions for further work 149

Appendix A: Performance integrals 150

A.1 The need for performance integrals 150

A.2 Using performance integrals to design a controller 151

A.3 Overview of the literature on performance integrals 151
A.3.1 Performance integrals for continuous time systems 151
A.3.2 Performance sums for discrete time systems 154

A.4 MacFarlane's procedure for calculating performance integrals 155

Appendix B: New derivation of the discrete time system performance sums S_n for n=0,1,2 and 3158

B.1 Discrete time system performance sums with n = 0 158

B.2 Discrete time system performance sums with n = 1 ..... *158*

B.3 Discrete time system performance sums with n = 2 ..... *159*

B.4 Discrete time system performance sums with n=3 ..... *161*

Appendix C: H_2 and H_∞ norms **163**

C.1 A note on terminology *163*

C.2 System norms *163*

C.2.1 H_2 - norm controllers and error minimization 164

C.2.2 Computing the H_2 norm 164

C.2.3 Physical interpretation of the H_2 norm 164

C.2.4 H_∞ - norm controllers and error minimization 165

C.2.5 Computing the H_∞ norm 165

C.2.6 Physical interpretation of the H_∞ norm 166

C.2.7 Advantages and disadvantages of using the H_∞ norm to design controllers 166

Appendix D: History and present state of tuning rules for PI and PID controllers **168**

D.1 Introduction *168*

D.2 Ziegler –Nichols *168*

D.2.1 Some general comments on Ziegler-Nichols tuning 168

D.2.2 Ziegler-Nichols method based on process reaction curves 169

D.2.3 Ziegler-Nichols method based on ultimate sensitivity 169

D.2.4 Minimizing performance integrals 170

D.2.5 Specifying Gain Margin and Phase Margin 170

D.2.6 Nyquist diagram 171

D.2.7 H_∞ - norm 171

D.2.8 Stable polyhedra in parameter space 171

Appendix E: Number triangles and performance measures **172**

Appendix F: Highlights of the history of Stability Theory **177**

Bibliography **178**

List of Figures

Figure 1.1 Organization of this thesis 6
Figure 2.1 A PI controller and a process in a unitary gain, negative feedback loop 15
Figure 2.2 Root locus for $p(s) = s(s-1)(s+2) + 3k(s+a)$ with $a = 1/70$ 16
Figure 2.3 Root locus for $p(s) = s(s-1)(s+2) + 3k(s+a)$ with $a = 1/63$ - this is the critical value of a that results in a breakpoint at a triple eigenvalue 16
Figure 2.4 Columns of integers raised to powers of 1 to 5. Subsequent columns show the differences between pairs of figures in the previous columns. The top row of each array (in bold) consists of Barnett - Jury coefficients. 25
Figure 2.5 A number triangle for generating the Barnett-Jury coefficients 26
Figure 2.6 Generating a number triangle similar to Pascal's triangle but with the diagonals weighted as shown. For example $\mathbf{2}\times 3 + \mathbf{3}\times 2 = 12$ and $\mathbf{4}\times 2100 + \mathbf{5}\times 3360 = 25200$ etc. etc. This scheme allows one to calculate the Barnett-Jury coefficients for any value of i. Coefficients from $i = 0$ to $i = 9$ are given in this figure. Jury [42] gives coefficients up to $i = 10$ 26
Figure 2.7 Jury's method [34] for calculating b_{ij} with $i = 4$ 27
Figure 2.8 Indexing protocol used in this section 27
Figure 2.9 Finite difference array for n^i 28
Figure 2.10 Top row of Figure 2.9 with $i = 4$ 28
Figure 2.11 Top row of Figure 2.9 with i=0,1,2,3,4 29
Figure 2.12 Stirling Numbers of the Second Kind written as a number triangle. An indexing protocol for the rows and columns is indicated. 31
Figure 2.13 Number triangle of Barnett-Jury coefficients with signs included. This is used to illustrate that all rows, except the first, sum to zero. This property is not remarked upon in the literature and may have been unnoticed up to now. 32
Figure 3.1 Negative feedback system where R(s) = reference input, U(s) = controller output, Y(s) = system output, D(s) = disturbance input. 35
Figure 3.2 Root locus of (3.3) with α as parameter. There is a quadruple eigenvalue at the leftmost breakpoint. 40
Figure 3.3 Root locus of (3.3) with β as parameter. There is a quadruple eigenvalue at the leftmost breakpoint. 40
Figure 3.4 Root locus of (3.3) with γ as parameter. There is a quadruple eigenvalue at the leftmost breakpoint. 41
Figure 3.5 Root locus of (3.3) with k as parameter. There is a quadruple eigenvalue at the leftmost breakpoint. 41
Figure 3.6 Nyquist plot for equation (3.6) 42
Figure 4.1 An example of a second order system 43
Figure 4.2 Root Locus of $p(s) = (s - j\omega_n)(s + j\omega_n) + 2\zeta\omega_n s$ with ζ as parameter 44
Figure 4.3 Plot of $4J_\alpha\omega_n$ against ζ showing a minimum of at $4J_\alpha\omega_n = 1.75$ at $\zeta = 1$ 46
Figure 4.4 PI controller and a first order unstable process in a unitary negative feedback control system. 47
Figure 4.5 Root locus of $p(s) = s(s-1) + k(s+a)$ with k as parameter 47
Figure 4.6 Plot of $J_\alpha(k) = \int_0^\infty \exp(-2.4t)\,\mathbf{x}^T(t)\mathbf{Q}_1\,\mathbf{x}(t)dt$ against k. 49
Figure 4.7 The gray region is part of the stability domain for the system in Figure 4.4. 50
Figure 4.8 Step Reference responses for system in Figure 4.4 – optimum stability design in solid, centroid design in dashed. 51
Figure 4.9 Step Disturbance responses for system in Figure 4.4 – optimum stability design in solid, centroid design in dashed. 51
Figure 4.10 Impulse Disturbance responses for system in Figure 4.4– optimum stability design in solid, centroid design in dashed. 51
Figure 5.1 Unitary negative feedback control system. 53
Figure 5.2 Root locus of $p(s) = (s+1)^4 + k$ with k as parameter. 54
Figure 5.3 Root locus for $p(s) = 5.215s(s+1)^4 + k1.8(5.215s+1)$ with k as parameter. 54
Figure 5.4 Root locus for (5.4) with k as parameter and with $a = 0.4$, which is below the critical value. The system has not yet reached a state of optimum stability. 55

Figure 5.5 Root locus for (5.4) with k as parameter and with $a = 0.64$ showing the breakpoint at a triple eigenvalue. This critical value of a, along with the corresponding value for k, allows us to design the PI controller that leads to optimum stability. 56
Figure 5.6 Root locus for (5.4) with k as parameter and with $a = 0.8$. The system has moved away from a state of optimum stability. 56
Figure 5.7 Step reference responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response, with no overshoot and the fastest settling time. 58
Figure 5.8 Step disturbance responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response and the fastest settling time, but it has the most overshoot 59
Figure 5.9 Impulse disturbance responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response, with greatest overshoot but the fastest settling time 59
Figure 5.10 Root locus of $p(s) = s(s+1)^4 + k(1.885)\left(s + 2/\pi\right)^2$ with k as parameter 61
Figure 5.11 Step reference responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has less overshoot and is smoother than the Ziegler Nichols system. 62
Figure 5.12 Step disturbance responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has larger overshoot but is smoother than the Ziegler Nichols system 62
Figure 5.13 Impulse disturbance responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has slightly larger overshoot but is smoother than the standard Ziegler Nichols system. 63
Figure 5.14 Root locus of equation (5.15) with $m = 8$, with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense. 64
Figure 5.15 Root locus of equation (5.15) with $m = 9$, with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense. 64
Figure 5.16 Open-loop unit step response for the process $G(s) = 1/(s+1)^4$. The values of $T_u = 1.2\ s$ and $T_g = 4.8\ s$ allow the design of a CHR-type PID controller. 66
Figure 5.17 Step reference responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from the Lyapunov equation results in the smoothest response, with no overshoot and the fastest settling time. The response with the controller designed using root locus based optimum stability is smooth and has no overshoot but has a longer settling time than the Lyapunov equation based controller. 67
Figure 5.18 Step disturbance responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from the Lyapunov equation results in a smooth response with no undershoot and the fastest settling time. The controller designed using root locus based optimum stability is also smooth and has no undershoot but has greater overshoot than the Lyapunov equation based controller. 68
Figure 5.19 Impulse disturbance responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from root results in the smoothest response with least undershoot and the fastest settling time 68
Figure 5.20 Open-loop unit step response for the process $G(s) = 1/(s+1)^5$. The values of $T_u = 2.0\ s$ and $T_g = 5.4\ s$ allow the design of a CHR-type PID controller. 69
Figure 5.21 Step reference, step disturbance, and impulse disturbance responses for the two systems with PID controllers designed for the process $G(s) = 1/(1+s)^5$ using Lyapunov based optimum stability. One controller (system responses in blue) has transfer function $C_1(s) = 0.475 + 0.1657/s + 0.3501s$ and the system has a single, real, rightmost eigenvalue at -0.48, the rest of the eigenvalues are complex and deeper into the left half plane. The second controller (system responses in red) has transfer function $C_2(s) = 0.515 + 0.169/s + 0.425s$. There is a pair of real, rightmost eigenvalues for this system at –0.382 and –0.389, the rest of the eigenvalues are complex and deeper into the left half plane. Controller $C_1(s)$ is used for comparison with other designs in Table 5.3. 70

Figure 5.22 Step reference responses for systems using five controllers from Table 5.3. The controllers designed using optimum stability result in the smoothest response, with no overshoot and the fastest settling time. 71
Figure 5.23 Step disturbance responses for systems using five controllers from Table 5.3. The controllers designed using optimum stability result in a smooth response with the greatest initial overshoot but the fastest settling time. 71
Figure 5.24 Impulse disturbance responses for systems using five of the controllers in Table 5.3. The controllers designed using optimum stability result in the smoothest response with greatest initial overshoot but with the least undershoot and the fastest settling time. 72
Figure 5.25 A second order controller and the double integrator process. 72
Figure 5.26 Step reference response, step disturbance response, and impulse disturbance response for the system in Figure 5.25 with a Ziegler Nichols controller given by equation (5.23) (in blue) and the controller given by equation (5.25) (in red) designed from optimum stability. 74
Figure 5.27 Step reference response, step disturbance response, and impulse disturbance response for the system in Figure 5.25 with a Ziegler Nichols controller given by equation (5.23) (in blue) and the controller given by equation (5.25) (in red) designed from optimum stability. 75
Figure 6.1 Open loop step response, or the "process reaction curve", for the FOLPD process $G(s) = e^{-6s}/\left(s+\frac{1}{4}\right)$. A step input is applied to the process at $t = 0$ and after a delay of 6 s the process begins to respond. The speed of response is dictated by the time constant of the process. After one time constant, which is 4 seconds, the response reaches $(1-e^{-1}) = 0.63$ of the final value. After about 30 s the process has settled down to a new operating point. Reaction curves of this shape are typical of many industrial processes [72] including processes as diverse as the air intake on a turbojet engine [73] and chemical mixing tanks or reaction vessels [73]. The problem of designing PI controllers for processes with this type of reaction curve has been under scrutiny for over sixty years [71]. 79
Figure 6.2 An illustrative example of a root locus diagram for a time-delay system. The system used has characteristic equation $p(s) = s + ke^{-s} = 0$ with k as parameter. There is a pole at $s = 0$ and a branch starts at this pole when $k = 0$. This branch meets another branch coming from $\sigma = -\infty$ to form a breakpoint. These two branches then cross the imaginary axis and approach $\sigma = \infty$. There is an infinite number of other branches that start at $\sigma = -\infty$ and approach $\sigma = \infty$ along asymptotes that are parallel to the real axis. Root loci for time-delay systems are symmetrical about the real axis but, unlike the delay-free case, the number of branches is infinite. The branches nearest to the real axis, called the primary branches, are the critical ones when considering stability since they are the first to cross the imaginary axis with subsequent crossings taking place at higher values of k 80
Figure 6.3 PI controller and a first-order lag plus time-delay process (FOLPD). 80
Figure 6.4 Primary branches of the root locus for $p(q) = q(q + L/\tau) + kL(q + FL)e^{-q} = 0$ with k as parameter, $L = 1$, $\tau = 5$, and $FL = 0.25$, which is just below the critical value. In addition to the branches that start at the poles at $q = 0$ and $q = -L/\tau$, and end at the zero at $q = -FL$, there is an infinite number of branches that start at $\sigma = -\infty$ and end at $\sigma = +\infty$. One of this infinite number of branches approaches the zero at $q = -FL$ and meets the branch going in the opposite direction forming the leftmost breakpoint. 82
Figure 6.5 Primary branches of the root locus for $p(q) = q(q + L/\tau) + kL(q + FL)e^{-q} = 0$ with k as parameter, $L = 1$, $\tau = 5$ and FL at the critical value $FL = 0.28845$. A single branch that originates at $\sigma = -\infty$ meets the two branches that originate at the poles at $q = 0$ and $q = -L/\tau$. Here three eigenvalues have coalesced to form a single point. The rightmost eigenvalue is as far to the left as possible at this point and the system is said to be optimally stable in the root locus sense. Optimum stability design gives nominal parameters that place the system's operating point at this triple eigenvalue. 82
Figure 6.6 Primary branches of the root locus for $p(q) = q(q + L/\tau) + kL(q + FL)e^{-q} = 0$ with $L = 1$, $\tau = 5$, $FL = 0.3$ and with k as parameter. Here FL is just above the critical value, the triple root evident in Figure 5 is gone, and the rightmost eigenvalue is no longer as far to the left as possible. The branch that originates at $\sigma = -\infty$ now terminates at the zero at $q = -FL$. The branches that start at the poles $q = 0$ and $q = -L/\tau$ simply approach $\sigma = +\infty$ along asymptotes.83
Figure 6.7 Step response and impulse response curves for various controller designs. Red curve is for minimum $J_{IT^2AE} = \int_0^\infty t^2 |e(t)| dt$ controller; green curve is for minimum

$J_{IES2_{\alpha-1}} = \int_0^\infty (\exp(2\alpha t)-1)\, e^2(t)dt$ controller, where $e(t)$ is the error signal and α is fixed at $\alpha = 1.5$ [76]; black curve is for the controller designed from optimum stability considerations using (6.15) and (6.16). (a) Impulse disturbance responses for $L = 0.5$. (b) Impulse disturbance responses for $L = 1.0$. (c) Step disturbance responses for $L = 2.0$. (d) Step reference responses for $L = 2.0$. We can see that the controllers designed from optimum stability result in systems with smoother response curves, less undershoot and overshoot, and settling times that are comparable the best controllers in [76]. 88

Figure 6.8 Stability region for the sample system consisting of the controller $C(s) = k_p + \frac{k_i}{s}$ and the process $G(s) = \frac{\frac{1}{4}}{s+\frac{1}{4}}e^{-s}$. The red region represents points that lead to system stability. Three possible controller design points are indicated. The centroid of the region, the point (3,1) used as an example design point in [79] and [44], and the optimum stability point. Performance measures show that the fact that the optimum stability point is nearest to the stability boundary is irrelevant. Indeed the centroid leads to the poorest performance even though it is furthest from the stability boundary. This poor performance is due to the fact when selecting a design point the designer must not only consider stability but also the degree of stability of the point. The system designed with parameter values chosen at the centroid is less stable (in the root locus sense) than one designed with parameter values at the optimum stability point. 89

Figure 6.9 Step response and impulse response curves for systems with various controller designs. The red curve is for the system with a controller that has parameter values selected at the centroid of the region of stability. The green curve is for the system with the example controller. The black curve is for the system with a controller designed from optimum stability. (a) Step reference responses. (b) Step disturbance responses. (c) Impulse disturbance responses. These diagrams illustrate the fact that, with the exception of the step disturbance response (b), optimum stability based design has similar settling times to the two other designs, and lead to smoother responses and less undershoot and overshoot. In the case of (b) the optimum stability design response is smooth but has a higher overshoot than the other designs. The initial oscillations in these plots are artefacts of the Pade approximation to the pure time delay. 91

Figure 6.10 Root locus for the time delay system with characteristic polynomial $p(s) = s + ke^{-s}$ from [75]. The equation for branches of this root locus was derived using the Bendrikov-Teodorchik equation and it is the same as the one derived from plotting rules in [75]. 93

Figure 6.11 Primary of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.15$ which is just below the critical value. 95

Figure 6.12 Primary branches of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.17157$ - which is the critical value. There is a triple eigenvalue at the breakpoint and the system is in a state of optimum stability at this point. 95

Figure 6.13 Primary branches of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.18$ which is just above the critical value. 96

Figure 6.14 Plot of the Lambert W function from [81] 97

Figure 7.1 Single-loop electric circuit, to illustrate maximum power transfer theorem 100

Figure 7.2 Electric circuit as a feedback loop 101

Figure 7.3 Single-loop feedback control system 101

Figure 7.4 Responses of third order system under PI and PID control 105

Figure 7.5 Open loop frequency response loci for the same systems as shown in Figure 7.4. 105

Figure 7.6 Root locus of equation (7.17) with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense. 107

Figure 7.7 Comparison of three PID designs for eighth-order process, step reference input 108

Figure 7.8 Comparison of three PID designs for eighth-order process, step disturbance input 109

Figure 8.1 Human body represented by a single-link inverted pendulum free to move only about the ankle. The centre of gravity (c.o.g.) is at a length h from the ankle. The body is maintained in an upright position by a torque about the ankle that opposes the torque due to the weight of the body. 111

Figure 8.2 Block diagram of the human balance control system. The controller consists of four feedback loops that model the semicircular canals, otolith, vision, and somatosensory systems. The transfer functions for these four subsystems are taken from [91] although the time delays have been modified. The control loops are combined into a single transfer function $f_6(s)$ that represents the muscle torque per unit moment of inertia about the ankles. (a,b,k_c,k_0) are the four parameters that must be selected. 112
Figure 8.3 Nyquist diagram for $p_b = p_2/(p_q + ap_1)$, $a = 5.0$ so $b = 1/0.08 = 12.5$ at optimum phase margin. 113
Figure 8.4 Stability region shown in green for p(s) with $k_0 = k_c = 0$. Optimum phase margin points from Table 8.1 are shown as black dots. 114
Figure 8.5 Stability region shown in green for p(s) with $a = b = 0$. Optimum phase margin points from Table 8.2 are shown as black dots. 115
Figure 8.6 Impulse response for the system in Figure 8.2 with $(a,b,k_c,k_0) = (5.56,10.0,8.5,4.44)$ - parameters selected by from Table 8.3. This is the least oscillatory impulse response using parameters in Table 8.3 116
Figure 8.7 Impulse responses for the system at the operating point $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ from [8]. This point was selected in [8] by recalling that human balance is governed by a compromise between optimum stability visual / somatosensory and optimum stability vestibular designs. When $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ the rightmost eigenvalues of the quiet standing polynomial is at –0.07 117
Figure 8.8 Impulse response for the system at the operating point $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$. This point was arrived at by using an optimization routine with starting point $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ [8]. When $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ the rightmost eigenvalues of the quiet standing polynomial is at –0.07. 118
Figure 8.9 Impulse responses for the operating point with the lowest ITSE from Table 8.4 - $(a,b,k_c,k_0) = (8.39,13.0,2.51,6.99)$ and the operating point $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ from section 8.3.3. The operating point form Table 8.4 has a lower peak impulse deflection than the operating point $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$. The operating point from Table 8.4 and $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ have comparable settling times. 120
Figure 8.10 Impulse response for the system at two operating points from Table 8.4 - $(a,b,k_c,k_0) = (8.39,13.0,2.51,6.99)$ and $(a,b,k_c,k_0) = (4.73,10.04,1.58,4.17)$ 120
Figure 8.11 Impulse response for the system at two operating points from Table 8.4 - $(a,b,k_c,k_0) = (6.49,13.30,3.08,9.47)$ and $(a,b,k_c,k_0) = (4.29,10.82,2.74,7.15)$ 121
Figure 8.12 Impulse response for the system at the operating point $(a,b,k_c,k_0) = (11.43,9.78,3.3,12.65)$ from Table 8.4. The oscillatory behavior is due to a complex pair of eigenvalues ($-0.18 \pm j0.66$) near to the real eigenvalue at –0.11. 121
Figure 9.1 Routh array for a polynomial $p(s+\sigma) = 0$ of order n (even) with a triple eigenvalue at $s = -\sigma$. Those entries in the upper right corner that are zero propagate down the array causing the last three entries in each column to go to zero. 132
Figure 9.2 Routh array for a polynomial $p(s+\sigma) = 0$ of order n (odd) with a triple eigenvalue at $s = -\sigma$. Those entries in the upper right corner that are zero propagate down the array causing the last three entries in each column to go to zero. 132
Figure 9.3 Stability region in the (k,f) for the polynomial $p(s) = s(s^2+1) + k(s^2+f)$. The optimum stability point $(\sqrt{3},1/9)$ derived from root locus is shown. 134
Figure C.1 Block diagram for a general feedback control system. 163
Figure E.1 Polynomials from [130] that minimise the ISE for "zero steady state step error systems". 172
Figure E.2 Polynomials from Figure E.1 with normalized frequency. 172
Figure E.3 Routh array that is synthesized by first choosing a column of 1's as the leftmost column and then calculating the rest of the array in the usual way. 172
Figure E.4 Zero steady state unit step responses for the polynomials synthesized by Parks [130] using the Routh array as illustrated in Figure E.3. 173
Figure E.5 Coefficients of the polynomials in Figure E.2 arranged as a number triangle. This figure illustrates the fact that the coefficients sum to form Fibonacci numbers. This number triangle has been studied elsewhere [181] and it was generated using the Routh array by Parks [130], albeit without the intention of generating Fibonacci numbers. 173

Figure E.6 Routh array that is synthesized by first choosing a column of 1's ending in a "2" as the leftmost column and then calculating the rest of the array in the usual way. 174
Figure E.7 Zero steady state unit step responses for the polynomials synthesized using the Routh array as illustrated in Figure E.6. 174
Figure E.8 A number triangle generated from the coefficients of polynomials that were synthesized using the Routh array with the leftmost column consisting of a column of 1's ending with a 2 as illustrated in Figure E.6. This figure illustrates that the coefficients of these polynomials sum to form Lucas numbers. This number triangle is already known [181] but has not been calculated before using the Routh array. 175
Figure E.9 Routh array that is synthesized by first choosing a left hand column consisting of binomial coefficients [182] 175
Figure E.10 Zero steady state unit step responses for the polynomials synthesized by dePaor [182] using the Routh Array as illustrated in Figure E.9. 176

List of Tables

Table 2.1 Illustration of the scheme used to show that the rule for generating row $i+1$ in Figure 2.9 from row i is valid for all i 30

Table 5.1 Comparison of the performance of four PI controllers designed for the process $G(s) = 1/(s+1)^4$ .58

Table 5.2 Comparison of four PID controllers for the process $G(s) = 1/(s+1)^4$. The PID controller designed using root locus based optimum stability shows considerably enhanced robustness margins. 67

Table 5.3 Comparison of six PID controllers designed for the process $G(s) = 1/(s+1)^5$. The controller designed using root locus based optimum stability has the best gain margin and excellent phase margin.....70

Table 6.1 PI controller parameters for the best eight controllers from [76] and from root locus based optimum stability. PI controller parameters are selected either by minimizing the performance integrals $J_{IT^2AE} = \int_0^\infty t^2 |e(t)| dt$ and $J_{IES2_{\alpha-1}} = \int_0^\infty (\exp(2\alpha t) - 1)\, e^2(t) dt$, where $e(t)$ is the error signal and α is fixed at $\alpha = 1.5$ [76] , or using (6.15) and (6.16), which are derived from optimum stability 86

Table 6.2 Comparison between the robustness measures of PI controllers designed by three different methods. 87

Table 6.3 Controller parameters, GM (gain margin), PM (phase margin), and DM (delay margin) for the example controller from [79] and [44], the controller designed with parameter values at the centroid of the region of stability , and the controller design based on optimum stability. The enhanced robustness measures show the effect of using the parameters derived from optimum stability. 90

Table 8.1 Optimum phase margin points - Visual / Somatosensory System only active. 114

Table 8.2 Optimum phase margin points - Vestibular System only active. 115

Table 8.3 Nineteen combinations of parameters (a, b, k_c, k_0) from Table 8.1 and Table 8.2 that lead to stability. The values for (a, b) were obtained by setting $k_c = k_0 = 0$ and finding the values of (a, b) that give optimum phase margin. The values for (k_c, k_0) were obtained by setting $a = b = 0$ and finding the values of (k_c, k_0) that give optimum phase margin. These combinations of points are very artificial; there is no relationship between the pairs of points other than they lead to characteristic polynomials with eigenvalues in the left half plane. Nature would select the operating point as a set of four rather than in pairs. The operating points in this Table lead to very oscillatory impulse responses. 116

Table 8.4 A selection of operating points that lead to Optimum stability with both the Visual / Somatosensory System and the Vestibular System active. These points were derived from the Lyapunov matrix equation and each set of parameters place the rightmost eigenvalue at –0.11. The model predicts that many people, each with different values for (a, b, k_c, k_0) , will all have the same "sense of balance". This is indeed what one experiences in everyday life. The ITSE parameter is included to allow one to select a set of parameters. 119

Table 8.5 The operating point with the lowest ITSE from Table 8.4 - $(a, b, k_c, k_0) = (8.39, 13.0, 2.51, 6.99)$ is compared with the operating point $(a, b, k_c, k_0) = (5.0, 9.5, 4.0, 2.0)$ from section 8.3.3. The operating point form Table 8.4 has an eigenvalue further to the left, a lower ITSE figure and a lower peak impulse deflection than the operating point $(a, b, k_c, k_0) = (5.0, 9.5, 4.0, 2.0)$. The operating point from Table 8.4 and $(a, b, k_c, k_0) = (5.0, 9.5, 4.0, 2.0)$ have comparable settling times. 119

Table D.1 Ziegler-Nichols Tuning rules [50] for P, PI, and PID controllers based on the process reaction curve of the process. 169

Table D.2 Ziegler-Nichols tuning rules [50] for P, PI, and PID controllers based on ultimate sensitivity. .169

Chapter 1: Control System design methodologies

1.1 Introduction

The purpose of this chapter is to give a description of the controller design methodologies developed in this thesis. Figure 1.1 shows the controller design methods developed, the processes to which they were applied, and where in the thesis each method is discussed.

1.2 Optimum stability from root locus

The root locus based controller design methodology described in this thesis has analytic and geometric manifestations. The root locus diagram provides the arena for the geometric manifestation. Specifically, the problem is to find those values for the system parameters that place the rightmost eigenvalue as far to the left as possible in the complex plane. These parameter values are then selected as the system's nominal design parameters. Putting the rightmost eigenvalue is as far to the left as possible is a balancing act that is performed within the context of the system dynamics. When a system has one or two design parameters this balancing act is done with the aid of the root locus diagram. A Lyapunov matrix equation based optimization procedure, described in section 1.4 is used for systems with more than two design parameters. By placing the rightmost eigenvalue as far to the left as possible we are minimizing the largest time constant $1/|\delta|$ where $|\delta|$, called the degree of stability [1] of the system, is the distance from the rightmost eigenvalue to the imaginary axis.

The merits and demerits of this approach will be discussed and a case is made for adopting the root locus based design methodology in many cases. For example, concentration on optimum stability makes for a trade-off between certain performance measures e.g. fast transient response and good disturbance rejection. This trade-off is not unreasonable for process control applications in particular.

In at least one interesting general case i.e. a second order system comprising integral control of a first order lag in section 4.1.2, the root locus based controller design methodology described in this thesis has an analytic manifestation also. Specifically, the analytic problem is to find the system parameters that minimise the performance integral $\int_0^\infty \exp(\alpha t)\left[e^2(t)+\frac{1}{\omega_n^2}\left[\frac{de(t)}{dt}\right]^2\right]dt$, where $e(t)$ is the system's error signal. A method for calculating this integral in developed in 2.4.2.

In the examples considered, when a system is in a state of optimum stability there are invariably several real eigenvalues at the same rightmost point. This leads to a very convenient method for calculating the system parameters for optimum stability. For example, if there is a triple real eigenvalue at the rightmost point $s=-\sigma$ then the characteristic polynomial has the factorization $p(s)=f(s)(s+\sigma)^3=0$ and we can also write $p'(s)=0$ and $p''(s)=0$. This observation is used frequently to calculate system parameters.

Several authors have used root locus techniques to choose parameter values for controllers. Wang et al. [2] and Mann et al. [3] develop tuning methods for PID controllers that place the closed-loop poles at a breakpoint in the root locus. Normey-Rico et al. [4] present a solution to the mobile robot path tracking problem by selecting parameter values that lead to a double real pole on the root locus. Basilio and Matos [5] tune PI and PID controllers by selecting parameter values that lead to a critically damped system. In doing this they sometimes places the rightmost eigenvalue as far to the left as possible. In addition nominal parameter values that place the rightmost eigenvalue as far to the left as possible are selected in [6-11].

1.3 Optimum stability using performance integrals

A system is said to be robust if it behaves in an acceptable manner when subjected to changes in the process or the environment. Dorf [12, 13] points out that classical techniques may be used to design controllers that are robust in the sense that they minimise an interal performance measure. In this thesis integral performance measures such as $J = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt$ are used. Power [14] considered a similar performance integral with no time weighting: $J = \int_0^\infty \left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt$ as did Grayson [15] and Gibson [16].

Gibson [16] points out that performance integrals that minimise the square of the error alone can result in systems with insufficient damping. He then points out, however, that if the derivatives of the error are included then this stricture does not apply. Gibson [16] also remarks that if a system has a persistent, residual, acceptable error then performance integrals with polynomial time weighting such as $\int_0^\infty t^n e^2(t)dt$ or $\int_0^\infty t|e(t)|dt$ go to infinity. This criticism does not apply to performance integrals with exponential time weighting as such integrals decay to zero as $t \to \infty$.

Grayson [15] remarks that the value of a performance integral depends on the initial conditions and he describes this as a disadvantage. However, this dependence on initial conditions actually allows a performance integral to reflect the system performance for different disturbance types e.g. impulsive, step, sloping etc. For this reason the dependence on initial conditions is actually a strong point of performance integral and affords an ability to distinguish, for example, when a system has a good step input response but a poor impulse disturbance response.

1.4 Optimum stability using the Lyapunov matrix equation

Optimum stability using the Lyapunov matrix equation is used frequently in this thesis. The approach taken was motivated by the following corollary in Kalman and Bertram [17]:

Corollary 3.2: For the continuous time, free, linear, stationary dynamic system $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ the real parts of the eigenvalues of a constant matrix $\mathbf{A}$ are $< \sigma$ if and only if given any symmetric positive definite matrix $\mathbf{Q}$ there exists a symmetric positive definite matrix $\mathbf{L}$ which is the unique solution of the set of $n(n+1)/2$ linear equations: $-2\sigma\mathbf{I}\mathbf{L} + \mathbf{A}^T\mathbf{L} + \mathbf{L}\mathbf{A} = -\mathbf{Q}$

The proof of this Corollary is obtained very easily by substituting $(\mathbf{A}-\sigma\mathbf{I})$ into the Lyapunov Matrix Equation $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ and rearranging the terms. This gives

$$(\mathbf{A}-\sigma\mathbf{I})^T\mathbf{L}+\mathbf{L}(\mathbf{A}-\sigma\mathbf{I})=\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}-2\sigma\mathbf{I}\,\mathbf{L}=-\mathbf{Q}$$

The significance of this Corollary here is that it can be used as part of an eigenvalue location optimization procedure as described next.

As σ gets more and more negative, the eigenvalues of $(\mathbf{A}-\sigma\mathbf{I})$ shift to the right by an amount equal to σ. (Also, as σ gets more and more positive, the eigenvalues of $(\mathbf{A}-\sigma\mathbf{I})$ shift to the left by an amount equal to σ). So by making σ more and more negative and at the same time solving $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}-2\sigma\mathbf{I}\,\mathbf{L}=-\mathbf{Q}$ for $\mathbf{L}$ and checking to see if $\mathbf{L}$ is positive definite, we can establish that σ that makes $\mathbf{L}$ no longer positive definite. We now know where the rightmost eigenvalue of the characteristic equation corresponding to $\mathbf{A}$ is.

This procedure for finding the rightmost eigenvalue may be described as follows:

take $\mathbf{Q}$ to be the Identity matrix, $\mathbf{I}$

take σ to be a small, negative number

select an initial set of parameters - for example, (a,b,k_c,k_0)

use the characteristic equation to create the companion form of the matrix $\mathbf{A}$

solve $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}-2\sigma\mathbf{I}\,\mathbf{L}=-\mathbf{Q}$ for $\mathbf{L}$

check if $\mathbf{L}$ is positive definite

if $\mathbf{L}$ is positive definite then decrease σ a little

solve $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}-2\sigma\mathbf{I}\,\mathbf{L}=-\mathbf{Q}$ again for $\mathbf{L}$

check if $\mathbf{L}$ is positive definite

continue decreasing σ until $\mathbf{L}$ is no longer positive definite

record the final value of σ at which $\mathbf{L}$ is positive definite as this is the location of the rightmost eigenvalue of $\mathbf{A}$ for the selected parameters (a,b,k_c,k_0).

This procedure is effectively a function that takes (a,b,k_c,k_0) as an input and gives σ as an output. If this procedure is treated as a function then an optimization algorithm can be used to find that input (a,b,k_c,k_0) that makes the output σ as negative as possible. This optimization procedure was implemented by using an add-on for Mathematica called Global Optimization [18]. Other packages failed to find a global optimum or exceeded the memory resources of the computer. The value of the Lyapunov matrix equation method is that it works for a system when several parameters have to be chosen simultaneously.

1.5 Optimum stability from the Routh-Hurwitz Criterion

The Routh array may be used to find domains of stability simply by going through the usual calculations and requiring that no entry be less than zero [19]. When these domains are drawn in parameter space, the centroid is chosen as the design point. This procedure has been used by many designers, for example [8, 9], although, as illustrated in section 8.4.2, it must be used with caution. This caution is necessary, as a flat parameter plane does not contain the complete information about the dynamics of the system. A height parameter is also necessary to obtain a complete picture. A useful measure for this height parameter is the distance from the imaginary axis to the rightmost eigenvalue, also called the margin of stability [1]. The optimum operating point is now a point on the top of a hill in the parameter-plane – margin-of-stability space. As this space has a characteristic topography, the best operating point may not be the centroid of the parameter plane as viewed form above.

When the Routh array for a characteristic polynomial is calculated, the columns contain functions of the system parameters. An interesting property of the Routh array, described in chapter 9, is that some of the entries in these columns go to zero when the system parameters have their optimum stability values i.e. the same values that are optimal according to the root locus. This fact is exploited in chapter 9 and column entries are used to find the parameters that lead to optimum stability.

1.6 Optimum stability from the Nyquist diagram - optimizing gain and phase margins

It is frequently possible to draw a tangent to the Nyquist diagram from the origin and then draw an arc from the point of tangency to the real axis. The point on the real axis obtained in this way corresponds to an optimum phase margin point [20]. Optimum phase margin design was used, for example, by Power in the study of inertially damped instrument servomechanisms [21]. Ho et al. [22] give tuning rules for PID controllers that simultaneously optimise Gain and Phase Margin. Phase margin optimization is applied in section 8.3.

The range of parameter values that results in stable behavior may be gleaned from the Nyquist diagram. Simultaneous minimisation of sensitivity and the optimisation of Gain Margin is discussed in [23].

It is sometimes mentioned that Gain and Phase margins are not reliable indicators of robustness [24, 25]. However, these reservations seem only to apply to contrived systems such as the following controller C(s) and process G(s) from [25]:

$$C(s)=\frac{a-s}{as-1},\quad (a>1)\ \text{and}\ \ G(s)=\frac{(s+3.3)(s+0.55)(1.7s^2+1.5s+1)}{(3.3s+1)(0.55s+1)(s^2+1.5s+1.7)}.$$

Conclusions drawn about robustness from Gain and Phase Margin values depend on the shape of the Nyquist plot and some discretion needs to be exercised.

1.7 The maximum power transfer theorem - another setting for optimality

Some years ago [26] I proved that nonlinear resistive loading of a series-wound, self-excited DC generator driven by a wind turbine, in such a way as to optimise power transfer from wind to electrical load, resulted in a very well damped dynamic response to varying wind speeds. This prompted me to explore whether there might be some other favorable consequences for control lying unexploited in results on optimum power transfer. Chapter 7 presents a resulting new idea for tuning PI and PID controllers for a class of asymptotically stable processes, discovered by viewing the Maximum Power Transfer Theorem of linear AC circuit theory as a relation in a single loop, negative feedback system. This approach brings together ideas from the cognate subjects of Circuit Theory and Control Theory, continuing an old but often overlooked tradition (Truxal, [27]). It is a contribution to the many methods already available for designing PI and PID controllers – see for example[28] and [29].

1.8 Thesis organization

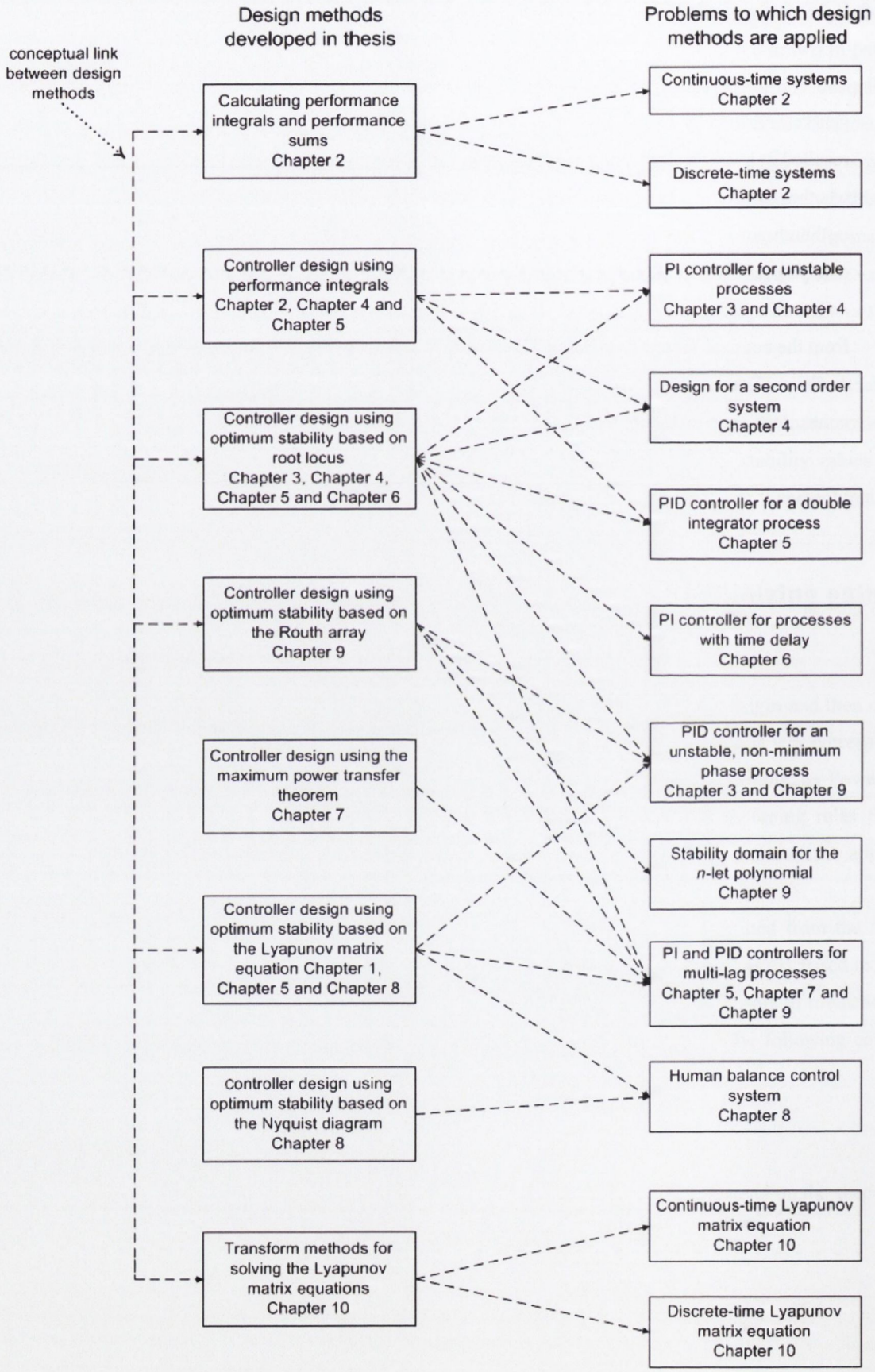


Figure 1.1 Organization of this thesis

Chapter 2 describes procedures for calculating performance integrals and performance sums within a unified framework provided by the Kronecker product and MacFarlane's procedure. MacFarlane's procedure is simplified and extended to discrete time systems.

Chapter 3 describes a root-locus-based optimum stability approach to design a PID controller for a second order, unstable, non-minimum phase process. The controller designed using root locus based optimum stability is of lower order than a H_∞ controller and it results in a system with better performance than the system using the H_∞ controller.

In chapter 4 two controller design methodologies are described – one that uses root locus based optimum stability and the other based on minimising an exponentially weighted performance integral.

Chapter 5 shows that systems with controllers that were designed for multi-lag processes using either Lyapunov based optimum stability or root locus based optimum stability exhibit greater robustness margins, and smoother response characteristics than systems with controllers designed using currently available methods. General formulas were given for root-locus-based optimum stability design for a PI controller, and a restricted class of PID controller, for process $G(s) = k_2 /(s+b)^m$.

Chapter 6 presents a new procedure for the design of PI controllers for an integrator with time delay and for general FOLPD process. By calculating gain margins, phase margins, delay margins, and plotting various response curves we see that the controllers that were designed using optimum stability are, by these standard measures, superior to controllers designed using currently available techniques.

Chapter 7 describes a new idea for tuning PI and PID controllers based on analogy with the maximum power transfer theorem from linear AC circuit theory. The approach is one that specifies the phase margin and the frequency at which it is effective. Explicit formulas are derived for calculating the max'-power-transfer-based PI controller parameters for the process $G(s) = k_2 /(s+b)^m$.

A model of the human balance control system is studied in chapter 8. Four parameters are selected – one for each major control loop. Nyquist analysis is used to select pairs of parameters that lead to Optimum Phase Margin but this is a graphical procedure and therefore approximate. The Lyapunov matrix equation was used to select all four parameters simultaneously – these parameters give optimum eigenvalue location.

A new procedure for designing controllers using the Routh array is described in chapter 9. This method is shown to be equivalent to the root-locus-based optimum stability method and leads to controllers that are identical to the root locus based controllers.

Chapter 10 describes new methods for solving the continuous-time and the discrete-time Lyapunov matrix equations based on the Laplace transform and the Z transform respectively.

1.9 Publications

B. Cogan, B. and A. M. dePaor, "Optimum stability and minimum complexity as desiderata in feedback control system design" in IFAC Conference on Control Systems Design, Bratislava, Slovakia, 2000.

A. M. dePaor and B. Cogan, "Making a connection: PI and PID Controller Tuning by analogy with the Maximum Power Transfer Theorem of Circuit Theory", Journal of Electrical Engineering, vol. 55, n.9-10, pp. 277-280, 2004.

Chapter 2: A unified treatment of control system performance measures for continuous and discrete time system

2.1 Introduction

The purpose of this chapter is to investigate methods for calculating performance measures. A design methodology studied in this thesis requires that the rightmost eigenvalue be as deep into the left-half plane as possible when the parameters have their nominal values. This design point often corresponds to a triple or quadruple breakpoint in the root locus. An objection that might be raised against this design strategy is that the sensitivity of the eigenvalues is at a maximum when the root locus crosses the real axis. We are then designing systems whose eigenvalues are very sensitive at the nominal parameter values and this may degrade the system performance. One way to address this concern is to use a performance measure to give a quantitative indication of the performance of the system at the design point. We can then calculate the performance measure at the design point and see if this eigenvalue sensitivity is manifested as degradation in performance of the system. Indeed, such a performance measure is defined in section 1.3. Analytically, the performance measure is found to have a unique minimum. When plotted against system parameter the curve is seen to have a broad, flat, shape around the design point. This indicates very acceptable system behavior for these parameter values despite the eigenvalue sensitivity.

The treatment of control system performance measures is unified in the sense that MacFarlane's original procedure for the calculation performance integrals [30] is used to produce new and simple derivations of well known and important results. MacFarlane's procedure, rooted in Lyapunov stability theory, is seen here to provide a general framework for the design of control systems and for the study of control system performance. Since optimality principles, and specifically optimal stability, are central issues in this thesis a thorough study of this procedure and these novel extensions is essential.

MacFarlane derived a general method for the calculation of performance integrals for continuous time systems [30] and then recast this procedure in Kronecker product form [31]. MacFarlane's original procedure is used in section 2.2 to derive his own Kronecker product equation. Symmetries in the solution are exploited in a new way in section 2.2.2 in order to minimise the number of calculations required. This method is simpler than those proposed by MacFarlane [30] and by Chen and Shieh [32]. A new method for calculating performance integrals with exponential time weighting is described in section 2.4.

Barnett derived a general formula for calculating performance sums for discrete time control systems [33]. Jury derived a recurrence relation for calculating the same performance sums [34]. In section 2.5 Jury's recurrence relation is shown to be a manifestation, in discrete time systems, of MacFarlane's original continuous time procedure. Barnett's formula is derived again in a new and simple way in section 2.6. In addition, an entirely new procedure for calculating these performance sums is also given in section 2.6. A new role for Stirling numbers of the Second Kind in these calculations is described in section 2.7. A new method for calculating performance sums with exponential time weighting is described in section 2.10.

2.2 Kronecker product method for calculating performance integrals

The following series of Lyapunov equations arises when applying MacFarlane's procedure – see [30] and Appendix A.4 of this thesis.

$$\mathbf{A}^T\mathbf{L}_1+\mathbf{L}_1\mathbf{A}=-\mathbf{Q},\ \mathbf{A}^T\mathbf{L}_2+\mathbf{L}_2\mathbf{A}=-\mathbf{L}_1,\ \mathbf{A}^T\mathbf{L}_3+\mathbf{L}_3\mathbf{A}=-\mathbf{L}_2,\ \mathbf{A}^T\mathbf{L}_4+\mathbf{L}_4\mathbf{A}=-\mathbf{L}_3 \text{ etc. etc.}$$

In order to develop another approach to these calculations we can take the fourth formula: $-\mathbf{L}_3=\mathbf{A}^T\mathbf{L}_4+\mathbf{L}_4\mathbf{A}$ and substitute this into $-\mathbf{L}_2=\mathbf{A}^T\mathbf{L}_3+\mathbf{L}_3\mathbf{A}$ to express $\mathbf{L}_2$ in terms of $\mathbf{L}_4$. We now can use this expression to express $\mathbf{L}_1$ in terms of $\mathbf{L}_4$. And finally use this expression to express $\mathbf{Q}$ in terms of $\mathbf{L}_4$. Examples of the equations thus obtained are:

$\mathbf{A}^T\mathbf{L}_1+\mathbf{L}_1\mathbf{A}=-\mathbf{Q}$

$-(\mathbf{A}^T)^2\mathbf{L}_2-2\mathbf{A}^T\mathbf{L}_2\mathbf{A}-\mathbf{L}_2\mathbf{A}^2=-\mathbf{Q}$

$(\mathbf{A}^T)^3\mathbf{L}_3+3(\mathbf{A}^T)^2\mathbf{L}_3\mathbf{A}+3\mathbf{A}^T\mathbf{L}_3\mathbf{A}^2+\mathbf{L}_3\mathbf{A}^3=-\mathbf{Q}$

$-(\mathbf{A}^T)^4\mathbf{L}_4-4(\mathbf{A}^T)^3\mathbf{L}_4\mathbf{A}-6(\mathbf{A}^T)^2\mathbf{L}_4\mathbf{A}^2-4(\mathbf{A}^T)\mathbf{L}_4\mathbf{A}^3-\mathbf{L}_4\mathbf{A}^4=-\mathbf{Q}$ and so on...

Following this procedure a general expression relating $\mathbf{Q}$ to $\mathbf{L}_n$ is found to be:

$$\sum_{j=0}^{n}(-1)^n\binom{n}{j}\left(\mathbf{A}^T\right)^{n-j}\mathbf{L}_n\mathbf{A}^j=\mathbf{Q} \tag{2.1}$$

where $\binom{n}{j}$ are binomial coefficients.

For example, to calculate $\int_0^\infty t^3\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ we could solve (2.1) for $\mathbf{L}_4$ and then we could calculate: $\int_0^\infty t^3\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt=(-1)^4 3!\mathbf{x}^T(0)\mathbf{L}_4\mathbf{x}(0)$.

Mansour et al. [35] derive $\mathbf{Q}=\sum_{j=0}^{n}(-1)^n\binom{n}{j}\left(\mathbf{A}^T\right)^{n-j}\mathbf{L}_n\mathbf{A}^j$ i.e. equation (2.1) of this thesis, and states that it may be solved for $\mathbf{L}_{n+1}$ but does not say how this could be done.

Equation (2.1) is a matrix equation that is linear in the unknown matrix $\mathbf{L}_n$. A typical term in this equation consists of the product of the matrix $\mathbf{L}_n$ flanked by $\mathbf{A}$ and $\mathbf{A}^T$ raised to some power. Such triple products of matrices may be rewritten in a convenient form using the following property of the Kronecker product [36]:

$$\mathbf{ALB}=\mathbf{Y} \text{ then: } (\mathbf{B}^T\otimes\mathbf{A})\vec{\mathbf{L}}=\vec{\mathbf{Y}}$$

where $\mathbf{A}\otimes\mathbf{B}$ represents the Kronecker product of the matrices $\mathbf{A}$ and $\mathbf{B}$ and $\vec{\mathbf{L}}$ a column vector that is constructed from $\mathbf{L}$ by turning the rows of $\mathbf{L}$ into columns and stacking them on top of each other - similarly for $\vec{\mathbf{Y}}$.

For example, using Kronecker products, the solution to the continuous time Lyapunov Matrix Equation $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ is obtained by solving $(\mathbf{A}^T\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{A}^T)\vec{\mathbf{L}}=-\vec{\mathbf{Q}}$ for the entries in the vector $\vec{\mathbf{L}}$ and then constructing the matrix $\mathbf{L}$. Also, the solution to the discrete time Lyapunov Matrix Equation $\mathbf{A}^T\mathbf{L}\mathbf{A}-\mathbf{L}=-\mathbf{Q}$ is obtained from $(\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I})\vec{\mathbf{L}}=-\vec{\mathbf{Q}}$ in the same way.

In order to derive an expression for time weighted performance integrals in a new way we can use the Kronecker product to rewrite equation (2.1) as:

$$(-1)^n \left\{ \sum_{j=0}^{n} \binom{n}{j} \left(\mathbf{A}^j \otimes \mathbf{A}^{n-j} \right)^T \right\} \vec{\mathbf{L}}_n = \vec{\mathbf{Q}} .$$

A further simplification is possible when the following equation is taken into account:

$$\sum_{j=0}^{n} \binom{n}{j} \left(\mathbf{A}^j \otimes \mathbf{A}^{n-j} \right)^T = \left(\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T \right)^n$$

So finally we can write:

$$(-1)^n \left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T \right]^n \vec{\mathbf{L}}_n = \vec{\mathbf{Q}} \tag{2.2}$$

Equation (2.2) is MacFarlane's Kronecker product formula [31]. The derivation given here is simpler than MacFarlane's derivation and is rooted in his own original method [30]. To apply this formula to calculate $J_n = \int_0^\infty t^n \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt$ we solve $(-1)^{n+1} \left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T \right]^{n+1} \vec{\mathbf{L}} = \vec{\mathbf{Q}}$ for the entries in the matrix $\mathbf{L}$ and then calculate $J_n = (-1)^{n+1} n! \mathbf{x}^T(0) \mathbf{L} \mathbf{x}(0)$.

The original imbedding of the Lyapunov equations into each other is now manifested in terms of Kronecker products like: $\left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T \right]^n$ i.e. the solution to the Lyapunov equation $\mathbf{A}^T \mathbf{L} + \mathbf{L} \mathbf{A} = -\mathbf{Q}$ raised to the power of n.

2.2.1 Condition for the existence of solutions for equation (2.2)

One may write equation (2.2) as $\vec{\mathbf{L}}_n = (-1)^{-n} [\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T]^{-n} \vec{\mathbf{Q}}$ only if $[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T]$ is invertable. The condition for the invertability of $[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T]$ may be derived as follows.

Let $\mathbf{v}_i \otimes \mathbf{v}_j$ be an eigenvector of $[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T]$ where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the eigenvectors of $\mathbf{A}^T$ belonging to the eigenvalues λ_i and λ_j respectively [36].

$$[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T][\mathbf{v}_i \otimes \mathbf{v}_j] = [\mathbf{A}^T \otimes \mathbf{I}][\mathbf{v}_i \otimes \mathbf{v}_j] + [\mathbf{I} \otimes \mathbf{A}^T][\mathbf{v}_i \otimes \mathbf{v}_j]$$

$$= \mathbf{A}^T \mathbf{v}_i \otimes \mathbf{I} \mathbf{v}_j + \mathbf{I} \mathbf{v}_i \otimes \mathbf{A}^T \mathbf{v}_j = \lambda_i \mathbf{v}_i \otimes \mathbf{v}_j + \lambda_j \mathbf{v}_i \otimes \mathbf{v}_j = (\lambda_i + \lambda_j)[\mathbf{v}_i \otimes \mathbf{v}_j] .$$

So $[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T]$ is invertable if and only if all possible combinations of $(\lambda_i + \lambda_j) \neq 0$. This requirement on $\mathbf{A}$ that ensures $\mathbf{A}^T \mathbf{L} + \mathbf{L} \mathbf{A} = -\mathbf{Q}$ has a unique solution is a well-known result that was derived originally by Lyapunov [37]. One implication of this result is that if $\mathbf{A}$ is Hurwitz there will always be a solution to $\mathbf{A}^T \mathbf{L} + \mathbf{L} \mathbf{A} = -\mathbf{Q}$ as $(\lambda_i + \lambda_j)$ can never be equal to zero in that case.

2.2.2 A new method for reducing the number of equations to be solved to a minimum

MacFarlane [30] gives a solution to the continuous time Lyapunov matrix equation $\mathbf{A}^T \mathbf{L} + \mathbf{L} \mathbf{A} = -\mathbf{Q}$ as $\mathbf{l} = \mathbf{B}^{-1} \mathbf{q}$ where the matrix $\mathbf{B}$, an $\frac{n}{2}(n+1) \times \frac{n}{2}(n+1)$ matrix, is formed from the matrix $\mathbf{A}$ by following an algorithm. This algorithm exploits the symmetry of $\mathbf{L}$ in order to reduce the number of equations to be solved to a minimum. He then states that in general $\mathbf{l}_n = \mathbf{B}^{-n} \mathbf{q}$ where $\mathbf{l}_n$ is the contracted vector solution to $\mathbf{A}^T \mathbf{L}_n + \mathbf{L}_n \mathbf{A} = -\mathbf{L}_{n-1}$. Performance integrals can be calculated using this $\mathbf{L}_n$ without recursively solving a series of Lyapunov equations. MacFarlane's algorithm was simplified by Chen and Shieh [32]. Another algorithm that exploits the symmetry of $\mathbf{L}$ to reduce the number of equations to be

solved to a minimum is given next. This new algorithm is simpler to implement than those proposed by MacFarlane [30] and by Chen and Shieh [32].

Solving MacFarlane's Kronecker product equation $(-1)^n\left[\mathbf{A}^T\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{A}^T\right]^n\vec{\mathbf{L}}=\vec{\mathbf{Q}}$ for $\mathbf{L}$ requires solving n^2 equations in n^2 unknowns. However, by exploiting the symmetry of $\mathbf{L}$ the number of equations to be solved may be reduced from n^2 to $\frac{n}{2}(n+1)$. If $\vec{\mathbf{L}}$ is written with only those elements along and above the main diagonal we get a contracted form of $\vec{\mathbf{L}}$ that we shall call $\vec{\mathbf{L}}_c$. $\vec{\mathbf{L}}_c$ is a column vector with $\frac{n}{2}(n+1)$ elements. With $n=4$ $\vec{\mathbf{L}}_c$ is: $\begin{pmatrix} l_{11}\\ l_{12}\\ l_{13}\\ l_{14}\\ \cdots\\ l_{22}\\ l_{23}\\ l_{24}\\ \cdots\\ l_{33}\\ l_{34}\\ \cdots\\ l_{44}\end{pmatrix}$. We can write $\vec{\mathbf{L}}_c$ in terms of $\vec{\mathbf{L}}$ as follows: $\vec{\mathbf{L}}=\mathbf{E}\vec{\mathbf{L}}_c$.

Here $\mathbf{E}$ is a 4^2 by $\frac{4}{2}(4+1)$ matrix that expands $\vec{\mathbf{L}}_c$ into $\vec{\mathbf{L}}$. The equation $\vec{\mathbf{L}}=\mathbf{E}\vec{\mathbf{L}}_c$ for the case where $n=4$ is:

$$\begin{pmatrix} l_{11}\\ l_{12}\\ l_{13}\\ l_{14}\\ \cdots\\ l_{21}\\ l_{22}\\ l_{23}\\ l_{24}\\ \cdots\\ l_{31}\\ l_{32}\\ l_{33}\\ l_{34}\\ \cdots\\ l_{41}\\ l_{42}\\ l_{43}\\ l_{44}\end{pmatrix}=\begin{pmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1\end{pmatrix}\begin{pmatrix} l_{11}\\ l_{12}\\ l_{13}\\ l_{14}\\ \cdots\\ l_{22}\\ l_{23}\\ l_{24}\\ \cdots\\ l_{33}\\ l_{34}\\ \cdots\\ l_{44}\end{pmatrix}$$

A general procedure for constructing the matrix E is given in section 2.2.3.

Equation (2.2) becomes $(-1)^n\left[\mathbf{A}^T\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{A}^T\right]^n\mathbf{E}\vec{\mathbf{L}}_c=\vec{\mathbf{Q}}$. We can now contract $\vec{\mathbf{Q}}$ by multiplying both sides of this equation by $\mathbf{E}^T$ to give:

$$(-1)^n\,\mathbf{E}^T\left[\mathbf{A}^T\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{A}^T\right]^n\mathbf{E}\,\vec{\mathbf{L}}=\mathbf{E}^T\vec{\mathbf{Q}} \tag{2.3}$$

However, $\mathbf{E}^T\vec{\mathbf{Q}}$ is not quite $\vec{\mathbf{Q}}_c$. For example, if $n=4$, $\mathbf{E}^T\vec{\mathbf{Q}}$ is given by:

$$\mathbf{E}^T\vec{\mathbf{Q}} = \begin{pmatrix} q_{11} \\ 2q_{12} \\ 2q_{13} \\ 2q_{14} \\ \ddot{q}_{22} \\ 2q_{23} \\ 2q_{24} \\ \ddot{q}_{33} \\ 2q_{34} \\ \ddot{q}_{44} \end{pmatrix}.$$

It would of course be possible to solve this minimal set of equations keeping in mind the presence of the factors of 2 on some of the elements of $\mathbf{E}^T\vec{\mathbf{Q}}$. However, it is preferable to correct this and to do so it is necessary to premultiply both sides equation (2.3) by a $\frac{n}{2}(n+1)$ by $\frac{n}{2}(n+1)$ matrix $\mathbf{D}$ that has the following structure in the case where $n=4$:

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

A general procedure for constructing the matrix $\mathbf{D}$ is given in section 2.2.4.

Now, $\mathbf{D}\mathbf{E}^T\vec{\mathbf{Q}}$ is equal to $\vec{\mathbf{Q}}_c$. Finally, we can now write the $\frac{n}{2}(n+1)$ equations in $\frac{n}{2}(n+1)$ unknowns as:

$$(-1)^n \mathbf{D}\mathbf{E}^T\left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T\right]^n \mathbf{E}\vec{\mathbf{L}}_c = \vec{\mathbf{Q}}_c \tag{2.4}$$

The method adopted in this thesis for calculating performance integrals is (i) solve equation (2.4) for $\vec{\mathbf{L}}_c$ (ii) construct the symmetric matrix $\mathbf{L}$ and finally (iii) calculate $\int_0^\infty t^n \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^{n+1} n!\mathbf{x}^T(0)\mathbf{L}\,\mathbf{x}(0),\ n = 0,1,2....$

2.2.3 Constructing the matrix E

The product: $\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T$ gives a n^2 by $\frac{n}{2}(n+1)$ matrix that, when $n=4$ is given by:

$$\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T = \begin{pmatrix} l_{11} \\ l_{12} \\ l_{13} \\ l_{14} \\ l_{21} \\ l_{22} \\ l_{23} \\ l_{24} \\ l_{31} \\ l_{32} \\ l_{33} \\ l_{34} \\ l_{41} \\ l_{42} \\ l_{43} \\ l_{44} \end{pmatrix} \begin{pmatrix} l_{11} & l_{12} & l_{13} & l_{14} & l_{22} & l_{23} & l_{24} & l_{33} & l_{34} & l_{44} \end{pmatrix}$$

$$= \begin{pmatrix} l_{11}l_{11} & l_{11}l_{12} & l_{11}l_{13} & l_{11}l_{14} & l_{11}l_{22} & l_{11}l_{23} & l_{11}l_{24} & l_{11}l_{33} & l_{11}l_{34} & l_{11}l_{44} \\ l_{12}l_{11} & l_{12}l_{12} & l_{12}l_{13} & l_{12}l_{14} & l_{12}l_{22} & l_{12}l_{23} & l_{12}l_{24} & l_{12}l_{33} & l_{12}l_{34} & l_{12}l_{44} \\ l_{13}l_{11} & l_{13}l_{12} & l_{13}l_{13} & l_{13}l_{14} & l_{13}l_{22} & l_{13}l_{23} & l_{13}l_{24} & l_{13}l_{33} & l_{13}l_{34} & l_{13}l_{44} \\ l_{14}l_{11} & l_{14}l_{12} & l_{14}l_{13} & l_{14}l_{14} & l_{14}l_{22} & l_{14}l_{23} & l_{14}l_{24} & l_{14}l_{33} & l_{14}l_{34} & l_{14}l_{44} \\ l_{21}l_{11} & l_{21}l_{12} & l_{21}l_{13} & l_{21}l_{14} & l_{21}l_{22} & l_{21}l_{23} & l_{21}l_{24} & l_{21}l_{33} & l_{21}l_{34} & l_{21}l_{44} \\ l_{22}l_{11} & l_{22}l_{12} & l_{22}l_{13} & l_{22}l_{14} & l_{22}l_{22} & l_{22}l_{23} & l_{22}l_{24} & l_{22}l_{33} & l_{22}l_{34} & l_{22}l_{44} \\ l_{23}l_{11} & l_{23}l_{12} & l_{23}l_{13} & l_{23}l_{14} & l_{23}l_{22} & l_{23}l_{23} & l_{23}l_{24} & l_{23}l_{33} & l_{23}l_{34} & l_{23}l_{44} \\ l_{24}l_{11} & l_{24}l_{12} & l_{24}l_{13} & l_{24}l_{14} & l_{24}l_{22} & l_{24}l_{23} & l_{24}l_{24} & l_{24}l_{33} & l_{24}l_{34} & l_{24}l_{44} \\ l_{31}l_{11} & l_{31}l_{12} & l_{31}l_{13} & l_{31}l_{14} & l_{31}l_{22} & l_{31}l_{23} & l_{31}l_{24} & l_{31}l_{33} & l_{31}l_{34} & l_{31}l_{44} \\ l_{32}l_{11} & l_{32}l_{12} & l_{32}l_{13} & l_{32}l_{14} & l_{32}l_{22} & l_{32}l_{23} & l_{32}l_{24} & l_{32}l_{33} & l_{32}l_{34} & l_{32}l_{44} \\ l_{33}l_{11} & l_{33}l_{12} & l_{33}l_{13} & l_{33}l_{14} & l_{33}l_{22} & l_{33}l_{23} & l_{33}l_{24} & l_{33}l_{33} & l_{33}l_{34} & l_{33}l_{44} \\ l_{34}l_{11} & l_{34}l_{12} & l_{34}l_{13} & l_{34}l_{14} & l_{34}l_{22} & l_{34}l_{23} & l_{34}l_{24} & l_{34}l_{33} & l_{34}l_{34} & l_{34}l_{44} \\ l_{41}l_{11} & l_{41}l_{12} & l_{41}l_{13} & l_{41}l_{14} & l_{41}l_{22} & l_{41}l_{23} & l_{41}l_{24} & l_{41}l_{33} & l_{41}l_{34} & l_{41}l_{44} \\ l_{42}l_{11} & l_{42}l_{12} & l_{42}l_{13} & l_{42}l_{14} & l_{42}l_{22} & l_{42}l_{23} & l_{42}l_{24} & l_{42}l_{33} & l_{42}l_{34} & l_{42}l_{44} \\ l_{43}l_{11} & l_{43}l_{12} & l_{43}l_{13} & l_{43}l_{14} & l_{43}l_{22} & l_{43}l_{23} & l_{43}l_{24} & l_{43}l_{33} & l_{43}l_{34} & l_{43}l_{44} \\ l_{44}l_{11} & l_{44}l_{12} & l_{44}l_{13} & l_{44}l_{14} & l_{44}l_{22} & l_{44}l_{23} & l_{44}l_{24} & l_{44}l_{33} & l_{44}l_{34} & l_{44}l_{44} \end{pmatrix} = \left(l_{ij}l_{kl}\right)$$

To define a procedure for deriving $\mathbf{E}$ from $\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T$ we first define a function ε_{ijkl}:

$$\varepsilon_{ijkl} = \begin{cases} \dfrac{1}{l_{ij}l_{kl}} & \text{if } i=k \text{ and } j=l \\ \dfrac{1}{l_{ij}l_{kl}} & \text{if } i=l \text{ and } j=k \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where $l_{ij} > 0$ and $l_{kl} > 0$. So now we have:

$$\mathbf{E} = \varepsilon_{ijkl}\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T \tag{2.6}$$

where $l_{ij}l_{kl}$ is a typical member of $\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T$ and ε_{ijkl} is as defined in equation (2.5). It is not necessary to say to which vector l_{ij} or l_{kl} belongs (i.e. $\vec{\mathbf{L}}$ or $\vec{\mathbf{L}}_c^T$) to apply equation (2.6).

2.2.4 Constructing the matrix D

To develop a strategy for constructing the matrix $\mathbf{D}$ we first write the product $\mathbf{E}^T\vec{\mathbf{Q}}(\mathbf{E}^T\vec{\mathbf{Q}})^T$ or

$$\mathbf{E}^T\vec{\mathbf{Q}}(\mathbf{E}^T\vec{\mathbf{Q}})^T = \begin{pmatrix} q_{11} \\ 2q_{12} \\ 2q_{13} \\ 2q_{14} \\ \cdots q_{22} \\ 2q_{23} \\ 2q_{24} \\ \cdots q_{33} \\ 2q_{34} \\ \cdots q_{44} \end{pmatrix} \begin{pmatrix} q_{11} & 2q_{12} & 2q_{13} & 2q_{14} & q_{22} & 2q_{23} & 2q_{24} & q_{33} & 2q_{34} & q_{44} \end{pmatrix}$$

$$= \begin{pmatrix} q_{11}q_{11} & 2q_{11}q_{12} & 2q_{11}q_{13} & 2q_{11}q_{14} & q_{11}q_{22} & 2q_{11}q_{23} & 2q_{11}q_{24} & q_{11}q_{33} & 2q_{11}q_{34} & q_{11}q_{44} \\ 2q_{12}q_{11} & 4q_{12}q_{12} & 4q_{12}q_{13} & 4q_{12}q_{14} & 2q_{12}q_{22} & 4q_{12}q_{23} & 4q_{12}q_{24} & 2q_{12}q_{33} & 4q_{12}q_{34} & 2q_{12}q_{44} \\ 2q_{13}q_{11} & 4q_{13}q_{12} & 4q_{13}q_{13} & 4q_{13}q_{14} & 2q_{13}q_{22} & 4q_{13}q_{23} & 4q_{13}q_{24} & 2q_{13}q_{33} & 4q_{13}q_{34} & 2q_{13}q_{44} \\ 2q_{14}q_{11} & 4q_{14}q_{12} & 4q_{14}q_{13} & 4q_{14}q_{14} & 2q_{14}q_{22} & 4q_{14}q_{23} & 4q_{14}q_{24} & 2q_{14}q_{33} & 4q_{14}q_{34} & 2q_{14}q_{44} \\ q_{22}q_{11} & 2q_{22}q_{12} & 2q_{22}q_{13} & 2q_{22}q_{14} & q_{22}q_{22} & 2q_{22}q_{23} & 2q_{22}q_{24} & q_{22}q_{33} & 2q_{22}q_{34} & q_{22}q_{44} \\ 2q_{23}q_{11} & 4q_{23}q_{12} & 4q_{23}q_{13} & 4q_{23}q_{14} & 2q_{23}q_{22} & 4q_{23}q_{23} & 4q_{23}q_{24} & 2q_{23}q_{33} & 4q_{23}q_{34} & 2q_{23}q_{44} \\ 2q_{24}q_{11} & 4q_{24}q_{12} & 4q_{24}q_{13} & 4q_{24}q_{14} & 2q_{24}q_{22} & 4q_{24}q_{23} & 4q_{24}q_{24} & 2q_{24}q_{33} & 4q_{24}q_{34} & 2q_{24}q_{44} \\ q_{33}q_{11} & 2q_{33}q_{12} & 2q_{33}q_{13} & 2q_{33}q_{14} & q_{33}q_{22} & 2q_{33}q_{23} & 2q_{33}q_{24} & q_{33}q_{33} & 2q_{33}q_{34} & q_{33}q_{44} \\ 2q_{34}q_{11} & 4q_{34}q_{12} & 4q_{34}q_{13} & 4q_{34}q_{14} & 2q_{34}q_{22} & 4q_{34}q_{23} & 4q_{34}q_{24} & 2q_{34}q_{33} & 4q_{34}q_{34} & 2q_{34}q_{44} \\ q_{44}q_{11} & 2q_{44}q_{12} & 2q_{44}q_{13} & 2q_{44}q_{14} & q_{44}q_{22} & 2q_{44}q_{23} & 2q_{44}q_{24} & q_{44}q_{33} & 2q_{44}q_{34} & q_{44}q_{44} \end{pmatrix} = \left(q_{ij}q_{kl}\right)$$

To define a procedure for deriving $\mathbf{D}$ from $\mathbf{E}^T\vec{\mathbf{Q}}(\mathbf{E}^T\vec{\mathbf{Q}})^T$ we first define a function δ_{ijkl}:

$$\delta_{ijkl} = \begin{cases} \dfrac{1}{q_{ij}q_{kl}} & \text{if } i = j = k = l \\ \dfrac{1}{8q_{ij}q_{kl}} & \text{if } i = k \text{ and } j = l \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where $q_{ij} > 0$ and $q_{kl} > 0$. So now we have

$$\mathbf{D} = \delta_{ijkl}\mathbf{E}^T\vec{\mathbf{Q}}(\mathbf{E}^T\vec{\mathbf{Q}})^T \tag{2.8}$$

where $q_{ij}q_{kl}$ is a typical member of $\mathbf{E}^T\vec{\mathbf{Q}}(\mathbf{E}^T\vec{\mathbf{Q}})^T$ and δ_{ijkl} is as defined in equation (2.7). Just as for the procedure described in section 2.2.3 for constructing the matrix E, it is not necessary to say to which vector q_{ij} or q_{kl} belongs (i.e. $\mathbf{E}^T\vec{\mathbf{Q}}$ or $(\mathbf{E}^T\vec{\mathbf{Q}})^T$) to apply equation (2.8).

2.2.5 Notes on the matrices E and D

(a) Upper triangular entries in both **L** and **Q** matrices were used above when constructing $\mathbf{L}_c$ and $\mathbf{Q}_c$. Lower triangular entries could also have been used. If this had been done, the same rules for constructing **E** and **D**, i.e. equations (2.6) and (2.8), apply. The resulting **E** and **D** matrices that would be obtained then are those given above turned upside down.

(b) It was not in fact necessary to use the product: $\vec{\mathbf{L}}\vec{\mathbf{L}}_c^T$ to create the matrix **E** as the product of any two vectors of lengths n^2 and $\frac{n}{2}(n+1)$ lead to the same matrix. Similarly, the matrix **D** could have been built up from the product of any two vectors with the same lengths as $\mathbf{E}^T\vec{\mathbf{Q}}$ and $(\mathbf{E}^T\vec{\mathbf{Q}})^T$. The vectors $\vec{\mathbf{L}}$, $\vec{\mathbf{L}}_c^T$, $\mathbf{E}^T\vec{\mathbf{Q}}$, and $(\mathbf{E}^T\vec{\mathbf{Q}})^T$ were used in the derivations above as they arose naturally.

2.3 Example of calculating time weighted performance measures by using equation (2.4)

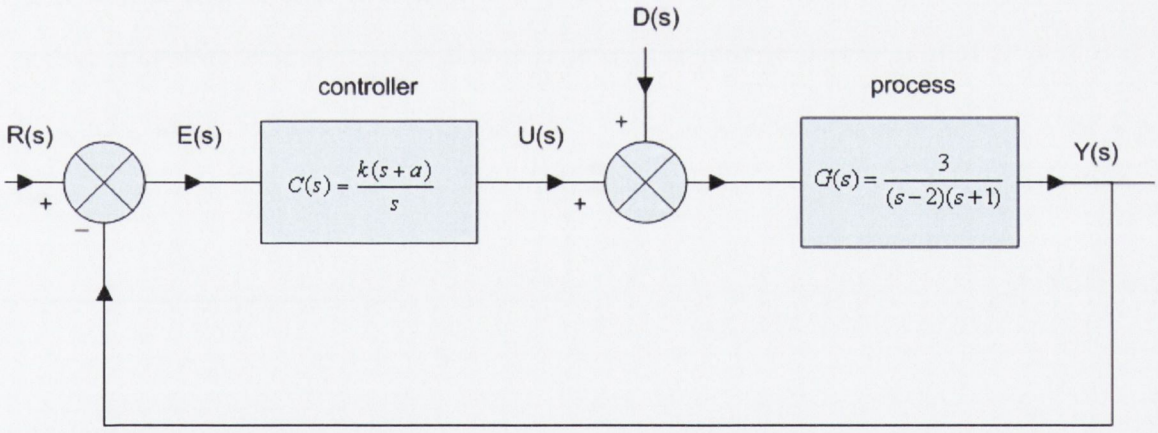


Figure 2.1 A PI controller and a process in a unitary gain, negative feedback loop

A PI controller $C(s)=\frac{k(s+a)}{s}$ is to be designed for the process $G(s)=\frac{3}{(s-2)(s+1)}$. The characteristic equation for the system is $p(s)=s^3-s^2+(3k-2)s+3ka$. If we want the nominal value of k to place all three eigenvalues at the same point, say $s=-\lambda$ then we can equate coefficients: $s^3-s^2+(3k-2)s+3ka=(s+\lambda)^3$ to find the appropriate values for a, k and for λ. Doing this results in $\lambda=\frac{1}{3}$, $k=\frac{7}{9}$ and $a=\frac{1}{63}$.

The following root locus diagrams illustrate the evolution of this breakpoint at a triple eigenvalue.

