

A DESIGN PROCEDURE FOR MODEL REFERENCE

ADAPTIVE CONTROL

by

Jonathan Hill

Thesis submitted to the Faculty of

Virginia Polytechnic Institute and State University

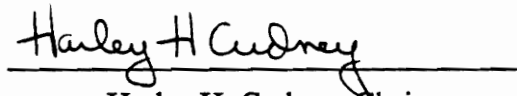
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

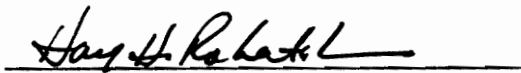
IN

MECHANICAL ENGINEERING

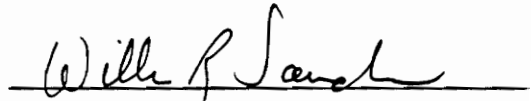
APPROVED:



Harley H. Cudney, Chair



Harry H. Robertshaw



William R. Saunders

August 10, 1995

Blacksburg, Virginia

c.2

LD
E655
P55
P95
H555
c.2

A DESIGN PROCEDURE FOR MODEL REFERENCE ADAPTIVE CONTROL

by

Jonathan Hill

Harley H. Cudney, Chairman

Mechanical Engineering

Abstract

In this study, we assess the robustness of four distinct control approaches: pole placement; the command generator tracker (CGT) approach to model reference control; model reference adaptive control (MRAC); and MRAC using a fixed feedback gain. We use a second order, single-input single-output (SISO) plant to examine the performance and stability of each method. This evaluation spans a broad range of design goals and uncertainty in models of the plant.

Pole placement and CGT designs are linear and relatively easy to implement, but require explicit knowledge of the plant. Although MRAC schemes require little knowledge of the plant's dynamic characteristics, such algorithms are non-linear and involve design variables whose effects are not readily apparent. Currently, there are no general design procedures for MRAC. In this study, we propose a method for designing an MRAC controller applied to a second order SISO plant. This method does not require the controller to be

tuned for different closed-loop performance goals. This procedure also creates a consistent basis for comparing the robustness of all four algorithms.

Pole placement and the CGT control perform as designed if the plant is modeled correctly. Under this circumstance, the adaptive controllers also perform at levels equivalent to the linear algorithms. However, conditions with plant modeling error highlight enormous differences among the four algorithms. Pole placement suffers the largest response error and for extreme testing conditions, instability. The CGT controller exhibits better performance than pole placement and remains stable over all testing variables. MRAC maintains a high performance level under severe testing conditions. MRAC requires minimal plant knowledge to guarantee stability and good performance.

Acknowledgments

I would like to express my gratitude to my committee chairman Dr. Harley Cudney for his guidance and time dedicated to working with me during my graduate studies. I also appreciate his faith by granting me the opportunity to explore areas beyond the scope of our original intention.

I also wish to thank my committee members, Dr. Harry Robertshaw and Dr. William Saunders, for their interest and consultation during my research.

I thank my associates in “the lab”, Rich Lomenzo, Anton Sumali, David Schmiel, Wes Holley, Tom McCray, Chris Niezrecki, and Larry Tentor who have endured my shenanigans and contributed to a strange but enjoyable place to work. Thanks to Randy Soper with whom I have discussed obscure linear algebraic manipulations and spent many late night hours fulfilling our graduate requirements.

The staff of the Cellar also deserves recognition. Without their “friendly service”, wonderful studying environment, and unlimited coffee, I would not have finished as quickly nor enjoyed the process.

Finally, I must express my gratitude to my parents. Their love, support, and at times, money, made life at Virginia Tech much more bearable during times of distress.

Table of Contents

Abstract.....	ii
Acknowledgments.....	iv
List of Tables.....	xi
List of Figures.....	xiv
I Introduction.....	1
1.1 Background.....	1
1.2 Objectives.....	4
1.3 Contribution.....	5
1.4 Outline.....	5
II Formulating Control Algorithms.....	7
2.1 The Basis for Control.....	7
2.1.1 Plant.....	7
2.1.2 Closed-Loop Performance.....	10
2.2 Method I: Pole Placement.....	11

2.2.1 Continuous Case.....	13
2.2.2 Discrete Case.....	19
2.2.3 Summary.....	25
2.3 Method II: The Command Generator Tracker Approach to Model Reference Control.....	25
2.3.1 Continuous Case.....	28
2.3.2 Discrete Case.....	33
2.3.3 Summary.....	37
2.4 Method III: Model Reference Adaptive Control.....	38
2.4.1 Continuous Case I.....	38
2.4.2 Continuous Case II.....	41
2.4.3 Discrete Case.....	43
2.5 Method IV: Model Reference Adaptive Control with Time-Invariant Output Feedback Gain.....	45
2.5.1 Continuous Case I.....	45
2.5.2 Continuous Case II.....	46
2.5.3 Discrete Case.....	47
2.6 Summary.....	48
III Analyzing Linear Control Algorithm Stability.....	50
3.1 Evaluating Relative Stability.....	50

3.2 Pole Placement.....	52
3.2.1 Continuous Case.....	53
3.2.2 Discrete Case.....	55
3.3 Command Generator Tracker.....	58
3.4 Summary.....	59
 IV Analyzing Stability of Model Reference Adaptive Control.....	61
4.1 Continuous Case I.....	63
4.1.1 Formulating Lyapunov Functions.....	63
4.1.2 Satisfying Stability Constraints.....	71
4.1.3 Summary.....	72
4.2 Continuous Case II.....	72
4.2.1 Formulating Lyapunov Functions for Known CGT Gains.....	73
4.2.2 Formulating Lyapunov Functions for Nominal CGT Gains.....	77
4.2.3 Satisfying Stability Constraints.....	81
4.2.4 Summary.....	81
4.3 Discrete Case.....	82
4.3.1 Formulating the Lyapunov Functions for Known CGT Gains.....	82
4.3.2 Formulating the Lyapunov Functions for Nominal CGT Gains.....	88
4.3.3 Satisfying Stability Constraints.....	92
4.3.4 Summary.....	93

4.4 Model Reference Adaptive Control with Time-Invariant Output Feedback Gain:	
Continuous Case I.....	93
4.5 Model Reference Adaptive Control with Time-Invariant Output Feedback Gain:	
Continuous Case II.....	95
4.5.1 Known CGT Gains.....	95
4.5.2 Nominal CGT Gains.....	97
4.6 Discrete Model Reference Adaptive Control with Time-Invariant Output Feedback	
Gain.....	97
4.6.1 Known CGT Gains.....	97
4.6.2 Nominal CGT Gains.....	99
4.7 Summary.....	100
V Discrete Design Procedures.....	101
5.1 Preliminary Design Considerations.....	101
5.1.1 Plant Modeling Error.....	103
5.1.2 Objectives.....	103
5.1.3 Discrete Time Domain.....	105
5.2 Pole Placement.....	107
5.2.1 Performance Gain Selection.....	107
5.2.2 Observer Gain Selection.....	108
5.3 Command Generator Tracker.....	109

5.3.1 Selecting Performance Gains.....	111
5.3.2 Selecting the Output Feedback Gain.....	114
5.4 Model Reference Adaptive Control.....	118
5.4.1 The Lyapunov Functions and ASPR Constraints.....	118
5.4.2 Choosing Weighting Matrices.....	125
5.5 Model Reference Adaptive Control with Fixed Feedback Gain.....	133
5.5.1 The Lyapunov Functions and ASPR Constraints.....	133
5.5.2 Choosing Weighting Matrices.....	134
5.6 Summary.....	136
 VI Comparison of the Four Algorithms.....	 138
6.1 Overview.....	138
6.2 Discrete Model Reference Control Using the CGT Approach.....	140
6.3 Model Reference Adaptive Control.....	144
6.3.1 Model Reference Adaptive Control for Explicit Knowledge of Plant Dynamics.....	145
6.3.2 Examining the Choice of Representative Error.....	157
6.3.3 Alternative Weighting Matrix Tuning.....	161
6.4 The Effects of Plant Modeling Error.....	165
6.5 Minimizing the Effects of Plant Modeling Error in Model Reference Adaptive Control.....	174

6.6 Analyzing Controller Robustness.....	176
VII Conclusions and Recommendations.....	205
References.....	208
Appendix A: Controllability and Observability.....	212
Appendix B: Transformation to Feedback Canonical Form.....	213
Appendix C: Strictly Positive Real Conditions.....	215
C.1 Terminology.....	216
C.2 Continuous Case I.....	218
C.3 Continuous Case II.....	222
C.4 Discrete Case.....	224
Appendix D: Predicting Representative Trajectory Values.....	233
D.1 Underdamped Case.....	233
D.2 Critically Damped Case.....	235
Vita.....	237

List of Tables

5-1	Relation of the Plant's Dynamic Characteristics to Closed-Loop Pole Locations for $T_s = 0.02$ seconds.....	116
5-1	Relation of the Plant's Dynamic Characteristics to Closed-Loop Pole Locations for $T_s = 0.0032$ seconds.....	116
6-1a	Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$	181
6-1b	CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$	182
6-1c	MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$	183
6-1d	Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$	184
6-2a	Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$	185
6-2b	CGT Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$	186
6-2c	MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$	187

6-2d	Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$	188
6-3a	Pole Placement Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	189
6-3b	CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	190
6-3c	MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	191
6-3d	Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	192
6-4a	Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	193
6-4b	CGT Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	194
6-4c	MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	195
6-4d	Fixed Gain MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	196

6-5a	Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.33$	197
6-5b	CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.33$	198
6-5c	MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.33$	199
6-5d	Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.33$	200
6-6a	Pole Placement Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.01$	201
6-6b	CGT Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.01$	202
6-6c	MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.01$	203
6-6d	Fixed Gain MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.01$	204

List of Figures

2-1	Block Diagram of a Continuous Pole Placement Control System with Full State Availability.....	13
2-2	Block Diagram of a Continuous Pole Placement Control System with a Reduced Order Observer.....	15
2-3	Block Diagram of Discrete Pole Placement Control System with a Reduced Order Observer.....	24
2-4	Block Diagram of Continuous Model Reference Control System Using the CGT Approach.....	32
2-5	Block Diagram of the Continuous Model Reference Adaptive Control System....	40
3-1	Break Point Location for a Continuous Pole Placement Control System with a Reduced Order Observer.....	53
3-2	Break Point Location for a Discrete Pole Placement Control System with a Reduced Order Observer.....	56
3-3	Broken Loop of the Continuous CGT Control (Unsimplified Form).....	58
3-4	Broken Loop of the Continuous CGT Control (Simplified Form).....	59
5-1	Pole and Zero Locations of the Continuous Nominal Plant.....	102
5-2	Continuous Plant Pole and Zero Locations.....	104
5-3	Root Locus Plot of the Discrete Nominal Plant.....	110
5-4	Root Locus Plots of Discrete Plant.....	115
5-5	Design Flowchart.....	137

6-1a	Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	140
6-1b	Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	141
6-2a	Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	142
6-2b	Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	142
6-2c	Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	143
6-2d	Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	143
6-3a	Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	146
6-3b	Discrete MRAC Using A Fixed Feedback Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	147
6-4a	Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	148

6-4b	Discrete MRAC Using A Fixed Feedback Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	149
6-5a	Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	150
6-5b	Discrete MRAC Using A Fixed Feedback Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	151
6-6a	Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	152
6-6b	Discrete MRAC Using A Fixed Feedback Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	155
6-7a	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	156
6-7b	Discrete MRAC Using A Fixed Feedback Gain for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$	157
6-8a	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Assuming an Increased Error Estimate by a Factor of 10.....	159
6-8b	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Assuming an Increased Error Estimate by a Factor of 100.....	160

6-8c	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Assuming an Increased Error Estimate by a Factor of 1000.....	161
6-9a	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = 1$	163
6-9b	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = 100$	163
6-9c	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^4$	164
6-9d	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^6$	164
6-9e	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^{10}$	165
6-10a	Discrete Pole Placement Control for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 0.7\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	166
6-10b	Discrete Pole Placement Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 0.7\zeta_p$	167
6-10c	Discrete Pole Placement Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	167

6-11a	Discrete CGT Control for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 0.7\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	168
6-11b	Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 0.7\zeta_p$	169
6-11c	Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	169
6-12a	Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 0.7\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	170
6-12b	Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 0.7\zeta_p$	171
6-12c	Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	172
6-13a	Discrete MRAC Using a Fixed Gain for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 0.7\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	172
6-13b	Discrete MRAC Using a Fixed Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 0.7\zeta_p$	173
6-13c	Discrete MRAC Using a Fixed Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 1.3\omega_p$ and $\zeta_{pm} = 1.3\zeta_p$	173

6-14a	G vs. PI for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = 0.7 \omega_p$ and $\zeta_{pm} = 1.3 \zeta_p$	174
6-14b	G vs. PI for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = 1.3 \omega_p$ and $\zeta_{pm} = 0.7 \zeta_p$	175
6-15a	Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 1.0$	181
6-15b	CGT Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 1.0$	182
6-15c	MRAC Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 1.0$	183
6-15d	Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 1.0$	184
6-16a	Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.17$	185
6-16b	CGT Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.17$	186
6-16c	MRAC Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.17$	187
6-16d	Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.5 \omega_{pm}$ and $\zeta_m = 0.17$	188
6-17a	Pole Placement Performance vs. Modeling Error for $\omega_m = 0.5 \omega_{pm}$ and $\zeta_m = 1.0$	189

6-17b	CGT Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	190
6-17c	MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	191
6-17d	Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$	192
6-18a	Pole Placement Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	193
6-18b	CGT Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	194
6-18c	MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	195
6-18d	Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$	196
6-19a	Pole Placement Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$	197
6-19b	CGT Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$	198
6-19c	MRAC Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$	199
6-19d	Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$	200
6-20a	Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$	201

6-20b CGT Performance vs. Modeling Error for $\omega_m = 15\omega_{pm}$ and $\zeta_m = 0.01$202

6-20c MRAC Performance vs. Modeling Error for $\omega_m = 15\omega_{pm}$ and $\zeta_m = 0.01$203

6-20d Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 15\omega_{pm}$ and $\zeta_m = 0.01$204

Chapter 1

Introduction

1.1 Background

The introduction of state space descriptions into control law design prompted the era of "modern control". This occurred in the late 1950's. Describing differential equations with state space matrices allowed a new perspective that was particularly conducive to computer calculation (Franklin *et al.*, 1991). The advantages of state space methods include an easier treatment of multi-input, multi-output systems and a simple approach to specify the closed-loop eigenvalues of the plant using pole placement.

During the 1950's, a new method of control grew into prominence. The academic nature of pole placement did not lend itself well to real world applications. The world, being a basically non-linear place, made plant identification a difficult endeavor for complex systems. For a method such as pole placement to effectively control a system, explicit knowledge of plant dynamics is required. The concept of adaptive control lessened the emphasis on a priori knowledge. Tsytkin (1966) described the adaptive process as "changing the parameters structure and possibly the controls of a system on the basis of information obtained during the control period so as to optimize (from one point of view or another) the state of a system when the operating conditions are either incompletely defined initially or changed".

Two fundamental types of adaptive control are explicit adaptive control and implicit adaptive control. The former uses a recursive plant identification algorithm and generates a control based upon the plant parameter estimates. Implicit adaptive control does not identify the plant. Instead, this approach specifies desirable closed-loop performance characteristics in a reference model. Using trajectory errors between the plant and reference model, and the state trajectories of the reference model, the control gains evolve into values forcing the plant's behavior to emulate that of the reference model. In this study, we are concerned with implicit (also known as direct) model reference adaptive control (MRAC).

Whitaker (1958) originated implicit adaptive control with his well known "M.I.T. rule". This simple algorithm relied on the error trajectories between the plant and reference model, and the sensitivity of error with respect to the adaptive gains. Its ease of implementation made the M.I.T. rule an attractive approach, however, stability could not be analytically guaranteed.

In 1961, LaSalle published Stability by Lyapunov's Direct Method. This treatment of non-linear differential equations provided a new means to assess adaptive control stability. Using Lyapunov's direct method, the designer analytically determines that a control scheme is stable by satisfying the constraints placed upon selected Lyapunov functions. Often, these functions represent some kind of energy that prove to be dissipative for a sufficient guarantee of stability.

Adaptive control grew rapidly over the next two decades. Until 1980, adaptive control designers faced two harsh constraints on the plant. Although stability conditions could be met, fulfilling Lyapunov constraints often required that the entire plant state vector be available (Erzberger, 1963) and that the open-loop plant be strictly positive real (Mabius and Kaufman, 1976). To illustrate the severity of such constraints, a single degree of freedom, spring-mass-damper system with velocity output could not be subjected to MRAC with guaranteed stability.

In his Ph.D. thesis, Sobel (1980) devised an implicit MRAC scheme that relaxed many of the conditions placed on the plant. The plant no longer needed to be strictly positive real, and output, instead of the state vector, became sufficient to promise stability. Sobel accomplished this by using an approach developed by O'Brien and Broussard (1979). Their work provided a basis for model reference linear control with the same relaxed plant constraints. Broussard's contribution is known as the command generator tracker (CGT) concept. Not only did the CGT design negate the need to use the plant state vector, the reference model could be of lower order than the plant. This eliminated the effect of the unmodeled dynamics of the plant on stability. The linear command generator tracker control became the foundation for Sobel's innovative MRAC algorithm.

Since 1980, stability concerns have been underscored by a growing interest in applications. However, most literature fails to include or address any sort of systematic design procedure. Model reference adaptive control involves weighting matrices, much akin to the "Q" and "R" weighting matrices in linear quadratic regulators. The "black box"

nature of MRAC makes design a more amorphous process than in linear quadratic regulation (LQR). Whereas stability may be ensured by meeting the sufficient conditions of Lyapunov, we gain little insight into performance from the Lyapunov functions. Kaufman *et al.* (1994) prescribe some guidelines for selecting weighting matrices. Their suggestions are rather vague and fail to address control law design for poor knowledge of plant dynamics.

In addition to the lack of published design procedures, let alone a hesitation to include weighting matrix values, there is surprisingly little documentation on discrete model reference adaptive control. Ionescu and Monopoli (1977) proposed the first discrete MRAC system. Sobel (1980) addresses discrete MRAC in great detail but to our knowledge, this has remained the most substantial contribution to date.

1.2 Objectives

In this work, we compare the robustness of four digital controllers: pole placement; the command generator tracker approach to model reference control; model reference adaptive control; and model reference adaptive control using one fixed gain. The term robustness implies the ability to achieve "reasonable" performance over a variety of uncertain conditions. In our study, these uncertain conditions are limited to an established range of plant modeling error. The performance goals consist of changing the closed-loop values for natural frequency and damping ratio of a second order plant.

Our inability to locate a derivation of a discrete model reference control algorithm leads us to propose and test a discrete CGT controller. Additionally, we must develop a

robust design procedure for this control approach.

Comparing the three algorithms in a consistent fashion precludes the possibility of fine tuning the MRAC algorithm for each set of testing conditions. For the range of design goals and plant modeling uncertainty, we seek a design procedure that minimizes error but does not require user-controller interaction.

1.3 Contribution

Comparing robustness demonstrates the conditions for which each controller is best suited. The results may serve as basis for designers wishing to weigh algorithm complexity and computation time against performance and stability.

Our development of the discrete time model reference controller provides an alternative perspective than that presented by Sobel (1980). Sobel's approach is complex and somewhat counterintuitive. In this study, we contrast his findings with ours

Our discrete MRAC design procedure, though applied to a simple system, is unique. Our insights and design methods may serve others wishing to work with more complex systems.

1.4 Outline

In Chapter 2, we define the control problem and formulate continuous and discrete versions of pole placement, model reference control, and model reference adaptive control. In Chapters 3 and 4, we derive stability conditions for the linear algorithms and non-linear algorithms, respectively.

We develop our discrete design procedures in Chapter 5 followed by simulating and comparing each control law in Chapter 6. These two chapters contain the most important contributions of this study.

In chapter 7, we draw our conclusions and suggest possibilities for future research.

Chapter 2

Formulating Control Algorithms

The control algorithms presented in this chapter consist of a pole placement (full state feedback) design applied to a single input, single output (SISO) system, the command generator tracker approach to model reference control, and three model reference adaptive control designs, each with a fixed gain option. Whereas we provide continuous and discrete algorithms, control laws are typically implemented on digital signal processors. Consequently, our simulations only involve the discrete algorithms. We formulate the continuous algorithms to lead the reader through this material with greater ease.

2.1 The Basis for Control

Controllers are used to change the dynamic behavior of a process or plant. The particular objectives of a controller may include specifying the closed-loop dynamics, rejecting disturbances, or tracking reference inputs. In this thesis, we assess the robustness of a variety of control laws applied to shaping the response of a process to a command. In essence, each of our control laws attempts to specify the eigenvalues of the closed-loop system while maintaining stability over a range of uncertain conditions.

2.1.1 Plant

The term plant refers to the process that is subject to control. We represent the dynamics of the plant in the state-space form. All analog control algorithms are based

upon the linear, time-invariant (LTI) state equations in the continuous time domain,

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) \quad (2.1-1)$$

and

$$y_p(t) = C_p x_p(t). \quad (2.1-2)$$

We formulate the digital control algorithms for the LTI state equations in the discrete time domain,

$$x_p(k+1) = \Phi_p x_p(k) + \Gamma_p u_p(k) \quad (2.1-3)$$

and

$$y_p(k) = C_p x_p(k). \quad (2.1-4)$$

Let the trajectories, for both the continuous and the discrete cases, be defined such that $x_p(\cdot)$ ($n \times 1$) is the plant state vector, $u_p(\cdot)$ ($m \times 1$) is the plant control vector, and $y_p(\cdot)$ ($q \times 1$) is the plant output vector. The term C_p ($q \times n$) represents the plant output matrix. The control designer specifies this matrix. We assume that C_p does not deviate from its established values.

In this study, we use four assumptions regarding the plant.

(1) The plant parameters exist within a specified range of uncertainty and variation.

- $\underline{a}_{ij} \leq a_{ij} \leq \bar{a}_{ij}$ for $i = 1, 2, \dots, n; j = 1, 2, \dots, n$
- $\underline{b}_{ij} \leq b_{ij} \leq \bar{b}_{ij}$ for $i = 1, 2, \dots, n; j = 1, 2, \dots, m$
- $\underline{\phi}_{ij} \leq \phi_{ij} \leq \bar{\phi}_{ij}$ for $i = 1, 2, \dots, n; j = 1, 2, \dots, n$
- $\underline{\gamma}_{ij} \leq \gamma_{ij} \leq \bar{\gamma}_{ij}$ for $i = 1, 2, \dots, n; j = 1, 2, \dots, m$

Note that a_{ij} represents the element of A_p corresponding to the i^{th} row and the j^{th} column, for example. The term \underline{a}_{ij} represents the minimum possible value of a_{ij} and \bar{a}_{ij} denotes its maximum possible value.

- (2) Within the bounds specified by assumption (1), all possible pairs (A_p, B_p) and (Φ_p, Γ_p) are controllable.
- (3) Within the bounds specified by assumption (1), all possible pairs (A_p, C_p) and (Φ_p, C_p) are observable.

Statements (2) and (3) are sufficient to claim that the sets $\{A_p, B_p, C_p\}$ and $\{\Phi_p, \Gamma_p, C_p\}$ are minimal realizations of the continuous and discrete plant, respectively (Vidyasagar, 1993). Appendix A contains definitions of controllability and observability.

- (4) The plant input matrices B_p and Γ_p have maximum rank. This requirement implies that the control inputs are linearly independent. Such a concern does not, however, apply to a single input system.

As we cover the various control algorithms, we constrain the plant further. The reader may note that the following claims are not yet made.

- (1) The open-loop plant is Hurwitz (all poles in the left-hand side of the complex plane).
- (2) The open-loop plant is minimum phase (all zeros in the left-hand side of the complex plane).

(3) The open-loop plant is proper or strictly proper (for the single input, single output (SISO) case, this means that the open-loop plant has a relative degree of 0 or 1, respectively).

2.1.2 Closed-Loop Performance

To evaluate the controlled plant's performance, we consider how closely the response emulates some ideal trajectory. The ideal trajectory is the designer's conception of the plant output during control. We discuss two basic approaches to controlling the response of the plant. In the first approach, pole placement (full state feedback), a compensator in the feedback loop accesses the plant's internal dynamic states and generates an input signal to define the closed-loop characteristic equation. The resulting characteristic equation is designed to have dynamic properties corresponding to the ideal trajectory. Thus, the output of the plant should match the ideal trajectory.

The model reference approach is an alternative to full state feedback. In this approach, the ideal trajectory is physically generated by another dynamic system: the reference model (or simply, "model"). This control law is designed so that the plant's output asymptotically tracks that of the model.

The state dynamics of the model define the characteristics of the ideal output. The designer specifies desirable performance characteristics of the plant through shaping the response of the model to a command input. The continuous LTI state equations that describe the reference model are

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m \mathbf{x}_m(t) + \mathbf{B}_m \mathbf{u}_m(t) \quad (2.1-5)$$

and

$$y_m(t) = C_m x_m(t). \quad (2.1-6)$$

The LTI state equations representing the discrete time reference model are

$$x_m(k+1) = \Phi_m x_m(k) + \Gamma_m u_m(k) \quad (2.1-7)$$

and

$$y_m(k) = C_m x_m(k). \quad (2.1-8)$$

The trajectories, for continuous and discrete cases, are such that $x_m(\cdot)$ ($n_m \times 1$) is the model state vector; $u_m(\cdot)$ ($m_m \times 1$) is the model input vector; and $y_m(\cdot)$ ($q \times 1$) is the model output vector. The matrix terms A_m and Φ_m ($n_m \times n_m$) are the model state dynamics matrices, B_m and Γ_m ($n_m \times m_m$) denote the model input matrices, and C_m ($q \times n_m$) represents the model output matrix.

We use three assumptions about the model:

- (1) The reference model is bounded input, bounded output (BIBO) stable.
- (2) The order of the model may be less than the order of the process.
- (3) The plant and model have the same canonical forms.

2.2 Method I: Pole Placement

In this section, we provide a review of pole placement theory, discuss our assumptions, and define the gains used in simulation. The reader may find most of this information in Friedland (1986) and Franklin *et al.* (1991) for the continuous case. Our presentation of discrete pole placement theory comes predominantly from Franklin *et al.*

(1990). This section contains no original theory but is pertinent to later discussion and simulations.

Pole placement designs use the internal dynamic states of the plant to place the eigenvalues of the closed-loop system. The internal dynamic states may be directly measured, or, if the system is observable, estimated.

In this study, we use six assumptions regarding pole placement.

- (1) The internal dynamic states are not all available through the output.
- (2) The output, alone, is not sufficient to assign closed-loop eigenvalues.
- (3) The plant's output signal is uncorrupted by noise.
- (4) The parameters defining the plant's state equations are explicitly known. The reader may infer that this statement contradicts assumption (1) of section 2.1.1. As chapter 6 will demonstrate, pole placement control performs at an optimum if the plant is modeled perfectly. Performance declines, however, with increasing modeling error. Pole placement theory does not consider real world application as it assumes that the designer has explicit knowledge of the plant; a major shortcoming of this approach.
- (5) This algorithm is restricted to the SISO case: $m = q = 1$.
- (6) The number of distinct eigenvalues that describe the ideal output trajectory must be equal to the order of the plant.

In our problem formulation, we present a reduced order observer (ROO) to estimate the unmeasured states. A reduced order observer is not the only method of state

estimation but tends to perform better than a full order observer in the presence of plant modeling error.

2.2.1 Continuous Case

We assume full state availability for the initial presentation of the pole placement formulation. In this scenario, the compensator extracts each dynamic state without measurement. The state vectors feed through a gain matrix to form an input signal. The

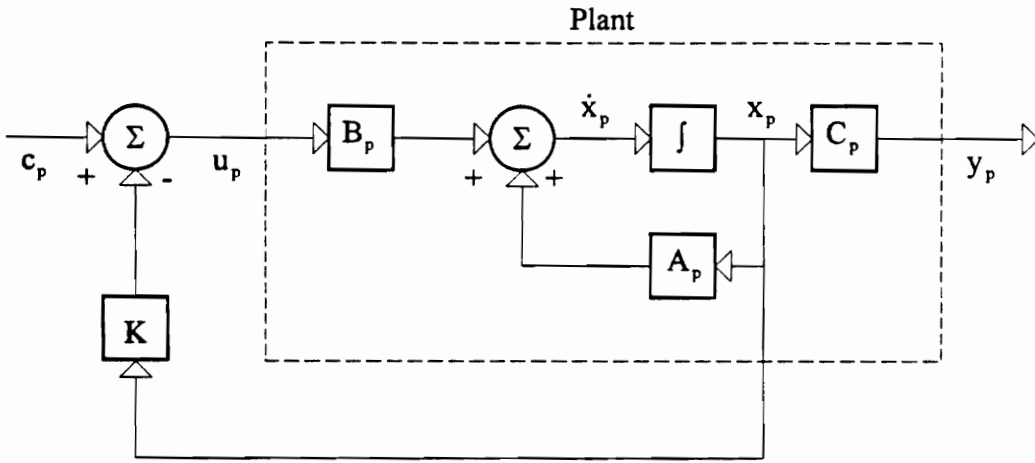


Figure 2-1
Block Diagram of a Continuous Pole Placement Control System with Full State Availability

control incorporates state feedback and a command input, $c_p(t)$. In Fig. 2-1, we illustrate this control method.

$$u_p(t) = c_p(t) - Kx_p(t) \quad (2.2-1)$$

In the gain matrix K ($m \times n$), we select values that produce the desired closed-loop characteristic equation,

$$|sI - A_p + B_p K| = 0, \quad (2.2-2)$$

thus shaping the plant's dynamic response. Note that 's' denotes the Laplace operator.

The designer calculates the determinant of Eq. (2.2-2), symbolically in terms of K, and equates the expression to the selected characteristic equation (the design goal). In the SISO case, this matrix is uniquely defined for a given set of closed-loop pole locations. By selecting K, the resulting closed-loop plant exhibits dynamics identical to the ideal trajectory. This method of eigenvalue assignment, however, requires explicit knowledge of A_p and B_p . Without this knowledge, the closed-loop plant will not behave as expected and may even become unstable.

For a system whose plant output matrix provides insufficient state information, we implement a reduced order observer. We review this approach, using the work of Friedland (1986). Figure 2-2 illustrates pole placement control with state observation.

The state equations describing the plant's dynamics are partitioned into submatrices corresponding to the states that are available through the output and to those which are not. The vector $x_{p1}(t)$ ($n_o \times 1$) denotes the n_o measured states and $x_{p2}(t)$ ($n_e \times 1$) represents the n_e unmeasured states.

$$\begin{bmatrix} \dot{x}_{p1}(t) \\ \dot{x}_{p2}(t) \end{bmatrix} = \begin{bmatrix} A_{p11} & A_{p12} \\ A_{p21} & A_{p22} \end{bmatrix} \begin{bmatrix} x_{p1}(t) \\ x_{p2}(t) \end{bmatrix} + \begin{bmatrix} B_{p1} \\ B_{p2} \end{bmatrix} u_p(t) \quad (2.2-3)$$

$$y_p(t) = \begin{bmatrix} C_{p1} & 0 \end{bmatrix} \begin{bmatrix} x_{p1}(t) \\ x_{p2}(t) \end{bmatrix} \quad (2.2-4)$$

The submatrices in these equations, Eq. (2.2-3) and Eq. (2.2-4), are A_{p11} ($n_o \times n_o$), A_{p12} ($n_o \times n_e$), A_{p21} ($n_e \times n_o$), A_{p22} ($n_e \times n_e$), B_{p1} ($n_o \times m$), B_{p2} ($n_e \times m$), and C_{p1} ($q \times n_o$).

The reduced order observer does not estimate the states in $x_{p1}(t)$ because those states are directly measured through the output. However, we must create an estimate of the vector, $x_{p2}(t)$. The estimate of $x_{p2}(t)$ is denoted $\tilde{x}_{p2}(t)$, and we assume it has the form,

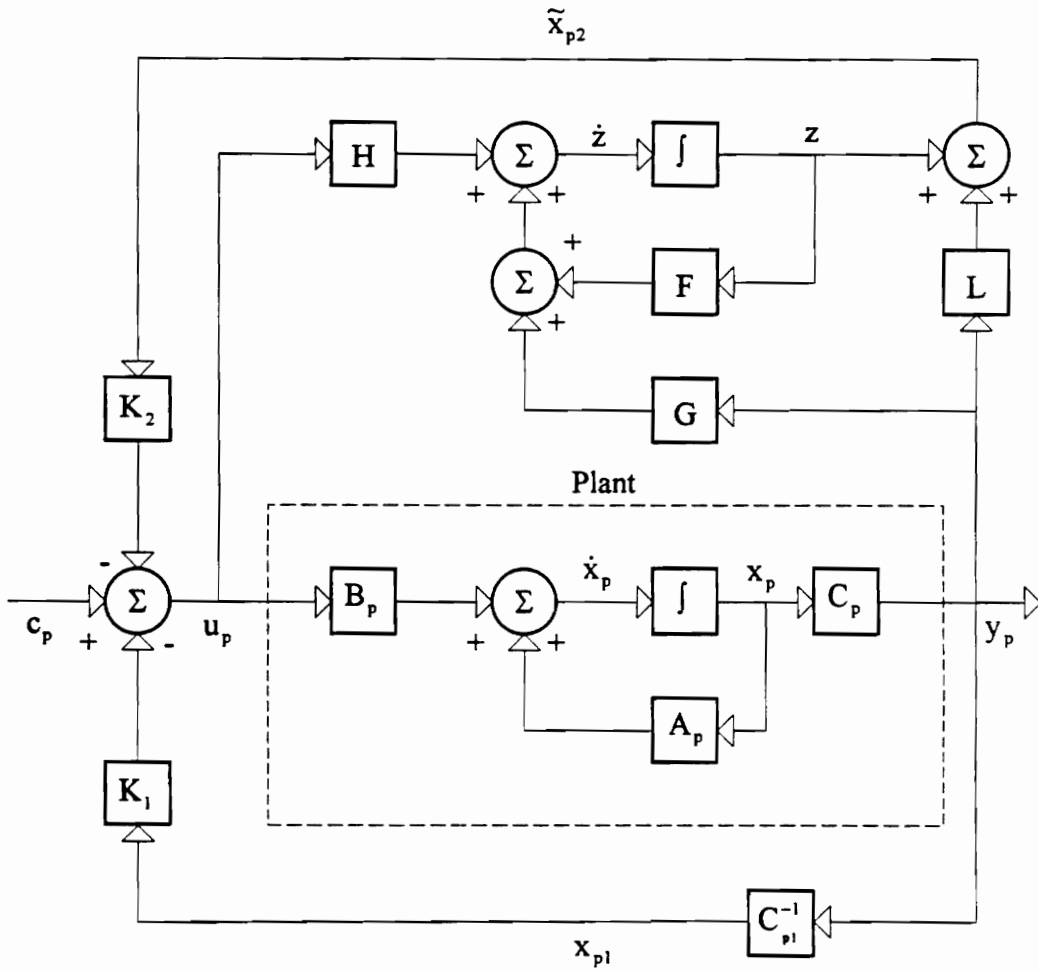


Figure 2-2
Block Diagram of a Continuous Pole Placement Control System with a Reduced Order Observer

$$\tilde{x}_{p2}(t) = Ly_p(t) + z(t). \quad (2.2-5)$$

We design the estimate of the unmeasured state vector to converge to the state vector's true value. Before this convergence occurs, error exists and performance suffers. The vector $z(t)$ serves as an intermediate state trajectory aiding in the analysis of error transients. The dynamics of $z(t)$ are governed by,

$$\dot{z}(t) = Fz(t) + Gy_p(t) + Hu_p(t). \quad (2.2-6)$$

The elements in this equation have the dimensions L ($n_e \times q$), F ($n_e \times n_e$), G ($n_e \times q$), H ($n_e \times m$), and z ($n_e \times 1$).

We examine the estimation error dynamics to specify L , F , G , and H . First, let's define the error as the difference between the true state and the estimated state,

$$e_2(t) = x_{p2}(t) - \tilde{x}_{p2}(t). \quad (2.2-7)$$

By differentiating the error equation, we can examine its transient behavior.

$$\dot{e}_2(t) = \dot{x}_{p2}(t) - \dot{\tilde{x}}_{p2}(t)$$

For \dot{x}_{p2} , we substitute Eq. (2.2-3) and differentiate Eq. (2.2-5) to substitute for $\dot{\tilde{x}}_{p2}(t)$.

$$\dot{e}_2(t) = A_{p21}x_{p1}(t) + A_{p22}x_{p2}(t) + B_{p2}u_p(t) - L\dot{y}_p(t) - \dot{z}(t)$$

Differentiating the plant output equation of Eq. (2.2-4) produces,

$$\dot{y}_p(t) = C_{p1}\dot{x}_{p1}(t),$$

which we then expand using Eq. (2.2-3), the definition of $\dot{x}_{p1}(t)$,

$$\dot{y}_p(t) = C_{p1}(A_{p11}x_{p1}(t) + A_{p12}x_{p2}(t) + B_{p1}u_p(t)).$$

We substitute this result into the error equation, yielding,

$$\begin{aligned}\dot{e}_2(t) = & A_{p21}x_{p1}(t) + A_{p22}x_{p2}(t) + B_{p2}u_p(t) \\ & - LC_{p1}(A_{p11}x_{p1}(t) + A_{p12}x_{p2}(t) + B_{p1}u_p(t)) \\ & - Fz(t) - GC_{p1}x_{p1}(t) - Hu_p(t)\end{aligned}\quad (2.2-8)$$

Using equations (2.2-4), (2.2-5) and (2.2-7), we express $z(t)$ as,

$$z(t) = x_{p2}(t) - e_2(t) - LC_{p1}x_{p1}(t). \quad (2.2-9)$$

Substituting this new expression for $z(t)$ into the error equation produces,

$$\begin{aligned}\dot{e}_2(t) = & Fe_2(t) + (A_{p21} - LC_{p1}A_{p11} + FLC_{p1} - GC_{p1})x_{p1}(t) \\ & + (A_{p22} - LC_{p1}A_{p12} - F)x_{p2}(t) \\ & + (B_{p2} - LC_{p1}B_{p1} - H)u_p(t)\end{aligned}\quad (2.2-10)$$

To ensure that the error is controllable and asymptotically approaches zero, the coefficients of the state and input trajectories in Eq. (2.2-10) are equated to zero. Error becomes a mathematical function of itself, such that

$$\dot{e}_2(t) = Fe_2(t), \quad (2.2-11)$$

where the eigenstructure of the F matrix dictates the rate at which observation error decays. After choosing the F matrix, we solve the next three equations to specify L , G , and H .

$$F = A_{p22} - LC_{p1}A_{p12} \quad (2.2-12a)$$

$$H = B_{p2} - LC_{p1}B_{p1} \quad (2.2-12b)$$

$$GC_{p1} = A_{p21} - LC_{p1}A_{p11} + FLC_{p1} \quad (2.2-12c)$$

We can now combine the pole placement design with the reduced order observer design and evaluate the dynamics of the entire closed-loop system. Let the matrix of feedback gains, K , be partitioned such that K_1 ($m \times n_o$) corresponds to the measured state vector and K_2 ($m \times n_e$) corresponds to the estimated state vector.

$$K = [K_1 \quad K_2] \quad (2.2-13)$$

Rewriting the plant's input signal in Eq. (2.2-1) to include the estimated state vector,

$$\begin{aligned} u_p(t) &= c_p(t) - (K_1 x_{p1}(t) + K_2 \tilde{x}_{p2}(t)) \\ &= c_p(t) - (K_1 x_{p1}(t) + K_2 x_{p2}(t) - K_2 e_2(t)) \\ &= c_p(t) - Kx_p(t) + K_2 e_2(t) \end{aligned} \quad (2.2-14)$$

The state space representation of the closed-loop design is augmented by the states describing estimation error, and is given as,

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} A_p - B_p K & B_p K_2 \\ 0 & F \end{bmatrix} \begin{bmatrix} x_p(t) \\ e_2(t) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} c_p(t). \quad (2.2-15)$$

The characteristic equation derived from this matrix expression demonstrates that the dynamics of a pole placement design and of the estimator's error may be specified independently (the separation principle). We choose closed-loop eigenvalues by assigning values to K and F in the characteristic equation,

$$|sI - A_p + B_p K| |sI - F| = 0. \quad (2.2-16)$$

Although the closed-loop dynamics are specified independently from the dynamics of state estimation error, the presence of error affects the response of the controlled plant. The matrix F must be selected such that the real components of its eigenvalues are significantly larger in magnitude than those of the closed-loop plant design. The poles that dominate the system's response should be associated with the closed-loop plant and not with the observation error. There is no absolute method of selecting the eigenstructure of F . In this study, we not only require sufficiently "quick" dynamics, but margins of stability as well. The observer gain must not induce instability. We address these issues in Chapters 3 and 5.

2.2.2 Discrete Case

The six assumptions pertaining to the continuous case remain valid for the discrete approach. In a manner analogous to the continuous pole placement design, the state trajectories of the discretized plant feed through a gain matrix, and with a command input, form the control signal. The control is defined as

$$u_p(k) = c_p(k) - Kx_p(k), \quad (2.2-17)$$

where $c_p(k)$ ($m \times 1$) is the command input and K ($1 \times n$) is the matrix of feedback gains.

The gain matrix is solved for the selected closed-loop characteristic equation,

$$|zI - \Phi_p + \Gamma_p K| = 0. \quad (2.2-18)$$

We assume this system has insufficient state information. Thus, directly implementing the control law in Eq. (2.2-17) is not possible. A reduced order observer provides estimates of states not measured. Let $x_{p1}(k)$ ($n_0 \times 1$) denote the n_0 measured

states and $x_{p2}(k)$ ($n_e \times 1$) represent the n_e unmeasured states. The plant's state equations can then be written as,

$$\begin{bmatrix} x_{p1}(k+1) \\ x_{p2}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_{p11} & \Phi_{p12} \\ \Phi_{p21} & \Phi_{p22} \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \end{bmatrix} + \begin{bmatrix} \Gamma_{p1} \\ \Gamma_{p2} \end{bmatrix} u_p(k). \quad (2.2-19)$$

The plant output vector assumes the state space description,

$$y_p(k) = \begin{bmatrix} C_{p1} & 0 \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \end{bmatrix}. \quad (2.2-20)$$

The submatrices in these matrix expressions are Φ_{p11} ($n_o \times n_o$), Φ_{p12} ($n_o \times n_e$), Φ_{p21} ($n_e \times n_o$), Φ_{p22} ($n_e \times n_e$), Γ_{p1} ($n_o \times m$), Γ_{p2} ($n_e \times m$), and C_{p1} ($q \times n_o$).

The inherent differences, between the continuous and discrete domains, result in dissimilar approaches to observer formulation. Franklin *et al.* (1990) provide the formulation of the discrete reduced order observer.

We arrange the terms in Eq. (2.2-19) to separate measurable quantities from unmeasurable quantities. Consider the first term,

$$x_{p1}(k+1) = \Phi_{p11}x_{p1}(k) + \Phi_{p12}x_{p2}(k) + \Gamma_{p1}u_p(k). \quad (2.2-21)$$

The product $\Phi_{p12}x_{p2}(k)$ is not measurable. We can reorder Eq. (2.2-21) so that all right hand terms are available through measurement¹,

$$\Phi_{p12}x_{p2}(k) = x_{p1}(k+1) - \Phi_{p11}x_{p1}(k) - \Gamma_{p1}u_p(k). \quad (2.2-22)$$

If we define the right hand side as,

¹ Although $x_{p1}(k+1)$ is not available at time step k , please grant us a "leap of faith". This will prove not to be an article of concern.

$$\bar{y}(k) = x_{p1}(k+1) - \Phi_{p11}x_{p1}(k) - \Gamma_{p1}u_p(k), \quad (2.2-23)$$

then Eq. (2.2-22) becomes,

$$\Phi_{p12}x_{p2}(k) = \bar{y}(k). \quad (2.2-24)$$

Note that this equation parallels the form of the plant output equation,

$$C_p x_p(k) = y_p(k). \quad (2.1-4)$$

Let's examine the second term of Eq. (2.2-19),

$$x_{p2}(k+1) = \Phi_{p21}x_{p1}(k) + \Phi_{p22}x_{p2}(k) + \Gamma_{p2}u_p(k).$$

If we define the measurable part of the right hand side as,

$$\bar{u}_p(k) = \Phi_{p21}x_{p1}(k) + \Gamma_{p2}u_p(k), \quad (2.2-25)$$

then

$$x_{p2}(k+1) = \Phi_{p22}x_{p2}(k) + \bar{u}_p(k). \quad (2.2-26)$$

The form of Eq. (2.2-26) parallels the form of the plant's state equation,

$$x_p(k+1) = \Phi_p x_p(k) + \Gamma_p u_p(k). \quad (2.1-3)$$

From Franklin *et al.* (1990), the general equation for a prediction estimator is

$$\tilde{x}(k+1) = \Phi \tilde{x}(k) + \Gamma u(k) + L(y(k) - C \tilde{x}(k)). \quad (2.2-27)$$

Using equations (2.2-23) through (2.2-26), we establish equivalencies to be substituted into the general estimator equation.

$$\tilde{x}(k+1) = \tilde{x}_{p2}(k+1) \quad (2.2-28a)$$

$$\Phi \tilde{x}(k) = \Phi_{p22} \tilde{x}_{p2}(k) \quad (2.2-28b)$$

$$\Gamma u(k) = \Phi_{p21}x_{p1}(k) + \Gamma_{p2}u_p(k) \quad (2.2-28c)$$

$$y(k) = x_{p1}(k+1) - \Phi_{p11}x_{p1}(k) - \Gamma_{p1}u_p(k) \quad (2.2-28d)$$

$$C\tilde{x}(k) = \Phi_{p12}\tilde{x}_{p2}(k) \quad (2.2-28e)$$

We apply these equivalencies to the general estimator equation, producing the reduced order version of Eq. (2.2-27),

$$\begin{aligned} \tilde{x}_{p2}(k+1) = & \Phi_{p22}\tilde{x}_{p2}(k) + \Phi_{p21}x_{p1}(k) + \Gamma_{p2}u_p(k) \\ & + L(x_{p1}(k+1) - \Phi_{p11}x_{p1}(k) - \Gamma_{p1}u_p(k) - \Phi_{p12}\tilde{x}_{p2}(k)) \end{aligned} \quad (2.2-29)$$

We examine the dynamics of the error trajectory to design the matrix L . Defining the observation error as,

$$e_2(k) = x_{p2}(k) - \tilde{x}_{p2}(k), \quad (2.2-30)$$

we then take a time step advance to produce

$$e_2(k+1) = \Phi_{p22}(x_{p2}(k) - \tilde{x}_{p2}(k)) - L\Phi_{p12}(x_{p2}(k) - \tilde{x}_{p2}(k)).$$

By definition of $e_2(k)$, this reduces to

$$e_2(k+1) = (\Phi_{p22} - L\Phi_{p12})e_2(k). \quad (2.2-31)$$

For simplicity, let's define a new matrix F as,

$$F = \Phi_{p22} - L\Phi_{p12}. \quad (2.2-32)$$

Equation (2.2-31) can then be described as,

$$e_2(k+1) = Fe_2(k), \quad (2.2-33)$$

where the eigenstructure of the F matrix governs the decay of state estimation error. After selecting F , we solve for L using Eq. (2.2-32). Combining the ROO states with the

plant states, we examine the dynamic characteristics of the entire closed-loop system. The matrix of feedback gains is partitioned such that K_1 corresponds to the measured state vector and K_2 corresponds to the estimated state vector. Now, we can write the control law as,

$$\begin{aligned} u_p(k) &= c_p(k) - (K_1 x_{p1}(k) - K_2 \tilde{x}_{p2}(k)) \\ &= c_p(k) - (K_1 x_{p1}(k) + K_2 x_{p2}(k) - K_2 e_2(k)) \quad (2.2-34) \\ &= c_p(k) - Kx_p(k) + K_2 e_2(k) \end{aligned}$$

Presenting the state equations for the complete system in an augmented matrix form,

$$\begin{bmatrix} x_p(k+1) \\ e_2(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_p - \Gamma_p K & \Gamma_p K_2 \\ 0 & F \end{bmatrix} \begin{bmatrix} x_p(k) \\ e_2(k) \end{bmatrix} + \begin{bmatrix} \Gamma_p \\ 0 \end{bmatrix} c_p(k) \quad (2.2-35)$$

The characteristic equation for this system is

$$|zI - \Phi_p + \Gamma_p K| |zI - F| = 0, \quad (2.2-36)$$

where 'z' represents the forward time step operator. The poles governing the observer's error dynamics must have transients that settle in sufficiently fewer time steps than those of $|zI - \Phi_p + \Gamma_p K| = 0$.

For simplicity in our future discussion of this algorithm, we introduce the matrices H and G .

$$H = \Gamma_{p2} - L\Gamma_{p1} \quad (2.2-37)$$

$$G = \Phi_{p21} - L\Phi_{p22} \quad (2.2-38)$$

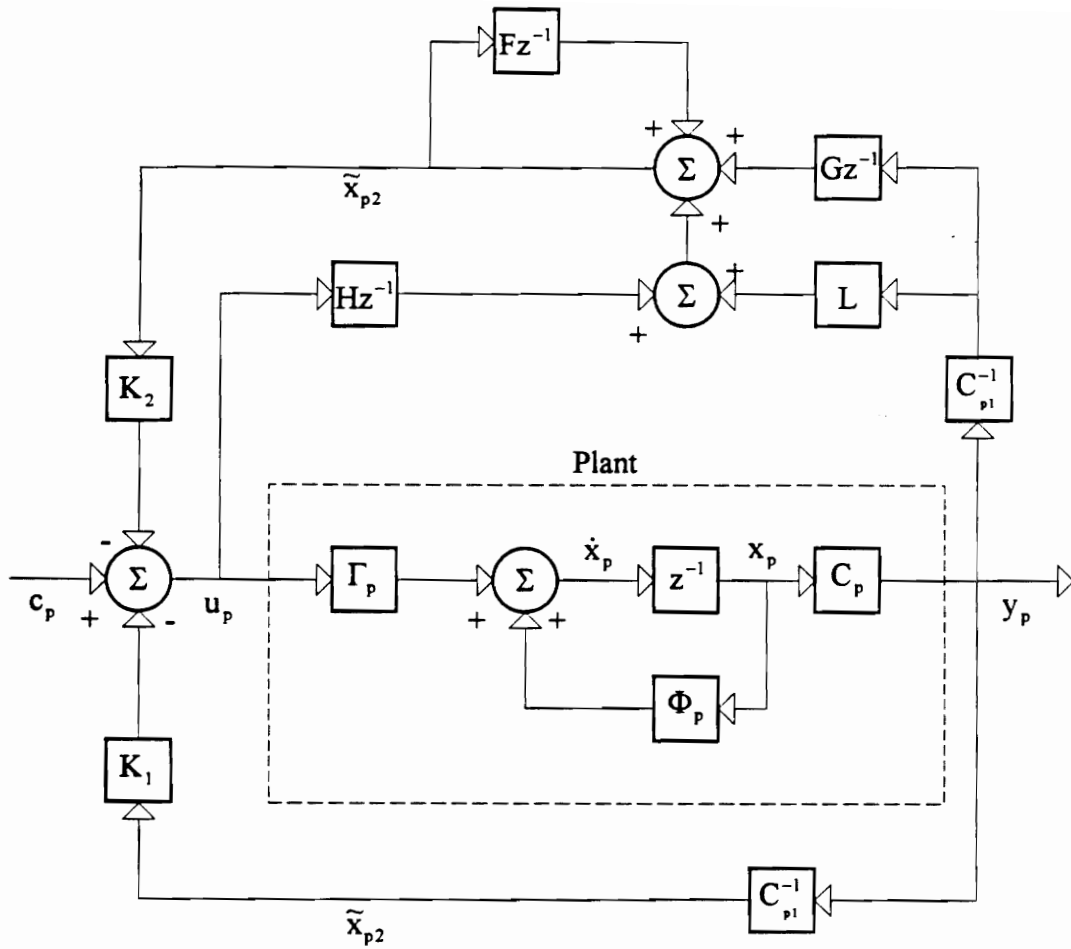


Figure 2-3
Block diagram of a Discrete Pole Placement Control System with a Reduced Order Observer

Equation (2.2-27) is easier to implement if considered using the discrete time operator, z (we can not use future time step values). This equation becomes

$$z\tilde{x}_{p2}(z) = F\tilde{x}_{p2}(z) + Gx_{p1}(z) + Hu_p(z) + Lzx_{p1}(z). \quad (2.2-39)$$

Division of both sides by the discrete time operator renders the equation into a form that is based upon past and current values instead of current and future values.

$$\tilde{x}_{p2}(z) = z^{-1}F\tilde{x}_{p2}(z) + z^{-1}Hu_p(z) + z^{-1}Gx_{p1}(z) + Lx_{p1}(z) \quad (2.2-40)$$

We illustrate the implementation of this controller in Fig. 2-3.

2.2.3 Summary

We reviewed the formulations of continuous and discrete pole placement control methods. We assumed the necessity of a reduced order observer and included its formulation as well. By the separation principle, closed-loop performance objectives and observer gain selection may be treated separately. However, the estimator gain must be appropriately chosen to limit the negative effects of observation error. As a final important note, closed-loop poles will not equal their design values unless the designer has a perfect model of the plant. Furthermore, it is conceivable that the closed-loop system be unstable if gains are calculated for an extremely poor plant model.

