ADAPTIVE CONTROL OF A LARGE ANTENNA DISH

by

Margaret H. LaBrecque

B.S., Electrical Engineering
Boston University
(1986)

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 5, 1989

© Massachusetts Institute of Technology 1989
All rights reserved

Signature of Author

Department of Electrical Engineering
June 5, 1989

Certified by

Leonard A. Gould
Professor, Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by

Michael A. Johnson
Technical Co-Supervisor, MIT Lincoln Laboratory

Accepted by

Anthony F. Hotz
Technical Co-Supervisor, MIT Lincoln Laboratory

Accepted by

Arthur C. Smith, Chairman
Departmental Committee on Graduate Students

JUL 11 1989

LIBRARIES
ADAPTIVE CONTROL OF A LARGE ANTENNA DISH

by

Margaret LaBrecque
B.S., Electrical Engineering
Boston University
(1986)

Submitted to the Department of Electrical Engineering and Computer Science
on June 5, 1989
in partial fulfillment of the requirements for the Degree of
Master of Science in Electrical Engineering

Abstract

Fixed compensators are often used for non-linear, time-varying plants because of
their simple structure and ease of implementation. Adaptive control has been suggested as
a method for maintaining a pre-specified closed loop performance in the presence of
changing plant dynamics. To date, successful practical application of adaptive control has
been limited. This thesis investigates the practicality of applying adaptive control to the 84-
foot Millstone antenna dish located in Westford, Massachusetts.

Thesis Supervisor: Leonard A. Gould
Title: Professor, Department of Electrical Engineering and Computer Science

Technical Co-Supervisor: Michael A. Johnson
Title: Staff Member, MIT Lincoln Laboratory
Technical Co-Supervisor: Anthony F. Hotz
Title: Staff Member, MIT Lincoln Laboratory
ACKNOWLEDGEMENT

Of all the experiences I have had as a graduate student at M.I.T., my tenure as a Research Assistant in the Control Systems Engineering Group at MIT Lincoln Laboratory has been the most educational and enjoyable. I am indebted to many people in Group 76 for their generosity and willingness to help.

As designer of the current Millstone antenna control algorithm, Tony Hotz has been of invaluable help. Without his patience and theoretical expertise, the completion of this thesis may not have been possible. I am deeply indebted to Mike Johnson for suggesting this thesis topic and for his encouragement and technical guidance. Although our contact was brief, I am very grateful to Professor Gould for his thorough evaluations of my thesis proposal and thesis. Thanks also go to Carl Much, the leader of Group 76, whose in-depth knowledge of the Millstone antenna system and accessibility were greatly appreciated.

My computationally intensive simulations could only be executed on the Micro-vax computers, the use of which was only made possible through the painstaking dedication of Stan Leigh, Jeff Brauch, and Mike Marino. When I had difficulties using the MacIntosh computers for word processing, I was told to use the G.A.S method -- the "go ask Stan" method. In addition to helping maintain the Micro-vax's, Mike Marino's knowledge of adaptive control and control systems in general was a great help. I also thank Professor Lena Valavani, who, as an adaptive control expert and a consultant to Group 76, helped me to understand and be aware of the potential problems associated with adaptive control.

Finally, I would like to thank my family for their encouragement and support.

This work was sponsored by the Department of the Air Force and performed at MIT Lincoln Laboratory. The views expressed in this document are those of the author and do not reflect the official policy or position of the U.S. Government or MIT Lincoln Laboratory. It is published for the exchange and stimulation of ideas.
# TABLE OF CONTENTS

Abstract 2

Acknowledgement 3

Table of Contents 4

List of Figures 6

1 Introduction 8

2 Adaptive Control 10
   2.1 Background 10
   2.2 Theoretical Formulation 11
   2.3 Parameter Adaptation Algorithm 16
      2.3.1 Variants of the General PAA 21

3 The Millstone Antenna and its Model 23
   3.1 Description of the Millstone Antenna 23
   3.2 Modelling Millstone 28
   3.3 Adjustments to the Model 32

4 Discrete Models 36

5 Simulation 42
   5.1 Simulation of the Adaptive Controller 42
   5.2 Simulation Experiments 47
      5.2.1 The General Case 47
      5.2.2 Unmodeled Dynamics 57
      5.2.3 Unpredictable Disturbances 64
      5.2.4 Large Magnitude Inputs 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Conclusions</td>
<td>68</td>
</tr>
<tr>
<td>References</td>
<td>71</td>
</tr>
<tr>
<td>Appendix A: Circuit Diagrams</td>
<td>73</td>
</tr>
<tr>
<td>Appendix B: Velocity Loop Transfer Function</td>
<td>78</td>
</tr>
<tr>
<td>Appendix C: Adjustments to Millstone Model</td>
<td>79</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2 Adaptive Control
2.1 Deterministic Case 15
2.2 Adaptive Case 17

3 The Millstone Antenna and its Model
3.1 Millstone Antenna Tower Pictorial Cutaway 24
3.2 Millstone System Structure 25
3.3 Millstone Radar Mechanical Schematic of Power Drive 27
3.4 Pole-zero Diagram 28
3.5 Block Diagram of Velocity Loop 29
3.6a Block Diagram of Velocity Loop Circuitry 30
3.6b Block Diagram of Meter Display and A/D Buffer Circuitry 31
3.7 Bode Plots of Actual Data and Model Data 33
3.8 Revised Pole-zero Diagram 34
3.9 Bode Plots of Actual Data and Revised Model Data 35

5 Simulation
5.1 Adaptive Case 48
5.2 Outputs for Adaptive and Fixed Compensation 49
5.3 Error and Control Input for Square Wave Command 51
5.4 Parameter Estimates for Square Wave Command 52
5.5 Error with alpha = 1.0 and alpha = 0.5 54
5.6 Effect of G(1/z) on Error 55
5.7 Error and Estimates for Sinusoidal Command 56
5.8 Errors for Adaptive and Fixed Compensation 58
5.9 Error with and without Anti-aliasing Pole 60
5.10 Output with and without Anti-aliasing Pole 61
5.11 Errors with Unmodeled Pole Pair at 5Hz 63
5.12 Error and Control Input with DC Offset on Output 65
Appendix A: Circuit Diagrams

A.1  Velocity Loop Circuitry  74-75
A.2  A/D Buffer and Meter Circuitry  76
A.3  Tachometer Lead Network  77
CHAPTER 1

INTRODUCTION

Greatly simplified control algorithms are frequently used for compensation of systems with complicated dynamics. For example, if the operating point of a system varies with time or frequency over a known range, the designer may choose to use a fixed compensation that is stable over the entire range of operating points. In doing so, optimal performance at all operating points is forfeited.

The production of reliable, flexible, and inexpensive microprocessors has made programmable digital compensators increasingly attractive to designers who wish to make on-line adjustments to their compensation parameters. These adjustable compensators enable the designer to achieve better performance in the presence of known changing dynamics or non-linearities. In the case of large or complicated mechanical structures, however, the dynamic characteristics of the plant are often poorly known in addition to being time-varying. To attain high performance in the presence of large and unpredictable variations of the plant's dynamic characteristics, "adaptive" control systems have been proposed.

The field of adaptive control addresses the question of how to make on-line compensation adjustments in a systematic manner. Since adaptive controllers are inherently time-varying and non-linear, the stability of the closed loop adaptive control system is dependent upon both reference and disturbance inputs [1]. Consequently, adaptive control algorithms often include demanding constraints on the input to the plant as well as the plant itself. These interesting features of adaptive control have led to many doctoral theses and to numerous heated debates on the practical stability of existing adaptive control algorithms.
This thesis will study the practicality of applying adaptive control to a large antenna dish, specifically the Millstone Antenna in Westford, MA designed at MIT Lincoln Laboratory. The software package, Matlab (The MathWorks, Inc.), will be used for analysis and simulation.
CHAPTER 2

ADAPTIVE CONTROL

2.1 Background

There are numerous adaptive control algorithms and configurations. Currently, "model reference" adaptive control is the most popular approach. The main feature of model reference adaptive control (MRAC) systems is the presence of a reference model which specifies the desired performance. The difference between the actual plant output and the desired performance (the performance error) is used as input to a parameter adaptation algorithm which alters the control parameters and/or estimated plant parameters such that some pre-specified control objective is asymptotically attained. This aspect of MRAC is termed "performance feedback".

The first recognized discrete-time MRAC algorithm for single-input, single-output systems was proposed by Ionescu and Monopoli in 1977 [2]. One of the fundamental characteristics of this design is the use of an augmented performance error to alleviate problems related to system delay and the inherent one-step delay present in discrete-time systems. Other basic features include a constant adaptation gain and a stability design based on the use of Lyapunov functions. Many of the presently existing discrete-time MRAC algorithms are variants of this basic design [3,4] and therefore will be termed "IM class" MRAC algorithms.

