

A PARALLEL OBSERVER SYSTEM FOR MULTIRATE
STATE ESTIMATION AND ALIASED OUTPUT
MEASUREMENTS

By
MAY-WIN L. THEIN

Bachelor of Science
Lehigh University
Bethlehem, Pennsylvania
1991

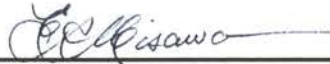
Master of Science
Lehigh University
Bethlehem, Pennsylvania
1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1999

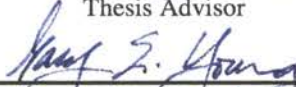
A PARALLEL OBSERVER SYSTEM FOR MULTIRATE
STATE ESTIMATION AND ALIASED OUTPUT
MEASUREMENTS


BY MAY-WIN L. THEIN

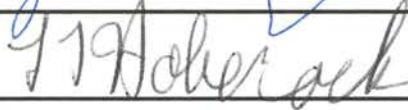
Thesis Approved:



Thesis Advisor









Dean of the Graduate College

PREFACE

Observer-based digital control systems are highly dependent upon the accuracy of the applied estimation technique for the effective update of the control signals. In turn, the estimation technique is dependent upon measurement feedback to update its state estimates. In many cases, the measurement is only available periodically. This work, motivated by the case of a disk drive system, proposes a multirate estimation technique that provides stable and accurate state estimates for every control sample step, despite the fact that the output measurement period is at a slower, but fixed, rate than that of the control input.

I would like to thank Mr. Otis Funches, Mr. Glenn Worstell, and Dr. Evert Cooper of Seagate Technology in Oklahoma City for their support of this research project.

I wish to express my thanks to my advisory committee members, Dr. Larry Hoberock, Dr. Gary Young, and Dr. Robert Whiteley, for their guidance throughout the development of my dissertation. Your comments and suggestions have been most helpful and have greatly improved the quality of my research.

I wish to express special gratitude to my advisor, Dr. Eduardo Misawa, for his continuous support and guidance throughout the duration of my studies at Oklahoma State University. Your sincere interest and concern in my success as a student, a professional, and an individual has helped me achieve many seemingly unreachable goals and prepare for my future life in academia.

I would also like to thank the past and present members of the Advanced Controls Laboratory. I will always remember your friendship, company, and occasional comic relief with fondness. I will truly miss you all.

I give my sincere gratitude to Margaret and Brian Bailey, who have always opened their hearts to provide me with a home away from home. Already considering you as family, I look forward to welcoming you as an official part of my own family very soon.

To my soul mate, my partner in life and my best friend, Richard Tamura, I give my deepest love and most sincere thanks for the unconditional patience, support, and love so generously given throughout these past years. This has been the best and most difficult part of my life, and I am truly grateful that you have stood beside me, and at times carried me, through all of it.

To my brothers and sister Moe, Mimi, and Minn Thein, I give my love and gratitude for the life time of support and love always shown and for their own successes which have, in turn, motivated me and provided the needed momentum during the last years of my studies. Although I was miles apart from you during these past years, I have always felt within me the needed love and encouragement from your supportive words, both spoken and unspoken.

Above all, I extend my utmost gratitude to my parents, San and Wendy Thein, who not only have provided love and support throughout my entire life, but also have provided the foundation for the person I have grown to become. You have taught me a deep perspective of life that I will always carry, and that continues to grow, within me. Neither words nor actions can fully express my gratitude for all that you have done. It is, then, with the deepest respect and love that I dedicate this thesis, the end result of my life's work, to you.

TABLE OF CONTENTS

1	Introduction.....	1
1.1	Problem Statement.....	1
1.2	Overview.....	2
1.2.1	General Multirate Systems.....	2
1.2.2	General Multirate Control Systems.....	3
1.2.3	Multirate Estimation	4
1.3	Contributions	7
1.4	Organization of Contents	8
2	The Problem Statement.....	9
3	A Parallel Observer System for Multirate State Estimation.....	11
3.1	Motivation and Background	11
3.2	The Parallel Observer System (POS).....	12
3.2.1	The Slow Observer System.....	12
3.2.2	The Fast Observer System	14
3.3	Parallel Observer Modifications for System Implementation.....	15
3.3.1	Implementation for Quickly Changing Control Inputs	15
3.3.2	Parallel Observer System with Uncertainty Estimation Compensation.....	18
3.4	Stability Analysis for the Parallel Observer System	22
3.4.1	Proof for Theorem 1: The ON Sample Points.....	25
3.4.2	Proof for Theorem 1: The INTER Sample Points.....	28
3.4.3	Maximum Error Bounds	29

4	Implementation of a Parallel Observer System to a Hard Disk Drive System.....	32
4.1	The IBM Magnetic Head/Disk Assembly.....	33
4.2	The HDA System Model	35
4.3	The Implementation of the Parallel Observer System to the IBM HDA	44
4.3.1	The Parallel Observer System Model	44
4.3.2	The Parallel Observer System Simulation Results.....	46
4.4	Comparison Luenberger Observers	48
4.4.1	The Luenberger Observer Models	49
4.4.2	The Comparative Luenberger Observers Simulation Results	50
4.5	The Present Observer System (PRES).....	52
4.5.1	The Present Observer System (PRES) Model.....	52
4.5.2	The HDA Test Simulation	53
4.5.3	The Present Observer System (PRES) Simulation Results.....	57
4.6	Summary.....	61
5	An Adaptive Parallel Observer System (APOS).....	64
5.1	Motivation.....	64
5.2	Existing Discrete Adaptive Observer Techniques	65
5.2.1	The Proposed Discrete Adaptive Observer System (DAO).....	66
5.2.2	Stability Analysis of the Proposed Adaptive Parallel Observer System	70
5.2.3	Maximum Error Bounds	75
6	Implementation of an Adaptive Parallel Observer System to a Hard Disk Drive System	78
6.1	DAO Implementation Issues.....	78
6.2	Implementation of the DAO to a Second Order Resonance Model.....	79
6.2.1	The Resonance Order Model	79
6.2.2	The DAO System Model	79
6.2.3	Resonance System Simulation Results	84

6.3	Implementation of the APOS to the IBM HDA.....	87
6.3.1	DAO Implementation Issues for the IBM HDA	90
6.3.2	The HDA System Model	91
6.3.3	The DAO System Model	91
6.3.4	HDA Simulation Results.....	97
6.3.5	HDA System Simulation Results.....	99
6.4	Summary Analysis of Simulation Results of the APOS	103
7	Tuning of the DAO Parameters	105
7.1	DAO Implementation Issues for the IBM HDA	105
7.1.1	The Scaled Disc Drive Model.....	106
7.1.2	DAO Disk Drive Implementation	107
7.2	Implementation of a Genetic Algorithm	110
7.3	Simulation Results	112
7.4	Effects of Noise on the APOS	116
7.5	Summary of Results.....	126
8	Damping Ratio and Aliased Resonant Frequency Approximation for the IBM HDA.....	127
8.1	Uniqueness Relationship of the DAO System Parameters.....	128
8.2	DAO Parameter Estimates at Aliased Resonant Frequencies	132
8.2.1	Case I: Slightly Aliased Resonant Frequency	133
8.2.2	Case II: Moderately Aliased Resonant Frequency	134
8.2.3	Aliased Resonant Frequencies in General.....	135
8.3	The Applied Neural Network.....	141
8.3.1	Training.....	141
8.3.2	Training Analysis.....	142
8.3.3	Results of IBM HDA System Parameter Implementation	144
8.4	Natural Frequency and Damping Ratio Approximation of the IBM HDA	146

8.5	Concluding Remarks.....	147
9	Conclusions and Future Work	149
9.1	Conclusions.....	149
9.2	Future Work.....	150
A	SIMULINK Implementation of the Discrete Adaptive Observer	157
B	Adaptive Parallel Observer System IBM HDA Parameter Estimate Results.....	166
C	Discrete Adaptive Observer Parameter Estimate Results of IBM HDA with Noise.....	172
C.1	Parameter Estimates Without Retuning of DAO	172
C.2	Parameter Estimates With Retuning of DAO	174
D	Adapted Genetic Algorithm Code for MATLAB	180

LIST OF TABLES

4.1	IBM HDA System Parameters.....	33
4.2	IBM HDA System States.....	34
7.1	Parameter Values and Estimates.....	116
7.2	Parameter Values and Estimates (With Noise).....	121
7.3	Parameter Values and Estimates (With Noise and Re-tuning).....	126
8.1	Parameter Values and Estimates for $\omega_n = 9.7e3$ rad/sec.....	134
8.2	Parameter Values and Estimates for $\omega_n = 1.3e4$ rad/sec.....	135
8.3	Percentage Estimates Errors for $\omega_n = k \times \omega_N$, $k=1,2,3,4,5$	140
8.4	Case I: Neural Network Parameter Values and Estimates for $\omega_n = 9.7e3$ rad/sec.....	144
8.5	Case II: Neural Network Parameter Values and Estimates for $\omega_n = 1.3e4$ rad/sec.....	144
8.6	Case I: Extraction of ζ and ω_n from DAO parameter estimates for $\omega_n = 9.7e3$ rad/sec.....	146
8.7	Case II: Extraction of ζ and ω_n from DAO parameter estimates for $\omega_n = 1.3e4$ rad/sec.....	146

LIST OF FIGURES

2.1	Cycle and INTER Sample Points.....	10
3.1	The Parallel Observer System.....	13
3.2	The Multirate Segment	13
4.1	IBM Head/Disk Assembly	35
4.2	HDA Power Amplifier	36
4.3	HDA Coil Dynamics.....	36
4.4	HDA VCM and Basecasting Dynamics.....	37
4.5	HDA Actuator Dynamics.....	37
4.6	HDA Power Amplifier Input Current	38
4.7	Simulation Actuator Model	38
4.8	Actuator Modeled Input and Actual Input	41
4.9	Actuator Position Responses.....	42
4.10	Actuator Velocity Responses.....	42
4.11	The Parallel Observer System and HDA System.....	44
4.12	The Parallel Observer System and Comparison Observers	45
4.13	Actuator Position Output and POS Position Estimates	47
4.14	Actuator Velocity and POS Velocity Estimates.....	47
4.15	POS and Comparison Observers Position Estimate Errors	51
4.16	POS and Comparison Observers Velocity Estimate Errors	52
4.17	Simulation Model of the Fourth Order HDA Actuator System for the POS and PRES	54
4.18	Total Input into Fourth Order HDA Actuator Model.....	55
4.19	POS and PRES Position Estimates	58

4.20	POS and PRES Position Estimate Errors.....	59
4.21	POS and PRES Position Estimates with Matched and Unmatched Uncertainties	59
4.22	POS and PRES Position Estimate Errors with Matched and Unmatched Uncertainties.....	60
4.23	PRES and POS Position Estimates with Matched and Increased Unmatched Uncertainties	61
4.24	PRES and POS Position Estimate Error with Matched and Increased Unmatched Uncertainties.....	62
5.1	APOS ON Sample Stability.....	73
6.1	Input Motor Current to a Second Order Resonance System	80
6.2	DAO Implementation on a Resonance Model	81
6.3	The Discrete Resonance Model	81
6.4	The Discrete Adaptive Observer.....	82
6.5	Actuator Position Output and Estimates	85
6.6	Actuator Velocity Output and Estimates	86
6.7	DAO and Slow Observer System Position Estimate Errors.....	86
6.8	DAO and Slow Observer System Velocity Estimate Errors	87
6.9	DAO Parameter Estimate of a_1	88
6.10	DAO Parameter Estimate of a_2	88
6.11	DAO Parameter Estimate of b_1	89
6.12	DAO Parameter Estimate of b_2	89
6.13	Actuator Position Responses.....	92
6.14	Actuator Velocity Responses.....	92
6.15	The Discrete Adaptive Observer.....	93
6.16	APOS, POS and Comparison Observers.....	95
6.17	Actuator Position Output and APOS Estimates	98
6.18	Actuator Velocity Output and APOS Estimates	99
6.19	DAO and Slow Observer System Position Estimate Errors.....	100

6.20	DAO and Slow Observer System Velocity Estimate Errors	100
6.21	APOS and Comparison Observers Position Estimate Errors	101
6.22	APOS and Comparison Observers Velocity Estimate Errors	101
6.23	DAO Parameter Estimate of a_1	102
6.24	DAO Parameter Estimate of b_1	102
7.1	The DAO Disk Drive Implementation	108
7.2	Actual Output and Observer Estimate	113
7.3	Output Observer Estimate Error	113
7.4	Actual Velocity and Observer Estimate	114
7.5	Velocity Observer Estimate Error	114
7.6	Parameter Estimate of a_1	115
7.7	Parameter Estimate of b_1	115
7.8	Actual Output, Signal With Noise, and DAO Estimate	117
7.9	DAO Output Estimate Error (With Noise)	118
7.10	DAO Velocity Estimate Error (With Noise)	118
7.11	Actual Output and Observer Estimate (With Noise)	119
7.12	Actual Velocity and Observer Estimate (With Noise)	119
7.13	Parameter Estimate of a_1 (With Noise)	120
7.14	Parameter Estimate of b_1 (With Noise)	121
7.15	Actual Output and Observer Estimate (With Noise and Re-tuning)	122
7.16	Actual Velocity and Observer Estimate (With Noise and Re-tuning)	123
7.17	Output Estimate Error (With Noise and Re-tuning)	123
7.18	Velocity Estimate Error (With Noise and Re-tuning)	124
7.19	Parameter Estimate of a_1 (With Noise and Re-tuning)	125
7.20	Parameter Estimate of b_1 (With Noise and Re-tuning)	125
8.1	Parameter a_1 as a function of ω_n and	128

8.2	Parameter a_2 as a function of ω_n and ζ	129
8.3	Parameter a_3 as a function of ω_n and ζ	129
8.4	Parameter a_4 as a function of ω_n and ζ	130
8.5	Parameter b_1 as a function of ω_n and ζ	130
8.6	Parameter b_2 as a function of ω_n and ζ	131
8.7	Parameter b_3 as a function of ω_n and ζ	131
8.8	Parameter b_4 as a function of ω_n and ζ	132
8.9	Estimate error of a_1 as a function of ω_n and ζ	136
8.10	Estimate error of a_2 as a function of ω_n and ζ	136
8.11	Estimate error of a_3 as a function of ω_n and ζ	137
8.12	Estimate error of a_4 as a function of ω_n and ζ	137
8.13	Estimate error of b_2 as a function of ω_n and ζ	138
8.14	Estimate error of b_2 as a function of ω_n and ζ	138
8.15	Estimate error of b_3 as a function of ω_n and ζ	139
8.16	Estimate error of b_4 as a function of ω_n and ζ	139
8.17	Mean Squared Error during Training.....	142
8.18	Regression Analysis of ω_n	143
8.19	Regression Analysis of ζ	143
8.20	Neural Network Errors for Extraction of ζ	145
8.21	Neural Network Errors for Extraction of ω_n	145
8.22	Errors for Extraction of ζ Using DAO Parameter Estimates.....	147
8.23	Errors for Extraction of ω_n Using DAO Parameter Estimates.....	148
A.1	The Discrete Adaptive Observer.....	158
A.2	$z(m)$	159
A.3	$\phi(m)$	159

A.4	$\phi_1(m)$	160
A.5	$\phi_2(m)$	160
A.6	$\Gamma(m)$	161
A.7	$p(m)$	162
A.8	State and Output Estimates	163
A.9	State Estimates of $x(m)$	163
A.10	Output Estimates of $y(m)$	164
A.11	$S(m) E p(m)$	164
A.12	Subsystem S: Vector 3	165
A.13	$\phi^T(m)$	165
A.14	Sub S 3	165
B.1	APOS Parameter estimate of a_2	167
B.2	APOS Parameter estimate of a_3	167
B.3	APOS Parameter estimate of a_4	168
B.4	APOS Parameter estimate of a_5	168
B.5	APOS Parameter estimate of b_2	169
B.6	APOS Parameter estimate of b_3	170
B.7	APOS Parameter estimate of b_4	170
B.8	APOS Parameter estimate of b_5	171
C.1	APOS Parameter estimate of a_2 Without Retuning	173
C.2	APOS Parameter estimate of a_3 Without Retuning	173
C.3	APOS Parameter estimate of a_4 Without Retuning	174
C.4	APOS Parameter estimate of b_2 Without Retuning	175
C.5	APOS Parameter estimate of b_3 Without Retuning	175
C.6	APOS Parameter estimate of b_4 Without Retuning	176
C.7	APOS Parameter estimate of a_2 With Retuning	176

C.8	APOS Parameter estimate of a_3 With Retuning.....	177
C.9	APOS Parameter estimate of a_4 With Retuning.....	177
C.10	APOS Parameter estimate of b_2 With Retuning.....	178
C.11	APOS Parameter estimate of b_3 With Retuning.....	178
C.12	APOS Parameter estimate of b_4 With Retuning.....	179

Chapter 1

Introduction

It is known that the performance of observer-based control techniques may be highly sensitive to the accuracy of the applied observer. Thus, the need to investigate methods of state estimation becomes critical. The control input, determined by the observer-based control technique, relies on the accuracy and availability of state estimates calculated from output measurements. This control input must be determined for each sample period (denoted by T_f) so that the system may be adjusted adequately. For many cases, however, the available output measurements are not available every sample period, as in typical disk drive applications. They, in fact, may only be available once every k sample periods.

1.1 Problem Statement

Equation (1.1) describes a linear, single-input continuous time system.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + W(t) \\ y(t) &= Cx(t)\end{aligned}\tag{1.1}$$

The system is controllable and observable. The state vector x is such that $x \in R^{n \times 1}$ and the measurement y is such that $y \in R^{p \times 1}$. The control input is $u(t)$ and $W(t)$ represents any uncertainties in the system.

In discrete time, the traditional estimation problem is to provide reliable state estimates to the controller $u(t)$. It is usually assumed that the measurement y is also available at every control sample step T_f , so that the observer may update its state estimates accordingly.

However, in this study, the output measurement sampling period, denoted by T_s , is such that $T_s = kT_f$, where k is an integer that satisfies the condition that $k \in Z_+$. In other words, the output measurement signal is available only once every k control time steps.

The estimation problem is to provide the estimation-based controller with accurate state estimates at each control sample period T_f , even though the measurement output needed to update the observer is available only once every k control sample steps, or once every system cycle. (Here the term *cycle* refers to the time period T_s , the time period between two consecutive measurements.)

1.2 Overview

1.2.1 General Multirate Systems

As the name suggests, the term multirate refers to any system whose signals are sampled at more than one rate. Multirate system analysis owes its beginnings to the age of digital processing. Significant points of multirate sampled-data control systems were first studied in the 1950's and 1960's with the stability analysis of Kranc [1], Ragazzini and Franklin [2], Jury [3], and then by Kalman and Bertram [4] who also introduce the use of time-invariant difference equations to describe multirate systems. Later, Araki and Yamamoto's work [5] derived a discrete time state space description for multirate systems in addition to four stability criteria.

A solution for the pole placement problem for multivariable, multirate sampled-data systems was proposed by Colaneri et al. [6] via the use of a periodic state observer and an output feedback controller.

Meyer [7] defined the concept of N-periodicity in multirate sampled-data systems and introduced the corresponding shift-varying operators generalized by N-periodicity. Applications of [1] were also used in this work which extended into implementations into multi-input, multi-output systems. In

contrast to the shift-varying parameters used in [7] to define multirate sampled-data systems, Longhi [8] used a state space representation which is an extension of the single rate case. He also analyzed the structural properties inherent to multirate systems, namely those of reachability, controllability, and stabilizability.

Further studies in stability of multirate sampled-data systems were made in Fang and Chu's work [9], where they present sufficient requirements for robust stability.

1.2.2 General Multirate Control Systems

Major work involving general multirate control was developed by Berg et al. [10]. They developed a method for determining sample rate selection, used a discrete time state space to model the system, and developed a constrained optimization control design for multirate systems. The work in [10] was continued by Mason and Berg [11], implementing an infinite time cost function to generate reduced order compensators. Berg and Mason [12] developed a parameter optimization algorithm to generate control laws, assuming a priori knowledge of the system sampling and control input rates.

A periodically time-varying multirate controller was used in Serrano and Ramadge's contribution [13] to solve the Sample Disturbance Decoupling Problem (SDDP) in multirate systems. A feedback control law was used to nullify the effects of the unknown output disturbance at specified periodic rates. Zhu and Skelton [14] confronted the issue of robustness in periodic discrete and multirate systems. They derived bounds for disturbance attenuation and stability with time varying structured and unstructured parameter variations.

For general multirate systems whose output and control sample rates are different, reduced order control methods were developed by Haddad and Kapila [15] and Patton et al. [16]. Haddad and Kapila used a discrete time periodically varying structure for their reduced-order controller. Patton et al. used eigenstructure assignment for use in feedback control. They determined a method to obtain minimum sampling rates and address the issue of intersample ripple.

There is noted work in the area of multirate controls specifically for the case when the output measurement is taken at a slower rate than the input control. An application was presented by

Natarajan [17], who designed a controller for a DC motor using LQG methods and states conditions for zero steady-state ripple. Colaneri et al. [18] addressed this type of multirate system by using LQG techniques and a state augmentation form to derive a prediction algorithm to develop a periodic control law. Scattolini and Schiavoni [19] proposed a technique for these types of multirate systems to stabilize the system despite the existence of added reference signals and disturbances.

1.2.3 Multirate Estimation

Multirate estimation was introduced in single rate systems by Henriksen and Mellichamp [20] for accurate parameter estimation of discrete, stiff systems. Estimation was implemented in input-output form at slow and fast sample rates, so as to be able to estimate parameters in the slow modes and fast modes, respectively.

Apostolakis [21] developed a methodology for prediction type estimation, a technique used in conjunction with a state feedback controller for multirate systems with synchronized sampling rates. In [22] and [23], Haddad et al. developed a reduced order multirate estimator to facilitate the existence of sensors with different sampling rates by the use of linear periodic time-varying (LPTV) state space matrices.

Although multirate systems were not dealt with directly by Demirbas [24], he did develop a discrete state estimation technique for nonlinear systems with missing measurements by use of interpolating functions for use at times when output measurements are not available, in addition to trellis diagram representations. More recently, Savkin and Petersen [25] developed a general information structure to accomplish the dual problem of state estimation and model validation for hybrid systems of continuous time models in addition to continuous time and discrete time measurements. This work also takes into account missing data and system uncertainties.

Chiang [26] developed a controller for actuator and Voice Coil Motor (VCM) dynamics for a disk drive which implements a state estimator predictor running at an integer multiple of the output sample rate.

Prediction/Correction-Based Estimation

In Phillips and Tomizuka's work of [27] and [28] and Lu and Fisher's work of [29] and [30], a multirate estimation technique was developed to increase controller performance. The technique is a linear model-based technique. In this approach, a Prediction/Correction type method is used which performs at two sampling rates: one at the input sampling rate k and the other at the output sampling rate kT_f . Note that $t_i \leq kT_f \leq t_{i+1}$, where $k = 0, 1, 2, \dots, n_i - 1$. Here n_i is such that $(t_i + n_i T_f) \leq t_{i+1} < (t_i + (n_i + 1)T_f)$. A prediction equation estimates the states at every k time steps. Meanwhile a correction equation estimates the states at every kT_f time steps. It follows that at kT_f , the corrector term is updated and that the predictor is updated every k steps. In this way, the controller, which, in this case, is running at a faster rate than that of the measured output, may be updated at its own sampling time.

As mentioned before, this method is model-based. It is a linear estimation technique. Furthermore, the estimator parameters for the predictor equation are calculated on-line, while those of the corrector equations may be calculated off-line.

Numerical Methods-Based Estimation

Ramachandran, Young, and Misawa [31] proposed two numerical methods techniques to estimate multirate systems. The techniques involve the use of a polynomial fit prediction method and another based on a Taylor Series fit prediction method. In the first approach, two polynomial predictions are used: the *Divided Difference Polynomial Prediction* and the *Newton-Gregory Polynomial Prediction*. For the Divided Difference Method, a quadratic is fit using the most recent output sample and those of the two previous ones. In the Newton-Gregory Method, another second degree polynomial is used, except that, here, the polynomial is of a specific form and a calculation of the coefficients is derived via a difference table. The latter approach develops the Taylor Series approximation about the most recent output measurement.

Perturbation Theory-Based Estimation

The work of Kando, Aoyama, and Iwazumi [32] and Shousse and Taylor [33] showed a multirate observer design that is used for singularly perturbed systems. They, also, as with Phillips and Tomizuka [27] and [28], used a slow and fast sampling observer together. The method in [32] uses decomposition to estimate the states. It is assumed that one has available measurements of states in each of the time rate scales, implying that more than one state is being measured although not the same state in both time scales. Shousse and Taylor used two separate reduced order observers, again, one in the fast rate and the other in the slow-sampling rate.

Multirate Kalman Filter Design

Most recently, Hara and Tomizuka [34], [35] proposed a modified state estimator of [36], which is designed to provide estimates at both measurement instances and at intersample instances during the updating of the control input. The method is model-based and uses a multirate Kalman filter design. The design is similar to the prediction/correction technique of [27], [28], [29], and [30] except that the same preceding measurement with the Kalman filter gain is used to update the state estimates each control sample instant until the next output measurement is available. This stochastic estimator is primarily designed to provide smooth state estimates at each control step so that resonance modes of the dynamic system are not excited. Therefore, the accuracy of the state estimates may not be as dependable.

Need for a New Multirate Estimation Approach

All of the multirate estimation techniques above that apply to the problem under study assume that the output measurement signal is not aliased by the existing system resonance frequencies. This may result in poor estimation performance. Therefore, a new multirate estimation technique is needed to proceed towards a solution to account for these aliased frequencies.

1.3 Contributions

In this study, the main contributions of this work are as follows:

1. A multirate estimation technique is proposed that provides reliable state estimates to an estimation-based controller for each control sample step, despite the fact that the output measurement is only available once every k control sample steps. The proposed multirate estimation technique is referred to as the Parallel Observer System.
2. The proposed Parallel Observer System is shown to produce stable estimates for each control sample step. This is shown through the development of a stability proof.
3. It is shown that the errors of the Parallel Observer System's state estimates are bounded even in the presence of matched and unmatched uncertainties. The maximum possible errors bounds are calculated conservatively. These error bounds, take into account matched and unmatched uncertainties, so long as the uncertainties, themselves, are bounded.
4. A modified form of the Parallel Observer System implements an already existing discrete adaptive observer technique to improve overall robustness in the presence of system parameter changes or differing parameter estimates during the control process. This modification results in a new structure of the Parallel Observer System and is called the Adaptive Parallel Observer System.
5. A genetic algorithm is implemented for the automated tuning of the applied discrete adaptive observer technique,
6. An neural network system is implemented to illustrate how the parameter estimates from the Adaptive Parallel Observer System may be used to estimate system damping ratio and aliased resonant frequency values.

1.4 Organization of Contents

The organization of the report is as follows: The problem statement is defined in Chapter 2. In Chapter 3, the multirate estimation technique, the Parallel Observer System is proposed. Optional modifications to the Parallel Observer System to improve performance are described, and a stability proof for the error bounds of the Parallel Observer System's state and uncertainty estimates is presented. The Parallel Observer System is applied to a magnetic disk drive, and the results are compared to that of 2 single rate observers and the present industry standard estimation technique in Chapter 4. In Chapter 5, a modified form of the Parallel Observer System is proposed which implements a Discrete Adaptive Observer system within the proposed multirate estimation technique. The stability proof for this Adaptive Parallel Observer System is given. Chapter 6 simulates this modified Parallel Observer System on the same magnetic disk drive, and the observer performances are compared. Chapter 7 describes the implementation of a genetic algorithm for the automated tuning of the observer parameters for use in the Adaptive Parallel Observer System. A neural network system is applied in Chapter 8 to exemplify how accurate estimates of system damping ratios and aliased resonant frequencies may be obtained from the Adaptive Parallel Observer System. Finally, Chapter 9 presents concluding remarks and areas of future work.

Chapter 2

The Problem Statement

The following equation is a discretized system of Equation (1.1) at a sample period of T_f (the control sampling period):

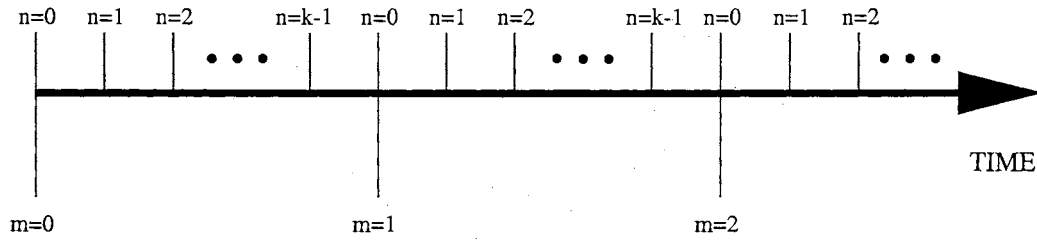
$$\begin{aligned}x(m, n + 1) &= A_f x(m, n) + B_f u(m, n) + W(m, n) \\y(m) &= C x(m)\end{aligned}\tag{2.1}$$

Here, A_f and B_f represent the discretized state space matrices of the system described in Equation (1.1) with the same disturbance $W(m, n)$ and output measurement $y(m)$ but in discrete-time form. The output measurement sampling period is $T_s = kT_f$, where k is an integer value with $k > 1$. Here, m refers to the output measurement cycle number and n refers to the control input sample step within the cycle. The estimation problem in this case is to provide dependable state estimates $\hat{x}(m, n)$ to the observer-based controller $u(m, n)$ at each of its sample steps n , despite the presence of uncertainties and the fact that the output measurements are only available once every cycle m . The cycle lasts a duration of kT_f . As one refers to Equation (2.1), the output measurement is available at $t = (m, 0)$ for every cycle m . In addition, the following terms are defined:

Definition 1. Referring to Figure 2.1, an *ON* sample point refer to those sample points where the output measurement is available, i.e. at $t = (m, 0)$.

Definition 2. An *INTER* sample point refers to those sample points in a cycle where the output

Control Input Points



Output Measurement Points

Figure 2.1: Cycle and INTER Sample Points

measurement is not available, i.e. at $t = (m, n)$ where $n = 1, 2, 3, \dots, k - 1$.

Definition 3. A *cycle*, as described earlier, refers to the time period between two consecutive output measurements and is denoted by m .

It should be noted that noise is not considered in this study. Furthermore, to realistically limit the scope of this project, the following assumptions are made:

Assumption 1. The system described in Equation (2.1) is stable and observable.

Assumption 2. The control input is available at each INTER sample point.

Chapter 3

A Parallel Observer System for Multirate State Estimation

3.1 Motivation and Background

Stable discrete observers have been established for many years. It can be argued, then, that the stability of already established discrete time estimation techniques may be used advantageously in multirate estimation.

The discrete time Luenberger Observer has a simple form and is easy to implement. Given the system in Equation (2.1), the Luenberger Observer has the following form

$$\begin{aligned}\hat{x}(m+1) &= A_d\hat{x}(m) + B_d u(m) + L[y(m) - \hat{y}(m)] \\ \hat{y}(m) &= C\hat{x}(m)\end{aligned}\tag{3.1}$$

where A_d and B_d describe the discretized system in Equation (2.1) and L is a Luenberger gain. The stability of the observer is guaranteed, so long as the observer gain L is chosen such that the eigenvalues of $[A_d - LC]$ are stable. In fact, the speed of convergence of the state estimates may be controlled by choosing appropriate eigenvalues of $[A_d - LC]$. Furthermore, the model-based structure of this state estimation technique offers easier manipulation and implementation

into systems which, themselves, have model-based structures.

It is evident that the use of the Luenberger Observer would be advantageous if its stability and simplicity may be exploited for use in multirate estimation.

3.2 The Parallel Observer System (POS)

A multirate state estimation technique is developed to take into account the situations where the output measurement is only periodically available. The proposed observer design involves two Luenberger Observers running parallel to each other, hence the name **Parallel Observer System (POS)**. The general schematic of the Parallel Observer System is shown in Figure 3.1. The multirate implementation of the system is shown in Figure 3.2. The state estimation method is model based. The POS method assumes that the control input signal into the plant is available (known) at every sampling period T_f and that the output measurement is available once every k sampling periods (or once every cycle) and at a fixed time in each cycle, leaving $k-1$ sampling instances each cycle where the output measurement is not available.

As mentioned earlier, the POS has two separate observers running in parallel. The Slow Observer System performs at the output measurement period T_s , and the Fast Observer System runs at the control input period T_f . Both systems are Luenberger Observers. The Slow Observer System state estimates are used as feedback by the Fast Observer System. By using a full order set of stable Slow Observer estimates during the times of available measurement and using these estimates to drive a Fast Observer, a stable set of estimates is available during all control sample points.

3.2.1 The Slow Observer System

The POS technique relies on a stable discrete-time Luenberger observer which is implemented at the same sample period as that of the output measurement. Since this observer's sample rate is the same as that of the measured output, the feedback output measurement is always available at each sample period T_s . In addition, since the control input sample rate is k times faster than that of the measured output, it is obvious that the control input values are also always available at every sample

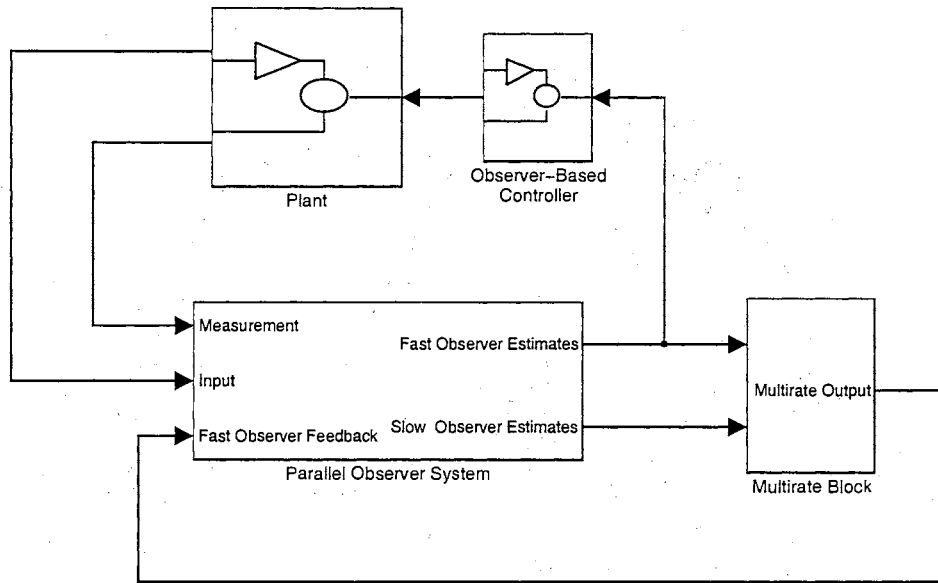


Figure 3.1: The Parallel Observer System

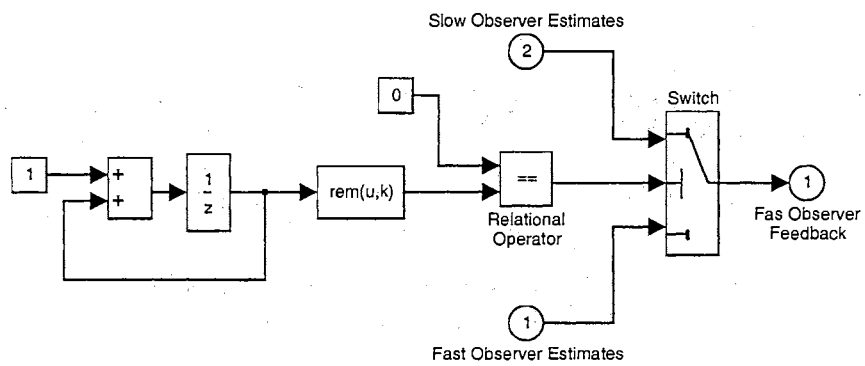


Figure 3.2: The Multirate Segment

time step. This observer system is referred to as the *Slow Observer System*. The structure of the Slow Observer is as follows:

$$\begin{aligned}\hat{x}_s(m+1) &= A_s \hat{x}_s(m) + B_s u(m) + L_s [y(m) - \hat{y}_s(m)] \\ \hat{y}_s(m) &= C \hat{x}_s(m)\end{aligned}\tag{3.2}$$

where m refers to the output measurement time step (or cycle). Also, A_s and B_s define the system in Equation (1.1) discretized with a sample period of T_s (kT_f), and L_s is the Luenberger gain. The values of the Slow Observer state estimates are held constant during the INTER sample points when the output measurement is not available. This structure is of the typical discrete-time Luenberger Observer system. Since it is running at the same rate as the output measurements and always has a control input value available, there is no multirate component.

3.2.2 The Fast Observer System

In this Parallel Observer System, another discrete-time Luenberger observer is running in parallel with the Slow Observer, except this parallel observer is implemented at the same sample rate as that of the input control. It follows that this observer system is referred to as the *Fast Observer System*. The proposed Fast Observer is of the following form:

$$\begin{aligned}\hat{x}_f(m, n+1) &= A_f \hat{x}_f(m, n) + B_f u(m, n) + L_f [y_f(m, n) - \hat{y}_f(m, n)] \\ \hat{y}_f(m, n) &= C_f \hat{x}_f(m, n)\end{aligned}\tag{3.3}$$

where

$$y_f(m, n) = \begin{cases} \hat{x}_s(m, n), & \text{if } n = 0 \\ \hat{y}_f(m, n), & \text{if } n = 1, 2, \dots, k-1 \end{cases}\tag{3.4}$$

Here, n refers to the control input time step and L_f is a Luenberger gain. Referring to Equations (3.3) and (3.4), one can see that this Fast Observer uses the state estimates of the Slow Observer as feedback during the ON sample points. It should be noted that since the Fast Observer uses the full order state estimates of the Slow Observer, C_f is the identity matrix of the same size as that of

the state vector. That is, $y_f \in R^{n \times 1}$. In other words,

$$y_f(m, n) = \begin{cases} \hat{x}_s(m, n), & \text{if } n = 0 \\ \hat{x}_f(m, n), & \text{if } n = 1, 2, \dots, k - 1 \end{cases} \quad (3.5)$$

Again, as with the Slow Observer, the Fast Observer also has the control input values available at each time step, since it is running at the same sample period of T_f . However, the Fast Observer System has *one* available set of estimates from the Slow Observer every k time steps. The whole premise of the Parallel Observer System is that the Slow Observer is stable. It is believed that the Fast Observer may take advantage of the Slow Observer's stability by using the Slow Observer's estimates of its full state for feedback, which occurs once every cycle. During the INTER sample points, however, the Fast Observer is forced to operate in an open-loop fashion.

The advantage to this Parallel Observer System is that state estimates from the Fast Observer are always available for use in the control input calculations. Meanwhile, the Slow Observer provides stable, full order state estimates which serve as feedback measurement equivalents to the Fast Observer during the ON sample points. In this way, the Fast Observer is designed to provide estimates at each control sample time step while converging to the stable Slow Observer's state estimates during the ON sample points.

3.3 Parallel Observer Modifications for System Implementation

3.3.1 Implementation for Quickly Changing Control Inputs

Since the output sampling period T_s is k times slower than that of the control sample period T_f , it is possible, therefore, that the control input changes too quickly for the Slow Observer. Hence, it becomes necessary to account for this quickly changing control input. The following system is that of Equation (1.1) discretized about the sampling period T_s :

$$\begin{aligned} x_s(m+1) &= A_s x(m) + B_s u(m) \\ y(m) &= C x(m) \end{aligned} \quad (3.6)$$

Note that the parameters used here are the same used for the Slow Observer.

If A_f and B_f correspond to the parameters of the discretization of Equation (1.1) about the control sampling period T_f such that

$$T_s = kT_f \quad (3.7)$$

then the following lemmas are proposed.

Lemma 1. *Given Equations (3.6) and (3.7), Equation (1.1) may be written as*

$$x(m+1) = A_f^k x(m) + B_s u(m) \quad (3.8)$$

Proof. This proof is quite trivial, as one notes that

$$A_s = e^{A T_s} \quad (3.9)$$

Here, A refers to the system defined in Equation (1.1). From Equations (3.6) and (3.7), the result in Equation (3.8) is easily established as follows:

$$\begin{aligned} A_s &= e^{A T_f k} \\ &= [A_f]^k \end{aligned} \quad (3.10)$$

□

Lemma 2. *Given Equations (3.6), (3.7), and (3.8) and that $x_s(m) = x_f(m, 0)$, then*

$$B_s u(m) = \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) \quad (3.11)$$

and Equation (3.6) may then be written as

$$x(m+1) = A_s x(m) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) \quad (3.12)$$

Proof. Equation (1.1) may be discretized about T_f such that

$$x(m, k) = A_f^k x(m, 0) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) \quad (3.13)$$

Note that $x(m, k) = x(m + 1)$. Then, by transitivity and Equations (3.8) and (3.13),

$$B_s u(m) = \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i)$$

and through this result and the substitution into Equation (3.6),

$$x(m + 1) = A_s x(m) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i)$$

□

Lemma 2 may also be perceived in the following equivalent manner:

$$\begin{aligned} \hat{x}_s(m + 1) &= A_s \hat{x}_s(m) + \int_0^{T_f} e^{A\tau} d\tau B u(m) \\ &+ \int_{T_f}^{2T_f} e^{A\tau} d\tau B u(m + T_f) + \int_{2T_f}^{3T_f} e^{A\tau} d\tau B u(m + 2T_f) \\ &+ \dots + \int_{T_s - T_f}^{T_s} e^{A\tau} d\tau B u(m + (k - 1)T_f) + L_s [y(m) - \hat{y}_s(m)] \end{aligned} \quad (3.14)$$

or

$$\begin{aligned} \hat{x}_s(m + 1) &= A_f^k \hat{x}_s(m) + \int_0^{T_f} e^{A\tau} d\tau B u(m) \\ &+ \int_{T_f}^{2T_f} e^{A\tau} d\tau B u(m + T_f) + \int_{2T_f}^{3T_f} e^{A\tau} d\tau B u(m + 2T_f) \\ &+ \dots + \int_{T_s - T_f}^{T_s} e^{A\tau} d\tau B u(m + (k - 1)T_f) + L_s [y(m) - \hat{y}_s(m)] \end{aligned} \quad (3.15)$$

Remark 1. Note that the implementation of Lemma (2) to the Slow Observer is now

$$\begin{aligned} \hat{x}_s(m + 1) &= A_s \hat{x}_s(m) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) + L_s [y(m) - \hat{y}_s(m)] \\ \hat{y}_s(m) &= C \hat{x}_s(m) \end{aligned} \quad (3.16)$$

Remark 2. The implementation of Lemma (2) to the Slow Observer may also be written as

$$\begin{aligned} \hat{x}_s(m + 1) &= A_f^k \hat{x}_s(m) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) + L_s [y(m) - \hat{y}_s(m)] \\ \hat{y}_s(m) &= C \hat{x}_s(m) \end{aligned} \quad (3.17)$$

All control inputs at any given time are accounted for, as the control input contribution is being calculated at the higher sample period T_f . In other words, no change in input goes undetected from the Slow Observer. This implementation is consistently used as necessary in multirate systems, as in the works of Hara and Tomizuka [34], [36] and in the works of Lu and Fisher [29], [30].

3.3.2 Parallel Observer System with Uncertainty Estimation Compensation

A common practice in state estimation problems is to augment the system under study to differentiate uncertainties which follow a matching condition, as in Franklin, Powell and Workman [37]. In this way, the observer state is augmented which allows for the estimation of these uncertainties in the system. A discrete-time system assuming matched uncertainties is presented below:

$$\begin{aligned}\dot{x}(t) &= [A + \Delta A] x(t) + [B + \Delta B] [u(t) + \Delta u(t)] \\ y(t) &= Cx(t)\end{aligned}\tag{3.18}$$

where ΔA and ΔB represent the unmodeled plant dynamics and Δu represents added input disturbances to the system. It is assumed that the magnitudes of parameter variances are not significantly large.

The term $w(t)$ represents matched uncertainties and is defined by the following relation:

$$B w(t) = \Delta A x(t) + \Delta B u(t) + B \Delta u(t) + \Delta B \Delta u(t)\tag{3.19}$$

Equation (3.18) may be written in a form with an augmented state to include the matched uncertainties, where

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Bw(t) \\ \dot{w}(t) &= 0 \\ y(t) &= Cx(t)\end{aligned}\tag{3.20}$$

A new augmented state $X(t)$ may be used such that

$$X(t) = \begin{bmatrix} x(t) \\ w(t) \end{bmatrix}\tag{3.21}$$

and Equation (3.20) may be written as

$$\begin{aligned}
\dot{X}(t) &= \begin{bmatrix} \dot{x}(t) \\ \dot{w}(t) \end{bmatrix} \\
&= \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) \\
Y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} X(t)
\end{aligned} \tag{3.22}$$

The corresponding Slow Observer for the discretized system of Equation 3.18, then, is

$$\begin{aligned}
\hat{x}_s(m+1) &= A_s \hat{x}_s(m) + B_s u(m) + B_s \hat{w}_s(m) + L_s [y(m) - \hat{y}_s(m)] \\
\hat{w}_s(m+1) &= \hat{w}_s(m) + l_{sw} [y(m) - \hat{y}_s(m)] \\
\hat{y}_s(m) &= C \hat{x}_s(m)
\end{aligned} \tag{3.23}$$

or, using Lemmas (1) and (2), as

$$\begin{aligned}
\hat{x}_s(m+1) &= A_s \hat{x}_s(m) + \sum_{i=0}^{k-1} A_f^{k-i-1} B_f u(m, i) + B_s \hat{w}_s(m) \\
&\quad + L_s [y(m) - \hat{y}_s(m)] \\
\hat{w}_s(m+1) &= \hat{w}_s(m) + l_{sw} [y(m) - \hat{y}_s(m)] \\
\hat{y}_s(m) &= C \hat{x}_s(m)
\end{aligned} \tag{3.24}$$

where $\hat{w}_s(m)$ is the Slow Observer estimate of the matched uncertainties and L_s and l_{sw} are appropriate Luenberger gains for the state estimates and uncertainty estimation, respectively. Using the same augmented form as Equation (3.22), one may define a new state matrix

$$\hat{X}_s(m) = \begin{bmatrix} \hat{x}_s(m) \\ \hat{w}_s(m) \end{bmatrix} \tag{3.25}$$

such that

$$\begin{aligned}
\hat{X}_s(m+1) &= \begin{bmatrix} \hat{x}_s(m) \\ \hat{w}_s(m) \end{bmatrix} \\
&= \begin{bmatrix} A_s & B_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_s(m) \\ \hat{w}_s(m) \end{bmatrix} + \begin{bmatrix} B_s \\ 0 \end{bmatrix} u(m) \\
&\quad + \begin{bmatrix} L_s \\ l_{sw} \end{bmatrix} [y(m) - \hat{y}_s(m)] \\
\hat{Y}_s(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \hat{X}_s(m)
\end{aligned} \tag{3.26}$$

The Fast Observer is

$$\begin{aligned}
\hat{x}_f(m, n+1) &= A_f \hat{x}_f(m, n) + B_f u(m, n) + B_f \hat{w}_f(m, n) \\
&\quad + L_f [y_f(m, n) - \hat{y}_f(m, n)] + L_{fw} [w_s(m, n) - B_f \hat{w}_f(m, n)] \\
\hat{w}_f(m, n+1) &= \hat{w}_f(m, n) + l_f [y_f(m, n) - \hat{y}_f(m, n)] \\
&\quad + l_{fw} [w_s(m) - \hat{w}_f(m, n)] \\
\hat{y}_f(m, n) &= \hat{x}_f(m, n)
\end{aligned} \tag{3.27}$$

where

$$\begin{aligned}
y_f(m, n) &= \begin{cases} \hat{x}_s(m), & \text{if } n = 0 \\ \hat{x}_f(m, n), & \text{if } n = 1, 2, \dots, k-1 \end{cases} \\
w_s(m, n) &= \begin{cases} \hat{w}_s(m), & \text{if } n = 0 \\ \hat{w}_f(m, n), & \text{if } n = 1, 2, \dots, k-1 \end{cases}
\end{aligned} \tag{3.28}$$

Here, $\hat{w}_f(m, n)$ is the Fast Observer estimation of the matched disturbance, and L_f , L_{fw} , l_f and l_{fw} are Luenberger gains. Again, using the same augmented form as Equation (3.22), one may define a new state matrix

$$\hat{X}_f(m, n) = \begin{bmatrix} \hat{x}_f(m, n) \\ \hat{w}_f(m, n) \end{bmatrix} \tag{3.29}$$

and rewrite Equation (3.27) as

$$\begin{aligned}
\hat{X}_f(m, n+1) &= \begin{bmatrix} \hat{x}_f(m, n+1) \\ \hat{w}_f(m, n+1) \end{bmatrix} \\
&= \begin{bmatrix} A_f & B_f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_f(m, n) \\ \hat{w}_f(m, n) \end{bmatrix} + \begin{bmatrix} B_f \\ 0 \end{bmatrix} u(m, n) \\
&\quad + \begin{bmatrix} L_f & L_{fw} \\ l_f & l_{fw} \end{bmatrix} \begin{bmatrix} [y_f(m, n) - \hat{y}_f(m, n)] \\ [w_f(m, n) - \hat{w}_f(m, n)] \end{bmatrix} \\
y_f(m, n) &= \begin{cases} \hat{x}_s(m), & \text{if } n = 0 \\ \hat{x}_f(m, n), & \text{if } n = 1, 2, \dots, k-1 \end{cases} \\
w_f(m, n) &= \begin{cases} \hat{w}_s(m), & \text{if } n = 0 \\ \hat{w}_f(m, n), & \text{if } n = 1, 2, \dots, k-1 \end{cases} \tag{3.30}
\end{aligned}$$

The following is the described in Equation (3.18) but with additional uncertainties that do not satisfy the matching condition of Equation (3.19):

$$\begin{aligned}
\dot{x} &= [A + \Delta A] x(t) + [B + \Delta B] [u(t) + \Delta u(t)] + \Delta(t) \\
y(t) &= C x(t) \tag{3.31}
\end{aligned}$$

The added uncertainty $\Delta(t)$ represents any existing unmatched uncertainties. Equation (3.31) may be written in a form with an augmented state to include the matched and unmatched uncertainties, that is, any other uncertainties that do not follow the matching condition of Equation (3.19), where

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + Bw(t) + \Delta(t) \\
\dot{w}(t) &= 0 \\
y(t) &= C x(t) \tag{3.32}
\end{aligned}$$

Here, $w(t)$ represents the uncertainties that follow the same matching condition as described in Equation (3.19). Using the same augmented state $X(m, n)$ as in Equation (3.21), Equation (3.32)

may written as

$$\begin{aligned}
\dot{X}(t) &= \begin{bmatrix} \dot{x}(t) \\ \dot{w}(t) \end{bmatrix} \\
&= \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} B & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u(t) \\ \delta_f(m, n) \end{bmatrix} \\
Y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} X(t)
\end{aligned} \tag{3.33}$$

The same Slow Observer System described in Equation (3.26) and the same Fast Observer System described in Equation (3.28) and (3.30) are applied for this case when unmatched uncertainties are added. The Slow and Fast Observer cannot detect the unmatched uncertainties, but can only compensate for them by using the observer correction feedback terms.

3.4 Stability Analysis for the Parallel Observer System

The following terms are defined for use in this section of development of stability proofs:

$$\begin{aligned}
e_s &= x(m) - \hat{x}_s(m) \\
e_{ws} &= w(m) - \hat{w}_s(m) \\
e_f &= x(m, n) - \hat{x}_f(m, n) \\
e_{wf} &= w(m, n) - \hat{w}_f(m, n) \\
\varepsilon_{sf}(m) &= \hat{x}_s(m) - \hat{x}_f(m, 0) \\
\varepsilon_{wsf}(m) &= \hat{w}_s(m) - \hat{w}_f(m, 0)
\end{aligned} \tag{3.34}$$

The proof of convergence for the Slow and Fast Observer Systems involve manipulations of Lemma 2 and the use of eigenvalue stability analysis. The proposed Slow Observer from Equation (3.23) is

$$\begin{aligned}
\hat{x}_s(m+1) &= A_s \hat{x}_s(m) + B_s u(m) + L_s e_s(m) \\
\hat{w}_s(m+1) &= \hat{w}_s(m) + B_s u(m) + l_{sw} e_s(m) \\
\hat{y}_s(m) &= C \hat{x}_s(m)
\end{aligned} \tag{3.35}$$

and the Fast Observer from Equation (3.27) is

$$\begin{aligned}
\hat{x}_f(m, n+1) &= A_f \hat{x}_f(m, n) + B_f u_f(m, n) + L_f \varepsilon_{sf}(m, n) + L_{fw} \varepsilon_{wsf}(m, n) \\
\hat{w}_f(m, n+1) &= \hat{w}_f(m) + l_f \varepsilon_{sf}(m, n) + l_{fw} \varepsilon_{wsf}(m, n) \\
\hat{y}_f(m, n) &= C \hat{x}_f(m, n)
\end{aligned} \tag{3.36}$$

where

$$\begin{aligned}
\varepsilon_{sf}(m, n) &= \begin{cases} \varepsilon_{sf}(m), & \text{if } n = 0 \\ 0, & \text{if } n = 1, 2, \dots, k-1 \end{cases} \\
\varepsilon_{wsf}(m, n) &= \begin{cases} \varepsilon_{wsf}(m), & \text{if } n = 0 \\ 0, & \text{if } n = 1, 2, \dots, k-1 \end{cases}
\end{aligned} \tag{3.37}$$

Assumption 3. The system described in Equation (3.18) is stable.

Assumption 4. The frequency content of the system in study is less than the half the sampling frequency.

Assumption 5. The sample rate of the Fast Observer System is fast enough to react to any corrections from the Slow Observer System.

Furthermore, define the matrices S , f , and F_f such that

$$S = \begin{bmatrix} [A_s - L_s C] & B_s \\ -l_{sw} C & 1 \end{bmatrix} \tag{3.38}$$

$$f = \begin{bmatrix} [A_f - L_f] & B_f \\ -l_f & 1 \end{bmatrix} \tag{3.39}$$

and

$$F_f = \begin{bmatrix} [A_s - A_f^{k-1} L_f] & [B_s - A_f^{k-1} L_f w] \\ -l_f & (1 - l_{fw}) \end{bmatrix} \tag{3.40}$$

Denoting $\delta_s(m)$ as the unmatched uncertainties during the ON sample points and $\delta_f(m, n)$ as the unmatched uncertainties during the INTER sample points, the following assumption is made:

Assumption 6. The unmatched uncertainties $\delta_s(m)$ and $\delta_f(m, n)$ are bounded such that

$$\max_i |\delta_s(i)| \leq \delta_{s_0} \quad (3.41)$$

$$\max_j |\delta_f(m, j)| \leq \delta_{f_0} \quad (3.42)$$

where δ_{s_0} and δ_{f_0} are finite values.

Theorem 1. *Let Assumptions (3) through (6) hold. In addition, it is assumed that k is a finite integer. Given the system in Equation (3.31), define the same S , f , and F_f described in Equations (3.38) through (3.40). If the proposed Slow Observer System described in Equation (3.23) and the Fast Observer System described in Equations (3.27)-(3.28) are designed such that the poles of S , f , and F_f are stable, then*

1. *The Slow Observer state estimates are stable and the errors of the state and uncertainty estimates have a finite bound.*
2. *The Fast Observer state and uncertainty estimates converge to the Slow Observer state and uncertainty estimates, respectively, during the ON sample points with a finite error bound.*
3. *The Fast Observer state and uncertainty estimates are stable during the INTER sample points.*

Furthermore, at a general time step n , the state estimates error has a finite bound of

$$\begin{aligned} \|e_f(m, n+1)\|_\infty &\leq A_0^{n+1} F_0 + A_0^n C_{f_0} \\ &\quad + \frac{1 - A_0^{n+1}}{1 - A_0} [B_0(\omega_0 + C_{\omega_0}) + \delta_{f_0}] \end{aligned} \quad (3.43)$$

and the uncertainty estimation error has a finite bound of

$$|e_{\omega_f}(m, n+1)| \leq \omega_0 + C_{\omega_0} \quad (3.44)$$

Proof: The proof for this theorem has two parts. First, bounded error is shown for the ON sample points. Then the proof is completed by showing bounded error during the INTER sample points for a finite k . The first part involves analyzing an augmented error dynamics matrix comprised of: (1) the error dynamics of the Slow Observer estimates of states and uncertainties to the actual states

and uncertainties, respectively, and (2) the error dynamics between the Slow and Fast Observers. The second part involves analyzing the error dynamics of the Fast Observer and calculating the propagated error during each of the INTER sample points.

Remark 3. Note that S and f are the exact structures of the dynamics of the Slow and Fast Observers, respectively. Therefore, the eigenvalues of S and f are, also, identically those of the Slow and Fast Observers.

3.4.1 Proof for Theorem 1: The ON Sample Points

The system in Equation (3.32) is represented at the slow and fast sampling time rates and the resulting systems are

$$\begin{aligned} x(m+1) &= A_s x(m) + B_s u(m) + B_s w(m) + \delta_s(m) \\ w(m+1) &= w(m) \\ y(m) &= C x(m) \end{aligned} \tag{3.45}$$

at the slow sampling rate and

$$\begin{aligned} x(m, n+1) &= A_f x(m, n) + B_f u(m, n) + B_f w(m, n) + \delta_f(m, n) \\ w(m, n+1) &= w(m, n) \\ y(m, n) &= C x_f(m, n) \end{aligned} \tag{3.46}$$

at the fast sampling rate, noting that the sets of model parameters (A_s, B_s) and (A_f, B_f) are analogous to those of the discretized systems at the slow and fast sample periods, T_s and T_f , respectively.