2.3 Method II: The Command Generator Tracker Approach to Model Reference

Control

O'Brien and Broussard (1979) developed a feedforward control technique that enables a plant output to track that of a reference model. This control is known as the command generator tracker (CGT) law. We derive this control law to achieve the ideal condition where the plant's output equals the reference model's output (perfect tracking),

$$y_p(\cdot) = y_m(\cdot). \quad (2.3-1)$$

The perfect tracking condition, illustrated by this equation, implies that

$$C_p x_p(\cdot) = C_m x_m(\cdot). \quad (2.3-2)$$

We introduce a new set of trajectories, the ideal trajectories, that always satisfy the perfect tracking condition. These trajectories are denoted by a superscript asterisk.

$$y_p^*(\cdot) = y_m(\cdot) \quad (2.3-3)$$

The ideal trajectories are generated by the response of the ideal plant to the ideal input. For the continuous case,

$$\dot{x}_p^*(t) = A_p^* x_p^*(t) + B_p^* u_p^*(t) \quad (2.3-4)$$

and

$$y_p^*(t) = C_p^* x_p^*(t). \quad (2.3-5)$$

For the discrete case,

$$x_p^*(k+1) = \Phi_p^* x_p^*(k) + \Gamma_p^* u_p^*(k) \quad (2.3-6)$$

and

$$y_p^*(k) = C_p^* x_p^*(k). \quad (2.3-7)$$

We constrain the ideal system realizations to be equal to the plant's system realization,

$$\{A_p^*, B_p^*, C_p^*\} = \{A_p, B_p, C_p\}, \quad (2.3-8)$$

or for the discrete case,

$$\{\Phi_p^*, \Gamma_p^*, C_p^*\} = \{\Phi_p, \Gamma_p, C_p\}. \quad (2.3-9)$$

Using these constraints, we determine an ideal input that causes perfect tracking when the ideal trajectories satisfy the state equations of the actual plant. Conversely, "when perfect

tracking occurs, the real plant trajectories become the ideal plant trajectory and the plant output becomes the ideal plant output" (Kaufman *et al.*, 1994).

Let's assume the ideal trajectories, $x_p^*(\cdot)$ and $u_p^*(\cdot)$, to be linear combinations of the model's state and input vectors. This assumption implies an open-loop control² approach which can be advantageous for stability. Let

$$\begin{bmatrix} x_p^*(\cdot) \\ u_p^*(\cdot) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m(\cdot) \\ u_m(\cdot) \end{bmatrix}. \quad (2.3-10)$$

The dimensions of the matrices in this equation are S_{11} ($n \times n_m$), S_{12} ($n \times m_m$), S_{21} ($m \times n_m$), and S_{22} ($m \times m_m$).

We also constrain the model input $u_m(\cdot)$ to be a type 0 or type 1 command (essentially, a time-invariant command). If the model input is time varying, its dynamics must be incorporated into the state dynamics matrix of the model (Kaufman *et al.*, 1994).

In this study, we assume that the plant and the model have the same canonical forms. We use the feedback (also called phase-variable) canonical form (see Appendix B), but do not claim this as necessary.

As a final note, Sobel (1980) presents continuous and discrete versions of the CGT law. Our discrete version differs from his. He provides no derivation and his result contains an extraneous term. As far as we know, our discrete CGT formulation can not

² This control is based on the reference model trajectories. Since no feedback loop surrounds the plant, we view the ideal control input as an open-loop approach.

be found in any other literature. Although we demonstrate in Chapter 6 that our version performs as expected, the reader may wish to examine our development closely.

2.3.1 Continuous Case

Equation (2.3-8) defines the ideal plant trajectories to satisfy the state dynamic equations of the plant. We substitute these for the actual state trajectories and place the state equations into matrix form.

$$\begin{bmatrix} \dot{x}_p^*(t) \\ y_p^*(t) \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x_p^*(t) \\ u_p^*(t) \end{bmatrix} \quad (2.3-11)$$

Substituting for $\begin{bmatrix} x_p^*(t) \\ u_p^*(t) \end{bmatrix}$ using Eq. (2.3-10) yields,

$$\begin{bmatrix} \dot{x}_p^*(t) \\ y_p^*(t) \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m(t) \\ u_m(t) \end{bmatrix}. \quad (2.3-12)$$

We differentiate the first term in Eq. (2.3-10) to produce,

$$\dot{x}_p^*(t) = S_{11} \dot{x}_m(t). \quad (2.3-13)$$

The model dynamics, given by $\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t)$, substitute into the previous equation.

$$\dot{x}_p^*(t) = S_{11} A_m x_m(t) + S_{11} B_m u_m(t). \quad (2.3-14)$$

Since $y_p^*(t) = C_m x_m(t)$, we establish,

$$\begin{bmatrix} \dot{x}_p^*(t) \\ y_p^*(t) \end{bmatrix} = \begin{bmatrix} S_{11} A_m & S_{11} B_m \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m(t) \\ u_m(t) \end{bmatrix}. \quad (2.3-15)$$

Considering equations (2.3-12) and (2.3-15), we have two linearly independent solutions for $\begin{bmatrix} \dot{x}_p(t) \\ \dot{y}_p(t) \end{bmatrix}$. Comparing these two equations,

$$\begin{bmatrix} S_{11}A_m & S_{11}B_m \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}. \quad (2.3-16)$$

This matrix expression produces four matrix equations and four matrix unknowns.

$$S_{11}A_m = A_p S_{11} + B_p S_{21} \quad (2.3-17a)$$

$$S_{11}B_m = A_p S_{12} + B_p S_{22} \quad (2.3-17b)$$

$$C_m = C_p S_{11} \quad (2.3-17c)$$

$$0 = C_p S_{12} \quad (2.3-17d)$$

If the matrix $\begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix}$ is nonsingular, we may define a new matrix Ω such that

$$\Omega = \begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix}. \quad (2.3-18)$$

Equation (2.3-16) may be rewritten as

$$\begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} S_{11}A_m & S_{11}B_m \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix},$$

where

$$S_{11} = \Omega_{11}S_{11}A_m + \Omega_{12}C_m, \quad (2.3-19a)$$

$$S_{12} = \Omega_{11}S_{11}B_m, \quad (2.3-19b)$$

$$S_{21} = \Omega_{21}S_{11}A_m + \Omega_{22}C_m, \quad (2.3-19c)$$

and

$$S_{22} = \Omega_{21} S_{11} B_m. \quad (2.3-19d)$$

The designer may solve for the unknown gains using either Eq. (2.3-17) or Eq. (2.3-19) or even a combination of both. The ease of solving may be a function of the plant's size and structure. Unfortunately, we cannot determine a closed-form matrix solution. In Chapter 5, we provide our solution process.

After solving for the unknown matrices, S_{21} and S_{22} , we construct the ideal input using Eq. (2.3-10),

$$u_p^*(t) = S_{21}x_m(t) + S_{22}u_m(t). \quad (2.3-20)$$

When the ideal plant trajectories satisfy the plant's state equations this control results in perfect tracking. If the plant's initial conditions differ from those of the model, then we violate the perfect tracking condition before we even begin! If the actual plant trajectories converge to the ideal trajectories then the ideal input can induce perfect tracking during the remainder of the response. We modify our control law to promote asymptotic tracking or equivalently, force the output error to asymptotically approach zero. We define error as the difference between the ideal state trajectories and the actual state trajectories of the plant,

$$e(t) = x_p^*(t) - x_p(t). \quad (2.3-22)$$

Let's examine the error dynamics. Differentiating Eq. (2.3-22) yields,

$$\dot{e}(t) = \dot{x}_p^*(t) - \dot{x}_p(t). \quad (2.3-23)$$

We substitute the state equations for the ideal plant and the model into this equation.

$$\dot{e}(t) = A_p x_p^*(t) + B_p u_p^*(t) - A_p x_p(t) - B_p u_p(t) \quad (2.3-24)$$

Using the definition of error from Eq. (2.3-23), we have,

$$\dot{e}(t) = A_p e(t) + B_p (u_p^*(t) - u_p(t)). \quad (2.3-25)$$

If the input to the plant assumes the form,

$$u_p(t) = u_p^*(t) + K_e (y_p^*(t) - y_p(t)), \quad (2.3-26)$$

where K_e is the stabilizing output feedback gain, then the CGT control becomes

$$u_p(t) = u_p^*(t) + K_e C_p e(t). \quad (2.3-27)$$

We substitute this control signal into Eq. (2.3-25) such that,

$$\dot{e}(t) = (A_p - B_p K_e C_p) e(t), \quad (2.3-28)$$

and see that the error asymptotically approaches zero provided that the K_e creates stable closed-loop error dynamics. The eigenstructure of Eq. (2.3-28) dictates the rate at which the plant's output converges to that of the model. Finally, we can write the CGT control as,

$$u_p(t) = S_{21} x_m(t) + S_{22} u_m + K_e C_p e(t). \quad (2.3-29)$$

In Fig. 2-4, we illustrate this model reference control scheme.

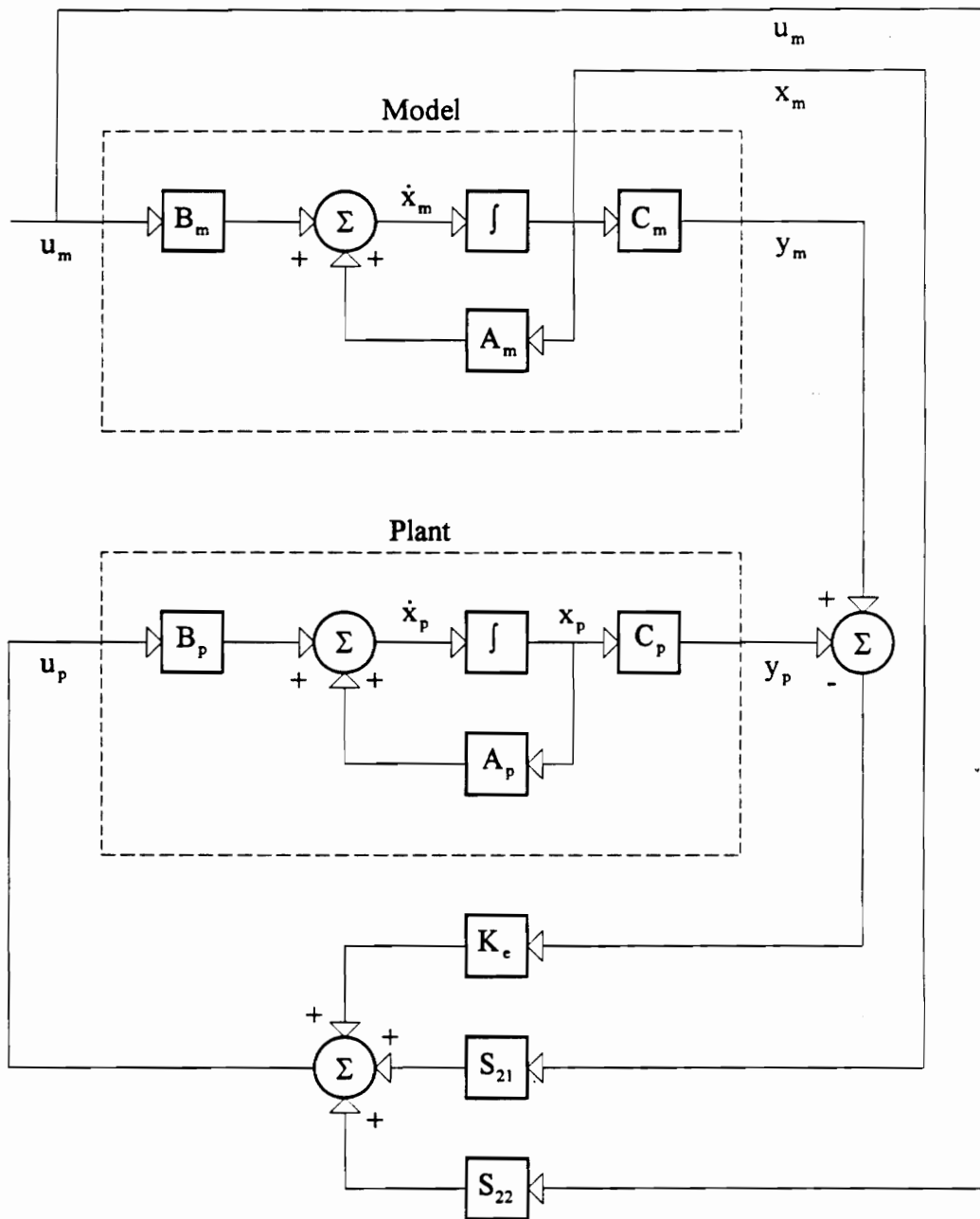


Figure 2-4
Block Diagram of a Continuous Model Reference Control System Using the CGT Approach

2.3.2 Discrete Case

Before we commence the discrete CGT formulation, we must mention to the reader that we were unable to locate this information in any literature. In his Ph.D. thesis, Sobel (1980) briefly addresses the issue. He only provides results analogous to Eq. (2.3-19):

$$S_{11} = \Omega_{11}S_{11}(\Phi_m - I) + \Omega_{12}C_m, \quad (2.3-30a)$$

$$S_{12} = \Omega_{11}S_{11}\Gamma_m, \quad (2.3-30b)$$

$$S_{21} = \Omega_{21}S_{11}(\Phi_m - I) + \Omega_{22}C_m \quad (2.3-30c)$$

and

$$S_{22} = \Omega_{21}S_{11}\Gamma_m \quad (2.3-30d)$$

where

$$\Omega = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} = \begin{bmatrix} \Phi_p - I & \Gamma_p \\ C_p & 0 \end{bmatrix}^{-1}. \quad (2.3-31)$$

We infer from these equations, by inspecting Ω , that

$$\begin{bmatrix} S_{11}(\Phi_m - I) & S_{11}\Gamma_m \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} \Phi_p - I & \Gamma_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}. \quad (2.3-32)$$

Sobel does not explain why he includes the identity matrix. If we examine the term in the first row, first column of this matrix equation,

$$S_{11}\Phi_m - S_{11} = \Phi_p S_{11} - S_{11} + \Gamma_p S_{21}, \quad (2.3-33a)$$

we see that the effect of the identity matrix in Eq. (2.3-32) is trivial. The other possible purpose for this matrix is in the calculation of,

$$S_{11}\Gamma_m = \Phi_p S_{12} - S_{12} + \Gamma_p S_{22}. \quad (2.3-33b)$$

Let's consider the term S_{12} as it applies to

$$0 = C_p S_{12}. \quad (2.3-33c)$$

If the plant output matrix is fully populated, then S_{12} must equal zero and there is no purpose for the identity matrix in Eq. (2.3-32). If the plant output matrix is not fully populated then S_{12} may be selected to solve Eq. (2.3-33b). It is possible that including the identity matrix helps to prevent non-singularities during the solution process. However, if its purpose is more significant than that, Sobel makes no indication. In our treatment of the discrete CGT controller, we create our own formulation, following the logical progression of the continuous CGT case.

Like the continuous CGT development, the ideal plant trajectories satisfy the state equations specified by the definition from Eq. (2.3-9).

$$\begin{bmatrix} x_p^*(k+1) \\ y_p^*(k) \end{bmatrix} = \begin{bmatrix} \Phi_p & \Gamma_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x_p^*(k) \\ u_p^*(k) \end{bmatrix}. \quad (2.3-34)$$

Substituting Eq. (2.3-10) into Eq. (2.3-34) yields,

$$\begin{bmatrix} x_p^*(k+1) \\ y_p^*(k) \end{bmatrix} = \begin{bmatrix} \Phi_p & \Gamma_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} x_m(k) \\ u_m(k) \end{bmatrix}. \quad (2.3-35)$$

A forward time step advance of $x_p^*(k)$ in Eq. (2.3-2) produces,

$$x_p^*(k+1) = S_{11}x_m(k+1) + S_{12}u_m(k+1). \quad (2.3-36)$$

Note that the input to the model, $u_m(k)$, is constant and may not be eliminated from this expression as it was in the continuous case. However, the model input has the property,

$$u_m(k+1) = u_m(k) \quad \forall k \geq 0. \quad (2.3-37)$$

This property allows the ideal state equation to be described as

$$x_p^*(k+1) = S_{11}x_m(k+1) + S_{12}u_m(k). \quad (2.3-38)$$

We substitute the model's state dynamics expression for $x_m(k+1)$ to produce

$$x_p^*(k+1) = S_{11}\Phi_m x_m(k) + (S_{11}\Gamma_m + S_{12})u_m(k). \quad (2.3-39)$$

We augment this equation with ideal discrete plant output equation, creating a form similar to Eq. (2.3-35).

$$\begin{bmatrix} x_p^*(k+1) \\ y_p^*(k) \end{bmatrix} = \begin{bmatrix} S_{11}\Phi_m & S_{11}\Gamma_m + S_{12} \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m(k) \\ u_m(k) \end{bmatrix} \quad (2.3-40)$$

Comparing this expression to Eq. (2.3-35) yields,

$$\begin{bmatrix} S_{11}\Phi_m & S_{11}\Gamma_m + S_{12} \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} \Phi_p & \Gamma_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}. \quad (2.3-41)$$

From this equation, we can extract four matrix equations and four matrix unknowns:

$$S_{11}\Phi_m = \Phi_p S_{11} + \Gamma_p S_{21}; \quad (2.3-42a)$$

$$S_{11}\Gamma_m + S_{12} = \Phi_p S_{12} + \Gamma_p S_{22}; \quad (2.3-42b)$$

$$C_m = C_p S_{11}; \quad (2.3-42c)$$

and

$$0 = C_p S_{12}. \quad (2.3-42d)$$

The reader may note that these equations are equivalent to those produced by Sobel (1980) in Eq. (2.3-33). The difference is that we do not use the additional identity matrix.

Equation (2.3-42) may be solved element by element, but if the matrix

$\Omega = \begin{bmatrix} \Phi_p & \Gamma_p \\ C_p & 0 \end{bmatrix}^{-1}$ exists, then we can also use the following equations.

$$S_{11} = \Omega_{11}S_{11}\Phi_m + \Omega_{12}C_m \quad (2.3-43a)$$

$$S_{12} = \Omega_{11}S_{11}\Gamma_m + \Omega_{11}S_{12} \quad (2.3-43b)$$

$$S_{21} = \Omega_{21}S_{11}\Phi_m + \Omega_{22}C_m \quad (2.3-43c)$$

$$S_{22} = \Omega_{21}S_{11}\Gamma_m + \Omega_{21}S_{12} \quad (2.3-43d)$$

The solution of these unknowns provides the discrete CGT solution where the ideal plant input is defined to be

$$u_p^*(k) = S_{21}x_m(k) + S_{22}u_m(k). \quad (2.3-44)$$

By examining the error dynamics we can determine a control signal to guarantee asymptotic tracking of the model's output. Defining error as,

$$e(k) = x_p^*(k) - x_p(k), \quad (2.3-45)$$

we take a time step advance of this error equation to analyze its dynamics.

$$e(k+1) = x_p^*(k+1) - x_p(k+1) \quad (2.3-46)$$

Substituting the ideal plant and actual plant state equations,

$$e(k+1) = \Phi_p x_p^*(k) + \Gamma_p u_p^*(k) - \Phi_p x_p(k) - \Gamma_p u_p(k). \quad (2.3-47)$$

By the definition of error in Eq. (2.3-45), this becomes

$$e(k+1) = \Phi_p e(k) + \Gamma_p (u_p^*(k) - u_p(k)). \quad (2.3-48)$$

Selecting the control signal as,

$$u_p(k) = u_p^*(k) + K_e C_p e(k), \quad (4.3-49)$$

where K_e is the stabilizing output feedback gain, error dynamics can be written as,

$$e(k+1) = (\Phi_p - \Gamma_p K_e C_p) e(k). \quad (2.3-50)$$

We then select K_e to control the transients of error.

Substituting the definition of the ideal input into Eq. (4.3-49), produces the final form of the control signal,

$$u_p(k) = S_{21} x_m(k) + S_{22} u_m(k) + K_e C_p e(k). \quad (2.3-51)$$

2.3.3 Summary

Like pole placement, CGT design involves two types of gain selection. The first corresponds to calculating gains for the control of dominant eigenvalue locations. These are the “performance” gains. The second type of gain, the output feedback gain, is associated with the decay of error.

The CGT controller is dependent on the accuracy of the plant model. If plant modeling error occurs, the CGT controller does not perform as expected. In contrast to pole placement, the CGT law is partially open-loop. Thus, stability concerns are lessened. In Chapter 3, our stability analysis reveals this clearly.

2.4 Method III: Model Reference Adaptive Control

Model reference control using the CGT approach requires full knowledge of the plant's dynamics. In many instances, a process (plant) may be incorrectly modeled, or it may not be possible to quantify all high order dynamics. If either problem occurs, the performance of the controller will suffer. However, if a CGT controller can adapt, it becomes possible to achieve the design goals specified by the reference model. This is made possible by replacing the fixed gains, S_{21} and S_{22} , with time varying gains; this is the basis for Sobel's (1980) formulation of the model reference adaptive control algorithms. In this section, we review Sobel's MRAC formulations for three different cases, including one discrete case.

The first version (case I) of MRAC requires that the open-loop plant be strictly positive real (refer to Appendix C). This rigid constraint is relaxed by the second version of MRAC which compensates by using a feedthrough term. The discrete version (case III) is based upon the continuous second version. There is no true discrete equivalent to case I.

2.4.1 Continuous Case I

Version I of the continuous MRAC algorithm uses the plant input form from Eq. (2.3-29), the CGT control signal. We replace these constant gains with matrices of time-varying or "adaptive" gains.

$$u_p(t) = K_x(t)x_m(t) + K_u(t)u_m(t) + K_e(t)C_p e(t) \quad (2.4-1)$$

In Fig. 2-5, we provide a block diagram of this controller.

The adaptive gains are concatenated into an $m \times n_r$ matrix for convenience,

$$K_r(t) = [K_e(t) \quad K_x(t) \quad K_u(t)]. \quad (2.4-2)$$

Note that n_r is the total number of elements in the output, the model state vector and the model input. The corresponding trajectories are concatenated into an $n_r \times 1$ matrix,

$$r(t) = \begin{bmatrix} C_p e(t) \\ x_m(t) \\ u_m(t) \end{bmatrix}. \quad (2.4-3)$$

The control signal is then expressed as,

$$u_p(t) = K_r(t)r(t). \quad (2.4-4)$$

The adaptive gain $K_r(t)$ consists of integral and proportional components.

$$K_r(t) = K_p(t) + K_I(t) \quad (2.4-5)$$

The proportional gain dominates the early transient response while the integral gain grows, "learning" which gain values minimize error. We calculate these gains using,

$$K_p(t) = v(t)r^T(t)\bar{T}, \quad (2.4-6)$$

$$\dot{K}_I(t) = v(t)r^T(t)T, \quad (2.4-7)$$

and by designating an initial condition for the integral gain,

$$K_I(0) = K_{I0}. \quad (2.4-8)$$

The $n_r \times n_r$ matrices T and \bar{T} are diagonal "weighting" matrices. They place emphasis on gains associated with each vector in $r(t)$. We may express these weighting matrices as,

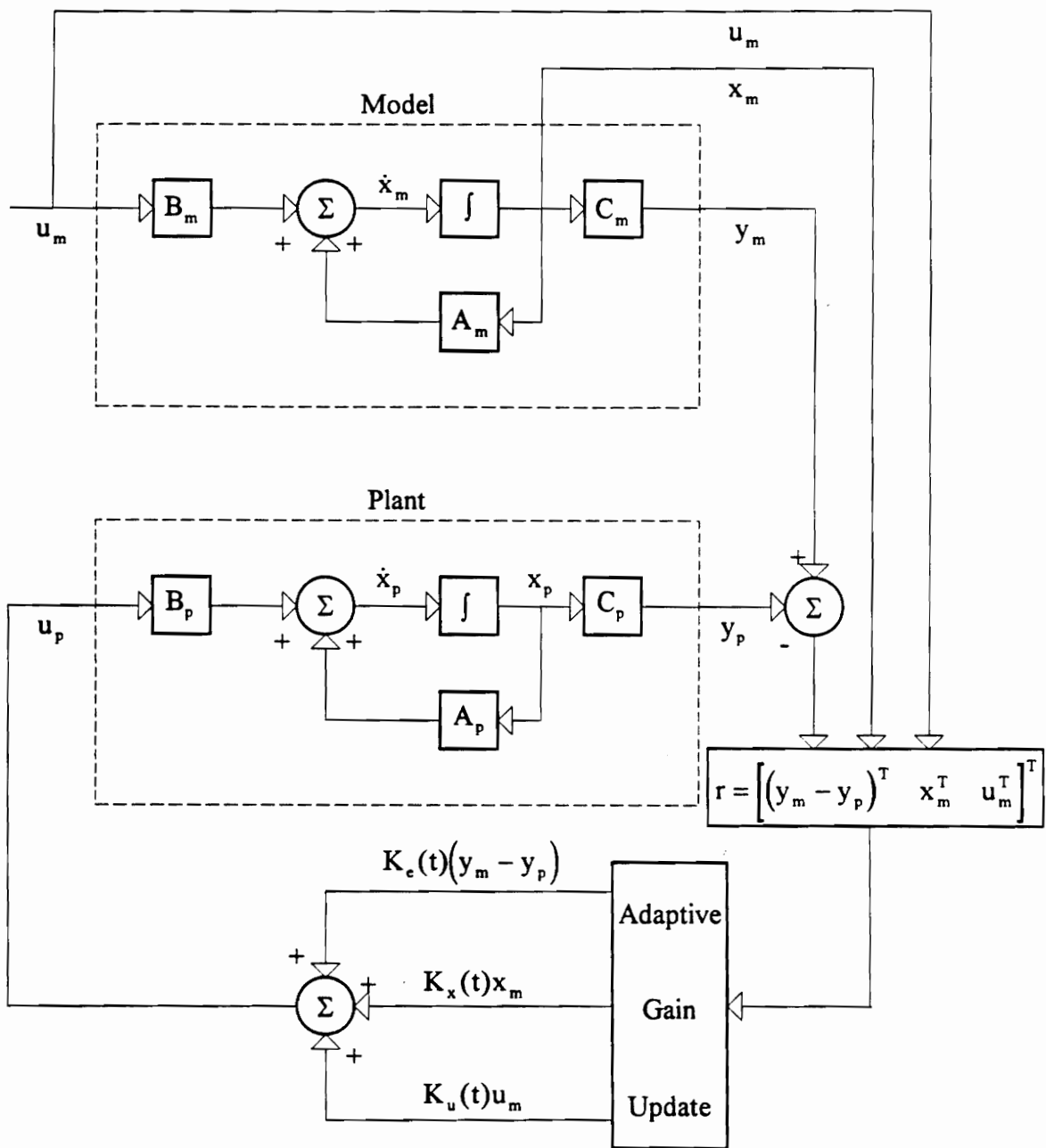


Figure 2-5
Block Diagram of the Continuous Model Reference Adaptive Control System

$$\bar{T} = \begin{bmatrix} \bar{T}_e & 0 & 0 \\ 0 & \bar{T}_x & 0 \\ 0 & 0 & \bar{T}_u \end{bmatrix} \quad (2.4-9a)$$

and

$$T = \begin{bmatrix} T_e & 0 & 0 \\ 0 & T_x & 0 \\ 0 & 0 & T_u \end{bmatrix}, \quad (2.4-9b)$$

where \bar{T}_e and T_e correspond to the output error, \bar{T}_x and T_x , correspond to the model's state vector, and \bar{T}_u and T_u correspond to the model's input.

As we mentioned previously, there are two different versions of the continuous algorithm. The distinction lies in the definition of the vector $v(t)$. In this first case, $v(t)$ is defined as the output error between the plant and the model.

$$v(t) = C_p e(t) = y_m(t) - y_p(t) \quad (2.4-10)$$

2.4.2 Continuous Case II

Version II of the continuous MRAC algorithm is identical to the first except for the specification of $v(t)$ in Eq. (2.4-10). In this second version, certain stability constraints are relaxed (discussed in Chapter 4), however, the algorithm requires some knowledge of the parameters that define the plant's dynamics. We define $v(t)$ as,

$$v(t) = QC_p e(t) + Gz(t), \quad (2.4-11)$$

where

$$z(t) = u_p^*(t) - u_p(t) + \tilde{K}_e C_p e(t). \quad (2.4-12)$$

In the last case, $v(t)$ was simply the output error. In case II, $v(t)$ is the error between the outputs, each of which involve a feedthrough term, $Gz(t)$. The vector $z(t)$ represents the error between the CGT control signal and the actual control signal. The need for a feedthrough term stems from a stability result to be covered in Chapter 4. The size and importance of G is addressed further in Chapter 5, our design chapter.

Unlike version I of the continuous algorithm, gains from the CGT controller appear in the definition of $z(t)$. The CGT terms require explicit knowledge of the plant's parameters. Requiring such knowledge can undermine the purpose of an adaptive controller. However, in Chapter 4, we demonstrate that only nominal and not exact values are required.

The reader may note that $u_p(t)$ is a function of $v(t)$ while simultaneously $v(t)$ is a function of $u_p(t)$. We may reconcile this dilemma by solving for $u_p(t)$ in closed form. To accomplish this, we introduce a new vector, $v_1(t)$.

$$v_1(t) = QC_p e(t) + G(u_p^*(t) + \tilde{K}_c C_p e(t)) \quad (2.4-13)$$

Note that $v_1(t)$ consists of terms that are immediately available to the controller and are not recursively defined. The trajectory $v(t)$ is rewritten in terms of $v_1(t)$.

$$v(t) = v_1(t) - Gu_p(t) \quad (2.4-14)$$

Using $v_1(t)$, we express the control signal as,

$$u_p(t) = \left[(v_1(t) - Gu_p(t)) r^T(t) \bar{T} + K_I(t) \right] r(t). \quad (2.4-15)$$

Solving for $u_p(t)$ in closed form,

$$u_p(t) = (v_1(t)r^T(t)\bar{T} + K_I(t))r(t) - Gu_p(t)r^T(t)\bar{T}r(t).$$

Grouping all $u_p(t)$ terms on the left yields,

$$(I + r^T(t)\bar{T}r(t)G)u_p(t) = (v_1(t)r^T(t)\bar{T} + K_I(t))r(t).$$

By premultiplying both sides by $(I + r^T(t)\bar{T}r(t)G)^{-1}$, we arrive at

$$u_p(t) = (I + r^T(t)\bar{T}r(t)G)^{-1}(K_I(t) + v_1(t)r^T(t)\bar{T})r(t). \quad (2.4-16)$$

2.4.3 Discrete Case

The discrete control algorithm uses the same form as the second version of the continuous MRAC algorithm,

$$u_p(k) = K_x(k)x_m(k) + K_u(k)u_m(k) + K_e(k)C_p e(k). \quad (2.4-17)$$

The adaptive gains, as well as the trajectories, are concatenated into $m \times n_r$ and $n_r \times 1$ matrices, respectively.

$$K_r(k) = [K_e(k) \quad K_x(k) \quad K_u(k)] \quad (2.4-18)$$

$$r(k) = \begin{bmatrix} C_p e(k) \\ x_m(k) \\ u_m(k) \end{bmatrix} \quad (2.4-19)$$

We assume that the time lag associated with machine computation is negligible compared to the sampling period. The control input to the plant is simply expressed as

$$u_p(k) = K_r(k)r(k). \quad (2.4-20)$$

We base gain calculation on the following set of equations.

$$K_r(k) = K_p(k) + K_I(k) \quad (2.4-21)$$

$$K_p(k) = v(k)r^T(k)\bar{T} \quad (2.4-22)$$

$$K_1(k+1) = K_1(k) + v(k)r^T(k)T \quad (2.4-23)$$

$$K_1(0) = K_{10} \quad (2.4-24)$$

$$v(k) = Fe(k) + Gz(k) \quad (2.4-25)$$

$$z(k) = u_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k) \quad (2.4-26)$$

The terms of the gain calculation procedure are dimensioned as v ($q \times 1$), z ($m \times 1$), F ($q \times n$), G ($q \times m$), T ($n_r \times n_r$) and \bar{T} ($n_r \times n_r$). Note that the error term in Eq. (2.4-17) and Eq. (2.4-18) is defined by Eq. (2.2-22a) where

$$e(k) = x_p^*(k) - x_p(k).$$

Whereas the error expression indicates that the plant state vector must be available, the F matrix will prove to be populated identically to the plant output matrix. As a result, the error that is actually used in the algorithm pertains to output based error rather than state based error.

Note that $z(k)$ is a function of $u_p(k)$ while simultaneously, $u_p(k)$ is a function of $z(k)$. We treat this issue by defining the ($q \times 1$) vector, $v_1(k)$.

$$v_1(k) = Fe(k) + G(u_p^*(k) + \tilde{K}_e C_p e(k)) \quad (2.4-27)$$

Now, we can rewrite $v(k)$.

$$v(k) = v_1(k) - Gu_p(k) \quad (2.4-28)$$

Expressing the control signal in terms of $v_1(k)$, we have,

$$u_p(k) = \left[(v_1(k) - Gu_p(k))r^T(k)\bar{T} + K_1(k) \right] r(k). \quad (2.4-29)$$

This equation is then solved for $u_p(k)$ in closed form.

$$u_p(k) = \left(I + r^T(k)\bar{T}r(k)G \right)^{-1} \left(K_1(k) + v_1(k)r^T(k)\bar{T} \right) r(k) \quad (2.4-30)$$

2.5 Model Reference Adaptive Control with a Time-Invariant Output Feedback Gain

Since the output error feedback gain is not calculated to influence the intended positions of eigenvalues, it does not need to adapt like the performance gains. Recall that the CGT control law only specifies that K_e produce a stable transfer function. With only a few minor changes in the fully adaptive MRAC algorithm, we can make the output feedback gain constant.

2.5.1 Continuous Case I

We modify the plant's control signal so that,

$$u_p(t) = K_x(t)x_m(t) + K_u(t)u_m(t) + K_e C_p e(t). \quad (2.5-1)$$

Again, for simplicity, the adaptive gains and trajectories are concatenated into the following matrices:

$$K_r(t) = \begin{bmatrix} K_x(t) & K_u(t) \end{bmatrix} \quad (2.5-2)$$

and

$$r(t) = \begin{bmatrix} x_m(t) \\ u_m(t) \end{bmatrix}. \quad (2.5-3)$$

With the control signal expressed as

$$u_p(t) = K_r(t)r(t) + K_e C_p e(t), \quad (2.5-4)$$

the calculation of gains corresponds to section 2.4.1 with appropriate dimensional changes in the weighting matrices (e.g., omit T_e).

2.5.2 Continuous Case II

The plant's control signal assumes the same form as version I where the adaptive gains and trajectories are concatenated. In Eq. (2.4-11), the definition of $z(t)$, K_e replaces \tilde{K}_e .

$$z(t) = u_p^*(t) - u_p(t) - K_e C_p e(t) \quad (2.5-5)$$

We introduce a new term $u_{p1}(t)$ ($m \times 1$) to represent the adaptive portion of the plant's control signal such that

$$u_p(t) = u_{p1}(t) + K_e C_p e(t) \quad (2.5-6)$$

where

$$u_{p1}(t) = K_r(t)r(t). \quad (2.5-7)$$

Equation (2.5-5) becomes

$$z(t) = u_p^*(t) - u_{p1}(t). \quad (2.5-8)$$

Recall that

$$v(t) = Q C_p e(t) + G z(t), \quad (2.4-10)$$

such that if

$$v_1(t) = Q C_p e(t) + G u_p^*(k), \quad (2.5-9)$$

then

$$v(t) = v_1(t) - Gu_{p1}(t). \quad (2.5-10)$$

The adaptive portion of the control signal may then be expressed using the vector, $v_1(t)$.

$$u_{p1}(t) = \left[(v_1(t) - Gu_{p1}(t))r^T(t)\bar{T} + K_I(t) \right] r(t) \quad (2.5-11)$$

This expression is manipulated into a closed-form solution for the vector, $u_{p1}(t)$.

$$u_{p1}(t) = \left(I + r^T(t)\bar{T}r(t)G \right)^{-1} \left(v_1(t)r^T(t)\bar{T} + K_I(t) \right) r(t) \quad (2.5-12)$$

Substituting Eq. (2.5-12) into Eq. (2.5-6) results in the final expression for the control signal.

$$u_p(t) = \left(I + r^T(t)\bar{T}r(t)G \right)^{-1} \left(v_1(t)r^T(t)\bar{T} + K_I(t) \right) r(t) + K_e C_p e(t) \quad (2.5-13)$$

2.5.3 Discrete Case

This formulation parallels the previous section. The discrete equivalents to the concatenations of Eq. (2.5-2) and Eq. (2.5-3) are established first. Next, the matrix \tilde{K}_e is replaced by the fixed gain, K_e , in the expression for $z(k)$,

$$z(k) = u_p^*(k) - u_p(k) + K_e C_p e(k). \quad (2.5-14)$$

With the above changes, the control signal becomes

$$u_p(k) = K_r(k)r(k) + K_e C_p e(k). \quad (2.5-15)$$

Introducing $u_{p1}(k)$ to represent the adaptive portion of the plant's control signal, we can write the total control as,

$$u_p(k) = u_{p1}(k) + K_e C_p e(k) \quad (2.5-16)$$

where

$$u_{p1}(k) = K_r(k)r(k). \quad (2.5-17)$$

Equation (2.5-14) becomes

$$z(k) = u_p^*(k) - u_{p1}(k). \quad (2.5-18)$$

Recall that

$$v(k) = Fe(k) + Gz(k) \quad (2.4-25)$$

such that if

$$v_1(k) = Fe(k) + Gu_p^*(k) \quad (2.5-19)$$

then

$$v(k) = v_1(k) - Gu_{p1}(k). \quad (2.5-20)$$

The adaptive portion of the control signal can be expressed using $v_1(k)$.

$$u_{p1}(k) = \left[(v_1(k) - Gu_{p1}(k))r^T(k)\bar{T} + K_1(k) \right] r(k) \quad (2.5-21)$$

We then solve for $u_{p1}(k)$ in closed form:

$$u_{p1}(k) = \left(I + r^T(k)\bar{T}r(k)G \right)^{-1} \left(v_1(k)r^T(k)\bar{T} + K_1(k) \right) r(k). \quad (2.5-22)$$

Substituting Eq. (2.5-22) into Eq. (2.5-16) results in the final expression for the control signal,

$$u_p(k) = \left(I + r^T(k)\bar{T}r(k)G \right)^{-1} \left(v_1(k)r^T(k)\bar{T} + K_1(k) \right) r(k) + K_e C_p e(k). \quad (2.5-23)$$

2.6 Summary

In this chapter, we formulated four types of algorithms in both the continuous and discrete domains: pole placement; model reference control using the CGT approach;

model reference adaptive control; and MRAC using a fixed output feedback gain. Aside from the discrete CGT controller, this chapter reviewed unoriginal work.

In subsequent chapters, we examine the stability of each algorithm to establish constraints on design variables (e.g.: the weighting matrices T and \bar{T}). Following the stability analysis, we examine the implications of the design variables on performance.

Chapter 3

Analyzing Linear Control Algorithm Stability

In this chapter, we discuss the stability of each linear control approach. Since the pole placement and CGT controllers are linear and time-invariant, we can analyze these controllers in the frequency domain. We develop sensitivity functions to contrast model reference control with pole placement. Analyzing sensitivity provides insight into relative stability. In Chapter 6, we will empirically examine the robustness of these two controllers.

3.1 Evaluating Relative Stability

In this section, we review a common stability analysis technique. Most of this information may be found in Wolovich (1994), Friedland (1986), and Anderson (1990).

The sensitivity transfer function reflects a system's ability to remain stable when the plant is modeled incorrectly. Sensitivity is a function of loop gain. To evaluate stability, we must first derive the loop gain. Before we explicitly discuss loop gain, it helps to understand the return difference. The return difference is essentially the denominator of a closed-loop transfer function. Let's consider a simple example to illustrate this concept. Given a plant's input-output transfer function,

$$G_p(s) = \frac{y_p(s)}{u_p(s)} = C_p [sI - A_p]^{-1} B_p,$$

and control using an output feedback compensator,

$$H_p(s) = -K,$$

the control signal is given by,

$$u_p(t) = c_p(t) - Ky_p(t),$$

where c_p is a command input. We calculate the closed-loop transfer function as,

$$G_{pcl}(s) = \frac{y_p(s)}{c_p(s)} = \frac{G_p(s)}{1 + H_p(s)G_p(s)}.$$

By definition (Maciejowski, 1989), the return difference is

$$1 + H_p(s)G_p(s).$$

We then define the loop gain, $L(s)$, as the return difference minus 1 or,

$$L(s) = H_p(s)G_p(s).$$

In our study, the loop gain is simply the product of all transfer functions in a loop. To calculate the loop gain, we usually “break” the path of an input or output and then calculate the product of all transfer functions and gains from the beginning of the break to the end.

Using the loop gain, we can form the sensitivity transfer function which is the inverse of the return difference. This function measures a controller’s sensitivity to a plant’s incorrectly modeled dynamics. The less “sensitive” a controlled plant is, the greater its relative stability will be. We relate sensitivity, S , to the loop gain by,

$$S(s) = \frac{1}{1 + L(s)} \quad (3.1-1)$$

In a plot of $L(j\omega)$ (a Nyquist plot), we can see that the magnitude of the return difference, $|1 + L(j\omega)|$, equals the distance from the -1 point to $L(j\omega)$. The minimum distance, between the -1 point and $L(j\omega)$, represents the margin of stability at a given frequency point (Wolovich, 1994). This technique considers both phase and gain margins. The stability margin may be determined using an identity developed from Eq. (3.1-1).

$$|1 + L(j\omega)| = |S(j\omega)|^{-1} \quad (3.1-2)$$

In this analysis, we seek a minimum value of $|S(j\omega)|^{-1}$ or, equivalently, a maximum of $|S(j\omega)|$. The “infinity norm” of $|S(j\omega)|$ represents the maximum sensitivity over all frequencies,

$$\|S\|_{\infty} = \max_{\omega} |S(j\omega)| = \bar{S}. \quad (3.1-3)$$

We define the stability margin, SM, as

$$SM = \bar{S}^{-1} = \min_{\omega} |1 + L(j\omega)|. \quad (3.1-4)$$

For the discrete controllers, the stability analysis follows the same procedure, except $z = e^{j\omega T_s}$, where T_s represents the sampling period.

3.2 Pole Placement

The continuous pole placement control law differs considerably from its discrete counterpart if state estimation is required. In this chapter, we only treat the cases requiring state estimation.

3.2.1 Continuous Case

The first step is to determine the loop gain, $L(s)$. In Fig. 3-1, we “break” the loop at the plant input, after the signal branches off to the observer.

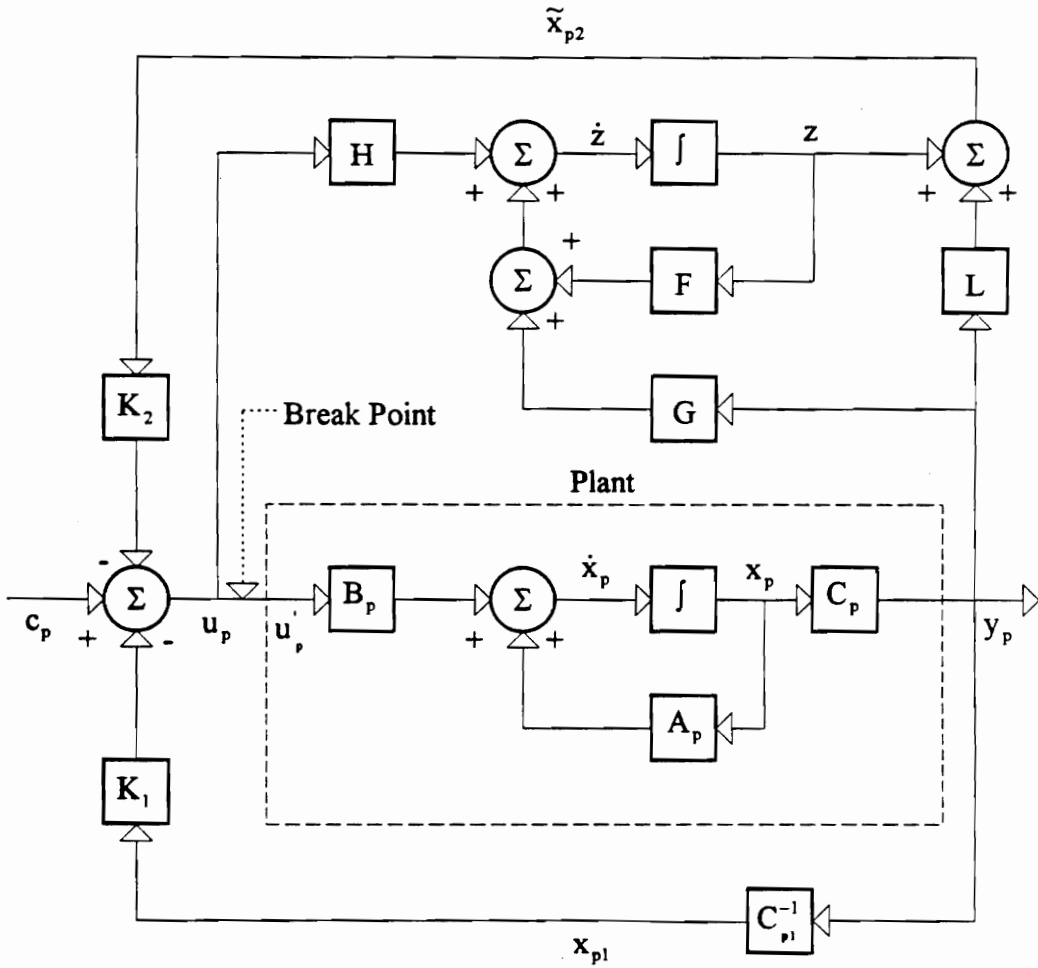


Figure 3-1
Break Point Location for a Continuous Pole Placement Control System with a Reduced Order Observer

We formulate the loop gain by considering u_p' the input, and u_p the output associated with the break point. Using this perspective, we can realize a system whose transfer function is the loop gain. Let's begin with the plant's state equations,

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p'(t) \quad (3.2-1)$$

and

$$y_p(t) = C_p x_p(t). \quad (2.1-2)$$

Now, we consider the dynamics of the trajectory $z(t)$. Recall that,

$$z(t) = \tilde{x}_{p2}(t) - L y_p(t) \quad (2.2-5)$$

and

$$\dot{z}(t) = F z(t) + G y_p(t) + H u_p(t). \quad (2.2-6)$$

Defining the vector $u_p(t)$ (the “output” in this scenario) as,

$$\begin{aligned} u_p(t) &= -K_1 x_{p1}(t) - K_2 \tilde{x}_{p2}(t) \\ &= -K_1 C_{p1}^{-1} C_p x_p(t) - K_2 (z(t) + L C_p x_p(t)), \\ &= -(K_1 C_{p1}^{-1} + K_2 L) C_p x_p(t) - K_2 z(t) \end{aligned} \quad (3.2-2)$$

we can write Eq. (2.2-6) as,

$$\dot{z}(t) = (F - H K_2) z(t) + [G - H (K_1 C_{p1}^{-1} + K_2 L)] C_p x_p(t). \quad (3.2-3)$$

We augment the $x_p(t)$ and $z(t)$ trajectories to form a state space realization of the loop gain:

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ [GC_{p1}^{-1} - H(K_1 C_{p1}^{-1} + K_2 L)]C_p & F - HK_2 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u_p'(t) \quad (3.2-4)$$

and

$$u_p(t) = -\left[(K_1 C_{p1}^{-1} + K_2 L)C_p \quad K_2\right] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}. \quad (3.2-5)$$

In the Laplace domain, loop gain is given by,

$$L(s) = -\left[(K_1 C_{p1}^{-1} + K_2 L)C_p \quad K_2\right] \begin{bmatrix} sI - A_p & 0 \\ -[GC_{p1}^{-1} - H(K_1 C_{p1}^{-1} + K_2 L)]C_p & sI - F + HK_2 \end{bmatrix}^{-1} \begin{bmatrix} B_p \\ 0 \end{bmatrix}. \quad (3.2-6)$$

3.2.2 Discrete Case

Developing $L(z)$ differs from the continuous case because of an alternative observer formulation. In Fig. (3-2), however, we break the control signal in a similar fashion. Again, we formulate the loop gain by considering u_p' the input, and u_p the output associated with the break point. Using this perspective, we can realize a system whose transfer function is the loop gain. Let's begin with the plant's state equations. From equations (2.1-3) and (2.1-4),

$$zx_p(z) = \Phi_p x_p(z) + \Gamma_p u_p'(z) \quad (3.2-7)$$

and

$$y_p(z) = C_p x_p(z). \quad (3.2-8)$$

Now, we consider the dynamics of the trajectory $\tilde{x}_{p2}(z)$. Recall that,

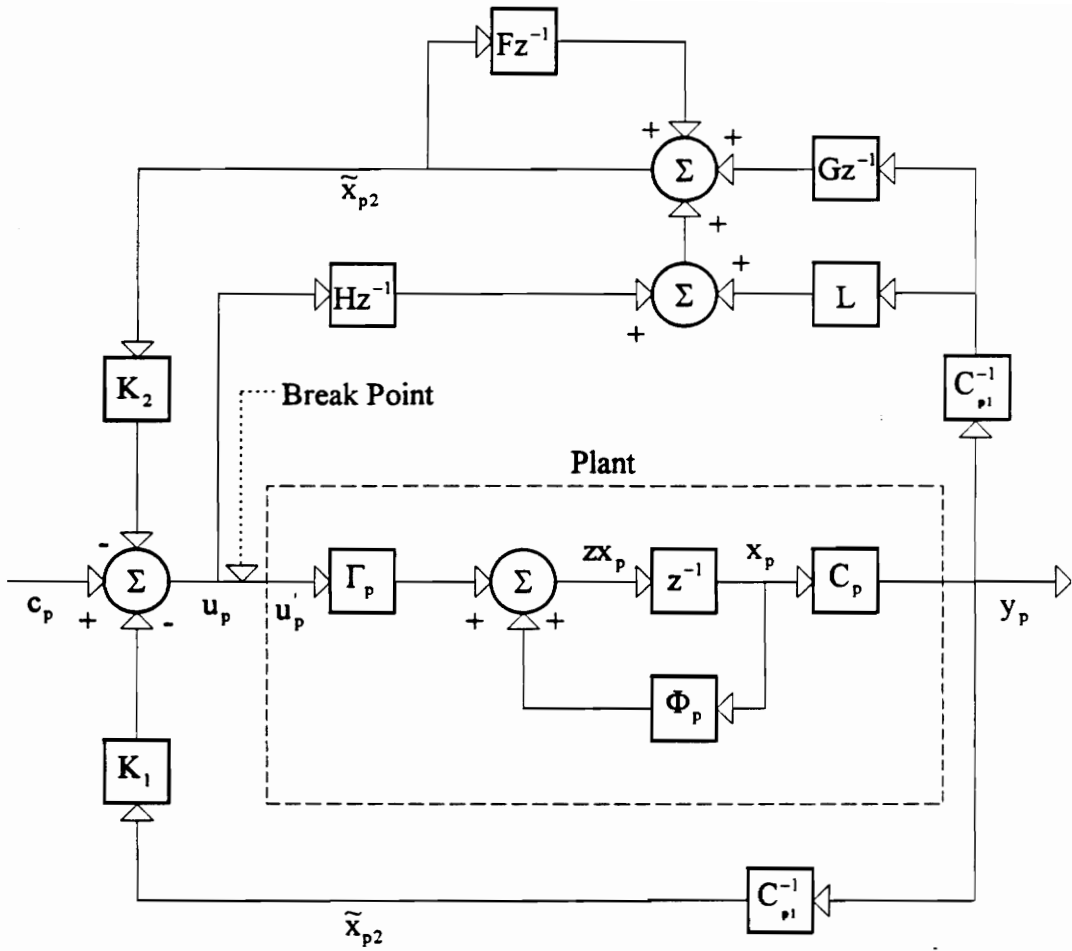


Figure 3-2
Break Point Location for a Discrete Pole Placement Control System with a
Reduced Order Observer

$$z\tilde{x}_{p2}(z) = F\tilde{x}_{p2}(z) + Gx_{p1}(z) + Hu_p(z) + Lzx_{p1}(z). \quad (2.2-39)$$

Since

$$x_{p1}(z) = C_{p1}^{-1}y_p(z) = C_{p1}^{-1}C_p x_p(z) \quad (3.2-9)$$

and

$$zx_{p1}(z) = C_{p1}^{-1}C_p zx_p(z) = C_{p1}^{-1}C_p \Phi_p x_p(z) + C_{p1}^{-1}C_p \Gamma_p u'_p(z), \quad (3.2-10)$$

we can write Eq. (2.2-39) as,

$$z\tilde{x}_{p2}(z) = F\tilde{x}_{p2}(z) + \left(GC_{p1}^{-1}C_p + LC_{p1}^{-1}C_p\Phi_p\right)x_p(z) + Hu_p(z) + LC_{p1}^{-1}C_p\Gamma_p u'_p(z). \quad (3.2-11)$$

Now, we derive the expression for $u_p(z)$.

$$u_p(z) = -K_1 x_{p1}(z) - K_2 \tilde{x}_{p2}(z) = -K_1 C_{p1}^{-1}C_p x_p(z) - K_2 \tilde{x}_{p2}(z) \quad (3.2-12)$$

Substituting this result into Eq. (3.2-11) yields,

$$z\tilde{x}_{p2}(z) = (F - HK_2)\tilde{x}_{p2}(z) + \left(GC_{p1}^{-1}C_p + LC_{p1}^{-1}C_p\Phi_p - HK_1 C_{p1}^{-1}C_p\right)x_p(z) + LC_{p1}^{-1}C_p\Gamma_p u'_p(z) \quad (3.2-13)$$

We augment the $x_p(z)$ and $\tilde{x}_{p2}(z)$ trajectories to form a state space realization of the loop gain. Note that the equations are converted to the time step description.

$$\begin{bmatrix} x_p(k+1) \\ \tilde{x}_{p2}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_p & 0 \\ GC_{p1}^{-1}C_p + LC_{p1}^{-1}C_p\Phi_p - HK_1 C_{p1}^{-1}C_p & F - HK_2 \end{bmatrix} \begin{bmatrix} x_p(k) \\ \tilde{x}_{p2}(k) \end{bmatrix} + \begin{bmatrix} \Gamma_p \\ LC_{p1}^{-1}C_p\Gamma_p \end{bmatrix} u'_p(k) \quad (3.2-14)$$

$$u_p(k) = -\begin{bmatrix} K_1 C_{p1}^{-1}C_p & K_2 \end{bmatrix} \begin{bmatrix} x_p(k) \\ \tilde{x}_{p2}(k) \end{bmatrix} \quad (3.2-15)$$

In the discrete domain, loop gain is given by,

$$L(z) = -\begin{bmatrix} K_1 C_{p1}^{-1}C_p & K_2 \end{bmatrix} \begin{bmatrix} zI - \Phi_p & 0 \\ -GC_{p1}^{-1}C_p - LC_{p1}^{-1}C_p\Phi_p + HK_1 C_{p1}^{-1}C_p & zI - F + HK_2 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_p \\ LC_{p1}^{-1}C_p\Gamma_p \end{bmatrix} \quad (3.2-16)$$

3.2 Command Generator Tracker

In a manner similar to that of section 3.1, we calculate loop gain as the first step to evaluate relative stability. Breaking the control signal in Fig. 2-4 produces the branches shown in Fig. 3-3. Due to similarities between the continuous and discrete cases, we calculate the loop gains in identical fashions.