The MRAC algorithm chosen for investigation is not of the "IM class" and uses an \textit{a priori} and an \textit{a posteriori} performance error instead of an augmented error to circumvent problems of delay. In addition, a variable (as opposed to constant) adaptation gain is used
to improve system performance and the stability design is based on the use of hyperstability [5] concepts. These terms will become more familiar in later sections. This algorithm, presented by I.D. Landau and R. Lozano in [6] and by Landau in [7], incorporates parameter adaptation algorithms inspired by recursive identification techniques. However, the use of these types of parameter adaptation algorithms complicates the stability analysis beyond that which can be handled by the use of Lyapunov functions; hence, the alternate stability design. For ease of reference, the algorithm will be termed "Landau's algorithm".

Landau's algorithm is chosen over the IM class algorithms because of its more intuitive nature (and thus ease of successful implementation and variation) and because adequate performance was not realized from the few IM class discrete-time MRAC algorithms that were tried. This thesis will concentrate on the chosen algorithm's performance via simulations and will not delve into theoretical non-linear, time-varying systems analysis. Note that Landau's algorithm can only be applied to minimum phase plants because it incorporates zero cancellation.

2.2 Theoretical Formulation

Suppose the plant which we would like to control is described by

\[ A(z^{-1})y(k+1) = B(z^{-1})u(k) \]  

(2.1)

where \( u(k) \) is the input, \( y(k) \) is the output, and \( z^{-1} \) is a delay operator. \( A(z^{-1}) \) specifies the state transitions of the plant and \( B(z^{-1}) \) contains the input control information for the plant. These polynomials are described by
\[ A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n} = 1 + z^{-1} A^*(z^{-1}) \quad \text{and} \quad (2.2) \]
\[ B(z^{-1}) = b_0 + b_1 z^{-1} + \ldots + b_m z^{-m} = b_0 + z^{-1} B^*(z^{-1}). \quad (2.3) \]

Since the plant is assumed to be minimum phase, all of the zeros of \( B(z^{-1}) \) lie inside the unit circle and therefore can be cancelled without leading to instability.

The model reference adaptive control approach will adhere to the following design constraints:

**Tracking**

Given a reference sequence \( u_m(k) \), the control \( u(k) \) should be designed such that the plant output satisfies the difference equation,

\[ C(z^{-1}) y(k+1) = D(z^{-1}) u_m(k), \quad (2.4) \]

where

\[ C(z^{-1}) = 1 + c_1 z^{-1} + \ldots + c_n z^{-n} \quad (2.5) \]

is an asymptotically stable polynomial and

\[ D(z^{-1}) = d_0 + d_1 z^{-1} + \ldots + d_m z^{-m}. \quad (2.6) \]

In other words, \( D(z^{-1}) / C(z^{-1}) \) is the **desired** closed loop model of the system.

**Regulation**

In regulation \( (u_m(k) = 0) \), an initial disturbance \((y(0) \neq 0)\) should be eliminated with dynamics defined by

\[ G(z^{-1}) y(k+1) = 0, \text{ for all } k > 0, \quad (2.7) \]
where
\[ G(z^{-1}) = 1 + g_1z^{-1} + \ldots + g_nz^{-n} = 1 + z^{-1}G^*(z^{-1}) \] (2.8)

is an asymptotically stable polynomial. Hence, \( G(z^{-1}) \) defines the closed loop adaptation response (e.g., how fast the system adapts to a disturbance).

Considering Eqn.(2.4), a reference model for tracking which will generate the desired trajectory will be defined as

\[ C(z^{-1})y_m(k+1) = D(z^{-1})u_m(k) \] (2.9)

where \( y_m(k+1) \) is the output of the reference model for tracking. The constraints of Eqn.(2.4) and Eqn.(2.7) will be satisfied if

\[ G(z^{-1}) [y(k+1) - y_m(k+1)] = 0. \] (2.10)

Adding the term \( G(z^{-1})y(k+1) \) to both sides of Eqn.(2.1) and taking into account Eqn.(2.2) and Eqn.(2.8), the plant equation can be reparametrized as

\[ G(z^{-1})y(k+1) = [G^*(z^{-1}) - A^*(z^{-1})] y(k) + B(z^{-1})u(k) \]
\[ = R(z^{-1})y(k) + b_0u(k) + B^*(z^{-1})u(k-1) \] (2.11)

where
\[ R(z^{-1}) = r_0 + r_1z^{-1} + \ldots + r_{n-1}z^{-(n-1)}, \] (2.12)
\[ r_i = g_{i+1} - a_{i+1}. \] (2.13)
Solving Eqn.(2.11) with respect to \( u(k) \) and taking into account Eqn.(2.10), the resulting desired control is

\[
u(k) = (1/b_0) [G(z^{-1})y_m(k+1) - B^*(z^{-1})u(k-1) - R(z^{-1})y(k)].
\]

(2.14)

The resulting control scheme for the case where \( A(z^{-1}) \) and \( B(z^{-1}) \) are deterministic (the non-adaptive case) is shown in Figure 2.1. Qualitatively, the zeros of the plant are cancelled by the \( z^{-1}B^*(z^{-1}) \) feedback and the poles are then freely adjusted by the \( R(z^{-1}) \) feedback. The \( (1/b_0) \) term in the forward path serves only to maintain the original DC gain. The dashed line path represents the \textit{a priori} error,

\[
e^0(k) = G(z^{-1}) [y(k) - y_m(k)]
\]

(2.15)

which is identically zero in the deterministic case. Equation (2.15) more clearly indicates why \( G(z^{-1}) \) specifies the closed loop adaptation response.

If the plant parameters are incompletely known (the adaptive case), \( e^0(k) \) will be different from zero and, as such, can be used in an adaptation algorithm with the following constraint:

\[
\lim_{k \to \infty} e^0(k+1) = \lim_{k \to \infty} G(z^{-1}) [y(k+1) - y_m(k+1)] = 0.
\]

(2.16)

If this constraint is satisfied, the control objective of Eqn.(2.10) will be attained asymptotically. In order to incorporate the estimates generated by the adaptation algorithm, the fixed controller of Eqn.(2.14) must be replaced by an adjustable controller. As the error of Eqn.(2.16) is caused by misalignment of the control parameters, one might
Figure 2.1: Deterministic Case
construct an adaptive controller which has the same form as Eqn.(2.14) except with the fixed parameters replaced by their adjustable estimates, i.e.,

\[
u(k) = \frac{1}{b_0(k)} [G(z^{-1})y_m(k+1) - \hat{B}^*(k,z^{-1})u(k-1) - \hat{R}(k,z^{-1})y(k)], \tag{2.17}
\]

where

\[
\hat{B}(k,z^{-1}) = \hat{b}_0(k) + \hat{b}_1(k)z^{-1} + \ldots + \hat{b}_m(k)z^{-m}
= \hat{b}_0(k) + z^{-1} \hat{B}^*(k,z^{-1}) \tag{2.18}
\]

and

\[
\hat{R}(k,z^{-1}) = \hat{r}_0(k) + \hat{r}_1(k)z^{-1} + \ldots + \hat{r}_{n-1}(k)z^{-(n-1)},
\hat{r}_i(k) = g_{i+1^{-}} \hat{a}_{i+1}(k). \tag{2.19}
\]

Figure 2.2 is a block diagram of the model reference adaptive controller which has been outlined above.

2.3 Parameter Adaptation Algorithm

As stated earlier, the parameter estimates of the adjustable controller are updated via a parameter adaptation algorithm (PAA). The PAA (as in Figure 2.2) generates the estimates \(\hat{R}(k,z^{-1})\) and \(\hat{B}(k,z^{-1})\) and is of the recursive least squares (RLS) type. Let us define a parameter vector,

\[
\hat{\Theta}(k)^T = [\hat{b}_0(k), \ldots, \hat{b}_m(k), -\hat{a}_1(k), \ldots, -\hat{a}_n(k)] \tag{2.20}
\]
Figure 2.2: Adaptive Case
and an observation vector,

$$\Phi(k)^T = \{ u(k), \ldots, u(k-m), y(k), \ldots, y(k-n+1) \} \quad (2.21)$$

such that $\hat{\Theta}(k)^T \Phi(k)$ equals $\hat{y}(k+1)$, the estimated output at time $(k+1)$. The classical least-squares identification algorithm attempts to minimize the cost function,

$$J(k) = \sum_{i=1}^{k} [y(i) - \hat{\Theta}(k)^T \Phi(i - 1)]^2 . \quad (2.22)$$