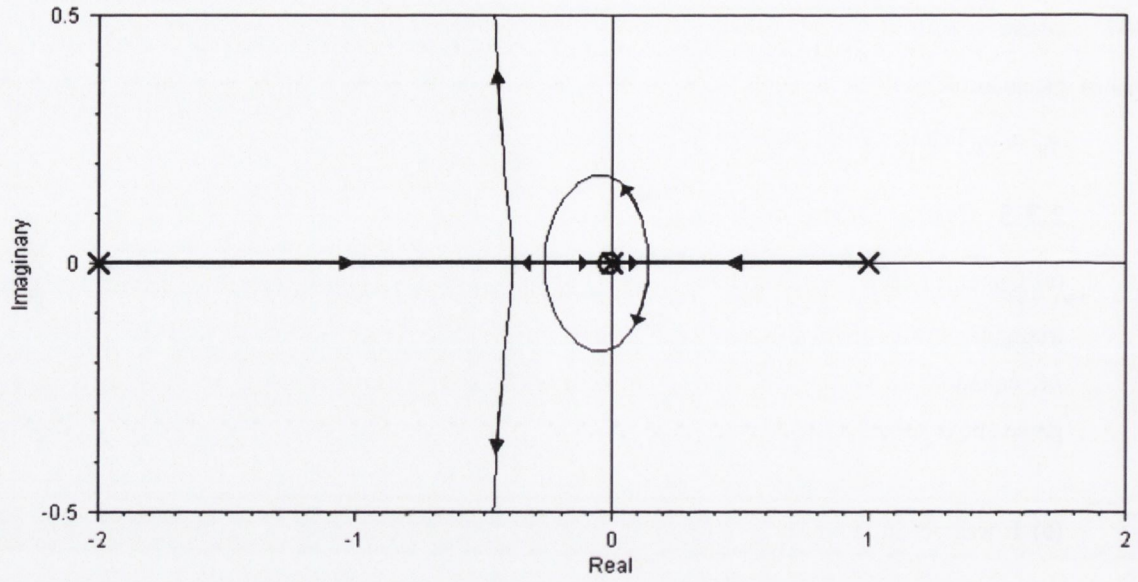


Figure 2.2 Root locus for $p(s) = s(s-1)(s+2) + 3k(s+a)$ with $a = 1/70$

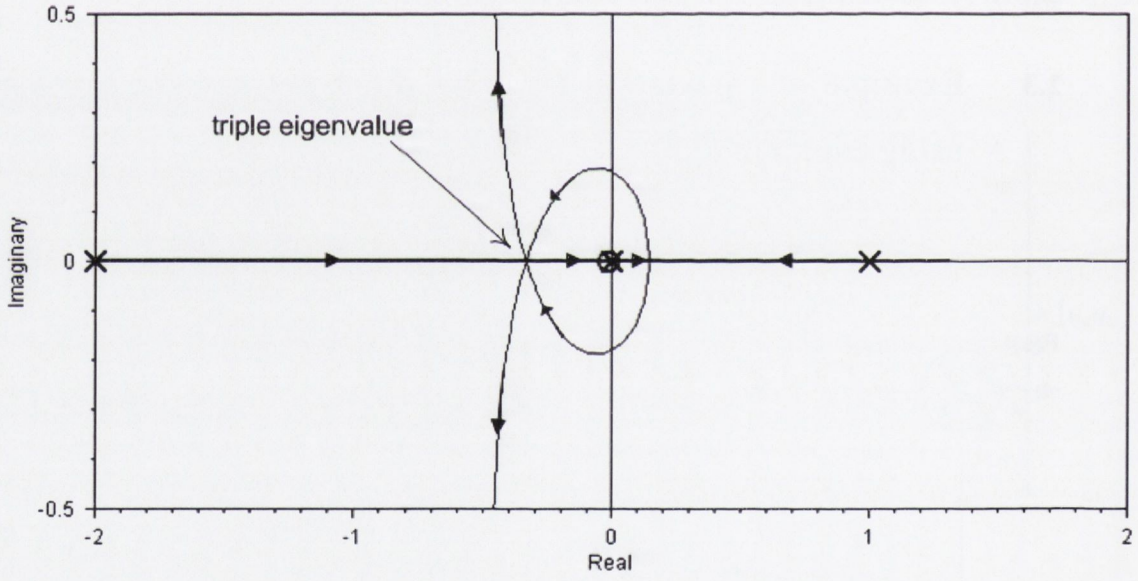


Figure 2.3 Root locus for $p(s) = s(s-1)(s+2) + 3k(s+a)$ with $a = 1/63$ - this is the critical value of a that results in a breakpoint at a triple eigenvalue.

Also, for a step disturbance $D(s) = \frac{1}{s}$ with $R(s) = 0$ the initial conditions are found to be $x^T(0) = (0, 0, -3)$ as follows.

$E(s) = \dfrac{-G(s)}{1 + C(s)G(s)} D(s)$; substituting for $C(s) = \frac{k(s+a)}{s}$, $G(s) = \frac{3}{(s-2)(s+1)}$, and $D(s) = \frac{1}{s}$ gives $E(s) = \dfrac{-3}{s^3 - s^2 + (3k-2)s + 3ka}$. This is the Laplace Transform of the solution to $\dfrac{d^3e}{dt^3} - \dfrac{d^2e}{dt^2} + (3k-2)\dfrac{de}{dt} + 3ka = 0$. So we may derive another expression for $E(s)$ as follows: $(s^3E(s) - s^2e(0) - se'(0) - e''(0)) - (s^2E(s) - se(0) - e'(0)) + (3k-2)(sE(s) - e(0)) + 3kaE(s) = 0$ giving:

$$E(s) = \frac{s^2 e(0) + (e'(0) - e(0)) + e''(0) - e'(0) + (3k-2)e(0)}{s^3 - s^2 + (3k-2)s + 3ka}$$

Equating coefficients in these two expressions for $E(s)$ gives $\mathbf{x}^T(0) = (e(0), e'(0), e''(0))^T = (0,0,-3)^T$.

We can now calculate the performance integral $J_3 = \int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ for the system in Figure 2.1 when it is subjected to a step disturbance $D(s)$. First we construct $\vec{\mathbf{L}}$, $\vec{\mathbf{L}}_c$, and $\mathbf{E}$, using equation (2.6):

$$\mathbf{L} = \begin{pmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{pmatrix}, \quad \vec{\mathbf{L}} = \begin{pmatrix} l_{11} \\ l_{12} \\ l_{13} \\ l_{21} \\ l_{22} \\ l_{23} \\ l_{31} \\ l_{32} \\ l_{33} \end{pmatrix}, \quad \vec{\mathbf{L}}_c = \begin{pmatrix} l_{11} \\ l_{12} \\ l_{13} \\ l_{22} \\ l_{23} \\ l_{33} \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \text{ where } \vec{\mathbf{L}} = \mathbf{E}\vec{\mathbf{L}}_c.$$

Next we construct $\vec{\mathbf{Q}}$, $\vec{\mathbf{Q}}_c$, and $\mathbf{D}$, using equation (2.8):

$$\mathbf{Q} = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix}, \quad \vec{\mathbf{Q}} = \begin{pmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{21} \\ q_{22} \\ q_{23} \\ q_{31} \\ q_{32} \\ q_{33} \end{pmatrix}, \quad \vec{\mathbf{Q}}_c = \begin{pmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{22} \\ q_{23} \\ q_{33} \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \text{ where } \mathbf{D}\mathbf{E}^T\vec{\mathbf{Q}} = \vec{\mathbf{Q}}_c.$$

For this calculation: $\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.037 & -0.333 & -1 \end{pmatrix}$ and we take: $\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $\vec{\mathbf{Q}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\vec{\mathbf{Q}}_c = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$.

Here $\mathbf{Q}$ is positive semi-definite and can be factored as $\mathbf{Q} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \end{pmatrix} = \boldsymbol{p}\boldsymbol{p}^T$. Since $(\boldsymbol{A}, \boldsymbol{p}^T)$ form an observable pair the necessary and sufficient condition for the asymptotic stability of the system described by $\mathbf{A}$ is that $\mathbf{L}$, the solution to $\mathbf{A}^T\mathbf{L} + \mathbf{L}\mathbf{A} = -\mathbf{Q}$, is positive definite [38].

Solving equation (2.4) for $\vec{\mathbf{L}}_c = \begin{pmatrix} 218.32 \\ 922.64 \\ 1053.63 \\ 3986.72 \\ 4613.2 \\ 5382.07 \end{pmatrix}$ one may construct $\mathbf{L} = \begin{pmatrix} 218.32 & 922.64 & 1053.63 \\ 922.64 & 3986.73 & 4613.22 \\ 1053.63 & 4613.22 & 5382.09 \end{pmatrix}$.

Finally, using $\mathbf{x}^T(0) = (0,0,-3)$ we find that $J_3 = \int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^4 3!\mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0) = 290633$.

We can use MacFarlane's procedure to check this calculation. We first have to solve a series of four Lyapunov equations to get L_4 and then calculate $J_3 = \int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^4 3!\mathbf{x}^T(0)\mathbf{L}_4\mathbf{x}(0)$.

$$\mathbf{A}^T\mathbf{L}_1 + \mathbf{L}_1\mathbf{A} = -\mathbf{Q} \text{ gives } \mathbf{L}_1 = \begin{pmatrix} 6.1875 & 15.1875 & 13.5 \\ 15.1875 & 47.2501 & 45.5626 \\ 13.5 & 45.5626 & 45.5626 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{L}_2 + \mathbf{L}_2\mathbf{A} = -\mathbf{L}_1 \text{ gives } \mathbf{L}_2 = \begin{pmatrix} 24.4688 & 82.6876 & 83.5314 \\ 82.6876 & 303.751 & 318.938 \\ 83.5314 & 318.938 & 341.72 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{L}_3 + \mathbf{L}_3\mathbf{A} = -\mathbf{L}_2 \text{ gives } \mathbf{L}_3 = \begin{pmatrix} 78.0469 & 303.75 & 330.329 \\ 303.75 & 1230.19 & 1366.88 \\ 330.329 & 1366.88 & 1537.74 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{L}_4 + \mathbf{L}_4\mathbf{A} = -\mathbf{L}_3 \text{ gives } \mathbf{L}_4 = \begin{pmatrix} 218.32 & 922.642 & 1053.63 \\ 922.642 & 3986.73 & 4613.22 \\ 1053.63 & 4613.22 & 5382.09 \end{pmatrix}$$

So now we can calculate $J_3 = \int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^4 3!\mathbf{x}^T(0)\mathbf{L}_4\mathbf{x}(0) = 290633$, as before.

2.4 Two methods for calculating exponentially weighted performance integrals

We find in section 4.1 that exponentially weighted performance integrals have a deep relationship with root locus based optimum stability design for a class of second order system. We therefore need to develop a method for calculating these integrals as such a method is not available in the literature. In the following two sections I give two methods for calculating these integrals.

2.4.1 Exponentially weighted performance integrals in terms of an infinite sum of matrices

We know from Appendix A.4 that MacFarlane's procedure may be written as:

$$\int_0^\infty t^n \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^{n+1} n!\mathbf{x}^T(0)L_{n+1}\mathbf{x}(0) \text{ where } \mathbf{A}^T\mathbf{L}_{n+1} + \mathbf{L}_{n+1}\mathbf{A} = -\mathbf{L}_n \text{ and } \mathbf{L}_0 = \mathbf{Q},\ n = 0,1,2...$$

It is interesting to note that an expression for an exponentially weighted performance integral may now be derived directly from this equation. Using MacFarlane's formula we get:

$$\text{for } n = 0 \quad \int_0^\infty \frac{t^0}{0!}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = -\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) \text{ where } \mathbf{A}^T\mathbf{L}_1 + \mathbf{L}_1\mathbf{A} = -\mathbf{Q}$$

for $n=1$ $\int_0^\infty \frac{t}{1!}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_2 + \mathbf{L}_2\mathbf{A} = -\mathbf{L}_1$

for $n=2$ $\int_0^\infty \frac{t^2}{2!}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = -\mathbf{x}^T(0)\mathbf{L}_3\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_3 + \mathbf{L}_3\mathbf{A} = -\mathbf{L}_2$ and so on....

Adding these equations we get:

$$\int_0^\infty \left[1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{t^4}{4!} + \ldots\right]\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \mathbf{x}^T(0)\left[-\mathbf{L}_1 + \mathbf{L}_2 - \mathbf{L}_3 + \mathbf{L}_4 \ldots\right]\mathbf{x}(0)$$

So, $\int_0^\infty e^t\mathbf{x}(t)\mathbf{Q}\mathbf{x}(t)dt = \mathbf{x}^T(0)\left[-\mathbf{L}_1 + \mathbf{L}_2 - \mathbf{L}_3 + \mathbf{L}_4 \ldots\right]\mathbf{x}(0)$ where $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3, \mathbf{L}_4 \ldots$ are solutions to the series of Lyapunov matrix equations $\mathbf{A}^T\mathbf{L}_{n+1} + \mathbf{L}_{n+1}\mathbf{A} = -\mathbf{L}_n$ and $\mathbf{L}_0 = \mathbf{Q}$. A more elegant and tractable procedure for calculating such integrals is described in section 2.4.2.

2.4.2 Exponentially weighted performance integrals in terms of the Lyapunov matrix equation

In this thesis we occasionally need to calculate the exponentially weighted performance integral:

$$J = \int_0^\infty \exp(\alpha t)\mathbf{x}^T(t)\mathbf{Q}\,\mathbf{x}(t)dt = \int_0^\infty \exp\left(\frac{\alpha t}{2}\right)\mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t)\exp\left(\frac{\alpha t}{2}\right)dt\,.$$

This may be done as follows. If $\mathbf{x}(t)$ is the solution of $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ then $\mathbf{x}(t).\exp\left(\frac{\alpha t}{2}\right)$ is the solution of $\frac{d\mathbf{x}(t)}{dt} = \left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)\mathbf{x}(t)$. This can be seen as follows. The solution to $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ is $\mathbf{x}(t) = \exp(\mathbf{A}t)\mathbf{x}(0)$.

Multiply both sides of this solution by $\exp\left(\frac{\alpha}{2}t\right)$ to get

$$\exp\left(\frac{\alpha}{2}t\right)\mathbf{x}(t) = \begin{pmatrix} \exp\left(\frac{\alpha}{2}t\right) & 0 & . & . & 0 \\ 0 & \exp\left(\frac{\alpha}{2}t\right) & 0 & . & 0 \\ 0 & 0 & \exp\left(\frac{\alpha}{2}t\right) & . & 0 \\ . & . & . & . & . \\ 0 & 0 & . & . & \exp\left(\frac{\alpha}{2}t\right) \end{pmatrix} \exp(\mathbf{A}t)\mathbf{x}(0)$$

$$= \exp\left(\frac{\alpha}{2}\mathbf{I}t\right)\exp(\mathbf{A}t)\mathbf{x}(0)$$

$$= \exp\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)t\,\mathbf{x}(0) \text{ provided } \left(\frac{\alpha}{2}\mathbf{I}\,t\right) \text{and } (\mathbf{A}t) \text{ commute, as they do.}$$

So, by analogy with the procedure described in Appendix A.4 we have:

$\int_0^\infty \exp(\alpha t)\,\mathbf{x}^T(t)\mathbf{Q}\,\mathbf{x}(t)dt = \mathbf{x}^T(0)\,\mathbf{L}\,\mathbf{x}(0)$ where $\mathbf{L}$ is the solution to $\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)^T\mathbf{L} + \mathbf{L}\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right) = -\mathbf{Q}$.

It is important to note that, for the integral $\int_0^\infty \exp(\alpha t)\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ to converge, we must have: $\lim_{t\to\infty}(\exp(\frac{\alpha}{2}t)\mathbf{x}(t)=0$. This means that $\left(\mathbf{A}+\frac{\alpha}{2}\mathbf{I}\right)$ must have all of its eigenvalues in the left half plane. This means in turn that all the eigenvalues of $\mathbf{A}$ must lie to the left of the line $\sigma=-\frac{\alpha}{2}$ in the (σ,ω) plane.

It is interesting to note that combining this result with the one obtained in section 2.4.1 we can see that the solution to $\left(\mathbf{A}+\frac{1}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}+\frac{1}{2}\mathbf{I}\right)=-\mathbf{Q}$ may be written as the infinite sum:

$$\mathbf{L}=\left[-\mathbf{L}_1+\mathbf{L}_2-\mathbf{L}_3+\mathbf{L}_4...\right]=\sum_{j=1}^{\infty}(-1)^j\mathbf{L}_j \tag{2.9}$$

where the $\mathbf{L}_j$'s are solutions to $\mathbf{A}^T\mathbf{L}_j+\mathbf{L}_j\mathbf{A}=-\mathbf{L}_{j-1}$, where $\mathbf{L}_0=\mathbf{Q}$.

An entirely different derivation of the procedure for calculating the exponentially weighted performance integral is given in section 10.3.

2.5 Calculating discrete time system performance measures - a new derivation of Jury's procedure

Barnett [33] derived an explicit formula for calculating performance sums for discrete time control systems: $S_i=\sum_{k=0}^{\infty}k^i\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k=\mathbf{x}_0^T\left[\sum_{j=1}^{i+1}b_{ij}\mathbf{L}_j\right]\mathbf{x}_0$ where $\mathbf{A}^T\mathbf{L}_{i+1}\mathbf{A}-\mathbf{L}_{i+1}=\mathbf{L}_i$, $\mathbf{L}_0=\mathbf{Q}$ and the coefficients b_{ij} are given by $b_{ij}=(-1)^{i+j-1}\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j-s)^i$.

Jury and Gutman [34] derived a recurrence relation for calculating the same performance sums:

$\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)=\mathbf{x}^T(0)\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_jk^{n-j}(-1)^{j+1}\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]+(-1)^n\mathbf{L}_1]x(0)$. He expressed $\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ in terms of k^{n-j} where $j\geq 1$ so we can calculate $\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ as a recurrence relation.

Barnett's and Jury's methods involve calculations such as: $S_3=\sum_{k=0}^{\infty}k^3\mathbf{x}_k^TQ\mathbf{x}_k=\mathbf{x}_0^T(6\mathbf{L}_4-12\mathbf{L}_3+7\mathbf{L}_2-\mathbf{L}_1)\mathbf{x}_0$ or $S_4=\sum_{k=0}^{\infty}k^4\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k=\mathbf{x}_0^T(24\mathbf{L}_5-60\mathbf{L}_4+50\mathbf{L}_3-15\mathbf{L}_2+\mathbf{L}_1)\mathbf{x}_0$ where $\mathbf{A}^T\mathbf{L}_1\mathbf{A}-\mathbf{L}_1=\mathbf{L}_0$, $\mathbf{A}^T\mathbf{L}_2\mathbf{A}-\mathbf{L}_2=\mathbf{L}_1$, $\mathbf{A}^T\mathbf{L}_3\mathbf{A}-\mathbf{L}_3=\mathbf{L}_2$ etc. , etc.

Later, Mansour et al. [35] derived both Jury's and Barnett's results using different methods.

In this section, the calculus of finite differences [39, 40] is used to derive Jury's recurrence relation. Barnett's formula is also derived by a new method in section 2.6. Neither Barnett [33], Jury and Gutman [34]

or Mansour et al. [35] used finite differences in their derivations - they all gave different, and more difficult derivations for their results than the methods described here.

A new and simple method for calculating performance sums for discrete time control systems is given in section 2.6. This method uses a new number triangle with many interesting properties.

In the following derivation the function $V(x(k)) = (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ is defined and used in the first step in the calculation of performance sums for discrete time control systems. This function is analogous to the function $V(x(t)) = t^n \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)$ used by MacFarlane [30] when calculating performance integrals for continuous time control systems. Special cases with n=0,1,2, and 3 are given in Appendix B.

Consider the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$ where all the eigenvalues of $\mathbf{A}$ lie inside the unit circle. Define a family of performance sums as $S_n = \sum_{k=0}^{\infty} k^n \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$.

The performance sum $S_n = \sum_{k=0}^{\infty} k^n \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ may be evaluated as follows:

First take $V_1(\mathbf{x}(k)) = (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$

Then $\Delta V_1(\mathbf{x}(k)) = \Delta[(k-1)^n](\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) + (k-1)^n \Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + \Delta[(k-1)^n]\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$

Recall that: $\Delta[(k-1)^n] = ((k+1)-1)^n - (k-1)^n = (k^n - (k-1)^n)$

$$\begin{aligned}
\Delta V_1(\mathbf{x}(k)) &= (k^n - (k-1)^n)\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)^n[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&\quad + (k^n - (k-1)^n)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&= k^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \\
&\quad (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - (k-1)^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \\
&\quad k^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&= k^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)
\end{aligned}$$

$$= k^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - [k^n + \sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j]\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

$$= k^n[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] - \sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

$$= k^n \mathbf{x}^T(k)[\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1]\mathbf{x}(k) - \sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

$$= -k^n \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) - \sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

Rearranging terms gives:

$$k^n \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = -\Delta V_1(\mathbf{x}(k)) - \sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

$$\sum_{k=0}^{\infty} k^n \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = -\sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k)) - \sum_{k=0}^{\infty}\sum_{j=1}^{n} {}^nC_j k^{n-j}(-1)^j \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

Recall that:

$$\Delta V_1(\mathbf{x}(k)) = k^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

$$\sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k)) = \sum_{k=0}^{\infty} k^n \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \sum_{k=0}^{\infty} (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

The expression $\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$ may be evaluated term by term as follows:

$(k=0)\quad \Delta V_1(\mathbf{x}(0)) = 0-(-1)^n\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)]$

$(k=1)\quad \Delta V_1(\mathbf{x}(1)) = 1^n\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2)]-0$

$(k=2)\quad \Delta V_1(\mathbf{x}(2)) = 2^n\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3)]-1^n\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2)]$

$(k=3)\quad \Delta V_1(\mathbf{x}(3)) = 3^n\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4)]-2^n\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3)]$

$(k=4)\quad \Delta V_1(\mathbf{x}(4)) = 4^n\mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5)]-3^n\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4)]$

$(k=5)\quad \Delta V_1(\mathbf{x}(5)) = 5^n\mathbf{x}^T(6)\mathbf{L}_1\mathbf{x}(6)]-4^n\mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5)]$

.....

Adding these terms gives:

$$\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k)) = -(-1)^n\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)]$$

So now we have:

$$\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = -\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_j k^{n-j}(-1)^j\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]+(-1)^n\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

$$= -\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_j k^{n-j}(-1)^j\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]+(-1)^n\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

$$= -\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_j k^{n-j}(-1)^j\mathbf{x}^T(0)\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k\mathbf{x}(0)]+(-1)^n\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

$$= -\mathbf{x}^T(0)\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_j k^{n-j}(-1)^j\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]+(-1)^n\mathbf{L}_1]\mathbf{x}(0)$$

So we can now write:

$$\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)\sum_{k=0}^{\infty}\sum_{j=1}^{n}[{}^nC_j k^{n-j}(-1)^{j+1}\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]+(-1)^n\mathbf{L}_1]\mathbf{x}(0) \qquad (2.10)$$

We have expressed $\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)$ in terms of k^{n-j} where $n\geq 1$ and $j\geq 1$. So we can now calculate $\sum_{k=0}^{\infty}k^n\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)$ as a recurrence relation.

Taking $n=0$ as a first case we know from Appendix B.1:

$\sum_{k=0}^{\infty}\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_1\mathbf{A}-\mathbf{L}_1 = -\mathbf{Q}$.

Taking $n=1$ in equation (2.10) we get:

$$\sum_{k=0}^{\infty}k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)\sum_{k=0}^{\infty}[\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]\mathbf{x}(0)-\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

to evaluate $\mathbf{x}^T(0)\sum_{k=0}^{\infty}[\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]\mathbf{x}(0)$ we will first expand the summation:

$$\mathbf{x}^T(0)\sum_{k=0}^{\infty}[\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]\mathbf{x}(0) = \mathbf{x}^T(0)[\mathbf{L}_1+\mathbf{A}^T\mathbf{L}_1\mathbf{A}+\mathbf{A}^{T2}\mathbf{L}_1\mathbf{A}^2+\mathbf{A}^{T3}\mathbf{L}_1\mathbf{A}^3+...]\mathbf{x}(0)$$

we notice that if we let $\mathbf{A}^T\mathbf{L}_2\mathbf{A}-\mathbf{L}_2 = -\mathbf{L}_1$ and substitute for $\mathbf{L}_1$ we get:

$$= \mathbf{x}^T(0)[(\mathbf{L}_2-\mathbf{A}^T\mathbf{L}_2\mathbf{A})+\mathbf{A}^T(\mathbf{L}_2-\mathbf{A}^T\mathbf{L}_2\mathbf{A})\mathbf{A}+\mathbf{A}^{T2}(\mathbf{L}_2-\mathbf{A}^T\mathbf{L}_2\mathbf{A})\mathbf{A}^2+\mathbf{A}^{T3}(\mathbf{L}_2-\mathbf{A}^T\mathbf{L}_2\mathbf{A})\mathbf{A}^3+...]\mathbf{x}(0)$$

$= \mathbf{x}^T(0)[\mathbf{L}_2]\mathbf{x}(0)-\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) = \mathbf{x}^T(0)[\mathbf{L}_2-\mathbf{L}_1]\mathbf{x}(0)$ as required.

Taking $n = 2$ in equation (2.10) we get:

$$\sum_{k=0}^{\infty} k^2 \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)\sum_{k=0}^{\infty}[2k\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]\mathbf{x}(0) - \mathbf{x}^T(0)\sum_{k=0}^{\infty}[\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k]\mathbf{x}(0) + \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

$$= \mathbf{x}^T(0)[2(\mathbf{L}_3 - \mathbf{L}_2)] - (\mathbf{L}_2) + \mathbf{L}_1]\mathbf{x}(0)$$

$$= \mathbf{x}^T(0)[2\mathbf{L}_3 - 3\mathbf{L}_2 + \mathbf{L}_1]\mathbf{x}(0)$$

where we have used the result for $n = 0$, and $n = 1$

Taking $n = 3$ in equation (2.10) we get:

$$\sum_{k=0}^{\infty} k^3 \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)\left\{\sum_{k=0}^{\infty}[{}^3C_1 k^2 \mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k - {}^3C_2 k\,\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k + {}^3C_3\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k] - \mathbf{L}_1\right\}\mathbf{x}(0)$$

$$= \mathbf{x}^T(0)\left\{\sum_{k=0}^{\infty}[3k^2\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k - 3k\,\mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k + \mathbf{A}^{T\,k}\mathbf{L}_1\mathbf{A}^k] - \mathbf{L}_1\right\}\mathbf{x}(0)$$

$$= \mathbf{x}^T(0)[3[2\mathbf{L}_4 - 3\mathbf{L}_3 + \mathbf{L}_2] - 3[\mathbf{L}_3 - \mathbf{L}_2] + \mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)$$

$$= \mathbf{x}^T(0)[6\mathbf{L}_4 - 12\mathbf{L}_3 + 7\mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)$$

where we have used the results for $n = 0$, $n = 1$, and $n = 2$. We can continue like this for all n. Equation (2.10) is Jury's recurrence relation [34] for calculating performance sums for discrete time control systems.

2.5.1 A note on one difference between discrete time and continuous time systems

It is interesting to note that the new derivation given here highlights why the formula for the discrete time case is different to the formula for the continuous time case. In deriving the formula for the continuous time case MacFarlane [30] starts by calculating the derivative of a product. This is given by:

$$\frac{d\,xy}{dt} = y\frac{dx}{dt} + x\frac{dy}{dt}$$

Similarly, the discrete time case starts with the calculation of the finite difference of a product. The finite difference of a product is given by:

$$\Delta(xy) = (\Delta x)y + x(\Delta y) + \Delta x\Delta y\,.$$

So there is an additional term i.e. $\Delta x\Delta y$ in the formula for finding the finite difference of a product. The fundamental mathematical difference between continuous time systems and discrete time systems that makes the calculation of performance measures for discrete time systems harder is the presence of the term $\Delta x\Delta y$ in the formula for the finite difference of a product. There are more terms in the formula for S_i than there are in the formula for J_n because of the presence of this $\Delta x\Delta y$ term.

2.5.2 A note on solving embedded discrete time Lyapunov equations

The series of discrete time Lyapunov equations encountered above is:

$\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$, $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$, $\mathbf{A}^T\mathbf{L}_3\mathbf{A} - \mathbf{L}_3 = -\mathbf{L}_2$, $\mathbf{A}^T\mathbf{L}_4\mathbf{A} - \mathbf{L}_4 = -\mathbf{L}_3$ etc. etc.

As for the continuous time case, we can back substitute for $\mathbf{L}_3$ in the third equation and for $\mathbf{L}_2$ in the second equation and so on.

Examples of the equations obtained by this back substitution procedure are:

$$\mathbf{A}^T\mathbf{L}_1\mathbf{A}-\mathbf{L}_1=-\mathbf{Q}$$
$$-(\mathbf{A}^T)^2\mathbf{L}_2\mathbf{A}^2+2\mathbf{A}^T\mathbf{L}_2\mathbf{A}-\mathbf{L}_2=-\mathbf{Q}$$
$$(\mathbf{A}^T)^3\mathbf{L}_3\mathbf{A}^3-3(\mathbf{A}^T)^2\mathbf{L}_3\mathbf{A}^2+3\mathbf{A}^T\mathbf{L}_3\mathbf{A}-\mathbf{L}_3=-\mathbf{Q}$$
$$-(\mathbf{A}^T)^4\mathbf{L}_4\mathbf{A}^4+4(\mathbf{A}^T)^3\mathbf{L}_4\mathbf{A}^3-6(\mathbf{A}^T)^2\mathbf{L}_4\mathbf{A}^2+4\mathbf{A}^T\mathbf{L}_4\mathbf{A}-\mathbf{L}_4=-\mathbf{Q}$$

Following this procedure a general expression for $-\mathbf{Q}$ in terms of $\mathbf{L}_n$ is found to be:

$$\sum_{j=0}^{n}(-1)^{n-j+1}\binom{n}{j}(\mathbf{A}^T)^{n-j}\mathbf{L}_n(\mathbf{A})^{n-j}=-\mathbf{Q}$$

As for the continuous time case, this matrix equation is linear in $\mathbf{L}_n$. Using the Kronecker product it may be transformed to the following form:

$$\sum_{j=0}^{n}(-1)^{n-j+1}\binom{n}{j}[(\mathbf{A}^T)^{n-j}\otimes(\mathbf{A}^T)^{n-j}]\vec{\mathbf{L}}_n=-\vec{\mathbf{Q}}$$

Finally, this may be simplified to:

$$(-1)^{n+1}[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]^n\vec{\mathbf{L}}_n=-\vec{\mathbf{Q}} \tag{2.11}$$

As for the continuous time case we can employ matrices $\mathbf{E}$ and $\mathbf{D}$ to reduce the number of equations to be solved to a minimum. The methods for constructing the $\mathbf{E}$ and $\mathbf{D}$ matrices are the same as described in sections 2.2.3 and 2.2.4. Using these matrices, the reduced version of equation (2.11) becomes:

$$(-1)^{n+1}\mathbf{D}\mathbf{E}^T[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]^n\mathbf{E}\vec{\mathbf{L}}_c=-\vec{\mathbf{Q}}_c \tag{2.12}$$

The structure of equation (2.12) is identical to that of equation (2.4) - the equivalent equation for the continuous time case. Here we have the solution to the discrete time Lyapunov matrix equation raised to the power of n. However, unlike its continuous time analogue (i.e. equation (2.4)) equation (2.12) does not have an application in the calculation of performance indices.

2.5.3 A note on the condition for the existence of solutions for equation (2.11)

One may write equation (2.11) as $\vec{\mathbf{L}}_n=-[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]^{-n}\vec{\mathbf{Q}}$ only if $[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]$ is invertable. The condition for the invertability of $[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]$ may be derived as follows.

Let $\mathbf{v}_i\otimes\mathbf{v}_j$ be an eigenvector of $[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]$ where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the eigenvectors of $\mathbf{A}^T$ belonging to the eigenvalues λ_i and λ_j respectively.

$$[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}][\mathbf{v}_i\otimes\mathbf{v}_j]=[\mathbf{A}^T\otimes\mathbf{A}^T][\mathbf{v}_i\otimes\mathbf{v}_j]-[\mathbf{I}\otimes\mathbf{I}][\mathbf{v}_i\otimes\mathbf{v}_j]$$
$$=[\mathbf{A}^T\mathbf{v}_i\otimes\mathbf{A}^T\mathbf{v}_j]-[\mathbf{I}\mathbf{v}_i\otimes\mathbf{I}\mathbf{v}_j]=[\lambda_i\mathbf{v}_i\otimes\lambda_j\mathbf{v}_j]-[\mathbf{v}_i\otimes\mathbf{v}_j]=[\lambda_i\lambda_j-1][\mathbf{v}_i\otimes\mathbf{v}_j]$$

So $[\mathbf{A}^T\otimes\mathbf{A}^T-\mathbf{I}\otimes\mathbf{I}]$ is invertable and $\mathbf{A}^T\mathbf{L}\mathbf{A}-\mathbf{L}=-\mathbf{Q}$ has a unique, symmetric solution $\mathbf{L}$, if and only if all possible combinations of $|\lambda_i\lambda_j|\neq 1$.

2.6 A new and simple method for computing the Barnett-Jury coefficients using a number triangle

As mentioned in section 2.5 Barnett [33] , Jury et al [34], and others [35, 41] derived methods for computing performance sums for discrete time control systems. In this thesis, the coefficients b_{ij} that occur in these equations will be called Barnett-Jury coefficients and the equations called Barnett-Jury equations.

It has not been remarked upon elsewhere that the Barnett-Jury coefficients also occur in another context. Take the integers and write them in a column. Then beside that column write a column of those integers to some power. In subsequent columns, write the difference between each pair of entries in the previous column. Examples of such arrays of numbers are given in Figure 2.4.

i = 1

n	**n^1**	⌐
1	**1**	↓
		1
2	2	
		1
3	3	
		1
4	4	
		1
5	5	
		1
6	6	
		1
7	7	
		1
8	8	
		1
9	9	
		1
10	10	

i = 2

n	**n^2**		**2!**
1	**1**		↓
		3	
2	4		**2**
		5	
3	9		2
		7	
4	16		2
		9	
5	25		2
		11	
6	36		2
		13	
7	49		2
		15	
8	64		2
		17	
9	81		2
		19	
10	100		

i = 3

n	**n^3**			**3!**
1	**1**			↓
		7		
2	8		**12**	
		19		**6**
3	27		18	
		37		6
4	64		24	
		61		6
5	125		30	
		91		6
6	216		36	
		127		6
7	343		42	
		169		6
8	512		48	
		217		6
9	729		54	
		271		
10	1000			

i = 4

n	**n^4**	*j* = 2	*j* = 3	*j* = 4	*j* = 5
1	**1**	↓	↓	↓	↓
		15			**4!**
2	16		**50**		↓
		65		**60**	
3	81		110		**24**
		175		84	
4	256		194		24
		369		108	
5	625		302		24
		671		132	
6	1296		434		24
		1105		156	
7	2401		590		24
		1695		180	
8	4096		770		24
		2465		204	
9	6561		974		
		3439			
10	10000				

i = 5

n	**n^5**	*j* = 2	*j* = 3	*j* = 4	*j* = 5	*j* = 6
1	**1**	↓	↓	↓	↓	↓
		31				
2	32		**180**			**5!**
		211		**390**		↓
3	243		570		**360**	
		781		750		**120**
4	1024		1320		480	
		2101		1230		120
5	3125		2550		600	
		4651		1830		120
6	7776		4380		720	
		9031		2550		120
7	16807		6930		840	
		15961		3390		120
8	32768		10320		960	
		26281		4350		
9	59049		14670			
		40951				
10	100000					

Figure 2.4 Columns of integers raised to powers of 1 to 5. Subsequent columns show the differences between pairs of figures in the previous columns. The top row of each array (in bold) consists of Barnett - Jury coefficients.

It will be shown below that the first row of each of the arrays in Figure 2.4, highlighted in bold type, consists of Barnett-Jury coefficients (ignoring signs for the moment). Writing the first row of each of these arrays as a number triangle give:

1

1 1

1 3 2

1 7 12 6

1 15 50 60 24

1 31 180 390 360 120

Figure 2.5 A number triangle for generating the Barnett-Jury coefficients

By inspection of Figure 2.5 a scheme for generating this triangle suggests itself. This scheme is illustrated in Figure 2.6.

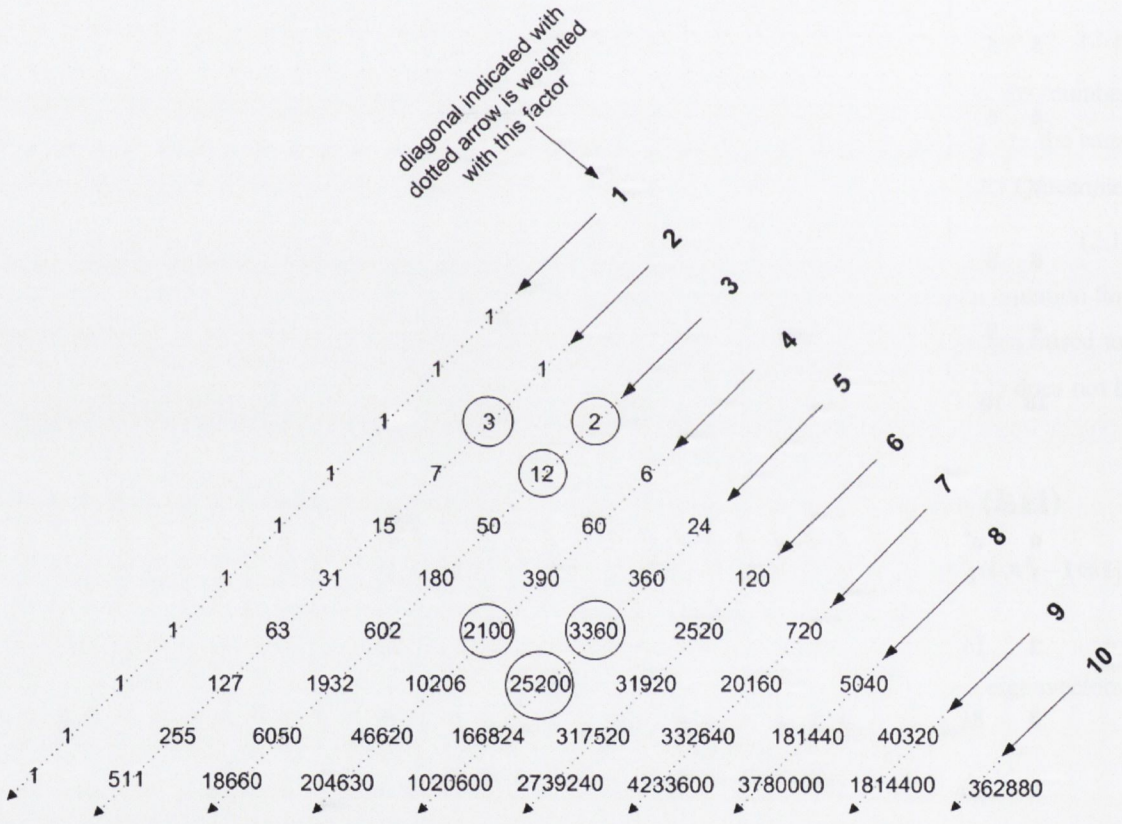


Figure 2.6 Generating a number triangle similar to Pascal's triangle but with the diagonals weighted as shown. For example $\mathbf{2}\times 3+\mathbf{3}\times 2=12$ and $\mathbf{4}\times 2100+\mathbf{5}\times 3360=25200$ etc. etc. This scheme allows one to calculate the Barnett-Jury coefficients for any value of i. Coefficients from $i=0$ to $i=9$ are given in this figure. Jury [42] gives coefficients up to $i=10$.

As in Pascal's triangle, calculating a new row is done by using the numbers in the previous row. The difference in this case is that the numbers in the previous row are weighted before adding them together. If these diagonal weighting factors are all set to 1 then this triangle becomes Pascal's triangle.

Jury [42] gives values for the coefficients in the expansions for $S_i = \sum_{k=0}^{\infty} k^i \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k$, $i = 1$ to $i = 10$.

There is a typographical error in his expansion for $S_{10} = \sum_{k=0}^{\infty} k^{10} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k$ - the coefficient for $\mathbf{L}_4$ is given as 877500 – it should be 874500. All the other coefficients given by Jury agree with those calculated using the number triangle in Figure 2.6.

It will be shown below that in making this array, one is actually performing the calculation using Barnett's equation. In [34] Jury proposes performing these calculations by using Pascal's triangle for $(x-y)^n$, weighting the entries appropriately and then summing the rows. Jury's procedure simply generates a single row of the number triangle in Figure 2.5. For example, if $i = 4$ Jury's procedure generated the fifth row in Figure 2.5 as illustrated in Figure 2.7.

$$
\begin{array}{ccccccccccc}
 & & & & \mathbf{1}\times 1^4 & & & & & = & 1 \\
 & & & \mathbf{1}\times 2^4 & & -\mathbf{1}\times 1^4 & & & & = & 15 \\
 & & \mathbf{1}\times 3^4 & & -\mathbf{2}\times 2^4 & & \mathbf{1}\times 1^4 & & & = & 50 \\
 & \mathbf{1}\times 4^4 & & -\mathbf{3}\times 3^4 & & \mathbf{3}\times 2^4 & & -\mathbf{1}\times 1^4 & & = & 60 \\
\mathbf{1}\times 5^4 & & -\mathbf{4}\times 4^4 & & \mathbf{6}\times 3^4 & & -\mathbf{4}\times 2^4 & & \mathbf{1}\times 1^4 & = & 24
\end{array}
$$

Figure 2.7 Jury's method [34] for calculating b_{ij} with $i = 4$.

The number triangle proposed in Figure 2.5 is simpler to generate and automatically gives the coefficients for all i.

The indexing protocol used next is chosen so that the general formulas derived coincide with those derived by Barnett [33]. In order to keep track of the indices, the number triangle is written out again with indexing terms indicated.

$$
\begin{array}{lccccccccccc}
 & & & & & & p=1 & & & & & \\
 & & & & & \swarrow & & p=2 & & & & \\
i=0 \rightarrow & & & & & 1 & & \swarrow & & p=3 & & \\
i=1 \rightarrow & & & & 1 & & 1 & & \swarrow & & p=4 & \\
i=2 \rightarrow & & & 1 & & 3 & & 2 & & \swarrow & & p=5 \\
i=3 \rightarrow & & 1 & & 7 & & 12 & & 6 & & \swarrow & & p=6 \\
i=4 \rightarrow & 1 & & 15 & & 50 & & 60 & & 24 & & \swarrow & \\
i=5 \rightarrow & 1 & & 31 & & 180 & & 390 & & 360 & & 120 & \\
 & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & \\
 & j=1 & & j=2 & & j=3 & & j=4 & & j=5 & & j=6 &
\end{array}
$$

Figure 2.8 Indexing protocol used in this section

To derive the relevant properties of the number triangle write out a general triangle for n^i.

n	$\mathbf{n^i}$	j = 2	j = 3	j = 4	. .	j = *i* + 1	j = *i* + 2
1	**1**	↓	↓	↓		↓	↓
		$2^i - 1^i$					
2	2^i		$3^i - 2\times 2^i + 1^i$				
		$3^i - 2^i$		$4^i - 3\times 3^i + 3\times 2^i - 1^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j-s)^i = i!$	
3	3^i		$4^i - 2\times 3^i + 2^i$				0
		$4^i - 3^i$		$5^i - 3\times 4^i + 3\times 3^i - 2^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+1-s)^i = i!$	
4	4^i		$5^i - 2\times 4^i + 3^i$				0
		$5^i - 4^i$		$6^i - 3\times 5^i + 3\times 4^i - 3^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+2-s)^i = i!$	
5	5^i		$6^i - 2\times 5^i + 4^i$				0
		$6^i - 5^i$		$7^i - 3\times 6^i + 3\times 5^i - 4^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+3-s)^i = i!$	
6	6^i		$7^i - 2\times 6^i + 5^i$				0
		$7^i - 6^i$		$8^i - 3\times 7^i + 3\times 6^i - 5^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+4-s)^i = i!$	
7	7^i		$8^i - 2\times 7^i + 6^i$				0
		$8^i - 7^i$		$9^i - 3\times 8^i + 3\times 7^i - 6^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+5-s)^i = i!$	
8	8^i		$9^i - 2\times 8^i + 7^i$				0
		$9^i - 8^i$		$10^i - 3\times 9^i + 3\times 8^i - 7^i$	. .	$\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j+6-s)^i = i!$	
9	9^i		$10^i - 2\times 9^i + 8^i$	.	. .	.	.
		$10^i - 9^i$	.	.	. .	.	.
10	10^i	.	.	.	. .	.	.
.	.	.	.	.	. .	.	.

Figure 2.9 Finite difference array for n^i

The first row of Figure 2.9 replicates exactly the calculation scheme that was suggested by Jury [34]. Also, the general term for the first row i.e. $\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j-s)^i$, is also the general term derived by Barnett [33] for $|b_{ij}|$ - that is the Barnett-Jury coefficients but without the signs. Repeating the previous example, if $i = 4$ the first row is shown in Figure 2.10.

$$\mathbf{1}\times 1^k = \mathbf{1}\times 1^4 = 1$$

$$\mathbf{1}\times 2^k - \mathbf{1}\times 1^k = \mathbf{1}\times 2^4 - \mathbf{1}\times 1^4 = 15$$

$$\mathbf{1}\times 3^k - \mathbf{2}\times 2^k + \mathbf{1}\times 1^k = \mathbf{1}\times 3^4 - \mathbf{2}\times 2^4 + \mathbf{1}\times 1^4 = 50$$

$$\mathbf{1}\times 4^k - \mathbf{3}\times 3^k + \mathbf{3}\times 2^k - \mathbf{1}\times 1^k = \mathbf{1}\times 4^4 - \mathbf{3}\times 3^4 + \mathbf{3}\times 2^4 - \mathbf{1}\times 1^4 = 60$$

$$\mathbf{1}\times 5^k - \mathbf{4}\times 4^k + \mathbf{6}\times 3^k - \mathbf{4}\times 2^k + \mathbf{1}\times 1^k = \mathbf{1}\times 5^4 - \mathbf{4}\times 4^4 + \mathbf{6}\times 3^4 - \mathbf{4}\times 2^4 + \mathbf{1}\times 1^4 = 24$$

Figure 2.10 Top row of Figure 2.9 with $i = 4$

Figure 2.10 is as found above in Figure 2.7 using Jury's procedure directly. So, we now know that for at least $i = 1,2,3,$ and 4, the top row of Figure 2.9 will replicate Jury's procedure and it will generate the correct coefficients.

It remains to prove that the rule for generating row $i+1$ in Figure 2.6 from row i is valid for all i. Writing out the first five rows of the number triangle indicates haw to write general term for different rows and diagonals.

$$
\begin{array}{ccccccccccccccccc}
 & & & & & & 1\times1^{0} & & & & & & & & & & \\
 & & & & 1\times1^{1} & & & & 1\times2^{1}-1\times1^{1} & & & & & & & & \\
 & & 1\times1^{2} & & & & 1\times2^{2}-1\times1^{2} & & & & 1\times3^{2}-2\times2^{2}+1\times1^{2} & & & & & & \\
 & 1\times1^{3} & & & 1\times2^{3}-1\times1^{3} & & & & 1\times3^{3}-2\times2^{3}+1\times1^{3} & & & & 1\times4^{3}-3\times3^{3}+3\times2^{3}-1\times1^{3} & & & & \\
1\times1^{4} & & 1\times2^{4}-1\times1^{4} & & & & 1\times3^{4}-2\times2^{4}+1\times1^{4} & & & & 1\times4^{4}-3\times3^{4}+3\times2^{4}-1\times1^{4} & & & & 1\times5^{4}-4\times4^{4}+6\times3^{4}-4\times2^{4}+1\times1^{4} & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}
$$

Figure 2.11 Top row of Figure 2.9 with i=0,1,2,3,4

For the purpose of this analysis I'll use the symbol $\overset{?}{=}$ to indicate that the equality is being checked.

$$\text{term from row i diagonal p} = \sum_{s=0}^{p-1}(-1)^s\binom{p-1}{s}(p-s)^i = \textbf{term 1}$$

$$\text{term from row i diagonal p+1} = \sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^i = \textbf{term 2}$$

$$\text{term from row i+1 diagonal p+1} = \sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^{i+1} = \textbf{term 3}$$

If the rule for generating the triangle is true in general, then the following identity must be true:

$$\underset{\downarrow}{\mathbf{p\times term\ 1}} \qquad \underset{\downarrow}{\mathbf{(p+1)\times term\ 2}} \qquad \underset{\downarrow}{\mathbf{term\ 3}}$$

$$(p)\sum_{s=0}^{p-1}(-1)^s\binom{p-1}{s}(p-s)^i + (p+1)\sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^i \overset{?}{=} \sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^{i+1}$$

First the limits of the summation in term 1 may be rewritten to give:

$$\underset{\downarrow}{\mathbf{p\times term\ 1}} \qquad \underset{\downarrow}{\mathbf{(p+1)\times term\ 2}} \qquad \underset{\downarrow}{\mathbf{term\ 3}}$$

$$(p)\sum_{s=1}^{p}(-1)^{s-1}\binom{p-1}{s-1}(p+1-s)^i + (p+1)\sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^i \overset{?}{=} \sum_{s=0}^{p}(-1)^s\binom{p}{s}(p+1-s)^{i+1}$$

This identity may be established term by term by following (p+1) steps shown in Table 2.1.

	term 1	+	**term 2**	=	**term 3**
Step 1	----------	----	term 2 with s = 0	=	term 3 with s = 0
Step 2	term 1 with s = 1	+	term 2 with s = 1	=	term 3 with s = 1
Step 3	term 1 with s = 2	+	term 2 with s = 2	=	term 3 with s = 2
.	.	.	.	.	.
.	.	.	.	.	.
Step p	term 1 with s = p - 1	+	term 2 with s = p - 1	=	term 3 with s = p - 1
Step (p + 1)	term 1 with s = p	+	term 2 with s = p	=	term 3 with s = p

Table 2.1 Illustration of the scheme used to show that the rule for generating row $i+1$ in Figure 2.9 from row i is valid for all i.

Step 1 is as follows:

$$\underset{\downarrow}{\mathbf{(p+1)\times term\ 2}} \qquad \underset{\downarrow}{\mathbf{term\ 3}}$$

$$(p+1)(-1)^0\binom{p}{0}(p+1)^i \;=\; (-1)^0\binom{p}{0}(p+1)^{i+1}$$

Step 2 to Step (p+1) are as follows:

$$\underset{\downarrow}{\mathbf{p\times term\ 1}} \qquad \underset{\downarrow}{\mathbf{(p+1)\times term\ 2}} \qquad \underset{\downarrow}{\mathbf{term\ 3}}$$

$$(p)\sum_{s=1}^{p}(-1)^{s-1}\binom{p-1}{s-1}(p+1-s)^i \;+\; (p+1)\sum_{s=1}^{p}(-1)^s\binom{p}{s}(p+1-s)^i \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s\binom{p}{s}(p+1-s)^{i+1}$$

$$(p)\sum_{s=1}^{p}(-1)^{s-1}\binom{p-1}{s-1}(p+1-s)^i + (p)\sum_{s=1}^{p}(-1)^s\binom{p}{s}(p+1-s)^i + \sum_{s=1}^{p}(-1)^s\binom{p}{s}(p+1-s)^i \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s\binom{p}{s}(p+1-s)^{i+1}$$

$$\sum_{s=1}^{p}(-1)^{s-1}\binom{p-1}{s-1} + \sum_{s=1}^{p}(-1)^s\binom{p}{s} + \sum_{s=1}^{p}(-1)^s\binom{p}{s}\frac{1}{p} \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s\binom{p}{s}\frac{(p+1-s)}{p}$$

$$\sum_{s=1}^{p}\left[(-1)^{s-1}\binom{p-1}{s-1} + (-1)^s\binom{p}{s} + (-1)^s\binom{p}{s}\frac{1}{p}\right] \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s\binom{p}{s}\frac{(p+1-s)}{p}$$

$$\sum_{s=1}^{p}(-1)^s\left[-\frac{(p-1)!}{(s-1)!(p-s)!} + \frac{p!}{s!(p-s)!} + \frac{(p-1)!}{s!(p-s)!}\right] \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s\frac{(p-1)!}{s!(p-s)!}(p+1-s)$$

$$\sum_{s=1}^{p}(-1)^s\left[-\frac{(p-1)!s!(p-s)!}{(s-1)!(p-s)!} + p! + (p-1)!\right] \;\overset{?}{=}\; \sum_{s=1}^{p}(-1)^s(p-1)!(p+1-s)$$

$$\sum_{s=1}^{p}(-1)^s\left[-(p-1)!s + p! + (p-1)!\right] \;=\; \sum_{s=1}^{p}(-1)^s(p-1)!(p+1-s)$$

So the rule for generating row $i+1$ from row i is valid in general and not just for the specific cases given above.

2.7 A relationship between Barnett-Jury coefficients and Stirling Numbers of the Second Kind

There is a close relationship between the Barnett-Jury coefficients and Stirling Numbers of the Second Kind. Stirling numbers, named for the Scottish mathematician Sir James Stirling (1692-1770), are denoted by $\mathbb{S}_n^{\{m\}}$ and are given by the expression [43]: $\mathbb{S}_n^{\{m\}} = \frac{1}{m!}\sum_{k=0}^{m}(-1)^{m-k}\binom{m}{k}k^n$. $\mathbb{S}_n^{\{m\}}$ is the number of ways of partitioning n elements into m non-empty subsets. For example the set $\{1,2,3\}$ can be partitioned into three subsets in one way: $\{\{1,\},\{2\},\{3\}\}$; into two subsets in three ways: $\{\{1,2\},\{3\}\},\{\{1,3\},\{2\}\}$, and $\{\{1\},\{2,3\}\}$; and into one subset in one way: $\{\{1,2,3\}\}$. Adopting Barnett's notational convention the expression for $\mathbb{S}_n^{\{m\}}$ becomes: $\mathbb{S}_i^{\{j\}} = \frac{1}{(j-1)!}\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j-s)^i$ The numbers $\mathbb{S}_i^{\{j\}}$ can be written as a number triangle:

$$
\begin{array}{lccccccccccc}
i=0 \rightarrow & & & & & & 1 & & & & & \\
i=1 \rightarrow & & & & & 1 & & 1 & & & & \\
i=2 \rightarrow & & & & 1 & & 3 & & 1 & & & \\
i=3 \rightarrow & & & 1 & & 7 & & 6 & & 1 & & \\
i=4 \rightarrow & & 1 & & 15 & & 25 & & 10 & & 1 & \\
i=5 \rightarrow & 1 & & 31 & & 90 & & 65 & & 15 & & 1 \\
 & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & j=1 & & j=2 & & j=3 & & j=4 & & j=5 & & j=6
\end{array}
$$

Figure 2.12 Stirling Numbers of the Second Kind written as a number triangle. An indexing protocol for the rows and columns is indicated.

If the Barnett-Jury coefficients multiplied by the factor: $1/(j-1)!$ then the Stirling numbers of the second kind result. Alternatively, if the Stirling numbers of the second kind are multiplied by $(j-1)!$ then the Barnett-Jury coefficients result. This may be written as: $(j-1)!\,\mathbb{S}_i^{\{j\}} = \left|b_{ij}\right|$. Incorporating the signs we can say: $(-1)^{i+j-1}(j-1)!\,\mathbb{S}_i^{\{j\}} = b_{ij}$

For example, in the $i=4$ row the Stirling numbers are 1,15,25,10,1. If these are multiplied by the appropriate factors we get: $1\times 0!=1, 15\times 1!=15,\ 25\times 2!=50,\ 10\times 3!=60$ and $1\times 4!=24$ giving the $i=4$ row of Barnett-Jury coefficients. So we can write, for $i=4$ for example:

$$S_4 = \sum_{k=0}^{\infty} k^4 \mathbf{x}_k^T \mathbf{Q}\mathbf{x}_k = \mathbf{x}_0^T\left(\sum_{j=1}^{5}(-1)^{j+1}(j-1)!\mathbb{S}_4^{\{j\}}\mathbf{L}_j\right)\mathbf{x}_0 \qquad (2.13)$$

Or in general:

$$S_n = \sum_{k=0}^{\infty} k^n \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k = \mathbf{x}_0^T \left(\sum_{j=1}^{n+1} (-1)^{i+j-1} (j-1)! \mathbb{S}_n^{\{j\}} \mathbf{L}_j \right) \mathbf{x}_0$$

Equation (2.13) is the first time that a discrete time performance sum has been written explicitly in terms of Stirling Numbers of the Second Kind.

The Barnett-Jury coefficients have several properties that seem to have gone unnoticed elsewhere. One such property is studied in the next section.

2.8 Proving a property of Barnett-Jury coefficients using Stirling Numbers of the Second Kind

If the signs of the coefficients are included in the number triangle, the sum across each row appears to be zero, except for the first row. This may be seen as follows:

| | | |

2.9 Further interesting and novel properties of the Barnett-Jury coefficients

A trivial corollary $\sum_{j=1}^{i+1} b_{ij} = 0$ is that the sums comprising of every second term in each row are equal. Examples of this in rows 6, 7 and 8 are as follows:

$$\begin{aligned} 1+180+360 &= 541 = 31+390+120 \\ 1+602+3360+720 &= 4683 = 63+2100+2520 \\ 1+1932+25200+20160 &= 47293 = 127+10206+31920+5040 \end{aligned}$$

Other properties of the Barnett-Jury coefficients are that the coefficient of $\mathbf{L}_{i+1}$ is always $i!$ and the coefficient of $\mathbf{L}_i$ is always $(i+1)!/2$. These results may be derived from properties of combinatorial coefficients.

Presumably the Barnett-Jury coefficients have other useful applications and interesting properties.

2.10 A method for calculating exponentially weighted performance sums

Consider $S_\alpha = \sum_{k=0}^{\infty} e^{\alpha k} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ where $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$

We can write this as: $S_\alpha = \sum_{k=0}^{\infty} \left[e^{\frac{\alpha}{2}k}\mathbf{x}^T(k)\right]\mathbf{Q}\left[e^{\frac{\alpha}{2}k}\mathbf{x}(k)\right]$

Let $\mathbf{y}(k) = e^{\frac{\alpha}{2}k}\mathbf{x}(k)$

Then $\mathbf{y}(k+1) = e^{\frac{\alpha}{2}(k+1)}\mathbf{x}(k+1)$

$$\begin{aligned} &= e^{\frac{\alpha}{2}}e^{\frac{\alpha k}{2}}\mathbf{A}\mathbf{x}(k) \\ &= (e^{\frac{\alpha}{2}}\mathbf{A})e^{\frac{\alpha k}{2}}\mathbf{x}(k) \\ &= (e^{\frac{\alpha}{2}}\mathbf{A})\mathbf{y}(k) \end{aligned}$$

Now we have $S_\alpha = \sum_{k=0}^{\infty} \mathbf{y}^T(k)\mathbf{Q}\mathbf{y}(k)$ where $\mathbf{y}(k+1) = (e^{\frac{\alpha}{2}}\mathbf{A})\mathbf{y}(k)$

$$\Rightarrow S_\alpha = \mathbf{y}^T(0)\mathbf{L}\mathbf{y}(0)$$

$$\Rightarrow S_\alpha = \sum_{k=0}^{\infty} e^{\alpha k}\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0) \qquad (2.15)$$

where $\mathbf{L}$ is the solution of $(e^{\frac{\alpha}{2}}\mathbf{A}^T)\mathbf{L}(e^{\frac{\alpha}{2}}\mathbf{A}) - \mathbf{L} = -\mathbf{Q}$

For convergence of this infinite sum we require that $(e^{\frac{\alpha}{2}}\mathbf{A})$ has all of its eigenvalues in the unit circle. Note: If λ is an eigenvalue of $\mathbf{A}$ then $e^{\frac{\alpha}{2}}\lambda$ is an eigenvalue of $e^{\frac{\alpha}{2}}\mathbf{A}$. (2.15) is derived differently in section 10.5.

2.11 Summary of results in this chapter

Performance integrals for continuous and discrete time control systems have been described. Procedures for calculating these performance integrals have been developed within a unified framework provided by the Kronecker product and MacFarlane's procedure [30]. MacFarlane's procedure has been simplified and extended to discrete time systems. All new procedures have been presented in a way that may be implemented easily using standard computer programs.

To calculate $J_n = \int_0^\infty t^n \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ the well-known continuous time control systems performance integrals for the system $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ we solve $(-1)^n \mathbf{D}\mathbf{E}^T\left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T\right]^n \mathbf{E}\vec{\mathbf{L}}_c = \vec{\mathbf{Q}}_c$ for the entries in the matrix $\mathbf{L}$ and then calculate $J_n = \int_0^\infty t^n \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^{n+1} n!\, \mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0)$.

The methods for calculating $S_n = \sum_{k=0}^{\infty} k^n \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$, the family of discrete time control system performance sums for the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$ proposed by Jury [34] and Barnett [33] and Mansour [35], are simplified by using the number triangle given in section 2.6. The coefficients that arise in these calculations have several interesting properties that have not been remarked upon before – including a relationship with Stirling numbers of the second kind. These properties are described in section 2.7.

The number of equations to be solved when calculating J_n or S_n may be reduced to a minimum by the use of the $\mathbf{E}$ and $\mathbf{D}$ matrices described in section 2.2.2, 2.2.3, and 2.2.4.This method is simpler than that derived by Chen and Shieh [32].

A proof is given in section 2.2.1 that $(-1)^{n+1}\left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T\right]^{n+1} \vec{\mathbf{L}} = \vec{\mathbf{Q}}$ is solvable if and only if all possible combinations of $(\lambda_i + \lambda_j) \neq 0$. A proof is given in section 2.5.3 that $(-1)^n[\mathbf{A}^T \otimes \mathbf{A}^T - \mathbf{I} \otimes \mathbf{I}]^{n+1}\vec{\mathbf{L}} = -\vec{\mathbf{Q}}$ is solvable if and only if all possible combinations of $\lambda_i\lambda_j \neq 1$.

Expressions for exponentially weighted performance integrals and sums were derived in sections 2.4.1, 2.4.2, and 2.10.

2.12 Suggestions for future work

(i) Investigate why the coefficients in $J_4 = \mathbf{x}_1^T(k)(-\mathbf{L}_2 + 14\mathbf{L}_3 - 36\mathbf{L}_4 + 24\mathbf{L}_5)\mathbf{x}_1(k)$ [33] i.e. {1,14,36,24} are available from row 3 ({1,7,12,6}) of the new number triangle proposed in section 2.6 by simply multiplying the Barnett-Jury coefficients by j to get: $\sum_{s=0}^{j-1} j(-1)^s \binom{j-1}{s}(j-s)^i$.

(ii) Prove that, ignoring signs, and taking the second number in each row the new number triangle in section 2.6 we get the sequence 1,3,7,15,31,63,... - these are Mersenne numbers i.e. numbers of the form $2^i - 1$.

(iii) Prove that the first two numbers in each row of the new number triangle in section 2.6 are uneven – the remaining numbers in each row are even.

(iv) Prove that the coefficient of $\mathbf{L}_{i+1}$ is always $i!$ and the coefficient of $\mathbf{L}_i$ is always $(i+1)!/2$.

Chapter 3: Using root locus based optimum stability to design a controller of minimum complexity for an unstable, non-minimum phase process

3.1 Introduction

The deceptively simple problem of designing a single-loop, error-actuated feedback system is considered anew. Fundamental concepts only are invoked. *Minimum controller complexity* achieves arbitrary eigenvalue assignment. *Optimum stability* places the rightmost eigenvalue as deep in the left half plane as possible. *Desirable side conditions* confer static disturbance rejection and unity static gain between reference input and process output. Root loci show that any system designed to have all eigenvalues equal is *optimally stable* with respect to variation of *any* design parameter from its nominal value. In a cautionary vein, gain and phase margins are used to compare a design arrived at here with an overly complex one yielded by the H_∞ approach.

3.2 A desirable controller structure

The following analysis illustrates how very desirable system behavior (the "two common secondary requirements" listed below) is obtained simply by including a pole at the origin in the transfer function of the controller. Silva et al. [44] refer to this as "the magic of integral control". This controller structure is used in the following sections.

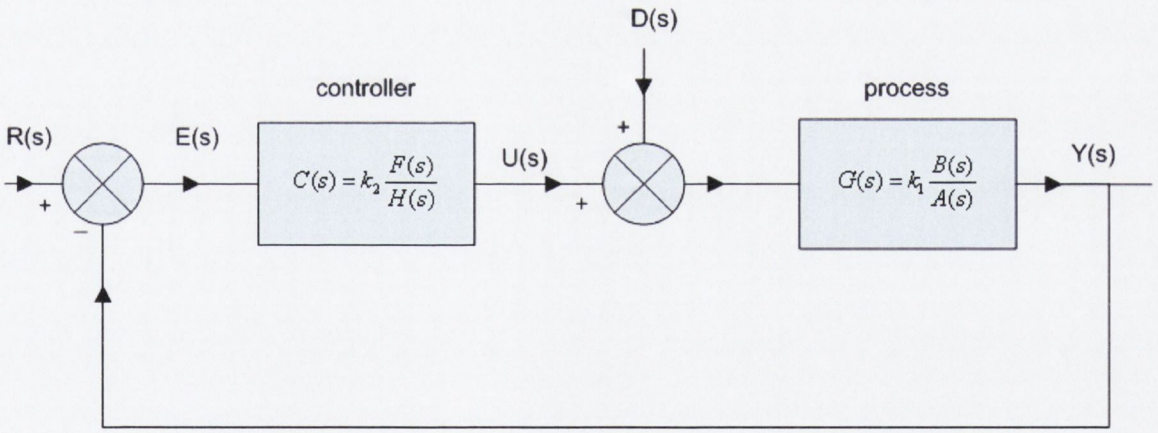


Figure 3.1 Negative feedback system where R(s) = reference input, U(s) = controller output, Y(s) = system output, D(s) = disturbance input.

The fundamental requirement of the controller C(s) is, of course, that the system described by the characteristic equation $p(s) = A(s)H(s) + k_1k_2F(s)B(s)$ must be stable. Two common secondary requirements are:

(a) the static gain between process output and reference input should be 1 i.e. $W_{yr}(0) = 1$

(b) the static gain between process output and disturbance input should be 0 i.e. $W_{yd}(0) = 0$

These two secondary requirements are met by ensuring that the controller has a pole at s = 0:

(a) $$W_{yr}(s) = \frac{G(s)C(s)}{1+G(s)C(s)} = \frac{k_1 k_2 F(s)B(s)}{A(s)H(s)+k_1 k_2 F(s)B(s)}$$

$$\text{Static gain} = W_{yr}(0) = \frac{k_1 k_2 F(0)B(0)}{A(0)H(0)+k_1 k_2 F(0)B(0)}$$

Static gain $W_{yr}(0)$ can be made equal to 1if H(0) = 0. This may be achieved if $H(s) = s\hat{H}(s)$.

(b) $$W_{yd}(s) = \frac{G(s)}{1+G(s)C(s)} = \frac{k_1 \frac{B(s)}{A(s)}}{1+\frac{k_1 k_2 F(s)B(s)}{A(s)H(s)}} = \frac{k_1 B(s)H(s)}{A(s)H(s)+k_1 k_2 F(s)B(s)}$$

$$\text{Static gain} = W_{yd}(0) = \frac{k_1 B(0)H(0)}{A(0)H(0)+k_1 k_2 F(0)B(0)}$$

Static gain $W_{yd}(0)$ can be made equal to 0 if H(0) = 0. This may be achieved if $H(s) = s\hat{H}(s)$.

So, in general, if a controller is designed with the following structure: $C(s) = \frac{k_2 F(s)}{sH(s)}$ we are assured of unity static gain between process output and reference input, and zero static gain between process output and disturbance input.

3.3 The minimum order necessary for a controller

This section shows that the order of the controller transfer function need at most be equal to the order of the transfer function of the process. This fact is used in the following sections.

Say $C(s) = k_2 \frac{F(s)}{sH(s)}$ is a controller for the process $G(s) = k_1 \frac{B(s)}{A(s)}$ where H(s), A(s), F(s), and B(s) are monic polynomials and C(s) and G(s) are proper rational functions. The characteristic equation for this system is: $P(s) = sH(s)A(s)+k_1 k_2 F(s)B(s)$.

If the orders of H(s), A(s), F(s) and B(s) are: c-1, p, c, and q ($\leq p$), respectively, then the order of P(s) is p + c and it has 1 + c + (c-1) = 2c free parameters (i.e. 1 for k_2, c for F(s), and (c-1) for H(s)). If $2c = p+c$, (that is, if c = p $\Rightarrow$ the order of F(s) = the order of A(s)), then P(s) is of order 2c and it has 2c free parameters and one has complete discretion about eigenvalue placement.

So, H(s) need never be of degree greater than c-1 and the controller need never be of degree greater than c.

A criticism that can be made of the H_∞ design process is that it can produce controllers that are of orders that are higher than necessary. For example, [24] gives a sixth order H_∞ controller for a second order

process whereas, within the context of the theory developed in this thesis, and as demonstrated in section 3.5, a second order controller suffices – this second order controller is said to be of minimum complexity.

3.4 H_∞ controllers can be fragile and of high order

Keel et al. [24] examine the stability margins of several controllers from the literature. (These controllers have been designed using H_2, H_∞, and other methods.) These controllers are found to be very fragile in the sense that tiny perturbations to the coefficients of the controller destabilise the control system. An inexactly realised controller will cause the entire control system to be unstable.

The following process for which Keel et al. [24] develop controller designs is from Doyle et al. [45]:

$$G(s) = \frac{s-1}{s^2 - s - 2} \tag{3.1}$$

- a second order, unstable process (pole at s = 2) with non-minimum phase (zero at s = 1).

In [24] a *sixth order* H_∞ controller for (3.1) is found to have a Gain Margin of 0.9992 and a Phase Margin of 0.1681 degrees. To quote from their paper "This means, roughly, that a reduction in gain of one part in one thousand will destabilise the closed loop system". They then go on to design a simple first order controller "placing closed poles on a circle of radius $\sqrt{2}$ spaced equidistantly in the left-half plane." This controller has Gain Margin 0.794 and Phase Margin -9.887 degrees. This new system can tolerate a gain reduction of 21% and the Phase Margin is improved by a factor of 60. It does not have the desirable side conditions referred to in section 3.2.

In the following section, a second order controller is designed for (3.1) from optimum root locus considerations and the dynamics of this controller compared to that of the sixth order H_∞ controller and Keel's first order controller.

3.5 Root locus based design for a controller and a new procedure for calculating system parameters

(3.1) is an unstable process (pole at s = 2) with non-minimum phase (zero at s = 1). The difficulty in designing a controller for it derives from the presence of a branch between the points (1,0) and (2,0) in the right half of the s-plane.

A controller for (3.1) can be synthesized using the root locus definition of optimum stability. The controller must impose a topology on the resulting system root locus that overcomes the presence of the branch on the positive real axis.

Referring to Figure 3.1, suppose one requires a second order stabilising controller that gives, simultaneously, (i) unity static gain between Y(s) and R(s); and (ii) zero static gain between Y(s) and D(s) and (iii) is of minimum order to allow complete discretion about eigenvalue placement. The pole at the origin of the following controller, and the four parameters (α, β, γ, k), ensure such behavior:

$$C(s) = \frac{k(s-\alpha)(s+\beta)}{s(s-\gamma)} \tag{3.2}$$

A controller with the same structure appears in section 5.4 where it is shown to be a PID controller with a low-pass filter in the D channel. The aim now is to find the values of (α, β, γ, k) which will make the system optimally stable in the root locus sense i.e. rightmost eigenvalue is as far to the left as possible.

The characteristic equation for the system in Figure 3.1 consisting of the controller in (3.2) with the process in (3.1) is:

$$p(s) = s(s-\gamma)(s+1)(s-2) + k(s-\alpha)(s+\beta)(s-1) \tag{3.3}$$

There are four system eigenvalues – two process eigenvalues and two controller eigenvalues. In the paper by Keel and Bhattacharyya [24] they assign all the eigenvalues to the same location (which therefore must be real and negative.) This is very profound for this study as it means that the resulting p(s) has the optimum stability property with respect to every system parameter - not just for α, β, γ, k - but also for the positions of the poles and zeros and the gain constant of the process. As any parameter passes through its nominal value, all others being held at theirs, the rightmost eigenvalue penetrates as deeply as possible into the left half plane. This observation follows at once from root locus considerations: if all eigenvalues are placed at the same location, that must be a common $2p^{th}$ order eigenvalue at the breakpoint on the loci for variation of each system parameter in turn, with all others held at their design values. As has been already referred to in section 1.2 assigning all eigenvalues to the same location also leads to a very convenient way of solving for the controller parameters α, β, γ, k. *It is important to appreciate that this method, though only illustrated here, generalizes readily to any order of process.*

Expanding and collecting terms in (3.3) gives:

$$p(s) = [s^4 - s^3 - 2s^2] + \gamma[-s^3 + s^2 + 2s] + k[s^3 - s^2] + k(\beta - \alpha)[s^2 - s] + k\alpha\beta\ [1 - s]$$

Or: $p(s) = p_a(s) + \gamma\ p_b(s) + kp_c(s) + k(\beta - \alpha)p_d(s) + k\alpha\beta\ p_e(s)$

Let $p(s) = (s+2)^4$ i.e. all of the eigenvalues of (3.3) are at $s = -2$ when α, β, γ and k have their nominal values.

Then, $\frac{dp}{ds} = 4(s+2)^3$; $\frac{d^2p}{ds^2} = 12(s+2)^2$; and $\frac{d^3p}{ds^3} = 24(s+2)$.

All four equations: $p(s) = 0$; $\frac{dp}{ds} = 0$; $\frac{d^2p}{ds^2} = 0$;and $\frac{d^3p}{ds^3} = 0$ are simultaneously satisfied at s = -2. As mentioned in section 1.2, this gives a very useful way to find the unknown system parameters.

First of all we will solve for $\gamma, k, k(\beta - \alpha)$, and $k\alpha\beta$. We get four simultaneous equations in these parameters and parameter groups:

$$\begin{pmatrix} p_b & p_c & p_d & p_e \\ \frac{dp_b}{ds} & \frac{dp_c}{ds} & \frac{dp_d}{ds} & \frac{dp_e}{ds} \\ \frac{d^2p_b}{ds^2} & \frac{d^2p_c}{ds^2} & \frac{d^2p_d}{ds^2} & \frac{d^2p_e}{ds^2} \\ \frac{d^3p_b}{ds^3} & \frac{d^3p_c}{ds^3} & \frac{d^3p_d}{ds^3} & \frac{d^3p_e}{ds^3} \end{pmatrix} \begin{pmatrix} \gamma \\ k \\ k(\beta-\alpha) \\ k\alpha\beta \end{pmatrix} = \begin{pmatrix} -p_a \\ -\frac{dp_a}{ds} \\ -\frac{d^2p_a}{ds^2} \\ -\frac{d^3p_a}{ds^3} \end{pmatrix} \tag{3.4}$$

At $s=-2$ equation (3.4) becomes:

$$\begin{pmatrix} 8 & -12 & 6 & 3 \\ -14 & 16 & -5 & -1 \\ 14 & -14 & 2 & 0 \\ -6 & 6 & 0 & 0 \end{pmatrix} \begin{pmatrix} \gamma \\ k \\ k(\beta-\alpha) \\ k\alpha\beta \end{pmatrix} = \begin{pmatrix} -16 \\ 36 \\ -56 \\ 54 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \gamma \\ k \\ k(\beta-\alpha) \\ k\alpha\beta \end{pmatrix} = \begin{pmatrix} 41.5 \\ 50.5 \\ 35 \\ 16 \end{pmatrix}. \text{ This gives : } \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ k \end{pmatrix} = \begin{pmatrix} 0.314463 \\ 1.00753 \\ 41.5 \\ 50.5 \end{pmatrix}$$

(These parameter values are derived using the Jeltsch-Fichera array in section 9.6.)

So the controller in (3.2) becomes:

$$C(s) = \frac{50.5(s-0.314463)(s+1.00753)}{s(s-41.5)} \tag{3.5}$$

It is interesting to note that the controller in equation (3.5) is unstable with non-minimum phase as is the process for which it was designed – given by equation (3.1). Also, the H_∞ controller from [24] for the process given by equation (3.1) is unstable.

For nominal controller and process parameters: $p(s)=(s+2)^4$. A root locus diagram for (3.3) with any one of the parameters α, β, γ or k as a variable will have a quadruple eigenvalue at the breakpoint at $s=-2$ just at the nominal value of that parameter. This means that all four eigenvalues are at $s=-2$ when the parameter has its nominal value. At that point, the rightmost eigenvalue is as deep into the left half plane as possible and the system is optimally stable in the root locus sense. At any value other than the nominal value there is at least one eigenvalue to the right of $s=-2$. It is important to note that this must also apply not only to the controller parameters but also to the process parameters.

The root loci are given in Figure 3.2, Figure 3.3, Figure 3.4, and Figure 3.5 with the parameters scaled such that the characteristic equation (3.3) becomes $p(s)=N(s)+\left(\frac{K}{K_{nom}}\right)K_{nom}M(s)$ and the quadruple eigenvalue appears at the breakpoint $s=-2$ when $\left(\frac{K}{K_{nom}}\right)=1$.

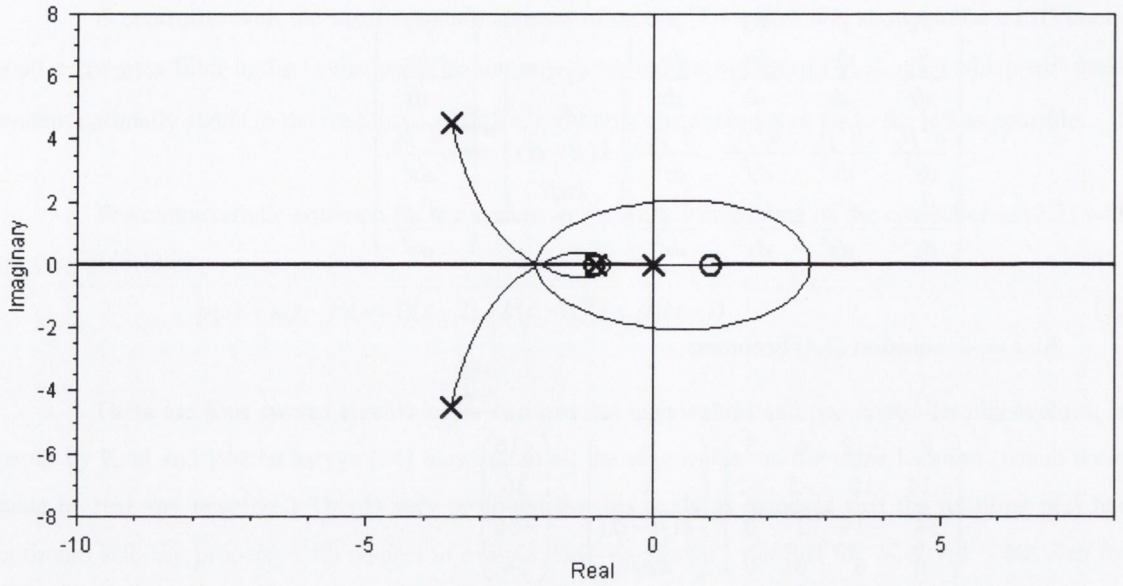


Figure 3.2 Root locus of (3.3) with α as parameter. There is a quadruple eigenvalue at the leftmost breakpoint.

