Adaptive Control of a Small Autonomous Underwater Vehicle

by

Bernd Clauberg

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 and Computer Science

at the

Massachusetts Institute of Technology
July, 1991

Signature of Author ________________________
Department of Electrical Engineering
and Computer Science June, 1991

Certified by ________________________
Dr. Jay Farrell
Technical Supervisor
The Charles Stark Draper Laboratory

Certified by ________________________
Dr. Munther A. Dahleh
Thesis Supervisor
Associate Professor
~~Department of Electrical Engineering~~ and Computer Science

Accepted by ________________________
Professor Campbell L. Searle
Chairman, Departmental Graduate Committee
Department of Electrical Engineering and Computer Science

Adaptive Control of a Small Autonomous Underwater Vehicle

by

Bernd Clauberg

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 and Computer Science

ABSTRACT

The main objective of this thesis was to develop an indirect adaptive depth and heading controller for the autonomous underwater vehicle (AUV) *Sea Squirt*. A comparison of the several estimation methods was done to determine which identification method worked best for the AUV. The identification methods that were researched included recursive least square (RLS), normalized least mean square (NLMS), output error minimization (OEM), and extended Kalman Filtering (KF). The extended Kalman Filter was chosen as the preferred method and used in all adaptive controllers implemented in this report.

The depth controllers implemented and tested using the KF estimated states and parameters included Pole-Placement, LQG, and Sliding Mode. All of these designs were implemented on-line on the AUV *Sea Squirt* and extensively compared both through off-line simulation and through in-water testing. The adaptive Pole-Placement design was chosen as the best controller for this application because it performed as well or even better than the more complex Sliding Mode and LQG designs.

Finally, a combined depth and heading controller was developed for the *Sea Squirt* using the Pole-Placement design. Several autonomous in-water tests were done to demonstrate the performance of the adaptive controller in a real-life environment.

Technical Supervisor: Dr. Jay Farrell
Thesis Supervisor: Dr. Munther A. Dahleh
Title: Associate Professor of Electrical
Engineering and Computer Science

ACKNOWLEDGMENT

First, I would like to thank Jay Farrell for all the time and effort he devoted to this thesis. His many comments and suggestions are greatly appreciated. It has been a pleasure to work with Jay and to have him as a friend. A special thanks also goes to my thesis advisor Munther Dahleh.

I'd also like to thank Bill Bonnice and Jim Bellingham for getting me involved in the MIT Sea Grant/CSDL *Sea Squirt* project. This project gave me a unique chance to address the implementation issues involved with on-line adaptive control. It also gave me the opportunity to see the result of my work on an actual vehicle and not just in simulation.

I want to thank all the members of the MIT Sea Grant College Program for their assistance and support especially Jim Bellingham and Tom Consi for the many hours they devoted to helping me during in-water testing with the *Sea Squirt*.

I would also like to thank all the members of the autonomous systems group for making my stay at Draper enjoyable. A special thanks goes to Krishna Govindarajan for his patients in helping me get started and to my office mate Mitch Livstone for being a good friend.

Finally, I'd like to thank Cedric Logan for helping me run the *Sea Squirt* and Doug Humphrey for writing and maintaining the vehicle software and for helping me with countless software problems.

I thank the Charles Stark Draper Laboratory for the financial support it provided under the Independent Research and Development Program.

Table of Contents

CHAPTER 1
INTRODUCTION 8
1.1 Control Design Problem 8
1.2 Thesis Organization 10
CHAPTER 2
VEHICLE DESCRIPTION AND DERIVATION OF DISCRETE TIME DYNAMICS 12
2.1 Vehicle Description 12
2.1.1 Physical Description of the Sea Squirt 12
2.1.2 Vehicle Hardware 14
2.1.3 Vehicle Software 18
2.2 Derivation of Simplified Dynamics for the Sea Squirt 19
CHAPTER 3
ADAPTIVE CONTROL 24
3.1 System Identification 25
3.1.1 Recursive Least Square 27
3.1.2 Normalized Least Mean Square 30
3.1.3 Output Error Minimization 32
3.1.4 Extended Kalman Filter 34
3.1.5 Summary of Identification Methods 37
3.2 Feedback Control 38
3.2.1 Pole-Placement (PP) 44
3.2.2 Linear Quadratic Gaussian (LQG) 51
3.2.3 Sliding Mode (SM) 53
CHAPTER 4
SIMULATION COMPARISON OF IDENTIFICATION METHODS AND CONTROL DESIGNS 65
4.1 Simulation Comparison of Identification Methods 65
4.2 Simulation Comparison of Control Designs 79
4.2.1 Comparison of Controllers After Parameter Convergence 81
4.2.2 Comparison of Controllers with Sensor Noise and Disturbances 84
4.2.3 Comparison of Controllers during Initial Parameter Estimation 88
4.2.4 Comparison of Controllers Based on Implementation Issues 92
4.2.5 Summary and Conclusion of Off-Line Control Design Comparison 93
CHAPTER 5
IMPLEMENTATION ISSUES 95

5.1 Transient Safety....................96
5.2 Preventing Parameter Drift....................98
5.3 Adding Excitation....................100
CHAPTER 6
ON-LINE CONTROL IMPLEMENTATION....................105
6.1 In-Water Depth Control Comparison....................105
6.1.1 Comparison of Controllers After Parameter Convergence....................106
6.1.2 Comparison of controllers during Initial Parameter Estimation....................109
6.1.3 Conclusions of Depth Control Comparison....................116
6.2 Adaptive Heading Control....................117
6.2.1 Heading Control Design....................117
6.2.2 In-Water Autonomous Depth and Heading Control of the Sea Squirt....................121
CHAPTER 7
CONCLUSIONS AND FUTURE APPLICATIONS....................137
7.1 Conclusions....................137
7.2 Future Improvements and Applications....................138
APPENDIX 1
DEFINITION OF VEHICLE RELATED VARIABLES....................140
APPENDIX 2
AUV EQUATIONS OF MOTION....................142
APPENDIX 3
DERIVATION OF DISCRETE TIME DEPTH DYNAMICS....................146
APPENDIX 4
ANALYTIC SOLUTION OF THE DISCRETE TIME RICCATI EQUATION....................154
REFERENCES....................158

List of Figures

2.1 CSDL/MIT Sea Grant AUV (*Sεa Squirt*)

2.2 Nonlinear Thruster Characteristics

2.3 Simplified Thruster Model

3.1 Block Diagram of the Indirect Adaptive Controller

3.2 Step Response of the Desired Closed-Loop System

3.3 Saturation Function

3.4 Discrete Time Sliding Surface with Boundary Layer

4.1 Comparison of Identification Methods for a Linear Plant

4.2 Comparison of Identification Methods for a Nonlinear Plant

4.3a Parameter Convergence Without Fast Buoyancy Id

4.3b Parameter Convergence with Fast Buoyancy Id

4.4 Comparison of Controllers After Parameter Convergence

4.5 Thrust Commands Corresponding to Figure 4.4

4.6a Performance of SM Controller After Parameter Convergence with Noise Added

4.6b Comparison of PP and LQG Controllers After Parameter Convergence with Noise Added

4.7 Thrust Commands Corresponding to Figures 4.6a and 4.6b

4.8 Depth Control for a 1 pound Sinusoidal Thrust Disturbance

4.9a Sliding Mode Controller Performance During Initial Parameter Convergence

4.9b PP and LQG Controller Performance During Initial Parameter Convergence

4.10a Parameter Estimates Corresponding to the SM Controller

4.10b Parameter Estimates Corresponding to the PP Controller

4.10c Parameter Estimates Corresponding to the LQG Controller

5.1 Safety Net for Adaptive Controller

6.1 In-Water Comparison of Controllers After Parameter Convergence

6.2 Thrust Commands Corresponding to Figure 6.1

6.3a PP Controller Performance During Initial Parameter Convergence

6.3b SM Controller Performance During Initial Parameter Convergence

6.3c LQG Controller Performance During Initial Parameter Convergence

6.4a Parameter Estimates Corresponding to the PP Controller

6.4b Parameter Estimates Corresponding to the SM Controller

6.4c Parameter Estimates Corresponding to the LQG Controller

6.5 Autonomous Pole-Placement Heading Control

6.6 Thrust Commands Corresponding to Figure 6.5

6.7a Heading parameters Corresponding to Figure 6.5

6.7b Heading parameters Corresponding to Figure 6.5

6.8 Depth Control Corresponding to Figure 6.5

6.9 Thrust Commands Corresponding to Figure 6.8

6.10a Parameter Estimates Corresponding to Figure 6.8

6.10b Parameter Estimates Corresponding to Figure 6.8

6.11 Depth Control With Parameter Change at 150 seconds

6.12a Parameter Estimates Corresponding to Figure 6.11

6.12b Parameter Estimates Corresponding to Figure 6.11

6.13 Heading Control Corresponding to Figure 6.11

6.14a Parameter Estimates Corresponding to Figure 6.13

6.14b Parameter Estimates Corresponding to Figure 6.13

A1.1 Representation of Euler Angles

A3.1 Taylor Series Coefficients of W(N+1)

CHAPTER 1

INTRODUCTION

In this thesis an indirect adaptive controller is developed for a small autonomous underwater vehicle (AUV), the *Sea Squirt.* This vehicle was designed in 1988 by the Charles Stark Draper Laboratory (CSDL) and the M.I.T. Sea Grant Program as a research aid for autonomous planning and control design. Although the control design was done specifically with this vehicle in mind, the conclusions are general enough to allow applications to other systems. The main objective of this research was to develop a controller that could achieve good performance and robustness without knowing the parameters (i.e. hydrodynamic coefficients) of the system a priori and without having to tune the gains of the controller by hand. The effectiveness of the control design is then demonstrated on the AUV *Sea Squirt.*

1.1 Control Design Problem

The precise control of autonomous underwater vehicles (AUV's) is difficult because it requires the knowledge of the vehicle's hydrodynamic model. The problem lies in the difficulty to obtain an accurate model without extensive tow tank experiments which are usually extremely costly and inaccurate if a scaled version of the vehicle has to be used. Also, the hydrodynamics of the vehicle can change during a mission. For example, the payload may change, increasing or decreasing the buoyancy and mass properties of the vehicle. Even the vehicle itself may change as new equipment such as sensors or additional batteries are installed in the AUV. To overcome these problems, a feedback controller could adapt in an on-line fashion so as to reduce the amount of plant uncertainty. The indirect adaptive controller uses an identification method that estimates the parameters of a dynamic model of the AUV during normal operation based on certain sensor readings (e.g., depth, heading) and the corresponding control inputs (e.g., thrust). The increased parameter accuracy gained by the identification method allows one to update the feedback gains of the controller, resulting in better performance than could have been achieved using a fixed controller based only on a priori knowledge of the system.

In this thesis several indirect adaptive controllers will be presented and compared in actual in-water tests. Four different identification methods and three different control design techniques will be discussed. The identification methods discussed include the most common techniques -- Recursive Least Square (RLS), Normalized Least Mean Square (NLMS), Output Error Minimization (OEM), and extended Kalman Filtering (KF). The control designs presented in this thesis are adaptive Pole-Placement (PP), adaptive Linear Quadratic Gaussian (LQG), and adaptive Sliding Mode (SM). All controllers were implemented in C on the testbed AUV *Sea Squirt* to evaluate and demonstrate their performance in a real-world environment.