For the ON sample points, the Fast Observer may be summed at each INTER sample point n until $n = k$ and, therefore, can be represented as

$$\begin{aligned} \hat{x}_f(m+1) &= A_s \hat{x}_f(m) - A_f^{k-1} L_f \varepsilon_{sf}(m) + B_s \hat{w}_f - A_f^{k-1} L_{fw} \varepsilon_{wsf}(m) \\ \hat{w}_f(m+1) &= \hat{w}_f(m) - l_f \varepsilon_{sf}(m) - l_{fw} \varepsilon_{wsf}(m) \end{aligned} \tag{3.47}$$

Thus, the corresponding error dynamics between the Slow Observer System and the Fast Observer System during the ON sample points are

$$\begin{aligned}\varepsilon_{sf}(m, n+1) &= L_s C e_s(m) + [A_s - A_f^{k-1} L_f] \varepsilon_{sf}(m) + [B_s - A_f^{k-1} L_{fw}] \varepsilon_{wsf}(m) \\ \varepsilon_{wsf}(m, n+1) &= l_{sw} C e_s(m) - l_f \varepsilon_{sf}(m) + [1 - l_{fw}] \varepsilon_{wsf}(m)\end{aligned}$$

Using the error definitions given in Equation (3.34), $e_s(m)$ and $e_{ws}(m)$ are such that the error dynamics for the Slow Observer System with uncertainty estimation is

$$e_s(m+1) = [A_s - L_s C] e_s(m) + B_s e_{ws}(m) + \delta_s(m) \quad (3.48)$$

$$e_{ws}(m+1) = e_{ws}(m) - l_{sw} C e_s(m) \quad (3.49)$$

If one uses an augmented form of the system such that

$$E_{wsf}(m) = \begin{bmatrix} e_s(m) \\ e_{ws}(m) \\ \varepsilon_{sf}(m) \\ \varepsilon_{wsf}(m) \end{bmatrix} \quad (3.50)$$

then the augmented system error dynamics becomes,

$$E_{wsf}(m+1) = A_{wsf} E_{wsf}(m) + \Delta_{wsf}(m) \quad (3.51)$$

where

$$A_{wsf} = \begin{bmatrix} [A_s - L_s C] & B_s & 0^{n \times n} & 0^{n \times 1} \\ -l_{sw} C & 1 & 0^{1 \times n} & 0 \\ L_s C & 0^{n \times 1} & [A_s - A_f^{k-1} L_f] & [B_s - A_f^{k-1} L_{fw}] \\ l_{sw} C & 0 & -l_f & (1 - l_{fw}) \end{bmatrix} \quad (3.52)$$

and

$$\Delta_{wsf} = \begin{bmatrix} \delta_s(m) \\ 0^{(n+2) \times 1} \end{bmatrix} \quad (3.53)$$

Furthermore, Equation (3.52) may be written as

$$A_{wsf} = \begin{bmatrix} S & 0^{(n+1) \times (n+1)} \\ \Lambda_s C & F_f \end{bmatrix}$$

where S and F are defined previously and

$$\Lambda C_s = \begin{bmatrix} L_s C & 0^{n \times 1} \\ l_{sw} C & 0 \end{bmatrix}$$

If E_0 is defined to be $E_{wsf}(m=0)$, then the propagated error based on E_0 and $\Delta_{wsf}(m)$ may be written as function of cycle m , such that if the error matrix $E_{wsf}(m)$ is summed at each ON sample point for for a general sample step m ,

$$E_{wsf}(m+1) = [A_{wsf}]^{m+1} E_0 + \sum_{i=0}^m [A_{wsf}]^{m-i} \Delta_{wsf}(i) \quad (3.54)$$

From the restriction on S , f and F defined in the theorem problem statement, it is known that A_{wsf} is stable. Let

$$A_{wsf_0} = \|A_{wsf}\|_{i\infty} \quad (3.55)$$

Then, because of the stability of A_{wsf} ,

$$\lim_{m \rightarrow \infty} [A_{wsf_0}]^m = 0 \quad (3.56)$$

In addition, it is previously assumed that $\delta_s(m)$ is bounded by δ_{s_0} , which means that

$$\|\Delta_{wsf}\|_{\infty} = \delta_{s_0} \quad (3.57)$$

Hence, it is shown that

$$\begin{aligned} \left\| \lim_{m \rightarrow \infty} E_{wsf}(m) \right\|_{\infty} &= \lim_{m \rightarrow \infty} \left[[A_{wsf_0}]^{m+1} E_0 + \left\| \sum_{i=0}^m [A_{wsf}]^{m-i} \Delta_{wsf}(i) \right\|_{\infty} \right] \\ &= \lim_{m \rightarrow \infty} \left[\sum_{i=0}^m [A_{wsf_0}]^{m-i} \delta_{s_0} \right] \\ &= \lim_{m \rightarrow \infty} \left[\delta_{s_0} \times \frac{1 - [A_{wsf_0}]^{m+1}}{1 - A_{wsf_0}} \right] \\ &= \frac{\delta_{s_0}}{1 - A_{wsf_0}} \end{aligned} \quad (3.58)$$

Therefore, it is shown that Part (1) and Part (2) of Theorem 1 are proven.

3.4.2 Proof for Theorem 1: The INTER Sample Points

Define the following parameters such that

$$\begin{aligned}
f_0 &= e_f(m, 0) \\
w_0 &= e_{wf}(m, 0) \\
f_{s_0} &= \varepsilon_f(m, 0) \\
w_{s_0} &= \varepsilon_{wf}(m, 0) \\
C_f &= L_f f_{s_0} + L_{fw} w_{s_0} \\
C_w &= l_f f_{s_0} + l_{fw} w_{s_0} \\
A_0 &= \|A_f\|_{i\infty} \\
B_0 &= \|B_f\|_{\infty} \\
F_0 &= \|f_0\|_{\infty} \\
C_{f_0} &= \|C_f\|_{i\infty} \\
C_{w_0} &= |C_w| \\
\omega_0 &= |w_0|
\end{aligned} \tag{3.59}$$

Using the error definitions given in Equations (3.34) and the Equations (3.30) and (3.46), $e_f(m, n)$ and $e_{wf}(m, n)$ are such that the error dynamics for the Fast Observer System with uncertainty estimation are

$$\begin{aligned}
e_f(m, n+1) &= A_f e_f(m, n) + B_f e_{wf}(m, n) - L_f \varepsilon_{sf}(m, n) \\
&\quad - L_{fw} \varepsilon_{wsf}(m, n) + \delta_f(m, n) \\
e_{wf}(m, n+1) &= e_{wf}(m, n) - l_f \varepsilon_{sf}(m, n) - l_{fw} \varepsilon_{wsf}(m, n)
\end{aligned} \tag{3.60}$$

Using the expressions defined for Theorem (1), one may calculate the resulting propagated error as

$$e_f(m, n+1) = A_f^{n+1} f_0 - A_f^n C_f + \sum_{i=0}^n A_f^i B_f w_0 - \sum_{i=0}^n A_f^i B_f C_w - \sum_{i=0}^n A_f^i \delta_f(m, i) \tag{3.61}$$

and

$$e_{wf}(m, n+1) = w_0 - C_w \tag{3.62}$$

for the Fast Observer state estimates and uncertainty estimates, respectively. Utilizing Schwartz's Inequality and the triangle inequality, the error bound of Equation (3.61) is

$$\begin{aligned} \|e_f(m, n+1)\|_\infty &\leq A_0^{n+1} F_0 + A_0^n C_{f_0} + \sum_{i=0}^n A_0^i B_0 \omega_0 \\ &\quad + \sum_{i=0}^n A_0^i B_0 C_{w_0} + \sum_{i=0}^n A_0^i \delta_{f_0} \end{aligned} \quad (3.63)$$

and

$$|e_{w_f}(m, n+1)| \leq \omega_0 + C_{w_0} \quad (3.64)$$

Using finite series analysis, one can simplify the second, third and fourth terms of Equation (3.63).

To satisfy Assumption 3, $A_0 < 1$. Then,

$$\begin{aligned} \sum_{i=0}^n A_0^i B_0 \omega_0 &= B_0 \omega_0 \frac{1 - A_0^{n+1}}{1 - A_0} \\ \sum_{i=0}^n A_0^i B_0 C_{w_0} &= B_0 C_{w_0} \frac{1 - A_0^{n+1}}{1 - A_0} \\ \sum_{i=0}^n A_0^i \delta_{f_0} &= \delta_{f_0} \frac{1 - A_0^{n+1}}{1 - A_0} \end{aligned} \quad (3.65)$$

The resulting calculation of the error bound of the Fast Observer estimates is

$$\|e_f(m, n+1)\|_\infty \leq A_0^{n+1} F_0 + A_0^n C_{f_0} + \frac{1 - A_0^{n+1}}{1 - A_0} [B_0 (\omega_0 + C_{w_0}) + \delta_{f_0}] \quad (3.66)$$

Q.E.D

Remark 4. It is proven that the state and uncertainty estimates of the Slow Observer are stable and that the Fast Observer's state and uncertainty estimates converge to that of the Slow Observer's state and uncertainty estimates, respectively, during the ON sample points. Then, it can be inferred that the state and uncertainty estimates of the Fast Observer are stable with a finite error bound during the ON sample points.

3.4.3 Maximum Error Bounds

Remark 5. In the worst case scenario, the error in the state estimates for the Fast Observer System continue to increase during the INTER sample points until the start of the next cycle (i.e., the next

ON point). Under this condition, the greatest maximum bound for any INTER sample point occurs at the time step just before the next ON point. At this time, $n = k - 1$ such that $t = (m, n + 1) = (m, k - 1)$ (t).

Corollary 1. *Given the results of Theorem 1 and the worst case scenario in which the maximum estimation error occurs at the time step of $t = (m, k - 1)$, an upper bound of the error estimates of the Fast Observer state estimates $\hat{x}_f(m, k - 1)$ is*

$$\begin{aligned} \|e_f(m, k - 1)\|_\infty &\leq A_0^{k-1}F_0 + A_0^{k-2}C_{f_0} \\ &\quad + \frac{1 - A_0^{k-1}}{1 - A_0} [B_0(w_0 + C_w) + \delta_{f_0}] \end{aligned} \quad (3.67)$$

and the upper bound of the uncertainty estimation $\hat{w}_f(m, k - 1)$ is

$$|e_{wf}(m, k - 1)| \leq w_0 + C_w \quad (3.68)$$

Proof. The proof, here, is straightforward. Using the fact that, at the sample point prior to the following cycle period, as described in Remark 5, $n = k - 2$ in Equations (3.64) and (3.66). Substituting this relation into Equations (3.43) and (3.44), the result is the same as that described in Equations (3.67) and (3.68), respectively. \square

Remark 6. The reader should note that the above bounds may be extremely conservative and may not be of practical use except to show that the INTER sample estimate bound is, indeed, finite.

Corollary 2. *If there are no unmatched uncertainties in the system and the Slow and Fast Observers have the same initial conditions, the error bound is further reduced to*

$$\begin{aligned} \|e_f(m, k - 1)\|_\infty &\leq A_0^{k-1}F_0 + B_0\omega_0 \frac{1 - A_0^{k-1}}{1 - A_0} \\ \|e_{wf}(m, k - 1)\|_\infty &\leq \omega_0 \end{aligned} \quad (3.69)$$

Proof. The proof, here, is again straightforward in that now, all terms except A_0 , B_0 , ω_0 , and F_0 are eliminated from Equations (3.67) and (3.68). \square

Remark 7. Furthermore, if the initial conditions of the actual states and uncertainties are the same as those for the Slow Observer and the Fast Observer, the parameters ω_0 and F_0 become 0. These initial conditions settings, of course, result in perfect state and uncertainty estimates. This can be easily seen, as all the remaining terms in Equation (3.69) are eliminated.

Chapter 4

Implementation of a Parallel Observer System to a Hard Disk Drive System

In this chapter, the Parallel Observer System is applied to an IBM disk drive system running in open-loop control. The POS is implemented on a reduced order form of this magnetic Head/Disk Assembly (HDA). Input disturbances and unmodeled resonance dynamics are incorporated into the disk drive system to illustrate the performance of the POS in the presence of matched and unmatched uncertainties.

For further illustrations, the multirate POS is then compared to two single rate Luenberger Observers. One observer runs at the rate of the output measurement and the other runs at the rate of the control input. Both are Luenberger Observers and have control input values and output measurements available at each of their respective sample time steps so that a basis of “best” and “worst” case estimates can be compared. In addition, the performance of the POS is compared to that of the method currently being implemented in the standard disk drive. This method is referred to as the Present Observer System.

Primary coil resistance, R_1	4.0 Ω
Shorted turn reflected to primary, R_2	2.33 Ω
Coil leakage inductance, L_1	0.171 mH
Shorted turn leakage inductance, L_2	0.137 mH
Coil and shorted turn mutual inductance, L_3	5.95 mH
Power amplifier output current sense resistor, R_{pa}	0.25 Ω
Actuator Mass, M_{ACT}	120 grams
Actuator back EMF constant, K_{BEMF}	15 Volts/meter/sec
Actuator force factor, K_F	15 $\frac{\text{Newton}}{\text{amp}}$
Voltage Controlled Motor (VCM) mass	2.0 kilograms
Base casting mass (including spindle)	4.0 kilograms
VCM/damper natural resonant frequency, ω_{VCM}	50.2 $\cdot \pi \frac{\text{rad}}{\text{sec}}$
VCM/damper damping ratio, ζ_{VCM}	0.707
Base casting/shock mounts natural resonant frequency, ω_{nsm}	5.2 $\cdot \pi \frac{\text{rad}}{\text{sec}}$
Base casting/shock mounts damping ratio, ζ_{nsm}	0.707
Track Pitch	2400 TPI (tracks per inch)
Disk Diameter	5 $\frac{1}{4}$ inches

Table 4.1: IBM HDA System Parameters

4.1 The IBM Magnetic Head/Disk Assembly

The disk drive in study, as previously mentioned, is that of an IBM Magnetic Head/Disk Assembly (HDA), as described in [38] and [39]. The SIMULINK diagrams which describe this system as modeled in [38] are shown in Figures 4.1 through 4.5. (The reader is referred to [40] and [41] for more information on MATLAB/SIMULINK). In addition, the disk drive parameters and states are described by Tables 4.1, 4.2 and the following relations:

$x_1(t)$	coil current
$x_2(t)$	shorted turn current
$x_3(t)$	motor voltage
$x_4(t)$	power amplifier compensation capacitor voltage
$x_5(t)$	actuator position
$x_6(t)$	actuator velocity
$x_7(t)$	VCM position
$x_8(t)$	VCM velocity
$x_9(t)$	base casting position
$x_{10}(t)$	base casting velocity

Table 4.2: IBM HDA System States

$$\begin{aligned}
\text{denom} &= L_1 L_2 + L_1 L_3 + L_2 L_3 \\
a_{11} &= \frac{-R_1(L_2 + L_3)}{\text{denom}} \\
a_{12} &= \frac{-R_2 L_3}{\text{denom}} \\
a_{14} &= \frac{-(L_2 + L_3)K_{ref}}{\text{denom}} \\
a_{21} &= \frac{-R_1 L_3}{\text{denom}} \\
a_{22} &= \frac{-R_2(L_1 + L_3)}{\text{denom}} \\
K_{VCM} &= M_{VCM} \cdot \omega_{VCM}^2 \\
D_{VCM} &= 2M_{VCM} \cdot \zeta_{VCM} \cdot \omega_{VCM} \\
K_{sm} &= (M_{VCM} + M_{BC})\omega_{nsm}^2 \\
D_{sm} &= 2(M_{VCM} + M_{BC})\zeta_{BC} \cdot \omega_{nsm}
\end{aligned} \tag{4.1}$$

The HDA is modeled by four main dynamic parts: the power amplifier dynamics, the coil dynamics, the Voice Coil Motor (VCM) and Basecasting dynamics, and the actuator dynamics. The input into the system is the desired coil voltage, which is fed through the power amplifier. The

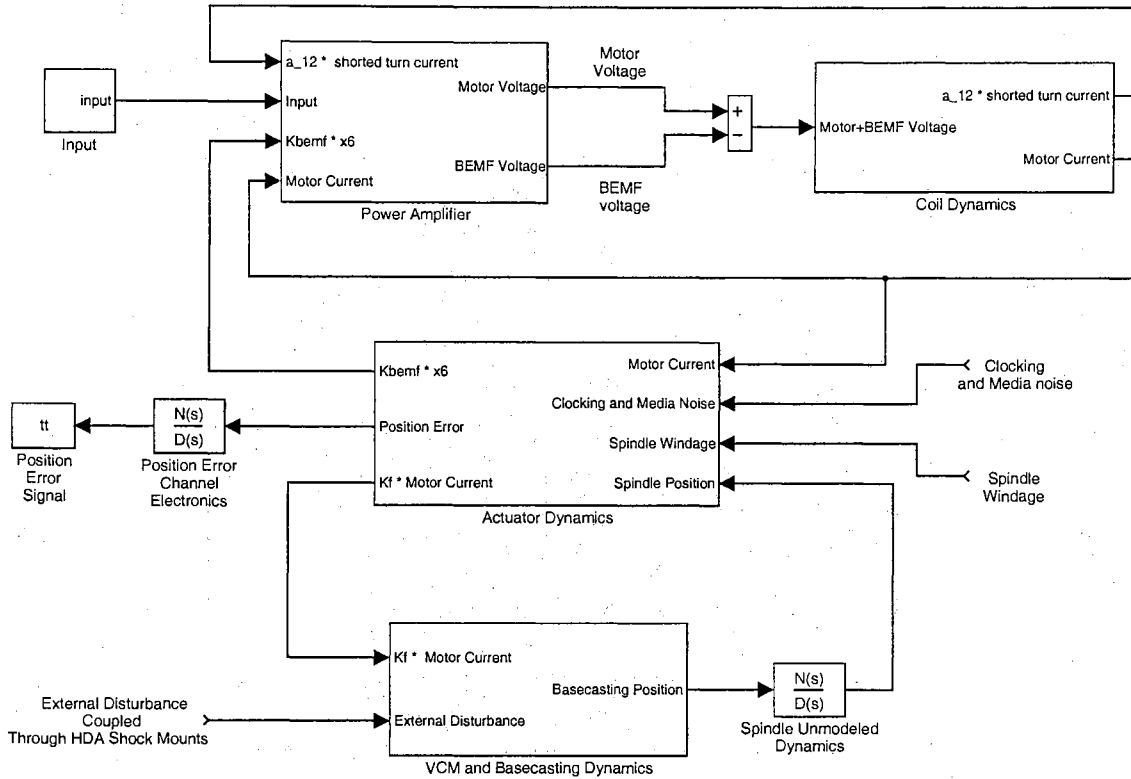


Figure 4.1: IBM Head/Disk Assembly

final output of the system is the position error signal, the relative error displacement between the read/write head and the center of the track. (For more details, the reader is referred to [38] and [39].) The actual input driving the power amplifier and, thus, the entire HDA system is a test input signal comprised of a series of step inputs, as shown in Figure 4.6.

4.2 The HDA System Model

The full model is of a tenth order linear system. However, for purposes of simulation, a reduced second order actuator dynamics model is used, placed in series with a second order resonance dynamics model. The SIMULINK diagram illustrating the model in study is shown in Figure 4.7. The input to the system is the motor current and is acting in open-loop control, which is the output of the coil dynamics. The output of the reduced order system is the actuator position.

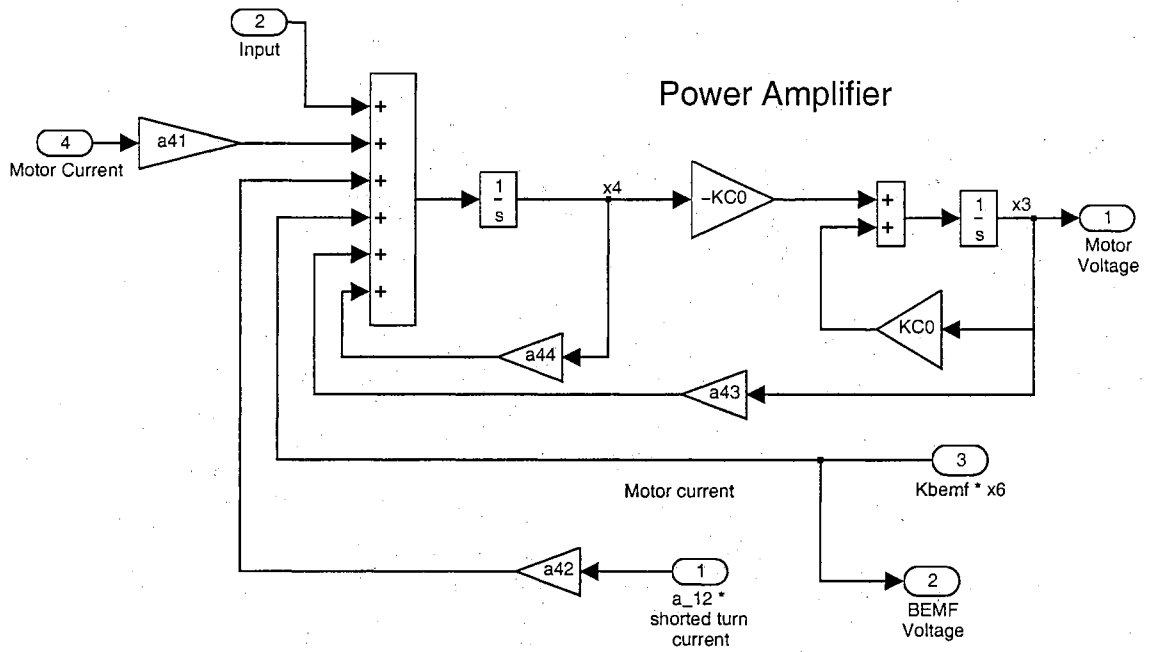


Figure 4.2: HDA Power Amplifier

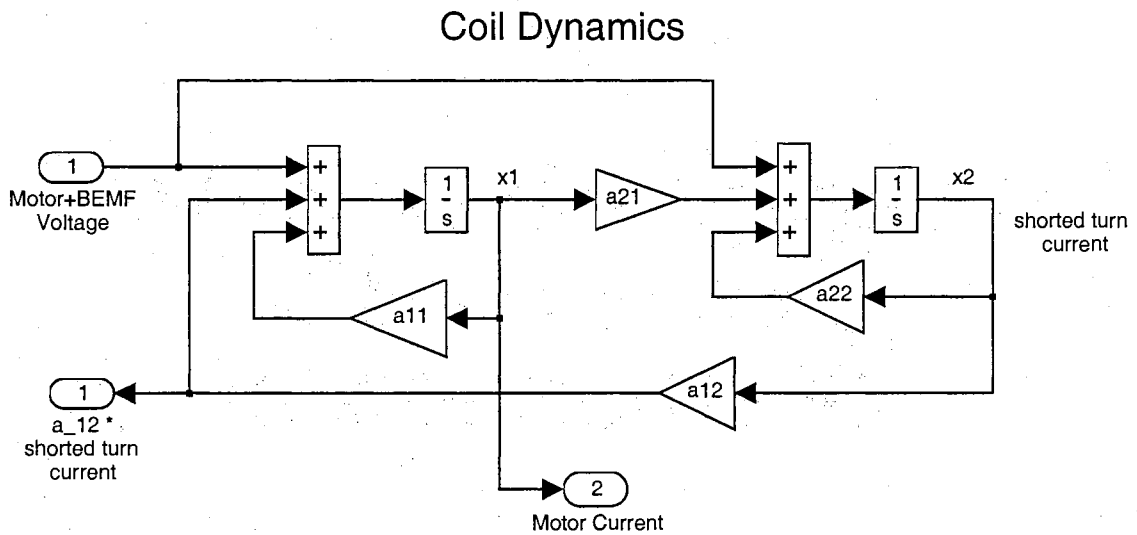


Figure 4.3: HDA Coil Dynamics

VCM and Basecasting Dynamics

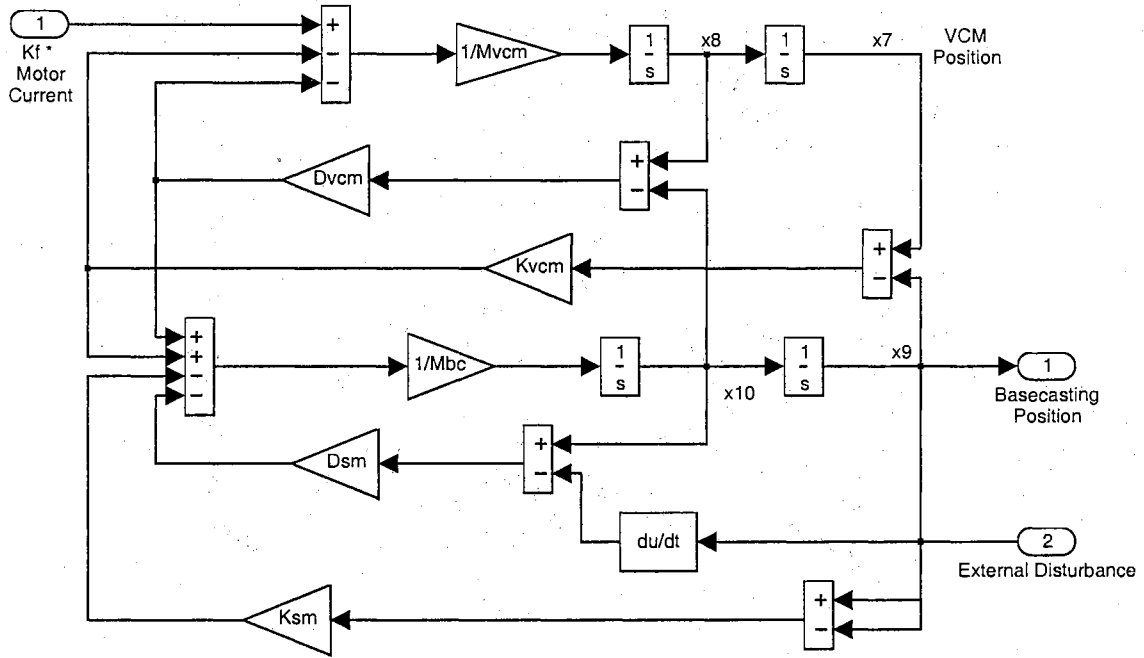


Figure 4.4: HDA VCM and Basecasting Dynamics

Actuator Dynamics

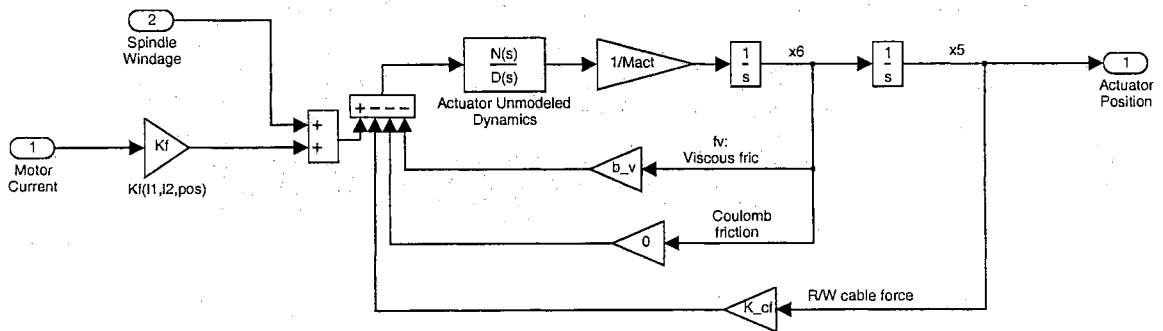


Figure 4.5: HDA Actuator Dynamics

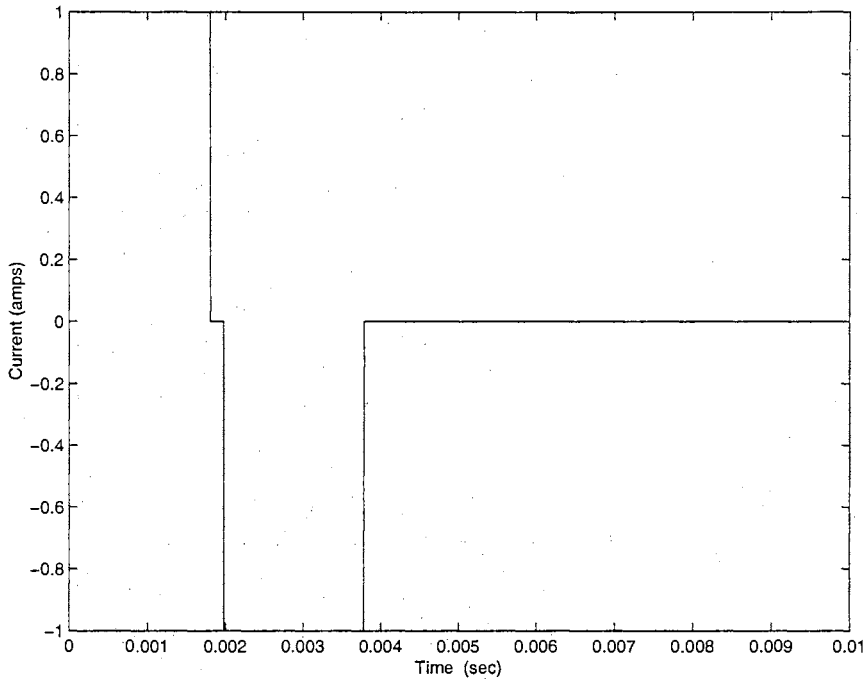


Figure 4.6: HDA Power Amplifier Input Current

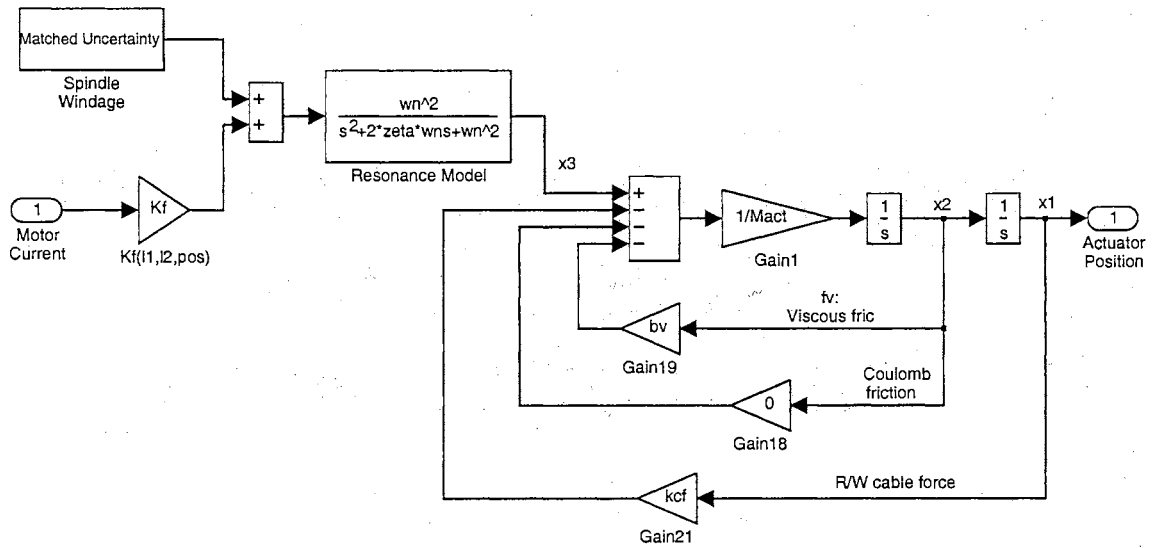


Figure 4.7: Simulation Actuator Model

The second order resonance model $R(s)$ which precedes the actuator dynamics is

$$R(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.2)$$

where ω_n is the resonance natural frequency and ζ is the damping ratio. In this simulation, $\omega_n = 350\text{rad/s}$ and $\zeta = 0.30$.

Neglecting Coulombic friction, the two second order systems joined together in series fashion results in the linear fourth order model:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{cf}}{M_{act}} & -\frac{b_v}{M_{act}} & \frac{1}{M_{act}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_f\omega_n^2 \end{bmatrix} u(t)$$

$$y(t) = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \quad (4.3)$$

Here, x_1 and x_2 denote the actuator position and velocity, respectively, while x_3 and x_4 denote the resonance dynamics. In addition, the motor current is $u(t)$ and the actuator position is $y(t)$. Also, M_{act} corresponds to the actuator mass, K_{cf} corresponds to the R/W cable force spring coefficient, b_v corresponds to the viscous friction coefficient, and K_f corresponds to the actuator force factor.

In this simulation of the HDA system, the disk drive model is run in continuous time, and sampled data signals of $u(t)$ and $y(t)$ are obtained from the continuous time system at sampling periods of T_f and $T_s = kT_f$, respectively, where k is 5. No process or measurement noises are introduced.

The Parallel Observer System is applied to the magnetic Head/Disk Assembly in discrete time. While the actual control input signal $i_m(t)$ to the actuator system is the motor current, and an added spindle windage input is incorporated as a sinusoid rather than a random input such that

$$u(t) = i_m(t) + 2\sin(185t) \quad (4.4)$$

This spindle windage input acts as the unmodeled matched uncertainty to the system. The model

used for the Parallel Observer System is the same as that of Equation (4.3) except for the additional spindle windage input and that the resonant frequency and damping ratio from the resonance model is estimated to be at different values than the actual respective values. That is,

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{cf}}{M_{act}} & -\frac{b_v}{M_{act}} & \frac{1}{M_{act}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\hat{\omega}_n^2 & -2\hat{\zeta}\hat{\omega}_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_f\hat{\omega}_n^2 \end{bmatrix} \hat{u}(t)$$

$$y(t) = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \quad (4.5)$$

where $\hat{\zeta}$, $\hat{\omega}_n$, and $\hat{u}(t)$ are such that

$$\begin{aligned} \hat{\zeta} &= 0.85 \zeta \\ \hat{\omega}_n &= 0.85 \omega_n \\ \hat{u}(t) &= i_m(t) \end{aligned} \quad (4.6)$$

and M_{act} , K_{cf} , b_v , and K_f are such that

$$\begin{aligned} M_{act} &= 120 \text{ grams} \\ K_{cf} &= 0.04 \text{ N/in} \\ b_v &= 0.01 \text{ N/(in/s)} \\ K_f &= 15 \text{ N/amp} \end{aligned} \quad (4.7)$$

It should be noted that the differences in the parameters of the resonance model correspond to unmatched uncertainties in the actuator dynamics. To illustrate the effects of the added matched and unmatched uncertainty dynamics, a plot of the position and velocity of the actual input and the estimated input are shown in Figure 4.8. In addition, a plot of the nominal model, the model accounting for matched uncertainties, and the actual system with matched and unmatched uncertainties are shown in Figures 4.9 and 4.10, respectively. These plots show the degree to which the

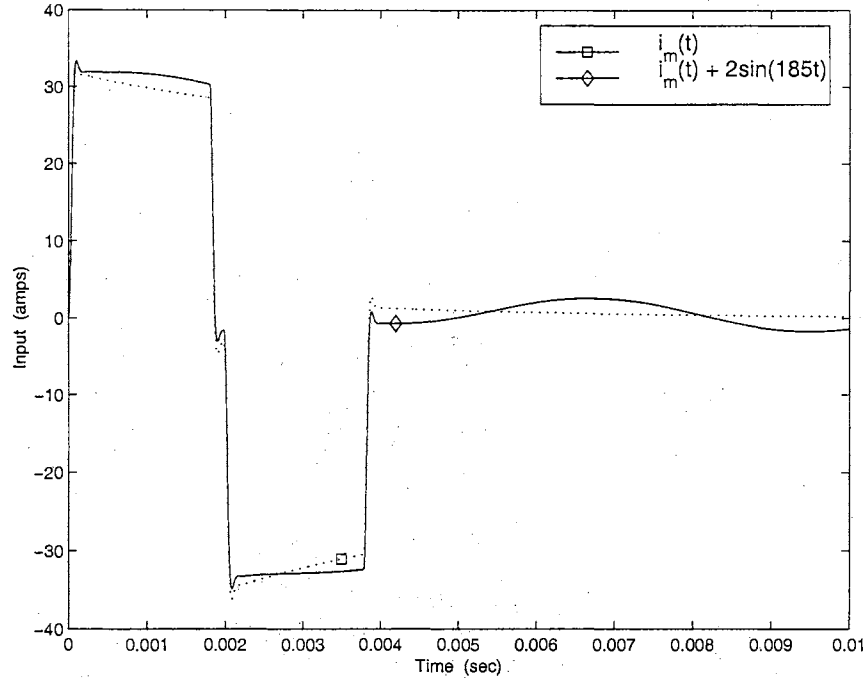


Figure 4.8: Actuator Modeled Input and Actual Input

matched and unmatched uncertainties change the actual system and to the extent to which the applied observers must adjust and correct themselves for these uncertainties.

Equation (4.5) is discretized via method of zero order hold at sampling periods of $T_f = 70\mu s$ for the Fast Observer System and $T_s = 5T_f = 350\mu s$ for the Slow Observer system. Also, the state vectors of both systems are respectively augmented to $X(m)$ and $X(m, n)$ for the Slow and Fast Observers, respectively, as discussed in Chapter 4, so as to accommodate matched uncertainty estimation.

The approximations $\hat{\omega}_n$ and $\hat{\zeta}$ are used for the resonant frequency and damping ratio, respectively, to calculate the system model parameters A_s , B_s , A_f , and B_f for the system model discretized at the corresponding slow and fast sample rates. The resulting final augmented discrete state space model does not take into account any of the matched or unmatched uncertainties.

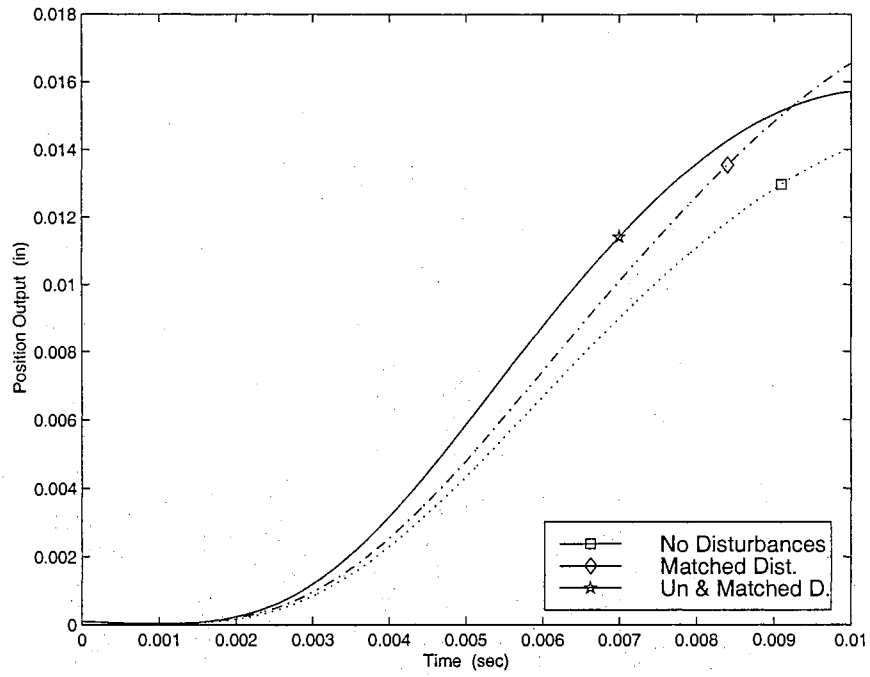


Figure 4.9: Actuator Position Responses

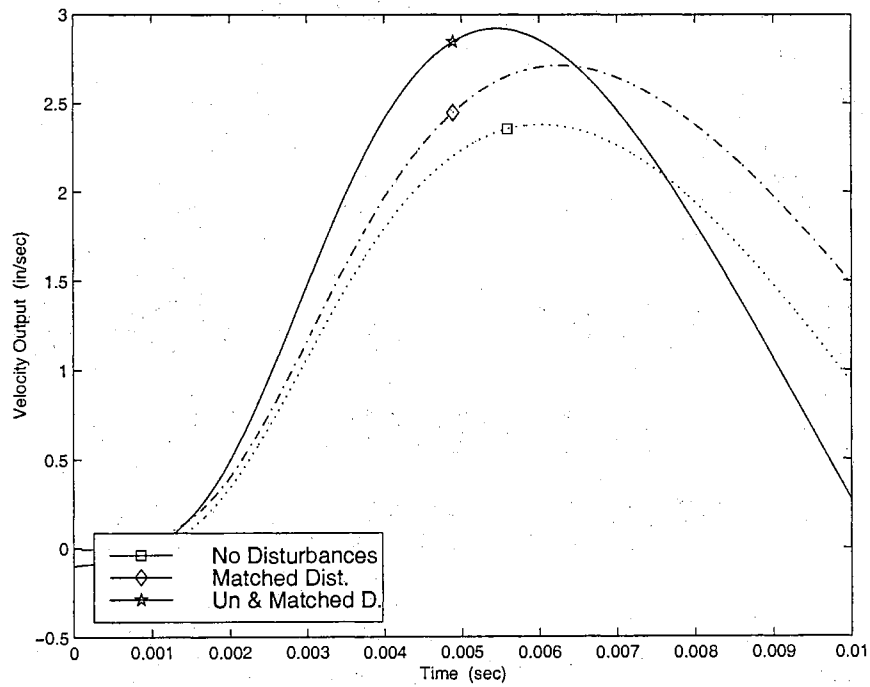


Figure 4.10: Actuator Velocity Responses

MATLAB calculates the system model parameters A_s, B_s to be

$$A_s = \begin{bmatrix} 1.0000e-00 & 3.4999e-04 & 5.0996e-07 & 5.8734e-11 & 6.8421e-09 \\ -1.1666e-04 & 9.9997e-01 & 2.9114e-03 & 5.0104e-07 & 7.7975e-05 \\ 0 & 0 & 9.9468e-01 & 3.4025e-04 & 7.9823e-02 \\ 0 & 0 & -3.0115e+01 & 9.4305e-01 & 4.5172e+02 \\ 0 & 0 & 0 & 0 & 1.0000e+00 \end{bmatrix}$$

$$B_s = \begin{bmatrix} 6.8421e-09 \\ 7.7975e-05 \\ 7.9823e-02 \\ 4.5172e+02 \\ 0 \end{bmatrix} \quad (4.8)$$

and A_f and B_f to be

$$A_f = \begin{bmatrix} 1.0000e-00 & 7.0000e-05 & 2.0416e-08 & 4.7512e-13 & 1.1044e-11 \\ -2.3333e-05 & 9.9999e-01 & 5.8329e-04 & 2.0344e-08 & 6.3076e-07 \\ 0 & 0 & 9.9978e-01 & 6.9625e-05 & 3.2410e-03 \\ 0 & 0 & -6.1622e+00 & 9.8922e-01 & 9.2433e+01 \\ 0 & 0 & 0 & 0 & 1.0000e+00 \end{bmatrix}$$

$$B_f = \begin{bmatrix} 1.1044e-11 \\ 6.3076e-07 \\ 3.2410e-03 \\ 9.2433e+01 \\ 0 \end{bmatrix} \quad (4.9)$$

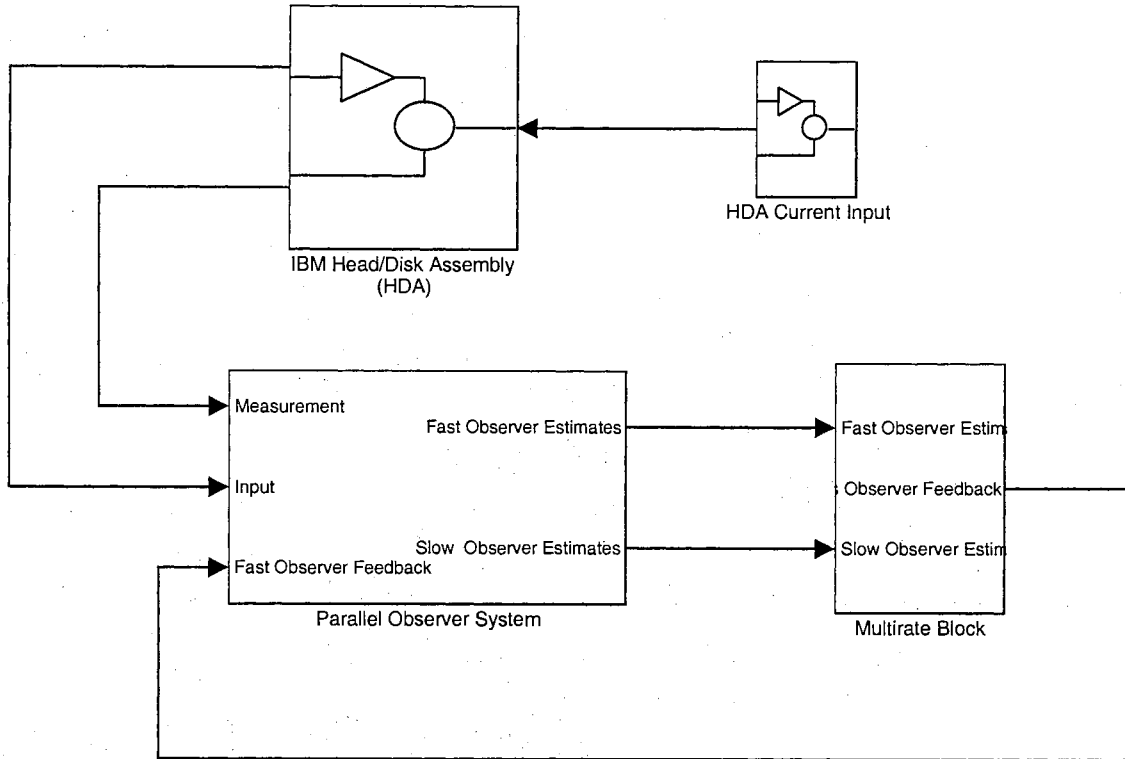


Figure 4.11: The Parallel Observer System and HDA System

4.3 The Implementation of the Parallel Observer System to the IBM HDA

4.3.1 The Parallel Observer System Model

The Parallel Observer System is applied, as in Equations (3.26) and (3.30). The general SIMULINK implementation of the POS in the simulations are shown in Figures 4.11 and 4.12. It should be noted that all observers implemented in this chapter are applied within the same simulation to insure that all observers share identical simulation conditions.

Using the parameters calculated in Equation (4.8), the Slow Observer System and the Fast Observer System are obtained, as in Equations (3.26) and (3.30). The Slow Observer System gains are calculated so that the observer poles are located arbitrarily close to the origin. The Fast Observer System observer gains are calculated using the MATLAB command DLQE.m [41] to obtain

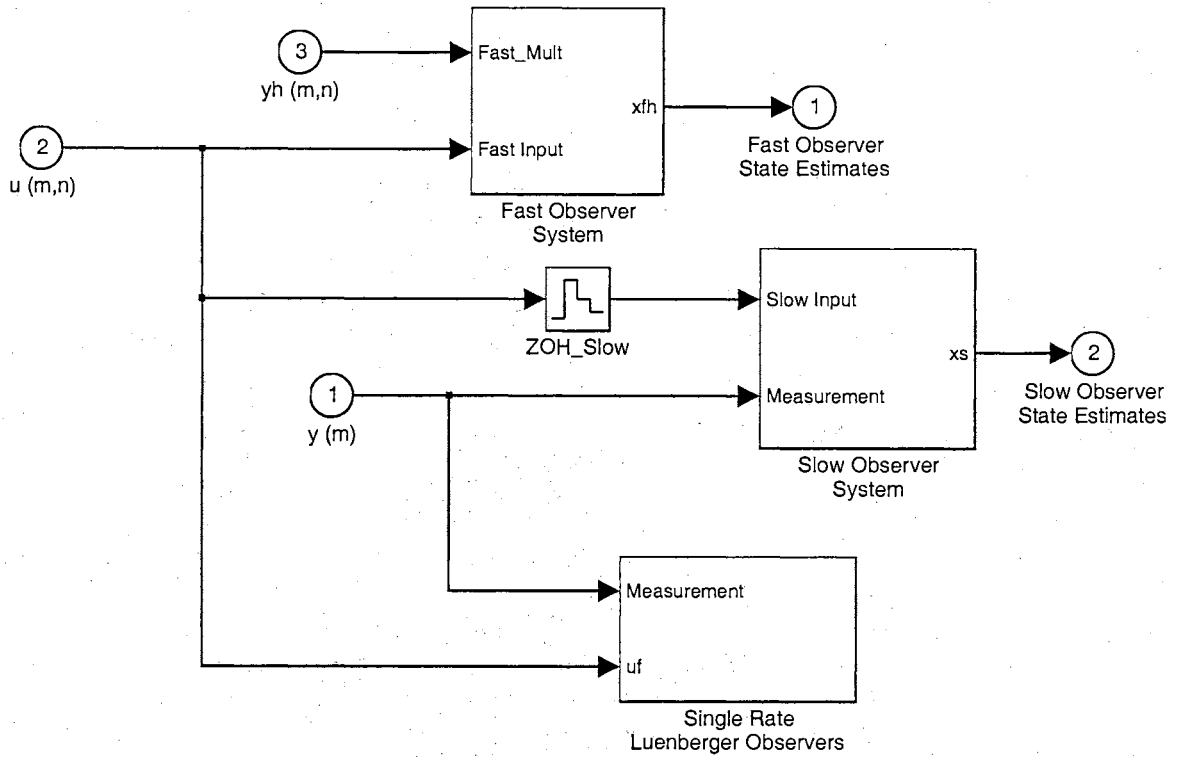


Figure 4.12: The Parallel Observer System and Comparison Observers

appropriate Kalman estimator gains. The DLQE.m command calculates the necessary parameters, including the Kalman gains, for the design of a discrete Kalman filter using the knowledge of the covariances of the discrete system's unbiased process and measurement noises. These calculated gains are guaranteed to be stable. In this case, no noise is introduced. Therefore, to tune the Kalman gains, the norm of the covariance matrix of the measurement noise is made very small. In addition, the coefficient matrix of the process noise is an identity matrix of the augmented state order. The norm of the covariance matrix of the process noise is increased accordingly for acceptable estimation results. The resulting gains for the Parallel Observer System are as follows:

$$\begin{bmatrix} L_s \\ l_{sw} \end{bmatrix} = \begin{bmatrix} 4.4437e + 00 \\ 1.5509e + 04 \\ 4.5863e + 06 \\ 7.4154e + 09 \\ 3.8496e + 06 \end{bmatrix} \quad (4.10)$$

and

$$\begin{bmatrix} L_f & L_{fw} \\ l_f & l_{fw} \end{bmatrix} = \begin{bmatrix} 1.0000e-00 & 7.0000e-05 & 2.0416e-08 & 4.7512e-13 & 1.1044e-11 \\ -2.3333e-05 & 9.9999e-01 & 5.8329e-04 & 2.0344e-08 & 6.3076e-07 \\ 5.4581e-26 & 3.7298e-22 & 9.9978e-01 & 6.9625e-05 & 3.2410e-03 \\ -1.1295e-24 & -4.4998e-21 & -6.1622e+00 & 9.8922e-01 & 9.2433e+01 \\ -7.7850e-27 & 4.1891e-25 & 2.0745e-21 & 5.9157e-17 & 1.0000e-00 \end{bmatrix}$$

The initial conditions for the Slow Observer System and the Fast Observer System are

$$\begin{bmatrix} \hat{x}_s(0) \\ \hat{w}_s(0) \end{bmatrix} = \begin{bmatrix} \hat{x}_f(0,0) \\ \hat{w}_f(0,0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.11)$$

while those for the actual system are set to

$$x(0,0) = \begin{bmatrix} 0.0001 \\ 0.1000 \\ 5.0000 \\ 2000.0 \end{bmatrix} \quad (4.12)$$

4.3.2 The Parallel Observer System Simulation Results

Simulations of the applied Parallel Observer System on the IBM magnetic Head/Disk Assembly are performed using MATLAB/SIMULINK. The comparison between the actual actuator position and the position estimates of the Parallel Observer System's Slow Observer System (SOS) and Fast Observer System (FOS) are shown in Figure 4.13. Those for the actual velocity and velocity estimates are shown in Figure 4.14.

Referring to Figure 4.13, it can be seen that both the Slow Observer System and Fast Observer System of the POS, after the initial transient, estimate the actuator position rather well, despite the added matched and unmatched uncertainties. The Fast Observer estimate tends to be very sensitive

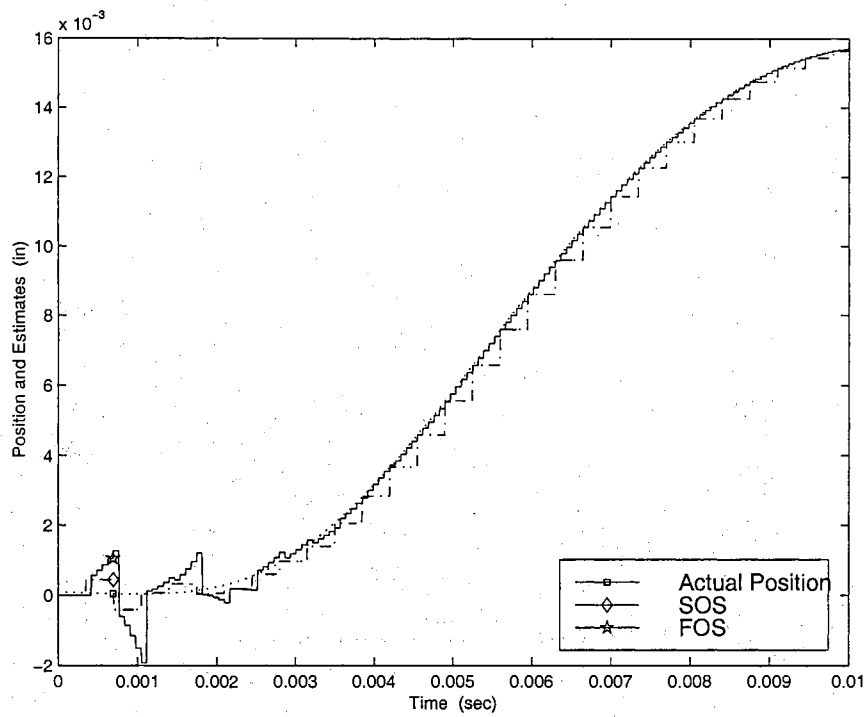


Figure 4.13: Actuator Position Output and POS Position Estimates

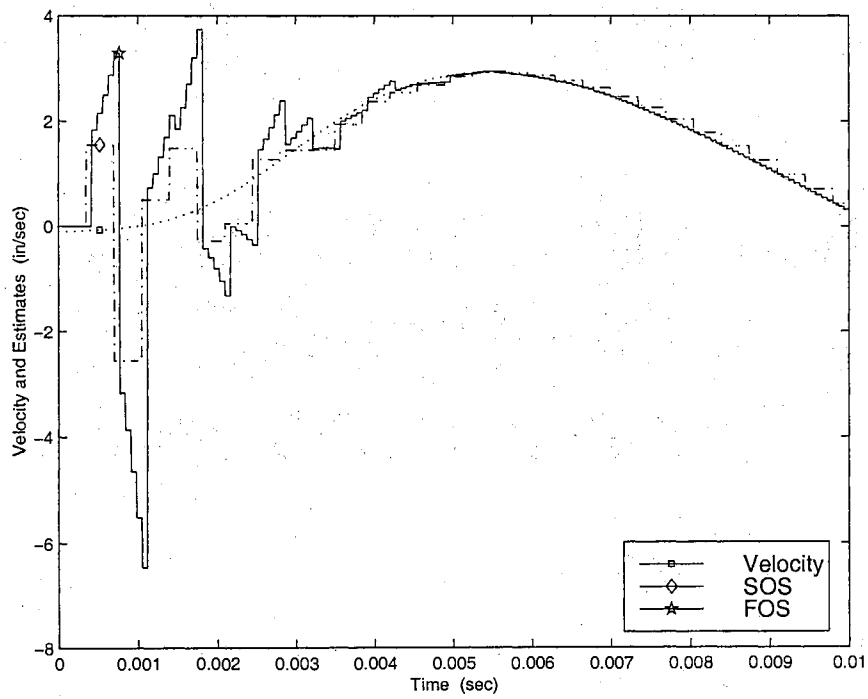


Figure 4.14: Actuator Velocity and POS Velocity Estimates

to the accuracy of the Slow Observer estimates, as can be seen by the marked transient oscillations at the points where the Slow Observer is still attempting to converge to the actual position signal.

The same behavior is noticed in Figure 4.14. The initial transient behavior before the actual velocity estimate converges to the actual signal is significantly greater and persists for a longer amount of time, most likely because the Slow Observer System estimates do not converge as quickly to the actual velocity values. However, as in the case of the position estimates, once convergence is obtained, estimation of velocity for both the Slow Observer and the Fast Observer remain highly accurate.

It should be noted that the Parallel Observer System gains are tuned for accuracy in position estimates, as opposed to any of the other state estimates, since, in a real application, the observer-based controller is driven mostly, if not entirely, by the position estimates obtained by the observer. Because of this tuning, the velocity estimation is not necessarily as accurate as it can possibly be.

4.4 Comparison Luenberger Observers

As mentioned previously, to have a basis for comparison, two separate, single rate observers are applied to the IBM HDA. Both of these observers are of the standard Luenberger form but run at different rates. The first single rate observer is applied at the measurement output sampling rate. It is referred to as the Slow Luenberger Observer. The second single rate observer is applied at the control input rate and is referred to as the Fast Luenberger Observer.