Notice that the cost function is just the square of the cumulative error up to time $k$ with $\hat{\Theta}(k)$, the most recent (or best) estimate of $\Theta$, replacing all the previous values of $\hat{\Theta}(i)$. At each step, the estimates move closer to their correct values. To find the updating formula for $\hat{\Theta}(k)$, the derivative of the cost function, $J(k)$ is taken with respect to the estimated parameter vector, $\hat{\Theta}(k)$, and set equal to zero, i.e.,

$$\frac{\delta J(k)}{\delta \hat{\Theta}(k)} = 0. \quad (2.23)$$

The solution to Eqn.(2.23) is

$$\hat{\Theta}(k) = F(k) \sum_{i=1}^{k} y(i) \Phi(i - 1), \quad (2.24)$$

where

$$F(k)^{-1} = \sum_{i=1}^{k} \Phi(i - 1) \Phi(i - 1)^T \quad (2.25)$$
and is always positive definite. F(k) is termed the "adaptation gain matrix". \( \hat{\Theta}(k+1) \) can be computed recursively knowing \( \hat{\Theta}(k) \) in the following manner [8]:

\[
\begin{align*}
\hat{\Theta}(k+1) &= \hat{\Theta}(k) + F(k)\Phi(k)e(k+1), \\
F(k+1)^{-1} &= F(k)^{-1} + \Phi(k)^T\Phi(k), \quad F(0) > 0,
\end{align*}
\]  

and

\[
e(k+1) = e^0(k+1) / (1 + \Phi(k)^TF(k)\Phi(k)).
\] (2.28)

The following definitions have been assumed:

\[
\begin{align*}
e^0(k+1) & \triangleq y(k+1) - \hat{\Theta}(k)^T\Phi(k) = y(k+1) - \hat{y}^0(k+1) \quad \text{and} \\
e(k+1) & \triangleq y(k+1) - \hat{\Theta}(k+1)^T\Phi(k) = y(k+1) - \hat{y}(k+1)
\end{align*}
\] (2.29, 2.30)

where \( e^0(k+1) \) is the \textit{a priori} error and \( e(k+1) \) is the \textit{a posteriori} error. Equation (2.28) relates \( e(k+1) \) to \( e^0(k+1) \) without having to first know \( \hat{\Theta}(k+1) \). Notice also that Eqn.(2.27) requires a matrix inversion. Fortunately, this inversion can be circumvented via the "matrix inversion lemma" [9] and reformulated as

\[
F(k+1) = F(k) - \frac{F(k)\Phi(k)\Phi(k)^TF(k)}{1 + \Phi(k)^TF(k)\Phi(k)}.
\] (2.31)

Restating Eqns.(2.26) through (2.28),

\[
\begin{align*}
\hat{\Theta}(k+1) &= \hat{\Theta}(k) + F(k)\Phi(k)e(k+1), \\
F(k+1) &= F(k) - \frac{F(k)\Phi(k)\Phi(k)^TF(k)}{1 + \Phi(k)^TF(k)\Phi(k)}, \quad F(0) > 0,
\end{align*}
\] (2.32, 2.33)

and
\[ e(k+1) = e^o(k+1) / (1 + \Phi(k)^T F(k) \Phi(k)). \quad (2.34) \]

There are numerous variants of the general PAA outlined above. Before addressing this issue, the convergence properties of the general PAA will be examined.

The PAA of Eqns. (2.32) through (2.34) will converge, provided that the system input, \( u(k) \), is *weakly persistently exciting* of order \( n \). By definition, this means that \( u(k) \) must be a stationary input whose discrete spectral distribution is non-zero at \( n \) points or more [10]. This proposition, however, does not strictly address the issue at hand. The PAA outlined above assumes that the *a priori* error, \( e^o(k) \), is the actual plant output minus the estimated plant output, \( [y(k+1) - \hat{y}^o(k+1)] \). In Landau's MRAC algorithm of Section 2.2, the *a priori* error is the filtered actual plant output minus the desired plant output, \( G(z^{-1})[y(k+1) - y_m(k+1)] \), and this error is used to drive \( [y(k+1) - \hat{y}^o(k+1)] \) to zero. Hence, the well established convergence proofs of the general PAA (used in an identification environment as opposed to a control environment) cannot be directly applied to the MRAC problem.

It is necessary that the PAA of Figure 2.2 achieve its goal, \( [y(k+1) - \hat{y}^o(k+1)] --> 0 \), for the constraint of Eqn.(2.16),

\[
\lim_{k \to \infty} e^o(k+1) = \lim_{k \to \infty} G(z^{-1}) [y(k+1) - y_m(k+1)] = 0, \quad (2.35)
\]

to be satisfied. Note, it is not necessary for the parameter estimates to converge to their true values. It is only necessary for the estimated output, \( \hat{y}^o(k+1) = \hat{\Theta}(k) \Phi(k) \), to converge to the actual output, \( y(k+1) \). The value of the parameter vector, \( \hat{\Theta}(k) \), that satisfies \( y(k+1) = \hat{y}^o(k+1) \) is not unique.
Given these considerations, in simulation the MRAC algorithm will be driven with "worst case" types of reference inputs (e.g., pure sine waves, constant inputs) to determine if the system can be sufficiently excited. Note that it is the control input, \( u(k) \), not the reference input, \( u_m(k) \), which must be sufficiently exciting. Refer back to Figure 2.2 for clarification.

2.3.1: Variants of the General PAA

The adaptation gain matrix of Eqn.(2.33) monotonically decreases in time, i.e.,

\[
x^T F(k+1)x \leq x^T F(k)x,
\]

where \( x \) is a vector of compatible dimension. Thus, new measurements are weighted less and less. Consequently, this adaptation gain, although appropriate for identifying static parameters, is inappropriate for estimating time-varying parameters. To achieve a more "alert" PAA, one could introduce a "forgetting factor", \( \lambda_1 \), to the cost function:

\[
J(k) = \sum_{i=1}^{k} \lambda_1^{k-i} \left[ y(i) - \hat{\theta}(k)^T \Phi(i - 1) \right]^2
\]

where \( 0 < \lambda_1 \leq 1 \). Here, old measurements are weighted less than new measurements. \( F(k) \) is updated as

\[
F(k+1) = \frac{1}{\lambda_1} \left[ F(k) - \frac{F(k)\Phi(k)\Phi(k)^T F(k)}{1 + \Phi(k)^T F(k) \Phi(k)} \right].
\]
The problem with this revised adaptation gain matrix is that some components of \( F(k) \) can grow too much.

Another way to keep the PAA alert is to simply make \( F(k) \) a constant,

\[
F(k) = F, \quad \text{for all } k. \tag{2.39}
\]

The disadvantage of this method is that the estimates will not necessarily move in the *best* direction at each step as in the case of the general PAA. A general form for updating the adaptation gain matrix, \( F(k) \) which overcomes these difficulties and allows the PAA to operate in various situations is

\[
F(k+1) = \frac{1}{\lambda_1(k)} \left[ F(k) - \frac{F(k)\Phi(k)\Phi(k)^TF(k)}{\lambda_1(k)\lambda_2(k) + \Phi(k)^TF(k)\Phi(k)} \right] \tag{2.40}
\]

where \( 0 < \lambda_1(k) \leq 1 \) and \( 0 < \lambda_2(k) \leq 2 \). Although \( \lambda_1(k) \) tends to increase the adaptation gain, \( \lambda_2(k) \) tends to decrease the adaptation gain. Hence, \( \lambda_1(k) \) and \( \lambda_2(k) \) can be tailored to suit various situations.

One variant of Eqn.(2.40) that yields good performance for time-varying parameters is the constant trace PAA. Choosing \( \lambda_1(k)/\lambda_2(k) \) to be a constant equal to \( \alpha \), one has

\[
F(k+1) = \frac{1}{\lambda_1(k)} \left[ F(k) - \frac{F(k)\Phi(k)\Phi(k)^TF(k)}{\alpha + \Phi(k)^TF(k)\Phi(k)} \right] \tag{2.41}
\]

where \( 0.5 \leq \alpha \leq 1 \) and \( \lambda_1(k) \) is chosen at each step such that the trace of \( F \) remains constant. This algorithm stays alert while strictly controlling the growth of \( F \). The constant trace algorithm preserves the least squares *direction* of the correction of the current estimate but alters the *magnitude* of the correction. For the reasons stated above, the constant-trace PAA will be used in implementing Landau's algorithm.
CHAPTER 3

THE MILLSTONE ANTENNA AND ITS MODEL

3.1 Description of the Millstone Antenna

Operational in the early 1960's, the Millstone Antenna was one of the first antennas to successfully track man-made satellites. Since then, the system has been updated periodically -- one of the most recent revisions being the conversion from analog to digital compensation. The digital compensation is implemented via an 8086/8087 Intel single board computer [11], the presence of which makes the implementation of discrete adaptive control very convenient.

The receiver dish of the Millstone antenna is 84 feet in diameter and is mounted approximately 100 feet off the ground as in Figure 3.1 [12]. The dish can rotate 360 degrees in azimuth and 90 degrees in elevation. In its normal mode of operation, the antenna system receives position and velocity commands from a Harris computer which processes analog voltages from a monopulse receiver. There is also a manual control panel and remote station where position and velocity commands can be set by hand. The commands are fed to the control microprocessor and an appropriate drive signal is generated which activates the gear train in the desired manner. Sensors located on the mount measure the response of the antenna to the control input generated by the microprocessor. In turn, the drive signal is adjusted so that the commanded position and velocity are achieved. The general structure of the system is shown in Figure 3.2.