The model's dynamic states and input do not contribute to the return difference equation because they are not in any feedback loop. In a stable system, a bounded input

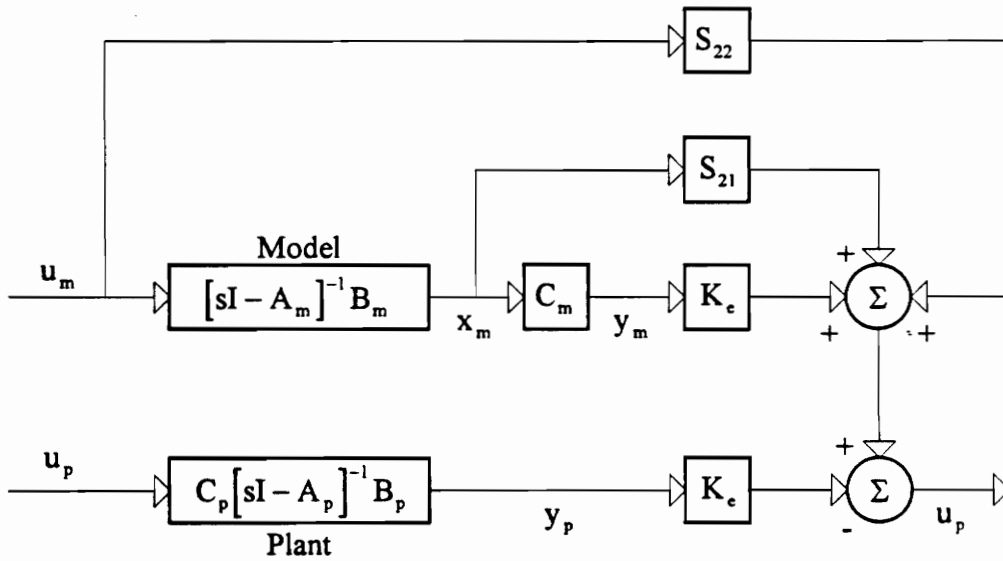


Figure 3-3
Broken Loop of the Continuous CGT Control (unsimplified form)

produces a bounded output. The contribution of the model is strictly open-loop. The model's input, state, and output trajectories are constrained to be stable (hence a bounded input). Therefore, the choices of S_{21} and S_{22} have no bearing on the stability of the closed-loop system. We begin by eliminating all signal paths involving $y_m(\cdot)$, $x_m(\cdot)$, and $u_m(\cdot)$. The

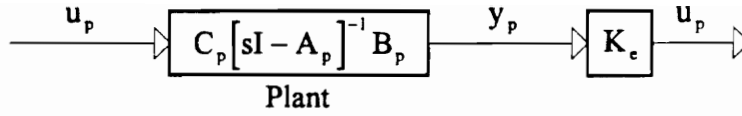


Figure 3-4
Return Difference of Continuous Time CGT Control
(Simplified Form)

contribution from the reference model may be treated as a disturbance. This simplifies the block diagram in Fig. 3-3.

The resulting loop gains for both the continuous and discrete time cases are based on Fig. 3-4. As the following equations indicate, stability is only a function of the output error feedback gain.

Continuous Case:

$$L(s) = K_e C_p [sI - A_p]^{-1} B_p \quad (3.2-1)$$

Discrete Case:

$$L(z) = K_e C_p [zI - \Phi_p]^{-1} \Gamma_p \quad (3.2-2)$$

Since the error feedback gain is not used in the specification of closed-loop performance, the concern of robustness is minimal. Performance and absolute stability are essentially separate issues wherein addressing one issue does not come at the expense of the other.

3.3 Summary

In this chapter, we presented a method of evaluating the relative stability of the linear controllers. From this treatment, we see that the stability of pole placement depends

on its performance gains and its estimator gains. In contrast, the CGT control only has one gain matrix associated with stability. More importantly, this CGT gain does not affect the placement of dominant eigenvalues.

When we compare these algorithms in Chapter 6, we refer to this development to explain trends in robustness. In the next chapter, we treat the stability of the non-linear control algorithms which requires a very different approach.

Chapter 4

Analyzing Stability of Model Reference Adaptive Control

The frequency domain approach to stability analysis applies only to LTI systems. Model reference adaptive control requires a different approach because such algorithms create nonlinear and time-variant closed-loop systems. Lyapunov's direct method (also called Lyapunov's second method) provides a means to determine the absolute stability of many systems (linear and nonlinear). The findings of the analysis provide design guidelines and constraints on MRAC terms such as the weighting matrices.

Using Lyapunov's direct method, we derive the conditions that are sufficient to guarantee stability. If we violate one of these conditions, our system does not necessarily become unstable. However, we adhere to the guidelines prescribed by the stability analysis even though violating a stability constraint does not always lead to instability.

In addition to deriving conditions for absolute stability, we prove that the adaptive integral gains are bounded. A control system is not practical if it requires infinite gains.

Our review of Lyapunov stability comes predominantly from Sobel (1980). Related information is also documented by Sobel *et al.* (1986) and Kaufman *et al.* (1994). Although the essence of this chapter is unoriginal, we have added many intermediate steps and explanations not found in any of our references. Additionally, we discuss strictly positive real systems which often pertain to the linear subsystems found in Lyapunov equations.

If the reader is unfamiliar with Lyapunov's direct method we suggest Vidyasagar (1993) and Slotine (1991) as excellent sources of information. Briefly and qualitatively, we explain the general idea behind a Lyapunov stability analysis. We chose a Lyapunov function, based upon trajectories of a system, to be positive definite. The minimum possible value of this function represents the designer's goal. Placing the trajectories into quadratic forms is an easy way to make the function positive definite. Let's assume that our function happens to represent energy. If the first derivative of this Lyapunov function, with respect to its trajectories, is negative definite, then the energy is dissipative for the continuous case. If our system is discrete, the time step difference of the Lyapunov function must be negative definite. Satisfying these constraints, the energy function approaches a minimum value. If the function is a quadratic of state variables, then the system reaches a finite steady-state and is stable.

Sobel (1980) selects an error-based positive definite Lyapunov function, V . For the continuous case, the first derivative of the Lyapunov function must be negative definite to guarantee that output error is bounded. To accomplish this for a discrete case, the quantity $[V(k+1) - V(k)]$ must be negative definite. This examination of stability is crucial for synthesizing each MRAC algorithm.

As a final note, we assume that the plant's dynamic parameters do not vary over the time of response (the plant is autonomous). Although the vast majority of processes in this world are indeed non-autonomous, we assume that any parameter changes occurring within the plant are negligible.

4.1 Continuous Case I

In this section, we examine the stability of MRAC case I. We formulate the Lyapunov function and its first derivative. Afterwards, we discuss how to satisfy the constraints implied by the Lyapunov functions.

We preface this development with an explanation of terminology: V is the Lyapunov function and \dot{V} is the first time derivative of the Lyapunov function. Together, we call them the “Lyapunov equations” that correspond to a particular system.

4.1.1 Formulating the Lyapunov Functions

First, we establish the Lyapunov function (Sobel, 1980), a quadratic function of $e(t)$ and $K_I(t)$.

$$V(e(t), K_I(t)) = e^T(t)Pe(t) + \text{trace} \left[S(K_I(t) - \tilde{K})T^{-1}(K_I(t) - \tilde{K})^T S^T \right] \quad (4.1-1)$$

The terms of this function are constrained such that T^{-1} ($n_r \times n_r$) is positive definite, P ($n \times n$) is positive definite, \tilde{K} ($m \times n_r$) is time invariant and arbitrarily chosen, and S ($m \times m$) is non-singular (invertible). We constrained P and T^{-1} to be positive definite because they are the only matrices that are not in a quadratic form. Note that we use the definition of error given by,

$$e(t) = x_p^*(t) - x_p(t). \quad (2.3-22)$$

The matrix \tilde{K} is termed the “dummy” gain matrix. Its value is constant, arbitrary, and may be assigned for convenience. The dummy gain matrix serves a purely symbolic

purpose in this mathematical development³. This matrix is partitioned in the same fashion as $K_I(t)$,

$$\tilde{K} = \begin{bmatrix} \tilde{K}_e & \tilde{K}_x & \tilde{K}_u \end{bmatrix}.$$

The Lyapunov function in Eq. (4.1-1) is positive for all non-zero values of $e(t)$ and of the quantity, $(K_I(t) - \tilde{K})$. The function V may attain the value of zero provided that the error is zero and that $K_I(t)$ reaches a bounded steady-state value equal to that of \tilde{K} . It is possible, however, that the Lyapunov function achieves a non-zero steady state value - when the error reaches zero, $K_I(t)$ may not equal \tilde{K} . This is not a problem because we only need to prove that the adaptive integral gain is bounded.

To ensure that $V(e, K_I)$ approaches a minimum (implying minimal error and a bounded gain), its first derivative must be negative definite. The first derivative of the Lyapunov function with respect to time is,

$$\dot{V} = \dot{e}^T(t)Pe(t) + e^T(t)P\dot{e}(t) + \text{trace}\left[S\dot{K}_I(t)T^{-1}(K_I(t) - \tilde{K})^T S^T\right] + \text{trace}\left[S(K_I(t) - \tilde{K})T^{-1}\dot{K}_I^T(t)S^T\right]$$

which becomes

$$\dot{V} = \dot{e}^T(t)Pe(t) + e^T(t)P\dot{e}(t) + 2\text{trace}\left[S(K_I(t) - \tilde{K})T^{-1}\dot{K}_I^T(t)S^T\right]. \quad (4.1-2)$$

We proceed to develop the error derivative in this equation. Recall the definition of error,

³ This statement may be difficult to accept. Admittedly, the dummy gain matrix appears useless at this juncture. Selecting compatible Lyapunov functions is, however, a much more difficult endeavor than their subsequent interpretation (which is all we are doing). We speculate that the dummy gain matrix simply made finding compatible Lyapunov equations an easier task.

$$\dot{e}(t) = A_p e(t) + B_p (u_p^*(t) - u_p(t)). \quad (2.3-25)$$

We can expand the error equation by substituting for $u_p^*(t)$ and $u_p(t)$. Recall that,

$$u_p^*(t) = S_{21}x_m(t) + S_{22}u_m(t) \quad (2.3-20)$$

and from equations (2.4-5) and (2.4-6),

$$u_p(t) = K_I(t)r(t) + C_p e^T(t)\bar{T}r(t).$$

Placing these definitions into Eq. (2.3-25) yields,

$$\dot{e}(t) = A_p e(t) + B_p (S_{21}x_m(t) + S_{22}u_m(t) - K_I(t)r(t) - C_p e^T(t)\bar{T}r(t)). \quad (4.1-3)$$

The third term of \dot{V} is simplified by the definition of \dot{K}_I ,

$$\dot{K}_I(t) = C_p e(t)r^T(t)T, \quad (2.4-7)$$

such that,

$$2\text{trace}[S(K_I(t) - \tilde{K})T^{-1}\dot{K}_I^T(t)S^T] = 2\text{trace}[S(K_I(t) - \tilde{K})r(t)e^T(t)C_p^T S^T].$$

The matrix products $S(K_I(t) - \tilde{K})r(t)$ and $e^T(t)C_p^T S^T$ are dimensioned $(m \times 1)$ and $(1 \times m)$, respectively. Hence,

$$2\text{trace}[S(K_I(t) - \tilde{K})r(t)e^T(t)C_p^T S^T] = 2e^T(t)C_p^T S^T S(K_I(t) - \tilde{K})r(t). \quad (4.1-4)$$

Note that we eliminate the time functional notation (e.g., $e \Rightarrow e(t)$) for clarity in presentation. Applying the results of equations (4.1-3) and (4.1-4) to Eq. (4.1-2), we have

$$\begin{aligned}
\dot{V} = & e^T P \left(A_p e - B_p C_p e r^T \bar{T} r - B_p K_I r + B_p S_{21} x_m + B_p S_{22} u_m \right) \\
& + \left(A_p e - B_p C_p e r^T \bar{T} r - B_p K_I r + B_p S_{21} x_m + B_p S_{22} u_m \right)^T P e \\
& + 2e^T C_p^T S^T S (K_I - \tilde{K}) r
\end{aligned}$$

Regrouping terms in quadratic form, we can better assess the negative definite character of \dot{V} .

$$\begin{aligned}
\dot{V} = & e^T (P A_p + A_p^T P) e - e^T (P B_p C_p + C_p^T B_p^T P) e r^T \bar{T} r \\
& - 2e^T P B_p K_I r + 2e^T P B_p (S_{21} x_m + S_{22} u_m) \\
& + 2e^T C_p^T S^T S (K_I - \tilde{K}) r
\end{aligned}$$

We then combine similar terms to produce,

$$\begin{aligned}
\dot{V} = & e^T (P A_p + A_p^T P) e - e^T (P B_p C_p + C_p^T B_p^T P) e r^T \bar{T} r \\
& + 2e^T (C_p^T S^T S - P B_p) K_I r - 2e^T C_p^T S^T S \tilde{K} r \\
& + 2e^T P B_p (S_{21} x_m + S_{22} u_m)
\end{aligned} \tag{4.1-5}$$

Since term three of \dot{V} is not quadratic, we can not claim that it will be negative definite.

We equate this term to zero using by constraining the plant output matrix,

$$C_p = (S^T S)^{-1} B_p^T P. \tag{4.1-6}$$

This convenient constraint reduces Eq. (4.1-4) to

$$\begin{aligned}\dot{V} = & e^T (PA_p + A_p^T P)e - 2e^T PB_p (S^T S)^{-1} B_p^T P e r^T \bar{T} r \\ & - 2e^T PB_p \tilde{K} r + 2e^T PB_p (S_{21} x_m + S_{22} u_m)\end{aligned}\quad (4.1-7)$$

Let's consider the third term of this equation. We see that by similarities in the third and fourth terms, we can define the dummy gain to simplify \dot{V} further. Expanding $\tilde{K}r$ produces,

$$\begin{aligned}\tilde{K}r &= \tilde{K}_e (y_m - y_p) + \tilde{K}_x x_m + \tilde{K}_u u_m \\ &= \tilde{K}_e (C_p x_p^* - C_p x_p) + \tilde{K}_x x_m + \tilde{K}_u u_m \\ &= \tilde{K}_e C_p e + \tilde{K}_x x_m + \tilde{K}_u u_m\end{aligned}\quad (4.1-8)$$

Substituting this matrix product into \dot{V} , we arrive at,

$$\begin{aligned}\dot{V} = & e^T (PA_p + A_p^T P)e - 2e^T PB_p (S^T S)^{-1} B_p^T P e r^T \bar{T} r \\ & - 2e^T PB_p (\tilde{K}_e C_p e + \tilde{K}_x x_m + \tilde{K}_u u_m) \\ & + 2e^T PB_p (S_{21} x_m + S_{22} u_m)\end{aligned}$$

Combining the third and fourth terms results in,

$$\begin{aligned}\dot{V} = & e^T \left[P(A_p - B_p \tilde{K}_e C_p) + (A_p - B_p \tilde{K}_e C_p)^T P \right] e \\ & - 2e^T PB_p (S^T S)^{-1} B_p^T P e r^T \bar{T} r \\ & + 2e^T PB_p \left[(S_{21} - \tilde{K}_x) x_m + (S_{22} - \tilde{K}_u) u_m \right]\end{aligned}\quad (4.1-9)$$

The final term is not a quadratic function of any trajectories. We can eliminate this term by defining the dummy gains to be equal to the CGT gains:

$$\tilde{K}_x = S_{21} \quad (4.1-10a)$$

and

$$\tilde{K}_u = S_{22}. \quad (4.1-10b)$$

These constraints eliminate the last term of Eq. (4.1-9) while all remaining terms are in quadratic form.

$$\begin{aligned} \dot{V} = & e^T(t) \left[P(A_p - B_p \tilde{K}_e C_p) + (A_p - B_p \tilde{K}_e C_p)^T P \right] e(t) \\ & - 2e^T(t) P B_p (S^T S)^{-1} B_p^T P e(t) r^T(t) \bar{r}(t) \end{aligned} \quad (4.1-11)$$

Since all trajectories are in quadratic forms, they can not affect whether or not a term of \dot{V} is negative definite. To constraint \dot{V} to be negative definite, we restrict the time-invariant matrices. Let's begin with the first term of Eq. (4.1-11). If,

$$P(A_p - B_p \tilde{K}_e C_p) + (A_p - B_p \tilde{K}_e C_p)^T P < 0. \quad (4.1-12)$$

then the first term is negative definite. This expression contains no terms associated with adaptation. From this term, we may infer a linear subsystem. This allows us to apply positivity concepts (Anderson, 1967) in determining constraints necessary in proving \dot{V} to be negative definite.

Let's begin by demonstrating exactly how we infer a linear subsystem from term 1 of Eq. (4.1-11). Consider a plant, analogous to the plant in equations (2.1-1) and (2.1-2), described by,

$$\dot{x}(t) = Ax(t) + Bu(t)$$

and

$$y(t) = Cx(t).$$

Suppose the control law is such that

$$u(t) = -KCx(t),$$

a standard output feedback controller. We then write the closed-loop state equation as

$$\dot{x}(t) = (A - BKC)x(t).$$

To examine the stability of this linear system, we select the Lyapunov function as,

$$V(x) = x^T(t)Px(t),$$

where P is a positive definite matrix. Then, we calculate the first derivative.

$$\dot{V}(x) = \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t).$$

Into this expression, we substitute the closed-loop state equation for $\dot{x}(t)$.

$$\dot{V}(x) = x^T(t)[(A - BKC)^T P + P(A - BKC)]x(t)$$

The form of this equation parallels that of Eq. (4.1-11). Thus, we may treat term 1 of Eq. (4.1-11) as if it were a simple linear system, allowing us to apply positivity concepts.

Using the strict positive realness concept, we can restrict the open-loop plant so that the controlled plant is stable for any negative output feedback gain. If the realization

$\{(A_p - B_p \tilde{K}_e C_p), B_p, C_p\}$ is strictly positive real (SPR), then the first term of \dot{V} must be negative definite⁴. Since this realization involves a dummy gain, whose value we do not know, we consider the realization to be almost strictly positive real (ASPR) if $\{A_p, B_p, C_p\}$ is output stabilizable. The motivation for considering positivity concepts is best described by Kaufman *et al.* who say, “ASPR transfer functions remain stable for any high [negative] definite feedback, be it fixed or nonlinear and nonstationary.”

In Appendix C, we derive the SPR requirements for this MRAC case. These conditions are satisfied if, for a stabilizing feedback gain \tilde{K}_e , known to exist,

$$P(A_p - B_p \tilde{K}_e C_p) + (A_p - B_p \tilde{K}_e C_p)^T P < 0 \quad (4.1-12)$$

and

$$C_p = (S^T S)^{-1} B_p^T P. \quad (4.1-6)$$

Although we did not establish Eq. (4.1-6) with any notion of SPR conditions, it suffices to say that MRAC case I requires the open-loop plant to be ASPR.

Let's examine the second term of \dot{V} in Eq. (4.1-11),

$$-2e^T(t) P B_p (S^T S)^{-1} B_p^T P e(t) r^T(t) \bar{T} r(t).$$

The only term that is not quadratic is \bar{T} . We constrain this weighting matrix to be positive semi-definite. Thus, this Lyapunov term becomes negative semi-definite. Since

⁴ The converse is not true. Not all systems that satisfy the negative definite Lyapunov criterion are SPR. In essence, a stable system is not always SPR, but all SPR systems are stable.

we already constrained the first Lyapunov term to be negative definite, the entire expression will then be negative definite provided that the second term is non-positive.

4.1.2 Satisfying the Stability Constraints

To begin satisfying the ASPR conditions, we can select the matrix product, $\tilde{K}_e C_p$, such that the eigenvalues of $(A_p - B_p \tilde{K}_e C_p)$ lie strictly in the left-hand side of the complex plane. This should produce a stable closed-loop equation for all possible plant parameters. By doing this, we satisfy Eq. (4.1-6). If the plant output matrix C_p satisfies Eq. (4.1-6), then the plant is ASPR.

As an alternative, we can arbitrarily choose the plant output matrix C_p (Sobel, 1980). Checking to see if this is a valid choice, we define a transfer function, $Z(s)$, which isolates the plant output matrix for evaluation.

$$Z(s) = C_p (sI - A_p + B_p \tilde{K}_e C_p)^{-1} B_p$$

For this transfer function to indicate an ASPR system, a frequency domain analysis can demonstrate that

$$F(\omega) = Z(j\omega) + Z^T(-j\omega) > 0 \quad \forall \omega.$$

This inequality is one sufficient SPR condition that we provide in Appendix C.

The stability limits on \tilde{K}_e can be determined by knowing C_p and the acceptable range for the values of the matrix product $\tilde{K}_e C_p$. This is of little concern unless the designer wishes to implement a non-adaptive output feedback gain.

The output error feedback gain does not need to be adaptive. As discussed in the continuous CGT analysis, the output error feedback gain's principle role is to compensate for a difference in the initial conditions between the plant and the model as well as stabilization (if the plant is not Hurwitz). This gain is not meant to directly influence model output tracking.

4.1.3 Summary

Below, we provide a list of constraints sufficient to guarantee Lyapunov.

- (1) T is a positive definite matrix
- (2) \bar{T} is a positive semidefinite matrix
- (3) $C_p = (S^T S)^{-1} B_p^T P$
- (4) $P(A_p - B_p \tilde{K}_e C_p) + (A_p - B_p \tilde{K}_e C_p)^T P < 0$

Note that statements (3) and (4) are the ASPR conditions. If these equations can be satisfied for all possible plant dynamic properties, then the frequency domain analysis is not necessary. If the designer does not want to calculate an output matrix because he or she already has one, its validity is best checked using the frequency domain method.

4.2 Continuous Case II

In this section, we demonstrate that version II of the continuous MRAC algorithms does not require the open-loop plant to be ASPR. The plant must be output stabilizable but need not be minimum phase. This control method involves a feedthrough term which may compensate for zeros that are positive or located on the imaginary axis.

As a consequence (for the relaxed plant conditions), this approach requires some knowledge of the CGT gains. Our development first assumes explicit knowledge of the CGT gains. Afterwards, we consider the effects of plant modeling error if our knowledge of the CGT gains is nominal.

4.2.1 Formulating the Lyapunov Functions for Known CGT Gains

We begin by using the same Lyapunov equations from the last section:

$$V(e(t), K_I(t)) = e^T(t)Pe(t) + \text{trace} \left[S(K_I(t) - \tilde{K})^T T^{-1} (K_I(t) - \tilde{K})^T S^T \right] \quad (4.1-1)$$

and

$$\dot{V} = \dot{e}^T(t)Pe(t) + e^T(t)P\dot{e}(t) + 2\text{trace} \left[S(K_I(t) - \tilde{K})^T T^{-1} \dot{K}_I^T(t) S^T \right]. \quad (4.1-2)$$

The initial constraints and dimensions specified by section 4.1 remain valid.

Our development of \dot{V} begins by considering the error transients. Recall that,

$$\dot{e}(t) = A_p e(t) + B_p (u_p^*(t) - u_p(t)), \quad (2.3-25)$$

and

$$u_p(t) = u_p^*(t) + \tilde{K}_e C_p e(t) - z(t). \quad (2.4-12)$$

We can rewrite the error equation as,

$$\dot{e}(t) = \bar{A}_p e(t) + B_p z(t), \quad (4.2-1)$$

where

$$\bar{A}_p = A_p - B_p \tilde{K}_e C_p. \quad (4.2-2)$$

We substitute the error equation from Eq. (4.2-1) and the definition,

$$\dot{K}_1(t) = v(t)r^T(t)T, \quad (2.4-7)$$

into \dot{V} . For convenience, we drop the functional notation,

$$\dot{V} = e^T \bar{A}_p^T P e + z^T B_p^T P e + e^T P \bar{A}_p e + e^T P B_p z + 2\text{trace}\left[S(K_1 - \tilde{K})T^{-1}T r v^T S^T\right].$$

Let's examine the first and third terms. The quadratic functions of e may be factored so that

$$e^T \bar{A}_p^T P e + e^T P \bar{A}_p e = e^T (\bar{A}_p^T P + P \bar{A}_p) e.$$

Next, we consider the non-quadratic second and fourth terms. Because the matrix products in \dot{V} are scalar,

$$z^T B_p^T P e = e^T P B_p z.$$

Finally, we examine the trace term, $2\text{trace}\left[S(K_1 - \tilde{K})r v^T S^T\right]$. The matrix products $S(K_1 - \tilde{K})r$ and $v^T S^T$ are dimensioned $m \times 1$ and $1 \times m$ respectively. Hence, we may apply the following identity,

$$2\text{trace}\left[S(K_1 - \tilde{K})r v^T S^T\right] = 2v^T S^T S(K_1 - \tilde{K})r.$$

We substitute these three results into \dot{V} producing,

$$\dot{V} = e^T (\bar{A}_p^T P + P \bar{A}_p) e + 2e^T P B_p z + 2v^T S^T S(K_1 - \tilde{K})r. \quad (4.2-3)$$

We recognize a linear subsystem in the first term of \dot{V} . From the previous section, we recall that the linear subsystem should be ASPR. However, this algorithm does not require the open-loop plant to be ASPR. If we add a feedthrough, then the plant

need only be output stabilizable. Let's examine a transfer function containing a feedthrough term,

$$Z(s) = J + C_p [sI - \bar{A}_p]^{-1} B_p. \quad (4.2-4)$$

This transfer function is SPR if

$$\bar{A}_p^T P + P \bar{A}_p = -LL^T < 0, \quad (4.2-5a)$$

$$PB_p = C_p^T - LW, \quad (4.2-5b)$$

and

$$W^T W = J + J^T. \quad (4.2-5c)$$

In Appendix C, we provide a derivation of Eq. (4.2-5). Although the feedthrough term J is hypothetical, the Lyapunov stability analysis can demonstrate how this metaphysical term relates to a physical MRAC gain.

We proceed to substitute the set of equations of (4.2-5) into \dot{V} .

$$\dot{V} = -e^T LL^T e + 2e^T (C_p^T - LW)z + 2v^T S^T S(K_I - \tilde{K})r \quad (4.2-6)$$

Placing the feedthrough term of Eq. (4.2-5c) into this equation as a zero matrix, we have

$$\dot{V} = -e^T LL^T e - 2e^T LWz - \underbrace{z^T (-J - J^T + W^T W)z}_{=0} + 2e^T C_p^T z + 2v^T S^T S(K_I - \tilde{K})r.$$

Next, we distribute terms into a quadratic form so that,

$$\dot{V} = -(e^T L + z^T W^T)(L^T e + Wz) + 2z^T Jz + 2e^T C_p^T z + 2v^T S^T S(K_I - \tilde{K})r,$$

and then apply a matrix transpose property to the first term.

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{Wz})^T (\mathbf{L}^T \mathbf{e} + \mathbf{Wz}) + 2\mathbf{z}^T \mathbf{Jz} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{z} + 2\mathbf{v}^T \mathbf{S}^T \mathbf{S} (\mathbf{K}_I - \tilde{\mathbf{K}}) \mathbf{r} \quad (4.2-7)$$

We expand the dummy gain as

$$\tilde{\mathbf{K}} = \begin{bmatrix} \tilde{\mathbf{K}}_e & \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix},$$

and define \mathbf{z} using Eq. (2.4-12):

$$\mathbf{z} = \tilde{\mathbf{K}} \mathbf{r} - (\mathbf{vr}^T \bar{\mathbf{T}} + \mathbf{K}_I) \mathbf{r}.$$

We associate different terms in \mathbf{z} ,

$$\mathbf{z} = (\tilde{\mathbf{K}} - \mathbf{K}_I) \mathbf{r} - \mathbf{vr}^T \bar{\mathbf{T}} \mathbf{r}, \quad (4.2-8)$$

to solve for the matrix product $(\mathbf{K}_I - \tilde{\mathbf{K}}) \mathbf{r}$.

$$(\mathbf{K}_I - \tilde{\mathbf{K}}) \mathbf{r} = -(\mathbf{z} + \mathbf{vr}^T \bar{\mathbf{T}} \mathbf{r}) \quad (4.2-9)$$

The expression for \dot{V} then becomes

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{Wz})^T (\mathbf{L}^T \mathbf{e} + \mathbf{Wz}) + 2\mathbf{z}^T \mathbf{Jz} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{z} - 2\mathbf{v}^T \mathbf{S}^T \mathbf{S} (\mathbf{z} + \mathbf{vr}^T \bar{\mathbf{T}} \mathbf{r}).$$

Distributing through the fourth term produces,

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{Wz})^T (\mathbf{L}^T \mathbf{e} + \mathbf{Wz}) + 2\mathbf{z}^T \mathbf{Jz} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{z} - 2\mathbf{v}^T \mathbf{S}^T \mathbf{S} \mathbf{z} - 2\mathbf{v}^T \mathbf{S}^T \mathbf{S} \mathbf{vr}^T \bar{\mathbf{T}} \mathbf{r}. \quad (4.2-10)$$

By definition,

$$\mathbf{v} = \mathbf{Q} \mathbf{C}_p \mathbf{e} + \mathbf{Gz}, \quad (2.4-11)$$

which we substitute into the fourth term of Eq. (4.2-10).

$$\begin{aligned}\dot{V} = & -(\mathbf{L}^T \mathbf{e} + \mathbf{Wz})^T (\mathbf{L}^T \mathbf{e} + \mathbf{Wz}) + 2\mathbf{z}^T \mathbf{Jz} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{z} \\ & - 2(\mathbf{e}^T \mathbf{C}_p^T \mathbf{Q}^T + \mathbf{z}^T \mathbf{G}^T) \mathbf{S}^T \mathbf{Sz} - 2\mathbf{v}^T \mathbf{S}^T \mathbf{Svr}^T \bar{\mathbf{T}}r\end{aligned}\quad (4.2-11)$$

If we restrict Q to be related to S by,

$$\mathbf{Q} = (\mathbf{S}^T \mathbf{S})^{-1}, \quad (4.2-12)$$

then we can simplify Eq. (4.2-11) further.

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{Wz})^T (\mathbf{L}^T \mathbf{e} + \mathbf{Wz}) - 2\mathbf{z}^T (\mathbf{G}^T \mathbf{S}^T \mathbf{S} - \mathbf{J}) \mathbf{z} - 2\mathbf{v}^T \mathbf{S}^T \mathbf{Svr}^T \bar{\mathbf{T}}r \quad (4.2-13)$$

Term 1 of Eq. (4.2-13) is a quadratic and is negative definite. With term 1 negative definite, the remaining terms need only be negative semidefinite. This implies that

$$\mathbf{G}^T \mathbf{S}^T \mathbf{S} \geq \mathbf{J} \quad (4.2-13)$$

and that

$$\mathbf{T} \geq 0. \quad (4.2-14)$$

Note that in Eq. (4.2-13), we relate G to J. This is the physical manifestation of the hypothetical feedthrough term, J.

4.2.2 Formulating the Lyapunov Functions for Nominal CGT Gains

The stability analysis up to this point assumed explicit knowledge of the CGT solution. Let's define the nominal values of the CGT gains. These values are approximations of a CGT solution that could be used to control a plant "reasonably" well over a range of plant parameter uncertainty. The vector $\hat{\mathbf{u}}_p^*$ represents the nominal ideal control signal defined as,

$$\hat{u}_p^* = \hat{S}_{21}x_m(t) + \hat{S}_{22}u_m(t). \quad (4.2-15)$$

The matrices \hat{S}_{21} and \hat{S}_{22} are the nominal CGT gains. The ideal control signal is a constituent of $z(t)$ in Eq. (2.4-12). Thus, the trajectory $z(t)$ must also have a nominal value. Since

$$z(t) = u_p^*(t) - u_p(t) + \tilde{K}_e C_p e(t), \quad (2.4-12)$$

it follows that the nominal trajectory be defined as,

$$\hat{z}(t) = \hat{u}_p^*(t) - u_p(t) + \tilde{K}_e C_p e(t). \quad (4.2-16)$$

The vector $v(t)$ remains a function of $z(t)$ and its nominal value is defined according to Eq. (2.4-11).

$$\hat{v}(t) = QC_p e(t) + G\hat{z}(t) \quad (4.2-17)$$

We introduce a new term defined as the difference between the nominal CGT control and the correct CGT control,

$$\Delta u(t) = \hat{u}_p^*(t) - u_p^*(t). \quad (4.2-18)$$

Expressing $\hat{z}(t)$ in terms of $\Delta u(t)$, we have,

$$\hat{z}(t) = \Delta u(t) + u_p^*(t) - u_p(t) + \tilde{K}_e C_p e(t). \quad (4.2-19)$$

We define a new vector $q(t)$.

$$q(t) = u_p^*(t) - u_p(t) + \tilde{K}_e C_p e(t) \quad (4.2-20)$$

Although the definitions of $q(t)$ and $z(t)$ appear identical, the control signal $u_p(t)$ is not calculated in the same way for both equations. Essentially, $q(t)$ does not equal $z(t)$,

because $q(t)$ does not assume explicit knowledge of the plants dynamics in $u_p(t)$ (and mathematically, $z(t)$ does). Now we can write $\hat{v}(t)$ in terms of $q(t)$ and $\Delta u(t)$. Since,

$$\hat{z}(t) = \Delta u(t) + q(t), \quad (4.2-21)$$

$$\hat{v}(t) = QC_p e(t) + G(\Delta u(t) + q(t)). \quad (4.2-22)$$

With the original Lyapunov function and its first derivative remaining valid for this treatment, we develop the error equation. Recall that

$$\dot{e}(t) = A_p e(t) + B_p(u_p^*(t) - u_p(t)), \quad (2.3-25)$$

which we can also describe by,

$$\dot{e}(t) = A_p e(t) + B_p(q(t) - \tilde{K}_e C_p e(t)).$$

If $\bar{A}_p = A_p - B_p \tilde{K}_e C_p$ then

$$\dot{e}(t) = \bar{A}_p e(t) + B_p q(t). \quad (4.2-23)$$

The similarities between this error equation and that of Eq. (4.2-1) (except that q is in place of z) permit a stability analysis consistent with the previous section through Eq. (4.2-6). Below, we present the nominal counterpart to Eq. (4.2-7).

$$\dot{V} = -(L^T e + Wq)^T (L^T e + Wq) + 2q^T Jq + 2e^T C_p^T q + 2\hat{v}^T S^T S(K_I - \tilde{K})r \quad (4.2-24)$$

If we select

$$\tilde{K} = [\tilde{K}_e \quad S_{21} \quad S_{22}]$$

then

$$(K_I - \tilde{K})r = -(q + \hat{v}r^T \bar{T}r). \quad (4.2-25)$$

We substitute this result into the last term of \dot{V} .

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q})^T (\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q}) + 2\mathbf{q}^T \mathbf{J} \mathbf{q} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{q} - 2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S} (\mathbf{q} + \hat{\mathbf{v}} \mathbf{r}^T \bar{\mathbf{T}} \mathbf{r})$$

By distributing $2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S}$ over the last term,

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q})^T (\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q}) + 2\mathbf{q}^T \mathbf{J} \mathbf{q} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{q} - 2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S} \mathbf{q} - 2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S} \hat{\mathbf{v}} \mathbf{r}^T \bar{\mathbf{T}} \mathbf{r}.$$

We substitute for $\hat{\mathbf{v}}$ in the fourth term of \dot{V} :

$$\begin{aligned} \dot{V} = & -(\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q})^T (\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q}) + 2\mathbf{q}^T \mathbf{J} \mathbf{q} + 2\mathbf{e}^T \mathbf{C}_p^T \mathbf{q} \\ & - 2(\mathbf{e}^T \mathbf{C}_p^T \mathbf{Q}^T + \Delta \mathbf{u}^T \mathbf{G}^T + \mathbf{q}^T \mathbf{G}^T) \mathbf{S}^T \mathbf{S} \mathbf{q} - 2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S} \hat{\mathbf{v}} \mathbf{r}^T \bar{\mathbf{T}} \mathbf{r} \end{aligned}$$

Assigning

$$\mathbf{Q} = (\mathbf{S}^T \mathbf{S})^{-1},$$

allows us to eliminate two non-quadratic terms in \dot{V} :

$$\dot{V} = -(\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q})^T (\mathbf{L}^T \mathbf{e} + \mathbf{W} \mathbf{q}) - 2\mathbf{q}^T (\mathbf{G}^T \mathbf{S}^T \mathbf{S} - \mathbf{J}) \mathbf{q} - 2\hat{\mathbf{v}}^T \mathbf{S}^T \mathbf{S} \hat{\mathbf{v}} \mathbf{r}^T \bar{\mathbf{T}} \mathbf{r} - 2\Delta \mathbf{u}^T \mathbf{G}^T \mathbf{S}^T \mathbf{S} \mathbf{q}. \quad (4.2-26)$$

This form of \dot{V} is identical to that of Eq. (4.2-13) with exception to the final term. We are unable to evaluate whether or not $2\Delta \mathbf{u}^T \mathbf{G}^T \mathbf{S}^T \mathbf{S} \mathbf{q}$ is positive definite. Sobel *et al.* (1986) interpret a result from LaSalle (1961) that "outside some hypersurface where $\mathbf{e}(t)$ and $[\mathbf{q}(t)]$ are sufficiently large, the Lyapunov derivative will be negative. Thus, ultimately when the hypersurface is reached, a bound on the error is defined." It suffices to say that we need not worry about stability, even for heinous plant modeling error. We propose that the effect can be minimized by making the product, $\mathbf{G}^T \mathbf{S}^T \mathbf{S}$, as small as possible.

Physically, this means a relatively small feedthrough term, thus, less dependence on the nominal CGT gains.

4.2.3 Satisfying the Stability Constraints

Satisfying the stability constraints is similar to the process for case I. We begin by selecting a matrix product $\tilde{K}_e C_p$ such that \bar{A}_p is Hurwitz over the entire range of plant uncertainty. Next, we select a non-singular matrix L and an output matrix C_p to solve Eq. (4.2-5b) for W .

$$W = L^{-1}(C_p^T - P B_p)$$

By constraining J to be a symmetric matrix we can solve Eq. (4.2-5c) as,

$$J = \frac{1}{2} W^T W.$$

If the reader prefers the frequency domain, we can prove that the transfer function,

$$Z(s) = J + C_p [sI - \bar{A}_p]^{-1} B_p,$$

is SPR for a given gain \tilde{K}_e . By defining

$$F(\omega) = Z(j\omega) + Z^T(-j\omega),$$

we validate the choice of J by checking that $F(\omega)$ is positive definite for all frequencies.

Note that J can compensate for non-minimum phase zeros.

4.2.4 Summary

We provide a brief review of the stability constraints derived for MRAC case II.

(1) T is a positive definite matrix

(2) \bar{T} is a positive semidefinite matrix

$$(3) \bar{A}_p^T P + P \bar{A}_p = -LL^T < 0$$

$$(4) PB_p = C_p^T - LW$$

$$(5) W^T W = J + J^T$$

$$(6) Q = (S^T S)^{-1}$$

$$(7) G^T S^T S \geq J$$

4.3 Discrete Case

Like case II of the continuous MRAC schemes, the plant does not need to be ASPR. Although this control approach is similar to case II, analyzing stability reveals some very different constraints.

4.3.1 Formulating the Lyapunov Functions for Known CGT Gains

In a manner analogous to case II of the continuous MRAC, we form a quadratic Lyapunov function of the error and integral gains.

$$V(e(k), K_I(k)) = e^T(k)Pe(k) + \text{trace} \left[S(K_I(k) - \tilde{K})^T T^{-1} (K_I(k) - \tilde{K})^T S^T \right] \quad (4.3-1)$$

The terms of this function are such that, T^{-1} ($n_r \times n_r$) is positive definite, P ($n \times n$) is positive definite, \tilde{K} ($m \times n_r$), and S ($m \times m$) is non-singular. Recall that

$$\tilde{K} = \begin{bmatrix} \tilde{K}_e & \tilde{K}_x & \tilde{K}_u \end{bmatrix},$$

which implies,

$$\tilde{K}r(k) = \tilde{K}_e C_p e(k) + \tilde{K}_x x(k) + \tilde{K}_u u_m(k). \quad (4.3-2)$$

With the Lyapunov function, $V(e(k), K_I(k))$, shown to be positive definite, ΔV must prove to be negative definite to guarantee stability. We define ΔV as,

$$\Delta V = V(k+1) - V(k). \quad (4.3-3)$$

Taking a time step advance in V , the difference equation becomes,

$$\begin{aligned} \Delta V = & e^T(k+1)Pe(k+1) - e^T(k)Pe(k) \\ & + \text{trace} \left[S(K_I(k+1) - \tilde{K})T^{-1}(K_I(k+1) - \tilde{K})^T S^T \right] \\ & - \text{trace} \left[S(K_I(k) - \tilde{K})T^{-1}(K_I(k) - \tilde{K})^T S^T \right] \end{aligned} \quad (4.3-4)$$

Let's simplify this expression by developing the error dynamics. We begin with

$$e(k+1) = \Phi_p e(k) + \Gamma_p (u_p^*(k) - u_p(k)). \quad (2.3-48)$$

If we add the product $\Gamma_p (\tilde{K}_e - \tilde{K}_e)C_p e(k)$, which equals zero, we can describe the error by,

$$e(k+1) = \bar{\Phi}_p e(k) + \Gamma_p z(k) \quad (4.3-5)$$

where

$$\bar{\Phi}_p = (\Phi_p - \Gamma_p \tilde{K}_e C_p).$$

Placing this result into term 1 of ΔV , we have

$$\begin{aligned} e^T(k+1)Pe(k+1) = & e^T(k)\bar{\Phi}_p^T P \bar{\Phi}_p e(k) + 2e^T(k)\bar{\Phi}_p^T P \Gamma_p z(k) \\ & + z^T(k)\Gamma_p^T P \Gamma_p z(k) \end{aligned} \quad (4.3-6)$$

Next, we define and develop the time step difference of the trace term, Δ_{trace} .

$$\Delta_{\text{trace}} = \text{trace} \begin{bmatrix} S(K_I(k) - \tilde{K} + v(k)r^T(k)T)T^{-1}(K_I(k) - \tilde{K} + v(k)r^T(k)T)^T S^T \\ -S(K_I(k) - \tilde{K})T^{-1}(K_I(k) - \tilde{K})^T S^T \end{bmatrix}$$

By regrouping terms,

$$\Delta_{\text{trace}} = \text{trace} \begin{bmatrix} S(K_I(k) - \tilde{K})T^{-1}(K_I(k) - \tilde{K})^T S^T \\ +2Sv(k)r^T(k)TT^{-1}(K_I(k) - \tilde{K})^T \\ +Sv(k)r^T(k)TT^{-1}Tr(k)v^T(k)S^T \\ -S(K_I(k) - \tilde{K})T^{-1}(K_I(k) - \tilde{K})^T S^T \end{bmatrix}$$

It immediately follows that,

$$\Delta_{\text{trace}} = \text{trace} \left[Sv(k)r^T(k)Tr(k)v^T(k)S^T + 2Sv(k)r^T(k)(K_I(k) - \tilde{K})^T S^T \right].$$

The dimensional characteristics of these matrix products allow us to commute S, producing

$$\Delta_{\text{trace}} = \text{trace} \left[Sv(k)v^T(k)S^T r^T(k)Tr(k) + 2r^T(k)(K_I(k) - \tilde{K})^T S^T Sv(k) \right].$$

Using identities similar to those in section 4.2.1, we arrive at the final expression for Δ_{trace} .

$$\Delta_{\text{trace}} = v^T(k)S^T Sv(k)r^T(k)Tr(k) + 2v^T(k)S^T S(K_I(k) - \tilde{K})^T r(k) \quad (4.3-7)$$

We substitute equations (4.3-6) and (4.3-7) into ΔV .

$$\begin{aligned}\Delta V = & \mathbf{e}^T(k) \left(\overline{\Phi}_p^T P \overline{\Phi}_p - P \right) \mathbf{e}(k) + 2\mathbf{e}^T(k) \overline{\Phi}_p^T P \Gamma_p \mathbf{z}(k) + \mathbf{z}^T(k) \Gamma_p^T P \Gamma_p \mathbf{z}(k) \\ & + \mathbf{v}^T(k) S^T S \mathbf{v}(k) \mathbf{r}^T(k) \text{Tr}(k) + 2\mathbf{v}^T(k) S^T S \left(K_I(k) - \tilde{K} \right) \mathbf{r}(k)\end{aligned}\quad (4.3-8)$$

To derive the conditions for which ΔV is negative definite, we define the transfer function,

$$S(z) = J + C_p \left(zI - \overline{\Phi}_p \right)^{-1} \Gamma_p. \quad (4.3-9)$$

We correlate this transfer function to the linear subsystem implied by the first term of ΔV . If $S(z)$ is ASPR then the first term of ΔV must be negative definite. Satisfaction of the negative definite criterion is achieved if and only if there exists real matrices L and W and a real symmetric positive definite matrix P , such that the following identities hold.

$$\overline{\Phi}_p^T P \overline{\Phi}_p - P = -LL^T < 0 \quad (4.3-10a)$$

$$\overline{\Phi}_p^T P \Gamma_p = C_p - LW \quad (4.3-10b)$$

$$W^T W = J + J^T - \Gamma_p^T P \Gamma_p \quad (4.3-10c)$$

In Appendix C, we provide a derivation of Eq. (4.3-10). Substituting this set of equations into ΔV , we have

$$\begin{aligned}\Delta V = & -\mathbf{e}^T(k) LL^T \mathbf{e}(k) + 2\mathbf{e}^T(k) \left(C_p - LW \right) \mathbf{z}(k) + \mathbf{z}^T(k) \left(J + J^T - W^T W \right) \mathbf{z}(k) \\ & + \mathbf{v}^T(k) S^T S \mathbf{v}(k) \mathbf{r}^T(k) \text{Tr}(k) + 2\mathbf{v}^T(k) S^T S \left(K_I(k) - \tilde{K} \right) \mathbf{r}(k)\end{aligned}\quad (4.3-11)$$

After we distribute terms,

$$\begin{aligned}
\Delta V = & -\mathbf{e}^T(k)\mathbf{L}\mathbf{L}^T\mathbf{e}(k) - 2\mathbf{e}^T(k)\mathbf{L}\mathbf{W}\mathbf{z}(k) - \mathbf{z}^T(k)\mathbf{W}^T\mathbf{W}\mathbf{z}(k) \\
& + 2\mathbf{e}^T(k)\mathbf{C}_p\mathbf{z}(k) + 2\mathbf{z}^T(k)\mathbf{J}\mathbf{z}(k) \\
& + \mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}\mathbf{v}(k)\mathbf{r}^T(k)\mathbf{T}\mathbf{r}(k) + 2\mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}(\mathbf{K}_I(k) - \tilde{\mathbf{K}})\mathbf{r}(k)
\end{aligned}$$

Next, we factor terms into quadratic form, producing

$$\begin{aligned}
\Delta V = & -(\mathbf{L}^T\mathbf{e}(k) + \mathbf{W}\mathbf{z}(k))^T(\mathbf{L}^T\mathbf{e}(k) + \mathbf{W}\mathbf{z}(k)) \\
& + 2\mathbf{z}^T(k)\mathbf{J}\mathbf{z}(k) + 2\mathbf{e}^T(k)\mathbf{C}_p^T\mathbf{z}(k) \\
& + \mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}\mathbf{v}(k)\mathbf{r}^T(k)\mathbf{T}\mathbf{r}(k) + 2\mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}(\mathbf{K}_I(k) - \tilde{\mathbf{K}})\mathbf{r}(k)
\end{aligned} \tag{4.3-12}$$

We assign the dummy gain matrix to be $\tilde{\mathbf{K}} = \begin{bmatrix} \tilde{\mathbf{K}}_e & \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}$ and develop $\mathbf{z}(k)$

from Eq. (2.4-26).

$$\mathbf{z}(k) = (\tilde{\mathbf{K}} - \mathbf{K}_I(k))\mathbf{r}(k) - \mathbf{v}(k)\mathbf{r}^T(k)\bar{\mathbf{T}}\mathbf{r}(k) \tag{4.3-13}$$

Using this identity, we solve for the matrix product $(\mathbf{K}_I(k) - \tilde{\mathbf{K}})\mathbf{r}(k)$.

$$(\mathbf{K}_I(k) - \tilde{\mathbf{K}})\mathbf{r}(k) = -\mathbf{z}(k) - \mathbf{v}(k)\mathbf{r}^T(k)\bar{\mathbf{T}}\mathbf{r}(k) \tag{4.3-14}$$

Substituting Eq. (4.3-14) into ΔV brings us to,

$$\begin{aligned}
\Delta V = & -(\mathbf{L}^T\mathbf{e}(k) + \mathbf{W}\mathbf{z}(k))^T(\mathbf{L}^T\mathbf{e}(k) + \mathbf{W}\mathbf{z}(k)) \\
& + 2\mathbf{z}^T(k)\mathbf{J}\mathbf{z}(k) + 2\mathbf{e}^T(k)\mathbf{C}_p^T\mathbf{z}(k) \\
& - 2\mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}\mathbf{z}(k) - \mathbf{v}^T(k)\mathbf{S}^T\mathbf{S}\mathbf{v}(k)\mathbf{r}^T(k)(2\bar{\mathbf{T}} - \mathbf{T})\mathbf{r}(k)
\end{aligned} \tag{4.3-16}$$

By constraining,

$$2\bar{T} - T \geq 0 \quad (4.3-16)$$

term 5 of ΔV becomes negative semidefinite. We advance this development by substituting,

$$v(k) = Fe(k) + Gz(k), \quad (2.4-25)$$

into the Lyapunov equation.

$$\begin{aligned} \Delta V = & -\left(L^T e(k) + Wz(k)\right)^T \left(L^T e(k) + Wz(k)\right) \\ & - v^T(k) S^T S v(k) r^T(k) (2\bar{T} - T) r(k) - 2z^T(k) (G^T S^T S - J) z(k) \\ & + 2e^T(k) (C_p^T - F^T S^T S) z(k) \end{aligned} \quad (4.3-17)$$

The requirement that ΔV be negative definite mandates, with respect to the third term, that,

$$G^T S^T S \geq J. \quad (4.3-18)$$

The final term of ΔV is not a quadratic function of a single trajectory. Hence, we set this term to zero by establishing the constraint of

$$F = (S^T S)^{-1} C_p. \quad (4.3-19)$$

Eliminating this final term places ΔV into its final form.

$$\begin{aligned}
\Delta V = & -\left(L^T e(k) + Wz(k)\right)^T \left(L^T e(k) + Wz(k)\right) \\
& - v^T(k) S^T S v(k) r^T(k) (2\bar{T} - T) r(k) \\
& - 2z^T(k) (G^T S^T S - J) z(k)
\end{aligned} \tag{4.3-20}$$

4.3.2 Formulating the Lyapunov Functions for Nominal CGT Gains

As mentioned in Chapter 2, the discrete MRAC algorithm requires some knowledge of the discrete CGT gains. Equation (2.4-26) defines $z(k)$ to include the ideal plant input, $u_p^*(k)$. With a nominal knowledge of the plant's dynamics, we can generate values of S_{21} and S_{22} to form a "reasonable" control resulting in bounded output error. This nominal CGT control is calculated for the model of the plant. We denote the nominal CGT ideal input by a "hat":

$$\hat{u}_p^*(k) = \hat{S}_{21} x_m(k) + \hat{S}_{22} u_m(k). \tag{4.3-21}$$

Directly using Eq. (2.4-25), which defines $z(k)$, is no longer possible. Defining a nominal $z(k)$, we have

$$\hat{z}(k) = \hat{u}_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k) + (u_p^*(k) - u_p^*(k)).$$

If we associate terms differently, this becomes

$$\hat{z}(k) = (\hat{u}_p^*(k) - u_p^*(k)) + (u_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k)). \tag{4.3-22}$$

We define two new terms, $\Delta u(k)$ and $q(k)$. The difference between the nominal ideal control and the correct ideal control defines $\Delta u(k)$.

$$\Delta u(k) = \hat{u}_p^*(k) - u_p^*(k) \tag{4.3-23}$$

The term $q(k)$ uses the same form as $z(k)$ in Eq. (2.4-26), although these two trajectories are not identical: $u_p(k)$ differs in each case.

$$q(k) = u_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k) \quad (4.3-24)$$

Substituting these two trajectories into Eq. (4.3-22) reveals

$$\hat{z}(k) = \Delta u(k) + q(k). \quad (4.3-25)$$

Using this expression, we can define the nominal value of $v(k)$ as

$$\hat{v}(k) = F e(k) + G \hat{z}(k) \quad (4.3-26)$$

or by Eq. (4.3-25),

$$\hat{v}(k) = F e(k) + G(\Delta u(k) + q(k)). \quad (4.3-27)$$

We perform another Lyapunov stability analysis with the modified control law.