4.4.1 The Luenberger Observer Models

The Slow Luenberger Observer is given as follows:

$$\begin{aligned}
 \hat{X}_{slo}(m+1, n) &= \begin{bmatrix} \hat{x}_{slo}(m+1) \\ \hat{w}_{slo}(m+1) \end{bmatrix} \\
 &= \begin{bmatrix} A_s & B_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{slo}(m) \\ \hat{w}_{slo}(m) \end{bmatrix} + \begin{bmatrix} B_s \\ 0 \end{bmatrix} u(m) \\
 &\quad + \begin{bmatrix} L_{slo} \\ l_{slo} \end{bmatrix} [y(m) - \hat{y}_{slo}(m)] \\
 \hat{Y}_{slo}(m) &= \begin{bmatrix} C & 0 \end{bmatrix} \hat{X}_{slo}(m)
 \end{aligned} \tag{4.13}$$

where A_s and B_s are the same as given in Equation (4.8). One should note that this is the exact form of the Slow Observer System, as both are single rate observers running at the same sampling period. In so doing, it is only logical to use the same observer gains from the Slow Observer System on the Slow Luenberger Observer. In other words:

$$\begin{aligned}
 L_{slo} &= L_s \\
 l_{slo} &= l_{sw}
 \end{aligned} \tag{4.14}$$

Also, the Slow Luenberger Observer is held constant during the INTER sample points. The Fast Luenberger Observer is given as follows:

$$\begin{aligned}
 \hat{X}_{fio}(m, n+1) &= \begin{bmatrix} \hat{x}_{fio}(m, n+1) \\ \hat{w}_{fio}(m, n+1) \end{bmatrix} \\
 &= \begin{bmatrix} \hat{A}_f & \hat{B}_f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{fio}(m, n) \\ \hat{w}_{fio}(m, n) \end{bmatrix} + \begin{bmatrix} \hat{B}_f \\ 0 \end{bmatrix} u(m, n) \\
 &\quad + \begin{bmatrix} L_{fio} \\ l_{fio} \end{bmatrix} [y_f(m, n) - \hat{y}_{fio}(m, n)] \\
 \hat{Y}_{fio}(m, n) &= \begin{bmatrix} C & 0 \end{bmatrix} \hat{X}_{fio}(m, n)
 \end{aligned} \tag{4.15}$$

where A_f and B_f are the same as given in Equation (4.9). The observer gains of the Fast Luenberger Observer are calculated using the same observer poles obtained from the Parallel Observer System's Fast Observer gains. These Fast Luenberger Observer gains are as follows:

$$\begin{bmatrix} L_{flo} \\ l_{flo} \end{bmatrix} = \begin{bmatrix} 4.4950e + 00 \\ 7.7787e + 04 \\ 1.1466e + 08 \\ 8.6198e + 11 \\ 1.8829e + 09 \end{bmatrix} \quad (4.16)$$

Again, one should note that this Fast Luenberger Observer form is the exact form of the Slow Observer System except that it is running at the faster sampling period and, more importantly, that it is given the knowledge of the output actuator position during each ON *and* INTER sample point.

The reason for choosing these single rate Luenberger Observers for comparison is to show the worst and best case scenarios. For the worst case, as in the Slow Luenberger Observer, an output measurement is used every ON sample point to update the observer, and then the observer is forced to hold its estimates constant during the INTER sample points until the next output measurement is available. In contrast, the best case scenario is when the output measurement is available during every control sample point and no multirate estimation is needed, as in the case of the Fast Luenberger Observer.

4.4.2 The Comparative Luenberger Observers Simulation Results

Comparing the results of the Parallel Observer System's Fast Observer System position estimate errors to those position estimate errors of the single rate Slow Luenberger Observer (SLO) and Fast Luenberger Observer (FLO) in Figure 4.15, one can see that, besides the initial transient, the POS is much more accurate than the Slow Luenberger Observer, as it accounts for INTER sample state estimation. Also, the errors of the POS estimate is highly affected by that of the Slow Observer System's estimate accuracy.

Although the Fast Luenberger Observer is still considerably faster than the POS, the POS

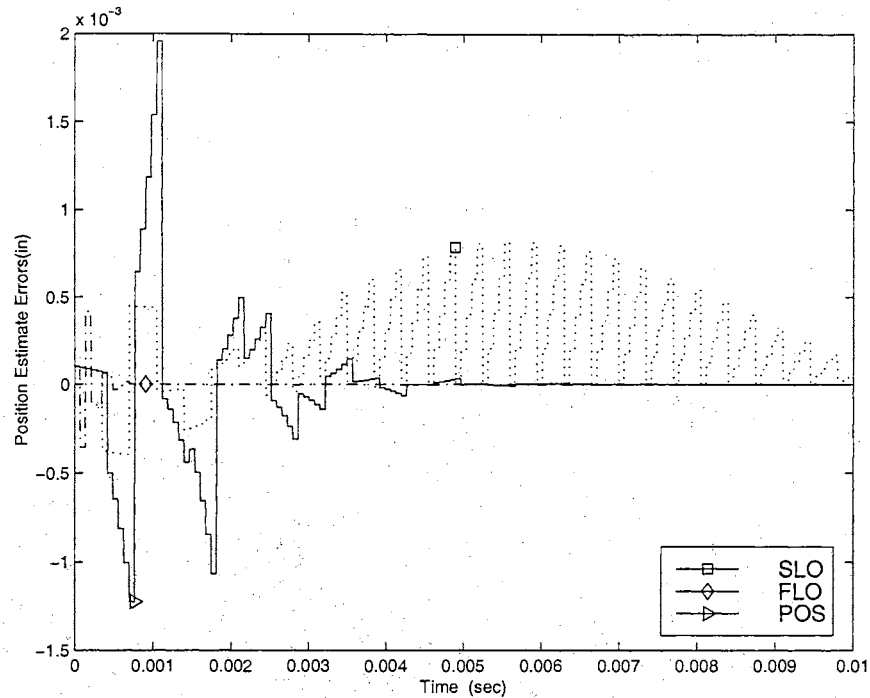


Figure 4.15: POS and Comparison Observers Position Estimate Errors

position error signal eventually decreases to zero just as in the best case situation of the Fast Luenberger Observer. The extremely quick convergence of the Fast Luenberger Observer state estimates is not surprising because the Fast Luenberger Observer has the added unfair advantage of having output measurement feedback during all sets of sample points.

The same trend is seen in Figure 4.16, where the velocity estimates of the Parallel Observer System and the single rate Slow Luenberger Observer and Fast Luenberger Observer are shown. The POS velocity estimates converge to actual velocity values relatively quickly, while the estimate errors of the Slow Luenberger Observer are still forced to oscillate during the INTER sample points. In addition, the velocity error signal of the POS has a much less amplitude of oscillation in its initial transient than even the best case Fast Luenberger Observer.

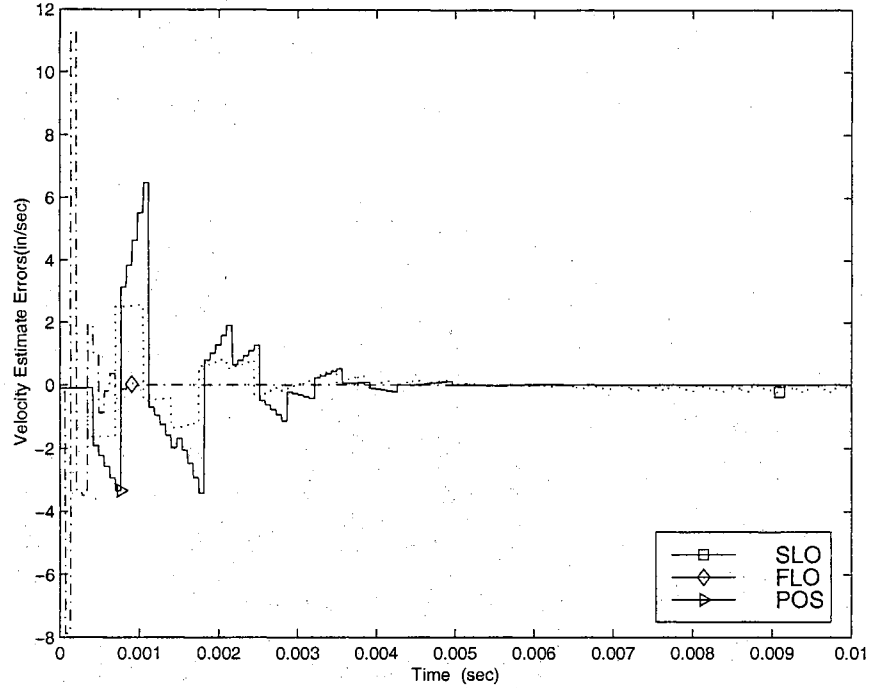


Figure 4.16: POS and Comparison Observers Velocity Estimate Errors

4.5 The Present Observer System (PRES)

In this section, the performance effects of aliased resonant frequencies and vibration on the Parallel Observer System compared to that of the industry standard estimation technique, referred to in this work as the Present Observer System (PRES), is analyzed. With this in mind, the emphasis of analysis is on the POS and the PRES when the resonant frequency is higher than that of half the measurement sampling frequency and when the system oscillates at this resonant frequency.

4.5.1 The Present Observer System (PRES) Model

The Present Observer System (PRES) which is used in today's standard industry disk drive is of the following form:

$$\begin{aligned}
 \hat{x}_{pres}(m, n+1) &= A_f \hat{x}_f(m, n) + B_f u(m, n) + L_{pres} [y_{pres}(m, n) - \hat{y}_{pres}(m, n)] \\
 \hat{y}_{pres}(m, n) &= C \hat{x}_{pres}(m, n)
 \end{aligned} \tag{4.17}$$

where

$$y_{pres}(m, n) = \begin{cases} y(m), & \text{if } n = 0 \\ \hat{y}_{pres}(m, n), & \text{if } n = 1, 2, \dots, k - 1 \end{cases} \quad (4.18)$$

Here, L_p is a Luenberger gain. The state estimates \hat{x}_{pres} are calculated every sample point and are updated during the ON sample points, as with the Fast Observer System of the POS, but the PRES uses the actual measurement output for feedback. Because both the Fast Observer System and PRES are running at the same sampling period and instances, it follows that the system parameters used, namely A_f and B_f are used in the PRES. This Present Observer is very similar to that of the Fast Observer System, except that the only feedback available is the single measurement output, as opposed to the full state estimate.

For purposes of comparison to the Parallel Observer System, the PRES is also augmented to allow for matched uncertainty estimation. That is,

$$\begin{aligned} \hat{X}_{pres}(m, n + 1) &= \begin{bmatrix} \hat{x}_{pres}(m, n + 1) \\ \hat{w}_{pres}(m, n + 1) \end{bmatrix} \\ &= \begin{bmatrix} A_f & B_f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{pres}(m, n) \\ \hat{w}_{pres}(m, n) \end{bmatrix} + \begin{bmatrix} B_f \\ 0 \end{bmatrix} u(m, n) \\ &\quad + \begin{bmatrix} L_{pres} \\ l_{pres} \end{bmatrix} \begin{bmatrix} y_{pres}(m, n) - \hat{y}_{pres}(m, n) \end{bmatrix} \end{aligned} \quad (4.19)$$

where

$$y_{pres}(m, n) = \begin{cases} y(m), & \text{if } n = 0 \\ \hat{x}_{pres}(m, n), & \text{if } n = 1, 2, \dots, k - 1 \end{cases} \quad (4.20)$$

Here, $\hat{w}_{pres}(m, n)$ is the Present Observer System estimation of the matched disturbance, and L_{pres} and l_{pres} are Luenberger gains.

4.5.2 The HDA Test Simulation

The scheme of the simulation system with the implemented Parallel Observer System and the Present Observer System is shown Figure 4.17. For these set of tests, the input to the system is a test signal

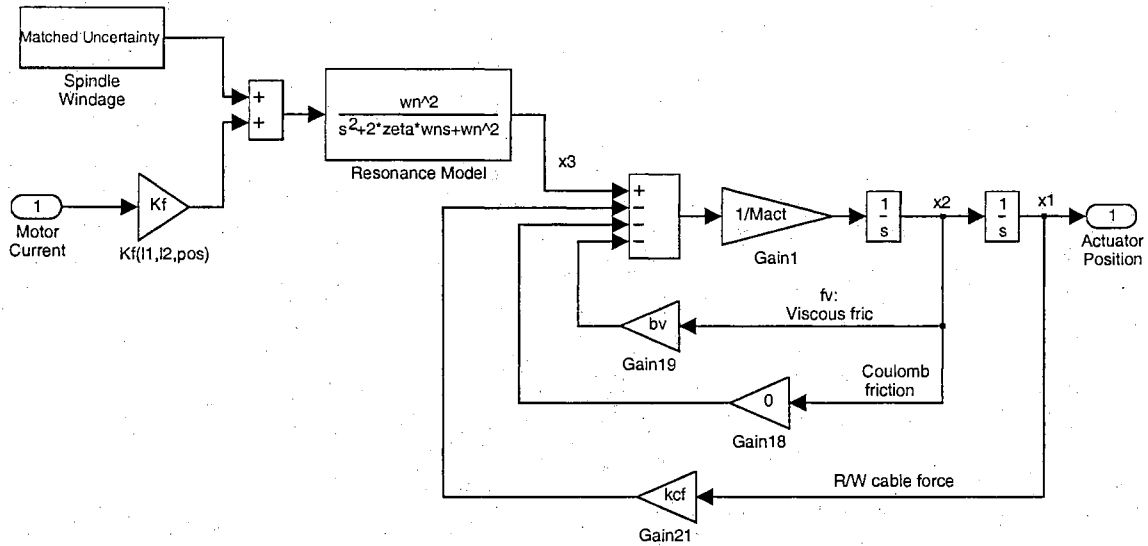


Figure 4.17: Simulation Model of the Fourth Order HDA Actuator System for the POS and PRES comprised of a series of step inputs. Furthermore, the system is continuously excited by an input sine wave which mimics a constant vibration input at the system resonant frequency. The resulting effective input is shown in Figure 4.18.

Two cases of simulations are presented. In the first case, there are no added matched or unmatched uncertainties. The sampling rate is the same as in the past simulations, that is a Nyquist frequency of $8.9760 \times 10^3 \text{ rad/s}$ ($1.4286 \times 10^3 \text{ Hz}$). The natural frequency is chosen to be aliased at a frequency of $1.5745 \times 10^4 \text{ rad/s}$ ($2.5059 \times 10^3 \text{ Hz}$). The damping ratio is set to 0.0995. In the second case, matched and unmatched uncertainties are added to the system. For both cases, the observer gains of the Present Observer are calculated using the closest available poles obtained from the Parallel Observer System's Fast Observer gains.

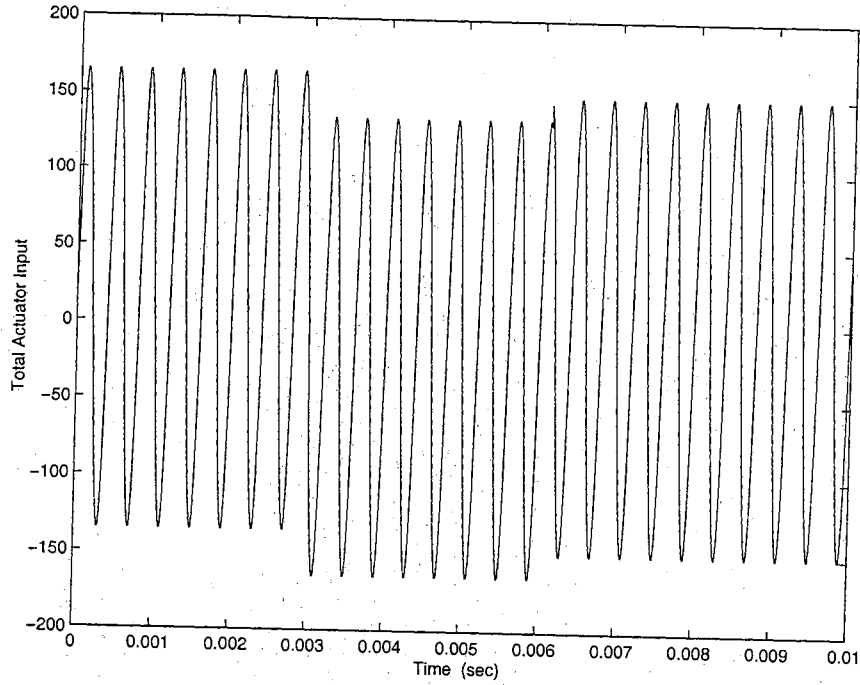


Figure 4.18: Total Input into Fourth Order HDA Actuator Model

Case I: Perfect Model

In the first case, a perfect model is used and no matched or unmatched uncertainties are introduced into the system.

$$A_s = \begin{bmatrix} 1.0000e-00 & 3.4999e-04 & 6.0269e-08 & 1.2383e-11 & 6.7521e-06 \\ -1.1666e-04 & 9.9997e-01 & -1.5320e-04 & 2.1469e-08 & 4.6047e-02 \\ 0 & 0 & 3.6128e-01 & -2.6456e-05 & 9.5807e+00 \\ 0 & 0 & 6.5586e+03 & 4.4418e-01 & -9.8380e+04 \\ 0 & 0 & 0 & 0 & 1.0000e+00 \end{bmatrix}$$

$$B_s = \begin{bmatrix} 6.7521e-06 \\ 4.6047e-02 \\ 9.5807e+00 \\ -9.8380e+04 \\ 0 \end{bmatrix} \quad (4.21)$$

and A_f and B_f are used for the Fast Observer and Present Observer parameters and are

$$A_f = \begin{bmatrix} 1.0000e-00 & 7.0000e-05 & 1.8514e-08 & 4.2522e-13 & 2.8535e-08 \\ -2.3333e-05 & 9.9999e-01 & 4.7792e-04 & 1.7182e-08 & 1.5812e-03 \\ 0 & 0 & 4.8886e-01 & 5.0890e-05 & 7.6671e+00 \\ 0 & 0 & -1.2616e+04 & 3.2941e-01 & 1.8924e+05 \\ 0 & 0 & 0 & 0 & 1.0000e+00 \end{bmatrix}$$

$$B_f = \begin{bmatrix} 2.8535e-08 \\ 1.5812e-03 \\ 7.6671e+00 \\ 1.8924e+05 \\ 0 \end{bmatrix} \quad (4.22)$$

The resulting Present Observer gains for this first case are

$$\begin{bmatrix} L_{pres} \\ l_{pres} \end{bmatrix} = \begin{bmatrix} 2.9915e+00 \\ 2.7261e+04 \\ -1.3696e+07 \\ -4.5555e+11 \\ 2.4012e+01 \end{bmatrix} \quad (4.23)$$

The gains for the Parallel Observer System are determined as explained earlier in this section and are as follows:

$$\begin{bmatrix} L_s \\ l_{sw} \end{bmatrix} = \begin{bmatrix} 2.9915e+00 \\ 2.7261e+04 \\ -1.3696e+07 \\ -4.5555e+11 \\ 2.4012e+01 \end{bmatrix} \quad (4.24)$$

$$\begin{bmatrix} L_f & L_{fw} \\ l_f & l_{fw} \end{bmatrix} = \begin{bmatrix} 1.0000e-00 & 7.0000e-05 & 1.8514e-08 & 4.2522e-13 & 2.8535e-08 \\ -2.3333e-05 & 9.9999e-01 & 4.7792e-04 & 1.7182e-08 & 1.5812e-03 \\ 1.6439e-26 & 7.6468e-23 & 4.8886e-01 & 5.0890e-05 & 7.6671e+00 \\ 3.9321e-22 & 5.2086e-19 & -1.2616e+04 & 3.2941e-01 & 1.8924e+05 \\ 2.0640e-27 & 3.7369e-24 & 1.7198e-20 & 4.1165e-16 & 1.0000e-00 \end{bmatrix}$$

Case II: Added Matched and Unmatched Uncertainties

In this second case, matched and unmatched uncertainties are present in the system. The matched uncertainty component is a spindle windage disturbance, a disturbance associated with the interaction of air across the spindle, and is a sine wave input with a magnitude of 30 and a frequency of 185 Hz. The unmatched uncertainties are implemented by using a resonance model where the actual resonant frequency and damping ratio are 0.94 and 1.13 times the actual natural frequency and damping ratio values, respectively.

4.5.3 The Present Observer System (PRES) Simulation Results

Figure 4.19 show the results of the POS and the PRES position estimates under the first case of simulations where there is no matched or unmatched uncertainties. It can be seen that the PRES position estimates are more accurate than the POS position estimates. Both observers are able to detect the forced oscillations at the aliased resonant frequency, but the PRES is able to capture the oscillations more accurately, although neither are truly accurate in its depiction of the forced oscillations. These results are seen more clearly in Figure 4.20 where the position estimate errors are shown. Both observer's estimate error oscillate about zero, with those of the PRES having smaller amplitudes of oscillation.

The reason for the POS's poorer performance may stem from the fact that the Fast Observer is dependent upon the accuracy of the Slow Observer's estimates. The aliased resonant frequency and the oscillation about this frequency could have a negative effect on the Slow Observer's estimates. In addition, the Fast Observer has an additional delay in feedback because it reacts and corrects its

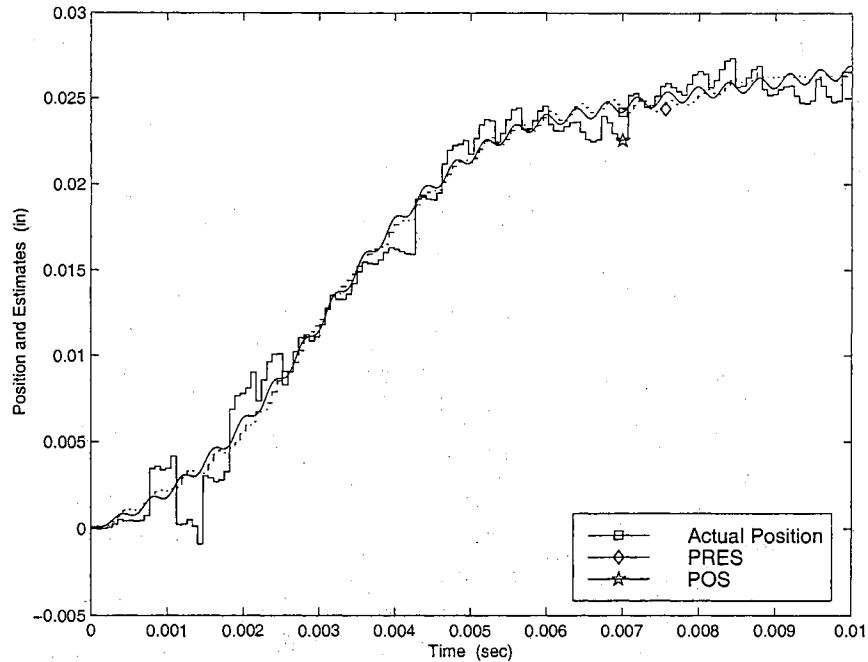


Figure 4.19: POS and PRES Position Estimates

measurements from the Slow Observer estimates only after the Slow Observer reacts and corrects its own estimates. The PRES does not have this “middle” stage in that it receives its feedback directly from the output measurement.

Figure 4.21 show the results of the POS and the PRES position estimates under the second case of simulations where there are added matched and unmatched uncertainties. As can be seen, the PRES position estimate shows signs of instability, while that of the POS position estimate do not seem to be affected by the presence of the added matched and unmatched uncertainties. The results of the simulation are seen more clearly in Figure 4.22 where the position estimate errors are shown. Here, again, both observer’s estimate error oscillate about zero, but whereas the PRES shows signs of instability, the estimate errors of the POS appear to be the same as without the uncertainties and remains stable.

The results of this latter case of simulations are not surprising, as the accuracy of the Present Observer is highly dependent upon the accuracy of the observer model. In addition, because the system is being excited at the resonant frequency and because the Present Observer shares similar dynamics, any significant deviation in position estimate could excite the dynamics of the Present

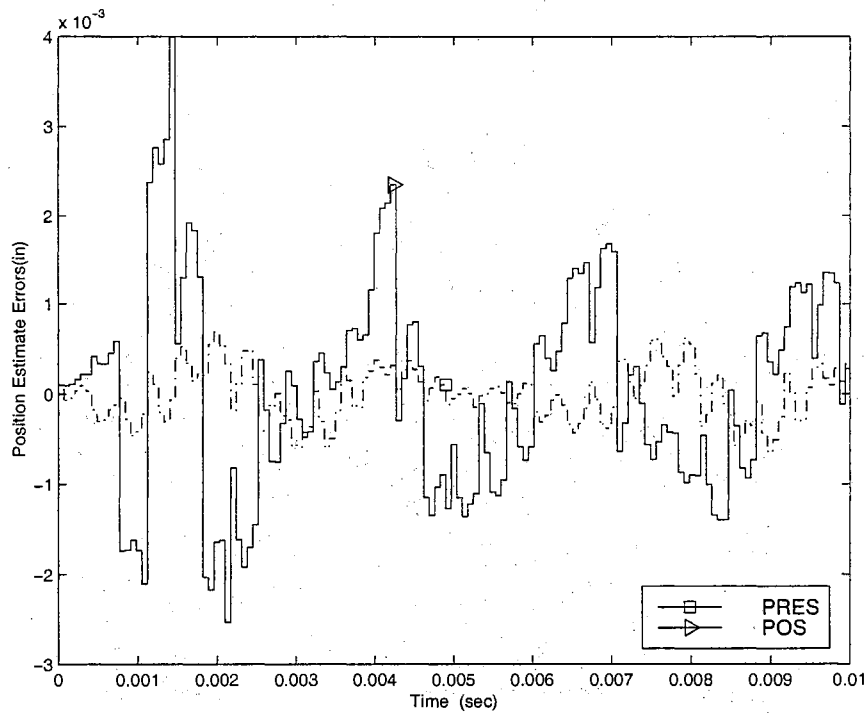


Figure 4.20: POS and PRES Position Estimate Errors

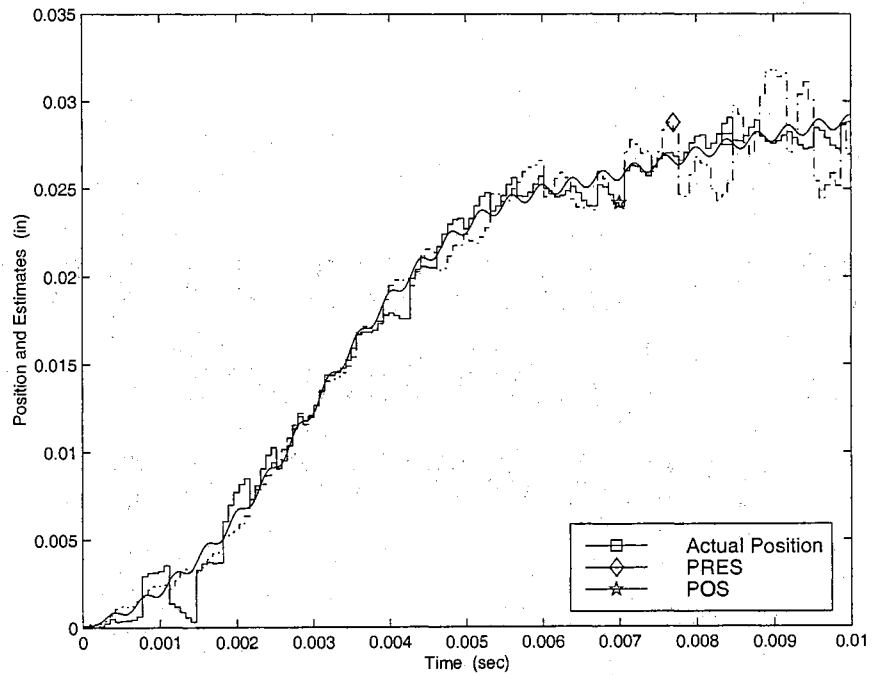


Figure 4.21: POS and PRES Position Estimates with Matched and Unmatched Uncertainties

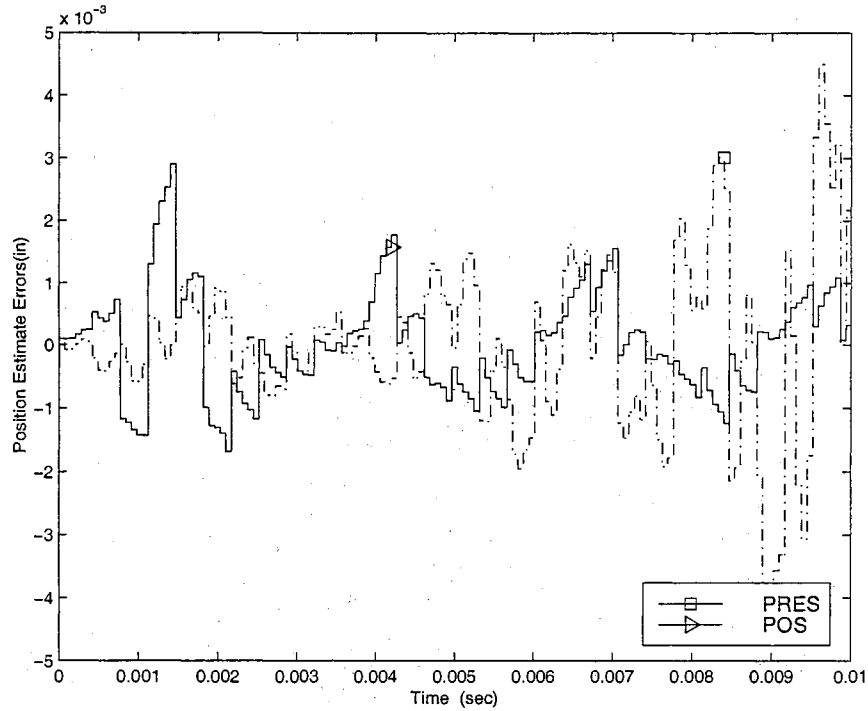


Figure 4.22: POS and PRES Position Estimate Errors with Matched and Unmatched Uncertainties

Observer System into system instability. Because of finite escape time, the INTER sample estimates of the PRES are bounded, but each deviation between estimate and measurement during the ON sample points effects the correction for the deviation. This situation with the system oscillation about the resonance frequency may be exciting the dynamics of the PRES, causing further deviations. As the deviations grow, so do the corrections, and instability eventually occurs:

This condition of instability is shown in Figures (4.23) and (4.24), where the simulation is rerun with the same conditions except that the estimated natural frequency is decreased slightly further to a value of 0.9 times the actual resonant frequency. Here, as the figure shows, the Present Observer quickly becomes unstable, while the Parallel Observer remains relatively unchanged.

The Parallel Observer System, on the other hand, has the advantage of being based upon a stable set of full state order estimates from the Slow Observer. The Fast Observer, although having the same observer model as the Present Observer System, uses these stable full state order estimates for feedback. Furthermore, as long as the gains of both the Slow and Fast Observer are chosen appropriately, the overall POS is designed to guarantee stability, regardless of added matched and

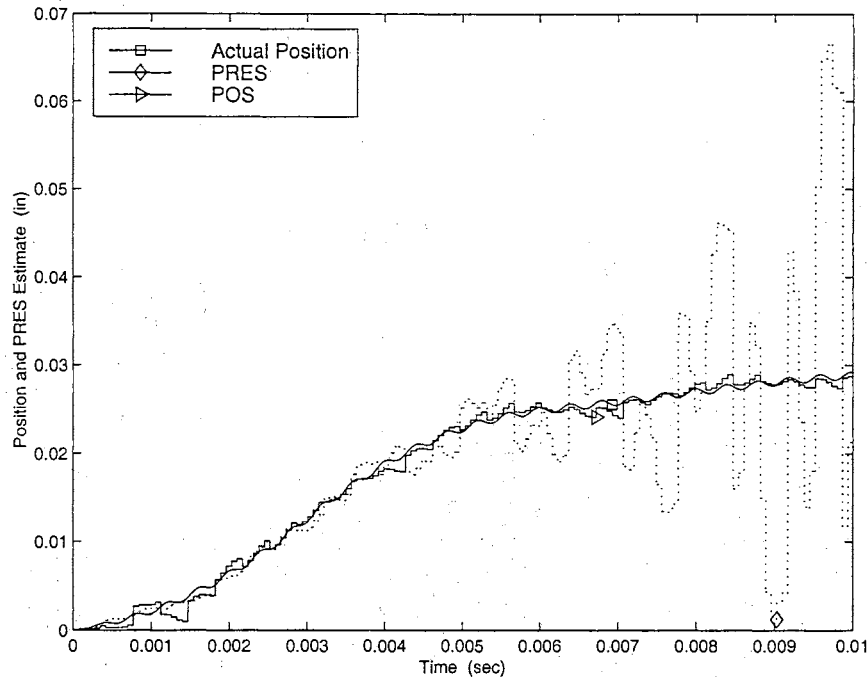


Figure 4.23: PRES and POS Position Estimates with Matched and Increased Unmatched Uncertainties

unmatched uncertainties in the system.

4.6 Summary

In short, the results of the simulations of the Parallel Observer System implemented on the IBM magnetic Head/Disk Assembly system are very promising. Despite added matched and unmatched uncertainties, in addition to different initial conditions, the Parallel Observer System is still able to produce accurate and convergent state estimates during the ON and INTER sample points.

The results of the POS's state estimation in this simulation shows superior performance to that of a single rate Slow Luenberger Observer, as the POS is able to produce accurate INTER sample estimates. The POS's state estimation is even comparable to a single rate Fast Luenberger Observer, despite the fact that the Fast Luenberger Observer is provided with output measurement feedback every ON and INTER sample point, while the POS has output measurements available only once out of every five sampling periods.

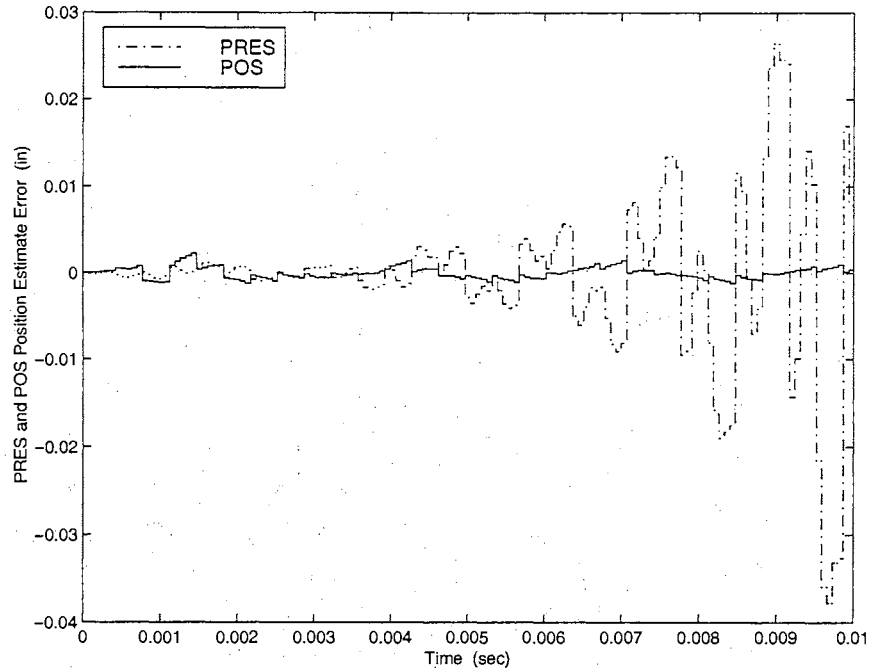


Figure 4.24: PRES and POS Position Estimate Error with Matched and Increased Unmatched Uncertainties

In addition, the implementation of the Present Observer to the IBM HDA system shows that the advantage to the POS under the conditions of system excitation at the aliased resonant frequency occurs in the presence of matched and unmatched uncertainties. The POS does not seem to be significantly affected by the presence of these uncertainties, whereas the Present Observer tends towards instability.

In short, the poles of the POS's Fast Observer System can be adjusted so that extremely fast convergence to the POS's Slow Observer's state estimates can be obtained. However, because the Fast Observer is designed to converge to the Slow Observer quickly, the Fast Observer is also highly affected by the accuracy of the estimates of the Slow Observer. In addition, the greater the speed of convergence of the Fast Observer to the Slow Observer, the greater the transient effects of the Slow Observer on the Fast Observer. This can be seen in the simulations where high amplitudes of oscillations occurs in the transients of the Fast Observer. It is noted, therefore, that the Slow Observer's state estimates must be accurate to insure the performance of the Fast Observer and the

overall Parallel Observer System.

Chapter 5

An Adaptive Parallel Observer System (APOS)

5.1 Motivation

The Parallel Observer System's Fast Observer is highly dependent upon the accuracy of the estimates of the Slow Observer, as seen in the example in the preceding chapter. Therefore, it is crucial to the success of the Parallel Observer System that the Slow Observer be made robust. Namely, if the system parameters change during the control process or if the model parameters do not match the actual system parameters, it would be very advantageous if the Slow Observer is made to account for these changing parameters or parameter uncertainties. Specifically, it would add greatly to the benefits and performance of the Parallel Observer System if the Slow Observer were modified to estimate the actual parameters, in addition to providing stable state and uncertainty estimates during the ON sample points. Specifically, it is desired that these parameter estimates aid in the estimation of multirate systems with output measurements which are only available at fixed intervals and which are aliased due to high resonance frequencies in the system model.

In this chapter, a Discrete Adaptive Observer System (DAO) is implemented in the Parallel Observer System, resulting in a new structure of the POS and is referred to as the Adaptive Parallel

Observer System (APOS). This new structure is very similar to that of the original POS in that two observers still are running parallel with each other. Instead of using a Luenberger Observer for both parallel observers, however, only the Fast Observer System is a Luenberger Observer. In this APOS, a Discrete Adaptive Observer System is used in place of the Slow Observer System. In this way, stable state estimates are provided by the Fast Observer at every control input sampling period, despite limited output measurements and any system parameter uncertainties or parameter changes that may occur during the control process. Furthermore, the DAO is also able to help estimate these uncertain and/or changing parameters.

5.2 Existing Discrete Adaptive Observer Techniques

The implementation of already existing discrete time adaptive observer techniques would improve the robustness of the Parallel Observer System by estimating system parameters or correcting inaccurate model parameter values.

The origins of adaptive estimation stem from the pioneer work by Carroll and Lindorff [42], which is intended for single-input single-output linear, continuous time systems. Many more works have followed. In fact, the Ei CompendexWeb [43] lists over 2500 references in adaptive estimation. The number of available discrete time adaptive observers, though, is far less. A couple of examples are those of Kudva and Narendra [44] and Suzuki and Andoh [45]. However, most of these available discrete adaptive observers, although all proven to be stable, have a slow rate of convergence.

Some methods which claim fast convergence are Shahrokhi and Morari [46] and Hong et al. [47]. These two techniques guarantee exponential convergence. However, they make use of several past measurements in each time step and require the inversion of a large matrix for each step. Because of the computational burden, these two methods are eliminated as possibilities. It is decided that the technique of Suzuki et al. [48], is to be implemented within the Parallel Observer System because of its smaller amount of required computation, its guaranteed stability, and its fast convergence properties.

5.2.1 The Proposed Discrete Adaptive Observer System (DAO)

As with most adaptive observers, whether continuous or discrete systems, the system to be observed is assumed to be in the proper observer canonical form. That is, the system should be written in the following form:

$$\begin{aligned} x(m+1) &= A_T x(m) + B_T u(m) \\ y(m) &= C_T x(m) \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} A_T &= \begin{bmatrix} -a_1 & 1 & 0 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & 0 & \cdots & 0 \\ -a_3 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ B_T &= [b_1 \ b_2 \ b_3 \ \cdots \ b_n]^T \\ C_T &= [1 \ 0 \ 0 \ 0 \ \cdots \ 0] \end{aligned} \tag{5.2}$$

According to Carroll and Lindorff [42], given a system

$$\begin{aligned} Z(m+1) &= A_z Z(m) + B_z u(m) \\ y(m) &= C_z Z(m) \end{aligned} \tag{5.3}$$

there exists a transformation matrix T such that $x_z = Tz$ and

$$\begin{aligned} A_T = T A_z T^{-1} &= \begin{bmatrix} -a_1 & 1 & 0 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & 0 & \cdots & 0 \\ -a_3 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ B_T &= T B_z \\ C_T = C_z T^{-1} &= [1 \ 0 \ 0 \ 0 \ \cdots \ 0] \end{aligned} \tag{5.4}$$

From Iwai et al. [49], the transformation matrix is given such that

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & 1 \end{bmatrix} \begin{bmatrix} C_z \\ C_z A_z \\ C_z [A_z]^2 \\ \vdots \\ C_z [A_z]^{n-1} \end{bmatrix} \quad (5.5)$$

The Discrete Adaptive Observer (DAO) as developed by Suzuki et al. assumes a single-input single-output linear plant. It is derived from an exponentially weighted least-squares method. The technique guarantees fast convergence of all state and parameter estimates and the asymptotic stability of the overall adaptive system, so long as the system input is sufficiently rich. The reader is referred to [48] for more details. The method is capable of estimating time-varying plant parameters quickly. The DAO scheme assumes the system of the following form:

$$\begin{aligned} x(m+1) &= A_T x(m) + B_T u(m) \\ y(m) &= C_T x(m) \end{aligned} \quad (5.6)$$

where

$$\begin{aligned} A_T &= \begin{bmatrix} a_1 & 1 & 0 & 0 & \cdots & 0 \\ a_2 & 0 & 1 & 0 & \cdots & 0 \\ a_3 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ B_T &= [b_1 \ b_2 \ b_3 \ \cdots \ b_n]^T \\ C_T &= [1 \ 0 \ 0 \ 0 \ \cdots \ 0] \end{aligned} \quad (5.7)$$

which is the same observer canonical output form as above. The system described in Equation (5.7) is re-written as

$$x(m+1) = Fx + p_1 v_1(m) + p_2 v_2(m), \quad x(0) = x_0 \quad (5.8)$$

where

$$F = \begin{bmatrix} f_1 & 1 & 0 & 0 & \cdots & 0 \\ f_2 & 0 & 1 & 0 & \cdots & 0 \\ f_3 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ f_n & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (5.9)$$

and such that F is a stable matrix. Vectors p_1 and p_2 are such that

$$p_{1i} = a_i - f_i, \quad i = 1, 2, 3, \dots, n \quad (5.10)$$

$$p_{2i} = b_i, \quad i = 1, 2, 3, \dots, n$$

$$p = [p_1^T \quad p_2^T]^T$$

and

$$v_1(m) = y(m) \quad (5.11)$$

$$v_2(m) = u(m)$$

Define the state variable filter vector $\phi(m)$ and the matrix $S(m)$, respectively, as

$$\phi_i(m+1) = F^T \phi_i(m) + C^T v_i(m), \quad i = 1, 2 \quad (5.12)$$

and

$$S_i = \begin{bmatrix} C \\ CF \\ CF^2 \\ \vdots \\ CF^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} \phi_i(m)^T \\ \phi_i(m)^T F \\ \phi_i(m)^T F^2 \\ \vdots \\ \phi_i(m)^T F^{n-1} \end{bmatrix}, \quad i = 1, 2 \quad (5.13)$$

such that

$$\phi(m) = [\phi_1(m)^T \quad \phi_2(m)^T]^T \quad (5.14)$$

$$S(m) = [S_1(m) \quad S_2(m)]$$

the adaptive state estimation expressions are

$$\hat{x}(m) = S(k) \hat{p}(m) + F^m \hat{x}(0) \quad (5.15)$$

$$\hat{y}(m) = \phi(m)^T \hat{p}(m) + C F^m \hat{x}(0)$$

where $\hat{x}(m)$ is the state estimate vector and $\hat{y}(m)$ is the output estimate scalar. In addition, $\hat{p}(m)$ is a vector of size $2n$ which denotes the parameter estimates at time m and is defined as

$$\hat{p}_i(m) = \begin{cases} \hat{a}_i(m) - f_i, & \text{if } i = 1, 2, 3, \dots, n \\ \hat{b}_i(m), & \text{if } i = n + 1, n + 2, n + 3, \dots, 2n \end{cases}$$

Here, $\hat{a}_i(m)$ are the parameter estimates of the matrix A_T , and $\hat{b}_i(m)$ are the parameter estimates of the matrix B_T . This adaptive state estimation technique is guaranteed to be asymptotically stable, provided that the parameter estimates of $\hat{p}(m)$ converge to the actual parameter values. That is, the adaptive observer system is stable if $\hat{a} \rightarrow a$ and $\hat{b} \rightarrow b$.

The adaptive law, based upon a weighted least-squares method, guarantees parameter convergence and is defined by the following relation:

$$\hat{p}(m+1) = \hat{p}(m) + \Gamma(\lambda, m+1) \phi(m+1) [z(m+1) - \phi(m+1)^T \hat{p}(m)] \quad (5.16)$$

where

$$\begin{aligned} \Gamma(\lambda, m+1) &= \Gamma_\lambda - \frac{\Gamma_\lambda \phi(m+1) \phi(m+1)^T \Gamma_\lambda}{1 + \phi(m+1)^T \Gamma_\lambda \phi(m+1)} \\ \Gamma_\lambda &= \frac{\Gamma(\lambda, m)}{\lambda^2} \\ z(m) &= y(m) - C F^m \hat{x}(0) \end{aligned} \quad (5.17)$$

Here, λ is a weighting coefficient whose values may vary such that $0 < \lambda < 1$. Suzuki et al. note in their work that Γ may be initialized as:

$$\Gamma(\lambda, 0) = d^2 I_{2n}, \quad d \gg 1 \quad (5.18)$$

As explained earlier, the APOS replaces the original Slow Observer System with the Discrete Adaptive Observer System described in Equations (5.8) through (5.18). The DAO transformed A_s matrix, denoted by A_T is obtained by comparing coefficients of the characteristic polynomials of the

given A_s and the n parameters of A_T , which is in observer canonical form. Then, the transformation matrix T is found using Equation (5.5). In turn, the remaining DAO system parameters are calculated.

It also must be noted that any changes or uncertainties in the system are interpreted by the Discrete Adaptive Observer as parameter changes. In so doing, any external disturbances or measurement noise are to be mistaken for system parameter changes. Thus, the Discrete Adaptive Observer will adjust itself accordingly in this incorrect assumption. In addition, although output measurements, as well as input data, are used to drive the DAO, the DAO still remains acting in an open-loop fashion. That is, the DAO does not have any correction terms, as in the Slow Observer or the Fast Observer, to correct itself for estimation errors.

5.2.2 Stability Analysis of the Proposed Adaptive Parallel Observer System

Assumption 7. Any changes in the system are due to system parameter changes and not due to any other disturbances to the system.

The following terms are used in this chapter of development of stability proofs, somewhat similar to those defined in Equation (5.19) of Chapter 3:

$$\begin{aligned}
 e_f &= x(m, n) - \hat{x}_f(m, n) \\
 e_{wf} &= w(m, n) - \hat{w}_f(m, n) \\
 \epsilon_{sf}(m) &= \hat{x}_{DAO}(m) - \hat{x}_f(m, 0) \\
 \epsilon_{wsf}(m) &= \hat{w}_{DAO}(m) - \hat{w}_f(m, 0)
 \end{aligned} \tag{5.19}$$

where, in this case, $\hat{x}_{DAO}(m)$ is the state estimates of the Discrete Adaptive Observer. The proof of convergence for the Adaptive Parallel Observer Systems involve manipulations of Lemma (2) and the use of stability analysis based on eigenvalues. In addition, it also assumes the stability of the DAO as given in [48].

The Fast Observer from Equation (3.27) is

$$\begin{aligned}
\hat{x}_f(m, n+1) &= A_f \hat{x}_f(m, n) + B_f u_f(m, n) + L_f \epsilon_{sf}(m, n) + L_{fw} \epsilon_{wsf}(m, n) \\
\hat{w}_f(m, n+1) &= \hat{w}_f(m) + l_f \epsilon_{sf}(m, n) + l_{fw} \epsilon_{wsf}(m, n) \\
\hat{y}_f(m, n) &= C \hat{x}_f(m, n)
\end{aligned} \tag{5.20}$$

where

$$\begin{aligned}
\epsilon_{sf}(m, n) &= \begin{cases} \epsilon_{sf}(m), & \text{if } n = 0 \\ 0, & \text{if } n = 1, 2, \dots, k-1 \end{cases} \\
\epsilon_{wsf}(m, n) &= \begin{cases} \epsilon_{wsf}(m), & \text{if } n = 0 \\ 0, & \text{if } n = 1, 2, \dots, k-1 \end{cases}
\end{aligned} \tag{5.21}$$

Assumption 8. All of the necessary requirements for the stability of the Discrete Adaptive Observer, as described in [48] are satisfied.

Denote $\delta_{DAO}(m)$ as the sum of the unmatched uncertainties and perturbations from the feedback term comparing the Discrete Adaptive Observer and the Fast Observer system during the ON sample points. In addition, denote $\delta_f(m, n)$, as before as the unmatched uncertainties during the INTER sample points, the following assumption is made:

Assumption 9. The unmatched uncertainties $\delta_{DAO}(m)$ and $\delta_f(m, n)$ are bounded such that

$$\max_i |\delta_{DAO}(i)| \leq \delta_{DAO_0} \tag{5.22}$$

$$\max_j |\delta_f(m, j)| \leq \delta_{f_0} \tag{5.23}$$

where δ_{DAO_0} and δ_{f_0} are finite values.

Re-define the following parameters such that

$$\begin{aligned}
f_0 &= e_f(m, 0) \\
w_0 &= e_{wf}(m, 0) \\
f_{DAO_0} &= \epsilon_f(m, 0) \\
w_{DAO_0} &= \epsilon_{wf}(m, 0) \\
C_{fDAO} &= L_f f_{DAO_0} + L_{fw} w_{DAO_0} \\
C_{wDAO} &= l_f f_{DAO_0} + l_{fw} w_{DAO_0} \\
A_0 &= \|A_f\|_{i\infty} \\
B_0 &= \|B_f\|_{\infty} \\
F_0 &= \|f_0\|_{\infty} \\
C_{fDAO_0} &= \|C_{fDAO}\|_{i\infty} \\
C_{wDAO_0} &= |C_{wDAO}| \\
\omega_0 &= |w_0|
\end{aligned} \tag{5.24}$$

Theorem 2. *Let Assumptions 3 through 5, 8, and 9 hold. In addition, it is assumed that k , the number of INTER sample point between two consecutive measurements, is a finite integer and that the matrix F_f in Equation (3.40) is stable. Given the system in Equation (3.31), the proposed Discrete Adaptive Observer described in Equations (5.8) through (5.18), and the Fast Observer System described in Equations (3.27)-(3.28), then*

1. *The Fast Observer state and uncertainty estimates during the ON sample points are stable.*
2. *The Fast Observer state and uncertainty estimates are stable during the INTER sample points.*

Furthermore, at a general time step n , the state estimate error has a finite bound of

$$\begin{aligned}
\|e_f(m, n+1)\|_{\infty} &\leq A_0^{n+1} F_0 + A_0^n C_{fDAO_0} \\
&\quad + \frac{1 - A_0^{n+1}}{1 - A_0} [B_0(\omega_0 + C_{wDAO_0}) + \delta_{f_0}]
\end{aligned} \tag{5.25}$$

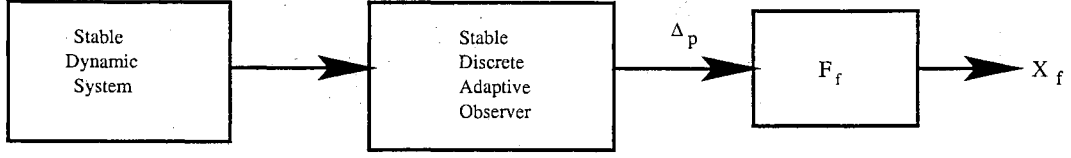


Figure 5.1: APOS ON Sample Stability

and the uncertainty estimation error has a finite bound of

$$|e_{wf}(m, n + 1)| \leq \omega_0 + C_{wDAO_0} \quad (5.26)$$

Proof: The proof for this theorem is very similar to that for the stability proof for the Parallel Observer System. First, stable estimates are shown for the ON sample points. The second part involves analyzing the error dynamics of the Fast Observer and calculating the propagated error during each of the INTER sample points.

Proof for Theorem 2: The ON Sample Points

On the ON sample points, the Fast Observer may be represented as

$$\begin{aligned}
 \begin{bmatrix} \hat{x}_f(m+1) \\ \hat{w}_f(m+1) \end{bmatrix} &= \begin{bmatrix} A_s & B_s \\ 0^{1 \times n} & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_f(m) \\ \hat{w}_f(m) \end{bmatrix} + \begin{bmatrix} B_s \\ 0 \end{bmatrix} u(m) \\
 &+ \begin{bmatrix} A_f^{k-1} L_f & A_f^{k-1} L_{fw} \\ l_f & l_{fw} \end{bmatrix} \begin{bmatrix} \varepsilon_{sf}(m) \\ \varepsilon_{wsf}(m) \end{bmatrix} \\
 &= \begin{bmatrix} A_s & B_s \\ 0^{1 \times n} & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_f(m) \\ \hat{w}_f(m) \end{bmatrix} + \begin{bmatrix} B_s \\ 0 \end{bmatrix} u(m) \\
 &+ \begin{bmatrix} A_f^{k-1} L_f & A_f^{k-1} L_{fw} \\ l_f & l_{fw} \end{bmatrix} \begin{bmatrix} \hat{x}_{DAO}(m) - \hat{x}_f(m) \\ \hat{w}_{DAO}(m) - \hat{w}_f(m) \end{bmatrix} \quad (5.27)
 \end{aligned}$$

Referring to Figure (5.1), one may rewrite Equation (5.27) simply as

$$\hat{X}_f(m+1) = F_f \hat{X}_f(m) + \Delta_p(m) \quad (5.28)$$

Using Assumption (9) and the fact that the system in Equation (3.18) is stable and so that the input to the system in Equation (3.18) is bounded, it can be inferred that Δ_p is also stable. Because of the stability of Δ_p , it can be seen that the overall dynamics described in Equation (5.28) is stable.

Proof for Theorem 2: The INTER Sample Points

The stability proof for the INTER sample points for the Adaptive Parallel Observer System is very similar to that for Theorem 1. Using the error definitions given in Equations (5.19) and the Equations (3.30) and (3.46), $e_f(m, n)$ and $e_{wf}(m, n)$ are such that the error dynamics for the Fast Observer System with uncertainty estimation are

$$\begin{aligned} e_f(m, n+1) &= A_f e_f(m, n) + B_f e_{wf}(m, n) - L_f \epsilon_{sf}(m, n) \\ &\quad - L_{fw} \epsilon_{wsf}(m, n) + \delta_f(m, n) \\ e_{wf}(m, n+1) &= e_{wf}(m, n) - l_f \epsilon_{sf}(m, n) - l_{fw} \epsilon_{wsf}(m, n) \end{aligned} \quad (5.29)$$

Using the expressions defined for Theorem (2), one may calculate the resulting propagated error as

$$\begin{aligned} e_f(m, n+1) &= A_f^{n+1} f_0 - A_f^n C_{fDAO} + \sum_{i=0}^n A_f^i B_f w_0 - \sum_{i=0}^n A_f^i B_f C_{wDAO} \\ &\quad - \sum_{i=0}^n A_f^i \delta_f(m, i) \end{aligned} \quad (5.30)$$

and

$$e_{wf}(m, n+1) = w_0 - C_{wDAO} \quad (5.31)$$

for the Fast Observer state estimates and uncertainty estimates, respectively. Utilizing Schwartz's Inequality and the triangle inequality, the error bound of Equation (5.30) is, most conservatively,

$$\begin{aligned} \|e_f(m, n+1)\|_\infty &\leq A_0^{n+1} F_0 + A_0^n C_{fDAO_0} + \sum_{i=0}^n A_0^i B_0 \omega_0 \\ &\quad + \sum_{i=0}^n A_0^i B_0 C_{wDAO_0} + \sum_{i=0}^n A_0^i \delta_{f_0} \end{aligned} \quad (5.32)$$

and

$$|e_{wf}(m, n+1)| \leq \omega_0 + C_{wDAO_0} \quad (5.33)$$

Using finite series analysis, one can simplify the second and third terms of Equation (5.33). For a stable system, $A_0 < 1$. Then,

$$\begin{aligned}\sum_{i=0}^n A_0^i B_0 \omega_0 &= B_0 \omega_0 \frac{1 - A_0^{n+1}}{1 - A_0} \\ \sum_{i=0}^n A_0^i B_0 C_{wDAO_0} &= B_0 C_{wDAO_0} \frac{1 - A_0^{n+1}}{1 - A_0} \\ \sum_{i=0}^n A_0^i \delta_{f_0} &= \delta_{f_0} \frac{1 - A_0^{n+1}}{1 - A_0}\end{aligned}\quad (5.34)$$

The resulting calculation of the error bound of the Fast Observer is

$$\|e_f(m, n + 1)\|_\infty \leq A_0^{n+1} F_0 + A_0^n F_0 + \frac{1 - A_0^{n+1}}{1 - A_0} [B_0 (\omega_0 + C_{wDAO_0}) + \delta_{f_0}] \quad (5.35)$$

Q.E.D

5.2.3 Maximum Error Bounds

Remark 8. As discussed previously, in the worst case scenario, the error in the state estimates for the Fast Observer System continue to increase during the INTER sample points until the start of the next cycle (i.e., the next ON point). Under this condition, then, the greatest maximum bound for any INTER sample point would occur at $n = k - 1$, or such that $(m, n + 1) = (m, k - 1)$ (at the time step just before the next ON point).

Corollary 3. *Given the results of Theorem 2 and the worst case scenario in which the maximum estimation error occurs at the time step of $t = (m, k - 1)$ (described in Remark 5), an upper bound of the error estimates of the Fast Observer state estimates $\hat{x}_f(m, k - 1)$ is*

$$\|e_f(m, k - 1)\|_\infty \leq A_0^k F_0 + A_0^{k-2} C_{fDAO_0} \quad (5.36)$$

$$+ \frac{1 - A_0^{k-1}}{1 - A_0} [B_0 (\omega_0 + C_{wDAO}) + \delta_{f_0}] \quad (5.37)$$

and the upper bound of the uncertainty estimation \hat{w}_f is

$$|e_{wf}(m, k - 1)| \leq \omega_0 + C_{wDAO} \quad (5.38)$$

Proof. The proof, here, is straightforward. Using the fact that, at the sample point prior to the following cycle period, as described in Remark (5), $n = k - 2$ in Equations (5.31) and (5.35). Substituting this relation into Equations (5.25) and (5.26), the result is the same as that described in Equations (5.37) and (3.68), respectively. \square

Remark 9. The reader should note, as before, that the above bounds may be extremely conservative and may not be of practical use except to show that the INTER sample estimate bound is, indeed, finite.