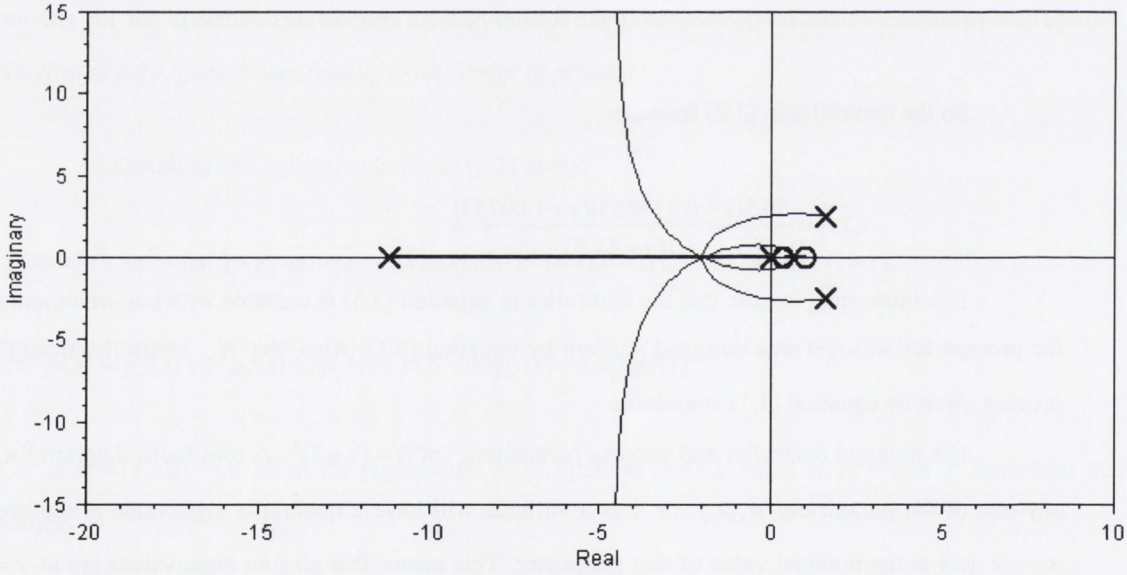


Figure 3.3 Root locus of (3.3) with β as parameter. There is a quadruple eigenvalue at the leftmost breakpoint.

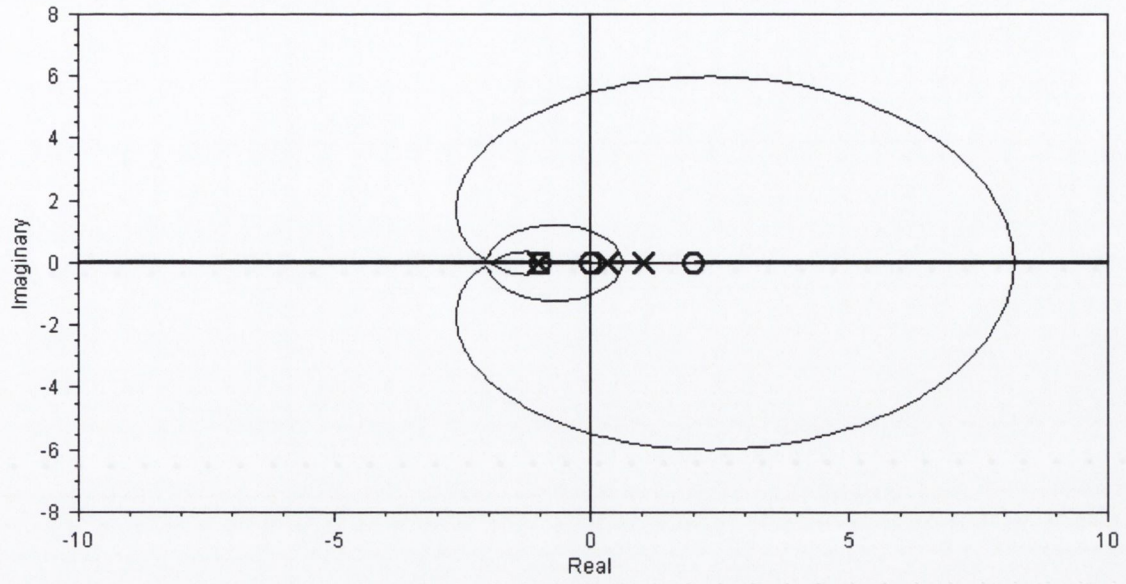


Figure 3.4 Root locus of (3.3) with γ as parameter. There is a quadruple eigenvalue at the leftmost breakpoint.

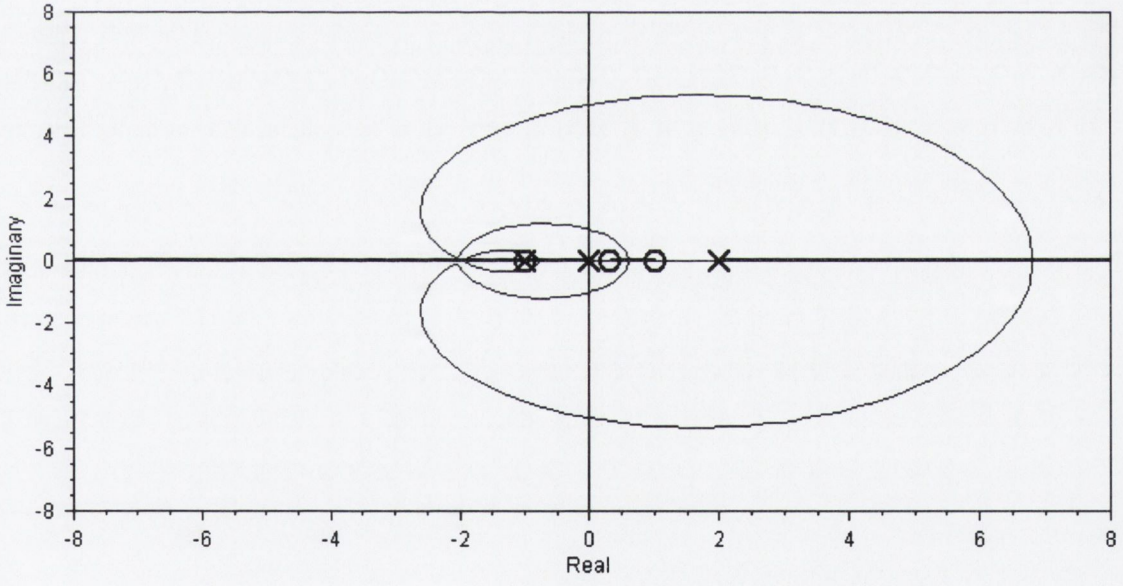


Figure 3.5 Root locus of (3.3) with k as parameter. There is a quadruple eigenvalue at the leftmost breakpoint.

The transfer function of the process $G(s) = \dfrac{s-1}{s^2 - s - 2}$ and the controller $C(s) = \dfrac{k(s-\alpha)(s+\beta)}{s(s-\gamma)}$ is:

$$G(s)C(s) = \frac{50.5(s-1)(s-0.314463)(s+1.00753)}{s(s-41.5)(s-2)(s+1)} \tag{3.6}$$

A Nyquist Plot of equation (3.6) is given in Figure 3.6.



Figure 3.6 Nyquist plot for equation (3.6)

From Figure 3.6 the Phase Margin is -6 degrees – the negative sign meaning a permitted swing of the (-1,0) point in the clockwise direction. Although modest by convention, this is an improvement by a factor of 36 on the H_∞ design. This is an improvement on the H_∞ design but not as good as the alternative first-order controller considered in [24]. However, the first order controller suggested in [24] does not have the additional features of the second order controller designed above, such as: (i) unity static gain between y and r; (ii) zero static gain between y and d; (iii) complete control over eigenvalue placement; and, crucially, (iv) independence of process parameters.

In order to illustrate the improvement in robustness of the system with the controller in (3.5) over the system with the sixth order H_∞ controller from [24] , we consider this transfer function in an error actuated loop with a pure proportional controller K, then asymptotic stability exists for $0.8715875 < K < 1.232892$. This means that the "k" in the optimally stable design (nominal value is 50.5) may vary between $43.9<k<62.1$ i.e. it may be increased by 23% or decreased by 14.7% before losing asymptotic stability.

3.6 Discussion and conclusions

A root locus based optimum stability approach was used to design a second order controller for a second order, unstable process with non-minimum phase. The design procedure developed in section 3.5 is general and is used repeatedly in this thesis. The performance of the controller using root locus was compared to a sixth order controller from the literature that was designed using H_∞ methods. The controller designed using root locus based optimum stability is of lower order than the H_∞ controller and it results in a system that is more robust and with very much enhanced performance when compared to the system using a H_∞ controller.

Chapter 4: Root locus based optimum stability and a relationship with a class of exponentially weighted performance integrals

The purpose of this chapter is to validate the use of root locus based optimum stability in the design of control systems. This is done (a) by showing in the cases of a second order process that design using root locus based optimum stability is related to the minimization of a stringent performance integral and (b) by showing that controllers designed using this method are superior to controllers designed by another popular approach. In the process, a deep relationship is explored between controllers that are designed by minimising this performance integral and "eigenvalue assigning" controllers that are designed using ideas from root locus based optimum stability. Specifically, it is shown that when the performance integral is a minimum then the rightmost eigenvalue is a far to the left as possible and vice versa. The performance integral utilized is the integral of functions of the square of the error and its derivative with exponential time weighting. The relationship between these two controller design methodologies is explored in the context of a standard second order system. An illustrative example of the design of a PI controller for an unstable is also given.

4.1 A second order system

4.1.1 Design for a second order system using root locus based optimum stability

Second order systems are of great use in control engineering as many interesting processes are of second order. In addition to this, second order systems are routinely used as a first approximation when modelling higher order systems [46].

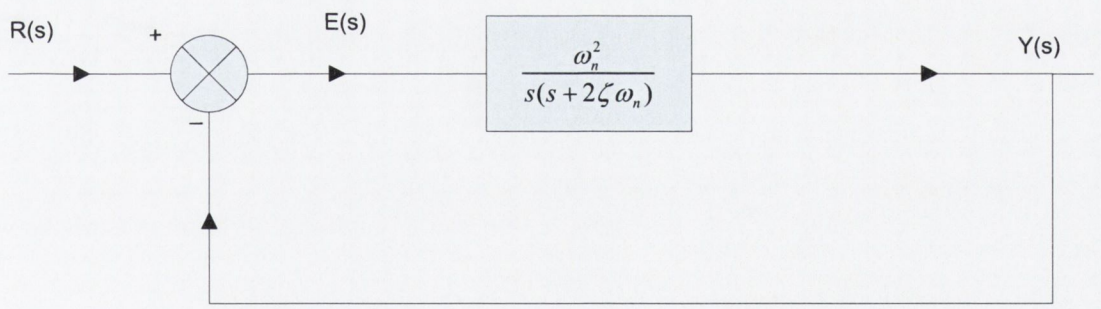


Figure 4.1 An example of a second order system

Here ω_n is the undamped natural frequency and ζ is the damping ratio. Given ω_n, root locus considerations will be used to find the value of the parameter ζ required to put the rightmost eigenvalue of the characteristic equation as far to the left as possible. The closed loop transfer function for the system in Figure 4.1 is $\frac{Y(s)}{R(s)} = W(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$. So the characteristic equation is:

$p(s) = s^2 + \omega_n^2 + 2\zeta\omega_n s = (s - j\omega_n)(s + j\omega_n) + 2\zeta\omega_n s$. The root locus for $p(s)$ with ζ as parameter is:

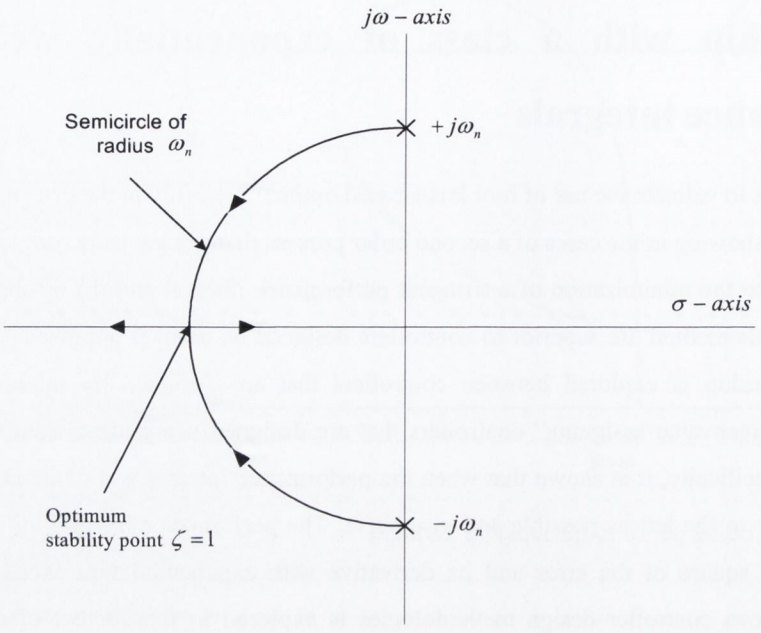


Figure 4.2 Root Locus of $p(s) = (s - j\omega_n)(s + j\omega_n) + 2\zeta\omega_n s$ with ζ as parameter

The optimum stability point is where the rightmost eigenvalue is as far to the left as possible. So $\zeta = 1$ makes this system optimally stable in the root locus sense. When $\zeta = 1$ the characteristic equation becomes $p(s) = s^2 + 2\omega_n s + \omega_n^2 = (s + \omega_n)^2$.

4.1.2 Design for a second order system by minimising a performance integral

For the system in Figure 4.1, a unit step input $R(s) = \frac{1}{s}$ gives $E(s) = \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ - the Laplace Transform of the solution to $\frac{d^2e(t)}{dt^2} + 2\zeta\omega_n \frac{de(t)}{dt} + \omega_n^2 e(t) = 0$ with $e(0) = 1$ and $\frac{de(0)}{dt} = 0$. So, if $\mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ and $x_1(t) = e(t)$, and $x_2(t) = \frac{de(t)}{dt}$ the equations of motion of the system in Figure 4.1 may be written as:

$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{pmatrix} \mathbf{x}$ or $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ with initial conditions $\mathbf{x}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Minimising the performance integral $J_{\alpha 1} = \int_0^\infty \exp(\alpha_1 t)\, \mathbf{x}^T(t) \mathbf{Q}_1 \mathbf{x}(t) dt$ with $\mathbf{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ proves not to be a satisfactory approach in this case. Minimising $J_{\alpha 1}$ leads to two possible values for α_1. The first, $\alpha_1 = +2\omega_n$, leads $J_{\alpha 1} = 1/0$ at $\zeta = 1$. The second, $\alpha_1 = 6\omega_n$, leads to an expression for $J_{\alpha 1}$ that does not have a unique minimum at $\zeta = 1$.

However, using $\mathbf{Q}_2 = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\omega_n^2} \end{pmatrix}$ in $\int_0^\infty \exp(\alpha_2 t)\,\mathbf{x}^T(t)\mathbf{Q}_2\,\mathbf{x}(t)dt$ leads to the integral $J_{\alpha 2}$:

$$J_{\alpha 2} = \int_0^\infty \exp(\alpha_2 t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt \tag{4.1}$$

Equation (4.1) proves to be an appropriate performance integralas it leads to a physically admissible value for ζ i.e. $\zeta = 1$, as well as to an expression for $J_{\alpha 2}$ that has a unique minimum at $\zeta = 1$.

We know from section 2.4.2

$$J_{\alpha 2} = \mathbf{x}^T(0)\mathbf{L}\,\mathbf{x}(0) \quad \text{where} \quad \left(\mathbf{A} + \frac{\alpha_2}{2}\mathbf{I}\right)^T \mathbf{L} + \mathbf{L}\left(\mathbf{A} + \frac{\alpha_2}{2}\mathbf{I}\right) = -\mathbf{Q}_2 \tag{4.2}$$

provided $\left(\mathbf{A} + \frac{\alpha_2}{2}\mathbf{I}\right)$ has all of its eigenvalues in the left half plane. (To simplify the notation in what follows we set $\alpha_2 = \alpha$ and $J_{\alpha 2} = J_\alpha$)

Equation (4.2) gives $J_\alpha = L_{11} = \dfrac{(2\zeta\omega_n - \frac{\alpha}{2})(2\zeta\omega_n - \alpha) + 2\omega_n^2}{2\left[(\frac{\alpha}{2})^2 - 2\zeta\omega_n(\frac{\alpha}{2}) + \omega_n^2\right]\left[2\zeta\omega_n - \alpha\right]}$. To find the value of ζ that makes J_α a minimum we calculate $\dfrac{\partial J_\alpha}{\partial \zeta} = \dfrac{\partial}{\partial \zeta}\left[\dfrac{\alpha^2 - 6\alpha\zeta\omega_n + (4 + 8\zeta^2)\omega_n^2}{-\alpha^3 + 6\alpha^2\zeta\omega_n - 4\alpha\omega_n^2 - 8\alpha\zeta^2\omega_n^2 + 8\zeta\omega_n^3}\right] = 0$

Calculating this derivative and equating to zero gives $\alpha = \pm\sqrt{4\omega_n^2(2\zeta^2 - 1)}$. From the Root Locus considerations in section 4.1.1, we have that optimum stability is obtained at $\zeta = 1$. This gives $\alpha = \pm 2\omega_n$.

The characteristic equation of $\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)$ with $\alpha = +2\omega_n$ is $s^2 + 2(\zeta - 1)\omega_n s + 2(1 - \zeta)\omega_n^2 = 0$. But this implies that, for asymptotic stability, $\zeta > 1$ and $\zeta < 1$. So $\alpha = +2\omega_n$ is not a possibility.

The characteristic equation of $\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)$ with $\alpha = -2\omega_n$ is $s^2 + 2(\zeta + 1)\omega_n s + 2(\zeta + 1)\omega_n^2 = 0$. This implies that, for asymptotic stability, $\zeta > -1$. So $\alpha = -2\omega_n$ makes the integral J_α a minimum and is a physically admissible choice.

So, for the second order system being discussed, $J_\alpha = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt$ has a unique minimum at $\alpha = -2\omega_n$ and $\zeta = 1$. This is the same value of ζ that was obtained from the root locus approach by putting the rightmost eigenvalue as far to the left as possible.

The value of $4\omega_n J_\alpha$ at $\zeta = 1$ may be calculated as: $4\omega_n J_\alpha\big|_{\zeta=1} = \dfrac{2 \times 2 \times \frac{3}{2} + 1}{4} = \dfrac{7}{4} = 1.75$.

The plot of $4J_\alpha\omega_n$ against ζ given in Figure 4.3 illustrates this minimum.

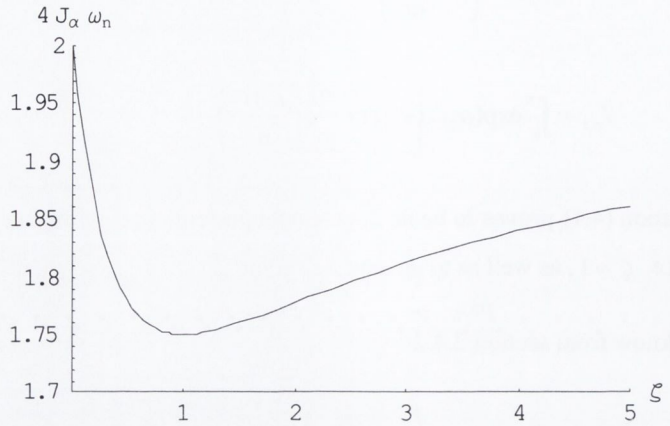


Figure 4.3 Plot of $4J_\alpha\omega_n$ against ζ showing a minimum of at $4J_\alpha\omega_n = 1.75$ at $\zeta = 1$

We can see Figure 4.3 that the minimum of $4J_\alpha\omega_n$ has a smooth, broad shape. Two observations may be made about this. First, it implies that the root locus design is robust to variations in ζ. For example, as ζ varies from 1 to 4, then $4J_\alpha\omega_n$ varies only from 1.75 to 1.85. So a 400% change in ζ leads to a less than 6 % change in J_α. A second observation that may be made about this graph is in relation to the sensitivity of the eigenvalues at the design point. Root locus based optimum stability leads to a choice of design parameters that place the eigenvalues of the system at a breakpoint in the root locus - the point of maximum eigenvalue sensitivity. The robustness of the design seen here shows that the sensitivity of the roots at the optimum stability point is irrelevant to the robustness of the system at that point. So, designing a system with maximum eigenvalue sensitivity has lead to a robust design and not to a degradation of performance.

4.2 PI controller design for the unstable process $G(s) = 1/(s-1)$

Two design methods for a PI controller for a simple first order process are considered. The first design is based on optimum stability in the root locus sense. The second design is based on finding the controller parameters that minimise a certain stringent performance integral. These two designs are seen to be identical.

4.2.1 Design for a PI controller for the process $G(s) = 1/(s-1)$ using root locus based optimum stability

The central idea in this method is to choose the system parameter that puts the rightmost eigenvalue as far to the left as possible [47]. That is, we design an eigenvalue-assigning controller. As an example of this method, Figure 4.4 shows a PI controller $C(s)$ and a simple first order process $G(s)$.

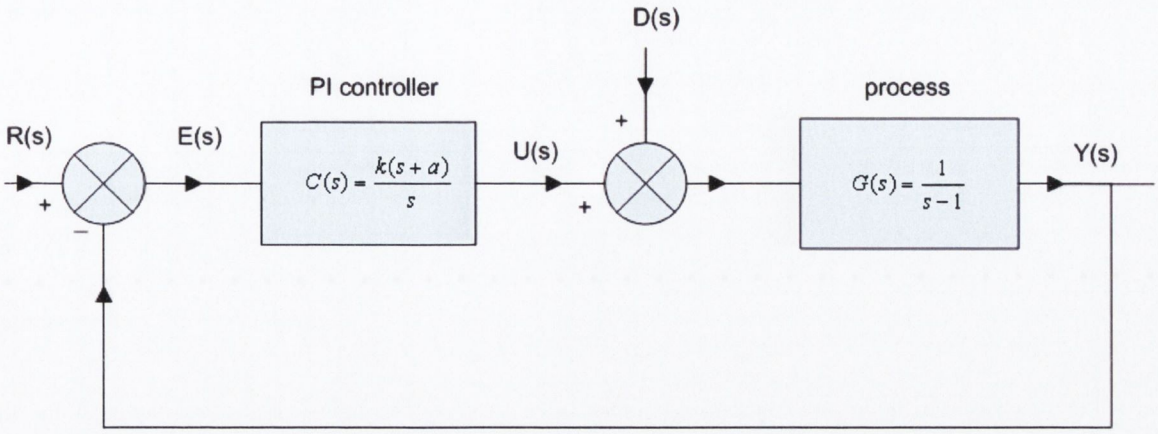


Figure 4.4 PI controller and a first order unstable process in a unitary negative feedback control system.

The root locus of the characteristic equation $p(s) = s(s-1) + k(s+a)$ for fixed $a > 0$ with varying k includes a circle of radius $r = \sqrt{a(a+1)}$ centered on the zero at $(-a, 0)$.

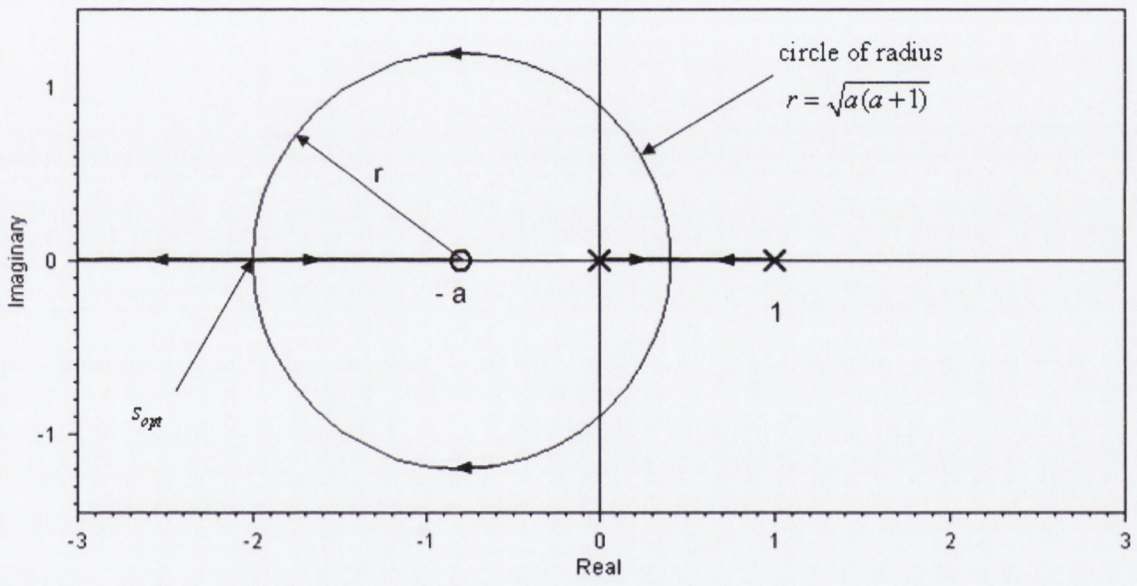


Figure 4.5 Root locus of $p(s) = s(s-1) + k(s+a)$ with k as parameter

From Figure 4.5 we can see that the rightmost eigenvalue is as far to the left as possible at the point s_{opt}. Choosing the design point at s_{opt} gives optimum stability (in the root locus sense) with respect to variations in k. At s_{opt} the characteristic equation becomes:

$$p(s) = [s + a + \sqrt{a(a+1)}]^2 = s^2 + 2[a + \sqrt{a(a+1)}]s + [a + \sqrt{a(a+1)}]^2 .$$

Equating coefficients gives: $k = (1 + 2a) + 2\sqrt{a(a+1)}$ and $a = \frac{1}{4k}(k-1)^2$.

If we choose as our design point $s_{opt} = -b$ then the characteristic equation becomes $(s+b)^2 = s^2 + 2bs + b^2$. Equating coefficients we get $k = 1 + 2b$ and $a = b^2/(1+2b)$ as our controller parameters.

For the purposes of this illustrative example we make the entirely arbitrarily choice of $s_{opt} = -2$ as the design point i.e. the rightmost eigenvalue is as far to the left as possible at $s = -2$. This gives: $(s+2)^2 = s(s-1) + k(s+a)$. This in turn gives $k = 5$ and $a = \frac{4}{5}$ are the values for (k,a) that put the rightmost eigenvalue as far to the left as possible. So $C(s) = 5 + \frac{4}{s}$ is the PI controller designed for the process $G(s) = 1/(s-1)$ using considerations of optimum stability.

4.2.2 Design for a PI controller for the process $G(s) = 1/(s-1)$ by minimising a performance integral

To get an indication of the performance of the controller in the example above we need to calculate a performance measure. We shall follow the conventional method [14, 48] and choose our state vector as $\vec{x} = \{e(t), \frac{de(t)}{dt}, \frac{d^2e(t)}{dt^2}, ...\}$ where $x_1(t) = e(t)$, the error signal.

First we need to calculate the initial conditions. Take $R(s) = 0$ and recall that the Laplace Transform of a step disturbance is $D(s) = 1/s$. Then the Laplace Transform of the error $e(t)$ is given by: $E(s) = \frac{-1}{s^2 + (k-1)s + 4k/5}$. E(s) is the Laplace Transform of the solution to: $\frac{d^2e}{dt^2} + (k-1)\frac{de}{dt} + \frac{4k}{5}e = 0$ with appropriate initial conditions. The Laplace Transform of this differential equation is: $[s^2E(s) - se(0) - \dot{e}(0)] + (k-1)[sE(s) - e(0)] + \frac{4k}{5}E(s) = 0$ So another expression for the Laplace Transform of the error is: $E(s) = \frac{se(0) + \dot{e}(0) + (k-1)e(0)}{s^2 + (k-1)s + 4k/5}$. Equating numerators in the two expressions for $E(s)$ we have: $e(0) = 0, \dot{e}(0) = -1$. Substituting $e(t) = x_1(t), \frac{de(t)}{dt} = x_2(t)$ gives $\frac{d\mathbf{x}}{dt} = \begin{pmatrix} 0 & 1 \\ -\frac{4k}{5} & -(k-1) \end{pmatrix} \begin{pmatrix} e(t) \\ \frac{de(t)}{dt} \end{pmatrix}$ with initial conditions $\mathbf{x}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$.

We want to choose a performance integral for the response of the above system to a step disturbance. Also, we want to see if this performance integral has a minimum at the parameter values chosen i.e. at $k = 5$ and $a = 4/5$.

Polynomial-time weighted performance integrals such as $\int_0^\infty (1 + a_1 t)e^2(t)dt$ and $\int_0^\infty (1 + a_1 t + a_2 t^2)e^2(t)dt$ proved unsuitable for this particular problem as they do not have a minimum in the

permitted range i.e. the range of values of k that give asymptotic stability. However, the performance integral $J_\alpha = \int_0^\infty \exp(\alpha t)\,\mathbf{x}^T(t)\mathbf{Q}_1\,\mathbf{x}(t)dt$ with $\mathbf{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ proved to be appropriate.

The integral J_α was calculated and then $\frac{dJ_\alpha}{dk} = 0$ was solved to give $\alpha = -2.4$ and $\alpha = -4$. The latter solution puts the eigenvalues of $\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)$ onto the imaginary axis and so was rejected as it would cause J_α to diverge. At $\alpha = -2.4$, $J_\alpha = \int_0^\infty \exp(\alpha t)\,\mathbf{x}^T(t)\mathbf{Q}_1\,\mathbf{x}(t)dt = \frac{5/4}{[(33/5)-k][k-(17/5)]}$ We see that J_α is a function of k so we can write $J_\alpha(k)$. Mathematica [49] gives the minimum value of $J_\alpha(k) = 0.488$ at $k = 5$. So the value of k that minimises $J_\alpha(k)$ is the same value of k that puts the rightmost eigenvalue as for to the left as possible, which in this case was chosen to be at $s = -2$. The plot of $J_\alpha(k)$ against k given below shows this unique minimum clearly.

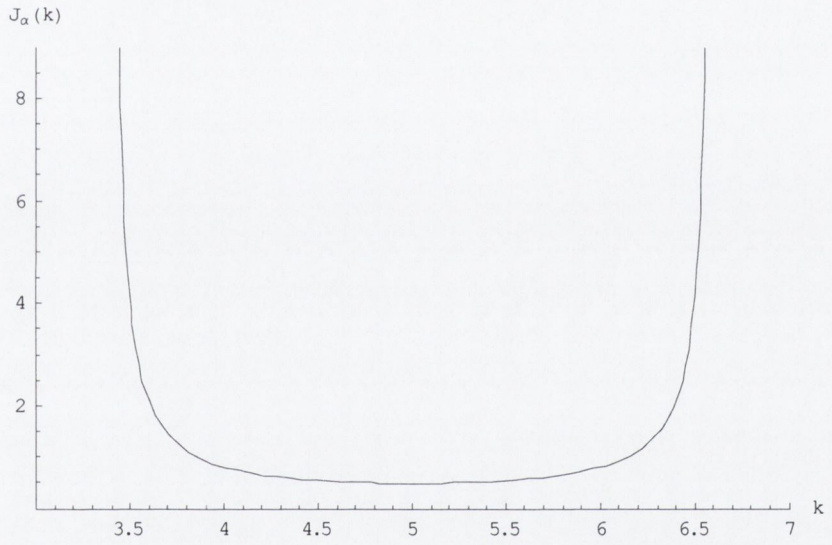


Figure 4.6 Plot of $J_\alpha(k) = \int_0^\infty \exp(-2.4t)\,\mathbf{x}^T(t)\mathbf{Q}_1\,\mathbf{x}(t)dt$ against k.

The plot in Figure 4.6 shows, that the criterion of optimality that puts the rightmost eigenvalue as far to the left as possible (i.e. using $k = 5$ as parameter value) is equivalent to minimising $J_\alpha(k) = \int_0^\infty \exp(-2.4t)\,\mathbf{x}^T(t)\mathbf{Q}_1\,\mathbf{x}(t)dt$. This is the same observation that was made for the plot of $4J_\alpha\omega_n$ against ζ in Figure 4.3. We now have reassurance of a minimised system performance measure complemented by the intuitively satisfying concept of optimum eigenvalue location.

A further observation may be made regarding this plot of $J_\alpha(k)$ against k- it has a smooth, broad shape. For example, as k varies from 4.5 to 5.5, $J_\alpha(k)$ varies between 0.54 and 0.488. So a 22% variation in the design parameter k leads to less than 10% change in $J_\alpha(k)$. As stated after Figure 3, this implies that the sensitivity of the roots at the optimum stability point is irrelevant to the robustness of the system at that point. So, designing a system with maximum eigenvalue sensitivity has again not lead to a degradation of performance.

4.2.3 Comparison between the PI controller from optimum stability and a PI controller derived from the centroid of a stability region

Figure 4.7 shows a section of the region of points in the (k,a) plane that lead to stability for the control system in Figure 4.4.

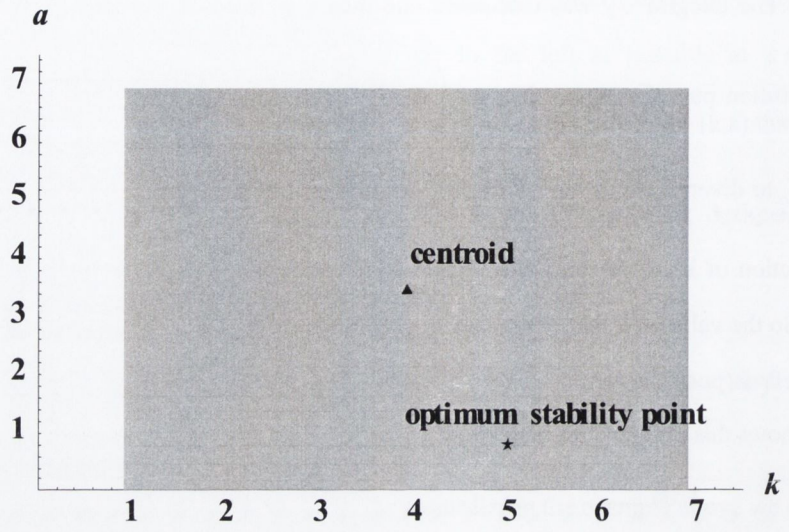


Figure 4.7 The gray region is part of the stability domain for the system in Figure 4.4.

Some authors choose the centroid of the region of stability as their design point. One drawback with selecting the centroid as the design point is that the stability region may be an infinite band so it is not possible to define the centroid. This is the case with the example considered here. However, it is possible to define a region of interest and select *its* centroid as the design point. The diagram above shows where the centroid of such a region is. It also shows the location of the optimum stability point. At first sight the location of the optimum stability point appears to be perilously close to the edge of the region of stability. However, when a comparison is made of the reference responses and the disturbance responses of the two systems it is seen that the optimum stability point behaves in a superior way as described next.

First the stability margins for the centroid system are GM=0.25, PM=42.1. Whereas the stability margins for the optimum stability system are GM=0.20, PM=69.5. The GM for the centroid design is slightly better than for the optimum stability design but the PM for the optimum stability design is far superior to that for the centroid design.

Figure 4.8, Figure 4.9, and Figure 4.10 show the optimum stability design responses (solid) and the centroid design responses (dashed) for the system in this example.

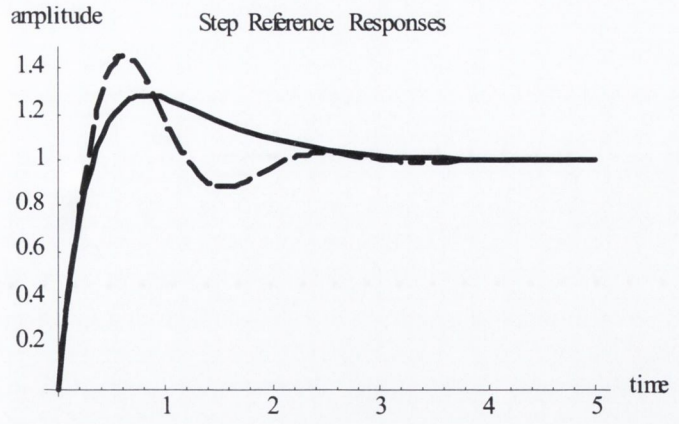


Figure 4.8 Step Reference responses for system in Figure 4.4 – optimum stability design in solid, centroid design in dashed.

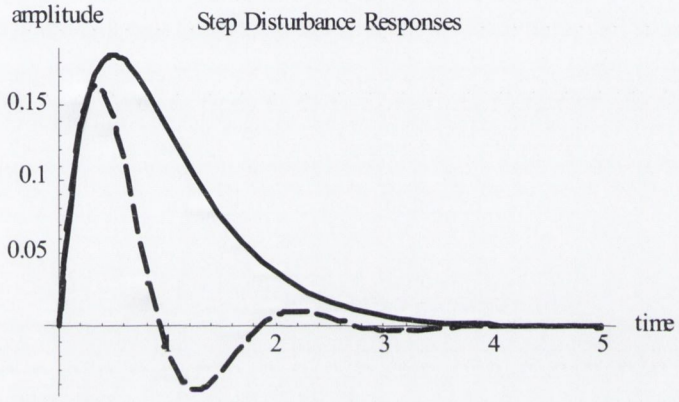


Figure 4.9 Step Disturbance responses for system in Figure 4.4 – optimum stability design in solid, centroid design in dashed.

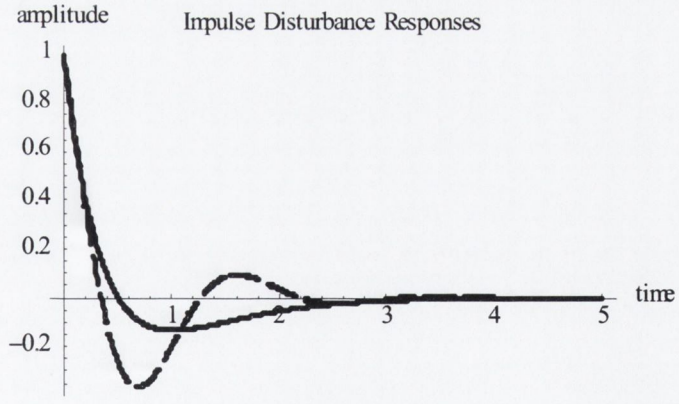


Figure 4.10 Impulse Disturbance responses for system in Figure 4.4– optimum stability design in solid, centroid design in dashed.

The disturbance responses and the reference responses show that the PI controller designed using optimum stability is superior to the PI controller that was designed by using the centroid of a stability region as the design point. The appeal of the centroid as a design point is simply its maximum distance from the stability boundary. However, this fails to take into account the degree of stability of the point. The optimum stability design point has the virtue of not only being in the stability region but of occupying a point of optimum stability.

4.3 Conclusions

Two controller design methodologies were described – one that uses root locus based optimum stability and the other based on minimising performance integrals of the form $\int_0^\infty \exp(\alpha t)\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$. The values for $\mathbf{Q}$ and for α are problem specific. Root locus is used to make the minimum value of α as large as possible. Two examples have been used to illustrate that the controllers obtained in both cases are identical. The first example was of a second order system and the second example was of a PI controller for a specific process.

In both of the examples a plot of the performance integral against the system parameter exhibits a smooth broad shape with a unique minimum point. This means that the system is robust to large variations in the design parameter as this result in small variations in the performance integral. This implies in turn that designing a system with maximum eigenvalue sensitivity does not lead to a degradation of performance.

In addition, the step reference response, the impulse disturbance response, and the step disturbance response of the optimum stability PI controller are shown to be superior to a PI controller that was designed by using the centroid of a stability region as the design point. The appeal of the centroid as a design point is simply its distance from the stability boundary. However, this fails to take into account the degree of stability of the point. The optimum stability point has the virtue of not only being in the stability region but of occupying a point of optimum stability.

4.4 Suggestions for future work

Investigate the range of controller and process combinations for which the performance integral J_α is calculable.

Chapter 5: PI and PID controller tuning using root locus based optimum stability

5.1 Introduction

Optimum stability considerations are used in this chapter to design PI and PID controllers for a variety of processes. The performance of these controllers is compared to that of controllers from the literature.

Throughout this chapter, systems of the type illustrated in Figure 5.1 are used.

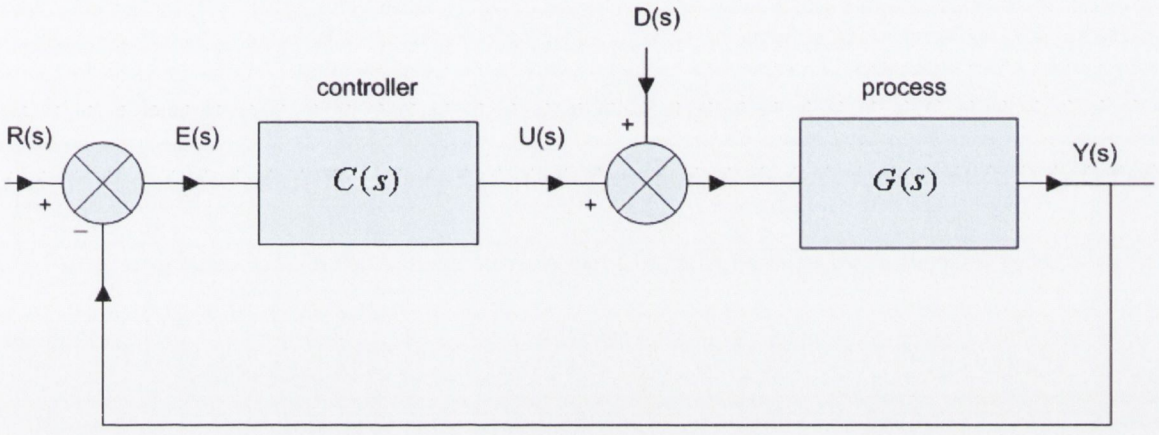


Figure 5.1 Unitary negative feedback control system.

5.2 PI controller design for multi-lag processes

5.2.1 Ziegler-Nichols tuning for PI controller for $G(s) = 1/(s+1)^4$

In 1942 Ziegler and Nichols [50] introduced tuning rules for several controllers, including the PID controller. Others [7, 22, 51-54] have also derived tuning methods for PID controllers as described in Appendix D.

It is assumed in Ziegler-Nichols designs that the closed loop system is asymptotically stable for pure proportional control for $0 < k < k_u$ (where k_u is called the ultimate sensitivity) and for $k = k_u$ the system sustains a harmonic oscillation with period T_u. The Ziegler-Nichols tuning parameters are: $k = 0.45k_u$ and $T_i = 0.83T_u$. The characteristic equation of the process $G(s) = 1/(s+1)^4$ with proportional control is $p(s) = (s+1)^4 + k$. The root locus of this equation is given in Figure 5.2.

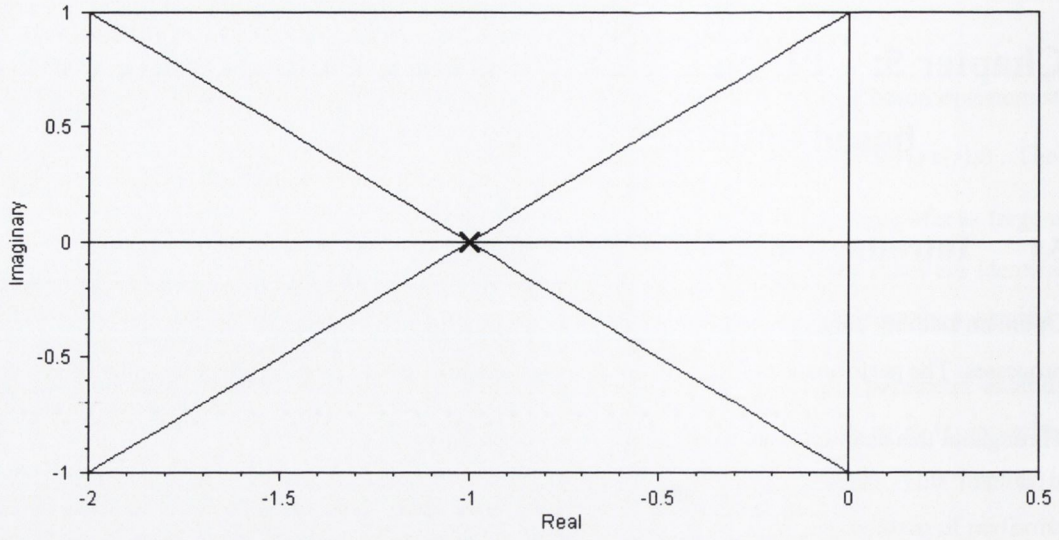


Figure 5.2 Root locus of $p(s) = (s+1)^4 + k$ with k as parameter.

From the root locus in Figure 5.2 one may see: $\omega_u = 2\pi / T_u = 1 \Rightarrow T_u = 2\pi$. Also, $k_u = \left(\sqrt{2}\right)^4 = 4$. The Ziegler-Nichols parameters are therefore $k = 0.45(4) = 1.8$ and $T_i = 0.83(2\pi) = 5.215$.

The Ziegler-Nichols controller is: $C(s) = 1.8\left(1 + \dfrac{1}{5.215s}\right) = 1.8\left(\dfrac{5.215s+1}{5.215s}\right)$ and the characteristic equation for the system in Figure 5.1 is $p(s) = 5.215s(s+1)^4 + 1.8(5.215s+1)$ (5.1)

Gain and Phase Margins for this system are 1.8 and 36^0.

5.2.2 Fine-tuning of the Ziegler-Nichols PI controller for the process $G(s) = 1/(s+1)^4$ using root locus

Equation (5.1) may be rewritten with a fine-tuning term k included as follows:

$$p(s) = 5.215s(s+1)^4 + k1.8(5.215s+1) \tag{5.2}$$

A root locus diagram equation (5.2) is given in Figure 5.3.

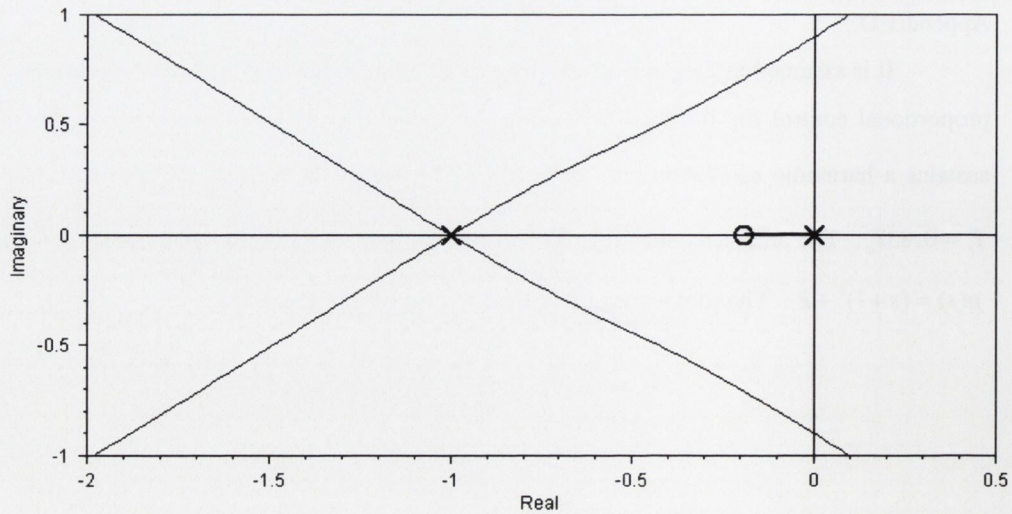


Figure 5.3 Root locus for $p(s) = 5.215s(s+1)^4 + k1.8(5.215s+1)$ with k as parameter.

It can be seen from Figure 5.3 that optimum stability in the root locus sense occurs when the real, left-moving, negative root has the same value of σ as the two right-going, complex roots. It is difficult to estimate when this happens from the diagram, but it may be found by solving p(s) for different values of k and comparing the roots. When k = 0.947, Mathematica's NSolve command gives the following roots of p(s): -0.1461, -0.1463 ± j 0.752, -1.78 ± j 0.798.

So, the rightmost eigenvalue of the Ziegler-Nichols design is as far to the left as possible when k = 0.947. Substituting this value back into p(s) gives:

$$p(s) = 5.215s(s+1)^4 + 1.7(5.215s+1) \tag{5.3}$$

with Gain Margin = 1.9 and Phase Margin =41° – improvements on the standard Ziegler-Nichols design.

5.2.3 PI controller for the process $G(s) = 1/(s+1)^4$ from root locus based optimum stability

A PI controller $C(s) = k(s+a)/s$ for the process $G(s) = 1/(s+1)^4$ may be designed using root locus based optimum stability. The characteristic equation for the system is:

$$p(s) = s(s+1)^4 + k(s+a) \tag{5.4}$$

The evolution of the root locus topology of (5.4) with a shows that a critical value of a results in a breakpoint in the root locus at a triple root. Figure 5.4, Figure 5.5, and Figure 5.6 show the root locus for (5.4) with k as parameter and with $a = 0.4$, $a = 0.64$ (the critical value), and $a = 0.8$.

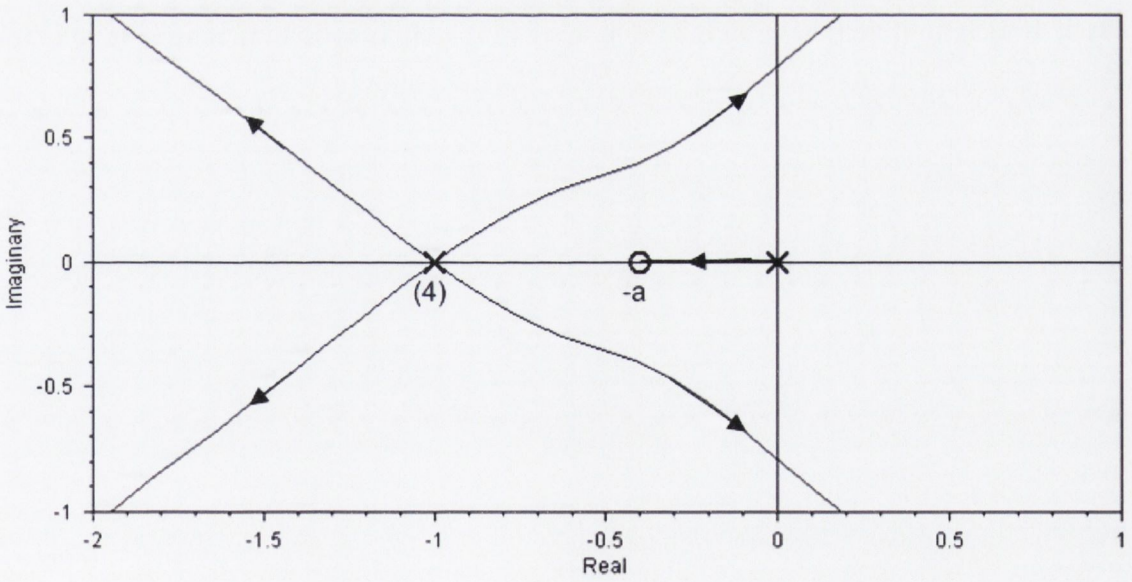


Figure 5.4 Root locus for (5.4) with k as parameter and with $a = 0.4$, which is below the critical value. The system has not yet reached a state of optimum stability.

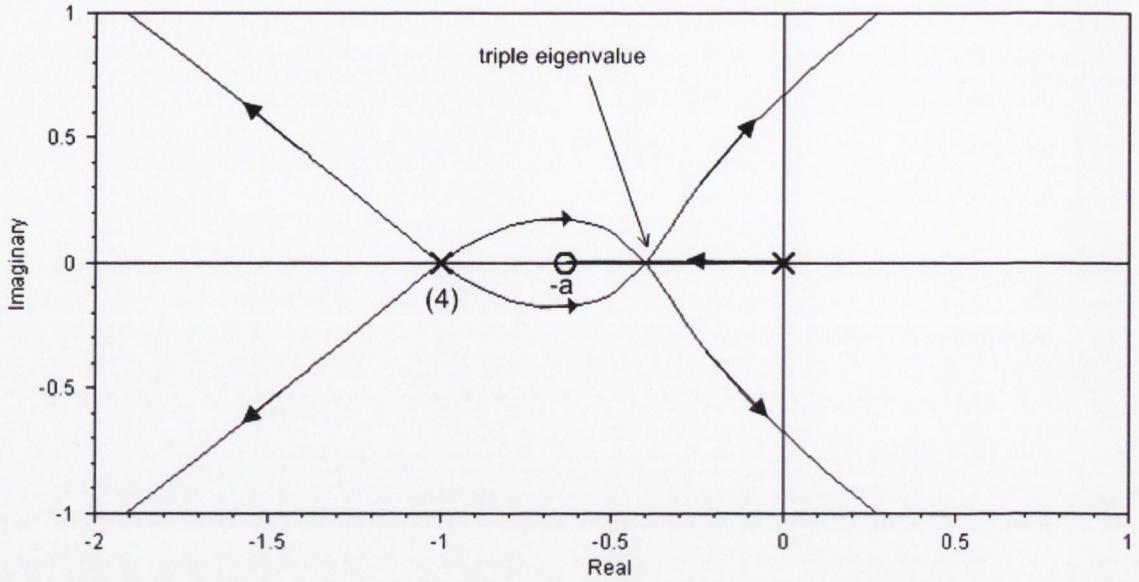


Figure 5.5 Root locus for (5.4) with k as parameter and with $a = 0.64$ showing the breakpoint at a triple eigenvalue. This critical value of a, along with the corresponding value for k, allows us to design the PI controller that leads to optimum stability.

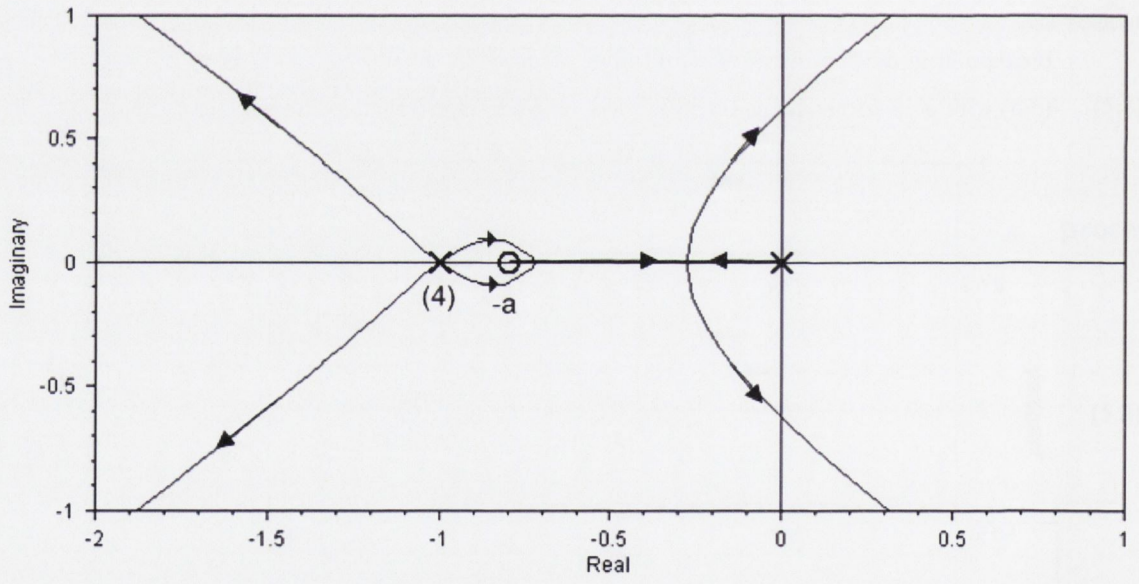


Figure 5.6 Root locus for (5.4) with k as parameter and with $a = 0.8$. The system has moved away from a state of optimum stability.

We know from section 1.2 that at the triple eigenvalue in Figure 5.5 the following equations hold in addition to (5.4):

$$p'(s) = (s+1)^4 + 4s(s+1)^3 + k = 0 \tag{5.5}$$

$$p''(s) = 8(s+1)^3 + 12s(s+1)^2 = 0 \tag{5.6}$$

Solving (5.6) for s gives $s = -1$ or $s = -\frac{2}{5}$ - which is the location of the triple eigenvalue.

Substituting $s = -\frac{2}{5}$ into (5.5) gives $k = 0.216$.

Substituting $s = -\frac{2}{5}$ and $k = 0.216$ into (5.4) gives $a = 0.64$.

So the PI controller that has been designed for the process $G(s) = 1/(s+1)^4$ using considerations of optimum stability is $C(s) = \frac{0.216(s+0.64)}{s} = 0.216 + \frac{0.138}{s}$. (These controller parameter values are also obtained using a Lyapunov matrix equation method described in section 8.4.2 and a Jeltsch-Fichera array method in 9.8.)

5.2.4 Other methods for tuning PI controllers

Ho et al. [52, 55] write down the following equations from the definitions of gain and phase margins:

$$\arg\left[C(j\omega_p)G(j\omega_p)\right] = -\pi \qquad A_m = \frac{1}{\left|C(j\omega_p)G(j\omega_p)\right|}$$

$$C(j\omega_g)G(j\omega_g) = 1 \qquad \phi_m = \arg\left[C(j\omega_g)G(j\omega_g)\right] + \pi$$

where A_m and ϕ_m are designer specified gain and phase margins; ω_p is the frequency at which the Nyquist curve has a phase of $-\pi$ (i.e. the phase crossover frequency); and ω_g is the frequency at which the Nyquist curve has an amplitude of 1 (i.e. the gain crossover frequency). When a PI controller is designed using this approach one obtains four highly nonlinear equations in four unknowns (ω_p, ω_g, T, and k). These equations may be solved numerically or approximate analytic solutions are also possible. These approximate analytic solutions depend on noticing that: $\arctan x \approx \frac{1}{4}\pi x \quad (|x| \le 1)$ and $\arctan x \approx \frac{1}{2}\pi - \frac{\pi}{4x} \quad (|x| > 1)$. Ho's controller for the process $G(s) = 1/(s+1)^4$ is given in Table 5.1.

Fung et al. [56] give a method for tuning PI controllers using the same starting point as Ho et al. [52, 55] but solve the equations by graphing them and reading the intersections off the graph. The controller from [56] for the process $G(s) = 1/(s+1)^4$ is given in Table 5.1.

5.2.5 Comparison of PI controllers for the process $G(s) = 1/(s+1)^4$

The four controller design methods described in sections 5.2.1 to 5.2.4 are now compared. Table 5.1 shows that the controller designed using optimum stability gives improved gain margin and phase margin compared to controllers designed using Ziegler Nichols tuning, Fung's method, or Ho's method.

year	Design	Controller	Gain Margin (absolute)	Phase Margin
1942	fine-tuned Ziegler-Nichols [50]	$C(s) = 1.7 + \frac{0.326}{s}$	1.9	41°
1995	Ho et al. [52, 55]	$C(s) = 1.112 + \frac{0.287}{s}$	2.6	61°
1998	Fung et al. [56]	$C(s) = 0.848 + \frac{0.297}{s}$	3	60°
2006	Root locus based optimum stability	$C(s) = 0.216 + \frac{0.138}{s}$	7.2	71°

Table 5.1 Comparison of the performance of four PI controllers designed for the process $G(s) = 1/(s+1)^4$.

Reference response, step disturbance response, and impulse disturbance response curves are given in Figure 5.7, Figure 5.8, and Figure 5.9.

Another PI controller for the process $G(s) = 1/(s+1)^4$ that could have could have been included in Table 5.1 is $C(s) = 0.76 + 0.36/s$ from Panagopoulos and Åström [57]. This controller and process lead to a system with Gain Margin of 2.73 and Phase Margin of 48.7° and time domain performance comparable with the systems proposed in [50, 52, 55, 56] so it was not included.

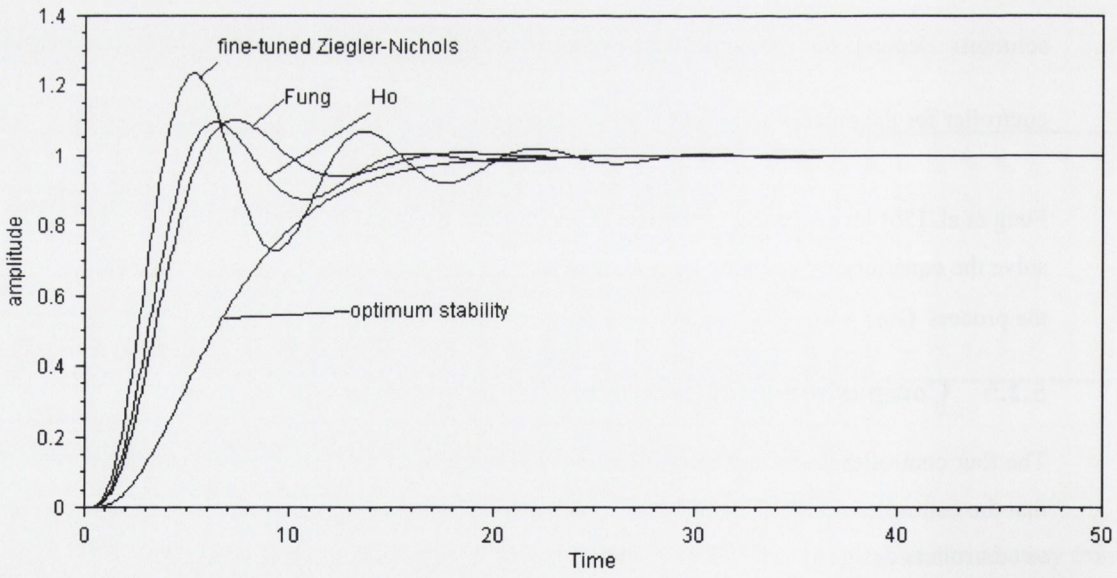


Figure 5.7 Step reference responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response, with no overshoot and the fastest settling time.

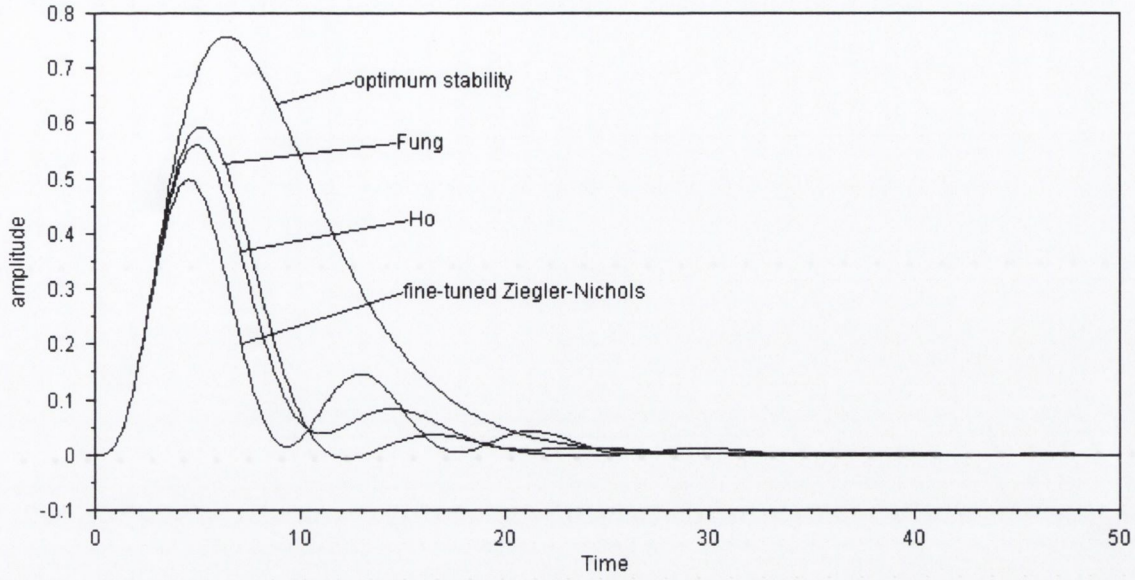


Figure 5.8 Step disturbance responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response and the fastest settling time, but it has the most overshoot.

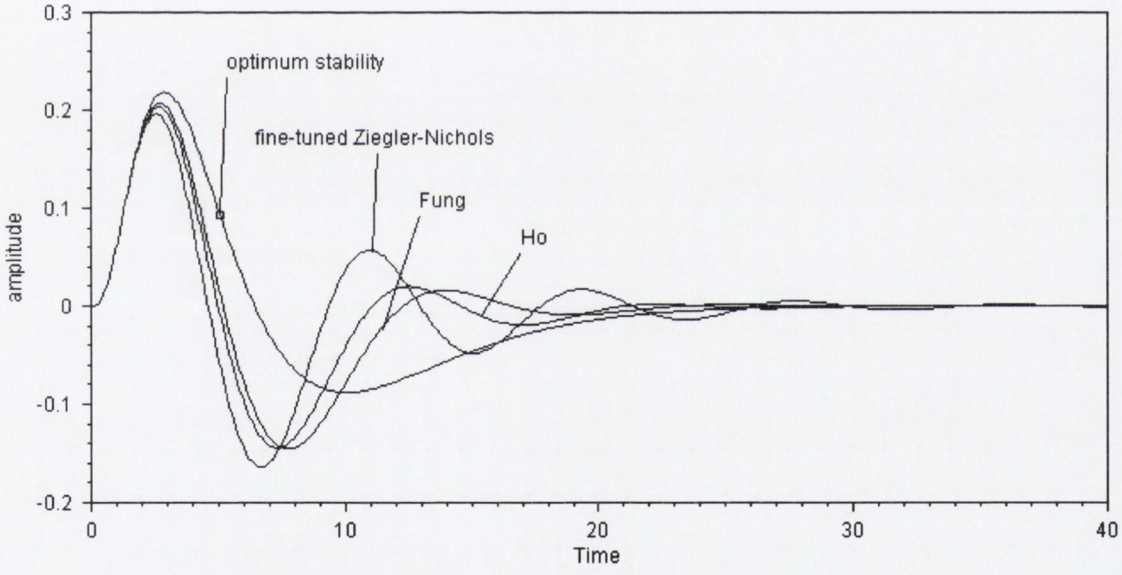


Figure 5.9 Impulse disturbance responses for systems using the four controllers in Table 5.1. The controller designed using root locus based optimum stability results in the smoothest response, with greatest overshoot but the fastest settling time.

From Figure 5.7, Figure 5.8, and Figure 5.9 we can see that the Ho, Fung, and Ziegler Nichols designs give more oscillatory responses (which, of course, is built in to the Ziegler Nichols method [50]) but in the case of the disturbance responses they give reduced initial overshoot. This is an illustration of the trade-off between stability and performance referred to in section 1.2. Table 5.1 shows that the controller designed from optimum stability results in very much improved gain margin and phase margin values.

5.2.6 Generalised root locus based optimum stability approach for designing a PI controller for the multi-lag process $G(s) = k_2/(s+b)^m$

This section gives a derivation of the parameters for a PI controller $C(s) = k_1(s+a)/s$ for the multi-lag process $G(s) = k_2/(s+b)^m$ using the optimum stability design approach.

The characteristic polynomial for the system is $p(s) = s(s+b)^m + k_1k_2(s+a) = 0$. If we take $k = k_1k_2$ we get:

$$p(s) = s(s+b)^m + k(s+a) = 0 \tag{5.7}$$

Differentiating (5.7) gives:

$$p'(s) = (s+b)^{m-1}(b + s(m+1)) + k = 0 \tag{5.8}$$

Differentiating (5.8) gives:

$$p''(s) = m(s+b)^{m-2}(2b + s(m+1)) = 0 \tag{5.9}$$

(5.9) gives:
$$s = \frac{-2b}{(m+1)} \tag{5.10}$$

(5.10) is the location of the triple eigenvalue and the point of optimum stability.

Substituting (5.10) into (5.8) gives:
$$k = b^m\left(\frac{m-1}{m+1}\right)^{m-1} \tag{5.11}$$

Substituting (5.10) and (5.11) into (5.7) gives:

$$a = \frac{4bm}{(m+1)^2} \tag{5.12}$$

Using (5.11) and (5.12) one may design a PI controller $C(s) = k_1(s+a)/s$ for the process $G(s) = k_2/(s+b)^m$ and thereby obtain a system with optimum stability.

5.3 PID controller designs for multi-lag processes

The performance of PID controllers for the processes $G(s) = 1/(s+1)^4$ and $G(s) = 1/(s+1)^5$ from the literature is compared to that of PID controllers designed using ideas from optimum stability.

5.3.1 Ziegler-Nichols tuning for a PID controller for the process $G(s) = 1/(s+1)^4$

The PID controller has the form:

$$C(s) = k\left(1 + \frac{1}{sT_i} + sT_d\right)$$

where k = gain of proportional channel; T_d = the derivative action time; T_i = the integral action time.

Say the transfer function of the process to be controlled is $G(s) = \frac{1}{(s+1)^4}$ - one of the proposed benchmark processes from Åström and Hägglund [58]. The Ziegler-Nichols tuning parameters are: $k = 0.6k_u$, $T_i = 0.5T_u$, and $T_d = 0.25T_i = 0.125T_u$

The root locus diagram in Figure 5.2 gives: $\omega_u = 1 \Rightarrow T_u = 2\pi \Rightarrow T_i = \pi$. Also $T_d = \pi/4$. And $k_u = 4$.

The Ziegler-Nichols controller designed from ultimate sensitivity is:

$$C(s) = 0.6*4\left\{1+\frac{1}{s\pi}+\frac{\pi}{4}s\right\} = 2.4\left\{\frac{4\pi s+4+\pi^2 s^2}{4\pi s}\right\} = 1.885\frac{\left(s+\frac{2}{\pi}\right)^2}{s}$$

The characteristic equation of this controller with $G(s) = 1/(s+1)^4$ is

$$p(s) = s(s+1)^4 + 1.885\left(s+\frac{2}{\pi}\right)^2 \tag{5.13}$$

The Ziegler-Nichols PID controller constrains the parameters in a particular way. In Ziegler–Nichols design we have $T_d = T_i/4$ so there are only two independent parameters - k and T_i. However, this is the structure required for root locus based design as we can fix one parameter and study root locus topology for different values of the other parameter. Root locus based design would not be possible with three parameters. An approach based on the Lyapunov matrix equation as described in section 1.4 or the Jeltsch-Fichera array as described in 9.8 would be required.

5.3.2 Fine-tuning the Ziegler-Nichols PID controller for the process $G(s) = 1/(s+1)^4$ using root locus

Fine-tuning of the Ziegler-Nichols controller is possible using root locus. Introducing a fine-tuning variable, k, into (5.13) gives:

$$p(s) = s(s+1)^4 + k(1.885)\left(s+\frac{2}{\pi}\right)^2 \tag{5.14}$$

The root locus of (5.14) with k as parameter is given in Figure 5.10.

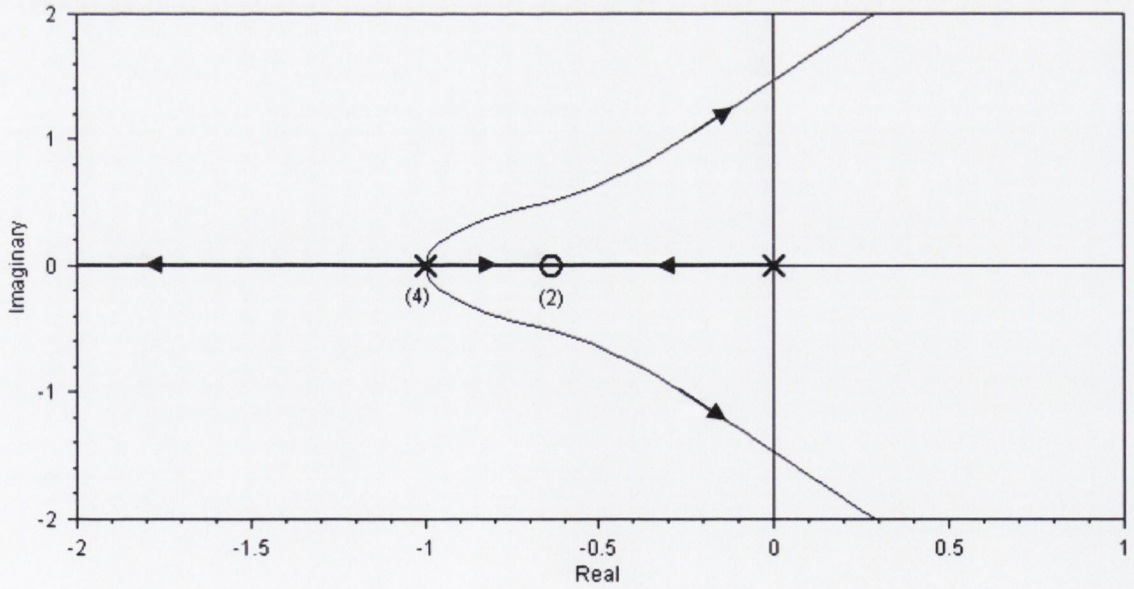


Figure 5.10 Root locus of $p(s) = s(s+1)^4 + k(1.885)(s+2/\pi)^2$ with k as parameter

It is possible to estimate graphically that the rightmost eigenvalue of the Ziegler-Nichols design is as far to the left as possible when $k = 0.566$. Using $k = 0.566$ in (5.14) gives a fine-tuned Ziegler-Nichols controller $C(s) = 1.067\frac{(s+2/\pi)^2}{s}$ and the improved system response characteristics shown in section 5.3.3.

5.3.3 Dynamics of Ziegler-Nichols and fine-tuned Ziegler-Nichols controllers

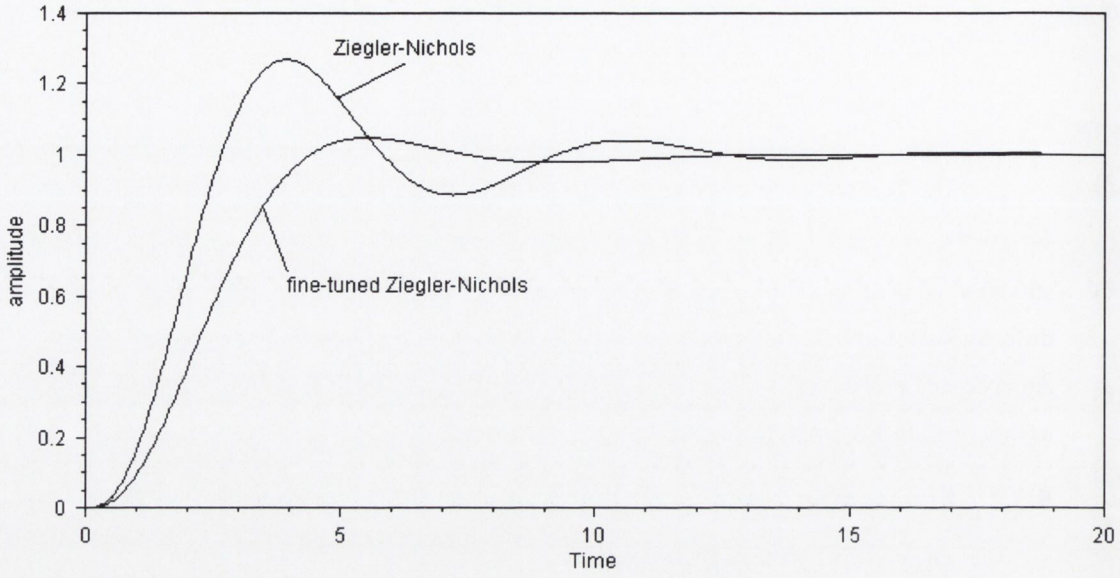


Figure 5.11 Step reference responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has less overshoot and is smoother than the Ziegler Nichols system.