1.2 Thesis Organization

Chapter 2 describes the details of the testbed underwater vehicle. All relevant hardware and software components are listed and described to give the reader an overview of the vehicle. The six-degree-of-freedom nonlinear equations of motion are presented for the testbed vehicle. These equations are then simplified by decoupling the depth and heading dynamics and reducing them to two second order independent systems. All assumptions used in this simplification will be discussed.

Chapter 3 motivates and describes the indirect adaptive controller. Four identification methods (RLS, NLMS, OEM, KF) are presented and their advantages and disadvantages are discussed. Three control methods (PP, LQG, SM) are discussed and compared in a similar fashion.

Chapter 4 compares the identification methods and controllers using off-line simulations. The estimation techniques are mainly compared based on how well they can identify the parameters of a known plant as a function of signal-to-noise ratio. Computational requirements of the algorithms are also considered as they must allow on-line implementation. The control design methods are compared based on their performance in a six-degree-of-freedom nonlinear model of the *Sea Squirt* dynamics. Since these methods are implemented in an indirect adaptive control setting, their performance is evaluated both before and after parameter convergence.

Chapter 5 discusses implementation issues such as transient safety (i.e. safe operation during parameter estimation transients) and sufficient excitation (i.e. signal being sufficient for correct identification of the plant parameters). Transient safety will be achieved by using a safety net approach which guarantees stability by assuring that the vehicle path remains within some distance of the desired trajectory even when the parameter estimates are poor. Since sufficient excitation cannot be guaranteed without the freedom to design the commanded depth and heading trajectories, conditions will be developed that allow either the estimation to automatically stop or to automatically add excitation when necessary. These conditions will prevent parameter drift and permit adaptation whenever it is needed.

Chapter 6 compares the depth controllers in actual in-water tests demonstrating the performance of each design both before and after the parameter estimates have converged. The "best" design is then chosen for the combined heading and depth control of the vehicle. The results of a full-scale autonomous test are then presented to demonstrate the effectiveness of the adaptive controller in a real-world environment.

Chapter 7 presents the conclusions of this research. A brief discussion of the future application and possible improvements of the indirect adaptive controller will be given.

CHAPTER 2

VEHICLE DESCRIPTION AND DERIVATION OF DISCRETE TIME DYNAMICS

Before discussing the equations of motion of the *Sea Squirt*, it is necessary to give a description of the vehicle including its physical shape, its hardware and software, as well as its basic operation. Simplified discrete time depth and heading dynamics are then derived from the general six-degree-of-freedom nonlinear model for underwater vehicles [1].

2.1 Vehicle Description

2.1.1 Physical Description of the *Sea Squirt*

The *Sea Squirt* is a battery operated, three feet long, 80 pound, autonomous submarine which has been developed by CSDL and MIT Sea Grant as a research and testbed vehicle (Fig. 2.1). As shown in Figure 2.1, the vehicle has one vertical thruster, one port (left) thruster, and one starboard (right) thruster. The vertical thruster is used in depth control while the other two thrusters are used in heading and speed control. There are no control surfaces such as stern planes, bow planes or rudders. The vertical thruster applies a force on the AUV which pushes it straight up or down depending on the direction of the applied force. The vertical thruster passes through the hull near the center of gravity (c.g.) and center of buoyancy (c.b.) so that induced pitch and roll angles remain small for non-zero vertical thrusts. Also, the x, y components of the center of drag (c.d.) are close to the center of the vertical thruster so that even for large vertical velocities (1 ft/s), the pitch and roll angles remain small.

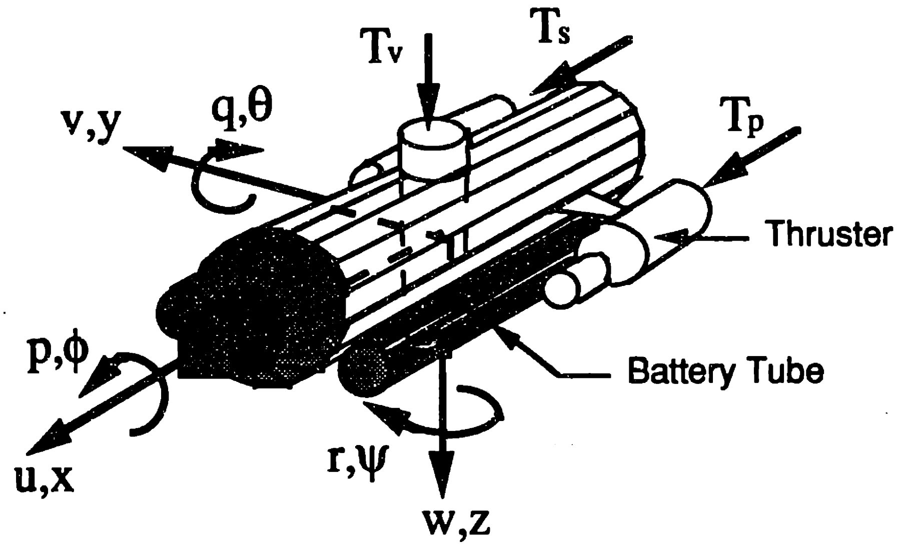


Figure 2.1 : CSDL/MIT Sea Grant AUV (*Sea Squirt*)[1]

The z-component of the c.d. is below the forward thrusters, due to the additional drag introduced by the external battery tubes, causing the vehicle to pitch downward for non-zero forward velocities or up for non-zero backward velocities. However, for slow forward speeds the pitch and roll angles are still small. As there are no actuators to control the pitch and roll of the vehicle, they are passively stabilized by placing the center of gravity (c.g.) below the

[1] All variables shown in Figure 2.1 are formally defined in Appendix 1.

center of buoyancy (c.b.). In general, the buoyancy of the AUV will always be slightly greater than its weight for safety reasons (this ensures that the vehicle will surface if no thrust is applied). The port and starboard thrusters are located along the sides of the vehicle (see Figure 2.1). This arrangement allows one to use the port and starboard thrusters to control both the vehicle heading and velocity independently. The hardware and software of the vehicle will be discussed in Sections 2.1.2 and 2.1.3.

2.1.2 Vehicle Hardware

The on-board computer is made up of a GESPAC single board 68020 microprocessor with math coprocessor, a 10-bit A/D and D/A board, and a communications board. Currently, there is no permanent storage available on the vehicle so that all data and programs have to be stored in volatile memory (which is 512 Kbytes). This means that the parameter estimates learned by the adaptive controller would be lost if the on-board computer lost power. A 40 meg hard disk will be added in the future to extend the data storage capability of the AUV and to enable permanent storage of data, in the case of the adaptive controller parameter estimates. As a comparison, the computing power of the on-board computer is approximately equal to that of a Mac II.

The sensors available on the AUV are summarized in Table 2.1. Our experiments will use only the depth and heading sensors. The yaw rate sensor was not used due to the fact that it drifts (i.e.

produces steady-state rate errors) with time. Although this problem could have been eliminated, it turned out that the rate measurement was not necessary for the control of the vehicle. Since the depth range for the depth sensor is only 0 to 60 feet (while the vehicle itself has a depth range of several hundred feet), a different depth sensor has to be used for missions requiring depth ranges over 60 feet. The advantage of using a depth sensor with a relatively small range is that the accuracy of the measurement is improved. This is mainly due to the fact that the entire depth range is quantized by the A/D converter into 1023 quantization steps. Clearly, if the range of the sensor is large, accuracy will be lost since the resulting quantization steps will also be large.

Table 2.1: Vehicle Sensors

Measured Variable	**Sensor Type**
depth	electronic pressure gauge (range is 0 to 60 feet depth)
heading	electronic flux gate compass
pitch	electronic capacitance type
roll	electronic capacitance type
yaw rate	gyro (measures turn rate)
speed	paddle wheel type (not currently in use)
temperature	electronic thermo couple

The power for the vehicle in autonomous operation is supplied by either silver zinc or D-size batteries. Each of the two large external battery containers of the vehicle can hold 21 D-cells (at 1.5 volts per cell) or 24 rechargeable silver zinc cells (at 1.85 volts per cell). The 21 Duracells are connected in series for a total of 32 volts with a rated maximum current output of about 1 amp. The 24 silver zincs are connected as two batteries (12 cells each) in parallel for a total of 22 volts with a maximum rated current of about 30 amps. One set of batteries (i.e. one container) powers the vehicle electronics including everything except the thrusters. The other set supplies the power to operate the thrusters. The thrusters need a considerable amount of current to operate at full power (up to 8 amps each) and, therefore, usually have to be powered by silver zinc batteries since these can supply much more current than the D-cells. The electronics can be powered by either silver zincs, D-cells, or, if a tether is attached, by a power supply.

The thrusters are driven by a pulse width modulator that applies a high frequency square wave of varying duty-ratio and constant amplitude to the thruster motors. The amplitude is determined by the battery voltage, V, (which may change as the batteries drain) and the duty-ratio is adjusted by the modulator according to the thrust commanded by the controller. For example, if the controller commands a thrust of 1 pound, this thrust is changed to a percentage of maximum thrust (which is 6.5 pounds for the thrusters on the *Sea Squirt*) and then the modulator applies a square wave voltage to the motors which varies between +V volts (positive V for positive thrust, negative V for negative thrust) and zero volts and has a duty-ratio of 1/6.5. The modulating period is much lower than the response time of the thrusters so that the total thrust output remains constant for a given thrust command.