Corollary 4. *If there are no unmatched uncertainties in the system and the Slow and Fast Observers have the same initial conditions, the error bound is further reduced to*

$$\begin{aligned} \|e_f(m, k-1)\|_\infty &\leq \begin{cases} A_0^k F_0 + B_0 \omega_0 \frac{1-A_0^{k-1}}{1-A_0}, & \text{if } A_0 \neq 1 \\ F_0 + B_0(k-1)\omega_0, & \text{if } A_0 = 1 \end{cases} \\ |e_{wf}(m, k-1)| &\leq \omega_0 \end{aligned} \quad (5.39)$$

Proof. The proof, here, is again straightforward in that now, all terms except A_0 , B_0 , ω_0 , and F_0 are eliminated from Equations (5.37) and (5.38). \square

Remark 10. Furthermore, if the initial conditions of the actual states and uncertainties are the same as those for the Slow Observer and the Fast Observer, the parameters A_0 , B_0 , ω_0 , and F_0 all become 0. These initial conditions settings, of course, result in perfect state and uncertainty estimates. This can be easily seen, as all the remaining terms in Equation(5.39) are eliminated, leaving

$$\begin{aligned} \|e_f(m, k-1)\|_\infty &= 0 \\ |e_{wf}(m, k-1)| &= 0 \end{aligned} \quad (5.40)$$

Remark 11. The reader should note that the stability proof of the INTER sample points of the Fast Observer System does not change, regardless of whether the Discrete Adaptive Observer or the Slow Observer is implemented, so long as the observer being used is stable. This infers the

fact that the Parallel Observer System with the Fast Observer System may be used with any stable “feedback” observer, such as the Slow Observer System and the DAO.

An important note here is that the overall stability for the Adaptive Parallel Observer System is guaranteed only if the DAO is also stable, as inferred in Assumption 8. The DAO, however, does not necessarily guarantee that the augmented state (to facilitate matched uncertainty estimation) of the Discrete Adaptive Observer will remain stable. Therefore, if the augmented state form is, in fact, applied to the APOS, there is no guarantee that the Discrete Adaptive Observer will interpret the matched uncertainties as contributions to the matched uncertainty term $w(m)$ and that it will not interpret these specific uncertainties as parameter changes or mismatches in system parameters. Furthermore, because the DAO assumes that any changes in the system are a result of changing system parameters, the DAO is expected to be very susceptible to noise and other external disturbances.

Chapter 6

Implementation of an Adaptive Parallel Observer System to a Hard Disk Drive System

6.1 DAO Implementation Issues

In the set of simulations performed in this chapter, it is seen that the DAO may either be tuned specifically for the accuracy of the state estimates at the expense of oscillating parameter estimates, or it may be tuned for the smooth convergence of the parameter estimates but with slower state estimate convergence. For the purposes of the following simulations, a “middle” ground is attempted in that the state estimation is relatively accurate while preventing excessive oscillation of the parameter estimates.

6.2 Implementation of the DAO to a Second Order Resonance Model

To illustrate the potential benefits of the Adaptive Parallel Observer System, the same second order resonance model used in Chapter 4 is used in this section to analyze the effectiveness of the Discrete Adaptive Observer. In so doing, the results of the DAO are compared to that of the Slow Observer System. In addition, a comparison of the DAO parameter estimates to the actual estimates are made. Note that these set of simulations are of single rate at the output measurement sampling period.

6.2.1 The Resonance Order Model

The same second order resonance model $R(s)$ used in the HDA simulation in Chapter 4 is used for this example. That is, $R(s)$ is such that

$$R(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6.1)$$

where, again, the natural frequency ω_n and damping ratio ζ are, respectively, 1500rad/s and 0.10. The system is discretized at the measurement sample rate of $T_s = 350\mu\text{s}$ and the resulting state space matrices are

$$\begin{aligned} A &= \begin{bmatrix} 8.6990e-01 & 3.1720e-04 \\ -7.1370e+02 & 7.7473e-01 \end{bmatrix} \\ B &= \begin{bmatrix} 1.3010e-01 \\ 7.1370e+02 \end{bmatrix} \end{aligned} \quad (6.2)$$

6.2.2 The DAO System Model

The input into the system is composed of two sinusoidal inputs, shown in Figure 6.1, such that $u(t) = \sin(535t) + \sin(615t)$. There are no disturbances added to the system. Therefore, the Discrete Adaptive Observer does not have an augmented state form. That is, since there is no need

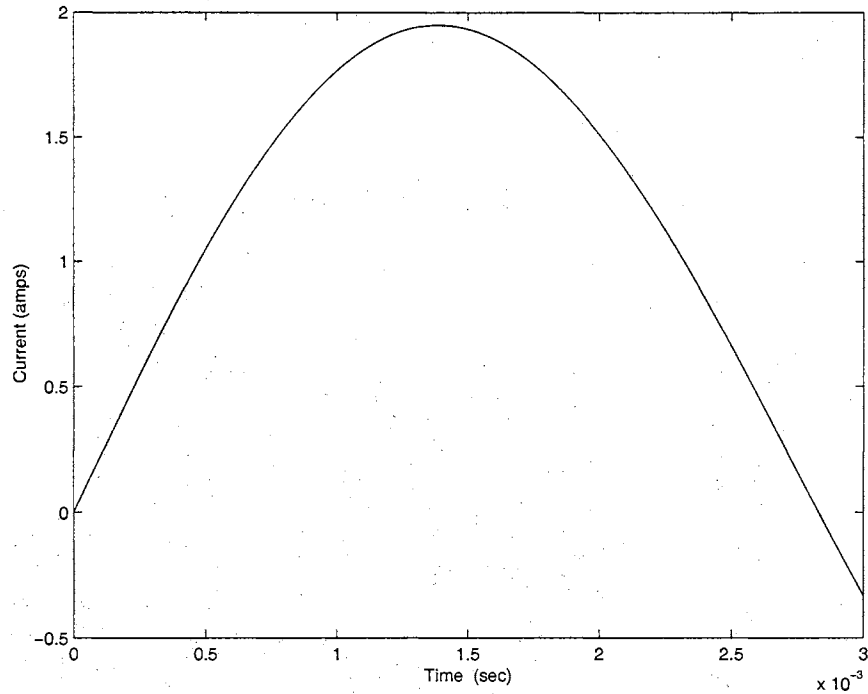


Figure 6.1: Input Motor Current to a Second Order Resonance System

for matched uncertainty estimation, the matched uncertainty term $w(m)$ is not included as part of the state matrix.

The SIMULINK diagrams of the general simulation of the DAO applied to this second order resonance model and the resonance model, itself, are shown in Figures 6.2 and 6.3, respectively. The general SIMULINK simulation of the Discrete Adaptive Observer is shown in Figure 6.4. For a more detailed look at the simulation, the reader is referred to the appendix.

As the original HDA system is not in the usable observer canonical output form, a transformation is needed, as described in Equations (5.5) through (5.7). In other words, a transformation matrix T is needed to transform the system in Equation (4.5) to the form of Equation (5.5) such that

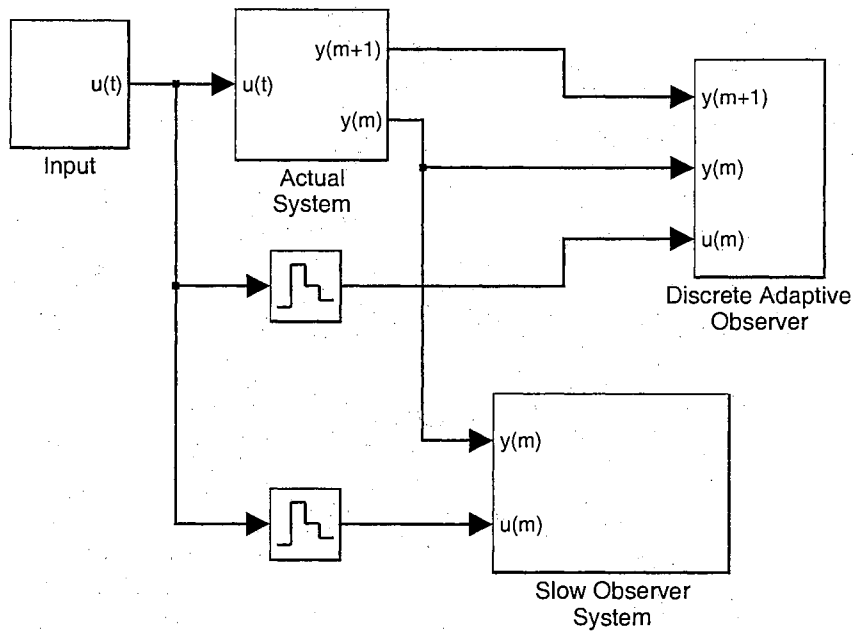


Figure 6.2: DAO Implementation on a Resonance Model

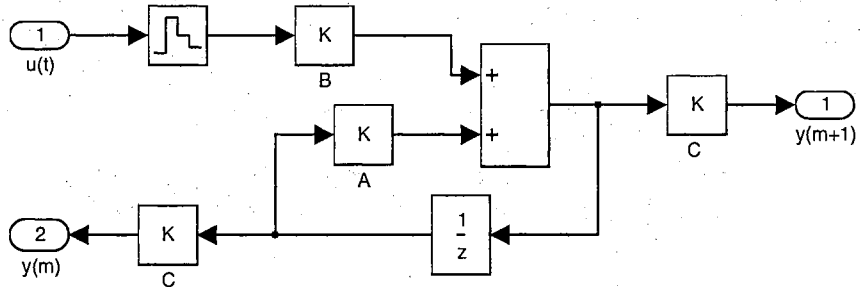


Figure 6.3: The Discrete Resonance Model

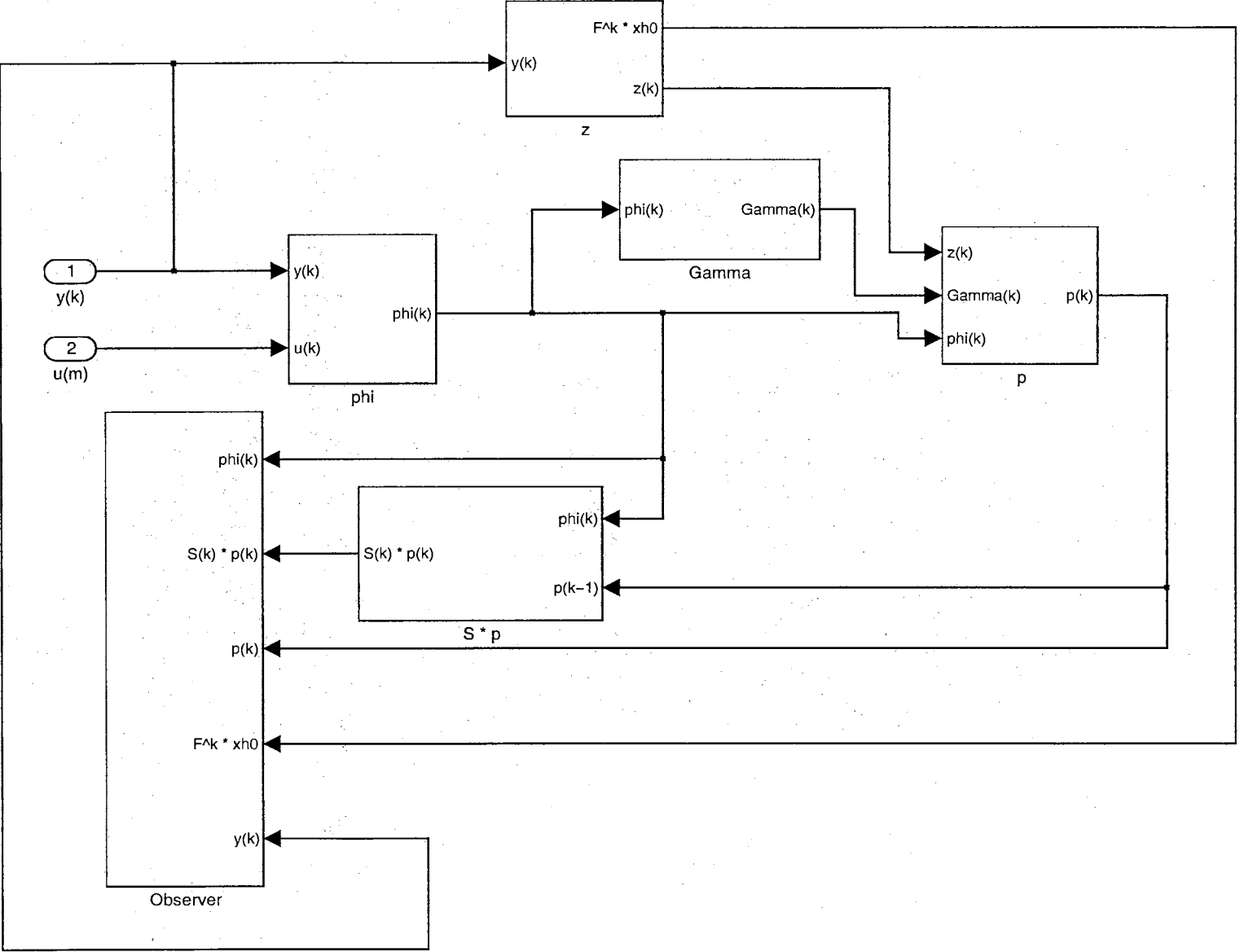


Figure 6.4: The Discrete Adaptive Observer

$x_T(m) = Tx(m)$ and

$$\begin{aligned}
 A_T = TA_sT^{-1} &= \begin{bmatrix} -a_1 & 1 \\ -a_2 & 0 \end{bmatrix} \\
 B_T &= TB_s \\
 C_T = CT^{-1} &= [1 \ 0]
 \end{aligned} \tag{6.3}$$

Using the method used in [49], the transformation matrix is given such that

$$T = \begin{bmatrix} 1 & 0 \\ a_1 & 1 \end{bmatrix} \begin{bmatrix} C \\ CA_s \end{bmatrix} \tag{6.4}$$

The resulting system for the Discrete Adaptive Observer (DAO) is as follows:

$$\begin{aligned}
 A_T = TAT^{-1} &= \begin{bmatrix} 1.6446e+00 & 1.0000e+00 \\ -9.0032e-01 & 0 \end{bmatrix} \\
 B_T = TB &= \begin{bmatrix} 1.3010e-01 \\ 1.2559e-01 \end{bmatrix} \\
 C_T = CT^{-1} &= [1 \ 0]
 \end{aligned} \tag{6.5}$$

where

$$T = \begin{bmatrix} 1.0000e+00 & 0 \\ -7.7473e-01 & 3.1720e-04 \end{bmatrix} \tag{6.6}$$

In addition, the DAO parameters are initialized as follows:

$$\begin{aligned}
 \phi(0) &= 0^{4 \times 1} \\
 \Gamma(0) &= d^2 I_4 \\
 \lambda &= \sqrt{0.5} \\
 \hat{p}(0) &= 0^{4 \times 1} \\
 d &= 10
 \end{aligned} \tag{6.7}$$

The state variable filter F is chosen, following the example of Suzuki et al., to be

$$F = \begin{bmatrix} 1.4900e+00 & 1.0000e+00 \\ -5.5000e-01 & 0 \end{bmatrix} \quad (6.8)$$

The Slow Observer System is also implemented in this simulation of the resonance system to compare with the DAO. However, as in the previous set of simulations, the estimated parameter values are used. That is,

$$\begin{aligned} \hat{\zeta} &= 0.85 \zeta \\ \hat{\omega}_n &= 0.85 \omega_n \end{aligned} \quad (6.9)$$

for the observer estimated model

$$\hat{R}(s) = \frac{\hat{\omega}_n^2}{s^2 + 2\hat{\zeta}\hat{\omega}_n s + \hat{\omega}_n^2} \quad (6.10)$$

The Slow Observer System is also applied in this simulation in the augmented format, as with the DAO, of Equation (3.2) where the observer gains L_s are chosen so that the observer poles are at $[1e-8 \pm 5e-8i \quad 1e-9]$. The resulting gain matrix is

$$L_s = \begin{bmatrix} 2.7383e+00 \\ 5.6753e+03 \\ 5.3019e+00 \end{bmatrix} \quad (6.11)$$

6.2.3 Resonance System Simulation Results

The simulation results of the Discrete Adaptive Observer (DAO) System and Slow Observer System (SOS) position estimates are shown in Figure (6.5). The DAO estimate converges directly to the actual position. Although the Slow Observer System allows for the estimation of matched uncertainties, it does not converge as quickly as the DAO because its design does not take into account any unmatched uncertainties.

The velocity estimate of the DAO is shown in Figure 6.6. There is a large initial transient where the Discrete Adaptive Observer is initializing itself and reacting to different initial conditions than the actual system. It should be noted that this transient behavior is inherent to general adaptive systems.

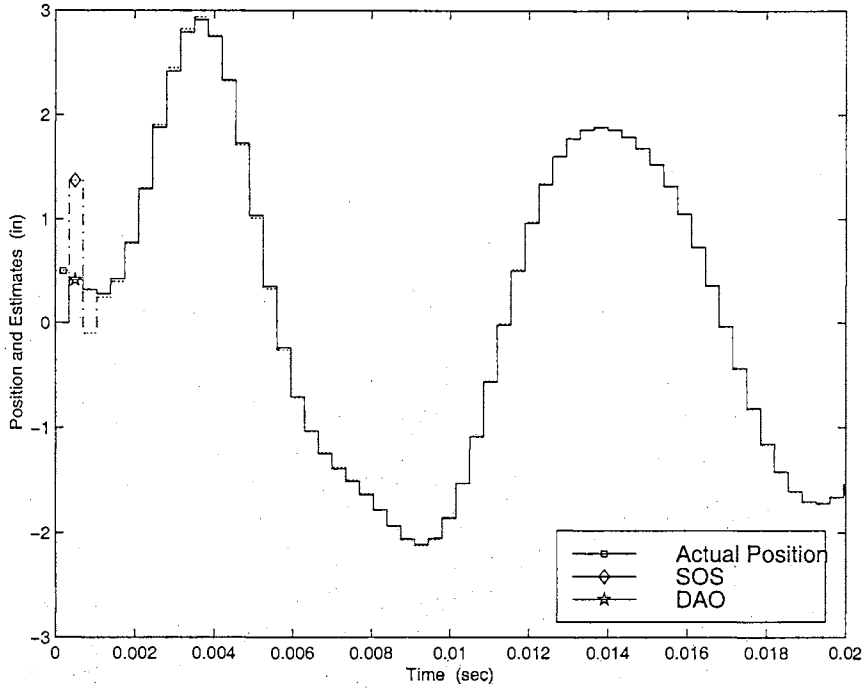


Figure 6.5: Actuator Position Output and Estimates

In addition, because of the relatively large difference between the estimated initial conditions and actual initial conditions, this transient is much more amplified. After the initial transient, however, the DAO converges to the velocity output.

The comparative results between the Discrete Adaptive Observer and the Slow Observer System are shown in Figures 6.7 and 6.8. The first plot confirms that the DAO position estimate error converges quickly to 0. The latter plot shows the initial DAO transient, as mentioned before, and then the decrease in error. Both plots show the SOS estimates converging to the actual states. Because it cannot adjust itself as quickly to the mismatched system parameters, however, the speed of convergence of the DAO is much quicker than that of the Slow Observer System.

Simulations of the DAO parameter estimates are shown in Figures 6.9 through 6.12. It can be seen that not only does the DAO position estimate converge quickly and that of the velocity estimate converge fairly quickly after the initial transient, but the estimates of the system parameters also settle to actual values. This is an added advantage over the original Slow Observer System in that corrections in parameter values are adjusted by the Discrete Adaptive Observer, while the Slow

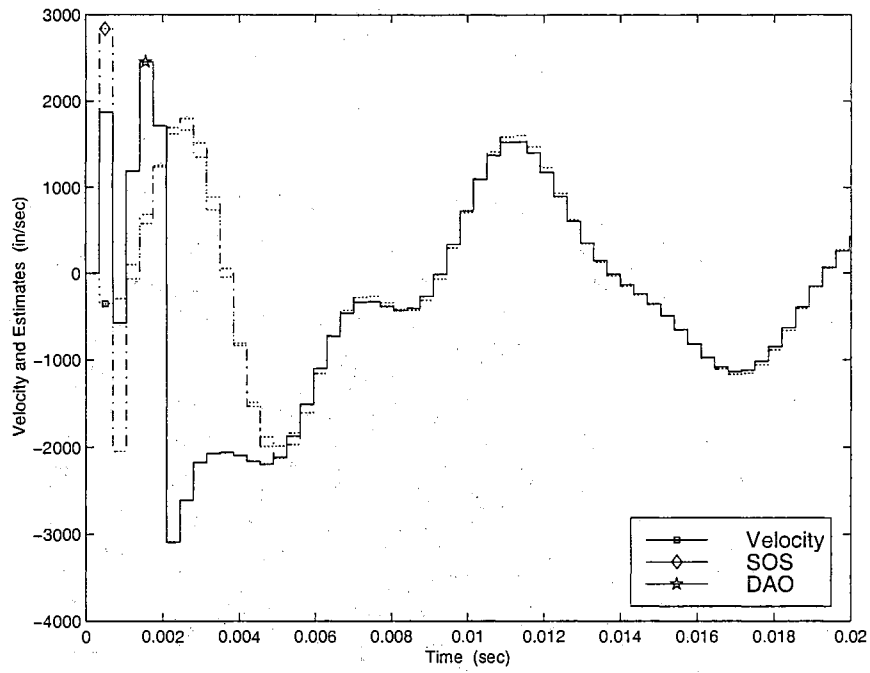


Figure 6.6: Actuator Velocity Output and Estimates

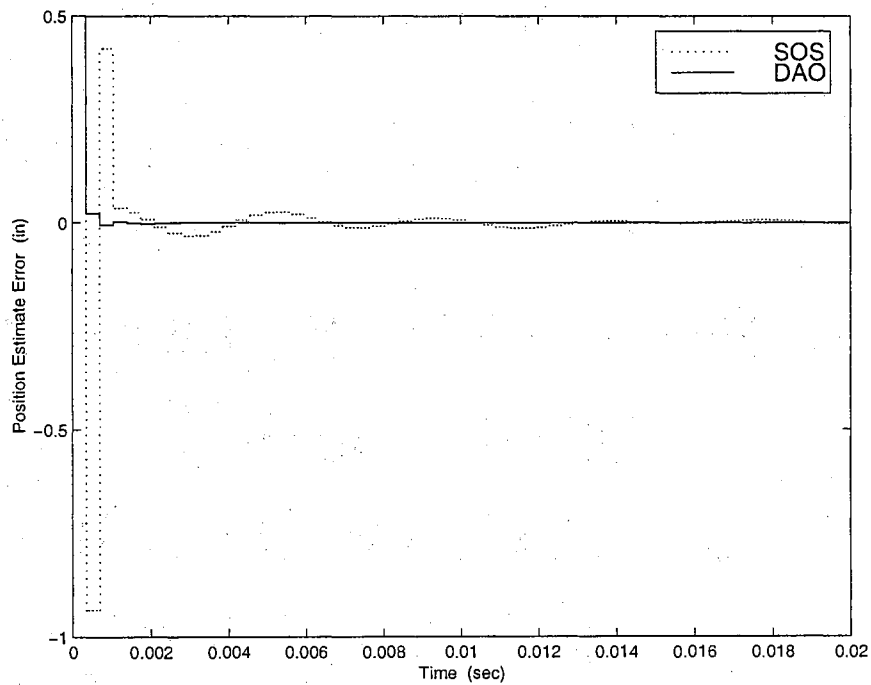


Figure 6.7: DAO and Slow Observer System Position Estimate Errors

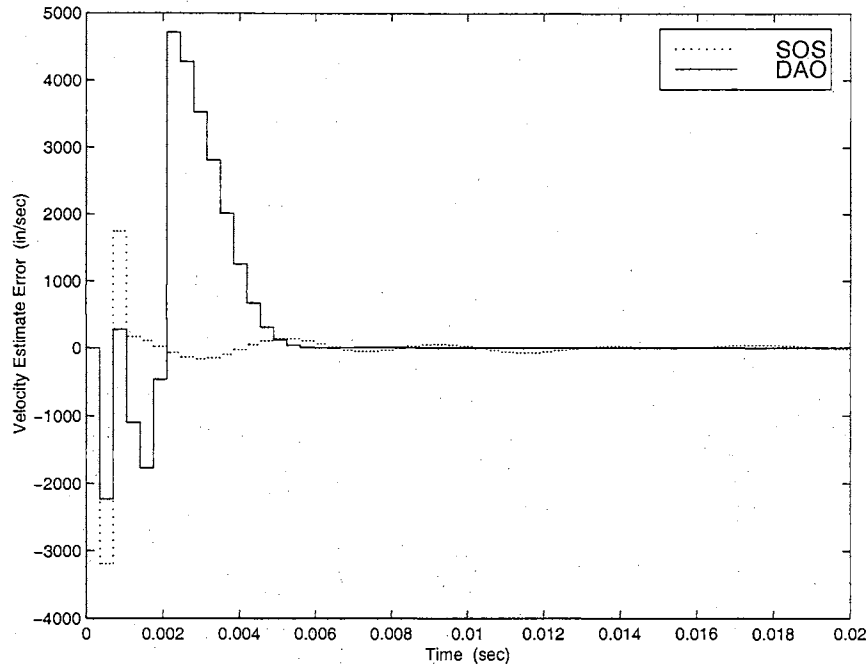


Figure 6.8: DAO and Slow Observer System Velocity Estimate Errors

Observer System can only adjust itself via its observer gain. In addition, any additional changes to the system will not cause as great a transient, as seen in the initialization, as, assuming the DAO has already converged to actual state estimate values, the DAO and the changing system will start with the same initial conditions.

6.3 Implementation of the APOS to the IBM HDA

The DAO algorithm as presented is implemented exactly on the IBM magnetic Head/Disk Assembly from Chapter 4. The same conditions are applied except that the input current to the power amplifier is changed to a pseudo random binary input signal. This is to supply a sufficiently rich input signal to satisfy the necessary conditions for the stability of the Discrete Adaptive Observer (DAO) implemented in the Adaptive Parallel Observer System (APOS).

The same matched uncertainty signal is also being used in this example, as in Chapter 4. In addition, the simulations are being run with the same augmented state form to allow for matched uncertainty estimation. It should be noted, however, that the DAO does not specifically account for

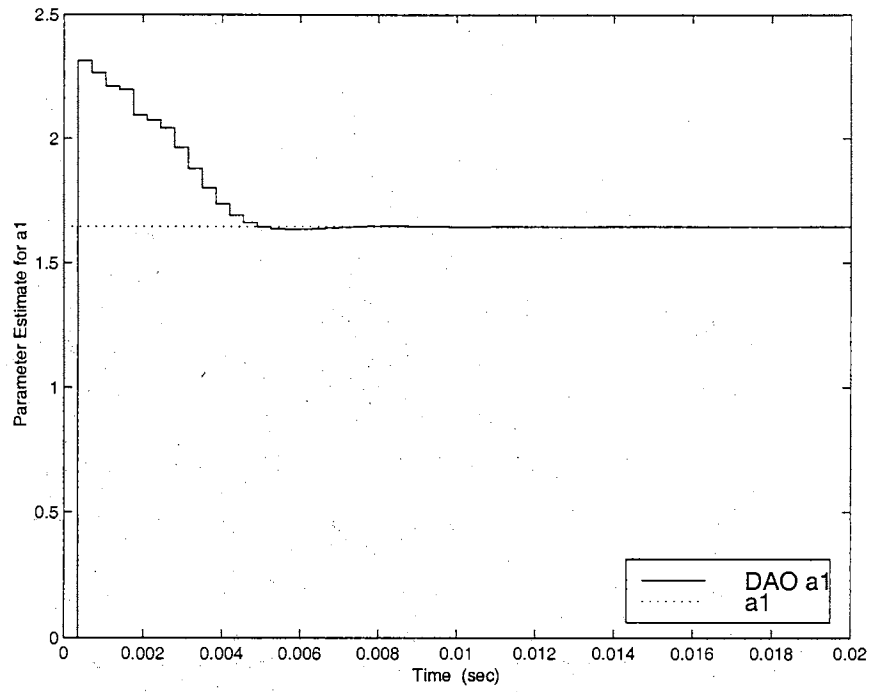


Figure 6.9: DAO Parameter Estimate of a_1

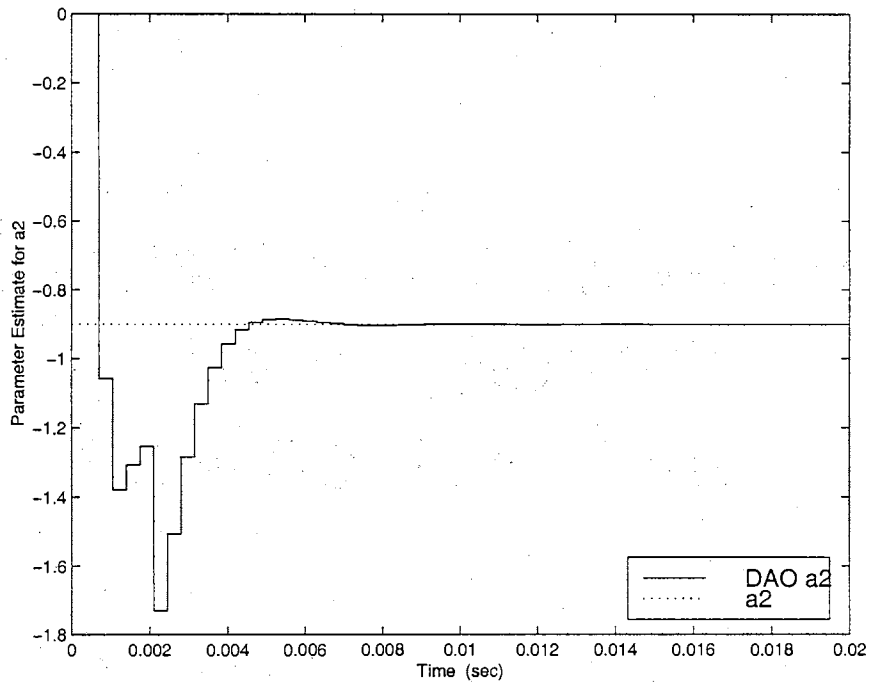


Figure 6.10: DAO Parameter Estimate of a_2

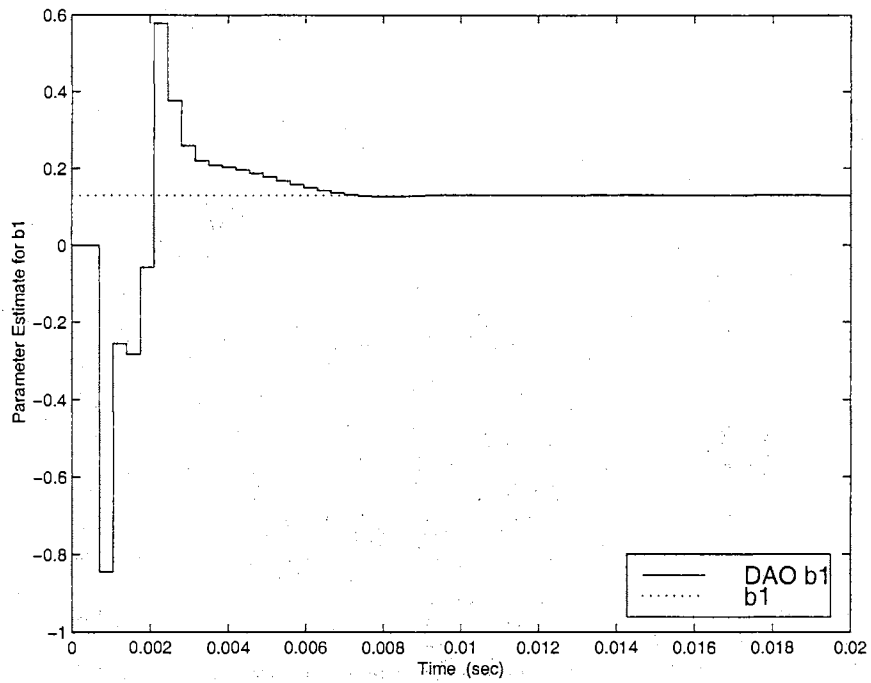


Figure 6.11: DAO Parameter Estimate of b_1

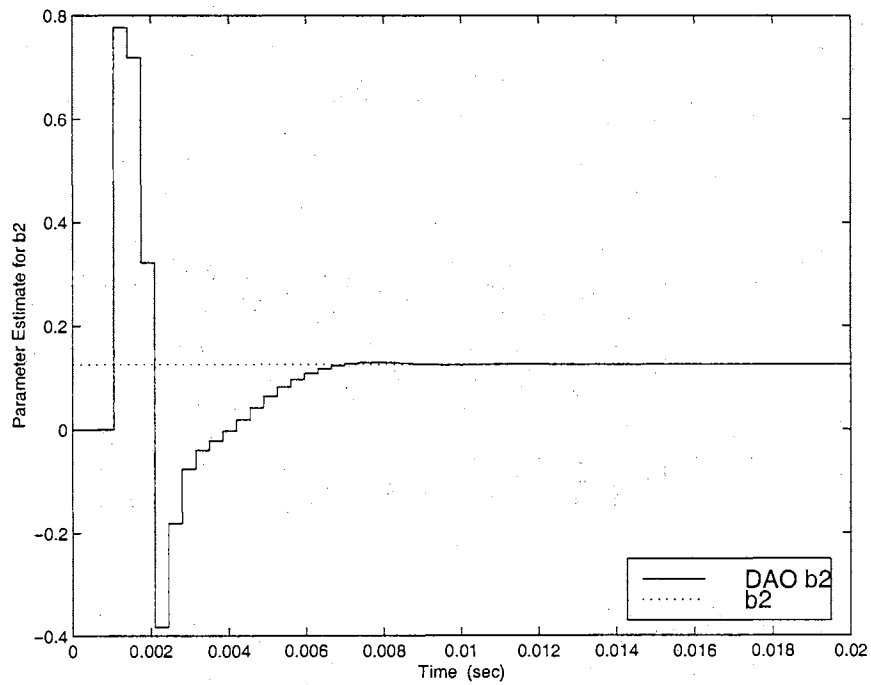


Figure 6.12: DAO Parameter Estimate of b_2

either matched uncertainties or the augmented state form. However, for purposes of comparison to the example of Chapter 4, these conditions are being consistently implemented.

As in the previous set of simulations, the APOS is compared to the single rate Slow Luenberger Observer and the Fast Luenberger Observer. To compare the POS to the modified structure APOS, the POS's Slow Observer System state estimates are shown with those of the APOS' Discrete Adaptive Observer (DAO) state estimates.

6.3.1 DAO Implementation Issues for the IBM HDA

The DAO algorithm as presented is implemented exactly on the IBM magnetic Head/Disk Assembly from Chapter 6 as described in Equations (5.8) through (5.18). However, to prevent unneeded parameter adaptation during times when the parameter estimates have already converged to their respective values, the parameters, according to the magnitude of the output estimation error at the previous sample time step, are held constant. This error, referred to as the error threshold e_{thresh} , is set to $1e-6$ inches for all of the simulations shown. This also prevents unnecessary parameter estimate oscillation due to the DAO continuously trying to obtain better parameter estimates when, in fact, it may not be possible.

In addition, the output measurement $y(m+1)$ as needed in Equation (5.17) is assumed to be unavailable and that only $y(m)$ may be used for estimation purposes. In effect, the parameter estimates being used for the observer is $\hat{p}(m-1)$ instead of $\hat{p}(m)$. This result, though, merely means that the parameter estimates obtained shows a one measurement time step delay, which should not noticeably affect the overall disk drive simulation.

6.3.2 The HDA System Model

In this simulation, as in Chapter 4, the same actuator and resonance dynamics from the IBM magnetic Head/Disk Assembly system are used as the simulation:

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{cf}}{M_{act}} & -\frac{b_v}{M_{act}} & \frac{1}{M_{act}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_f\omega_n^2 \end{bmatrix} u(t) \\
 y(t) &= [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \tag{6.12}
 \end{aligned}$$

where $\omega_n = 350\text{rad/s}$ and $\zeta = 0.30$.

To illustrate the effects of the new input, the matched uncertainties, and the unmodeled dynamics on the actuator output, a plot of the actuator position output under each of these conditions is shown in Figure 6.13, and those for the actuator velocity are shown in Figure 6.14.

6.3.3 The DAO System Model

The DAO system is applied to the IBM magnetic Head/Disk Assembly with the presence of the same matched and unmatched uncertainties as before. As in the case as the of the POS, the APOS is also augmented to facilitate matched uncertainty estimation. The general SIMULINK simulation of the Adaptive Observer System is shown in Figures 6.15 and 6.16. For a more detailed look at the simulations, the reader is referred to Appendix A.

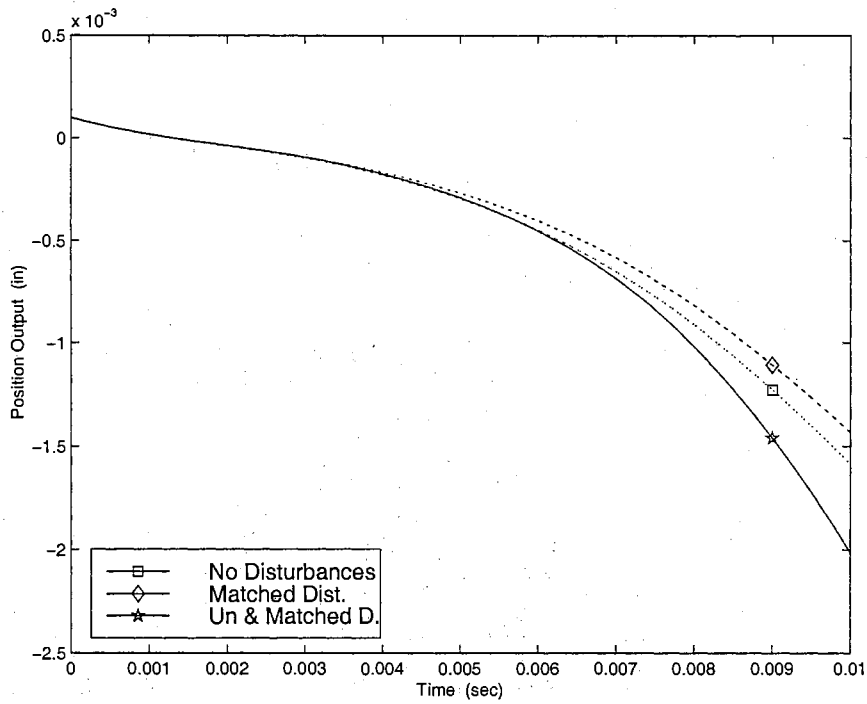


Figure 6.13: Actuator Position Responses

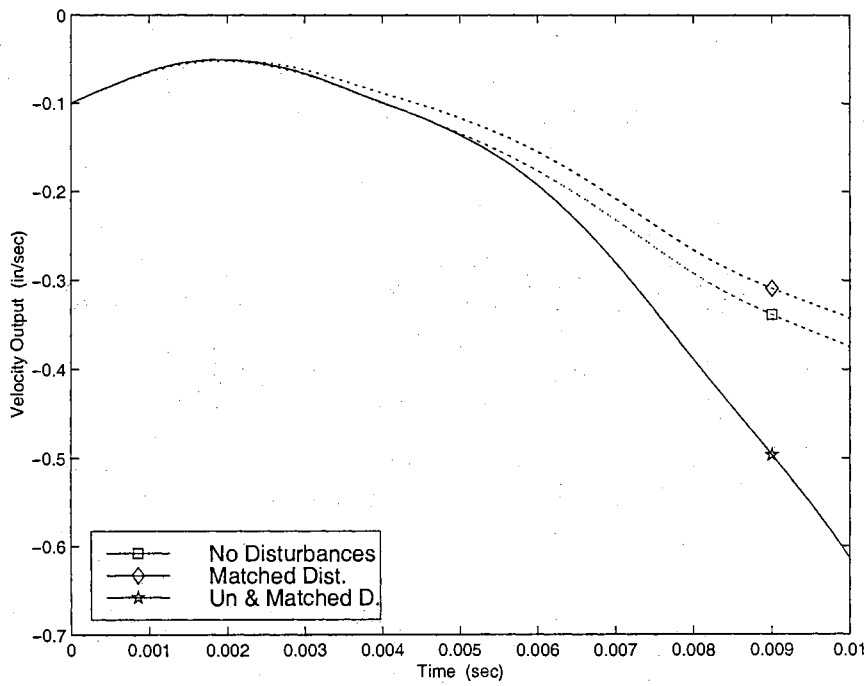


Figure 6.14: Actuator Velocity Responses

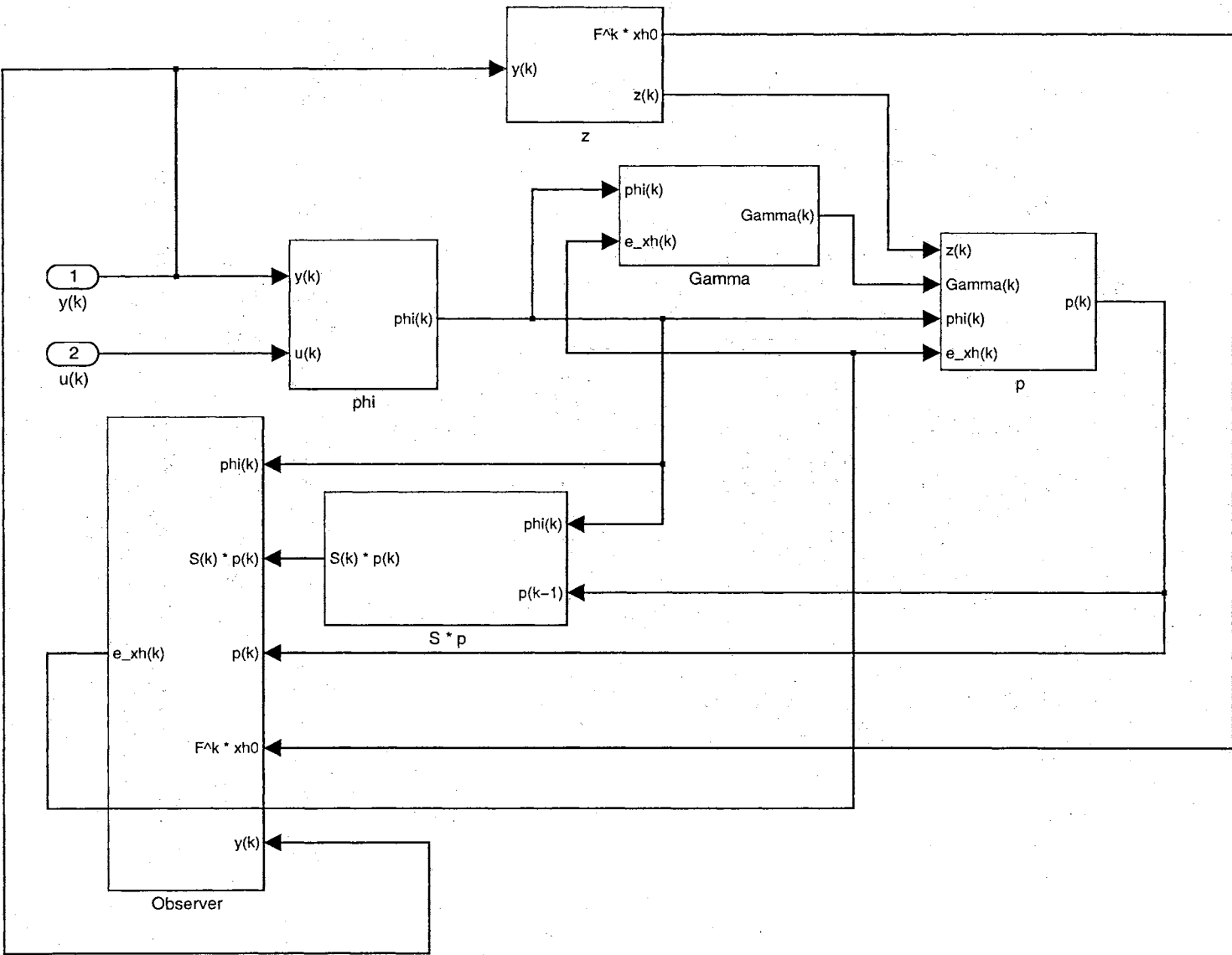


Figure 6.15: The Discrete Adaptive Observer

The same model is used in this example simulation as in Equation (4.5) of Chapter 4. That is,

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{cf}}{M_{act}} & -\frac{b_v}{M_{act}} & \frac{1}{M_{act}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\hat{\omega}_n^2 & -2\hat{\zeta}\hat{\omega}_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_f\hat{\omega}_n^2 \end{bmatrix} \hat{u}(t) \\
 y(t) &= [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \tag{6.13}
 \end{aligned}$$

where $\hat{\zeta}$, $\hat{\omega}_n$, and $\hat{u}(t)$ are such that

$$\begin{aligned}
 \hat{\zeta} &= 0.85 \zeta \\
 \hat{\omega}_n &= 0.85 \omega_n \\
 \hat{u}(t) &= i_m(t) \tag{6.14}
 \end{aligned}$$

It should be noted that the design of the DAO does not necessarily take into account the matched certainties of the system, regardless of whether or not the system is augmented to allow for uncertainty estimation, as these conditions do not satisfy those stated in the proof for stability in [48].

As the original HDA system is not in the usable observer canonical output form, a transformation is needed, as described in Equations (5.5) through (5.7). In other words, a transformation matrix T is needed to transform the system in (4.5) to the form of (5.7) such that

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 & 0 \\ a_2 & a_1 & 1 & 0 & 0 \\ a_3 & a_2 & a_1 & 1 & 0 \\ a_4 & a_3 & a_2 & a_1 & 1 \end{bmatrix} \begin{bmatrix} C \\ CA_T \\ C[A_T]^2 \\ C[A_T]^3 \\ C[A_T]^4 \end{bmatrix} \tag{6.15}$$

The resulting system for the Discrete Adaptive Observer (DAO), which replaces the Slow Observer

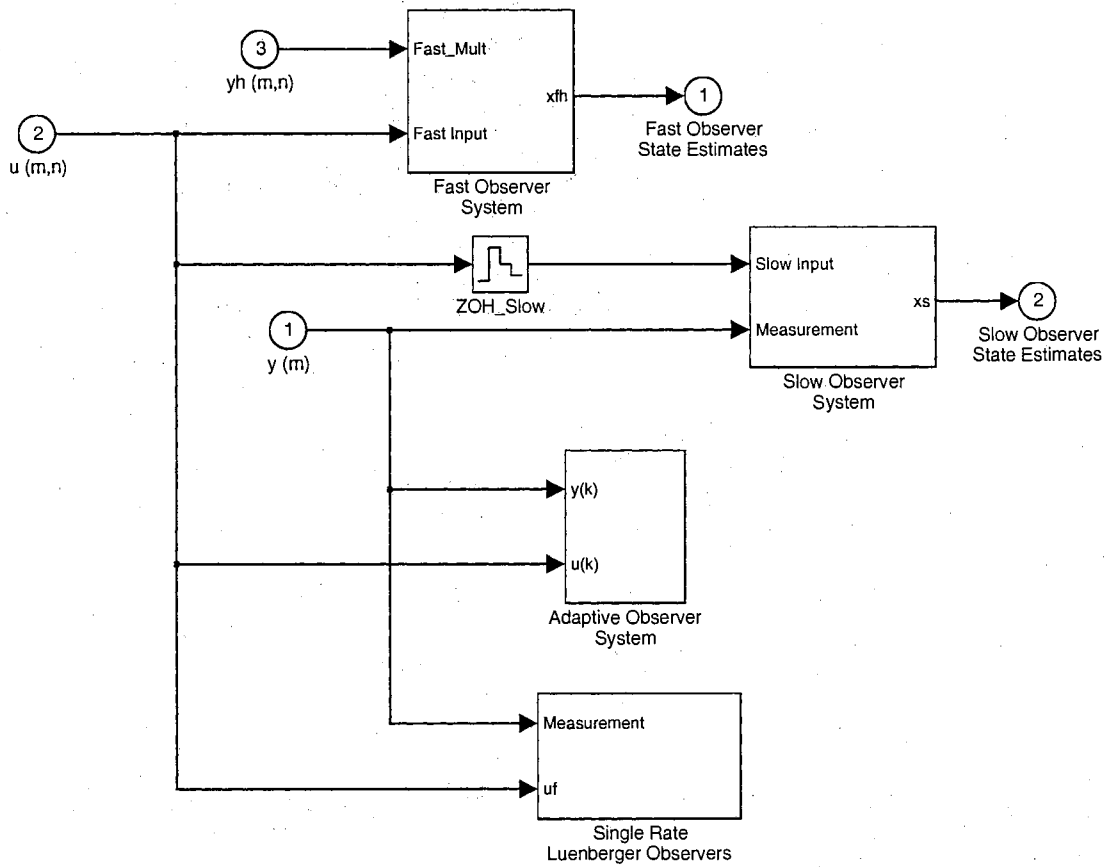


Figure 6.16: APOS, POS and Comparison Observers

System, is as follows:

$$\begin{aligned}
 A_T = T A_s T^{-1} &= \begin{bmatrix} 4.9377e+00 & 1 & 0 & 0 & 0 \\ -9.7614e+00 & 0 & 1 & 0 & 0 \\ 9.6579e+00 & 0 & 0 & 1 & 0 \\ -4.7825e+00 & 0 & 0 & 0 & 1 \\ 9.4825e-01 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 B_T = T B_s &= \begin{bmatrix} 6.8421e-09 \\ 6.7586e-08 \\ -7.8710e-10 \\ -6.7014e-08 \\ -6.6274e-09 \end{bmatrix} \\
 C_T = C T^{-1} &= [1 \ 0 \ 0 \ 0 \ 0]
 \end{aligned} \tag{6.16}$$

where

$$T = \begin{bmatrix} 1.0000e+00 & 0 & 0 & 0 & 0 \\ -3.9377e+00 & 3.4999e-04 & 5.0996e-07 & 5.8734e-11 & 6.8421e-09 \\ 5.8237e+00 & -1.0282e-03 & -4.8360e-07 & 1.7299e-10 & 7.4428e-08 \\ -3.8342e+00 & 1.0101e-03 & -5.0993e-07 & -1.7453e-10 & 7.3641e-08 \\ 9.4825e-01 & -3.3189e-04 & 4.8357e-07 & -5.7194e-11 & 6.6274e-09 \end{bmatrix}$$

In addition, the DAO parameters are initialized as follows:

$$\begin{aligned}
 \phi(0) &= 0^{10 \times 1} \\
 \Gamma(0) &= d^2 I_{10} \\
 \lambda &= 0.55 \\
 d &= 10
 \end{aligned} \tag{6.17}$$

$$\begin{aligned}
\hat{p}(0) &= \begin{bmatrix} [A_T]_{ii} \\ B_T \end{bmatrix} \\
&= \begin{bmatrix} 4.9114e+00 \\ -9.7611e+00 \\ 9.6579e+00 \\ -4.7825e+00 \\ 9.4825e-01 \\ 6.8421e-09 \\ 6.7586e-08 \\ -7.8710e-10 \\ -6.7014e-08 \\ -6.6274e-09 \end{bmatrix}
\end{aligned} \tag{6.18}$$

The state variable filter F are arbitrarily chosen such that the resulting eigenvalues are 0.003 ± 0.001 , 0.01 ± 0.002 and 0.0003 . The resulting state variable filter is

$$F = \begin{bmatrix} 2.6300e-02 & 1 & 0 & 0 & 0 \\ -2.4180e-04 & 0 & 1 & 0 & 0 \\ 8.9420e-07 & 0 & 0 & 1 & 0 \\ -1.2872e-09 & 0 & 0 & 0 & 1 \\ 3.1200e-13 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.19}$$

The same Fast Observer System model used in the example simulation in Chapter 4 is also implemented in this simulation. In addition, the same Fast Observer System observer gains in Equation (4.11) are used in this simulation as well.

6.3.4 HDA Simulation Results

The simulation results of the Adaptive Parallel Observer System position estimates are shown in Figure 6.17. There is a large initial transient where the Discrete Adaptive Observer is initializing itself. It should be noted that this transient behavior is inherent to general adaptive systems. In

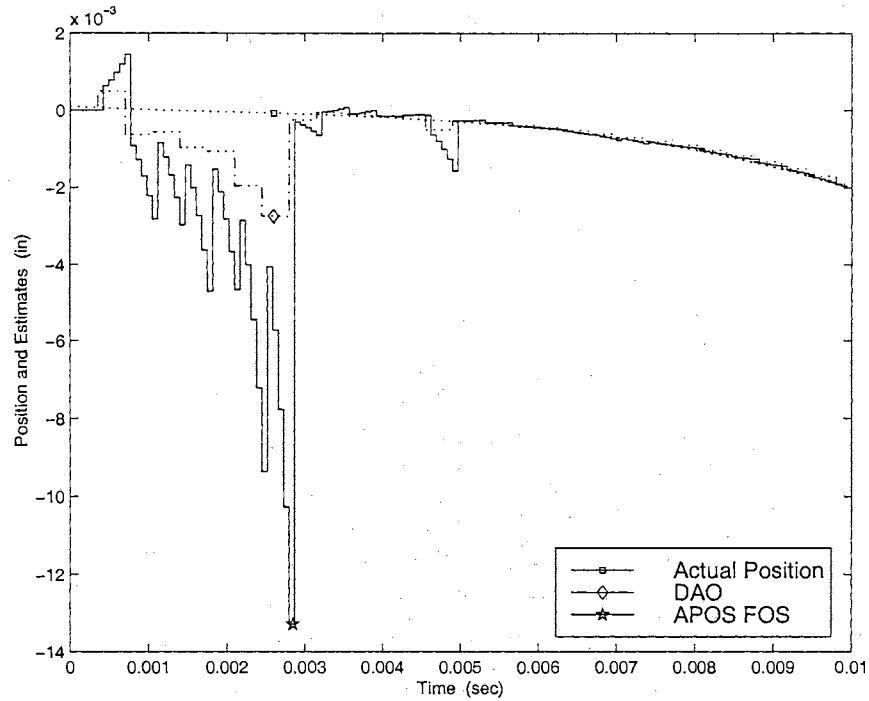


Figure 6.17: Actuator Position Output and APOS Estimates

addition, the parameter and state estimates are starting with different initial conditions from the actual state and system parameters. A large transient is seen from the Fast Observer System because of the DAO's initialization period. After the DAO begins to converge to the actual position signal, however, it can be seen that the Fast Observer System position estimates converge also.

Analogous results are obtained in the velocity estimates of the APOS, as shown in Figure 6.18. After the initial transient, both the DAO and the Fast Observer System converge to the velocity output. Again, it is clearly seen that the Fast Observer System is greatly affected by the accuracy of the state estimates of the driving observer which, in this case, is the DAO.

To view the results with the comparative Luenberger Observers between that of the APOS' Discrete Adaptive Observer and the POS's Slow Observer System, the reader is referred to Figures 6.19 and 6.20, where the position estimate errors and the velocity estimate errors are shown, respectively. Both plots show the initial DAO transient and then the decrease in error estimates of both the DAO and the Slow Observer System. The comparative results between the single rate Slow Luenberger Observer and Fast Luenberger Observer are shown in Figures 6.21 and 6.22. Again, the same results

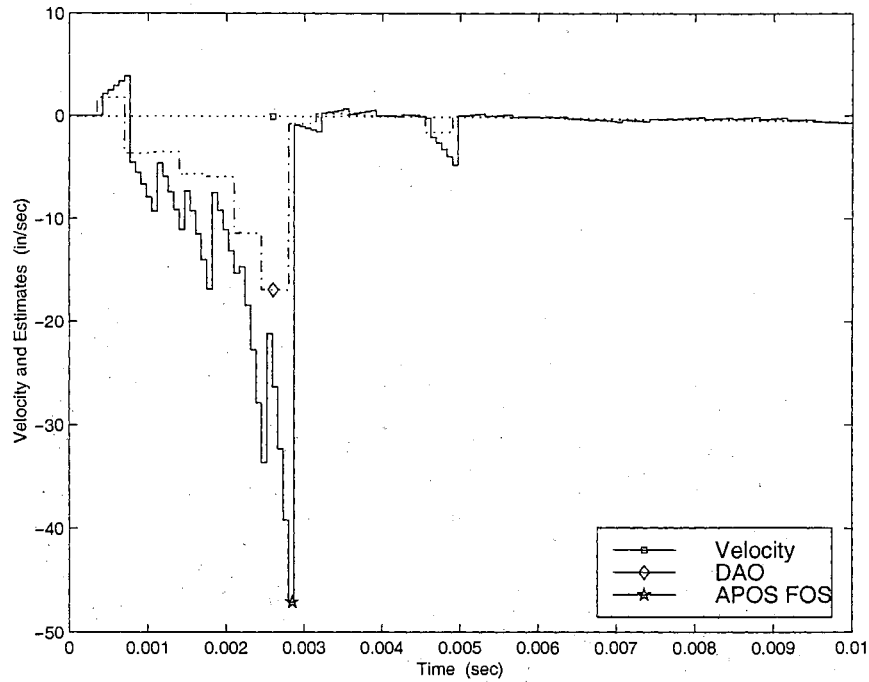


Figure 6.18: Actuator Velocity Output and APOS Estimates

can be seen in this set of comparisons.

Typical simulations of the DAO parameter estimates are shown in Figures 6.23 and 6.24. The rest of the ten DAO parameter estimates may be found in the Appendix B. These simulations show that the parameter estimates vary little from their initial values and show no positive convergence to actual values. As pointed out earlier, however, the implemented DAO is not guaranteed to be stable for the case of augmented states or for matched certainty estimation, as is in this set of simulations. In addition, it is possible that the DAO has found other parameters that fit the model description and these set of parameter estimates simply do not happen to be the intended model parameter values. This reasoning is supported by the fact that matched uncertainties are introduced in the system, which may easily be mistaken by the DAO as a mismatch in system parameters.