Beginning with

$$e(k+1) = \Phi_p e(k) + \Gamma_p (u_p^*(k) - u_p(k)), \quad (2.3-48)$$

we define the control as

$$u_p(k) = u_p^*(k) - q(k) + \tilde{K}_e C_p e(k), \quad (4.3-28)$$

and place this into the error expression.

$$e(k+1) = \Phi_p e(k) + \Gamma_p (q(k) - \tilde{K}_e C_p e(k))$$

If $\bar{\Phi}_p = \Phi_p - \Gamma_p \tilde{K}_e C_p$ then

$$e(k+1) = \bar{\Phi}_p e(k) + \Gamma_p q(k) \quad (4.3-29)$$

which permits a development consistent with equations (4.3-5) through (4.3-11). The equivalent to Eq. (4.3-12) for nominal CGT values is

$$\begin{aligned}\Delta V = & -\left(L^T e(k) + Wq(k)\right)^T \left(L^T e(k) + Wq(k)\right) \\ & + 2q^T(k)Jq(k) + 2e^T(k)C_p^T q(k) \\ & + \hat{v}^T(k)S^T S \hat{v}(k)r^T(k)Tr(k) + 2\hat{v}^T(k)S^T S \left(K_1(k) - \tilde{K}\right)r(k)\end{aligned}\quad (4.3-30)$$

By defining the dummy gain matrix as

$$\tilde{K} = \begin{bmatrix} \tilde{K}_e & S_{21} & S_{22} \end{bmatrix},$$

we can derive the identity,

$$\left(K_1(k) - \tilde{K}\right)r(k) = -\left(q(k) + \hat{v}(k)r^T(k)\bar{Tr}(k)\right). \quad (4.3-31)$$

We substitute for this matrix product in Eq. (4.3-30), so that ΔV becomes,

$$\begin{aligned}\Delta V = & -\left(L^T e(k) + Wq(k)\right)^T \left(L^T e(k) + Wq(k)\right) \\ & + 2q^T(k)Jq(k) + 2e^T(k)C_p^T q(k) + \hat{v}^T(k)S^T S \hat{v}(k)r^T(k)Tr(k) \\ & - 2\hat{v}^T(k)S^T S \hat{v}(k)r^T(k)\bar{Tr}(k) - 2\hat{v}^T(k)S^T S q(k)\end{aligned}\quad (4.3-32)$$

Using Eq. (4.3-27), we expand term 6 of ΔV .

$$\begin{aligned}
\Delta V = & -\left(\mathbf{L}^T \mathbf{e}(k) + \mathbf{W} \mathbf{q}(k)\right)^T \left(\mathbf{L}^T \mathbf{e}(k) + \mathbf{W} \mathbf{q}(k)\right) \\
& + 2\mathbf{q}^T(k) \mathbf{J} \mathbf{q}(k) + 2\mathbf{e}^T(k) \mathbf{C}_p^T \mathbf{q}(k) + \hat{\mathbf{v}}^T(k) \mathbf{S}^T \hat{\mathbf{v}}(k) \mathbf{r}^T(k) \mathbf{T} \mathbf{r}(k) \\
& - 2\hat{\mathbf{v}}^T(k) \mathbf{S}^T \hat{\mathbf{v}}(k) \mathbf{r}^T(k) \bar{\mathbf{T}} \mathbf{r}(k) - 2\mathbf{e}^T(k) \mathbf{F}^T \mathbf{S}^T \mathbf{S} \mathbf{q}(k) \\
& - 2\Delta \mathbf{u}^T(k) \mathbf{G}^T \mathbf{S}^T \mathbf{S} \mathbf{q}(k) - 2\mathbf{q}(k)^T \mathbf{G}^T(k) \mathbf{S}^T \mathbf{S} \mathbf{q}(k)
\end{aligned} \tag{4.3-33}$$

We apply the same matrix constraints, used in the previous section, and evolve ΔV into its final form.

$$\begin{aligned}
\Delta V = & -\left(\mathbf{L}^T \mathbf{e}(k) + \mathbf{W} \mathbf{q}(k)\right)^T \left(\mathbf{L}^T \mathbf{e}(k) + \mathbf{W} \mathbf{q}(k)\right) \\
& - \hat{\mathbf{v}}^T(k) \mathbf{S}^T \hat{\mathbf{v}}(k) \mathbf{r}^T(k) (2\bar{\mathbf{T}} - \mathbf{T}) \mathbf{r}(k) \\
& - 2\mathbf{q}^T(k) (\mathbf{G}^T \mathbf{S}^T \mathbf{S} - \mathbf{J}) \mathbf{q}(k) - 2\Delta \mathbf{u}^T(k) \mathbf{G}^T \mathbf{S}^T \mathbf{S} \mathbf{q}(k)
\end{aligned} \tag{4.3-34}$$

All terms in Eq. (4.3-34), except for the final term, are known to result in a negative definite function. The last term is dependent upon the values of $\Delta \mathbf{u}(k)$ and of $\mathbf{z}(k)$. The term $\Delta \mathbf{u}(k)$ is function of the model's state and input trajectories, both of which are assumed to be bounded in section 2.1.2. Sobel (1980) extends a result of LaSalle (1961) such that there exists a value of k , after which $\mathbf{e}(k)$ and $\mathbf{z}(k)$ are bounded and stable. This discrete algorithm, in a worst case scenario, guarantees that error is bounded but does not necessarily converge to zero. Again, we conjecture that minimizing $\mathbf{G}^T \mathbf{S}^T \mathbf{S}$ reduces MRAC sensitivity to errors in the plant model.

4.3.3 Satisfying the Stability Constraints

The procedure for selecting an appropriate matrix, J , stems from the solution of Eq. (4.3-10). First, \tilde{K}_e is selected to stabilize the open-loop state dynamics matrix. If Φ_p is open-loop stable, then \tilde{K}_e may equal the zero matrix. Next, we choose a non-singular value of L and plant output matrix C_p to solve Eq. (4.3-10a), demonstrating the existence of P . Then using Eq. (4.3-10b), we determine W .

$$W = L^{-1} \left[C_p^T - \overline{\Phi}_p^T P \Gamma_p \right]$$

Using Eq. (4.2-10c), we solve for J .

$$J = \frac{1}{2} \left(W^T W + \Gamma_p^T P \Gamma_p \right)$$

The designer can also meet ASPR conditions by showing that, for a given \tilde{K}_e ,

$$S(z) = J + C_p \left[zI - \overline{\Phi}_p \right]^{-1} \Gamma_p$$

is a SPR transfer function. Defining the function,

$$F(e^{j\omega T_s}) = S(e^{j\omega T_s}) + S^T(e^{-j\omega T_s})$$

where T_s is the sampling period, F must be positive definite for all frequencies.

The value of J and the resulting transfer matrix $S(z)$ depend upon nominal values of the plant's state dynamics and input matrices. If the nominal values deviate from the true values to a great extent, then it is possible that the discrete MRAC algorithm may perform poorly.

4.3.4 Summary

We review the sufficient stability constraints for the discrete MRAC system.

(1) T is a positive definite matrix

(2) $2\bar{T} - T \geq 0$

(3) $\bar{\Phi}_p^T P \bar{\Phi}_p - P = -LL^T < 0$

(4) $\bar{\Phi}_p^T P \Gamma_p = C_p - LW$

(5) $W^T W = J + J^T - \Gamma_p^T P \Gamma_p$

(6) $G^T S^T S \geq J$

(7) $F = (S^T S)^{-1} C_p$

4.4 MRAC with Time-Invariant Output Feedback Gain: Continuous Case I

A fixed feedback gain can simplify stability concerns in both the continuous and the discrete time cases. The Lyapunov function used in Eq. (4.1-1) remains the same. We truncate the dummy gain matrix to compensate for the fixed error feedback gain; the algorithm does not generate this gain on-line.

$$\tilde{K} = \begin{bmatrix} \tilde{K}_x & \tilde{K}_u \end{bmatrix}$$

Developing the Lyapunov function's first derivative requires us to consider error dynamics. This development differs somewhat from the original case in that Eq. (4.1-3) no longer holds. Instead,

$$\dot{e}(t) = A_p e(t) + B_p (S_{21} x_m(t) + S_{22} u_m(t) - K_I(t) r(t) - C_p e^T(t) \bar{T} r(t) - K_e C_p e(t)),$$

which may be regrouped to produce

$$\dot{e}(t) = (A_p - B_p K_e C_p) e(t) + B_p (S_{21} x_m(t) + S_{22} u_m(t) - K_I(t) r(t) - C_p e^T(t) \bar{T} r(t)).$$

Defining,

$$\bar{A}_p = A_p - B_p K_e C_p$$

allows us to use a development identical to section 4.1.1. The error dynamics then become,

$$\dot{e}(t) = \bar{A}_p e(t) + B_p (S_{21} x_m(t) + S_{22} u_m(t) - K_I(t) r(t) - C_p e^T(t) \bar{T} r(t)), \quad (4.4-1)$$

which is concurrent with section 4.1.1.

The next alteration, in this development, occurs at Eq. (4.1-7): the dummy gain expansion. The concatenated gain matrices are now of lower order, no longer including the output feedback gain. We use the matrix product,

$$\tilde{K} r = \tilde{K}_x x_m + \tilde{K}_u u_m,$$

which leads to the next expression of \dot{V} ,

$$\begin{aligned} \dot{V} = & e^T (P \bar{A}_p + \bar{A}_p^T P) e - 2e^T P B_p (S^T S)^{-1} B_p^T P e^T \bar{T} r \\ & - 2e^T C_p^T S^T S (\tilde{K}_x x_m + \tilde{K}_u u_m) + 2e^T P B_p (S_{21} x_m + S_{22} u_m) \end{aligned} \quad (4.4-2)$$

Expanding \bar{A}_p reveals,

$$\begin{aligned}
\dot{V} = & e^T \left[P(A_p - B_p K_e C_p) + (A_p - B_p K_e C_p)^T P \right] e \\
& - 2e^T P B_p (S^T S)^{-1} B_p^T P e r^T \bar{T} r \\
& + 2e^T P B_p \left[(S_{21} - \tilde{K}_x) x_m + (S_{22} - \tilde{K}_u) u_m \right]
\end{aligned} \tag{4.4-3}$$

Equation (4.4-3) is identical to Eq. (4.1-8) except that the known feedback gain K_e replaces the unknown dummy gain \tilde{K}_e . Setting the dummy gain equal to the ideal CGT control eliminates the final term in Eq. (4.4-3). This brings us to the final expression of \dot{V} .

$$\begin{aligned}
\dot{V} = & e^T(t) \left[P(A_p - B_p K_e C_p) + (A_p - B_p K_e C_p)^T P \right] e(t) \\
& - 2e^T(t) P B_p (S^T S)^{-1} B_p^T P e(t) r^T(t) \bar{T} r(t)
\end{aligned} \tag{4.4-4}$$

The first term of Eq. (4.4-4) must be negative definite. The values of P and K_e are chosen, along with a valid selection of C_p , such that

$$P(A_p - B_p K_e C_p) + (A_p - B_p K_e C_p)^T P < 0. \tag{4.4-5}$$

Since the values of both K_e and C_p are known, satisfying this equation ensures that the closed-loop plant is SPR rather than ASPR. The remainder of the constraints are identical to those enumerated in section 4.1.

4.5 MRAC with Time-Invariant Output Feedback Gain: Continuous Case II

4.5.1 Known CGT Gains

The Lyapunov function and its first derivative (from section 4.2) remain valid. We

begin by examining the error expression defined as,

$$\dot{e}(t) = A_p e(t) + B_p (u_p^*(t) - u_p(t)). \quad (2.3-25)$$

Using,

$$u_p(t) = u_{p1}(t) + K_e C_p e(t), \quad (2.5-6)$$

we can expand the error dynamics further.

$$\dot{e}(t) = A_p e(t) + B_p (u_p^*(t) - u_{p1}(t) - K_e C_p e(t))$$

Recall that

$$z(t) = u_p^*(t) - u_{p1}(t). \quad (2.5-8)$$

We substitute this into the error expression, producing

$$\dot{e}(t) = \bar{A}_p e(t) + B_p z(t), \quad (4.5-1)$$

where

$$\bar{A}_p = A_p - B_p K_e C_p. \quad (4.5-2)$$

Mathematically, the only differences between this development and that of section 4.2 are that K_e has replaced \tilde{K}_e and that the dimensions within the trace term differ. Using the fixed gain, our stability analysis in section 4.2.1 is applicable through Eq. (4.2-6). It then becomes necessary to examine $z(t)$. By defining

$$\tilde{K} = [S_{21} \quad S_{22}],$$

the control signal error, $z(t)$, becomes

$$z(t) = \tilde{K}r(t) - (v(t)r^T(t) + K_I)r(t).$$

The remainder of the development is consistent from Eq. (4.2-8) to the end of the section 4.2.1.

4.5.2 Nominal CGT Gains

We begin by examining $z(t)$ defined with nominal CGT gains. We establish that,

$$\hat{z}(t) = \hat{u}_p^*(t) - u_{pl}(t) \quad (4.5-3)$$

which may also be written as,

$$\hat{z}(t) = \hat{u}_p^*(t) - u_p^*(t) + u_p^*(t) - u_{pl}(t).$$

Since the definition of $\Delta u(t)$ remains

$$\Delta u(t) = \hat{u}_p^*(t) - u_p^*(t),$$

we define

$$q(t) = u_p^*(t) - u_{pl}(t), \quad (4.5-4)$$

such that

$$\hat{z}(t) = \Delta u(t) + q(t).$$

The remainder of the stability analysis presented in section 4.2.2 applies to the fixed gain option. The designer must be certain, however, that the selected output feedback gain results in a closed-loop matrix \bar{A}_p that is Hurwitz for all possible parameters.

4.6 Discrete MRAC with Time-Invariant Output Feedback Gain

4.6.1 Known CGT Gains

This development is similar to that of section 4.3. The Lyapunov function of Eq. (4.3-1) remains the same, however the dummy gain matrix no longer includes the output

feedback gain.

$$\tilde{K} = \begin{bmatrix} \tilde{K}_x & \tilde{K}_u \end{bmatrix}$$

Equation (4.3-2) must be described accordingly. Expanding the matrix product $\tilde{K}r(k)$ produces,

$$\tilde{K}r(k) = \tilde{K}_x x_m(k) + \tilde{K}_u u_m(k).$$

The Lyapunov difference equation development is consistent with Eq. (4.3-3) and Eq. (4.3-4). We define error differently, however. We begin by examining

$$e(k+1) = \Phi_p e(k) + \Gamma_p (u_p^*(k) - u_p(k)). \quad (2.3-48)$$

Using the definition,

$$u_p(k) = u_{p1}(k) + K_e C_p e(k), \quad (2.5-16)$$

and recalling that,

$$z(k) = u_p^*(k) - u_{p1}(k), \quad (2.5-18)$$

we can describe error as,

$$e(k+1) = \bar{\Phi}_p e(k) + \Gamma_p z(k) \quad (4.6-1)$$

where

$$\bar{\Phi}_p = (\Phi_p - \Gamma_p K_e C_p). \quad (4.6-2)$$

Equations (4.3-6) through (4.3-12) apply to the fixed gain option. The definition of $z(k)$ changes, however.

$$z(k) = S_{21}x_m(k) + S_{22}u_m(k) - (v(k)r^T(k)\bar{T} + K_I(k))r(k) \quad (4.6-3)$$

If define the dummy gain by,

$$\tilde{K} = [S_{21} \quad S_{22}],$$

then

$$z(k) = -\left(K_1(k) - \tilde{K}\right)r(k) - v(k)r^T(k)\bar{T}r(k), \quad (4.6-4)$$

which is identical to Eq. (4.3-15). The stability analysis then parallels the remainder of section 4.3.1.

4.6.2 Nominal CGT Gains

When only a nominal knowledge of the CGT gains is possible, the stability analysis becomes somewhat different for a fixed output feedback gain. The nominal representation of $z(k)$ changes. From

$$z(k) = u_p^*(k) - u_{p1}(k) \quad (2.5-18)$$

we describe the nominal version as,

$$\hat{z}(k) = \hat{u}_p^*(k) - u_{p1}(k) + \left(u_p^*(k) - u_p^*(k)\right)$$

(after adding a zero term). Changing the vector association, we write,

$$\hat{z}(k) = \left(\hat{u}_p^*(k) - u_p^*(k)\right) + \left(u_p^*(k) - u_{p1}(k)\right).$$

If

$$q(k) = u_p^*(k) - u_{p1}(k)$$

then

$$\hat{z}(k) = \Delta u(k) + q(k).$$

The remainder of the stability analysis from Eq. (4.3-20) to the end of section 4.3.2 applies to MRAC using a fixed output feedback gain.

4.7 Summary

In this chapter, we presented the stability analyses for each type of adaptive controller used in this study. We provided the constraints on design variables and discussed the mathematical manifestation of plant modeling error. In the next chapter, we examine how design variables may be chosen to result in “good” controller performance traits, completing our treatment of robustness.

Chapter 5

Discrete Design Procedures

In this chapter, we propose design methods for each of the four discrete control approaches: pole placement, CGT, MRAC, and MRAC using a fixed output feedback gain. For pole placement, our only concern is selecting an appropriate observation gain. Selecting the performance gains is a set process (adhering to guidelines in Chapter 2), based on the nominal plant.

For the CGT controller, we present our method of selecting the output feedback gain. Like pole placement, calculating performance gains relies on the accuracy of the nominal plant.

Our philosophy on MRAC is relatively simple. We almost arbitrarily satisfy Lyapunov stability requirements. We focus largely on selecting sample rate, weighting matrices, and the relative sizes of the proportional and integral weighting matrices.

5.1 Preliminary Design Considerations

We apply each control to the LTI plant in the testing environment. Testing conditions include plant modeling error and reference model selection (closed-loop performance specification). It is important that the reader understand that control laws are not developed for the characteristics of the actual plant. The control laws are designed for a model of this plant which may or may not be accurate. This plant estimation is termed the nominal plant or equivalently, the plant model.

The nominal plant is second order, having a mass of 3.375 lb., 1% damping, and a natural frequency of 6 Hz.

$$m_{pm} = 3.375 \text{ lb.} \quad (5.1-1)$$

$$\zeta_{pm} = 0.0100 \quad (5.1-2)$$

$$\omega_{pm} = 6 \text{ Hz} = 37.70 \text{ r/s} \quad (5.1-3)$$

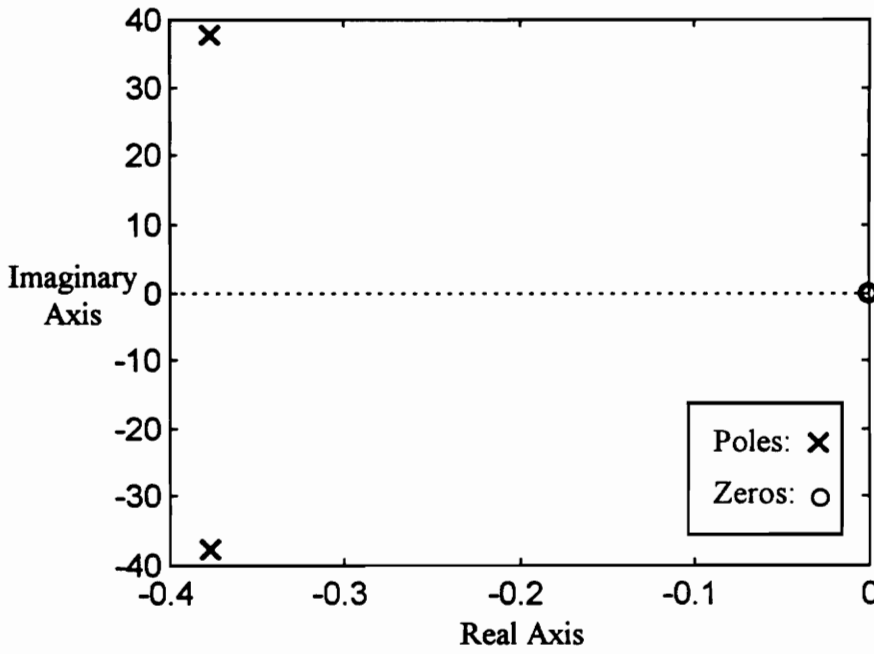


Figure 5-1
Pole and Zero Locations of the Continuous Nominal Plant

In the state space form of $\dot{x}_{pm}(t) = A_{pm}x_{pm}(t) + B_{pm}u_p(t)$ and $y_{pm}(t) = C_{pm}x_{pm}(t)$,

$$A_{pm} = \begin{bmatrix} -0.7540 & -1421 \\ 1 & 0 \end{bmatrix}, \quad (5.1-4)$$

$$\mathbf{B}_{pm} = \begin{bmatrix} 0.2963 \\ 0 \end{bmatrix}, \quad (5.1-5)$$

and

$$\mathbf{C}_{pm} = [1 \ 0]. \quad (5.1-6)$$

In Fig. 5-1, we provide a plot of the complex plane containing the pole and zero locations of the open-loop nominal plant.

5.1.1 Plant Modeling Error

We assume that the dynamic characteristics of the actual plant can differ from the nominal plant by as much as 30% of its damping ratio and 30% of its natural frequency. Plant damping ratio and natural frequency are denoted ζ_p and ω_p , respectively. These values exist within the ranges,

$$0.0070 \leq \zeta_p \leq 0.0130 \quad (5.1-7)$$

and

$$26.40 \text{ r/s} \leq \omega_p \leq 49.01 \text{ r/s}. \quad (5.1-8)$$

We illustrate the range of the actual plant's dynamic characteristics at the four extreme points in Fig. 5-2. The linear control algorithms rely heavily upon the accuracy of the nominal plant. Comparing these pole and zero locations to those of the nominal plant demonstrates a considerable disparity. These differences can have a substantial impact on the effectiveness of control gains.

5.1.2 Objectives

As we discussed in section 2.1.2, the object of control is to force a plant output to

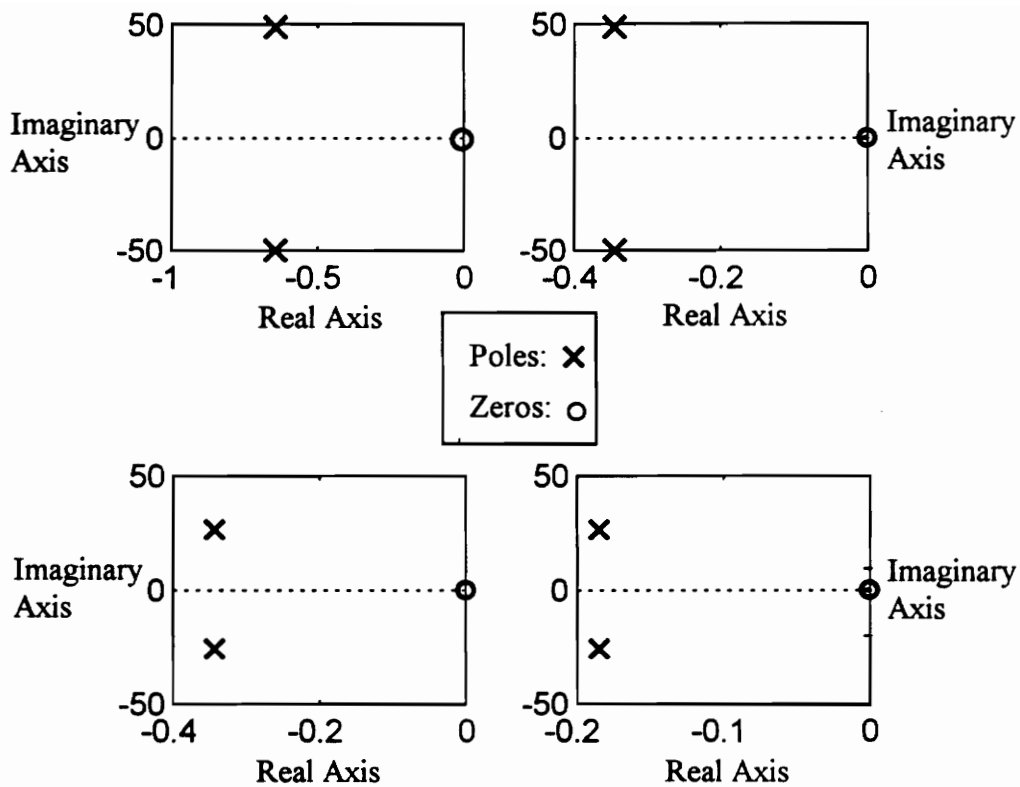


Figure 5-2
Continuous Plant Pole and Zero Locations*

*Clockwise From Top Left: Maximum Frequency and Maximum Damping; Maximum Frequency and Minimum Damping; Minimum Frequency and Minimum Damping; and Minimum Frequency and Maximum Damping

emulate an ideal trajectory. We assign values of damping and natural frequency to the ideal trajectory. Assuming that the ideal trajectory may be generated by a system of equivalent mass and structure, we can define a desired characteristic equation or reference model corresponding to this trajectory. The performance characteristics, defining the ideal trajectory, translate into damping and frequency specifications. We select our damping goal, ζ_m , to fall within the range

$$0.0100 \leq \zeta_m \leq 1.000. \quad (5.1-9)$$

The natural frequency objective, ω_m , varies through a range of +/- 50% of the nominal plant's natural frequency.

$$18.85 \text{ r/s} \leq \omega_m \leq 56.55 \text{ r/s} \quad (5.1-10)$$

5.1.3 Discrete Time Domain

We design our controllers for digital implementation. Therefore, we must select a sample rate that is suitable to implement all four controllers. Franklin *et al.* (1990) advise sampling at frequencies between 6 and 40 times the closed-loop frequency (or equivalently, 3 and 20 times the Nyquist frequency) of the plant. Increasing the sample rate produces a better approximation of an analog signal. In general, high sampling frequencies yield better control results than do low sampling frequencies. Also, we use a zero order hold (ZOH) approximation for which “quick” sample rates improve accuracy. However, there is usually a threshold after which the effects of quicker sampling become negligible. Since we conduct computer simulations, sampling at excessively high rates is costly in terms of computation time. We wish to minimize the sampling rate without compromising significant performance.

First, we consider a basis for selecting the sample rate. Our philosophy is one of overcompensation in that we consider the highest possible frequency between the actual plant and the reference model. We describe this as overcompensation because we will not necessarily need to sample as quickly as we do, but we will never sample at an insufficient rate.

Let's consider the plant. We know the natural frequency of the nominal plant and its value does not change. The actual plant, however, has a natural frequency that could be as much as 30% greater than the frequency of the nominal plant. We deem this maximum natural frequency to be a “critical point of consideration”, denoting it $\omega_p|_{\max}$.

$$\omega_p|_{\max} = 49.01 \text{ r/s.} \quad (5.1-11)$$

The natural frequency of the ideal trajectory or reference model is another critical point of consideration. This defines the closed-loop (or controlled) frequency of the plant, provided that it is correctly modeled. We cannot realistically assume this. We do assume that if the ideal trajectory contains a natural frequency lower than that of the nominal plant, the plant's frequency will not exceed $\omega_p|_{\max}$.

In the continuous domain, we assume that the control gain associated with changing the natural frequency is proportional to $\omega_{pm}^2 - \omega_m^2$.⁵ If the ideal trajectory is defined by a natural frequency greater than the nominal plant's, the resulting linear control will be excessive. The plant's natural frequency will be much greater than expected. For the worst case scenario, we assume the plant's natural frequency to be $\omega_p|_{\max}$. This conservative precaution predicts a closed-loop natural frequency of $\sqrt{\omega_m^2 + \omega_p|_{\max}^2} - \omega_{pm}$ or $\sqrt{\omega_m^2 + 980.6}$.

We select the closed-loop natural frequency to equal $\omega_p|_{\max}$,

⁵ We base this on pole placement calculations which are the most sensitive to plant modeling error.

$$\omega_{cl} = \omega_p|_{\max}, \quad (5.1-12a)$$

if the natural frequency goal is lower than that of the open-loop nominal plant. If the ideal trajectory calls for an increased natural frequency, we choose

$$\omega_{cl} = \sqrt{\omega_m^2 + \omega_p|_{\max}^2 - \omega_{pm}^2} \quad (5.1-12b)$$

as our closed-loop natural frequency.

We finally consider the adaptation characteristic of the MRAC algorithms. To aid the "learning" process of these algorithms, we want to ensure that the sample rate is fast enough to enable accurate integration. Based on our conservative approach in selecting ω_{cl} , sampling at 10 times the Nyquist frequency will produce sufficient smoothness without requiring excessive simulation time. The sampling frequency is

$$\omega_s = 20\omega_{cl} = \frac{1}{T_s}, \quad (5.1-13)$$

where T_s is the sampling period.

5.2 Pole Placement

We will briefly discuss the pole placement design procedure. This is one of the most academic control approaches and may be found in many digital control texts such as Franklin *et al.* (1990). There is no way to design for robustness when we pick closed-loop pole locations. The only aspect of this section, that may interest the reader, is our selection of the observer gain.

5.2.1 Performance Gain Selection

Since the order of the plant and reference model are identical, we can solve for a

feedback gain matrix that equates their characteristic equations. We use MATLAB, a dynamic simulations package, to calculate this matrix of feedback gains. This matrix, K , creates the equality,

$$\left| zI - \Phi_{pm} + \Gamma_{pm} K \right| = \left| zI - \Phi_m \right| = 0. \quad (5.2-1)$$

As a note to the reader, we also use MATLAB to convert all continuous state matrices to the discrete domain. We specify a zero order hold for this transformation. Our particular transformation maintains the physical significance of the states (e.g., the discrete states remain defined by position and velocity). The resulting canonical form, however, is not conducive to pole placement. For this reason, we find MATLAB particularly useful.

5.2.2 Observer Gain Selection

Let's consider the dynamics of the pole placement controller. Recall the characteristic equation,

$$\left| zI - \Phi_{pm} + \Gamma_{pm} K \right| \left| zI - F \right| = 0, \quad (2.2-36)$$

whose dynamic elements are specified separately. The observation error dynamics are governed by the eigenstructure of F . Since we only estimate position, F is merely a time constant. We choose F to be the least dominant eigenvalue of the entire system. For a second order system, 95% settling time is approximated as $\frac{2}{\zeta_{cl}\omega_{cl}}$. We select F to be 5 times this value in the continuous domain (Friedland, 1986). We then approximate the discrete equivalent to F using the transformation, $z = e^{sT_s}$. This transformation produces

$$F = e^{-\left(\frac{10T_s}{\zeta_{cl}\omega_{cl}}\right)} \quad (5.2-2)$$

In the continuous domain, the values of real closed-loop poles may be evaluated by the formula,

$$\left(\zeta_p \omega_p\right)_{cl} = \left(\zeta_m \omega_m + \zeta_p \omega_p - \zeta_{pm} \omega_{pm}\right).$$

This expression is maximized for the greatest possible value of $\zeta_p \omega_p$ or 0.6371.

$$\left(\zeta_p \omega_p\right)_{cl} = \left(\zeta_m \omega_m + 0.6371 - \zeta_{pm} \omega_{pm}\right) \quad (5.2-3)$$

5.3 Command Generator Tracker

The CGT approach to model reference control, like pole placement, involves two separate gain selection procedures. The designer, in accordance with section 2.3.2, calculates S_{21} and S_{22} . These are the feedforward gains: the gains directly responsible for specifying the performance of the plant (we consider these "performance" gains). Section 2.3.2 also mentions an output feedback gain, K_e . This gain causes error, between the outputs of the plant and model, to converge at a specified rate. The output feedback gain is necessary when initial conditions of the plant and model differ. It can also improve performance when the plant is not accurately modeled.

The performance gains do not affect the stability of the controlled plant. Their open-loop contribution to the control signal is bounded by the stability of the reference model (which we constrain to be stable). The only possible source of instability is the output feedback gain. Let's consider the discrete nominal plant and its root locus plot, provided in Fig. 5-3. This plot shows that as the output feedback gain increases, closed-

loop poles move from their open-loop positions, and converge to the real axis. After the closed-loop poles reach the real axis, they diverge in opposite directions. One becomes increasingly dominant as it approaches +1 on the unit circle. The other, after crossing the origin,

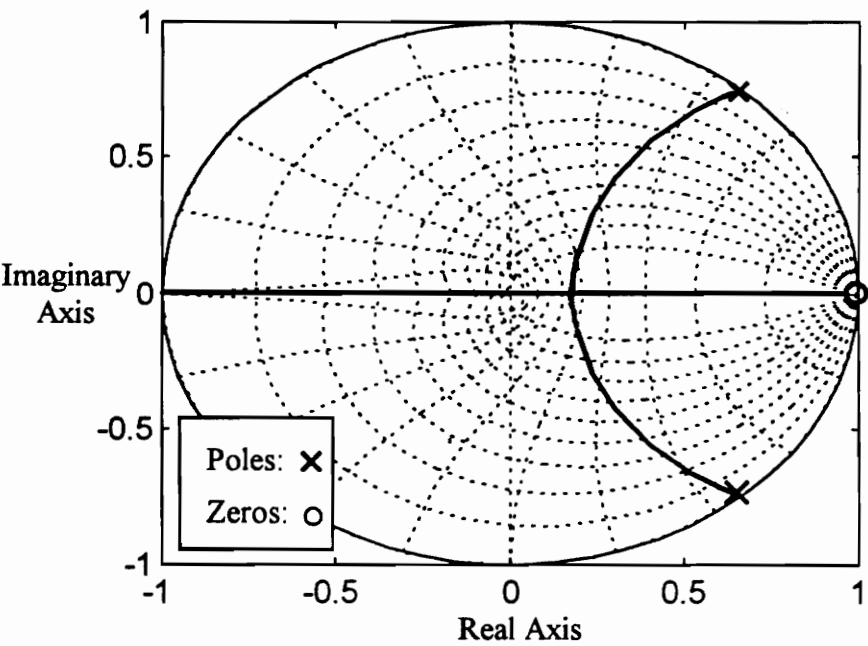


Figure 5-3
Root Locus Plot of the Discrete Nominal Plant

becomes more dominant. This pole eventually moves into the region of instability, outside the unit circle. Naturally, we do not want a closed-loop pole located in the unstable region. This is simple enough to prevent. However, we also face the task of selecting an operating point that produces "quick" error transients. We focus on the region where the closed-loop poles converge to the real axis, maximizing error decay rate. Our goal is to

prevent one pole from becoming more dominant than the other, and produce unnecessarily "slow" transients.

5.3.1 Selecting Performance Gains

The designer does not consider stability when selecting the gains, S_{21} and S_{22} . One can only hope for an accurate estimate of the plant's dynamic characteristics. Our solution process is rather straightforward though we imagine it is not the only way to determine S_{21} and S_{22} .

Discretizing the state matrices can alter their canonical forms. This is almost always (if not invariably) true when one wishes to maintain the physical significance of trajectories over the two domains. After calculating the discrete representations of the nominal plant and reference model, we convert the state matrices into the feedback canonical form (see Appendix B), using the transformation matrices T_{pm} and T_m for the plant model and reference model, respectively. Let these transformed matrices be denoted by a "hat" such that

$$\{\Phi_{pm}, \Gamma_{pm}, C_{pm}\} \Rightarrow \{\hat{\Phi}_{pm}, \hat{\Gamma}_{pm}, \hat{C}_{pm}\} \quad (5.3-1)$$

and

$$\{\Phi_m, \Gamma_m, C_m\} \Rightarrow \{\hat{\Phi}_m, \hat{\Gamma}_m, \hat{C}_m\}. \quad (5.3-2)$$

As an illustration to the reader, we provide an example of actual matrices calculated in equations (5.3-1) and (5.3-2). For a sample period of 0.0032 seconds, model frequency 10% greater than the plant, and a model damping ratio of 0.17,

$$\{\hat{\Phi}_{pm}, \hat{\Gamma}_{pm}, \hat{C}_{pm}\} = \left\{ \begin{bmatrix} 1.9830 & -0.9976 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, (10^{-3}) \begin{bmatrix} 0.9462 & -0.9462 \end{bmatrix} \right\} \quad (5.3-3)$$

and

$$\{\hat{\Phi}_m, \hat{\Gamma}_m, \hat{C}_m\} = \left\{ \begin{bmatrix} 1.9336 & -0.9531 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, (10^{-3}) \begin{bmatrix} 0.9241 & -0.9241 \end{bmatrix} \right\}. \quad (5.3-4)$$

From Eq. (2.3-20),

$$\begin{bmatrix} \hat{S}_{11} \hat{\Phi}_m & \hat{S}_{11} \hat{\Gamma}_m + \hat{S}_{12} \\ \hat{C}_m & 0 \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_{pm} & \hat{\Gamma}_{pm} \\ \hat{C}_{pm} & 0 \end{bmatrix} \begin{bmatrix} \hat{S}_{11} & \hat{S}_{12} \\ \hat{S}_{21} & \hat{S}_{22} \end{bmatrix} \quad (5.3-5)$$

where \hat{S}_{11} , \hat{S}_{12} , \hat{S}_{21} and \hat{S}_{22} correspond to the feedback canonical form. We may immediately notice that

$$\hat{C}_m = \hat{C}_{pm} \hat{S}_{11}. \quad (5.3-6)$$

In this expression, there are two equations and four unknowns. Without a loss of generality, we constrain the \hat{S}_{11} (2×2) to be diagonal and solve Eq. (5.3-6) element by element.

From Eq. (5.3-5) we see that

$$0 = \hat{C}_{pm} \hat{S}_{12}. \quad (5.3-7)$$

This solution is trivial. We choose \hat{S}_{21} to equal a 2×1 zero matrix.

Next, we define the matrix Ω .

$$\Omega = \begin{bmatrix} \hat{\Phi}_{pm} & \hat{\Gamma}_{pm} \\ \hat{C}_{pm} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \quad (5.3-8)$$

Multiplying Ω on both sides of Eq. (5.3-3) yields the set of equations,

$$\hat{S}_{11} = \Omega_{11}\hat{S}_{11}\hat{\Phi}_m + \Omega_{12}\hat{C}_m, \quad (5.3-9)$$

$$\hat{S}_{12} = \Omega_{11}\hat{S}_{11}\hat{\Gamma}_m + \Omega_{11}\hat{S}_{12}, \quad (5.3-10)$$

$$\hat{S}_{21} = \Omega_{21}\hat{S}_{11}\hat{\Phi}_m + \Omega_{22}\hat{C}_m, \quad (5.3-11)$$

and

$$\hat{S}_{22} = \Omega_{21}\hat{S}_{11}\hat{\Gamma}_m + \Omega_{21}\hat{S}_{12}. \quad (5.3-12)$$

Already knowing \hat{S}_{11} and \hat{S}_{12} , we need not concern ourselves with equations (5.3-9) and (5.3-10). The solutions to \hat{S}_{21} and \hat{S}_{22} are attained simply by using equations (5.3-11) and (5.3-12).

The gains, \hat{S}_{21} and \hat{S}_{22} , must be converted from feedback canonical form into the original coordinate frame. The form of the ideal input, from Eq. (2.3-2), is

$$u_p^*(k) = \hat{S}_{21}\hat{x}_m(k) + \hat{S}_{22}u_m(k). \quad (5.3-13)$$

From the transformation procedure provided in Appendix B, we infer that

$$\hat{x}_m(k) = T_m x_m(k), \quad (5.3-14)$$

which we substitute into the ideal input equation,

$$u_p^*(k) = \hat{S}_{21}T_m x_m(k) + \hat{S}_{22}u_m(k).$$

Since

$$S_{21} = \hat{S}_{21}T_m \quad (5.3-15)$$

and

$$S_{22} = \hat{S}_{22}, \quad (5.3-16)$$

we can convert the ideal input back to its original form given by Eq. (2.3-2).

5.3.2 Selecting the Output Feedback Gain

For the range of possible plant characteristics, we select a feedback gain K creating closed-loop poles that do not touch the real axis in the discrete complex plane. Critical damping is difficult enough to achieve, but exceeding it would be detrimental to performance. We question which type of plant uncertainty requires the smallest gain, K , for closed-loop poles to lie on the real axis. Let's examine the four extreme cases of erroneous plant modeling.

In Fig. 5-4, we see that open-loop poles with relatively low natural frequencies produce the most dominant closed-loop poles. These poles travel the smallest distances to contact the real axis. Although it is difficult to discern from these plots, the plant having the lowest natural frequency with the highest possible damping, converges to the largest value on the real axis. In Table 5-1, we provide the gains and pole positions associated with real axis convergence.

We see from this table, however, that the most dominant intersection does not correspond to the lowest possible crossing gain (the feedback gain at which convergence occurs).

Let's consider the region of the z -plane in which the plant's roots lie. This entails the right half of the unit circle. As we increase the sampling frequency, the open-loop region condenses around $+1$ on the real axis. The trends in Table 5-1 are not affected by

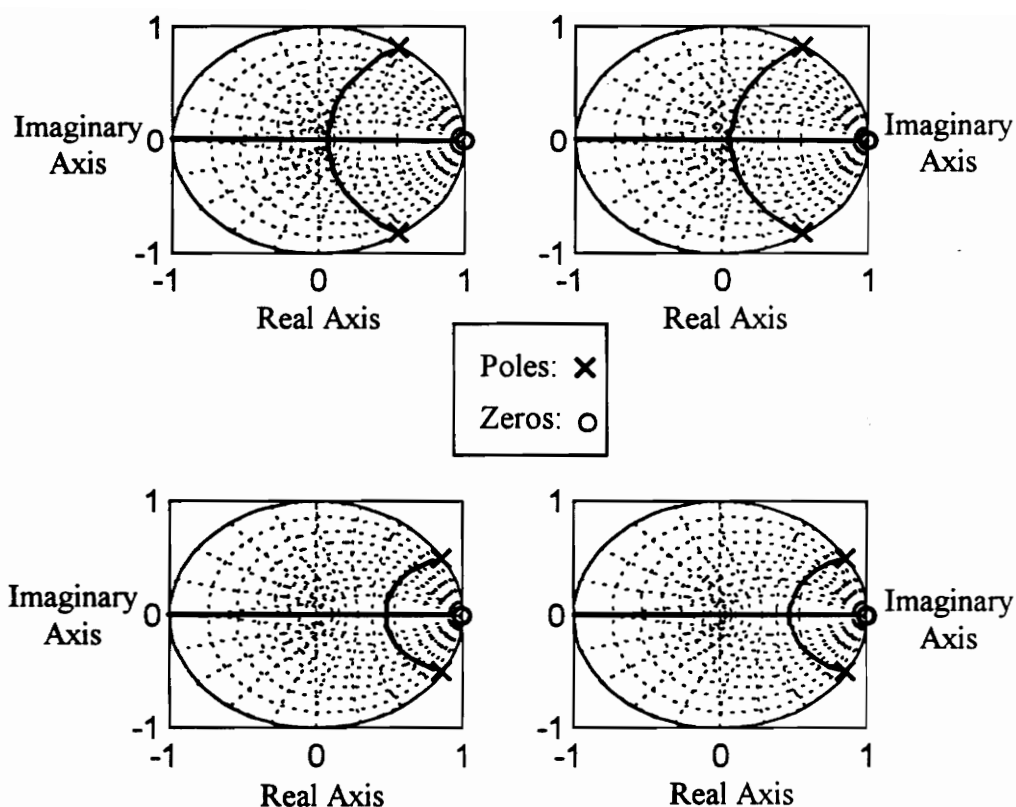


Figure 5-4*
Root Locus Plots of Discrete Plant

*Clockwise From Top Left: Maximum Frequency and Maximum Damping; Maximum Frequency and Minimum Damping; Minimum Frequency and Minimum Damping; and Minimum Frequency and Maximum Damping

an increased sampling frequency. In fact, we would not normally sample so slowly; we chose T_s only to produce clear root locus plots. Table 5-2 provides the same type of information, but for a more realistic sample rate. For the increased sample rate, the differences are less pronounced, but the trends in Table 5-1 are consistent with those of Table 5-2.

Table 5-1
Relation of the Plant's Dynamic Characteristics to Closed-Loop Pole Locations for $T_s = 0.02$ seconds

Possible Plant Dynamic Characteristics	Point of Real Axis Convergence	Associated Feedback Gain
Maximum Frequency Maximum Damping	0.0645	195.8
Maximum Frequency Minimum Damping	0.0618	197.0
Minimum Frequency Maximum Damping	0.480	134.6
Minimum Frequency Minimum Damping	0.479	135.4

Table 5-2
Relation of the Plant's Dynamic Characteristics to Closed-Loop Pole Locations for $T_s = 0.0032$ seconds

Possible Plant Dynamic Characteristics	Point of Real Axis Convergence	Associated Feedback Gain
Maximum Frequency Maximum Damping	0.8433	301.2
Maximum Frequency Minimum Damping	0.8431	303.7
Minimum Frequency Maximum Damping	0.9155	168.5
Minimum Frequency Minimum Damping	0.9154	169.6

Consider two open-loop poles having identical natural frequencies and different damping ratios. It may seem peculiar that a pole crossing the real axis, closer to +1 than the other, corresponds to a higher feedback gain. There are two reasons for this occurrence. First, open-loop poles with lower damping ratios are further from the origin than those having higher damping ratios. This causes each "branch" (Hale, 1988) from the open-loop pole locations to converge upon the real axis at a point defined as

$$cg = \frac{\sum \text{poles} - \sum \text{zeros}}{(\text{number of poles}) - (\text{number of zeros})} \quad (5.3-17)$$

We find cg simply by summing the real components of each open-loop pole and subtracting 1.

The open-loop pole with greater damping is closer to the origin, and, closer to the real axis. Having a shorter distance to travel, this pole requires a smaller feedback gain to reach the real axis.

These trends cause a design dilemma. We do not want closed-loop poles surpassing the point of intersection on the real axis. On the other hand, we should avoid needlessly "slow" poles by selecting conservative feedback gains. These alternatives are not equally detrimental.

From experience we understand that poles are very sensitive in this critical region; for a given change in feedback gain, poles diverge further along the real axis than the change in location in their approach. If we select the lowest gain of 168.5 from Table 5-2, the most dominant possible pole has a magnitude of 0.9160 and is complex. This

corresponds to the minimum damping ratio and frequency. Selecting the slightly higher gain, 169.6 from Table 5-2, leads to the most dominant possible pole, having a magnitude of 0.9243. This pole is divergent along the real axis. This is a significant difference for such a slight change in the choice of gains. We choose the gain corresponding to the plant having the highest possible damping ratio and the lowest possible natural frequency.

5.4 Model Reference Adaptive Control

In this section, we examine the significance of the Lyapunov equations and their relation to performance. We will then propose a method of selecting weighting matrices. We point out which design variables are arbitrary and treat them as simply as possible. Additionally, our methods consider the nonlinearity of MRAC and minimize its sensitivity to initial conditions.

5.4.1 The Lyapunov Functions and ASPR Constraints

We begin by providing a short review of the Lyapunov equations. From Eq. (4.3-1),

$$V(e(k), K_1(k)) = e^T(k)Pe(k) + \text{trace} \left[S(K_1(k) - \tilde{K})^T (K_1(k) - \tilde{K})^T S^T \right]$$

and from Eq. (4.3-34),

$$\begin{aligned} \Delta V = & -\left(L^T e(k) + Wq(k)\right)^T \left(L^T e(k) + Wq(k)\right) \\ & - \hat{v}^T(k) S^T S \hat{v}(k) r^T(k) (2\bar{T} - T) r(k) \\ & - 2q^T(k) (G^T S^T S - J) q(k) - 2\Delta u^T(k) G^T S^T S q(k) \end{aligned}$$

Recall that

$$q(k) = u_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k), \quad (4.3-24)$$

$$\hat{z}(k) = \Delta u(k) + q(k), \quad (4.3-25)$$

$$\hat{v}(k) = F e(k) + G(\Delta u(k) + q(k)) = F e(k) + G \hat{z}(k), \quad (4.3-26,27)$$

$$\Delta u(k) = \hat{u}_p^*(k) - u_p^*(k), \quad (4.3-23)$$

and

$$\hat{u}_p^*(k) = \hat{S}_{21} x_m(k) + \hat{S}_{22} u_m(k).$$

Our first step is to select \tilde{K}_e . In the feedthrough vector, $z(k)$, we see that the plant's actuation is a function of the adaptive signal's failure to converge to $u_p^*(k) + \tilde{K}_e C_p e(k)$. The feedback matrix \tilde{K}_e essentially penalizes the controller for not converging to its feedthrough term contribution: $\tilde{K}_e C_p e(k)$. Assuming this to be the case, we would not want to encourage the adaptive feedback gain to converge to detrimental values (or be penalized for not doing so). We select

$$\tilde{K}_e = K_e, \quad (5.4-1)$$

with the understanding that the output feedback gain designed for CGT optimizes performance for the worst possible plant characteristics. It is possible to select this gain to be zero without compromising stability. However, our plant is so lightly damped that using a non-zero \tilde{K}_e improves performance, particularly when the reference model is heavily damped.

Next, we satisfy the ASPR stability constraints. Beginning with Eq. (4.3-10a),

$$\overline{\Phi}_{pm}^T P \overline{\Phi}_{pm} - P = -LL^T < 0 \quad (5.4-2)$$

where

$$\overline{\Phi}_{pm} = \left(\Phi_{pm} - \Gamma_{pm} \tilde{K}_e C_{pm} \right), \quad (5.4-3)$$

we can arbitrarily select a real, 2×2 matrix L (e.g., the identity matrix), and solve the symmetric positive definite matrix, P . Recall from the root locus plots in Fig. 5-4, that the selection of \tilde{K}_e places the poles well within the unit circle. Satisfying the ASPR conditions is almost trivial but does affect the choice of one design variable, G . Let's define

$$Q = L^T L, \quad (5.4-4)$$

where

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}. \quad (5.4-5)$$

We also define

$$\overline{\Phi}_{pm} = \begin{bmatrix} \overline{\Phi}_{11} & \overline{\Phi}_{12} \\ \overline{\Phi}_{21} & \overline{\Phi}_{22} \end{bmatrix}, \quad (5.4-6)$$

and

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}. \quad (5.4-7)$$

We substitute these equations into Eq. (5.4-2).

$$\begin{bmatrix} \phi_{11}^2 p_{11} + \phi_{11} \phi_{21} p_{21} + \phi_{11} \phi_{21} p_{12} + \phi_{21}^2 p_{22} & \phi_{11} \phi_{12} p_{11} + \phi_{12} \phi_{21} p_{21} + \phi_{11} \phi_{22} p_{12} + \phi_{21} \phi_{22} p_{22} \\ \phi_{11} \phi_{12} p_{11} + \phi_{11} \phi_{22} p_{11} + \phi_{12} \phi_{21} p_{12} + \phi_{21} \phi_{22} p_{22} & \phi_{12}^2 p_{11} + \phi_{12} \phi_{22} p_{21} + \phi_{12} \phi_{22} p_{12} + \phi_{22}^2 p_{22} \end{bmatrix} - \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = - \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

We solve for the scalar components of P using the system of equations,

$$\begin{bmatrix} \phi_{11}^2 - 1 & \phi_{11} \phi_{21} & \phi_{11} \phi_{21} & \phi_{21}^2 \\ \phi_{11} \phi_{12} & \phi_{11} \phi_{22} - 1 & \phi_{12} \phi_{21} & \phi_{21} \phi_{22} \\ \phi_{11} \phi_{12} & \phi_{12} \phi_{21} & \phi_{11} \phi_{22} - 1 & \phi_{21} \phi_{22} \\ \phi_{12}^2 & \phi_{12} \phi_{22} & \phi_{11} \phi_{22} & \phi_{22}^2 - 1 \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{22} \end{bmatrix} = - \begin{bmatrix} q_{11} \\ q_{12} \\ q_{21} \\ q_{22} \end{bmatrix}. \quad (5.4-8)$$

Defining,

$$\Xi = \begin{bmatrix} \phi_{11}^2 - 1 & \phi_{11} \phi_{21} & \phi_{11} \phi_{21} & \phi_{21}^2 \\ \phi_{11} \phi_{12} & \phi_{11} \phi_{22} - 1 & \phi_{12} \phi_{21} & \phi_{21} \phi_{22} \\ \phi_{11} \phi_{12} & \phi_{12} \phi_{21} & \phi_{11} \phi_{22} - 1 & \phi_{21} \phi_{22} \\ \phi_{12}^2 & \phi_{12} \phi_{22} & \phi_{11} \phi_{22} & \phi_{22}^2 - 1 \end{bmatrix}, \quad (5.4-9)$$

if Ξ is non-singular, then we can solve for the elements in P by,

$$\begin{bmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{22} \end{bmatrix} = -\Xi^{-1} \begin{bmatrix} q_{11} \\ q_{12} \\ q_{21} \\ q_{22} \end{bmatrix}. \quad (5.4-10)$$

We proceed to satisfy the next ASPR constraint by solving for W using Eq. (4.3-10b),

$$W = L^{-1} \left[C_{pm}^T - \overline{\Phi}_{pm}^T P \Gamma_{pm} \right] \quad (5.4-11)$$

The solution for J immediately follows using,

$$J = 0.5(W^T W + \Gamma_{pm}^T P \Gamma_{pm}). \quad (5.4-12)$$

To conclude this treatment of the ASPR constraints, we remind the reader that we make only one choice: the value of L . Ultimately, the choice of L affects value of J , the hypothetical feedthrough term. Regarding stability, the value of L does not matter as long as a positive definite P matrix exists. However, the value of J does indeed affect performance. Let's consider the other variables involved in the Lyapunov stability analysis.