The elevation axis exhibits significant non-linearities and is therefore the proposed candidate for adaptive control. Specifically, the response of the elevation axis is a function
Figure 3.1: Millstone Antenna Tower Pictorial Cutaway
Figure 3.2: Milestone System Structure
of elevation angle. Elevation motor rotation is transferred through pinion and sector gears to the elevation torque tube (see Figure 3.3 [13]) which supports the main reflector, sub-
reflector, and RF feed assembly, and rotates on ball bearings mounted in pillow blocks. The drive system for the elevation axis contains two 30-horsepower motors, each driving a separate reduction gear train. As in Figure 3.3, two pinions drive individual sector gears. The overall reduction ratio from motor to antenna is 2230:1.

The entire elevation assembly is counterweighted by concrete blocks housed in welded frames supported by the torque tube. These counterweights have been adjusted to minimize the unbalance (the difference between the moments generated by the counterweights and the moments generated by the antenna dish assembly) and the most recent measurements show a maximum unbalance of 9400 ft-lbs with an unstable equilibrium of 55 degrees [14]. Evidently, the unbalance is a function of elevation angle and is hence a major contributor to the angle dependency of the response of the elevation axis. The adverse effects of this unbalance on the response are somewhat reduced by the use of counter-torque to pre-load the gear train. [The use of counter-torque greatly alleviates frictional dead zone and gear backlash as well.]

In addition to the unbalance, there are mechanical non-linearities (e.g., torsional compliance) and structural modes that are not included in the model of the system. These factors also contribute to the angle dependency and unpredictability of the response. Furthermore, environmental conditions such as wind, snow, and ice induce unmodeled external torques and disturbances that are inherently time-varying. In light of these circumstances, adaptive control may be an appropriate compensation approach.
Figure 3.3: Millstone Radar Mechanical Schematic of Power Drive
3.2 Modelling Millstone

As the velocity loop contains all of the relevant system dynamics beyond the digital controller, the adaptive controller will not be concerned with anything outside of this loop. The most current model of the velocity loop is shown in Figure 3.5 [15]. Block diagrams of the velocity loop, meter display and A/D buffer circuitry are shown in Figures 3.6(a-b). Detailed circuit diagrams are contained in Appendix A for completeness. The derived transfer function in terms of the system parameters is contained in Appendix B. The closed loop poles and zeros of the velocity loop are plotted in Figure 3.4:

\[ s_{n1} = -6.37 \]
\[ s_{n2} = -92.2 \]
\[ s_{p1} = -1.86 \pm 3.02 \, i \]
\[ s_{p2} = -114 \pm 63.7 \, i \]
\[ s_{p3} = -23,700 \]

**Figure 3.4: Pole-zero Diagram**
Figure 3.5: Block Diagram of Velocity Loop
Figure 3.6a: Block diagram of Velocity Loop Circuitry
Figure 3.6b: Block diagram of Meter Display and A/D Buffer Circuitry
Clearly, the dominant dynamics of the system lie below 7 rad/sec (about 1 Hz). A reduced-order model of the velocity loop will be used in the model reference adaptive control algorithm if possible.

3.3 Adjustments to the Model

Bode plots of the actual velocity loop and the velocity loop model are shown in Figure 3.7. The solid line represents the real data and the dashed line represents the model data. There is significant discrepancy. Since both sets of data exhibit second-order plant characteristics in the frequency range of interest (below a few Hz), it would probably suffice to simply derive a second-order model from the actual velocity loop data. However, since it is not known at this point whether a reduced-order model can be used in Landau's algorithm, an alternate method will be used.

As it is not known which velocity loop parameters are in error and since, from the transfer function of the velocity loop given in Appendix B, the dominant poles are determined by many parameters, all the parameters (including those parameters which are in all likelihood correct) will be varied by ±10% such that the current model more closely matches the actual data. [A few parameters are adjusted by ±15% because of less confidence in their modeled values or because they are themselves dependent on numerous parameter values.] It is hoped that this method of adjustment will be less biased than that of adjusting a few parameter values radically.

The Bode plots of Figure 3.7 indicate that the actual velocity loop has less peaking and a higher -3dB frequency than the model. Consequently, the directions of the model parameter adjustments are determined by how each parameter affects the peaking and roll-off frequency. The actual parameter adjustments and revised transfer function derivation
Figure 3.7: Bode Plots of Actual Data and Model Data
are indicated in the Matlab file, newmodel.m, provided in Appendix C. Bode plots of the revised velocity loop model and the actual velocity loop are shown in Figure 3.9. Now, both the actual data and the model data have a bandwidth of about 0.89Hz and very little peaking. The revised pole-zero diagram is shown in Figure 3.8:

Figure 3.8: Revised Pole-zero Diagram
Figure 3.9: Bode Plots of Actual Data and Revised Model Data
CHAPTER 4

DISCRETE MODELS

In order to implement Landau's discrete model reference adaptive control algorithm described in Chapter 2, discrete representations of the actual velocity loop, $A(z^{-1})/B(z^{-1})$, the desired velocity loop response, $D(z^{-1})/C(z^{-1})$, and the closed loop adaptation response, $G(z^{-1})$, must be formulated. For now, a reduced-order model of the velocity loop will be assumed to facilitate matters. The reduced-order velocity loop model retains the dominant dynamics below a few Hz and is specified as follows:

$$\frac{A(s)}{B(s)} = \frac{3.1608 \ (s + 5.79)}{(s + 3.03 + 3.02i)(s + 3.03 - 3.02i)} \quad (4.1)$$

where $A(s)/B(s)$ represents the continuous-time transfer function between the output mount velocity and the commanded input velocity.

The digital control board samples the output data and updates the input data at 50Hz. Therefore, a zero-order hold (ZOH) of $T = 20$ms ($1/50$Hz) should be added to the model as in Figure 4.1:

![Figure 4.1: Addition of Zero-Order Hold](image)
To discretize $A(s)/B(s)$, the Matlab function "c2d" is used. This function takes as
its input the state-space description of $A(s)/B(s)$,

$$\dot{x} = Ax + Bu \quad (4.2)$$
$$y = Cx + Du, \quad (4.3)$$

where

$x$ is a vector representing the states of the system,

$\dot{x}$ represents the derivative of $x$,

$u$ is the input to the system,

$y$ is the output of the system,

and the bold-face type signifies a matrix. Next, "c2d" applies a ZOH of the specified
sampling period to the system inputs. Finally, the continuous-time state-space description
is discretized via the Padé approximation $[16]$ as

$$x(k+1) = \Phi x(k) + \Gamma u(k), \quad (4.4)$$

where

$$\Phi \sim e^{At} \quad \text{and}$$

$$\Gamma \sim \left( \int_0^T e^{At} dt \right) B. \quad (4.6)$$

The output matrices describing $y$ do not change (see $[17]$ for a complete description).

Using the discrete-time state-space description, the discrete transfer function,$A(z^{-1})/B(z^{-1})$ in terms of the delay operator, $z^{-1}$, can easily be found. This "roundabout"
continuous to discrete conversion is used because Matlab conveniently includes the ZOH and quickly yields the discretized system. One could equivalently use the bilinear transformation on the continuous-time system plus ZOH to achieve the same result. Later, when the addition of a ZOH is not desired, the bilinear and backwards difference transformations will be used for discretization. Using the Matlab "c2d" function in the manner stated above, the discrete transfer function is computed as

$$\frac{A(z^{-1})}{B(z^{-1})} = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{6.2982 \times 10^{-2} z^{-1} - 5.6091 \times 10^{-2} z^{-2}}{1 - 1.8791 z^{-1} + 0.8860 z^{-2}}. \quad (4.7)$$

Since $z^{-1}$ is the delay operator, an equivalent time domain system description can be found as

$$y(k+1) = 1.8791 y(k) - 0.8860 y(k-1) + 6.2982 \times 10^{-2} u(k) - 5.6091 \times 10^{-2} u(k-1) \quad (4.8)$$

where $u(k)$ is the input at time $k$ and $y(k+1)$ is the output.