6.3.5 HDA System Simulation Results

It is seen that the state estimates of the Adaptive Parallel Observer system, after the pass of final transients, converges to those of the actual states, despite the added input disturbances and system

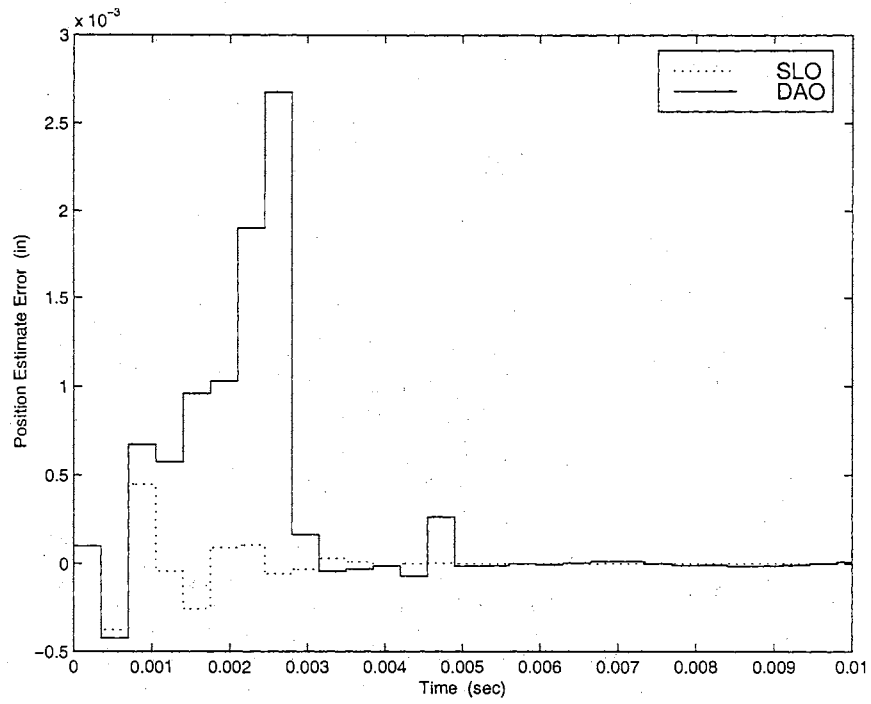


Figure 6.19: DAO and Slow Observer System Position Estimate Errors

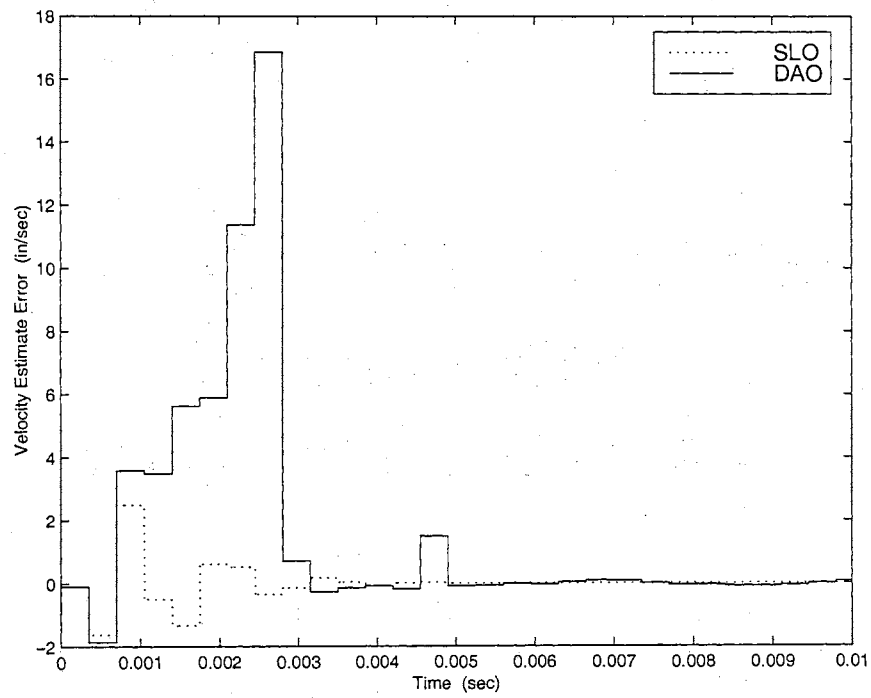


Figure 6.20: DAO and Slow Observer System Velocity Estimate Errors

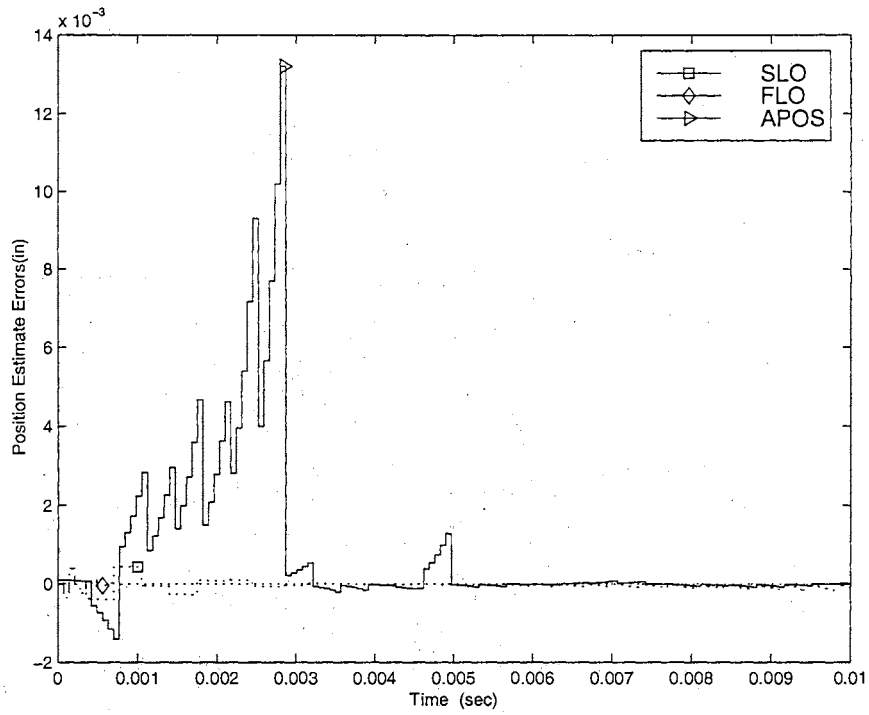


Figure 6.21: APOS and Comparison Observers Position Estimate Errors

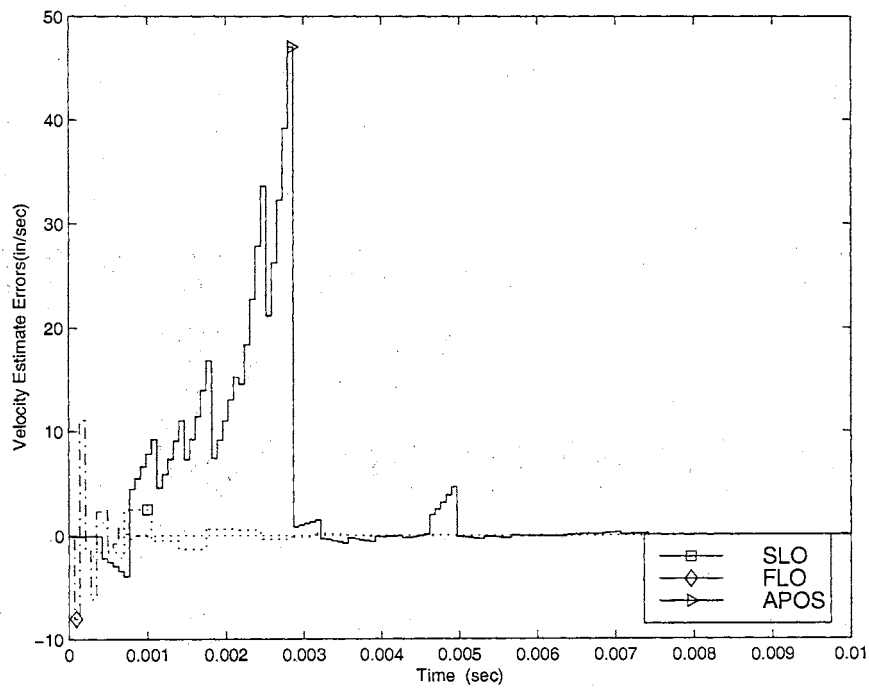


Figure 6.22: APOS and Comparison Observers Velocity Estimate Errors

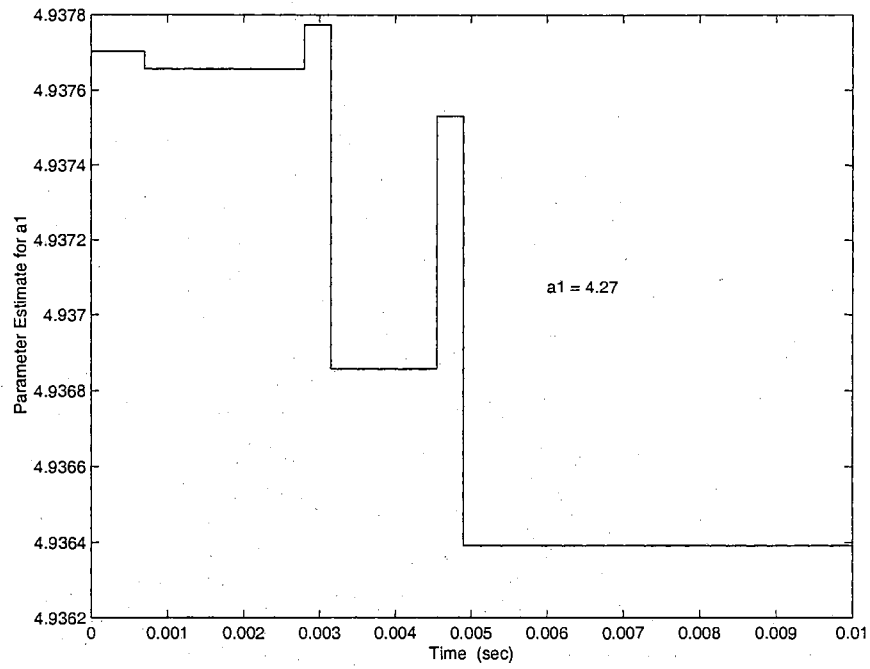


Figure 6.23: DAO Parameter Estimate of a_1

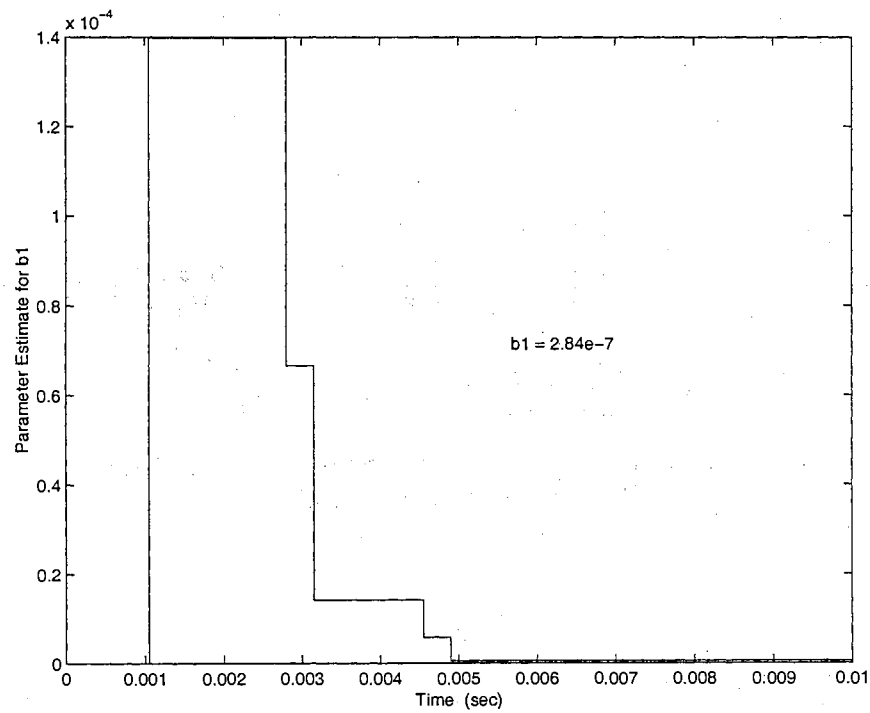


Figure 6.24: DAO Parameter Estimate of b_1

state augmentation. The transients are large but are indicative of general adaptive schemes and applications. These transients are magnified in these simulations due to the offset of initial state and parameter values between those of the APOS's Discrete Adaptive Observer and the actual system. Furthermore, it can be argued that, for the case of a disk drive system, any transients are insignificant because these transients disappear before the end of the computer system's "boot-up" period.

The DAO's parameter estimates do not converge to actual parameter values of the HDA system, possibly because stability and convergence of state and parameter estimates are not guaranteed under condition of state augmentation and added input disturbances. Yet, the state estimates still converge to actual state values. This infers that the DAO has found another set of A_T and B_T parameters that satisfy state estimate convergence.

It must be pointed out that tuning of the Adaptive Parallel Observer System, specifically of the Discrete Adaptive Observer is not a trivial matter. As discussed earlier in this chapter, a "tradeoff" exists between fast convergence of state estimates and parameter estimation oscillation. In addition, the added tuning parameter of the error threshold e_{thresh} introduces further tuning complications, as there is also a tradeoff here between accuracy of estimation and estimation oscillation. The tuning of the DAO is especially critical, since the accuracy of the Fast Observer System estimates relies on the estimates of the Discrete Adaptive Observer.

6.4 Summary Analysis of Simulation Results of the APOS

It is shown in the simulations that the Adaptive Parallel Observer system may be very beneficial in providing convergent state and parameter estimates, as long as the implemented Discrete Adaptive Observer is reliable. This reliability is guaranteed, among other necessary conditions, when there are no external inputs or disturbances and when the augmented state form of the system is not used. This is not the case in the latter set of simulations. Large transients are observed in the initializations of the APOS but disappear in an appropriate range of time.

The tuning of the APOS is difficult in that not only must the observer gains for the Fast Observer

System be chosen appropriately, but also the parameters λ and e_{thresh} . These parameters must be chosen so as to sustain accurate state and parameter estimation without significant oscillation in parameter estimates.

Chapter 7

Tuning of the DAO Parameters

It is found that the DAO from Suzuki et al. is accurate and shows fast convergence of states and parameter estimates for a second order resonance model. However, finding the optimal observer parameters for the DAO is not trivial.

The use of genetic algorithms in control applications is a relatively new concept. Furthermore, the added application of the genetic algorithm to tune state estimation parameters used for control applications is newer still. As a result of the difficulties in finding optimal observer parameters, a genetic algorithm (GA) is employed to automate this process. This GA uses the *rank fitness* method for the crossover selection process due to its simplicity to implement and its control over the selection process. This chapter presents the application of a genetic algorithm implemented off-line to obtain optimal observer parameters for the DAO.

7.1 DAO Implementation Issues for the IBM HDA

The DAO algorithm as presented is implemented exactly on the IBM magnetic Head/Disk Assembly as described in Equations (5.8) through (5.18). However, as before, the error threshold, e_{thresh} , is implemented to prevent unnecessary parameter estimate oscillation due to the DAO continuously trying to obtain better parameter estimates when, in fact, it may not be possible. Also as before, the parameter estimates being used for the observer is $\hat{p}(m - 1)$ instead of $\hat{p}(m)$.

In this case, however, the IBM HDA model is scaled by time, input, and state values for numerical conditioning and for more effective implementation of the Discrete Adaptive Observer. The entire HDA model, for purposes of simulation, consists of the current input into the actuator and the fourth order model described in Equation (4.3).

7.1.1 The Scaled Disc Drive Model

Given that the state space representation of the HDA system is

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{7.1}$$

the system is scaled using the following equations:

$$\begin{aligned}\frac{dX}{d\tau} &= \bar{A} + \bar{B}U \\ Y &= \bar{C}X\end{aligned}\tag{7.2}$$

where

$$\begin{aligned}\bar{A} &= t_{max}[P]^{-1}AP \\ \bar{B} &= t_{max}[P]^{-1}BQ \\ \bar{C} &= \frac{CP}{R}\end{aligned}\tag{7.3}$$

$$\begin{aligned}x &= PX \\ u &= QU \\ y &= RY \\ t &= t_{max}\tau\end{aligned}\tag{7.4}$$

and

$$\begin{aligned}
 P &= \begin{bmatrix} O(\max|x_1|) & 0 & 0 & 0 \\ 0 & O(\max|x_2|) & 0 & 0 \\ 0 & 0 & O(\max|x_3|) & 0 \\ 0 & 0 & 0 & O(\max|x_4|) \end{bmatrix} \\
 Q &= \max|u| \\
 R &= \max|y|
 \end{aligned} \tag{7.5}$$

7.1.2 DAO Disk Drive Implementation

The Discrete Adaptive Observer is applied to the disk drive system. The main emphasis of the following simulations is to show the performance of the DAO output and parameter estimates, given the implemented scaled model and the use of a genetic algorithm for optimal tuning of observer parameters.

The SIMULINK simulation diagram of the overall system is shown in Figure 7.1, and the implemented DAO is shown in Figure 6.15. For a detailed view of the specific parts of the DAO system, the reader is referred to the appendix. In this implementation, the DAO is running external to the system. For the system shown, $k = 5$ and $T_f = 75 \mu s$. These parameters are arbitrarily chosen. Furthermore, it should be noted here that the model described in [38] uses a single rate control/estimation strategy and that the sampling period used is $100 \mu s$.

The fourth order continuous time disk drive model is

$$\begin{aligned}
 \frac{dX}{d\tau} &= \bar{A}X + \bar{B}U \\
 Y &= CX
 \end{aligned} \tag{7.6}$$

where

$$\bar{A} = \begin{bmatrix} 0 & 1.0000e+00 & 0 & 0 \\ -3.3333e-05 & -8.3333e-04 & 8.3333e-01 & 0 \\ 0 & 0 & 0 & 1.0000e+02 \\ 0 & 0 & -9.4090e+01 & -1.9303e+01 \end{bmatrix}$$

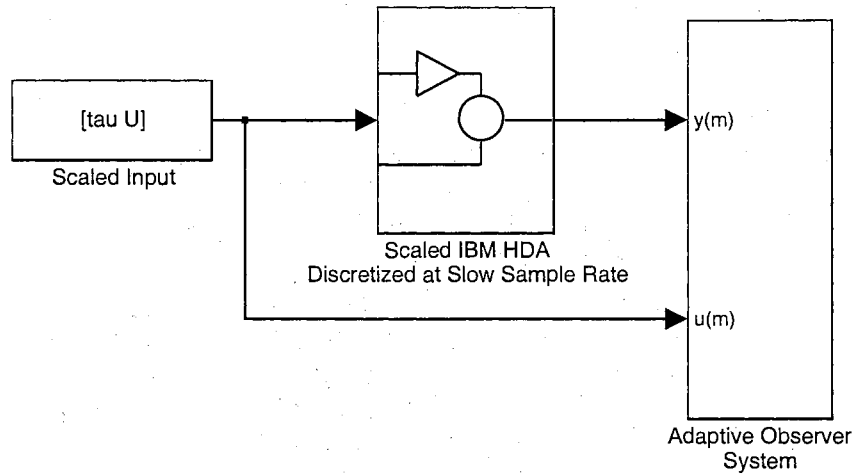


Figure 7.1: The DAO Disk Drive Implementation

and

$$\bar{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.4114e + 06 \end{bmatrix}$$

and the discretization of the system at the measured output rate is,

$$\begin{aligned} X(m+1) &= \bar{A}_s X(m) + \bar{B}_s U(m) \\ Y(m) &= CX(m) \end{aligned} \quad (7.7)$$

where \bar{A}_s and \bar{B}_s are the slow discretization matrices of the system described in Equation (7.6).

$$\bar{A}_s = \begin{bmatrix} 1.0000e-00 & 3.4999e-02 & 2.0826e-04 & 2.9425e-04 \\ -1.1666e-06 & 9.9997e-01 & 1.4802e-03 & 1.5146e-02 \\ 0 & 0 & -7.1019e-01 & -1.7321e-01 \\ 0 & 0 & 1.6297e-01 & -6.7675e-01 \end{bmatrix}$$

$$\bar{B}_s = \begin{bmatrix} 4.5322e + 00 \\ 4.1529e + 02 \\ 2.5653e + 04 \\ -2.4446e + 03 \end{bmatrix}$$

The DAO transformed \bar{A}_s matrix, denoted by \bar{A}_d is obtained by comparing coefficients of the characteristic polynomials of the given \bar{A}_s and the n parameters of \bar{A}_d , which is in observer canonical form. Then, the transformation matrix \bar{T} is found using Equation (5.5). In turn, the remaining DAO system parameters are calculated. The \bar{F} matrix is an arbitrarily chosen stable matrix in observer canonical form. The roots of \bar{F} are arbitrarily chosen to be $0.003 + 0.001i$, $0.003 - 0.001i$, $0.01 + 0.002i$, and $0.01 - 0.002i$.

Using the transformation \bar{T} such that

$$\bar{T} = \begin{bmatrix} 1.0000e + 00 & 0 & 0 & 0 \\ 3.8697e - 01 & 3.4999e - 02 & 2.0826e - 04 & 2.9425e - 04 \\ -8.7805e - 01 & 4.8542e - 02 & 3.2446e - 05 & 4.0878e - 04 \\ -5.0883e - 01 & 1.7809e - 02 & -6.7438e - 05 & 1.9462e - 04 \end{bmatrix}$$

the DAO system model is as follows:

$$X_d(m+1) = \bar{A}_d X_d(m) + \bar{B}_d U(m)$$

$$Y(m) = \bar{C}_d X_d(m)$$

where

$$\bar{A}_d = \begin{bmatrix} 1.9967e + 00 & 1.0000e + 00 & 0 & 0 \\ -9.9709e - 01 & 0 & 1.0000e + 00 & 0 \\ 4.2747e - 12 & 0 & 0 & 1.0000e + 00 \\ -6.3060e - 22 & 0 & 0 & 0 \end{bmatrix}$$

$$\bar{B}_d = \begin{bmatrix} 7.6394e-02 \\ 7.6504e-02 \\ -1.7810e-06 \\ -3.5733e-17 \end{bmatrix}$$

$$\bar{C}_d = C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} = C$$

and

$$\bar{F} = \begin{bmatrix} 2.6000e-02 & 1.0000e+00 & 0 & 0 \\ -2.3400e-04 & 0 & 1.0000e+00 & 0 \\ 8.2400e-07 & 0 & 0 & 1.0000e+00 \\ -1.0400e-09 & 0 & 0 & 0 \end{bmatrix}$$

The parameter estimates for \bar{A}_d and \bar{B}_d are initialized to zero.

7.2 Implementation of a Genetic Algorithm

The tuning of the DAO is difficult in that decreasing λ causes the state and parameter estimates to converge at a faster rate. However, if the value of λ is too small, the estimates show significant oscillation and possible system instability. It is also found that decreasing e_{thresh} guarantees more accurate state and parameter estimates, but decreasing it beyond a certain value results in excessive oscillations in the state and parameter estimates and eventual system instability.

The genetic algorithm in this paper is applied as an off-line technique to choose optimal values of λ and e_{thresh} . The genetic algorithm (GA) follows the standard rank fitness method format with a few exceptions. The first exception is that there is a separate mutation multiplier for each parameter, since each parameter must search through a different range of values. It should be noted that the probability of a mutation occurring is set at 10%. Second, whenever an intermediate generation is created, it is added to the old generation and all values are retained to create the new generation. The reason for this implementation is so that a history of parameters may be viewed, along with any existing trends. Third, instead of having a predetermined value of the cost function that terminates

the process, the GA stops once the two hundredth generation has been evaluated. These exceptions combined with the basic logic behind the genetic algorithm are converted into a code [50] written for MATLAB to tune the parameters of the observer. (The reader is referred to the appendix for more information.)

The individual is the ordered pair (λ, e_{thresh}) , where the elements are the two design parameters of the DAO. The genetic algorithm uses the cost function \mathcal{J} . Here, \mathcal{J} is a function of the output estimation error and its derivative such that

$$\begin{aligned}\mathcal{J} &= \sum_{i=0}^{T_{max}} [e_Y^T(i)e_Y(i) + e_V^T(i)e_V(i)] \\ e_Y(i) &= Y(i) - \hat{Y}(i) \\ e_V(i) &= V(i) - \hat{V}(i)\end{aligned}\tag{7.8}$$

It should be noted that since the GA is applied as an off-line technique to choose observer parameters for the DAO, it is assumed only for the implementation of the GA that the output of both the scaled position Y and velocity V are known to take into account state and parameter oscillation errors. If, in fact, the GA is implemented on-line, then the quantity $(e_Y(i) - e_Y(i-1))/\Delta t$ may be used in place of $e_V(i)$.

The initial population is set to

$$\text{Initial Population} = \begin{bmatrix} (0.78215, 3.1124e-6) \\ (0.68451, 1.1410e-4) \\ (0.59632, 2.1203e-2) \\ (0.46980, 2.9012e-1) \\ (0.46980, 3.1124e-6) \\ (0.78215, 2.9012e-1) \end{bmatrix}\tag{7.9}$$

The mutation rate is set to 0.92651 and 0.8 for λ and e_{thresh} , respectively, and the algorithm ends within 200 generations. In addition, the bounds of λ and e_{thresh} are set such that $0.40 \leq \lambda \leq 1$ and $0 \leq e_{thresh} \leq 0.3$ for stability reasons. The results of the rank fitness method of the genetic algorithm after 9 complete simulations yield consistent values of 0.49476 and 0 for λ and e_{thresh} ,

respectively. Note that the value of e_{thresh} obtained is an expected result, since no noise is added to the HDA system.

7.3 Simulation Results

The state variable filter F is arbitrarily chosen such that the resulting eigenvalues are $0.003 \pm 0.001j$ and $0.01 \pm 0.002j$. The Discrete Adaptive Observer System is implemented on the IBM HDA with no added disturbances. In addition, the input to the HDA system is a pseudo-random binary input. The design parameter values for λ and e_{thresh} obtained by the genetic algorithm are applied. Figure 7.2 shows the response of the actual position and that of the position estimate, where one sees that the output estimate matches well with the actual output. For a more detailed view, the reader is referred to Figure 7.3, where the error estimation can be seen more clearly. Here, it is shown that the DAO has a noticeable transient in the beginning of the simulation. This transient behavior, however, is expected, as this characteristic is indicative of most adaptive systems. After the initial transient, the estimate error converges directly to zero.

Figure (7.4) shows the results of the actual velocity and that of the velocity estimate. It is seen that the velocity estimate has a noted amount of oscillation during the characteristic “transient” phase. However, the velocity estimate does, in fact, converge to the actual velocity. The characteristics of the velocity estimate are seen more clearly in Figure 7.5, where the velocity error estimate is shown. Here, the large oscillations are seen. In addition, the negligible oscillations, as the error estimate proceeds to converge to zero, exist but are just barely detectable.

A typical response of the parameter estimates for \bar{A}_d is represented for parameter \bar{a}_1 as shown in Figure 7.6. All parameters values converge to actual values very quickly after the passing of the initial transient period. Convergence of all the parameter estimates occurs concurrently, which is at the time period shortly following the characteristic transient period. Once parameter estimates are within the range of actual parameter values, convergence is maintained with no noticeable oscillations.

A typical response of the parameter estimates for \bar{B}_d is represented for parameter \bar{b}_1 as shown in

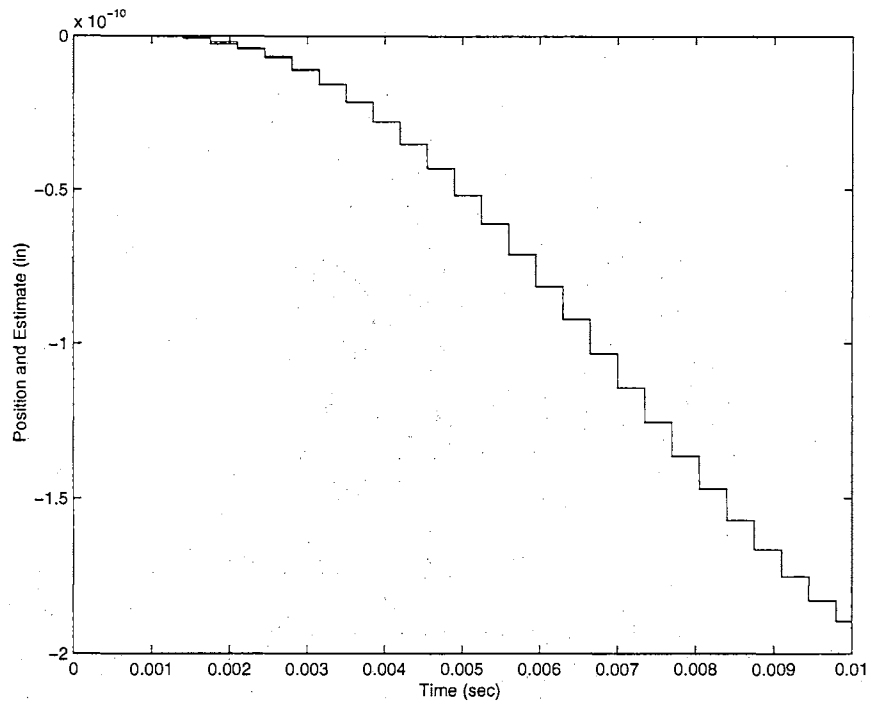


Figure 7.2: Actual Output and Observer Estimate

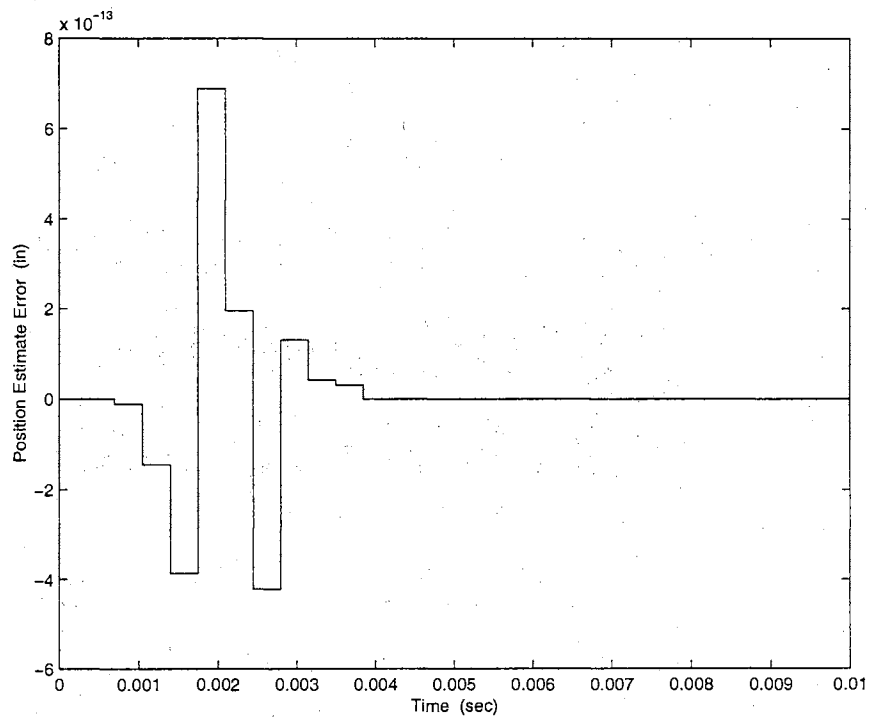


Figure 7.3: Output Observer Estimate Error

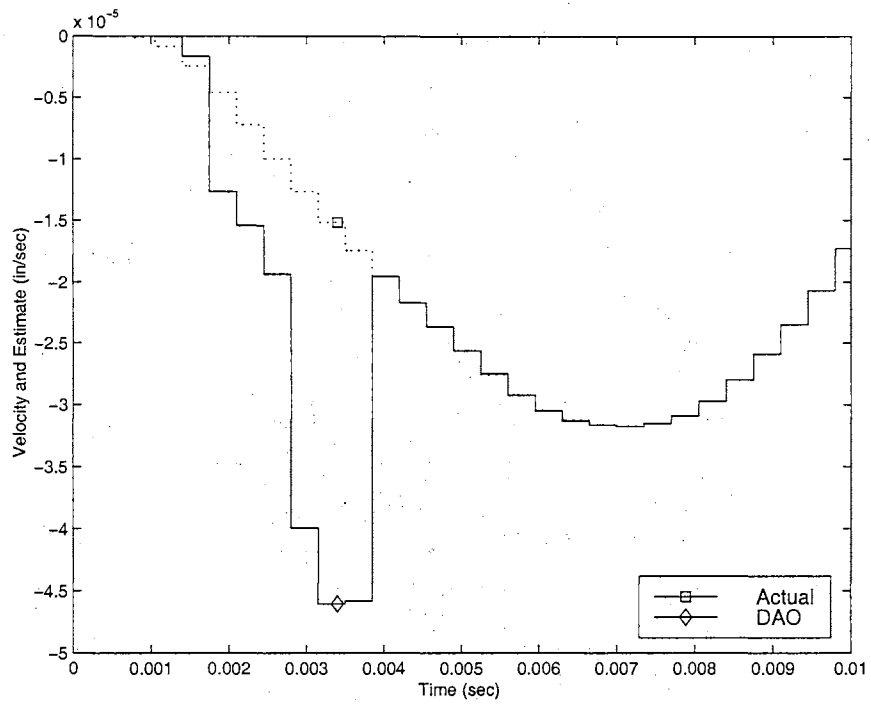


Figure 7.4: Actual Velocity and Observer Estimate

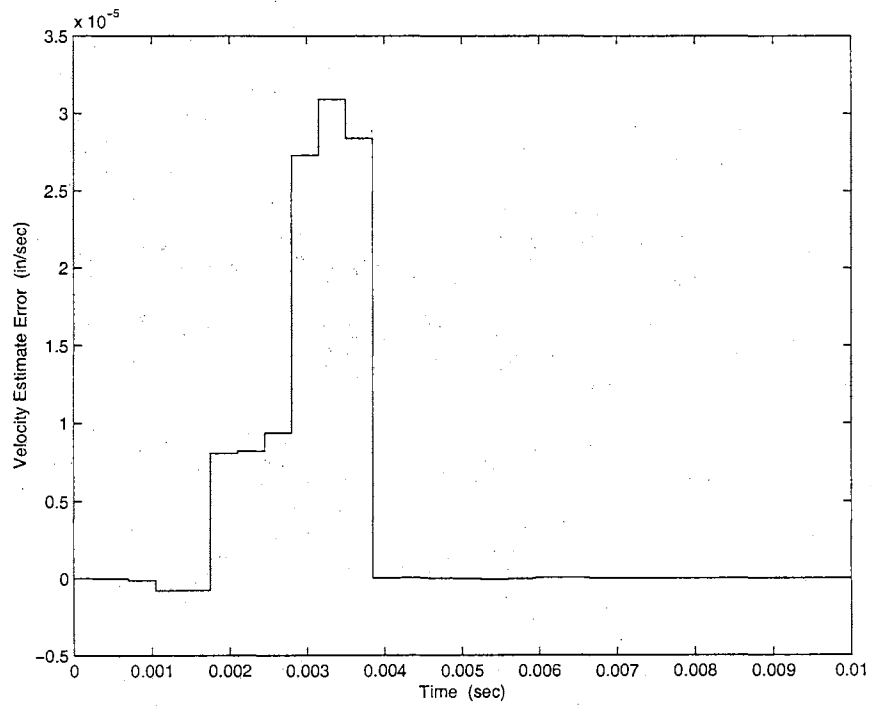


Figure 7.5: Velocity Observer Estimate Error

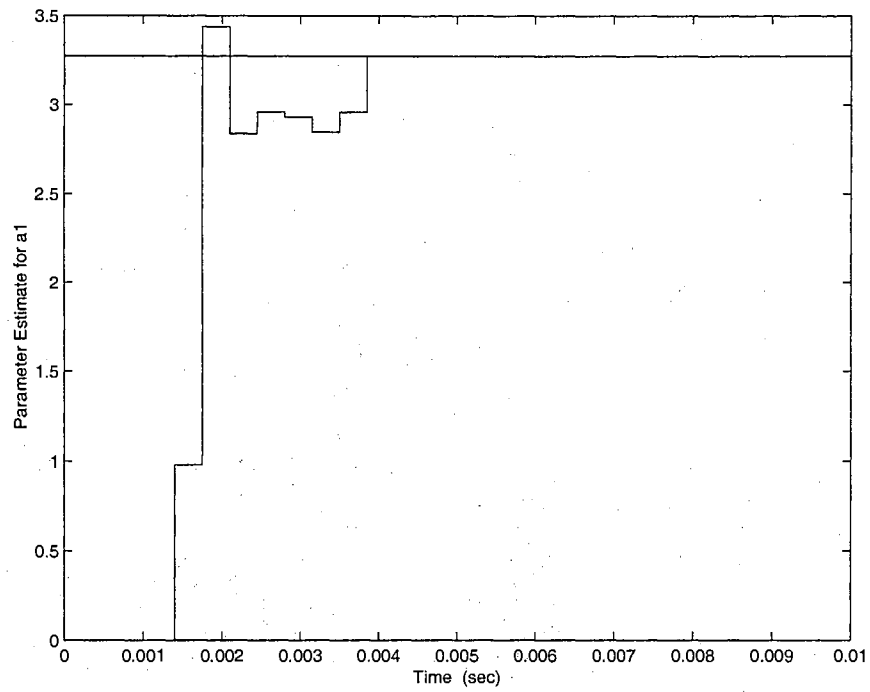


Figure 7.6: Parameter Estimate of \bar{a}_1

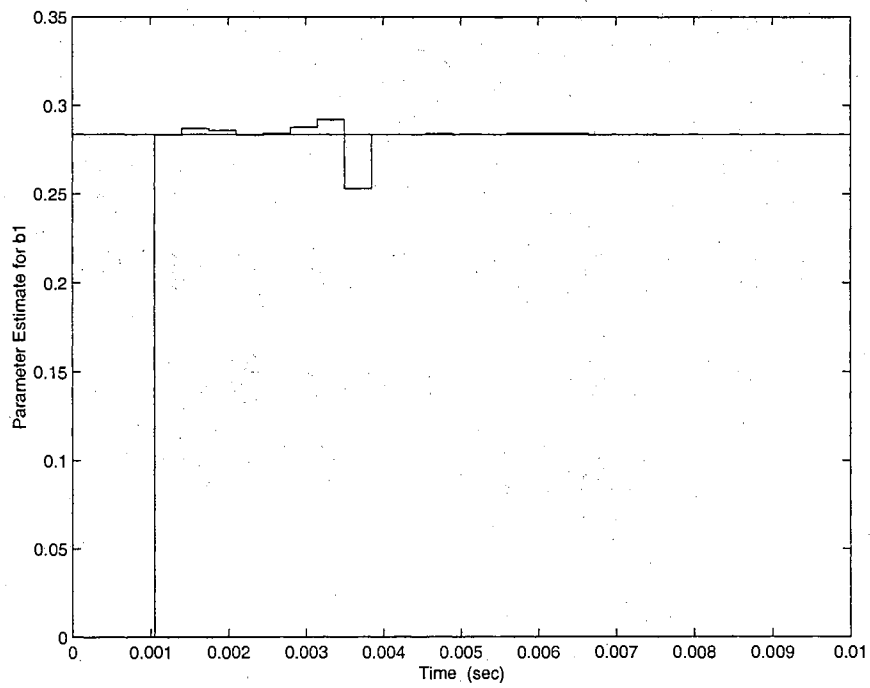


Figure 7.7: Parameter Estimate of \bar{b}_1

Parameter	Actual	Estimate	% Error
\bar{a}_1	3.2729e+00	3.2469e+00	7.9439e-01
\bar{a}_2	-4.2029e+00	-4.2027e+00	5.5667e-03
\bar{a}_3	2.5870e+00	2.5870e+00	3.0577e-05
\bar{a}_4	-6.5703e-01	-6.5703e-01	1.3305e-06
\bar{b}_1	2.8360e-01	2.8360e-01	9.8869e-06
\bar{b}_2	2.8040e+00	2.8040e+00	7.8697e-07
\bar{b}_3	2.5735e+00	2.5735e+00	4.1001e-07
\bar{b}_4	2.2019e-01	2.2019e-01	1.2167e-05

Table 7.1: Parameter Values and Estimates

Figure 7.7. The same results are obtained for those parameter estimates of \bar{B}_d as obtained for \bar{A}_d . There is no estimate oscillation after the initial transient, and convergence is quickly established and maintained.

The actual parameter values, along with the final parameter estimates are shown in Table 7.1, where the final percentage errors of the estimates are also calculated. The largest error occurs in the estimation of \bar{a}_1 , which is 0.8%. All other parameter estimates are accurate to at least 4 significant figures.

7.4 Effects of Noise on the APOS

For this implementation, two sets of simulations are run on the IBM HDA. The first set of simulations involves the implementation of the DAO where the observer parameters λ and e_{thresh} are tuned by the genetic algorithm with the assumption of a noise-free signal. These same observer parameters are used in the second set of simulations, but noise simulated by a normally distributed random number generator with a standard deviation of 5% of a track width is applied to the disk drive. The second set of simulations involves the disk drive in the presence of the same noise as in the previous set of simulations, but the DAO observer parameters are re-tuned with the genetic algorithm to

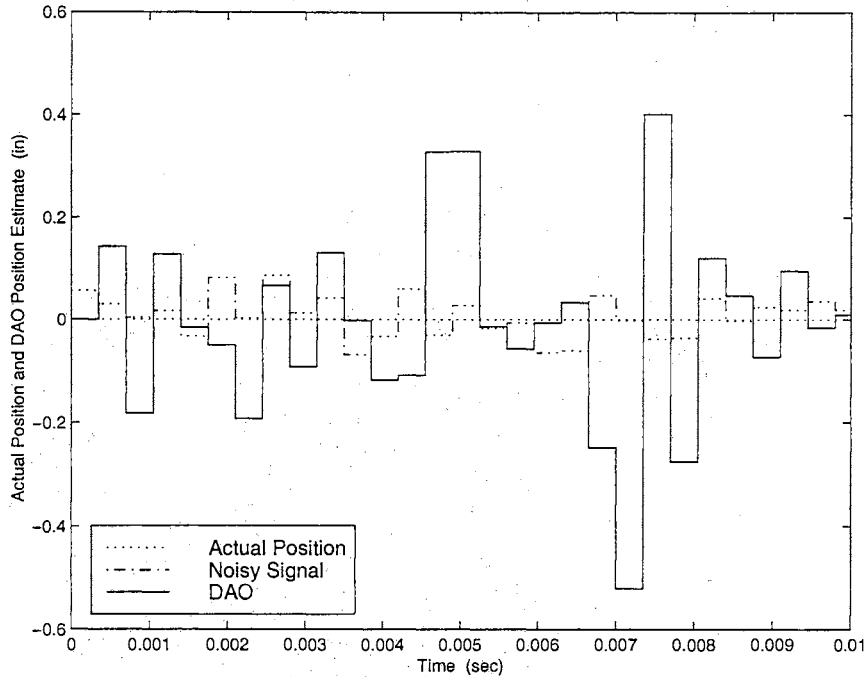


Figure 7.8: Actual Output, Signal With Noise, and DAO Estimate

include the added noise.

The actual output and output signal with noise is shown in Figure 7.8, along with the DAO position output estimate. It should be noted from this figure that the noise significantly overpowers the true position signal. It is shown that the DAO is significantly affected by the noise, which is not surprising, considering the level of the noise compared to the true position output. As can be seen more clearly in Figure 7.9, the output estimate error is notably larger than the level of even that of the noise in the measurement signal. This same characteristic also occurs in the velocity error estimate, as shown in Figure 7.10. In Figures 7.11 and 7.12, one sees that because the DAO estimates are significantly inaccurate, the output and velocity estimates of the Fast Observer System are, in turn, significantly oscillatory and unreliable, but they are stable.

Sample responses of the parameter estimates for \bar{A}_T and \bar{B}_T for the case of added noise and without GA re-tuning are shown in Figure 7.13 and 7.14, where the parameter estimate responses of \bar{a}_1 and \bar{b}_1 , respectively, are given. (The reader is referred to the appendix for the responses of the remaining parameters.) All parameter estimates of \bar{A}_T are significantly inaccurate and do not even

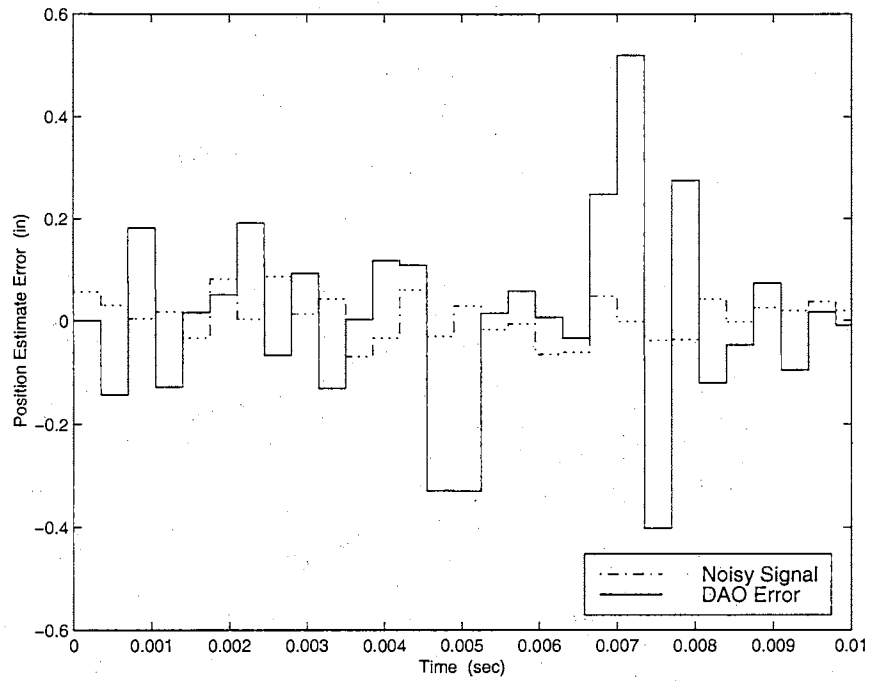


Figure 7.9: DAO Output Estimate Error (With Noise)

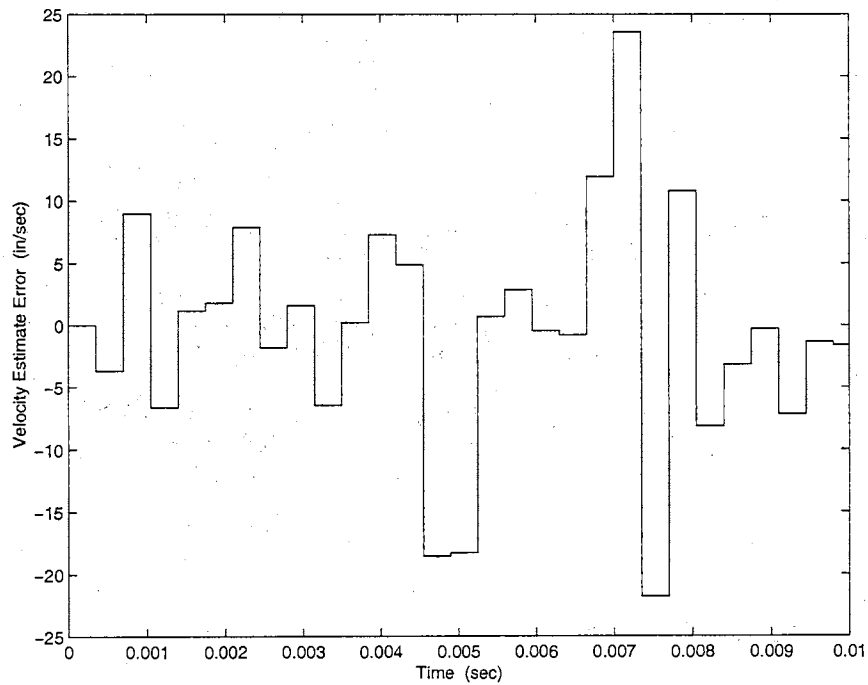


Figure 7.10: DAO Velocity Estimate Error (With Noise)

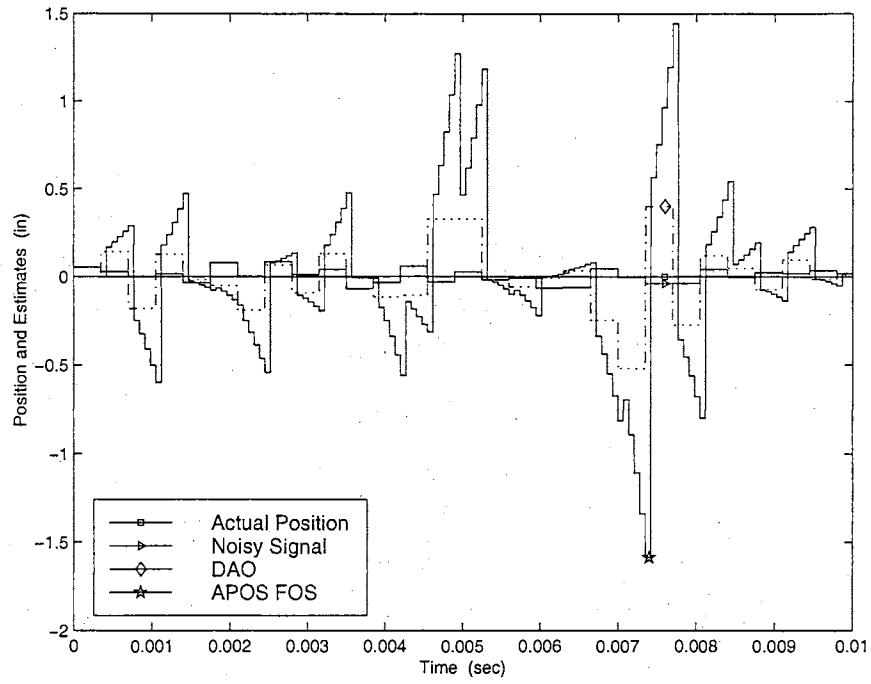


Figure 7.11: Actual Output and Observer Estimate (With Noise)

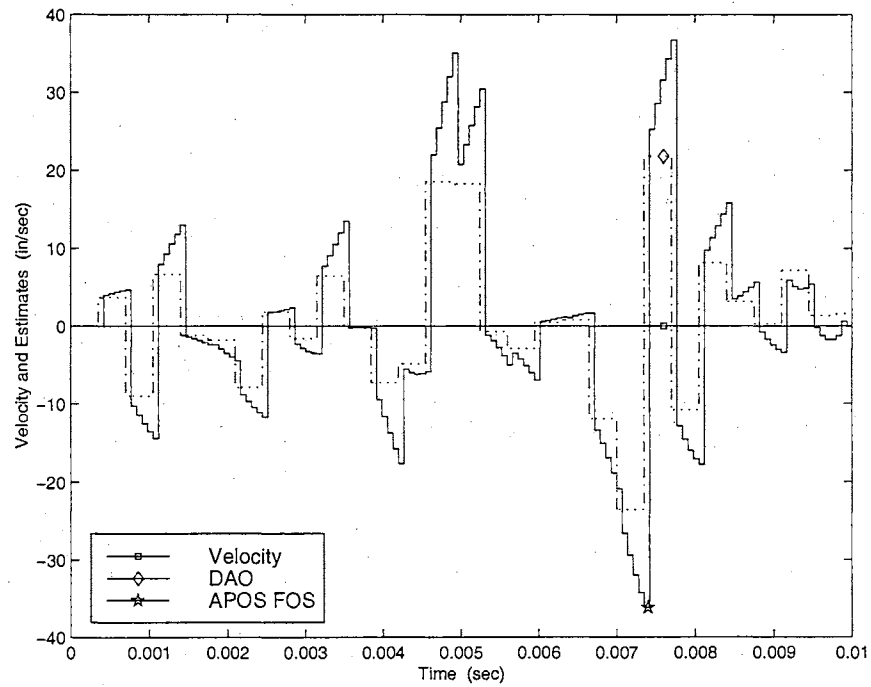


Figure 7.12: Actual Velocity and Observer Estimate (With Noise)

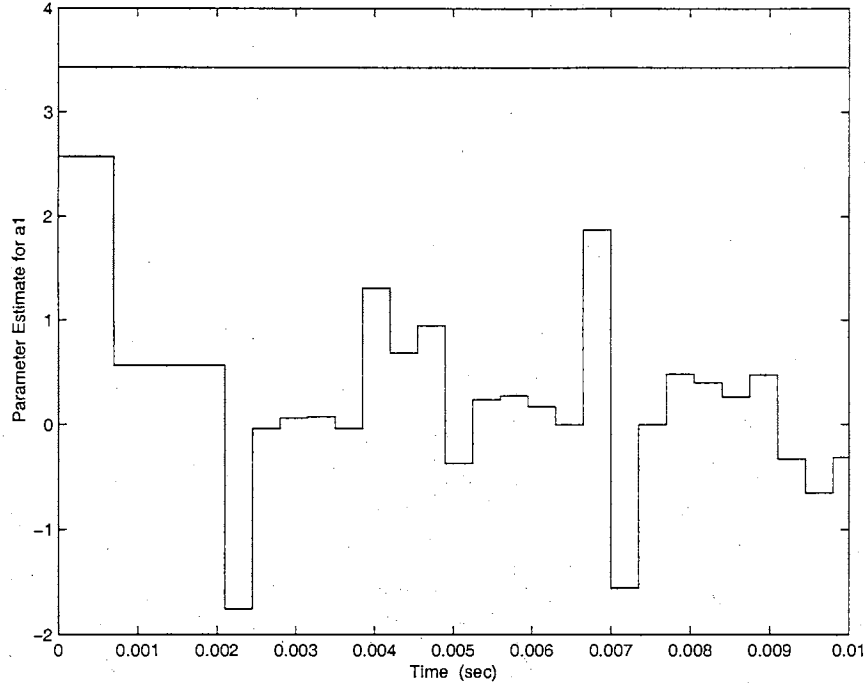


Figure 7.13: Parameter Estimate of \bar{a}_1 (With Noise)

oscillate about its true values. The parameter estimates of \bar{B}_T oscillate about their actual values, although the magnitudes of oscillation are also significant.

It is seen that noise does greatly affect the final parameter estimates. The actual parameter values, along with the final parameter estimates are shown in Table 7.2, where the final percentage errors of the estimates are also calculated. Here, the parameters estimate errors are as large as $7 \times 10^8\%$.

In the second set of simulations, where the DAO is re-tuned by the GA to run in the presence of the noise, the cost function \mathcal{J} is used such that

$$\begin{aligned} \mathcal{J} &= \sum_{i=0}^{T_{max}} [e_y^T(i)e_y(i) + e_v^T(i)e_v(i)] \\ e_y(i) &= y(i) - \hat{y}(i) \\ e_v(i) &= [e_y(i) - e_y(i-1)]/\Delta t \end{aligned} \quad (7.10)$$

Note that this form of the cost function is used so as to fairly correlate the effects of the noise on the velocity output without assuming that the velocity output is known. The result of the

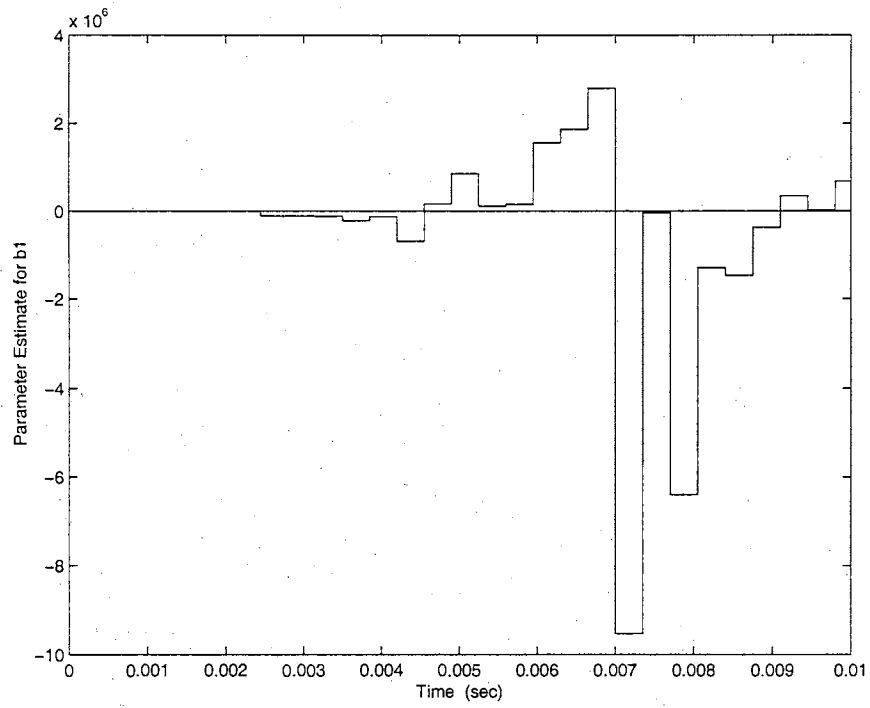


Figure 7.14: Parameter Estimate of \bar{b}_1 (With Noise)

Parameter	Actual	Estimate	% Error
a_1	3.4309e+00	-3.4237e-01	1.0998e+02
a_2	-4.7318e+00	-3.0733e-01	9.3505e+01
a_3	3.1708e+00	-1.9957e-01	1.0629e+02
a_4	-8.6994e-01	3.5842e-01	1.4120e+02
b_1	2.9925e-01	6.7951e+05	2.2707e+08
b_2	3.1179e+00	-1.1369e+06	3.6462e+07
b_3	3.0303e+00	-3.7333e+05	1.2320e+07
b_4	2.7515e-01	1.8997e+06	6.9044e+08

Table 7.2: Parameter Values and Estimates (With Noise)

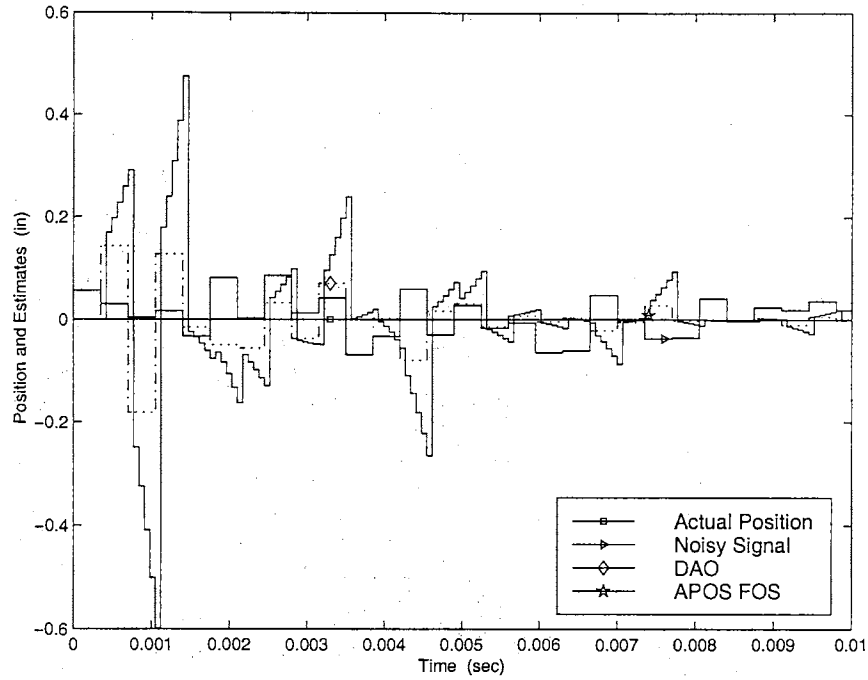


Figure 7.15: Actual Output and Observer Estimate (With Noise and Re-tuning)

GA simulation using this form of the cost function yield values of 1 and $1.65e-2$ for λ and e_{thresh} , respectively.