One problem with using the thruster controllers described above is that there is no guarantee that the thrust commanded by the controller is actually achieved since no feedback is used. If a torque or propeller speed sensor was available, feedback could be used to control the thrust more accurately. Also, the thrusters have nonlinear dynamics and dead zones (see Figure 2.2). The dead zones are partially inverted by experimentally determining the approximate location of the forward and reverse set points indicated in Figure 2.2. The nonlinearities, other than the dead zones, shown in Figure 2.2 are ignored by the thruster controller since they can not be easily measured or calculated. The thruster characteristics assumed by the controller are shown in Figure 2.3. [2]

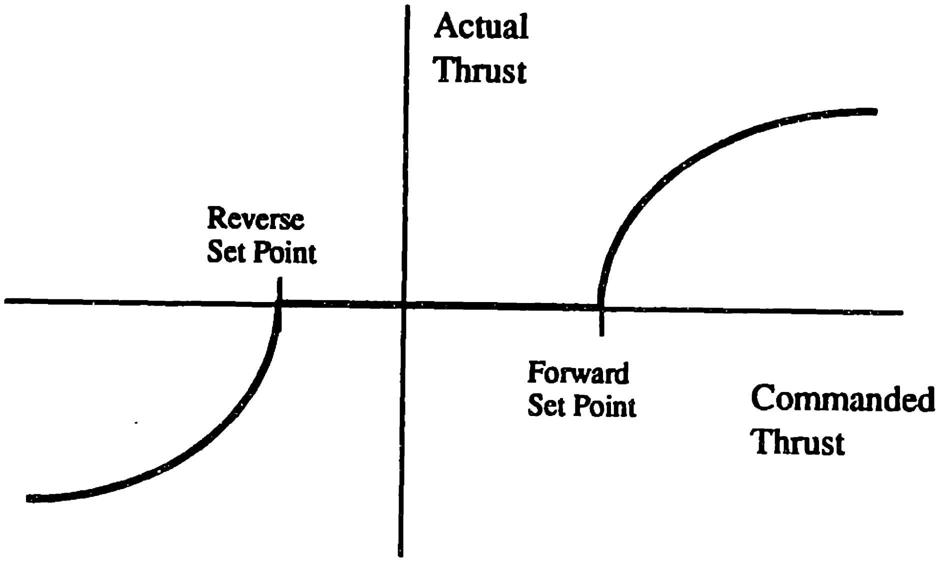


Figure 2.2: Nonlinear Thruster Characteristics

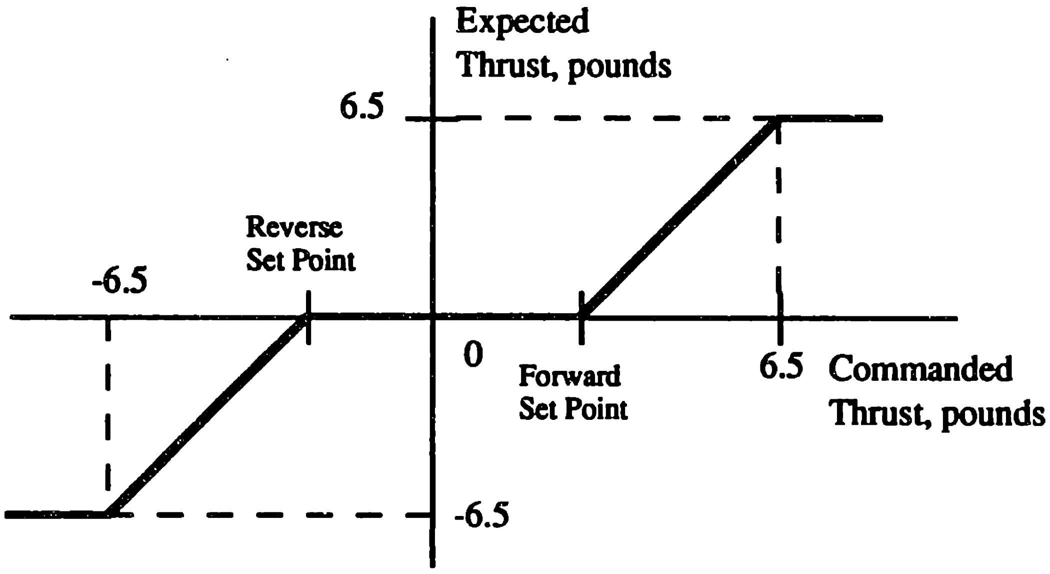


Figure 2.3: Simplified Thruster Model

2.1.3 Vehicle Software

For implementation purposes all software programs (i.e. controllers and identification algorithms) will be implemented in C and downloaded from a portable computer (zenith 286) to the on-board computer via a serial port by connecting a tether. The vehicle can be operated with or without the tether, but all data collected during a mission must be passed through the tether to the portable computer for permanent storage. The on-board computer uses an OS-9 multitasking operating system, which allows the user to run more than one program at a time. Since there is only one main processor, the programs are executed according to a hierarchy with the most important programs having priority over less important programs.

For example, modules that read the sensors and update the thrust commands will always be executed at the sampling period, T, even if other programs such as data storage routines have to be temporarily halted. To avoid halting programs that depend on being executed at regular intervals (such a. the controller which should be executed at the sampling frequency of 5 Hz), care must to be taken in determining the computation time of the code involved. This means that the adaptive controllers presented in this thesis and implemented on the vehicle had to be coded in such a way that the on-board microprocessor could perform all necessary calculations in considerably less than one sampling period (or 0.2 seconds). Although currently the combined computational load induced by all programs other than the controller (including routines for reading the sensors, storing data, generating heading, depth and speed commands, etc.) is less than 20% of the total CPU time, it was decided that no more than 50% of the CPU time could be used by the controller to allow at least 30% CPU time for routines that might be added in the future such as survey, obstacle avoidance, and navigation routines.

2.2 Derivation of Simplified Dynamics for the *Sea Squirt*

In order to implement on-line parameter estimation for the underwater vehicle dynamic model, it was necessary to determine the equations that describe the dynamics of the *Sea Squirt*. The nominal parameters for the standard submarine equations of motion [1] are contained in [2]. The equations are reprinted in Appendix 2 for convenience. In order to implement the adaptive controller in an on-line fashion, it is desirable to simplify the equations of motion as much as possible without neglecting any crucial terms. This simplification reduces the number of states and the number of parameters that have to be identified which, in turn, reduces the computation time of the estimation and control routines and increases the convergence rate of the parameter estimates; however, care must be taken in the model reduction, for if the simplified identification model is incapable of representing the actual dynamics the estimation is doomed to failure. One significant advantage of the *Sea Squirt* dynamics is that the depth and heading dynamics can be decoupled resulting in two single-input-single-output (SISO) control problems instead of one multi-input-multi-output (MIMO) control problem which is much more difficult to deal with especially in an on-line control design.

Let us consider the depth dynamics first. The vertical thruster acts approximately through the center of gravity (c.g.), center of buoyancy (c.b.), and the center of drag (c.d.) and, since the c.g. is below the c.b., it can be assumed that both pitch and roll angles resulting from vertical motion will remain small as long as the disturbances such as waves are negligible. This is a reasonable assumption since the AUV will normally operate at a depth of at least several feet where the wave effects are considerably less than at the surface. As discussed in Section 2.1.1, the z-component of the c.d. is below the center of the forward thrusters, causing the vehicle to pitch down for non-zero forward velocities. Also, the roll angle of the vehicle will increase if the vehicle experiences significant side slip, v, (side slip increases with forward speed and with yaw rate).

Since the vehicle is designed to operate with small pitch and roll angles (sensors such as the compass will not work properly otherwise) and since neither pitch nor roll is directly controllable, it is assumed that the normal operating velocities of the vehicle will be chosen such that these angles remain small. Using these assumptions allows one to approximate the equations of motion for the depth dynamics given in Appendix 2 by the following set of equations:

$$\frac{d\ depth}{dt} = w \tag{2.2.1a}$$

$$\frac{dw}{dt} = a\,w + b\,w\,|w| + \frac{(W-B) + T_v}{mass\text{-}Z_{w'}} \tag{2.2.1b}$$

where

$$a = \frac{1}{mass\text{-}Z_{w'}} Z_{uw}\, u_o$$

$$b = \frac{1}{mass\text{-}Z_{w'}} Z_{w|w|}$$

u_o = some nominal forward velocity

w = depth velocity as a function of time

T_v = vertical thrust (pounds)

All terms not defined here are given in Appendix 1.

The previous equations assume that u_o varies slowly (i.e. its derivative is small) so that the identification routine can track the parameters that change as a function of u_o.

Since the controller will be implemented on a digital computer, it was desirable to convert the above continuous time equations to the corresponding discrete time representation. This transformation is not straight forward due to the nonlinearities involved. The derivation is presented in Appendix 3. The following discrete time depth dynamics were obtained.

$$d(N+1) = d(N) + T\, w(N) + b_1\, (T_v(N)+ c) \tag{2.2.2a}$$

$$w(N+1) = a_1\, w(N) + a_2\, w(N)\, |w(N)| + b_2\, (T_v(N) + c) \tag{2.2.2b}$$

where

d = depth

w = depth velocity

T = sampling period (.2 seconds)

b_1, a_1, a_2, b_2, c = parameters to be identified

Note the similarities between the discrete time equations (2.2.2) and the continuous time equations (2.2.1). The model given in equations (2.2.2) is a good approximation of the six-degree-of-freedom nonlinear model under the assumption that pitch and roll angles as well as their derivatives are small and $w<1$ ft/s.

The heading dynamics for the *Sea Squirt* are almost identical to its depth dynamics. The differential thrust of the two horizontal thrusters provides a torque to affect yaw acceleration and any net positive (or negative) offset in both thrusters determines the forward (or backward) velocity of the vehicle. As discussed earlier in this chapter, it is assumed that both pitch and roll angles remain small during heading control. Also, it is assumed that the side slip velocity, v, is small and, therefore, does not significantly affect heading. This is a reasonable assumption since the vehicle drag in the y-direction is large. Furthermore, the vehicle is fairly symmetric (in terms of drag) about the y-z plane so that even when side slip occurs, the resulting heading change will be small. Of course, the true validity of all assumptions will be determined by on-line performance. With these assumptions the form of the heading dynamics turns out to be identical to that of the depth dynamics with depth replaced by yaw (heading) and depth rate replaced by yaw rate (heading rate). Although there is no heading term corresponding to the buoyancy term, c, in the depth dynamics, this biasing term was retained in the heading model to account for any steady-state differences in the two horizontal thrusters as well as the effects of cross currents. For simplicity, the same names will be used for the heading and depth parameters except that all heading parameters will begin with an *h* (i.e. hb_1, ha_1, ha_2, hb_2, hc).

CHAPTER 3

ADAPTIVE CONTROL

The indirect adaptive controller consists of a parameter identification scheme and a controller whose gains are calculated in an on-line fashion from the identified plant model. The structure of the plant model must be known a priori, but the coefficients or parameters involved can be estimated using the input/output information available. The block diagram in Figure 3.1 shows the operation of the indirect adaptive controller. Once the system identification has identified the model parameters, a controller can be designed on-line based on this estimated plant.

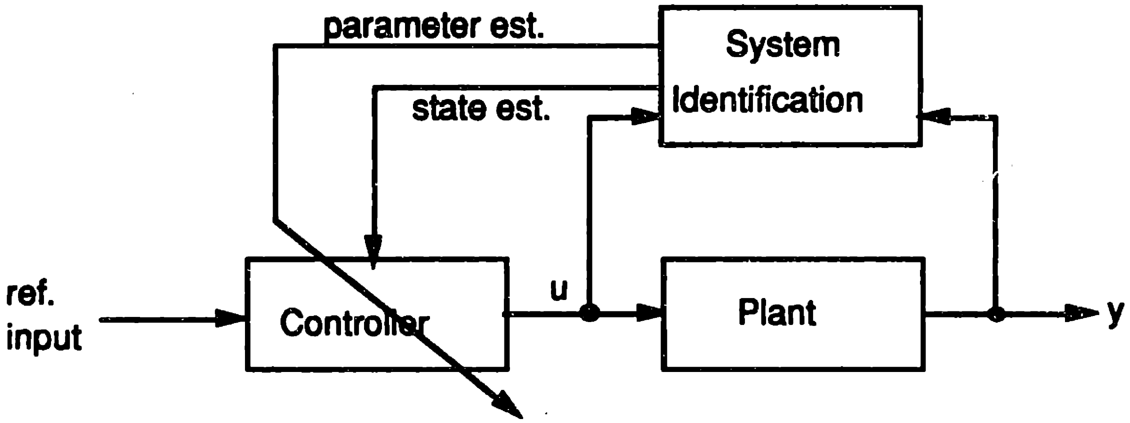


Figure 3.1: Block Diagram of the Indirect Adaptive Controller

In the block diagram the reference input is compared to the estimated plant output, $\hat{y}$ (i.e. the state estimates), to produce an internal error signal. The controller uses this error signal to generate the control, u, which drives the actuators of the system. The identification block estimates the plant parameters (and perhaps the states) from the control signal, u, and the output measurement, y. This information is then used to update the controller gains according to one of several control design methodologies. Depending on the identification methodology either the actual measurement or the filtered estimates are used in the control law. Sections 3.1 and 3.2 describe the different identification methods and control designs which will be implemented usıng the above indirect adaptive control scheme.