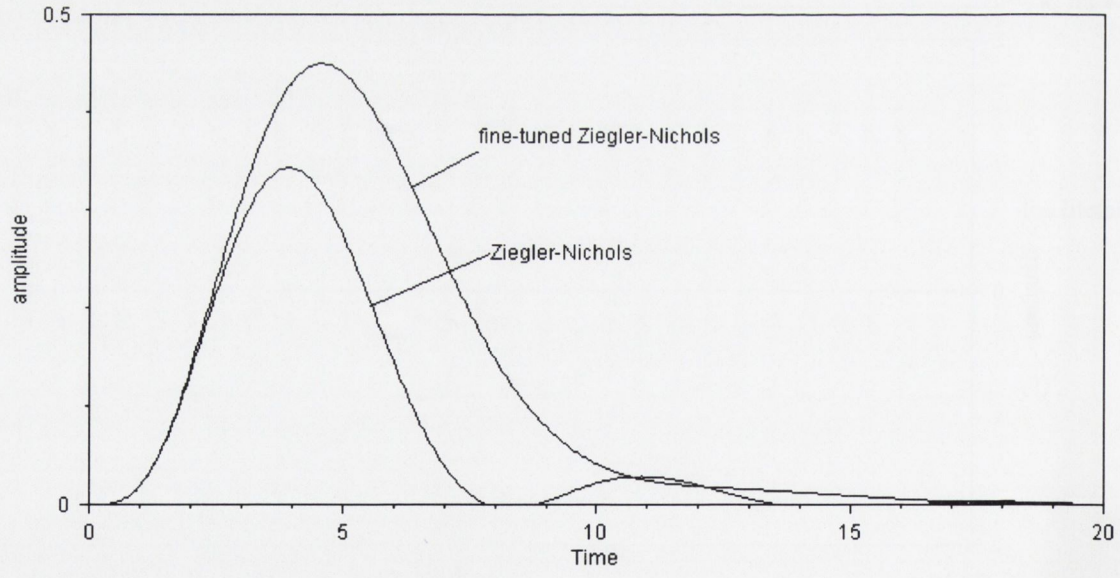


Figure 5.12 Step disturbance responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has larger overshoot but is smoother than the Ziegler Nichols system

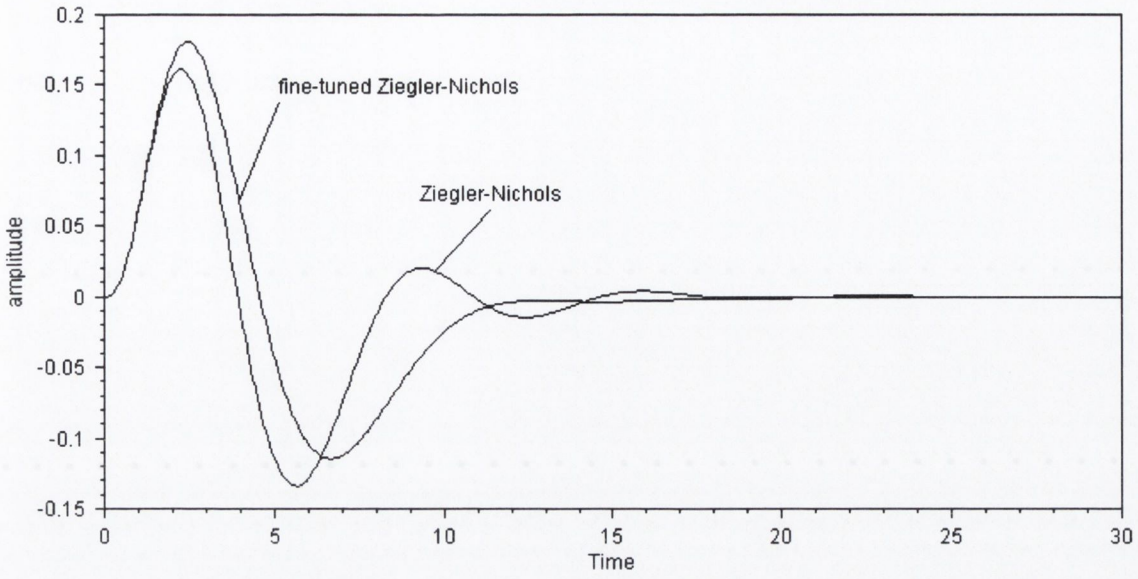


Figure 5.13 Impulse disturbance responses for Ziegler Nichols and fine-tuned Ziegler Nichols systems. The fine-tuned Ziegler Nichols system has slightly larger overshoot but is smoother than the standard Ziegler Nichols system.

5.3.4 Discussion of the Ziegler-Nichols and fine-tuned Ziegler-Nichols controllers

Comparing the response curves given in Figure 5.11, Figure 5.12, and Figure 5.13 we can see that he fine-tuned Ziegler Nichols controller produces more desirable behavior than the standard Ziegler-Nichols design. Also the Gain Margin (GM) of the Ziegler Nichols system is only 3 whereas the GM of the fine-tuned Ziegler Nichols system is 5.4. The Phase Margin (PM) of the Ziegler Nichols system is only 42° whereas the PM of the fine-tuned Ziegler Nichols system is 63.7°.

5.3.5 Generalised root locus based optimum stability approach for designing a PID controller for the process $G(s) = k_2/(s+b)^m$

Consider the restricted class of PID controllers $C(s) = \dfrac{k_1(s+a)^2}{s}$ with $T_i = \dfrac{2}{a}$ and derivative action time $T_d = \dfrac{1}{2a}$. The characteristic equation for this PID controller and the process $G(s) = \dfrac{k_2}{(s+b)^m}$ is:

$$p(s) = s(s+b)^m + k(s+a)^2 . \tag{5.15}$$

where $k = k_1 k_2$ and $b > 0$.

To show illustrative examples of typical root loci of (5.15) we anticipate the root-locus-based optimum stability design method described next to say that $k_2 = 1,\ b = 1,\ m = 8$ gives $k_1 = 0.287, a = 0.545$; and $k_2 = 1,\ b = 1,\ m = 9$ gives $k_1 = 0.3, a = 0.5$.

The root locus with respect to k of (5.15) with m even $(m = 8), k_2 = 1,\ b = 1$ is given in Figure 5.14.

The root locus with respect to k of (5.15) with m odd $(m = 9), k_2 = 1,\ b = 1$ is given in Figure 5.15.

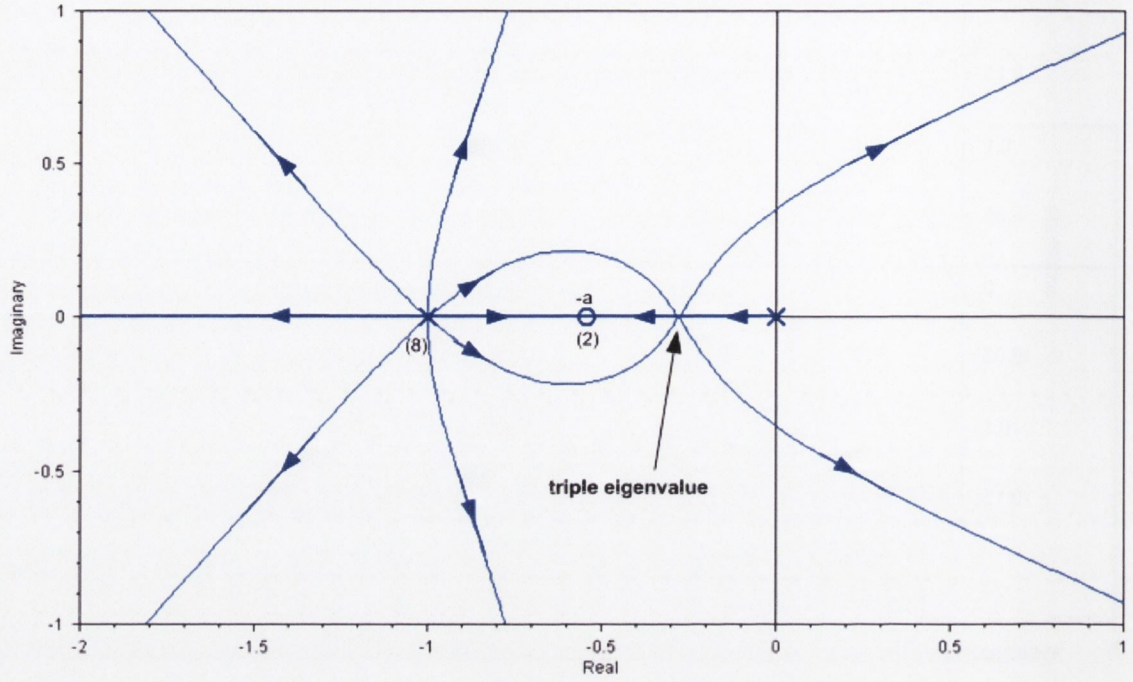


Figure 5.14 Root locus of equation (5.15) with $m = 8$, with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense.

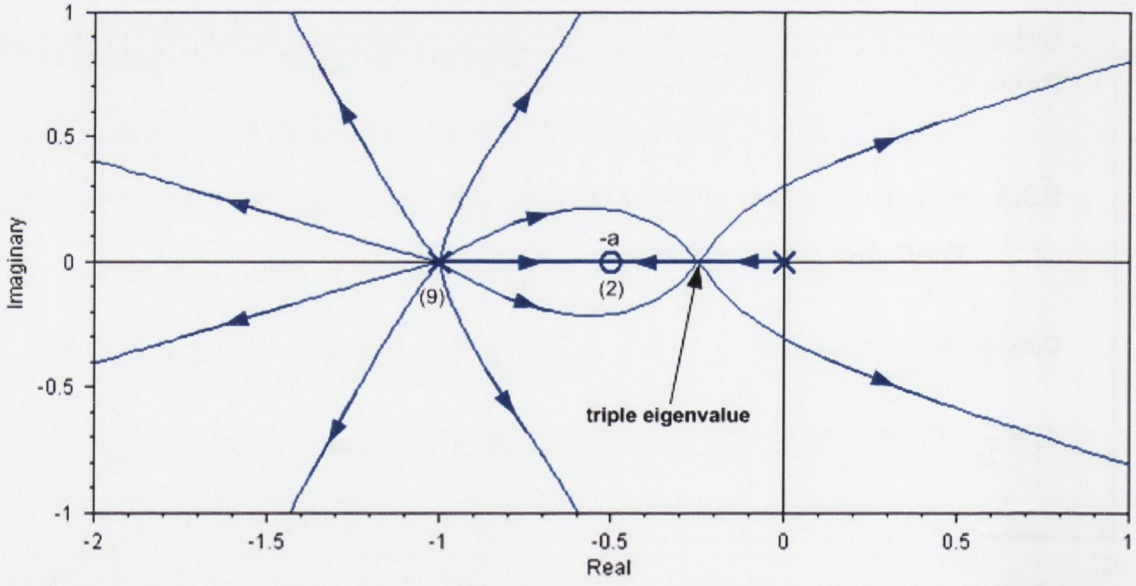


Figure 5.15 Root locus of equation (5.15) with $m = 9$, with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense.

We can see from Figure 5.14 and Figure 5.15 that for even and odd values of m, the root locus of (5.15) has a breakpoint at a triple eigenvalue. These diagrams also illustrate the property $b > a$. I will now derive a general method for calculating the parameters for a PID controller $C(s) = k_1(s+a)^2/s$ for the process $G(s) = k_2/(s+b)^m$ that leads to a system with optimum stability.

In general, if there is a triple eigenvalue at q in the root locus then the characteristic equation must factorize as $p(s) = f(s)(s+q)^3$ at that point $s = -q$ where $f(s)$ is a polynomial of order $(m-2)$. So we have $p(-q) = 0$, $p'(-q) = 0$ and $p''(-q) = 0$. This gives at $s = -q$:

$$p(s) = s(s+b)^m + k(s+a)^2$$

$$= N_1(s) + k(s+a)^2 = 0 \text{ at } s = -q$$

$$p'(s) = N_2(s) + 2k(s+a) = 0 \text{ at } s = -q .$$

$$p''(s) = N_3(s) + 2k = 0 \text{ at } s = -q .$$

$$p''(s) = 0 \Rightarrow k = -\frac{N_3(s)}{2} \tag{5.16}$$

$$p'(s) = 0 \Rightarrow N_2(s) - N_3(s)(s+a) = 0$$

$$\Rightarrow \left(\frac{N_2(s)}{N_3(s)}\right)^2 = (s+a)^2 \tag{5.17}$$

$$p(s) = 0 \Rightarrow N_1(s) - \frac{N_3(s)}{2}(s+a)^2 = 0$$

$$\Rightarrow \frac{2N_1(s)}{N_3(s)} = (s+a)^2 \tag{5.18}$$

Equating (5.17) and (5.18) gives $\left(\frac{N_2(s)}{N_3(s)}\right)^2 = \frac{2N_1(s)}{N_3(s)} \Rightarrow (N_2(s))^2 = 2N_1(s)N_3(s)$

$$\Rightarrow (N_2(s))^2 - 2N_1(s)N_3(s) = 0 \tag{5.19}$$

where $N_1(s) = s(s+b)^m$, $N_2(s) = ((m+1)s+b)(s+b)^{m-1}$, and $N_3(s) = m((m+1)s+2b)(s+b)^{m-2}$.

In general, equation (5.19) becomes

$$(m^2-1)s^2 + 2b(m-1)s - b^2 = 0 \tag{5.20}$$

As an example of the use of this method we design a PID controller for the process $G(s) = 1/(s+1)^9$. Equation (5.20) gives:

$$80s^2 + 16\,bs - b^2 = 0 . \tag{5.21}$$

With $b = 1$, equation (5.21) gives $s = -0.25$ or $s = 0.05$. Since we are considering a stable system the negative breakpoint is the relevant one. Substituting $k_2 = 1$, $b = 1$, and $s = -0.25$ into equation (5.16) gives $k_1 = 0.3003$. We can now find $a = 0.5$ from equation (5.17) or (5.18). The resulting PID controller designed using root locus based optimum stability for the process $G(s) = 1/(s+1)^9$ is:

$$C(s) = \frac{0.3003(s+0.5)^2}{s} \tag{5.22}$$

5.3.6 PID controller for $G(s)=1/(s+1)^4$ derived by various methods in the literature

The PID controller $C(s)=\dfrac{0.2837(s+0.8319)^2}{s}$ was derived using the algorithm described in section 5.3.5.

The PID controller $C(s)=\left(0.778+\dfrac{0.289}{s}+0.556\right)$ was derived using the algorithm described in section 1.4. This is a three term PID controller and it places the rightmost eigenvalues of the system at $-0.55\pm j0.18$ and $-0.55\pm j0.41$.

Ang et al. [59] discuss PID controllers designed by the software package PIDeasy [60]. They use PIDeasy to design the PID controller $C(s)=0.83\left(1+\dfrac{2.61}{s}+0.43s\right)$ for the process $G(s)=1/(s+1)^4$.

Stefani et al. [61] describe the two CHR methods for tuning PID controllers that were published in 1952 by Chein, Hrones and Reswick [62]. The CHR methods uses the unit step response of the open-loop process to estimate values for two time parameters - T_u and T_g. The "overdamped" CHR method is used here in preference to the "20% overshoot " method as the former leads to smoother time responses and better robustness margins – PM of 42.4° and GM of 2.7 compared to PM of 21.1° and GM of 1.7.

The open-loop unit step response for the process $1/(s+1)^4$ is given in Figure 5.16.

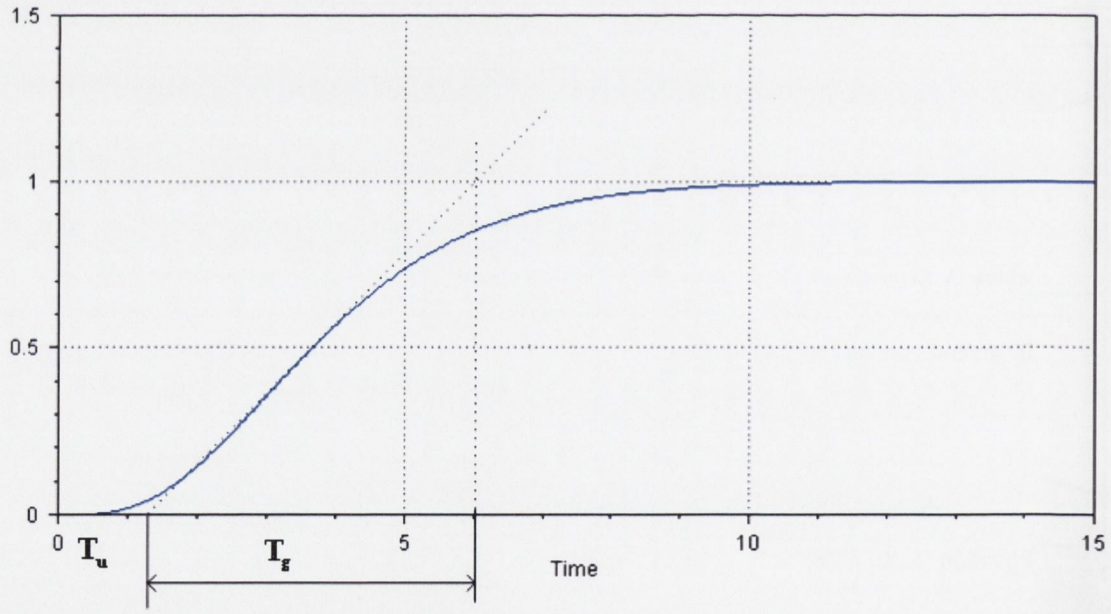


Figure 5.16 Open-loop unit step response for the process $G(s)=1/(s+1)^4$. The values of $T_u=1.2\ s$ and $T_g=4.8\ s$ allow the design of a CHR-type PID controller.

From Figure 5.16 we have $T_u=1.2\ s$ and $T_g=4.8\ s$. Using [61], these values allow us to calculate $k_p=2.4, T_i=4.8\ s$ and $T_d=0.6\ s$. So the "overdamped" CHR PID controller is $C(s)=2.4\left[1+\dfrac{1}{4.8s}+0.6s\right]$.

5.3.7 Comparison of five PID controllers for the process $G(s)+1/(s+1)^4$

In this section we summarise the properties of the four controllers designed in sections 5.3.1 to 5.3.6

year	Design	Controller	Gain Margin	Phase Margin
1942, 2006	Fine-tuned Ziegler-Nichols [50]	$C(s)=\frac{1.067}{s}\left(s+\frac{2}{\pi}\right)^2$	5.38	63.7°
1952	Chein et al. [62]	$C(s)=2.4\left(1+\frac{1}{4.8s}+0.6s\right)$	2.72	42.4°
2005	PIDeasy [59]	$C(s)=0.83\left(1+\frac{1}{0.383s}+0.43s\right)$	5.21	62.5°
2006	Lyapunov equation based optimum stability	$C(s)=0.778\left(1+\frac{1}{2.69s}+0.715s\right)$	8.63	68.9°
2006	Root locus based optimum stability	$C(s)=\frac{0.2837(s+0.8319)^2}{s}$	11.7	72.5°

Table 5.2 Comparison of four PID controllers for the process $G(s)=1/(s+1)^4$. The PID controller designed using root locus based optimum stability shows considerably enhanced robustness margins.

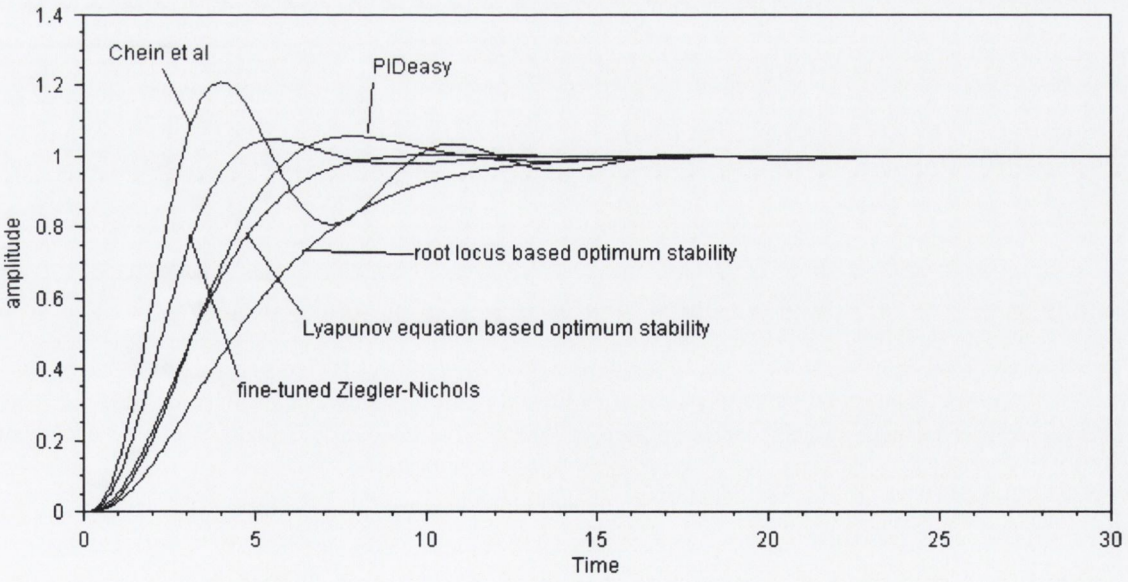


Figure 5.17 Step reference responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from the Lyapunov equation results in the smoothest response, with no overshoot and the fastest settling time. The response with the controller designed using root locus based optimum stability is smooth and has no overshoot but has a longer settling time than the Lyapunov equation based controller.

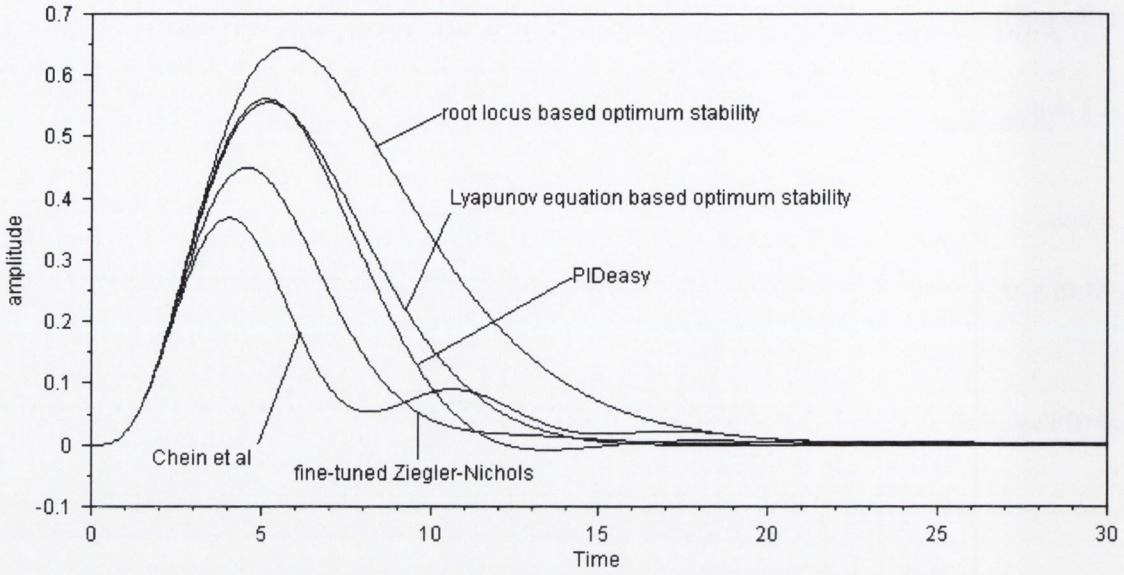


Figure 5.18 Step disturbance responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from the Lyapunov equation results in a smooth response with no undershoot and the fastest settling time. The controller designed using root locus based optimum stability is also smooth and has no undershoot but has greater overshoot than the Lyapunov equation based controller.

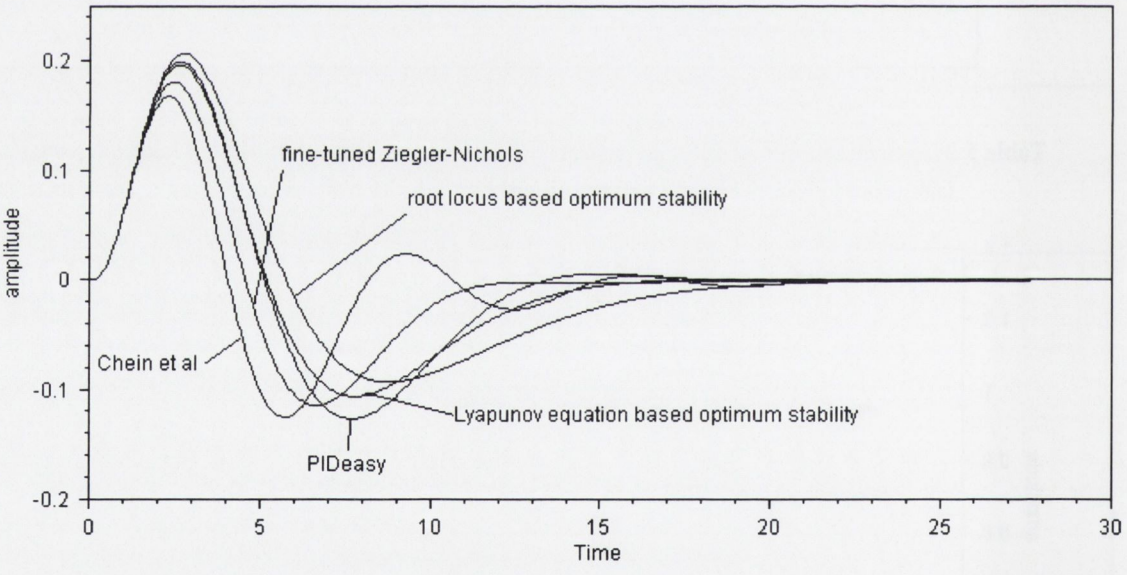


Figure 5.19 Impulse disturbance responses for system using the five controllers in Table 5.2. The controller designed using optimum stability from root results in the smoothest response with least undershoot and the fastest settling time.

5.3.8 Discussion

Systems with controllers that were designed for the process $G(s) = 1/(s+1)^4$ using either Lyapunov based optimum stability or root locus based optimum stability exhibit greater robustness margins, and smoother response characteristics than systems with controllers designed using a selection of other methods from the literature. Systems with controllers designed using optimum stability tend to exhibit greater peak disturbance responses.

5.3.9 PID controllers for the process $G(s) = 1/(s+1)^5$ designed by various methods

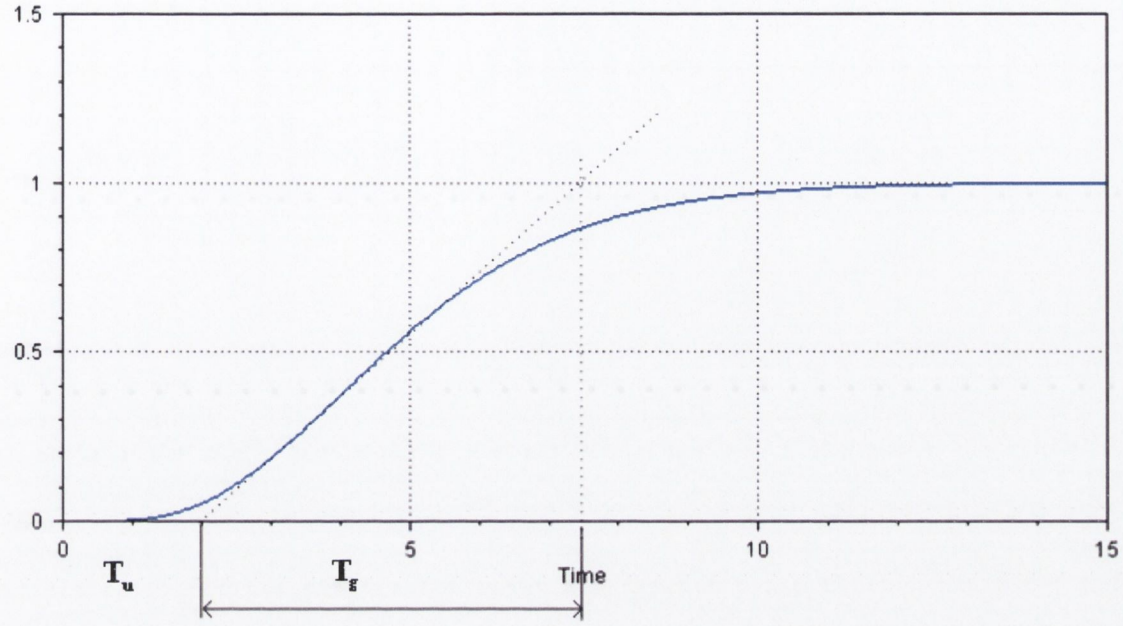


Figure 5.20 Open-loop unit step response for the process $G(s) = 1/(s+1)^5$. The values of $T_u = 2.0\ s$ and $T_g = 5.4\ s$ allow the design of a CHR-type PID controller.

As in section 5.3.6 the "overdamped" CHR method is used here in preference to the "20% overshoot " method as the former leads to smoother time responses and better robustness margins – PM of 59.5° and GM of 2.4 compared to PM of 26.9° and GM of 1.5.

From Figure 5.20 we have $T_u = 2.0\ s$ and $T_g = 5.4\ s$ - using [61] these values allow us to calculate $k_p = 1.62$, $T_i = 5.4\ s$ and $T_d = 1.0\ s$. So the "overdamped" CHR PID controller is $C(s) = 1.62\left[1 + \dfrac{1}{5.4s} + s\right]$.

Karimi et al. [63] give the controller $C(s) = 1.35\left(1 + \dfrac{1}{2.81s} + 1.27s\right)$, designed using Bode's Integral, for the process $G(s) = 1/(s+1)^5$.

The standard Ziegler-Nichols PID controller for the process $G(s) = 1/(s+1)^5$ is $C(s) = 1.73 + \dfrac{0.4}{s} + 1.87s$.

Using the general formulas in section 5.3.5 we can use root locus based optimum stability to design the PID controller $C(s) = \dfrac{0.2688(s+0.7403)^2}{s}$ for the process $G(s) = 1/(s+1)^5$.

Two controllers can be designed using the Lyapunov matrix equation method described in section 1.4. One controller has transfer function $C_1(s) = 0.475 + 0.1657/s + 0.3501s$ and the system has a single, real, rightmost eigenvalue at -0.48, the rest of the eigenvalues are complex and deeper into the left half plane. The second controller has transfer function $C_2(s) = 0.515 + 0.169/s + 0.425s$. There is a pair of real, rightmost eigenvalues for this system at –0.382 and –0.389, the rest of the eigenvalues are complex and deeper into the left half plane. Controller $C_1(s)$ is used for comparison with other designs. The response curves for systems using these two controllers are compared in Figure 5.21.

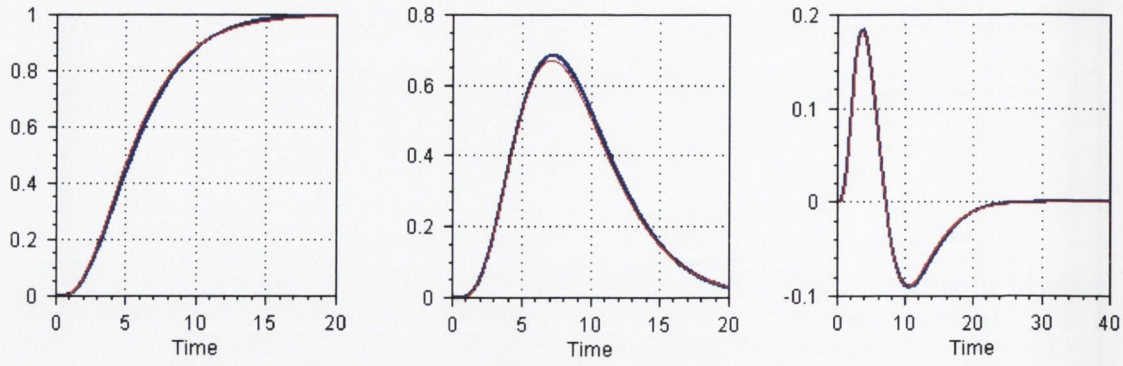


Figure 5.21 Step reference, step disturbance, and impulse disturbance responses for the two systems with PID controllers designed for the process $G(s) = 1/(1+s)^5$ using Lyapunov based optimum stability. One controller (system responses in blue) has transfer function $C_1(s) = 0.475 + 0.1657/s + 0.3501s$ and the system has a single, real, rightmost eigenvalue at -0.48, the rest of the eigenvalues are complex and deeper into the left half plane. The second controller (system responses in red) has transfer function $C_2(s) = 0.515 + 0.169/s + 0.425s$. There is a pair of real, rightmost eigenvalues for this system at –0.382 and –0.389, the rest of the eigenvalues are complex and deeper into the left half plane. Controller $C_1(s)$ is used for comparison with other designs in Table 5.3.

5.3.10 Comparison of six PID controllers for the process $G(s) = 1/(s+1)^5$

year	Design	Controller	Gain Margin (absolute)	Phase Margin
1942	Ziegler-Nichols [63]	$C(s) = 1.73 + \frac{0.4}{s} + 1.87s$	2.29	52.1°
1952	Chein et al. [62]	$C(s) = 1.62 + \frac{0.3}{s} + 1.62s$	2.43	59.5°
2003	Karimi et al. [63]	$C(s) = 1.35 + \frac{0.48}{s} + 1.71s$	2.95	50.1°
2006	Lyapunov matrix equation based optimum stability	$C_1(s) = 0.475 + \frac{0.1657}{s} + 0.35s$	6.50	70.0°
2006	Lyapunov matrix equation based optimum stability	$C_2(s) = 0.515 + \frac{0.169}{s} + 0.425s$	6.58	71.2°
2006	Root locus based optimum stability controller	$C(s) = \frac{0.2688(s+0.7403)^2}{s}$	7.08	70.9°

Table 5.3 Comparison of six PID controllers designed for the process $G(s) = 1/(s+1)^5$. The controller designed using root locus based optimum stability has the best gain margin and excellent phase margin.

The results in Table 5.3 show that the PID controllers designed using optimum stability shows considerably enhanced robustness measures. Using controller $C_1(s)$ results in a single, real, rightmost eigenvalue at -0.48, the rest of the eigenvalues are complex and deeper into the left half plane. Using controller $C_2(s)$ gives a pair of real, rightmost eigenvalues at –0.382 and –0.389, the rest of the eigenvalues are complex and deeper into the left half plane. The robustness margins for $C_2(s)$ are slightly better than $C_1(s)$ but Figure 5.21 shows that the response graphs for both controllers are indistinguishable. Controller $C_1(s)$ is used for comparison with other designs.

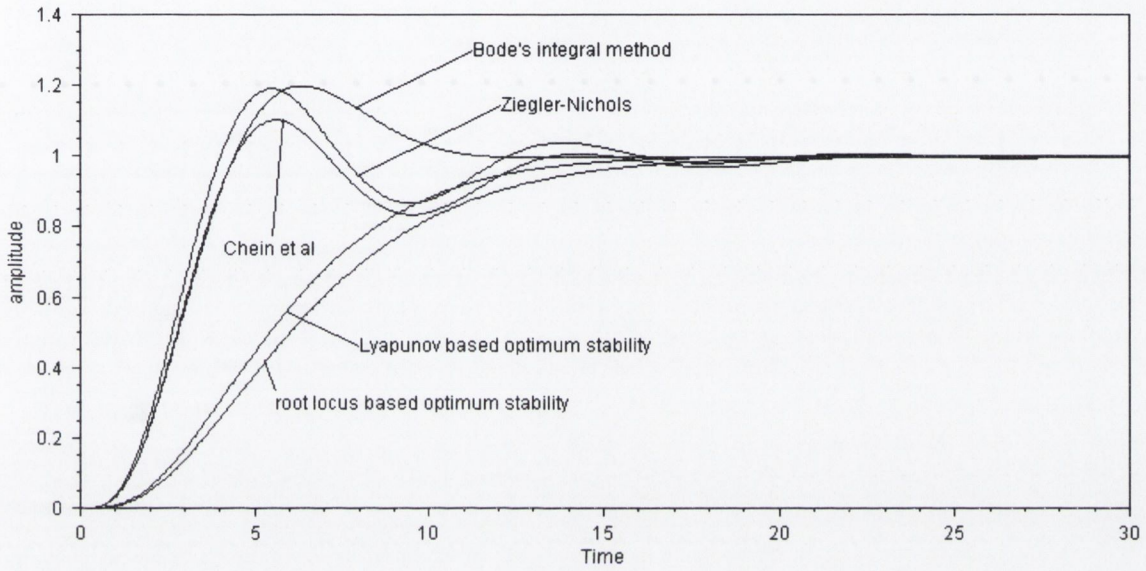


Figure 5.22 Step reference responses for systems using five controllers from Table 5.3. The controllers designed using optimum stability result in the smoothest response, with no overshoot and the fastest settling time.

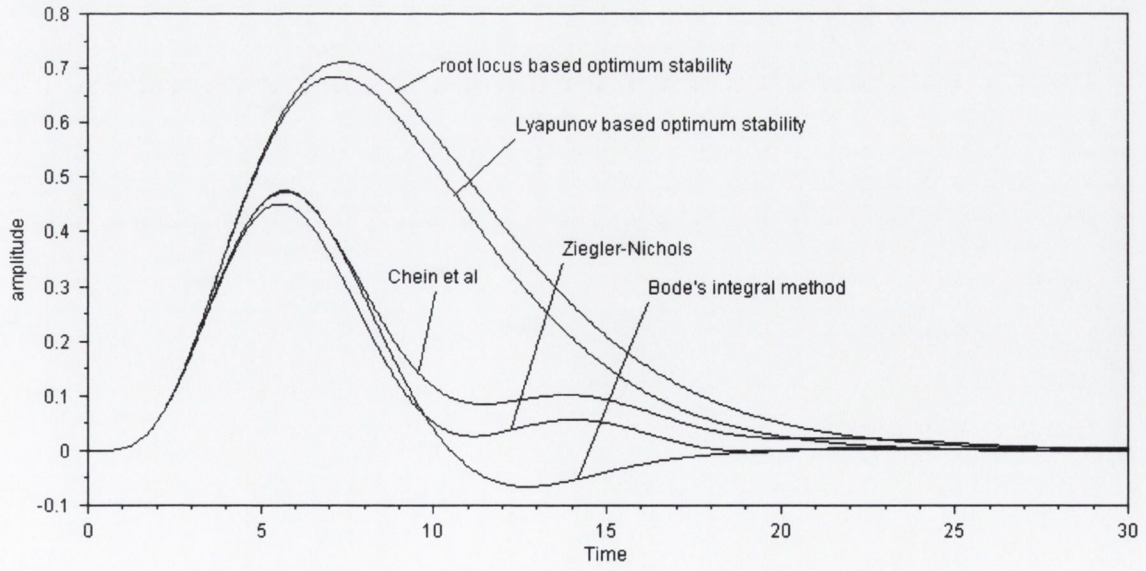


Figure 5.23 Step disturbance responses for systems using five controllers from Table 5.3. The controllers designed using optimum stability result in a smooth response with the greatest initial overshoot but the fastest settling time.

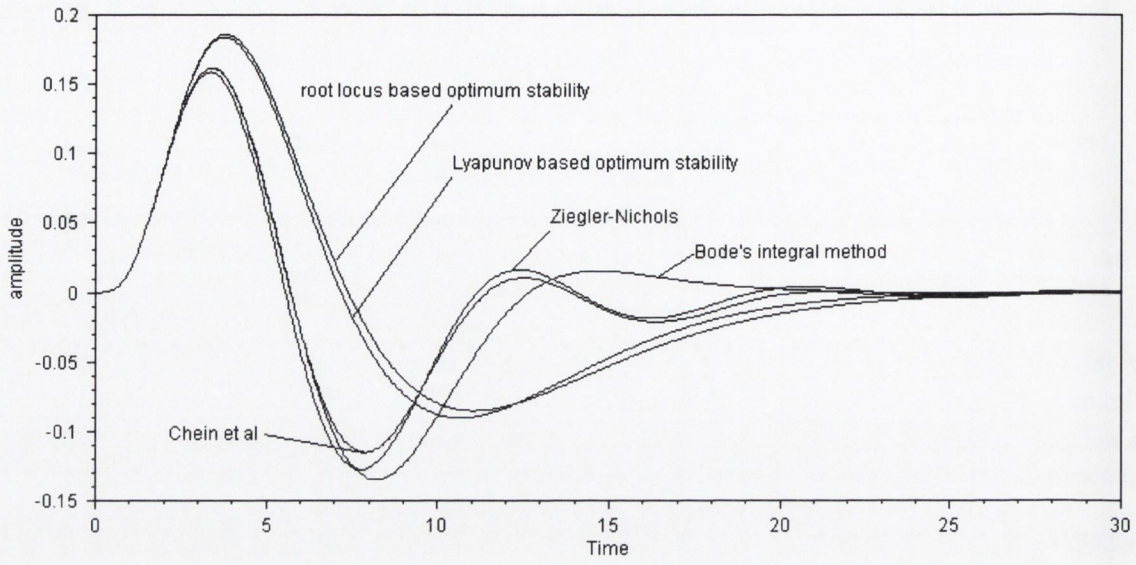


Figure 5.24 Impulse disturbance responses for systems using five of the controllers in Table 5.3. The controllers designed using optimum stability result in the smoothest response with greatest initial overshoot but with the least undershoot and the fastest settling time.

5.3.11 Discussion

Systems with controllers that were designed for the process $G(s) = 1/(s+1)^5$ using either Lyapunov based optimum stability or root locus based optimum stability exhibit greater robustness margins, and smoother response characteristics than systems with controllers designed using a selection of other methods from the literature. These characteristics seem to be obtained at the expense of other possible measures – such as settling time. Systems with controllers designed using optimum stability tend to exhibit greater peak disturbance responses.

5.4 PID controller design for the double integrator process

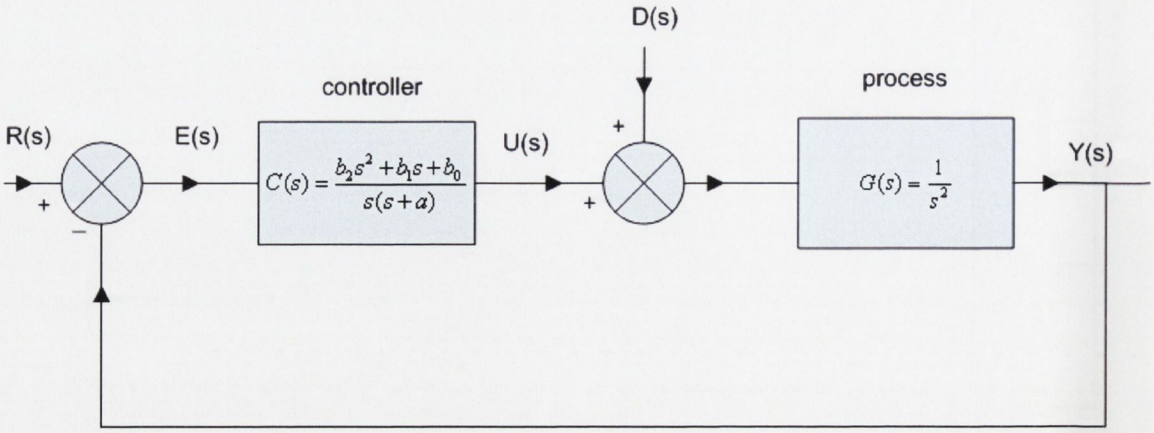


Figure 5.25 A second order controller and the double integrator process.

The double integrator process $G(s)=1/s^2$ is of interest as it is used when modelling single degree of freedom translational and rotational motion [64] including physical processes such as the attitude control of a satellite or a rolling ball on a tilting beam [65].

A Ziegler Nichols PID controller for the double integrator process [66] is:

$$C(s)=\frac{22}{7}\frac{(s+1)^2}{s} \tag{5.23}$$

The characteristic equation for the system in Figure 5.25 is $p(s)=s^3(s+a)+(b_2s^2+b_1s+b_0)$. If we put all the eigenvalues of $p(s)$ at $s=-1$ then the characteristic equation becomes $p(s)=(s+1)^4=s^4+4s^3+6s^2+4s+1$. Equating coefficients we get $b_0=1, b_1=4, b_2=6, a=4$ so an alternative controller is:

$$C(s)=\frac{6s^2+4s+1}{s(s+4)} \tag{5.24}$$

The controller $C(s)=\dfrac{b_2s^2+b_1s+b_0}{s(s+a)}$ in Figure 5.25 is actually a PID controller with a low-pass filter in the D channel. It is common practice to introduce such a filter [4]. The presence of this filter avoids saturating the controller when the input changes rapidly. It also limits the susceptibility of the controller to measurement noise. This PID structure can be seen more clearly after rearrangement of terms:

$$C(s)=K\left[1+\frac{\frac{1}{T_i}}{s}+\frac{sT_d}{1+\alpha sT_d}\right]=K\left[1+\frac{\frac{1}{T_i}}{s}+\frac{\frac{s}{\alpha}}{s+\frac{1}{\alpha T_d}}\right]=K\left[\frac{s^2+\frac{1}{\alpha T_d}s+\frac{1}{T_i}s+\frac{1}{\alpha T_dT_i}+\frac{s^2}{\alpha}}{s\left(s+\frac{1}{\alpha T_d}\right)}\right]$$

$$=K\left[\frac{\left(1+\frac{1}{\alpha}\right)s^2+\left(\frac{1}{T_i}+\frac{1}{\alpha T_d}\right)s+\frac{1}{\alpha T_dT_i}}{s\left(s+\frac{1}{\alpha T_d}\right)}\right]=\frac{b_2s^2+b_1s+b_0}{s(s+a)}$$

The parameter values for the PID controller are:

$$a=\frac{1}{\alpha T_d}\Rightarrow T_d=\frac{1}{a\alpha}$$

$$\frac{b_0}{b_1}=\frac{\frac{1}{\alpha T_dT_i}}{\frac{1}{T_i}+\frac{1}{\alpha T_d}}=\frac{1}{\alpha T_d+T_i}=\frac{1}{\frac{1}{a}+T_i}\quad\Rightarrow\frac{b_1}{b_0}=\frac{1}{a}+T_i\Rightarrow T_i=\frac{b_1}{b_0}-\frac{1}{a}$$

$$\frac{b_2}{b_0}=\frac{1+\frac{1}{\alpha}}{\frac{1}{\alpha T_dT_i}}=\frac{1+\frac{1}{\alpha}}{a\left[\frac{ab_0}{b_1a-b_0}\right]}\Rightarrow a\left[\frac{ab_0}{b_1a-b_0}\right]\times\frac{b_2}{b_0}-1=\frac{1}{\alpha}$$

$$b_0=\frac{K}{\alpha T_dT_i}\Rightarrow K=b_0\alpha T_dT_i\Rightarrow K=\frac{b_0}{a}\left[\frac{b_1}{b_0}-\frac{1}{a}\right]$$

The parameters for the PID controller $C(s)=\dfrac{b_2s^2+b_1s+b_0}{s(s+a)}=\dfrac{6s^2+4s+1}{s(s+4)}$ may be calculated as follows:

$$T_i = 4 - \frac{1}{4} = \frac{15}{4}$$

$$\frac{1}{\alpha} = \frac{16 \times 1}{16-1} \times 6 - 1 = \frac{16}{15} \times 6 - 1 = \frac{27}{5} \Rightarrow \alpha = \frac{5}{27}$$

$$T_d = \frac{1}{a\alpha} = \frac{1}{\frac{4 \times 5}{27}} = \frac{27}{20}$$

$$K = \frac{1}{4}\left[4 - \frac{1}{4}\right] = \frac{1}{4}\left[\frac{15}{4}\right] = \frac{15}{16}.$$

The system with a Ziegler Nichols controller in equation (5.23) was found to have better time responses and robustness margins than the system with a controller designed using optimum stability (5.24). Response curves for the system in Figure 5.25 with the Ziegler Nichols controller and the controller designed using optimum stability in equation (5.24) are given in Figure 5.26.

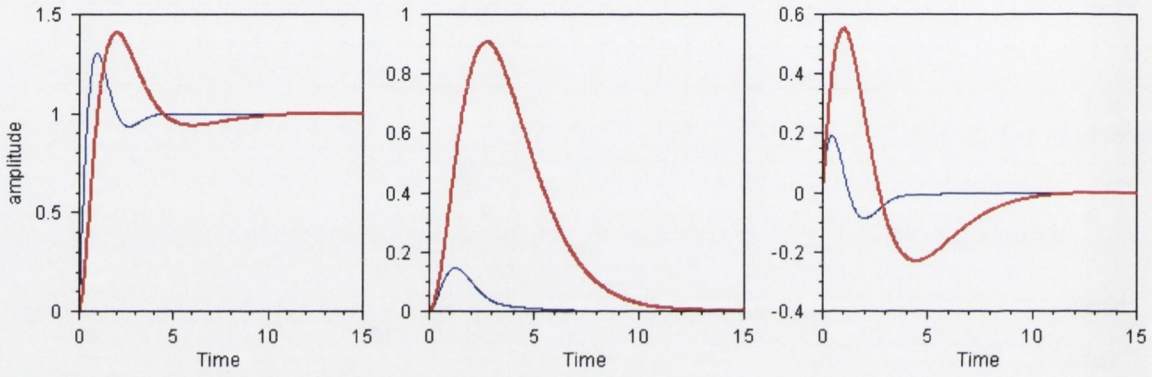


Figure 5.26 Step reference response, step disturbance response, and impulse disturbance response for the system in Figure 5.25 with a Ziegler Nichols controller given by equation (5.23) (in blue) and the controller given by equation (5.25) (in red) designed from optimum stability.

This can be easily remedied by simply choosing a system with a controller that is designed to place all the eigenvalues further to the left than initially selected. The controller in equation (5.24) puts the system eigenvalues at $s = -1$. By trial and error we find that it is necessary to design a controller that places all the system eigenvalues at $s = -4$ in order to improve on the performance of the Ziegler Nichols controller in (5.23). When all of the eigenvalues are at $s = -4$ the characteristic equation becomes $p(s) = s^4 + 16s^3 + 96s^2 + 256s + 256$. Equating coefficients we find a controller designed using optimum stability:

$$C(s) = \frac{96s^2 + 256s + 256}{s(s+16)} \tag{5.25}$$

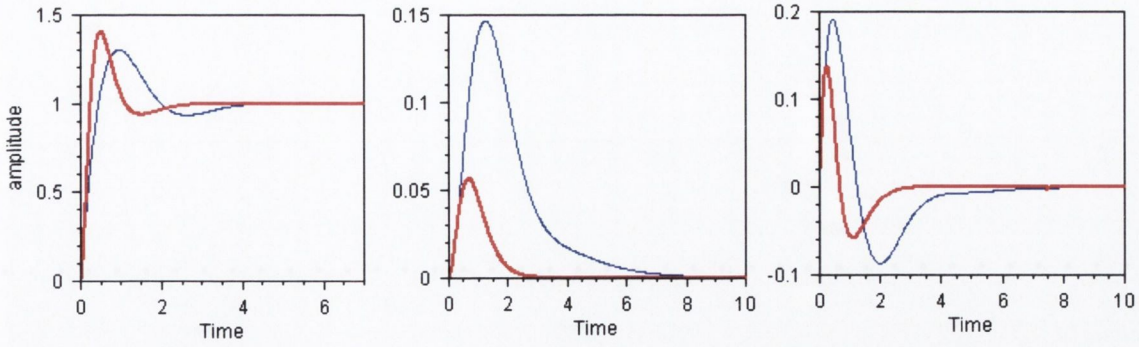


Figure 5.27 Step reference response, step disturbance response, and impulse disturbance response for the system in Figure 5.25 with a Ziegler Nichols controller given by equation (5.23) (in blue) and the controller given by equation (5.25) (in red) designed from optimum stability.

The response curves in Figure 5.27 show the improved behavior of the system in Figure 5.25 when a controller that places the systems eigenvalues at $s=-4$ is used.

The system in Figure 5.25 with the Ziegler Nichols controller given by equation (5.23) has Gain Margin 0.159 and Phase Margin 57.3^0. The same system with the controller designed using optimum stability given by equation (5.25) has Gain Margin 0.2 and Phase Margin 43.6^0. So the controller designed using optimum stability shows slightly improved Gain Margin but a smaller Phase Margin.

The response curves in Figure 5.27 show that, in spite of the smaller Phase Margin, the controller designed using optimum stability leads to shorter settling times and, in the case of step disturbance and impulse disturbance, reduced overshoot.

5.4.1 Performance integral for the system in Figure 5.25

To illustrate a difficulty with calculating exponentially weighted performance integrals for some systems we return for the moment to the controller in equation (5.24). If we replace the "6" in $C(s)=\dfrac{6s^2+4s+1}{s(s+4)}=\dfrac{6(s^2+\frac{4}{6}s+\frac{1}{6})}{s(s+4)}$ with a variable parameter k, then the characteristic equation becomes $p(s)=s^3(s+4)+k(s^2+\frac{2}{3}s+\frac{1}{6})$. This should have optimum stability for $k=6$, at the quadruple breakpoint: $s=-1$.

To calculate the exponentially weighted performance integral $J_\alpha=\int_0^\infty e^{\alpha t}\mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t)dt$ we first choose $\mathbf{Q}=\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$, $D(s)=\dfrac{1}{s}$ and $R(s)=0$.

We can now write: $E(s)=-\left[\dfrac{k(s^2+\frac{2}{3}s+\frac{1}{6})}{s(s+4)}E(s)+\dfrac{1}{s^3}\right]$

So $E(s)=\dfrac{(s+4)}{s^3(s+4)+k(s^2+\frac{2}{3}s+\frac{1}{6})}=\dfrac{(s+4)}{s^4+4s^3+ks^2+\frac{2k}{3}s+\frac{k}{6}}$.

Also $\frac{d^4e}{dt^4}+4\frac{d^3e}{dt^3}+k\frac{d^2e}{dt^2}+\frac{2k}{3}\frac{de}{dt}+\frac{k}{6}e=0$.

Finally : $\left[s^4+4s^3+ks^2+\frac{2}{3}ks+\frac{k}{6}\right]E(s)=$

$$s^3e(0)+s^2\frac{de}{dt}(0)+s\frac{d^2e}{dt^2}(0)+\frac{d^3e}{dt^3}(0)+4s^2e(0)+4s\frac{de}{dt}(0)+4\frac{d^2e}{dt^2}(0)+kse(0)+k\frac{de}{dt}(0)+\frac{2}{3}ke(0)$$

This gives: $e(0)=0,\frac{de}{dt}(0)=0,\frac{d^2e}{dt^2}(0)=0,\frac{d^3e}{dt^3}(0)=1$

So the differential equations for e(t) maybe written:

$$\frac{d\mathbf{x}}{dt}=\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\\frac{-k}{6}&\frac{-2k}{3}&-k&-4\end{pmatrix}\mathbf{x}(t),\quad \mathbf{x}(0)=\begin{pmatrix}e(0)\\e'(0)\\e''(0)\\e'''(0)\end{pmatrix}=\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

This means that $J_\alpha=\mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0)=l_{44}$ where $\left(\mathbf{A}+\frac{\alpha}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}+\frac{\alpha}{2}\mathbf{I}\right)=-\mathbf{Q}$.

Solving for J_α and calculating $\frac{\partial J_\alpha}{\partial k}=0$ at $k=6$ gives $\alpha=1.9824$, $\alpha=1.99118$, and $\alpha=2.00441$ minimise J_α. Unfortunately, the denominator of J_α with $k=6.0$ is extremely sensitive to α so when $\alpha=1.9824$ the denominator of J_α is 7.69×10^{-17} and when $\alpha=1.99118$ the denominator of J_α is 6.60×10^{-20}. The values $k=6.0$ and $\alpha=1.9824$ lead to a minimum $J_\alpha=3.83\times10^{13}$. The large parameter values and the extremely sensitive dependence of J_α on α indicate that there is something wrong with using this integral as a performance indicator.

To follow this particular line of inquiry would require selecting different structure for $\mathbf{Q}$, for example $\mathbf{Q}=\begin{pmatrix}1&0&0&0\\0&1/\omega_n^2&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$. This in turn requires us to select a value for ω_n, the underdamped natural frequency. There is no obvious method for selecting ω_n here so this approach was abandoned.

5.5 Summary of results in this chapter

Root locus based optimum stability was used to design PI controllers for the sample process $G(s)=1/(s+1)^4$. Root locus based optimum stability and Lyapunov matrix equation based optimum stability was used to design and PID controllers for the sample processes $G(s)=1/(s+1)^4$, $G(s)=1/(s+1)^5$, $G(s)=1/s^2$.

General formulas were given for:

(a) root locus based optimum stability design for a PI controllers for process $G(s)=k_2/(s+b)^m$

(b) root locus based optimum stability design of a restricted class of PID controllers for process $G(s) = k_2 /(s+b)^m$.

The performance of these systems was compared to that of systems from the literature. It was seen that controllers designed using optimum stability methods produce systems with enhanced performance compared to controllers designed using the other methods.

A root locus method for fine-tuning the standard Ziegler-Nichols controller parameters was described in sections 5.2.2 and 5.3.2. - the resulting controller out-performs the original with respect to significant performance criteria.

5.6 Suggestions for future work

Apply the methods presented to other benchmark processes from [58].

Chapter 6: PI controller design for processes with time delay using the root locus method

6.1 A procedure for designing PI controllers for general first-order lag plus time-delay process

6.1.1 Introduction

This chapter describes a general procedure for the design of a PI controller for a general first-order lag plus time-delay (i.e. FOLPD) process. We derive two equations that allow the designer to calculate the PI controller parameters (k_p, k_i) using only the process parameters. The central idea of this general design procedure is to use the root locus method to select the system parameters that put the system's rightmost eigenvalue as far to the left as possible in the complex plane. These parameters are chosen as the system's nominal design parameters. When the system is operating at this point it is said to be operating at a point of optimum stability in the root locus sense. The procedure presented here is an extension to the root locus based procedure for designing controllers for delay-free processes that has been described previously in [20, 47, 67] and in chapters 3, 4, and 5 above. There are many advantages to designing controllers for time-delay systems from the optimum stability perspective and these are described below.

The study of time-delay systems began in the 18th century with the work of such luminaries as Euler, Lagrange, and Laplace [68]. Since the 1930s [69-71] control engineers have recognized the importance of time delays in process models and have developed techniques for designing controllers for these processes. Motivated by the widespread use of PI controllers [72] and the importance of time delays in process models, the problem of tuning a PI controller for a process with time delay has been of interest for over sixty years [71]. The open-loop step response of the first-order lag plus time-delay (FOLPD) process:

$$G(s) = \frac{k_2 e^{-sL}}{\left(s + \frac{1}{\tau}\right)} \tag{6.1}$$

is shown in Figure 6.1, where k_2 is the steady state process gain, L is the time delay of the process, and τ is its time constant.

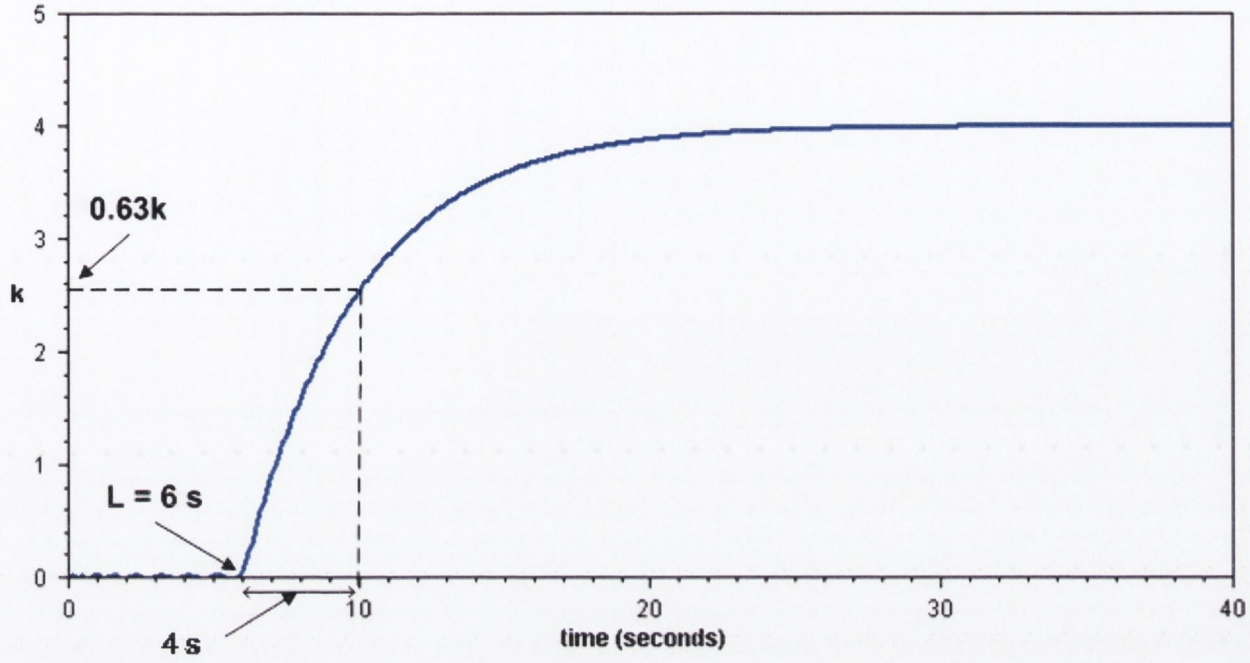


Figure 6.1 Open loop step response, or the "process reaction curve", for the FOLPD process $G(s) = e^{-6s}/\left(s+\frac{1}{4}\right)$. A step input is applied to the process at $t = 0$ and after a delay of 6 s the process begins to respond. The speed of response is dictated by the time constant of the process. After one time constant, which is 4 seconds, the response reaches $(1-e^{-1}) = 0.63$ of the final value. After about 30 s the process has settled down to a new operating point. Reaction curves of this shape are typical of many industrial processes [72] including processes as diverse as the air intake on a turbojet engine [73] and chemical mixing tanks or reaction vessels [73]. The problem of designing PI controllers for processes with this type of reaction curve has been under scrutiny for over sixty years [71].

In this section I develop a root locus method to select controller parameters that place the system's rightmost eigenvalue as far to the left as possible in the complex plane. We derive equations that allow the designer to calculate these parameters using the process parameters k_2, L, and τ.

6.1.2 Root locus diagrams for a PI controller and a general FOLPD process

Classical root locus plotting rules [48] can be extended to systems with time delay as described in [74] and [75]. The characteristic polynomial for a typical time-delay system is given by $p(s) = N(s) + k\,M(s)e^{-sL}$, where $N(s)$ and $M(s)$ are polynomials in $s = \sigma + j\omega$ and L is the time delay. $N(s)$ and $M(s)$ are of degree n and m, respectively, with $m \le n$. In root locus terminology, the poles and zeros of the root locus are given by the roots of $N(s) = 0$ and $M(s) = 0$, respectively. The gain k is a real number.

Root loci for time-delay systems are symmetrical about the real axis, but unlike the delay-free case, the number of branches is infinite. Setting $k = 0$, the branches start at the poles and at $\sigma = -\infty$. Setting $k = \pm\infty$, the branches terminate at the zeros and at $\sigma = +\infty$. Branches that do not terminate at a zero approach $\sigma = +\infty$ along asymptotes. These asymptotes are infinite in number and are parallel to the real axis. An example of a root locus diagram for a time-delay system is given in Figure 6.2.

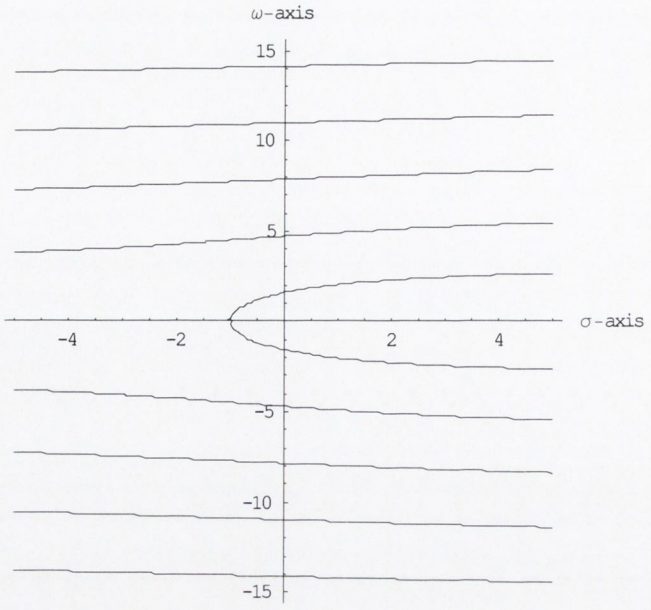


Figure 6.2 An illustrative example of a root locus diagram for a time-delay system. The system used has characteristic equation $p(s) = s + ke^{-s} = 0$ with k as parameter. There is a pole at $s = 0$ and a branch starts at this pole when $k = 0$. This branch meets another branch coming from $\sigma = -\infty$ to form a breakpoint. These two branches then cross the imaginary axis and approach $\sigma = \infty$. There is an infinite number of other branches that start at $\sigma = -\infty$ and approach $\sigma = \infty$ along asymptotes that are parallel to the real axis. Root loci for time-delay systems are symmetrical about the real axis but, unlike the delay-free case, the number of branches is infinite. The branches nearest to the real axis, called the primary branches, are the critical ones when considering stability since they are the first to cross the imaginary axis with subsequent crossings taking place at higher values of k.

The PI controller used here has the transfer function $C(s) = k_1\left(1 + \frac{F}{s}\right)$, where k_1 is the gain of the proportional element, $k_1 F$ is the gain of the integrator element, and $F = 1/T_i$, where T_i is the reset time of the controller [75]. An equivalent form for PI controllers is $C(s) = k_p + \frac{k_i}{s}$, where k_p is the gain of the proportional element and k_i is the gain of the integrator element. The transfer function for a FOLPD process is given by (6.1). Figure 6.3 shows a PI controller connected to a FOLPD process in a negative feedback configuration.

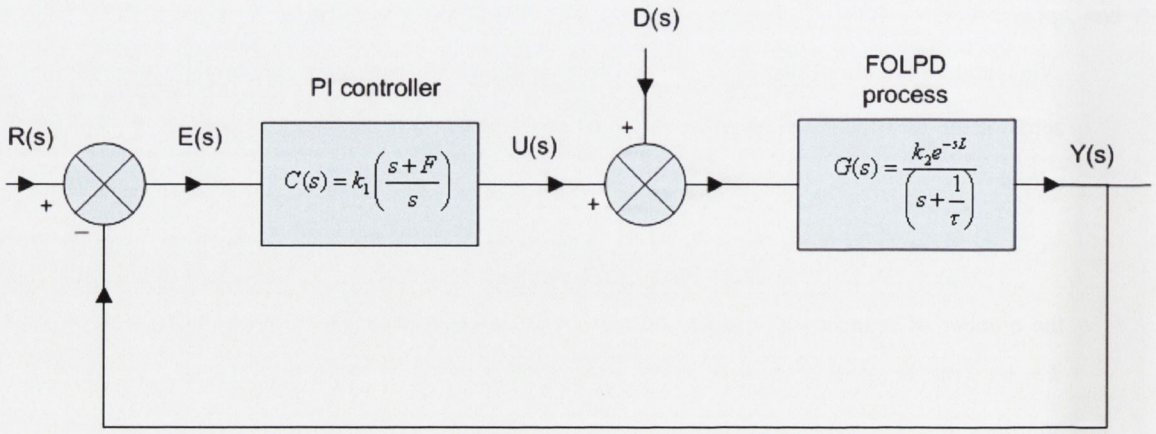


Figure 6.3 PI controller and a first-order lag plus time-delay process (FOLPD).

Letting $k = k_1 k_2$, the characteristic polynomial of the system becomes $p(s) = s(s + \frac{1}{\tau}) + ke^{-sL}(s + F)$. To simplify the derivations below we use the scaled variable $q = sL$ and consider the scaled characteristic equation

$$p(q) = q(q + \frac{L}{\tau}) + kLe^{-q}(q + FL) = 0. \tag{6.2}$$

The dependence of the root locus of (6.2) on the product FL shows the existence of a critical value of FL that results in a breakpoint in the root locus at a triple root. As will be shown later, this triple root occurs where the rightmost eigenvalue is as far to the left as possible. Formulas for calculating the parameters that result in the triple root and therefore produce an optimally stable system are also derived later. We calculate that, for example, if $L = 1$ and $\tau = 5$, then the critical value for FL is 0.28845.

Figure 6.4, Figure 6.5, and Figure 6.6 show the root locus plots for (6.2) with $L = 1$, $\tau = 5$ and with $FL = 0.25, FL = 0.28845,$ and $FL = 0.3$ respectively. Only those branches nearest to the real axis, the 'primary' branches, are shown. The primary branches are the critical branches when considering stability since they are the first to cross the imaginary axis with subsequent crossings taking place at higher gains [74], [75]. A relationship between $|k_{s=j\omega}|$, the gains at which the branches cross the imaginary axis, and the values of ω at the crossing points, can be established by returning for the moment to the unscaled characteristic polynomial. The gain condition is $|k_1 k_2| = |k| = |s|\left|s + \frac{1}{\tau}\right| \exp(\sigma L) / |s + F|$; using it we can see that values of $|k|$ when the branches cross the imaginary axis are given by $|k_{s=j\omega}| = \omega\sqrt{\omega^2 + \frac{1}{\tau^2}} / \sqrt{\omega^2 + F^2}$. Since the numerator of the expression for $|k_{s=j\omega}|$ is of order ω^2 and the denominator is of order ω, it is clear that larger values of ω imply larger values of $|k_{s=j\omega}|$.

$FL = 0.25$ is just below the critical value. The primary branches of the corresponding root locus of (6.2) are given in Figure 6.4. The primary branches of the root locus of (6.2) at the critical value of FL, that is $FL = 0.28845$, are shown on Figure 6.5. The two leftmost breakpoints on Figure 6.4 have coalesced to form a triple root.

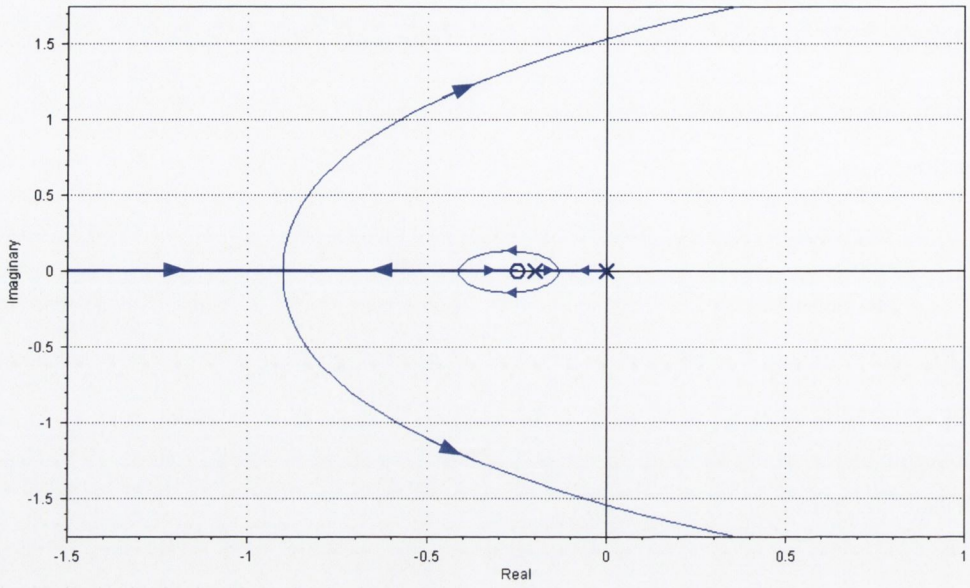


Figure 6.4 Primary branches of the root locus for $p(q) = q(q + L/\tau) + kL(q + FL)e^{-q} = 0$ with k as parameter, $L = 1$, $\tau = 5$, and $FL = 0.25$, which is just below the critical value. In addition to the branches that start at the poles at $q = 0$ and $q = -L/\tau$, and end at the zero at $q = -FL$, there is an infinite number of branches that start at $\sigma = -\infty$ and end at $\sigma = +\infty$. One of this infinite number of branches approaches the zero at $q = -FL$ and meets the branch going in the opposite direction forming the leftmost breakpoint.

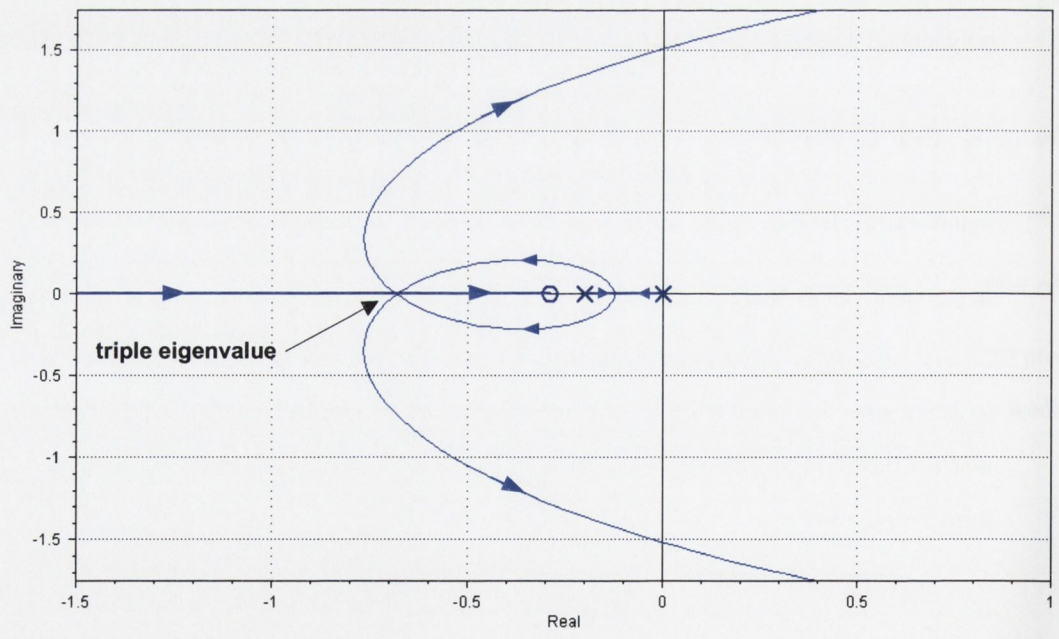


Figure 6.5 Primary branches of the root locus for $p(q) = q(q + L/\tau) + kL(q + FL)e^{-q} = 0$ with k as parameter, $L = 1$, $\tau = 5$ and FL at the critical value $FL = 0.28845$. A single branch that originates at $\sigma = -\infty$ meets the two branches that originate at the poles at $q = 0$ and $q = -L/\tau$. Here three eigenvalues have coalesced to form a single point. The rightmost eigenvalue is as far to the left as possible at this point and the system is said to be optimally stable in the root locus sense. Optimum stability design gives nominal parameters that place the system's operating point at this triple eigenvalue.

As FL becomes larger, the real breakpoints to the left of the zero cease to exist. For example, at $L = 1,\ \tau = 5$, and $FL = 0.3$ the primary branches of the root locus of (6.2) are shown in Figure 6.6.

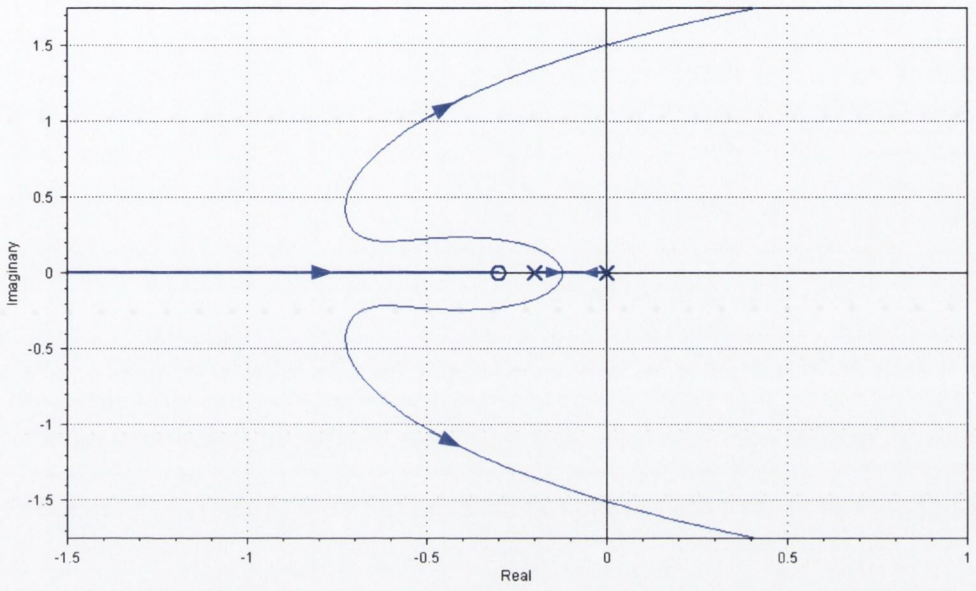


Figure 6.6 Primary branches of the root locus for $p(q) = q\left(q + L/\tau\right) + kL(q + FL)e^{-q} = 0$ with $L = 1,\ \tau = 5, FL = 0.3$ and with k as parameter. Here FL is just above the critical value, the triple root evident in Figure 5 is gone, and the rightmost eigenvalue is no longer as far to the left as possible. The branch that originates at $\sigma = -\infty$ now terminates at the zero at $q = -FL$. The branches that start at the poles $q = 0$ and $q = -L/\tau$ simply approach $\sigma = +\infty$ along asymptotes.

6.2 Exploiting the triple eigenvalue

The procedure described here is an extension to the one described in section 1.2.

When there is a triple eigenvalue at $q = -a$ in the root locus, the scaled characteristic equation $p(q) = q(q + \frac{L}{\tau}) + kLe^{-q}(q + FL) = 0$ must have the factorization

$$p(q) = (q + a)^3 f(q) = 0, \tag{6.3}$$

where $f(q)$ has an infinite number of roots. Differentiating (6.3) with respect to q gives

$$p'(q) = (q + a)^2[3f(q) + f'(q)(q + a)] = 0. \tag{6.4}$$

Differentiating (6.4) with respect to q gives

$$p''(q) = (q + a)[6f(q) + 6(q + a)f'(q) + (q + a)^2 f''(q)] = 0. \tag{6.5}$$

The equations $p(-a) = 0,\ p'(-a) = 0$, and $p''(-a) = 0$, being (6.3), (6.4), and (6.5) evaluated at $q = -a$, are the key to the following derivation of the PI controller design equations.