As can be seen from Figure 7.15 and 7.16, the DAO and FOS estimates still remain relatively inaccurate, as expected, due to the significance of the noise level compared to the actual output position. However, the estimates do appear to start to converge to the actual position and velocity values, respectively, as the oscillatory behavior does decrease significantly with time. As seen in Figures 7.17 and 7.18, where the position and velocity estimate errors are shown, respectively, there is large oscillation in the initial transient phase, but the oscillations decrease dramatically with time. In fact, the magnitude of oscillation of the position error decreases to below the magnitude of that of the noise in the output signal.

Sample responses of the parameter estimates for A_T and B_T for the case of added noise and with GA re-tuning are shown in Figure 7.19 and 7.20, where the parameter estimate responses of \bar{a}_1 and \bar{b}_1 , respectively, are given. (The reader is referred to the appendix for the responses of the remaining parameters.) The responses are more or less the same as those without the retuning of

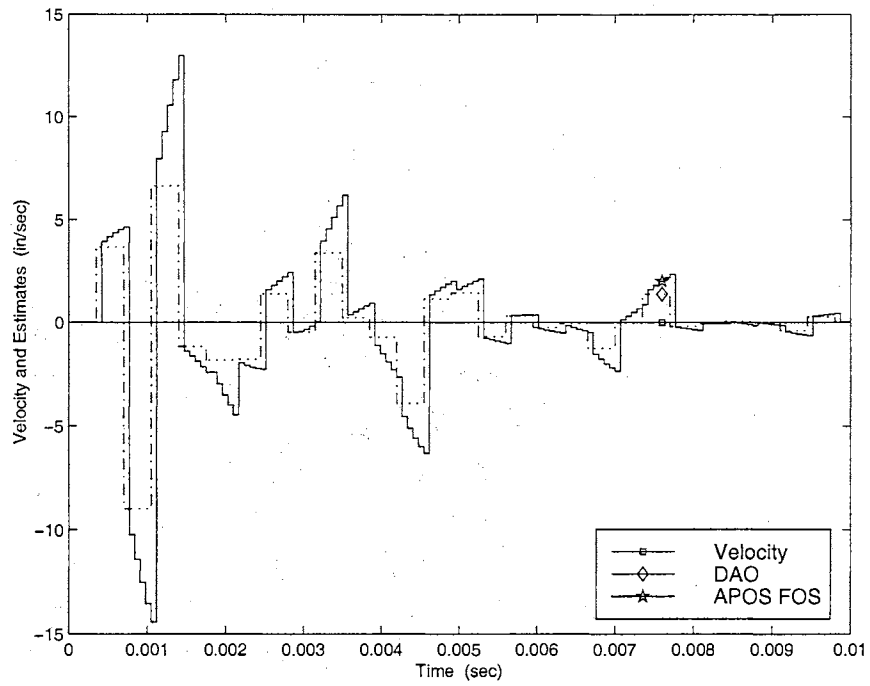


Figure 7.16: Actual Velocity and Observer Estimate (With Noise and Re-tuning)

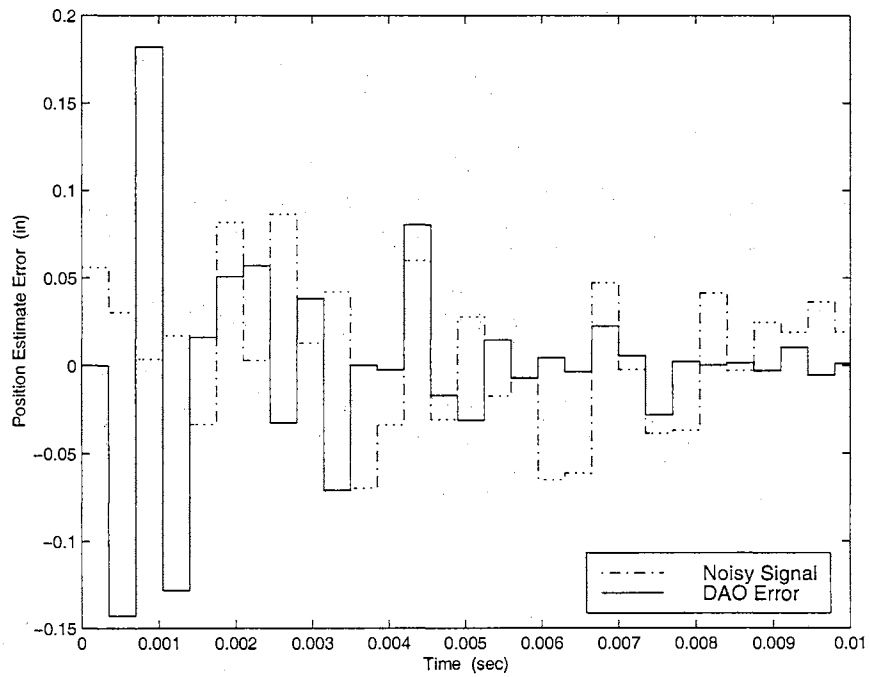


Figure 7.17: Output Estimate Error (With Noise and Re-tuning)

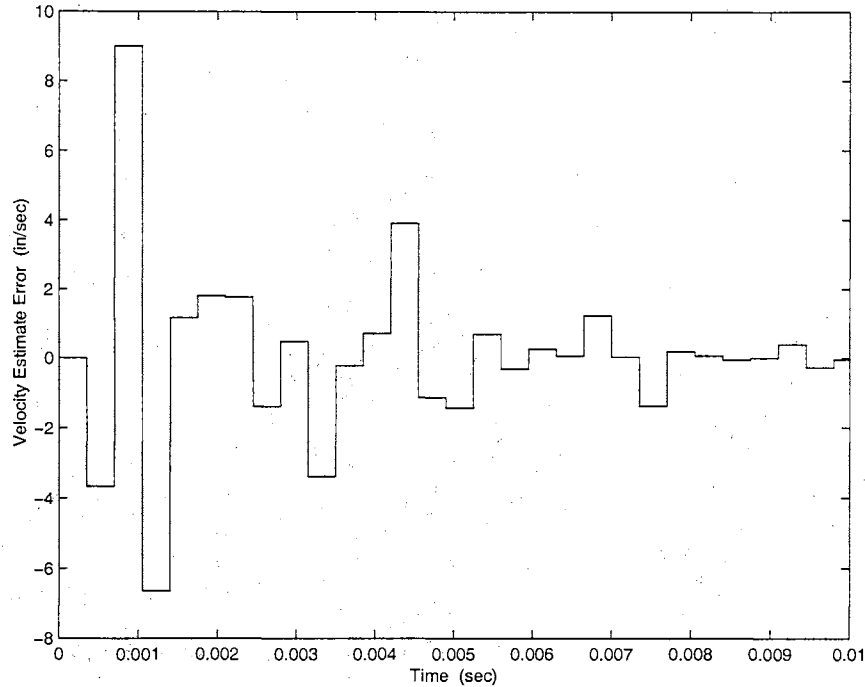


Figure 7.18: Velocity Estimate Error (With Noise and Re-tuning)

the DAO, except that there is less oscillation present and the overall response of the DAO is slower. This, of course, is expected, as λ is now set to 1.

The actual parameter values, along with the final parameter estimates are shown in Table 7.3, where the final percentage errors of the estimates are also calculated. The errors are significantly less than that without retuning but are still unacceptably high, as high as over 5000%.

To summarize, it appears that the Discrete Adaptive Observer is very susceptible to noise, as expected. For one, the noise levels in these simulations are significant compared to the actual position output. In addition, the DAO assumes that any perturbations in the system are assumed to be a result of system parameter changes, thereby forcing itself to try to adapt to the noise. However, with the consideration of the comparatively significant noise levels, the APOS state estimates are not too intolerable. It is also seen that re-tuning the DAO with the use of the genetic algorithm improves overall state and parameter estimates significantly. In fact, the DAO position estimate appears to converge to the actual position value, and its position estimate error decreases to levels below that of the corruptive noise. With or without retuning with the implemented genetic algorithm, however,

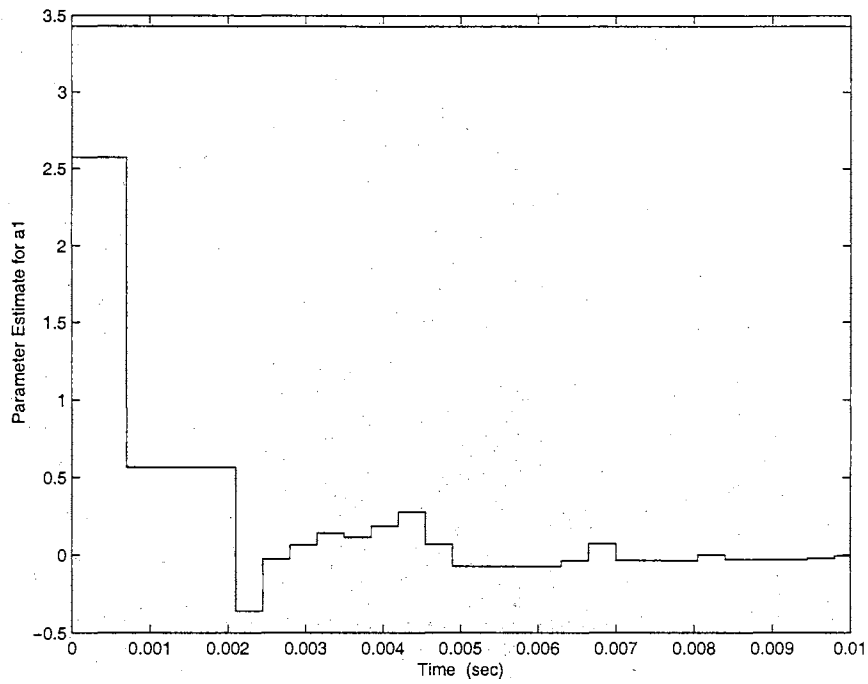


Figure 7.19: Parameter Estimate of \bar{a}_1 (With Noise and Re-tuning)

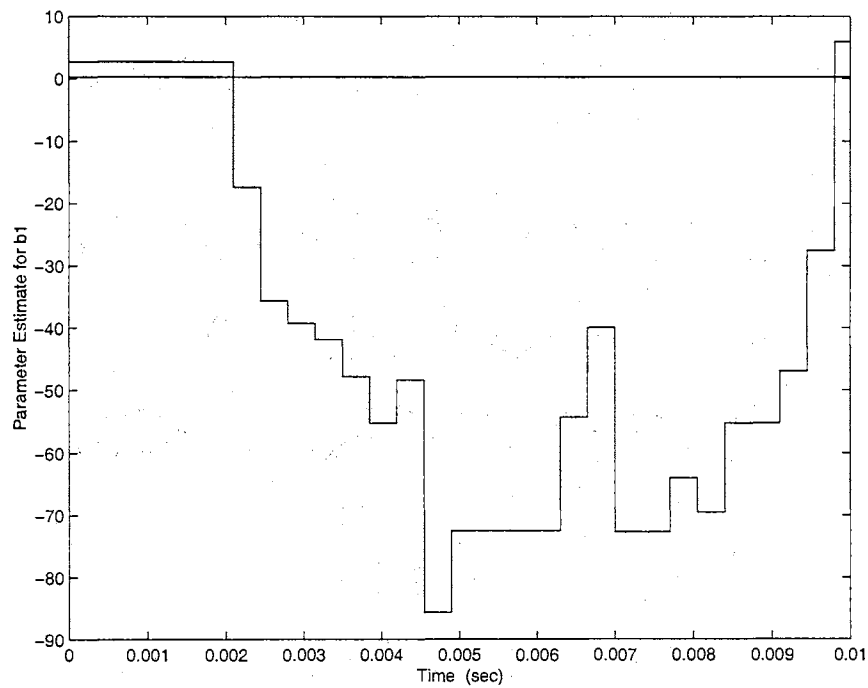


Figure 7.20: Parameter Estimate of \bar{b}_1 (With Noise and Re-tuning)

Parameter	Actual	Estimate	% Error
a_1	3.4309e+00	-3.0645e-02	1.0089e+02
a_2	-4.7318e+00	1.2475e-01	1.0264e+02
a_3	3.1708e+00	-1.0127e-01	1.0319e+02
a_4	-8.6994e-01	1.0971e-01	1.1261e+02
b_1	2.9925e-01	5.9824e+00	1.8991e+03
b_2	3.1179e+00	-9.8094e+00	4.1461e+02
b_3	3.0303e+00	2.0018e+01	5.6058e+02
b_4	2.7515e-01	1.4526e+01	5.1794e+03

Table 7.3: Parameter Values and Estimates (With Noise and Re-tuning)

the parameter estimates continue to be very inaccurate, which is expected, as any such disturbances that do exist are interpreted by the DAO as system parameter changes.

7.5 Summary of Results

Simulation results show that the genetic algorithm used is effective in automating the process of choosing optimal observer gains off-line for the Discrete Adaptive Observer. Accurate estimation of state outputs and especially of system parameters is achieved from the DAO as a result of the GA chosen optimal observer gains. It is found, however, that the DAO is greatly susceptible to noise, as it interprets any noise and other disturbances as changes in system parameters. However, the state estimates are significantly improved after retuning the GA with the presence of noise, although the parameter estimates still continue to be inaccurate.

Chapter 8

Damping Ratio and Aliased Resonant Frequency

Approximation for the IBM HDA

The Parallel Observer System (POS) is presented in attempts to help solve the dual problem of needing an estimation technique of a multirate architecture while taking into account the effects of aliased output measurements. Eventually, a scheme of not only adequate state estimation is desired, but accurate parameter estimation is also desired so as to estimate any modeling errors and/or parameter changes, especially those of aliased resonance frequencies.

In this chapter, the Discrete Adaptive Observer is applied to the same form of the IBM Head/Disk Assembly System as the last chapter to exemplify how system parameters, namely aliased resonant frequencies, may be extracted from the DAO. A neural network is trained for function approximation using the system parameter estimates obtained by the DAO and by using a priori information about the disk drive model. This neural network is primarily developed in attempts to uncover estimates of aliased resonant frequencies and of damping ratios in the disk drive system.

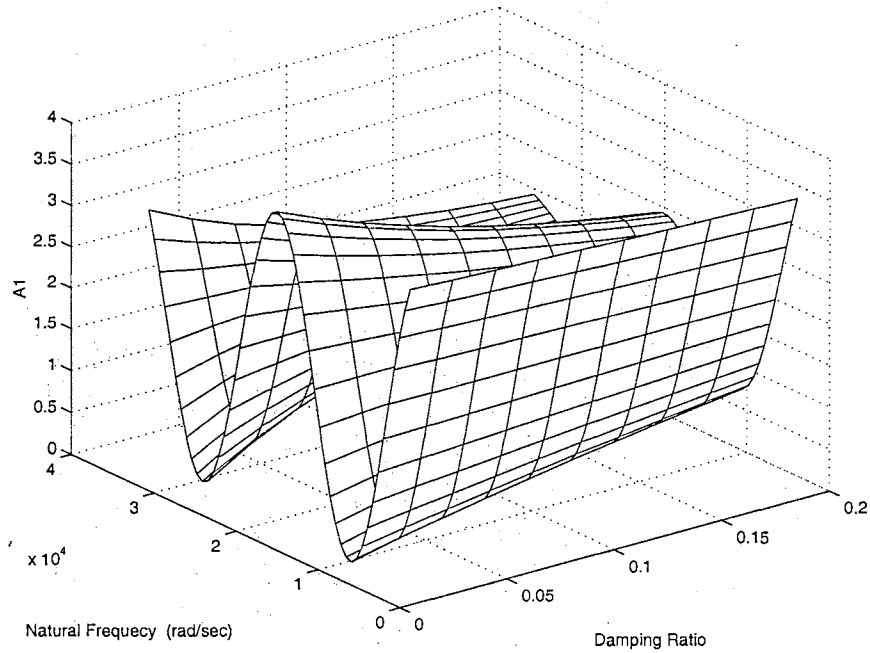


Figure 8.1: Parameter \bar{a}_1 as a function of ω_n and ζ

8.1 Uniqueness Relationship of the DAO System Parameters

A preliminary feasibility analysis is performed to determine whether the DAO system parameter values and the values of the resonant frequency and damping ratio share a unique relationship. Using a scaled model of Equation (6.12) with varying values of ζ and ω_n and the transformation relations in Equations (5.5) through (5.7), the relationships between the DAO parameters and the system damping ratios and natural frequencies are determined.

Figures 8.1 through 8.4 show the DAO parameters \bar{a}_1 through \bar{a}_4 as a function of the IBM HDA natural frequency and damping ratio. Likewise, Figures 8.5 through 8.8 show the DAO parameters \bar{b}_1 through \bar{b}_4 as a function of the disk drive natural frequency and damping ratio. As one may observe from these plots, using the relationships of each of the DAO system parameters with the corresponding values of natural frequency ω_n and damping ratio ζ , it may be possible to construct and train a neural network to approximate the resulting ω_n and ζ , given accurate DAO system parameter estimates.

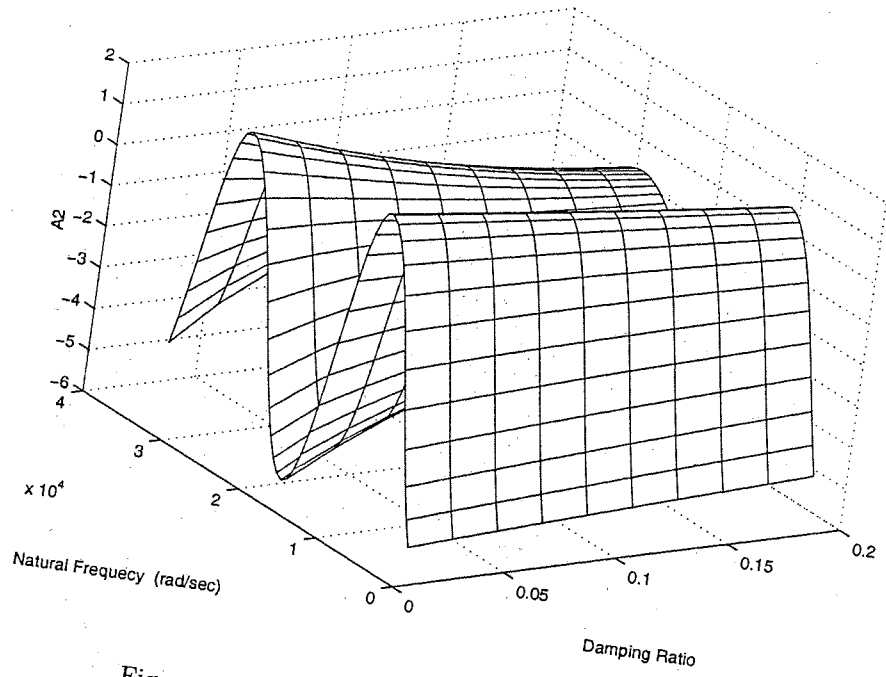


Figure 8.2: Parameter \bar{a}_2 as a function of ω_n and ζ

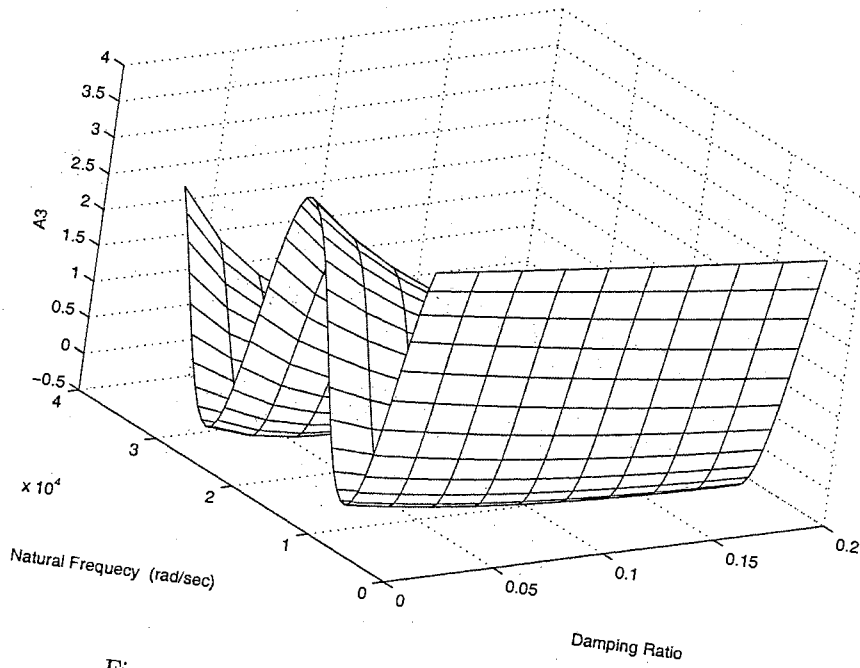


Figure 8.3: Parameter \bar{a}_3 as a function of ω_n and ζ

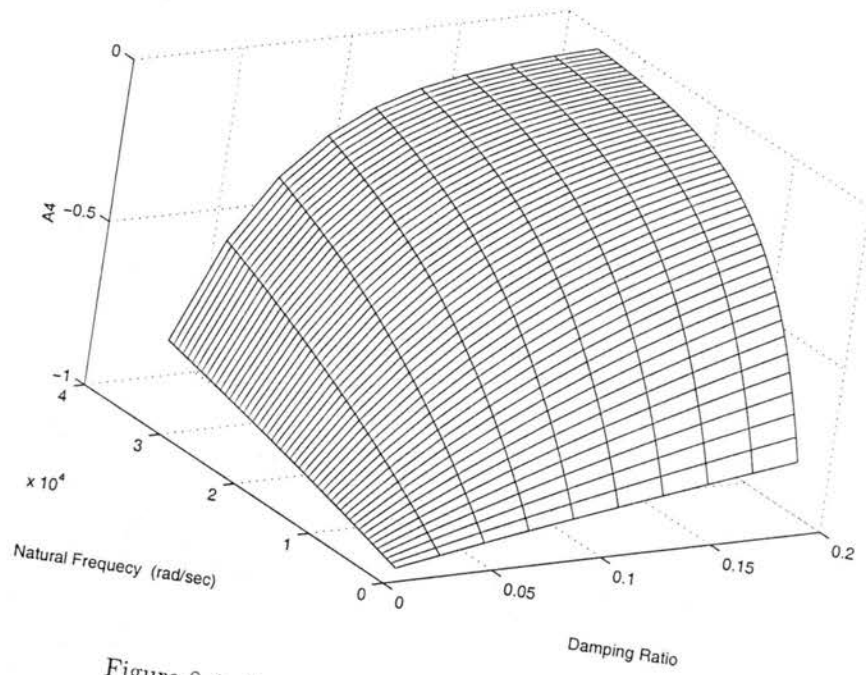


Figure 8.4: Parameter \bar{a}_4 as a function of ω_n and ζ

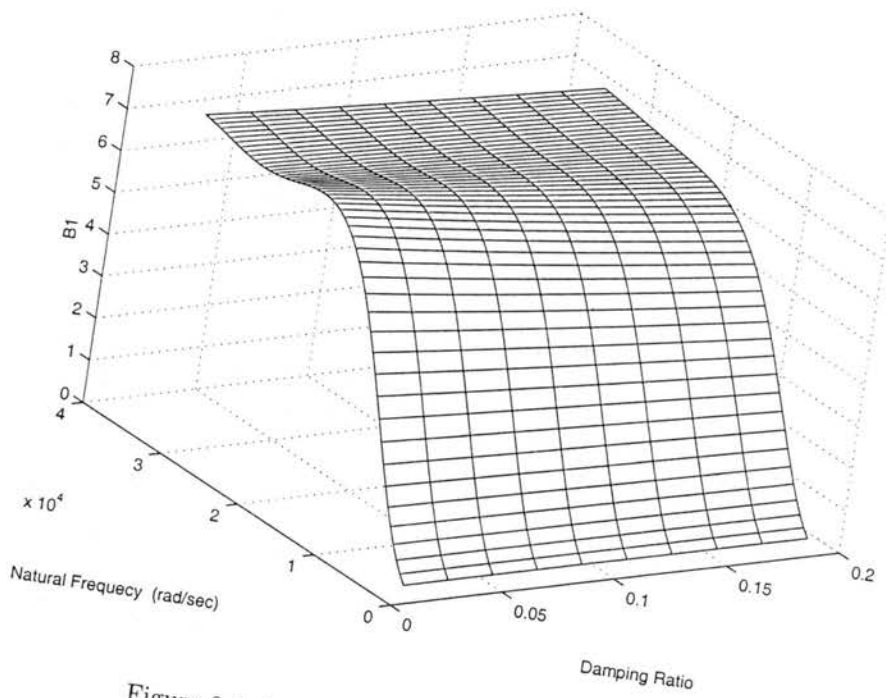


Figure 8.5: Parameter \bar{b}_1 as a function of ω_n and ζ

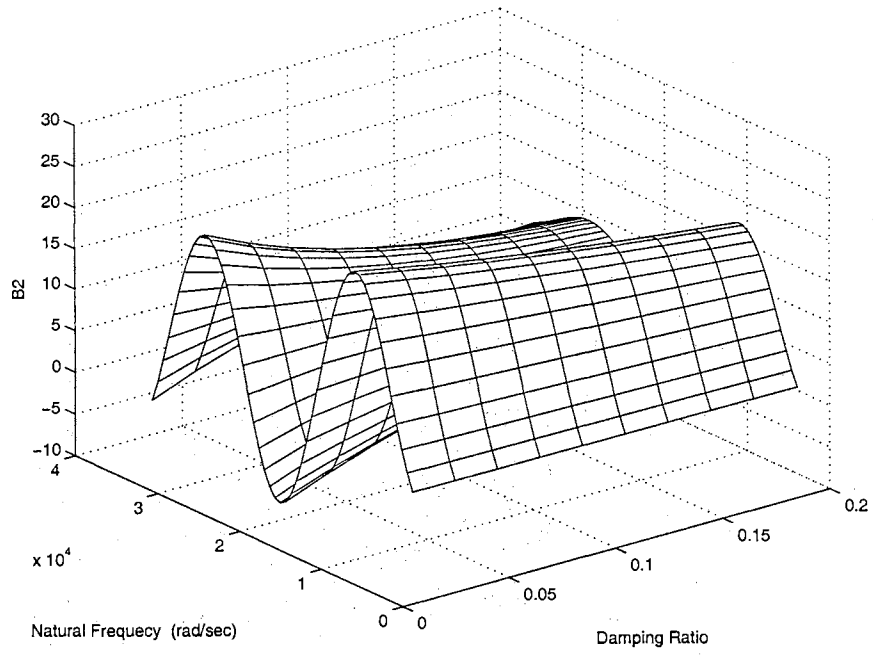


Figure 8.6: Parameter \bar{b}_2 as a function of ω_n and ζ

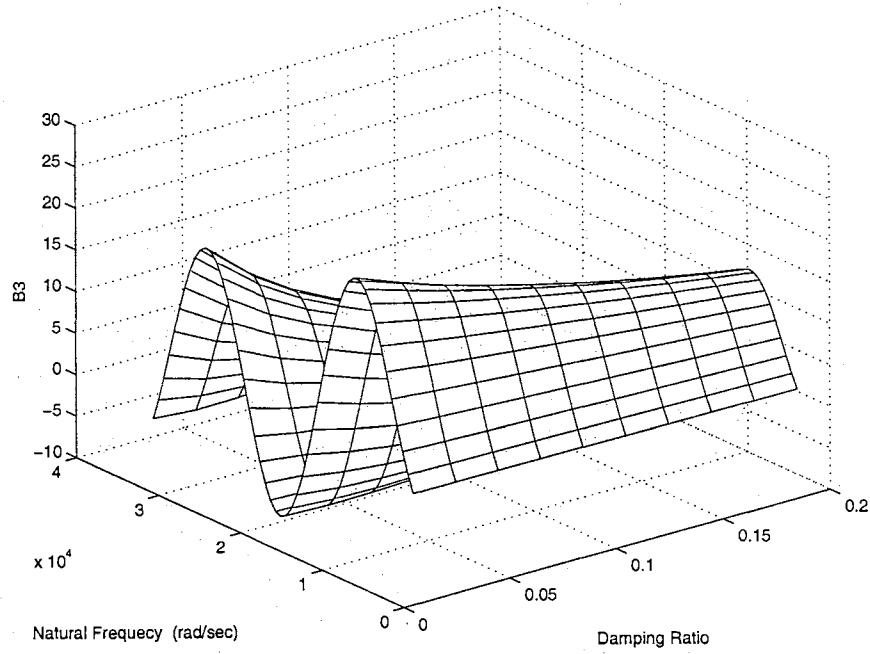


Figure 8.7: Parameter \bar{b}_3 as a function of ω_n and ζ

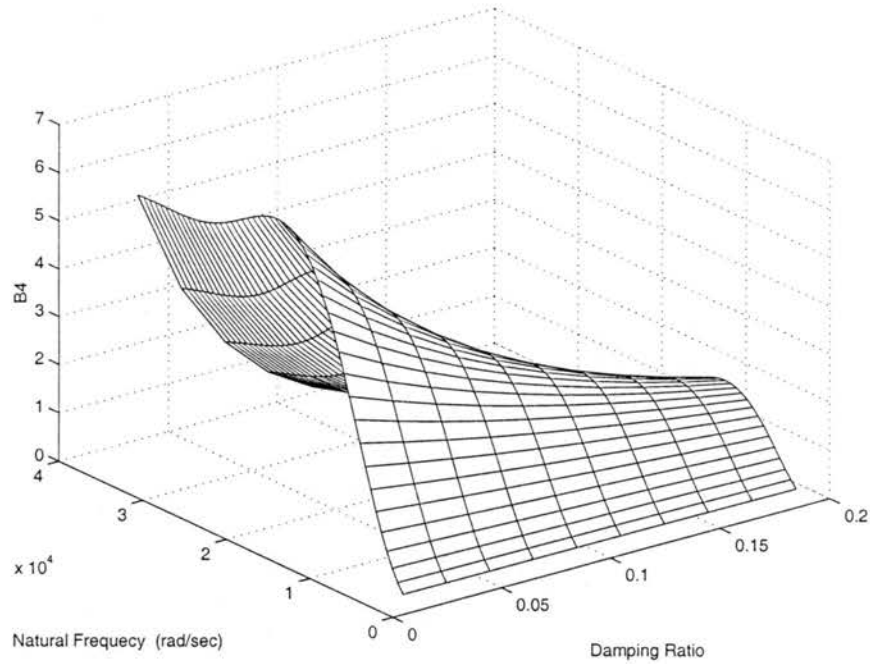


Figure 8.8: Parameter \bar{b}_4 as a function of ω_n and ζ

8.2 DAO Parameter Estimates at Aliased Resonant Frequencies

For purposes of performance analysis of the extraction of aliased resonant frequencies, two cases of natural frequencies are applied to the IBM HDA system. For both cases, the same sampling frequency is maintained, resulting in a fixed Nyquist frequency of $1.4286 \times 10^3 \text{ Hz}$ ($8.976 \times 10^3 \text{ rad/s}$). In the first case, the applied resonant frequency is slightly aliased at $1.5438 \times 10^4 \text{ Hz}$ ($9.7 \times 10^3 \text{ rad/s}$). In the second case, the applied resonant frequency is moderately aliased at $2.0690 \times 10^3 \text{ Hz}$ or ($1.3 \times 10^4 \text{ rad/s}$). For both cases, the damping ratio is fixed at 0.0995.

For these simulations, it is assumed that the true resonant frequency and damping ratio is approximately known. This assumption is valid, otherwise, it would be impossible to differentiate the actual frequency from the infinite number of possible aliasing frequencies. As a result, this information is used for simulations in this chapter. More specifically, the actual values of \bar{A}_T and \bar{B}_T are calculated with the knowledge of the true values of ζ and ω_n . The parameter estimates of

\bar{A}_T and \bar{B}_T are initialized at 75% and 80%, respectively, of their actual values such that at $t=0$,

$$\hat{p}(0) = \begin{bmatrix} 4.3377e - 01 \\ 9.4900e - 01 \\ -2.7691e - 01 \\ -3.8163e - 01 \\ 3.6258e + 00 \\ 1.6730e + 01 \\ 1.2810e + 01 \\ 2.3074e + 00 \end{bmatrix}$$

In addition, for fairness to the DAO performance simulation, it is assumed that only the resonant frequency and the damping ratio are in question and that all other system parameters are well known. Second, the DAO is re-tuned to the case of a natural frequency of $9.7 \times 10^3 \text{ rad/s}$ and a damping ratio of 0.0995. The genetic algorithm parameters are re-tuned by changing the initial population and the mutation rates. The resulting λ and e_{thresh} values are 0.49476 and 0, respectively. The same observer parameters are used for both sets of aliased natural frequencies. Furthermore, as in the previous chapter, the DAO is performed on the scaled version of the IBM HDA with no added noise.

8.2.1 Case I: Slightly Aliased Resonant Frequency

Table 8.1 shows the results of the DAO parameter estimates for the case of a slightly aliased resonant frequency of $\omega_n = 9.7 \times 10^3 \text{ rad/s}$. Despite the slight aliasing, the DAO is still able to estimate the system parameters well. In fact, the estimate errors are well under 1%, except for the estimate error of \bar{a}_1 , which is at 4.2%.

The extreme accuracy of the DAO parameter estimates under the condition of a slightly aliased resonant frequency may seem unexpected. However, as one may recall, the knowledge of the approximate values of the the aliased resonant frequency and the damping ratio is incorporated into the DAO by initializing the parameter estimates to the corresponding parameter values using these

Parameter	Actual	Estimate	% Error
\bar{a}_1	6.1303e-01	5.8706e-01	4.2357e+00
\bar{a}_2	1.2650e+00	1.2652e+00	1.4864e-02
\bar{a}_3	-3.6922e-01	-3.6923e-01	2.6559e-03
\bar{a}_4	-5.0883e-01	-5.0881e-01	4.2006e-03
\bar{b}_1	4.5322e+00	4.5322e+00	5.9031e-05
\bar{b}_2	2.0912e+01	2.0912e+01	7.1554e-04
\bar{b}_3	1.6013e+01	1.6012e+01	3.6818e-03
\bar{b}_4	2.8842e+00	2.8841e+00	3.0130e-03

Table 8.1: Parameter Values and Estimates for $w_n = 9.7e3$ rad/sec

approximate values. In addition, the DAO is solely driven by a parameter update law. Under this condition, it is logical that, despite the presence of an aliased resonant frequency, the parameters converge to the system parameters corresponding to the closest aliased frequency, rather than to any other aliased frequency.

In this case the parameter estimates are initialized corresponding to the true aliased natural frequency of 9.7×10^3 rad/s, but with a 20%-25% error. Therefore, it is logical that the parameter estimates eventually converge to the estimates associated with the 9.7×10^3 rad/s frequency, as opposed to an aliased frequency of 19.4×10^3 rad/s, for example, or any other integer multiple of the actual frequency. Of course, this may not be the general case if the actual parameter values corresponding to two sets of aliased frequencies are similar.

8.2.2 Case II: Moderately Aliased Resonant Frequency

Table 8.2 shows the results of the DAO parameter estimates for the case of a moderately aliased resonant frequency of $\omega_n = 1.3 \times 10^4$ rad/s. Despite the moderate aliasing, the DAO is still able to estimate the system parameters well. In fact, the estimate errors are well under 1%, except for the estimate error of \bar{a}_1 , which is at 1.5%.

Parameter	Actual	Estimate	% Error
\bar{a}_1	1.7661e+00	1.7401e+00	1.4722e+00
\bar{a}_2	-9.3654e-01	-9.3630e-01	2.5500e-02
\bar{a}_3	5.7481e-01	5.7481e-01	1.5391e-03
\bar{a}_4	-4.0435e-01	-4.0434e-01	9.2619e-04
\bar{b}_1	6.0562e+00	6.0562e+00	3.9058e-05
\bar{b}_2	1.1587e+01	1.1587e+01	2.9606e-05
\bar{b}_3	4.3091e+00	4.3091e+00	3.6746e-04
\bar{b}_4	3.1332e+00	3.1331e+00	1.1682e-03

Table 8.2: Parameter Values and Estimates for $w_n = 1.3e4$ rad/sec

As in Case I of slightly aliased resonant frequency, it is seen that the DAO, after initializing the parameter estimates accordingly, is very accurate in estimating the true parameters, despite the moderately aliased resonant frequency. It shows that the parameter estimates eventually converge to the estimates associated with the 1.3×10^4 rad/s frequency. Again, this may not have been the case if the actual parameter values corresponding to the actual aliased resonant frequency were similar to that of another aliased frequency.

8.2.3 Aliased Resonant Frequencies in General

For the general case of resonant frequencies ranging from 1650 rad/s to 3.35×10^4 rad/s and damping ratios ranging from 0.011 to 0.191, the reader is referred to Figures 8.9 through 8.16. In these figure, the DAO parameter estimate errors are shown as a function of the damping ratio and resonant frequency.

It can be seen that estimate errors are minimal except about half the sampling frequency and most significantly about the actual sampling frequency. At the sampling frequency, since the resonant frequency and the sampling frequency are identical, the output detects no changing response from the resonant frequency. In essence, any contributed response effected by the resonant frequency

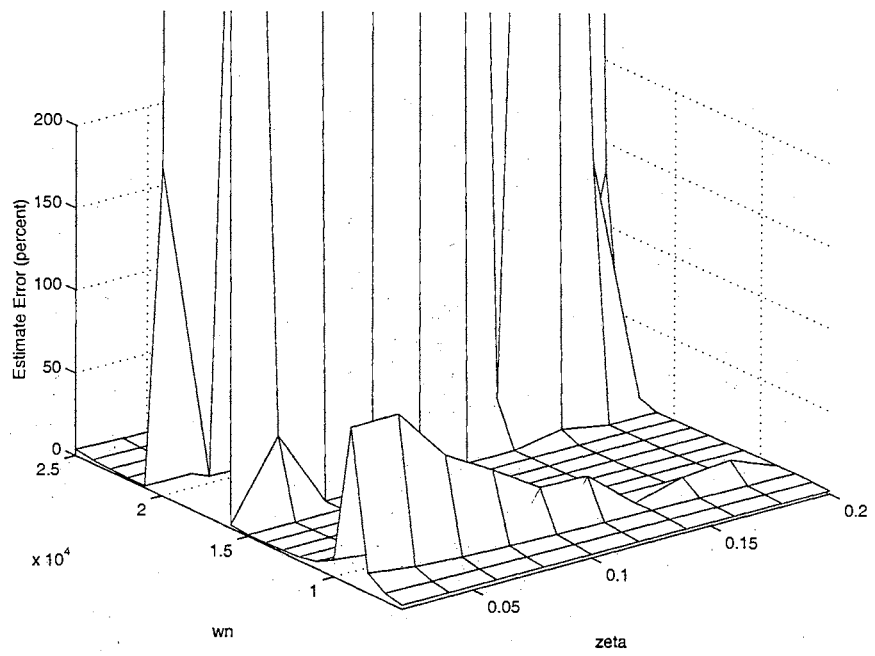


Figure 8.9: Estimate error of \bar{a}_1 as a function of ω_n and ζ

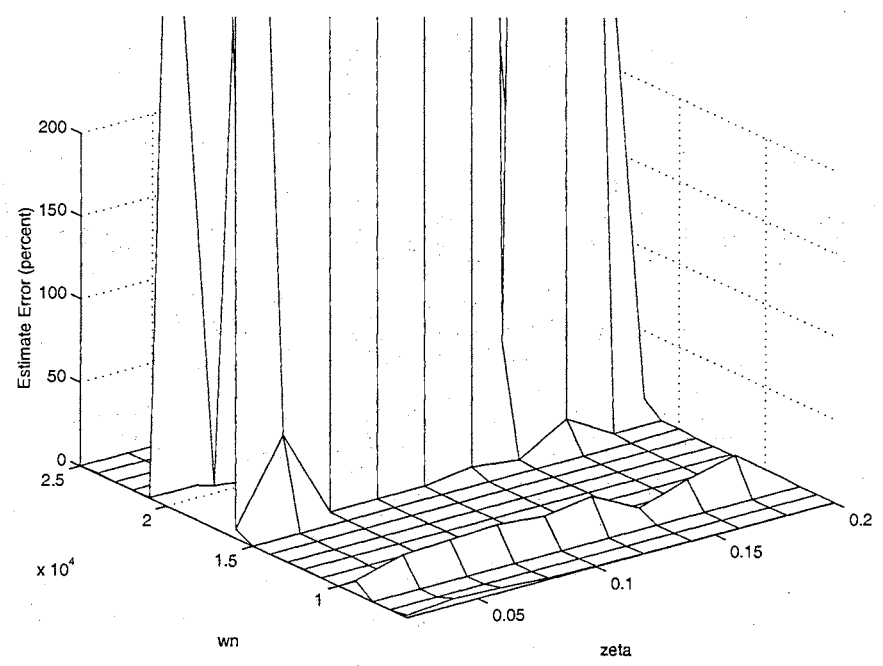


Figure 8.10: Estimate error of \bar{a}_2 as a function of ω_n and ζ

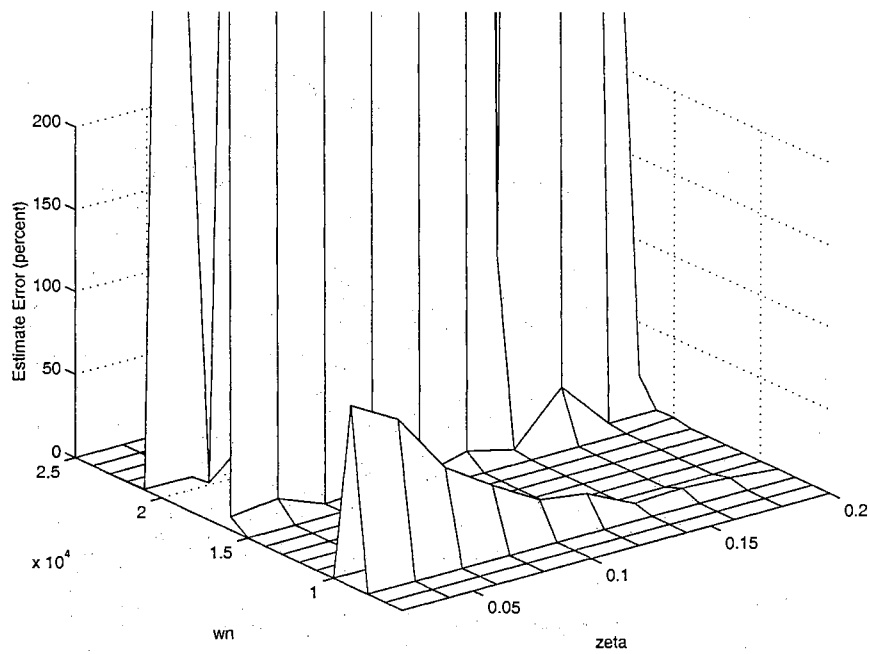


Figure 8.11: Estimate error of \bar{a}_3 as a function of ω_n and ζ

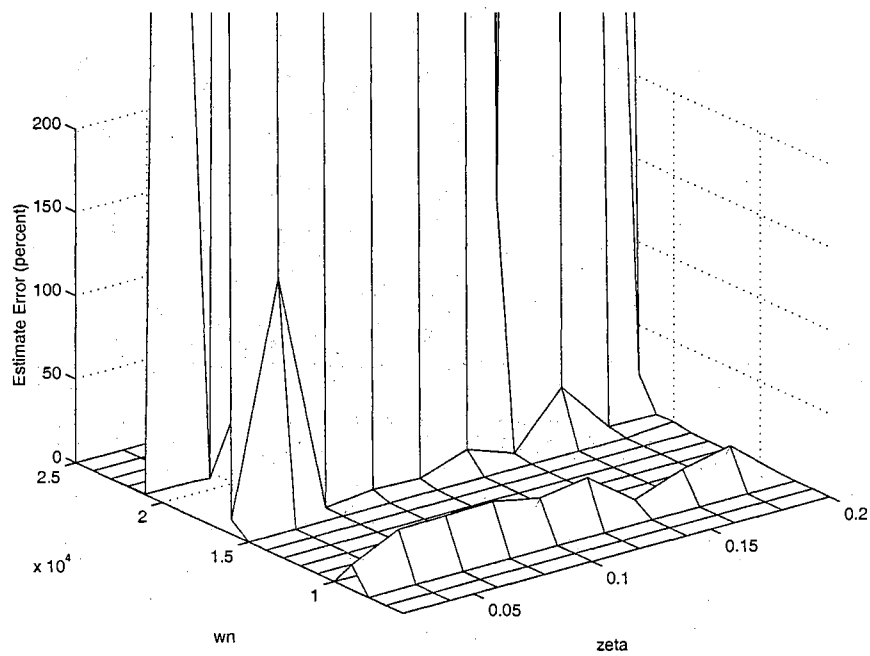


Figure 8.12: Estimate error of \bar{a}_4 as a function of ω_n and ζ

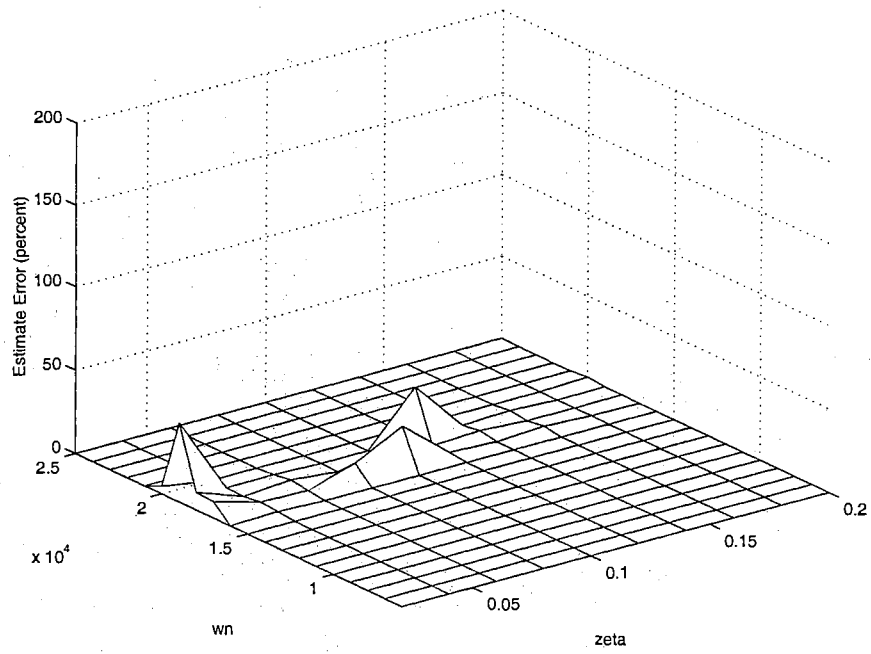


Figure 8.13: Estimate error of \bar{b}_1 as a function of ω_n and ζ

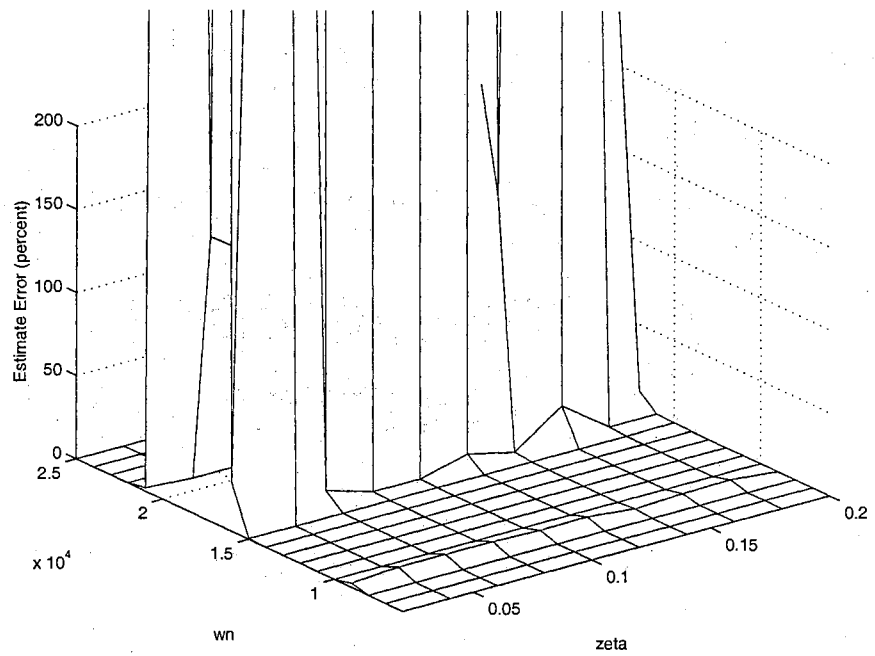


Figure 8.14: Estimate error of \bar{b}_2 as a function of ω_n and ζ

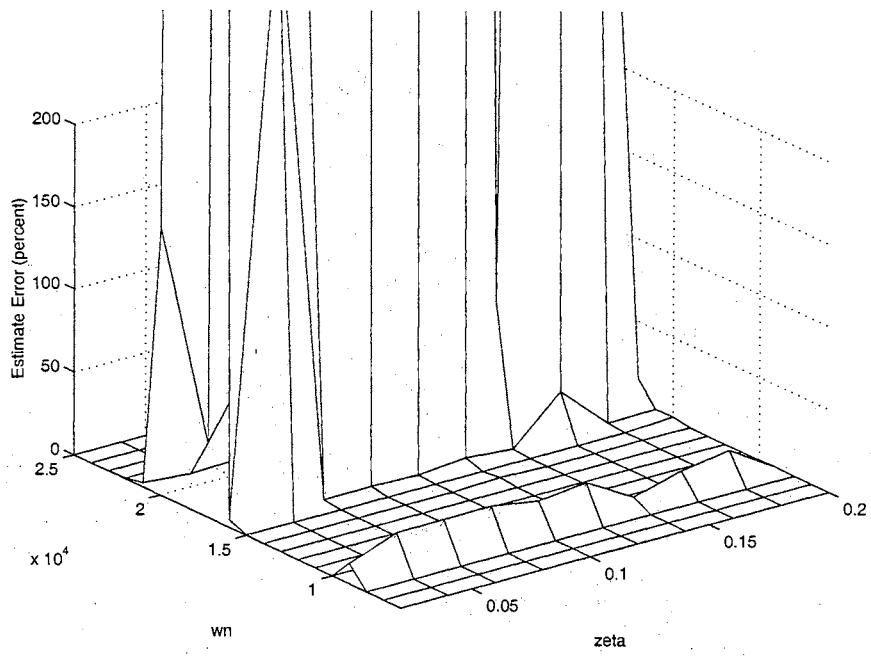


Figure 8.15: Estimate error of \bar{b}_3 as a function of ω_n and ζ

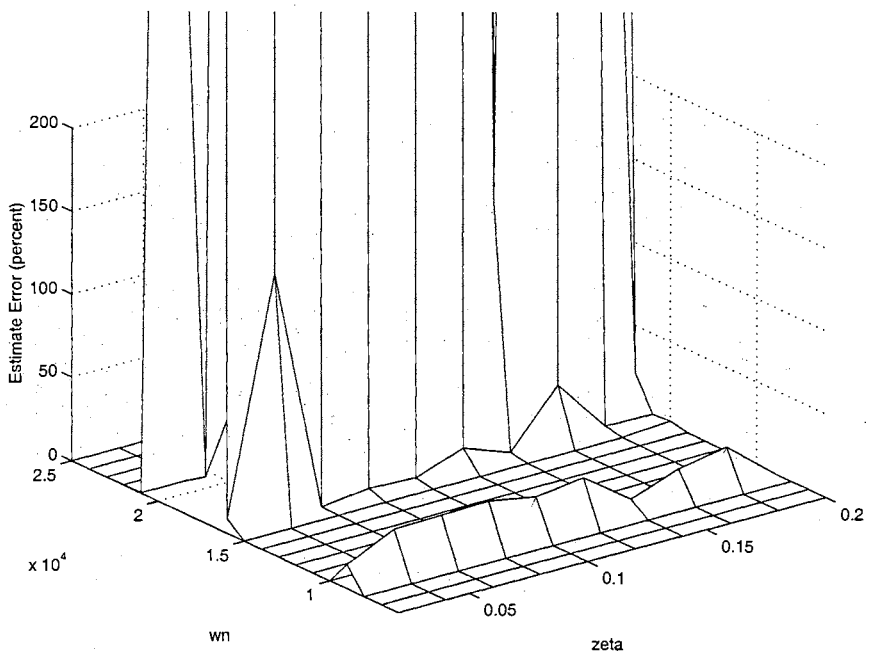


Figure 8.16: Estimate error of \bar{b}_4 as a function of ω_n and ζ

Parameter	$\omega_n = \omega_N$	$\omega_n = 2 \times \omega_N$	$\omega_n = 3 \times \omega_N$	$\omega_n = 4 \times \omega_N$	$\omega_n = 5 \times \omega_N$
\bar{a}_1	6.0613e+01	1.7328e+03	2.4387e+01	7.4993e+02	1.4274e+01
\bar{a}_2	1.6872e+01	1.1540e+03	3.5768e+01	1.8299e+04	8.0756e+01
\bar{a}_3	5.3896e+01	1.1774e+04	6.2254e+00	8.6817e+04	1.1714e+01
\bar{a}_4	2.8253e+01	3.5138e+04	4.1404e+01	2.7627e+05	4.7624e+01
\bar{b}_1	2.1258e-03	2.8729e+00	2.5352e-03	1.0138e+01	7.6416e-03
\bar{b}_2	4.3135e+00	2.3139e+04	1.0193e+01	1.9155e+04	1.0832e+01
\bar{b}_3	2.2666e+01	2.1630e+04	2.6234e+01	8.2009e+04	2.9741e+01
\bar{b}_4	2.8271e+01	3.5097e+04	4.1367e+01	2.7565e+05	4.7303e+01

Table 8.3: Percentage Estimates Errors for $\omega_n = k \times \omega_N, k = 1, 2, 3, 4, 5$

cannot be detected in the output measurement.

For the resonant frequency at the Nyquist frequency ω_N , which is half of the sampling frequency, there still is a set of intermediate points available, resulting in the presence of a small degree of resonance information to the otherwise undetectable signal. Because of this, although there is somewhat of a significant estimate error, the error is not nearly as large as that corresponding to resonant frequencies at the sampling rate or multiples thereof.

Table 8.3 shows a summary of estimate percentage errors for resonant frequencies that are an integer multiple of the Nyquist frequency ω_N . The damping ratio for these simulations are set to 0.06, as this is the damping ratio at which the highest errors occur. The simulations are performed for resonant frequencies up to and including 5 times the Nyquist frequency, as this is the limit at which the control input rate is aliased. As can be seen from the summary of errors, somewhat significant errors occur at odd integer multiples of the Nyquist frequency but very large errors occur at even integer multiples of the Nyquist frequency, which correspond to integer multiples of the sampling frequency. It is noted that these errors, although extremely large, only appear at or about these critical frequencies.

In short, except for the frequencies near or about the Nyquist frequency and especially the sam-

pling frequency, or multiples thereof, the DAO provides highly accurate parameter estimate results. Most of these parameter estimates are well under 1% error. Therefore, if a method of extracting the aliased resonant frequencies along with the damping ratio may be constructed, then using these accurate parameter estimate results with this method should yield accurate DAO estimates, as long as the resonant frequencies to be estimated are not within the range of any critical frequencies.

8.3 The Applied Neural Network

Assuming that accurate DAO parameter estimates are obtained, a neural network is constructed and trained to extract estimates of the system aliased resonant frequency and damping ratio. The neural network constructed and trained for this back propagation application is via the MATLAB Neural Network Toolbox [41]. The constructed neural network is chosen to be two layers for the sake of simplicity. There are eight inputs, determined by the eight system parameters for the fourth order IBM HDA model. The hidden layer is set to fifty neurons so as not to exceed reasonable computational limits. (It should be noted that the number of neurons should be decreased, so as to avoid over parametrization.) A log sigmoid transfer function is chosen for the first layer and a pure linear transfer function for the output layer, as suggested in [51]. There are two outputs in the neural network: one for the determination of ω_n and the other for the determination of ζ . In the training of this network, 500 training points are used, simulated in MATLAB with 300 epochs with a performance goal of 0 Mean Squared Error.