3.1 System Identification

The system identification is perhaps the most important element of the indirect adaptive controller because the performance of the controller is directly related to the accuracy of the plant model available. Since the controller is designed on-line based on the parameter estimates obtained from the identification method, any inaccuracies in these parameter estimates will degrade the performance of the controller. For this reason, it was thought necessary to research several estimation methods before deciding which would be best suited for the problem at hand. Several commonly used methods are given below:

1. Recursive Least-Square (RLS)
2. Normalized Least-Mean-Square (NLMS)

3. Output Error Minimization (OEM)

4. Extended Kalman Filter (KF)

Before discussing each of the above methods in detail, it should be noted that they all use the following general update equation for recursive estimation given by

$$\hat{\theta}(N+1) = \hat{\theta}(N) + K(N)\left[\, y(N+1) - \hat{y}(N+1)\right] \qquad (3.1.1)$$

where

$\hat{\theta}(N+1)$ = the parameter estimates at time N+1

$\hat{y}(N+1)$ = the estimate of the output based on information available at time N

$K(N)$ = some gain vector

$y(N+1)$ = the output at time N+1

In equation 3.1.1, only the gain vector, K(N), changes from one method to the other. Sections 3.1.1 to 3.1.4 discuss how K(N) is calculated for each of the four identification methods listed previously. All of the methods are discussed extensively in the literature and therefore, only the basic equations necessary for their implementation are given in the sections to follow. Table 3.1 will summarize the advantages and disadvantages of each of the identification methods. Since the indirect adaptive controller will be implemented on-line on a digital computer, discrete time equations are used.

3.1.1 Recursive Least Square

Unlike the extended Kalman Filter, the RLS, NLMS, and OEM methods cannot be applied to a nonlinear plant without modifications. In order to clarify this point, we will present the RLS, NLMS, and OEM methods as they apply to a linear plant model and then discuss the modifications necessary to apply these methods to the reduced order nonlinear *Sea Squirt* model given in equations (2.2.2).

RLS Applied to a Linear Plant Model

Assume that the plant to be identified is given by the linear discrete time equation

$$y(N+1) = \sum_{i=0}^{n-1} a_i\ y(N-i) + \sum_{i=0}^{m-1} b_i\ u(N-i) \tag{3.1.2}$$

where a_i and b_i are the unknown parameters.
Define the parameter vector

$$\theta = [a_1\ a_2\ ...\ a_n\ \ b_1\ b_2\ ...\ b_m]^T \tag{3.1.3}$$

and a regressor

$$X(N) = [y(N)\ y(N\ -1)\ ...\ y(N\text{-}n\ +1)\ \ u(N)\ u(N\text{-}1)\ ...\ u(N\text{-}m\text{+}1)]^T \tag{3.1.4}$$

so that the estimated output is given by

$$\hat{y}(N+1) = X^T(N)\ \hat{\theta}(N) \tag{3.1.5}$$

The RLS method minimizes the cost function

$$J(\theta) = || y - \hat{y} ||_2 \qquad (3.1.6)$$

It can be shown that the minimum cost is achieved using the following parameter update technique [3].

Let the gain vector be given by

$$K(N) = \frac{P(N)\, X(N)}{X^T(N)\, P(N)\, X(N) + \lambda} \qquad (3.1.7)$$

where

$$P(N+1) = \frac{P(N) - K(N)\, X^T(N)\, P(N)}{\lambda} \qquad (3.1.8)$$

Lambda is a forgetting factor which allows the RLS method to track parameter changes "forgetting" old information exponentially fast. Lambda has to satisfy the constraint

$$0 < \lambda \leq 1 \qquad (3.1.9)$$

If lambda is equal to 1, all information will be weighed equally and there is no forgetting. If lambda is very close to 0, only the most recent input/output information is considered. In the literature the RLS method with the weighting factor, λ, not being equal to 1 is sometimes referred to as Recursive Weighted Least Square (RWLS).

As discussed in [3], P is usually initialized as a large scalar constant times the identity matrix if only limited a priori knowledge of the system is available. This is done because a large P will result in large parameter gains which, in turn, cause the parameter estimates to converge fast. For λ equal to 1, the parameter convergence will slow down as time goes on since P, and therefore K, will decrease. This means that the gains calculated in (3.1.7) will eventually be driven toward zero making the algorithm unable to track time varying parameters. Making λ less than 1 will prevent P and K from becoming too small (see (3.1.8)) and therefore, enable RWLS to track time varying parameters.

RLS Applied to a Nonlinear Plant Model *(Sea Squirt* depth dynamics)

In order to apply RLS (as well as NLMS and OEM) to the *Sea Squirt* depth dynamics given in equations (2.2.2), an appropriate regressor, X(N), and parameter vector, θ, must be defined.

The reduced order nonlinear *Sea Squirt* depth dynamics given in equations (2.2.2) can be rewritten in the following form by combining equations (2.2.2a) and (2.2.2b).

$$\begin{aligned} d(N+1) = {} & d(N) + a_1\, T\, w(N-1) + a_2\, T\, w(N-1)\, |w(N-1)| \\ & + b2\, T\, T_v(N-1) + b_1\, T_v(N) + L \end{aligned} \tag{3.1.10}$$

$$L = (b_2\, T + b_1)\, c \tag{3.1.11}$$

where

d = depth

w = depth velocity

T = sampling period (.2 seconds)

T_v = vertical thrust

b_1, a_1, a_2, b_2, L = parameters to be identified

Define the parameter vector

$$\theta = [\ b1\ \ a1\ \ a2\ b2\ L]' \qquad (3.1.12)$$

and a regressor

$$X(N) = [\ T_V(N) \quad T\,w(N) \quad T\,w(N\text{-}1)\,|w(N\text{-}1)| \quad T\,T_V(N\text{-}1) \quad 1]' \qquad (3.1.13)$$

Since a depth velocity measurement is not available on the *Sea Squirt*, w(N) in eq. (3.1.13) has to be approximated as

$$w(N) \cong (d(N) - d(N\text{-}1))/T. \qquad (3.1.14)$$

Then,

$$d(N+1) \cong X(N)^T\,\theta + d(N). \qquad (3.1.15)$$

Although the above choice of X(N) and θ results in the identification of parameter L instead of the buoyancy estimate, c can be determined easily using the relation given in equation (3.1.11). Given the parameter and regressor vector, equations (3.1.1), (3.1.7), and (3.1.8) apply for both linear and nonlinear plants and, therefore, need not be changed.

3.1.2 Normalized Least Mean Square

As the name suggests this method minimizes the mean value of the square of the output error, $y - \hat{y}$. It can be shown [3] that the necessary gain equation is given by

$$K(N) = \frac{\mu\, X(N)}{\varepsilon + X^T(N)\, X(N)} \qquad (3.1.16)$$

where ε and μ are constants and $X^T(N)\ X(N)$ is the normalizing term. The regressor, X(N), in (3.1.16) is the same as that for RLS given in (3.1.4) (or (3.1.13) for the nonlinear *Sea Squirt* depth dynamics). The ε is included to ensure that the denominator is always strictly positive and therefore, ε is chosen to be a small positive number.

The μ in (3.1.16) determines the speed of convergence of algorithm and has to satisfy the following constraint to ensure stability:

$$0 < \mu < 2 \qquad (3.1.17)$$

The stability proof is given in [3]. Note that the RLS equations (3.1.7) and (3.1.8) reduce to equation (3.1.16) when P is replaced by a scalar times the identity matrix.

The convergence rate of NLMS is fixed by μ. The convergence rate of RLS varies depending on the covariance of the regressor. Since the covariance in low noise conditions is small, RLS generally converges faster than NLMS under these conditions. The reason NLMS is sometimes preferred to RLS is that it performs better in situations where the sensor noise is high. This is demonstrated through simulation in Section 4.1. NLMS also has the advantage that it uses less CPU time than RLS as shown in Table 3.1. In the NLMS method the convergence rate of some parameters may be substantially different from that of other parameters if the magnitude of the parameters differ significantly. Although this problem can be eliminated by scaling the parameter vector, θ, and the regressor, X, it is not easy to determine the scaling factors especially if the parameters are not known a priori. Scaling the parameter and regressor vector may also help by decreasing numeric problems when the computational accuracy is limited.

3.1.3 Output Error Minimization

This method is similar to the RLS and NLMS in that the goal is still to have $\hat{y}$ approximate y in the least squares sense. The main difference between this method and RLS or NLMS is that it utilizes the regressor (for the linear plant model)

$$\hat{X}(N) = [\hat{y}(N)\ \hat{y}(N-1)\ \ldots\ \hat{y}(N-n+1)\quad u(N)\ u(N-1)\ \ldots\ u(N-m+1)]^T \qquad (3.1.18)$$

Note that the estimated output is used in the regressor instead of the actual output as was the case for the RLS and NLMS methods. As a result of using the estimated output in the regressor, the OEM is able to produce unbiased parameter estimates. As is discussed [3], RLS and NLMS are both biased estimators. For the reduced order nonlinear *Sea Squirt* model, the OEM method uses the same regressor as given in equation (3.1.13) with the depth rate replaced by the estimated depth rate.

Even though the regressor chosen here has the advantage of providing unbiased parameter estimates, it makes it difficult to find a closed-form solution of the desired minimization as discussed in [3]. Using the approximations discussed in the reference (i.e. small μ) a closed-form solution can be obtained. For the method discussed here, the gain is given by (see [3])

$$K(N) = \mu \Psi(N) \tag{3.1.19}$$

where

$$\Psi(N) = X(N) + \sum_{i=1}^{n} \hat{a}_i(N)\, \Psi(N-i) \tag{3.1.20}$$

The $\hat{a}_i(N)$ is the estimate at time N of a_i given in (3.1.2). For the nonlinear *Sea Squirt* depth dynamics the gain will be calculated as $K(N) = \mu\ X(N)$ since equation (3.1.20), which assumes a finite impulse response (FIR), does not apply to the nonlinear model.

The μ is a constant that has to be determined according to the noise conditions and desired rate of convergence just like that of the NLMS method. However, the above recursive estimation algorithm for OEM is not necessarily stable for all choices of μ if the plant has poles close to the unit circle. This means that μ has to be chosen small enough to ensure stability causing the convergence rate to be relatively slow. This is one of the main disadvantages of OEM compared to the KF method discussed in Section 3.1.4. Also, the OEM presented above requires that the parameters and regressor be scaled properly as was the case in NLMS.