6.3 Root locus based design for a PI controller for a general FOLPD process

We now show how to choose the design parameters for the PI controller $C(s)$ in Figure 6.3 using root locus based optimum stability. We deduce from studying Figure 6.4, Figure 6.5, and Figure 6.6 that the characteristic equation for the system in Figure 6.3 has a triple root when the parameters are such that the rightmost eigenvalue is as far to the left as possible. We know from (6.3), (6.4), and (6.5) that the characteristic equation and its first two derivatives are equal to zero at this root. We now apply this idea to (6.2). (6.2) and its first two derivatives are rearranged to obtain

$$kLe^{-q} = \frac{-q\left(q+\frac{L}{\tau}\right)}{(q+FL)}, \tag{6.6}$$

$$kLe^{-q} = \frac{-2\left(2q+\frac{L}{\tau}\right)}{(1-q-FL)}, \tag{6.7}$$

and

$$kLe^{-q} = \frac{2}{(2-q-FL)}. \tag{6.8}$$

(6.6), (6.7), and (6.8) will now be solved for the three unknowns $F, k,$ and q. Equating the right hand sides of (6.6) and (6.7) gives

$$FL = \frac{-Lq^2 - q^2\tau - q^3\tau}{L + Lq + 2q\tau + q^2\tau}. \tag{6.9}$$

Equating the right hand sides of (6.7) and (6.8) gives

$$FL = \frac{2L - Lq + 2\tau + 2q\tau - 2q^2\tau}{L + 2\tau + 2q\tau}. \tag{6.10}$$

Equating (6.9) and (6.10) gives the following cubic in q

$$\left(q+\frac{L}{2\tau}\right)\left(q^2+\left(4+\frac{L}{\tau}\right)q+2\left(1+\frac{L}{\tau}\right)\right)=0. \tag{6.11}$$

The three roots of (6.11) are

$$q_1 = -\frac{L}{2\tau}, \tag{6.12}$$

$$q_2 = \sqrt{2+\frac{L^2}{4\tau^2}} - 2 - \frac{L}{2\tau}, \tag{6.13}$$

and $$q_3 = -\sqrt{2+\frac{L^2}{4\tau^2}} - 2 - \frac{L}{2\tau}. \tag{6.14}$$

Root q_1 is extraneous since it gives different values for kL in (6.6), (6.7), or (6.8) and q_3 leads to a negative value for k and to instability. However, q_2 is the physically allowable root since it gives a positive value for k and asymptotic stability. Substituting q_2 into (6.9) or (6.10) gives

$$FL = \frac{\left(5+\frac{L}{2\tau}\right)\sqrt{2+\frac{L^2}{4\tau^2}} - 7 - \frac{L}{2\tau} - \frac{L^2}{4\tau^2}}{\sqrt{2+\frac{L^2}{4\tau^2}} - 1}. \tag{6.15}$$

Substituting for q_2 and FL in (6.6), (6.7) or (6.8) gives

$$kL = 2\left[\sqrt{2+\frac{L^2}{4\tau^2}} - 1\right] e^{-2-\frac{L}{2\tau}+\sqrt{2+\frac{L^2}{4\tau^2}}}. \tag{6.16}$$

(6.15) and (6.16) may be used to design the PI controller in Figure 6.3 using only the process parameters $k_2, L,$ and τ. The controller parameters obtained using (6.15) and (6.16) lead to a system that is optimally stable in the root locus sense.

6.3.1 Comparing designs from optimum stability with designs from performance integrals

Wade and Johnson [76] study the performance of PI controllers for the four FOLPD processes $G(s) = \frac{e^{-Ls}}{s+1}$ parameterized by the time delay L, where L takes on the values 0.25 s, 0.5 s, 1.0 s, and 2.0 s. Each controller is designed by finding the values for the controller parameters (k_p, k_i) that minimize a performance integral. Ten performance integrals are used in [76] and a controller is considered to be well designed if it results in a system that is robust but not too conservative. A design is considered to be robust but not too conservative if the system's gain margin (GM) is > 2.25 dB and its phase margin (PM) is in or very near the range $20^o < PM < 65^o$. Using these performance measures [76] concludes that the best controller designs result from the time-weighted performance integrals $J_{IT^2AE} = \int_0^\infty t^2 |e(t)| dt$ and $J_{IES2_{\alpha-1}} = \int_0^\infty (\exp(2\alpha t) - 1)\, e^2(t)\, dt$, where $e(t)$ is the error signal and α is fixed at the value 1.5 [76].

Table 6.1 gives the (k_p, k_i) values for the four controllers designed using (6.15) and (6.16), and for the eight best controllers from [76].

Controller $C(s) = k_p + \frac{k_i}{s}$		Process $G(s) = \frac{e^{-Ls}}{s+1}$			
Controller design criterion	Controller parameters	$L = 0.25$	$L = 0.5$	$L = 1.0$	$L = 2.0$
minimize J_{IT^2AE}	k_p	1.84	0.85	0.59	0.48
	k_i	1.85	0.85	0.48	0.26
minimize $J_{IES2_{\alpha=1}}$	k_p	2.18	1.44	0.89	0.55
	k_i	1.96	1.08	0.56	0.27
optimum stability	k_p	1.66	0.77	0.37	0.21
	k_i	2.14	0.81	0.37	0.20

Table 6.1 PI controller parameters for the best eight controllers from [76] and from root locus based optimum stability. PI controller parameters are selected either by minimizing the performance integrals $J_{IT^2AE} = \int_0^\infty t^2 |e(t)| dt$ and $J_{IES2_{\alpha=1}} = \int_0^\infty (\exp(2\alpha t) - 1)\, e^2(t) dt$, where $e(t)$ is the error signal and α is fixed at $\alpha = 1.5$ [76], or using (6.15) and (6.16), which are derived from optimum stability

Controller $C(s)=k_p+\frac{k_i}{s}$	Process $G(s)=\frac{e^{-Ls}}{s+1}$											
Controller design criterion	$L=0.25$			$L=0.5$			$L=1.0$			$L=2.0$		
	GM	PM	DM	GM	PM	DM	GM	PM	DM	GM	PM	DM
minimize J_{IT^2AE}	3.41	63.48	0.60	3.70	65.65	1.35	2.95	66.14	2.29	2.46	69.43	4.29
minimize $J_{IES2_{\alpha-1}}$	2.92	61.57	0.50	2.32	59.72	0.80	2.13	63.91	1.63	2.23	70.04	4.02
optimum stability	3.64	57.80	0.56	4.02	66.13	1.46	4.27	68.92	3.27	3.99	68.06	6.05

Table 6.2 Comparison between the robustness measures of PI controllers designed by three different methods.

Table 6.2 shows the results for twelve controllers - the eight best of forty controllers from [76] designed by selecting controller parameters that minimize the performance integrals $J_{IT^2AE}=\int_0^\infty t^2|e(t)|dt$ and $J_{IES2_{\alpha-1}}=\int_0^\infty(\exp(2\alpha t)-1)\,e^2(t)dt$, where $e(t)$ is the error signal, and α is fixed at the value 1.5 [76], and the four controllers designed from optimum stability using (6.15) and (6.16). Gain margin (GM in absolute units), phase margin (PM in degrees), and delay margin (DM in seconds) - that is, the additional delay which if added in cascade with the forward path would bring the system to the boundary of stability - are used to quantify robustness.

From Table 6.2 we can conclude that for a time delay of 0.25 s the GM for controllers designed from (6.15) and (6.16) is larger than that achieved by all forty controllers presented in [76]. The PM and DM for the controllers designed from (6.15) and (6.16) are a little smaller than the best controllers in [76]. For time delays of 0.5 s, and 1.0 s the GM, PM, and DM for controllers designed from (6.15) and (6.16) are larger that the best of the forty controllers in [76]. For time delay of 2.0 s the GM and DM of the controllers designed from (6.15) and (6.16) are larger than the best controllers in [76]. Also, for time delay of 2.0 s the PM of the controller designed from (6.15) and (6.16) (68.06°) is smaller than the PMs of the best controllers in [76] (69.43° and 70.04°) and is nearer to the range of PMs ($20^\circ < PM < 65^\circ$) considered by [76] to produce a good design.

In Table 6.2 the values for GM, PM, and delay margin (DM) were calculated using both Mathematica's Control System Professional [77] and Program CC [78]. Although the values obtained from these two programs agreed with each other, they did not always agree with those reported in [76].

The conclusions we draw from Table 6.2 apply only to the forty PI controllers designed in [76] using ten performance integrals and the four PI controllers designed from (6.15) and (6.16). We cannot make the claim that (6.15) and (6.16) will always produce the best controllers, but we can say, for the specific processes considered, that the designs based on them are superior to the those based on ten commonly-used performance integrals.

Figure 6.7 illustrates the different time responses obtained with controllers designed using performance integrals and the controllers designed using (6.15) and (6.16). Figure 3 shows where the reference input (R(s)) and the disturbance input (D(s)) are applied.

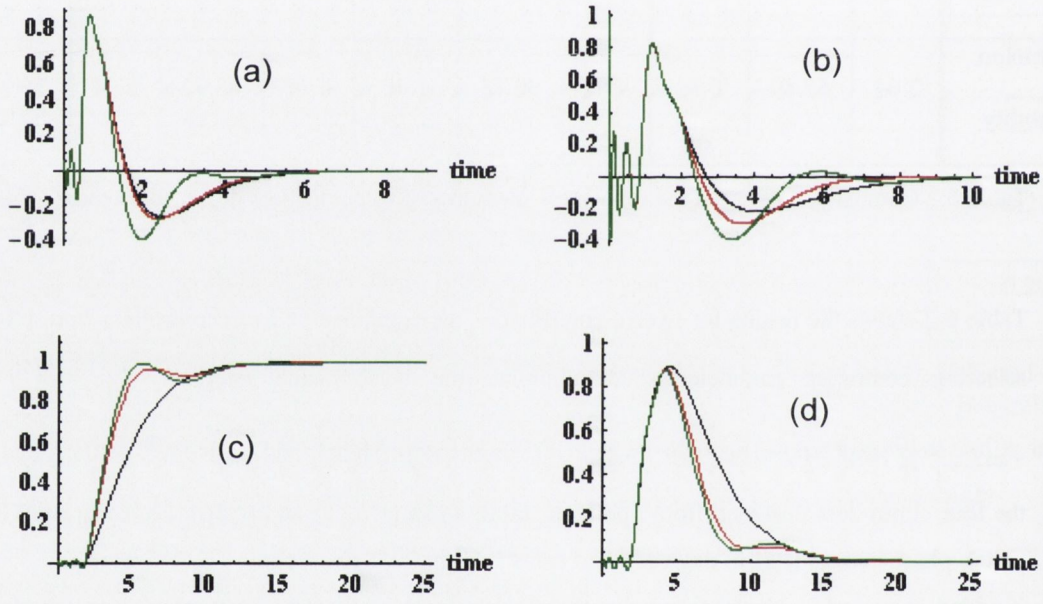


Figure 6.7 Step response and impulse response curves for various controller designs. Red curve is for minimum $J_{IT^2AE} = \int_0^\infty t^2 |e(t)| dt$ controller; green curve is for minimum $J_{IES2_{\alpha-1}} = \int_0^\infty (\exp(2\alpha t) - 1)\, e^2(t) dt$ controller, where $e(t)$ is the error signal and α is fixed at $\alpha = 1.5$ [76]; black curve is for the controller designed from optimum stability considerations using (6.15) and (6.16). (a) Impulse disturbance responses for $L = 0.5$. (b) Impulse disturbance responses for $L = 1.0$. (c) Step disturbance responses for $L = 2.0$. (d) Step reference responses for $L = 2.0$. We can see that the controllers designed from optimum stability result in systems with smoother response curves, less undershoot and overshoot, and settling times that are comparable the best controllers in [76].

6.3.2 Comparing designs from optimum stability with designs from stability region methods

[79] and [44] consider the problem of designing a PI controller $C(s) = k_p + \frac{k_i}{s}$ for the process $G(s) = \frac{k}{1+Ts} e^{-Ls}$. They derive an algorithm for determining the sets of points in the (k_p, k_i) plane that lead to system stability. The stability region for a PI controller with $G(s) = \frac{\frac{1}{4}}{s+\frac{1}{4}} e^{-s}$, a sample process from [79] and [44], is shown in Figure 6.8.

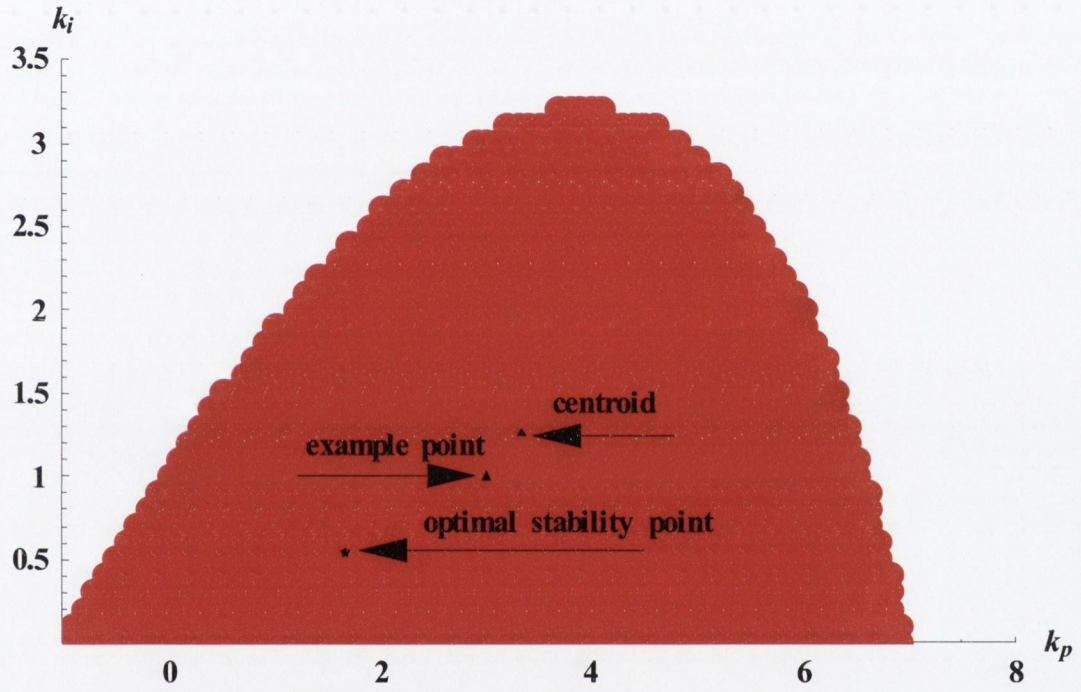


Figure 6.8 Stability region for the sample system consisting of the controller $C(s) = k_p + \frac{k_i}{s}$ and the process $G(s) = \frac{\frac{1}{4}}{s+\frac{1}{4}} e^{-s}$. The red region represents points that lead to system stability. Three possible controller design points are indicated. The centroid of the region, the point (3,1) used as an example design point in [79] and [44], and the optimum stability point. Performance measures show that the fact that the optimum stability point is nearest to the stability boundary is irrelevant. Indeed the centroid leads to the poorest performance even though it is furthest from the stability boundary. This poor performance is due to the fact when selecting a design point the designer must not only consider stability but also the degree of stability of the point. The system designed with parameter values chosen at the centroid is less stable (in the root locus sense) than one designed with parameter values at the optimum stability point.

In [79], [44] the point $(k_p, k_i) = (3,1)$ is used as an example design point for a PI controller for the process $G(s) = \frac{\frac{1}{4}}{s+\frac{1}{4}} e^{-s}$. The design point $(k_p, k_i) = (1.659, 0.535)$ is found using (6.15) and (6.16). Yet another design point that has an intuitive appeal is the centroid of the stability region $(k_p, k_i) = (3.4, 1.2)$. The intuitive appeal of the centroid is the fact that it is as deep into the stability region as possible. This intuition is incorrect and the appeal of the centroid as a design point is unfounded. When selecting a design point within

the stability region the designer must check the degree of stability of the point. The robustness measures for the three designs are given in Table 6.3.

controller $C(s) = k_p + \frac{k_i}{s}$ process $G(s) = \frac{\frac{1}{4}}{s+\frac{1}{4}} e^{-s}$		Robustness measures		
Controller design criterion	Controller parameters	GM (absolute)	PM (degrees)	DM (seconds)
example controller	$k_p = 3.0$ $k_i = 1.0$	2.0	40.10	0.90
centroid of region of stability	$k_p = 3.4$ $k_i = 1.2$	1.76	32.60	0.65
optimum stability	$k_p = 1.66$ $k_i = 0.54$	3.65	57.80	2.26

Table 6.3 Controller parameters, GM (gain margin), PM (phase margin), and DM (delay margin) for the example controller from [79] and [44], the controller designed with parameter values at the centroid of the region of stability , and the controller design based on optimum stability. The enhanced robustness measures show the effect of using the parameters derived from optimum stability.

From Table 6.3 we see that the design point that results in the worst performance is the centroid whereas the design point that results in the best performance is the one derived from (6.15) and (6.16). The example design point $(k_p, k_i) = (3,1)$ mentioned above is close to the centroid but is biased towards the point given by (6.15) and (6.16). The bias of the example design towards the point given by (6.15) and (6.16) might explain the fact that the robustness measures for the example design are better than the centroid design but not as good as the design obtained from (6.15) and (6.16). The example design point from [79] and [44], the centroid, and the design point from (6.15) and (6.16) are all shown on the stability region in Figure 6.8.

The response curves in Figure 6.9 compare the performance of the example controller, the centroid controller, and the controller designed from (6.15) and (6.16). Figure 6.3 shows where the reference input (R(s)) and the disturbance input (D(s)) are applied.

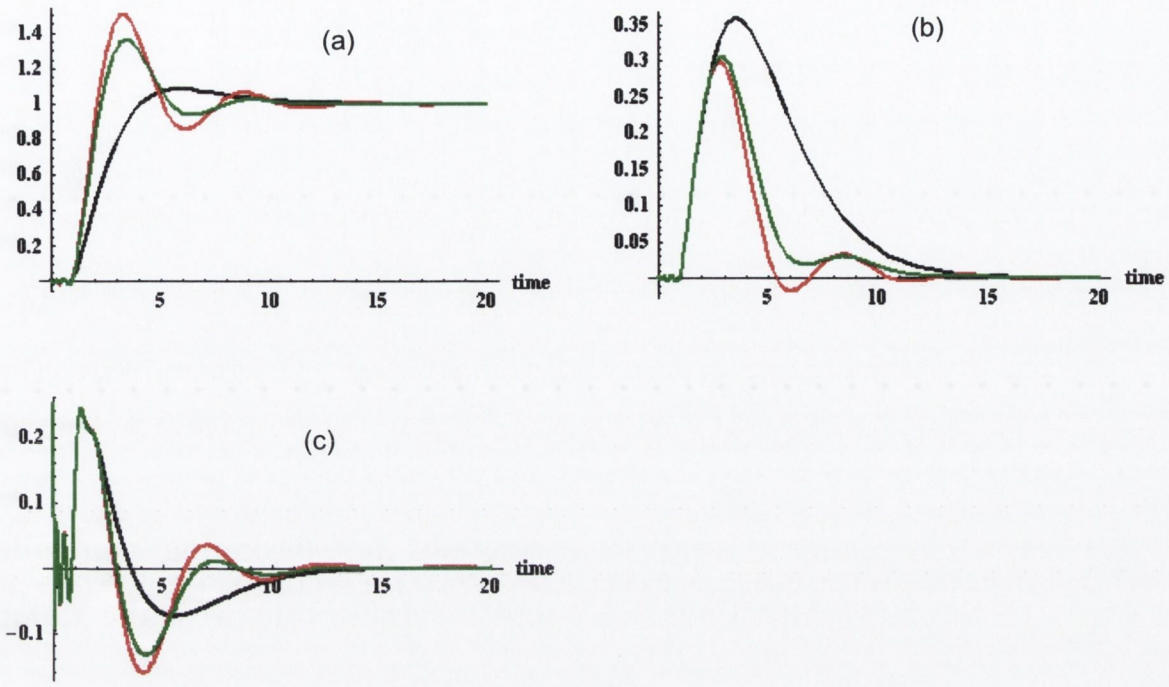


Figure 6.9 Step response and impulse response curves for systems with various controller designs. The red curve is for the system with a controller that has parameter values selected at the centroid of the region of stability. The green curve is for the system with the example controller. The black curve is for the system with a controller designed from optimum stability. (a) Step reference responses. (b) Step disturbance responses. (c) Impulse disturbance responses. These diagrams illustrate the fact that, with the exception of the step disturbance response (b), optimum stability based design has similar settling times to the two other designs, and lead to smoother responses and less undershoot and overshoot. In the case of (b) the optimum stability design response is smooth but has a higher overshoot than the other designs. The initial oscillations in these plots are artefacts of the Pade approximation to the pure time delay.

6.4 Discussion of PI controller designed using optimum stability for the FOLPD process

For the examples considered, and using standard robustness measures, we can say that for time delays of 0.25 s and greater controller designs based on (6.15) and (6.16) are superior to controllers designed using standard performance integrals or domain of stability considerations.

In the case of controllers designed using parameters at the centroid of a stability region, it seems that the appeal of the centroid as a design point is simply its maximum distance from the stability boundary. However, the choice of the centroid as a design point fails to take into account the degree of stability of the system at that point. The design point found using (6.15) and (6.16) has the advantage of not only being in the stability region but of occupying a point of optimum stability in the root locus sense. The robustness measures and response curves show the enhanced performance obtained by using the optimum stability point as a design point rather than the centroid.

In addition to their simplicity and ease of use (6.15) and (6.16), the fundamental design equations, emphasize the role of the ratio L/τ. In particular, it is not the magnitude of the delay that dictates the system performance but the ratio between the delay and the time constant of the process.

The design approach described in this chapter does not try to optimize the settling time. The settling time depends on the location of all the poles and all of the zeros. Since a time-delay system has an infinite number of poles, the problem of optimization of the settling time would almost certainly be intractable. By placing the rightmost eigenvalue as far to the left as possible we are minimizing the largest time constant $\frac{1}{|\delta|}$, where $|\delta|$ is the distance from the rightmost eigenvalue to the imaginary axis. $|\delta|$ is called the "degree of stability" [1] of the system. Our results indicate that minimizing the largest time constant has a favorable effect on the settling time.

6.5 The analytic root locus method for designing controllers for systems with time delay and a relationship with the Lambert W function

In this section I introduce an analytic method for designing controllers for time delay systems that is equivalent to the geometric method described earlier in this chapter. This analytic method can be used to design controllers for time-delay processes that lead to systems with optimum stability in the root locus sense.

We have two ways to plot root loci for time delay systems – the one based on the Bendrikov-Teodorchik equation [80] and the other is based on plotting rules [75] used earlier in this thesis.

There is nothing in the derivation of the Bendrikov-Teodorchik equation (also called the root locus equation) that excludes terms with factors such as e^{-sT}. Recall that the root locus equation [80] is given by:

$$\mathrm{Re}\{N(s)\}\,\mathrm{Im}\{M(s)\} - \mathrm{Re}\{M(s)\}\,\mathrm{Im}\{N(s)\} = 0 \tag{6.17}$$

Equation (6.17) describes a set of curves in the (σ,ω) plane and it may be derived from the requirement that k is a real number. These curves are the root locus of the characteristic equation:

$$p(s) = N(s) + k\,M(s) = 0 \text{ for } -\infty < k < \infty .$$

I begin the section with an illustrative example of plotting root loci using the analytic method and then in section 6.5.2 I apply this analytic method to the problem of PI control of an integrator with time delay.

Finally, in 6.5.5 I draw attention to potentially very useful equivalence between equation (6.17) and the Lambert W function.

6.5.1 Illustrative example of the analytic method from Palm [75]

Plot the root locus of the characteristic equation:

$$p(s) = N(s) + k\,M(s) = s + ke^{-Ts} = 0 \tag{6.18}$$

Here the root locus equation (6.17) gives:

$$e^{-\sigma T}(\omega Cos(\omega T) + \sigma Sin(\omega T)) = 0 \tag{6.19}$$

Equation (6.19) can be simplified to give:

$$\omega Cos(\omega T) + \sigma Sin(\omega T) = 0 \tag{6.20}$$

Figure 6.10 shows equation (6.20) plotted in the (σ, ω) - plane (with $T = 1$). Figure 6.10 agrees with the plot derived from plotting rules by Palm [75]

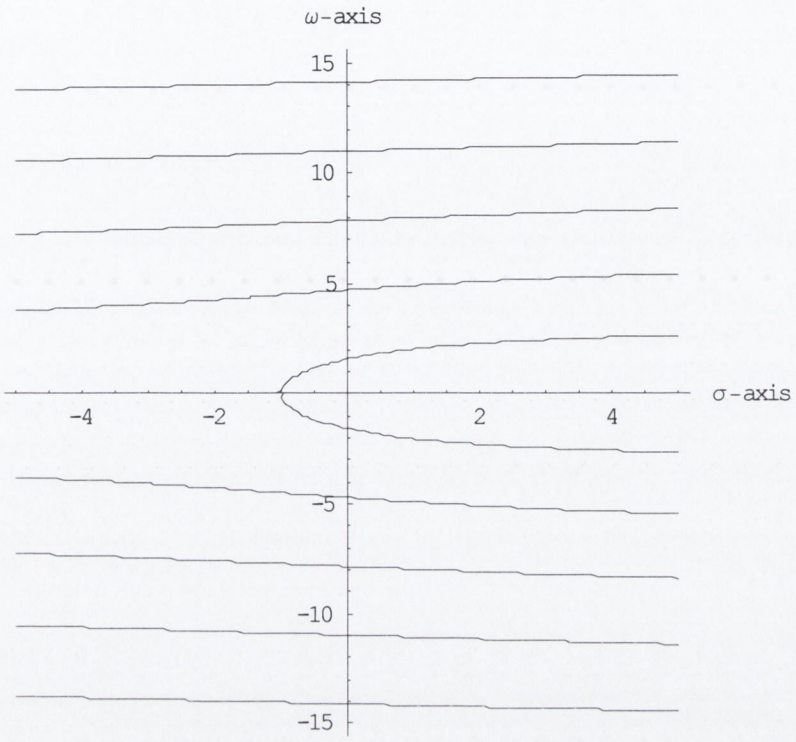


Figure 6.10 Root locus for the time delay system with characteristic polynomial $p(s) = s + ke^{-s}$ from [75]. The equation for branches of this root locus was derived using the Bendrikov-Teodorchik equation and it is the same as the one derived from plotting rules in [75].

6.5.2 Analytic method for designing PI controllers for an integrator with time delay

Processes modeled as an integrator with time delay are encountered in the literature. For example, the process $G(s) = \dfrac{e^{-0.2s}}{s}$ is used by Normey-Rico et al. [4] as a model of a mobile robot.

A PI controller $C(s) = k(s + F)/s$ and an integrator with time delay has characteristic equation:

$$p(s) = s^2 + k(s + F)e^{-sT} = 0 \tag{6.21}$$

Equation (6.17) gives the following root locus equation for $p(s)$ for $-\infty < k < \infty$:

$$\omega(2F\sigma + \sigma^2 + \omega^2)Cos(\omega T) + (F\sigma^2 + \sigma^3 - F\omega^2 + \sigma\omega^2)Sin(\omega T) = 0 \tag{6.22}$$

Checking for the case $T = 0$ gives the root locus equation: $\omega(2F\sigma + \sigma^2 + \omega^2) = 0$. We get $\omega = 0$ (as usual) or $2F\sigma + \sigma^2 + \omega^2 = 0$ i.e. a circle with radius F and centre $(-F, 0)$. This is also found using geometric methods. We can also see using the analytic method that this circle is part of the root locus for $\omega T = 2n\pi, n = 0,1,2,...$ because $Cos(\omega T) = 1$ and $Sin(\omega T) = 0$ at these points.

Rearranging (6.21) gives:

$$k(s) = \frac{-e^{sT}s^2}{F + s} \tag{6.23}$$

Calculating $\frac{dk(s)}{ds} = 0$ gives $s(Ts^2 + (FT+1)s + 2F) = 0$. This in turns gives possible breakpoints at: $s = 0, s = \frac{-1-FT-\sqrt{T^2F^2-6FT+1}}{2T}, s = \frac{-1-FT+\sqrt{T^2F^2-6FT+1}}{2T}$. We have $s \in \Re^1$ if F is such that $T^2F^2 - 6FT + 1 = 0$. This happens at the critical value of F given by:

$$F = \frac{0.171572875...}{T} \tag{6.24}$$

We can ignore the other possible value for F, i.e. $F = \frac{5.8284271247...}{T}$, as it leads to instability.

We also have critical value of s at which the breakpoint occurs:

$$s = \frac{-1-FT}{2T} \tag{6.25}$$

Finally, substituting equation (6.25) into (6.23) we can calculate the critical value of k as follows:

$$k_{critical} = -\frac{e^{-\frac{(1+FT)}{2}}(1+FT)^2}{2T(FT-1)} \tag{6.26}$$

At $T = 1$ equation (6.24) gives: $F = 0.1715...$; equation (6.25) gives $s = -0.5857...$; equation (6.26) gives $k = 0.4611...$ These values agree with the ones obtained using the method described in 6.1 and 6.2.

6.5.3 Evolution of the root locus for different values of F – illustration of the triple eigenvalue

For $F = 0.15$ i.e. F is below the critical value, the root locus equation (6.22) is:

$$\omega(0.3\sigma + \sigma^2 + \omega^2)Cos(\omega) + (0.15\sigma^2 + \sigma^3 - 0.15\omega^2 + \sigma\omega^2)Sin(\omega) = 0 \tag{6.27}$$

Equation (6.27) is plotted in Figure 6.11.

For $F = 0.171572875$ i.e. F is at the critical value, the root locus equation (6.22) is:

$$\omega(0.343146\sigma + \sigma^2 + \omega^2)Cos(\omega) + (0.171573\sigma^2 + \sigma^3 - 0.171573\omega^2 + \sigma\omega^2)Sin(\omega) = 0 \tag{6.28}$$

Equation (6.28) is plotted in Figure 6.12.

For $F = 0.18$ i.e. F is above the critical value, the root locus equation (6.22) is:

$$\omega(0.36\sigma + \sigma^2 + \omega^2)Cos(\omega) + (0.18\sigma^2 + \sigma^3 - 0.18\omega^2 + \sigma\omega^2)Sin(\omega) = 0 \tag{6.29}$$

Equation (6.29) is plotted in Figure 6.13.

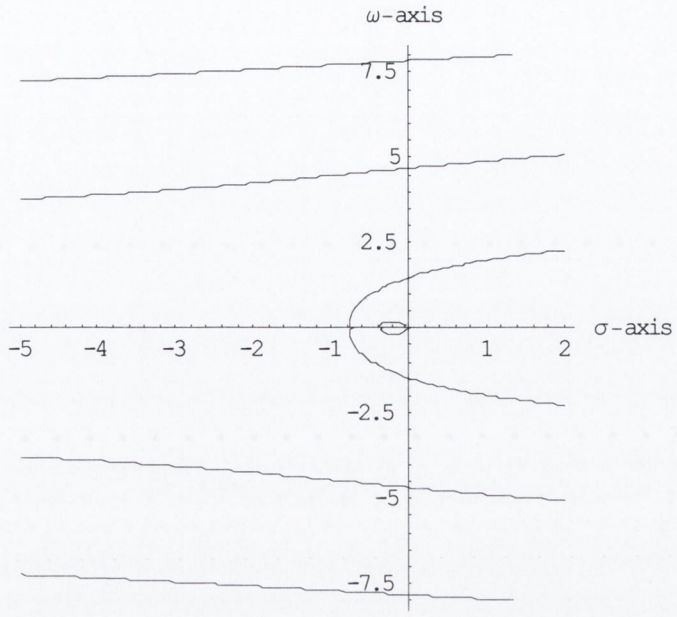


Figure 6.11 Primary of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.15$ which is just below the critical value.

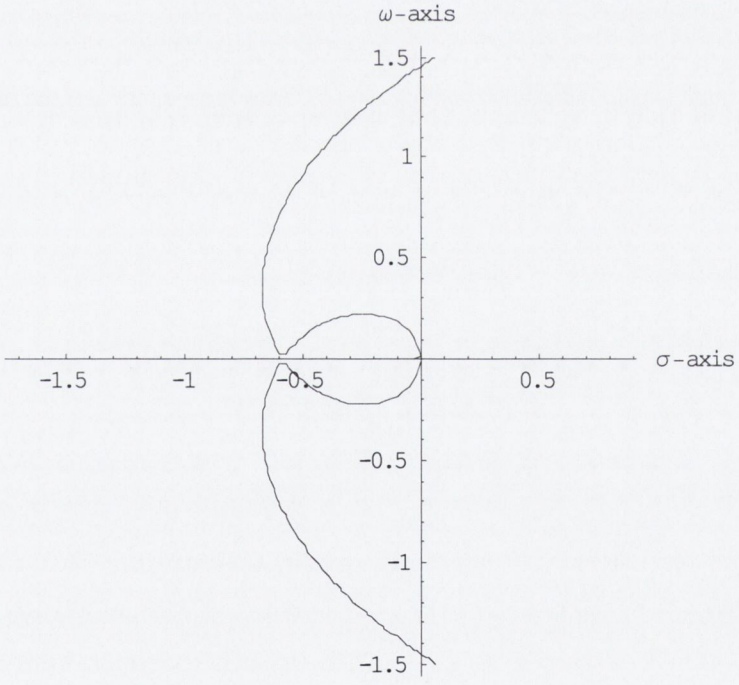


Figure 6.12 Primary branches of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.17157$ - which is the critical value. There is a triple eigenvalue at the breakpoint and the system is in a state of optimum stability at this point.

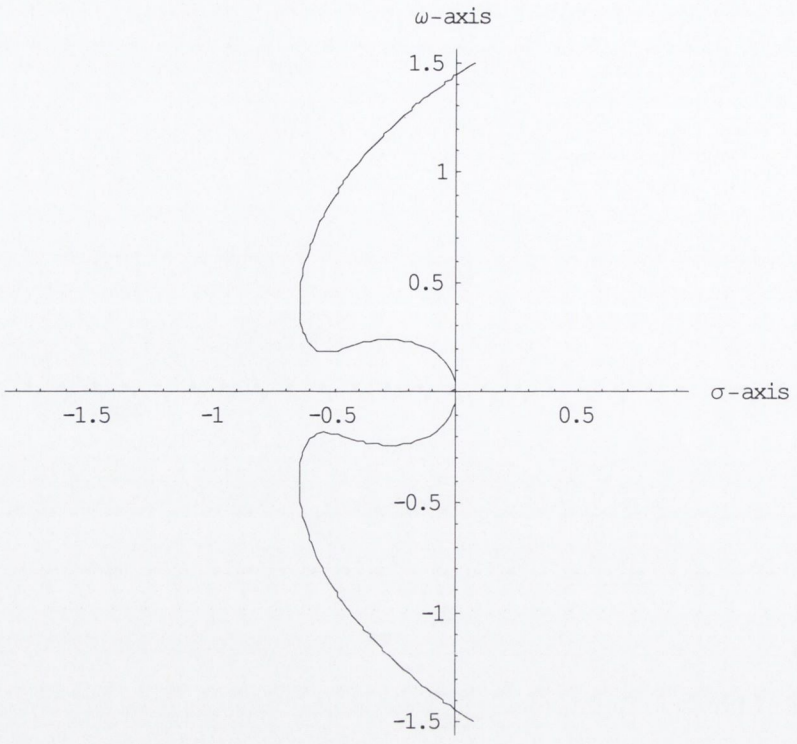


Figure 6.13 Primary branches of the root locus for the characteristic equation for a PI controller with an integrator process with time delay T=1, $p(s) = s^2 + k(s+F)e^{-s} = 0$. In this diagram $F = 0.18$ which is just above the critical value.

Figure 6.11, Figure 6.12, and Figure 6.13 show the evolution of the root locus of with the parameter F. The existence of a triple eigenvalue is evident in Figure 6.12.

6.5.4 An example of the gain equation

The gain equation [80] for equation (6.17) is:

$$-k = \frac{\mathrm{Re}\{N(s)\}\,\mathrm{Re}\{M(s)\} + \mathrm{Im}\{N(s)\}\,\mathrm{Im}\{M(s)\}}{[\mathrm{Re}\{M(s)\}]^2 + [\mathrm{Im}\{M(s)\}]^2} \tag{6.30}$$

For example, in the case of the root locus equation (6.22), the gain equation (6.30) becomes:

$$k(F^2 + 2F\sigma + \sigma^2 + \omega^2) + e^{T\sigma}(F\sigma^2 + \sigma^3 - F\omega^2 + \sigma\omega^2)Cos(\omega T) - e^{T\sigma}\omega(2F\sigma + \sigma^2 + \omega^2)Sin(\omega T) = 0 \tag{6.31}$$

The curve described by (6.31) is orthogonal to the root locus curve described by equation (6.22). When a value of k is selected, the gain curve (6.31) intersects the root locus curve (6.22) at the locations of the eigenvalues for that value of k [80].

6.5.5 A relationship between the Bendrikov-Teodorchik equation and the Lambert W function

The Lambert W function is the solution to equation (6.32):

$$z = W(z)e^{W(z)} \tag{6.32}$$

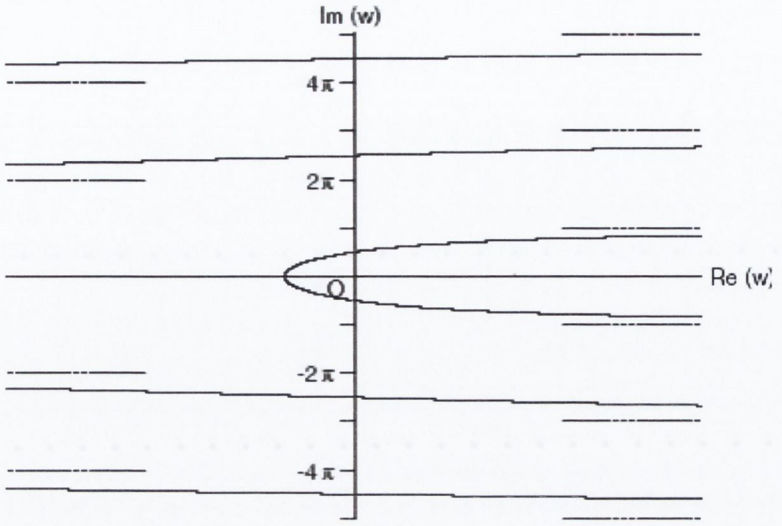


Figure 6.14 Plot of the Lambert W function from [81].

Figure 6.14 is very reminiscent of the root locus plot for time delay system in Figure 6.10.

The root locus equation for Figure 6.10 is given by equation (6.33):

$$\omega Cos(\omega T) + \sigma Sin(\omega T) = 0 \tag{6.33}$$

Equation (6.33) may be rewritten as:

$$\omega Cot(\omega T) + \sigma = 0 \tag{6.34}$$

Lambert W function is usually written in a form that is similar to equation (6.34) [81].

The knowledge that there is a relationship between the root locus equation for some time-delay systems and the Lambert W function appears not to be appreciated and could be very useful as this function appears in the analysis of many systems [82].

6.6 Calculating performance integrals for time delay systems

In this section I examine a method for calculating performance integrals of the type:

$$J_\alpha = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt \tag{6.35}$$

for time-delay systems. In sections 4.1.2 and 4.2.2 the performance integral given by equation (6.35) was found to have a minimum at the same parameter values that put the rightmost eigenvalue as far to the left as possible. It would be interesting to investigate if this also occurs in time-delay systems but one must first develop a method to calculate these integrals.

Parseval's Theorem states:

$\int_0^\infty f(t)g(t)dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} F(s)G(-s)ds$ where $F(s)$ is the Laplace transform of $f(t)$ and $G(s)$ is the Laplace

Transform of $g(t)$ i.e. $F(s) = \mathcal{L}\{f(t)\}$ and $G(s) = \mathcal{L}\{g(t)\}$. So, we have: $\int_0^\infty f^2(t)dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} F(s)F(-s)ds$

To calculate $\int_0^\infty t^q e^2(t)dt$ we use $\int_0^\infty t^q e^2(t)dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} F(s)E(-s)ds$ where $F(s) = \mathcal{L}\{t^q e(t)\} = (-1)^q \frac{d^q E(s)}{ds^q}$

So we can rewrite this integral as: $\int_0^\infty t^q e^2(t)dt = \frac{(-1)^q}{2\pi j}\int_{-j\infty}^{j\infty} \frac{d^q E(s)}{ds^q} E(-s)ds$

If we want to use Parseval's Theorem to evaluate an integral like: $\int_0^\infty \exp(\alpha t)\{e^2(t)\}dt$ we first write

the integral as: $\int_0^\infty \left\{\exp(\frac{\alpha}{2}t)e(t)\right\}^2 dt$ and then use can use Parseval's Theorem to say:

$$\int_0^\infty \left\{\exp(\frac{\alpha}{2}t)e(t)\right\}^2 dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \left\{\int_0^\infty \left\{\exp(-st)\exp(\frac{\alpha}{2}t)e(t)dt\right\}\int_0^\infty \left\{\exp(st)\exp(\frac{\alpha}{2}t)e(t)dt\right\}\right\}ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \left\{\int_0^\infty \left\{\exp(-(s-\frac{\alpha}{2})t)e(t)dt\right\}\int_0^\infty \left\{\exp(-(-s-\frac{\alpha}{2})t)e(t)dt\right\}\right\}ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} E(s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})ds$$

This result is already in the literature - see for example [83] and [84]. I have calculated these integrals using the Lyapunov matrix equation approach described by [85] and compared my results with the residue approach used by [83] and found that the two answers agree. However, these are the only examples of performance integrals for time delay systems that I could find. I could not find a description of how to calculate performance integrals for time delay systems of the type:

$$J_\alpha = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt$$

If we want to use Parseval's Theorem to evaluate J_α we first write the integral as:

$$J_\alpha = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt = \int_0^\infty \exp(\alpha t)e^2(t)dt + \frac{1}{\omega_n^2}\int_0^\infty \exp(\alpha t)\left(\frac{de(t)}{dt}\right)^2 dt$$

The first of these integrals i.e. $\int_0^\infty \exp(\alpha t)e^2(t)dt$, has already been evaluated above.

The second integral is rewritten as:

$$\frac{1}{\omega_n^2}\int_0^\infty \exp(\alpha t)\left(\frac{de(t)}{dt}\right)^2 dt = \int_0^\infty \left\{\frac{1}{\omega_n}\exp(\frac{\alpha}{2}t)\left(\frac{de(t)}{dt}\right)\right\}^2 dt$$

We can now use Parseval's Theorem to rewrite this as:

$$\int_0^\infty \left\{\frac{1}{\omega_n}\exp(\frac{\alpha}{2}t)\left(\frac{de}{dt}\right)\right\}^2 dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \left\{\int_0^\infty \left\{\frac{1}{\omega_n}\exp(-st)\exp(\frac{\alpha}{2}t)\left(\frac{de}{dt}\right)dt\right\}\int_0^\infty \left\{\frac{1}{\omega_n}\exp(st)\exp(\frac{\alpha}{2}t)\left(\frac{de}{dt}\right)dt\right\}\right\}ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \left\{\int_0^\infty \left\{\frac{1}{\omega_n}\exp(-(s-\frac{\alpha}{2})t)\left(\frac{de}{dt}\right)dt\right\}\int_0^\infty \left\{\frac{1}{\omega_n}\exp(-(-s-\frac{\alpha}{2})t)\left(\frac{de}{dt}\right)dt\right\}\right\}ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left\{\frac{1}{\omega_n}\left((s-\frac{\alpha}{2})E(s-\frac{\alpha}{2})-e(0)\right)\right\}\left\{\frac{1}{\omega_n}\left((-s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})-e(0)\right)\right\}ds$$

$$= \frac{1}{2\pi j}\left\{\int_{-j\infty}^{j\infty}E(s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})\,ds+\int_{-j\infty}^{j\infty}\frac{1}{\omega_n}\left[(s-\frac{\alpha}{2})E(s-\frac{\alpha}{2})-e(0)\right]\frac{1}{\omega_n}\left[(-s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})-e(0)\right]ds\right\}$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left\{E(s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})+\hat{E}(s-\frac{\alpha}{2})\hat{E}(-s-\frac{\alpha}{2})\right\}ds$$

We can now write:

$$J_\alpha = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left\{2\,E(s-\frac{\alpha}{2})E(-s-\frac{\alpha}{2})+\hat{E}(s-\frac{\alpha}{2})\hat{E}(-s-\frac{\alpha}{2})\right\}ds \tag{6.36}$$

Where: $\mathcal{L}\{e(t)\} = E(s) = \dfrac{B(s)}{A(s)}$ and $\hat{E}(s) = \dfrac{sB(s)-e(0)A(s)}{\omega_n A(s)}$.

Equation (6.36) was not used to calculate J_α as there was no clear way to select the initial conditions.

6.7 Conclusions

This chapter presents a new procedure for the design of PI controllers for general FOLPD processes. This procedure is based on the root locus method and a concept of optimum stability. We choose as our design point those parameter values that place the rightmost eigenvalue as far to the left as possible in the complex plane and we derive (6.15) and (6.16), two simple equations that allow the designer to calculate the controller parameter values using the process parameters only. When the controller parameter values are such that the system's rightmost eigenvalue is as far to the left as possible we say that the system is operating at a point of optimum stability in the root locus sense. We then apply this method to the design of PI controllers for specific FOLPD processes that are presently discussed in the literature. We compare PI controllers that were designed using optimum stability with PI controllers that were designed using a variety of performance integral and domain of stability considerations. By calculating gain margins, phase margins, delay margins, and plotting various response curves we see that the controllers that were designed using optimum stability offer, by these standard measures, enhanced performance when compared with the other controllers. The procedure presented extends a root-locus-based design procedure for delay-free processes described previously in chapters 3, 4, and 5, and in references [20] and [47].

We also describe the application of the analytic root locus to the design of a PI controller for an integrator with time delay and draw attention to a relationship between the root locus equation for a system with time delay and the Lambert W function.

6.8 Suggestions for further work

Investigate the application of the apparent link between Lambert W functions and the root locus for time-delay systems mentioned in section 6.5.5.

Evaluate equation (6.36) (possibly using step functions as initial conditions) and see if the parameter values that minimise it are the same parameter values that put the rightmost eigenvalue as far to the left as possible.

Chapter 7: PI and PID controller tuning by analogy with the Maximum Power Transfer Theorem of Circuit Theory

7.1 Introduction

Some years ago [26] I proved that nonlinear resistive loading of a series-wound, self-excited DC generator driven by a wind turbine, in such a way as to optimise power transfer from wind to electrical load, resulted in a very well damped dynamic response to varying wind speeds. This prompted me to explore whether there might be some other favorable consequences for control lying unexploited in results on optimum power transfer. This chapter presents a resulting new idea for tuning PI and PID controllers for a class of asymptotically stable processes, discovered by viewing the Maximum Power Transfer Theorem of linear AC circuit theory as a relation in a single loop, negative feedback system. This approach brings together ideas from the cognate subjects of Circuit Theory and Control Theory, continuing an old but often overlooked tradition (Truxal, [27]). It is a contribution to the many methods already available for designing PI and PID controllers – see O'Dwyer [28] and Datta et al. [29] for literature reviews.

Figure 7.1 shows a single-loop linear electric circuit, operating in steady state under sinusoidal excitation at angular frequency ω radians per second.

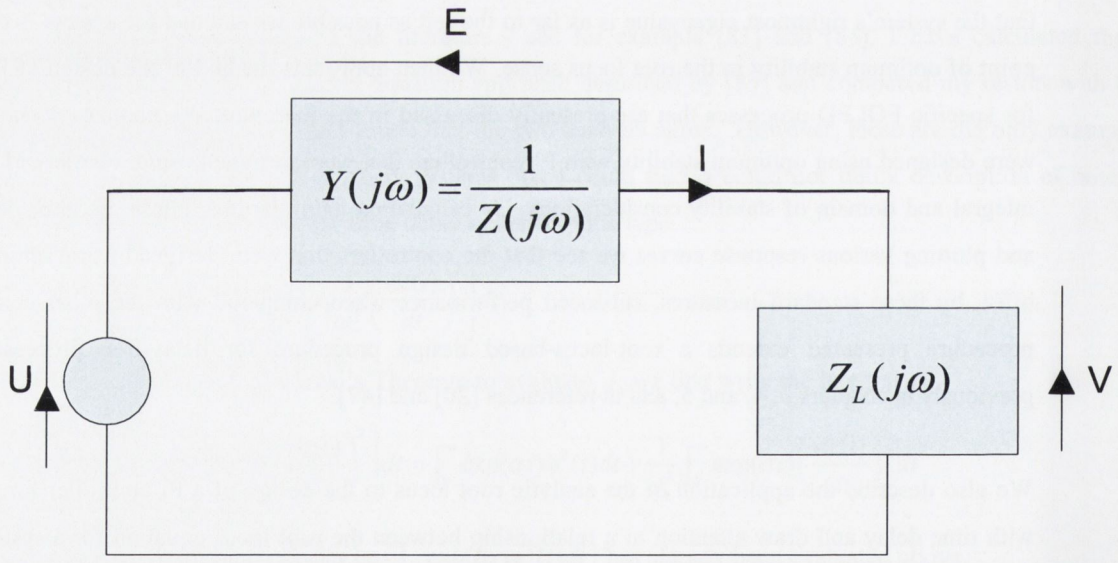


Figure 7.1 Single-loop electric circuit, to illustrate maximum power transfer theorem

The upper case quantities such $\tilde{V}$ as are phasor (complex number) representatives of real sinewaves. Thus, $\tilde{V} = |\tilde{V}| \exp(j\angle\tilde{V})$ is the phasor representative of the sinusoidal voltage $v = |\tilde{V}| \sin(\omega t + \angle\tilde{V})$. The source has sinusoidal emf and internal impedance $Z(j\omega)$ or, as we shall more conveniently characterise it below, admittance $Y(j\omega) = 1/Z(j\omega)$, where $Z(j\omega) = r + jx$ ohms. The Maximum Power Transfer Theorem states that, in order to maximise the mean power delivered to the load over any integral number of cycles, the

load impedance should have the value $Z_L(j\omega) = r - jx$, i.e., it should be the complex conjugate of the source impedance. In terms of source admittance, $Y(j\omega) = [r - jx]/[r^2 + x^2]$, this leads to magnitude and phase conditions

$$|Z_L(j\omega)| = 1/|Y(j\omega)| \quad \text{and} \quad \angle Z_L(j\omega) = \angle Y(j\omega) \tag{7.1}$$

Equation (7.1) gives the magnitude and phase relationships that must exist between the source admittance and the load impedance for maximum power transfer to occur between the source and the load.

From Figure 7.1 we write the relations

$$\begin{aligned} \tilde{V} &= Z_L(j\omega).\tilde{I} \\ \tilde{I} &= Y(j\omega).\tilde{E} \\ \tilde{E} &= \tilde{U} - \tilde{V} \end{aligned} \tag{7.2}$$

Equation (7.2) may be represented by the single-loop error-actuated feedback system shown on Figure 7.2 (see, for example, the same idea in a different context, D'Azzo and Houpis [86]).

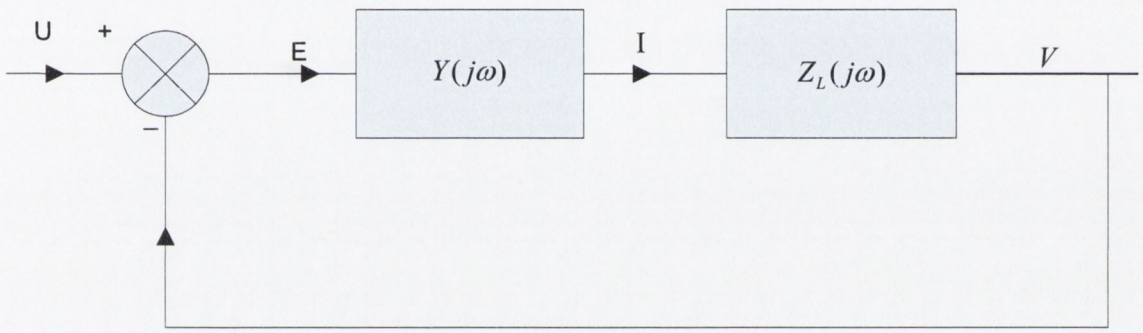


Figure 7.2 Electric circuit as a feedback loop

Figure 7.2 suggests that, in designing the controller $C(s)$ for the process $G(s)$ in Figure 7.3, we might explore the counterpart of equation (7.1), i.e., we might examine the possibility of specifying a design angular frequency ω in such a way that:

$$\begin{aligned} |C(j\omega)| &= 1/|G(j\omega)| \\ \angle C(j\omega) &= \angle G(j\omega) \end{aligned} \tag{7.3}$$

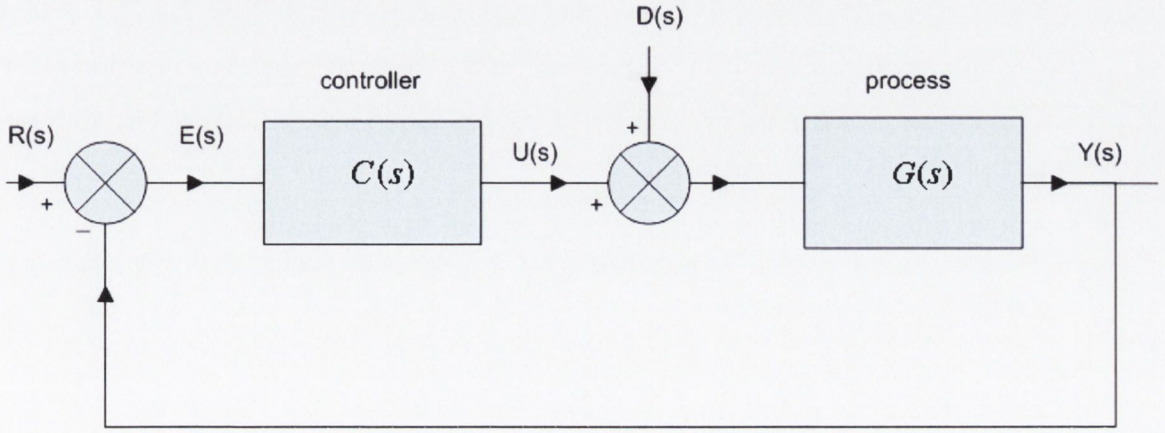


Figure 7.3 Single-loop feedback control system

Equation (7.3) shows that on invoking this idea, the gain of the controller would be the inverse of that of the process under control, at the chosen design frequency, and the phase shift introduced by the controller would be equal to that introduced by the process. In the section 7.4 a comparison is made between three controller designs - one based on an exploration of this idea, another on a root-locus based optimum stability, and a third due to Datta et al. [29].

7.2 Preliminary development of the theory

In this initial presentation, both for analytical convenience and for ease of comparison with a case study by Datta et al.[29], attention is restricted to process transfer functions of the form:

$$G(s) = \frac{k_2}{(s+b)^m} \tag{7.4}$$

with $k_2, b > 0$ and m is a positive integer. Equation (7.4) represents a cascade of m identical first order lags, each with time constant 1/b. The method applies to any asymptotically stable process, but this particular process offers the advantage of comparison with other designs.

For economy in notation, controller transfer functions are taken to have the form:

$$C(s) = \frac{k_1(s+a)^n}{s} \tag{7.5}$$

with $k_1, a > 0$ and n = 1,2.

The case n = 1 corresponds to PI control with integral action time T_i = 1/a:

$$C(s) = k_1\frac{(s+a)}{s} = k_1\left[1+\frac{1}{s\frac{1}{a}}\right] = k_1\left[1+\frac{1}{sT_i}\right]$$

The case n = 2 gives a restricted class of PID control with $T_i = \frac{2}{a}$ and derivative action time $T_d = \frac{1}{2a}$,

$$C(s) = k_1\frac{(s+a)^2}{s} = k_1\frac{s^2+2as+a^2}{s} = 2ak_1\left[1+\frac{1}{s\frac{2}{a}}+\frac{1}{2a}s\right] = 2ak_1\left[1+\frac{1}{sT_i}+T_d s\right]$$

It is interesting to note that in the PID case (i.e. $n = 2$) we have $T_d = T_i/4$. That is actually the classical Ziegler-Nichols relation [50] between T_d and T_i. As already stated in section 5.3.1 the Ziegler-Nichols PID controller is not a true three-term controller. In Ziegler–Nichols design we have $T_d = T_i/4$ so there are only two independent parameters - k and T_i. However, this is the structure required for root locus based design as we can fix one parameter and study root locus topology for the other parameter. Root locus based design would not be possible with three parameters.

It is convenient, and immediately interpretable in terms of Nyquist stability theory [48], to base our design procedure on choosing ω so that:

$$\angle G(j\omega) = -\frac{\pi}{2}+\frac{\phi}{2} \tag{7.6}$$

with the angle ϕ, $\{0 < \phi < \pi\}$ to be specified by the designer. For an asymptotically stable G(s) with G(0) > 0, such a choice is always possible. This leads, via equations (7.4) and (7.6) to:

$$\angle G(j\omega) = -\frac{\pi}{2} + \frac{\phi}{2}$$

$$= -m\tan^{-1}\left(\frac{\omega}{b}\right)$$

$$\Rightarrow \tan^{-1}\left(\frac{\omega}{b}\right) = \frac{\pi - \phi}{2m}$$

$$\Rightarrow \omega = b\tan\left(\frac{\pi - \phi}{2m}\right) \tag{7.7}$$

Equation (7.6) and the second line of equation (7.3): $\angle C(j\omega) = \angle G(j\omega)$ give:

$$\angle C(j\omega) = -\frac{\pi}{2} + \frac{\phi}{2}$$

$$= -\frac{\pi}{2} + n\tan^{-1}\left(\frac{\omega}{a}\right)$$

or $$n\tan^{-1}\left(\frac{\omega}{a}\right) - \frac{\pi}{2} = -\frac{\pi}{2} + \frac{\phi}{2} \tag{7.8}$$

Therefore:

$$\tan^{-1}\left(\frac{\omega}{a}\right) = \frac{\phi}{2n}$$

$$\frac{\omega}{a} = \tan\left(\frac{\phi}{2n}\right)$$

$$a = \frac{\omega}{\tan\left(\frac{\phi}{2n}\right)}$$

Finally, using equation (7.7) to substitute for ω we can evaluate the first controller parameter as:

$$a = b\frac{\tan\left(\frac{\pi - \phi}{2m}\right)}{\tan\left(\frac{\phi}{2n}\right)} \tag{7.9}$$

The first line of equation (7.3): $|C(j\omega)| = 1/|G(j\omega)|$ now yields:

$$\frac{k_1(\omega^2 + a^2)^{n/2}}{\omega} = \frac{(\omega^2 + b^2)^{m/2}}{k_2} \tag{7.10}$$

Once the phase margin ϕ and the controller structure (i.e. PI with $n = 1$ or PID with $n = 2$) have been chosen and ω has been evaluated from equation (7.7) and a from equation (7.9), k_1 follows from equation (7.10). We know the values of k_2, b and m from the process transfer function $G(s) = k_2/(s+b)^m$. An explicit equation for the second controller parameter, k_1, can be derived from equation (7.10) as follows:

$$\frac{k_1(\omega^2+a^2)^{n/2}}{\omega} = \frac{(\omega^2+b^2)^{m/2}}{k_2}$$

$$k_1 = \frac{\omega(\omega^2+b^2)^{m/2}}{k_2(\omega^2+a^2)^{n/2}}$$

$$= \frac{b^m\omega\left(1+\dfrac{\omega^2}{b^2}\right)^{m/2}}{k_2\, a^n\left(1+\dfrac{\omega^2}{a^2}\right)^{n/2}}$$

$$= \frac{b^m b\tan\left(\dfrac{\pi-\phi}{2m}\right)\left[1+\tan^2\left(\dfrac{\pi-\phi}{2m}\right)\right]^{m/2}\tan^n\left(\dfrac{\phi}{2n}\right)}{k_2 b^n\tan^n\left(\dfrac{\pi-\phi}{2m}\right)\left[1+\tan^2\left(\dfrac{\phi}{2n}\right)\right]^{n/2}}$$

$$k_1 = \frac{b^{m-n+1}}{k_2}\frac{\tan^n\left(\dfrac{\phi}{2n}\right)\left[1+\tan^2\left(\dfrac{\pi-\phi}{2m}\right)\right]^{m/2}}{\tan^{n-1}\left(\dfrac{\pi-\phi}{2m}\right)\left[1+\tan^2\left(\dfrac{\phi}{2n}\right)\right]^{n/2}} \tag{7.11}$$

These design equations give:

$$\angle G(j\omega)C(j\omega) = -\pi+\phi \tag{7.12}$$

Since $|G(j\omega)C(j\omega)| = 1$, equation (7.12) shows that the procedure works because the system has a specified phase margin ϕ, effective at the design angular frequency ω, which is extracted from the Nyquist diagram of $G(j\omega)$ at the phase angle given in equation (7.6). Specification of the phase margin—a classical robustness measure—is valuable, as it often gives sensitive control of the amount of damping in a system.

7.3 Comparison of PI and PID controller designs for the process $G(s) = 1/(s+1)^3$ based on maximum power transfer

Choose the process $G(s) = \dfrac{k_2}{(s+a)^m} = \dfrac{1}{(s+1)^3}$ i.e. $b = k_2 = 1$, and $m = 3$. Specify the phase margin $\phi = \pi/4$.

This leads to $\omega = \tan(\pi/8) = 0.4142$. The resulting controllers are:

$$\text{PI: } C(s) = \frac{0.4853(s+1)}{s} \tag{7.13}$$

$$\text{PID: } C(s) = \frac{0.1165(s+2.0824)^2}{s} \tag{7.14}$$

Responses of the process output to unit step reference and disturbance inputs are shown in Figure 7.4.

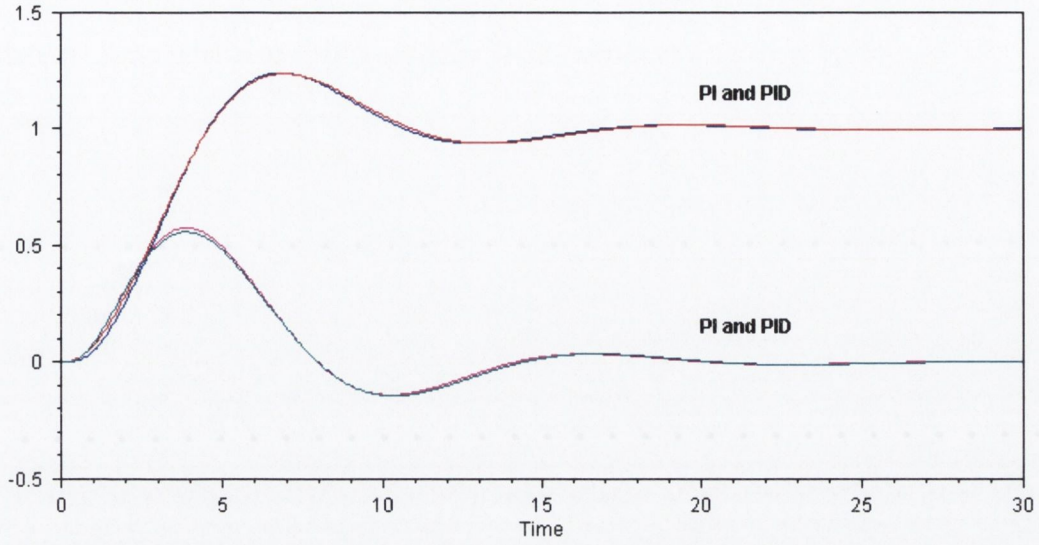


Figure 7.4 Responses of third order system under PI and PID control

In Figure 7.4 we see that PI and PID control are almost indistinguishable. This may at first sight seem surprising. However, the gain margins are so large—6.507 for PID control (equivalent to 16.267 dB), and 4.121 (equivalent to 12.300 dB) for PI—that the dominant indicator of damping is the phase margin, and this is the same, $\pi/4$, in both cases.

The frequency responses of C(s)G(s) are shown on Figure 7.5. These confirm asymptotic stability using the Nyquist criterion [48].

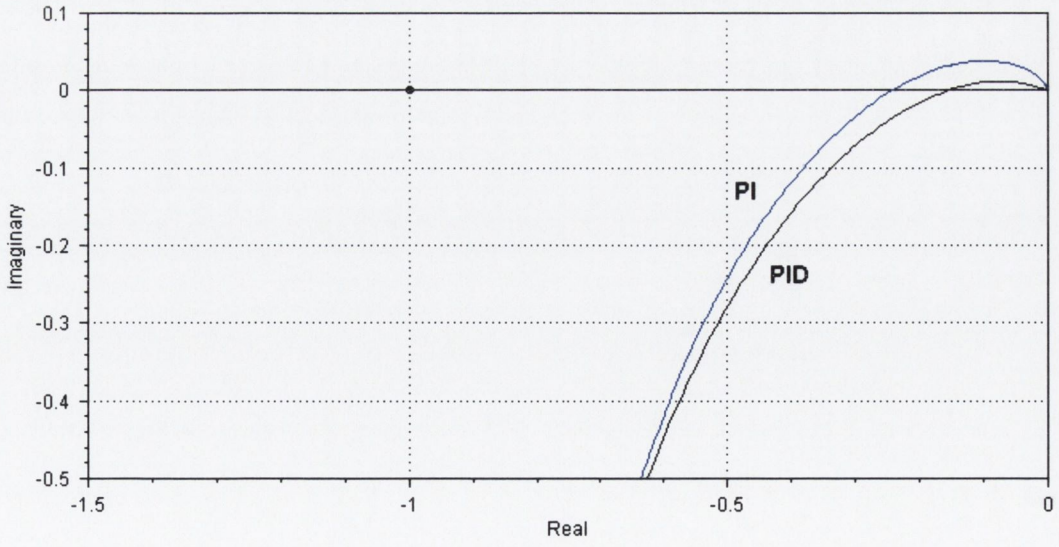


Figure 7.5 Open loop frequency response loci for the same systems as shown in Figure 7.4.

7.4 Comparison of PID controller designs based on optimum stability, max power transfer, and centroid of stability region for the process $G(s) = 1/(s+1)^8$

We now consider PID control of the process $G(s) = \dfrac{k_2}{(s+b)^m}$ with $k_2 = b = 1$ and $m = 8$. An interesting study of this system has been made by Datta [29] using optimum stability ideas in a parameter space. They wrote the PID controller transfer function in the form

$$C(s) = k_p + k_i/s + k_d s \tag{7.15}$$

For each chosen value of k_p they plotted a triangular domain of asymptotic stability in the (k_i, k_d) plane. They noted the radius of the largest circle that would just fit in this domain. They then searched for the value of k_p which maximised this radius, and chose as design parameters the values of k_i and k_d at the centre of this largest circle, along with the corresponding value of k_p. They found $k_p = 1.32759$, $k_i = 0.42563$, $k_d = 5.15291$. Corresponding process output responses to unit step reference and unit step disturbance, which are not shown by Datta et al. [29] are on Figure 7.7 and Figure 7.8.

In applying the present design idea to this process, we chose the phase margin $\phi = 56.04^o = 0.9781$ radians. (This specific value was motivated by comparison with an optimum stability design below.) The resulting "max power transfer" PID controller is:

$$C(s) = \frac{0.4636(s+0.5453)^2}{s} \tag{7.16}$$

Responses are compared on Figure 7.7 and Figure 7.8.

We also compare the present design with a PID controller based on a root locus-inspired principle of optimum stability [8, 9, 10], i.e., that the rightmost eigenvalue should, subject to structural relations between system parameters, lie as deep in the left half plane as possible.

The characteristic equation for the PID controller $C(s) = \dfrac{k_1(s+a)^2}{s}$ and the process $G(s) = \dfrac{k_2}{(s+b)^8}$ is:

$$p(s) = s(s+b)^8 + k(s+a)^2 . \tag{7.17}$$

where $k = k_1 k_2$. The root locus of equation (7.17) with respect to k is given in Figure 7.6.

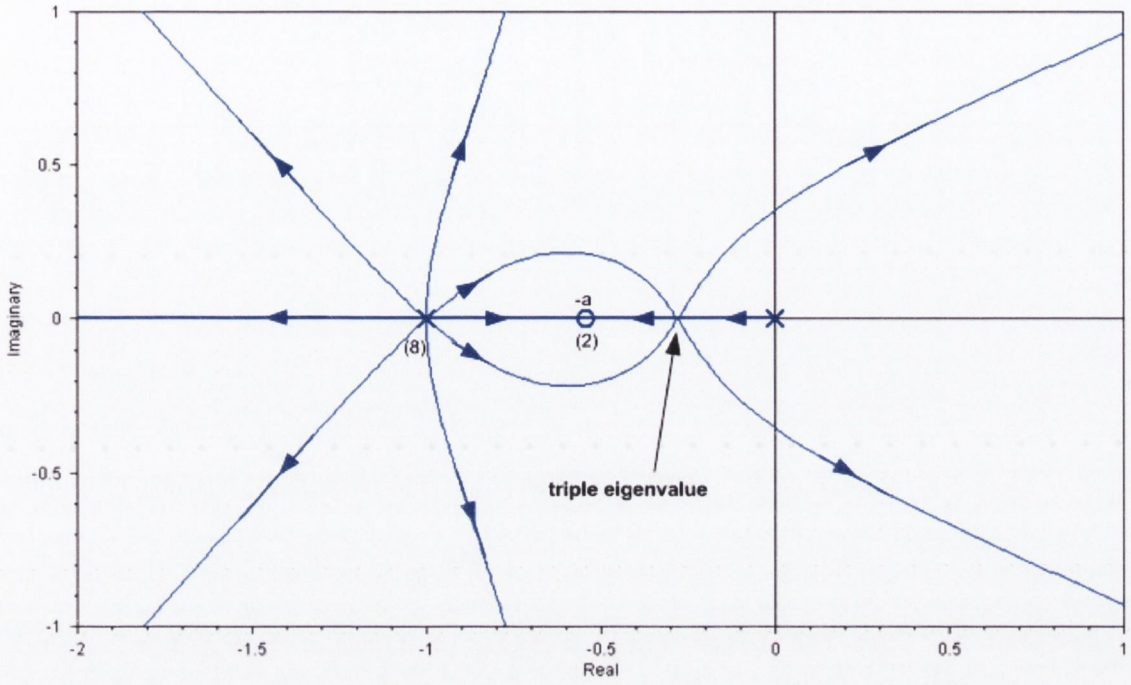


Figure 7.6 Root locus of equation (7.17) with respect to k. This diagram is to illustrate a root locus based design method. The nominal value of the parameter k is chosen as that value that results in a breakpoint at a triple eigenvalue. This is called a point of optimum stability in the root locus sense.

Following the general method described in section 5.3.5 the resulting root locus based optimum stability PID controller is: $C(s) = \dfrac{0.2874(s+0.5453)^2}{s}$ (7.18)

which differs only in gain from the "max power transfer" controller in equation (7.16). The fact that the gain in equation (7.18) is less than in equation (7.16) could not readily have been predicted before its calculation.

We have just computed that the value $a = 0.5453$ gives a triple eigenvalue in the root locus of the characteristic equation, plotted with respect to k, and deduced that designing for the value of k which places three eigenvalues at this breakpoint confers optimum stability. In this example, but not in all that we have studied, optimum stability yields real, equal, dominant eigenvalues—three in this case—thus generalizing the idea of critical damping in a second order system.

Once optimum stability considerations had led to the value of $a = 0.5453$ the corresponding phase margin ϕ and gain k_1 were evaluated as follows. In this case $n = 2$, and $m = 8$ so equation (7.9) gives:

$$a = b\frac{\tan\left(\dfrac{\pi-\phi}{2m}\right)}{\tan\left(\dfrac{\phi}{2n}\right)} = \frac{\tan\left(\dfrac{\pi-\phi}{16}\right)}{\tan\left(\dfrac{\phi}{4}\right)} \tag{7.19}$$

Given that $a = 0.5453$ equation (7.19) may be solved for ϕ by rearranging it as:

$$\phi_{m+1} = 4\tan^{-1}\left(\frac{\tan\left(\frac{\pi-\phi_m}{16}\right)}{0.5453}\right) \tag{7.20}$$

Equation (7.20) may be evaluated iteratively and converges to give $\phi = 56.04^o = 0.9781$ radians. Alternatively, the equation may be solved using Mathematica [49]. Knowing ϕ we can now use equation (7.11) to give $k_1 = 0.4636$.

The root locus of the characteristic equation with respect to k is shown on Figure 7.6. For $k_2 = 1$, $k_1 = 0.2874$ (optimum stability, as in equation (7.18)) the triple rightmost eigenvalue lies at the breakpoint, $s = -0.2791$, whereas for $k_1 = 0.4636$ (optimum power transfer analogy, equation (7.16)) the rightmost eigenvalues form the complex pair s = -0.1420 ± 0.2159.

It is interesting to note from Figure 7.7 that the optimum power transfer analogy and root locus-based optimum stability give the same settling time of approximately 30 seconds, but that the latter has no overshoot. The parameter plane idea invoked by Datta et al. gives quite an underdamped response, which has not settled in 100 seconds. With regard to disturbance rejection, as portrayed on Figure 7.8, the optimum power transfer analogy gives tighter control than root locus-based optimum stability, but at the expense of undershoot. The output excursion is more restricted with the Datta et al.[4] controller, but at the expense of a much longer settling time.

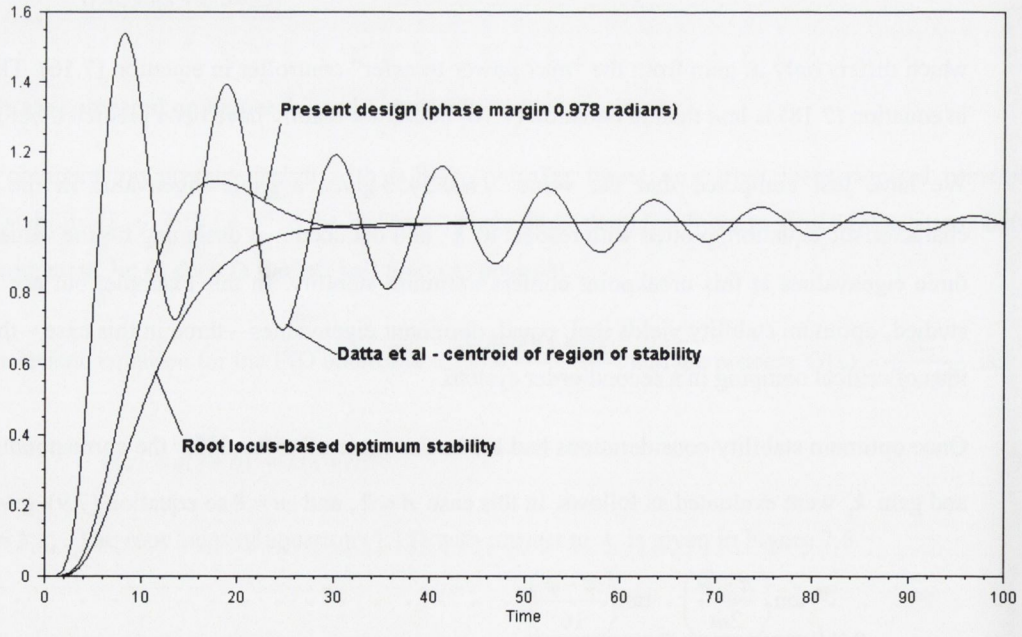


Figure 7.7 Comparison of three PID designs for eighth-order process, step reference input

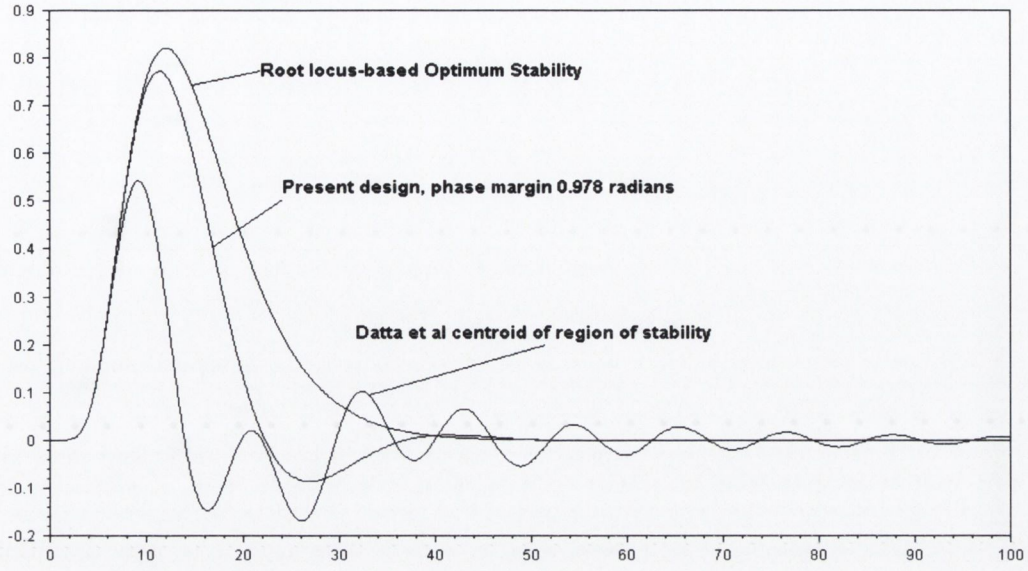


Figure 7.8 Comparison of three PID designs for eighth-order process, step disturbance input

7.5 Summary of results

A new idea for tuning PI and PID controllers has been presented, based on analogy with the maximum power transfer theorem from linear AC circuit theory. The approach has been identified as one that specifies the phase margin and the frequency at which it is effective. It has been illustrated by designs for third order and eighth order members of a restricted class of asymptotically stable processes, considered by Datta et al. [4]. Explicit formulas, involving the process parameters only, were derived for calculating the PI controller parameters for the process $G(s) = k_2 /(s+1)^m$. In the case m = 3, it is interesting to note that the performances of the PI and PID controllers are indistinguishable, both for reference input following and for disturbance rejection, despite significant differences in gain margin. In the case m = 8, the performance is similar in time scale to, though distinguished in overshoot (reference tracking) and undershoot (disturbance rejection) from, controllers designed by a root locus-based optimum stability approach. Controllers designed by an optimum parameter space approach [4] give a much more oscillatory behavior and longer settling time.