Next, the desired response of the velocity loop, $D(z^{-1})/C(z^{-1})$, must be formulated and discretized. Currently, the controller on Millstone increases the effective velocity loop bandwidth to 1.5Hz from 0.89Hz. The bandwidth of 1.5Hz will be maintained; furthermore, the desired closed response is specified to be purely second-order (no zeros) with a damping ratio, $\delta$, of 0.707. These constraints are satisfied by the following continuous-time transfer function:

$$\frac{D(s)}{C(s)} = \frac{1}{s^2/\omega_n^2 + 2\delta s/\omega_n + 1}, \quad (4.9)$$
where the natural frequency, \( \omega_n \), equals 9.42 rad/sec (1.5Hz). To discretize \( D(s)/C(s) \), the bilinear transformation,

\[
s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}
\]

(4.10)

where \( T = 20\text{ms} \) is used. [The Matlab "c2d" conversion function would needlessly add another \( ZOH \) if used.] The resulting discrete transfer function is

\[
\frac{D(z^{-1})}{C(z^{-1})} = \frac{d_0 + d_1 z^{-1} + d_2 z^{-2}}{1 + c_1 z^{-1} + c_2 z^{-2}}
\]

\[
= \frac{7.7721 \times 10^{-3} + 1.5544 \times 10^{-2} z^{-1} + 7.7721 \times 10^{-3} z^{-2}}{1 - 1.7356 z^{-1} + 0.76671 z^{-2}}.
\]

(4.11)

Since the model reference adaptive controller would be hard-pressed to make the actual velocity loop plus \( ZOH \) equal the desired model with no added delay, a unit delay (20ms) is introduced in the desired model, i.e.,

\[
\frac{D(z^{-1})}{C(z^{-1})} = z^{-1} \frac{D(z^{-1})}{C(z^{-1})} = \frac{d_0 z^{-1} + d_1 z^{-2} + d_2 z^{-3}}{1 + c_1 z^{-1} + c_2 z^{-2}}.
\]

(4.12)

In discrete-time, this is equivalent to the system description,

\[
y_m(k+1) = 1.7356 y_m(k) - 0.76671 y_m(k-1) + 7.7721 \times 10^{-3} u_m(k) \\
+ 1.5544 \times 10^{-2} u_m(k-1) + 7.7721 \times 10^{-3} u_m(k-2)
\]

(4.13)

where \( u_m(k) \) is the input at time \( k \) and \( y_m(k+1) \) is the output.
Finally, the closed loop adaptation response, \( G(z^{-1}) \), must be formulated and discretized. Notice from Eqn.(2.19),

\[
\hat{r}_i(k) = g_{i+1} \hat{a}_{i+1}(k),
\]

(4.14)

that the coefficients of \( G(z^{-1}) \) adjust the location of the poles of the closed loop adaptive system. As an arbitrary starting point, one can choose the continuous-time bandwidth of \([1/G(s)]\) to be a factor of 5 to 10 times greater than the plant bandwidth. Since the present "plant" or velocity loop, \( A(s)/B(s) \), has a bandwidth of about 0.89Hz, \( G(s) \) is chosen to have a double root at 9.55Hz (60 rad/sec). This choice puts the bandwidth of \([1/G(s)]\) at around 6Hz. Keep in mind that this is merely a starting point. The adaptation response time can be shortened by moving the poles of \([1/G(s)]\) in (consider Eqn.(4.14)) and lengthened by moving the poles out. The adaptation response can also be varied by adjusting the number of poles of \([1/G(s)]\). One extreme case is the "dead-beat" controller, \( G(z^{-1}) = 1 \), which produces the fastest adaptation response time but is usually severely underdamped.

To discretize \( G(s) \), the backwards difference method is preferred to the bilinear transformation because it maintains \( G(z^{-1}) \) in polynomial as opposed to transfer function form. To see this effect, consider Eqn.(4.10). The backwards difference method maps \( s \) to \( z^{-1} \) as

\[
s = \frac{1 - z^{-1}}{T}.
\]

(4.15)

Using the backwards difference method transformation,

\[
G(s) = (s + 60)(s + 60),
\]

(4.16)
maps to
\[
G(z^{-1}) = 1 + g_1 z^{-1} + g_2 z^{-2},
\]
\[
= 1 - 0.90909z^{-1} + 0.20661z^{-2}, \tag{4.17}
\]

which filters the output \(y(k+1)\) as
\[
y(k+1) = y(k+1) - 0.90909y(k-1) + 0.20661y(k-2). \tag{4.18}
\]

Note that Landau's MRAC algorithm requires that the first coefficient of \(G(z^{-1})\) equal one. In implementation, whatever constant factor is necessary for \([1/G(z^{-1})]\) to have unity DC gain is used as a scale factor in the algorithm (the adaptation gain matrix is multiplied by this scale factor).
CHAPTER 5

SIMULATION

In order to verify the performance and investigate the stability of Landau's algorithm, numerous simulations must be performed. The need for and advantage of simulation will become clear as the simulated plant (velocity loop) is arbitrarily varied in ways the actual velocity loop could not be (or, at least, could not be in a safe context). Naturally, if the algorithm is unstable, it is preferable to find this out in simulation. As mentioned in the introduction, MRAC systems are inherently non-linear and time-varying and therefore their stability is dependent on, among other things, the reference input. As stated in Chapter 2, worst case type inputs (those inputs without rich frequency content) will be used in attempts to drive Landau's algorithm unstable. Remember, it is the control input, \( u(k) \), not the user designated input, \( u_m(k) \), which must be sufficiently exciting.

5.1: Simulation of the Adaptive Controller

Given the theoretical formulation of Chapter 2 and the discrete models of Chapter 4, one can now proceed with the actual software implementation of Landau's MRAC algorithm. The control law to be implemented is as in Eqn.(2.17):

\[
u(k) = \frac{1}{b_0(k)} [ G(z^{-1}) y_m(k+1) - \hat{B}^*(k,z^{-1}) u(k-1) - \hat{R}(k,z^{-1}) y(k) ].
\] (5.1)
The nominal values of all of the discrete-time control parameters are designated in Chapter 4. However, some of the parameters involved in the estimation of $\hat{B}(k, z^{-1})$ and $\hat{R}(k, z^{-1})$ need to be set. Recall the constant trace version of the general parameter adaptation algorithm (PAA) outlined in Chapter 2:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + F(k)\Phi(k)e(k+1),$$  

(5.2)

$$F(k+1) = \frac{1}{\lambda_1(k)} \left[ F(k) - \frac{F(k)\Phi(k)\Phi(k)^TF(k)}{\alpha + \Phi(k)^TF(k)\Phi(k)} \right], \quad 0.5 \leq \alpha \leq 1,$$

(5.3)

and

$$e(k+1) = e^0(k+1) / (1 + \Phi(k)^TF(k)\Phi(k)),$$

(5.4)

where

$$e^0(k+1) = y(k+1) - y_m(k+1),$$

(5.5)

$$\hat{\Theta}(k)^T = [\hat{b}_0(k), \ldots, \hat{b}_m(k), -\hat{a}_1(k), \ldots, -\hat{a}_n(k)],$$

(5.6)

$$\Phi(k)^T = [u(k), \ldots, u(k-m), y(k), \ldots, y(k-n+1)],$$

(5.7)

and $\lambda_1(k)$ is computed at each step such that the trace of $F$ remains constant. In the present situation, $m$ (as in Eqn.(5.7)) represents the order of the numerator of the plant (or velocity loop), and $n$ represents the order of the denominator of the plant. From the transfer function of $A(z^{-1})/B(z^{-1})$ given by Eqn.(4.7), $m$ equals one and $n$ equals two. The definitions of $\hat{\Theta}(k)$ and $\Phi(k)$ indicate that both of the vectors have length $(n+m+1)$. For compatibility, $F(k)$, the adaptation gain matrix, must be an $(n+m+1)$ by $(n+m+1)$ square matrix. As a starting point, $F(0)$ is chosen to be
\[ F(0) = 100 \cdot \begin{bmatrix}
| \hat{b}_0(0) | & 0 \\
| \hat{b}_1(0) | & | \hat{a}_1(0) | \\
0 & | \hat{a}_2(0) | 
\end{bmatrix} \]  

(5.8)

where \( |*| \) denotes absolute value. Choosing the individual parameter adaptation gains proportional to their nominal magnitudes ensures they will converge at similar rates.

It is now possible to fully implement Landau's discrete MRAC algorithm outlined in Chapter 2. Recalling the desired reference model,

\[ \frac{D(z^{-1})}{C(z^{-1})} = \frac{d_0 z^{-1} + d_1 z^{-2} + d_2 z^{-3}}{1 + c_1 z^{-1} + c_2 z^{-2}} , \]  

(5.9)

and the adaptation response polynomial,

\[ G(z^{-1}) = 1 + g_1 z^{-1} + g_2 z^{-2} , \]  

(5.10)

let us define for notational simplicity the vectors

\[ \mathbf{d} = [ d_0, d_1, d_2 ] , \]  

(5.11)

\[ \mathbf{c} = [ c_1, c_2 ] \quad \text{and} \]  

(5.12)

\[ \mathbf{g} = [ 1, g_1, g_2 ] . \]  

(5.13)

Next, initialize \( \Theta(0) \) and \( \Phi(0) \):

\[ \Theta(0)^T = [ \hat{b}_0, \hat{b}_1, -\hat{a}_1, -\hat{a}_2 ] \quad \text{and} \]  

(5.14)
\[ \Phi(0)^T = \begin{bmatrix} u(-1), u(-2), y(-1), y(-2) \end{bmatrix} = \begin{bmatrix} 0, 0, 0, 0 \end{bmatrix}. \] (5.15)

The values for \( \hat{\Theta}(0) \) will be determined from available information on the velocity loop described in Chapter 3. Finally, calculate the value of the "constant trace":

\[ \text{constant trace} = \text{trace} ( F(0) ) = \sum ( |\hat{\Theta}(0)| ). \] (5.16)

From this point, the algorithm proceeds recursively for \( n \) iterations as follows:

For \( k = 1 \) to \( n \), the measurement vector is defined as

\[ \Phi(k)^T = \begin{bmatrix} u(k), u(k-1), y(k), y(k-1) \end{bmatrix}. \] (5.17)

The plant (velocity loop) output and filtered plant output are computed as

\[ y(k) = \begin{bmatrix} b_0, b_1, -a_1, -a_2 \end{bmatrix} \Phi(k-1) \] (5.18)

and

\[ y_f(k) = g \begin{bmatrix} y(k), y(k-1), y(k-2) \end{bmatrix}^T. \] (5.19)

[Note: In actual implementation, the velocity output would be measured from the tachometer output.] Next, the reference output and filtered reference output are determined by

\[ y_m(k+1) = d \begin{bmatrix} u_m(k), u_m(k-1), u_m(k-2) \end{bmatrix}^T - c \begin{bmatrix} y_m(k), y_m(k-1) \end{bmatrix}^T, \] (5.20)

and
\[ y_{\text{mf}}(k+1) = g \left[ y_{\text{m}}(k+1), y_{\text{m}}(k), y_{\text{m}}(k-1) \right]^T. \]  

(5.21)

The \textit{a priori} error is found by subtracting the filtered actual plant output from the filtered reference model output,

\[ e^0(k) = y_t(k) - y_{\text{mf}}(k). \]  

(5.22)

Now, the parameter adaptation is performed. The adaptation gain matrix, \( F(k-1) \) is updated as

\[ F(k-1) = \left[ F(k-2) - \frac{F(k-2)\Phi(k-2)\Phi(k-2)^TF(k-2)}{0.5 \leq \alpha \leq 1} \right]. \]  

(5.23)

It is then scaled to maintain a constant trace:

\[ F(k-1) = \left( \frac{\text{constant trace}}{\text{sum( trace(}F(k-1)\text{))}} \right) \times F(k-1). \]  

(5.24)

The \textit{a posteriori} error is calculated as

\[ e(k) = e^0(k) \frac{1}{1 + \Phi(k-1)^TF(k-1)\Phi(k-1)} , \]  

(5.25)

and the parameter vector is updated:

\[ \hat{\Theta}(k) = \hat{\Theta}(k-1) + F(k-1)\Phi(k-1)e(k). \]  

(5.26)

Finally, the control input can be computed as
\[ u(k) = \frac{1}{b_0(k)} \left[ y_{m1}(k+1) - \hat{b}_1(k)u(k-1) - \hat{r}_0(k)y(k) - \hat{r}_1(k)y(k-1) \right] \]  

(5.27)

where

\[ \hat{r}_0(k) = g_1 - \hat{a}_1(k) \] and

\[ \hat{r}_1(k) = g_2 - \hat{a}_2(k). \]  

(5.28)

(5.29)

The Millstone velocity loop saturates for commanded velocities in excess of \( \pm 5.6 \) volts. Therefore, when the computed magnitude of \( u(k) \) exceeds 5.6 volts, the control input will be 5.6 times the sign of \( u(k) \). Figure 2.2 is repeated as Figure 5.1 for reference. Recall that the sampling rate of the adaptive controller is 50Hz and therefore iterations are performed every 20ms.

### 5.2: Simulation Experiments

#### 5.2.1: The General Case

The top diagram in Figure 5.2 shows the outputs of the adaptive controller for a square wave input of unit magnitude. Here \( \alpha \) has been set equal to one and the initial parameter estimate vector, \( \hat{\Theta}(0) \), is started 25% off the correct value. This is 25% off in a "worst case" sense: in continuous-time, the real parts of the poles of \( A(s)/B(s) \) are moved 25% closer to the \( j\omega \)-axis and the imaginary parts of the poles are moved 25% further away from the real axis. Keep in mind that since the actual plant is being simulated, the correct value for the estimates is known.

Starting at \( t = 5 \) seconds (\( k = 250 \)), the damping ratio of the actual plant, \( A(z^{-1})/B(z^{-1}) \), is varied. The term damping ratio is used loosely here because \( A(s)/B(s) \) is not purely second order (recall, there is a zero at \( s = -5.79\text{rad/sec} \)). The damping ratio is
Figure 5.1: Adaptive Case
Figure 5.2: Outputs for Adaptive and Fixed Compensation
ramped from 0.707 to 0.33 over a period of 4 seconds. The natural frequency, \( \omega_n = 4.2781 \), is held constant. The dominant pole pair of \( A(s)/B(s) \) moves from its initial position of \( s_1^i, s_2^i = -3.03 \pm 3.02i \) at \( t = 5 \) seconds to its final position of \( s_1^f, s_2^f = -1.41 \pm 4.04i \) at \( t = 9 \) seconds. In discrete-time, the final description of \( A(z^{-1})/B(z^{-1}) \) is found using the Matlab "c2d" function as before. The difference between the discrete-time coefficients,

\[
[1, a_1^i, a_2^i] - [1, a_1^f, a_2^f] \quad \text{and} \quad [b_0^i, b_1^i] - [b_0^f, b_1^f] \quad (5.28)
\]
determines the slope of the ramp variation in each of the coefficients over the specified time interval. Here, the specified time interval is from \( t = 5 \) to \( 9 \) seconds or from \( k = 250 \) to \( 450 \) iterations.

Notice from the top diagram in Figure 5.2 that the variation in damping ratio in the adaptive controller is hardly detectable. Although both \( y \), the actual plant output, and \( y_m \), the desired plant output, are plotted in the top diagram, the performance error, \( (y - y_m) \), is so small that the two plots are indistinguishable. The bottom diagram of Figure 5.2 indicates the output of the nominal closed loop velocity loop, \( A(z^{-1})/B(z^{-1}) \), subjected to the same variation in damping ratio without any adaptation. Obviously, for any closed loop second order system with fixed compensation, a variation in the damping ratio from 0.707 to 0.33 will cause a large variation in the response. Thus, for this particular situation, the adaptive controller certainly has superior performance.

Figure 5.3 shows the performance error, \( (y - y_m) \), and the control input, \( u \). Figure 5.4 shows the procession of the parameter estimates, \( \hat{\Theta} \). The final values of the parameter estimates is not crucial; what is important is that they do not show any signs of diverging. In discrete-time, the misalignment of 25% may appear deceivingly small; it must be remembered that small deviations in the difference equation coefficients can have a large
Figure 5.4: Parameter Estimates for Square Wave Command
effect since the entire left-half s-plane is mapped to the interior of the unit circle in the z-plane.

Let us examine the effect of the value of $\alpha$ on the adaptive controller. The same simulation outlined above where the reference input was a square wave is performed, this time with $\alpha$ equal to 0.5 instead of one. Figure 5.5 shows the performance error in both cases. It is evident that the larger the value of $\alpha$, the faster the controller adapts. If the control input is driving the plant too hard, $\alpha$ can be reduced to decrease the drive. In this particular example, the difference between the control inputs for $\alpha = 0.5$ and $\alpha = 1.0$ is small.

Next, the effect of the polynomial $G(z^{-1})$ on the controller will be examined. Remember, $G(z^{-1})$ affects the closed loop adaptation response. In the original simulation discussed earlier, $G(s)$ had a double root at 60rad/sec. Figure 5.6 shows what happens to the performance error when $G(s)$ has a double root at 30rad/sec and then at 120rad/sec. At 30rad/sec, the error is less "damped" and takes longer to settle. At 120 rad/sec, the error is more "damped" and settles more quickly. The choice of a double root at 60rad/sec for $G(s)$ seems to be appropriate for this system.

How does the system react to a sinusoidal input? As before, the parameter estimate vector, $\hat{\Theta}(0)$, is initialized at 25% off its correct value and $\alpha$ equals one. The input is $\sin(2\pi t) + 3\sin(20\pi t)$, a sum of sine waves at 1Hz and 10Hz. Starting at $t = 10$ seconds ($k = 500$), the damping ratio of the actual plant is varied from 0.707 to 0.05 over a period of 4 seconds (200 iterations). Figure 5.7 shows the procession of the performance error, $(y - y_m)$, and the parameter estimates. Again, Landau's algorithm does not appear to have a problem adapting to a radical change in damping ratio.

It is also interesting to observe the response of the controller when the initial parameter vector estimate, $\hat{\Theta}(0)$, is started 50% off its correct value. The damping ratio of the actual plant is varied from 0.707 to 0.33 from 15 to 19 seconds. As a reminder, $\alpha$ is
Figure 5.5: Error with alpha = 1.0 and alpha = 0.5
Figure 5.6: Effect of G(1/z) on Error
equal to one and F(0), the adaptation gain matrix, is a diagonal matrix with $\Theta(0)$ along its diagonal. The reference input is chosen to be $\sin(0.2\pi t)$, a 0.1Hz sine wave.