3.1.4 Extended Kalman Filter

The extended Kalman Filter also minimizes the cost function

$$J(\theta) = \| y - \hat{y} \|_2 \tag{3.1.21}$$

just as the other methods do, but the KF generates the optimal linear parameter and state estimates given that the measurement noise, r, and process noise, q, is white and Gaussian. To define r and q more clearly, let's look at the state space representation of the plant given by

$$x(N) = f\left(x(N\text{-}1),\ u(N),\ \theta\right) + q[N] \tag{3.1.22}$$

$$y(N) = C\,x(N) + r[N] \tag{3.1.23}$$

where

$x(N)$ = state vector

$y(N)$ = output

q = process noise

r = measurement noise

Define the covariance of q and r as

$$\delta[\tau]\,Q = E\{q[N]^T q[N\text{-}\tau]\}$$

$$\delta[\tau]\,R = E\{r[N]^T r[N\text{-}\tau]\}$$

The main difference between the KF and the other 3 identification methods discussed is that the KF assumes a structure for the state dynamics to obtain state estimates. It estimates both the states, x, and the parameters, θ, by augmenting the state vector as follows:

$$z(N) = \begin{bmatrix} \theta(N) \\ x(N) \end{bmatrix} = \begin{bmatrix} \theta(N-1) \\ f\{x(N-1), u(N), \theta\} \end{bmatrix} \qquad (3.1.24)$$

The output is then given by

$$y(N) = h(z(N)) \qquad (3.1.25)$$

where h is a function of the augmented state.

Since we are now dealing with state estimates as well as parameter estimates, the general update equation for recursive estimation has to be written in two parts. One part extrapolates the state estimates using the equation

$$\hat{x}(N)^{-} = \hat{f}\left(\hat{x}(N-1)^{+}, u(N), \hat{\theta}\right) \qquad (3.1.26)$$

It is assumed that the parameter estimates remain constant over one sampling period so that only the state vector is updated, not the augmented state vector. The superscript (+) and superscript (-) are introduced to differentiate the state estimate extrapolation from the state estimate update which is obtained using the Kalman gain, K(N), as follows:

$$\hat{z}(N)^{+} = \hat{z}(N)^{-} + K(N)\left[y(N) - H(N)\,\hat{z}(N)^{-}\right] \qquad (3.1.27)$$

where $\hat{z}(N)$ is the estimate of the augmented state given in (3.1.24).

It can be shown [4] that the optimal gain (assuming r and q to be Gaussian) is given by

$$K(N) = P(N)^{-}\,H(N)\left[R + H^{T}(N)\,P(N)^{-}\,H(N)\right]^{-1} \qquad (3.1.28)$$

where

$$P(N)^{-} = F(N-1)\,P(N-1)^{+}\,F^{T}(N-1) + Q \qquad (3.1.29)$$

$$P(N)^{+} = \left(I - K(N)\,H(N)\right)P(N)^{-} \qquad (3.1.30)$$

F and H are the linearized versions of f and h and are given by

$$F(N) = \begin{bmatrix} I & 0 \\ \frac{\partial f(x)}{\partial \theta} & \frac{\partial f(x)}{\partial x} \end{bmatrix} \qquad H(N) = \begin{bmatrix} \frac{\partial h(x)}{\partial \theta} \\ \frac{\partial h(x)}{\partial x} \end{bmatrix} \qquad (3.1.31)$$

The KF is an unbiased estimator just as the OEM method. Since the OEM method only adjusts parameters, instability can result by $(y-\hat{y})$ diverging. The KF overcomes this difficulty by adjusting both the state estimates and parameter estimates. Unlike RLS, NLMS, and OEM, the KF can be directly applied to systems where the parameters being identified do not appear linearly in the dynamics. Another advantage of the KF is that it does not require the scaling of parameters since the covariance matrix, Q, of the parameter's driving noise can be selected to accommodate scaling differences. However, in some cases the scaling of the parameters may increase the performance of the KF by decreasing numeric problems when computational accuracy is limited. The main disadvantage of the KF is its complexity. It involves more computation than the other methods because it takes the dynamics of the system and noise processes into consideration to obtain state estimates. This means that for a second order system with 5 parameters, the KF has to deal with matrices of dimension 7 by 7 while the RLS method only has to deal with matrices of dimension 5 by 5.

3.1.5 Summary of Identification Methods

Table 3.1 summarizes the properties of each of the four identification methods described in Sections 3.1.1 to 3.1.4. The main advantages and disadvantages are given. The approximate number of floating point operations given in Table 3.1 for each of the methods were obtained through Matlab simulations and apply to the reduced order depth dynamics (5 parameters and 2 states) of the *Sea Squirt* obtained in Chapter 2. The number of operations listed include all calculations necessary for each update of the parameter (and state) vector. It should be noted that the code used in calculating the floating point operations was not optimized for the given plant to make the comparison more general. In practice, the number of computations of the RLS and KF can be reduced substantially by taking into consideration the structure of the plant (i.e. certain matrix multiplications can be simplified if the matrix contains only a few nonzero elements). However, the relative outcome of the comparison should be the same with NLMS and OEM having the least computations and the KF the most.

Table 3.1: Summary of the Estimation Methods

	RLS	NLMS	OEM	KF
Conv. Rate	Fast convergence rate	Slower convergence rate than RLS	Slower convergence than other methods	Relatively fast convergence rate
Number of Calc.	Number of floating point calculations = 623	Number of floating point calculations = 42	Number of floating point calculations = 47	Number of floating point calculations = 2389
Ease of Imp.	Simple to implement	Simple to implement	Simple to implement	More complex algorithm than other methods
Biasness	Biased estimator	Biased estimator	Unbiased estimator	Unbiased estimator
State Est.	No	No	No	Yes
Noise Stability	Yes	Yes	Some stability problems for poles close to the unit circle	Yes

3.2 Feedback Control

The main objective of this thesis was to design an adaptive controller for the *Sea Squirt* with good performance characteristics. However, since the *Sea Squirt* dynamics are not well known and change frequently, choosing reasonable performance specifications was not straightforward. The performance criteria that we were mostly interested in was fast response time with little or no overshoot, low output variance once the desired depth (or heading) was obtained and low power consumption (i.e. unnecessary thrust variations should be avoided). The problem lies in quantifying these performance specifications for the *Sea Squirt*. In order to specify an achievable response time, past in-water data was used as a guideline. For example, the maximum depth velocity of the vehicle was known to be less than 1 ft/s which means that the response time for a 1 foot step command must be larger than 1 second. However, since we are interested in the response time including the time it takes to accelerate and decelerate the vehicle (since small overshoot is desired), the best possible response time must be of the order of several seconds. Although we now have an approximation on the lower bound of the achievable response time, perhaps a more exact bound could be obtained from the reduced order *Sea Squirt* dynamics (equations 2.2.2) with the parameter estimates (a_1, a_2, b_1, b_2) obtained through off-line identification using in-water data. This is the approach that was used since the model obtained from in-water data should be a good approximation to the actual vehicle assuming that the parameter estimates are correct. To verify the validity of this model (at least in terms of the response time), several simulations were done using the six-degree-of-freedom nonlinear model of the *Sea Squirt*.

For the *Sea Squirt* model given in equation (2.2.2) with the corresponding parameter estimates obtained from in-water data using the extended Kalman Filter (a1=0.9, a2=-0.2, b1=0.002, and b2=0.02), the best achievable response time of the vehicle to a 1 foot depth command change was found to be 3.6 seconds. This response time was obtained by applying maximum thrust (5 pounds) for 2.4 seconds and minimum thrust (-5 pounds) for 1.2 seconds to the reduced order model (assuming no net buoyancy). This means that 3.6 seconds is a lower bound for the response time (for a 1 foot depth change with small overshoot) assuming that the parameter estimates obtained from the extended Kalman Filter are correct.

However, since this lower bound can only be achieved in the ideal case (i.e. no net buoyancy, no unmodeled dynamics, correct thruster calibration, charged batteries, etc.), the performance specification on the desired response time must be greater than 3.6 seconds if it is to be achievable. Also, it is not desirable to operate at the thrust limits if the depth error is small since this would cause unnecessary power consumption (thrust variations) in the presence of noise. With these considerations in mind (and after several simulations using the six-degree-of-freedom nonlinear *Sea Squirt* model) it was decided that a response time of 5 seconds should be achievable with little or no overshoot on the actual vehicle. This response time applies to a 1 foot depth command change. Step command changes larger than 2 feet (the response is assumed to be linear for command changes less than 2 feet) will have longer response times due to actuator saturation. A similar approach was used to obtain a bound for the heading response time. It will be assumed in this chapter that the response time is measured as the time required for the depth to reach the commanded depth to within 5% for a 1 foot depth change.

Another performance specification is that the output (depth or heading) variations from the desired setpoint should be as small as possible. However, in order to quantify this specification, a covariance analysis of the vehicle dynamics would be necessary. This is not possible because the plant parameters as well as the modeling uncertainty is not known. One way to overcome this problem would be to assume bounds on the modeling uncertainty as well as the noise and disturbances. If these bounds were known, one could design a controller to minimize (at least for the non-adaptive case) the output variance based on these bounds. However, in the case of the *Sea Squirt*, these bounds are not known. Therefore, it was decided to design the controller to have the desired response time to a unit step change in depth with little or no overshoot and then evaluate its performance in terms of the output variance under various conditions (i.e. noise, disturbances, etc.) using the six-degree-of-freedom nonlinear simulation. The above discussion on the output variance specification also applies to the thrust variance.

The following list summarizes the performance specifications for the *Sea Squirt* controller:

1. Bandwidth - The response time to a unit step command change to within 5% of the desired depth should be approximately 5 seconds. This specification applies only to the linear portion of the response (i.e. within 2ft of the commanded depth) since the response time to large step command changes will take longer due to actuator saturation. The 5 second specification was chosen loose enough to ensure it would be achievable given reasonable variations in the vehicle's buoyancy and dynamics.
2. Overshoot - It is undesirable to have significant (more than 5%) overshoot in closed-loop response of the vehicle.

3. Output variance - Once the desired depth has been obtained, the variations from this setpoint should be small.

4. Thrust variance - Since the power consumption of the vehicle increases with increased control action, unnecessary oscillations in the thrust commands should be avoided.

As with the identification methods, it was thought necessary to compare several different control design methods before deciding which one would be best for the AUV. Although there are many popular control designs used today, not all of them are suited for on-line implementation on the *Sea Squirt*. For example, the iterative nature of robust control design techniques would have been too complex for the limited computing power available. The control techniques that were implemented on the *Sea Squirt* include pole-placement, sliding mode, and LQG. These methods were chosen for several reasons:

1. The pole-placement was chosen because it allows the direct placement of the closed-loop eigenvalues. This makes it a very simple algorithm to obtain the desired closed-loop bandwidth (and hence response time). Also, since the eigenvalues are related to the damping, one can easily place the closed-loop poles such that the step response of the closed-loop system will have little or no overshoot.