Aside from the weighting matrices, two design scalars S and G , and the matrix F must still be addressed. From Chapter 4, we know that S and G are constrained by,

$$G^T S^T S \geq J. \quad (4.3-18)$$

The matrix F is a function of S and is determined by,

$$F = (S^T S)^{-1} C_p. \quad (4.3-19)$$

We must develop a criterion for the selection of S and G . It is simple to satisfy the stability requirement, but what should their relative sizes be and how much greater than J should $G^T S^T S$ be?

Let's examine the troublesome fourth term of ΔV ,

$$\Delta V_4 = -2\Delta u^T(k) G^T S^T S q(k). \quad (5.4-13)$$

This term is linear with respect to $q(k)$, meaning that we cannot readily assess its sign. To examine ΔV_4 more closely, let's substitute Eq. (4.3-23) for $\Delta u(k)$ and Eq. (4.3-24) for $q(k)$.

$$\Delta V_4 = -2(\hat{u}_p^*(k) - \hat{u}_p(k))G^T S^T S(u_p^*(k) - u_p(k) + \tilde{K}_e C_p e(k)) \quad (5.4-14)$$

If the ideal CGT control law is correct then this term equals zero. When the plant is incorrectly modeled, however, the magnitude of this term correlates to the severity of the erroneous CGT gains. In Chapter 4, we discussed the fact that this term will not compromise stability. It can, however, lead to a compromise in performance. As we formulate various Lyapunov equations, we notice that terms are not always in a quadratic form. When terms are not quadratic it is impossible to evaluate their sign. When faced with such a predicament, the most common remedy is to constrain design variables such that the particular term equals zero, regardless of the vectors involved. Unfortunately, we can not use this method for our case. The matrix product $G^T S^T S$ must be greater than or equal to J (which is positive definite). Therefore, we are unable to set the term to zero. At best, we minimize the effect of ΔV_4 by setting,

$$G^T S^T S = J. \quad (5.4-15)$$

As mentioned previously, the choice of L leads to the solution for J . Using the solution steps from Eq. (5.4-1) through Eq. (5.4-12), we can iterate and find a minimum value of J . We assume that L has the form,

$$L = vI \quad (5.4-16)$$

Incrementing the value of v for each iteration, we select the lowest value of J . In the next chapter, we will assess our assumption regarding the form of L .

Let's compare the discrete MRAC algorithm to the case I continuous MRAC algorithm. The fundamental difference lies in the definition of the vector, $v(\cdot)$. From Eq. (2.4-9),

$$v(t) = C_p e(t).$$

In contrast, the discrete algorithm defines this vector as

$$\hat{v}(k) = Fe(k) + G\hat{z}(k).$$

Let's assume that by constraining

$$F = (S^T S)^{-1} C_p = C_p, \quad (5.4-17)$$

implying that

$$(S^T S)^{-1} = I, \quad (5.4-18)$$

we isolate G as the major distinction between the two algorithms. Our choice of S will be factored into the weighting matrix selection process and has no independent significance. The relaxed constraints on the open-loop plant may now be traced exclusively to G (e.g., if $G = 0$ then the algorithms are the same except for their time operators). The scalar G is defined simply as,

$$G^T S^T S = G^T (I) = G = J \quad (5.4-19)$$

We treat G as a measure of MRAC's reliance on knowledge of the plant's dynamics. The lower G is, the less we need be concerned with poor modeling. By minimizing J , we achieve this.

5.4.2 Choosing Weighting Matrices

The weighting matrices are the most dominant factor in generating the adaptive gains. Ideally, we want the sum of the proportional and integral gains to equal the correct CGT gains at all times. Depending on the sampling frequency, such an achievement would produce near perfect performance.

During a response, the proportional gain should dominate the transient stage while the integral gain grows to "good" values before achieving steady-state. If we operated under ideal conditions, for which $u_p^*(k)$ is correct, there would be no need for an integral gain because our steady-state is zero. In reality, the integral gain does play an important role in the transient response. When operating in the presence of plant modeling error, the proportional gains become more oscillatory. This is because we design the proportional gains to equal the nominal CGT gains. Since these CGT gains are often incorrect, the proportional gains "jump" to different values, trying to find an acceptable operating region. The integral gains, if sized appropriately, act to "remember" which proportional gains best minimize output error. Thus the integral gains grow, becoming more dominant, and should eventually obscure the effects of the proportional gains at the end of the transient response.

Before delving into our procedure for selecting weighting matrices, we need to address the issue of gain convergence. Whereas we design the MRAC *control signal* to converge to a perfect CGT *control signal*, we can not claim that the MRAC *gains* converge to the CGT *gains*. From Sobel (1980), the model's trajectories, u_m and x_m , must

"contain a sufficient number of distinct frequencies" for this type of gain convergence to occur.

We select the proportional weighting matrix to produce a control that approximates the CGT control. Placing the proportional gain in the vicinity of the CGT gain will promote adaptive convergence to that gain. If the CGT gain is not based on the correct plant model, then the proportional adaptive gain will diverge. This is not a problem, however. The key to this endeavor is selecting the relative sizes of values within the proportional weighting matrix. We make the MRAC algorithm relatively insensitive to the magnitude of initial conditions; this is important because MRAC is nonlinear. Setting the weighting matrices in such a fashion, helps the gains to converge to values that minimize error. To accomplish these tasks, we use the information at our disposal. We pick representative values of the model trajectories and estimate a representative value of the output error.

Kaufman *et al.* (1994) suggest selecting " \bar{T} . . . such that K_p , for a representative value of x_m^0 of the model state vector at a representative error e_y^0 , will be equal to the CGT gain." This corresponds to the first continuous case of MRAC. Using their concept, we approach the discrete case.

We must select "reasonable" trajectories to calculate the proportional weighting matrix. These trajectory values correspond to the period of time for which the proportional gains are most dominant. This occurs during the during early transients. Most notably, the trajectories consist of position and velocity of the model. Since the

model has no command input, we are not concerned with the input component of the proportional weighting matrix. There are a number of possibilities for selecting representative values of the model's position and velocity. We may consider their averages, maximums, values at critical points, etc. The difficult part, however, is selecting the representative output error.

We define the representative model trajectories at the time average between the critical points of position and velocity that do not correspond to values of zero (Appendix D provides the method of critical point calculation). As an example, let's consider the reference model having an initial displacement. The initial position is a critical point and the initial velocity, though equal to zero, is also a critical point. The next critical point of velocity corresponds to the first non-boundary condition extreme. We establish the point of trajectory evaluation at 1/2 of the time for velocity to reach its second critical point. At this evaluation point, both velocity and position are 1/4 the way through their oscillatory cycle (if the model is underdamped). If the model is critically damped, velocity has only one critical point that is not a boundary condition. We define our representative model trajectory x_m^0 at the time of evaluation, t_e . We convert t_e to its discrete equivalent k_e by rounding

$$k_e = \frac{t_e}{T_s} \quad (5.4-20)$$

to the nearest integer.

To predict a representative value for the output error e_y^0 , we examine the trajectories of the nominal plant when responding to an initial displacement (identical to the reference model). In doing so, we assume that the plant is uncontrolled. This permits us to use the methods of Appendix D regarding free response. The representative plant trajectories are then evaluated at t_c . The output error at t_c *could* define e_y^0 . This assignment is very conservative because it assumes no control. Instead, we assume the magnitude of representative error to be 0.1% of the representative model velocity's magnitude.

Next, we consider the uncontrolled plant output at t_c and examine the sign of e_y . From this, we determine if the plant's uncontrolled tendency is to lead or lag behind the model's response. Assuming that e_y is not zero from the boundary condition to the critical point (e.g., the reference model is not identical to the plant), we define

$$e_y^0 = 0.001 * \text{sign}(e_y(t_s)) * C_m x_m^0. \quad (5.4-21)$$

We view the representative error as a sort of tolerance. If the true error equals the representative value, we expect $\tilde{K}_e = K_e(k)$. If error exceeds its representative value, the adaptive gain correspondingly grows to improve tracking. By picking the representative error, we are essentially defining our control expectation. However, picking a very small e_y^0 will result in very high weighting matrix values, hence, high computational gains.

As the reader shall see, e_y^0 affects the overall magnitude of the weighting matrices. It does not, however, impact the relative sizes of the weighting values associated with performance.

After generating representative trajectory values, we calculate the proportional weighting matrix. The goal is to create a control signal equal to the correctly modeled CGT input. Since we normally do not have the correct CGT gains at our disposal, we rely on the nominal values. If the plant produces more error than we expect, the adaptive gains can gradually generate a control corresponding to the correct CGT control signal. Basing our calculations on the nominal CGT gains, we establish a reasonable starting point.

We consider the nominal CGT gains as,

$$\tilde{K} = [K_e \quad S_{21}] = [K_e \quad S_{21v} \quad S_{21x}], \quad (5.4-22)$$

where the performance gains, S_{21v} and S_{21x} , correspond to the model's velocity and position, respectively. Since the model has no input, we need not consider any input related gains (e.g., S_{22}). The elements of the proportional weighting matrix are defined as,

$$\bar{T} = \begin{bmatrix} \bar{T}_e & 0 & 0 \\ 0 & \bar{T}_v & 0 \\ 0 & 0 & \bar{T}_x \end{bmatrix}, \quad (5.4-23)$$

where \bar{T}_e , \bar{T}_v and \bar{T}_x correspond to output error, model velocity and model displacement, respectively. We define the representative trajectories as, x_{m1}^0 and x_{m2}^0 , which correspond

to the model's velocity and position, respectively. These trajectories, along with e_y^0 define the representative concatenated vector,

$$r^0 = \begin{bmatrix} e_y^0 \\ x_{m1}^0 \\ x_{m2}^0 \end{bmatrix}. \quad (5.4-24)$$

For the set of representative trajectories, we solve the proportional weighting matrix for the condition that,

$$K_r(k) = \tilde{K}. \quad (5.4-25)$$

From Eq. (2.4-28), the closed-form definition of $u_p(k)$ and Eq. (2.4-19) where

$$u_p(k) = K_r(k)r(k),$$

the overall gain can be written as

$$K_r(k) = \left(I + r^T(k)\bar{T}r(k)G \right)^{-1} \left(K_I(k) + v_1(k)r^T(k)\bar{T} \right). \quad (5.4-26)$$

Substituting this equation into Eq. (5.4-25) yields,

$$\left(I + r^T(k)\bar{T}r(k)G \right)^{-1} \left(K_I(k) + v_1(k)r^T(k)\bar{T} \right) = \tilde{K}.$$

After pre-multiplying both sides by $\left(I + r^T(k)\bar{T}r(k)G \right)$, we arrive at

$$K_I(k) + v_1(k)r^T(k)\bar{T} = \tilde{K} + r^T(k)\bar{T}r(k)G\tilde{K}. \quad (5.4-27)$$

Now, we must consider $K_I(k)$. At the point of evaluation, we assume that $K_I(k)$ has converged to a percentage of the nominal CGT gains, or mathematically, that

$$K_I(k) = \rho\tilde{K}. \quad (5.4-28)$$

Using this assumption we rewrite Eq. (5.4-27).

$$\mathbf{v}_1(\mathbf{k})\mathbf{r}^T(\mathbf{k})\bar{\mathbf{T}} = (1-\rho)\tilde{\mathbf{K}} + \mathbf{r}^T(\mathbf{k})\bar{\mathbf{T}}\mathbf{r}(\mathbf{k})\tilde{\mathbf{G}}\tilde{\mathbf{K}} \quad (5.4-29)$$

Substituting the representative values for their time-varying counterparts and expanding matrix products into vectors and scalars, we produce

$$\mathbf{v}_1^0 \begin{bmatrix} \mathbf{e}_y^0 \bar{\mathbf{T}}_e & \mathbf{x}_{m1}^0 \bar{\mathbf{T}}_v & \mathbf{x}_{m2}^0 \bar{\mathbf{T}}_x \end{bmatrix} = (1-\rho)\tilde{\mathbf{K}} + \left[(\mathbf{e}_y^0)^2 \bar{\mathbf{T}}_e + (\mathbf{x}_{m1}^0)^2 \bar{\mathbf{T}}_v + (\mathbf{x}_{m2}^0)^2 \bar{\mathbf{T}}_x \right] \tilde{\mathbf{G}}\tilde{\mathbf{K}}, \quad (5.4-30)$$

where

$$\mathbf{v}_1^0 = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{e}_y^0 + \tilde{\mathbf{G}}\tilde{\mathbf{K}}\mathbf{r}^0. \quad (5.4-31)$$

We arrange the elements of $\bar{\mathbf{T}}$ onto the left hand side of Eq. (5.4-30) and expand the CGT gain matrices.

$$\begin{aligned} \mathbf{v}_1^0 \begin{bmatrix} \mathbf{e}_y^0 \bar{\mathbf{T}}_e & \mathbf{x}_{m1}^0 \bar{\mathbf{T}}_v & \mathbf{x}_{m2}^0 \bar{\mathbf{T}}_x \end{bmatrix} - \left[(\mathbf{e}_y^0)^2 \bar{\mathbf{T}}_e + (\mathbf{x}_{m1}^0)^2 \bar{\mathbf{T}}_v + (\mathbf{x}_{m2}^0)^2 \bar{\mathbf{T}}_x \right] \mathbf{G} \begin{bmatrix} \mathbf{K}_e & \mathbf{S}_{21v} & \mathbf{S}_{21x} \end{bmatrix} \\ = (1-\rho) \begin{bmatrix} \mathbf{K}_e & \mathbf{S}_{21v} & \mathbf{S}_{21x} \end{bmatrix} \end{aligned} \quad (5.4-32)$$

We construct Eq. (5.4-32) as a system of linear equations to solve for the elements of $\bar{\mathbf{T}}$ in vector form.

$$\begin{bmatrix} \mathbf{v}_1^0 \mathbf{e}_y^0 - (\mathbf{e}_y^0)^2 \mathbf{G}\mathbf{K}_e & -(\mathbf{x}_{m1}^0)^2 \mathbf{G}\mathbf{K}_e & -(\mathbf{x}_{m2}^0)^2 \mathbf{G}\mathbf{K}_e \\ -(\mathbf{e}_y^0)^2 \mathbf{G}\mathbf{S}_{21v} & \mathbf{v}_1^0 \mathbf{x}_{m1}^0 - (\mathbf{x}_{m1}^0)^2 \mathbf{G}\mathbf{S}_{21v} & -(\mathbf{x}_{m2}^0)^2 \mathbf{G}\mathbf{S}_{21v} \\ -(\mathbf{e}_y^0)^2 \mathbf{G}\mathbf{S}_{21x} & (\mathbf{x}_{m1}^0)^2 \mathbf{G}\mathbf{S}_{21x} & \mathbf{v}_1^0 \mathbf{x}_{m2}^0 - (\mathbf{x}_{m2}^0)^2 \mathbf{G}\mathbf{S}_{21x} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{T}}_e \\ \bar{\mathbf{T}}_v \\ \bar{\mathbf{T}}_x \end{bmatrix} = (1-\rho) \begin{bmatrix} \mathbf{K}_e \\ \mathbf{S}_{21v} \\ \mathbf{S}_{21x} \end{bmatrix} \quad (5.4-33)$$

By defining,

$$\eta = \begin{bmatrix} v_1^0 e_y^0 - (e_y^0)^2 GK_e & -(x_{m1}^0)^2 GK_e & -(x_{m2}^0)^2 GK_e \\ -(e_y^0)^2 GS_{21v} & v_1^0 x_{m1}^0 - (x_{m1}^0)^2 GS_{21v} & -(x_{m2}^0)^2 GS_{21v} \\ -(e_y^0)^2 GS_{21x} & (x_{m1}^0)^2 GS_{21x} & v_1^0 x_{m2}^0 - (x_{m2}^0)^2 GS_{21x} \end{bmatrix}^{-1}, \quad (5.4-34)$$

we solve for \bar{T} using,

$$\bar{T} = \left| \eta(1-\rho) \begin{bmatrix} K_e & 0 & 0 \\ 0 & S_{21v} & 0 \\ 0 & 0 & S_{21x} \end{bmatrix} \right|. \quad (5.4-35)$$

We added the absolute value function value because \bar{T} is constrained to be positive definite. The relative sizes of terms within \bar{T} are crucial to establish initial condition insensitivity. Although we use the absolute value function, which likely ruins the pure mathematical solution to \bar{T} , the sign of the response error compensates for this discrepancy.

Now we must select the integral matrix T , and estimate its value at the evaluation point to generate ρ . We constrain T to be proportional to \bar{T} such that,

$$T = \mu \bar{T}. \quad (5.4-36)$$

We intend for the integral gain to be dominant (relative to the proportional gain) at the end of the transient response. The weighting matrix ratio μ is then inversely proportional to the settling time of the plant, t_f (finish time). Experimentally, we noticed that choosing

$$\mu = \frac{T_s}{t_f}, \quad (5.4-37)$$

where t_f is calculated as the 95% settling time, produces excellent adaptation. If this ratio is excessive, the integral gain grows too rapidly and performance suffers. If the ratio is too small, the effects of integral gain are negligible. This would defeat the purpose of adaptive control because the controller should “learn” during the transient response.

As an approximation, we assume the integral gain grows linearly with time, during the early transient response. We expect minimal contribution from this gain. We find that assigning,

$$\rho = \frac{t_e}{t_f}, \quad (5.4-38)$$

yields good results in the weighting matrices.

5.5 Model Reference Adaptive Control With Fixed Feedback Gain

The design procedure for MRAC using a fixed output feedback gain is very similar to the fully adaptive MRAC design method. We make some simplifications, however, satisfying the Lyapunov stability constraints are no different. The weighting matrix selection is a reduced form of the previous case.

5.5.1 The Lyapunov Functions and ASPR Constraints

Our first step is to select the fixed output feedback gain. We choose K_e as calculated for the CGT controller. As we previously discussed, this gain optimizes performance for the worst possible plant characteristics without compromising stability.

If we define,

$$\bar{\Phi}_{pm} = (\Phi_{pm} - \Gamma_{pm} K_e C_{pm}), \quad (5.5-1)$$

to replace Eq. (5.4-3), then this part of the MRAC design is identical to that of section 5.4.1.

5.5.2 Choosing Weighting Matrices

We use identical representative trajectories and the same point of evaluation as presented for the fully adaptive MRAC algorithm. In a fashion similar to section 5.4.2, let the nominal CGT gain be represented as,

$$\tilde{K} = \begin{bmatrix} S_{21v} & S_{21x} \end{bmatrix}. \quad (5.5-2)$$

We define the proportional weighting matrix as,

$$\bar{T} = \begin{bmatrix} \bar{T}_v & 0 \\ 0 & \bar{T}_x \end{bmatrix} \quad (5.5-3)$$

We then form the concatenated trajectory vector, given as,

$$r^0 = \begin{bmatrix} x_{m1}^0 \\ x_{m2}^0 \end{bmatrix}. \quad (5.5-4)$$

Using the design goal of

$$K_r(k_e) = \tilde{K},$$

we rewrite the expression for the overall gain as a closed-form solution.

$$\left(I + r^T(k_e) \bar{T} r(k_e) G \right)^{-1} \left(v_1(k_e) r^T(k_e) \bar{T} + K_1(k_e) \right) = \tilde{K} \quad (5.5-5)$$

Pre-multiplying both sides by $\left(I + r^T(k_e) \bar{T} r(k_e) G \right)$ yields,

$$v_1(k_e) r^T(k_e) \bar{T} + K_1(k_e) = \tilde{K} \left(I + r^T(k_e) \bar{T} r(k_e) G \right).$$

We then subtract the integral gain from both sides and make the substitution of $K_I(k_e) = \rho \tilde{K}$.

$$v_i(k_e)r^T(k_e)\bar{T} = (1-\rho)\tilde{K} + r^T(k_e)\bar{T}r(k_e)G\tilde{K} \quad (5.5-6)$$

We substitute the representative values for the vectors evaluated at k_e and expand the matrix products.

$$v_i^0 \begin{bmatrix} x_{m1}^0 \bar{T}_v & x_{m2}^0 \bar{T}_x \end{bmatrix} = (1-\rho)\tilde{K} + \begin{bmatrix} (x_{m1}^0)^2 \bar{T}_v & (x_{m2}^0)^2 \bar{T}_x \end{bmatrix} G \begin{bmatrix} S_{21v} & S_{21x} \end{bmatrix} \quad (5.5-7)$$

We reconstruct this expression as a system of linear equations.

$$\begin{bmatrix} v_i^0 x_{m1}^0 - (x_{m1}^0)^2 GS_{21v} & -(x_{m2}^0)^2 GS_{21v} \\ -(x_{m1}^0)^2 GS_{21x} & v_i^0 x_{m2}^0 - (x_{m2}^0)^2 GS_{21x} \end{bmatrix} \begin{bmatrix} \bar{T}_v \\ \bar{T}_x \end{bmatrix} = (1-\rho) \begin{bmatrix} S_{21v} \\ S_{21x} \end{bmatrix} \quad (5.5-8)$$

We solve for \bar{T} using,

$$\bar{T} = \left[\eta(1-\rho) \begin{bmatrix} S_{21v} & 0 \\ 0 & S_{21x} \end{bmatrix} \right], \quad (5.5-9)$$

where

$$\eta = \begin{bmatrix} v_i^0 x_{m1}^0 - (v_i^0)^2 GS_{21v} & -(x_{m2}^0)^2 GS_{21v} \\ -(x_{m1}^0)^2 GS_{21x} & v_i^0 x_{m2}^0 - (x_{m2}^0)^2 GS_{21x} \end{bmatrix}^{-1}. \quad (5.5-10)$$

The selection of the integral weighting matrix T is identical to that of the fully adaptive MRAC algorithm where

$$T = \mu \bar{T} \quad (5.4-36)$$

and

$$\mu = \frac{T_s}{t_f} . \quad (5.4-37)$$

5.6 Summary

In this chapter, we presented our design methods for four different control algorithms: pole placement; CGT; fully adaptive MRAC; and MRAC with a fixed output feedback gain.

The pole placement design reveals little insight as this is a very common control method. The only decision regarding design pertains to the selection of the reduced order estimator gain.

Selecting the performance gains for the CGT controller is a function of the algorithm formulation in Chapter 2. In this chapter, solving for these gains involved a canonical form transformation which may or may not be the only possible approach. We choose our feedback gain to optimize the performance for the worst possible plant modeling scenario.

For both of the MRAC algorithms we address every design step from satisfying stability constraints to suggesting a method of weighting matrix selection. We justify every decision, but in some cases our conclusions are based upon empirical knowledge and not analysis. In Chapter 6, we will demonstrate the effectiveness of our design which requires no user-controller interaction.

In Fig. 5-5, we provide our design paths for each type of algorithm. Using the procedures in this chapter, we will empirically compare each control method in Chapter 6.

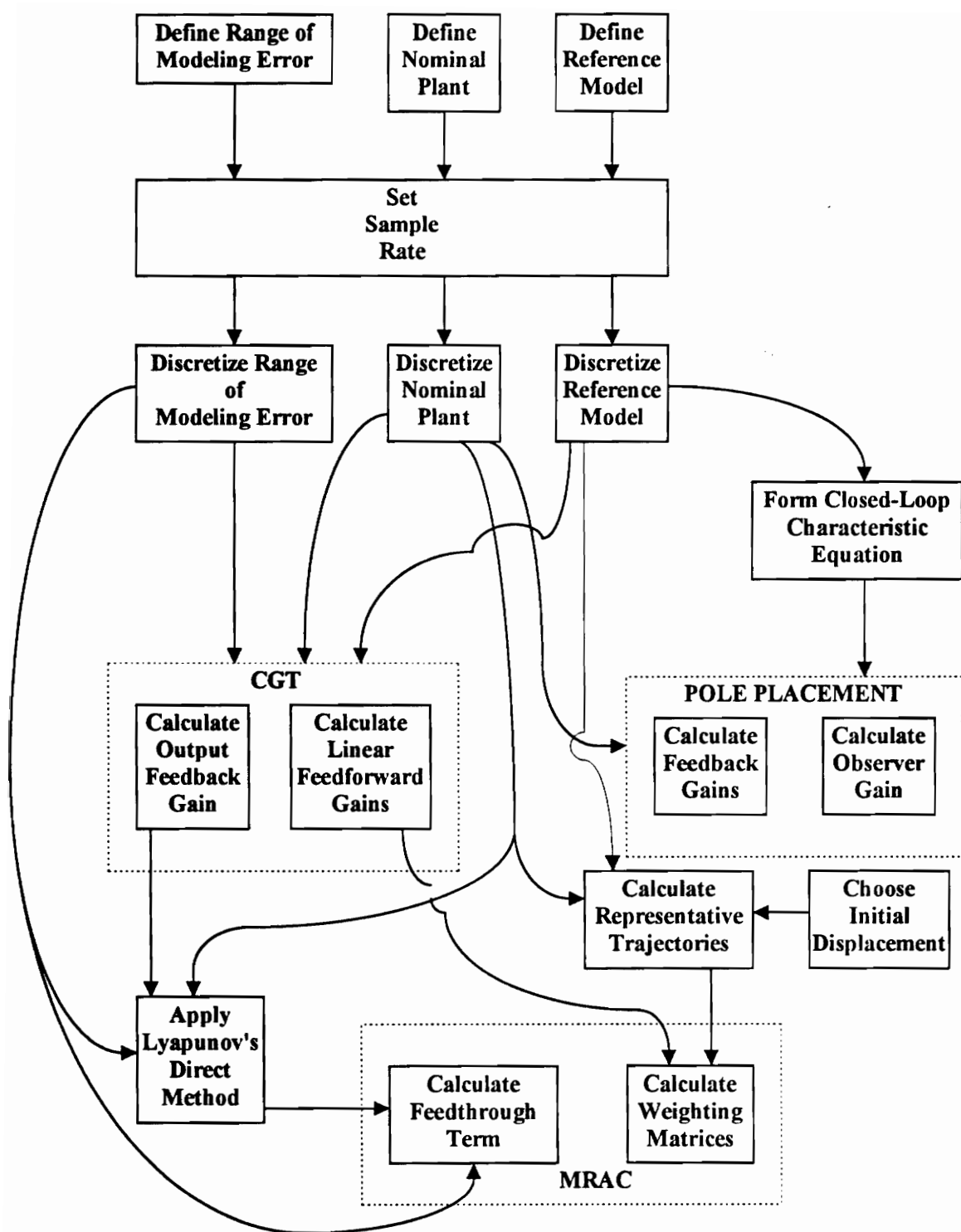


Figure 5-5
Design Flowchart

Chapter 6

Results

In this chapter, we present example plant responses to illustrate the effectiveness of our control strategies. We also address assumptions made in previous chapters, pointing out their shortcomings and their assets.

In the latter half of this chapter, we compare the robustness of all four controllers subjected to a broad range of testing conditions. We demonstrate that adaptive control does not always perform better than linear control. We do not mean to imply, however, that linear control is better in general. Linear control appears advantageous for some circumstances.

6.1 Overview

We begin by testing our discrete CGT design. Recall from Chapter 2 that our formulation is original. Although we inferred our results to be identical to Sobel's (1980), our gain solution process in Chapter 5 differs from his approach. In spite of these subtle differences, our discrete CGT controller performs as expected.

In the following section, we provide sample response plots of our two discrete types of MRAC. We assume explicit plant knowledge for this presentation. We demonstrate that our weighting matrix selection process (from Chapter 5) yields "good" results. For each example, we declare our weighting matrix values, the ratio of the proportional weighting matrix to the integral weighting matrix, and a quantitative measure

of performance. To measure performance we use a performance index (PI), defined as,

$$PI = \frac{1}{k_f} \sum_{i=1}^{k_f} |C_p e(i)|. \quad (6-1)$$

Physically, the performance index is a measure of the average absolute error between the velocities of the plant and the reference model. We normalize PI by dividing the summation result by the length of each simulation. Dividing by the simulation length, PI does not penalize responses having relatively long settling times.

Recall from Chapter 5 that we base weighting matrix selection on a representative value of the output error. We will assess the choice of

$$e_y^0 = 0.001 * \text{sign}(e_y(t_c)) * C_m x_m^0. \quad (5.4-21)$$

In this section, we also simulate MRAC responses using an alternative, though common, weighting matrix selection process. For this scenario, the designer tunes the weighting matrices to achieve desired effects. We demonstrate that our method yields equivalent results without any tuning.

In section 6.4, we examine the four controllers using various design objectives in the presence of plant modeling error. Afterwards, we assess our proposal to minimize the non-quadratic Lyapunov term. This involves a minimization of the feedthrough term G . We look at the changes in the performance index for varying values of G .

The final section includes mesh plots of performance index over the full range of plant modeling error. We consider all four algorithms for several types of performance objectives. This analysis reveals which algorithms are best suited for particular types of

performance goals.

6.2 Discrete Model Reference Control Using the Command Generator Tracker Approach

Let's begin by assuming that the initial conditions between the plant and the model differ. Setting the plant's initial displacement and velocity to zero, the control must actuate the plant from rest. The response then converges to the output of the reference model. This demonstrates one role of the output error feedback gain, K_e . In Fig. 6-1a and Fig. 6-1b, we provide two different examples of CGT.

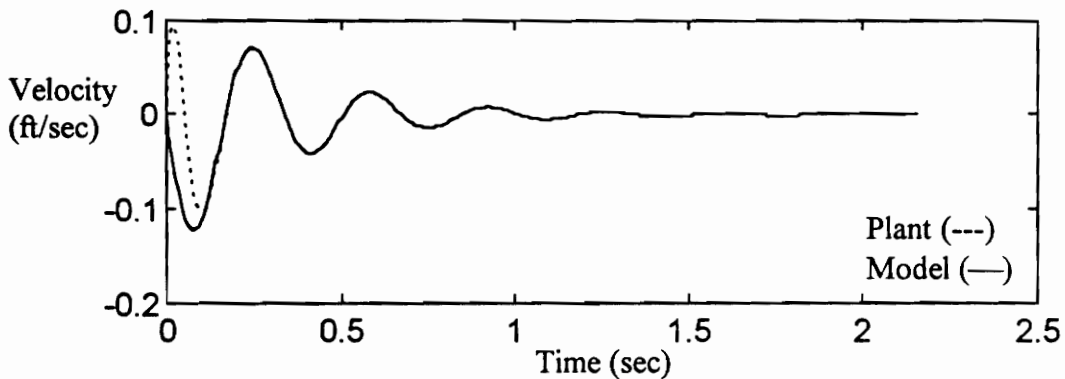


Figure 6-1a
Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

In each of these plots, we can see that the output feedback gain serves a role very similar to a state estimator gain. The initial error is fed through K_e , causing actuation that is additional to the open-loop contribution of the performance gain S_{22} . Because the feedback gain specifies stable dynamic properties, error decays and the plant eventually

tracks the output of the reference model. In Fig. 6-1b, however, the contribution of the feedback gain is not sufficient enough to create a “good” response. The settling time specified in the reference model exceeds the ability of the feedback gain to actuate a reasonable decay in tracking error. Although the dynamics of error are “quicker” than the settling time of the model, we can see that the plant’s response is somewhat undesirable. If we had explicit knowledge of the plant’s dynamics, we could push the feedback gain to its absolute limit.

Let’s imagine a reference model that specifies critical damping. There exists no velocity feedback gain capable of producing satisfactory error transients. Although we do not demonstrate this, the reader may see in Fig. 6-1b how bad the response for a model damping ratio of 0.61 is. Suffice it to say, setting the initial conditions of the plant and model to be equal is a wise precaution.

For the remainder of this chapter, we only consider cases in which the reference

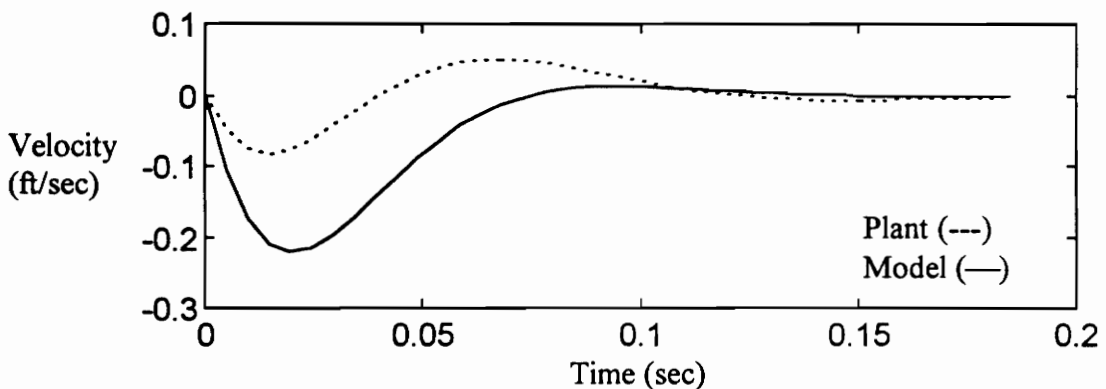


Figure 6-1b
Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.61$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

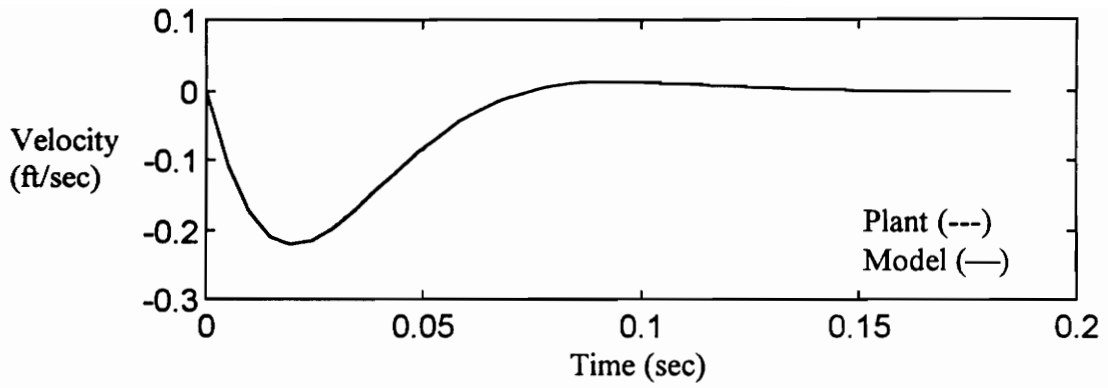


Figure 6-2a

Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

model and the plant have identical initial conditions. This enables us to better judge a controller by its performance gains rather than how effectively it rejects initial condition error. In Fig. 6-2a, we illustrate a scenario equivalent to that of Fig. 6-1b except there is no initial condition error.

As we can see in Fig. 6-2a through Fig. 6-2d, the CGT approach effectively

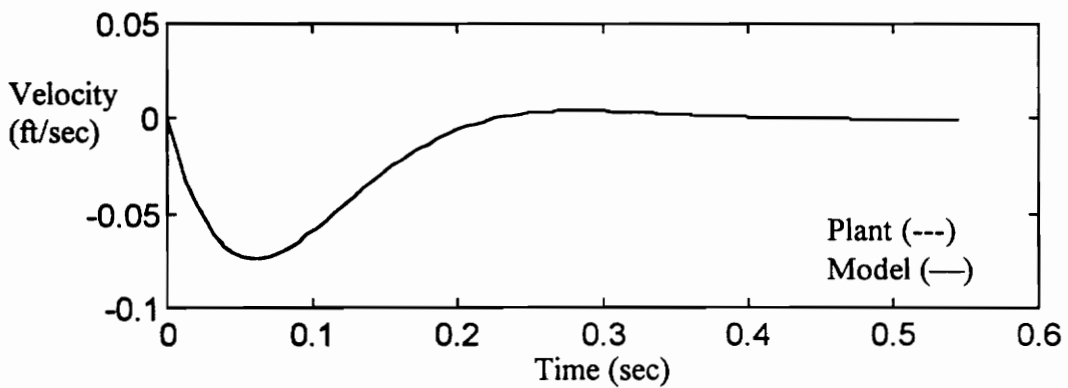


Figure 6-2b

Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.61$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

controls the plant. This method is as effective as pole placement. Since the performance gains contribute as an open-loop control, this approach has stability advantages over pole placement. This claim will be further evidenced in subsequent sections.

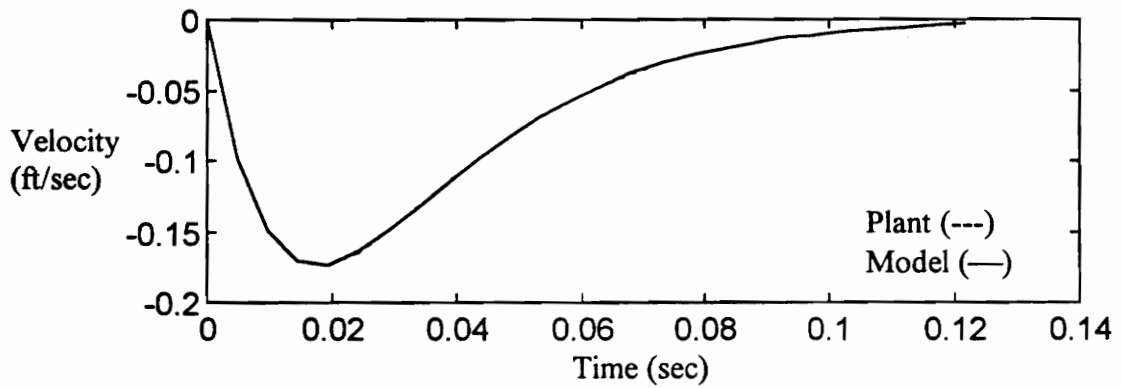


Figure 6-2c

Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.00$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

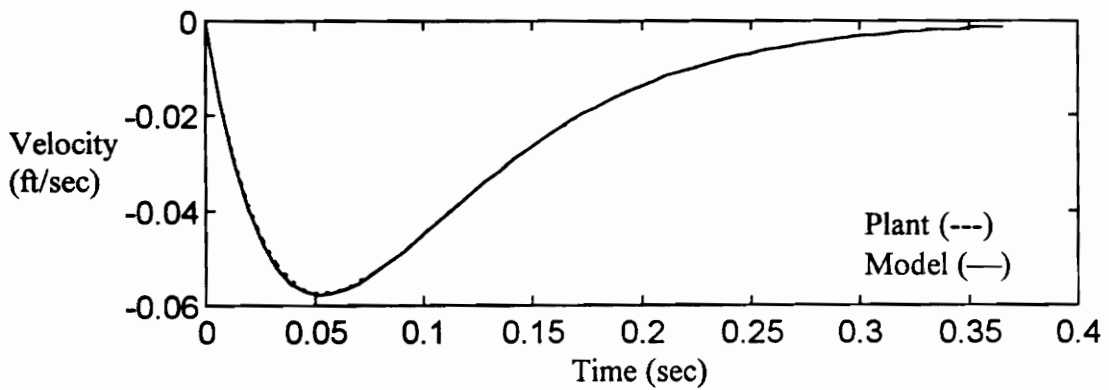


Figure 6-2d

Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

6.3 Model Reference Adaptive Control

In this section, we document example responses of the plant controlled by the two types of discrete MRAC. We span a broad and representative range of the possible design objectives.

We then look at MRAC responses to assess our choice of the representative error, used in weighting matrix selection. Recall from Chapter 5, we assumed the representative error to be 0.1% of the model's velocity at the point of evaluation. We will demonstrate that increasing this estimate, lowers the overall magnitude of the weighting matrices. By the same token, a relatively low estimate of the representative error tends to produce a relatively low performance index.

In Chapter 5, we based our weighting matrix calculations on predicted values of the model state trajectories and the estimation of error. Using these values, we calculated weighting matrices to generate gains equal to the CGT control. Sometimes this is not mathematically possible (because the procedure can require negative elements in the weighting matrices). However, the elements in these matrices are still sized appropriately. Their overall magnitude is mostly a function of estimated output error, but this does not interfere with relative sizing within the matrices.

In nearly every case that we reviewed, weighting matrix selection seemed to be a trial and error process. Most of the weighting matrices are based on the identity matrix, multiplied by an overall gain. Having no relative differences among the elements in these weighting matrices, it is obvious that some elements can contribute to unnecessarily high

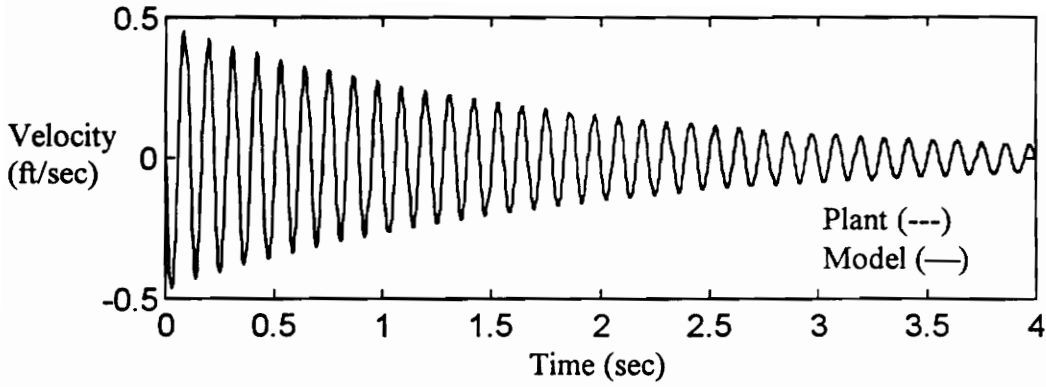
computational gains. In our sterile testing environment, this is of no concern. In this section, we will illustrate the effects of increasing the overall gain on an identity-based weighting matrix. We then compare the resulting performance with our MRAC design.

6.3.1 Model Reference Adaptive Control for Explicit Knowledge of Plant Dynamics

For each set of design goals, we illustrate the fully adaptive MRAC system and MRAC using the fixed output feedback gain. Our design goals cover maximum and minimum possible reference model specifications. We also examine a moderately demanding reference model. In these simulations, the model of the plant is perfect (the nominal plant and actual plant are identical).

Along with each response, we provide the proportional weighting matrix, the ratio between the proportional and integral weighting matrix, and the performance index. Doing so, we can highlight differences between the fully adaptive MRAC system, MRAC using a fixed gain, and the MRAC system designed with alternative weighting matrix selection.

Let's first examine the adaptive control cases involving a high positive change in the plant's natural frequency with no change in the plant's damping. Figures 6-3a and 6-3b illustrate the plant response associated with each of the two MRAC approaches. Note that the demands of the reference model translate to a relatively high response settling time. So that the reader may examine the response with clarity, we truncate the response substantially. The performance indexes, however, represent the entire time of response.



$$\bar{T} = \begin{bmatrix} 1.471(10)^9 & 0 & 0 \\ 0 & 1.397(10)^5 & 0 \\ 0 & 0 & 3.030(10)^9 \end{bmatrix}$$

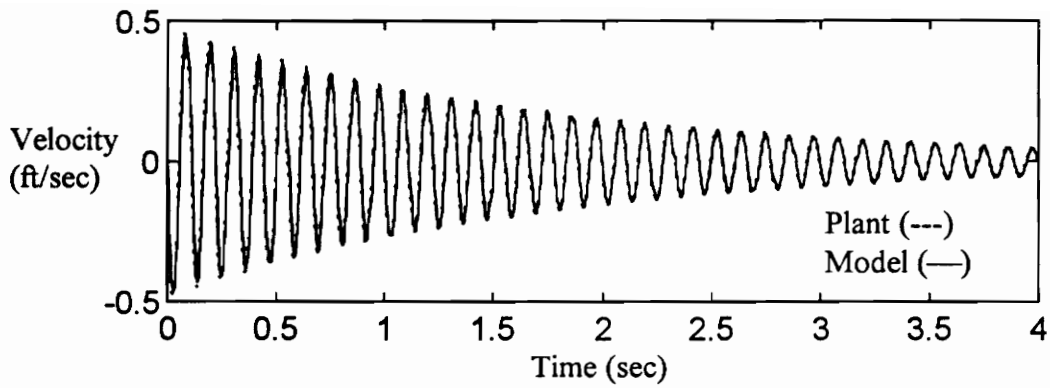
$$T = 7.959(10)^{-4} \bar{T}$$

$$PI = 2.397(10)^{-4} \text{ ft / s}$$

Figure 6-3a

Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

By examining these two response, we see that MRAC achieves the performance specifications of the reference model. The fully adaptive MRAC performs somewhat better than the MRAC using the fixed gain. As we can see from the plots, this difference is not significant. Note that the weighting matrix gains, associated with performance, are identical in both cases. We expect our design procedure to produce this effect.



$$\bar{T} = \begin{bmatrix} 1.397(10)^5 & 0 \\ 0 & 3.030(10)^9 \end{bmatrix}$$

$$T = 7.959(10)^{-4} \bar{T}$$

$$PI = 4.531(10)^{-4} \text{ ft / s}$$

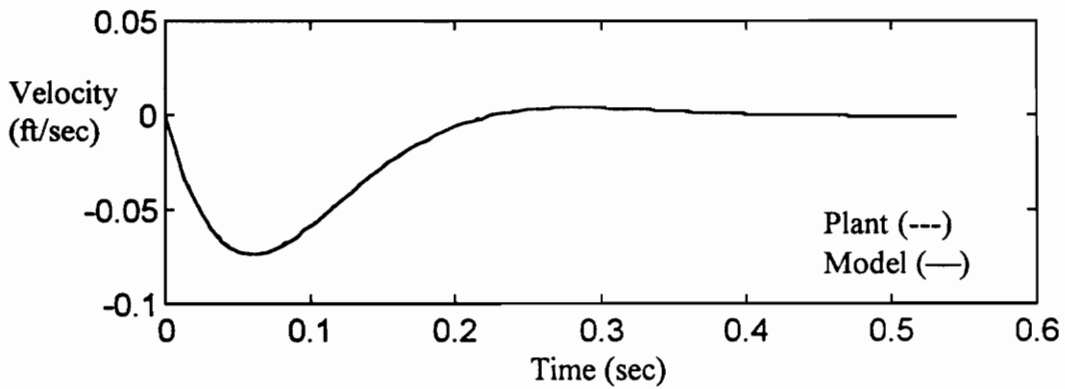
Figure 6-3b

Discrete MRAC Using a Fixed Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

One advantage of using a fixed gain MRAC lies in the lack of a potentially high weighting gain associated with output error. The control signal resulting from this computational term is a function of output error cubed. Using the fixed feedback gain, the control signal is linear in output error. If noise is present in the output, the negative effects are, first, propagated by the cubic function of error. Then, the high weighting gain worsens this predicament. Using a fixed feedback gain, although it may not optimize performance in the ideal case (no signal noise), may produce better results in a “real

world” application.

In figures 6-4a and 6-4b, we can examine the effectiveness of MRAC applied to the lowest possible design frequency with moderate damping. Again, we see that the fully adaptive MRAC performs marginally better than its fixed gain counterpart. Both, however, demonstrate further that our design procedure is effective in controlling the plant over a wide range of performance objectives.

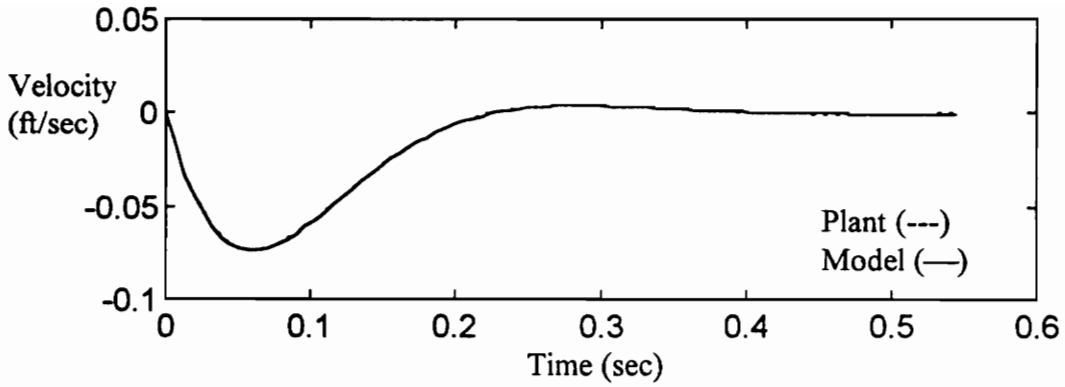


$$\bar{T} = \begin{bmatrix} 6.081(10)^{10} & 0 & 0 \\ 0 & 2.461(10)^7 & 0 \\ 0 & 0 & 8.892(10)^9 \end{bmatrix}$$

$$T = 2.353(10)^{-2} \bar{T}$$

$$PI = 8.650(10)^{-5} \text{ ft / s}$$

Figure 6-4a
Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$



$$\bar{T} = \begin{bmatrix} 2.461(10)^7 & 0 \\ 0 & 8.892(10)^9 \end{bmatrix}$$

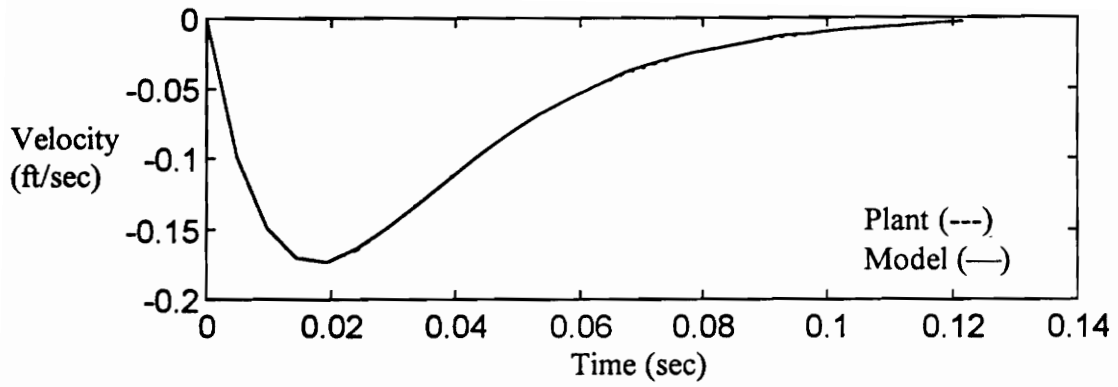
$$T = 2.353(10)^{-2} \bar{T}$$

$$PI = 1.033(10)^{-4} \text{ ft / s}$$

Figure 6-4b
Discrete MRAC Using a Fixed Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.67$ with
 $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

We extend our survey to a reference model demanding the lowest settling time. Frequency requirements are maximum and damping is critical. Figures 6-5a and 6-5b demonstrate the control action. The weighting matrices in this case tend to be relatively high. This makes sense because the corresponding CGT gains are also relatively high.