An interesting observation is that the controller designed by using max power transfer considerations: $C(s) = 0.4636(s+0.5453)^2 / s$ is very similar to the one designed from root locus based optimum stability considerations $C(s) = 0.2874(s+0.5453)^2 / s$ - the difference being in the value of the gains.

7.6 Suggestion for further work

Investigate the similarity between the controller in equation (7.16), designed from max power transfer, and the controller in equation (7.18), designed using root locus based optimum stability.

Chapter 8: Nyquist and Lyapunov based optimum stability in modeling the control of balance during quiet standing

8.1 Introduction

Dizziness is one of the most common complaints that patients bring to their doctors with about 40% of adults experience clinically significant dizziness at some time in their lives [87] - this makes the study of the human balance control system an important one. Analysis of unsupported standing has been of interest for some time and has received much attention from engineers, e.g. see [88].

Maintaining balance requires the brain to utilize inputs from at least three systems - the visual system, the muscular system, and the vestibular system [87]. The vestibular organs of the inner ear consist of two major subsystems. The first subsystem consists of semicircular canals that are filled with a liquid and when the head is nodded up and down tiny pressure differences between the ends of the canal generates a signal to the brain that indicates angular acceleration. So the semicircular canals sense rotational movement. A second subsystem - the otolith apparatus – consists of a pebble-sized bone embedded in a jellylike substance and floating on hairs. These hairs project out of sensory cells and the signal from those hairs that report the greatest load from the otolith is accepted by the brain as the "down" direction. The otolith apparatus responds to linear accelerations; it senses linear motion and orientation with respect to gravity.

In [8] de Paor et al. present a model of the human balance control system during quiet standing and determine regions of stability using Routh-Hurwitz and Nyquist techniques. They then chose the operating point for their model to ensure optimum stability in the sense that this point is at the centroid of the region of stability thus optimizing Vector Margin [20]. By assuming that the human balance control system is a compromise between optimally stable visual / somatosensory and optimally stable vestibular designs, a very close match is found between the predictions of their model and data gathered from subjects. Also, the authors of [8] excited their model with Gaussian random noise and compared its behavior with data collected from subjects and found an excellent correlation.

In this chapter a model of the human balance control system is developed. This model is a modification of the model described by de Paor et al. [8]. A tenth order characteristic equation with four parameters arises in the modified model of the human balance control system. Two methods based on optimum stability are used to select these operating parameters. The first method, described in section 8.3, involves setting two parameters to zero and then selecting the remaining pair of parameters using the Nyquist diagram. In this way, pairs of parameters that give an optimum phase margin are selected as the design parameters. An alternative parameter selection method based on the Lyapunov equation is used in section 8.4. This method allows the selection of all four parameters simultaneously and is based on using the Lyapunov equation to optimize eigenvalue location. A comparison is then made between the behavior of the model at operating points derived from optimum vector margin [8], optimum phase margin, and optimum eigenvalue location.

8.2 The characteristic equation of the model

The fundamental physical model used in [8] is the inverted pendulum and an average human being [89] with mass 74.2kg, height of centre of gravity 0.93m, moment of inertia for saggital sway 73.27kgm^2. A block diagram showing the transfer functions of the various components is given in Figure 8.2.

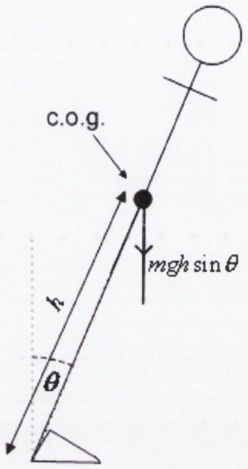


Figure 8.1 Human body represented by a single-link inverted pendulum free to move only about the ankle. The centre of gravity (c.o.g.) is at a length h from the ankle. The body is maintained in an upright position by a torque about the ankle that opposes the torque due to the weight of the body.

Taking clockwise as positive, we can write the equation of motion of the model in Figure 8.1:

$$J\frac{d^2\theta}{dt^2} = mghSin(\theta) - F\frac{d\theta}{dt} - \hat{f}_6 \tag{8.1}$$

where $F\frac{d\theta}{dt}$ is the viscous torque due to stretching of the muscles and $\hat{f}_6$ is the torque due to contraction of the muscles. For small θ (8.1) becomes $\frac{d^2\theta}{dt^2} + \frac{F}{J}\frac{d\theta}{dt} - \frac{mgh}{J}\theta = -\frac{\hat{f}_6}{J}$. On substituting the parameter values given above and using $\frac{F}{J} = 2$ from [90] (8.1) becomes $\frac{d^2\theta}{dt^2} + 2\frac{d\theta}{dt} - 9.24\theta = -f_6$ - where f_6 is the torque per unit moment of inertia. This may be written in transfer function form as: $\left[s^2 + 2s - 9.24\right]\theta(s) = -f_6(s)$ or $\left[(s+4.2)(s-2.2)\right]\theta(s) = -f_6(s)$.

Figure 8.2 shows the block diagram of a model of the human balance control system. This model is essentially the one developed Delaney [91] except that Delaney used a 150ms time delay in the vision loop and zero time delay in the path to the upper summing point. In the model used here, the 150ms delay is redistributed between the vision loop (115ms) and the path to the upper summing point (35ms). This is in line with results in the literature [90].

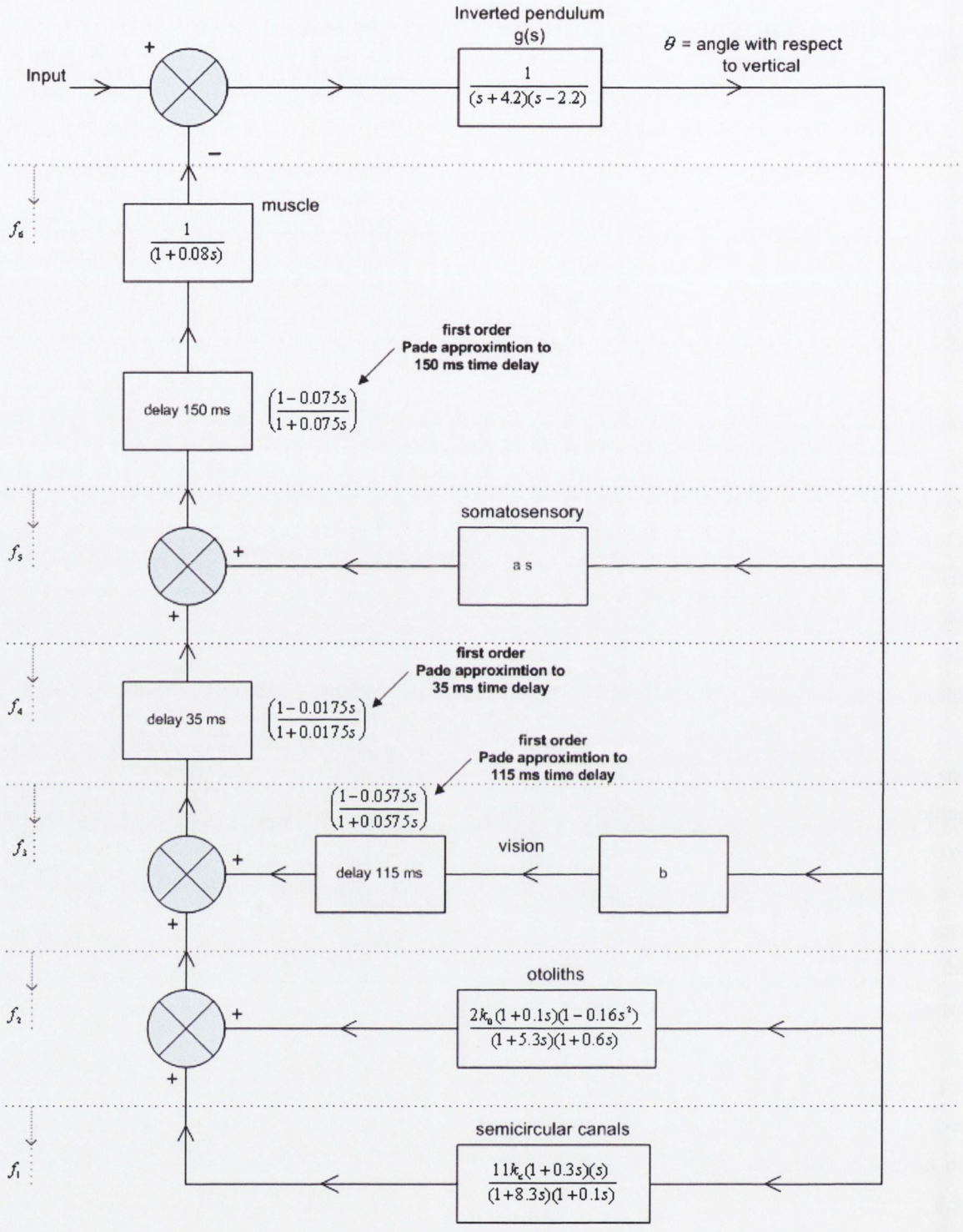


Figure 8.2 Block diagram of the human balance control system. The controller consists of four feedback loops that model the semicircular canals, otolith, vision, and somatosensory systems. The transfer functions for these four subsystems are taken from [91] although the time delays have been modified. The control loops are combined into a single transfer function $f_6(s)$ that represents the muscle torque per unit moment of inertia about the ankles. (a, b, k_c, k_0) are the four parameters that must be selected.

The transfer function $f_6(s)$ that represents the restoring torque about the ankles may be built up as follows:

$$f_1 = \frac{11k_c(1+0.3s)(s)}{(1+8.3s)(1+0.1s)};\ f_2 = \frac{2k_0(1+0.1s)(1-0.16s^2)}{(1+5.3s)(1+0.6s)} + f_1;\ f_3 = b\left(\frac{1-0.0575s}{1+0.0575s}\right) + f_2;\ f_4 = \left(\frac{1-0.0175s}{1+0.0175s}\right) f_3;$$

$$f_5 = as + f_4;\ f_6 = \left(\frac{1-0.075s}{1+0.075s}\right)\left(\frac{1}{(1+0.08s)}\right) f_5$$

Combining these transfer functions gives:

$$f_6 = \left(\left(\frac{11k_c(1+0.3s)(s)}{(1+8.3s)(1+0.1s)} + \frac{2k_0(1+0.1s)(1-0.16s^2)}{(1+5.3s)(1+0.6s)} + \frac{1-0.0575s}{1+0.0575s} b \right) \left(\frac{1-0.0175s}{1+0.0175s} \right) + as \right) \left(\frac{1-0.075s}{1+0.075s} \frac{1}{(1+0.08s)} \right)$$

$f_6(s)$ represents the muscle torque per unit moment of inertia about the ankles.

The disturbance transfer function is $w(s) = \dfrac{g(s)}{1+g(s)\,f_6(s)}$

The characteristic polynomial is: $p(s) = 1+g(s)\,f_6(s)$ (8.2)

8.3 Selecting pairs of parameters to optimise phase margin on the Nyquist diagram

8.3.1 Nyquist analysis with visual / somatosensory system only active

With $k_c = k_0 = 0$, i.e. the subject might have brain damage and receives no information from the inner ear; $p(s)$ reduces to $p_q(s)+ap_1(s)+bp_2(s)=0$. If a is fixed then $p(s)$ may be rearranged to give: $\dfrac{-bp_2(s)}{p_q(s)+ap_1(s)}+1=0$. Two series of Nyquist plots were created using Mathematica [92]. From these plots, two tables of values of a and b were drawn up. One set of results was generated by fixing a and then reading that value of b that gave optimum phase margin from the Nyquist plot of $p_b = p_2/(p_q+ap_1)$. This procedure was repeated by fixing b and taking a for optimum phase margin from the Nyquist plot of $p_a = p_1/(p_q+bp_2)$. These results are summarised in Table 8.1. Figure 8.3 shows a sample Nyquist diagram.

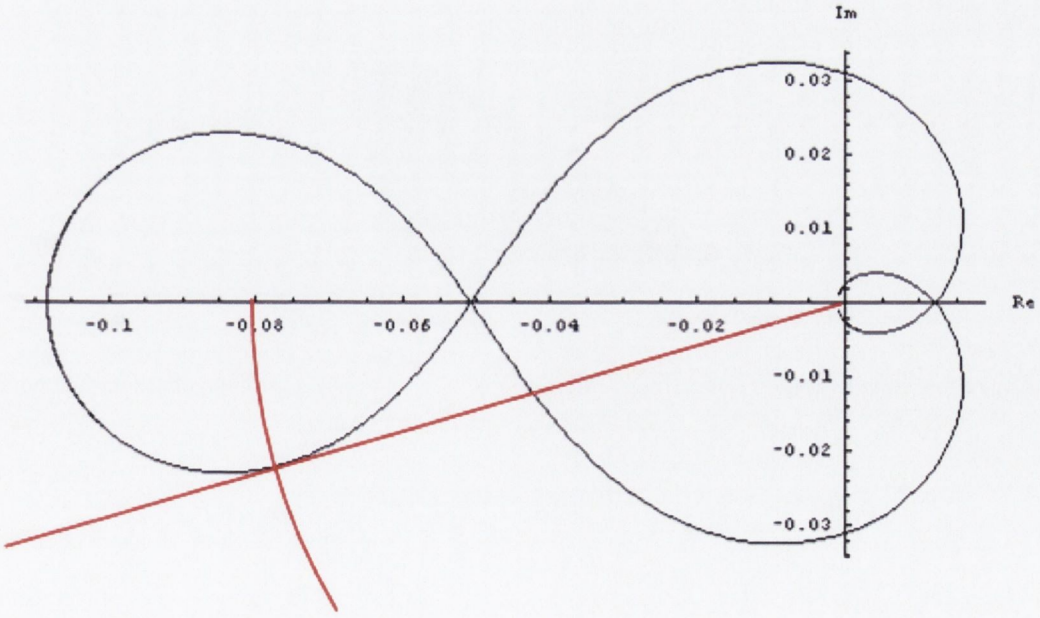


Figure 8.3 Nyquist diagram for $p_b = p_2/(p_q+ap_1)$, $a=5.0$ so $b=1/0.08=12.5$ at optimum phase margin.

The two sets of a and b values obtained in this way are given in Table 8.1 and they are plotted in the stability region on the parameter plane in Figure 8.4.

a	b	b	a
3	10.75	10	5.56
3.5	11.24	11	5.62
4	11.76	12	5.95
4.5	12.25	12.5	6.33
5	12.50	13	6.45
5.5	13.89	14	6.62
6	14.71	14.5	6.76
6.5	15.38	15	7.04
7	16.67	15.5	7.19
7.5	17.24	16	7.24
8	18.18	16.5	7.58
8.5	19.23	17	7.69
9	20.83	17.5	7.81
9.5	21.74	18	8
10	22.22	19	8.26
10.5	26.32	20	8.47

Table 8.1 Optimum phase margin points - Visual / Somatosensory System only active.

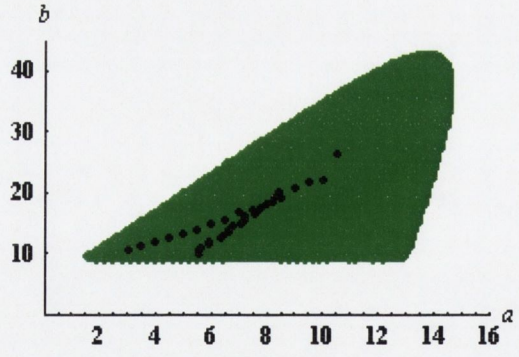


Figure 8.4 Stability region shown in green for p(s) with $k_0 = k_c = 0$. Optimum phase margin points from Table 8.1 are shown as black dots.

8.3.2 Nyquist analysis with vestibular system only active

With $a = b = 0$ (i.e. the subject has no visual or other sensory clues) $p(s)$ reduces to $p(s) = p_q(s) + k_0 p_3(s) + k_c p_4(s)$. Just as for the a and b parameters above, two series of Nyquist plots were created. From these plots two tables of values of k_0 and k_c were drawn up. One set of results was generated by fixing k_0 and reading that value of k_c that gave optimum phase margin from the Nyquist plot of $p_{k_c} = p_4/(p_q + k_0 p_3)$. This procedure was repeated by fixing k_c and taking k_0 for optimum phase margin

from the Nyquist plot of $p_{k_0} = p_3/(p_q + k_c p_4)$. The two sets of k_c and k_0 values obtained in this way are given in Table 8.2. These values are plotted in the stability region on the (k_c, k_0) parameter in Figure 8.5.

k_0	k_c	k_c	k_0
6	9.35	8.5	4.44
6.5	9.43	9	5.26
7	9.62	9.5	6.45
7.5	9.71	10	7.69
8	9.80	10.5	8.93
8.5	9.90		
9	10.0		
9.5	10.2		
10.5	10.31		
11	10.42		
11.5	10.53		
12	10.64		
12.5	10.75		
13	10.87		

Table 8.2 Optimum phase margin points - Vestibular System only active.

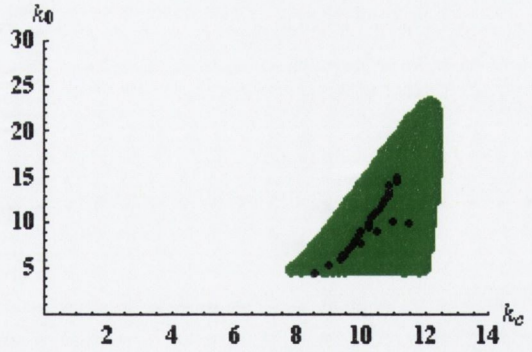


Figure 8.5 Stability region shown in green for p(s) with $a = b = 0$. Optimum phase margin points from Table 8.2 are shown as black dots.

8.3.3 Dynamics of the model when two pairs of operating points are combined

Pairs of operating points obtained for (a,b) and (k_c, k_0) were combined to form sets of points (a,b,k_c,k_0). Some of these combinations lead to a characteristic equation that has roots in the right half plane: this is due to the simplification used when the two cases of $a = b = 0$ and $k_c = k_0 = 0$ were studied. Other combinations lead to a stable system: the nineteen combinations of (a,b,k_c,k_0) from Table 8.1 and Table 8.2 that lead to a stable system are given in Table 8.3.

a	b	k_c	k_0	location of rightmost eigenvalue
5.56	10.0	8.5	4.44	-0.07
5.56	10.0	9.0	5.26	-0.08
5.56	10.0	9.5	6.45	-0.07
5.56	10.5	9.35	6.0	-0.06
5.56	10.5	9.43	6.5	-0.05
5.56	10.5	9.62	7.0	-0.01
5.56	10.5	9.71	7.5	-0.0009
5.56	10.5	8.5	4.44	-0.07
5.56	10.5	9.0	5.26	-0.07
5.56	10.5	9.5	6.45	-0.03
5.62	11.0	8.5	4.44	-0.07
5.62	11.0	9.0	5.26	-0.08
5.95	11.5	8.5	4.44	-0.07
5.95	11.5	9.0	5.26	-0.05
5.95	12.0	8.5	4.44	-0.07
5.95	12.0	9.0	5.26	-0.01
6.33	12.5	8.5	4.44	-0.07
6.45	13.0	8.5	4.44	-0.05
6.63	13.5	8.5	4.44	-0.01

Table 8.3 Nineteen combinations of parameters (a,b,k_c,k_0) from Table 8.1 and Table 8.2 that lead to stability. The values for (a,b) were obtained by setting $k_c = k_0 = 0$ and finding the values of (a,b) that give optimum phase margin. The values for (k_c,k_0) were obtained by setting $a = b = 0$ and finding the values of (k_c,k_0) that give optimum phase margin. These combinations of points are very artificial; there is no relationship between the pairs of points other than they lead to characteristic polynomials with eigenvalues in the left half plane. Nature would select the operating point as a set of four rather than in pairs. The operating points in this Table lead to very oscillatory impulse responses.

We select the point $(a,b,k_c,k_0) = (5.56,10.0,8.5,4.44)$ from Table 8.3 and show its impulse response in Figure 8.6. The impulse response is very oscillatory – a feature of all nineteen operating points in Table 8.3.

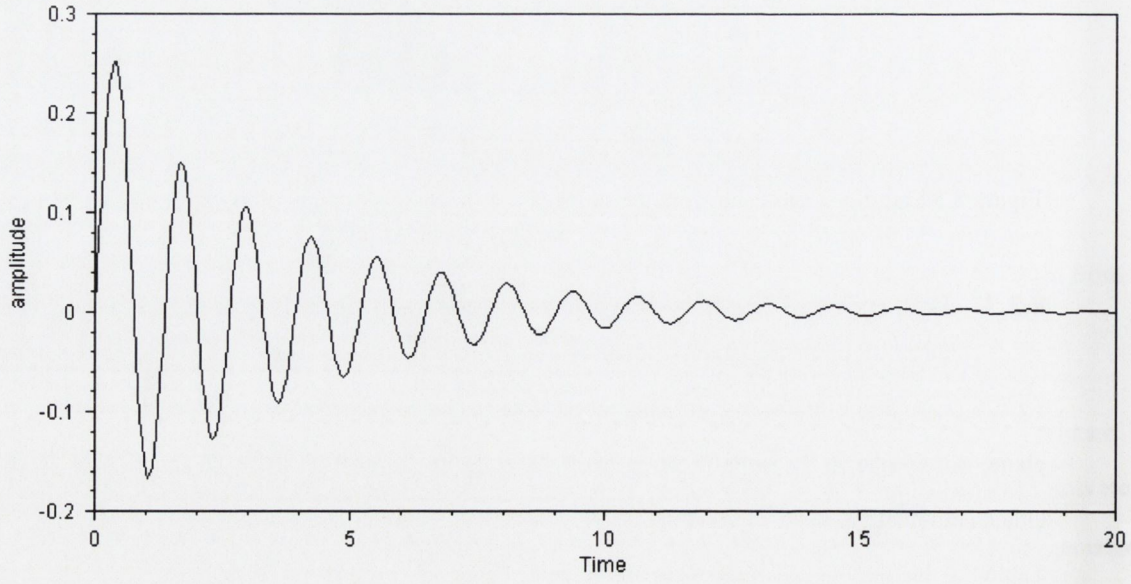


Figure 8.6 Impulse response for the system in Figure 8.2 with $(a,b,k_c,k_0) = (5.56,10.0,8.5,4.44)$ - parameters selected by from Table 8.3. This is the least oscillatory impulse response using parameters in Table 8.3.

The settling time for the impulse response in Figure 8.6 is quite long – about 20 seconds. So the point $(a,b,k_c,k_0) = (5.56,10.0,8.5,4.44)$ from Table 8.3, a combination of optimum phase margin points that gives the best impulse response, has lead to a system that is stable but not to one with behavior that corresponds to our experience.

In [8] the authors explored the possibility that human balance is governed by a compromise between optimum stability visual / somatosensory and optimum stability vestibular designs. Guided by this, they derive two points from a combination of vector margin and root locus techniques. These points are: $P(1) = (a,b,k_c,k_0) = (6.763,14.76,0,0)$ and $P(2) = (a,b,k_c,k_0) = (0,0,14,7.3)$. Three operating points for the system were taken at $P = P(1) - g(P(2) - P(1))$ for g = 0, 0.3, and 0.8. The resulting operating points are: $(a,b,k_c,k_0) = (6.76,14.76,0,0)$; $(4.73,10.33,4.2,2.19)$; $(1.35,2.95,11.2,5.84)$. Although their analysis applies only to their model and not to the modified version studied here, it is still interesting to see the behavior of the modified model at these points. The impulse response of the system at the point corresponding to $g = 0.3$ is given in Figure 8.7.

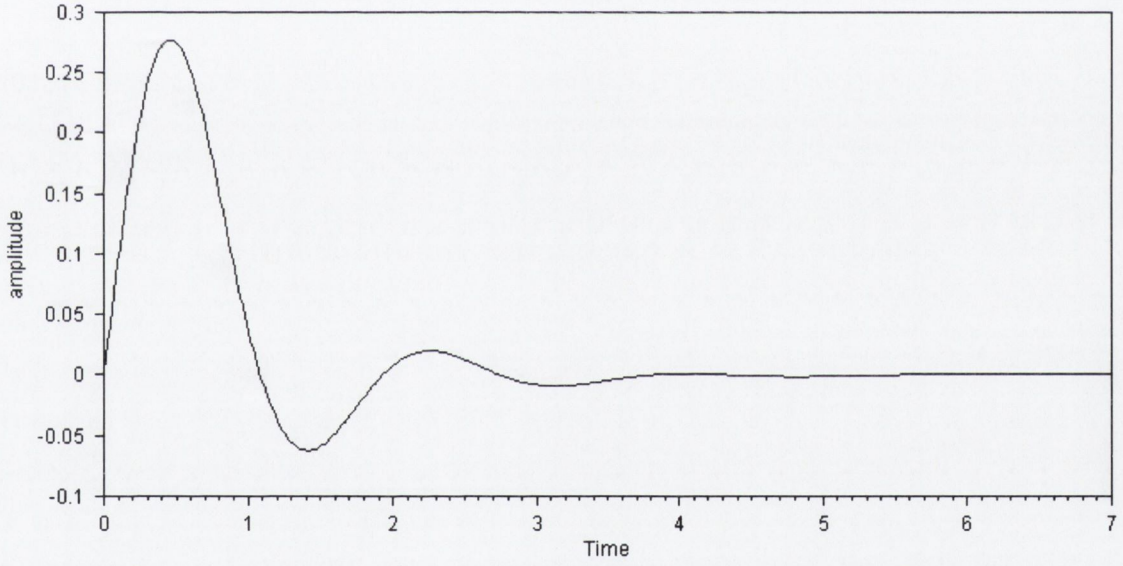


Figure 8.7 Impulse responses for the system at the operating point $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ from [8]. This point was selected in [8] by recalling that human balance is governed by a compromise between optimum stability visual / somatosensory and optimum stability vestibular designs. When $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ the rightmost eigenvalues of the quiet standing polynomial is at –0.07.

When $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ the rightmost eigenvalue of the characteristic polynomial is at – 0.07. An excellent response is obtained when $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ and recovery of the upright position is achieved in about 4 seconds. This is better than the 20 seconds required when the point $(a,b,k_c,k_0) = (5.56,10.0,8.5,4.44)$ obtained from optimum phase margin considerations, is used. However, the second point $(a,b,k_c,k_0) = (1.35,2.95,11.2,5.84)$ given by [8] leads to a highly oscillatory impulse response in the modified model.

Using the point $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ as a starting point of an optimization routine leads to $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ with impulse response given in Figure 8.8. The impulse response in Figure 8.8 is very smooth and recovery takes place in about 3 or 4 seconds. The operating point $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ seems to make to model behave in a very realistic way.

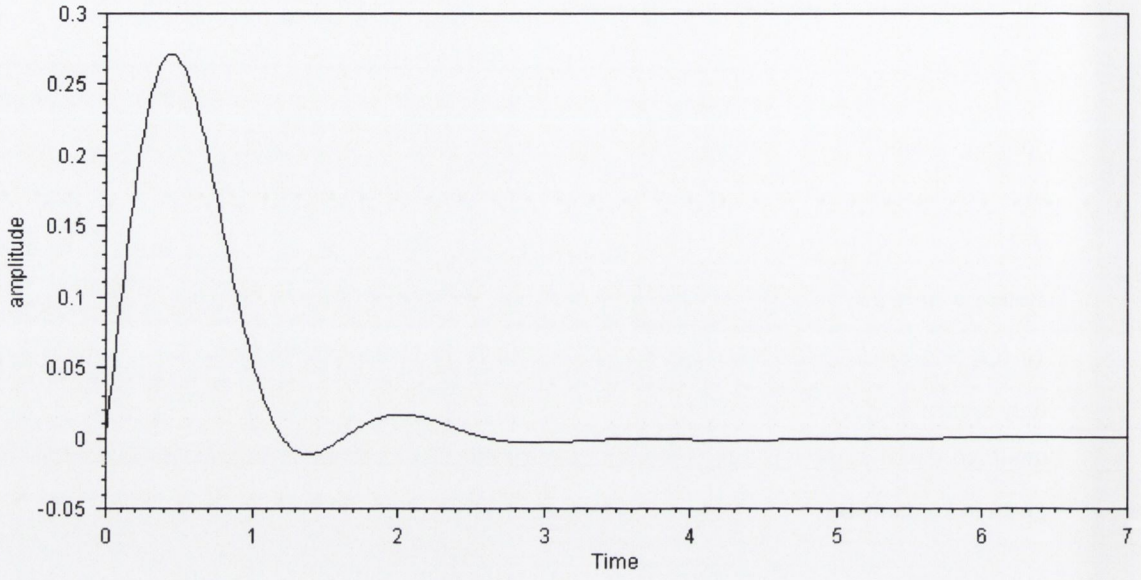


Figure 8.8 Impulse response for the system at the operating point $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$. This point was arrived at by using an optimization routine with starting point $(a,b,k_c,k_0) = (4.73,10.33,4.2,2.19)$ [8]. When $(a,b,k_c,k_0) = (5.0,9.5,4.0,2.0)$ the rightmost eigenvalues of the quiet standing polynomial is at –0.07.

8.4 Selecting all four parameters simultaneously

There are at least two approaches to the problem of selecting all four parameters simultaneously and not in pairs as in the previous sections. One approach is to derive a stability region in a four-dimensional parameter space (a,b,k_c,k_0) and use the centre of the largest hyper-sphere that could be drawn in this region used as a design point. A similar problem arises when one studies a polynomial whose coefficients may take any value from a given range. The resulting polynomials are represented as points in an n-dimensional polynomial-coefficient space. Those points that correspond to Hurwitz polynomials form a stability region. Ackermann [93] , Soh [94], and Bhattacharyya [95] describe how to draw the largest hyper-sphere in such a region. Bhattacharyya's approach was implemented without success in section 8.4.1.

Another approach is to use an optimization algorithm in combination with the Lyapunov matrix equation to select a design point that optimizes eigenvalue location. This method turned out to be quite successful and it is described in section 8.4.2.

8.4.1 Optimum vector margin design

Bhattacharyya et al. [95] describe a procedure for calculating the radius of the largest stability hypersphere that can be drawn around a point in parameter space. This procedure proved to be too unwieldy. Even though the parameter space has only four dimensions, the polynomial is of tenth order and, in this case, Bhattacharyya's procedure leads inexorably to calculations involving polynomials of order 52. This method was abandoned as such polynomials are very prone to rounding errors when evaluated.

8.4.2 Optimum stability using the Lyapunov matrix equation

The optimization algorithm described in 1.4 was applied to equation (8.2) to find sets of parameters that place the rightmost eigenvalue as far to the left as possible. In this case, the optimization algorithm is used to emulate evolution and adaptation in nature. Many applications of this algorithm resulted in systems with the rightmost eigenvalue at –0.11 and Table 8.4 shows a selection of the associated system parameters. In an attempt to make a selection from these parameters I have included the ITSE as an auxiliary criterion. Some of the results obtained are given in Table 8.4.

a	b	k_c	k_0	ITSE: $\int_0^\infty t\,e^2(t)dt$
3.57	13.68	0.59	2.39	0.984
3.89	10.04	2.58	7.94	0.208
4.29	10.82	2.74	7.15	0.080
4.73	10.04	1.58	4.17	0.127
5.42	13.21	2.23	5.56	0.031
6.17	10.11	1.89	8.27	0.137
6.49	13.30	3.08	9.47	0.20
6.91	12.11	4.12	12.31	0.019
7.33	8.12	3.94	11.65	0.081
8.39	13.0	2.51	6.99	0.014
9.39	11.22	1.77	5.09	0.024
9.98	13.77	2.74	7.62	0.017
11.43	9.78	3.20	12.65	131.49

Table 8.4 A selection of operating points that lead to Optimum stability with both the Visual / Somatosensory System and the Vestibular System active. These points were derived from the Lyapunov matrix equation and each set of parameters place the rightmost eigenvalue at –0.11. The model predicts that many people, each with different values for (a,b,k_c,k_0), will all have the same "sense of balance". This is indeed what one experiences in everyday life. The ITSE parameter is included to allow one to select a set of parameters.

operating point (a,b,k_c,k_0)	location of rightmost eigenvalue	ITSE	peak deflection of impulse response	settling time of impulse response (approx.)
(5.0,9.5,4.0,2.0)	-0.07	0.022	0.28	10s
(8.39,13.0,2.51,6.99)	-0.11	0.014	0.22	10s

Table 8.5 The operating point with the lowest ITSE from Table 8.4 - $(a,b,k_c,k_0)=(8.39,13.0,2.51,6.99)$ is compared with the operating point $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$ from section 8.3.3. The operating point form Table 8.4 has an eigenvalue further to the left, a lower ITSE figure and a lower peak impulse deflection than the operating point $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$. The operating point from Table 8.4 and $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$ have comparable settling times.

Table 8.5 returns for the moment to the operating point $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$ and compares its impulse response, as shown in Figure 8.8, with the impulse response of the operating point with lowest ITSE from Table 8.4.

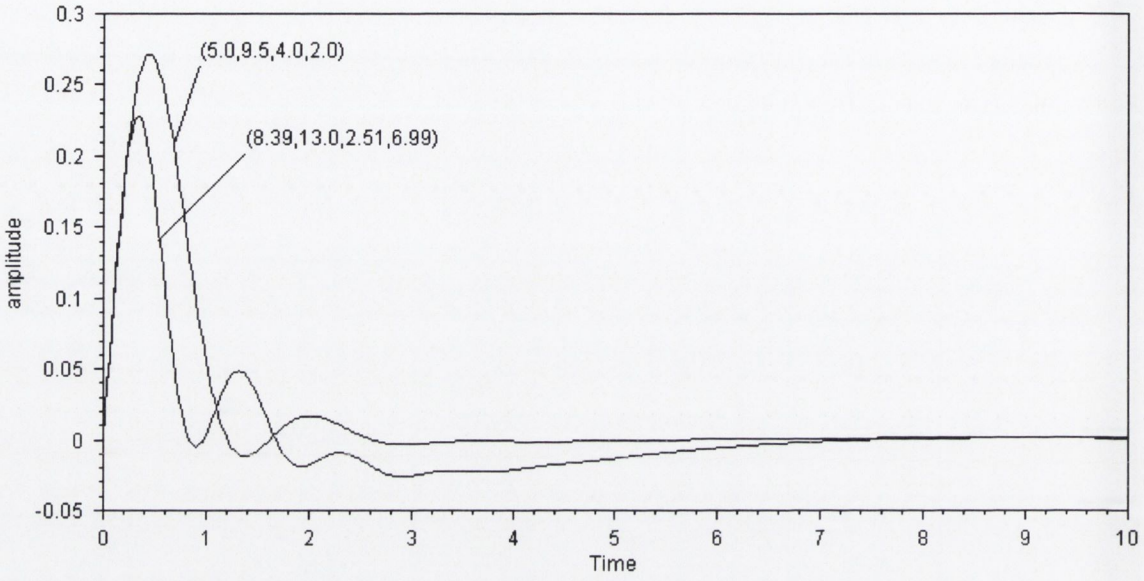


Figure 8.9 Impulse responses for the operating point with the lowest ITSE from Table 8.4 - $(a,b,k_c,k_0)=(8.39,13.0,2.51,6.99)$ and the operating point $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$ from section 8.3.3. The operating point form Table 8.4 has a lower peak impulse deflection than the operating point $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$. The operating point from Table 8.4 and $(a,b,k_c,k_0)=(5.0,9.5,4.0,2.0)$ have comparable settling times.

Time responses for three systems are shown in Figure 8.10, Figure 8.11, and Figure 8.12. These figures show impulse disturbance responses that illustrate the best, intermediate, and worst designs from Table 8.4.

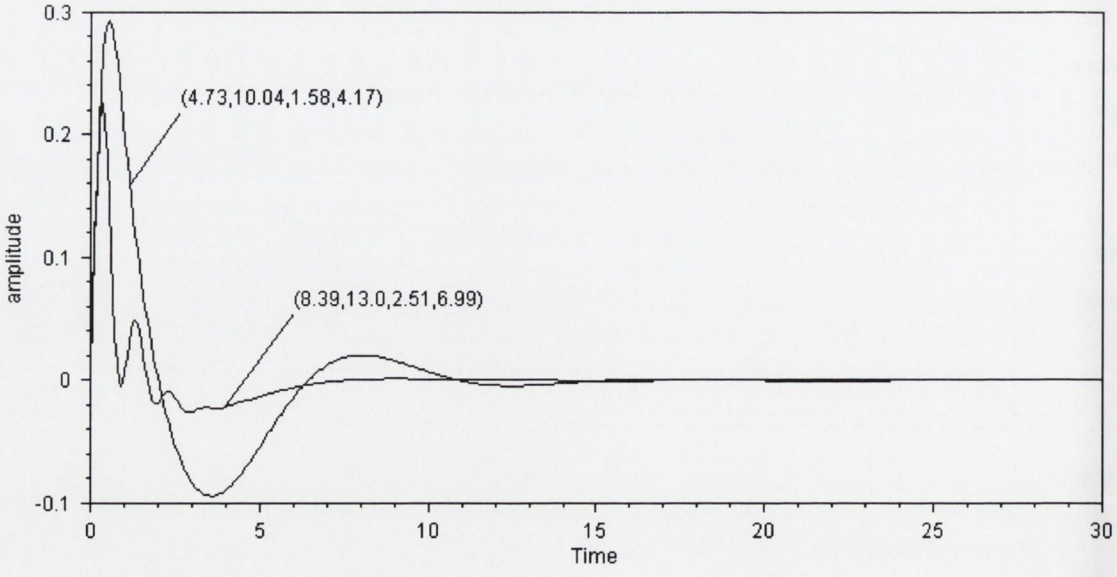


Figure 8.10 Impulse response for the system at two operating points from Table 8.4 - $(a,b,k_c,k_0)=(8.39,13.0,2.51,6.99)$ and $(a,b,k_c,k_0)=(4.73,10.04,1.58,4.17)$.

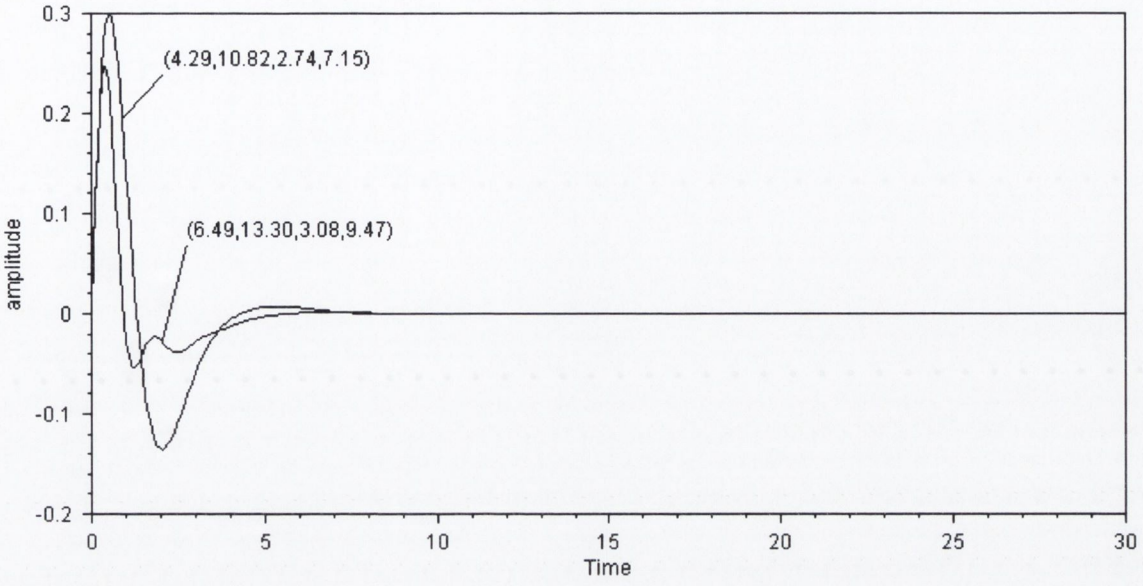


Figure 8.11 Impulse response for the system at two operating points from Table 8.4 - $(a,b,k_c,k_0) = (6.49,13.30,3.08,9.47)$ and $(a,b,k_c,k_0) = (4.29,10.82,2.74,7.15)$.

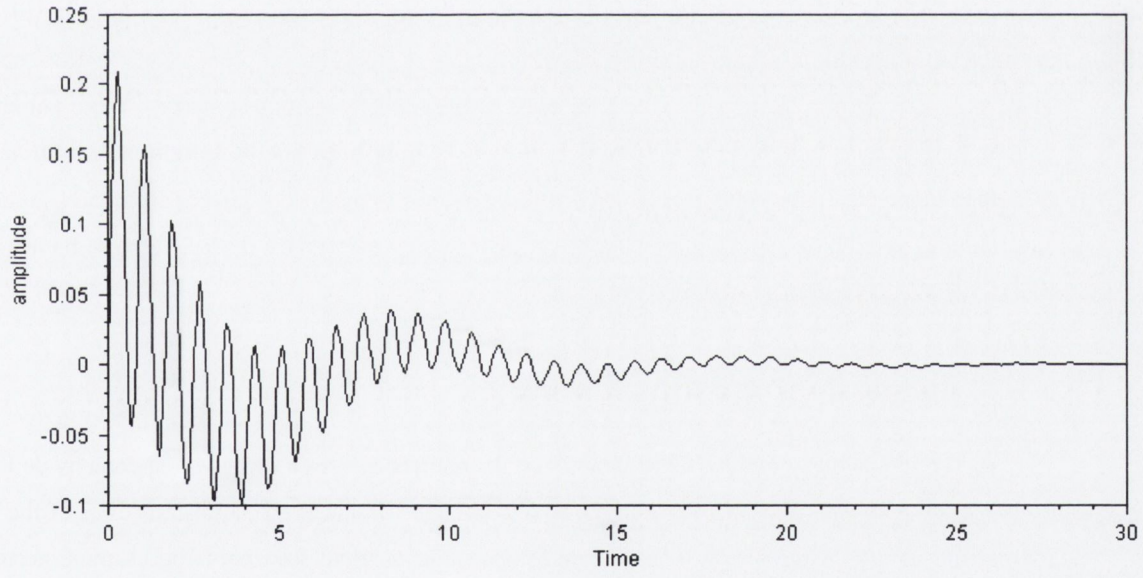


Figure 8.12 Impulse response for the system at the operating point $(a,b,k_c,k_0) = (11.43,9.78,3.3,12.65)$ from Table 8.4. The oscillatory behavior is due to a complex pair of eigenvalues ($-0.18 \pm j0.66$) near to the real eigenvalue at –0.11.

Two results are immediately apparent from Table 8.4. First, the rightmost eigenvalue has been shifted to the left of that obtained by dePaor [8] . dePaor's parameters [8] placed the rightmost eigenvalue at –0.07 whereas the parameters obtained in this study place it at –0.11. Thus, the magnitude of the shortest time constant has been increased by a factor of 1.6.

A second result apparent from Table 8.4 is that many combinations of (a,b,k_c,k_0) were found to have the optimum stability property. Finding multiple sets of parameters, each giving the same stability

margin, indicates in this case that the underling mathematical model is sound. The model predicts that many people, each with different values for (a,b,k_c,k_0), will all have the same "sense of balance". This is indeed what one experiences in everyday life. A mathematical model of the human balance control system that was optimally stable for a single set of parameter values and that predicted different values of stability margin for every person would not agree with our experience.

Initially it might seem counterintuitive that systems with the same index of stability exhibit completely different ITSEs and impulse response curves. However, it has to be born in mind that the time response depends not only on the location of the rightmost eigenvalue but also on a complicated interaction among the whole constellation of all the poles and all the zeros – this is especially evident in this system as it has many eigenvalues. Although the results obtained in previous chapters are very encouraging, the present analysis can serve as a note of warning that, in systems with many poles and zeros, simply placing the rightmost eigenvalue as far to the left as possible is not sufficient to guarantee a good time response. Any single quality indicator on its own is not a reliable predictor of performance.

8.5 Summary of results in this chapter

A model of the human balance control system has been studied. When the three time delays in the model are replaced by 1st order Pade approximations the model's characteristic polynomial is order ten. It is necessary to select four parameters – one for each major control loop. Nyquist analysis was used to select pairs of parameters that lead to Optimum Phase Margin but this is a graphical procedure and therefore approximate. The Lyapunov matrix equation was used to select four parameters simultaneously – these parameters gave optimum eigenvalue location. The parameters selected using Nyquist analysis tended to lead to systems that are underdamped and have very oscillatory impulse responses. Parameters derived from the Lyapunov matrix equation and optimum eigenvalue location, gave impulse responses that were fare less oscillatory and settled down after about 6 seconds to 10 seconds.

8.6 Suggestions for future work

One could carry out the same analysis on the models derived above, as was done by de Paor [8] i.e. excite the system with Gaussian Random noise and compute the autocorrelation functions of the outputs. A comparison of these results with data gathered from subjects could then be made. A more accurate match would indicate better parameters. The parameter tuning and the inclusion of "mild non-linearities" referred to by dePaor[8] could also be done.

It would also be very interesting to adapt the methods of [93]and [94] and use it to choose a true optimum vector margin operating point from the four dimensional (a,b,k_c,k_0) stability region. The distance from the rightmost eigenvalue to the real axis dictates the topography of the interior of the stability region and this must be taken into account when choosing an operating point within the region. This would result in a five dimensional problem in (a,b,k_c,k_0,σ) -space.

Chapter 9: Optimum stability from the Routh array and the Jeltsch-Fichera array

9.1 Introduction

In this chapter I explore the use of the Routh and Jeltsch-Fichera arrays for designing systems with optimum stability. (The Routh array can be used in synthetic mode as discussed in Appendix E.)

An advantage of root locus diagrams is that they indicate the degree of stability of a point and not only whether it is stable or not. The Nyquist diagram also indicates the degree of stability in the form of Gain and Phase Margins. On the other hand, the Lyapunov matrix equation and the Routh array, in the standard form, give only a criterion for stability and no indication of the degree of stability. It is possible to introduce degree-of-stability information into the standard Routh Array and into the Lyapunov equation by making very simple changes to their form. For example, as was seen in section 8.4.2, solving the Lyapunov matrix equation $(\mathbf{A}+\sigma)^T\mathbf{L}+\mathbf{L}(\mathbf{A}+\sigma)=-\mathbf{Q}$ rather than $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$. In this chapter I explore the properties of the Routh array and the Jeltsch-Fichera array of $p(s+\sigma)=0$ rather than $p(s)=0$. As in the case of the modified Lyapunov equation, σ is a measure of the degree of stability of the system.

9.2 Some properties of the Routh and Jeltsch-Fichera arrays

9.2.1 Constructing the Routh and Jeltsch-Fichera arrays

In 1979 Jeltsch [96] published a version of the Routh array that avoids division and is optimal in the sense that the rate of growth of the entries in the array is minimised. Such arrays are usually called Jeltsch arrays. However, in 1995 Jeltsch [97] found that Fichera [98] had already discovered this array in 1947. For this reason the name Jeltsch-Fichera array will be used to describe these arrays in this thesis.

The absence of division that characterises the entries in the Jeltsch-Fichera array turns out to be convenient in this thesis as will be seen below.

For $f(s)=a_0s^n+b_0s^{n-1}+a_1s^{n-2}+b_1s^{n-3}+....$ the Routh array may be constructed as follows:

$$r_{ij}=-\frac{1}{r_{i-1,1}}\begin{vmatrix} r_{i-2,1} & r_{i-2,j+1} \\ r_{i-1,1} & r_{i-1,j+1} \end{vmatrix},\ j=1,2,3...$$ with the initial conditions $r_{0j}=a_{j-1}$, $r_{1j}=b_{j-1}, j=1,2,3...$

For $f(s)=a_0s^n+b_0s^{n-1}+a_1s^{n-2}+b_1s^{n-3}+....$ the optimal fraction free Routh array (i.e. the Jeltsch-Fichera array) may be constructed as follows:

$$n_{ij}=-\frac{1}{d_i}\begin{vmatrix} n_{i-2,1} & n_{i-2,j+1} \\ n_{i-1,1} & n_{i-1,j+1} \end{vmatrix},\ j=1,2,3...$$ where $d_i=\begin{cases} 1 & \text{for } i=2,3,... \\ n_{i-3,1} & \text{for } i=4,5,... \end{cases}$

with the initial conditions $n_{0j}=r_{0j}=a_{j-1}$, $n_{1j}=r_{1j}=b_{j-1}, j=1,2,3...$

9.2.2 Stability from the Jeltsch-Fichera array

A point worth remembering about the Jeltsch-Fichera array is that the number of sign changes in the left hand column does not indicate the number of eigenvalues in the right half plane [19].

For example, the Routh array for $s^4+s^3-12s^2-28s-16=(s+2)^2(s+1)(s-4)$ is:

1	-12	-16
1	-28	
16	-16	
-27		
-16		

So k, the number of sign changes in the left hand column, is $k=V(1,1,16,-27,-16)=1$ - where V is a function, defined on a list of numbers, which returns the number of changes of sign in the list. So, there is 1 eigenvalue in the right half plane. (In fact, the appearance of a negative sign anywhere in the Routh array indicates instability [19], and the number of sign changes in the first column gives the number of eigenvalues in the right half plane.)

The Jeltsch-Fichera array for $s^4+s^3-12s^2-28s-16$ is:

1	-12	-16
1	-28	
16	-16	
-432		
6912		

In this case, to find the number of eigenvalues in the right half plane you have to calculate:

$k=V\left(1,1,\frac{16}{1},\frac{-432}{16},\frac{6912}{-432}\right)=1$ So, we see again that there is 1 eigenvalue in the right half plane. Note that $\left(1,1,\frac{16}{1},\frac{-432}{16},\frac{6912}{-432}\right)$ is in fact the first column of the Routh array.

To find the number of eigenvalues in the RHP from the Jeltsch-Fichera array, you have to divide a member by the previous one and look at the number variations in signs of this sequence.

In general, Barnett [19] gives:

$$k=V\left(n_{01},n_{11},\frac{n_{21}}{n_{11}},\frac{n_{31}}{n_{21}}....\frac{n_{n1}}{n_{n-1,1}}\right) \tag{9.1}$$

for the Jeltsch-Fichera array. $\left(n_{01},n_{11},\frac{n_{21}}{n_{11}},\frac{n_{31}}{n_{21}}....\frac{n_{n1}}{n_{n-1,1}}\right)$ is in fact the first column of the Routh array.

As a more general example, consider $p(s)=(s+c)(s^2+es+d^2)^2$. The first column of the Jeltsch-Fichera array of this polynomial is:

$$1$$

$$c+2e$$

$$2e(c^2+d^2+2ce+e^2)$$

$$2e(2d^2e^3+c^3(d^2+e^2)+2c^2e(d^2+e^2)+c(d^4+3d^2e^2+e^4))$$

$$4d^2e^4(c^2+d^2+ce)^2$$

$$4cd^6e^4(c^2+d^2+ce)^2$$

Using equation (9.1) we get:

$$k=V\left(\begin{array}{l} 1, c+2e, \dfrac{2e(c^2+d^2+2ce+e^2)}{c+2e}, \\ \dfrac{2e(2d^2e^3+c^3(d^2+e^2)+2c^2e(d^2+e^2)+c(d^4+3d^2e^2+e^4))}{2e(c^2+d^2+2ce+e^2)}, \\ \dfrac{4d^2e^4(c^2+d^2+ce)^2}{2e(2d^2e^3+c^3(d^2+e^2)+2c^2e(d^2+e^2)+c(d^4+3d^2e^2+e^4))}, \\ \dfrac{4cd^6e^4(c^2+d^2+ce)^2}{4d^2e^4(c^2+d^2+ce)^2} \end{array}\right)$$

but this is equal to the first column of the Routh Array:

$$1$$

$$c+2e$$

$$\frac{2e(c^2+d^2+2ce+e^2)}{c+2e}$$

$$\frac{2d^2e^3+c^3(d^2+e^2)+2c^2e(d^2+e^2)+c(d^4+3d^2e^2+e^4)}{(c^2+d^2+2ce+e^2)}$$

$$\frac{2d^2e^3(c^2+d^2+ce)^2}{2d^2e^3+c^3(d^2+e^2)+2c^2e(d^2+e^2)+c(d^4+3d^2e^2+e^4)}$$

$$cd^4$$

So one can go back and forward between the Routh and Jeltsch-Fichera arrays very easily.

9.3 Optimum stability, the principal minors of a Hurwitz matrix, and the Jeltsch-Fichera array

Barnett [19] shows that the first column of the Jeltsch-Fichera array is composed of the leading principal minors of the Hurwitz matrix. For example, the Jeltsch-Fichera array for $p(s)=s^4+s^3-12s^2-28s-16=0$ is:

1	-12	-16
1	-28	
16	-16	
-432		
6912		

Also, the Hurwitz matrix for $p(s)$ is $\mathbf{H} = \begin{pmatrix} 1 & -28 & 0 & 0 \\ 1 & -12 & -16 & 0 \\ 0 & 1 & -28 & 0 \\ 0 & 1 & -12 & -16 \end{pmatrix}$. The principal minors of $\mathbf{H}$ are: $\mathbf{H}_1 = 1$,

$\mathbf{H}_2 = \begin{vmatrix} 1 & -28 \\ 1 & -12 \end{vmatrix} = 16$, $\mathbf{H}_3 = \begin{vmatrix} 1 & -28 & 0 \\ 1 & -12 & -16 \\ 0 & 1 & -28 \end{vmatrix} = -432$ and $\mathbf{H}_4 = |\mathbf{H}| = 6912$.

Say $\mathbf{H}$ is an $m \times m$ Hurwitz matrix for a characteristic polynomial and $\mathbf{H} = \mathbf{H}_m$, $\mathbf{H}_{m-1}$, $\mathbf{H}_{m-2}$, $\mathbf{H}_{m-3}$ ... $\mathbf{H}_1$ are the m principal minors of $\mathbf{H}$. Say also that there is a p-fold optimum stability point for the system polynomial. Then we now know from [19] and from section 9.5 that at this optimum stability point the following p principal minors of $\mathbf{H}$ will be equal to zero: $\mathbf{H}_m = 0$, $\mathbf{H}_{m-1} = 0$, $\mathbf{H}_{m-2} = 0$, $\mathbf{H}_{m-3} = 0$, ... $\mathbf{H}_{m-p} = 0$.

This property of Hurwitz matrices provides another method for determining the system parameters that ensure optimum stability. It was not pursued in this thesis, as it is equivalent to the Jeltsch-Fichera array approach.

9.4 Optimum stability, the Jeltsch-Fichera array, and the Routh array

The Routh and Jeltsch-Fichera arrays both display an interesting property when the system described by the characteristic equation is optimally stable in the root locus sense. This property may best be illustrated by examples.

Example 1: The Routh and Jeltsch-Fichera arrays for $p_1(s) = (s+a)(s+b)^3$

The first column of the Routh array for the polynomial $p_1(s) = (s+a)(s+b)^3$ is:

$$1$$
$$a+3b$$
$$\frac{b(3a^2+9ab+8b^2)}{a+3b}$$
$$\frac{8b^2(a+b)^3}{3a^2+9ab+8b^2}$$
$$ab^3$$

Assume that both a and b are in the left half plane and that a is further to the left than b.

The polynomial $p_1(s)$ has a triple root at $s = -b$. This is the optimum stability point in terms of root locus. If we shift the $j\omega$-axis to the left as far as the point $s = -b$ we are effectively considering the polynomial $p_1(s+\sigma) = p_1(s-b)$. At this point $b = 0$ and the first column of the Routh array becomes:

1

a

0

0

0

We see that the three entries in the "tail" of the first column of the Routh array have gone to zero.

The first column of the Jeltsch-Fichera array for the polynomial $p_1(s) = (s+a)(s+b)^3$ is:

1

$a+3b$

$b(3a^2+9ab+8b^2)$

$8b^3(a+b)^3$

$8ab^6(a+b)^3$

At the point $b=0$ the first column of the Jeltsch-Fichera array becomes:

1

a

0

0

0

We see that the three entries in the "tail" of the first column of the Jeltsch-Fichera array have gone to zero also.

Example 2: The Routh and Jeltsch-Fichera arrays for $p_2(s) = (s^2+as+b)(s+c)^3$

The first column of the Routh array for the polynomial $p_2(s) = (s^2+as+b)(s+c)^3$ is:

1

$a+3c$

$$\frac{ab+3a^2c+9ac^2+8c^3}{a+3c}$$

$$\frac{c(8a^3c^2+8c^5+3a^2(3bc+8c^3)+a(3b^2+11bc^2+24c^4))}{ab+3a^2c+9ac^2+8c^3}$$

$$\frac{8ac^2(b+c(a+c))^3}{8a^3c^2+8c^5+3a^2(3bc+8c^3)+a(3b^2+11bc^2+24c^4)}$$

bc^3

Assume that all the roots of $p_2(s)$ are in the left half plane and that the roots of (s^2+as+b) are further to the left than c. The polynomial $p_2(s)$ has a triple root at $s=-c$. This is the optimum stability point in terms of root locus. If we shift the $j\omega-$axis to the left as far as the point $s=-c$ we are effectively considering the polynomial $p_2(s+\sigma) = p_2(s-c)$. At this point $c=0$ and the first column of the Routh array becomes:

1

a

b

0

0

0

We see that the three entries in the "tail" of the first column of the Routh array have gone to zero.

The first column of the Jeltsch-Fichera array for the polynomial $p_2(s) = (s^2 + as + b)(s + c)^3$ is:

1

$a + 3c$

$ab + 3a^2c + 9ac^2 + 8c^3$

$c(8a^3c^2 + 8c^5 + 3a^2(3bc + 8c^3) + a(3b^2 + 11bc^2 + 24c^4))$

$8ac^3(b + c(a + c))^3$

$8abc^6(b + c(a + c))^3$

At the point $c = 0$ and the Jeltsch-Fichera array becomes:

1

a

ab

0

0

0

We see that the three entries in the "tail" of the first column of the Jeltsch-Fichera array have gone to zero also.

Example 3: The Routh and Jeltsch-Fichera arrays for $p_3(s) = (s^2 + as + b)(s + c)^4$

The first column of the Routh array for the polynomial $(s^2 + as + b)(s + c)^4$ is:

1

$a + 4c$

$$\frac{ab + 4a^2c + 16ac^2 + 20c^3}{a + 4c}$$

$$\frac{4c(5a^3c^2 + 16c^5 + 4a^2c(b + 5c^2) + a(b^2 + 6bc^2 + 29c^4))}{ab + 4a^2c + 16ac^2 + 20c^3}$$

$$\frac{c^2(16a^4c^3 + 16c^7 + a^3(29bc^2 + 64c^4) + 4a^2(5b^2c + 17bc^3 + 24c^5) + a(5b^3 + 22b^2c^2 + 41bc^4 + 64c^6))}{5a^3c^2 + 16c^5 + 4a^2c(b + 5c^2) + a(b^2 + 6bc^2 + 29c^4)}$$

$$\frac{16ac^3(b + c(a + c))^4}{16a^4c^3 + 16c^7 + a^3(29bc^2 + 64c^4) + 4a^2(5b^2c + 17bc^3 + 24c^5) + a(5b^3 + 22b^2c^2 + 41bc^4 + 64c^6)}$$

bc^4

At this point $c = 0$ and the first column of the Routh array becomes:

1

a

b

0

0

0

0

We see that the four entries in the "tail" of the Routh array have gone to zero.

The first column of the Jeltsch-Fichera array for the polynomial $(s^2 + as + b)(s + c)^4$ is:

1

$a + 4c$

$ab + 4a^2c + 16ac^2 + 20c^3$

$4c(5a^3c^2 + 16c^5 + 4a^2c(b + 5c^2) + a(b^2 + 6bc^2 + 29c^4))$

$4c^3(16a^4c^3 + 16c^7 + a^3(29bc^2 + 64c^4) + 4a^2(5b^2c + 17bc^3 + 24c^5) + a(5b^3 + 22b^2c^2 + 41bc^4 + 64c^6))$

$64ac^6(b + c(a + c))^4$

$64abc^{10}(b + c(a + c))^4$

At the point $c = 0$ and the first column of the Jeltsch-Fichera array becomes:

1

a

ab

0

0

0

0

We see that the four entries in the "tail" of the first column of the Jeltsch-Fichera array have gone to zero also.

A trend is becoming apparent in these arrays. If an eigenvalue has multiplicity n then the last n terms in the first column of both the Routh and Jeltsch-Fichera arrays become equal to zero when one shifts the $j\omega$ - axis onto the multiple eigenvalue. In the optimum stability studies so far in this thesis we have chosen to place the operating point at a multiple eigenvalue. A different method for choosing the design parameters is now apparent i.e. choose those parameters that make the last n terms in the first column of the Routh or Jeltsch-Fichera arrays of $p(s + \sigma) = 0$ equal to zero.

We will see in the next section that it is not only the last n terms in the first column but also the last n terms in every column of the Routh and Jeltsch-Fichera arrays go to zero.

9.5 The last n entries of all of the columns the Routh and Jeltsch-Fichera Arrays vanish at an optimum stability point

In this section r_{ij} will be used to indicate the i^{th} entry in the j^{th} column of the Routh array and n_{ij} will be used to indicate the i^{th} entry in the j^{th} column of the Jeltsch-Fichera array.

We have found that the Routh Array for a system that is optimally stable in the root locus sense has a peculiar feature. Specifically, if the characteristic equation is of the form $p(s) = (s+\sigma)^n$ then the last n entries in each column of the array is zero. This results in a family of non-linear equations in the system parameters and σ that may be solved for those values that lead to optimum stability. The reason behind this peculiarity is explored below.

The Routh Array is usually written out for polynomial $p(s)$ with constant coefficients. In order to introduce degree of stability information we can write out the Routh Array for $p(s+\sigma) = 0$. The value of σ that makes a left column entry in the array equal to zero is a measure of the degree of stability of the polynomial.

We know from elementary calculus that for any polynomial

$p(s) = a_n(x-a)^n + a_{n-1}(x-a)^{n-1} + ... + a_1(x-a) + a_0(x-a)$ we can write: $a_k = \dfrac{p^{(k)}(a)}{k!}$ where $p^{(k)}(a)$ is the k^{th} derivative of $p(s)$ evaluated at $s = a$.

For example, if

$$p(s) = (s-\sigma)^4 + a(s-\sigma)^3 + b(s-\sigma)^2 + c(s-\sigma) + d$$
$$= s^4 + (a-4\sigma)s^3 + (6\sigma^2 - 3a\sigma + 6)s^2 + (-4\sigma^3 + 3a\sigma^2 - 2b\sigma + c)s + (\sigma^4 - a\sigma^3 + b\sigma^2 - c\sigma + d)$$

we notice that the coefficients are:

$$\frac{p^{(0)}(-\sigma)}{0!} = \sigma^4 - a\sigma^3 + b\sigma^2 - c\sigma + d$$
$$\frac{p^{(1)}(-\sigma)}{1!} = -4\sigma^3 + 3a\sigma^2 - 2b\sigma + c$$
$$\frac{p^{(2)}(-\sigma)}{2!} = 6\sigma^2 - 3a\sigma + 6$$
$$\frac{p^{(3)}(-\sigma)}{3!} = a - 4\sigma$$
$$\frac{p^{(4)}(-\sigma)}{4!} = 1$$

So we can write the equation $p(s) = (s-\sigma)^4 + a(s-\sigma)^3 + b(s-\sigma)^2 + c(s-\sigma) + d$ as:

$$p(s) = \frac{p^{(4)}(-\sigma)}{4!}s^4 + \frac{p^{(3)}(-\sigma)}{3!}s^3 + \frac{p^{(2)}(-\sigma)}{2!}s^2 + \frac{p^{(1)}(-\sigma)}{1!}s + \frac{p^{(0)}(-\sigma)}{0!}$$

The first two rows of the Routh array for $p(s)$ become:

$\frac{p^{(4)}(-\sigma)}{4!}$	$\frac{p^{(2)}(-\sigma)}{2!}$	$\frac{p^{(0)}(-\sigma)}{0!}$
$\frac{p^{(3)}(-\sigma)}{3!}$	$\frac{p^{(1)}(-\sigma)}{1!}$	0
$\vdots$	$\vdots$	$\vdots$

The Routh algorithm for creating entries $r_{ij} = \dfrac{r_{i-1,1}\, r_{i-2,j+1} - r_{1-2,1}\, r_{i-1,j+1}}{r_{i-1,1}}$ may be pictured as:

.	.	.	.	.	.
$r_{i-2,1}$	.	.	.	$r_{i-2,j+1}$	.
$r_{i-1,1}$	.	.	.	$r_{i-1,j+1}$	.
.	.	.	$r_{ij}\nearrow$	.	.
.	.	.	.	.	.

We can see immediately that if, for example, $\frac{p^{(1)}(-\sigma)}{1!} = 0$ and $\frac{p^{(0)}(-\sigma)}{0!} = 0$, then these two zeros propagate along the bottom of the array causing the last two entries in each column go to zero.

We know from section 1.2 that if, for example, there is a triple breakpoint in the root locus at $s = -a$ the characteristic equation becomes, for example:

$$p(s) = (s+a)^3 = 0 \qquad \text{at } s = -a \tag{9.2}$$

Differentiating equation (9.2) gives: $p'(s) = 3(s+a)^2 = 0 \qquad \text{at } s = -a$ (9.3)

Differentiating equation (9.3) gives: $p''(s) = 6(s+a) = 0 \qquad \text{at } s = -a$

In general, if the order of the characteristic equation n is even, and there is a threefold breakpoint in the root locus then the top right hand corner of the Routh array looks like:

....	$r_{1,\frac{n}{2}-7}$	$r_{1,\frac{n}{2}-6}$	$r_{1,\frac{n}{2}-5}$	$r_{1,\frac{n}{2}-4}$	$r_{1,\frac{n}{2}-3}$	$r_{1,\frac{n}{2}-2}$	$r_{1,\frac{n}{2}-1}=0$	$r_{1,\frac{n}{2}+1}=0$
....	$r_{2,\frac{n}{2}-7}$	$r_{2,\frac{n}{2}-6}$	$r_{2,\frac{n}{2}-5}$	$r_{2,\frac{n}{2}-4}$	$r_{2,\frac{n}{2}-3}$	$r_{2,\frac{n}{2}-2}$	$r_{2,\frac{n}{2}-1}=0$	0
....	$r_{3,\frac{n}{2}-7}$	$r_{3,\frac{n}{2}-6}$	$r_{3,\frac{n}{2}-5}$	$r_{3,\frac{n}{2}-4}$	$r_{3,\frac{n}{2}-3}$	$r_{3,\frac{n}{2}-2}=0$	$r_{3,\frac{n}{2}-1}=0$	0
....	$r_{4,\frac{n}{2}-7}$	$r_{4,\frac{n}{2}-6}$	$r_{4,\frac{n}{2}-5}$	$r_{4,\frac{n}{2}-4}$	$r_{4,\frac{n}{2}-3}$	$r_{4,\frac{n}{2}-2}=0$	0	0
....	$r_{5,\frac{n}{2}-7}$	$r_{5,\frac{n}{2}-6}$	$r_{5,\frac{n}{2}-5}$	$r_{5,\frac{n}{2}-4}$	$r_{5,\frac{n}{2}-3}=0$	$r_{5,\frac{n}{2}-2}=0$	0	0
....	$r_{6,\frac{n}{2}-7}$	$r_{6,\frac{n}{2}-6}$	$r_{6,\frac{n}{2}-5}$	$r_{6,\frac{n}{2}-4}$	$r_{6,\frac{n}{2}-3}=0$	0	0	0
....	$r_{7,\frac{n}{2}-7}$	$r_{7,\frac{n}{2}-6}$	$r_{7,\frac{n}{2}-5}$	$r_{7,\frac{n}{2}-4}=0$	$r_{7,\frac{n}{2}-3}=0$	0	0	0
....	$r_{8,\frac{n}{2}-7}$	$r_{8,\frac{n}{2}-6}$	$r_{8,\frac{n}{2}-5}$	$r_{8,\frac{n}{2}-4}=0$	0	0	0	0
....	$r_{9,\frac{n}{2}-7}$	$r_{9,\frac{n}{2}-6}$	$r_{9,\frac{n}{2}-5}=0$	$r_{9,\frac{n}{2}-4}=0$	0	0	0	0
....	⋮	⋮	⋮	0	0	0	0	0

Figure 9.1 Routh array for a polynomial $p(s+\sigma)=0$ of order n (even) with a triple eigenvalue at $s=-\sigma$. Those entries in the upper right corner that are zero propagate down the array causing the last three entries in each column to go to zero.

We can see from Figure 9.1, and from the method for constructing the Routh array, that for a polynomial of even order the last three entries in each column become zero at a point of optimum stability.

If n is odd, and there is a threefold breakpoint in the root locus then the top right hand corner of the Routh array looks like:

...	$r_{1,\frac{n-13}{2}}$	$r_{1,\frac{n-11}{2}}$	$r_{1,\frac{n-9}{2}}$	$r_{1,\frac{n-7}{2}}$	$r_{1,\frac{n-5}{2}}$	$r_{1,\frac{n-3}{2}}$	$r_{1,\frac{n-1}{2}}$	$r_{1,\frac{n+1}{2}}=0$
...	$r_{2,\frac{n-13}{2}}$	$r_{2,\frac{n-11}{2}}$	$r_{2,\frac{n-9}{2}}$	$r_{2,\frac{n-7}{2}}$	$r_{2,\frac{n-5}{2}}$	$r_{2,\frac{n-3}{2}}$	$r_{2,\frac{n-1}{2}}=0$	$r_{2,\frac{n+1}{2}}=0$
...	$r_{3,\frac{n-13}{2}}$	$r_{3,\frac{n-11}{2}}$	$r_{3,\frac{n-9}{2}}$	$r_{3,\frac{n-7}{2}}$	$r_{3,\frac{n-5}{2}}$	$r_{3,\frac{n-3}{2}}$	$r_{3,\frac{n-1}{2}}=0$	0
...	$r_{4,\frac{n-13}{2}}$	$r_{4,\frac{n-11}{2}}$	$r_{4,\frac{n-9}{2}}$	$r_{4,\frac{n-7}{2}}$	$r_{4,\frac{n-5}{2}}$	$r_{4,\frac{n-3}{2}}=0$	$r_{4,\frac{n-1}{2}}=0$	0
...	$r_{5,\frac{n-13}{2}}$	$r_{5,\frac{n-11}{2}}$	$r_{5,\frac{n-9}{2}}$	$r_{5,\frac{n-7}{2}}$	$r_{5,\frac{n-5}{2}}$	$r_{5,\frac{n-3}{2}}=0$	0	0
...	$r_{6,\frac{n-13}{2}}$	$r_{6,\frac{n-11}{2}}$	$r_{6,\frac{n-9}{2}}$	$r_{6,\frac{n-7}{2}}$	$r_{6,\frac{n-5}{2}}=0$	$r_{6,\frac{n-3}{2}}=0$	0	0
...	$r_{7,\frac{n-13}{2}}$	$r_{7,\frac{n-11}{2}}$	$r_{7,\frac{n-9}{2}}$	$r_{7,\frac{n-7}{2}}$	$r_{7,\frac{n-5}{2}}=0$	0	0	0
...	$r_{8,\frac{n-13}{2}}$	$r_{8,\frac{n-11}{2}}$	$r_{8,\frac{n-9}{2}}$	$r_{8,\frac{n-7}{2}}=0$	$r_{8,\frac{n-5}{2}}=0$	0	0	0
...	$r_{9,\frac{n-13}{2}}$	$r_{9,\frac{n-11}{2}}$	$r_{9,\frac{n-9}{2}}=0$	$r_{9,\frac{n-7}{2}}=0$	0	0	0	0
...	$r_{10,\frac{n-13}{2}}$	$r_{10,\frac{n-11}{2}}$	$r_{10,\frac{n-9}{2}}=0$	$r_{10,\frac{n-7}{2}}=0$	0	0	0	0
...	⋮	⋮	⋮	0	0	0	0	0

Figure 9.2 Routh array for a polynomial $p(s+\sigma)=0$ of order n (odd) with a triple eigenvalue at $s=-\sigma$. Those entries in the upper right corner that are zero propagate down the array causing the last three entries in each column to go to zero.

Again we can see from Figure 9.2, and from the method for constructing the Routh array, that for a polynomial of odd order the last three entries in each column becomes zero at a point of optimum stability. In general, if there is an p-fold breakpoint in the root locus of a characteristic polynomial then the last p entries in the Routh array of that polynomial go to zero.

There is a close relationship between the Routh Array and the Jeltsch-Fichera array. In particular, if r_{i1} is the i^{th} entry of the first column of the Routh array then the i^{th} entry in the first column of the Jeltsch-Fichera array is $n_{i1} = r_{i1}n_{i-1}$. So if $r_{i1} = 0$ then $n_{i1} = 0$. Generally if $r_{ij} = 0$ then $n_{ij} = 0$ so if the last p entries of a tail of the Routh array vanish then the last p entries of the corresponding tail of the Jeltsch-Fichera array will vanish too.

In the following sections it will be seen that the entries of the Routh array contain quotients that can become indeterminate at the optimum stability point. This problem is not encountered with the Jeltsch-Fichera array as it does not contain quotients.

9.6 PID controller for an unstable, non-minimum phase process

Section 3.5 uses root locus based optimum stability to design a PID controller $C(s) = \dfrac{k(s-\alpha)(s+\beta)}{s(s-\gamma)}$ for the unstable, non-minimum phase process $G(s) = \dfrac{s-1}{s^2-s-2}$. We can now use the Jeltsch-Fichera array to design $C(s)$. The characteristic polynomial of the system is given by equation (9.4):

$$p(s) = s(s-\gamma)(s+1)(s-2) + k(s-\alpha)(s+\beta)(s-1) \tag{9.4}$$

As in section 3.5 we have $p(s) = (s+2)^4$ so our optimum stability point is a four-fold eigenvalue at $s = -2$. Forming the Jeltsch-Fichera array for $p(s+\sigma) = p(s-2) = 0$ we now know that the last four entries in each column are equal to zero. Each term in a non-linear express in α, β, γ and k but they can be solved as follows.

$n_{21} = 0$ gives:

$$k = \gamma + 9 \tag{9.5}$$

Substituting (9.5) into $9n_{12} + 3n_{22} + n_{13} = 0$ gives:

$$\gamma = 41.5 \tag{9.6}$$

We now have $k = 50.5$. Substituting $k = 50.5, \gamma = 41.5$ into $n_{12} = 0$ gives:

$$\beta = \frac{35}{50.5} + \alpha \tag{9.7}$$

Substituting (9.7) into $n_{22} = 0$ gives;

$$50.5\alpha^2 + 35\alpha - 16 = 0 \tag{9.8}$$

Solving (9.8) gives $\alpha = 0.314463$ and $\alpha = -1.00753$.

We select the positive value for α to get:

$$\alpha = 0.314463 \tag{9.9}$$

Substituting (9.9) into (9.7) gives:

$$\beta = 1.00753 \tag{9.10}$$

The values for α, β, γ and k obtained here are the same values obtained in section 3.5 using root locus.

9.7 Stability domain and optimum stability point for the n-let polynomial

Typically, some of the coefficients of a characteristic equation are made up from system parameters. For example, the characteristic equation $p(s) = s(s^2+1) + k(s^2+f)$ is encountered in the n-let effect [99]: the magnification of torque generated by a paralysed muscle in response to electrical stimulation by segmenting the stimulating pulses optimally in time. Stability domains are regions in parameter space that are made up of points that correspond to parameter values that make the characteristic equation Hurwitz. One way of finding stability domains is to use the Routh array for $p(s) = s(s^2+1) + k(s^2+f)$ as follows:

$$\begin{array}{ll} 1 & 1 \\ k & kf \\ 1-f & \\ kf & \end{array}$$

This system is asymptotically stable (and $p(s)$ is Hurwitz) if and only if all the entries in the left hand column are positive. This means that $k > 0$, $f < 1$ and $f > 0$.

Picking various value of f and experimentation with root locus analysis with k as parameter shows that $p(s)$ has a triple root at the optimum stability point. At this point $p(s) = 0$, $\frac{dp(s)}{ds} = 0$, and $\frac{d^2 p(s)}{ds^2} = 0$ simultaneously. Solving these equations gives $(k, f) = (\sqrt{3}, \frac{1}{9})$ and $s = -1/\sqrt{3}$.