2. The sliding mode algorithm was chosen because it enables one to easily incorporate modeling errors into the control algorithm by explicitly taking into consideration nonlinear dynamics of the system. The details on how to represent modeling errors and disturbances (if bounds are known on these uncertainties) in discrete sliding mode are given in [5]. Since SM uses a reference trajectory which determines the closed-loop system behavior, the performance specifications (small overshoot and desired bandwidth) can be delt with easily by correctly choosing the desired reference trajectory and the controller parameters discussed in Section 3.2.3. Although the algorithm is more involved than the pole-placement, it is simpler in terms of implementation than the LQG method.

3. The LQG method was chosen since given that the plant model is correct and given that the control and state weighting matrices can be found to satisfy the response time specification (with little or no overshoot), it should optimize the state/control variance tradeoff. This is an advantage since the output and control variance performance constraints could not be directly addressed by the PP and SM designs. Although the LQ control system (i.e. LQG without an observer) has guaranteed phase and gain margins, these properties cannot be guaranteed for the LQG controller. However, the fact that these properties cannot be guaranteed for the LQG design does not imply that they may not be achieved in this specific application. The extend to which these properties do carry over in this

application could give the LQG design an additional advantage over the other control design methods and, therefore, is worth evaluating. One drawback to this algorithm is that it involves the solution of the matrix Riccati equation which makes it much more computationally complex than either of the above methods.

The following sections describe the pole-placement, sliding mode, and LQG methods, pointing out there advantages and disadvantages. Only the discrete version of each of the controllers will be discussed because the parameter identification was also done in discrete time and thus only the discrete time parameter estimates are available. Since the control design will be done on-line using these parameter estimates, discrete time control is required. The exact equations used in the implementation on the *Sea Squirt* will be given.

3.2.1 Pole-Placement (PP)

Since pole-placement (as well as LQG) is a linear control design, it is not clear what should be done to account for the nonlinear terms such as the one present in the reduced *Sea Squirt* dynamics given in equations (2.2.2). Although the effects of the drag term could have been reduced by subtracting $a_2 w|w|/b_2$ from the control input or by using some gain-scheduling technique based on w, neither proved necessary here (since the performance specifications were met without using these techniques). Linearizing equations (2.2.2) about w=0 (recall that w is the depth velocity) eliminates the nonlinear term. For this reasons it was decided that a pole-placement design based only on the linear portion of the model given by equations (2.2.2) would be sufficient here (the nonlinear term is maintained in the model for accuracy of the parameter identification). Therefore, the open-loop system is approximated as follows:

$$x(N+1) = A\ x(N) + B\ u(N) \qquad (3.2.1)$$

where

$$\begin{aligned} x(N+1) &= \text{the state vector at time } N+1 \\ &= [x_1\ x_2]^T \\ u(N) &= \text{the control input at time } N \end{aligned}$$

If we define the control signal as

$$u(N) = -K\ e(N) \qquad (3.2.2)$$

where e(N) is the error between the reference command, r(N), and the state vector, x(N), then the closed-loop system is given by

$$x(N+1) = A_{cl}\ x(N) + B\ K\ \ r(N) \qquad (3.2.3)$$

where

$$A_{cl} = A - B\ K$$

In the pole-placement design K is chosen such that the eigenvalues of A_{cl} are at the desired location [6], [7]. For low order systems (such as the reduced order *Sea Squirt* dynamics derived in Chapter 2) equation (3.2.3) can be solved analytically for the feedback gain vector K. This results in very simple code when implemented on a digital computer and minimal computational overhead.

In general, the desired pole location depends on the performance that is needed and achievable by the actuators. Especially undesirable is when actuator saturation occurs unnoticed. Since the identification algorithm uses the commanded control, not the actual control, to estimate the parameters, any difference between the two control values will cause incorrect parameter estimation.

Since the closed-loop poles are being placed in discrete time, their magnitudes have to be less than one in order for the system to be stable. Also, the poles should be chosen close to the real axis and should have a positive real part if oscillations are not desirable. Determining the exact location of the desired poles can be done in several ways. One method is to determine the continuous time poles based on the desired frequency response and then approximate these poles in discrete time. Another method is to use simulation in discrete time to find the poles that meet the desired performance specifications with minimal actuator saturation. Clearly, there is some trial-and-error involved in the latter method. For low order systems either method is acceptable.

Pole-Placement Applied to the *Sea Squirt*

The open-loop reduced order discrete time depth dynamics for the *Sea Squirt* are reprinted below (3.2.4) for convenience.

$$d(N+1) = d(N) + T\ w(N) + b_1\ (T_v(N)+ c) \tag{3.2.4a}$$

$$w(N+1) = a_1\ w(N) + a_2\ w(N)\ |w(N)| + b_2\ (T_v(N) + c) \tag{3.2.4b}$$

where

d = depth
w = depth velocity
T = sampling period (.2 seconds)
b_1,a_1,a_2,b_2,c = parameters to be identified

Since equations (3.2.4) include a buoyancy term, c, which represents a constant, steady force on the vehicle, the control law given in (3.2.2) would result in a non-zero steady-state error. To eliminate this error, one can either add an integrator state to the system which integrates the depth error, or one can subtract out the buoyancy term using the buoyancy estimate obtained from the identification method. Adding an integrator state causes the system dimension to increase from two to three which increases the computational complexity of the algorithm considerably and makes it more difficult to determine the desired closed-loop pole locations. Also, integral control leads to problems such as integrator wind-up. The only advantage of using the integrator method over the other method is that it would eliminate steady-state error even if the buoyancy estimate is poor. However, it was found that the parameter estimates converge as a group so that if the buoyancy estimate is poor, most likely all parameter estimates will be poor as well, resulting in degraded performance. Therefore, even if the integrator eliminates steady-state error with a poor buoyancy estimate, the overall response of the closed-loop system will more than likely be less than desirable anyway. Since the performance of the closed-loop system will be degraded by a poor buoyancy estimate, no matter which of the above methods is used to eliminate steady-state error, we do not use integral control; and, we develop in Chapter 4 an improved method to quickly estimate the buoyancy.

The pole-placement algorithm implemented on the *Sea Squirt* is given by

$$T_v(N) = K_1 \; (\mathrm{ref}(N) - d(N)) - K_2 \; w(N) - \hat{c} \qquad (3.2.5)$$

where $\hat{c}$ corresponds to the net buoyancy estimate and T_v is the vertical thrust. It should be noted that the actual depth rate, w(N), is not measured, but its estimate is assumed to be available and used in the place of w(N). As will be discussed in the next chapter, the depth, d(N), will also be estimated and used instead of the actual depth to filter out noise and quantization errors. Substituting equation (3.2.5) into equations (3.2.4) will result in the cancellation of the "c" term as desired. If the buoyancy estimate is correct, the steady-state error will be zero.

For the *Sea Squirt* the closed-loop pole locations were chosen to meet the performance specifications given at the beginning of this section as good as possible. Since small overshoot was desired, the poles had to be chosen positive (positive real part) and close to the real axis to obtain a damping close to 1. Since actuator limitations had to be considered as well, the exact pole locations were chosen based on the six-degree-of-freedom nonlinear simulation of the *Sea Squirt* such that the response time (bandwidth) and damping specifications were met. Through these simulations it was found that pole locations of 0.7 and 0.8 resulted in a closed-loop control system which satisfactorily met the desired performance specifications. Figure 3.2 shows the linear response corresponding to these pole locations. As can be seen from the figure, the response has no overshoot as desired and meets the response time specification. It should be noted that the control signal is limited to +/- 5 pounds for all controllers implemented on the *Sea Squirt* to avoid the actual thruster saturation limits. This ensures that the thrusters operate only in the linear region of the model (Figure 2.3) so that the estimation algorithm receives accurate data.

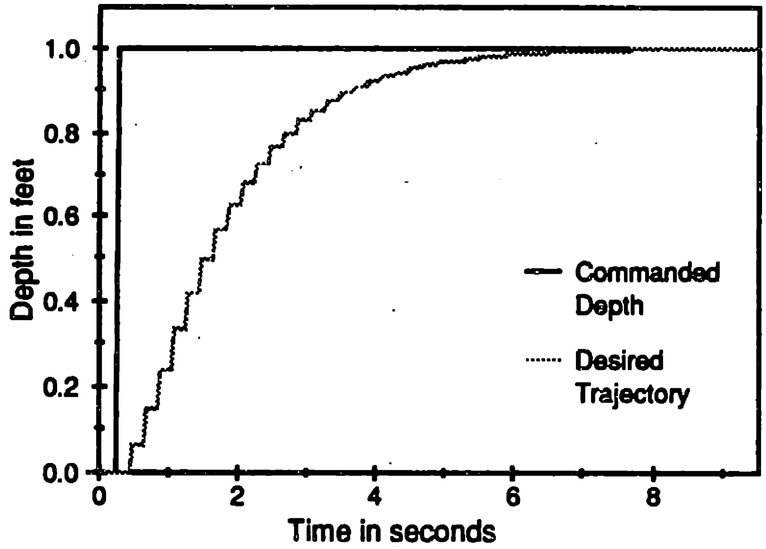


Figure 3.2: Step Response Corresponding to Poles at 0.7 and 0.8

Due to saturation the linear response shown in Figure 3.2 will not be attainable if the depth (or heading) error is large enough. For example, if a depth change of 100 feet is commanded, the vehicle will descend at the maximum attainable rate (given by the 5 lb thrust limit) until the depth error is less than some amount. For the poles chosen above this amount turned out to be about 2 feet for the *Sea Squirt.* For the last few feet the response of the vehicle should be identical to that dictated by the desired poles. Depending on the application of the vehicle, it may be desirable to generate a reference rate so that the depth velocity can be controlled as well as the depth itself. This change can easily be made in the future and requires only that the rate error be included in equation (3.2.5) instead of the rate, w(N).

In order to implement the above pole-placement algorithm one has to solve for the feedback gain vector, K, given the parameter estimates a_1,b_1, and b_2, and the desired closed-loop poles. This was done by analytically finding the characteristic equation and equating it to the desired closed-loop equation. For the system given in equations (3.2.4), the two feedback gains could be found analytically in a straightforward manner and are given in equation (3.2.6) below

$$K_1 = \frac{1}{\det A} [-b_2(p_1 + 1 + a_1) - b_2(p_2 - a_1)] \qquad (3.2.6a)$$

$$K_2 = \frac{1}{\det A} [-(T\, b_2 - a_1 b_1)(p_1 + 1 + a_1) + b_1(p_2 - a_1)] \qquad (3.2.6b)$$

where

$\det A = [-b_1 b_2 - b_2(T\, b_2 - a_1 b_1)]$
p_1, p_2 = the desired closed-loop poles

Since the identification method estimates the unknown parameters in equation (3.2.6) on-line, the pole-placement will be adaptive and the response of the closed-loop system should converge on the desired closed-loop response shown in Figure 3.2 as the parameter estimates converge (assuming that they converge to the correct values).

3.2.2 Linear Quadratic Gaussian (LQG)

The only difference between the LQG design technique and the pole-placement method is in the way in which the feedback gain vector, K, is calculated and therefore, equation (3.2.5) will also apply here. As discussed above, the estimated state vector will be used in place of the actual states (when the KF is used) to filter out some of the measurement noise (for this reason LQG is used and not LQR). In LQG, K will be determined such that the following cost function

$$J = \sum_{i=0}^{\infty} (x^T Q x + u^T R u) \tag{3.2.7}$$

is minimized. For the minimization all parameters including c are assumed to be correct.