Note that the ratio between the weighting matrices is considerably larger than our previous examples. We want our integral gains to achieve effective performance values before the plant transients expire. It is difficult to comment on the integral gains at this



$$\bar{T} = \begin{bmatrix} 6.945(10)^9 & 0 & 0 \\ 0 & 1.259(10)^7 & 0 \\ 0 & 0 & 2.745(10)^9 \end{bmatrix}$$

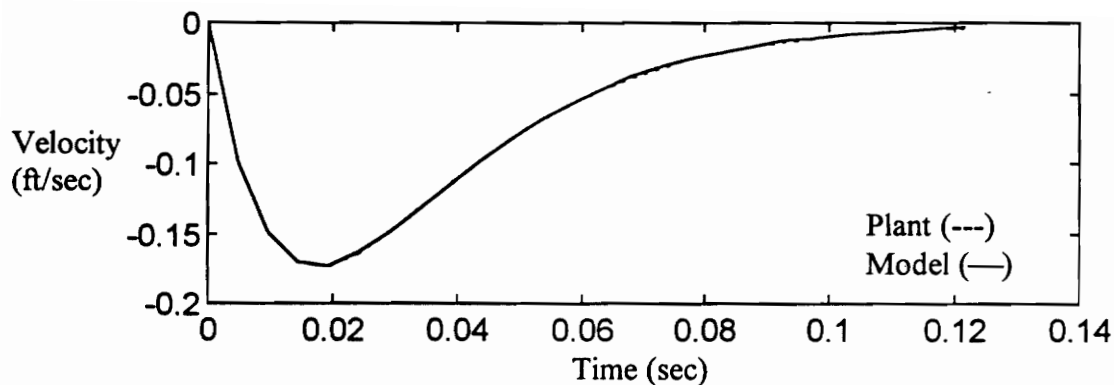
$$T = 0.08\bar{T}$$

$$PI = 2.581(10)^{-4} \text{ ft / s}$$

Figure 6-5a

Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

point. For a purely transient plant response (where the steady-state is zero) and no modeling error, the integral gains effectively serve no purpose. Until we examine the effects of plant modeling error, it is difficult to demonstrate their significance. In this section our purpose was only to show that the sizing of the integral weighting matrices do not produce adverse effects for this ideal case.



$$\bar{T} = \begin{bmatrix} 1.259(10)^7 & 0 \\ 0 & 2.745(10)^9 \end{bmatrix}$$

$$T = 0.08\bar{T}$$

$$PI = 2.658(10)^{-4} \text{ ft / s}$$

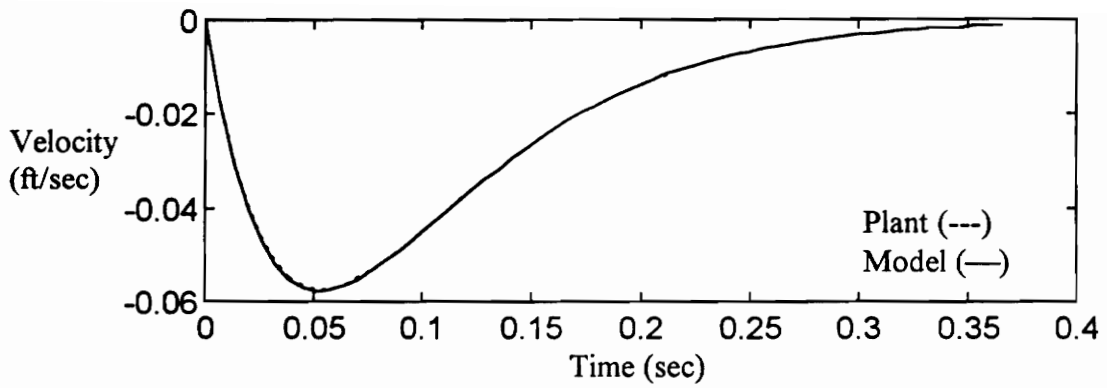
Figure 6-5b

Discrete MRAC Using a Fixed Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

Next, we look at another critically damped case that requires the maximum reduction in the plant's natural frequency. This performance requirement is successfully satisfied by the two MRAC algorithms depicted by figures 6-6a and 6-6b. Note that the performance index of the fixed gain MRAC is slightly lower than the fully adaptive version.

When model frequencies are lower than the plant's, we notice that the adaptive proportional feedback gain is initially negative. This corresponds to negative output error for which the plant's output is greater than the model's. As output error, the feedthrough term, $Gz(k)$, becomes dominant in defining the adaptive gains. In this feedthrough term (recall from Chapter 5), we select \tilde{K}_e to be the intended adaptive feedback gain. Failure to converge results in a “penalizing” actuation. Our choice of \tilde{K}_e equals the value of the positive CGT feedback gain. If the adaptive feedback gain enters a negative region to minimize error, it will then grow to positive values when the sign of error becomes positive.

Since the adaptive feedback gain remains in the negative region for some time, the integral component grows negative as well. When the proportional gain changes to a positive value, the total feedback gain remains negative for a brief period of time while the integral component approaches a positive value. We speculate that this sudden shift in value, results in an adaptive feedback gain that is mildly detrimental to performance for a short while. We assume that our observations of gain behavior explain the differences between the two MRAC systems. This drastic feedback gain shift does not occur when the model “leads” the plant. When the model frequency is larger than the plant's (the model leading the plant), we note that the fully adaptive MRAC achieves better performance than the fixed gain MRAC.



$$\bar{T} = \begin{bmatrix} 6.102(10)^{10} & 0 & 0 \\ 0 & 3.779(10)^7 & 0 \\ 0 & 0 & 8.848(10)^9 \end{bmatrix}$$

$$T = 3.509(10)^{-2} \bar{T}$$

$$PI = 8.82(10)^{-5} \text{ ft / s}$$

Figure 6-6a

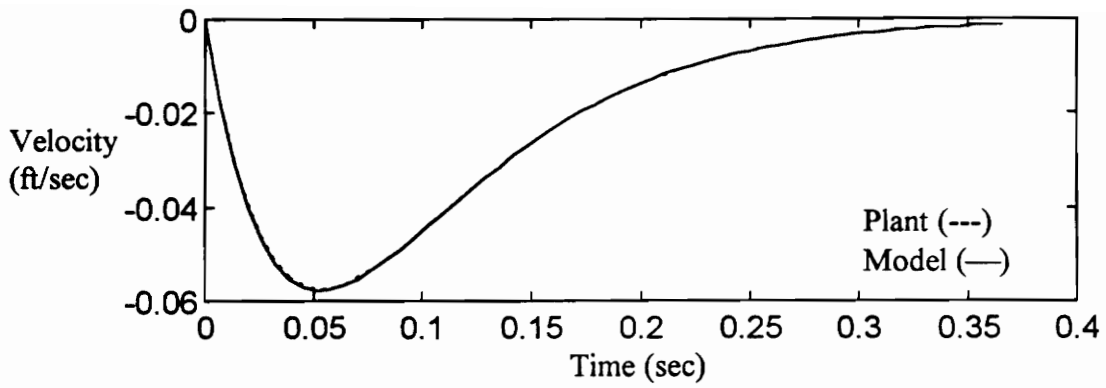
Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

Let's consider the case of critical damping. The model specifies repeated poles on the real axis of the z-plane. An output feedback gain can only specify the error's rate of decay by a finite amount. For reference models having low settling times, error dynamics are sometimes "slower" than the dynamics of the model. In the case illustrated by Fig. 6-5 the decay rate of error is approximately twice as slow as the reference model. In the case of Fig. 6-6, however, due to the lower natural frequency, error transients are about 1.5 times quicker than the transients of the reference model. Recall that the fully adaptive

MRAC is designed to converge to the “neighborhood” of the CGT feedback gain, or equivalently, the feedback gain used by the fixed gain MRAC. Let’s consider the first case. The feedback gain, calculated conservatively for the worst possible plant, cannot produce error transients quicker than the model. The fixed gain MRAC is condemned to use this value. The fully adaptive MRAC, on the other hand, converges to gains that minimize error. The adaptive feedback gain can exceed the conservative threshold calculated for the CGT controller, thus producing a better response than the fixed gain MRAC.

In the second case, the fixed gain MRAC performs slightly better than its counterpart. Requiring the plant to have a lower natural frequency means that initially, the reference model will lag behind the plant. Although the adaptive feedback gain converges to the CGT gain region, the proportional element of the overall gain does not completely nullify the adverse effects of the integral component. For a longer response time this situation should be rectified. However, heavily damped performance requirements tend to expose a weakness in adaptive control when only evaluating on the basis of plant transients.

When comparing these two MRAC systems, we identify response time and reference model frequency as two contributing factors to differences in performance.



$$\bar{T} = \begin{bmatrix} 3.779(10)^7 & 0 \\ 0 & 8.848(10)^9 \end{bmatrix}$$

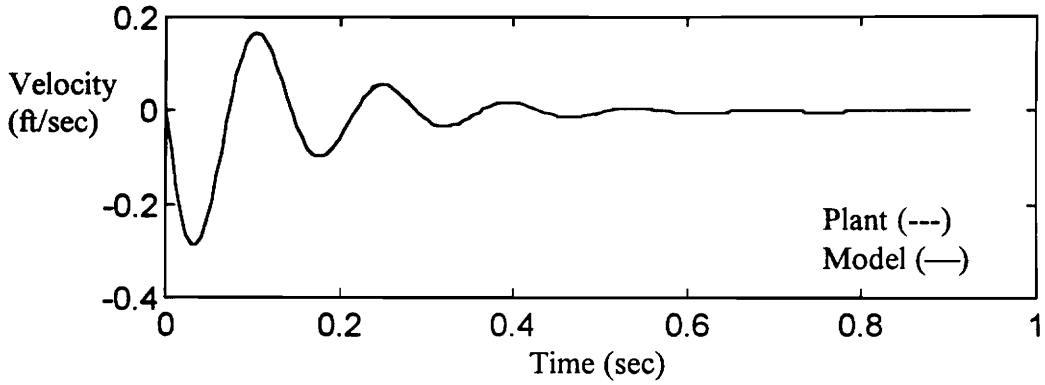
$$T = 3.509(10)^{-2} \bar{T}$$

$$PI = 8.812(10)^{-5} \text{ ft / s}$$

Figure 6-6b
Discrete MRAC Using a Fixed Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$ with $\omega_{pm} = \omega_p$
and $\zeta_{pm} = \zeta_p$

In our final MRAC example, using the ideal plant modeling conditions, we show the response of a plant with moderate performance demands. These responses are provided in figures 6-7a and 6-7b. In this case, the fully adaptive MRAC outperforms the fixed gain MRAC. This result is consistent with our suggestions made in the last example.

To conclude this subsection, we have shown that our MRAC weighting matrix selection process works very well for the operating conditions involving explicit knowledge of plant dynamics. Also, the method with which we choose the ratio between proportional and integral weighting matrices is effective. Although we encountered one



$$\bar{T} = \begin{bmatrix} 2.148(10)^9 & 0 & 0 \\ 0 & 6.642(10)^5 & 0 \\ 0 & 0 & 1.157(10)^9 \end{bmatrix}$$

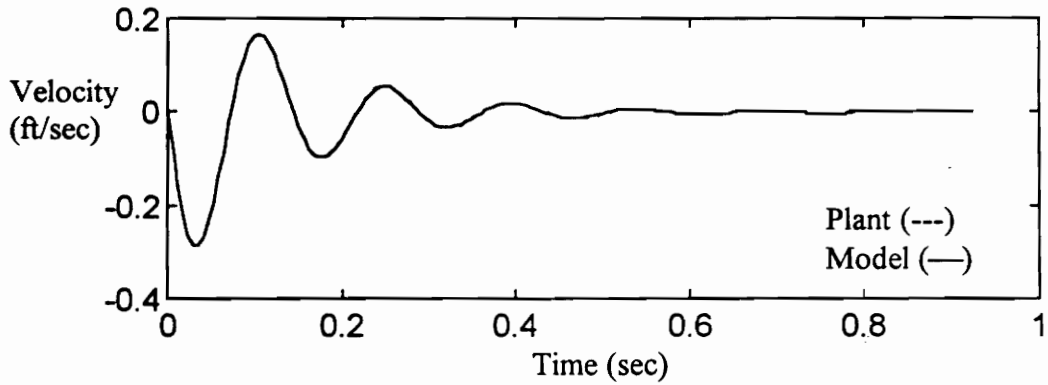
$$T = 1.258(10)^{-2} \bar{T}$$

$$PI = 1.587(10)^{-4} \text{ ft / s}$$

Figure 6-7a

Discrete MRAC for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

instance where the integral error gain caused a mild decline in performance, the problem is self-correcting over time. The role of output feedback (adaptive and fixed) diminishes gradually over the course of the transient response. The other adaptive gains are most responsible for model tracking. We do, however, attribute the slight performance differences (between the two MRACs) to the adaptive feedback gain. This is little more than a tangent and is not a major concern or interest to this study.



$$\bar{T} = \begin{bmatrix} 6.642(10)^5 & 0 \\ 0 & 1.157(10)^9 \end{bmatrix}$$

$$T = 1.258(10)^{-2} \bar{T}$$

$$PI = 3.216(10)^{-4} \text{ ft / s}$$

Figure 6-7b
Discrete MRAC Using a Fixed Gain for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with
 $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

6.3.2 Examining the Choice of Representative Error

In Chapter 5, we based weighting matrix values on a predicted value of output error. Our choice produces relatively high overall weighting matrix values, though it does not affect the relative sizes of the internal performance related values (e.g. T_x). In this section, we evaluate whether or not our estimation is excessively small. We shall see if higher error estimates produce a better performance index. If they do not, we must ask,

“does better performance warrant the use of such high computational gains?”.

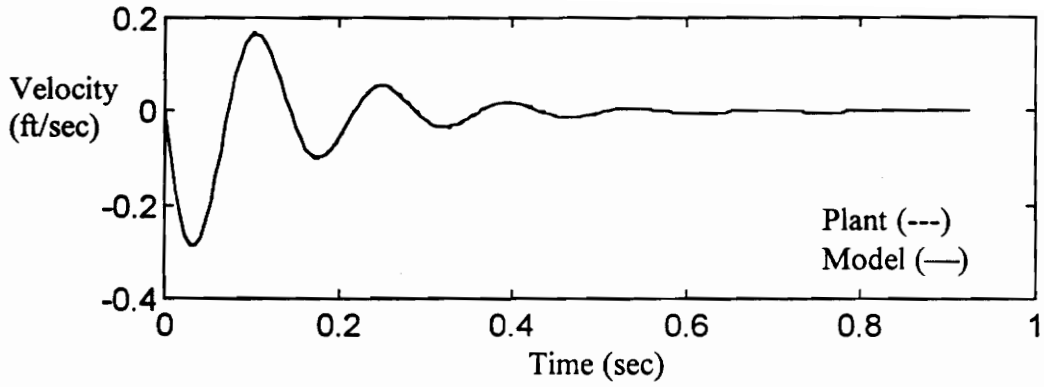
Recall that we select the representative error to be 0.1% of the predicated reference model velocity at the point of evaluation. We shall examine MRAC responses for estimations increased by factors of 10, 100, and 1000. We test each of these values as applied to the performance demands used in Fig. 6-7a. To establish a frame of reference, the critical values from this response are,

$$\bar{T} = \begin{bmatrix} 2.148(10)^9 & 0 & 0 \\ 0 & 6.642(10)^5 & 0 \\ 0 & 0 & 1.157(10)^9 \end{bmatrix}$$

and

$$PI = 1.587(10)^{-4} \text{ ft / s .}$$

In Fig. 6-8a, we see that performance declines with an increase in estimated error. Also, all the terms in the weighting matrix decrease, but not by the same amount. The \bar{T}_x component decreases by an order of magnitude. The element associated with error, however, decreases by two orders of magnitude. This makes sense because the performance gains are a linear function of error while the feedback gain is a quadratic function of error.



$$\bar{T} = \begin{bmatrix} 2.148(10)^7 & 0 & 0 \\ 0 & 6.642(10)^4 & 0 \\ 0 & 0 & 1.157(10)^8 \end{bmatrix}$$

$$PI = 8.601(10)^{-4} \text{ ft / s}$$

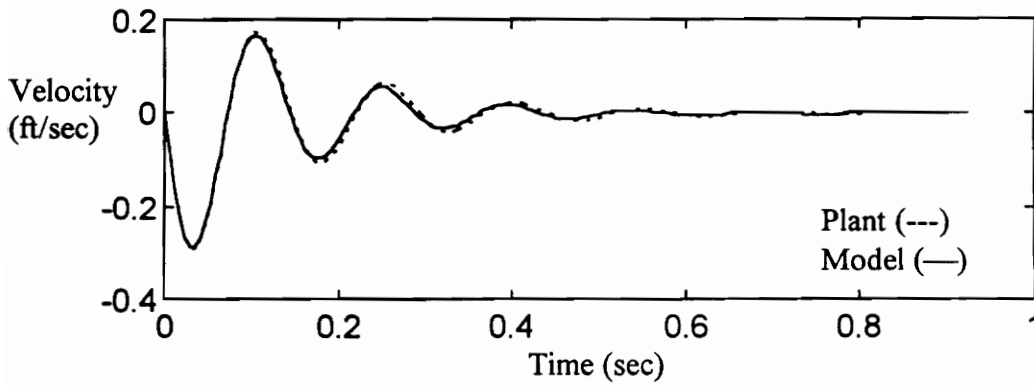
$$e_y^0 = 0.01 * \text{sign}(e_y(t_s)) * C_m x_m^0$$

Figure 6-8a

Discrete MRAC for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

Assuming an Increased Error Estimate by a Factor of 10

In Fig. 6-8b, we increase the error estimate by a factor of 100. The weighting matrix values continue to decrease in a fashion consistent with the previous example. By the same token, performance also declines. In Fig 6-8c, these two trends continue. The effectiveness of the controller becomes undesirable and not worth the advantages of low computational gains.



$$\bar{T} = \begin{bmatrix} 2.148(10)^5 & 0 & 0 \\ 0 & 6.642(10)^3 & 0 \\ 0 & 0 & 1.157(10)^7 \end{bmatrix}$$

$$PI = 4.857(10)^{-3} \text{ ft / s}$$

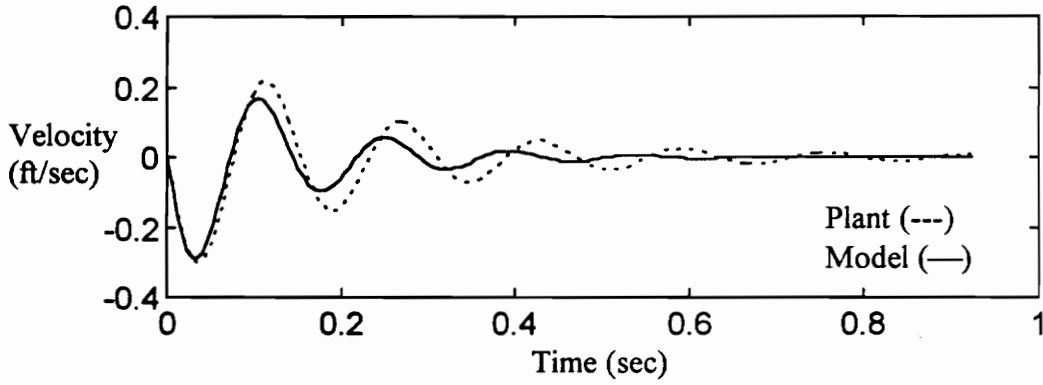
$$e_y^0 = 0.1 * \text{sign}(e_y(t_s)) * C_m x_m^0$$

Figure 6-8b

Discrete MRAC for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$

Assuming an Increased Error Estimate by a Factor of 100

The rate at which the error weighting values decreases is much greater than the weight values directly associated with performance. The responses shown in figures 6-8a and 6-8b are quite reasonable. If we had motivation (the need to minimize disturbance effects, for example), we could lower the overall weighting matrix gain without a large compromise in performance. Since we currently have no need to do this, we shall table such endeavors for a later date.



$$\bar{T} = \begin{bmatrix} 2.148(10)^3 & 0 & 0 \\ 0 & 6.642(10)^2 & 0 \\ 0 & 0 & 1.157(10)^6 \end{bmatrix}$$

$$PI = 2.753(10)^{-2} \text{ ft / s}$$

$$e_y^0 = \text{sign}(e_y(t_s)) * C_m x_m^0$$

Figure 6-8c

Discrete MRAC for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$
Assuming an Increased Error Estimate by a Factor of 1000

6.3.3 Alternative Weighting Matrix Tuning

In many MRAC designs, weighting matrices use the form,

$$\bar{T} = \alpha I, \quad (6-2)$$

where α represents the overall weighting matrix gain. In this section, we demonstrate that this can be an effective approach. However, the resulting weighting matrices are

unnecessarily high, even though they only serve computational roles. Part of the problem is a lack of relative sizing within the weighting matrix. In our examples, we note that the elements within weighting matrices have very different relative sizes. Another problem with the form of Eq. (6-2) is that there is no rational basis for selecting the overall gain. Designers must often test the effectiveness, either on-line or in simulation, to establish reasonable values.

In this subsection, we demonstrate that our method of weighting matrix selection is equivalent, if not better than tuning a matrix of the form given by Eq. (6-2). We show how the designer gradually converges to an appropriate selection. To do this, we select the performance requirements of $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$.

In figures 6-9a through 6-9e, we demonstrate the gradual tuning process necessary to achieve satisfactory performance. We make no claims regarding the number steps taken to accomplish this task. However, it certainly takes more steps than our procedure which requires no tuning at all.

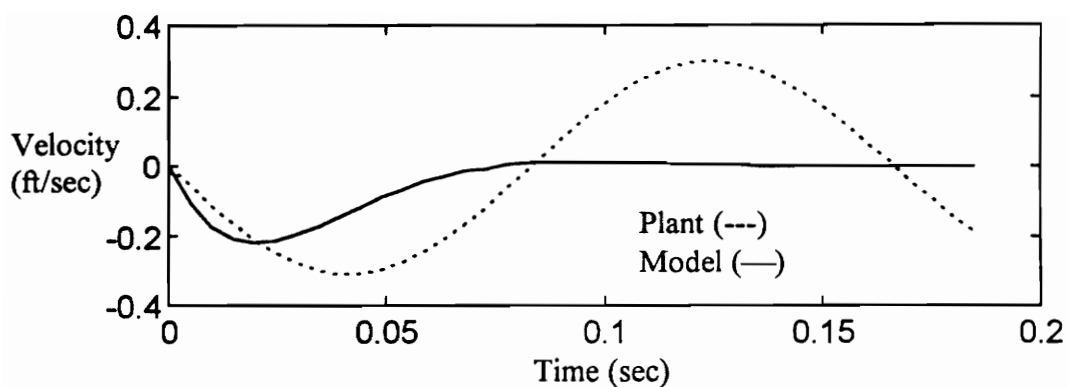


Figure 6-9a
Discrete MRAC Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = 1$

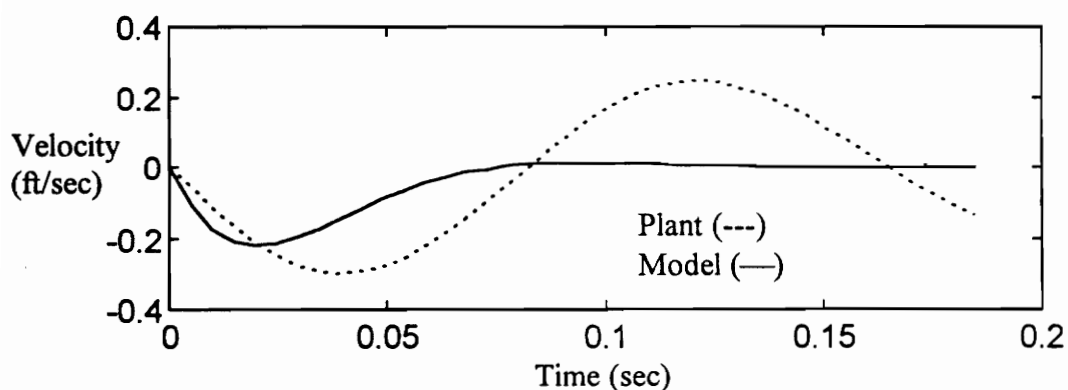


Figure 6-9b
Discrete MRAC Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^4$

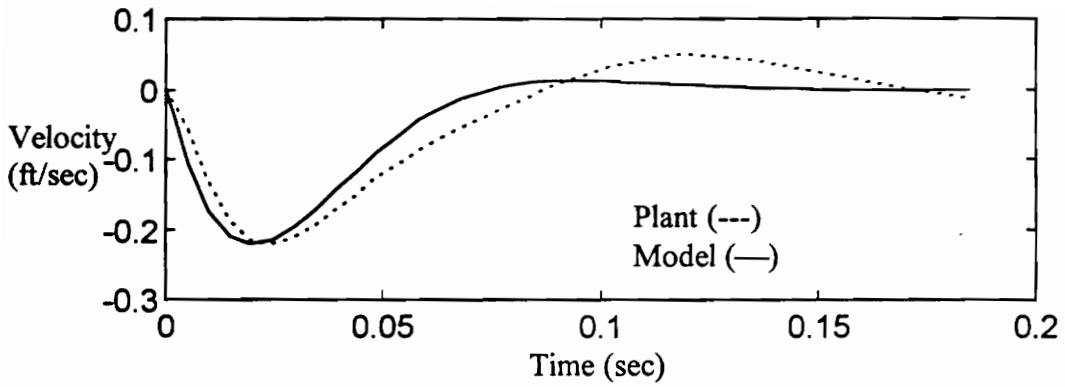


Figure 6-9c

Discrete MRAC Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^4$

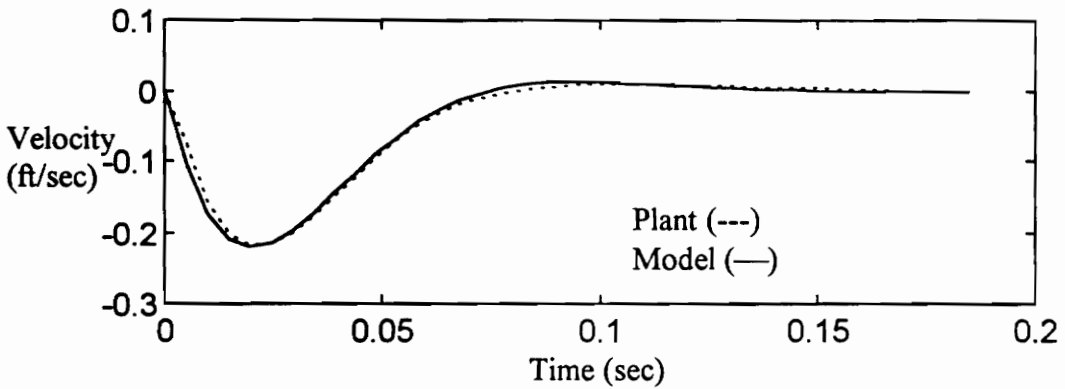


Figure 6-9d

Discrete MRAC Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^6$

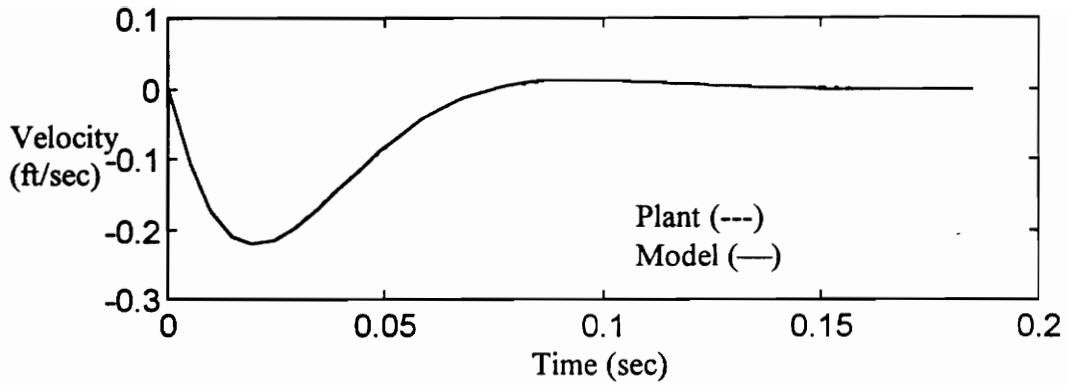


Figure 6-9e
Discrete MRAC Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_{pm} = \omega_p$ and $\zeta_{pm} = \zeta_p$, Using an Alternative Weighting Matrix Selection where $\alpha = (10)^{10}$

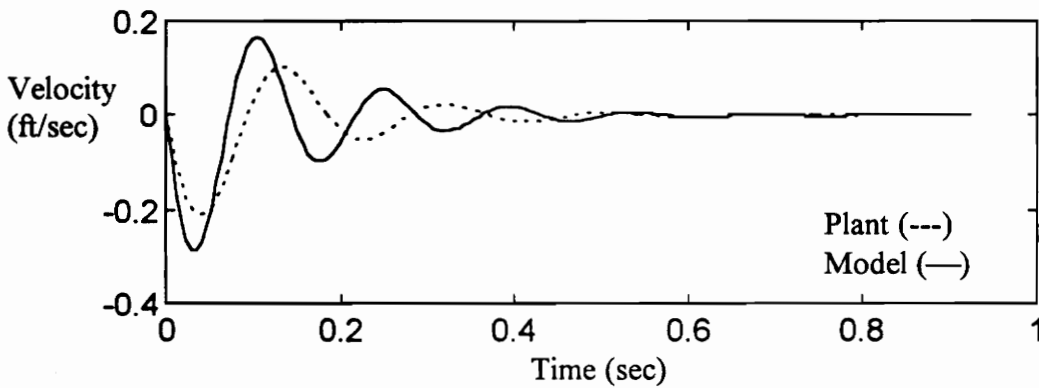
6.4 The Effects of Plant Modeling Error

Thus far, we have only considered the ideal case of correct plant modeling. To compare the four control methods in a more realistic fashion, we provide example responses for a set of design goals and modeling error. In the next section, we present a much more comprehensive examination of controller robustness. Here, we intend to briefly and qualitatively demonstrate the shortcomings of controllers in the presence of plant modeling error.

Let's begin with pole placement, implemented over various conditions and illustrated in figures 6-10a through 6-10c. The output error associated with pole placement is extremely pronounced. Although the reduced order observer offers marginal insensitivity to plant modeling error, it is clear that the pole placement controller does not come very close to its design objectives, except in Fig. 6-10b. In Fig. 6-10b, the plant's frequency is underestimated and the plant's damping, overestimated. These estimation

errors counteract each other in terms of the plant's settling time. Since the reference model specifies moderately high damping, this control approach does not appear to be entirely poor. However, the apparent success is due merely to circumstance (e.g., if reference model damping were lower, the error would be much more pronounced).

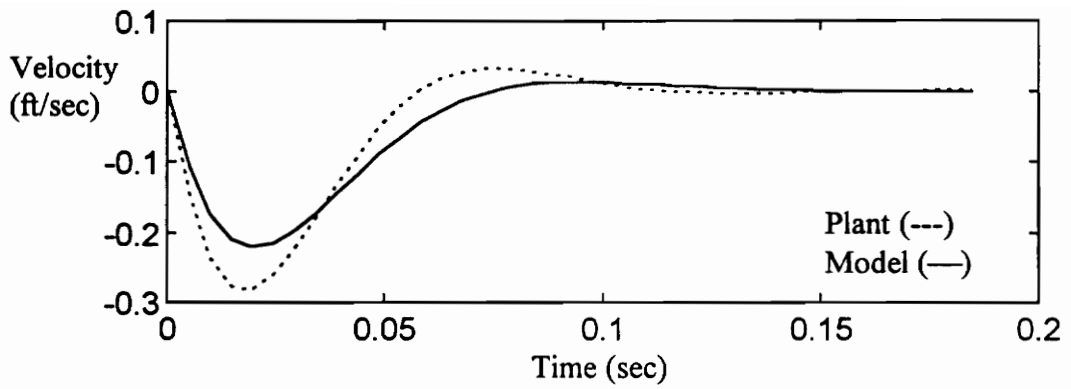
Note that we do not display any pole placement cases resulting in instability. This



$$PI = 0.0296 \text{ ft / s}$$

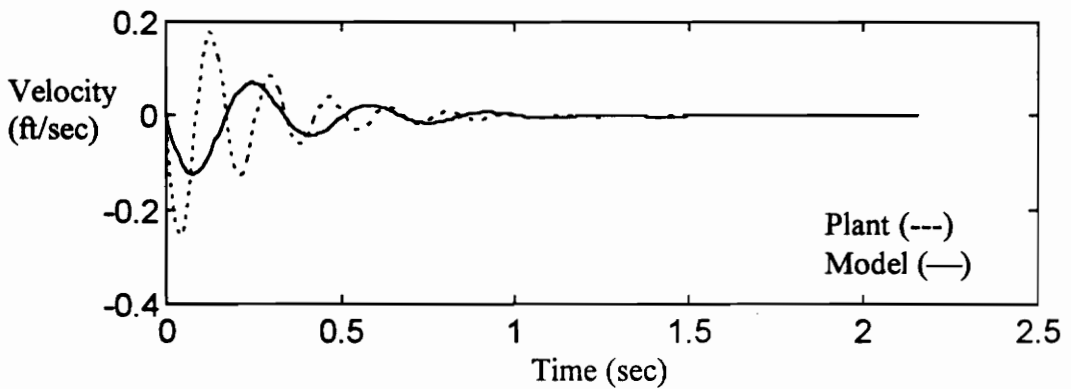
Figure 6-10a
Discrete Pole Placement Control for $\omega_m = 1.17 \omega_{pm}$ and $\zeta_m = 0.17$ with
 $\omega_p = 0.7 \omega_{pm}$ and $\zeta_p = 1.3 \zeta_{pm}$

is not to say that the closed-loop system never becomes unstable. In the next section, we demonstrate several instances of pole placement instability. In this section, we wish to illustrate that pole placement lacks a sufficient means to reduce the effects of plant modeling error. The observer only acts to recreate the position state of the actual plant. If the ROO successfully estimates position this does not change the fact that the performance gains are incorrect.



$$PI = 0.0192 \text{ ft / s}$$

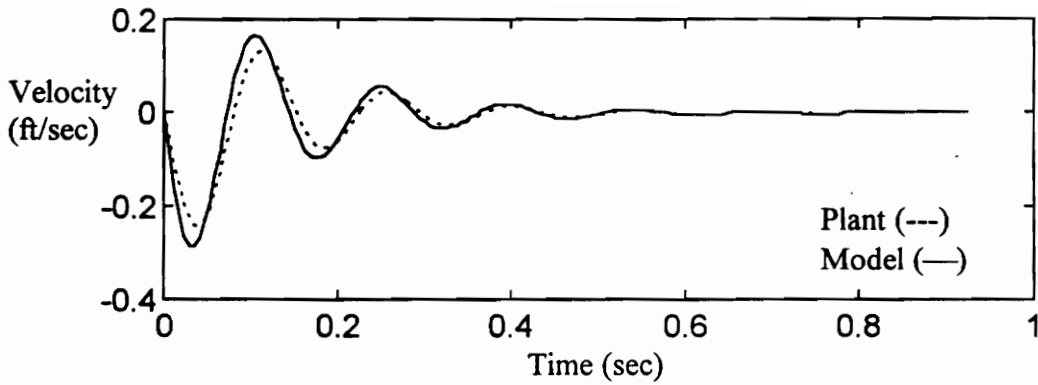
Figure 6-10b
Discrete Pole Placement Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with
 $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 0.7\zeta_{pm}$



$$PI = 0.0237 \text{ ft / s}$$

Figure 6-10c
Discrete Pole Placement Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with
 $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 1.3\zeta_{pm}$

Figures 6-11a through 6-11c illustrate the CGT controller subjected to the same set of design goals and plant modeling error.



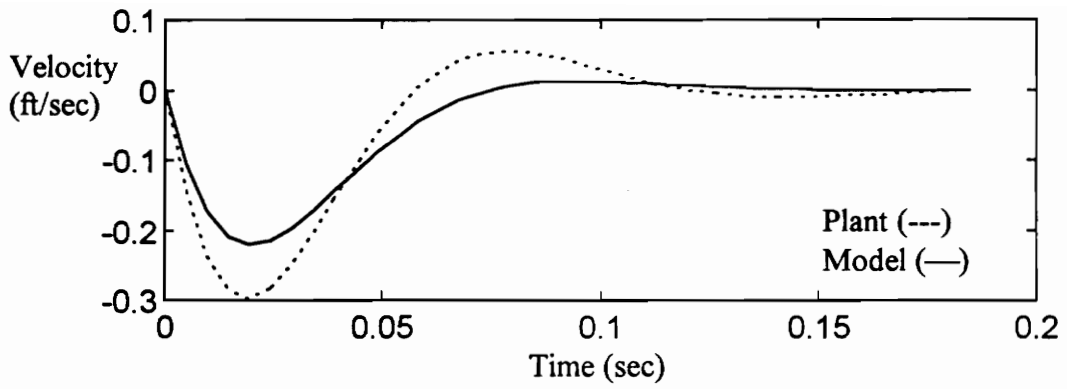
$$PI = 0.01002 \text{ ft / s}$$

Figure 6-11a

Discrete CGT Control for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 0.70\omega_{pm}$ and $\zeta_p = 1.30\zeta_{pm}$

This linear control approach demonstrates superior performance to pole placement. There are two major differences between these two methods. The CGT performance gains contribute to the control signal in an open-loop fashion. As discussed in Chapter 3, poor plant modeling can not result in instability with respect to the performance gains. The only source of instability is the output feedback gain K_e . Our design procedure will not allow K_e to create an unstable closed-loop system.

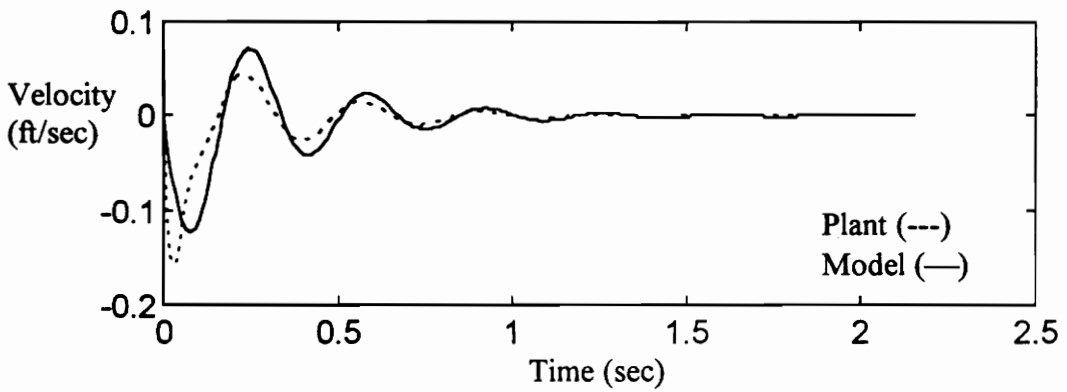
The output feedback gain asymptotically drives the error between the model and the plant to zero. It's effects are limited, however, by its conservative design.



$$PI = 0.004914 \text{ ft / s}$$

Figure 6-11b

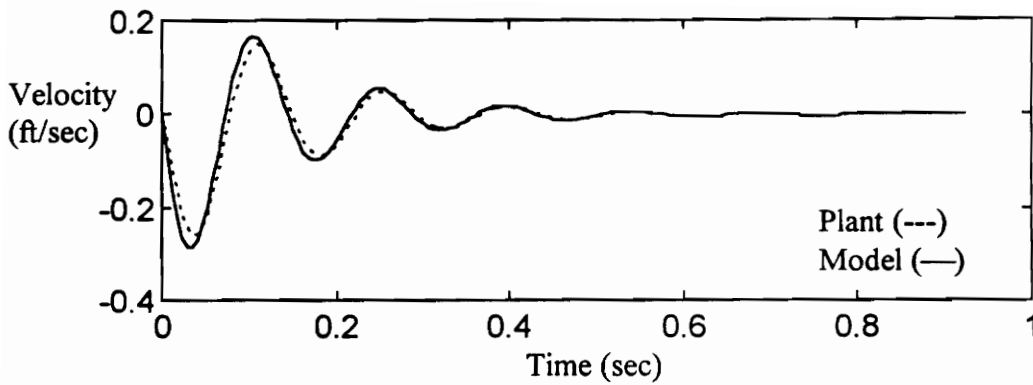
Discrete CGT Control for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 0.7\zeta_{pm}$



$$PI = 0.01623 \text{ ft / s}$$

Figure 6-11c

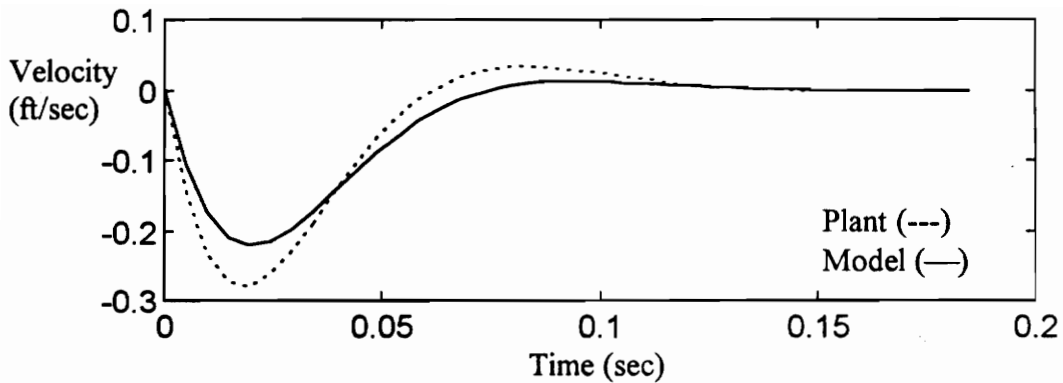
Discrete CGT Control for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 1.3\zeta_{pm}$



$$PI = 0.007764 \text{ ft / s}$$

Figure 6-12a
Discrete MRAC Control for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 0.70\omega_{pm}$
and $\zeta_p = 1.30\zeta_{pm}$

Next, we examine some MRAC response plots. Beginning with the fully adaptive MRAC, we observe that this control actuates better plant performance than the linear algorithms, except for the case having the lowest settling time. The response of Fig. 6-12b yields a higher (worse) performance index than the CGT controller. It is likely that the adaptive controller did not have sufficient information to integrate to error minimizing gains during the early transients. If we examine Fig. 6-11b and Fig. 6-12 b closely, we see that MRAC forces the plant response to converge to the model's output at approximately 0.13 seconds. The CGT control accomplishes this at about 0.175 seconds. Quantitatively, the CGT controller is better. Qualitatively, we must not dismiss MRAC as being worse in this application. As we begin our comprehensive analysis, we must be wary that the CGT controller has a decisive (and delusive) advantage for "quick" responses.



$$PI = 0.01701 \text{ ft / s}$$

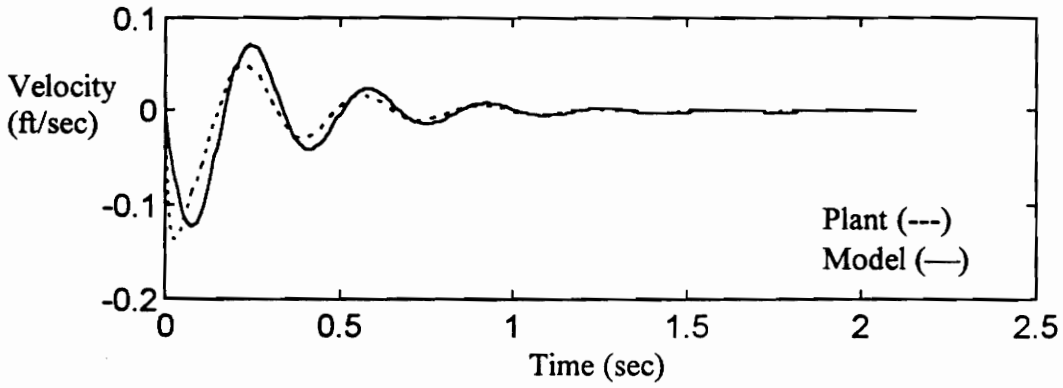
Figure 6-12b

Discrete MRAC for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 0.7\zeta_{pm}$

Let's compare the two types of MRAC controllers. We examined them previously for the case of perfect plant modeling. Now, we check to see if the trends observed in section 6.3 apply to this more realistic case.

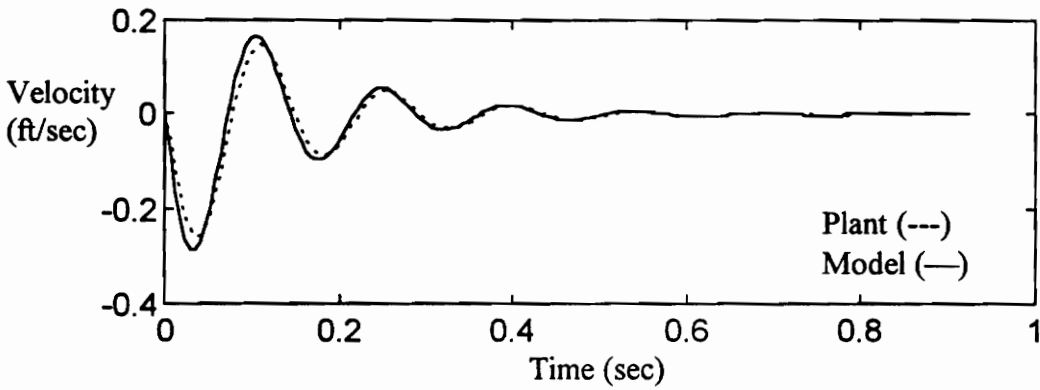
Across the board, the fully adaptive MRAC surpasses the fixed gain MRAC in performance. Recall that earlier, we discussed a problem associated with the reference model frequency being lower than the plant's. This caused the plant response to lead that of the model. The adaptive feedback gain reached negative values to "slow" the plant. Because of the integral gain, the adaptive feedback gain was sluggish in leaving the negative operating region. This effect allowed the fixed gain MRAC to perform better than the fully adaptive version. We hypothesized that, given enough time, the fully

adaptive MRAC would exceed the performance of its counterpart. Comparing Fig. 6-12 c to Fig. 6-13c, we see that this hypothesis can be true.



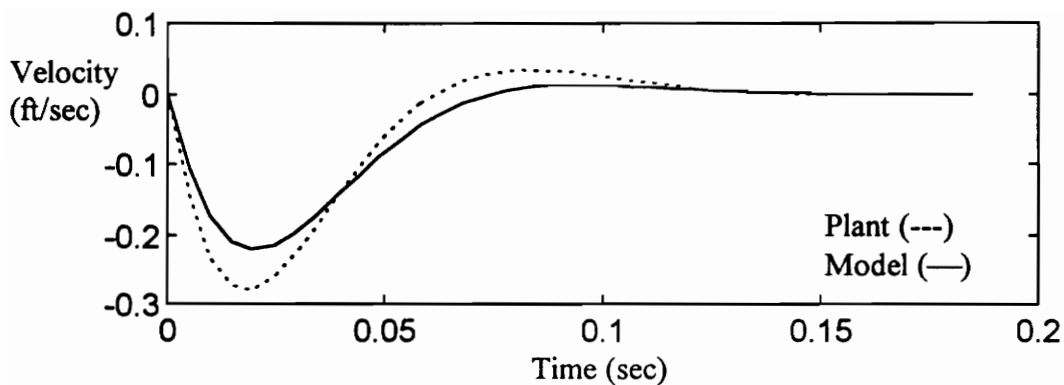
$$PI = 0.006669 \text{ ft / s}$$

Figure 6-12c
Discrete MRAC for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 1.3\zeta_{pm}$



$$PI = 0.008013 \text{ ft / s}$$

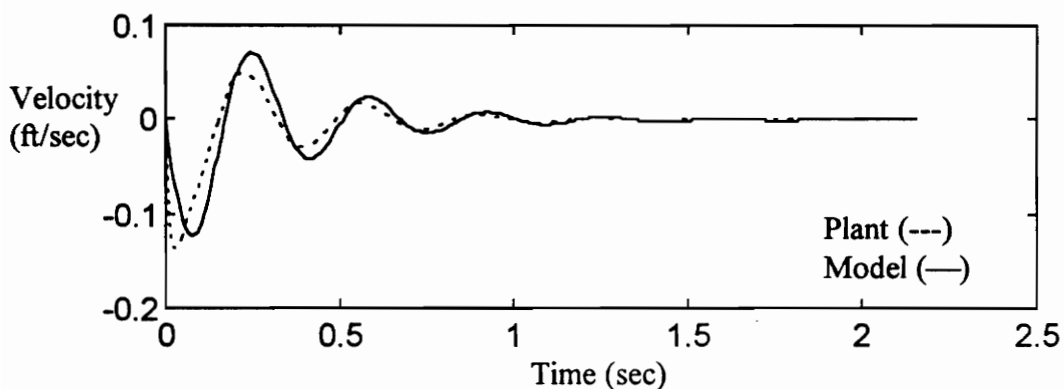
Figure 6-13a
Discrete MRAC Control Using a Fixed Gain for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 0.70\omega_{pm}$ and $\zeta_p = 1.30\zeta_{pm}$



$$PI = 0.01717 \text{ ft / s}$$

Figure 6-13b

Discrete MRAC Control Using a Fixed Gain for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 0.7\zeta_{pm}$



$$PI = 0.006712 \text{ ft / s}$$

Figure 6-13c

Discrete MRAC Control Using a Fixed Gain for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 1.3\zeta_{pm}$

6.5 Minimizing the Effects of Plant Modeling Error in Model Reference Adaptive Control

In Chapter 4, we conjectured that selecting the lowest possible feedthrough term, G , would minimize the dependence on erroneous CGT gains. In Chapter 5, we proposed a method of selecting the minimal hypothetical feedthrough term J for an assumed form of the matrix L . In our design procedure, we equate the feedthrough terms, G and J . Let's check the validity of our proposal by comparing the performance index of responses using different feedthrough terms than those presented in section 6.3.

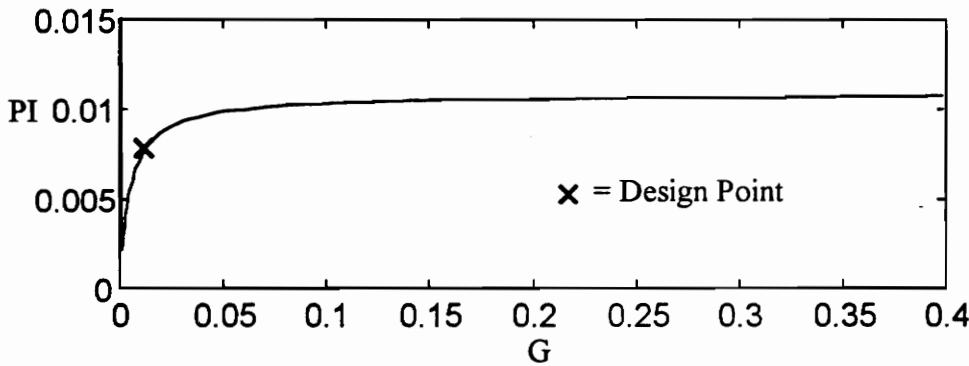


Figure 6-14a

G vs. PI for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.17$ with $\omega_p = 0.70\omega_{pm}$ and $\zeta_p = 1.30\zeta_{pm}$

We generated responses and performance indexes that correspond to a range of values for G . Note that increasing G does not affect stability. However, selecting G below a certain threshold, as we can see from Fig. 6-14a and Fig. 6-14b, the system

becomes unstable. This rapid system divergence occurs immediately before (in terms of G) we achieve a minimum performance index. Increasing G to large magnitudes does not appear to affect any significant change on performance, although performance does gradually become worse. Our design places G in a fairly safe region, balancing stability

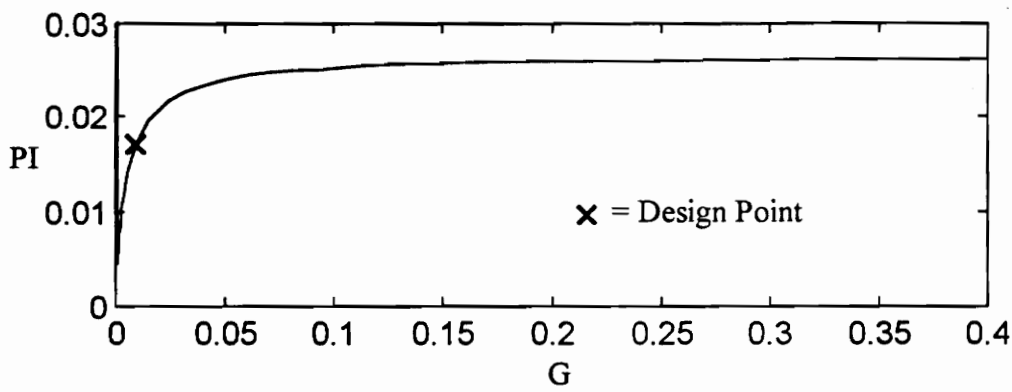


Figure 6-14a

G vs. PI for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.67$ with $\omega_p = 1.3\omega_{pm}$ and $\zeta_p = 0.7\zeta_{pm}$

with performance.

There are two notable problems associated with our attempt to minimize the effects of plant modeling error. The first is our assumed form of L from Eq. (5.4-13). We minimized for a given L , but did not explore how to choose an appropriate matrix L . The second problem essentially nullifies our first concern. Lyapunov stability results dictate the sufficient conditions for stability, not the necessary conditions. Without any well defined boundaries to work with, it is difficult to attempt minimization or optimization in any absolute sense. Whereas our solution of G is not optimal, it does yield a notable

insensitivity to plant modeling error. We conclude that our approach, though not analytically justified, yields robust simulation results.

6.6 Analyzing Controller Robustness

In this section, we generate responses for all four algorithms for various design goals. In each case, we determine the performance indexes over 49 points of possible plant dynamic characteristics. For each design goal we provide a mesh plot relating the performance index to modeling error and a table providing the same information numerically.

In Fig. 6-15a through Fig. 6-15d, we simulate the controllers for the maximum reference model frequency and critical damping. We predict the CGT controller to surpass the performance of the two adaptive algorithms. We base this hypothesis on the short settling time which is not conducive to complete adaptation.

Our results indicate that the CGT control is far superior, while pole placement, MRAC, and the fixed gain MRAC all perform at about the same level. Increasing the size of the weighting matrices and the sample rate could improve performance. However, our design does demonstrate reasonable effectiveness for this very demanding case.