Plotting the Routh array information and the Root locus information in the (k, f) parameter plane gives:

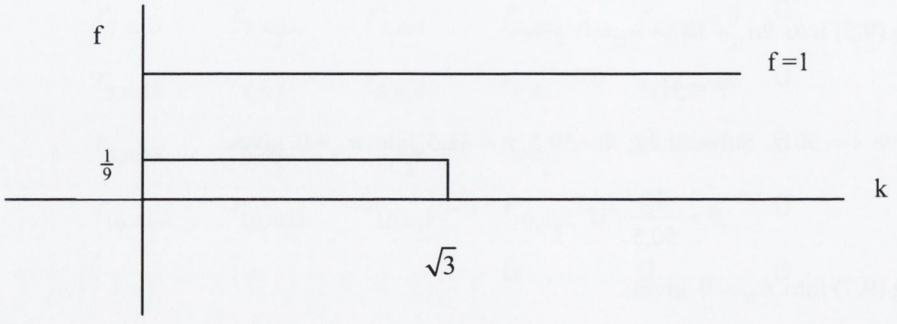


Figure 9.3 Stability region in the (k, f) for the polynomial $p(s) = s(s^2+1) + k(s^2+f)$. The optimum stability point $(\sqrt{3}, 1/9)$ derived from root locus is shown.

Note that in this case there is no centroid to choose as a design point so the method adopted by Datta [29] and others cannot be used here.

The Routh array indicates that the stability domain is the band of points bounded by $k = 0$, $f = 1$, and $f = 0$. Root locus indicates that the point $(k, f) = (\sqrt{3}, \frac{1}{9})$ is significant but there is nothing in the Routh array of $p(s) = s(s^2 + 1) + k(s^2 + f)$ to indicate what is so special about the point $(k, f) = (\sqrt{3}, \frac{1}{9})$. This is because some critical information has been omitted from consideration in the Routh approach. That critical information is the margin of stability of each point. We not only need to know if a point is stable but also how stable it is. The margin of stability can be included in the Routh array by considering the polynomial $p(s + \sigma) = 0$ rather than simply $p(s) = 0$. Writing out the Routh array for $p(s + \sigma) = 0$ gives:

$$\begin{array}{ll} 1 & 3\sigma^2 - 2\sigma k + 1 \\ k - 3\sigma & -\sigma^3 + k\sigma^2 - \sigma + kf \\ \dfrac{(k-3\sigma)(3\sigma^2 - 2\sigma k + 1) - (-\sigma^3 + k\sigma^2 - \sigma + kf)}{(k - 3\sigma)} & \\ -\sigma^3 + k\sigma^2 - \sigma + kf & \end{array}$$

Looking at each term in the left hand column one can see by inspection:

$k - 3\sigma = 0$ at $k = \sqrt{3}$, $\sigma = -\frac{1}{\sqrt{3}}$

$\dfrac{(k-3\sigma)(3\sigma^2 - 2\sigma k + 1) - (-\sigma^3 + k\sigma^2 - \sigma + kf)}{(k - 3\sigma)}$ is undefined at $k = \sqrt{3}$, $\sigma = -\frac{1}{\sqrt{3}}$, $f = \frac{1}{9}$

$-\sigma^3 + k\sigma^2 - \sigma + kf = 0$ at $k = \sqrt{3}$, $\sigma = -\frac{1}{\sqrt{3}}$, $f = \frac{1}{9}$

In this simple case it is easy to see that the parameter values that make each term equal to zero are the same terms that ensure optimum stability in the sense of root locus. There is one difficulty with the term $\dfrac{(k-3\sigma)(3\sigma^2 - 2\sigma k + 1) - (-\sigma^3 + k\sigma^2 - \sigma + kf)}{(k - 3\sigma)}$ as this in undefined at $k = \sqrt{3}$, $\sigma = -\frac{1}{\sqrt{3}}$. This may be overcome by considering the Jeltsch-Fichera array for $p(s + \sigma) = 0$ as follows:

$$\begin{array}{l} 1 \\ k - 3\sigma \\ k - fk - 2\sigma - 2k^2\sigma + 8k\sigma^2 - 8\sigma^3 \\ -(fk - \sigma(1 - k\sigma + \sigma^2))(2k^2\sigma + k(-1 + f - 8\sigma^2) + 2(\sigma + 4\sigma^3)) \end{array}$$

The second, third, and fourth and entries in the Jeltsch-Fichera array all equal zero at $k = \sqrt{3}, f = \dfrac{1}{9}, \sigma = \dfrac{1}{\sqrt{3}}$.

9.8 Generalised Routh array based optimum stability approach for designing a PI controller for the process $G(s) = k_2/(s+b)^m$

The characteristic equation for the combination of a PI controller $C(s) = k_1(s+a)/s$ and the process $G(s) = k_2/(s+b)^m$ is $p(s) = s(s+b)^m + k(s+a)$ where $k = k_1 k_2$. In section 5.2.6 we found that the following parameters lead to a system with optimum stability: $\sigma = \dfrac{-2b}{(m+1)}$, $k = b^m \left(\dfrac{m-1}{m+1}\right)^{m-1}$ and $a = \dfrac{4bm}{(m+1)^2}$.

For example, for the process $G(s) = 12/(s+5)^8$ these formulas give the PI controller parameters $k = 13452.7$ and $a = 1.975$. The optimum stability point is $\sigma = -1.111$.

We can construct the Jeltsch-Fichera array for $p(s+\sigma) = 0, \sigma < 0$ and, knowing that the last three entries of each column are equal to zero at the optimum stability point, derive these values again as follows.

$n_{24} = 0 \Rightarrow \sigma = -10/9$.

$n_{15} = 0 \Rightarrow k = 67258.7 \Rightarrow k_1 = 13452.7$.

$n_{25} = 0 \Rightarrow a = 1.975$.

So the PI controller derived from the Jeltsch-Fichera array is the same as the one derived from root locus. PI controllers for other processes of the type $G(s) = k_2/(s+b)^m$ can be designed in this way. This is another method, equivalent to the root locus method, for designing a system with optimum stability.

9.9 Summary of results in this chapter

A new procedure for designing controllers using the optimum stability approach is described. This method, based on the Routh array, is equivalent to the root-locus based optimum stability method and leads to controllers that are identical to the root locus based controllers.

9.10 Suggestions for future work

Apply the Routh array controller design approach to systems with other benchmark processes [58].

Chapter 10: Transform Methods for solving the Lyapunov matrix equations with applications

10.1 Introduction

When calculating performance integrals we frequently have to solve either the continuous time or the discrete time Lyapunov matrix equation. The purpose of this chapter is to describe a new method for solving these equations.

Several new results are presented in this chapter. Also, a new method is used here to derive a known result i.e. [83, 100, 101] on the use of Laplace Transforms in calculating performance integrals (described in section 10.2). It is an advantage of the new derivation procedure described here that some novel extensions are very natural and obvious. For example, previous authors did not extended their results to general time weighting. This is done in section 10.2.2. Another new result, in section 10.4, shows how to solve the continuous time Lyapunov Matrix Equation using the Laplace Transform. Also, section 10.5 shows how to solve the discrete time Lyapunov Matrix Equation using the $\mathcal{Z}$ Transform. Finally, these results are extended to the calculation of performance sums for discrete time systems in section 10.6.

One advantage of the method described here is that once a single Lyapunov Matrix equation has been solved, an infinite number of related Lyapunov Matrix equations have also been solved also. Another benefit of the approach taken here is that it allows one easily to study how the eigenvalues of $\mathbf{L}$, the solution to the Lyapunov Matrix equation $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$, are related to the eigenvalues of $\mathbf{A}$, the system matrix.

10.2 Calculating time weighted performance integrals using the Laplace transform

10.2.1 Performance integrals with polynomial time weighting using the Laplace transform

Loo [100], Ramar and Ramaswami [101], and Zhuang and Atherton [83] ([100] was simplified by Power [102]) pointed out that the following Theorem allows one to calculate time weighted performance integrals using Laplace Transforms:

$\mathcal{L}\{t^n e^2(t)\} = (-1)^n \left(\frac{\mathrm{d}^n}{\mathrm{ds}^n}\right)\mathcal{L}\{e^2(t)\}$ where $\mathcal{L}\{e^2(t)\}$ is the Laplace Transform of the square of the error signal.

For example: Let $\mathbf{A}=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -27 & -27 & -19 \end{pmatrix}, \mathbf{Q}=\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $\mathbf{x}(0)=(1,0,0)$

Using MacFarlane's procedure we get $\int_0^\infty t^3\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^3\mathbf{x}^T(0)\mathbf{L}_4\mathbf{x}(0) = 1.01806.$

Using the Laplace transform approach we first get:

$$\int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \frac{s^5+76s^4+1940s^3+19496s^2+65502s+91476}{s^6+76s^5+1940s^4+19550s^3+68580s^2+136296s+104976}$$

In this case we want to calculate $\int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$. so we use the above Theorem to get

$$\int_0^\infty t^3 \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^3 \frac{d^3}{ds^3}\left(\frac{s^5 + 76s^4 + 1940s^3 + 19496s^2 + 65502s + 91476}{s^6 + 76s^5 + 1940s^4 + 19550s^3 + 68580s^2 + 136296s + 104976} \right)\Bigg|_{s=0} = 1.01806$$

i.e. the same answer as the MacFarlane approach.

10.2.2 Performance integrals with general time weighting

If g(t) is a polynomial in t then $\mathcal{L}\{g(t)f(t)\} = g\left(-\frac{\mathrm{d}}{\mathrm{ds}}\right)\mathcal{L}\{f(t)\}$ [103]. We can use this to calculate a performance integral that is weighted by any function that can be expressed as a power series in "t". This was not pointed out in [100], [101] or [83].

Another Theorem that says: $\mathcal{L}\left\{\frac{1}{t}f(t)\right\} = \int_s^\infty F(s)ds$, provided $\lim_{t\to 0}\frac{f(t)}{t}$ exists [103].

So, provided the limit exists, this would allow us to calculate performance integrals that punish severely those errors that occur early on. As time progresses, the errors would matter less and less. I don't know an application where such a performance integral would be of value.

10.3 Analytic Solutions to the Lyapunov matrix equation

Malkin [104] pointed out that if $\mathbf{A}$ is Hurwitz then

$$\mathbf{L} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}dt \tag{10.1}$$

is a solution to:

$$\mathbf{A}^T\mathbf{L} + \mathbf{L}\mathbf{A} = -\mathbf{Q} \tag{10.2}$$

This can be seen by substituting (10.1) into (10.2) to get:

$$\begin{aligned}
&\mathbf{A}^T\left(\int_0^\infty e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}dt\right) + \left(\int_0^\infty e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}dt\right)\mathbf{A} \\
&= \int_0^\infty \left(\mathbf{A}^T e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t} + e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}\mathbf{A}\right)dt \\
&= \int_0^\infty \frac{d}{dt}\left(e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}\right)dt \\
&= e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}\Big|_0^\infty = 0 - \mathbf{Q} = -\mathbf{Q}
\end{aligned}$$

Malkin's [104]result may now be extended to show that

$$\mathbf{L} = \int_0^\infty e^{-st}e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}dt = \int_0^\infty e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t}\mathbf{Q}e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t}dt \tag{10.3}$$

is a solution to

$$\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L} + \mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right) = -\mathbf{Q}\,. \tag{10.4}$$

This also can be shown by substituting (10.3) into (10.4) as follows:

$$\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L} + \mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right) = -\mathbf{Q}$$

$$= \left(\mathbf{A}-\frac{s}{2}I\right)^T\int_0^\infty e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t}\mathbf{Q}e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t}dt + \int_0^\infty e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t}\mathbf{Q}e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t}dt\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$$

$$= \int_0^\infty \left[\left(\mathbf{A} - \frac{s}{2}\mathbf{I} \right)^T e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t} \mathbf{Q} e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t} + e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t} \mathbf{Q} e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t} \left(\mathbf{A} - \frac{s}{2}\mathbf{I} \right) \right] dt$$

$$= \int_0^\infty \frac{d}{dt} \left[e^{\left(A-\frac{s}{2}I\right)^T t} Q e^{\left(A-\frac{s}{2}I\right)t} \right] dt = \int_0^\infty d \left[e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t} \mathbf{Q} e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t} \right] = e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T t} \mathbf{Q} e^{\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)t} \Bigg|_0^\infty = 0 - \mathbf{Q} = -\mathbf{Q}$$

So (10.3) is a solution to (10.4).

In section 2.4.2 we derived a method for calculating exponentially weighted performance integrals $\int_0^\infty \exp(-\alpha t)\mathbf{x}^T\mathbf{Q}\mathbf{x}dt$. The analysis above is another way to derive this result. That is, $\int_0^\infty \exp(-\alpha t)\mathbf{x}^T\mathbf{Q}\mathbf{x}dt = \mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0)$ where $\mathbf{L}$ is the solution to:

$$\left(\mathbf{A} - \frac{\alpha}{2}\mathbf{I} \right)^T \mathbf{L} + \mathbf{L}\left(\mathbf{A} - \frac{\alpha}{2}\mathbf{I} \right) = -\mathbf{Q} \tag{10.5}$$

10.4 Using the Laplace transform to solve the continuous time Lyapunov matrix equation

We can write $\mathbf{L} = \int_0^\infty e^{-st} e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A}t} dt$ i.e. the solution to equation (10.4), as the Laplace Transform: $\mathcal{L}\{e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}\}$ and we can calculate these integrals by looking up tables. So by altering Malkin's integral to look like a Laplace transform we can solve (10.4) by looking up tables.

For example, if we call the state transition matrix $e^{\mathbf{A}t} = \mathbf{\Phi} = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix}$ and if $\mathbf{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ then we can write $\mathbf{L} = \mathcal{L}\{\mathbf{\Phi}^T\mathbf{Q}\mathbf{\Phi}\} = \mathcal{L}\begin{pmatrix} \Phi^2{}_{11} + \Phi^2{}_{21} & \Phi_{11}\Phi_{12} + \Phi_{21}\Phi_{22} \\ \Phi_{11}\Phi_{12} + \Phi_{21}\Phi_{22} & \Phi^2{}_{12} + \Phi^2{}_{22} \end{pmatrix}$. So, for example, $l_{11} = \mathcal{L}\{\Phi^2{}_{11} + \Phi^2{}_{21}\}$.

All the results in the following sections can be derived from this. Also, if we now let $s = 0$ we have the solution matrix for $\mathbf{A}^T\mathbf{L} + \mathbf{L}\mathbf{A} = -\mathbf{Q}$.

As another example if $\mathbf{Q} = \begin{pmatrix} 1 & 0 & . & 0 \\ 0 & 0 & . & . \\ . & . & . & . \\ 0 & . & . & 0 \end{pmatrix}$ then the performance integral $\int_0^\infty e^{-\alpha t}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ may be written as $\int_0^\infty e^{-\alpha t}x_1^2(t)dt$. This last integral is reminiscent of the Laplace Transform of $x_1^2(t)$ provided we say $\alpha = s$. So we can write $\int_0^\infty e^{-\alpha t}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \int_0^\infty e^{-st}x_1^2(t)dt = \mathcal{L}\{\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)\} = \mathcal{L}\{x_1^2(t)\}$.

From section 2.4.2 we know that given a continuous time linear system $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$, if $\mathbf{A}$ is Hurwitz, we can write $\int_0^\infty e^{-\alpha t}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \mathbf{x}^T(0)\mathbf{L}\mathbf{x}(0)$ where $\mathbf{L}$ is obtained from equation (10.5).

These facts suggest that the Lyapunov matrix equation (10.5) may be solved using Laplace transforms. (The converse is also true but less interesting i.e. you can calculate the Laplace transform of certain functions using the Lyapunov matrix equation.)

Note that in the expression $\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$ different $\mathbf{Q}'s$ give different combinations of solutions. For example, if all $q_{ij}=0$ except $q_{11}=1$ then $\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)=x_1^2(t)=error^2(t)$. However, if $\mathbf{Q}=\mathbf{I}$ then $\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)=x_1^2(t)+x_2^2(t)$. So, depending on the structure of $\mathbf{Q}$, the Laplace transform of different combinations of the solutions to $\frac{d\mathbf{x}(t)}{dt}=\mathbf{A}\mathbf{x}(t)$ are solutions to (10.4). The following examples illustrate these ideas.

10.4.1 Example 1 - Laplace transform solutions to the continuous time Lyapunov matrix equation

Solve the Lyapunov matrix equation $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ with $\mathbf{A}=\begin{pmatrix}0 & 1\\ -2 & -1\end{pmatrix}$ and $\mathbf{Q}=\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$ using Laplace Transforms. Note that different $x(0)'s$ (i.e. different initial conditions) may be used to pick off different entries in $\mathbf{L}$. So $\begin{pmatrix}1 & 0\end{pmatrix}\mathbf{L}\begin{pmatrix}1\\0\end{pmatrix}=l_{11}$; $\begin{pmatrix}0 & 1\end{pmatrix}\mathbf{L}\begin{pmatrix}0\\1\end{pmatrix}=l_{22}$ and $\begin{pmatrix}1 & 1\end{pmatrix}\mathbf{L}\begin{pmatrix}1\\1\end{pmatrix}=l_{11}+2l_{12}+l_{22}$ giving l_{12} and from symmetry of $\mathbf{L}$ we also have l_{21}.

Calculating the state transition matrix

The state transition matrix $\mathbf{\Phi}(t)$ gives a solution vector for any initial condition: $\mathbf{x}(t)=\mathbf{\Phi}(t)\mathbf{x}(0)$. $\mathbf{\Phi}(t)$ may be found by calculating either $\mathbf{\Phi}(t)=\mathcal{L}^{-1}\{(s\mathbf{I}-\mathbf{A})^{-1}\}$ or $\mathbf{\Phi}(t)=e^{\mathbf{A}t}$.

$$\mathbf{\Phi}(t)=\begin{pmatrix}\frac{1}{7}e^{-\frac{t}{2}}\left[7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right] & \frac{2e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}\\ \frac{-4e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}} & -\frac{1}{7}e^{-\frac{t}{2}}\left[-7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\end{pmatrix}$$

Calculating l_{11}

To find l_{11} take $\mathbf{x}(0)=\begin{pmatrix}1\\0\end{pmatrix}$ to get the solution vector $\mathbf{x}(t)=\begin{pmatrix}x_1(t)\\x_2(t)\end{pmatrix}=\begin{pmatrix}\frac{1}{7}e^{-\frac{t}{2}}\left[7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\\ \frac{-4e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}\end{pmatrix}$

Using Laplace Transform Tables we find:

$$\mathcal{L}\{x_1^2(t)\}=\mathcal{L}\left\{\left(\frac{1}{7}e^{-\frac{t}{2}}\left[7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\right)^2\right\}=\frac{s^2+3s+6}{s^3+3s^2+10s+8}$$

So now we have $\mathcal{L}\{x_1^2(t)\}=\int_0^\infty e^{-st}x_1^2(t)dt=\mathbf{x}^T(0)L\,\mathbf{x}(0)=\begin{pmatrix}1 & 0\end{pmatrix}\mathbf{L}\begin{pmatrix}1\\0\end{pmatrix}=l_{11}=\frac{s^2+3s+6}{s^3+3s^2+10s+8}$

This agrees with the l_{11} found by solving $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ directly.

Calculating l_{22}

To find l_{22} take $\mathbf{x}(0)=\begin{pmatrix}0\\1\end{pmatrix}$ to get the solution vector $\mathbf{x}(t)=\begin{pmatrix}x_1(t)\\x_2(t)\end{pmatrix}=\begin{pmatrix}\dfrac{2e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}\\-\frac{1}{7}e^{-\frac{t}{2}}\left[-7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\end{pmatrix}$.

Using Laplace Transform Tables, we find:

$$\mathcal{L}\{x_1^{\,2}(t)\}=\mathcal{L}\left\{\left(\frac{2e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}\right)^2\right\}=\frac{2}{s^3+3s^2+10s+8}$$

So now we have $\mathcal{L}\{x_1^2(t)\}=\int_0^\infty e^{-st}x_1^2(t)dt=\mathbf{x}^T(0)\mathbf{L}\,\mathbf{x}(0)=\begin{pmatrix}0 & 1\end{pmatrix}\mathbf{L}\begin{pmatrix}0\\1\end{pmatrix}=l_{22}=\dfrac{2}{s^3+3s^2+10s+8}$

Again, this agrees with the l_{22} found by solving $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ directly.

Calculating l_{21} (which, from the symmetry of $\mathbf{L}$, is equal to l_{12})

If we take $\mathbf{x}(0)=\begin{pmatrix}1\\1\end{pmatrix}$ we get the solution vector

$$\mathbf{x}(t)=\begin{pmatrix}x_1(t)\\x_2(t)\end{pmatrix}=\begin{pmatrix}\dfrac{2e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}+\frac{1}{7}e^{-\frac{t}{2}}\left[7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\\\dfrac{-4e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}-\frac{1}{7}e^{-\frac{t}{2}}\left[-7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\end{pmatrix}$$

Using Laplace Transform Tables, we find:

$$\mathcal{L}\{x_1^{\,2}(t)\}=\mathcal{L}\left\{\left(\frac{2e^{-\frac{t}{2}}Sin(\frac{\sqrt{7}t}{2})}{\sqrt{7}}+\frac{1}{7}e^{-\frac{t}{2}}\left[7Cos(\frac{\sqrt{7}t}{2})+\sqrt{7}Sin(\frac{\sqrt{7}t}{2})\right]\right)^2\right\}=\frac{s^2+5s+12}{s^3+3s^2+10s+8}$$

We know $\begin{pmatrix}1 & 1\end{pmatrix}\mathbf{L}\begin{pmatrix}1\\1\end{pmatrix}=l_{11}+2l_{12}+l_{22}\ \Rightarrow\ 2l_{12}=\begin{pmatrix}1 & 1\end{pmatrix}\mathbf{L}\begin{pmatrix}1\\1\end{pmatrix}-l_{11}-l_{22}$.

$$2l_{12}=\frac{s^2+5s+12}{s^3+3s^2+10s+8}-\frac{s^2+3s+6}{s^3+3s^2+10s+8}-\frac{2}{s^3+3s^2+10s+8}=\frac{2s+4}{s^3+3s^2+10s+8}$$

So $l_{12}=l_{21}=\dfrac{s+2}{s^3+3s^2+10s+8}$.

So the solution to $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ with $\mathbf{A}=\begin{pmatrix}0 & 1\\-2 & -1\end{pmatrix}$ and $\mathbf{Q}=\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$ is:

$$\mathbf{L}=\frac{1}{(s+1)(s^2+2s+8)}\begin{pmatrix}s^2+3s+6 & s+2\\ s+2 & 2\end{pmatrix} \tag{10.6}$$

The matrix $\mathbf{L}$ given by equation (10.6) indeterminate at the roots of $(s+1)(s^2+2s+8)$ i.e. at $s=-1$, $s=-1\pm j\sqrt{7}$. The explanation for this is as follows. We know from section 2.2.1 that the solution to $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ exist except when a combination of eigenvalues of $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ sum to zero i.e $(\lambda_i+\lambda_j)=0$. The eigenvalues of $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ are $(\lambda_1,\lambda_2)=\frac{1}{2}(-1-j\sqrt{7}-s,-1+j\sqrt{7}-s)$. All possible combinations of these eigenvalues are: $\lambda_1+\lambda_1=-1-j\sqrt{7}-s$, $\lambda_2+\lambda_2=-1+j\sqrt{7}-s$ and $\lambda_1+\lambda_2=-1-s$.

So solutions do not exist if any of these sums is zero i.e. at $s=-1$, $s=-1-j\sqrt{7}$ or at $s=-1+j\sqrt{7}$. But these are just the roots of $(1+s)(s^2+2s+8)$. So we would not expect $\mathbf{L}$ to be defined at these values of s.

10.4.2 Extending Example 1 to infinitely many Lyapunov Matrix Equations

$s=0 \Rightarrow \mathbf{L}=\begin{pmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix} 0 & 1 \\ -2 & -1 \end{pmatrix}$

$s=1 \Rightarrow \mathbf{L}=\frac{1}{22}\begin{pmatrix} 10 & 3 \\ 3 & 2 \end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix} 0 & 1 \\ -2 & -1 \end{pmatrix}-\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}=\begin{pmatrix} -\frac{1}{2} & 1 \\ -2 & -\frac{3}{2} \end{pmatrix}$

$s=2 \Rightarrow \mathbf{L}=\frac{1}{48}\begin{pmatrix} 16 & 4 \\ 4 & 2 \end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix} 0 & 1 \\ -2 & -1 \end{pmatrix}-\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}=\begin{pmatrix} -1 & 1 \\ -2 & -2 \end{pmatrix}$

Since $\mathbf{A}$ is Hurwitz this process can continue indefinitely as a non-Hurwitz $\mathbf{A}$ will never result from forming $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ - the real parts of the eigenvalues will just get more and more negative.

10.4.3 Example 2 - Laplace transform solutions to the continuous time Lyapunov matrix equation

Solve $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ with $\mathbf{A}=\begin{pmatrix} -6 & 1 \\ -2 & -3 \end{pmatrix}$, and $\mathbf{Q}=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ using Laplace Transforms. We must calculate Laplace transforms like: $\int_0^\infty e^{-st}\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt=\int_0^\infty e^{-st}[x_1^2(t)+x_2^2(t)]dt=\mathcal{L}\{x_1^2(t)+x_2^2(t)\}$

In this case, $\mathbf{\Phi}(t)=\mathcal{L}^{-1}\{(s\mathbf{I}-\mathbf{A})^{-1}\}=e^{-5t}\begin{pmatrix} 2-e^t & -1+e^t \\ 2-2e^t & -1+2e^t \end{pmatrix}$

Calculating l_{11}

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}=\mathbf{\Phi}(t)\begin{pmatrix} 1 \\ 0 \end{pmatrix}=e^{-5t}\begin{pmatrix} 2-e^t \\ 2-2e^t \end{pmatrix} \text{ so } l_{11}=\mathcal{L}\{x_1^2(t)+x_2^2(t)\}=\frac{s^2+15s+66}{(s+8)(s+9)(s+10)}$$

Calculating l_{22}

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}=\mathbf{\Phi}(t)\begin{pmatrix} 0 \\ 1 \end{pmatrix}=e^{-5t}\begin{pmatrix} -1+e^t \\ -1+2e^t \end{pmatrix} \text{ so } l_{22}=\mathcal{L}\{x_1^2(t)+x_2^2(t)\}=\frac{s^2+21s+114}{(s+8)(s+9)(s+10)}$$

Calculating $l_{12}(=l_{21})$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}=\mathbf{\Phi}(t)\begin{pmatrix} 1 \\ 1 \end{pmatrix}=e^{-5t}\begin{pmatrix} 2-e^t-1+e^t \\ 2-2e^t-1+2e^t \end{pmatrix}=e^{-5t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ so}$$

$$l_{21}=l_{12}=\frac{1}{2}\left(\mathcal{L}\{x_1^2(t)+x_2^2(t)\}-l_{11}-l_{22}\right)=\frac{-(s+18)}{(s+8)(s+9)(s+10)}$$

We now have $\mathbf{L}=\frac{1}{(s+8)(s+9)(s+10)}\begin{pmatrix} s^2+15s+66 & -(s+18) \\ -(s+18) & s^2+21s+114 \end{pmatrix}$ is the solution to $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ with $\mathbf{A}=\begin{pmatrix} -6 & 1 \\ -2 & -3 \end{pmatrix}$ and $\mathbf{Q}=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

As already seen in Example 1, this matrix become indeterminate for certain values of s, i.e. at $s=-8,-9$, or -10. The explanation for this is identical to that given in Example 2. We know from section 2.2.1 that the solution to $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ exist except when a combination of eigenvalues of $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ sum to zero i.e. $(\lambda_i+\lambda_j)=0$. In this case the eigenvalues of $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ are $(\lambda_1,\lambda_2)=(-5-\frac{s}{2},-4-\frac{s}{2})$. All possible combinations of these eigenvalues are: $\lambda_1+\lambda_1=-10-s$, $\lambda_2+\lambda_2=-8-s$ and $\lambda_1+\lambda_2=-9-s$. So solutions do not exist if any of these sums is zero i.e. at $s=-10$, $s=-8$ or at $s=-9$. But these are just the roots of $(s+8)(s+9)(s+10)$. So we would not expect $\mathbf{L}$ to be defined at these values of s.

10.4.4 Extending Example 2 to infinitely many Lyapunov Matrix Equations

$s=0\Rightarrow\mathbf{L}=\begin{pmatrix}\frac{11}{120} & -\frac{1}{40}\\ -\frac{1}{40} & \frac{19}{120}\end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix}-6 & 1\\ -2 & -3\end{pmatrix}$

$s=1\Rightarrow\mathbf{L}=\begin{pmatrix}\frac{41}{495} & -\frac{19}{990}\\ -\frac{19}{990} & \frac{68}{495}\end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix}-6 & 1\\ -2 & -3\end{pmatrix}-\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}=\begin{pmatrix}-\frac{11}{2} & 1\\ -2 & -\frac{7}{2}\end{pmatrix}$

$s=2\Rightarrow\mathbf{L}=\begin{pmatrix}\frac{5}{66} & -\frac{1}{66}\\ -\frac{1}{66} & \frac{4}{33}\end{pmatrix}$ is the solution to $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ where $\mathbf{A}=\begin{pmatrix}-6 & 1\\ -2 & -3\end{pmatrix}-\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}=\begin{pmatrix}-7 & 1\\ -2 & -4\end{pmatrix}$

Since $\mathbf{A}$ is Hurwitz this process can continue indefinitely as a non-Hurwitz $\mathbf{A}$ will never result from forming $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)$ - the real parts of the eigenvalues will just get more and more negative.

10.4.5 Example 3 - Laplace transform solutions to the continuous time Lyapunov matrix equation

Solve $\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)^T\mathbf{L}+\mathbf{L}\left(\mathbf{A}-\frac{s}{2}\mathbf{I}\right)=-\mathbf{Q}$ with $\mathbf{A}=\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}$, $\lambda_1,\lambda_2<0$, $\mathbf{Q}=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$ using Laplace Transforms.

Note in this case: $\mathbf{\Phi}(t)=\begin{pmatrix}e^{t\lambda_1} & 0\\ 0 & e^{t\lambda_2}\end{pmatrix}$. Also, the eigenvalues of $\mathbf{A}$ are (λ_1,λ_2).

Calculating l_{11}

$\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix}=\mathbf{\Phi}(t)\begin{pmatrix}1\\ 0\end{pmatrix}=\begin{pmatrix}e^{t\lambda_1}\\ 0\end{pmatrix}$ so $l_{11}=\mathcal{L}\{x_1^2(t)+x_2^2(t)\}=\frac{1}{(s-2\lambda_1)}$

Calculating l_{22}

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \mathbf{\Phi}(t)\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ e^{\lambda_2 t} \end{pmatrix} \text{ so } l_{22} = \mathcal{L}\{x_1^2(t) + x_2^2(t)\} = \frac{1}{(s - 2\lambda_2)}$$

Calculating $l_{12} = l_{21}$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \mathbf{\Phi}(t)\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} e^{t\lambda_1} \\ e^{t\lambda_2} \end{pmatrix} \text{ so } l_{21}(= l_{12}) = \frac{1}{2}\left(\mathcal{L}\{x_1^2(t) + x_2^2(t)\} - l_{11} - l_{22}\right) = 0$$

So $\mathbf{L} = \begin{pmatrix} \frac{1}{s - 2\lambda_1} & 0 \\ 0 & \frac{1}{s - 2\lambda_2} \end{pmatrix}$. The eigenvalues of $\mathbf{L}$ are $\mu_{1,2} = \left(\frac{1}{s - 2\lambda_1}, \quad \frac{1}{s - 2\lambda_2}\right)$. If $s = 0$ then we have the standard result that the eigenvalues of $\mathbf{L}$ are $\mu_{1,2} = \left(\frac{1}{-2\lambda_1}, \quad \frac{1}{-2\lambda_2}\right)$.

10.4.6 Discussion

The fact that it is possible to solve the Lyapunov matrix equation using the Laplace transform is interesting. Also, this procedure is analytical rather than numerical as, for example solving equation (2.4) (i.e. $(-1)^n \mathbf{D}\mathbf{E}^T\left[\mathbf{A}^T \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}^T\right]^n \mathbf{E}\vec{\mathbf{L}}_c = \vec{\mathbf{Q}}_c$) for $\mathbf{L}$ or using Barnett and Storey's method [105] so there are no problems with rounding errors.

One advantage of the Laplace transform method of solution is that once you have solved the Lyapunov matrix equation for one matrix $\mathbf{A}$ you have solved it for an infinite number of related $\mathbf{A}$'s.

Barnett's method [105] for solving the Lyapunov matrix equation requires only the inversion of $\mathbf{A}$. The Laplace Transform method described above requires solving the differential equation (by calculating $\mathcal{L}^{-1}\{(sI - \mathbf{A})^{-1}\}$ or by some other means) and calculating numerous objects that look like $\mathcal{L}\{x_1^2(t)\}$. So there is no computational advantage. It may be possible to get around this by using a canonical form for the matrix $\mathbf{A}$ and deriving a corresponding canonical form for the matrix $\mathbf{L}$.

Looking at these results another way, we have a means for solving ordinary differential equations by solving the Lyapunov matrix equation.

10.5 General solution to the discrete time Lyapunov matrix equation using the z transform

A set of results that are entirely analogous to those derived above for the continuous time Lyapunov matrix equation may now be derived for the discrete time Lyapunov matrix equation.

$$\mathbf{L} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{Q}e^{\mathbf{A}t}dt \tag{10.7}$$

is a solution to the continuous time Lyapunov matrix equation (10.2). Equation (10.7) has the structure

$$\mathbf{L} = \int_0^\infty \begin{pmatrix} \text{state transition} \\ \text{matrix} \end{pmatrix}^T \mathbf{Q} \begin{pmatrix} \text{state transition} \\ \text{matrix} \end{pmatrix} dt .$$

This suggests that the equation

$$\mathbf{L} = \sum_{k=0}^{\infty} \begin{pmatrix} \text{state transition} \\ \text{matrix} \end{pmatrix}^T \mathbf{Q} \begin{pmatrix} \text{state transition} \\ \text{matrix} \end{pmatrix}$$

could be a solution to the discrete time Lyapunov matrix equation:

$$\mathbf{A}^T\mathbf{L}\mathbf{A} - \mathbf{L} = -\mathbf{Q} \tag{10.8}$$

In the discrete time case we have $\mathbf{x}(k) = \mathbf{A}^k\mathbf{x}(0)$ so we can regard $\mathbf{A}^k$ as the discrete time version of the state transition matrix. So we assume that

$$\mathbf{L} = \sum_{k=0}^{\infty} (\mathbf{A}^T)^k \mathbf{Q}\mathbf{A}^k \tag{10.9}$$

is a solution to (10.8). We substitute (10.9) into (10.8) to get:

$$\begin{aligned}
&\mathbf{A}^T\left(\sum_{k=0}^{\infty}(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k\right)\mathbf{A} - \sum_{k=0}^{\infty}(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k \\
&= \sum_{k=0}^{\infty}\left[(\mathbf{A}^T)^{k+1}\mathbf{Q}\mathbf{A}^{k+1} - (\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k\right] \\
&= \mathbf{A}^T\mathbf{Q}\mathbf{A} - \mathbf{Q} + (\mathbf{A}^T)^2\mathbf{Q}\mathbf{A}^2 - \mathbf{A}^T\mathbf{Q}\mathbf{A} + (\mathbf{A}^T)^3\mathbf{Q}\mathbf{A}^3 - (\mathbf{A}^T)^2\mathbf{Q}\mathbf{A}^2 .... \\
&= -\mathbf{Q}
\end{aligned}$$

So, $\mathbf{L} = \sum_{k=0}^{\infty}(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k$ is indeed a solution of $\mathbf{A}^T\mathbf{L}\mathbf{A} - \mathbf{L} = -\mathbf{Q}$. LaSalle [106] derives this result in a different way. It is interesting to note that the solutions to the continuous time and discrete time Lyapunov matrix equations have identical structures.

As in the continuous time case, we can now say that if (10.9) is the solution to (10.8) then

$$\mathbf{L} = \sum_{k=0}^{\infty} e^{-ks}(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k = \sum_{k=0}^{\infty} e^{-\frac{k}{2}s}(\mathbf{A}^T)^k\mathbf{Q}e^{-\frac{k}{2}s}\mathbf{A}^k \tag{10.10}$$

is a solution of

$$(e^{-\frac{k}{2}s}\mathbf{A}^T)\mathbf{L}(e^{-\frac{k}{2}s}\mathbf{A}) - \mathbf{L} = -\mathbf{Q} . \tag{10.11}$$

But $\sum_{k=0}^{\infty} e^{-ks}(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k = \mathcal{Z}\{(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k\}$ i.e. the $\mathcal{Z}$ transform of $(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k$. So $\mathcal{Z}\{(\mathbf{A}^T)^k\mathbf{Q}\mathbf{A}^k\}$ is a solution of $(e^{-\frac{k}{2}s}\mathbf{A}^T)\mathbf{L}(e^{-\frac{k}{2}s}\mathbf{A}) - \mathbf{L} = -\mathbf{Q}$. Choosing different $k's$ results in different Lyapunov matrix equations so, when we have solved one Lyapunov matrix equation we have solved an infinite number of related Lyapunov matrix equations.

We can solve (10.11) using $\mathcal{Z}$-transforms and then let $s = 0$ to find the solution to (10.8). As for the continuous time case described in section 10.4, we need to find the state transition matrix here too. In the discrete time case this means we have to calculate $(\mathbf{A}^T)^k$ or $(\mathbf{A})^k$ before we can solve the discrete time Lyapunov matrix equation. Alternatively, the state transition matrix in a discrete time control system may be calculated s follows [107] $\mathbf{A}^k = \mathcal{Z}^{-1}\{(z\mathbf{I} - \mathbf{A})^{-1}z\}$.

We met (10.11) before in section 2.10 in the context of exponentially weighted performance sums. Here, we have derived again the formula for calculating the exponentially weighted performance sums but this time using the $\mathcal{Z}$ transform.

It is nice to see that the Laplace Transform can solve the continuous time Lyapunov matrix equation and the $\mathcal{Z}$ Transform can solve the discrete time Lyapunov matrix equation.

10.6 Calculating time weighted performance sums using $\mathcal{Z}$ transforms

Jury [108] gives the following identities that are analogous to those found in the continuous time case: if $\mathcal{Z}\{f(t)\} = F(z)$ then $: \mathcal{Z}\{t f(t)\} = -Tz\frac{d}{dz}F(z)$ and $\mathcal{Z}\{t^k f(t)\} = -Tz\frac{d}{dz}F_1(z)$ where $F_1(z) = \mathcal{Z}\{t^{k-1} f(t)\}$ where $k > 0$ and an integer. As for the continuous time case these identities can be used to calculate time weighted performance sums for the discrete time case.

10.7 Suggestions for future work

(a) Mäkilä [109] has shown that the standard unilateral Laplace transform may be used to solve initial value problems for linear constant coefficient differential equations with jump discontinuities (steps) in the input. An example of such an equation is: $\frac{d^3y}{dt^3} + 3\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + y = 2\frac{d^2u}{dt^2} - u$, where $u(t) = 1$ for $t < 0$, and $u(t) = 4$ for $t > 0$. It would be interesting to see if the methods described in this chapter extend to these problems.

(b) Time weighted performance integrals may be evaluated using the Routh array [110, 111] so it should be possible to find a link between Laplace transforms and the entries in the Routh array.

(c) Investigate the Matrix Laplace Transform described by MacFarlane [112] and the Matrix $\mathcal{Z}$ - transform described by Jury [34], and see if one has solved every Lyapunov Matrix equation once the solution to one equation is known.

Chapter 11: Summary and Conclusions

Procedures for calculating performance integrals J_n were developed within a unified framework provided by the Kronecker product and MacFarlane's procedure [30]. MacFarlane's procedure has been simplified and extended to discrete time systems. All new procedures have been presented in a way that may be implemented easily using standard computer programs.

The methods for calculating discrete time control system performance sums S_n for the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$ proposed by Jury [34] and Barnett [33] and Mansour [35], are simplified by using the number triangle given in section 2.6. The coefficients that arise in these calculations have several interesting properties that have not been remarked upon before – including a relationship with Stirling numbers of the second kind. These properties are described in section 2.7.

The number of equations to be solved when calculating J_n or S_n may be reduced to a minimum by the use of the **E** and **D** matrices described in section 2.2.2, 2.2.3, and 2.2.4.This method is simpler than that derived by Chen and Shieh [32].

Expressions for exponentially weighted performance integrals and sums were derived in sections 2.4.1, 2.4.2, and 2.10.

A root locus based optimum stability approach was used to design a second order controller for a second order, unstable process with non-minimum phase from Doyle [45]. The performance of the controller using root locus was compared to a sixth order controller from the literature designed using H_∞ methods. The controller designed using root locus based optimum stability is of lower order than the H_∞ controller and it results in a system that is more robust and with very much enhanced performance when compared to the system using a H_∞ controller.

In chapter 4 two controller design methodologies were described – one that uses root locus based optimum stability and the other based on minimising performance integrals of the form $\int_0^\infty \exp(\alpha t)\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$. The values for **Q** and for α are problem specific. Root locus is used to make the minimum value of α as large as possible. Two examples were used to illustrate that the controllers obtained in both cases are identical. The first example was of a second order system and the second example was of a PI controller for a specific unstable process. In both of the examples a plot of the performance integral against the system parameter exhibits a smooth broad shape with a unique minimum point. This means that the system is robust to large variations in the design parameter as this result in small variations in the performance integral. This implies in turn that designing a system with maximum eigenvalue sensitivity does not lead to a degradation of performance.

In addition, the step reference response, the impulse disturbance response, and the step disturbance response of the optimum stability PI controller are shown to be superior to a PI controller that was designed by using the centroid of a stability region as the design point. The appeal of the centroid as a design point is simply its distance from the stability boundary. However, this fails to take into account the degree of stability of the point. The optimum stability point has the virtue of not only being in the stability region but of occupying a point of optimum stability.

It was shown that systems with controllers that were designed for multi-lag processes using either Lyapunov based optimum stability or root locus based optimum stability exhibit greater robustness margins, and smoother response characteristics than systems with controllers designed using a selection of other methods from the literature. However, systems with controllers designed using optimum stability tend to exhibit greater peak disturbance responses.

General formulas were given for root locus based optimum stability design for a PI controller, and a restricted class of PID controller for process $G(s) = k_2/(s+b)^m$. A root locus method for fine-tuning the standard Ziegler-Nichols controller parameters was described in sections 5.2.2 and 5.3.2. The resulting controller outperforms the original Ziegler-Nichols design.

Chapter 6 presents a new procedure for the design of PI controllers for general FOLPD process. Equations (6.15) and (6.16) are derived - two simple equations that allow the designer to calculate the controller parameter values using the process parameters only. This method is applied to the design of PI controllers for specific FOLPD processes that are presently discussed in the literature and compare PI controllers that were designed using optimum stability with PI controllers that were designed using a variety of performance integral and domain of stability considerations. By calculating gain margins, phase margins, delay margins, and plotting various response curves we see that the controllers that were designed using optimum stability offer, by these standard measures, enhanced performance when compared with the other controllers.

The application of the analytic root locus to the design of a PI controller for an integrator with time delay is described and attention drawn to a relationship between the root locus equation for a system with time delay and the Lambert W function.

A new idea for tuning PI and PID controllers was presented, based on analogy with the maximum power transfer theorem from linear AC circuit theory. The approach has been identified as one that specifies the phase margin and the frequency at which it is effective. It has been illustrated by designs for third order and eighth order members of a restricted class of asymptotically stable processes, considered by Datta et al. [4]. Explicit formulas, involving the process parameters only, were derived for calculating the max power transfer based PI controller parameters for the process $G(s) = k_2/(s+b)^m$. Controllers designed by an optimum parameter space approach due to Datta et al. [4] give a much more oscillatory behavior and longer settling time.

An interesting observation is that the controller designed by using max power transfer considerations is very similar to the one designed from root locus based optimum stability considerations - the difference being in the value of the gain.

A model of the human balance control system was studied and four parameters selected – one for each major control loop. Nyquist analysis was used to select pairs of parameters that lead to Optimum Phase Margin but this is a graphical procedure and therefore approximate. The Lyapunov matrix equation was used to select all four parameters simultaneously – these parameters gave optimum eigenvalue location. The parameters selected using Nyquist analysis tended to lead to systems that are underdamped and have very oscillatory impulse responses. Parameters derived from the Lyapunov matrix equation and optimum

eigenvalue location, gave impulse responses that were fare less oscillatory and settled after about 6 seconds to 10 seconds.

A new procedure for designing controllers using the Routh array was described. This method is shown to be equivalent to the root-locus based optimum stability method and leads to controllers that are identical to the root locus based controllers.

New methods to solve the continuous-time and discrete-time Lyapunov matrix equations were described: these methods employ the Laplace transform and the $\mathcal{Z}$ transform respectively.

11.1 Suggestions for further work

Several suggestions for further work were made in sections 2.12, 4.4, 5.6, 6.8, 7.6, 8.6, 9.10, and 10.7. Other suggestions are:

Controller design based on other possible definitions of optimum stability. For example, using the circle criterion where the radial distance from the centre of the circle to the Nyquist diagram is to be as large as possible.

Controller design based on optimum stability using Lyapunov functions. This would involve the synthesis of systems to satisfy a Lyapunov function V, where dV/dt is as negative as possible. The use of Lyapunov theory to synthesize systems has received some attention [15, 113, 114] and the notion of synthesizing a control system that makes dV/dt as negative as possible is mentioned by Kalman and Bertram [17] and by Brogan [107].

Appendix A: Performance integrals

A.1 The need for performance integrals

A standard way of designing robust control systems is to choose controller parameters that optimise a performance measure [12, 13]. Integrals of the time-weighted error squared such as:

$I_n = \int_0^\infty t^n \, \vec{x}^{\,T}(t) Q \vec{x}(t) dt = \int_0^\infty t^n \, e^2(t) dt$, where $Q = \begin{pmatrix} 1 & 0 & . & 0 \\ 0 & 0 & . & 0 \\ . & . & . & . \\ 0 & 0 & 0 & 0 \end{pmatrix}$, $\vec{x}^{\,T}(t)$ is the state vector in phase variable form, the error squared signal is given by $e^2(t) = \vec{x}^{\,T}(t) Q \vec{x}(t)$, are well-established performance measures. Although these integrals have been used since the 1940's they are still relevant today – see for example Ackermann [93] , Wade and Johnson [76], and Albertos and Sala [115]. Procedures for calculating these integrals have been described elsewhere [30, 105] as well as in section 2.2.

Some authors follow Anderson and Moore [116] and include the process input $\vec{u}(t)$ in the performance integral to get $\int_0^\infty f(t)[\vec{x}^{\,T}(t) Q \vec{x}(t) + \vec{u}^{\,T}(t) R \vec{u}(t)] \, dt$ where the matrices Q and R are problem specific and $f(t)$ is a time weighting function. Their intention is to choose $\vec{u}(t)$ to minimise this integral and thereby design a controller that is optimal in that sense. The approach taken in this paper, and by many authors such as Wade and Johnson[76], Visioli [117], Zhuang and Atherton [118], Dan-Isa and Atherton [119], Ho et al. [22], and Åström et al. [120], is to set $\vec{u}(t) = 0$ and concentrate on minimisation of the time-weighted error squared signal alone. Choosing parameters that minimise some performance measure is a standard way of designing robust control systems. Also, performance measures may be used to give a figure of merit to the design of a system. When used along with other indicators such as step input response, disturbance rejection, settling time, overshoot, etc., they give a design method or else a method for assessing a deign methodology. System performance may be assessed objectively by using an index of performance such as $\int_0^\infty e^2(t) dt$ or $\int_0^\infty t e^2(t) dt$ as a quality index – where $e(t) =$ error.

Ackermann [93] refers to such measures as being of great importance and he uses these measures routinely as a performance indicator [121]. Datta et al. [29] list the minimisation of such integrals as one of the three new controller design techniques. Albertos and Sala [115] describe optimization strategies, including the minimisation of such performance integrals, as "..a powerful strategy with significant impact in practice" and "…of extraordinary importance in general control theory".

Performance integrals such as $\int_0^\infty t e^2(t) dt$ or $\int_0^\infty \exp(\alpha t) e^2(t) dt$, where $e(t)$ is the error signal, may be considered as system norms. These integrals are norms in the sense that they give a measure of the distance between the desired output and the actual output. An integral such as $\int_0^\infty t e^2(t) dt$ satisfies the three requirements of a norm (for $t > 0$) i.e. it is a real number; it is either 0 or positive; it is only zero if $e(t) = 0$.

(These integrals converge iff $e(t) \to 0$ as $t \to \infty$). If the distance represented by such an integral is minimised then the behaviour of the system is optimised with respect to that norm. Designing control systems to minimise such norms is a long established approach [30, 122-124]. It is shown in section 4.1 that, in some at least one interesting and general case, the root locus approach is actually a graphical technique that minimises the system norm $J = \int_0^\infty \exp(\alpha t)\left\{e^2(t) + \frac{1}{\omega_n^2}\left(\frac{de(t)}{dt}\right)^2\right\}dt$. This criterion minimises the accumulated time-weighted sum of the square of the error plus the square of its derivative. A discussion of the use of performance integrals in control system design is given in section A.2.

A.2 Using performance integrals to design a controller

The following is an outline of a controller design procedure based on the minimization of a performance integral.

(a) Choose a controller structure such as a PI controller: $C(s) = k_p + \frac{k_i}{s}$.

(b) Find the vector of initial conditions $\vec{x}(0)$ corresponding to the stimulus being studied.

(c) Evaluate I_n - this will be a function of the controller parameters (k_p, k_i).

(d) Find those values of controller parameters (k_p, k_i) that minimise I_n. We now have the controller parameters (k_p, k_i) that minimise I_n in response to the stimulus chosen in step (b).

A.3 Overview of the literature on performance integrals

In 1965 MacFarlane [112] gave many references for the history of the problem of calculating performance integrals for continuous time systems, starting in 1887 with Volterra. However, he does not mention performance measures for discrete time systems, as work on this topic did not begin until 1970.

A.3.1 Performance integrals for continuous time systems

1943

Hall [122] proposes $\int_0^\infty e^2(t)dt$ as a performance integral– where $e(t) =$ error.

1951

Nims [123] proposes the time-weighted performance integral: $\int_0^\infty te(t)dt$

1953

Graham and Lathrop [125] compared eight figures of merit for the transient response of Linear Time Invariant systems. These figures of merit include $\int_0^\infty e(t)dt$, $\int_0^\infty t|e(t)|dt$, $\int_0^\infty t^2|e(t)|^2 dt$, $\int_0^\infty t^2e^2(t)dt$, etc. This paper created great interest in $ITAE = \int_0^\infty t|e(t)|dt$. The ITAE does not penalise large initial errors as much as long duration transients. Graham and Lathrop [125] used analogue computer studies to derive tables of standard transfer functions that minimise the ITAE for different inputs.

1954

Westcott [124] provides a detailed study of the performance integral: $\int_0^\infty te^2(t)dt$

1961

Schultz and Rideout [126] provide a comprehensive survey of the work to that date and discusses performance measures for systems with random inputs.

1963

MacFarlane [30] gives a solution to the continuous time Lyapunov matrix equation $\mathbf{A}^T\mathbf{L}+\mathbf{L}\mathbf{A}=-\mathbf{Q}$ as $\vec{\mathbf{l}}=\mathbf{B}^{-1}\vec{\mathbf{q}}$ where the matrix $\mathbf{B}$, an $\frac{n}{2}(n+1)\times\frac{n}{2}(n+1)$ matrix, is formed from the matrix $\mathbf{A}$ by following an algorithm. This algorithm exploits the symmetry of $\mathbf{L}$ in order to reduce the number of equations to be solved to a minimum. He then states that in general $\vec{\mathbf{l}}_n=\mathbf{B}^{-n}\vec{\mathbf{q}}$ where $\vec{\mathbf{l}}_n$ is the contracted vector solution to $\mathbf{A}^T\mathbf{L}_n+\mathbf{L}_n\mathbf{A}=\mathbf{L}_{n-1}$. Performance integrals can be calculated using this $\mathbf{L}_n$ without recursively solving a series of Lyapunov equations. MacFarlane's algorithm was simplified by Chen and Shieh [32]. Their approach was misunderstood initially [127, 128] and up to now has been considered to be the most efficient possible by, for example, Gajic and Qureshi [129].

In section 2.2.2 of this thesis a different algorithm is given that exploits the symmetry of $\mathbf{L}$ to reduce the number of equations to be solved to a minimum. This algorithm is simpler than those proposed by MacFarlane [30] and by Chen and Shieh [32].

1964

P. C. Parks [130] expresses the system matrix in Schwarz canonical form. He then integrates the Lyapunov function $V=-2b_1^2x_n^2$ [17] and shows how to calculate $\int_0^\infty x^2(t)dt$. He then calculates $\int_0^\infty tx^2(t)dt$, and $\int_0^\infty t^2x^2(t)dt$ for a second order system by first integrating by parts and then using the Lyapunov equation approach again. He is actually using an early version MacFarlane's procedure [30] independently of MacFarlane.

J. E. Diamessis [131, 132] also expresses the system matrix in Schwarz canonical form. He then changes variables from x to w using $x=e^{-(\mu/2)t}w$. Using the idea of a moment generating function from probability theory he then derives an expression for $\int_0^\infty t^kx^2(t)dt$ that is similar to that derived by Parks [130].

1966 and 1967

Lehoczky [111] and Csaki and Lehoczky [110] show how to calculate the following integrals using the second, third, fourth etc. columns of the Routh array: $I_n=\int_0^\infty x^2(t)dt$, $J_n=\int_0^\infty tx^2(t)dt$, and $K_n=\int_0^\infty t^2x^2(t)dt$ where x(t) is a solution to a differential equation of degree n. Ramar and Ramaswami [101] describe a generalisation of this.

1967

H. M. Power [133] puts the system matrix into Routh canonical form and then calculates $\int_0^\infty x^2(t)dt$ by calculating $\dot{V} = -2b_1 x_k^2$, and then integrating both sides. He extends this result to integrals such as $\int_0^\infty \sum(\alpha_k x_k)^2 dt$.

S. G. Loo [100] writes $F(s) = \int_0^\infty e^{-st} t^k x^2(t)dt = \sum_{k=0}^{\infty} \frac{(-s)^k}{k!} \int_0^\infty t^k x^2(t)dt = \sum_{k=0}^{\infty} \frac{(-s)^k}{k!} J_k$. He then regards $F(s)$ as a moment generating function for the integrals J_k. He can then write $J_0 = F(0)$ and $J_k = (-1)^k F^{(k)}(0)$, $k = 1,2,3...$ where $F^{(k)}(0)$ is the kth derivative of F evaluated at $s = 0$. This procedure is identical to that used by Diamessis [131, 132]. A simplification of Loo's procedure was given by Power [102].

1969

MacFarlane [31] uses Kronecker Products and Kronecker Sums [36, 134, 135] to evaluate a wide range of functionals of the dynamical behaviour of continuous time linear systems. In particular, he gives the following expression for the integral: $\int_0^\infty (t^r \mathbf{x} \otimes \mathbf{x})dt = (-1)^{(r+1)} r!(\mathbf{A} \oplus \mathbf{A})^{-(r+1)} \mathbf{x}(0) \otimes \mathbf{x}(0)$. Where $\otimes$ represents the Kronecker product and $\oplus$ represents the Kronecker sum. This formula is derived in section 2.2 in a simple way. MacFarlane [31] extends this result to general time weighting functions that can be expressed as a power series in t. MacFarlane's own original procedure [30] is used in section 2.4.2 to provide a new derivation for this equation.

1971

Anderson and Moore [116] discuss the design of closed loop systems using of integrals of the type $\int_0^\infty \exp(2\alpha t)\{\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)\}dt$. They study the problem of choosing a control law (i.e. a form for $\mathbf{u}(t)$) that minimises this integral. The problem addressed in this thesis is that of choosing a point in the parameter plane that minimises a similar integral that has $\mathbf{u}(t) = 0$.

1984

Nishikawa et al. [136] choose PID parameters so as to minimise the exponentially weighted performance integral $J(\beta) = \int_0^\infty [\Delta x(t) e^{\beta t}]^2 dt$ where $\Delta x(t) = error$ and β is chosen to suit a particular process.

1987

Hwang [137] says that, when t is large, performance integrals such as $I = \int_0^\infty t^k \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ can place too much weight on the tail of the impulse response. He suggests the use of an additional term such as $\exp(-at)$ to mitigate this effect and advances $\int_0^\infty t^k \exp(-at)\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ as a performance integral. He then describes a method for calculating these measures for the discrete time case.

1994

Cai and Wang [138] minimise $J(x,t)=\int_0^\infty \{e(x,t)\exp(kt)\}^2 dt$ when designing the control loop in a high performance switching mode power supply. He introduces the exponential term in order to improve the transient response and to accelerate the convergence of his algorithm. He reports that this method results in: system stability, response time increased by a factor of 2, and disturbance rejection improved by up to 30%.

2004

[115] demonstrates that the performance integral $J=\frac{1}{2}\int_0^\infty e^{2\alpha t}[\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)+\mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)]dt$ can be used if a minimum exponential stability $e^{-\alpha t}, (\alpha>0)$ is desired. For this integral to be finite a solution must be obtained such that $x(t)$ and $u(t)$ "...are bounded in the norm $Me^{-\alpha t}$ (M an unknown constant)." He also defines the discrete time performance sum $J=\frac{1}{2}\sum_{k=0}^{\infty}\alpha^{2k}(\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k+\mathbf{u}_k^T\mathbf{R}\mathbf{u}_k)$ and states that this leads to a system response bounded by $M\beta^k$, $(\beta<1, \alpha=\frac{1}{\beta}>1)$.

This study

Two expressions are derived for the exponentially weighted performance integral in section 2.4. An entirely different method for calculating performance integrals with generalised time weighting is given in section 10.2.2.

A.3.2 Performance sums for discrete time systems

1970

Man [139] derives the following formula for a time weighted performance sum for linear discrete time systems: $J_r=\sum_{k=0}^{\infty}k^r\mathbf{x}_k^T\mathbf{S}_0\mathbf{x}_k=\mathbf{x}^T(0)\sum_{i=1}^{\infty}\sum_{j=1}^{r}[{}^rC_j i^{r-j}(-1)^{j+1}\mathbf{A}^{T\,i}\mathbf{L}_1\mathbf{A}^i]\mathbf{x}(0)$ where $\mathbf{S}_j-\mathbf{A}^T\mathbf{S}_j\mathbf{A}=\mathbf{S}_{j-1}$. He does not give a general expression for b_{ij} but he gives examples up to $J_3=\sum_{k=0}^{\infty}k^3\mathbf{x}_k^T\mathbf{S}_0\mathbf{x}_k=x_0^T(6\mathbf{S}_4-12\mathbf{S}_3+7\mathbf{S}_2-\mathbf{S}_1)\mathbf{x}_0$ where $\mathbf{S}_1$ is given by $\mathbf{S}_1-\mathbf{A}^T\mathbf{S}_1\mathbf{A}=\mathbf{Q}$, $S_2\ \mathbf{S}_2-\mathbf{A}^T\mathbf{S}_2\mathbf{A}=\mathbf{S}_1$ etc. etc.

1974

Barnett [33] improves the formula due to Man [139] by deriving a general expression for b_{ij} using Kronecker products and by finding the sum of an infinite series of matrices. He derives the following formula for the time weighted performance sum: $J_i=\sum_{k=0}^{\infty}k^i\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k=\mathbf{x}_0^T\left[\sum_{j=1}^{i+1}b_{ij}\mathbf{L}_j\right]\mathbf{x}_0$ where $\mathbf{A}^T\mathbf{L}_{i+1}\mathbf{A}-\mathbf{L}_{i+1}=\mathbf{L}_i$, $\mathbf{L}_0=\mathbf{Q}$ and the coefficients b_{ij} are given by $b_{ij}=(-1)^{i+j-1}\sum_{s=0}^{j-1}(-1)^s\binom{j-1}{s}(j-s)^i$.

Their procedures involve calculations such as: $J_4=\sum_{k=0}^{\infty}k^4\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k=\mathbf{x}_0^T(24\mathbf{L}_5-60\mathbf{L}_4+50\mathbf{L}_3-15\mathbf{L}_2+\mathbf{L}_1)\mathbf{x}_0$ where $\mathbf{A}^T\mathbf{L}_1\mathbf{A}-\mathbf{L}_1=\mathbf{L}_0$, $\mathbf{A}^T\mathbf{L}_2\mathbf{A}-\mathbf{L}_2=\mathbf{L}_1$, $\mathbf{A}^T\mathbf{L}_3\mathbf{A}-\mathbf{L}_3=\mathbf{L}_2$ etc., etc.

A new derivation for Barnett's formula based on finite differences [39, 40] is given in section 2.6. A novel and simple method for calculating the coefficients b_{ij} is given in section 2.6.

1975

Jury [34] derives a recursive method to calculate for calculating the b_{ij}'s for Man's [139] result. This process is derived in a new way in section 2.5and simplified greatly in section 2.6. He also extends the result to quadratic sums with more general time weighting.

1979

Mansour [35] derives time weighted performance sums for continuous and discrete time systems using the transformation to Schwartz matrix form. He then estimates the abscissa of stability of continuous time systems and the margin of stability of discrete time systems as functions of the time weighted performance sums for different initial conditions. He derives $J_r = \mathbf{x}^T(0)\left[\sum_{i=1}^{r+1}\sum_{j=0}^{i-1}(-1)^{r-i+j-1}(i-j)^r\binom{i-1}{j}L_i\right]\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_{i+1}\mathbf{A}-\mathbf{L}_{i+1}=\mathbf{L}_i$. This is an elaboration of a formula due to Man [139] and was also derived by Barnett [33] and Jury and Gutman [34].

1984

Fukata and Tamura [41] derive results similar to those in [139], [33] [34] and [35] .
He extends these results to sampled data systems.

1998 - this study

The procedures described above for calculating the performance sum for discrete time control systems were still in use in 1998 by, for example, Al-Sunni and LEwis [140].

As mentioned above references [33-35] derived complicated expansions for $J_i = \sum_{k=0}^{\infty} k^i \mathbf{x}_k^T \mathbf{Q}\mathbf{x}_k$. A simple and novel number triangle that may be used to calculate the coefficients in these expansions is given in section 2.6.

A new derivation for Barnett's formula based on finite differences [39, 40] is given in section 2.6.

An entirely different an novel procedure for calculating performance sums with general time weighting based on $\mathcal{Z}$ -transforms is described in section 10.6.

A.4 MacFarlane's procedure for calculating performance integrals

A procedure to calculate performance integrals using the Lyapunov Matrix Equation is described by MacFarlane [30]. An example of the application of this procedure is given below. This detailed example is given here in order to demonstrate clearly MacFarlane's procedure for continuous time systems as an entirely analogous procedure is developed in section 2.5 for discrete time systems.

Consider the system $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$ and the initial conditions $\mathbf{x}(0)$.

Note: if $\mathbf{Q} = \begin{pmatrix} 1 & 0 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ . & . & . & . & . \\ 0 & . & . & . & 0 \\ 0 & 0 & . & 0 & 0 \end{pmatrix}$ then $\mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t) = (error)^2 = e^2(t)$.

The classical Integral of Time x Squared Error (ITSE) for this system is $J = \int_0^\infty t\,\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$ and it may be evaluated as follows.

Let $V_1 = t\mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)$.

Then $\frac{dV_1}{dt} = \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t) + t[\mathbf{x}^T(t)(\mathbf{A}^T\mathbf{L}_1 + \mathbf{L}_1\mathbf{A})\mathbf{x}(t)]$.

Let $\mathbf{A}^T\mathbf{L}_1 + \mathbf{L}_1\mathbf{A} = -\mathbf{Q}$ (call this the "first Lyapunov equation")

So $\frac{dV_1}{dt} = \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t) - t\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$

Now integrate both sides of this last equation from 0 to ∞:

$\underset{t\to\infty}{Lim} V_1(t) - V_1(0) = \int_0^\infty \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)dt - \int_0^\infty t\,\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt$

If the system is asymptotically stable then every component in $\mathbf{x}(t)$ is a superposition of decaying exponentials. Therefore $\mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)$ is a superposition of decaying exponentials and these will dominate the polynomial t term. So $\underset{t\to\infty}{Lim} V_1(t) = \underset{t\to\infty}{Lim}(t\,\mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)) = 0$. Also: $V_1(0) = 0$. So the LHS = 0 and we have:

$$\int_0^\infty \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)dt = \int_0^\infty t\,\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = ITSE$$

Then $ITSE$ can now be written as $\int_0^\infty \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)dt$

Now, let $V_2 = \mathbf{x}^T(t)\mathbf{L}_2\mathbf{x}(t)$

Then $\frac{dV_2}{dt} = \mathbf{x}^T(\mathbf{A}^T\mathbf{L}_2 + \mathbf{L}_2\mathbf{A})\mathbf{x}$

Let $\mathbf{A}^T\mathbf{L}_2 + \mathbf{L}_2\mathbf{A} = -\mathbf{L}_1$ (call this the "second Lyapunov equation")

So $\frac{dV_2}{dt} = -\mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)$

Now, integrate this last equation from 0 to ∞ to get $\underset{t\to\infty}{Lim} V_2(t) - V_2(0) = -\int_0^\infty \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)dt$

As before, $\underset{t\to\infty}{Lim}(\mathbf{x}^T(t)\mathbf{L}_2\mathbf{x}(t)) = 0$.

For $t = 0$, $-V_2(0) = -\mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$.

Therefore: $\mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0) = \int_0^\infty \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)dt = ITSE$

So, MacFarlane's procedure [30] for calculating, for example, the ITSE consists of solving $\mathbf{A}^T\mathbf{L}_1 + \mathbf{L}_1\mathbf{A} = -\mathbf{Q}$ for $\mathbf{L}_1$; then solving $\mathbf{A}^T\mathbf{L}_2 + \mathbf{L}_2\mathbf{A} = -\mathbf{L}_1$ for $\mathbf{L}_2$; then calculating $ITSE = \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$. It is not necessary to solve the equations of motion in order to do these calculations. It is sufficient to known the initial conditions and to solve two imbedded Lyapunov equations.

In general, MacFarlane's procedure may be written as:

$$\int_0^\infty t^n \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = (-1)^{n+1} n! \mathbf{x}^T(0)\mathbf{L}_{n+1}\mathbf{x}(0) \text{ where } \mathbf{A}^T\mathbf{L}_{n+1} + \mathbf{L}_{n+1}\mathbf{A} = -\mathbf{L}_n \text{ and } \mathbf{L}_0 = \mathbf{Q},\ n = 0,1,2...$$

$\mathbf{x}(0)$ is the vector of initial conditions, and $\mathbf{L}_{n+1}$ is the solution to the Lyapunov matrix equation $\mathbf{A}^T\mathbf{L}_{n+1} + \mathbf{L}_{n+1}\mathbf{A} = -\mathbf{L}_n$ [129] for the dynamical system $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\,\mathbf{x}(t)$. The matrix $\mathbf{Q}$ may be positive semi-definite [38]. Different structures for $\mathbf{Q}$ may be used to combine the error signal and its derivatives e.g. using

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & . & 0 \\ 0 & 1 & 0 & . & 0 \\ 0 & 0 & 0 & . & 0 \\ . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$ in I_n leads to the integral: $\int_0^\infty t^n \left[e^2(t) + \left(\frac{de(t)}{dt} \right)^2 \right] dt$.

An important feature to note is that given the initial conditions and the matrix $\mathbf{Q}$ then I_n is unique because the solution to the Lyapunov Matrix equation is unique. It is this uniqueness of I_n that makes it a useful performance measure.

Our interest in these integrals arises from this intimate connection which we have found between minimisation of exponentially weighted performance integrals and a principle of optimum stability which involves, in the continuous-time case, choosing nominal system parameters that place the rightmost eigenvalue as deep in the left half plane as possible, subject to structural relations between system parameters. Thus, root locus methods give us a graphical procedure for optimizing α - and therefore the degree of stability. That is, we can use root locus methods to make the minimum absolute value of α as large as possible. Optimum stability is discussed in [20, 47, 67].

Appendix B: New derivation of the discrete time system performance sums S_n for n=0,1,2 and 3

In the following derivations the function $V(x(k)) = (k-1)^n \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ is defined and used in the first step in the calculation of performance sums for discrete time control systems. This new function is analogous to the function $V(x(t)) = t^n \mathbf{x}^T(t)\mathbf{L}_1\mathbf{x}(t)$ used by MacFarlane [30] when calculating performance integrals for continuous time control systems.

Consider the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$ where all the eigenvalues of $\mathbf{A}$ lie inside the unit circle. Define a family of performance sums as $S_n = \sum_{k=0}^{\infty} k^n \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$.

B.1 Discrete time system performance sums with n = 0

The performance sum S_0 may be evaluated as follows: $S_0 = \sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$. This may be shown as follows.

First let $V(\mathbf{x}(k)) = \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$

Then $\Delta V(\mathbf{x}(k)) = V(\mathbf{x}(k+1)) - V(\mathbf{x}(k))$

$$
\begin{aligned}
&= \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&= [\mathbf{A}\mathbf{x}(k)]^T\mathbf{L}_1[\mathbf{A}\mathbf{x}(k)] - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&= \mathbf{x}^T(k)[\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1]\mathbf{x}(k) \\
&= -\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)
\end{aligned}
$$

$$
\begin{aligned}
S_0 = \sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) &= -\sum_{k=0}^{\infty} [\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&= -[-\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)] = \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)
\end{aligned}
$$

So, to evaluate S_0 we must first solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$ and then calculate $S_0 = \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$. Ogata [141] derives this result in a different way.

B.2 Discrete time system performance sums with n = 1

The performance sum S_1 may be evaluated as follows: $S_1 = \sum_{k=0}^{\infty} k\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = \mathbf{x}^T(0)[\mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)$

Where $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ and $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$. This may be shown as follows:

First take $V_1(\mathbf{x}(k)) = (k-1)(\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k))$

$$\begin{aligned}
\text{Then } \Delta V_1(\mathbf{x}(k)) &= \Delta[k-1](\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) + (k-1)\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + \Delta[k-1]\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&= \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + (\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) \\
&= \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)([\mathbf{A}\mathbf{x}(k)]^T\mathbf{L}_1[\mathbf{A}\mathbf{x}(k)] - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) + ([\mathbf{A}\mathbf{x}(k)]^T\mathbf{L}_1[\mathbf{A}\mathbf{x}(k)] - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) \\
&= \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)\mathbf{x}^T(k)(\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1)\mathbf{x}(k) + \mathbf{x}^T(k)(\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1)\mathbf{x}(k) \\
&= \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - (k-1)\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) - \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) \\
&= \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) + \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) - \mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)
\end{aligned}$$

$$k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \Delta V_1(\mathbf{x}(k))$$

$$\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k))$$

To evaluate $\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)$ we will first find $\sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ and then $\sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k))$

From Example 1: $\sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$ where $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ and $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$.

Now to find: $\sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k))$.

$$\Delta V_1(\mathbf{x}(k)) = \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + (\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k))$$

$$\begin{aligned}
\sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k)) &= \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) - \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) + \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) - \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) \\
&+\mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + (0)\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - (0)\mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) \\
&+\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) + (1)\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - (1)\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) + \mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) \\
&+\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) + (2)\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) - (2)\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) + \mathbf{x}^T(4)\mathbf{L}_1 x(4) - \mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) \\
&+\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) + (3)\mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5) - (3)\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) + \mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5) - \mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) \\
&\text{.....} \\
&\text{.....(all terms cancel except } \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)) \\
&\text{.....} \\
&= \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)
\end{aligned}$$

So now we can write:

$$\begin{aligned}
\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) &= \sum_{k=0}^{\infty} \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty} \Delta V_1(\mathbf{x}(k)) \\
&= \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0) - \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0) \\
&= \mathbf{x}^T(0)[\mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)
\end{aligned}$$

So $S_1 = \sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = \mathbf{x}^T(0)[\mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)$

So, to evaluate S_1 we must first solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$ and then solve $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$ for $\mathbf{L}_2$.

We can then calculate $S_1 = \mathbf{x}^T(0)[\mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)$.

B.3 Discrete time system performance sums with n = 2

The performance sum $S_2 = \sum_{k=0}^{\infty} k^2\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ may be evaluated as follows:

$S_2 = \sum_{k=0}^{\infty} k^2 \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = \mathbf{x}^T(0)[\mathbf{L}_1 - 3\mathbf{L}_2 + 2\mathbf{L}_3]\mathbf{x}(0)$. Where we first solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$, then solve $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$ for $\mathbf{L}_2$ and finally solve $\mathbf{A}^T\mathbf{L}_3\mathbf{A} - \mathbf{L}_3 = -\mathbf{L}_2$ for $\mathbf{L}_3$.

This may be shown as follows.

First take $V_1(\mathbf{x}(k)) = (k-1)^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$

Then $\Delta V_1(\mathbf{x}(k)) = \Delta[(k-1)^2](\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) + (k-1)^2\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + \Delta[(k-1)^2]\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$

Recall that: $\Delta[(k-1)^2] = \Delta[k^2 - 2k + 1] = ((k+1)^2 - 2(k+1) + 1) - (k^2 - 2k + 1) = 2k - 1$

$$\begin{aligned}
\Delta V_1(\mathbf{x}(k)) &= (2k-1)\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k-1)^2[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + (2k-1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k \\
&= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k^2 - 2k + 1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&\quad + (2k-1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^2\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - k^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&\quad - 2k\,\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) + 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&\quad + 2k\,\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)
\end{aligned}$$

$$\begin{aligned}
\Delta V_1(\mathbf{x}(k)) &= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^2\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - k^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) \\
&= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^2[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] \\
&= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^2\mathbf{x}^T(k)[A^T\mathbf{L}_1A - \mathbf{L}_1]\mathbf{x}(k) \\
&= 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - k^2\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)
\end{aligned}$$

Rearranging terms gives: $k^2\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \Delta V_1(\mathbf{x}(k))$

So: $\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = 2\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$

To evaluate $\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ we will first find $2\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ then $\sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ and finally $\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$.

From Example 2 we have $2\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = 2\mathbf{x}^T(0)[\mathbf{L}_3 - \mathbf{L}_2]\mathbf{x}(0)$

From Example 1 we have $\sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$

It remains to calculate $\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$

$\Delta V_1(\mathbf{x}(k)) = 2k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^2\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - k^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$

$$\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k)) = -\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$
$$+2\mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) - \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1)$$
$$+4\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) + 4\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - 4\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2)$$
$$+6\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - \mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) + 9\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) - 9\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3)$$
$$+8\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) - \mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) + 16\mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5) - 16\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4)$$
$$\text{.....}$$
$$\text{.....(all terms cancel except } -\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0))$$
$$\text{.....}$$
$$= -\mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$

Gathering terms together we get:

$$\sum_{k=0}^{\infty}k^2\,\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) = 2\sum_{k=0}^{\infty}k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$$
$$= 2\mathbf{x}^T(0)[\mathbf{L}_3 - \mathbf{L}_2]\mathbf{x}(0) - \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0) + \mathbf{x}^T(0)\mathbf{L}_1\mathbf{x}(0)$$
$$= \mathbf{x}^T(0)[2\mathbf{L}_3 - 2\mathbf{L}_2 - \mathbf{L}_2 + \mathbf{L}_1]\mathbf{x}(0)$$
$$= \mathbf{x}^T(0)[2\mathbf{L}_3 - 3\mathbf{L}_2 + \mathbf{L}_1]\mathbf{x}(0)$$
$$= \mathbf{x}^T(0)[\mathbf{L}_1 - 3\mathbf{L}_2 + 2\mathbf{L}_3]\mathbf{x}(0)$$

So, to evaluate S_2 we must first solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$, then solve $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$ for $\mathbf{L}_2$ and finally solve $\mathbf{A}^T\mathbf{L}_3\mathbf{A} - \mathbf{L}_3 = -\mathbf{L}_2$ for $\mathbf{L}_3$. We can then calculate $S_2 = \mathbf{x}^T(0)[\mathbf{L}_1 - 3\mathbf{L}_2 + 2\mathbf{L}_3]\mathbf{x}(0)$.

B.4 Discrete time system performance sums with n=3

The performance sum $S_3 = \sum_{k=0}^{\infty}k^3\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$ may be evaluated as follows:

$S_3 = \mathbf{x}^T(0)[-\mathbf{L}_1 + 7\mathbf{L}_2 - 12\mathbf{L}_3 + 6\mathbf{L}_4]\mathbf{x}(0)$. Where we solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$; solve $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$ for $\mathbf{L}_2$; solve $\mathbf{A}^T\mathbf{L}_3\mathbf{A} - \mathbf{L}_3 = -\mathbf{L}_2$ for $\mathbf{L}_3$; and finally solve $\mathbf{A}^T\mathbf{L}_4\mathbf{A} - \mathbf{L}_4 = -\mathbf{L}_3$ for $\mathbf{L}_4$.

This may be shown as follows.