Just as the closed-loop poles were the design parameters in the pole-placement algorithm, the Q and R matrices are the design parameters that will determine the characteristics of the closed-loop system in LQG. Both Q and R will be fixed for this control design (i.e. neither will be updated on-line). Which of the two design methods, PP or LQG, is easier to implement (i.e. placing poles or finding the weighting matrices) depends on the design criterion that needs to be met and how difficult it is to represent in terms of the cost function given in equation (3.2.7). This point will be discussed later in this section when the LQG design is applied to the *Sea Squirt.*

The feedback gain vector that minimizes the cost function in equation (3.2.7) is given by

$$K = \left(R + B^T X B\right)^{-1} B^T X A \qquad (3.2.8)$$

where X is the positive semidefinite solution of the discrete time matrix Riccati equation

$$X - A^T X A + A^T X B\left(R + B^T X B\right)^{-1} B^T X A - Q = 0 \qquad (3.2.9)$$

The proof is given in [6].

LQG Applied to the *Sea Squirt*

Although the Riccati equation can be solved numerically, the general algorithm is fairly complex and computationally intense [8] and [9]. In order to implement the algorithm on-line, the steady-state Riccati equation was solved analytically for the second order dynamics of the *Sea Squirt* (linear portion of equations 3.2.4). The analytic solution for this particular plant is shown in Appendix 4.

Since the LQG cost function determines the feedback gains which optimize the state/control variance tradeoff for any given R and Q, the output and control variance should be low as desired. However, finding the state weighting matrix, Q, and the control weighting, R, such that the closed-loop response has small overshoot (for a step command change) and the desired response time given in the performance specifications, turned out to be more difficult than placing the closed-loop poles as was done in the pole-placement design. The problem was that these specifications could not be easily represented by the LQG cost function. For simplicity, the Q matrix was chosen to be diagonal so that only one weight had to be adjusted for each state. As was done in the PP design, the six-degree-of-freedom simulation was used to find the values of Q and R such that the desired performance specifications were met (at least in simulation). The two non-zero elements in the Q matrix were determined to give the desired damping and R (a scalar here) was adjusted to give the proper control weighting. Although other methods could have been used to determine Q and R, the approach used here seemed to work well for the given plant and performance specifications. The final values chosen for Q and R were as follows:

$$R = 0.15$$

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}$$

where $q_1 = 4$ and $q_2 = 1$. Although simulation was used to set Q and R, we rely on the reduced sensitivity of the closed-loop system to ensure that the desired closed-loop characteristics are obtained on the actual vehicle as well.

3.2.3 Sliding Mode (SM)

It should be pointed out that the discrete time sliding mode method has been researched in the literature only to a limited extend and has some drawbacks relative to the continuous time version. However, since the implementation on the AUV calls for a discrete time controller, the discrete version will be used here. Unlike the pole-placement and LQG, the sliding mode control method is nonlinear. This enables one to incorporate certain types of nonlinear model structures in the feedback control law.

To simplify the discussion of the discrete sliding mode control algorithm, the regulator problem will be discussed first. It shows how the control law is chosen such that all states are forced to zero from an arbitrary initial state. The regulator design will then be extended to the more general trajectory following case. The following discussion on discrete SM summarizes only the basics of the method as it applies to the problem at hand. A more complete presentation of both continuous and discrete sliding mode can be found in [5]. Whenever possible, the same notation will be used here as in [5] since most of the equations presented in this subsection were obtained from this reference.

The Regulator Problem

As discussed in [5], the discrete time sliding mode technique can be applied to linear systems and to nonlinear systems having the following form:

$$x'(N+1) = A'\ x(N) \qquad (3.2.10a)$$

$$x_n(N+1) = f(x(N),N) + b(x(N),N)\ u(N) \qquad (3.2.10b)$$

where

$$x'(N) = [x_1(N)\ x_2(N)\ \ldots\ x_{n-1}(N)]^T$$

$x_n(N)$ = the n^{th} state

$$x(n) = \begin{bmatrix} x'(N) \\ x_n(N) \end{bmatrix}$$

$u(N)$ = a scalar input

A' = a constant (n-1) by (n) matrix

Define the sliding surface as

$$s(N) = 0$$

where

$$s(N) = e^T x(N) \tag{3.2.11}$$

with e being an n dimensional real valued vector of the form $e = [e' \; 1]^T$ (e' is an n-1 dimensional vector).

The vector, e, is selected such that certain system properties (i.e. stability and performance) are satisfied when s(N) = 0. The control problem lies in assuring that the sliding surface (s(N)=0) is reached and that the system will remain on the surface once it is reached. The control law is formulated in two parts. One causes the system to move toward the sliding surface and another moves it along the surface. The control needed to maintain the system on the sliding surface is found by assuming that s(N)=0 and forcing s(N+1) = s(N).

Letting s(N+1) = s(N) gives

$$s(N) = e'^T A' x(N) + f + b\, u(N) \tag{3.2.12}$$

Solving for u(N) gives us the control required to keep s(N+1) = s(N)

$$u(N) = \frac{1}{b} \{ s(N) - e'^T A' x(N) - f\} \tag{3.2.13}$$

To allow for the case when the initial state does not lie on the sliding surface, the control law in (3.2.13) must be modified by adding a term that will force the system onto the surface. Equation (3.2.13) then becomes

$$u(N) = \frac{1}{b} \{ s(N) - e'^T A' x(N) - f - \eta \; sgn(s(N))\} \qquad (3.2.14)$$

where sgn( ) is the "sign" function (i.e. +/- 1 depending on the sign of s(N)).

Equation (3.2.14) guarantees that a region of radius η around the sliding surface will be reached within (|s(N=0)| / n+1) sampling periods after which the state can b guaranteed to stay within a region bounded by +/- η around the surface as discussed in [5]. The sign term in equation (3.2.14) could cause s to oscillate in this region with the average value of s not necessarily zero even when the model is correct and noise and disturbances are absent. This is due to the fact that the control law in (3.2.14) does not force s toward zero once it is less than η. It was found that changing the condition used to develop equation (3.2.12), namely s(N+1) = s(N), to

$$s(N+1) = \mu \; s(N) \qquad (3.2.15)$$

where

$$0 < \mu < 1$$

forces the average value of s towards zero even if it is less than η. This still ensures that the state stays on the sliding surface as desired. With this modification, the control input becomes

$$u(N) = \frac{1}{b} \{ \mu\, s(N) - e^{'T} A' x(N) - f - \eta \ \mathrm{sgn}(s(N)) \} \qquad (3.2.16)$$

Although the oscillations caused by the sign term in (3.2.14) still persist, they are centered around s=0 with magnitude $\eta/2$ if the control input of (3.2.16) is used instead of (3.2.14). Since the sign term would cause undesirable actuator chattering, it is replaced by a saturation function as was done in [5] so that u(N) becomes:

$$u(N) = \frac{1}{b} \{ \mu\, s(N) - e^{'T} A' x(N) - f - \eta \ \mathrm{sat}(s(N)/\Phi) \} \qquad (3.2.17)$$

where sat( ) is the saturation function shown in Figure 3.3.

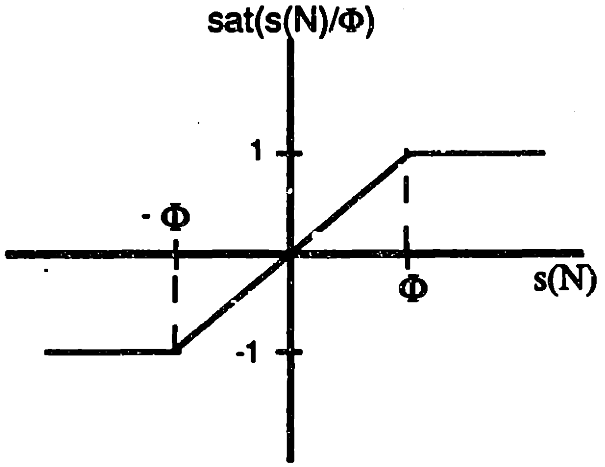


Figure 3.3: Saturation Function

The numerical value for η is determined according to the modeling uncertainty and expected noise and disturbances while Φ can be selected to obtain the desired linear s dynamics off the surface as discussed in [5]. Inside the boundary (i.e. $|s|<\Phi$), the s dynamics are given by

$$s(N+1) = \mu s(N) - \frac{\eta}{\Phi} s(N) \qquad (3.2.18)$$

assuming no disturbances or noise. The eigenvalue of equation (3.2.18) is equal to $\mu - \eta/\Phi$.

Selecting the Sliding Surface

The sliding surface is selected using a pole-placement technique which determines e' such that the closed-loop eigenvalues are at the desired location. As explained in [5], one eigenvalue always has to be chosen to be zero. The eigenvalue placement is done on the closed-loop system shown in equation (3.2.19) obtained by substituting equation (3.2.17) into equation (3.2.10):

$$x(N+1) = \begin{bmatrix} A' \\ -e'^T A' \end{bmatrix} x(N) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \{\mu s(N) - \eta \ \mathrm{sat}(s(N)/\Phi)\} \qquad (3.2.19)$$

The vector, e', can now be found using standard pole-placement techniques for the system in equation (3.2.19).

Trajectory Following

Up to now only the regulator problem was addressed. To extend the above discussion to the trajectory following case, the state vector, x, is replaced with the state error defined as

$$\tilde{x}(N) = x(N) - x_d(N) \qquad (3.2.20)$$

where $x_d(N)$ is the desired state vector at time N.

The control law for trajectory following is given by:

$$u(N) = \frac{1}{b} \{ \mu\ s(N) - e'^T A' x(N) - f + e^T x_d(N+1) - \eta\ \mathrm{sat}(s(N)/\Phi))\} \quad (3.2.21)$$

resulting in the closed-loop error dynamics:

$$\tilde{x}(N+1) = \begin{bmatrix} & A' \\ -e'^T & A' \end{bmatrix} \tilde{x}(N) \quad (3.2.22)$$

As discussed in [5], the desired state vector, $x_d(N)$, must be consistent with the system dynamics. For the system given in equation (3.2.10), if $x_{n_d}(N)$ is the desired trajectory, the other elements of $x_d(N)$ are given by:

$$x'_d(N) = A' x_d(N-1) \quad (3.2.23)$$

where $x'_d(N) = [\ x_{1_d}(N) \dots x_{n-1_d}(N)]^T$.

Sliding Mode Control Applied to the *Sea Squirt*

It was pointed out earlier that one of the drawbacks of the discrete sliding mode method is that it requires the system to be either linear or of the form given in equation (3.2.10). The reduced order *Sea Squirt* depth dynamics given in equation (3.2.4) fall into neither one of these categories. The problem is that b_1 in equation (3.2.4) is not necessarily equal to zero which means that these dynamics do not have the form of equation (3.2.10). There are several possibilities to overcome this problem.