In our next group of figures and tables, we examine performance for the maximum model frequency at a low damping ratio of 0.17. These specifications result in a longer transient period, allowing the MRAC systems to adapt and reject the effects of plant modeling error. The CGT controller should yield better results than pole placement, but should result in a PI lower than the MRAC systems. Because the reference model initially

leads the plant, the adaptive feedback gain should promote better error transients than the fixed gain MRAC if the response time is sufficient for accurate integration.

Examining Fig. 6-1a through Fig. 6-1d, we find our predictions to be mostly correct. Let's consider figures 6-1c and 6-1d: the two MRAC systems. Our predicted trends are true except for the row of values corresponding to maximum positive frequency modeling error in Table 6-2c and 6-2d. We observe that as the plant's natural frequency approaches that of the reference model, the effects of adaptive feedback become less pronounced. In this circumstance, there is less lag between the plant and reference model. This implies less error to drive the adaptive feedback gain. Because the transient response is relatively short, the adaptive gain has insufficient information to accurately integrate to a performance level surpassing the fixed feedback gain.

Our next set of design objectives consist of a minimal reference model frequency and critical damping. Because of the extended transient period, relative to that of Figs. 6-15, we expect differences between the MRAC systems and the CGT control to be less severe. However, the CGT controller probably yields a better performance index.

Examining Figs. 6-17 and Tables 6-3, we can conclude that our generalizations are accurate. There are some trends, however, that we did not foresee.

In Fig 6-17a, the pole placement controller is unstable for certain values of plant modeling error. Since the reference model requires a reduction in the plant's frequency, the feedback gains are calculated to move the poles away from the origin of the z-plane. By overestimating the plant's actual open-loop pole locations, this controller

overcompensates for the design objective. The feedback gains are excessive, placing poles in the unstable region of the z -plane.

As expected, the CGT controller performs best of the four algorithms. Since the MRAC systems have more time to adapt, the difference are not as drastic as the critical damping case.

Examining Table 6-3c and Table 6-3d, we see that the two MRAC systems bear much similarity. We observe a new trend, however. Recall that the adaptive feedback gain grows negatively, when the reference model lags behind the plant's response. The amount of lag is minimal for the lowest possible plant frequency. Although the integral feedback gain grows negatively, the longer time of response allows the error gain to continue adapting and gradually reach an optimal value.

For plants having greater natural frequencies, we see that the fixed gain MRAC performs better than the fully adaptive version. In these tables, we witness a compromise between adaptation time and frequency difference between the actual plant and the reference model.⁶

Next, we specify the reference model to have the minimum possible frequency and a low damping ratio of 0.17. This yields a relatively long transient response for which we expect the adaptive controllers to exceed the performance of the linear controllers.

Let's first consider pole placement, illustrated by Fig. 6-18a. Like the results from Fig. 6-17a, where the reference model's frequency is minimal, we see that this controller

⁶ Damping is also a factor but its effects (regarding model error) are negligible compared to frequency.

creates unstable closed-loop dynamics. Again, this is due to overestimating the plant's natural frequency, thus the feedback gains overcompensate

The adaptive algorithms, illustrated by Fig. 6-18c and Fig. 6-18d, are indeed superior to the linear controllers. Let's look at the difference between these two adaptive approaches. Recall that the fixed gain MRAC tends to perform better for minimum plant frequency and maximum plant damping. It takes the fully adaptive algorithm some time to converge to the optimal feedback value. This is why the fixed gain version produces marginally better performance in this row of Table 6-4d.

Our next design objectives entail a modest increase in the plant's frequency and a damping ratio of 0.33. We expect the CGT controller and the MRAC systems to have similar performance, but cannot predict which is best for this moderate settling time. We do, however, assume that the pole placement controller performs most poorly.

The CGT controller yields the lowest performance index. Its asymptotic tracking ability exceeds that of the adaptive performance gains. Apparently, this transient response is not lengthy enough to sufficiently expose the shortcomings of the CGT control method. Recall from section 6.4, that although the CGT controller may have a lower PI, the MRAC system tends to converge to the model output sooner during the transient response. It is in the initial stage of transients, where MRAC begins to "learn", that the CGT control demonstrates a margin of superiority.

This particular response calls for a model frequency greater than the plant's (the model response leads the plant's), and, the transients are long enough such that the fully

adaptive MRAC exceeds the performance of the fixed gain MRAC for every possible testing condition.

Finally, we present our best evidence for adaptive control. In Figs. 6-20 and Tables 6-6, we define the reference model to have a maximum natural frequency and a very low damping ratio of 0.01 (equal to the nominal plant). This produces a response time much longer than any of our previous cases.

In Fig. 6-20a, we have another example of pole placement instability. This time, the light damping ratio requires closed-loop poles located close to the edge of the unit circle (in the z -plane). When the plant's natural frequency is largely underestimated, the feedback gains place poles at unstable locations.

Regarding the CGT controller, the length of response exposes the fact that model output tracking does not occur until the model output equals zero at steady-state. We provide this information in Fig. 6-20b and Table 6-6b.

The MRAC algorithms' performance clearly exceeds that of the linear controllers. The results between the two types of MRAC are equitable. There are some slight trends evident in Tables 6-6c and 6-6d. We would expect, from trends thus far, that the fully adaptive MRAC produce the lowest performance index. However, this appears to be true only for cases involving overestimations of the actual plant's frequency. Although, this appears to be a new phenomenon, it is minor and underscored by the vast superiority of MRAC over linear control.

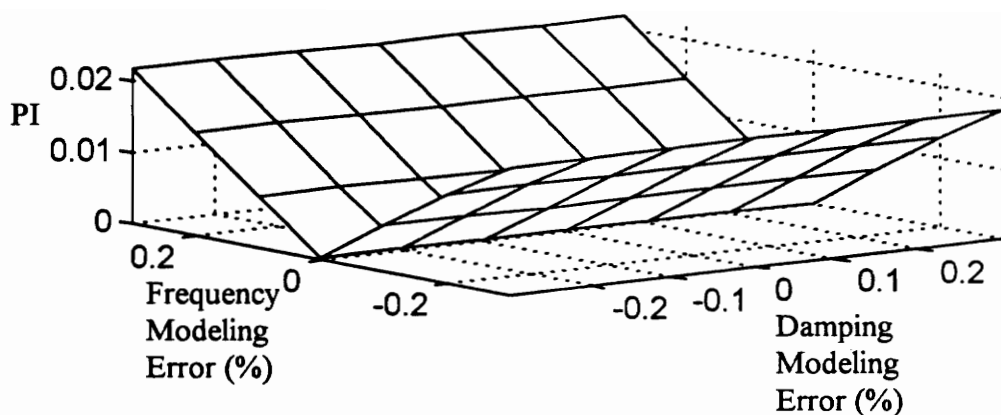


Figure 6-15a
Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-1a
Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	1.7967	1.7989	1.8012	1.8034	1.8057	1.8079	1.8102
$-0.2\omega_{pm}$	1.2550	1.2578	1.2605	1.2633	1.2661	1.2688	1.2716
$-0.1\omega_{pm}$	0.6422	0.6458	0.6494	0.6530	0.6567	0.6603	0.6639
No Error	0.0318	0.0279	0.0248	0.0236	0.0230	0.0223	0.0217
$+0.1\omega_{pm}$	0.7284	0.7231	0.7179	0.7127	0.7075	0.7023	0.6971
$+0.2\omega_{pm}$	1.4579	1.4517	1.4455	1.4393	1.4331	1.4269	1.4208
$+0.3\omega_{pm}$	2.1861	2.1793	2.1726	2.1659	2.1591	2.1524	2.1457

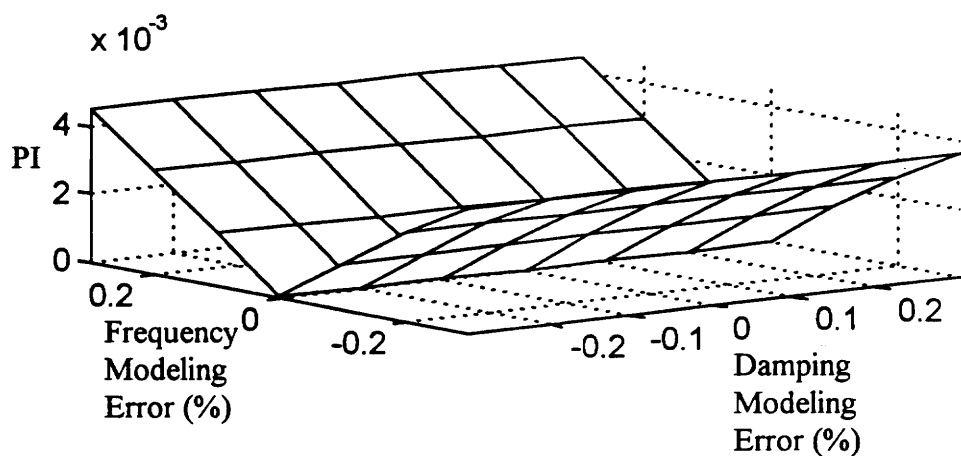


Figure 6-15b
CGT Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-1b
CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	3.6384	3.6417	3.6449	3.6481	3.6513	3.6545	3.6577
$-0.2\omega_{pm}$	2.5957	2.6005	2.6053	2.6101	2.6148	2.6196	2.6243
$-0.1\omega_{pm}$	1.3586	1.3655	1.3723	1.3790	1.3858	1.3926	1.3993
No Error	0.0537	0.0446	0.0364	0.0304	0.0285	0.0269	0.0253
$+0.1\omega_{pm}$	1.5468	1.5345	1.5223	1.5101	1.4979	1.4858	1.4736
$+0.2\omega_{pm}$	3.0683	3.0537	3.0392	3.0247	3.0102	2.9957	2.9813
$+0.3\omega_{pm}$	4.5402	4.5230	4.5058	4.4887	4.4716	4.4546	4.4376

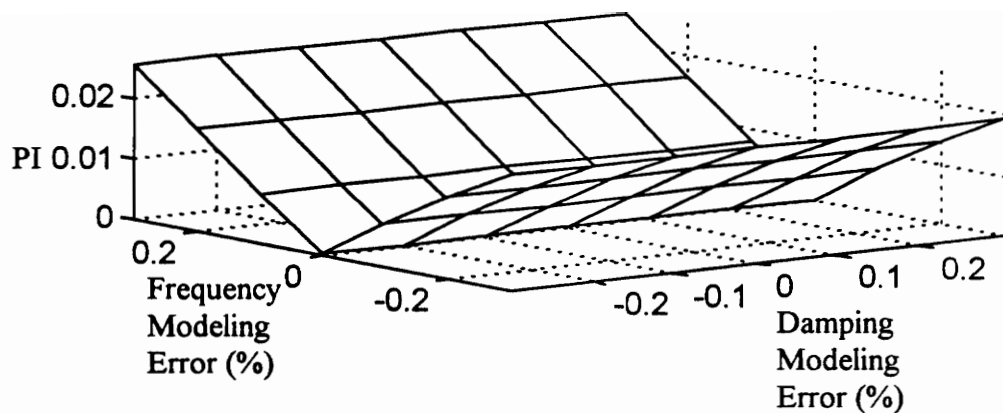


Figure 6-15c
MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-1c
MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	1.9183	1.9207	1.9230	1.9254	1.9277	1.9301	1.9324
$-0.2\omega_{pm}$	1.3644	1.3674	1.3705	1.3736	1.3767	1.3797	1.3828
$-0.1\omega_{pm}$	0.7174	0.7213	0.7253	0.7293	0.7332	0.7371	0.7411
No Error	0.0314	0.0296	0.0277	0.0258	0.0239	0.0220	0.0202
$+0.1\omega_{pm}$	0.8317	0.8253	0.8189	0.8126	0.8062	0.7999	0.7936
$+0.2\omega_{pm}$	1.6932	1.6856	1.6781	1.6706	1.6631	1.6556	1.6481
$+0.3\omega_{pm}$	2.5680	2.5591	2.5501	2.5412	2.5323	2.5234	2.5146

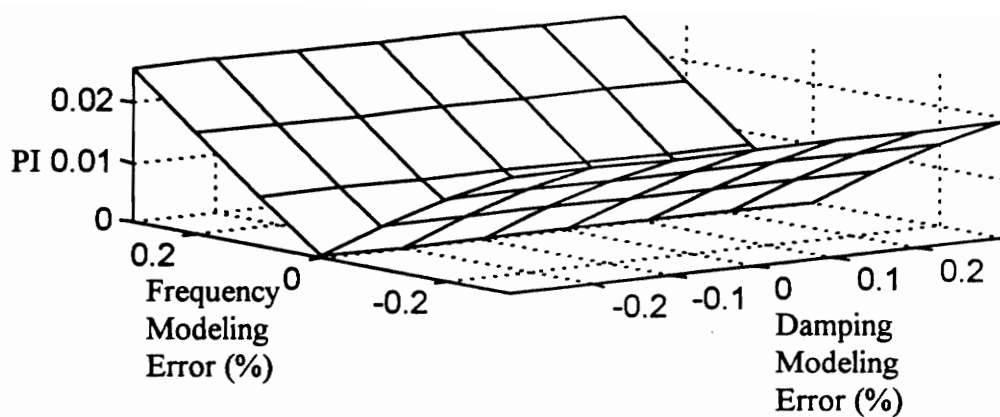


Figure 6-15d

Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-1d

Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	1.9304	1.9328	1.9351	1.9375	1.9398	1.9422	1.9445
$-0.2\omega_{pm}$	1.3746	1.3777	1.3808	1.3838	1.3869	1.3900	1.3930
$-0.1\omega_{pm}$	0.7244	0.7284	0.7324	0.7364	0.7403	0.7443	0.7482
No Error	0.0315	0.0299	0.0282	0.0266	0.0249	0.0233	0.0216
$+0.1\omega_{pm}$	0.8323	0.8259	0.8195	0.8131	0.8067	0.8003	0.7939
$+0.2\omega_{pm}$	1.6971	1.6895	1.6819	1.6743	1.6668	1.6593	1.6518
$+0.3\omega_{pm}$	2.5742	2.5652	2.5562	2.5472	2.5383	2.5294	2.5204

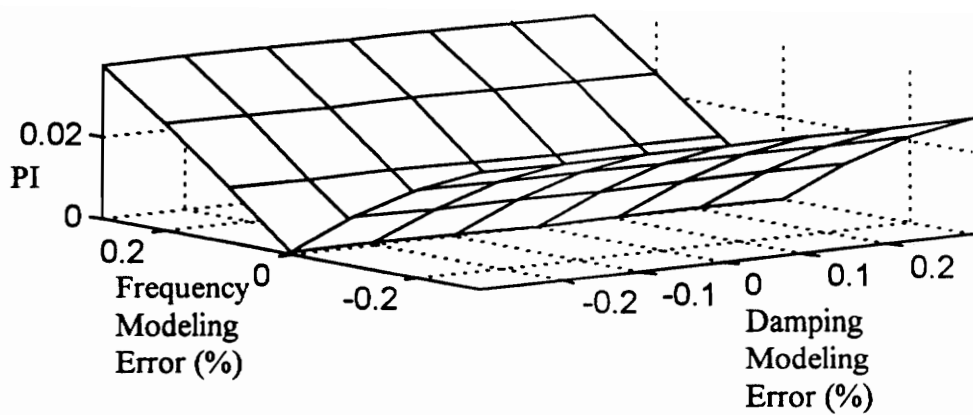


Figure 6-16a
Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-2a
Pole Placment Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	2.8180	2.8142	2.8104	2.8068	2.8033	2.7999	2.7965
$-0.2\omega_{pm}$	2.1135	2.1102	2.1070	2.1038	2.1006	2.0977	2.0955
$-0.1\omega_{pm}$	1.1596	1.1598	1.1602	1.1607	1.1612	1.1618	1.1629
No Error	0.0693	0.0511	0.0332	0.0154	0.0129	0.0264	0.0409
$+0.1\omega_{pm}$	1.3262	1.3147	1.3033	1.2922	1.2819	1.2716	1.2617
$+0.2\omega_{pm}$	2.6002	2.5817	2.5640	2.5469	2.5303	2.5138	2.4979
$+0.3\omega_{pm}$	3.7070	3.6851	3.6636	3.6429	3.6224	3.6021	3.5822

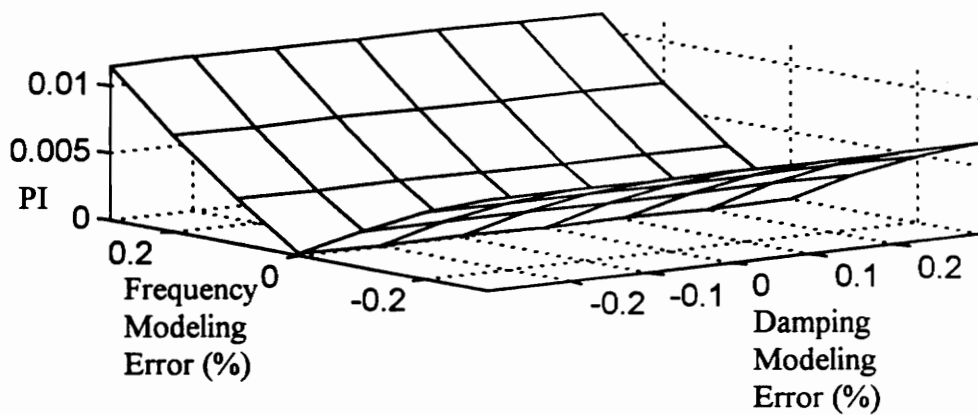


Figure 6-16b
CGT Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-2b
CGT Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	0.6728	0.6730	0.6733	0.6735	0.6738	0.6740	0.6743
$-0.2\omega_{pm}$	0.4922	0.4928	0.4934	0.4940	0.4945	0.4951	0.4957
$-0.1\omega_{pm}$	0.2683	0.2690	0.2698	0.2706	0.2715	0.2726	0.2738
No Error	0.0152	0.0108	0.0065	0.0032	0.0053	0.0080	0.0120
$+0.1\omega_{pm}$	0.3323	0.3302	0.3281	0.3260	0.3240	0.3220	0.3201
$+0.2\omega_{pm}$	0.7156	0.7129	0.7102	0.7075	0.7049	0.7022	0.6996
$+0.3\omega_{pm}$	1.1501	1.1459	1.1416	1.1374	1.1332	1.1289	1.1247

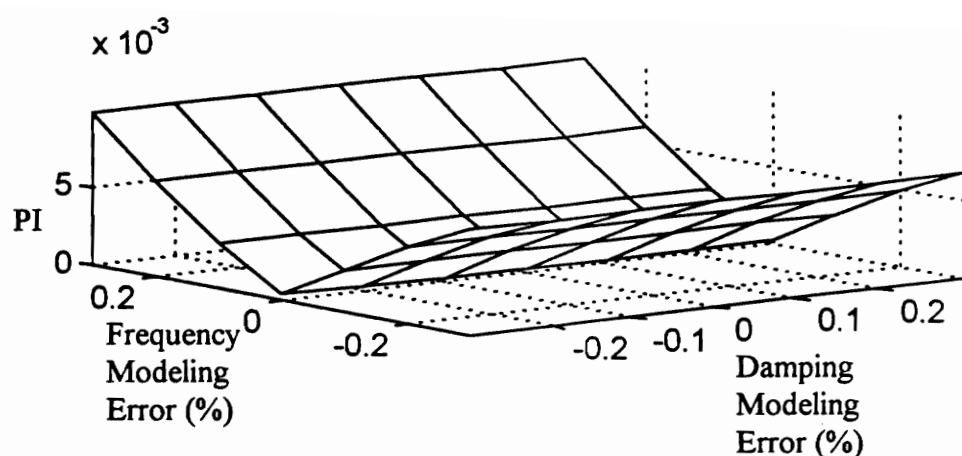


Figure 6-16c
MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-2c
MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	6.7374	6.7398	6.7430	6.7461	6.7492	6.7524	6.7555
$-0.2\omega_{pm}$	4.8719	4.8766	4.8813	4.8864	4.8915	4.8967	4.9018
$-0.1\omega_{pm}$	2.6652	2.6741	2.6830	2.6919	2.7008	2.7097	2.7185
No Error	0.3451	0.3145	0.2856	0.2657	0.2583	0.2633	0.2864
$+0.1\omega_{pm}$	2.9330	2.9168	2.9006	2.8846	2.8687	2.8529	2.8386
$+0.2\omega_{pm}$	6.1782	6.1579	6.1377	6.1177	6.0977	6.0779	6.0582
$+0.3\omega_{pm}$	9.8249	9.7970	9.7693	9.7451	9.7225	9.7000	9.6774

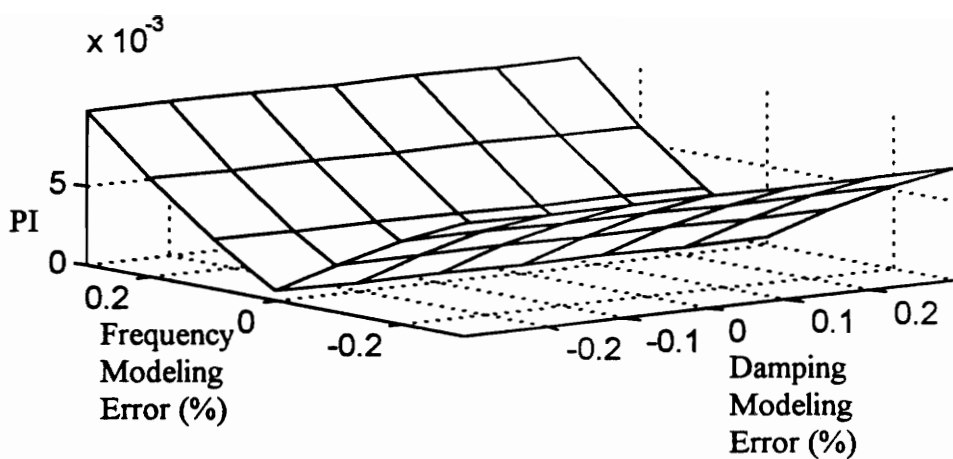


Figure 6-16d
Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-2d
Fixed Gain MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	7.0986	7.0995	7.1005	7.1014	7.1024	7.1037	7.1054
$-0.2\omega_{pm}$	5.2371	5.2404	5.2436	5.2469	5.2502	5.2535	5.2568
$-0.1\omega_{pm}$	3.0284	3.0347	3.0411	3.0474	3.0538	3.0601	3.0665
No Error	0.6507	0.6147	0.5805	0.5551	0.5423	0.5417	0.5591
$+0.1\omega_{pm}$	3.0861	3.0656	3.0456	3.0255	3.0055	2.9855	2.9658
$+0.2\omega_{pm}$	6.2309	6.2081	6.1854	6.1627	6.1401	6.1175	6.0950
$+0.3\omega_{pm}$	9.7935	9.7632	9.7329	9.7039	9.6783	9.6528	9.6274

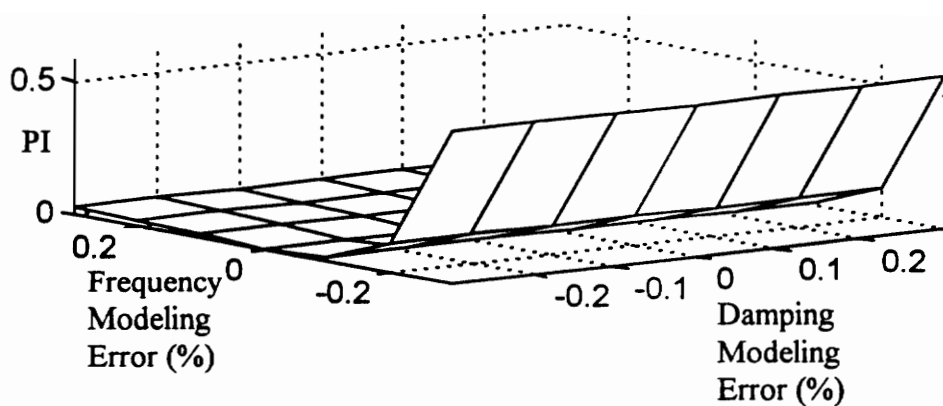


Figure 6-17a
Pole Placement Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-3a
Pole Placement Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	5.8160	5.7970	5.7781	5.7593	5.7405	5.7219	5.7034
$-0.2\omega_{pm}$	1.1234	1.1209	1.1184	1.1160	1.1135	1.1111	1.1087
$-0.1\omega_{pm}$	0.1790	0.1790	0.1790	0.1790	0.1790	0.1790	0.1791
No Error	0.0009	0.0008	0.0008	0.0008	0.0010	0.0014	0.0018
$+0.1\omega_{pm}$	0.1471	0.1465	0.1460	0.1455	0.1450	0.1445	0.1439
$+0.2\omega_{pm}$	0.2425	0.2420	0.2414	0.2408	0.2403	0.2397	0.2391
$+0.3\omega_{pm}$	0.3136	0.3130	0.3124	0.3118	0.3111	0.3105	0.3099

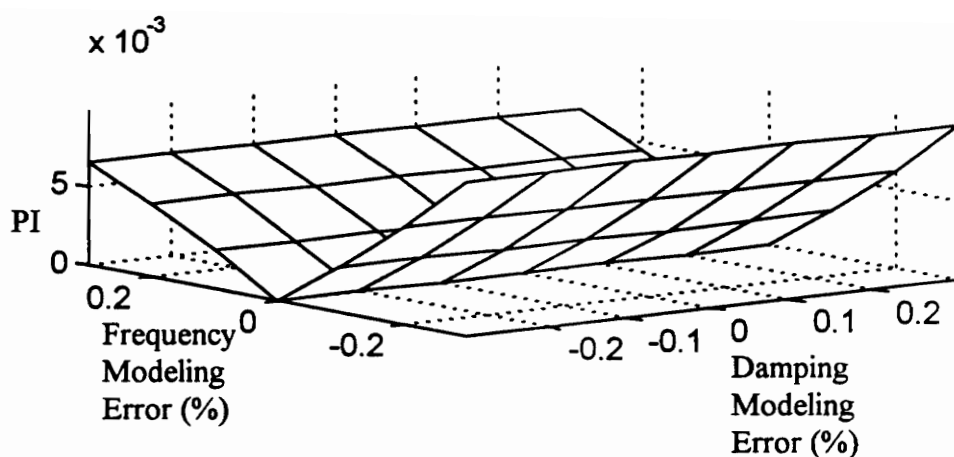


Figure 6-17b
CGT Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-3b
CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	9.8151	9.8134	9.8116	9.8098	9.8080	9.8062	9.8045
$-0.2\omega_{pm}$	6.1277	6.1284	6.1291	6.1298	6.1305	6.1315	6.1327
$-0.1\omega_{pm}$	2.8639	2.8669	2.8699	2.8729	2.8759	2.8789	2.8819
No Error	0.0302	0.0303	0.0308	0.0334	0.0390	0.0448	0.0505
$+0.1\omega_{pm}$	2.4700	2.4633	2.4566	2.4500	2.4433	2.4367	2.4300
$+0.2\omega_{pm}$	4.6415	4.6332	4.6249	4.6166	4.6083	4.6000	4.5918
$+0.3\omega_{pm}$	6.5304	6.5205	6.5106	6.5008	6.4910	6.4811	6.4713

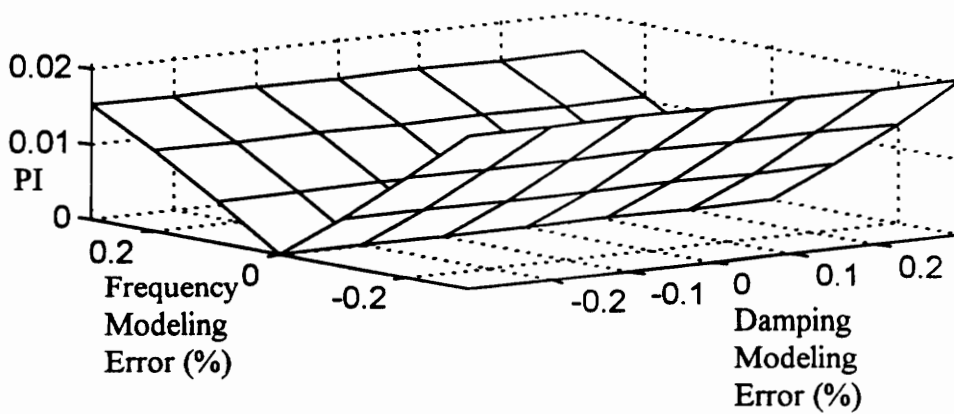


Figure 6-17c
MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-3c
MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	2.0544	2.0543	2.0542	2.0542	2.0541	2.0540	2.0539
$-0.2\omega_{pm}$	1.3272	1.3277	1.3282	1.3286	1.3291	1.3295	1.3300
$-0.1\omega_{pm}$	0.6342	0.6350	0.6358	0.6366	0.6374	0.6383	0.6391
No Error	0.0076	0.0079	0.0082	0.0088	0.0098	0.0110	0.0123
$+0.1\omega_{pm}$	0.5619	0.5604	0.5589	0.5575	0.5560	0.5545	0.5530
$+0.2\omega_{pm}$	1.0636	1.0618	1.0601	1.0583	1.0566	1.0548	1.0531
$+0.3\omega_{pm}$	1.5113	1.5092	1.5072	1.5052	1.5032	1.5012	1.4992

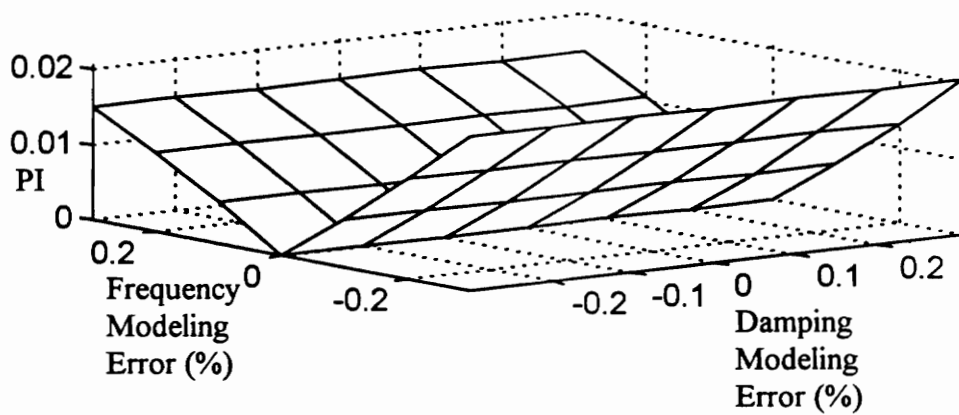


Figure 6-17d

Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Table 6-3d

Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 1.0$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	2.0618	2.0617	2.0616	2.0616	2.0615	2.0614	2.0613
$-0.2\omega_{pm}$	1.3308	1.3313	1.3317	1.3322	1.3327	1.3331	1.3336
$-0.1\omega_{pm}$	0.6358	0.6366	0.6374	0.6383	0.6391	0.6399	0.6407
No Error	0.0076	0.0078	0.0082	0.0088	0.0099	0.0111	0.0123
$+0.1\omega_{pm}$	0.5614	0.5599	0.5584	0.5570	0.5555	0.5540	0.5525
$+0.2\omega_{pm}$	1.0632	1.0615	1.0597	1.0580	1.0562	1.0545	1.0527
$+0.3\omega_{pm}$	1.5110	1.5090	1.5070	1.5049	1.5029	1.5009	1.4989

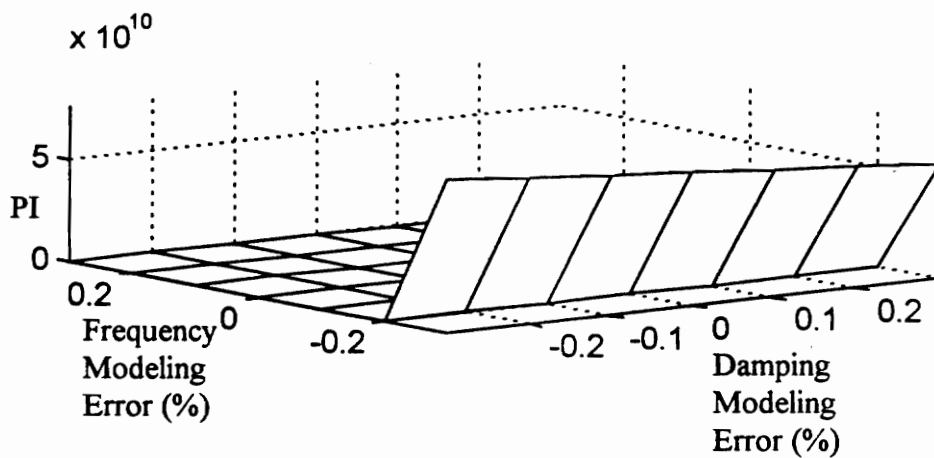


Figure 6-18a
Pole Placement Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-4a
Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	$7.6(10)^{12}$	$7.2(10)^{12}$	$6.9(10)^{12}$	$6.5(10)^{12}$	$6.3(10)^{12}$	$6.0(10)^{12}$	$5.7(10)^{12}$
$-0.2\omega_{pm}$	$2.0(10)^4$	$2.0(10)^4$	$1.9(10)^4$	$1.9(10)^4$	$1.8(10)^4$	$1.8(10)^4$	$1.8(10)^4$
$-0.1\omega_{pm}$	4.6402	4.5519	4.4669	4.3848	4.3057	4.2287	4.1544
No Error	0.0484	0.0304	0.0140	0.0053	0.0222	0.0388	0.0550
$+0.1\omega_{pm}$	1.8294	1.8153	1.8016	1.7884	1.7758	1.7634	1.7514
$+0.2\omega_{pm}$	2.2242	2.2062	2.1891	2.1723	2.1560	2.1400	2.1244
$+0.3\omega_{pm}$	2.4937	2.4703	2.4477	2.4261	2.4054	2.3853	2.3657

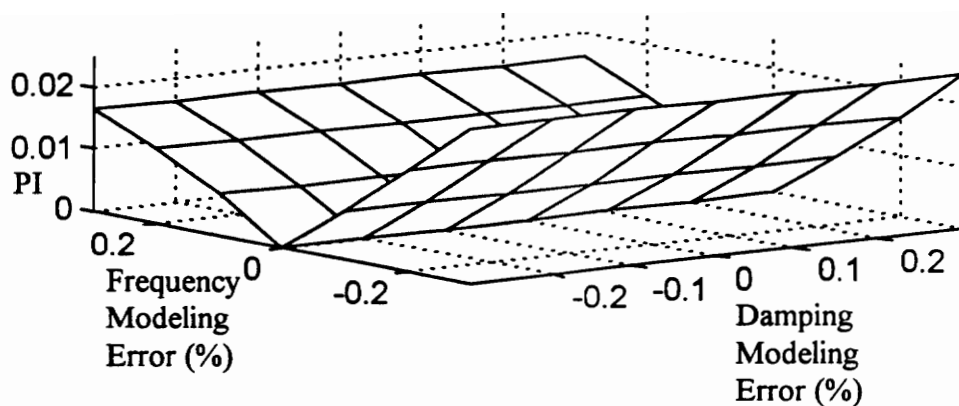


Figure 6-18b
CGT Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-4b
CGT Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	2.4809	2.4793	2.4777	2.4761	2.4745	2.4729	2.4713
$-0.2\omega_{pm}$	1.6140	1.6133	1.6126	1.6119	1.6112	1.6106	1.6099
$-0.1\omega_{pm}$	0.7641	0.7641	0.7642	0.7642	0.7642	0.7643	0.7643
No Error	0.0093	0.0070	0.0050	0.0033	0.0064	0.0095	0.0125
$+0.1\omega_{pm}$	0.6500	0.6488	0.6477	0.6466	0.6455	0.6445	0.6434
$+0.2\omega_{pm}$	1.1871	1.1858	1.1845	1.1832	1.1819	1.1806	1.1793
$+0.3\omega_{pm}$	1.6318	1.6303	1.6289	1.6274	1.6259	1.6245	1.6230

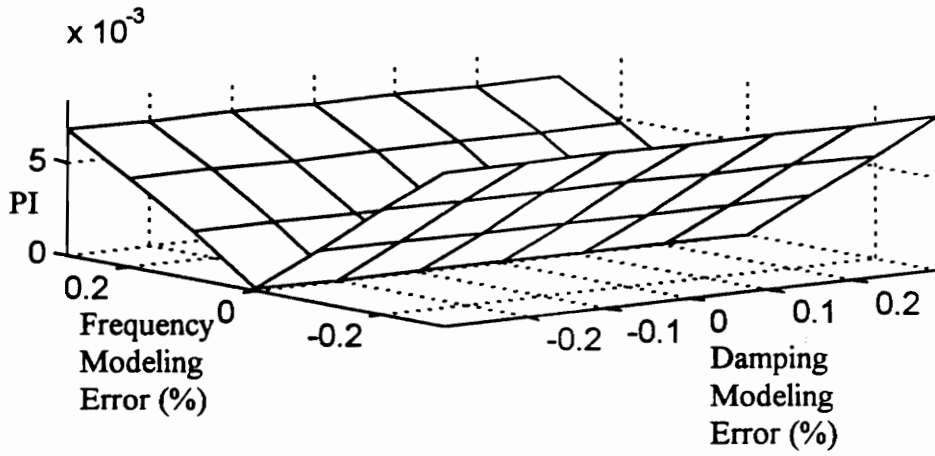


Figure 6-18c
MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-4c
MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	8.2633	8.2599	8.2565	8.2532	8.2498	8.2465	8.2431
$-0.2\omega_{pm}$	5.5649	5.5634	5.5621	5.5609	5.5597	5.5585	5.5572
$-0.1\omega_{pm}$	2.7369	2.7374	2.7383	2.7392	2.7405	2.7418	2.7433
No Error	0.0978	0.0912	0.0859	0.0812	0.0830	0.0898	0.0984
$+0.1\omega_{pm}$	2.5542	2.5499	2.5456	2.5414	2.5372	2.5330	2.5288
$+0.2\omega_{pm}$	4.7774	4.7719	4.7664	4.7608	4.7554	4.7499	4.7444
$+0.3\omega_{pm}$	6.7052	6.6990	6.6928	6.6866	6.6804	6.6744	6.6685

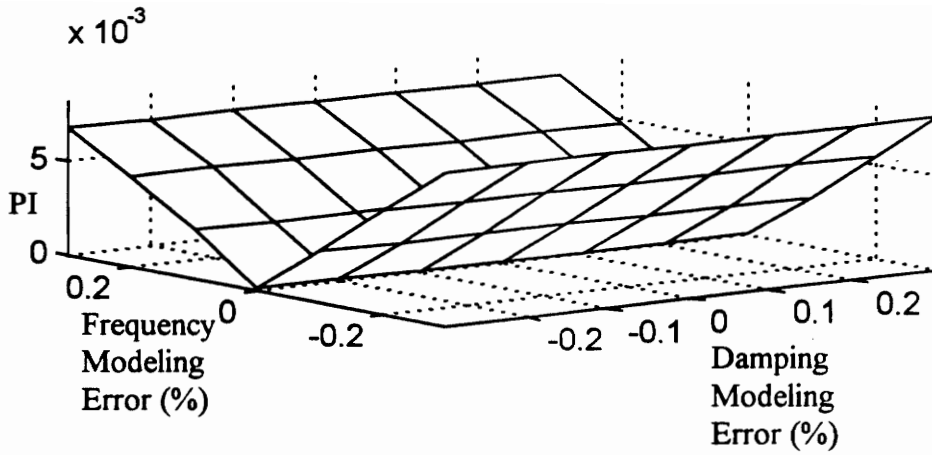


Figure 6-18d

Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Table 6-4d

Fixed Gain MRAC Performance Index (PI $\times 1000$) vs. Modeling Error for $\omega_m = 0.5\omega_{pm}$ and $\zeta_m = 0.17$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	8.1842	8.1807	8.1772	8.1738	8.1703	8.1669	8.1634
$-0.2\omega_{pm}$	5.4931	5.4914	5.4898	5.4883	5.4869	5.4854	5.4840
$-0.1\omega_{pm}$	2.6882	2.6883	2.6889	2.6894	2.6904	2.6914	2.6925
No Error	0.1389	0.1307	0.1239	0.1177	0.1180	0.1234	0.1305
$+0.1\omega_{pm}$	2.6049	2.6006	2.5963	2.5920	2.5878	2.5835	2.5793
$+0.2\omega_{pm}$	4.8263	4.8208	4.8153	4.8098	4.8043	4.7989	4.7935
$+0.3\omega_{pm}$	6.7484	6.7422	6.7361	6.7299	6.7238	6.7179	6.7120

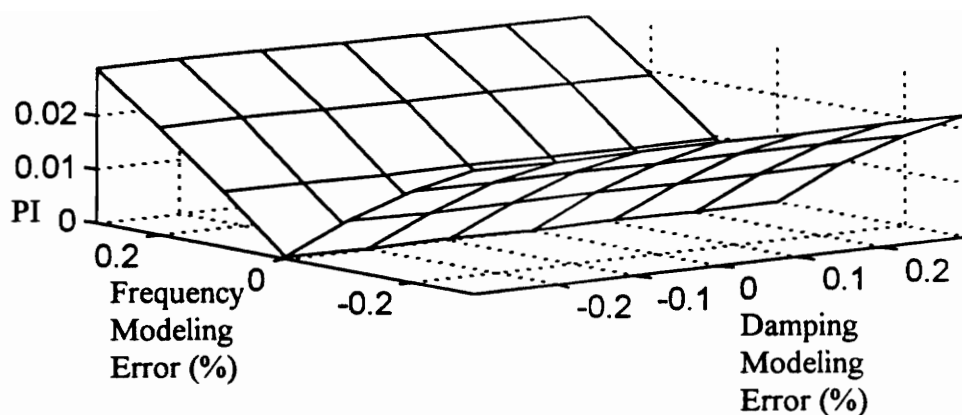


Figure 6-19a
Pole Placement Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$.

Table 6-5a
Pole Placement Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	2.2844	2.2839	2.2834	2.2829	2.2824	2.2819	2.2817
$-0.2\omega_{pm}$	1.6870	1.6871	1.6872	1.6873	1.6879	1.6886	1.6893
$-0.1\omega_{pm}$	0.9085	0.9106	0.9128	0.9156	0.9183	0.9211	0.9239
No Error	0.0341	0.0244	0.0146	0.0052	0.0071	0.0148	0.0239
$+0.1\omega_{pm}$	1.0175	1.0095	1.0015	0.9935	0.9856	0.9778	0.9700
$+0.2\omega_{pm}$	2.0074	1.9962	1.9850	1.9739	1.9629	1.9520	1.9412
$+0.3\omega_{pm}$	2.8975	2.8838	2.8702	2.8567	2.8433	2.8300	2.8168

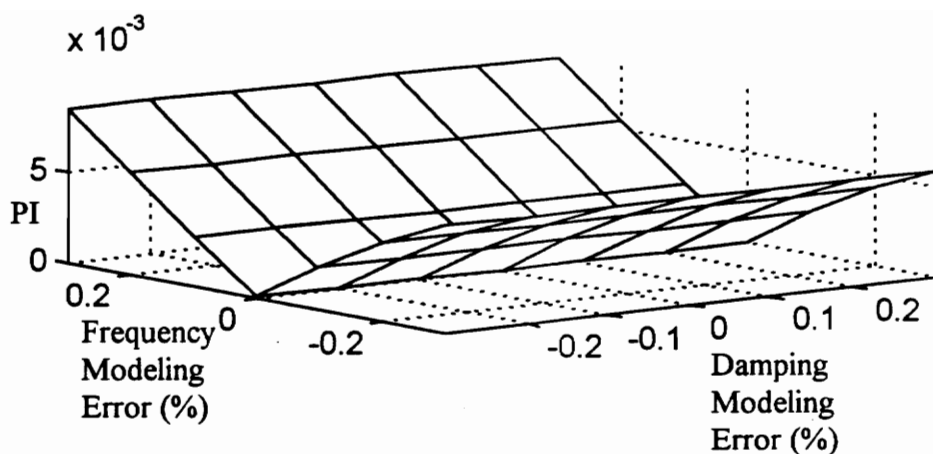


Figure 6-19b
CGT Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Table 6-5b
CGT Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	5.9019	5.9063	5.9106	5.9150	5.9194	5.9238	5.9282
$-0.2\omega_{pm}$	4.3134	4.3185	4.3237	4.3289	4.3340	4.3392	4.3443
$-0.1\omega_{pm}$	2.3232	2.3319	2.3406	2.3492	2.3579	2.3673	2.3777
No Error	0.0882	0.0624	0.0366	0.0134	0.0213	0.0432	0.0680
$+0.1\omega_{pm}$	2.7266	2.7074	2.6882	2.6691	2.6501	2.6311	2.6122
$+0.2\omega_{pm}$	5.6026	5.5767	5.5510	5.5253	5.4998	5.4744	5.4491
$+0.3\omega_{pm}$	8.4709	8.4391	8.4073	8.3758	8.3443	8.3130	8.2818

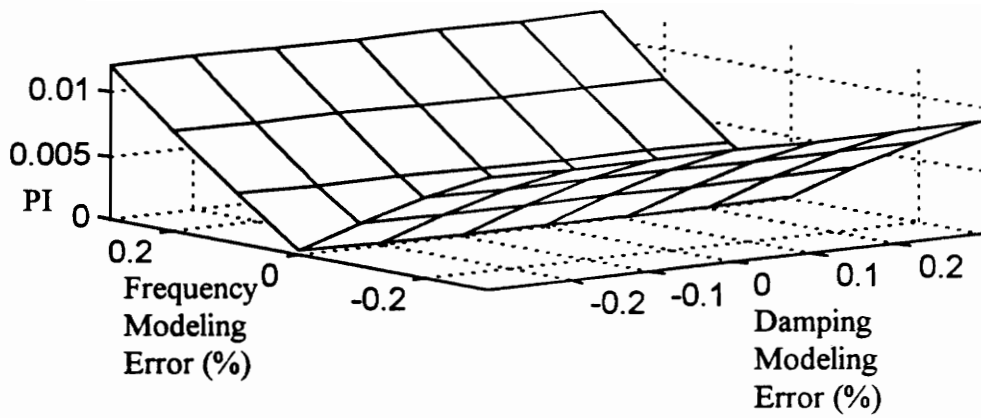


Figure 6-19c
MRAC Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Table 6-5c
MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	0.8768	0.8775	0.8782	0.8789	0.8796	0.8803	0.8810
$-0.2\omega_{pm}$	0.6264	0.6275	0.6287	0.6299	0.6310	0.6322	0.6333
$-0.1\omega_{pm}$	0.3357	0.3368	0.3380	0.3392	0.3404	0.3416	0.3428
No Error	0.0251	0.0221	0.0193	0.0168	0.0152	0.0163	0.0188
$+0.1\omega_{pm}$	0.3816	0.3791	0.3766	0.3741	0.3716	0.3691	0.3666
$+0.2\omega_{pm}$	0.7849	0.7818	0.7787	0.7756	0.7725	0.7694	0.7663
$+0.3\omega_{pm}$	1.2116	1.2078	1.2040	1.2003	1.1965	1.1928	1.1891

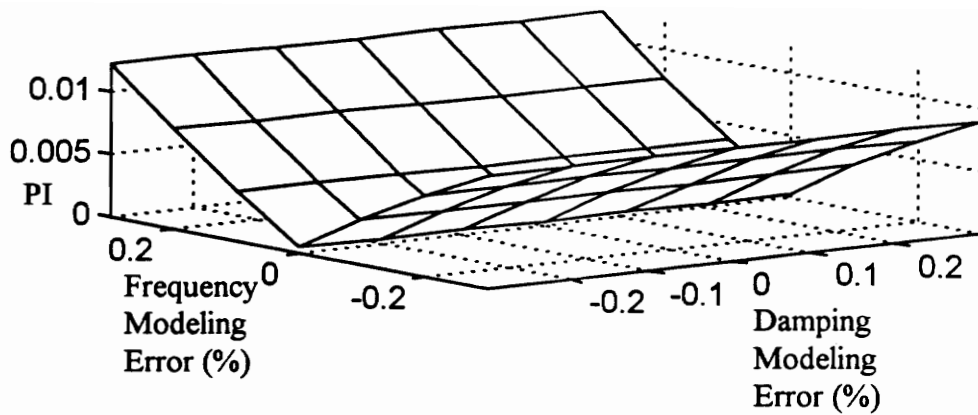


Figure 6-19d

Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Table 6-5d

Fixed Gain MRAC Performance Index (PI x 100) vs. Modeling Error for $\omega_m = 1.17\omega_{pm}$ and $\zeta_m = 0.33$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	0.8954	0.8960	0.8966	0.8972	0.8979	0.8985	0.8991
$-0.2\omega_{pm}$	0.6449	0.6459	0.6469	0.6480	0.6490	0.6500	0.6510
$-0.1\omega_{pm}$	0.3551	0.3561	0.3571	0.3580	0.3590	0.3600	0.3610
No Error	0.0454	0.0419	0.0386	0.0355	0.0334	0.0339	0.0358
$+0.1\omega_{pm}$	0.3922	0.3895	0.3867	0.3839	0.3812	0.3784	0.3757
$+0.2\omega_{pm}$	0.7945	0.7911	0.7877	0.7843	0.7810	0.7776	0.7743
$+0.3\omega_{pm}$	1.2222	1.2183	1.2145	1.2106	1.2068	1.2029	1.1991

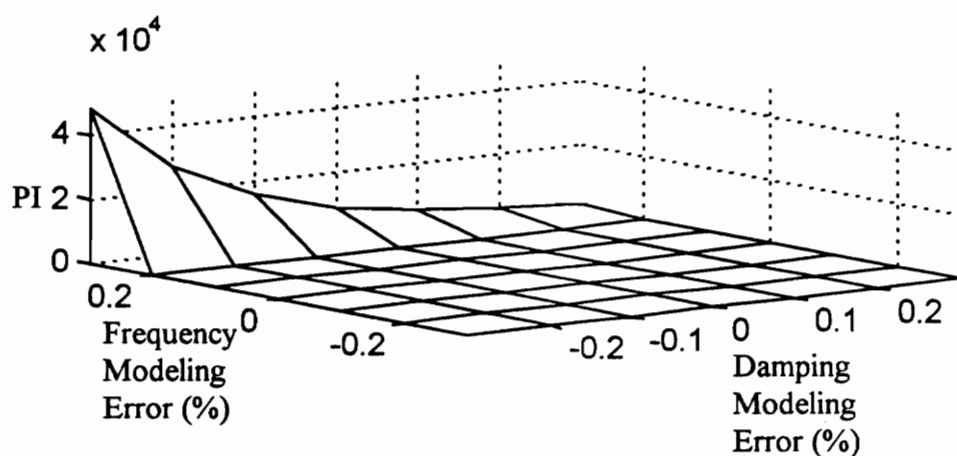


Figure 6-20a
Pole Placement Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Table 6-6a
Pole Placement Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	0.4358	0.4355	0.4353	0.4350	0.4348	0.4345	0.4343
$-0.2\omega_{pm}$	0.4378	0.4372	0.4365	0.4359	0.4353	0.4348	0.4342
$-0.1\omega_{pm}$	0.4389	0.4364	0.4340	0.4317	0.4295	0.4274	0.4254
No Error	0.1148	0.0714	0.0340	0.0015	0.0270	0.0521	0.0743
$+0.1\omega_{pm}$	9.1059	6.6035	4.8889	3.7013	2.8668	2.2723	1.8420
$+0.2\omega_{pm}$	$1.9(10)^3$	$1.2(10)^3$	$0.7(10)^3$	$0.4(10)^3$	$0.3(10)^3$	$0.2(10)^3$	$0.1(10)^3$
$+0.3\omega_{pm}$	$4.8(10)^5$	$2.8(10)^5$	$1.6(10)^5$	$0.9(10)^5$	$0.5(10)^5$	$0.3(10)^5$	$0.2(10)^5$

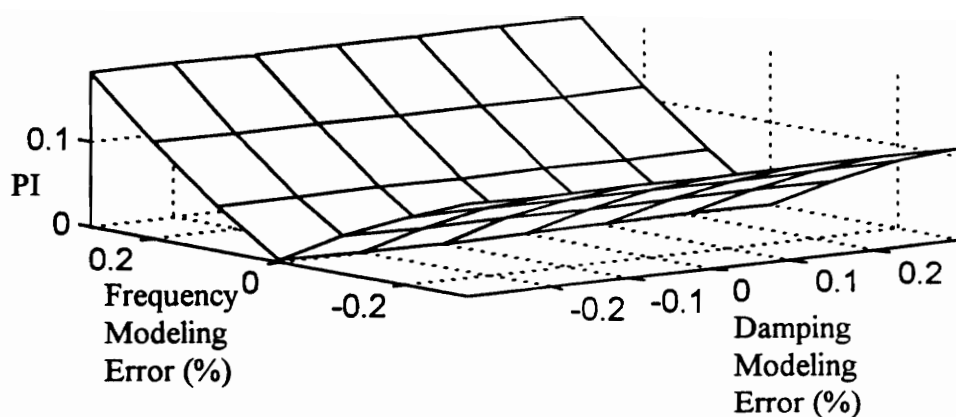


Figure 6-20b
CGT Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Table 6-6b
CGT Performance Index (PI x 10) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	1.0946	1.0938	1.0930	1.0922	1.0915	1.0908	1.0901
$-0.2\omega_{pm}$	0.8010	0.8003	0.7996	0.7989	0.7983	0.7977	0.7971
$-0.1\omega_{pm}$	0.4400	0.4393	0.4388	0.4382	0.4378	0.4375	0.4373
No Error	0.0219	0.0146	0.0074	0.0003	0.0072	0.0143	0.0214
$+0.1\omega_{pm}$	0.5271	0.5259	0.5249	0.5241	0.5233	0.5227	0.5222
$+0.2\omega_{pm}$	1.1424	1.1402	1.1382	1.1362	1.1344	1.1326	1.1309
$+0.3\omega_{pm}$	1.8383	1.8345	1.8308	1.8272	1.8236	1.8202	1.8169

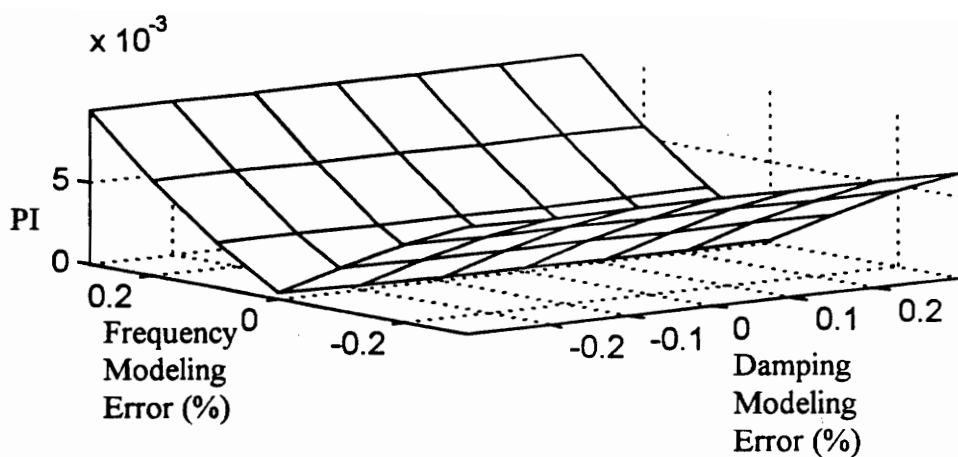


Figure 6-20c
MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Table 6-6c
MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	6.4765	6.4728	6.4693	6.4661	6.4631	6.4600	6.4571
$-0.2\omega_{pm}$	4.6767	4.6735	4.6705	4.6676	4.6651	4.6627	4.6604
$-0.1\omega_{pm}$	2.5709	2.5687	2.5669	2.5656	2.5648	2.5643	2.5643
No Error	0.3041	0.2766	0.2524	0.2397	0.2491	0.2718	0.3002
$+0.1\omega_{pm}$	2.7282	2.7250	2.7224	2.7205	2.7195	2.7192	2.7199
$+0.2\omega_{pm}$	5.7961	5.7897	5.7837	5.7783	5.7736	5.7696	5.7659
$+0.3\omega_{pm}$	9.2627	9.2532	9.2433	9.2343	9.2248	9.2172	9.2092

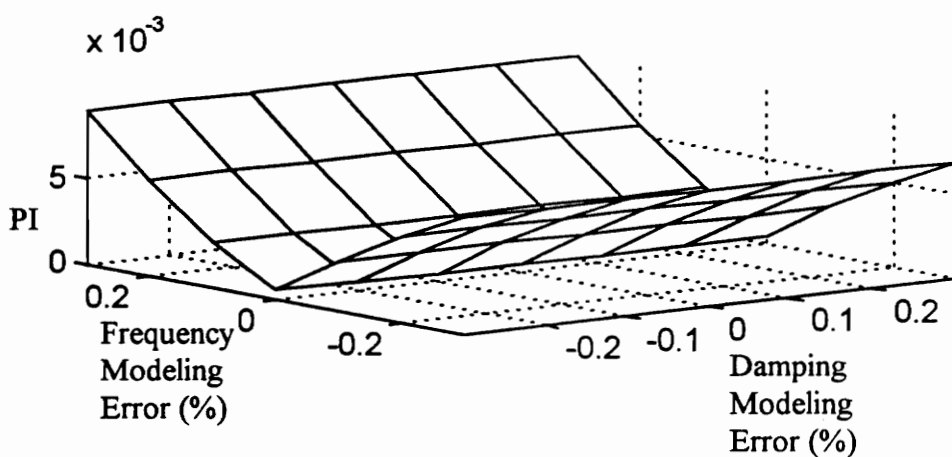


Figure 6-20d
Fixed Gain MRAC Performance vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Table 6-6d
Fixed Gain MRAC Performance Index (PI x 1000) vs. Modeling Error for $\omega_m = 1.5\omega_{pm}$ and $\zeta_m = 0.01$

Plant Model Error	$-0.3\zeta_{pm}$	$-0.2\zeta_{pm}$	$-0.1\zeta_{pm}$	No Error	$+0.1\zeta_{pm}$	$+0.2\zeta_{pm}$	$+0.3\zeta_{pm}$
$-0.3\omega_{pm}$	6.8551	6.8508	6.8467	6.8429	6.8392	6.8355	6.8319
$-0.2\omega_{pm}$	5.0488	5.0447	5.0407	5.0370	5.0335	5.0301	5.0269
$-0.1\omega_{pm}$	2.9223	2.9184	2.9152	2.9124	2.9100	2.9080	2.9064
No Error	0.5279	0.4969	0.4692	0.4531	0.4590	0.4781	0.5025
$+0.1\omega_{pm}$	2.6898	2.6830	2.6769	2.6716	2.6671	2.6633	2.6606
$+0.2\omega_{pm}$	5.6387	5.6296	5.6210	5.6126	5.6050	5.5981	5.5916
$+0.3\omega_{pm}$	8.9993	8.9861	8.9731	8.9603	8.9481	8.9364	8.9251

Chapter 7

Conclusions and Recommendations

Our model reference adaptive control design procedure yielded excellent results in simulation. Considerations of representative model trajectories, representative output error, sampling time, and transient response time factored into our weighting matrix selection. We used Lyapunov stability to results to determine a feedthrough term, minimizing the effects of plant modeling error. In the presence of plant modeling error, these design considerations were sufficient to produce responses well correlated with the reference model's. The model reference adaptive controller performed particularly well over long transient periods, where the adaptive gains could accurately converge to values minimizing output error.