The top diagram of Figure 5.8 shows the performance error, $(y - y_m)$, for the adaptive controller. The bottom figure shows the difference, $(y_n - y_m)$, where $y_n$ is generated by the nominal closed loop velocity loop with its fixed compensation. Because of their large magnitude differences, the errors are plotted on different vertical scales. [Recall, $y_m$ is generated by the desired closed loop model: a purely second-order response with a bandwidth of 1.5Hz and a damping ratio of 0.707.] Notice the small 0.1Hz sinusoidal modulation on the adaptive controller error. The amplitude of this modulation decreases with time and the error is unbiased. At $t = 40$ seconds, the amplitude is on the order of $1 \times 10^{-5}$. Hence, as predicted, the adaptation error goes to zero.

5.2.2: Unmodeled Dynamics

To examine the potential problems associated with unmodeled dynamics, it is natural to consider the Millstone velocity loop dynamics that were left out of the reduced-order model. The most dominant unmodeled pole was actually ignored earlier. Figure 3.5, which detailed the velocity loop, indicated a first order low-pass filter between the tachometer output and the A/D converter. The purpose of this filter is for anti-aliasing. The sampling rate of the digital controller is 50Hz and at the folding frequency, 25Hz, the response of the velocity loop is highly attenuated. Therefore, this simple first-order anti-aliasing filter is more than adequate. The anti-aliasing filter pole is at $s = -53.5\text{rad/sec}$ or about 8.5Hz. This pole is about a decade beyond the 0.89Hz -3dB frequency of the nominal actual velocity loop.
Figure 5.8: Errors for Adaptive and Fixed Compensation
In order to determine whether or not this pole should be modeled, two simulations will be performed: one with the pole included in the actual plant description, \( A(z^{-1})/B(z^{-1}) \), and the estimated plant description \( \hat{A}(z^{-1})/\hat{B}(z^{-1}) \), and another with the pole included only in the actual plant description. In the latter case, the output, \( \hat{y}(k) \), of the second-order estimator of the velocity loop will have to match the third-order velocity loop model actual output, \( y(k) \). Figures 5.9 and 5.10 show the error and output with and without the anti-aliasing pole included in the estimation model. The model reference adaptive controller appears to work much better when the anti-aliasing pole is included in the estimation model.

Similar experiments are performed with the rest of the known unmodeled dynamics, the closest being the zero at \( s = -102 \text{ rad/sec} \) or about 16.3Hz. Neither this unmodeled zero nor the unmodeled pole pair at \( s = -125 \pm 119i \text{ rad/sec} \) (19.9 ± 18.9i Hz) has a significant effect on the performance of the adaptive controller. Therefore, only the anti-aliasing pole will be added to the reduced-order model. The new continuous-time reduced-order velocity loop model is

\[
\frac{A(s)}{B(s)} = \frac{1.691 (s + 5.79)}{(s + 3.03 + 3.02i)(s + 3.02 - 3.02i)(s + 53.5)}. \tag{5.31}
\]

Using the Matlab "c2d" function, the new discrete velocity loop transfer function is

\[
\frac{A(z^{-1})}{B(z^{-1})} = \frac{b_0 z^{-1} + b_1 z^{-2} + b_2 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}
= \frac{2.4339 \times 10^{-2} z^{-1} - 4.617 \times 10^{-3} z^{-2} - 1.5195 \times 10^{-2} z^{-3}}{1 - 2.2221 z^{-1} + 1.5305 z^{-2} - 0.30391 z^{-3}}. \tag{5.32}
\]
Figure 5.9: Error with and without Anti-aliasing Pole

anti-aliasing pole unmodeled

anti-aliasing pole modeled
Figure 5.10: Output with and without Anti-aliasing Pole
The parameter vector, \( \hat{\Theta}(0) \), becomes

\[
\hat{\Theta}(0)^T = [ \hat{b}_0, \hat{b}_1, \hat{b}_2, -\hat{a}_1, -\hat{a}_2, -\hat{a}_3 ] .
\]  \hspace{1cm} (5.33)

G\( (z^{-1}) \) will remain unchanged but a trailing zero will be added for compatibility,

\[
g = [ 1, g_1, g_2, 0 ] .
\]  \hspace{1cm} (5.34)

In addition, F, the adaptation gain matrix is now a 6 x 6 square matrix.

What happens when the unmodeled dynamics are unknown? To determine this, unmodeled dynamics are introduced into the actual plant model, \( A(z^{-1})/B(z^{-1}) \), while the plant estimator model, \( \hat{A}(z^{-1})/\hat{B}(z^{-1}) \), remains unchanged. Figure 5.11 shows the error when an unmodeled pole pair with a damping ratio of 0.05 and a real part of 5Hz is included in the actual plant model. Now, the third-order estimator output must match the fifth-order actual plant output. The reference input in the top diagram is a 1Hz sine wave of amplitude 2.5. In the bottom diagram, the reference input is a 20 second period square wave. In both cases, \( \hat{\Theta}(0) \) is started 50% off its correct value. From these results, the performance of the algorithm does not appear to be adversely affected by the presence of the unmodeled pole pair. Similar experiments are performed with a variety of reference inputs for unmodeled pole pairs at 7.5Hz and 10Hz. The results are similar.

These observations are quite unexpected, given the warnings of studies by Rohrs et al in [1,18]. In his thorough 400-page doctoral thesis, Adaptive Control in the Presence of Unmodeled Dynamics, (MIT '82), Rohrs concluded that "none of the adaptive control algorithms considered can be used with confidence in most practical control system designs because instability will set in with a high probability." Rohrs did, however, concentrate on the IM class of MRAC algorithms because their instability (and stability) mechanisms are
Figure 5.11: Errors with Unmodeled Pole Pair at 5Hz
better understood. Given that the instability mechanisms for Landau's algorithm are less
well understood, it is certainly necessary for the user to always test his or her particular
system for robustness in the presence of unmodeled dynamics. Rohr's suggestions for
other potentially troublesome situations are heeded and the results are shown next.

5.2.3: Unpredictable Disturbances

Depending on the disturbance rejection properties of the plant, disturbances can
propagate through to the output. Sensor noise and 60Hz hum commonly present
themselves as output disturbances. Using the "new" third-order reduced-order model
derived in Eqns.(5.31) and (5.32), Landau's algorithm is simulated with disturbances
injected directly onto the output. There are no unmodeled dynamics and the initial
parameter estimate vector, \( \hat{\Theta}(0) \), is started 25% off its correct value. The reference input is
a 20 second period square wave of unit amplitude.

Figure 5.12 shows the performance error and control input when a DC offset of
0.01 is injected onto the actual plant output, \( y(k) \). For a DC offset level of 0.01, the
control input compensates by a DC bias (relative to the reference input) of over -2 volts.
At this rate, the control input will saturate for a DC offset level on the output of about 0.03.
If the control input is continually required to exceed the saturation voltage but is not
allowed to, the algorithm will not be able to bring the error down. In fact, when a DC
offset of 0.05 is added to \( y(k) \), the error quickly reaches a DC value of about 10.5 with a
non-decreasing modulation equal to the reference input. The control input, as one might
expect, rails continuously between \( \pm 5.6 \) volts (the input saturation voltage). The same is
ture for a sinusoidal disturbance injected directly onto the output. The controller is unstable
for sinusoidal disturbance amplitudes in excess of about 0.1 volts. Clearly, Landau's
Figure 5.12: Error and Control Input with DC Offset on Output
algorithm is intolerant of unmodeled output disturbances of any significant amplitude. It so happens that the output of the real Millstone velocity loop is very clean; however, in general, this aspect must be given serious consideration by potential users of Landau's algorithm.

According to Rohrs, the IM class algorithms do not fare much better: "any disturbance whose spectrum has frequency peaking will produce instability" [1]. These results certainly suggest that it is preferable to use adaptive control around a nominally stable closed loop plant as opposed to an open loop plant since the lack of disturbance rejection in the latter case will likely cause problems. The result of this section leads directly to the next consideration.

5.2.4: Large Magnitude Inputs

In Section 5.2.3, it was shown how the adaptive controller goes unstable if the control input, u(k), continually saturates. Clearly, this has serious consequences for large magnitude (relative to the saturation voltage) inputs of long duration. The allowable magnitude and duration is obviously dependent on the type of excitation. For example, if the parameter estimates are started 25% off their correct values and the reference input is a 4 volt amplitude sine wave of frequency 1Hz, the controller is stable. The performance error is between +0.08 and -0.06. However, if a 4.5 volt sine wave of the same frequency is input, the controller is unstable. The control input, u(k) rails continually between ± 5.6 volts. For a square wave input, the maximum tolerable input amplitude is around 4 volts. Given this prospect, the potential user of model reference adaptive control must be prepared to run extensive simulations (hopefully, with a priori information on what types of inputs can be expected) and to construct various "safety nets" in case of the unexpected. An
example of a safety net would be to turn off the adaptation whenever the control input magnitude is equal to the saturation voltage for longer than some pre-specified time period.