8.3.1 Training

For this network, the training occurs off-line and involves varying values of natural frequency ω_n and damping ratio ζ such that $1650 \text{ rad/s} \leq \omega_n \leq 3.35 \times 10^4 \text{ rad/s}$ and $0.011 \leq \zeta \leq 0.191$. These ranges of ω_n and ζ are used to calculate the exact system parameters, the data presented in Figures 8.1 through 8.8, as a function of ω_n and ζ . In so doing, one obtains the set of training points where the input is now the true DAO parameters and the output is the true values of ω_n and ζ .

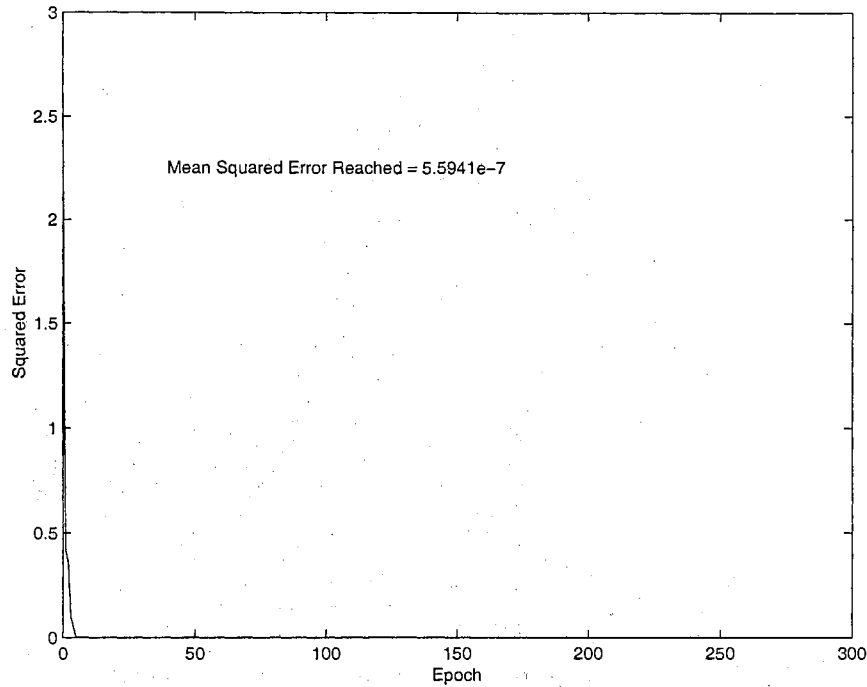


Figure 8.17: Mean Squared Error during Training

8.3.2 Training Analysis

Pre-processing of data, principle component analysis, post-processing of data, and post-training analysis are performed throughout the duration of the training of the neural network. During the principle component analysis, MATLAB determines that half of the 8 parameter inputs are redundant. During the training of the network, the final performance goal of 0 Mean Squared Error is not met but, instead, reaches a Mean Squared Error of $5.5941e-7$. The reader is referred to Figure 8.17 for the training results, where the response of Mean Squared Error with respect to the progression in epochs is shown.

The results of the post-training analysis, where a regression analysis is performed, is shown in Figures 8.18 and 8.19 for the regression analysis of ω_n and ζ , respectively. The regression analysis for both ω_n and ζ show a practically perfect correlation between the targets and actual output.

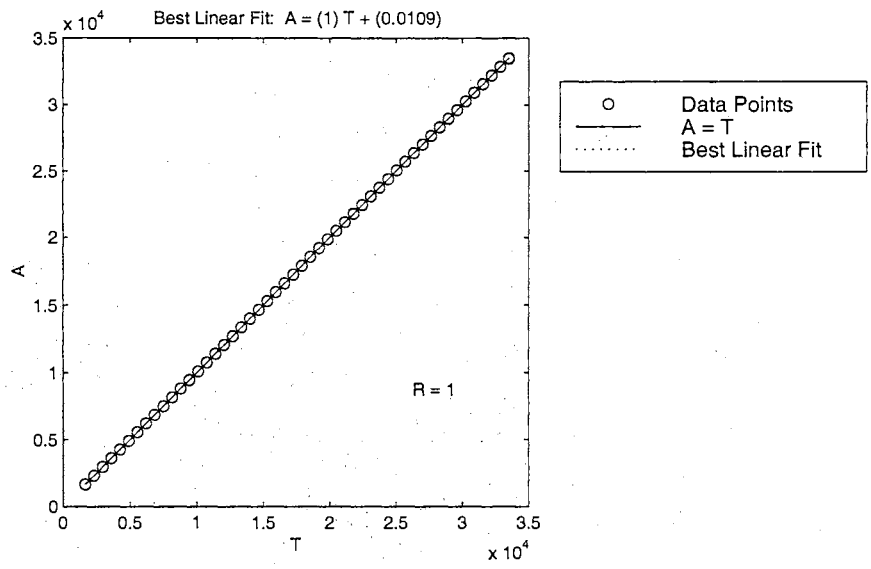


Figure 8.18: Regression Analysis of ω_n

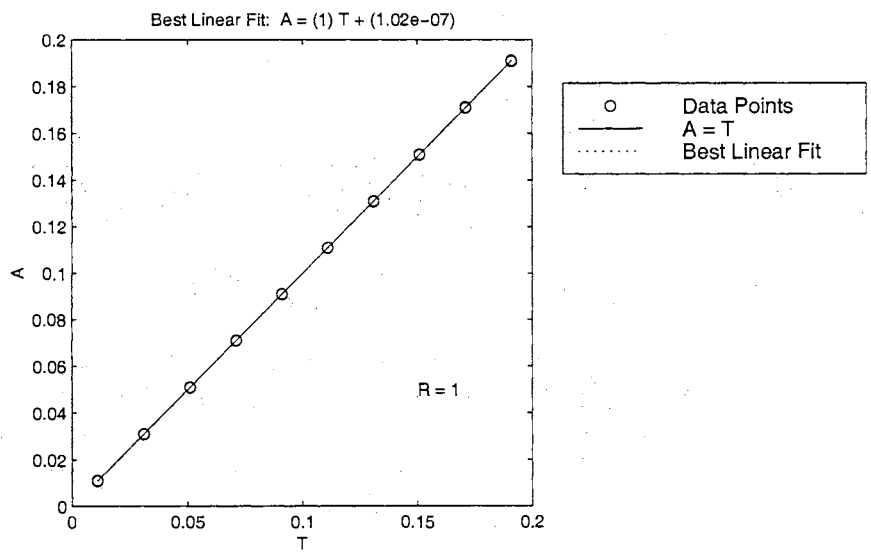


Figure 8.19: Regression Analysis of ζ

Parameter	Actual	Estimate	% Error
ζ	9.9500e-02	9.9531e-02	3.0737e-02
ω_n	9.7000e+03	9.6885e+03	1.1820e-01

Table 8.4: Case I: Neural Network Parameter Values and Estimates for $w_n = 9.7e3$ rad/sec

Parameter	Actual	Estimate	% Error
ζ	9.9500e-02	9.9447e-02	5.3178e-02
ω_n	1.3000e+04	1.2997e+04	2.6157e-02

Table 8.5: Case II: Neural Network Parameter Values and Estimates for $w_n = 1.3e4$ rad/sec

8.3.3 Results of IBM HDA System Parameter Implementation

For testing of the neural network, the neural network is applied to the IBM HDA system parameters, assuming that the exact parameter values are known. The two cases are analyzed, where $\omega_n = 9.7 \times 10^3$ rad/s for the first case and $\omega_n = 1.3 \times 10^4$ rad/s for the second case. Again, for both cases, $\zeta = 0.0995$. The results for the first and second cases are shown in Table 8.4 and 8.5, respectively. For both cases it is seen that the neural network is highly accurate in extracting correct values of aliased natural frequencies and damping ratios. As noted, however, these extracted values are the result of perfect knowledge of the IBM HDA in the DAO system parameter form.

For the general performance of the neural network for the case of varying resonant frequencies and damping ratios, the reader is referred to Figures 8.20 and 8.21. One sees that the neural network is highly accurate for all values of resonant frequencies and damping ratios, except for very small values of ζ , where the resulting error in damping ratio and natural frequencies is as high as 7% and 25%, respectively. For these smaller values of ζ , however, the neural network may be re-trained specifically to concentrate on this smaller range.

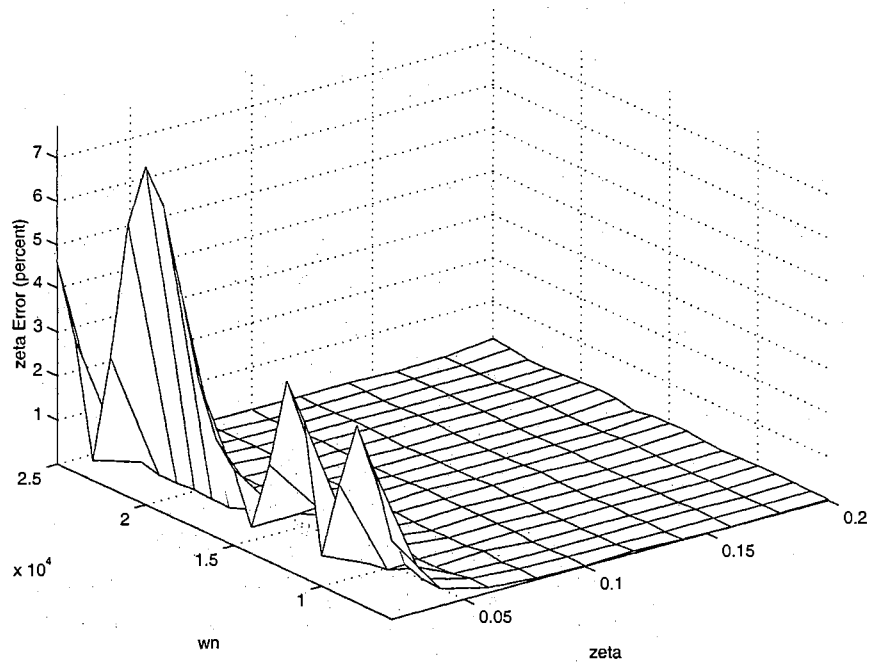


Figure 8.20: Neural Network Errors for Extraction of ζ

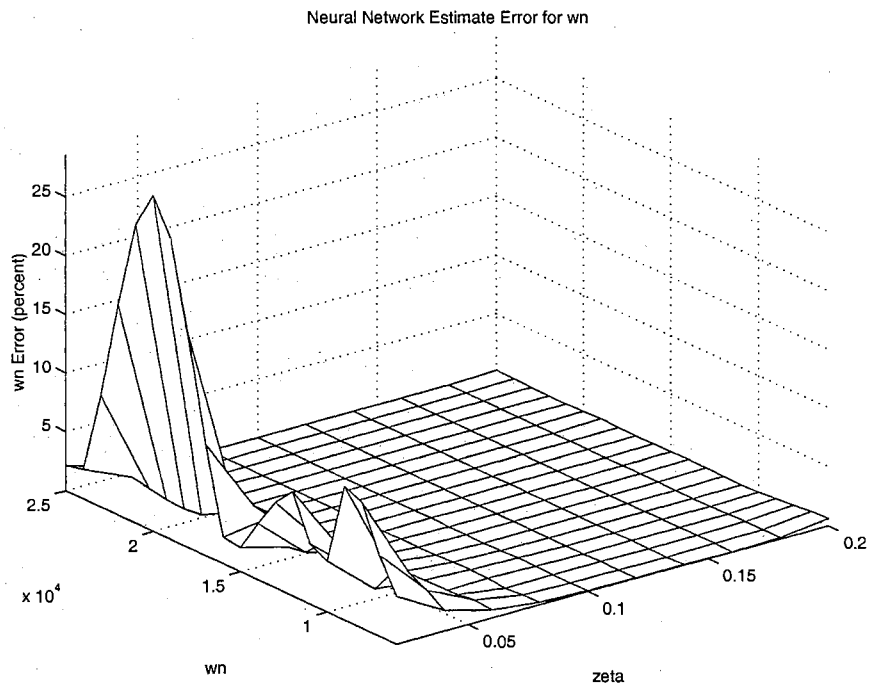


Figure 8.21: Neural Network Errors for Extraction of ω_n

Parameter	Actual	Estimate	% Error
ζ	9.9500e-02	9.9298e-02	2.0310e-01
ω_n	9.7000e+03	9.6840e+03	1.6505e-01

Table 8.6: Case I: Extraction of ζ and ω_n from DAO parameter estimates for $w_n = 9.7e3$ rad/sec

Parameter	Actual	Estimate	% Error
ζ	9.9500e-02	9.9261e-02	2.4003e-01
ω_n	1.3000e+04	1.2975e+04	1.8990e-01

Table 8.7: Case II: Extraction of ζ and ω_n from DAO parameter estimates for $w_n = 1.3e4$ rad/sec

8.4 Natural Frequency and Damping Ratio Approximation of the IBM HDA

The results of the neural network training are implemented on the final DAO parameter estimates on the scaled IBM HDA without noise for two cases: (1) $\omega_n = 9.7 \times 10^3$ rad/sec and (2) $\omega_n = 1.3 \times 10^4$ rad/s. For both cases, the damping ratio $\zeta = 0.0995$. As the reader recalls, despite the fact that the resonant frequencies are both aliased, the DAO correctly determines the correct system parameter estimates to within at most 5% error.

The results for the first and second cases are shown in Table 8.4 and 8.5, respectively. For the first case, the neural network determines that the damping ratio ζ is 9.9298×10^{-2} and that the natural frequency ω_n is 9.6840×10^3 rad/s. For the second case, the neural network determines that the damping ratio ζ is 9.9261×10^{-2} and that the natural frequency ω_n is 1.2975×10^4 rad/s. From the results it is seen that the percentage error does increase, obviously, if the true parameter values are not used to extract the appropriate resonant frequencies and damping ratios. However, this increase is not noticeable, as the estimate errors of ζ and ω_n using the DAO parameter estimates still remain less than 1%.

For the general performance of the neural network using actual DAO parameter estimates, the reader is referred to Figures (8.22) and (8.23). Here, it is shown that the combined effort of the DAO

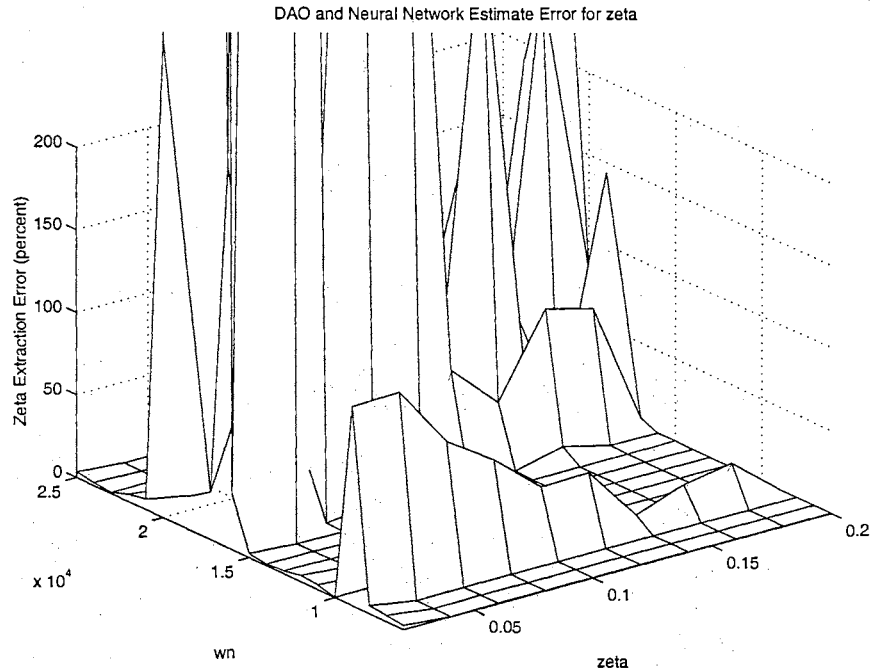


Figure 8.22: Errors for Extraction of ζ Using DAO Parameter Estimates

parameter estimates and neural network is highly accurate for all values of resonant frequencies and damping ratios, except for very small values of ζ and, as expected, for values of ω_n that lie near or on integer multiples of the Nyquist frequency. However, at very small values of ζ , the errors in this range of damping ratios are overshadowed by the large errors at resonant frequencies located near or at the system sampling frequency.

8.5 Concluding Remarks

Through the example of the IBM HDA, it is shown that the Discrete Adaptive Observer may be substituted into the Parallel Observer System resulting in the Adaptive Parallel Observer Systems to obtain accurate values of damping ratios and aliased resonant frequencies. In the case of the IBM HDA, there does appear to be a unique relationship between the values of DAO parameter estimates and the values of ζ and ω_n . In addition, by using the knowledge of the approximate values of the aliased natural frequency to initialize the DAO parameter estimates, the DAO is capable of accurately estimating system parameters, despite the presence of aliasing. It is seen, however that

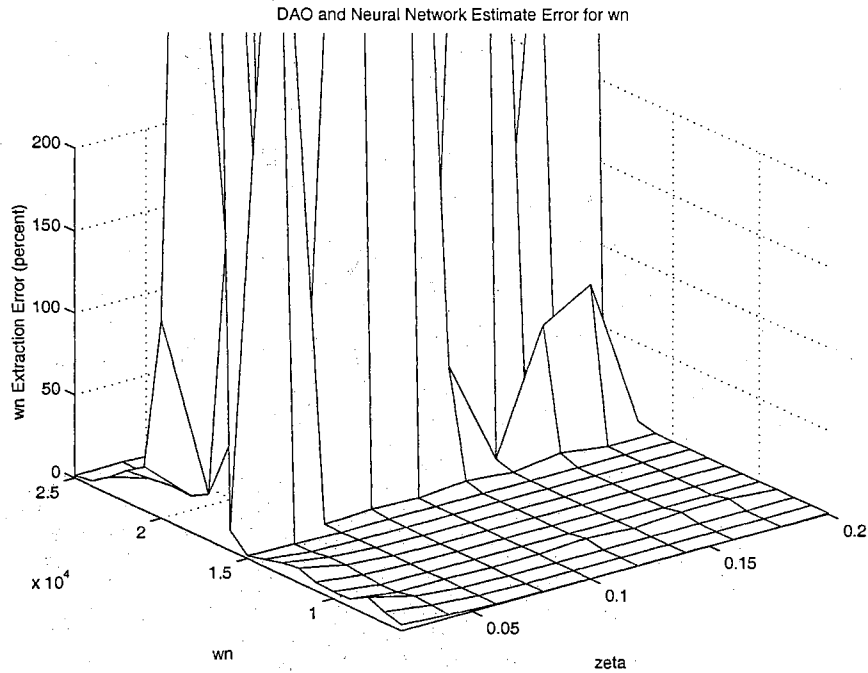


Figure 8.23: Errors for Extraction of ω_n Using DAO Parameter Estimates

this is not the case for frequencies at or near the Nyquist frequency and especially the sampling frequency.

By proper training of a neural network system, very accurate estimates of system damping ratios and natural frequencies may be extracted from DAO parameter estimates, so long as the DAO parameter estimates are accurate. Therefore, accurate estimates of ζ and ω_n can only be obtained when $\omega_n \ni j\omega_N, j \in Z_+$.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

A Parallel Observer System (POS) is proposed as a multirate estimation technique which takes into account periodically available output measurements. The Fast Observer is proven to converge to a stable Slow Observer during the ON sample points and provide stable state estimates during the INTER sample points with finite error bound. These results are illustrated with simulations of an IBM magnetic Head/Disk Assembly. It is seen that the accuracy of the Fast Observer is highly dependent upon the accuracy of the Slow Observer's state estimates.

Furthermore, a modification of the Parallel Observer System is proposed where a Discrete Adaptive Observer (DAO) System is implemented to not only provide stable state estimates, but also to estimate system parameters which may be initially incorrect or which may be changing during the control process. The overall modified Parallel Observer System, the Adaptive Parallel Observer System (APOS), is proven to be stable and to provide convergent state and parameter estimates. The Discrete Adaptive Observer is applied to a second order resonance model and the Adaptive Parallel Observer is applied to the IBM HDA. The simulations show that although large transients exist, state estimate do converge to actual values. In addition, given no additional input disturbances, a non-augmented state form, and an appropriately scaled system model, accurate parameter estimates

are obtained.

The success of the APOS is highly contingent upon the accuracy of the applied DAO. With the application of a genetic algorithm to the implemented Discrete Adaptive Observer of the APOS, the tuning of observer parameters is a straightforward, automated task. In addition, with some prior knowledge of the system and using the DAO with a trained neural network system, accurate values of system damping ratios and aliased resonant frequencies may be obtained. This accuracy is achieved so long as the resonant frequencies are not integer multiples of the Nyquist frequency and especially not of the sampling frequency and as long as the neural network is properly trained within the appropriate range of parameter values.

9.2 Future Work

Because the Discrete Adaptive Observer interprets any disturbances as system parameter changes, the entire APOS is highly susceptible to noise. In addition, the APOS is not guaranteed stable under state augmentation to allow for matched uncertainty estimation. Therefore, future work involves the study of noise and its effects on the APOS and how the APOS may be modified so as to allow for matched uncertainty estimation. In addition, for practical applications, it may be more suitable to apply another method, other than a neural network, for system parameter extraction, such as a polynomial fit procedure. Hence, future work also involves investigating other types of numerical techniques for this purpose.

Bibliography

- [1] G. M. Kranc. Input-output analysis of multirate feedback system. *IEEE Transactions on Automatic Control*, AC(3):21–28, November 1957.
- [2] John H. Ragazzini and Gene F. Franklin. *Sampled Data Control Systems*. McGraw-Hill Series in control Systems Engineering. McGraw-Hill, New York, 1958.
- [3] E. I. Jury. A note on multirate sampled-data systems. *IEEE Transactions on Automatic Control*, AC(12):319–320, June 1967.
- [4] R. E. Kalman and J. E. Bertram. A unified approach to the theory of sampling systems. *Journal of the Franklin Institute*, 267:405–436, 1959.
- [5] Mitsuhiro Araki and Koichiro Yamamoto. Multivariable multirate sampled-data systems: State-space description, transfer characteristics, and nyquist criterion. *IEEE Transactions on Automatic Control*, 31(2):145–154, February 1986.
- [6] P. Colaneri, R. Scattolini, and N. Schiavoni. Stabilization of multirate sampled-data linear systems. *Automatica*, 26(2):377–380, 1990.
- [7] David G. Meyer. A new class of shift-varying operators, their shift-invariant equivalents, and multirate digital systems. *IEEE Transactions on Automatic Control*, 35(4):429–433, April 1990.
- [8] Sauro Longhi. Structural properties of multirate sampled-data systems. *IEEE Transactions on Automatic Control*, 39(3):692–696, March 1994.

- [9] Yuguang Fang and Chu Xuedao. Stability analysis of multirate sampled-data systems. In *Proceedings of the IEEE American Control Conference*, pages 3241–3242, 1994.
- [10] Martin C. Berg, Naftali Amit, and J. David Powell. Multirate digital control system design. *IEEE Transactions on Automatic Control*, 33(5):1139–1150, December 1988.
- [11] Gregory S. Mason and Martin C. Berg. Reduced-order multirate compensator synthesis. *Journal of Guidance, Control, and Dynamics*, 15(3):700–706, 1992.
- [12] Martin C. Berg and Gregory S. Mason. New multirate sampled-data control law structure and synthesis algorithm. *Journal of Guidance, Control, and Dynamics*, 15(5):1183–1191, 1992.
- [13] L. J. Serrano and P. J. Ramadge. Sampled disturbance decoupling with stability using multirate control. *IEEE Transactions on Automatic Control*, 36(9):1061–1065, September 1991.
- [14] Guoming Zhu and Robert E. Skelton. Robust properties of periodic discrete and multirate systems. *IEEE Transactions on Automatic Control*, 37(5):610–615, May 1992.
- [15] Wassim M. Haddad and Vikram Kapila. A periodic fixed-structure approach to multirate control. *IEEE Transactions on Automatic Control*, 40(2):301–307, February 1995.
- [16] R. J. Patton, G. P. Liu, and Y. Patel. Sensitivity properties of multirate feedback control systems, based on eigenstructure assignment. *IEEE Transactions on Automatic Control*, 40(2):337–342, February 1995.
- [17] K. Natarajan. Multirate dc motor speed controller design for zero steady-state ripple. In *Proceedings of the IEEE American Control Conference*, pages 2760–2764, 1991.
- [18] P. Colaneri, R. Scattolini, and N. Schiavoni. Lqg optimal control of multirate sampled-data systems. *IEEE Transactions on Automatic Control*, 37(5):675–682, May 1992.
- [19] Riccardo Scattolini and Nicola Schiavoni. On the output control of multirate systems subject to arbitrary exogenous signals. *IEEE Transactions on Automatic Control*, 38(4):643–646, April 1993.

- [20] R. E. Young, R. Henriksen, and D. A. Mellichamp. A multi-rate decentralized parameter estimation method for stiff systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1902–1907, 1987.
- [21] Ioannis S. Apostolakis. A time invariant approach to optimal control and estimation of multirate systems. In *Proceedings of the IEEE American Control Conference*, pages 2772–2773, 1991.
- [22] Wassim M. Haddad, Dennis S. Bernstein, and Hsing-Hain Huang. Reduced-order multirate estimation for stable and unstable plants. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2892–2897, 1990.
- [23] Wassim M. Haddad, Dennis S. Bernstein, and Vikram Kapila. Reduced-order multirate estimation. *Journal of Guidance, Control, and Dynamics*, 17(4):712–721, 1994.
- [24] Kerim Demirbas. State estimation for nonlinear discrete dynamic systems with missing observations. *Journal of the Franklin Institute*, 372(1):49–59, 1990.
- [25] Andrey V. Savkin and Ian R. Petersen. Model validation and state estimation for uncertain continuous-time systems with missing discrete-continuous data. In *Proceedings of the IEEE Conference on Decision and Control*, pages 570–573, 1996.
- [26] Wen-Wei Chiang. Multirate state-space digital controller for sector servo systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1902–1907, 1990.
- [27] Anthony M. Phillips and Masayoshi Tomizuka. Robust wide-range controller using multirate estimation and control for velocity regulation and tracking. In *Proceedings of the ASME Advances in Information Storage and Processing Systems*, volume 1, pages 189–194, 1995.
- [28] Anthony M. Phillips and Masayoshi Tomizuka. Multirate estimation and control under time-varying data sampling with application to information storage devices. In *Proceedings of the IEEE American Control Conference*, pages 4151–4155, 1996.
- [29] Weiping Lu and D. Grant Fisher. Output estimation with multi-rate sampling. *International Journal of Control*, 48(1):149–160, 1988.

- [30] Weiping Lu and D. Grant Fisher. Multirate adaptive inferential estimation. *IEE Proceedings-D*, 139(2):181–189, 1992.
- [31] Parthasarathy Ramachandran, Gary E. Young, and Eduardo A. Misawa. Intersample output estimation with multirate sampling. In *Proceedings of the IEEE Conference on Control Applications*, pages 576–581, 1996.
- [32] Hisashi Kando, Tomohiki Aoyama, and Tetsuo Iwazumi. Multirate observer design via singular perturbation theory. *International Journal of Control*, 50(5):2005–2023, 1989.
- [33] Kenneth R. Shousse and David G. Taylor. Discrete-time observers for singularly perturbed continuous-time systems. *IEEE Transactions on Automatic Control*, 40(2):224–235, February 1995.
- [34] Takeyori Hara and Masayoshi Tomizuka. On the performance enhancement of multi-rate controllers for hard disk drive. In *1998 Asia-Pacific Magnetic Recording Conference*, 1998.
- [35] Takeyori Hara and Masayoshi Tomizuka. On the performance enhancement of multi-rate controllers for hard disk drive. *IEEE Transaction on Magnetics*, 1999. To be published.
- [36] Takeyori Hara and Masayoshi Tomizuka. Multi-rate controller for hard disk drive with redesign of state estimator. In *Proceedings of the IEEE American Control Conference*, pages 3033–3037, 1998.
- [37] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addison Wesley, Reading, Massachusetts, second edition, 1990.
- [38] Evert Steen Cooper. *Minimum Time Control with Minimum Vibration and with Power Limiting, with Application to the Magnetic Disk File*. PhD thesis, Santa Clara University, 1993.
- [39] R. Todd Lyler and Thomas Rendon. A technical report on a general disk drive model. Technical Memorandum ACL-98-009, Advanced Controls Laboratory, School of Mechanical & Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA, 1998.

- [40] The MathWorks, Inc. *Using SIMULINK*, for 2 edition, June 1997.
- [41] The MathWorks, Inc. *Using MATLAB*, for 5.1 edition, June 1997.
- [42] Robert L. Carroll and David P. Lindorff. An adaptive observer for single-input single-output linear systems. *IEEE Transactions on Automatic Control*, 18(5):428–435, October 1973.
- [43] Engineering Information Village. Ei compendexweb, 1995.
- [44] P. Kudva and L.S. Narendra. The discrete adaptive observer. In *Proceedings of the IEEE Conference on Decision and Control*, 1974.
- [45] Takashi Suzuki and Masanori Andoh. Design of a discrete adaptive observer based on hyperstability theorem. *International Journal of Control*, 26(4):643–653, 1977.
- [46] Mohammad Shahrokhi and Manfred Morari. A discrete adaptive observer and identifier with arbitrarily fast rate of convergence. *IEEE Transactions on Automatic Control*, 27(2):506–509, April 1982.
- [47] Yeon-Chan Hong, Jong-Hwan Kim, and Keh-Kun Choi. Discrete adaptive observer with exponential weighting properties. *IEEE Transactions on Automatic Control*, 34(2):229–231, February 1989.
- [48] Takashi Suzuki, Nakamura Takumi, and Masanori Andoh. Discrete adaptive observer with fast convergence. *International Journal of Control*, 31(6):1107–1119, 1980.
- [49] Zenta Iwai, Kazuo Mano, and Takehiko Inada. An adaptive observer for single-input single-output linear systems with inaccessible input. *International Journal of Control*, 32(1):159–174, 1980.
- [50] Thomas Rendon. Genetic algorithm code for matlab. Genetic Algorithm Code written for MATLAB incorporates tuning of the Discrete Adaptive Observer parameters for us in the IBM Head/Disk Assembly System, December 1998.

[51] Martin T. Hagan, B. Demuth, Howard, and Mark Beale. *Neural Network Design*. PWS Publishing Company, 1996.

Appendix A

SIMULINK Implementation of the Discrete Adaptive Observer

The following is a list of figures which illustrate the implementation of the Discrete Adaptive Observer in the Adaptive Parallel Observer System. Figure A.1 give the general application of the DAO. Figure A.2 describes the simulation process for the adaptive variable z . Figures A.3 through A.5 illustrate how the state variable filter matrix F is applied to calculate the adaptive parameters ϕ_1 and ϕ_2 . Figure A.6 shows the calculation steps for the adaptive variable Γ . The calculation of the parameter estimates \hat{p} are shown in Figure A.7. The state estimate $\hat{x}(m)$ and the output estimate $\hat{y}(m)$ are obtained by the SIMULINK blocks shown in Figures A.8 though Figure A.10.

The remaining figures in this section of the appendix describes the calculation of the values $S(m)\hat{p}(m)$.

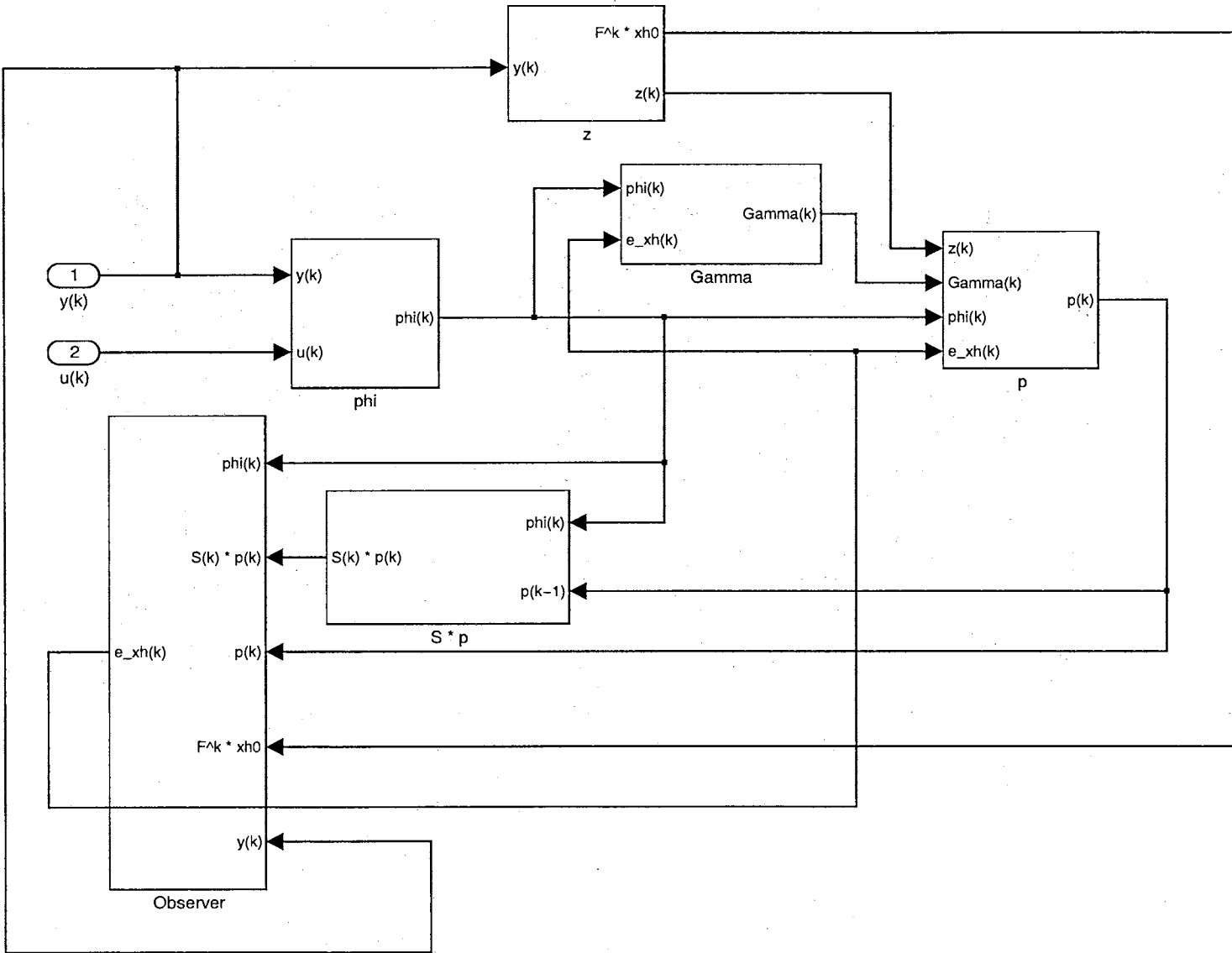


Figure A.1: The Discrete Adaptive Observer

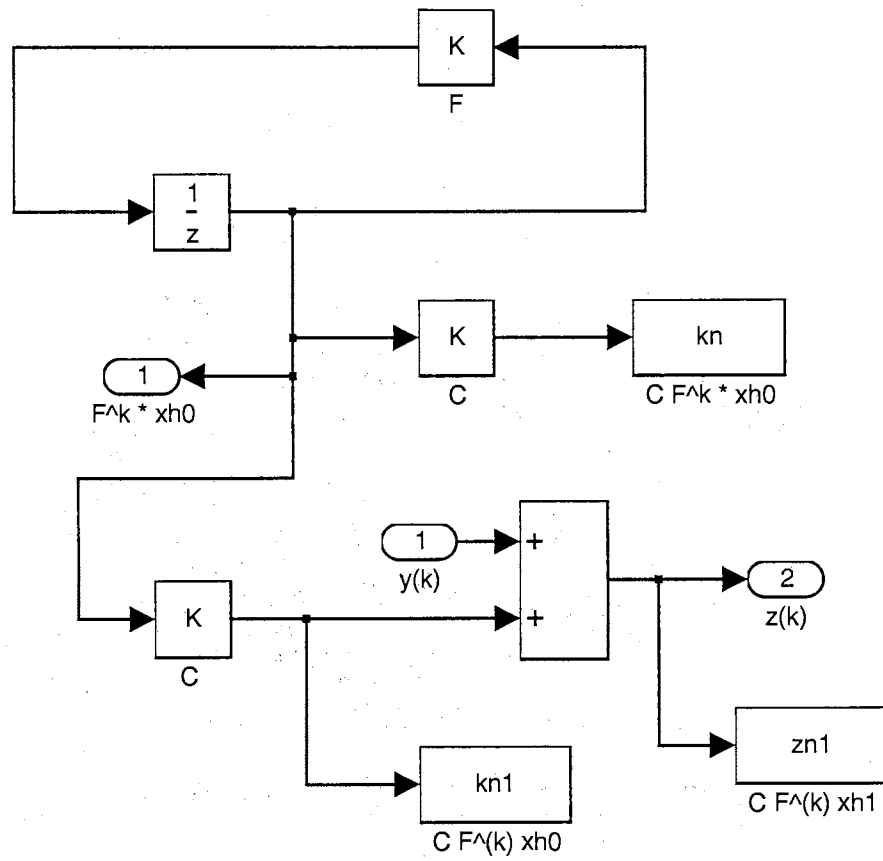


Figure A.2: $z(m)$

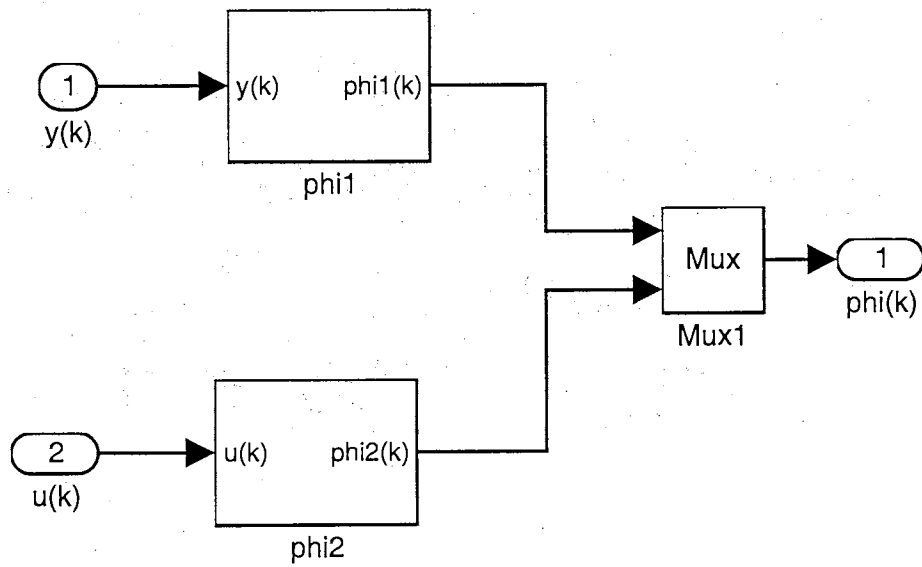


Figure A.3: $\phi(m)$

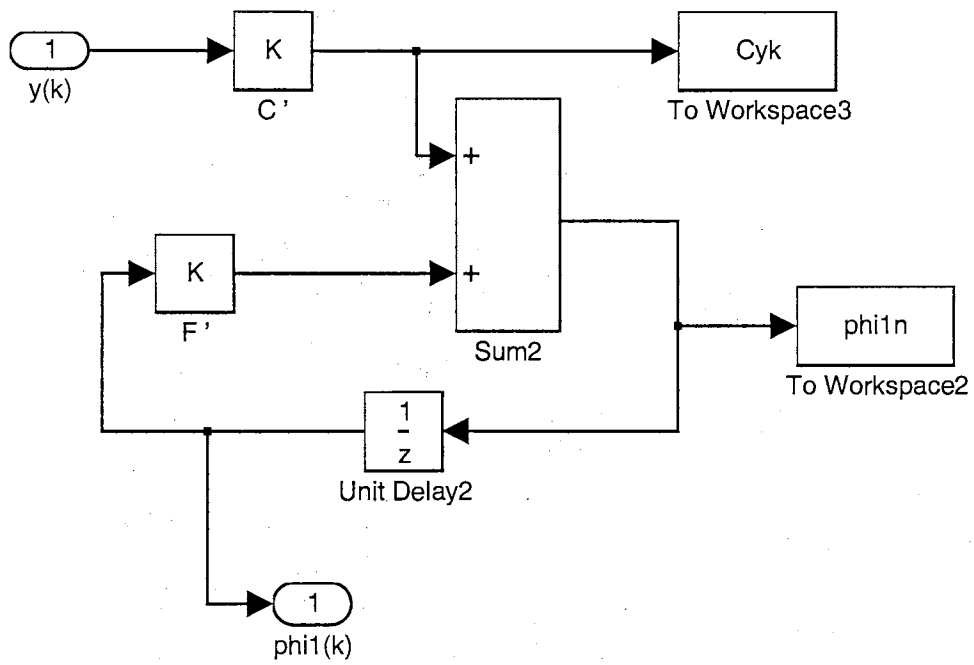


Figure A.4: $\phi_1(m)$

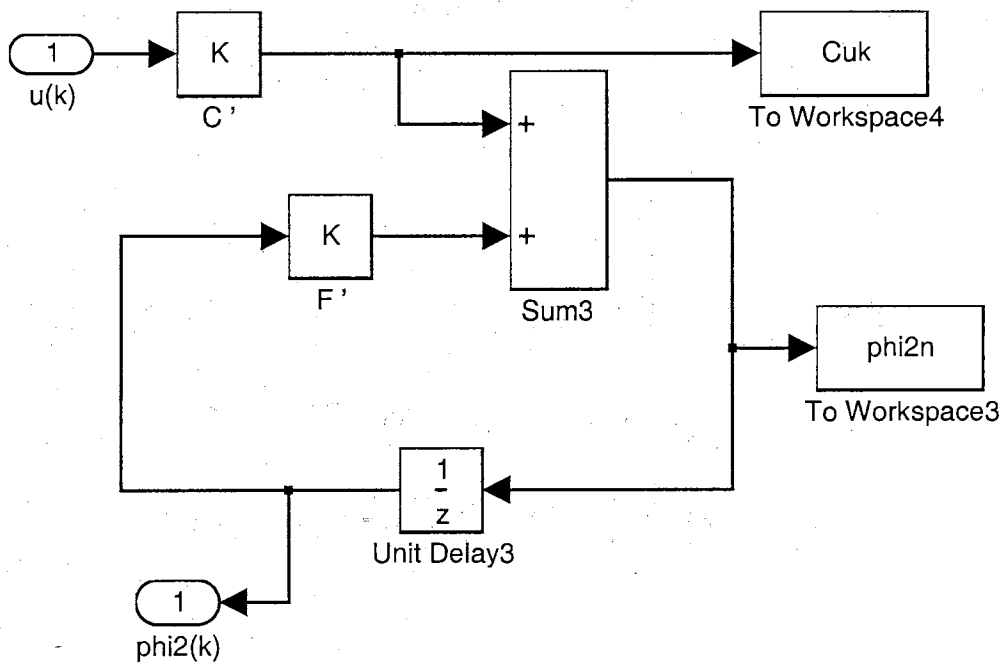


Figure A.5: $\phi_2(m)$

Figure A.6: $\Gamma(m)$

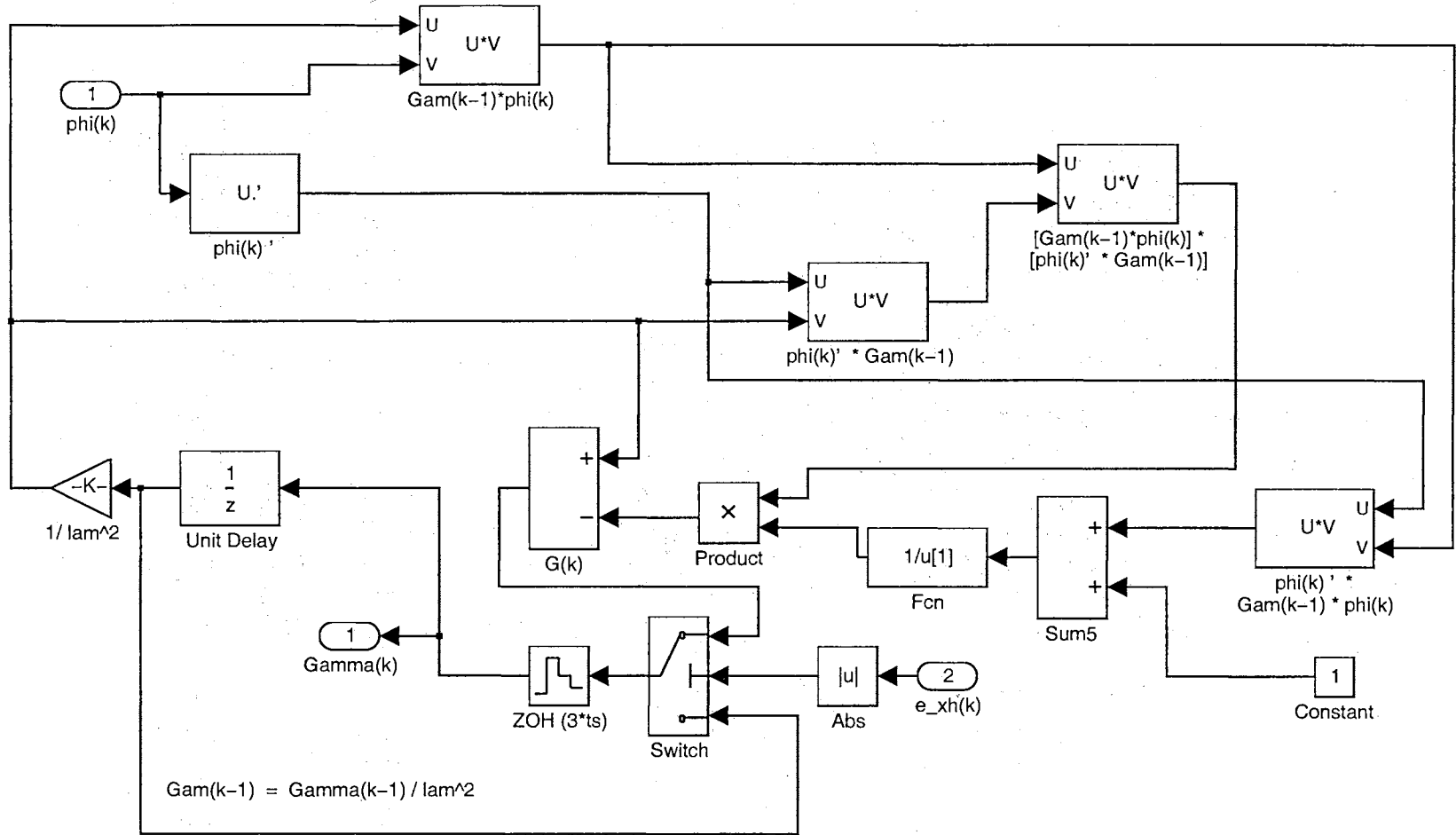
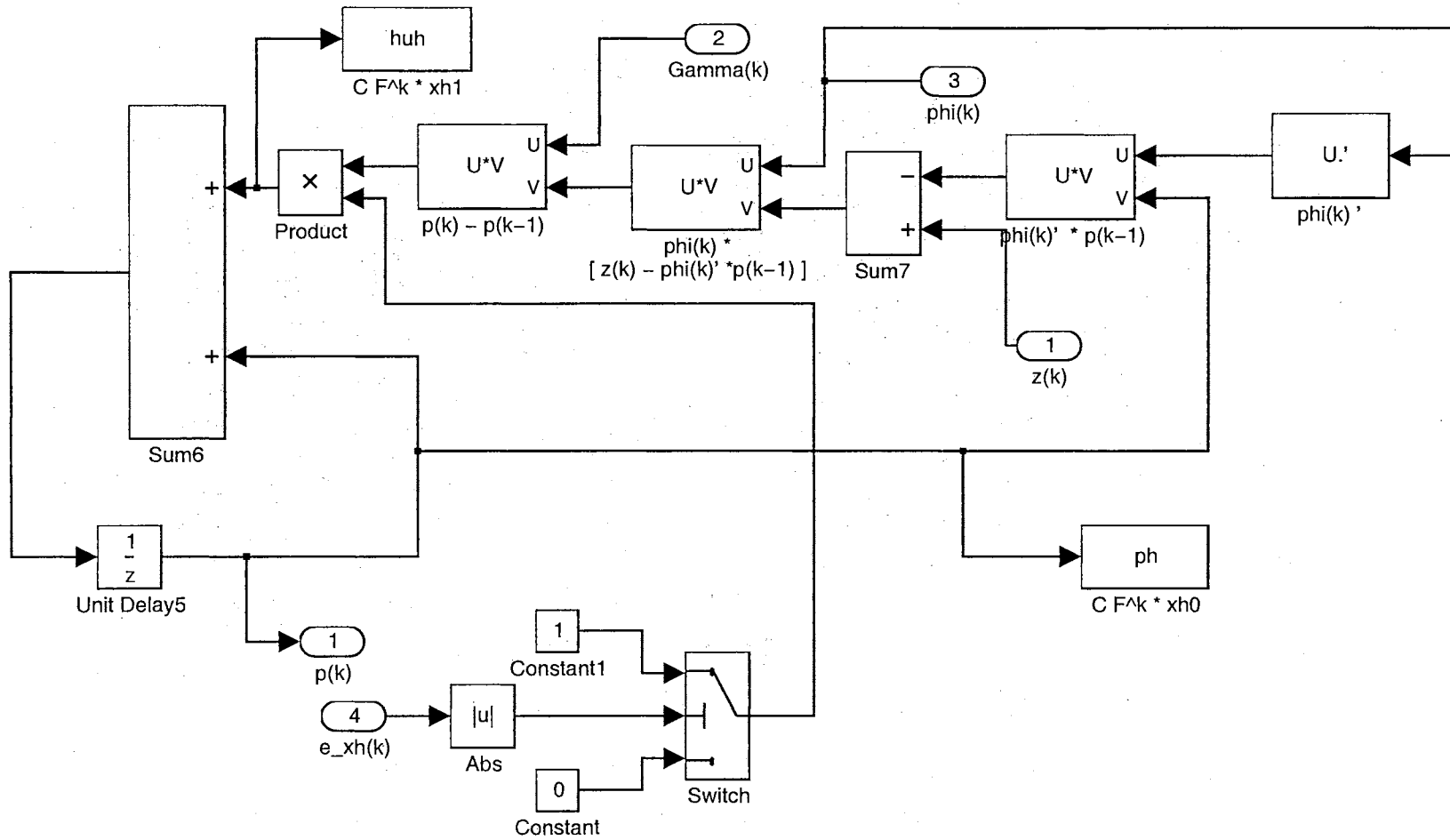


Figure A.7: $\hat{p}(m)$



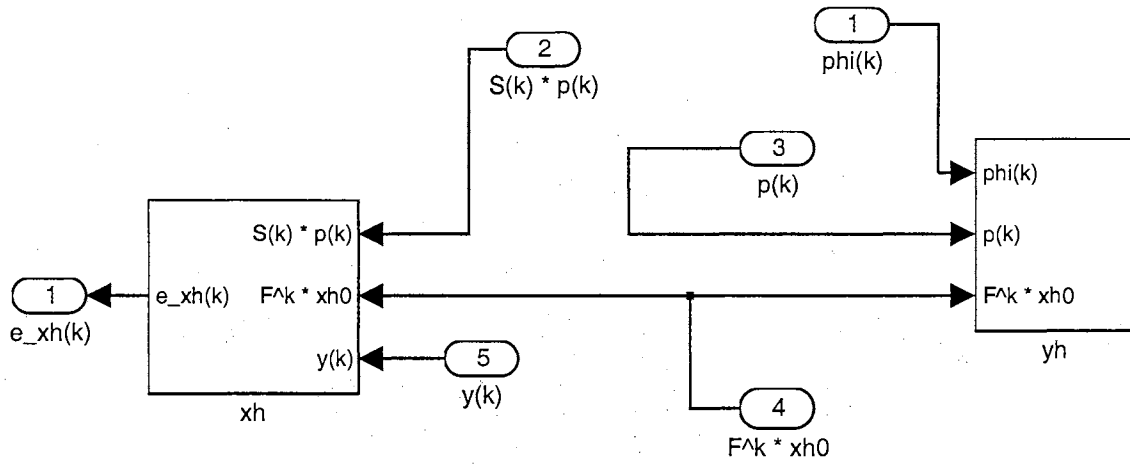


Figure A.8: State and Output Estimates

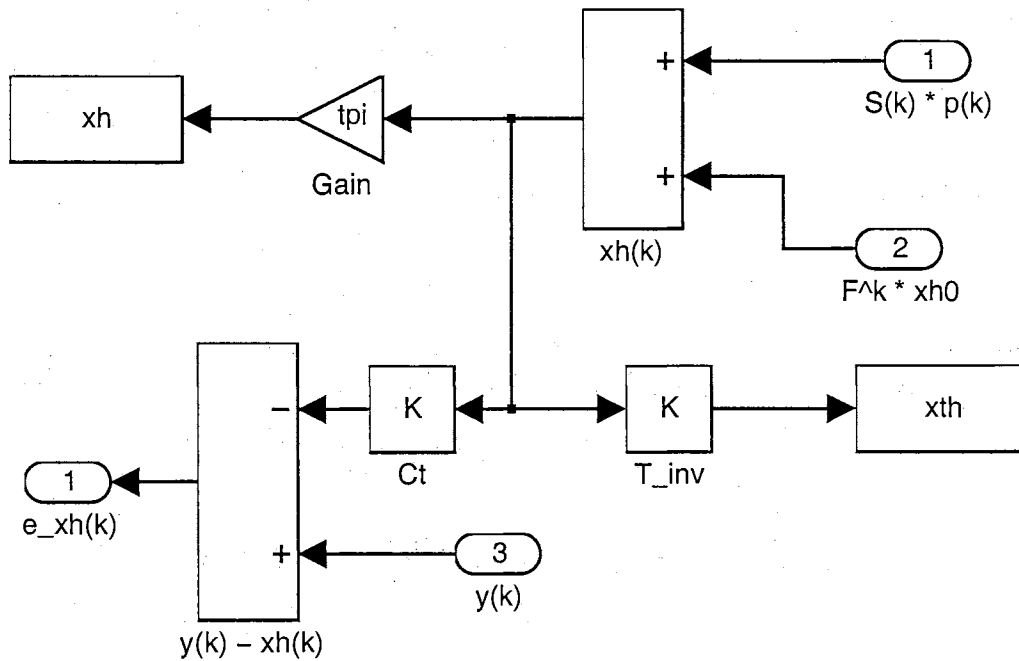


Figure A.9: State Estimates $\hat{x}(m)$

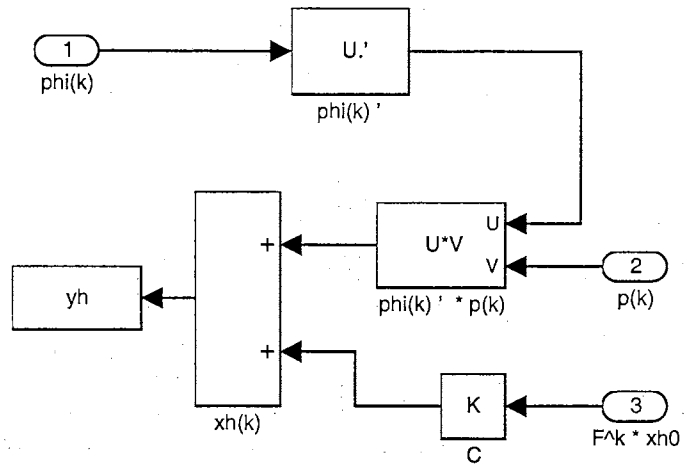


Figure A.10: Output Estimates $\hat{y}(m)$

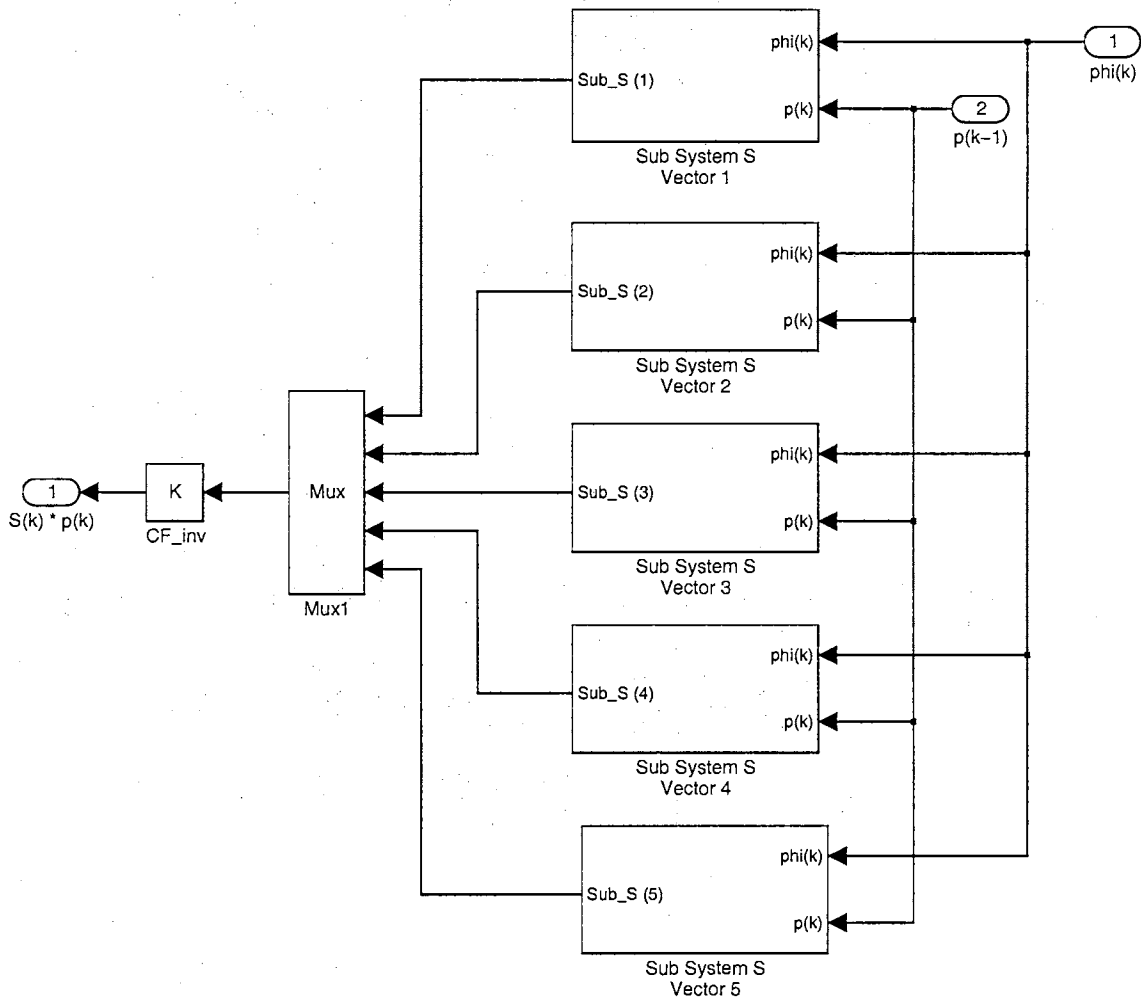


Figure A.11: $S(m)\hat{p}(m)$

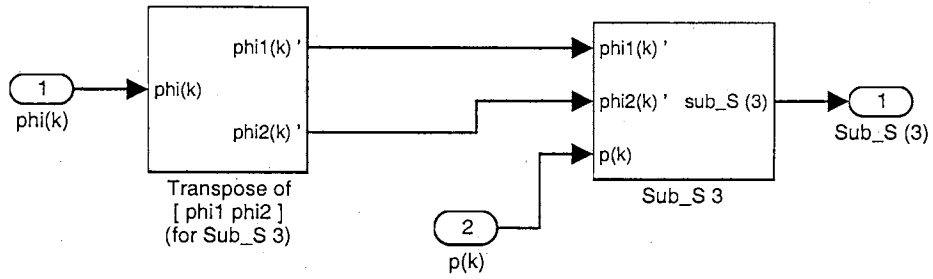


Figure A.12: Subsystem S: Vector 3

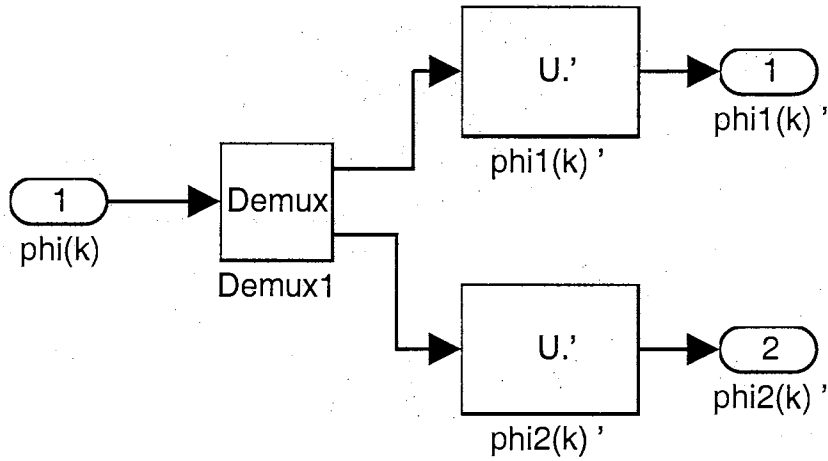


Figure A.13: $\phi(m)^T$

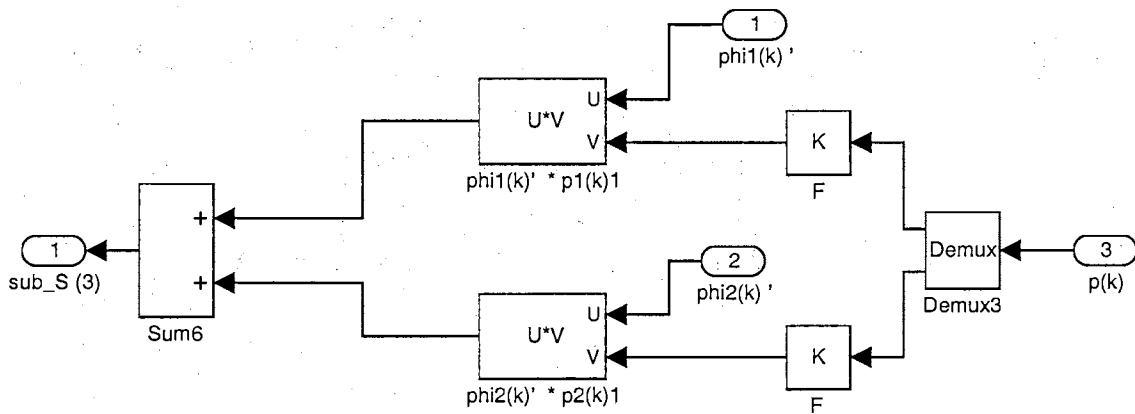


Figure A.14: Sub S 3

Appendix B

Adaptive Parallel Observer System

IBM HDA Parameter Estimate

Results

It should be emphasized that this application includes a system augmentation to allow for matched certainty estimation and that the DAO cannot guarantee stable results in this case. Figures B.1 through B.4 represent the parameter estimation results of the A_T matrix parameters when the Discrete Adaptive Observer is implemented on the IBM magnetic Head/Disk Assembly described in Chapter 6.