A major benefit of our design procedure is the absence of user-controller interaction. This procedure only requires an approximate model of the plant and a specified reference model. Although our study assumes a very simple system, where the reference model and plant are both second order, this design method is innovative and may be applied to a larger class of systems.

In the comparative analysis of robustness, we found that generally, adaptive control exceeds the ability of linear controllers to track the response of the reference model. Although the linear model reference control method yielded lower performance indexes periodically, the adaptive methods begin to track the model

output in fewer time steps than the linear methods. This attribute of adaptive control is advantageous, but not sufficiently recognized by our quantitative evaluation.

We raised two issues worthy of further investigation. First, we should examine the effects of signal noise for a particular choice of representative error. Our selection was very low, resulting in a relatively high feedback weighting value. This high computational value did not pose a problem in our “sterile” testing environment. However, it is possible to achieve equitable performance for lower feedback weighting values.

The two types of MRAC schemes produced similar results. We hypothesized that MRAC, using a fixed feedback gain, would have a lower sensitivity to noise than the fully adaptive version. The fully adaptive MRAC performed marginally better than its counterpart in our study. We should reexamine these methods under more realistic testing conditions.

We did not attempt control for an input to the reference model. Our design procedure can be easily modified to account for this. Using a reference model command, the plant has a non-zero steady state. This gives the designer a perspective to better evaluate and design the ratio between proportional and integral weighting matrices. A non-zero steady state also accentuates the benefits of adaptive control over linear control.

Another possibility for continuing research is the treatment of higher order and MIMO systems. Our design procedure must be evaluated for this larger class of systems to be truly effective in real world applications.

References

- Ackermann, Jurgen, 1985. Sampled-Data Control Systems, Springer-Verlag, New York, NY.
- Anderson, B. D. O., and Moore J. B., 1990. Optimal Control, Prentice Hall, Englewood Cliffs, New Jersey.
- Anderson, B. D. O. "A System Theory Criterion for Positive Real Matrices", J. SIAM Control, Vol. 5, No. 2, 1967, pp. 171-182.
- Astrom, Karl Johan, and Wittenmark, Bjorn, 1989. Adaptive Control, Addison-Wesley Publishing Company, Inc., New York, NY.
- Bar-Kana, Izhak. "Adaptive Control: A Simplified Approach", Advances in Control and Dynamics, Vol. 25, 1987, pp. 187-235.
- Bar-Kana, Izhak. "On Parallel Feedforward and Simplified Adaptive Control", IFAC Adaptive Systems in Control and Signal Processing, Lund, Sweden, 1986, pp. 99-103.
- Bar-Kana, Izhak, and Kaufman, H. "Discrete Direct Multivariable Adaptive Control", Proc. 1st IFAC Workshop Adaptive Systems in Control and Signal Processing, San Francisco, USA, 1983, pp. 357-361.
- Bryson, Arthur E., and Ho, Yu-Chi, 1975. Applied Optimal Control, Hemisphere Publishing Corporation, Washington, D.C.
- Erzberger, H., "On the Use of Algebraic Methods in the Analysis and Design of Model Following Control Systems", NASA TN D-4663, 1963.
- Franklin, Gene F., Powell, J. David, and Emami-Naeini, Abbas, 1991. Feedback Control of Dynamic Systems, Addison-Wesley Publishing Company, Inc., New York, NY.
- Franklin, Gene F., Powell, J. David, and Workman, Michael L., 1990. Digital Control of Dynamic Systems, Addison-Wesley Publishing Company, Inc., New York, NY.
- Friedland, Bernard, 1986. Control System Design, McGraw Hill, Inc., New York, NY.
- Goodwin, G. C., and Middleton, R. H. "Continuous and Discrete Adaptive Control", Advances in Control and Dynamics, Vol. 25, 1987, pp. 151-185.

Goodwin, G. C., Leal, R. Lozano, Maynes, D. Q., and Middleton, R. H. "Rapprochement between Continuous and Discrete Model Reference Adaptive Control", *Automatica* 22:2 pp. 199-207.

Hale, Francis J., 1988. Introduction to Control System Analysis and Design, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Hang, Chang-Chieh, and Parks, Patrick, C. "Comparative Studies of Model Reference Adaptive Control Systems", *IEEE Transactions on Automatic Control*, Vol. AC-18, No. 5, October 1973, pp. 419-428.

Hitz, L., and Anderson, B. D. O. "Discrete Positive-Real Functions and their Application to System Stability", *Proc. IEE*, Vol. 116, No. 1, January 1969, pp. 153-155.

Ih, C-H C., Bayard, D. S., Ahmed, A., and Wang, S. J. "Experiments in Multivariable Adaptive Control of a Large Flexible Structure", *Proc. Guidance, Navigation, and Control Conf.*, AIAA, Boston, MA, August 1989, pp. 1207-1217

Ionescu, I., and Monopoli, R. V., "Discrete Model Reference Adaptive Control with an Augmented Error Signal", *Automatica* Vol. 13, 1977, p. 507.

Kaufman, H., Bar-Kana, I., and Sobel, K., 1994. Direct Adaptive Control Algorithms, Springer-Verlag, New York, NY.

Kreisselmeir, Gerhard, and Narendra, K. S. "Stable Model Reference Adaptive Control in the Presence of Bounded Disturbances", *IEEE Trans. on Automatic Control*, Vol. AC-27, No. 6, December 1982, pp. 1169-1175.

Kuo, Benjamin C., 1980. Digital Control Systems, Holt, Rinehart and Winston, Inc., New York, NY.

Johnson, C. D. "A New Approach to Adaptive Control", *Advances in Control and Dynamics*, Vol. 27, 1988, pp. 1-72.

Landau, Y.D., 1979. Adaptive Control: The Model Reference Approach, Marcel Dekker, Inc., New York, NY.

LaSalle, J., and Lefshetz, S., 1961. Stability by Lyapunov's Direct Method, Academic Press, New York, NY.

Mabius, L., and Kaufman, H. "An Adaptive Flight Controller for the F-8 without Explicit Parameter Identification", *Proc. 1976 IEEE Conf. on Decision and Control*, Florida, December, 1976, pp. 9-14.

Mabius, L., and Kaufman, H. "An Implicit Algorithm for a Linear Model Reference Control System", Proc. 1975 IEEE Conf. on Decision and Control, Houston, Texas, December 1975, pp. 864-865.

Maciejowski, J. M., 1989. Multivariable Feedback Design, Addison-Wesley Publishing Company, Workingham, England.

Meirovitch, Leonard, 1990. Dynamics and Control of Structures, John Wiley and Sons, Inc., New York, NY.

Messer, R. Scott, "Analytical and Experimental Study of Control Effort Associated with Model Reference Adaptive Control", Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, July 1992.

Messer, R. Scott, Haftka, R. T., and Cudney, H. H. "Analytical and Experimental Study of Control Effort Associated with Model Reference Adaptive Control", Proc. 33rd Structures, Structural Dynamics, and Materials Conf., AIAA, Dallas, Texas, April 1991, pp. 1568-1579.

Miller, D. E., and Davison, E. J. "On Necessary Assumptions in Continuous Time Model Reference Adaptive Control", Proc. 28th Conf. on Decision and Control, Tampa, Florida, December 1989, pp. 1573-1578.

Narendra, K. S., and Lin, Youn-Hao. "Stable Discrete Adaptive Control", IEEE Trans. on Automatic Control, Vol. AC-25, No. 3, June 1980, pp. 456-461.

Narendra, K. S., Lin, Youn-Hao, and Valavani, Lena S. "Stable Adaptive Controller Design, Part II: Proof of Stability", IEEE Trans. on Automatic Control, Vol. AC-25, No. 3, June 1980, pp. 440-448.

O'Brien, M. J., and Broussard, J. R. "Feed-forward Control to Track the Output of a Forced Model", 17th IEEE Conf. on Decision and Control, January 1979, pp. 1149-1155.

Ogata, Katsuhiko, 1990. Modern Control Engineering, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Ortega, Romeo, and Kreisselmeier, Gerhard. "Discrete-Time Model Reference Adaptive Control for Continuous-Time Systems Using Generalized Sampled-Data Hold Functions", IEEE Transactions on Automatic Control, Vol. 35, No. 3, March 1990, pp. 334-338.

Parks, P.C. "Lyapunov Redesign of Model Reference Adaptive Control Systems", IEEE Transactions on Automatic Control, AC-11, July 1966, pp. 362-367.

- Perry, William L., 1988. Elementary Linear Algebra, McGraw Hill, Inc., New York, NY.
- Rao, S. S., 1990. Mechanical Vibrations, Addison-Wesley Publishing Company, New York, NY.
- Slotine, Jean-Jacques E., and Li, Weiping, 1991. Applied Nonlinear Control, Prentice Hall, Inc., Englewood Cliffs, New Jersey.
- Sobel, K., and Kaufman, H. "Direct Model Reference Adaptive Control for a Class of MIMO Systems" *Advances in Control and Dynamic Systems*, Vol. XXIV, 1986, pp. 246-314.
- Sobel, K., Kaufman, H., and Mabius, L. "Implicit Adaptive Control for a Class of MIMO Systems", *IEEE Trans. Aerospace Electron. Syst.*, Vol. 18, No. 5, September 1982, pp. 576-589.
- Sobel, K. "Model Reference Adaptive Control for Multi-Input Multi-Output Systems", Ph.D Thesis, Rensselaer Polytechnic Institute, Troy, New York, June 1980.
- Sobel, K., Kaufman, H., and Mabius, L. "Model Reference Adaptive Control Systems without Parameter Identification", *Proc. 18th Conf. on Decision and Control*, Fort Lauderdale, Florida, 1979, pp. 347-351.
- Su, Wei, and Sobel, K. "A New Implementation of Simple Adaptive Control for Non-ASPR Plants", *Proc. 29th Conf. on Decision and Control*, Honolulu, Hawaii, December 1990, pp. 3251-3253.
- Tao, G., and Ioannou, P. A. "Model Reference Adaptive Control for Plants with Unknown Relative Degree", *Proc. Automatic Control Conf.*, June 1989, pp. 2297-2302.
- Tsytkin, Y. Z., "Adaptation and Learning in Automatic Systems", *Aut. Remote Control* 27, 1966, pp. 16-51.
- Vidyasagar, M., 1993. Nonlinear Systems Analysis, Prentice Hall, Inc., Englewood Cliffs, New Jersey.
- Whitaker H. P., "Design of a Model Reference Adaptive System for Aircraft", M.I.T. Instrumentation Laboratory, Report R-164, Sept., 1958.
- Wolovich, William A., 1994. Automatic Control Systems, Holt, Rhinehart and Winston, Inc., New York, NY.

Appendix A

Controllability and Observability

The tests for controllability and observability are identical for both continuous and discrete time systems. These two tests are discussed for the generalized system, presented as,

$$\dot{x}(t) \text{ or } x(k+1) = Ax(\cdot) + Bu(\cdot) \quad (\text{A-1})$$

and

$$y(\cdot) = Cx(\cdot). \quad (\text{A-2})$$

The vectors x , y , and u are dimensioned $n \times 1$, $q \times 1$, and $m \times 1$, respectively. The matrices A , B , and C are dimensioned $n \times n$, $n \times m$, and $q \times n$, respectively

The test for controllability requires the formation of the controllability matrix ξ .

$$\xi = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \quad (\text{A-3})$$

The pair (A, B) is controllable if

$$\text{rank}(\xi) = n. \quad (\text{A-4})$$

The observability test requires the formation of the observability matrix O .

$$O = [C \quad CA \quad CA^2 \quad \dots \quad CA^{n-1}]^T \quad (\text{A-5})$$

The pair (A, C) is observable if

$$\text{rank}(O) = n. \quad (\text{A-6})$$

Appendix B

Transformation to Feedback Canonical Form

The transformation of both continuous and discrete systems follow the same procedure. For simplicity, we present the method of transformation for the SISO case.

For the generalized system

$$\dot{x}(t) \text{ or } x(k+1) = Ax(\cdot) + Bu(\cdot) \quad (\text{B-1})$$

and

$$y(\cdot) = Cx(\cdot), \quad (\text{B-2})$$

we seek a nonsingular transformation matrix T such that

$$\hat{A} = TAT^{-1} \quad (\text{B-3a})$$

$$\hat{B} = TB \quad (\text{B-3b})$$

$$\hat{C} = CT^{-1} \quad (\text{B-3c})$$

where the minimal realization $\{\hat{A}, \hat{B}, \hat{C}\}$ represents the system in feedback canonical form.

Feedback canonical form is demonstrated as,

$$\hat{A} = \begin{bmatrix} x & x & \cdots & x & x & x \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & \vdots & \ddots & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots & 0 \\ 0 & 1 & \ddots & \vdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \quad (\text{B-4a})$$

and

$$\hat{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \quad (\text{B-4b})$$

The upper row of \hat{A} consists of the coefficients of the system's characteristic equation. With a knowledge of \hat{A} and \hat{B} , the reader can construct a controllability matrix based upon this system realization.

$$\hat{\xi} = [\hat{B} \quad \hat{A}\hat{B} \quad \hat{A}^2\hat{B} \quad \dots \quad \hat{A}^{n-1}\hat{B}] \quad (\text{B-5})$$

Forming the controllability matrix ξ , derived from $\{A, B, C\}$, the transformation matrix may be calculated as,

$$T = \hat{\xi}\xi^{-1}. \quad (\text{B-6})$$

Appendix C

Strictly Positive Real Conditions

The stability discussion of Chapter 4 involves a “leap of faith” that perhaps the incredulous reader would prefer not to make. We hope that this section will provide some insight into the jargon-filled realm of stability. Specifically, we shall derive the strictly positive real conditions as applied to the linear subsystem of each adaptive algorithm. First, we provide a review of the stability conditions from Chapter 4. These equations are placed in a general form for convenience.

Continuous Case I:

$$A^T P + PA = -LL^T < 0 \quad (4.1-10)$$

$$C = B^T P \quad (4.1-5)$$

Continuous Case II:

$$A^T P + PA = -LL^T < 0 \quad (4.2-5a)$$

$$PB = C^T - LW_0 \quad (4.2-5b)$$

$$W_0^T W_0 = J + J^T \quad (4.2-5c)$$

Discrete Case:

$$A^T P A - P = -LL^T < 0 \quad (4.3-10a)$$

$$A^T P B = C^T - LW_0 \quad (4.3-10b)$$

$$W_0^T W_0 = J + J^T - B^T P B \quad (4.3-10c)$$

C.1 Terminology

Let's begin with a few definitions. For the SISO case, the term positive real essentially means that the poles and zeros of a transfer function lie in the left-hand side or on the imaginary axis of the complex plane and that the system transfer function has a relative degree of 0 or 1. The mathematical requirement of a positive real transfer function $Z(s)$ is (Slotine and Li, 1991),

$$\operatorname{Re}[Z(s)] \geq 0 \quad \forall \operatorname{Re}[s] \geq 0. \quad (\text{C1-1})$$

Describing a system as strictly positive real implies that all poles and zeros of a transfer function lie strictly in the left-hand side of the complex plane or

$$\operatorname{Re}[Z(s - \delta)] \geq 0 \quad \forall \operatorname{Re}[s] \geq 0 \text{ and some } \delta > 0. \quad (\text{C1-2})$$

Equation (C1-2) may be difficult to use when applied to higher order transfer functions. As an alternative, we may state that a transfer function $Z(s)$ is strictly positive real if and only if

$$Z(s) \text{ is Hurwitz (eigenvalues are strictly stable)} \quad (\text{C1-3a})$$

and

$$\operatorname{Re}[Z(j\omega)] > 0 \quad \forall \omega \geq 0. \quad (\text{C1-3b})$$

We can infer from Eq. (C1-3b) that the Nyquist plot of $Z(j\omega)$ lies in the right half of the complex plane. From this we may establish the following claims of a strictly positive real system (Slotine and Li, 1991):

- $Z(s)$ has a relative degree of 0 or 1.

- $Z(s)$ is strictly minimum phase (all zeros lie in the left hand complex plane and none on the imaginary axis).

We now extend this discussion to the MIMO case where we have a matrix of transfer functions, $Z(s)$. From Anderson (1967), the $n \times n$ transfer matrix $Z(s)$ is positive real if

$$Z(s) \text{ has elements that are analytic (defined, not infinite) for } \operatorname{Re}[s] > 0, \quad (\text{C1-4a})$$

$$Z^*(s) = Z(s^*) \text{ for } \operatorname{Re}[s] > 0, \quad (\text{C1-4b})$$

and

$$Z(s) + Z^T(s^*) \text{ is positive semi-definite (non-negative) for } \operatorname{Re}[s] > 0. \quad (\text{C1-4c})$$

Note that the superscript asterisk denotes complex conjugation. Equation (C1-4c) can be proven equivalent to

$$Z(j\omega) + Z^T(-j\omega) \text{ is positive semi-definite for } \forall \operatorname{Re}[\omega] \quad (\text{Hitz and Anderson, 1968}) \quad (\text{C1-4d})$$

The transfer matrix $Z(s)$ is strictly positive real if Eq. (C1-4) is satisfied for $Z(s-\delta)$ and some $\delta > 0$ or

$$Z(s) \text{ has elements that are analytic (defined, not infinite) for } \operatorname{Re}[s] \geq 0, \quad (\text{C1-5a})$$

$$Z^*(s) = Z(s^*) \text{ for } \operatorname{Re}[s] \geq 0, \quad (\text{C1-5b})$$

and

$$Z(s) + Z^T(s^*) \text{ is positive definite for } \operatorname{Re}[s] \geq 0. \quad (\text{C1-5c})$$

We have, additionally that

$$Z(j\omega) + Z^T(-j\omega) \text{ is positive definite for } \forall \text{Re}[\omega]. \quad (\text{C1-5d})$$

We shall present an algebraic criterion for the transfer matrix $Z(s)$ to be strictly positive real. Let

$$Y(s) = Z(s) + Z^T(-s). \quad (\text{C.1-6})$$

Note that this implies,

$$Y(s) = Y^T(-s). \quad (\text{C.1-7})$$

If $s = j\omega$ then

$$Y(j\omega) = Z(j\omega) + Z^T(-j\omega). \quad (\text{C.1-8})$$

From equations (C.1-5d) and (C.1-7), we see that $Y(j\omega)$ is positive definite Hermitian.

Because of this property, we may factor $Y(s)$, the $n \times n$ matrix of rank r , such that

$$Y(s) = W^T(-s)W(s) \quad (\text{C.1-9})$$

where $W(s)$ is an $r \times n$ matrix of rank r . Factorization principles dictate that if $Z(s)$ has elements that are Hurwitz then $W(s)$ will as well (Anderson, 1967). The existence of $W(s)$ and its properties is contingent upon $Z(s)$ being strictly positive real. If $W(s)$ has elements that are not Hurwitz, then $Z(s)$ is can not be strictly positive real. We constrain $W(s)$ to have elements that are Hurwitz and develop a realization of $Y(s)$ based upon $W(s)$. From Eq. (C.1-6) we can also realize $Y(s)$ in terms of $Z(s)$. After transforming these two realizations such that they are in the same canonical forms we equate similar terms, algebraically demonstrating the strictly positive real constraints on $Z(s)$.

C.2 Continuous Case I

We represent the linear subsystem in section 4.1 by the input-output transfer function $Z(s)$,

$$Z(s) = C[sI - A]^{-1}B. \quad (C.2-1)$$

Using $Z(s)$, we establish a set of state equations where x is the state vector, u is the input and y is the output.

$$s\mathbf{x}(s) = A\mathbf{x}(s) + B\mathbf{u}(s) \quad (C.2-2)$$

$$\mathbf{y}(s) = C\mathbf{x}(s) \quad (C.2-3)$$

Way may equivalently say that the set $\{A, B, C\}$ is a realization of $Z(s)$. If $\{A, B, C\}$ is a minimal realization (having no extraneous state variables), then there exists an infinity of other minimal realizations that describe the same transfer matrix. We proceed under the assumption that $\{A, B, C\}$ is indeed a minimal realization, and that other minimal realizations can each be calculated using a particular nonsingular transformation matrix T .

Let the set $\{F, K, L^T\}$ represent a minimal realization of $W(s)$. By equating similar realizations of $Y(s)$ we demonstrate the conditions necessary to satisfy the strictly positive real conditions for $Z(s)$. The realizations given by Eq. (C.1-6) and Eq. (C.1-9) are both minimal (Anderson, 1967).

Let the set $\{A_1, B_1, C_1\}$ represent the realization of $Y(s)$ derived from Eq. (C.1-6).

$$\begin{aligned} \{A_1, B_1, C_1\} &\Rightarrow Y(s) = C[sI - A]^{-1}B - B^T[sI + A^T]^{-1}C^T \\ \{A_1, B_1, C_1^T\} &= \left\{ \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}, \begin{bmatrix} B \\ C^T \end{bmatrix}, \begin{bmatrix} C \\ -B^T \end{bmatrix} \right\} \end{aligned} \quad (C.2-4)$$

Let the set $\{A_2, B_2, C_2\}$ represent the realization of $Y(s)$ derived from Eq. (C.1-9).

$$\{A_2, B_2, C_2\} \Rightarrow Y(s) = -K^T [sI + F^T]^{-1} L L^T [sI - F]^{-1} K$$

$$\{A_2, B_2, C_2^T\} = \left\{ \begin{bmatrix} F & 0 \\ L L^T & -F^T \end{bmatrix}, \begin{bmatrix} K \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -K^T \end{bmatrix} \right\} \quad (C.2-5)$$

Assuming that $Z(s)$ is strictly positive real, $W(s)$ must be Hurwitz. We examine the state dynamics matrix of $W(s)$ to derive the constraints for the Hurwitz condition. Let w represent the state vector of $W(s)$. Using Lyapunov's direct method, we formulate a positive definite function in terms of the state variable w .

$$V(w) = w^T P w \quad (C.2-6)$$

where P is an $n \times n$ positive definite matrix. We proceed to develop the first derivative of $V(w)$ and constrain it to be negative definite.

$$\dot{V}(w) = \dot{w}^T P w + w^T P \dot{w} \quad (C.2-7)$$

Proving $W(s)$ to be Hurwitz, we consider the homogeneous equation,

$$\dot{w}(t) = F w(t).$$

We substitute this expression into Eq. (C.2-7), producing

$$\dot{V}(w) = (F w)^T P w + w^T P (F w).$$

Factoring out w , we have

$$\dot{V}(w) = w^T (F^T P + P F) w. \quad (C.2-8)$$

The Lyapunov first derivative must be negative definite to guarantee asymptotic stability. Asymptotic stability implies that F contains strictly stable eigenvalues, or, that the elements of $W(s)$ are Hurwitz. We conveniently assign

$$F^T P + PF = -LL^T < 0 \quad (C.2-9)$$

so that Eq. (C.2-8) is negative definite. This assignment aids in the transformation of $\{A_2, B_2, C_2\}$ into the same canonical form as $\{A_1, B_1, C_1\}$. The new realization is related to $\{A_2, B_2, C_2\}$ by the process,

$$\{A_3, B_3, C_3\} = \{TA_2T^{-1}, TB_2, C_2T^{-1}\}$$

for which

$$T = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix}.$$

We define A_3 as,

$$A_3 = TA_2T^{-1} = \begin{bmatrix} F & 0 \\ F^T P + PF + LL^T & -F^T \end{bmatrix}.$$

Substituting for LL^T using Eq. (C.2-9) reduces A_3 to

$$A_3 = \begin{bmatrix} F & 0 \\ 0 & -F^T \end{bmatrix}. \quad (C.2-10a)$$

Now, we transform the rest of the realization.

$$B_3 = TB_2 = \begin{bmatrix} K \\ PK \end{bmatrix} \quad (C.2-10b)$$

$$C_3 = C_2T^{-1} = [K^T P \quad -K^T] \quad (C.2-10c)$$

The new realization for $Y(s)$ is

$$\{A_3, B_3, C_3^T\} = \left\{ \begin{bmatrix} F & 0 \\ 0 & -F^T \end{bmatrix}, \begin{bmatrix} K \\ PK \end{bmatrix}, \begin{bmatrix} K^T P \\ -K^T \end{bmatrix} \right\}. \quad (C.2-11)$$

With the first and third realizations in the same canonical forms, we may satisfy the strictly positive real conditions for $Z(s)$. We relate this realization to that the defined, yet theoretical matrix $W(s)$. Equating Eq. (C.2-11) to Eq. (C.2-4) shows that the matrices A and F are identical (Anderson, 1967). It follows that A must satisfy Eq. (C.2-10).

$$A^T P + PA = -LL^T < 0. \quad (C.2-12)$$

This is equivalent to Eq. (4.1-10)

Since $K = B$ and $C^T = PK$,

$$C^T = PB.$$

By transposing both sides of the last expression, we arrive at the more conventional presentation used in Eq. (4.1-5).

$$C = B^T P. \quad (C.2-13)$$

C.3 Continuous Case II

We extend the results of section C.2 to examine a system using a feedthrough term. With Eq. (C.1-6) and Eq. (C.1-9) remaining valid, we develop the new realizations of $Z(s)$ and $W(s)$.

$$\{A, B, C, J\} \Rightarrow Z(s) = J + C[sI - A]^{-1} B \quad (C.3-1)$$

$$\{F, K, L^T, W_0\} \Rightarrow W(s) = W_0 + L^T[sI - F]^{-1} K \quad (C.3-2)$$

Realizing the transfer matrix $Y(s)$ in terms of $Z(s)$,

$$\{A_1, B_1, C_1, D_1\} \Rightarrow Y(s) = J + C[sI - A]^{-1}B + J^T - B^T[sI + A^T]^{-1}C^T,$$

or equivalently,

$$\{A_1, B_1, C_1^T, D_1\} = \left\{ \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}, \begin{bmatrix} B \\ C^T \end{bmatrix}, \begin{bmatrix} C \\ -B^T \end{bmatrix}, J + J^T \right\}. \quad (C.3-3)$$

We realize $Y(s)$ is in terms of $W(s)$.

$$\begin{aligned} \{A_2, B_2, C_2, D_2\} \Rightarrow Y(s) &= (W_0^T - K^T[sI + F^T]^{-1}L)(W_0 + L^T[sI - F]^{-1}K) \\ &= W_0^T W_0 + W_0^T L^T[sI - F]^{-1}K \\ &\quad - K^T[sI + F^T]^{-1}L W_0 - K^T[sI + F^T]^{-1}L L^T[sI - F]^{-1}K \end{aligned}$$

This can be written as,

$$\{A_2, B_2, C_2^T, D_2\} = \left\{ \begin{bmatrix} F & 0 \\ LL^T & -F^T \end{bmatrix}, \begin{bmatrix} K \\ L W_0 \end{bmatrix}, \begin{bmatrix} W_0^T L^T \\ -K^T \end{bmatrix}, W_0^T W_0 \right\}. \quad (C.3-4)$$

Using the transformation procedure of Eq. (C.2-10) and the same matrix T , we develop a third realization of $Y(s)$.

$$\{A_3, B_3, C_3^T, D_3\} = \left\{ \begin{bmatrix} F & 0 \\ 0 & -F^T \end{bmatrix}, \begin{bmatrix} K \\ PK + L W_0 \end{bmatrix}, \begin{bmatrix} W_0^T L^T + K^T P \\ -K^T \end{bmatrix}, W_0^T W_0 \right\} \quad (C.3-5)$$

By equating the first and third realizations of $Y(s)$, we develop the strictly positive real constraints on $Z(s)$:

$$A^T P + P A = -L L^T \quad (C.3-6a)$$

$$P B = C^T - L W_0 \quad (C.3-6b)$$

$$W_0^T W_0 = J + J^T \quad (C.3-6c)$$

These conditions are equivalent to Eq. (4.2-5).

C.4 Discrete Case

We develop the discrete strictly positive real constraints for the generalized discrete system,

$$Z(z) = J + C(zI - A)^{-1}B. \quad (C.4-1)$$

From $Z(z)$, we establish the following state equations for which x is the state vector, u is the input, and y is the output.

$$zx(z) = Ax(z) + Bu(z) \quad (C.4-2)$$

$$y(z) = Cx(z) + Ju(z) \quad (C.4-3)$$

Let the set $\{A, B, C, J\}$ be a minimal realization of $Z(z)$. Because we only discuss the characteristics of an open-loop plant (and not any means of control), the results from the continuous discussion on strictly positive real properties may be mapped directly into the z -plane by means of the bilinear transform:

$$s = \frac{z-1}{z+1}. \quad (C.4-4)$$

Let $Z(s)$ be realized by the set $\{A_c, B_c, C_c, J_c\}$. We also give the equations of (C.3-6) a continuous denotation using a subscript “c”.

$$A_c^T P_c + P_c A_c = -L_c^T L_c \quad (C.3-6a)$$

$$P_c B_c = C_c^T - L_c W_{0c} \quad (C.3-6b)$$

$$W_{0c}^T W_{0c} = J_c + J_c^T \quad (C.3-6c)$$

Using the bilinear transform, our goal is to realize Eq. (C.3-6) in terms of discrete matrices. Substituting for s in Eq. (C.2-2) yields,

$$(z - 1)x(k) = A_c(z + 1)x(z) + B_c(z + 1)u(z). \quad (C.4-5)$$

We convert this equation into a finite-difference expression and group future time step values on the left.

$$(I - A_c)x(k + 1) - B_c u(k + 1) = (I + A_c)x(k) + B_c u(k) \quad (C.4-6)$$

To alleviate the problems associated with future time step, we introduce a new vector p , defined as,

$$p(k + 1) = (I - A_c)x(k + 1) - B_c u(k + 1). \quad (C.4-7)$$

The vector p may be equivalently defined as,

$$p(k) = (I - A_c)x(k) - B_c u(k),$$

thus providing a new expression for $x(k)$.

$$x(k) = (I - A_c)^{-1} p(k) + (I - A_c)^{-1} B_c u(k) \quad (C.4-8)$$

Substituting equations (C.4-7) and (C.4-8) into Eq. (C.4-6) produces a state equation in terms p and u , and is given as,

$$p(k + 1) = (I + A_c)(I - A_c)^{-1} p(k) + 2(I - A_c)^{-1} B_c u(k). \quad (C.4-9)$$

Equation (C.4-8) is converted into the transfer function form.

$$\frac{p(z)}{u(z)} = 2 \left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1} (I - A_c)^{-1} B_c \quad (C.4-10)$$

To realize the input-output transfer matrix $Z(z)$ we must substitute Eq. (C.4-8) for $x(k)$ in $y(k) = C_c x(k) + J_c u(k)$.

$$y(k) = C_c \left[(I - A_c)^{-1} p(k) + (I - A_c)^{-1} B_c u(k) \right] + J_c u(k)$$

Terms can be regrouped such that

$$y(k) = C_c (I - A_c)^{-1} p(k) + \left[C_c (I - A_c)^{-1} B_c + J_c \right] u(k). \quad (C.4-11)$$

We convert this finite difference equation into the z domain and then divide through by the input to create the transfer function $Z(z)$.

$$Z(z) = \frac{y(z)}{u(z)} = C_c (I - A_c)^{-1} \frac{p(z)}{u(z)} + C_c (I - A_c)^{-1} B_c + J_c$$

Using Eq. (C.4-10), we substitute the transfer function $\frac{p(z)}{u(z)}$ into $Z(z)$. This yields,

$$\begin{aligned} Z(z) = 2C_c (I - A_c)^{-1} \left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1} (I - A_c)^{-1} B_c \\ + C_c (I - A_c)^{-1} B_c + J_c \end{aligned} \quad (C.4-12)$$

Although this equation is a legitimate realization of $Z(z)$, the current form hinders the ensuing mathematical development. Life will be easier on us if we isolate C_c on the left side of $\left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1}$. Additionally, we will manipulate the constituents of the state dynamics matrix and the plant input matrix to aid further development. Consider the following element of $Z(z)$ from Eq. (C.4-12).

$$(I - A_c)^{-1} \left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1} (I - A_c)^{-1} \quad (C.4-12a)$$

Using the property of matrix inverses: if the matrices A and B are square and invertible and if the matrix AB is invertible then $(AB)^{-1} = B^{-1}A^{-1}$ (Perry, 1988), we can regroup the expression in (C.4-12a).

$$\left\{ \left[zI - (I + A_c)(I - A_c)^{-1} \right] (I - A_c) \right\}^{-1} (I - A_c)^{-1}$$

Within the braces, $(I - A_c)$ distributes over the adjacent term.

$$\left[z(I - A_c) - (I + A_c) \right]^{-1} (I - A_c)^{-1} \quad (\text{C.4-12b})$$

We apply the matrix inverse property, arriving at,

$$\left\{ (I - A_c) \left[z(I - A_c) - (I + A_c) \right] \right\}^{-1}. \quad (\text{C.4-12c})$$

Distributing $(I - A_c)$ over the right side produces

$$\left[z(I - A_c)^2 - (I - A_c)(I + A_c) \right]^{-1}.$$

Using the identity of

$$(I - A_c)(I + A_c) = (I + A_c)(I - A_c),$$

we make a commutative change in the previous expression. We also insert the identity matrix I which shall be defined subsequently.

$$\left[z(I - A_c)^2 - (I + A_c)(I - A_c)I \right]^{-1}$$

If we define the identity matrix such that,

$$I = (I - A_c)(I - A_c)^{-1},$$

substituting for I produces

$$\left[z(I - A_c)^2 - (I + A_c)(I - A_c)(I - A_c)^{-1} \right]^{-1}. \quad (\text{C.4-12d})$$

Since $(I + A_c)$ commutes with $(I - A_c)$, we have

$$\left[z(I - A_c)^2 - (I - A_c)^2(I + A_c)(I - A_c)^{-1} \right]^{-1}. \quad (\text{C.4-12e})$$

The term $(I - A_c)^2$ is factored out to the left. Then we use the property of matrix inverses, yielding,

$$\left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1} (I - A_c)^{-2}. \quad (\text{C.4-12f})$$

We substitute this expression back into Eq. (C.4-12) and arrive at the final realization of $Z(z)$ in terms of continuous matrices.

$$Z(z) = 2C_c \left[zI - (I + A_c)(I - A_c)^{-1} \right]^{-1} (I - A_c)^{-2} B_c + C_c (I - A_c)^{-1} B_c + J_c \quad (\text{C.4-13})$$

The matrices A_c , B_c , C_c and J_c must be defined so that $Z(z)$ assumes the form of Eq. (C.4-1):

$$Z(z) = C(zI - A)^{-1} B + J.$$

Equating Eq. (C.4-13) to Eq. (C.4-1) produces the relations:

$$A = (I + A_c)(I - A_c)^{-1} \quad (\text{C.4-14a})$$

$$B = 2(I - A_c)^{-2} B_c \quad (\text{C.4-14b})$$

$$C = C_c \quad (\text{C.4-14c})$$

$$J = C(I - A_c)^{-1} B_c + J_c \quad (\text{C.4-14d})$$

From Eq. (C.4-14), we can solve A_c , B_c , and J_c in closed form.

$$A_c = (A + I)^{-1}(A - I) \quad (C.4-15a)$$

$$B_c = 2(A + I)^{-2}B \quad (C.4-15b)$$

$$J_c = J - C(A + I)^{-1}B. \quad (C.4-15c)$$

We proceed to substitute for the continuous matrices in equations (C.3-6).

Equation (C.3-6a) becomes

$$(A^T - I)(A^T + I)^{-1}P_c + P_c(A + I)^{-1}(A - I) = -L_c^T L_c < 0. \quad (C.4-16)$$

If we make the assumptions that

$$P_c = 0.5(A^T + I)P(A + I) \quad (C.4-17)$$

and

$$L_c = L, \quad (C.4-18)$$

Eq. (C.4-16) becomes

$$0.5(A^T - I)P(A + I) + 0.5(A^T + I)P(A - I) = -LL^T < 0.$$

Distributing through all terms, he have

$$0.5(A^T P A + A^T P - P A - P) + 0.5(A^T P A - A^T P + P A - P) = -LL^T < 0,$$

which evolves into the final discrete strictly positive real equivalent to Eq. (C.3-6a).

$$A^T P A - P = -LL^T < 0 \quad (C.4-19)$$

This concludes the derivation of Eq. (4.3-10c).

With the established definitions of the continuous matrices, we make substitutions into Eq. (C.3-6b) to arrive at its discrete equivalent. Eq. (C.3-6b) becomes

$$(A^T + I)P(A + I)^{-1}B = C^T - LW_{0c}.$$

Under the assumption that

$$W_{0c} = W_0 - L^T(A + I)^{-1}B, \quad (C.4-20)$$

Eq. (C.3-6b) can evolve as,

$$(A^T + I)P(A + I)^{-1}B = C^T - L[W_0 - L^T(A + I)^{-1}B]$$

$$(A^TP + P)(A + I)^{-1}B = C^T - LW_0 - LL^T(A + I)^{-1}B$$

$$(A^TP + P + LL^T)(A + I)^{-1}B = C^T - LW_0$$

Substituting for LL^T using Eq. (C.4-19) permits further development.

$$(A^TP + P + A^TPA - P)(A + I)^{-1}B = C^T - LW_0$$

$$A^TP(A + I)(A + I)^{-1}B = C^T - LW_0$$

$$A^TPB = C^T - LW_0 \quad (C.4-21)$$

Thus, we have derived the discrete strictly positive real constraint from Eq. (4.3-10b).

Finally, we determine the discrete equivalent to $W_{0c}^T W_{0c} = J_c + J_c^T$. We substitute

for J_c and W_{0c} from equations (C.4-15c) and (C.4-20), respectively.

$$(W_0^T - B^T(A^T + I)^{-1}L)(W_0 - L^T(A + I)^{-1}B) = J - C(A + I)^{-1}B + J^T - B^T(A^T + I)^{-1}C^T$$

Distributing terms on the left side of this equation produces

$$W_0^T W_0 - W_0^T L^T(A + I)^{-1}B - B^T(A^T + I)^{-1}LW_0 + B^T(A^T + I)^{-1}LL^T(A + I)^{-1}B$$

$$= J + J^T - C(A + I)^{-1}B - B^T(A^T + I)^{-1}C^T$$

Next, we group similar terms.

$$\begin{aligned} & W_0^T W_0 + (C - W_0^T L^T)(A + I)^{-1} B + B^T (A^T + I)^{-1} (C^T - L W_0) + B^T (A^T + I)^{-1} L L^T (A + I)^{-1} B \\ & = J + J^T \end{aligned}$$

Recall that

$$A^T P A - P = -L L^T \quad (C.4-19)$$

and

$$A^T P B = C^T - L W_0 \quad (C.4-21)$$

or equivalently,

$$B^T P A = C - W_0^T L^T.$$

We use these results to produce

$$\begin{aligned} & W_0^T W_0 + (B^T P A)(A + I)^{-1} B + B^T (A^T + I)^{-1} (A^T P B) - J - J^T \\ & = B^T (A^T + I)^{-1} (A^T P A - P)(A + I)^{-1} B \\ & = 0.5 B^T \left[(A^T + I)^{-1} (A^T P A - P) \right] (A + I)^{-1} B \\ & \quad + 0.5 B^T (A^T + I)^{-1} \left[(A^T P A - P)(A + I)^{-1} B \right] \end{aligned}$$

Grouping similar terms,

$$W_0^T W_0 - J - J^T$$

$$= 0.5B^T \left\{ \left[(A^T + I)^{-1} (A^T P A - P) - 2P A \right] (A + I)^{-1} + (A^T + I)^{-1} \left[(A^T P A - P)(A + I)^{-1} - 2A^T P \right] \right\} B$$

$$= 0.5B^T (A^T + I)^{-1} \left\{ \left[(A^T P A - P) - 2(A^T + I)P A \right] + \left[(A^T P A - P) - 2A^T P (A + I) \right] \right\} (A + I)^{-1} B$$

$$= -0.5B^T (A^T + I)^{-1} \left[(A^T P A + P A + P A + P) + (A^T P + P + A^T P A + A^T P) \right] (A + I)^{-1} B$$

$$= -0.5B^T (A^T + I)^{-1} \left\{ \left[(A^T + I)P A + P(A + I) \right] + \left[(A^T + I)P + A^T P(A + I) \right] \right\} (A + I)^{-1} B$$

$$= -0.5B^T \left[P A (A + I)^{-1} + (A^T + I)^{-1} P + P(A + I)^{-1} + (A^T + I)^{-1} A^T P \right] B$$

$$= -0.5B^T \left[(P A + P)(A + I)^{-1} + (A^T + I)^{-1} (P + A^T P) \right] B$$

$$= -0.5B^T \left[P(A + I)(A + I)^{-1} + (A^T + I)^{-1} (A^T + I)P \right] B$$

We now arrive at the final expression which is equivalent to Eq. (4.3-10c).

$$W_0^T W_0 = J + J^T - B^T P B \quad (C.4-22)$$

Appendix D

Predicting Representative Trajectory Values

We base weighting matrix calculations on the approximate sizes of model and plant state vectors. If done correctly, the control algorithm achieves:

- (1) Insensitivity to initial conditions
- (2) Minimization of the error between the MRAC proportional gains and the CGT gains

To find representative trajectory values, we first determine the maximum and minimum values of position and velocity of the model. The model's trajectories are governed by their characteristic equation and initial conditions. We assume that there is no command input to the plant. We provide a review from Rao (1990) on free vibration with viscous damping for the underdamped and critically damped cases. Although we specifically treat the reference model, substitutions can be made to evaluate the nominal plant.

D.1 Underdamped Case

The position of the model, $x_{m2}(t)$, is defined in time by

$$x_{m2}(t) = X_m e^{-\zeta_m \omega_{mn} t} \sin(\omega_{md} t + \phi_m). \quad (D.1-1)$$

The terms ζ_m and ω_{mn} are the model's damping ratio and natural frequency, respectively.

The model's damped natural frequency, ω_{md} , is defined as,

$$\omega_{md} = \omega_{mn} \sqrt{1 - \zeta_m^2} \quad (D.1-2)$$

The remainder of the terms of the homogeneous solution in Eq. (D.1-1) are

$$X_m = \sqrt{C_{m1}^2 + C_{m2}^2} \quad (D.1-3)$$

for which,

$$C_{m1} = x_{m2}(t=0) = x_{m20} \quad (D.1-4)$$

and

$$C_{m2} = \frac{x_{m1}(t=0) + \zeta_m \omega_{mn} x_{m20}}{\omega_{md}} = \frac{x_{m10} + \zeta_m \omega_{mn} x_{m20}}{\omega_{md}}, \quad (D.1-5)$$

and

$$\phi_m = \tan^{-1}(C_{m1} / C_{m2}). \quad (D.1-6)$$

By differentiating $x_{m2}(t)$, we derive the model's velocity, $\dot{x}_{m1}(t)$.

$$\dot{x}_{m1}(t) = X_m e^{-\zeta_m \omega_{mn} t} \left[-\zeta_m \omega_{mn} \sin(\omega_{md} t + \phi_m) + \omega_{md} \cos(\omega_{md} t + \phi_m) \right] \quad (D.1-7)$$

To find the relative extrema of position and velocity, we consider the critical points of time where $t = 0$ and $t = \infty$. Additionally, relative extrema of position and velocity occur at times for which velocity and acceleration equal zero, respectively. Let's derive acceleration by taking the first derivative of velocity in Eq. (D.1-7).

$$\ddot{x}_{m1}(t) = -X_m e^{-\zeta_m \omega_{mn} t} \left[(\omega_{md}^2 - \zeta_m^2 \omega_{mn}^2) \sin(\omega_{md} t + \phi_m) + 2\zeta_m \omega_{mn} \omega_{md} \cos(\omega_{md} t + \phi_m) \right] \quad (D.1-8)$$

We find the critical points for position by setting Eq. (D.1-7) equal to zero. We denote these critical times as t_{crlxm2} .

$$\tan(\omega_{md} t_{crlxm2} + \phi_m) = \frac{\omega_{md}}{\zeta_m \omega_{mn}} \quad (D.1-9)$$

The tangent function is periodic and we consider its value for all possibilities resulting in a positive $t_{\text{crmx}2}$. We introduce a counting integer, n , as a counting index for the calculation of critical time points.

$$t_{\text{crxm}2} = \frac{\tan^{-1}\left(\frac{\omega_{\text{md}}}{\zeta_{\text{m}} \omega_{\text{mn}}}\right) - \phi_{\text{m}} + n\pi}{\omega_{\text{md}}} \quad \forall \text{ positive } t_{\text{crmx}2} \text{ and integers } n \quad (\text{D.1-10})$$

Using a similar procedure, we determine the critical time points for velocity by setting Eq. (D.1-8) to zero.

$$t_{\text{crxm}1} = \frac{\tan^{-1}\left(\frac{2\zeta_{\text{m}} \omega_{\text{mn}} \omega_{\text{md}}}{\zeta_{\text{m}}^2 \omega_{\text{m}}^2 - \omega_{\text{md}}^2}\right) - \phi_{\text{m}} + n\pi}{\omega_{\text{md}}} \quad \forall \text{ positive } t_{\text{crmx}1} \text{ and integers } n \quad (\text{D.1-11})$$

D.2 Critically Damped Case

For the circumstance where $\zeta_{\text{m}} = 1$, the position of the model is defined by

$$x_{\text{m}2}(t) = e^{-\omega_{\text{mn}} t} (C_{\text{m}1} + C_{\text{m}2} t), \quad (\text{D.2-1})$$

where

$$C_{\text{m}1} = x_{\text{m}2}(t=0) = x_{\text{m}20} \quad (\text{D.2-2})$$

and

$$C_{\text{m}2} = x_{\text{m}1}(t=0) + \omega_{\text{mn}} x_{\text{m}20} = x_{\text{m}10} + \omega_{\text{mn}} x_{\text{m}20}. \quad (\text{D. 2-3})$$

Differentiating the position equation provides us with an expression for velocity.

$$x_{\text{m}1}(t) = e^{-\omega_{\text{mn}} t} [-\omega_{\text{mn}} C_{\text{m}1} + (1 - \omega_{\text{mn}} t) C_{\text{m}2}] \quad (\text{D. 2-4})$$

Now, we calculate the critical points by first deriving an expression for acceleration.

$$\dot{x}_{m1}(t) = -\omega_{mn} e^{-\omega_{mn} t} \left[-\omega_{mn} C_{m1} + (2 - \omega_{mn} t) C_{m2} \right] \quad (D. 2-5)$$

Similar to the underdamped case, we calculate critical points for position, t_{cmax2} , and velocity, t_{cmax1} .

$$t_{cmax2} = \frac{C_{m2} - \omega_{mn} C_{m1}}{\omega_{mn} C_{m2}} \quad (D. 2-6)$$

$$t_{cmax1} = \frac{2C_{m2} - \omega_{mn} C_{m1}}{\omega_{mn} C_{m2}} \quad (D. 2-7)$$

Vita

Jonathan Hill was born August 2, 1972, in Leesburg, Virginia. After graduating from Thomas Jefferson High School for Science and Technology in 1990, he attended Virginia Tech. Three years later, he graduated with a Bachelor of Science Degree in Mechanical Engineering and enrolled in the graduate program of the same department. After fulfilling the requirements for the Master of Science Degree in August of 1995, he plans to attend the Marshall-Wythe School of Law at the College of William and Mary. In 1998, he hopes to retire from academia.

A handwritten signature in black ink, reading "Jonathan Hill". The signature is written in a cursive style with a large, sweeping initial 'J' and a long, horizontal stroke at the end.