K. J. Aström [19] and S. Boyd and S. S. Sastry [20] have all suggested that the problems of unmodeled dynamics and unpredictable disturbances can always be overcome by sufficient excitation of the control input. [Suggestions have been made to monitor the control input richness at all times and to inject perturbations or turn the adaptation off when needed.] Surprisingly, though, the subject of control input saturation has largely been ignored by almost all of the literature on MRAC. Clearly all real systems have some kind of limitation on the control input. Given that one makes certain practicality-based assumptions on what is likely to be a realistic simulation scenario, saturation of the control input is the only obvious mechanism to cause Landau's algorithm to go unstable. However, given that it is a non-linear, time-varying system, one can only make heuristic arguments concerning the underlying instability mechanisms.
CHAPTER 6

CONCLUSIONS

Given the simulation results of Chapter 5, there are some obvious questions that come to mind concerning the practical application of Landau's MRAC algorithm and MRAC in general:

What conditions must be satisfied in order to use Landau's algorithm?

1. The plant output should be reasonably clean. Simulations should be run to determine what level of disturbance can be tolerated.

2. The control input must have a larger magnitude swing than the reference input. Simulations can determine exactly how much more swing the control input needs.

As shown in Section 5.2.2, Landau's algorithm in the examined context can tolerate unmodeled dynamics reasonably well. Also, sufficiency of excitation does not appear to be a problem.

What are the advantages of adaptive control?

Assuming that one can reasonably estimate the number of significant (see Section 5.2.2) poles and zeros:
1. A pre-specified closed loop performance criterion can be maintained in the presence of time-varying changes in the plant dynamics. These changes can be extreme.

2. In the case of a time-invariant but poorly characterized plant, the adaptive controller will self-tune the compensation given the desired closed loop response.

When should adaptive control be used?

The answer to this question depends on a number of factors. Certainly, if fixed compensation will produce a closed loop response that is within desired specifications, an adaptive controller which is computationally intensive and requires "safety nets" and extensive simulation is not justified. Although the closed loop response for a fixed compensator may be time-varying, it is often the case that the settling time and/or transient accuracy is not so strictly specified that this variation in response matters. It is believed that the Millstone antenna falls into this category. The dependency of the closed loop response on elevation angle only produces a small variation in response. There is no closed loop velocity loop data categorized by weather conditions available. Except in extreme conditions where the motors would be driven too hard regardless of the compensation approach, there does not appear to be significant performance degradation due to weather fluctuations.

What kind of situations may warrant adaptive control?

1. The plant has small time variations that lead to violation of a strictly specified closed loop response.
2. The time variations of the plant are so large that even a loose closed loop response specification cannot be maintained.

Even in these cases, adaptive control must be weighted against adjustable "classical" controllers which may use estimators and/or additional measurements.

Clearly, adaptive control will not be fully accepted into mainstream control applications until further advancements are made. Although the hypotheses of the early 1980's (such as those proposed by Rohrs et al [1,18]) led many researchers away from adaptive control, there is still considerable research interest in the area. For the present time, users of adaptive control must be acutely aware of the potential hazards.
REFERENCES


Appendix A: Circuit Diagrams

Appendix A.1 Velocity Loop Circuitry Diagrams

Appendix A.2 A/D Buffer and Meter Circuitry Diagram

Appendix A.3 Tachometer Lead Network Analysis
Figure A.1a: Velocity Loop Circuitry
Figure A.1b: Velocity Loop Circuitry
Figure A.2: A/D Buffer and Meter Circuitry

Potentiometers are set at nominal current values.
Appendix A.3: Tachometer Lead Network Analysis

\[ R_p = R \parallel R_1 \]

Using superposition, ground \( v_2 \):

\[ v_{01} = \frac{R_p(1/sC)}{R_p(1/sC) + R(1/sC)} v_1 \]

\[ v_{01} = \frac{R_p/(sCR_p + 1)}{R/(sCR + 1) + R_p/(sCR_p + 1)} v_1 \]

\[ v_{01} = \frac{R_p(sCR + 1)}{R(sCR_p + 1) + R_p(sCR + 1)} v_1 \]

\[ v_{01} = \frac{R_p(sCR + 1)}{(R + R_p)[2CRR_p/(R + R_p)s + 1]} v_1 \]

By symmetry, grounding \( v_1 \):

\[ v_{02} = \frac{R_p(sCR + 1)}{(R + R_p)[2CRR_p/(R + R_p)s + 1]} v_2 \]

Hence:

\[ v_0 = v_{01} + v_{02} \]

* This analysis is provided because of discrepancy with existing documents, namely [15].
Appendix B: Velocity Loop Transfer Function in Terms of Actual Parameters

Note: Because of the complexity of the transfer function, the numerator and denominator will be given separately and the denominator will be expressed as a sum of polynomials. The parameter names used here are the same as in Figures 3.5 and 3.6a.

Numerator  =  57.3 \ K_1 \ K_t \ N \ (196.1) \ \{ \ \tau_4 \tau_2 \ s^2 + (\tau_4 + \tau_2) \ s + 1 \ \}

Den1 = \{ \ \tau_4 \tau_2 \tau_g \ s^3 + [ \ \tau_4 \tau_2 + (\tau_4 + \tau_2)\tau_g \ ] \ s^2 + (\tau_4 + \tau_2 + \tau_g) \ s + 1 \ \} *

\{ \ J \ \tau_a \ s^2 + (J + \mu \tau_a) \ s + (\mu + K_1K_vn^2) \}

Den2 = 6.2 \ K_1 \ K_{cs} \ \{ \ \tau_3 \tau_2 \ J \ s^4 + [ \ \tau_3 \tau_2 \mu + (\tau_3 + \tau_2)J \ ] \ s^3 + [ \ (\tau_3 + \tau_2)\mu + J \ ] \ s^2 + \mu \ s \}

Den3 = K_2K_1 \ K_t \ N \ \{ \ \tau_1 \tau_4 \ s^2 + (\tau_1 + \tau_4) \ s + 1 \ \}

Denominator = Den1 + Den2 + Den3
APPENDIX C: Adjustments to Millstone Model

%CONSTANT DECLARATIONS

\(\tau_1 = 1.1 \times 0.15;\)  % Tachometer Lead Network zero
\(\tau_2 = 0.9 \times 0.01085;\)  % Tachometer Lead Network pole
\(\tau_3 = 0.9 \times 0.00705;\)  % Current Feedback filter zero
\(\tau_4 = 1.1 \times 0.157;\)  % Current Feedback filter pole
\(k_{ta} = 1.1 \times 0.95;\)  % Tachometer gain
\(n = 1.1 \times 2230;\)  % Gear Ratio
\(kp_{ot} = 1.1 \times 0.713;\)  % Tach Feedback Potentiometer setting
\(k_2 = 10 \times k_{ta} \times n \times kp_{ot} \times 19.61 \times 0.0362;\)
\(k_v = 0.9 \times 1.2;\)  % Back EMF of motors
\(\tau_{ua} = 1.1 \times 0.0015;\)  % Motor pole
\(k_1 = 0.85 \times 64.789;\)  % Power Amp gain * Motor Generator % gain divided by Motor Generator resistance
\(k_{cs} = 0.9 \times 0.05;\)  % Current Sensor gain
\(\tau_{aug} = 0.9 \times 0.026;\)  % Motor Generator pole
\(m_1 = 1.15 \times 265.5;\)  % Mount Friction
\(j = 0.9 \times 4.15e6;\)  % Mount Inertia
\(k_t = 1.15 \times 0.9;\)  % Torque Sensitivity

%TRANSFER FUNCTION: OUTPUT VELOCITY/INPUT VELOCITY

% All vectors represent powers in 's' going from right to left with the rightmost element representing a constant term (s^0).

%NUMERATOR

num = [\(\tau_4 \times \tau_2, (\tau_4 + \tau_2), 1;\)]
% 57.3 factor is for rad/sec to deg/sec conversion.
% 1.2742 factor is potentiometer "play" to bring DC % gain to unity.
numerator = 57.3 \times 1.2742 \times k_1 \times k_t \times n \times 196.1 \times num;

%DENOMINATOR

den1a = [\(\tau_4 \times \tau_2 \times \tau_{aug}, (\tau_4 \times \tau_2 + (\tau_4 + \tau_2) \times \tau_{aug}), \ldots\)
\((\tau_4 + \tau_2 + \tau_{aug}), 1;\)]
den1b = [\(j \times \tau_{aug}, j + m_1 \times \tau_{aug}, m_1 + k_t \times k_v \times n^2;\)]
"Conv" function multiplies two polynomials in 's'.
den1 = conv(den1a, den1b);
den2 = [\(\tau_3 \times \tau_2 \times j, (\tau_3 \times \tau_2 \times m_1 + (\tau_3 + \tau_2) \times j), \ldots\)
\((\tau_3 + \tau_2) \times m_1 + j), m_1, 0];
den2 = 6.2 \times k_1 \times k_{cs} \times den2;
den3 = [\(\tau_1 \times \tau_4, (\tau_1 + \tau_4), 1;\)]
den3 = k_2 \times k_1 \times k_t \times n \times den3;
denominator = den1 + den2 + den3;