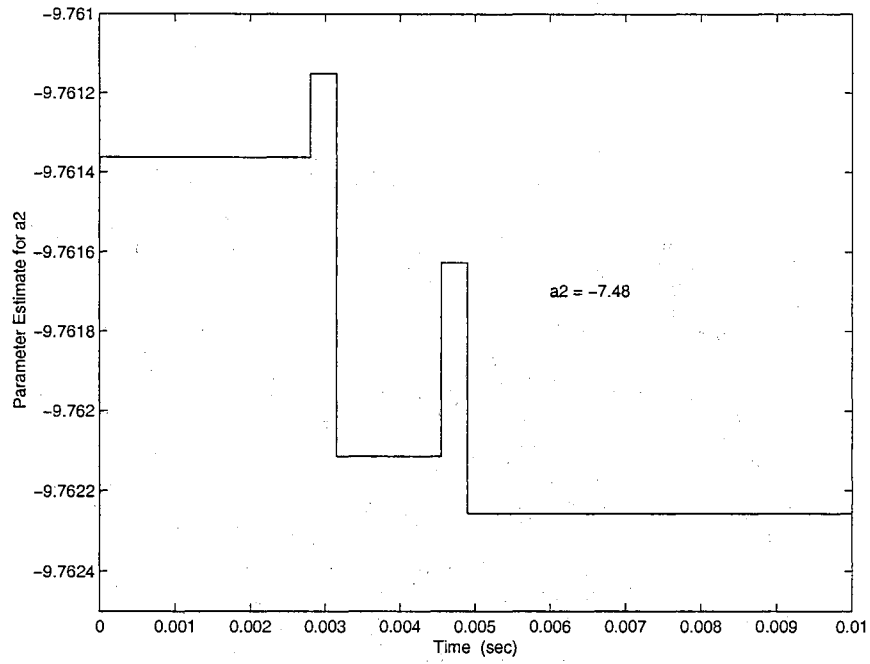


Figure B.1: APOS Parameter estimate of a_2

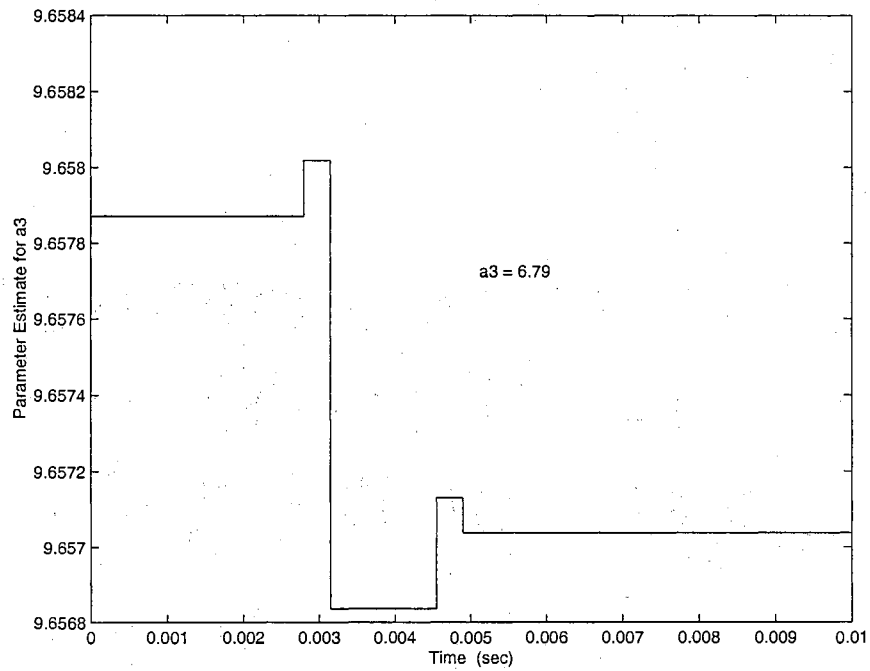


Figure B.2: APOS Parameter estimate of a_3

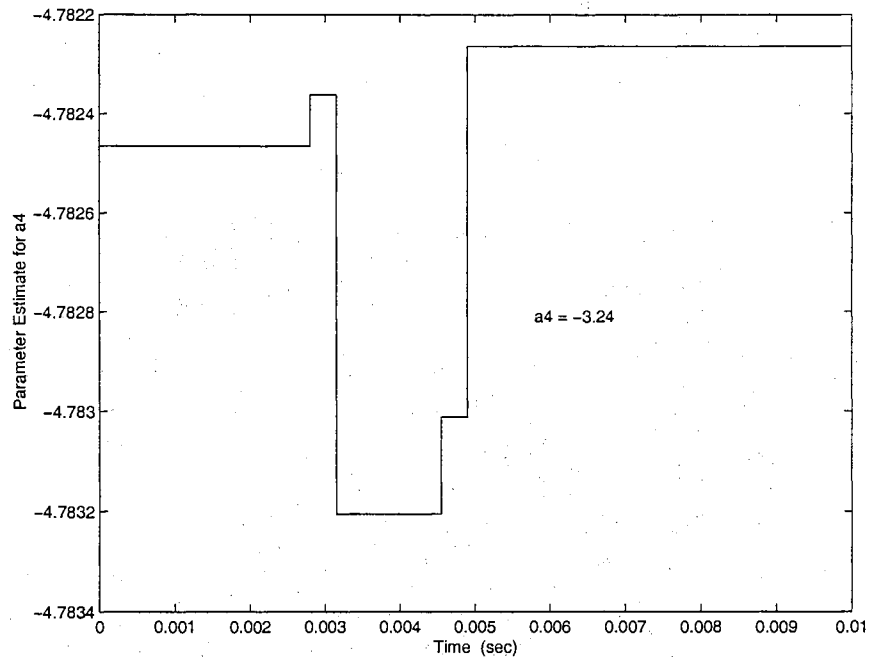


Figure B.3: APOS Parameter estimate of a_4

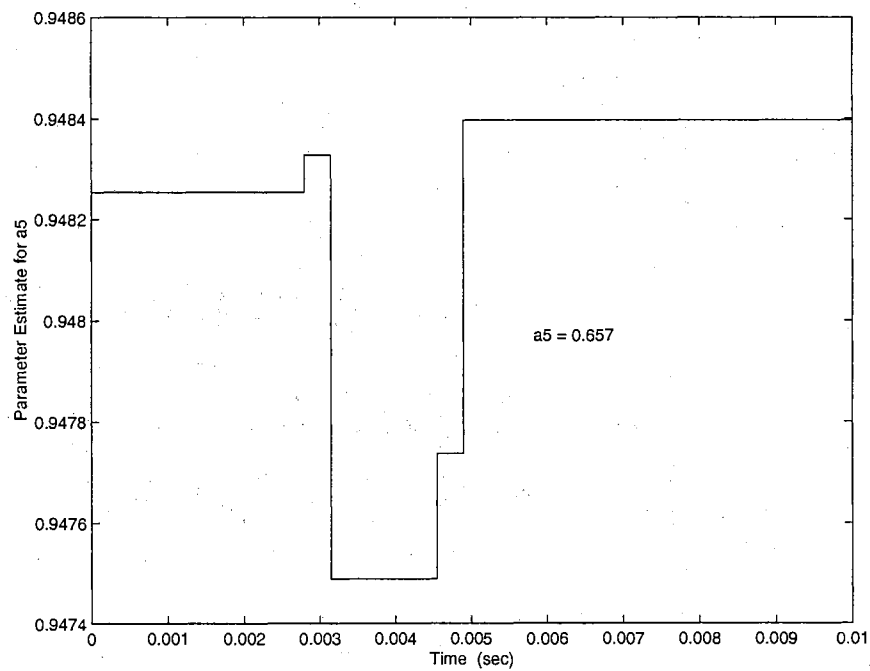


Figure B.4: APOS Parameter estimate of a_5

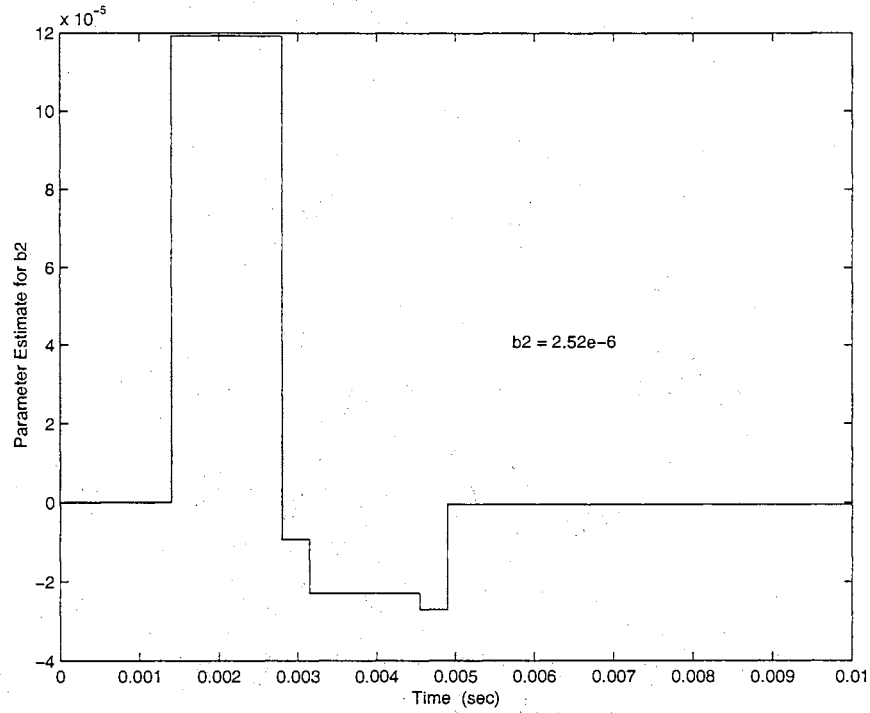


Figure B.5: APOS Parameter estimate of b_2

In addition, Figures B.5 through B.8 represent the parameter estimation results of the B_T matrix parameters when the Discrete Adaptive Observer is implemented on the IBM magnetic Head/Disk Assembly described in Chapter 6.

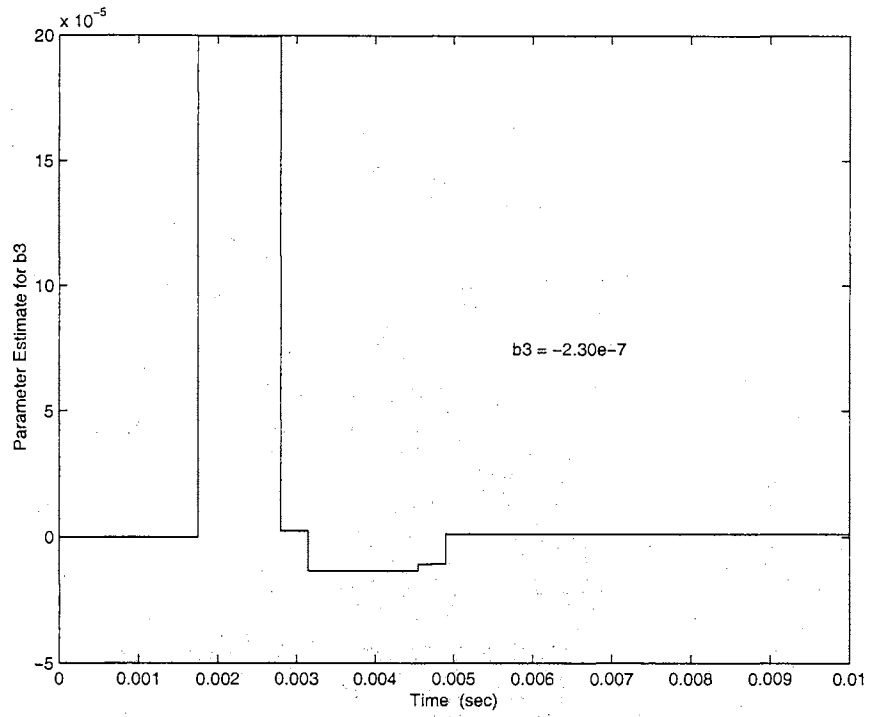


Figure B.6: APOS Parameter estimate of b_3

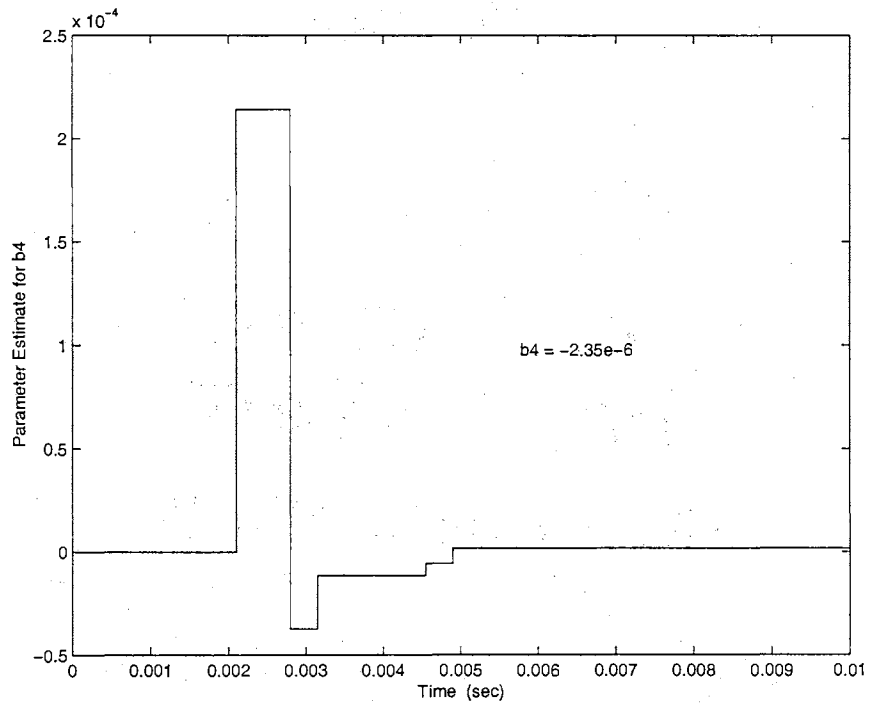


Figure B.7: APOS Parameter estimate of b_4

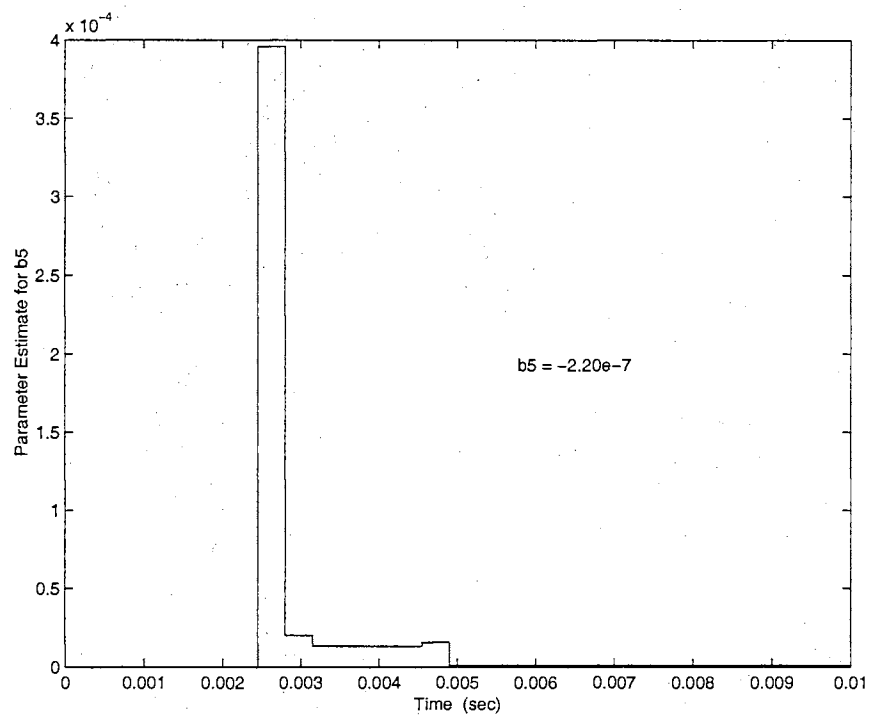


Figure B.8: APOS Parameter estimate of b_5

Appendix C

Discrete Adaptive Observer

Parameter Estimate Results of

IBM HDA with Noise

C.1 Parameter Estimates Without Retuning of DAO

Figures (C.1) through (C.3) represent the parameter estimation results of the \bar{A}_T matrix parameters when the Discrete Adaptive Observer is implemented on the IBM magnetic Head/Disk Assembly described in Chapter 8 with noise and without retuning of the Discrete Adaptive Observer.

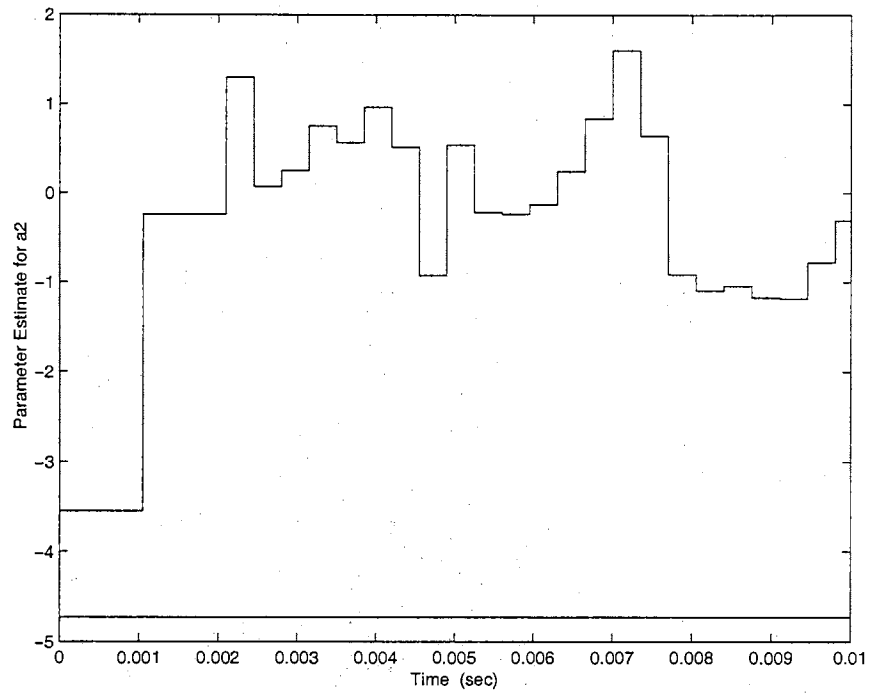


Figure C.1: APOS Parameter estimate of \bar{a}_2 Without Retuning

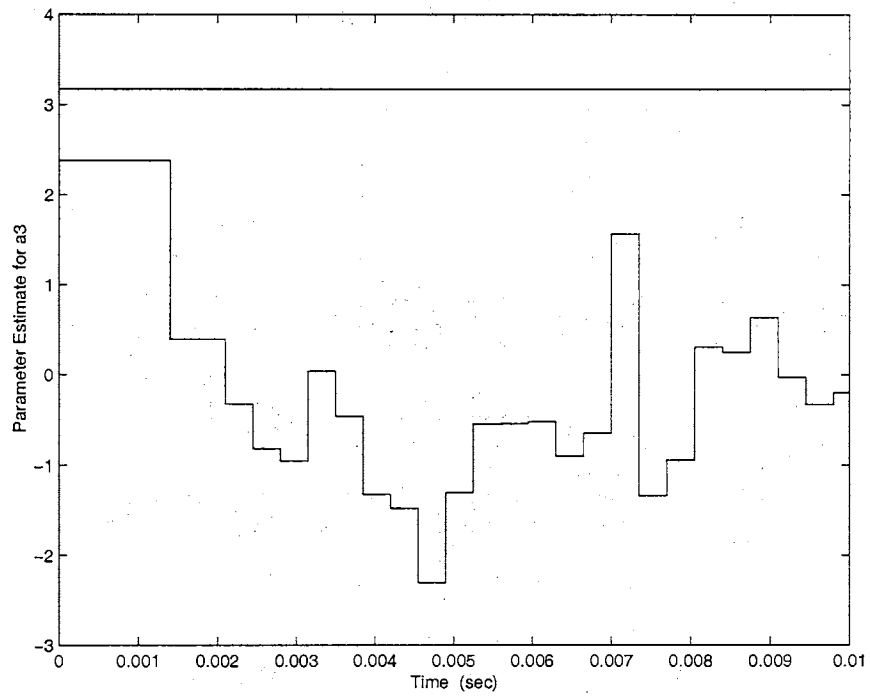


Figure C.2: APOS Parameter estimate of \bar{a}_3 Without Retuning

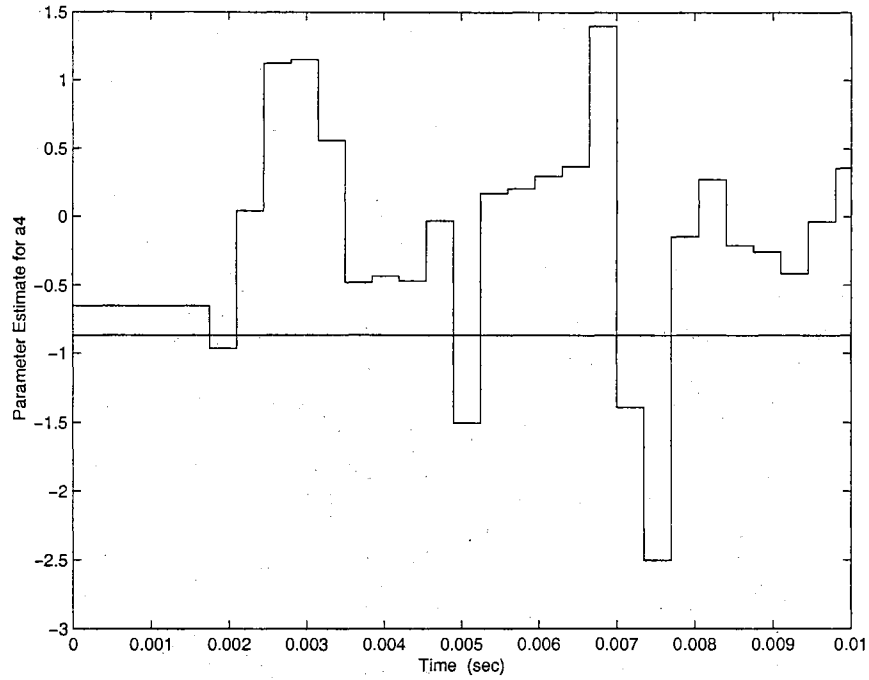


Figure C.3: APOS Parameter estimate of \bar{a}_4 Without Retuning

C.2 Parameter Estimates With Retuning of DAO

Figures (C.7) through (C.9) represent the parameter estimation results of the \bar{A}_T matrix parameters when the Discrete Adaptive Observer is implemented on the IBM magnetic Head/Disk Assembly described in Chapter 8 with noise and retuning of the Discrete Adaptive Observer.

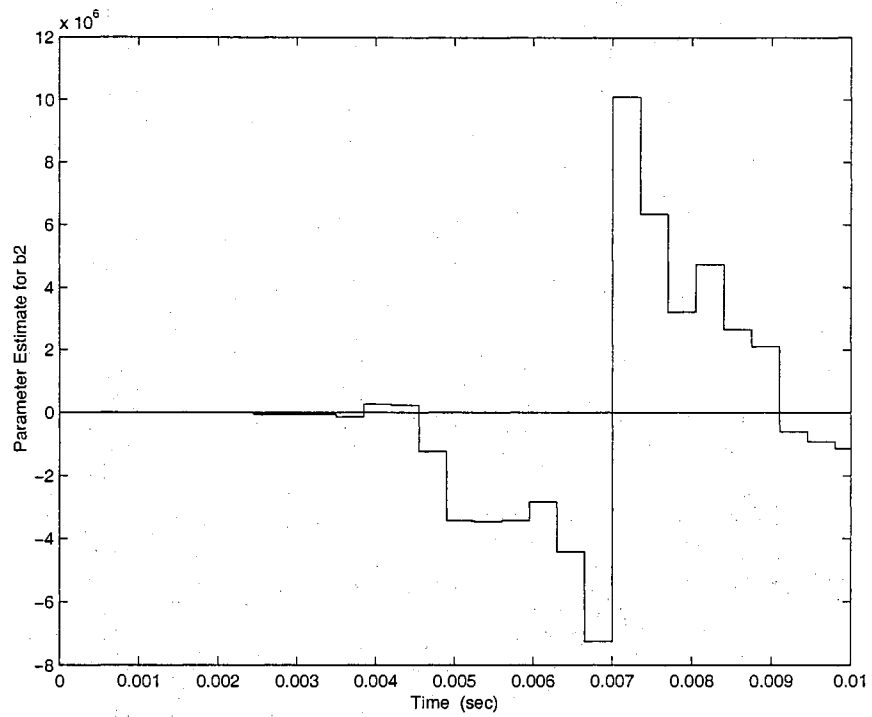


Figure C.4: APOS Parameter estimate of \bar{b}_2 Without Retuning

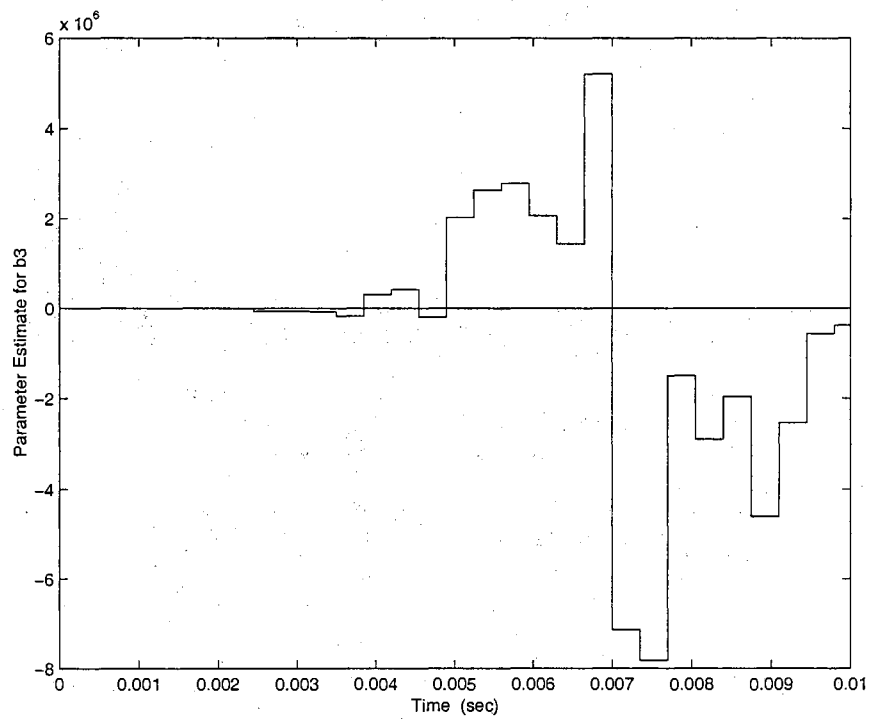


Figure C.5: APOS Parameter estimate of \bar{b}_3 Without Retuning

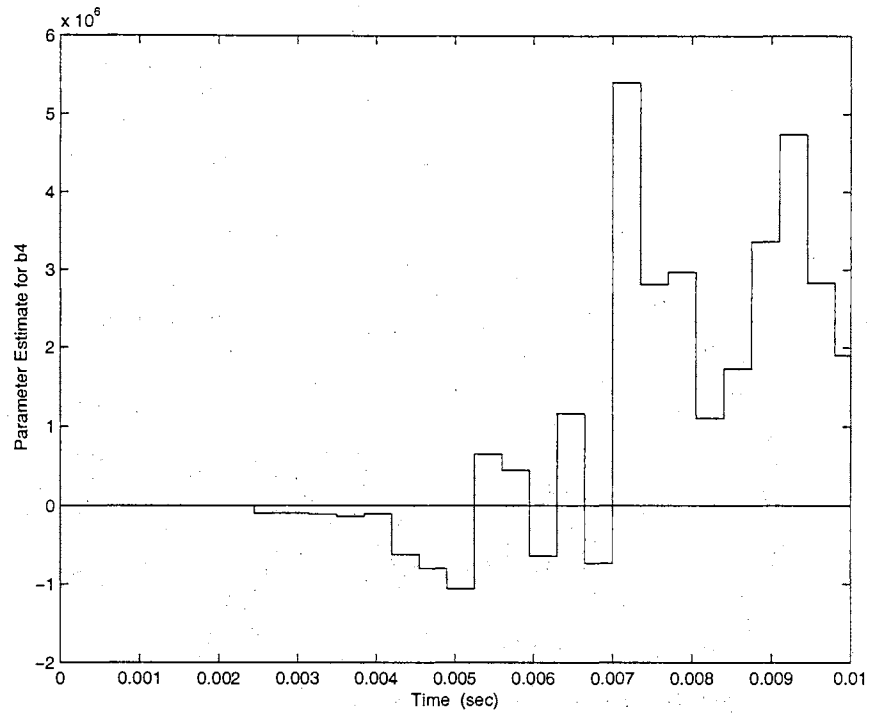


Figure C.6: APOS Parameter estimate of \bar{b}_4 Without Retuning

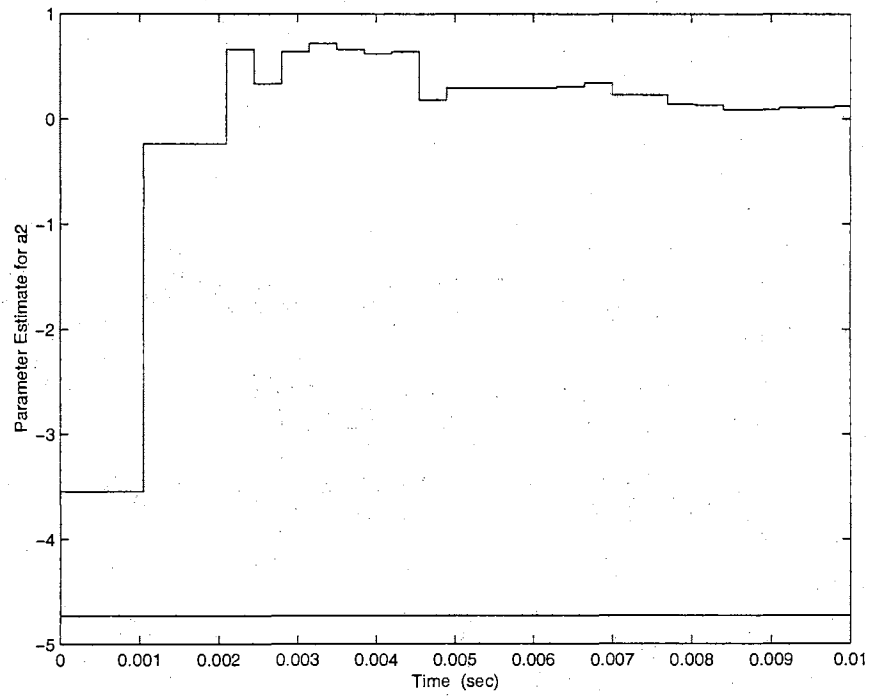


Figure C.7: APOS Parameter estimate of \bar{a}_2 With Retuning

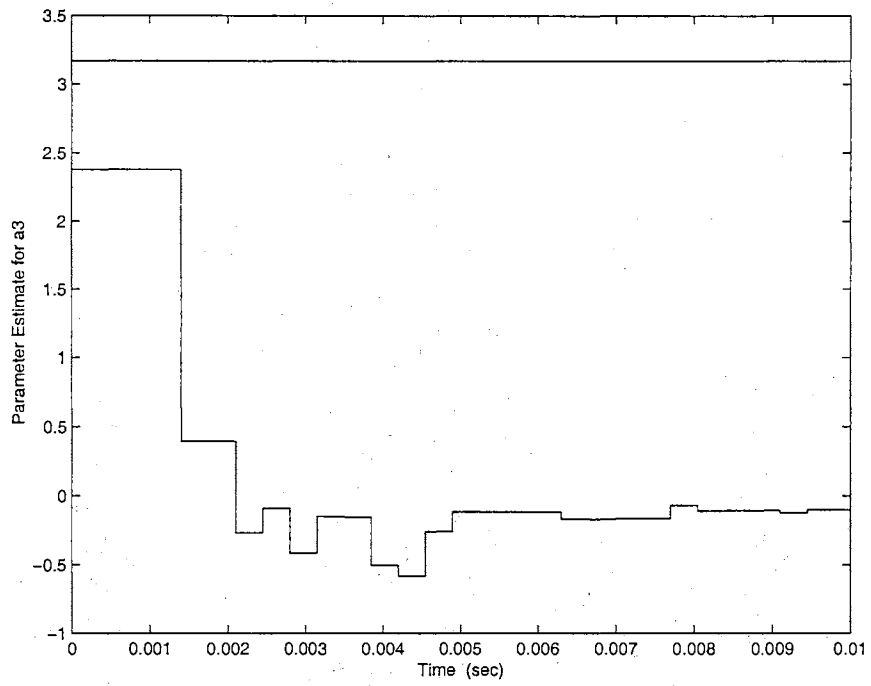


Figure C.8: APOS Parameter estimate of \bar{a}_3 With Retuning

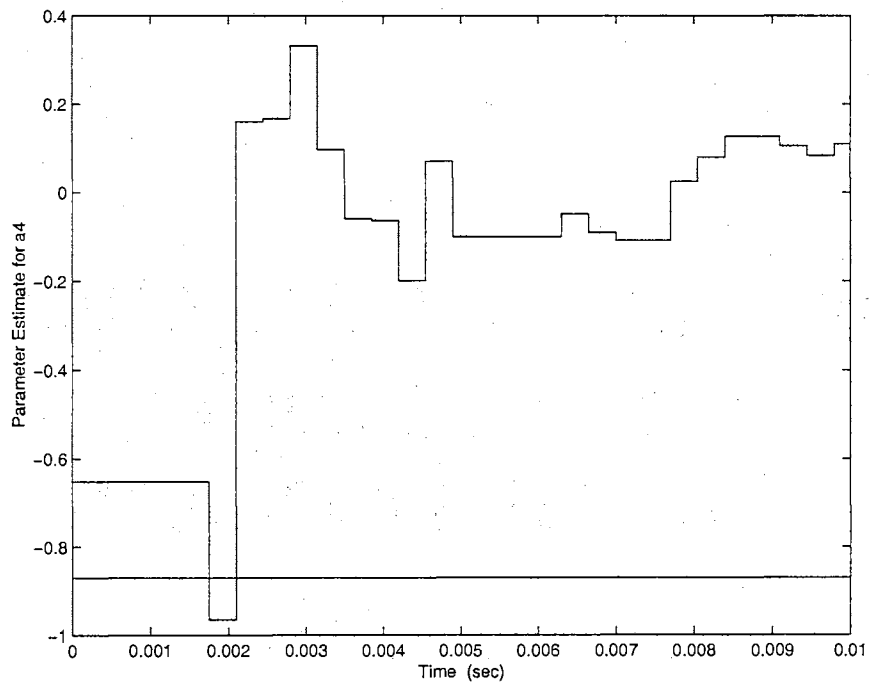


Figure C.9: APOS Parameter estimate of \bar{a}_4 With Retuning

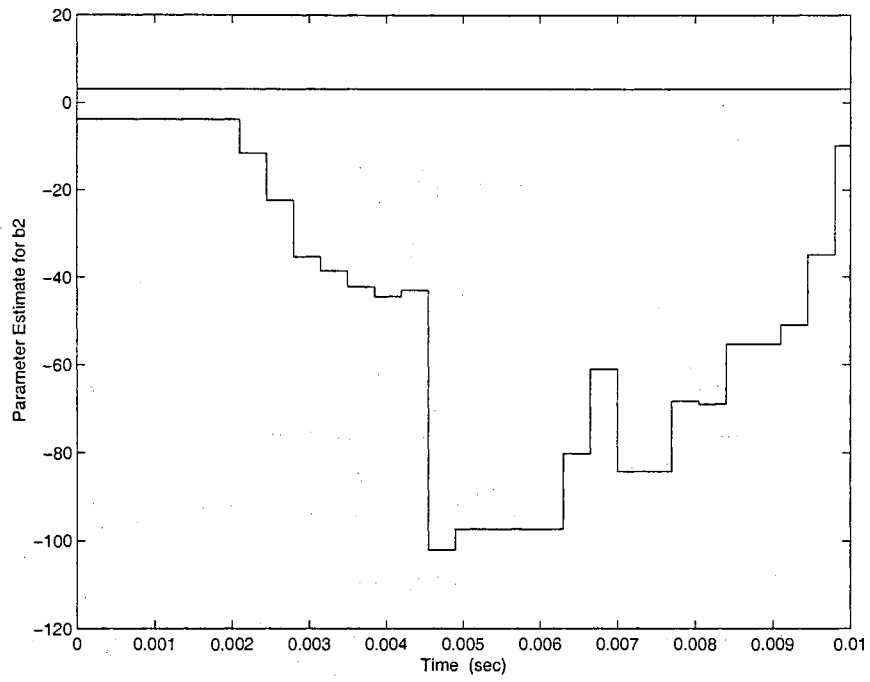


Figure C.10: APOS Parameter estimate of \bar{b}_2 With Retuning

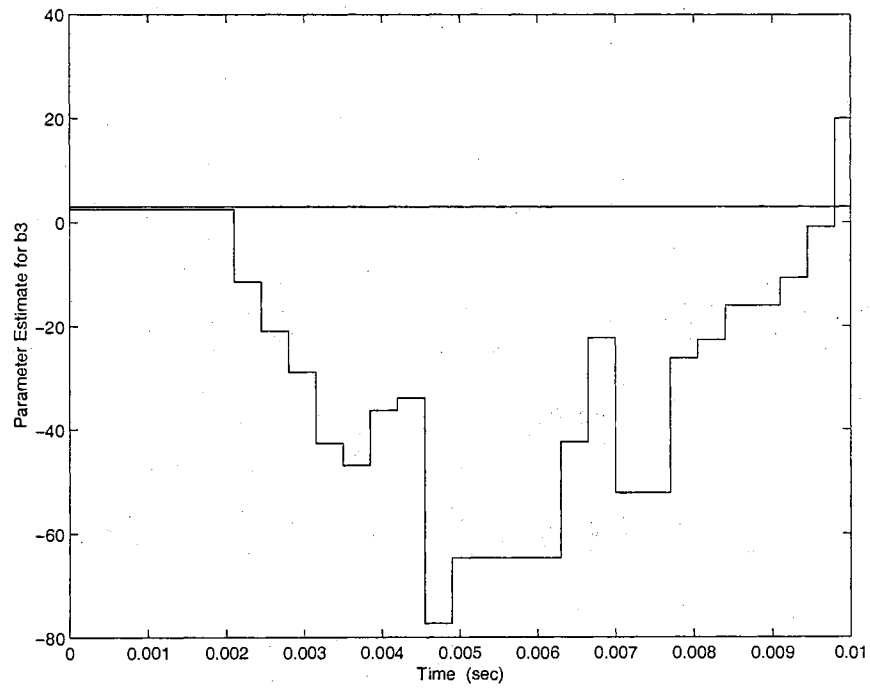


Figure C.11: APOS Parameter estimate of \bar{b}_3 With Retuning

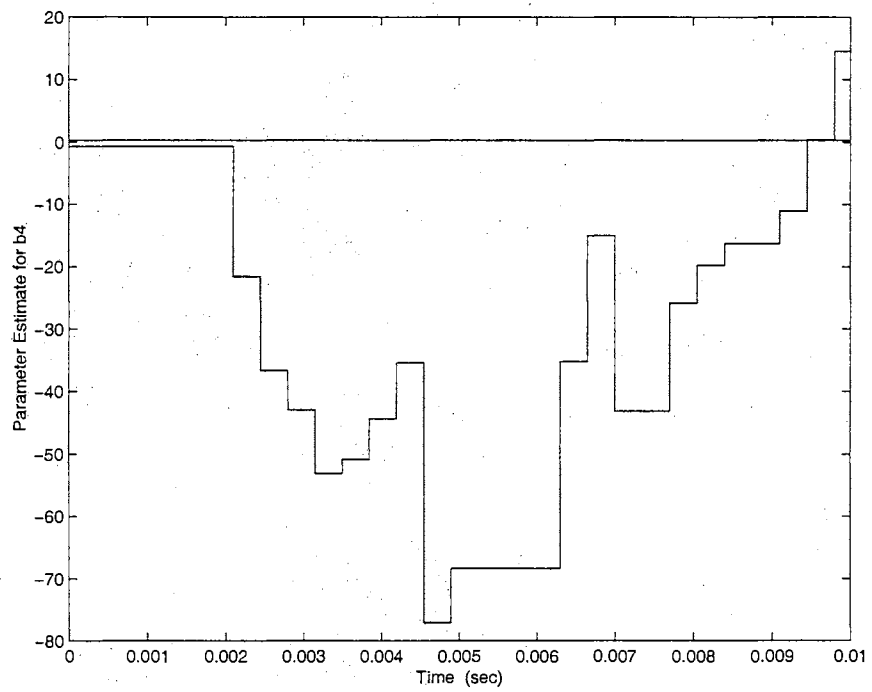


Figure C.12: APOS Parameter estimate of \bar{b}_4 With Retuning

Appendix D

Adapted Genetic Algorithm Code for MATLAB

This section of the appendix describes the genetic algorithm (GA) code [50] adapted specifically for the implementation of the Discrete Adaptive Observer (DAO) to the IBM Head/Disk Assembly (HDA) system used in this thesis. The code is package of M-files generated for use in MATLAB and as an off-line method to tune the DAO parameters λ and e_{thresh} for optimal estimation of the IBM HDA's states and parameters.

The original code is modified so as to incorporate a cost function which considers the output estimation error as well as the output velocity error, or the equivalent thereof, of the HDA actuator dynamics. The code is also modified by the initial population values and the mutation factors. The bounds of possible values of DAO design parameters are preset for reasons of system stability. In addition, instead of terminating the genetic algorithm program after a specified performance or "score", as it is referred to in the program, the GA ends after a specified number of generations. For more information, the reader is referred to [50].

The following is the list of M-files in the GA code:

- ga.m - MATLAB macros file for GA code
- start.m - defines initial settings for GA
- setup.m - defines the cost function used to determine the population "score"
- fit.m - ranks population according to fitness values
- mate.m - creates parent matrices and a child matrix

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% ga.m - main MATLAB executable file
%%      Calls: start.m, setup.m, fit.m, mate.m
%%      Determines maximum number of generations before program termination
%%      Output: generation number, population score
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear
start

gen=1;
stay=0;

while stay==0
    setup

    fit

    old_gains=gains;
    mate

    gen=gen+1;
    [q,w]=size(score);

    for col=1:w
        rk=1;
        sm=0;
        for row=1:w
            if score(col)>score(row)
                rk=rk+1;
            elseif score(col)==score(row)
                sm=sm+1;
            end
        end
    end
end
```

```

        if sm>=2
            last=sm-1;
            for sig=1:last
                best(rk+sig)=col;
            end
        end
        best(rk)=col;
    end

lesserr=best(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if gen >200
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    stay=1;
end

end

member=lesserr;
gen
old_pop_size

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% start.m - initialization file
%%      Sets:  mutation factors for lambda and thresh
%%              GA mutation rate
%%              initial population values
%%              choice of GA cost function
%%      Output: none
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

op_size=6; %INITIAL POPULATION SIZE
Pfactor=0.7; %PROBABILITY FACTOR OF SELECTING A SURVIVOR

```

```

old_pop_size=1;

```

```

MUTATE_THRESH=0.97644; % mutation factor for thresh
MUTATE_LAM=0.992651; % mutation factor for lambda

```

```

MUTATION_RATE=0.99; % GA mutation rate

```

```

ISETOT=100;

```

```

gains = [0.99999 0.0000031124;
         0.68451 0.00011410;
         0.59632 0.021203;
         0.469800 0.29012;
         0.869800 0.0000031124;
         0.78215 0.29012]; %INITIAL POPULATION VALUES

```

```

T=1; %SETTLING TIME

```

```

standard=0; %VARIABLE THAT ALLOWS FOR CHOICE OF COST FUNCTION.
            %standard=1 IMPLIES USE OF ISE COST FUNCTION
            %standard= anything else IMPLIES USE OF MODIFIED ISE.

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%  setup.m - determines cost function values (score) for population
%%      Calls: simulation macros for DAO implementation on IBM HDA
%%      Calculates for each set of DAO gains:
%%              error in output estimation
%%              error in equivalent velocity estimation
%%              population score (from predetermined cost function)
%%      Output: score
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

% CALCULATION OF THE INDIVIDUAL FITNESS

```

TOTERR=0;
for n=1:pop_size
    TOTERR=TOTERR +score(n);
end

```

```

AVGERR=(TOTERR/pop_size);
changes=pop_size/3;

```

```

for m=1:pop_size
    fit(m)=1-(score(m)/TOTERR);
end

```

% RANK EACH MEMBER FROM HIGHEST TO LOWEST FITNESS VALUE
% STANDARD ISE PERFORMANCE

```

for c=1:pop_size
    rank=1;
    same=0;
    for r=1:pop_size
        if fit(c)<fit(r)
            rank=rank+1;
        elseif fit(c)==fit(r)
            same=same+1;
        end
    end
end

```

```
        end
    end
    if same>=2
        last=same-1;
        for sig=1:last
            survive(rank+sig)=c;
        end
    end
    end
    survive(rank)=c;
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% fit.m - converts population scores to fitness values
%%      Uses:  score values calculated in setup.m
%%      Assigns fitness values to population
%%      Ranks members according to fitness values (highest to lowest)
%%      Output: none
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

j=old_pop_size;
while j<=pop_size
    lam=gains(j,1);
    thresh=gains(j,2);

```

```

%%%INSERT THE SIMULATION FILE HERE%%%%%%%%
DAO_initial_ga

```

```

%CALCULATION OF THE PERFORMANCE PARAMETER (SCORE)

```

```

ISETOT=0;
ISE_MODTOT=0;

e=yd-ya;
edot=sc(:,2)-va;
esqrd=e.*e;
edotsq=edot.*edot;
ISE(1)=esqrd(1)*t(1);
ISE_MOD(1)=(esqrd(1)+edotsq(1))*t(1);
[omeg,tau]=size(t);
for i=2:omeg
    ISE(i)=esqrd(i)*(t(i)-t(i-1));
    ISE_MOD(i)=(esqrd(i)+edotsq(i))*(t(i)-t(i-1));
    ISETOT=ISETOT + ISE(i);

```



```
ISE_MODTOT=ISE_MODTOT+ISE_MOD(i);
end
if standard==1
    score(j)=ISETOT;
else
score(j)=ISE_MODTOT;
end
j=j+1;
end

score
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% fit.m - creates parent matrices and offspring matrix
%%     Creates two parents
%%     Generates offspring
%%     Confirms mutation according to mutation rate
%%     Eliminates repetitive offspring
%%     Maintains population at an even number
%%     Output: random number generated to determine mutation process
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% CREATING THE PARENTS

```

```

father(1,1)=gains(survive(1),1);
father(1,2)=gains(survive(1),2);
selection=1;
hits=1;
f=2;
m=1;

survive(pop_size+1)=0;
for s=1:pop_size
    if s>=selection
        survive(s)=survive(s+1);
    end
end
a=1;

while m<=(pop_size/2)
    zones=pop_size-(hits);
    if zones==1
        prob(1)=1;
    else
        prob(1)=Pfactor;
    end
end

```

```

for k=2:zones
    prob(k)=Pfactor*(1-prob(k-1))+prob(k-1);
end
choice=rand;
prob(zones)=1;
for d=1:zones
    if choice<=prob(d)
        selection=d;
        hits=hits+1;
        break;
    end
end

if f==m
    father(f,1)=gains(survive(selection),1);
    father(f,2)=gains(survive(selection),2);
    f=f+1;
else
    mother(m,1)=gains(survive(selection),1);
    mother(m,2)=gains(survive(selection),2);
    m=m+1;
end

for s=1:pop_size
    if s>=selection
        survive(s)=survive(s+1);
    end
end
a=a+1;
end

```

%CREATING THE OFFSPRING

```

index=1;
count=1;
while index<=pop_size
    child(index,1)=father(count,1);
    child(index,2)=mother(count,2);

    index=index+1;

    child(index,1)=mother(count,1);
    child(index,2)=father(count,2);

```

```

        index=index+1;
        count=count+1;
end

old_pop_size=pop_size;

% CHECKING FOR MUTATION AND EXECUTING IT.

mutate=rand;

[cho,rho]=size(child);

if mutate<=MUTATION_RATE
    mutate

    child(cho+1,1)=MUTATE_LAM*child(cho-3,1);
    child(cho+1,2)=MUTATE_THRESH*child(cho-3,2);

    child(cho+2,1)=MUTATE_LAM*child(cho,1);
    child(cho+2,2)=MUTATE_THRESH*child(cho,2);
end

%gains=child;

new_pop_size=pop_size;
new_row=old_pop_size+1;

% CHECK TO SEE IF THE NEW GENE IS ALREADY IN THE POPULATION

for co=1:new_pop_size

    for ro=1:pop_size
        match=0;
        for p=1:2
            if child(co,p)==gains(ro,p)
                match=match+1;
            end
        end
        if match==2

```

```

                break;
            end
        end
    end
    if match<2
        for me=1:2
            gains(new_row,me)=child(co,me);
        end
        pop_size=pop_size+1;
        new_row=new_row+1;
    end
end

% TO ENSURE THAT THE POPULATION SIZE REMAINS EVEN

rmd=rem(pop_size,2);

if rmd>0
    pop_size=pop_size+1;

    gains(pop_size,1)=MUTATE_LAM*gains(pop_size-1,1);
    gains(pop_size,1)=MUTATE_THRESH*gains(pop_size,2);
end

%IF THE POPULATION SIZE DOES NOT CHANGE THEN MUTATE TWO MEMBERS

if pop_size==old_pop_size

    gains(pop_size+1,1)=MUTATE_LAM*gains(pop_size/2,1);
    gains(pop_size+1,2)=MUTATE_THRESH*gains(pop_size/2,2);

    gains(pop_size+2,1)=MUTATE_LAM*gains(pop_size-2,1);
    gains(pop_size+2,2)=MUTATE_THRESH*gains(pop_size-2,2);

    pop_size=pop_size+2;
end

[g_size,mu]=size(gains);
for th=1:g_size
    if gains(th,1)<0.4
        gains(th,1)=0.76844;
    end
end

```

```
elseif gains(th,1)>1
    gains(th,1)=0.76844;
end
if gains(th,2)>0.3
    gains(th,2)=0.15461;
end
end
```

VITA

May-Win L. Thein

Candidate for the Degree of

Doctor of Philosophy

Thesis: A PARALLEL OBSERVER SYSTEM FOR MULTIRATE STATE ESTIMATION AND ALIASED OUTPUT MEASUREMENTS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Rangoon, Burma, on November 6, 1969, the daughter of San and Wendy Thein.

Education: Graduated from North Penn Senior High School, Lansdale, Pennsylvania in June 1987. Received Bachelor of Science degree and Master of Science degree in Mechanical Engineering from Lehigh University, Bethlehem, Pennsylvania in June 1991 and September 1992, respectively. Completed the requirements for the Doctor of Philosophy degree with a major in Mechanical Engineering at Oklahoma State University in May 1999.

Experience: Employed by CertainTeed Corporation, Blue Bell, Pennsylvania, Summer 1988 and Summer 1990; employed by BFGoodrich Aerospace, Hatfield, Pennsylvania, Summer 1991; employed as a teaching assistant by the Mechanical Engineering and Mechanics Department, Lehigh University; employed as a teaching assistant, instructor and research assistant by the School of Mechanical and Aerospace Engineering, Oklahoma State University in 1992-1997, 1997, and 1997 to present, respectively.

Honors: Sigma Xi, Tau Beta Pi, Pi Tau Sigma, Phi Eta Sigma, Who's Who Among Students in American Colleges and Universities, Lehigh University Women's Club Award, Marcella S. Geltman Scholarship, Dean's List, National Merit Scholar.

Professional Memberships and Services: American Society of Mechanical Engineers; Institute of Electrical and Electronics Engineers; Reviewer for journal and conference papers, 1995-present.