First take $V_1(\mathbf{x}(k)) = (k-1)^3\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$

Then $\Delta V_1(\mathbf{x}(k)) = \Delta[(k-1)^3](\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)) + (k-1)^3\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)] + \Delta[(k-1)^3]\Delta[\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$

Recall that:

$$\Delta[(k-1)^3] = \Delta[k^3 - 3k^2 + 3k - 1] = ((k+1)^3 - 3(k+1)^2 + 3(k+1) - 1) - (k^3 - 3k^2 + 3k - 1) = 3k^2 - 3k + 1$$

$$\Delta V_1(\mathbf{x}(k)) = (3k^2 - 3k + 1)\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + (k^3 - 3k^2 + 3k - 1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$$
$$+(3k^2 - 3k + 1)[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$$
$$= 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^3\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - k^3\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$
$$-3k^2x^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) + 3k^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + 3k\,\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$
$$-\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + 3k^2\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - 3k^2\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$
$$-3k\,\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) + 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$$

Cancelling terms gives

$$\Delta V_1(\mathbf{x}(k)) = 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - k^3\mathbf{x}^T(k)\mathbf{L}_1x(k) + k^3\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1)$$
$$= 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^3[\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1) - \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)]$$
$$= 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^3\mathbf{x}^T(k)[\mathbf{A}^T\mathbf{L}\mathbf{A} - \mathbf{L}]\mathbf{x}(k)$$
$$= 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - k^3\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)$$

Rearranging terms gives:

$$k^3\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \Delta V_1(\mathbf{x}(k))$$

So:

$$\sum_{k=0}^{\infty} k^3\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = 3\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$$

To evaluate $\sum_{k=0}^{\infty} k^3\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k)$ we will first find $3\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$, then $3\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$, then $\sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k)$ and finally $\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$.

From Example 3 we have $3\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = 3\mathbf{x}^T(0)[\mathbf{L}_2 - 3\mathbf{L}_3 + 2\mathbf{L}_4]\mathbf{x}(0)$

From Example 2 we have $3\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = 3\mathbf{x}^T(0)[\mathbf{L}_3 - \mathbf{L}_2]\mathbf{x}(0)$

From Example 1 we have $\sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) = \mathbf{x}^T(0)\mathbf{L}_2\mathbf{x}(0)$

It remains to calculate $\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))$

$$\Delta V_1(\mathbf{x}(k)) = 3k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - k^3\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + k^3\mathbf{x}^T(k+1)\mathbf{L}_1\mathbf{x}(k+1)$$

$$\begin{aligned}\sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k)) = {} & \mathbf{x}^T(0)\mathbf{L}_1 x(0)\\ & +3\mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) - 3\mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) - \mathbf{x}^T(1)\mathbf{L}_1\mathbf{x}(1) + \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2)\\ & +12\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - 6\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) + \mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) - 8\mathbf{x}^T(2)\mathbf{L}_1\mathbf{x}(2) + 8\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3)\\ & +27\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - 9\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) + \mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) - 27\mathbf{x}^T(3)\mathbf{L}_1\mathbf{x}(3) + 27\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4)\\ & +48\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) - 12\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) + \mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) - 64\mathbf{x}^T(4)\mathbf{L}_1\mathbf{x}(4) + 64\mathbf{x}^T(5)\mathbf{L}_1\mathbf{x}(5)\\ & \ldots\ldots\\ & \ldots\ldots\text{(all terms cancel except } \mathbf{x}^T(0)L_1\mathbf{x}(0))\\ & \ldots\ldots\\ & = \mathbf{x}^T(0)L_1\mathbf{x}(0)\end{aligned}$$

Gathering terms together we get:

$$\begin{aligned}\sum_{k=0}^{\infty} k^3\mathbf{x}^T(k)[Q]\mathbf{x}(k) &= 3\sum_{k=0}^{\infty} k^2\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - 3\sum_{k=0}^{\infty} k\,\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) + \sum_{k=0}^{\infty}\mathbf{x}^T(k)\mathbf{L}_1\mathbf{x}(k) - \sum_{k=0}^{\infty}\Delta V_1(\mathbf{x}(k))\\ &= \mathbf{x}^T(0)[3(\mathbf{L}_2 - 3\mathbf{L}_3 + 2\mathbf{L}_4) - 3(\mathbf{L}_3 - \mathbf{L}_2) + \mathbf{L}_2 - \mathbf{L}_1]\mathbf{x}(0)\\ &= \mathbf{x}^T(0)[-\mathbf{L}_1 + 7\mathbf{L}_2 - 12\mathbf{L}_3 + 6\mathbf{L}_4]\mathbf{x}(0)\end{aligned}$$

So, to evaluate S_3 we first solve $\mathbf{A}^T\mathbf{L}_1\mathbf{A} - \mathbf{L}_1 = -\mathbf{Q}$ for $\mathbf{L}_1$, solve $\mathbf{A}^T\mathbf{L}_2\mathbf{A} - \mathbf{L}_2 = -\mathbf{L}_1$ for $\mathbf{L}_2$, solve $\mathbf{A}^T\mathbf{L}_3\mathbf{A} - \mathbf{L}_3 = -\mathbf{L}_2$ for $\mathbf{L}_3$ and finally solve $\mathbf{A}^T\mathbf{L}_4\mathbf{A} - \mathbf{L}_4 = -\mathbf{L}_3$ for $\mathbf{L}_4$. We can then calculate $S_3 = \sum_{k=0}^{\infty} k^3\mathbf{x}^T(k)[\mathbf{Q}]\mathbf{x}(k) = \mathbf{x}^T(0)[-\mathbf{L}_1 + 7\mathbf{L}_2 - 12\mathbf{L}_3 + 6\mathbf{L}_4]\mathbf{x}(0)$.

In general, to calculate S_j we need to know $S_{j-1}, S_{j-2}, \ldots S_1$.

Appendix C: H_2 and H_∞ norms

Control system design methodologies are rooted in classical ideas such as the Nyquist diagram, Bode plots, the Routh array etc. For example, H_∞ controller design is an analytic method that has a manifestation in the Nyquist and Bode diagrams. Specifically, H_∞ technique is an analytic method for finding either (i) the peak gain value of the Bode magnitude plot or (ii) the distance in the complex plane from the origin to the farthest point on the Nyquist plot of $G(j\omega)$ [142] – where $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is the transfer function matrix of the linear, time-invariant, stable system $\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu}, \mathbf{y} = \mathbf{Cx}$.

C.1 A note on terminology

The use of the letter H in H_2 and H_∞ is a reference to fact that these objects are norms in Hardy spaces. These spaces are named after the English mathematician G.H.Hardy (1877-1947). On the other hand, Stein [143] wonders if the "H" might also stand for "Hype".

C.2 System norms

The chief measures of robustness are the gain and phase margin [144] . Some other useful measures are descried in this section. A general feedback control system may be represented as follows [25, 45, 145, 146] .

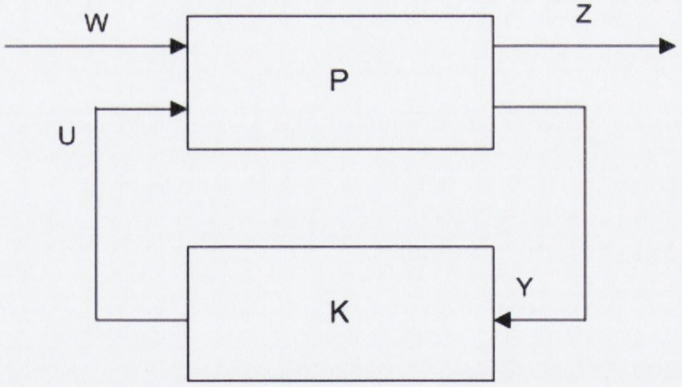


Figure C.1 Block diagram for a general feedback control system.

P is the linear time invariant process to be controlled (the inputs to P and the outputs from P are, in general, vector-valued functions)

K is the controller

Y is the vector of sensor measurements

U is the vector of inputs generated by K

W in the vector of all exogenous inputs to P

Z is the vector of variables we wish to control

The general control problem is to design a controller that will keep the size of the variable, z, small in the presence of w. So the size of the closed loop transfer function from w to z is to be small. Appropriate measures of the size of a transfer function need to be defined and such measures are called system norms [45]. This analytic approach to controller design can be more useful than the geometric approach in the case of multivariable systems where graphical methods may fail.

C.2.1 H_2 - norm controllers and error minimization

Let $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be transfer function matrix of a linear, time-invariant, stable system given by $\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu}, \mathbf{y} = \mathbf{Cx}$. The H_2 norm of G(s), denoted by $\|G\|_2$, is defined [25, 45, 145, 146] as:

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} trace\left[G(j\omega) G^*(j\omega) \right] d\omega \right)^{\frac{1}{2}}$$

$$= \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} \sum_{i=1}^{r} \sigma_i^2 \left[G(j\omega) \right] d\omega \right)^{\frac{1}{2}}$$

where σ_i denotes the i^{th} singular value, $G^*(j\omega)$ is the complex conjugate transpose of $G(j\omega)$, and r is the rank of $G(j\omega)$. The H_2 control problem is to find a controller that stabilises the process and minimises the H_2 - norm of the transfer matrix from w to z.

Alternatively [147], the H_2 -optimal control problem is to find a controller such that the 2-norm of the integral square error (ISE) measure, $\left\| \int_0^\infty e^2(t) dt \right\|_2$, is minimised for a specific input. Integral square error measures and other performance measures are discussed in chapter 2.

C.2.2 Computing the H_2 norm

If $\mathbf{L}_c$ is the controllability Gramian of $(\mathbf{A}, \mathbf{B})$ (i.e. $\mathbf{L}_c$ is the solution of $\mathbf{A}\mathbf{L}_c + \mathbf{L}_c\mathbf{A}^T = -\mathbf{B}\mathbf{B}^T$) and $\mathbf{L}_0$ is the observability Gramian of $(\mathbf{A}, \mathbf{C})$ (i.e. $\mathbf{L}_o$ is the solution to $\mathbf{A}^T\mathbf{L}_o + \mathbf{L}_o\mathbf{A} = -\mathbf{C}^T\mathbf{C}$) then:

$$\|G(s)\|_2 = \left[trace\left(\mathbf{C}\mathbf{L}_c\mathbf{C}^T \right) \right]^{\frac{1}{2}} = \left[trace\left(\mathbf{B}^T\mathbf{L}_0\mathbf{B} \right) \right]^{\frac{1}{2}}$$

This procedure for computing the H_2 norm involves the solution of linear Lyapunov equations and can be done without iteration. Examples of these calculations and of H_2 controller synthesis are given in [25, 45, 145, 146, 148].

C.2.3 Physical interpretation of the H_2 norm

If G(s) is the transfer function of a system driven by independent, zero mean, unit intensity white noise, then the sum of the variances of the outputs is the square of the H_2 norm. So, the H_2 norm of G(s) gives an exact measure of the power or signal strength of the output of a system driven with unit intensity white noise.

C.2.4 H_∞ - norm controllers and error minimization

Each member of a set of inputs to system produces a corresponding error. The H_∞ optimal controller is designed to minimise the worst error that can arise from any input in the set.

The H_∞ norm of an error signal can be expressed as [147] $\min_c \|e(t)\|_\infty = \min_c \sup_\omega |S(j\omega)W(j\omega)|$ where, $\min_c$ means the minimum over all controllers, $S(j\omega)$ is the sensitivity function and $W(j\omega)$ is a designer specified, frequency-dependant weighting function. So the H_∞ - optimal controller minimizes the maximum magnitude of the weighted sensitivity function over frequency range ω, or in mathematical terms, minimises the ∞ - norm of the sensitivity function weighted by $W(j\omega)$ [149].

According to Boyd et al. [150] "The H_∞ norm arises in control theory as a measure of disturbance rejection...".

Green and Limebeer [145] say that "H_∞ optimal control is a frequency domain optimization and synthesis theory that was developed in response to the need for a synthesis procedure that explicitly addresses questions of modeling errors." Later he says: "H_∞ control problems can be cast as constrained minimisation problems. The constraints come from an internal stability requirement and the object we seek to minimise is the infinity norm of some closed-loop transfer function."

Ackermann [142] and others [45, 147] state that the H_∞ norm of $G(s)$ may also be viewed as (i) the peak gain value of the Bode magnitude plot or (ii) the distance in the complex plane from the origin to the farthest point on the Nyquist plot of $G(j\omega)$.

Panagopoulous and Åström [57] have shown that H_∞ design is related to classical design and specifically to the Nyquist diagram.

We could also say that the H_∞ norm is the largest gain of the system taken over all frequencies.

Let $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be transfer function matrix of a linear, time-invariant, stable system. The H_∞ norm of G(s), denoted by $\|G(s)\|_\infty$, is defined [25, 45, 145, 146] as :

$$\|G(s)\|_\infty = \sup_\omega \sigma_{\max}\left[G(j\omega)\right]$$

where sup is the supremum (or least upper bound), and $\sigma_{\max}\left[G(j\omega)\right]$ is the largest singular value of $G(j\omega)$. So $\|G(s)\|_\infty$ is the supremum of the function $\sigma_{\max}\left[G(j\omega)\right]$. The H_∞ norm is the maximum value of $\sigma_{\max}\left[G(j\omega)\right]$ over all frequencies ω.

C.2.5 Computing the H_∞ norm

Consider the transfer function $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, with $\mathbf{A}$ stable. If $\gamma > 0$, then $\|G\|_\infty < \gamma$ iff the Hamiltonian matrix $\mathbf{H} = \begin{pmatrix} \mathbf{A} & \frac{1}{\gamma^2}\mathbf{B}\mathbf{B}^T \\ -\mathbf{C}^T\mathbf{C} & -\mathbf{A}^T \end{pmatrix}$ has no eigenvalues on the $j\omega - axis$. This fact allows us to compute a bound γ on $\|G\|_\infty$ such that $\|G\|_\infty < \gamma$. To find $\|G\|_\infty = \gamma_{\min}$, select a $\gamma > 0$ and test if $\mathbf{H}$ has eigenvalues on the $j\omega - axis$. If it does, increase γ and recompute the eigenvalues of $\mathbf{H}$. If it does not have

these eigenvalues then decrease γ and recompute the eigenvalues of $\mathbf{H}$. This iteration is continued until $\gamma_{\min}$ is calculated to the desired tolerance. Examples of these calculations and of H_∞ controller synthesis are given in [25, 45, 145, 146, 148]. The H_∞ control problem is to find a controller that stabilises the process and minimises the H_∞ - norm of the transfer matrix from w to z.

Computing the H_∞ - norm of a transfer function or synthesizing H_∞ - norm controllers may be done using commercial software such as MatLab [151], or freeware such as OPT [152] or Scilab [153]. Many algorithms are based on a search method and Boyd et al. [154, 155] have described a very efficient bisection algorithm for calculating H_∞ norms.

C.2.6 Physical interpretation of the H_∞ norm

The H_∞ norm has a physically meaningful interpretation for the system y(s) = G(s)u(s) [156]. When a system is driven with a unit magnitude sinusoidal input at a specific frequency, $\sigma_{\max}\left[G(j\omega)\right]$ is the largest possible output size for the corresponding sinusoidal input. So the H_∞ norm is largest possible amplification over all frequencies of a unit sinusoidal input. That is, it quantifies the greatest increase in energy that can occur between the input and output of a given system.

C.2.7 Advantages and disadvantages of using the H_∞ norm to design controllers

Some of the benefits of H_∞ design are as follows [157]:

1. The synthesis problem has well defined stability and robustness properties which can easily be predicted once the system is specified.
2. Design iterations, which enable trade offs to be achieved, can easily be accomplished.
3. For certain given classes of uncertainty, robustness margins can be guaranteed and the steps in going from the uncertainty to the optimal problem are straightforward.
4. Software is readily available in most commercial packages and users do not need a high degree of skill.
5. The links to LQG solutions often enable stochastic properties to be optimised in addition to robustness.

Some of the disadvantages with H_∞ design are as follows [157]:

1. Robust solutions may not give adequate transient responses or other properties and hence often such requirements have to be relaxed. The reason for using the H_∞ approach is therefore less obvious in this case.
2. Although it is not necessary to understand the theory to be able to use the packages, it is a daunting prospect to try to understand the underlying theory without the help of formal courses.
3. Although H_∞ problems are theoretically tractable it is not clear that the physical robustness problems match the theoretical problem posed.
4. The maximization of stability and robustness margins is a rather more complicated problem than the sensitivity minimisation H_∞ design problem suggests.

Two further disadvantages, described by Keel et al. [24] are:

5. H_∞ techniques produce controllers of higher order than necessary – Keel describes an example of how H_∞ design technique produce a sixth order controller for a second order process. See section 3.4.

6. H_∞ techniques produce fragile controllers – Keel shows how a H_∞ controller for a test process produces a closed loop system that becomes unstable if the gain is reduced by one part in a thousand - see section 3.4.

In [158] Campos-Delgado and Zhou point out that:

7. "…there is no guarantee from the current state-of-the-art design techniques [sic, including the H_∞ technique] that the controllers obtained through these techniques are stable themselves." This is a problem as: "… unstable controllers tend to be highly sensitive to model uncertainties, unmodelled nonlinearities, and sensor/actuator faults."

Ho and Lin remark in [159]:

8. "..design of the optimal or robust PID controller is a computationally intractable task using H_∞ and μ - synthesis design techniques."

and in Ho et al. [160] say:

9. "Indeed most of the optimization techniques of modern optimal control including H_2, H_∞ and L_1 Optimal and μ cannot be directly used in applications because they cannot accommodate fixed structure controllers such as PID." So these controller design methodologies cannot accommodate constraints on the controller order or structure.

Finally, Paganini [161] remarks:

10. "In terms of disturbance rejection H_∞ - optimal control favors allpass closed-loop transfer functions, paying the price of increased sensitivity over a large bandwidth to reduce sensitivity at the worst frequency, a poor choice under the broadband disturbances of most real-world applications. This problem can be alleviated by frequency-weighted H_∞, but weight selection becomes a largely ad hoc procedure, attempting to "distort" H_∞ into a measure of the response to broadband noise…"

This is an inevitable consequence of the H_∞ design approach of making the worst-case frequency response as good as possible – the focus is directed away from the other, possibly more likely, inputs.

Appendix D: History and present state of tuning rules for PI and PID controllers

D.1 Introduction

PI and PID controllers are very common in industry. For example, Ho [22] reports that more than 90% of control loops for process control systems in Japan are of the PID type. However, most of these controllers are badly tuned [54]. So the development of a simple method that can be used to tune a PID controller to a process in order to achieve desired closed loop performance has been of great practical interest for many years. Several software products designed specifically for tuning PID controllers are available [162].

The history of PID controllers is given by Bennett [163] and an interesting retrospective by Ziegler and Nichols appeared in 1993 [164]. There are several excellent summaries of the progress made to date on PID controller tuning and applications. Lelic and Gagic [165] presented a guide to over 300 articles on PID controllers that were published in various journals between 1990 and 1999. O'Dwyer [28] has described the tuning rules that were proposed between 1942 and 2005 for PI and PID controllers for processes with time delay. In addition, Johnson and Moradi [166] and Åström and Hagglund [72] provide very recent reviews of the area. The future of PID control is discussed by Åström and Hagglund in [167, 168].

D.2 Ziegler –Nichols

D.2.1 Some general comments on Ziegler-Nichols tuning

Systematic tuning of P, PI, and PID controllers started over 60 years ago in 1942 with Ziegler and Nichols [50]. Although the Ziegler-Nichols tuning rules are generally considered to be heuristic and empirical, a frequency domain interpretation of these rules has been given by dePaor [169]. He has found, for example, that a system tuned by the ultimate sensitivity method is guaranteed to have closed loop stability and adequate phase margin.

Generally, the advantages of the Ziegler-Nichols tuning methods are that they:

(a) produce a system with a "reasonable" response rather than one with a response that is "optimised" in some sense [170].

(b) do not require a model of the process itself but relies on obtaining parameters from the process step response or from the value of gain that makes the process marginally stable

(c) are easy for process operators to remember and apply, as an operator does not require any familiarity with transfer functions.

(d) generally result in systems with good disturbance rejection.

On the other hand, the disadvantages of the Ziegler-Nichols tuning methods are that they:

(a) were developed in the context of a particular PID controller i.e. Taylor's Fulscope 100 controller. Ziegler and Nichols did not intend that their values should not be used for any other PID configuration [170].

(b) produce values that are intended only as a good starting point for a search for "better" parameter values.

(c) are not suitable for many processes, including those having only dead time or a very large ratio of dead time to first order time constant [170].

(d) can result in a closed-loop system that are sensitive to parameter variations and with damping of $\zeta \cong 0.2$ which is too small for many applications [171] as it results in a high percent overshoot.

(e) produce systems that are sensitive to model uncertainty [120].

(f) give no systematic means to adjust the parameters in order to reduce overshoot on the closed loop step response.

(g) result in systems where the control signal is high which may lead to actuator saturation.

D.2.2 Ziegler-Nichols method based on process reaction curves

This method consists placing the process in an open loop without a controller and following these steps [5]:

1) obtain the process step response

2) draw the steepest tangent to the response

3) estimate the tangent's slope R and the lag time L - the intercept of the tangent with the time axis

4) calculate k_p, k_i and k_d from Table D.1 to get $C(s) = k(1 + T_d s + 1/T_i s)$

Controller	k	T_i	T_d
P	$1/RL$	---	---
PI	$0.9/RL$	$3.3L$	---
PID	$1.2/RL$	$2L$	$L/2$

Table D.1 Ziegler-Nichols Tuning rules [50] for P, PI, and PID controllers based on the process reaction curve of the process.

In using this method, only one test is required; the system is not brought close to instability during the test. However, it is suitable only for processes with monotonic step response. Also, it can be difficult to estimate the parameters accurately from the process reaction curve.

D.2.3 Ziegler-Nichols method based on ultimate sensitivity

This method consists placing the process in a closed loop with the controller and following these steps [14]:

1) leave k_i and k_d constant, vary k_p until the system is marginally stable and behave like a harmonic oscillator with period T. At this point $k_p = S_u$, the "ultimate sensitivity".

2) calculate k_p, k_i and k_d from Table D.2.

Controller	k_p	k_i	k_d
P	$S_u/2$	---	---
PI	$0.45S_u$	$0.83T$	---
PID	$0.6S_u \le k_p \le S_u$	$T/2$	$T/8$

Table D.2 Ziegler-Nichols tuning rules [50] for P, PI, and PID controllers based on ultimate sensitivity.

The following point is important to note in the context of this thesis. By virtue of the fact that the system is tuned to a marginally stable state and then deliberately brought away from this critical point, this tuning method introduces a "degree of stability" in the root locus sense. So, intrinsic to this tuning method is the introduction of a root locus type stability margin. This might explain the Ziegler Nichols parameters often lead to a response that is superior to, for example, parameter plane methods. This is investigated further in section 4.2.3

In practice, it is not often prudent to drive a process to the limit of stability. Also, the assumption that the process will behave in this fashion excludes commonly encountered processes such as [169]: $G(s)=\frac{k}{(s+a)}$, $G(s)=\frac{k}{(s+a)(s+b)}$, $G(s)=\frac{k}{s^2}$ and $G(s)=\frac{k}{(s-a)}e^{-sL}$., with $a,b,k,L>0$. Finally, system oscillations due to hysteresis or to a saturating element can be mistaken as the stability limit and the incorrect value of S_u found. Usually several trials are required to find S_u.

D.2.4 Minimizing performance integrals

Zhuang and Atherton [83, 118, 172] integral performance integrals to design PID and other controllers. They also describe a method for evaluating the integral $\int_0^\infty [t^r e(t)]^2 dt$ for time delay systems. Dan-Isa and Atherton [119] describe a MATLAB [151] based program they have written to evaluate the integral $\int_0^\infty [t^r e(t)]^2 dt$ and the sum $\sum_{k=0}^{\infty}[k^r e(k)]^2$.

Nishikawa et al. [136] choose PID parameters so as to minimise the exponentially weighted performance integral $J(\beta)=\int_0^\infty [\Delta x(t)e^{\beta t}]^2 dt$ where $\Delta x(t)=error$ and β is chosen to suit he particular process in question.

Åström et al. [120] express load disturbance rejection in terms of the integrated error due to a load disturbance in the form of a unit step at the process input i.e. $\int_0^\infty e(t)dt$.

D.2.5 Specifying Gain Margin and Phase Margin

[22, 52, 53, 55] and [56] have propose alternative tuning methods for PI and PID controllers. These consist of [22, 52, 53, 55] deriving approximate analytic expressions for k and T_i for the first-order lag plus time-delay process $G(s)=\frac{K_p}{1+s\tau}e^{-sL}$ in terms of user specified Gain Margin and Phase Margin values. Alternatively, graphical methods are used by [56] to find k and T_i.

Between 1999 and 2001 Liu and Daly described three methods for tuning PID controllers for a number of industrial systems with time delay. Their first [173] is a frequency domain method that applies a minimax optimization procedure simultaneously to analytic expressions for gain margin, phase margin, cross-over frequency and steady state error. Their second [174] is a time domain tuning approach that uses MATLAB's [151] optimisation algorithms to minimise the integral $\int_0^\infty e^2(t)dt$ or the sum $\sum_{k=0}^{\infty}e^2(k)$. Their third method [175] is a blend of the first two.

D.2.6 Nyquist diagram

Munro's [176] method consists of using Mathematica [177] to create thousands of Nyquist plots for the closed lop system. His method is to grid over one PID parameter and then scan the other two. For each set of three parameters he uses a Nyquist plot to decide whether the closed loop system is stable or not. In this way he can create spaces for all stabilising PID controllers. An operating point may then be chosen within this space.

Munro's method is numerically intensive and does not indicate "a degree stability" for each point so there is no guide as to which point to choose within the space of stabilising controllers.

D.2.7 H_∞ - norm

As traditional H_∞ controller design places no constraint on controller complexity it must be adapted to produce controllers of a desired structure such as a PID controller. One approach is to carry out a brute force optimal search the set of all stabilizing PID controllers for a given process [29]. Computationally efficient methods are still a subject of research and one such computationally efficient method is described by Ho [178]. In terms of CPU time, Ho's search method is ten time more efficient than Datta et al. [29].

Panagopoulos and Åström [57] have shown that traditional method for designing PID controllers [120] are related to designed using H_∞ methods. Specifically, they show that the requirement that the H_∞ be a minimum is equivalent to requirement that Nyquist curve of the loop transfer should lie outside a contour which encloses the critical point. They give an explicit formula for this contour and show that it is bounded internally and externally by circles, which are related to the maximum of the sensitivity function and the complementary sensitivity function. Thus they establish a relation between classical design conditions and H_∞ robust control.

D.2.8 Stable polyhedra in parameter space

Datta et al. [29] use a generalization of the Hermite-Biehler theorem to develop a procedure for plotting sets of all stabilising PID controllers for a given process. These sets are presented as convex polygons in parameter space. Munro [176] and Munro and Solyemez [179] develop an equivalent procedure based on the Nyquist criterion. Ackermann [180] generalizes the previous results and offers a third method for plotting these polygons. He then chooses the centroid of a polygon as an operating point and claims that its distance from the stability boundary is an indication of robustness.

However, this approach is flawed as described, for example, in section 6.3.2, and consideration needs to be given to the degree of stability of the operating point. Unfortunately, this turns the three-dimensional parameter space (k_p, k_i, k_d) into the four-dimensional (k_p, k_i, k_d, σ) space. The robust operating point is then a global maximum of a surface in this four-dimensional space.

Appendix E: Number triangles and performance measures

In 1963 P. C. Parks [130] presented the following polynomials :

$$s+\omega_o$$
$$s^2+\omega_o s+\omega_o^2$$
$$s^3+\omega_o s^2+2\omega_o^2 s+\omega_o^3$$
$$s^4+\omega_o s^3+3\omega_o^2 s^2+2\omega_o^3 s+\omega_o^4$$
$$s^5+\omega_o s^4+4\omega_o^2 s^3+3\omega_o^3 s^2+3\omega_o^4 s+\omega_o^5$$
$$s^6+\omega_o s^5+5\omega_o^2 s^4+4\omega_o^3 s^3+6\omega_o^4 s^2+3\omega_o^5 s+\omega_o^6$$
$$s^7+\omega_o s^6+6\omega_o^2 s^5+5\omega_o^3 s^4+10\omega_o^4 s^3+6\omega_o^5 s^2+4\omega_o^6 s+\omega_o^7$$
$$s^8+\omega_o s^7+7\omega_o^2 s^6+6\omega_o^3 s^5+15\omega_o^4 s^4+10\omega_o^5 s^3+10\omega_o^6 s^2+4\omega_o^7 s+\omega_o^8$$

Figure E.1 Polynomials from [130] that minimise the ISE for "zero steady state step error systems".

If the frequency is normalized these polynomials become:

$$s+1$$
$$s^2+s+1$$
$$s^3+s^2+2s+1$$
$$s^4+s^3+3s^2+2s+1$$
$$s^5+s^4+4s^3+3s^2+3s+1$$
$$s^6+s^5+5s^4+4s^3+6s^2+3s+1$$
$$s^7+s^6+6s^5+5s^4+10s^3+6s^2+4s+1$$
$$s^8+s^7+7s^6+6s^5+15s^4+10s^3+10s^2+4s+1$$

Figure E.2 Polynomials from Figure E.1 with normalized frequency.

Parks synthesizes these polynomials from the Routh array by choosing a column of 1's as the first column, calculating the rest of the Routh array, and then reading off the polynomials from the first two rows. An example of such an array is:

1	4	3
1	3	1
1	2	0
1	1	0
1	0	0
1	0	0

Figure E.3 Routh array that is synthesized by first choosing a column of 1's as the leftmost column and then calculating the rest of the array in the usual way.

Figure E.3 is the Routh Array for the polynomial: $P(s)=s^5+s^4+4s^3+3s^2+3s+1$. These polynomials are optimal in the sense that, for example, the transfer function $W_{yr}=\frac{1}{P(s)}$ minimises the ISE or

$J_0 = \int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = \int_0^\infty e^2(t)dt$ performance integral for a zero steady state step error system. Examples of the zero state unit step responses of these transfer functions are given in Figure E.4.

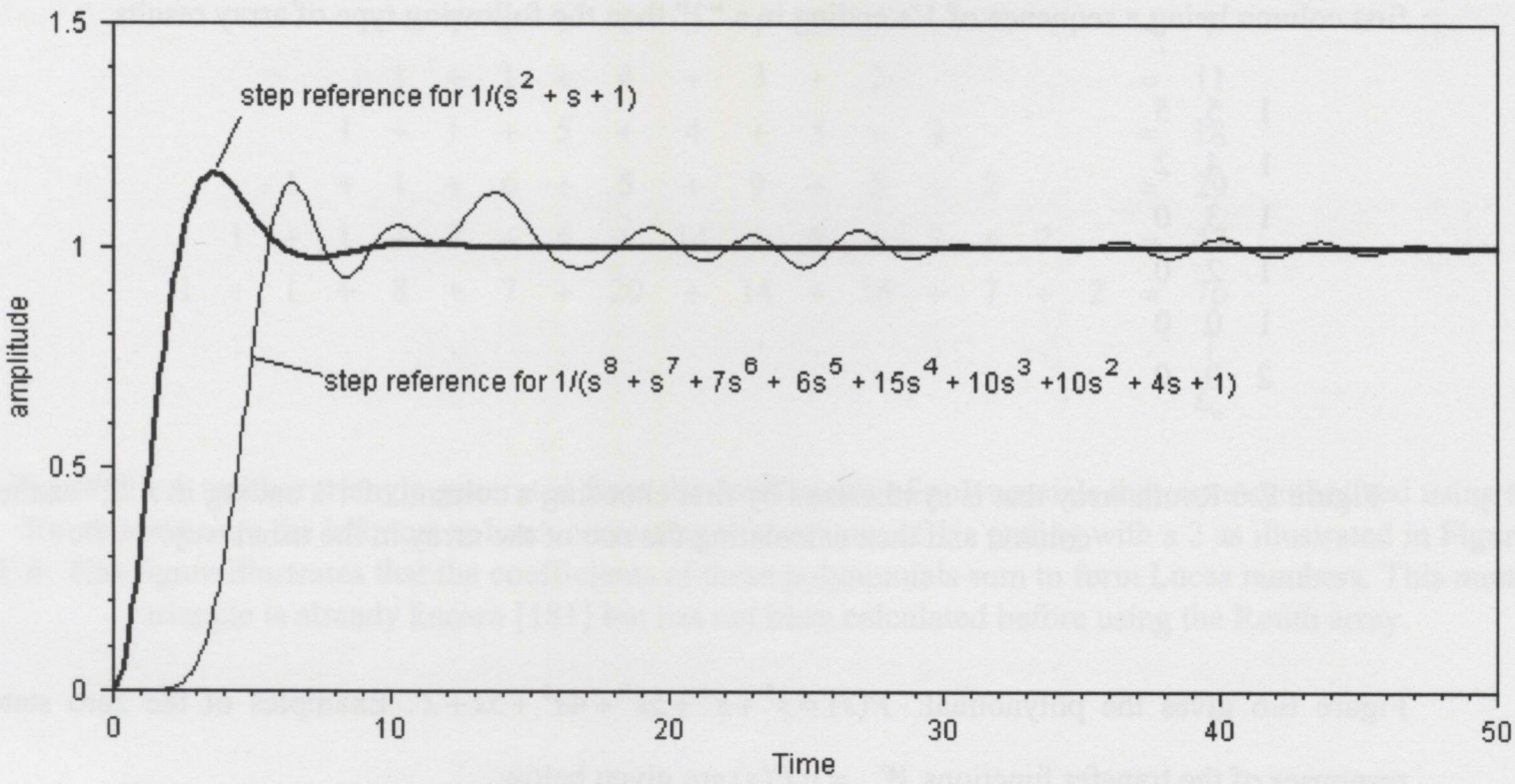


Figure E.4 Zero steady state unit step responses for the polynomials synthesized by Parks [130] using the Routh array as illustrated in Figure E.3.

Figure E.4 shows the zero steady state unit step responses for: $\frac{1}{s^2+s+1}$ and $\frac{1}{s^8+s^7+7s^6+6s^5+15s^4+10s^3+10s^2+4s+1}$. The coefficients of these polynomials form an interesting number triangle [181]:

$$
\begin{array}{ccccccccccccccccc}
 & & & & & & & 1 & & & & & & & & = & 1 \\
 & & & & & & 1 & + & 1 & & & & & & & = & 2 \\
 & & & & & 1 & + & 1 & + & 1 & & & & & & = & 3 \\
 & & & & 1 & + & 1 & + & 2 & + & 1 & & & & & = & 5 \\
 & & & 1 & + & 1 & + & 3 & + & 2 & + & 1 & & & & = & 8 \\
 & & 1 & + & 1 & + & 4 & + & 3 & + & 3 & + & 1 & & & = & 13 \\
 & 1 & + & 1 & + & 5 & + & 4 & + & 6 & + & 3 & + & 1 & & = & 21 \\
1 & + & 1 & + & 6 & + & 5 & + & 10 & + & 6 & + & 4 & + & 1 & = & 34 \\
1 & + & 1 & + & 7 & + & 6 & + & 15 & + & 10 & + & 10 & + & 4 & + & 1 & = & 55 \\
 & . & & . & & . & & \nearrow & & . & & \nearrow & & . & & . & \uparrow \\
 & & & & & & & & & & & & & & & & F_n
\end{array}
$$

Figure E.5 Coefficients of the polynomials in Figure E.2 arranged as a number triangle. This figure illustrates the fact that the coefficients sum to form Fibonacci numbers. This number triangle has been studied elsewhere [181] and it was generated using the Routh array by Parks [130], albeit without the intention of generating Fibonacci numbers.

This number triangle and the following properties have been noted elsewhere [181]:

The sequence "1, 3, 6, 10, 15…." is made of triangular numbers.

The sequence " 1, 4, 10 …" is made of tetrahedral numbers.

Summing across the rows gives F_n - the Fibonacci numbers.

The following method of construction of another interesting number triangle has not, to my knowledge, been noted elsewhere. If the Routh array is used again in synthetic mode but this time with the first column being a sequence of 1's ending in a "2" then the following type of array results:

1	5	5
1	4	2
1	3	0
1	2	0
1	0	0
2	0	0

Figure E.6 Routh array that is synthesized by first choosing a column of 1's ending in a "2" as the leftmost column and then calculating the rest of the array in the usual way.

Figure E.6 gives the polynomial: $P(s) = s^5 + s^4 + 5s^3 + 4s^2 + 5s + 2$. Examples of the zero state unit step responses of the transfer functions $W_{yr} = 1/P(s)$ are given below.

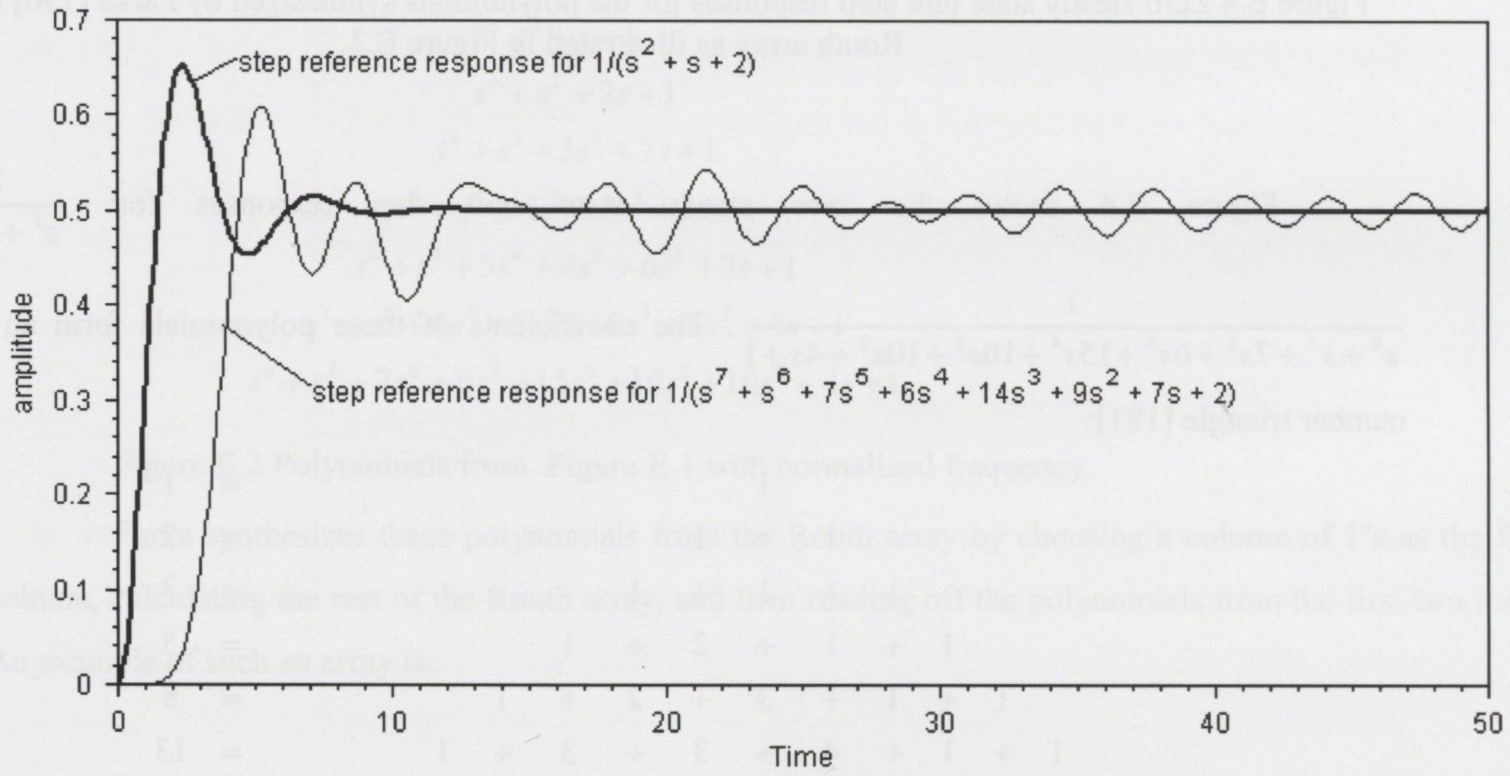


Figure E.7 Zero steady state unit step responses for the polynomials synthesized using the Routh array as illustrated in Figure E.6.

Figure E.7 shows the step reference responses for the transfer functions: $\frac{1}{s^2 + s + 2}$ and $\frac{1}{s^7 + s^6 + 7s^5 + 6s^4 + 14s^3 + 9s^2 + 7s + 2}$. These polynomials were synthesized using the Routh array with the right column consisting of a series of 1's ending with a 2 as illustrated in Figure E.6. These responses settle down a little more slowly than the Parks polynomials in Figure E.4.

If the process illustrated in Figure E.6 is repeated and the coefficients of the resulting polynomials arranged in a triangle then you get the number triangle in Figure E.8.

																1														=	1
														1	+	2														=	3
												1	+	1	+	2														=	4
								1	+	1	+	3	+	2																=	7
						1	+	1	+	4	+	3	+	2																=	11
				1	+	1	+	5	+	4	+	5	+	2																=	18
			1	+	1	+	6	+	5	+	9	+	5	+	2															=	29
		1	+	1	+	7	+	6	+	14	+	9	+	7	+	2														=	47
	1	+	1	+	8	+	7	+	20	+	14	+	16	+	7	+	2													=	76
.		.		.		.		.		.		.		.		.														↑	
																														L_n	

Figure E.8 A number triangle generated from the coefficients of polynomials that were synthesized using the Routh array with the leftmost column consisting of a column of 1's ending with a 2 as illustrated in Figure E.6. This figure illustrates that the coefficients of these polynomials sum to form Lucas numbers. This number triangle is already known [181] but has not been calculated before using the Routh array.

The number triangle in Figure E.8 is already known [181] but has not been calculated before using the Routh array. The sum across the rows of the number triangle in Figure E.8 gives L_n - the Lucas numbers. This suggests that there may be a relationship between Cauchy Indices, Lucas numbers, and Fibonacci numbers.

The relevance of these facts to this thesis is that sometimes the systems modeled by polynomials synthesized in this way behave very well and, in the case described by Parks [130], minimise the ISE index. The polynomials synthesized by dePaor [182] seem to minimise a very stringent performance measure. dePaor [182] has described the synthesis of polynomials using the Routh array with the first column given by binomial coefficients. An example of such an array is:

1	$\frac{63}{5}$	$\frac{31}{5}$
5	13	1
10	6	0
10	1	0
5	0	0
1	0	0

Figure E.9 Routh array that is synthesized by first choosing a left hand column consisting of binomial coefficients [182].

The synthesized Routh array in Figure E.9 gives the Pascal-Routh [182] polynomial:

$$P(s) = s^5 + 5s^4 + \tfrac{63}{5}s^3 + 13s^2 + \tfrac{31}{5}s + 1$$

The zero state unit step response of these Pascal-Routh polynomials is excellent and their behavior is far superior to that found in Parks polynomials. Examples of the zero state unit step responses of these transfer functions $W_{yr} = 1/P(s)$ are given below.

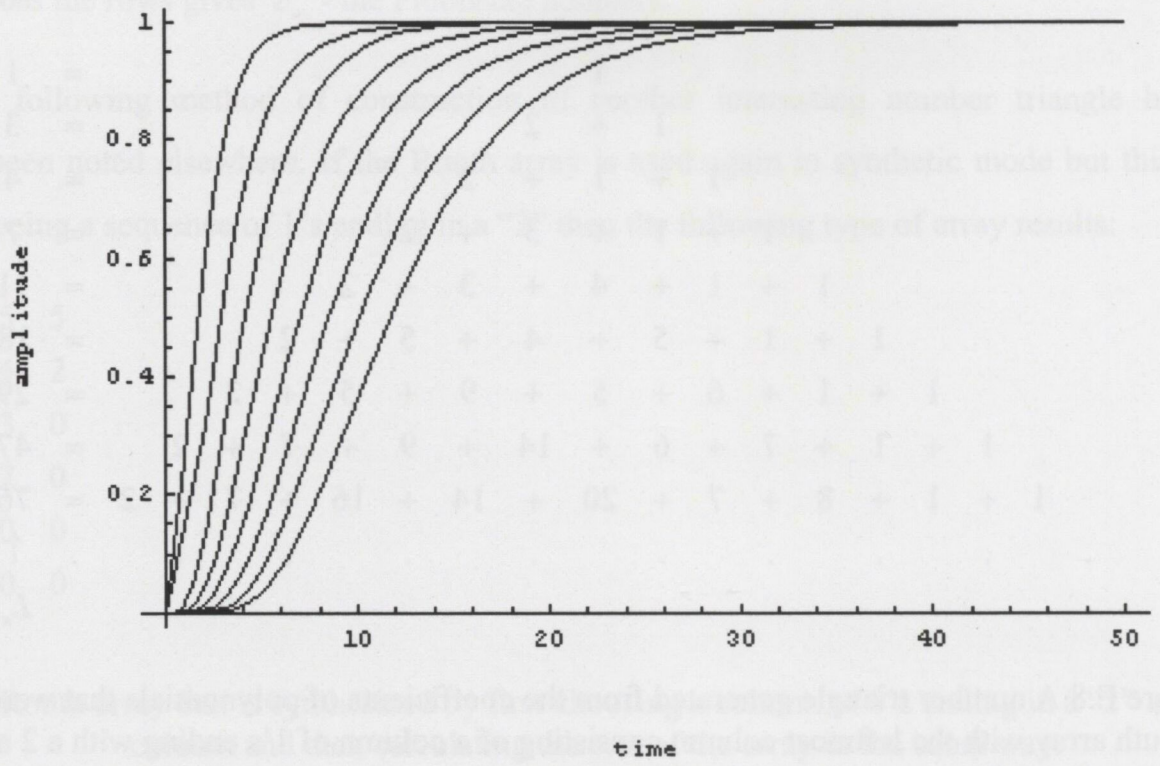


Figure E.10 Zero steady state unit step responses for the polynomials synthesized by dePaor [182] using the Routh Array as illustrated in Figure E.9.

These Pascal-Routh polynomials, synthesized by dePaor [182], seem to minimise a very stringent performance measure. Similar behavior was noticed when the first column of the Routh array consisted of a pattern of numbers that begins with a 1, become larger in the middle, and then becomes 1 again e.g. 1,5,5,5,5,5,1. It would be interesting to investigate this in detail and to discover which performance measure they satisfy.

This could lead to novel method for control system synthesis. A performance measure is selected and this could correspond to a first column for the Routh array. We can then synthesize a transfer function that minimises the corresponding performance measure.

Appendix F: Highlights of the history of Stability Theory

In 1867 Thomson and Tait [183] stated: "There is scarcely a question in dynamics more important for Natural Philosophy than the stability or instability of motion". Some authors have given part of this story [184]. A list of some of the high points in the development of stability theory is given below.

Newton (1686) - stability of the sun - moon - earth system and of the solar system

Laplace (1783) - stability of the solar system

Sturm (1829) – a method for finding the number of real zeros of a polynomial between given limits

Cauchy (1831) – Cauchy Index method for counting the number of zeros (with positive real parts) of a polynomial in a given domain

Sturm (1836) – introduces the Sturm Sequence for calculating the Cauchy Index

Cauchy (1837) – complete solution to the problem of counting the number of zeros of a polynomial in a given domain

Airy (1840) - velocity control of a telescope

Hermite (1854) - criterion for deciding on the signs of the real parts of the roots of a polynomial

Maxwell (1857) - stability of the rings of Saturn

Maxwell (1868) - stability of governors

Kronecker (1869) – formula for the number of zeros of a polynomial in a domain

Vyshnegradskii (1876) - stability of governors

Routh (1877) - algorithm for deciding on the signs of the real parts of the roots of a polynomial - this is a tabular scheme for calculating the number of changes of sign in a Sturm sequence.

Poincaré (1881) - stability of the solar system; general problem of the stability of motion

Lyapunov (1892) - general problem of the stability of motion

Hurwitz (1895) - criterion for deciding on the signs of the real parts of the roots of a polynomial

Synge (1924) – the stability of motion and the geometry of geodesics in Riemannian space [185]

Nyquist (1932) - stability criterion for feedback amplifiers

Harris (1942) - extension of the Nyquist criterion to feedback control systems

Hall (1943) - extension of the Nyquist criterion to feedback control systems

Bodé (1945) - frequency response method

Evans (1948, 1950) - root locus method

Bibliography

[1] H. Leipholz, *Stability Theory*, 2nd ed. Stuttgart: John Wiley and Sons, 1987.

[2] Q.-G. Wang, T.-H. Lee, H.-W. Fung, Q. Bi, and Y. Zhang, "PID Tuning for Improved Performance," *IEEE Transactions on Automatic Control*, vol. 7, pp. 457-465, 1999.

[3] G. K. I. Mann, B.-G. Hu, and R. G. Gosine, "Time-domain based design and analysis of new PID tuning rules," *IEE Proceedings on Control Theory Applications*, vol. 148, pp. 251-261, 2001.

[4] J. Normey-Rico, I. Alcala, Gomez-Ortega, and E. F. Camacho, "Mobile robot path tracking using a robust PID controller," *Control Engineering Practice*, vol. 9, pp. 1209-1214, 2001.

[5] J. C. Basilio and S. R. Matos, "Design of PI and PID Controllers with Transient Performance Specifications," *IEEE Transactions on Automatic Control*, vol. 45, pp. 364-370, 2002.

[6] A. M. de Paor, "A modified Smith predictor and controller for unstable processes with time delay," *International Journal of Control*, vol. 41, pp. 1025-1036, 1985.

[7] A. M. de Paor and M. O'Malley, "Controllers of Ziegler-Nichols type for Unstable Process with time Delay," *Int. J. Control*, vol. 49, pp. 1273-1284, 1989.

[8] A. M. de Paor, T. E. Ward, and J. D. Delaney, "On The Control of Balance During Quiet Standing," presented at Measurement '99, Slovakia, 1999.

[9] A. M. de Paor, "Aerodynamic design of optimum wind turbines," *Applied Energy*, vol. 12, pp. 221-228, 1982.

[10] H. M. Power, "Analytical root locus design study of servomechanisms with inertial damping," *International Journal of Electrical Engineering Education*, vol. 9, pp. 411-424, 1971.

[11] H. M. Power, "Analytical design of servomechanisms with spring-coupled inertial dampers," *International Journal of Control*, vol. 1971, pp. 497-511, 1971.

[12] R. C. Dorf and R. H. Bishop, "Design Using Performance Indices," in *The Control Handbook*, W. S. Levine, Ed.: CRC Press, IEEE Press, 1996, pp. 169-173.

[13] R. C. Dorf and R. H. Bishop, *Modern Control Systems - 8th edition*, 8 ed. Melno Park, CA: Addison-Wesley, 1998.

[14] H. M. Power and R. J. Simpson, *Introduction to Dynamics and Control*. Maidenhead, Berkshire: McGraw - Hill, 1978.

[15] L. P. Grayson, "The Status of Synthesis Using Lyapunov's Method," *Automatica*, vol. 3, pp. 91-121, 1965.

[16] J. E. Gibson, *Nonlinear Automatic Control*: McGraw-Hill Book Company, 1963.

[17] R. E. Kalman and J. E. Bertram, "Control System Analysis and Design Via the "Second Method" of Lyapunov - Part 1: Continuous - Time Systems," *Transactions of the ASME - Series D - Journal of Basic Engineering*, vol. 82, pp. 371-393, 1960.

[18] C. Loehle, "Global Optimization 5.2," Loehle Enterprises, 1258 Windemere Ave., Naperville, IL 60564, USA, 2006.

[19] S. Barnett, *Polynomials and Linear Control Systems*. New York: Marcel Dekker Inc., 1983.

[20] A. M. de Paor, "Concepts of optimum stability for linear feedback systems," *International Journal of Electrical Engineering Education*, vol. 36, pp. 46-64, 1999.

[21] H. M. Power, "Design of Inertially Damped Sevomechanisms with Maximum Phase Margins," *IEEE Transactions on Automatic Control*, pp. 568, 1969.

[22] W. K. Ho, K. W. Lim, and W. Xu, "Optimal Gain and Phase Margin Tuning for PID Controllers," *Automatica*, vol. 34, pp. 1009-1014, 1998.

[23] W. Yan and B. D. O. Anderson, "The Simultaneous Optimization Problem for Sensitivity and Gain Margin," *IEEE Transactions on Automatic Control*, vol. 35, pp. 558-563, 1990.

[24] L. H. Keel and S. P. Bhattacharyya, "Robust, Fragile, or Optimal?," *IEEE Transactions on Automatic Control*, vol. 42, pp. 1098-1105, 1997.

[25] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. New Jersey: Prentice-Hall Inc., 1996.

[26] A. M. de Paor and B. Cogan, "Windspeed-dependant underdamping and its cure in the self-excited, series-wound, aerogenerator," *Applied Energy*, vol. 42, pp. 253-267, 1992.

[27] J. G. Truxal, *Automatic Feedback Control Synthesis*: McGrath-Hill, 1955.

[28] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*: Imperial College Press, 2003.

[29] A. Datta, M.-T. Ho, and S. P. Bhattacharyya, *Structure and Synthesis of PID Controllers*: Springer, 2000.

[30] A. G. J. MacFarlane, "The Calculation of Functionals of the Time and Frequency Response of a Linear Constant Coefficient Dynamical System.," *Quart. Journ. Mech. and Applied Math. - Part 2*, vol. XVI, pp. 259-271, 1963.

[31] A. G. J. MacFarlane, "Functional Matrix Theory for the General Linear Electrical Network - Part 5," *Proceedings of the Institution Of Electrical Engineers*, vol. 116, pp. 1745-1747, 1969.

[32] C. Chen and L. Shieh, "A note on expanding $PA+A^TP=-Q$," *IEEE Transactions on automatic Control*, vol. 13, pp. 122-123, 1968.

[33] S. Barnett, "Evaluation of discrete linear system performance indices using Kronecker products.," *International Journal of Control*, vol. 20, pp. 849-855, 1974.

[34] E. I. Jury and S. Gutman, "On the Time-Weighted Quadratic Sum of Linear Discrete Systems," *Transactions of the ASME - Journal of Dynamic Systems, Measurement, and Control*, pp. 441-444, 1975.

[35] M. Mansour, E. I. Jury, and L. F. Chaparro, "Estimation of the Margin of Stability for linear continuous and discrete systems," *International Journal of Control*, vol. 30, pp. 49-69, 1979.

[36] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*: Cambridge University Press, 1991.

[37] A. M. Lyapunov, *The General Problem of the Stability of Motion*: Taylor and Francis, 1992.

[38] H. M. Power, "Lyapunov matrix equations with positive semidefinite input matrices," *Electronics Letters*, vol. 4, pp. 325-326, 1968.

[39] E. Meijering, "A Chronology of Interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, pp. 319-342, 2002.

[40] M. R. Spiegel, *Calculus of Finite Differences and Difference Equations*. New York: McGraw Hill, 1971.

[41] S. Fukata, Tamura, H.I., "The evaluation of time weighted quadratic performance indices for discrete and sample- data linear systems," *International Journal of Control*, vol. 39, pp. 135-142, 1984.

[42] E. I. Jury, *Inners and Stability of Dynamic Systems*: Robert E. Krieger Publishing Company, 1982.

[43] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*. New York: Dover Publications, Inc.,, 1965.

[44] G. J. Silva, A. Datta, and S. P. Bhattacharyya, *PID Controllers for Time-Delay Systems*. Boston: Birkhauser, 2005.

[45] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. New York: Macmillan Publishing Company, 1992.

[46] S. M. Shinners, *Modern Control System Theory and Design*. New York: John Wiley and Sons Inc., 1992.

[47] B. Cogan and A. M. dePaor, "Optimum stability and minimum complexity as desiderata in feedback control system design," presented at IFAC Conference on Control Systems Design, Bratislava, Slovakia, 2000.

[48] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed. Melno Park, CA: Addison Wesley, 1994.

[49] S. Wolfram, "Mathematica 5.1," Wolfram Research, Inc., 100 Trade Canter Drive, Champaign, IL 61820, USA, 2005.

[50] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Trans. ASME*, vol. 64, pp. 759-768, 1942.

[51] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang, "Performance and Gain and Phase Margins of Well-Known PID Tuning Formulas," *IEEE Trans. Control System Technology*, vol. 4, pp. 473-477, 1996.

[52] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID Controllers Based on Gain and Phase Margin Specifications," *Automatica*, vol. 31, pp. 497-502, 1995.

[53] W. K. Ho and W. Xu, "PID Tuning for Unstable Processes based on Gain and Phase Margin Specifications," *IEE Proceedings - Control Theory Appl.*, vol. 145, pp. 392-396, 1998.

[54] Z. Shafiei and A. T. Shenton, "Tuning of PID-Type Controllers for Stable and Unstable Systems with Time Delay," *Automatica*, vol. 30, pp. 1609-1615, 1994.

[55] W. K. Ho, C. C. Hang, and J. H. Zhou, "Performance and Gain and Phase Margins of Well-Known PI Tuning Formulas," *IEEE Transactions on Control Systems Technology*, vol. 3, pp. 245-248, 1995.

[56] H.-W. Fung, Q.-G. Wang, and T.-H. Lee, "PI Tuning in Terms of Gain and Phase Margins," *Automatica*, vol. 34, pp. 1145-1149, 1998.

[57] H. Panagopoulos and K. J. Astrom, "PID control design and H-infinity loop shaping," *International Journal of Robust and Nonlinear Control*, vol. 10, pp. 1249-1261, 2000.

[58] K. J. Åström and T. Hägglund, "Benchmark Systems for PID Control," presented at Digital Contol: Past, Present and Future of PID Control (PID '00), Terrassa, Spain, 2000.

[59] K. H. Ang, G. Chong, and Y. Li, "PID Control System Analysis, Design, and Technology," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 559-576, 2005.

[60] Y. Li, W. Feng, K. C. Tan, X. K. Zhu, X. Guan, and K. H. Ang, "PIDeasy and automated generation of optimal PID controllers," presented at 3rd Asia-Pacific Conf. Control and Measurement, Dunhuang, P. R. China, 1998.

[61] R. S. Stefani, B. Shahian, C. J. Savant, and G. H. Hostetter, *Design of Feedback Control Systems*, 4 ed: Oxford University Press, 2002.

[62] K. L. Chein, J. A. Hrones, and J. B. Reswick, "On the Automatic Control of Generalized Passive Systems," *Trans ASME*, vol. 74, pp. 175-185, 1952.

[63] A. Karimi, D. Garcia, and R. Longchamp, "PID Controller Tuning Using Bode's Integrals," *IEEE Transactions on Automatic Control*, vol. 11, pp. 812-821, 2003.

[64] V. J. Rao and D. S. Bernstein, "Naive Control of the Double Integrator," *IEEE Control Systems Magazine*, pp. 86-97, 2001.

[65] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems - Theory and Design*, 2nd ed: Prentice-Hall International, Inc., 1990.

[66] A. M. de Paor, "Control of the double integrator process: a cautionary tale in discreteization," *International Journal of Electrical Engineering Education*, vol. 32, pp. 307-318, 1995.

[67] A. M. de Paor, "The root locus method: famous curves, control designs and non-control applications," *International Journal of Electrical Engineering Education*, vol. 37, pp. 344-356, 2000.

[68] K. Gu, V. Kharitonov, and J. Chen, *Stability of Time -Delay Systems*. Boston: Birkhauser, 2003.

[69] D. R. Hartree, A. Porter, A. Callender, and A. B. Stevenson, "Time lag in a control system (II)," *Proc. Roy. Soc. London A, 161*, pp. 460-476, 1937.

[70] A. Callender, D. R. Hartree, and A. Porter, "Time-Lag in a Control System," *Philosophical Transactions of the Royal Society of London - Series A - Mathematical and Physical Sciences*, vol. 235, pp. 415-444, 1936.

[71] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 2nd ed. London: Imperial College Press, 2006.

[72] K. J. Åström and T. Hagglund, *Advanced PID Control*: Instrument Society of America, 2006.

[73] V. B. Kolmanovskii and V. R. Nosov, *Stability of Functional Differential Equations*. London: Academic Press, 1986.

[74] K. Ogata, *Modern Control Engineering*. New Jersey: Prentice Hall, 2002.

[75] W. J. Palm III, *Control System Engineering*: John Wiley & Sons, 1986.

[76] M. J. Wade and M. A. Johnson, "Towards automatic real -time controller tuning and robustness," presented at 38th IAS Annual Meeting and Industry Applications Conference, Industry Control Centre, Strathclyde University, Glasgow, UK, 2003.

[77] S. Wolfram, "Mathematica - Control System Professional 2." Champaign, Illinois: Wolfram Research, 2003.

[78] P. M. Thompson, "Program CC - Tutorial and User's Guide," 4th ed. Hawthorn, California: Systems Technology, Inc., 1989.

[79] G. J. Silva, A. Datta, and S. P. Bhattacharyya, "PI stabilization of first-order systems with time delay," *Automatica*, vol. 37, pp. 2025-2031, 2001.

[80] B. Cogan, "Use of the Analytic Method and Computer Algebra to plot Root Loci," *International Journal of Electrical Engineering Education*, vol. 35, pp. 350-356, 1998.

[81] H. Shinozaki and T. Mori, "Robust stability analysis of linear time-delay systems by Lambert W function: Some extreme point results," *Automatica*, vol. 42, pp. 1791-1799, 2006.

[82] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W Fuction," *Advances in Computational Mathematics*, vol. 5, pp. 329-359, 1996.

[83] M. Zhuang and D. P. Atherton, "Tuning of PID controllers with integral performance criteria," presented at IEE Control '91, 1991.

[84] J. E. Marshall, H. Gorecki, K. Walton, and A. Korytowski, *Time-delay systems - stability and performance criteria with applications*: Ellis Horwood, 1992.

[85] S. Boyd and C. Barratt, "Linear Controller Design - Limits of Performance," Prentice Hall, 1991.

[86] J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design Conventional and Modern*, 3rd ed. New York: Mc Graw Hill, 1988.

[87] J. M. Furman and S. P. Cass, *Balance Disorders: A Case-Study Approach*. Philadelphia: F. A. Davis, 1996.

[88] P. Loughlin and M. Redfern, "Special Issue on Analysis and Modeling of Human Postural Control," *IEEE Engineering in Medicine and Biology Magazine*, 2003.

[89] H. Hemani, F. C. Weimer, C. S. Robinson, C. W. Stockwell, and V. S. Cvetovic, "Biped stability considerations with vestibular models," *IEEE Transactions on Automatic Control*, vol. 23, pp. 1074-1079, 1978.

[90] Y. I. Pal'tsev and R. V. Aggashyan, "Role of feedback for ensuring the stability of the vertical posture in man," *Biofizika*, vol. 19, pp. 954-960, 1974.

[91] J. D. Delaney, "The human balance control system: a therapeutic aid and mathematical control model," in *Electronic and Electrical Engineering*. Dublin: National University of Ireland, 1998.

[92] S. Wolfram, "Mathematica - Control System Professional," 1 ed. Champaign, Illinois, 1996.

[93] J. Ackermann, *Robust Control - Systems with Uncertain Physical Parameters*. New York: Springer-Verlag, 1994.

[94] C. Soh, C. Berger, and K. Dabke, "On the stability properties of polynomials with perturbed coefficients," *IEEE Transactions on Automatic Control*, vol. 30, pp. 1033-1036, 1985.

[95] S. P. Bhattacharyya, H. Chapellat, and L. H. Keel, *Robust Control - The Parametric Approach*: Prentice Hall, 1995.

[96] R. Jeltsch, "An optimal fraction free Routh array," *International Journal of Control*, vol. 30, pp. 653-660, 1979.

[97] R. Jeltsch, "Stability of Time Discretization, Hurwitz Determinants and Order Stars," presented at Stability Theory - Hurwitz Centenary Conference, Centro Stefano Franscini, Ascona, 1995.

[98] G. Fichera, "Alcune osservazioni sulle condizioni di stabilita per le equazioni algebraiche a coefficienti reali.," *Bolletino della Unione Mathematica Italia, Ser. III*, vol. 2, pp. 103-109, 1947.

[99] A. M. de Paor, "A model for n-let behaviour in functional electrical stimulation of skeletal muscle," *Journal of Electrical Engineering*, vol. 51, pp. 285-289, 2000.

[100] S. G. Loo, "Method of Evaluating Time-Weighted Performance Integrals," *Electronics Letters*, vol. 3, pp. 523 - 524, 1967.

[101] K. Ramar and B. Ramaswami, "A new method of evaluating quadratic and time-weighted quadrtic performance indices," *International Journal of Control*, vol. 13, pp. 83-96, 1971.

[102] H. M. Power, "Comments on Method of Evaluating Time-Weighted Performance Integrals," *Electronics Letters*, vol. 4, pp. 3,4, 1968.

[103] J. Williams, *Laplace Transforms*, vol. 10. London: George Allen & Unwin Ltd., 1973.

[104] I. G. Malkin, *Theory of Stability of Motion*: Department of Commerce, USA, 1958.

[105] S. Barnett and C. Storey, *Matrix Methods in Stability Theory*. London: Thomas Nelson and Sons Ltd., 1970.

[106] J. P. LaSalle, *The Stability and Control of Discrete Processes*, vol. 62: Springer Verlag, 1986.

[107] W. D. Brogan, *Modern Control Theory - 3rd edition*: Prentice Hall, 1991.

[108] E. I. Jury, *Theory and application of the z - transform method*: John Wiley and Sons, Inc., 1964.

[109] P. M. Mäkilä, "A note on the Laplace transform method for initial value problems," *International Journal of Control*, vol. 79, pp. 36-41, 2006.

[110] F. Csaki and J. Lehoczky, "Comments on "A Remark on Routh's Array"," *IEEE Transactions on Automatic Control*, pp. 462-463, 1967.

[111] J. Lehoczky, "The determination of simple quadratic integrals by Routh coefficients," *Perodica Polytechnica, Elec. Eng.*, vol. 10, pp. 153-166, 1966.

[112] A. G. J. MacFarlane, "Functional Matrix Theory for the General Linear Electrical Network - Part 2," *Proceedings of the Institution Of Electrical Engineers*, vol. 112, pp. 763-770, 1965.

[113] J. L. Peczkowski, "Synthesis by Liapunov's Direct Method," *Transactions of the ASME - Journal of Dynamic Systems, Measurement, and Control*, vol. September, pp. 151-156, 1971.

[114] H. M. Power, "Formal Models for Biological Oscillators Derived by Using Liapunov Theory in Synthetic Mode," *Proceedings of the Royal Irish Academy*, vol. 78 Sect. A, pp. 235-245, 1978.

[115] P. Albertos and A. Sala, *Multivariable Control Systems - An Engineering Approach*. London: Springer, 2004.

[116] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Englewood Cliffs, Hemel Hempstead: Prentice-Hall Inc., 1971.

[117] A. Visioli, "Optimal tuning of PID controllers for integral and unstable processes," *IEE Proceedings on Control Theory Applications*, vol. 148, pp. 180-184, 2001.

[118] M. Zhuang and D. P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proceedings-D*, vol. 140, pp. 216-224, 1993.

[119] A. Dan-Isa and D. P. Atherton, "Time-domain method for the design of optimal linear controllers," *IEE Proceedings on Control Theory Applications*, vol. 144, pp. 287-292, 1997.

[120] K. J. Åström, H. Panagopoulos, and T. Hägglund, "Design of PI Controllers based on Non-Convex Optimization," *Automatica*, vol. 34, pp. 585-601, 1998.

[121] N. Hohenbichler and J. Ackermann, "Synthesis of Robust PID Controllers for Time Delay Systems," presented at European Control Conference '03, Cambridge, UK, 2003.

[122] A. C. Hall, *The Analysis and Synthesis of Servomechanisms*: Technology Press, Massachusetts Institute of Technology, 1943.

[123] P. T. Nims, "Some design criteria for automatic controls," *Trans. AIEE*, vol. 70, pp. 606-611, 1951.

[124] J. H. Westcott, "The minimum-moment-of-error-squared criterion: a new performance criterion for servos," *Proc. IEE*, vol. 101, pp. 471-480, 1954.

[125] D. Graham and R. C. Lathrop, "The Synthesis of "Optimum" Transient Response: Criteria and Standard Forms," *American Institutuin of Electrical Engineers*, vol. 72, pp. 273-288, 1953.

[126] W. C. Schultz and V. C. Rideout, "Control System Performance Measures: Past, Present, and Future," *IRE Transactions on Automatic Control*, vol. AC-6, pp. 22-35, 1961.

[127] S. Barnett, "Comment on "A note on expanding PA+A^TP=-Q"," *IEEE Transactions on automatic Control*, vol. 13, pp. 756, 1968.

[128] S. G. Margolis, "Comments on "A note on expanding PA+A^TP=-Q"," *IEEE Transactions on automatic Control*, vol. 13, pp. 596-597, 1968.

[129] Z. Gajic and M. Qureshi, *Lyapunov Matrix Equation in System Stability and Control*, vol. 195, first ed. London: Academic Press, 1995.

[130] P. C. Parks, "A New Proof of the Hurwitz Stability Criterion by the Second Method of Lyapunov, with Applications to "Optimum" Transfer functions," presented at Proceedings of the Joint Automatic Control Conference, 1963.

[131] J. E. Diamessis, "A New Method for Calculating System Performance Measures," *Proceedings of the Institution Of Electrical Engineers*, vol. 52, pp. 1240, 1964.

[132] J. E. Diamessis, "Time Domain Calculation of System Performance Criteria," *IEEE Transactions on Education*, vol. E-10, pp. 51 - 52, 1966.

[133] H. M. Power, "The Companion Matrix and Liapunov Functions for Linear Multivariable Time-Invarient Systems," *Journal of The Franklin Institute*, vol. 283, pp. 214-234, 1967.

[134] R. Bellman, *Introduction to Matrix Analysis*. New York: McGrath-Hill, 1970.

[135] J. W. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Transactions on Circuits and Systems*, vol. 25, pp. 772-781, 1978.

[136] Y. Nishikawa, N. Sannomiya, T. Ohta, and H. Tanaka, "A Method for Auto-tuning of PID Control Parameters," *Automatica*, vol. 20, pp. 321-332, 1984.

[137] C. Hwang, "Weighted Time-Moments of Discrete-Time Systems," *IEEE Transactions on Automatic Control*, vol. 32, pp. 249-251, 1987.

[138] X. S. Cai and Y. D. Wang, "Optimal Control for the PWM Switching-Mode Power Supply," 1994.

[139] F. T. Man, "Evaluation of Time-Weighted Quadratic Performance Indices for Linear Discrete Time Systems," *IEEE Transactions on Automatic Control*, vol. 15, pp. 496-497., 1970.

[140] F. M. Al-Sunni and F. L. Lewis, "Optimal Output Tracker Using a Time-Weighted Performance Index," *Optimal Control Applications and Methods*, vol. 19, pp. 55-62, 1998.

[141] K. Ogata, *Discrete -Time Control Systems*: Prentice-Hall, Inc., 1987.

[142] J. Ackermann, *Robust Control - The Parameter Space Approach*, 2nd ed: Springer, 2003.

[143] G. Stein, "Does "H" stand for "Hype"?," presented at IEE Computing and Control Division Colloquium on Controller Design and Applications Using H-infinity, Savoy Place, London, 1993.

[144] J. W. Helton, "Worst case analysis in the frequency domain: the H-infinity approach to control," *IEEE Transactions on Automatic Control*, vol. 30, pp. 1154-1170, 1985.

[145] M. Green and D. J. N. Limebeer, *Linear Robust Control*. New Jersey: Prentice-Hall, 1995.

[146] L. Lublin, S. Grocott, and M. Athans, "H2 (LQG) and H-infinity Control," in *The Control Handbook*, W. S. Levine, Ed.: CRC Press, 1996, pp. 651-661.

[147] R. S. Burns, *Advanced Control Theory*: Butterworth Heinemann, 2001.

[148] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-Space Solutions to Standard H2 and H-infinity Control Problems," *IEEE Transactions on Automatic Control*, vol. 34, pp. 831-847, 1989.

[149] W. S. Levine, *The Control Handbook*, 1996.

[150] S. Boyd, C. Barratt, and S. Norman, "Linear controller design: limits of performance via convex optimization," *Proceedings of the IEEE*, vol. 78, pp. 529-574, 1990.

[151] MathWorks, "MatLab," 13 ed: The Math Works, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, 2003.

[152] J. W. Helton and O. Merino, *Classical control using H-infinity methods*. Philadelphia: SIAM - Society for Industrial and Applied Mathematics, 1998.

[153] C. Gomez, "Engineering and Scientific Computing with Scilab," Birkhauser, 1999.

[154] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the H-infinity norm of a transfer matrix and related problems," *Mathematics of Controls, Signals, and Systems*, vol. 2, pp. 207-219, 1989.

[155] S. Boyd, V. Balakrishnan, and P. Kabamba, "On computing the H-infinity norm of a transfer matrix," presented at American Control Conference, Atlanta, Georgia, 1988.

[156] W. S. Levine, "The Control Handbook," CRC Press, 1996.

[157] M. A. Johnson and M. J. Grimble, "H infinity control design and quantitative feedback theory," presented at IEE Colloquium on Advances in the Application of Robust Controllers, 1993.

[158] D. U. Campos-Delgado and K. Zhou, "A parametric optimization approach to H-infinity and H-2 strong stabilization," *Automatica*, vol. 39, pp. 1205-1211, 2003.

[159] M. T. Ho and C.-Y. Lin, "PID Controller Design for Robust Performance," *IEEE Transactions on Automatic Control*, vol. 48, pp. 1404-1409, 2003.

[160] M.-T. Ho, A. Datta, and S. P. Bhattacharyya, "Control System Design Using Low Order Controllers: Constant Gain, PI and PID," presented at American Control Conference, Albuquerque, New Mexico, 1997.

[161] F. Paganini, "Frequency domain conditions for robust H-2 performance," *IEEE Transactions on Automatic Control*, vol. 44, pp. 38-49, 1999.

[162] V. J. Van Doren, "Loop Tuning Fundamentals," in *Control Engineering Europe*, vol. 4, 2003, pp. 36-38.

[163] S. Bennett, "The Past of PID Controllers," presented at Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 2000.

[164] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Transactions of the ASME*, vol. 220, pp. 220-222, 1993.

[165] M. Lelic and Z. Gajic, "A Reference Guide to PID Controllers in the Nineties," presented at Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 2000.

[166] M. A. Johnson and M. H. Moradi, *PID Control: New Identification and Design Methods*: Springer Verlag, 2005.

[167] K. J. Åström and T. Hägglund, "The Future of PID Control," *Control Engineering Practice*, vol. 9, pp. 1163-1175, 2001.

[168] K. J. Åström and T. Hägglund, "The Future of PID Control," presented at Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 2000.

[169] A. M. de Paor, "A fiftieth anniversary celebration of the Ziegler-Nichols PID controller," *International Journal of Electrical Engineering Education*, vol. 30, pp. 303-316, 1993.

[170] D. W. Pessen, "A new look at PID controller tuning," *Journal of Dynamic Systems, Measurement, and Control*, vol. 116, pp. 553-557, 1994.

[171] K. J. Astrom and T. Hagglund, "Automatic tuning of PID controllers," in *The Control Handbook*, W. S. Levine, Ed.: IEEE Press, 1996, pp. 817-826.

[172] M. Zhuang and D. P. Atherton, "Optimal PID controller settings using integral performance criteria," presented at American Control Conference, 1991.

[173] G. P. Liu and S. Daly, "Optimal-tuning PID controller design in the frequency domain with application to a rotary hydrolic system," *Control Engineering Practice*, vol. 7, pp. 821-830, 1999.

[174] G. P. Liu and S. Daly, "Optimal-tuning nonlinear PID control of hydrolic systems," *Control Engineering Practice*, vol. 8, pp. 1045-1053, 2000.

[175] G. P. Liu and S. Daly, "Optimal-tuning PID control for industrial systems," *Control Engineering Practice*, vol. 9, pp. 1185-1194, 2001.

[176] N. Munro, "The systematic design of PID controllers," presented at IEE Colloquium on Symbolic Computation for Control, 1999.

[177] S. Wolfram, "Mathematica 4.2," Wolfram Research, Inc., 100 Trade Canter Drive, Champaign, IL 61820, USA, 2002.

[178] M. T. Ho, "Synthesis of H-infinity PID controllers: A parametric approach," *Automatica*, vol. 39, pp. 1069-1075, 2003.

[179] N. Munro and M. T. Solyemez, "Fast calculation of stabilising PID controllers for uncertain parameter systems," presented at 3rd IFAC symposium on robust control design, Prague, Czech Republic, 2000.

[180] J. Ackermann, "Stable polyhedra in parameter space," *Automatica*, vol. 39, pp. 937-943, 2003.

[181] T. Koshy, *Fibonacci and Lucas Numbers with Applications*: John Wiley & Sons, Inc., 2001.

[182] A. M. de Paor, "Pascal - Routh polynomials: a first exploration in feedback system design," *International Journal of Control*, vol. 76, pp. 386-389, 2003.

[183] W. Thomson and P. G. Tait, *Treatise on Natural Philosophy*. Oxford: Oxford University Press, 1867.

[184] S. Bennett, *A History of Control Engineering 1800-1930*: Peter Peregrinus Ltd, 1979.

[185] J. L. Synge, "On The Geometry of Dynamics," *Philosophical Transactions of the Royal Society of London - Series A - Mathematical and Physical Sciences*, vol. 226, pp. 31-106, 1926.