One method is to transform the *Sea Squirt* dynamics by changing the states from d(N) and w(N) to d(N) and d(N-1). Although the resulting system will be in the form of equation (3.2.10), the transformed states will no longer include the depth rate. As will be shown in Chapter 4, using the first order difference rather than the depth rate estimate of the KF degrades the performance of the controller considerably due to the large quantization step size of the A/D converter on the *Sea Squirt*. Therefore, it was decided that a state transformation would not be desirable in this case.

An alternative method of adjusting the *Sea Squirt* dynamics to the form of equation (3.2.10) is to neglect b_1 and incorporate its affects into the modeling uncertainty (i.e. η). Clearly, this can be done only if b_1 is known to be relatively small as was the case for the *Sea Squirt*. The six-degree-of-freedom nonlinear simulation was used in determining η because the actual values of the parameters (and hence the modeling uncertainty) were not known for the AUV. How to choose η when bounds are known for the plant uncertainty and noise and disturbances is discussed in detail in [5]. Using this latter method, the *Sea Squirt* depth dynamics, given in equations (3.2.4), can be written in the form of equation (3.2.10) resulting in the following control law:

$$T_v(N) = \frac{1}{b_2}\{\ \mu\ s(N) - e'^T A'\ x(N) - f + e^T x_d(N+1) - \eta\ \mathrm{sat}(s(N)/\Phi))\} \tag{3.2.24}$$

where

$$A' = [1\ T]$$

$$\begin{aligned} f &= a_1\, w(N) + a_2\, w(N)\, |w(N)| + b2\, c \\ e^T &= [e'\ 1] \\ &= [\lambda\ 1] \end{aligned}$$

As was done with the PP and LQG controller, the state estimates (when available) will be used instead of the actual states and the plant estimate will be used instead of the actual plant dynamics to make the algorithm adaptive. It should be noted that the adaptive sliding mode algorithm presented here is not the same as those presented in [5], [10] or [11]. The algorithm in [7] and [11] use continuous time sliding mode and therefore, could not be used with the discrete time system discussed here. Our method differs from that of [5] in that an extended Kalman Filter is used to generate parameter and state estimates. The benefit of the state estimates is discussed in Chapter 4.

The model reference was generated such that it had the same second order desired response (poles at .7 and .8) as shown in Figure 3.2. This was done since the same performance specifications apply to both the SM and PP design.

The constants μ, λ, η, and Φ, were selected as follows. It should be noted that some or all of these parameters could have been made time varying as was done in [5], but were not because it would have added unnecessary complexity to the algorithm. The slope of the sliding surface, s, shown in Figure 3.4 is given by $-\lambda$ and was determined by placing the poles of the system in equation (3.2.19) at the desired location. Although the desired poles of the trajectory were .7 and .8, one of the poles of equation (3.2.19) has to be at 0 as discussed earlier. The second pole was chosen to be at .8 resulting in a λ of 1.

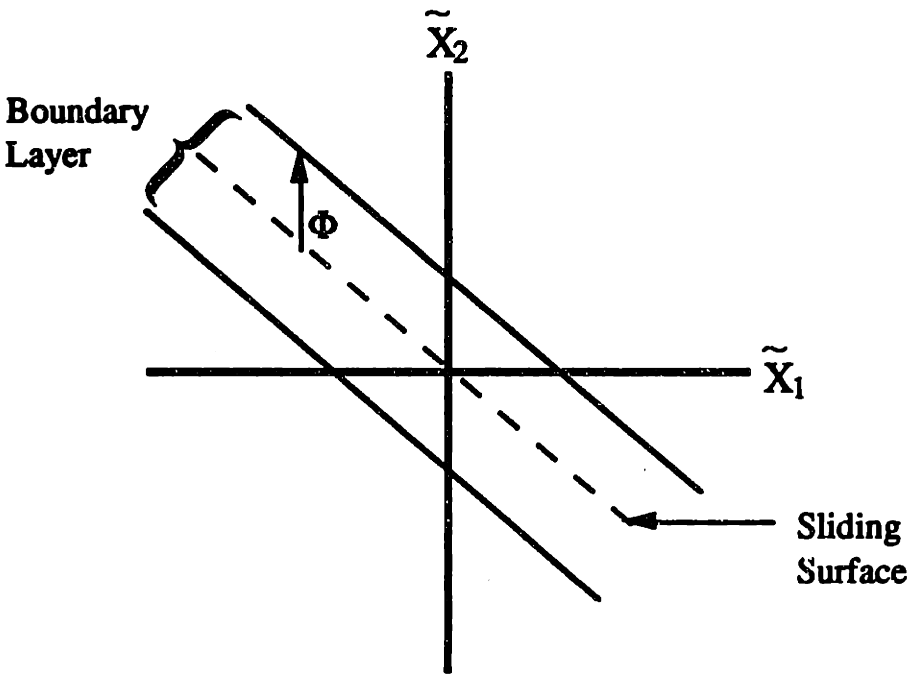


Figure 3.4: Discrete Time Sliding Surface with Boundary Layer

Determining the constants, μ, Φ, and η is somewhat more difficult than finding λ. This is due to the fact that the dynamics of s inside of the boundary layer are affected by all three constants as was shown in equation (3.2.18) and therefore, μ, Φ, and η cannot be chosen independently. Although η is usually set equal to or larger than the total system uncertainty (including modeling errors, noise, and disturbances), it cannot be quantified easily. Not only is the modeling uncertainty not known, it also changes as the identification routine improves the parameter estimates over time. For this reason system stability during the initial parameter convergence will be guaranteed using a safety net approach as discussed in Chapter 5. This means that η can be chosen to take into account only those uncertainties that persist after parameter convergence. If one assumes that the identification method correctly estimates the parameters for the reduced order *Sea Squirt* dynamics given in equations (3.2.4) then the "only" modeling uncertainty which has to be captured by η is that caused by the inability of equation (3.2.4) to fully capture all the dynamics of the actual system. Since it would have been difficult to quantify this uncertainty mathematically, the six-degree-of-freedom nonlinear *Sea Squirt* simulation was used to determine which value of η worked best. This is reasonable since the unmodeled dynamics of the actual vehicle should be similar to those of the simulation after the parameter estimates have converged (assuming that they converged to the correct parameters). However, before simulations can be done to determine the most favorable value of η, the constants Φ and μ have to be chosen.

Since the pole of the s dynamics inside of the boundary layer is given by $\mu-\eta/\Phi$ (see equation 3.2.18), the only choices of η and Φ that result in stable s dynamics are $\eta/\Phi < 1+\mu$ where $0<\mu<1$. Furthermore, the steady state value of s can be guaranteed to stay within the boundary layer, Φ, only if the eigenvalue of equation (3.2.18) is greater than zero (as discussed in [5]) which would limit the choice of η and Φ to $\eta/\Phi < \mu$. In order to allow a reasonable range of values for η and Φ and still satisfy the constraints $0<\mu<1$ and $\eta/\Phi < \mu$, μ was chosen to be .9.

The boundary layer thickness, Φ, shown in Figure 3.4, was determined along with η using the six-degree-of-freedom nonlinear simulation. Simulations with the control law designed with good parameter estimates as well as with poor parameter estimates were done to determine the most favorable values of Φ and η. The values of Φ and η which worked best overall were Φ=.5 and η=.1. The resulting pole of the s dynamics inside the boundary layer is .9-.1/.5 = .7.

CHAPTER 4

SIMULATION COMPARISON OF IDENTIFICATION METHODS AND CONTROL DESIGNS

4.1 Simulation Comparison of Identification Methods

The objective of the identification part of the indirect adaptive controller is to obtain the best possible parameter estimates of an underwater vehicle model given only the control inputs, and noisy sensor readings. Since underwater sensors generally have a low signal-to-noise ratio, the identification methods were compared mainly based on how well they performed as a function of sensor noise. However, several other issues related to on-line implementation on the *Sea Squirt* had to be addressed as well.

One consideration is the computational complexity of the estimation algorithm. Any estimation method that requires too many computations or too much memory is not acceptable. The computation time required by each of the four identification methods (RLS, NLMS, OEM, KF) discussed in Chapter 3 was given in Table 3.1. After implementing these estimation techniques in C on a Mac II and checking their execution time, it was found that all methods could be implemented on the *Sea Squirt* (recall that the *Sea Squirt* has the equivalent computing power of a Mac II) with more than enough CPU time and memory left for the control algorithm assuming second order depth and heading dynamics.

Another consideration for on-line system identification is the problem of insufficient excitation. All estimation methods rely on sufficiently exciting inputs (in this case thrust commands) for correct parameter estimation. Generally, this means that the control inputs must contain enough frequency components within the bandwidth of the system. A formal definition of sufficient excitation is given in [3] and [12]. Since the control inputs of the *Sea Squirt* are dependent on the desired vehicle path, sufficient excitation cannot always be guaranteed. This problem will be addressed in Chapter 5.

In order to compare the four identification methods (RLS, NLMS, OEM, KF) in simulation, it was decided that the comparison should be done for several linear plants as well as for the reduced order nonlinear *Sea Squirt* depth dynamics. This was done to reduce the chances that the outcome of the comparison will be plant specific. All the linear plant models used in the comparisons were chosen to have different pole locations. Since similar results were obtained for all the linear plant models compared, only one of these comparisons will be presented here as a typical example. A comparison of the identification methods applied to the reduced order nonlinear *Sea Squirt* dynamics will also be presented.

Comparison of the Identification Methods Applied to a Linear Plant

Let's assume that the discrete time model of the system to be identified is given by

$$y(N) - 1.5\ y(N-1) + .7\ y(N-2) = u(N-1) + .5\ u(N-2) + e(N) \quad (4.1.1)$$

where

$y(N)$ = the output at time N

$u(N-1)$ = the input at time N-1

$e(N)$ = zero mean gaussian noise applied at time N

This model is not intended to represent underwater vehicle dynamics and was chosen for comparison purposes only. It should be noted that the above system was also used in [13] to test different identification methods. The parameters to be identified are the coefficients of equation (4.1.1), so that the estimated output is given by

$$\hat{y}(k) + p_1 \hat{y}(k-1) + p_2 \hat{y}(k-2) = p_3 u(k-1) + p_4 u(k-2) \qquad (4.1.2)$$

where

$\hat{y}(N)$ = estimated output at time N

p_i = parameters to be identified

As in the reference [13], a Pseudo Random Binary Signal (PRBS) sequence with amplitude ±1 was chosen as the input. Although a PRBS input would not be encountered in real life, it was chosen because it ensures persistent excitation of the system. Any sufficiently exciting input signal could have been used. The problem of insufficient excitation will be addressed in Chapter 5.

The comparison results shown in Figure 4.1 were obtained through Matlab simulations of the above system. This figure plots the Euclidean norm of the final parameter error, after the parameters had sufficient time to converge (500 iterations), as a function of the signal to measurement noise ratio. All methods were initialized with the same parameter estimates and were allowed to converged for the same period of time. The values for μ, λ, ε, Q, and R, which determine the performance of the individual estimation methods, were adjusted to obtain the best possible parameter estimates at low signal-to-noise ratios and then kept constant. Since the actual parameters to be identified were constant in these simulations, the forgetting factor of the RLS method was set to 1 (i.e. no forgetting).

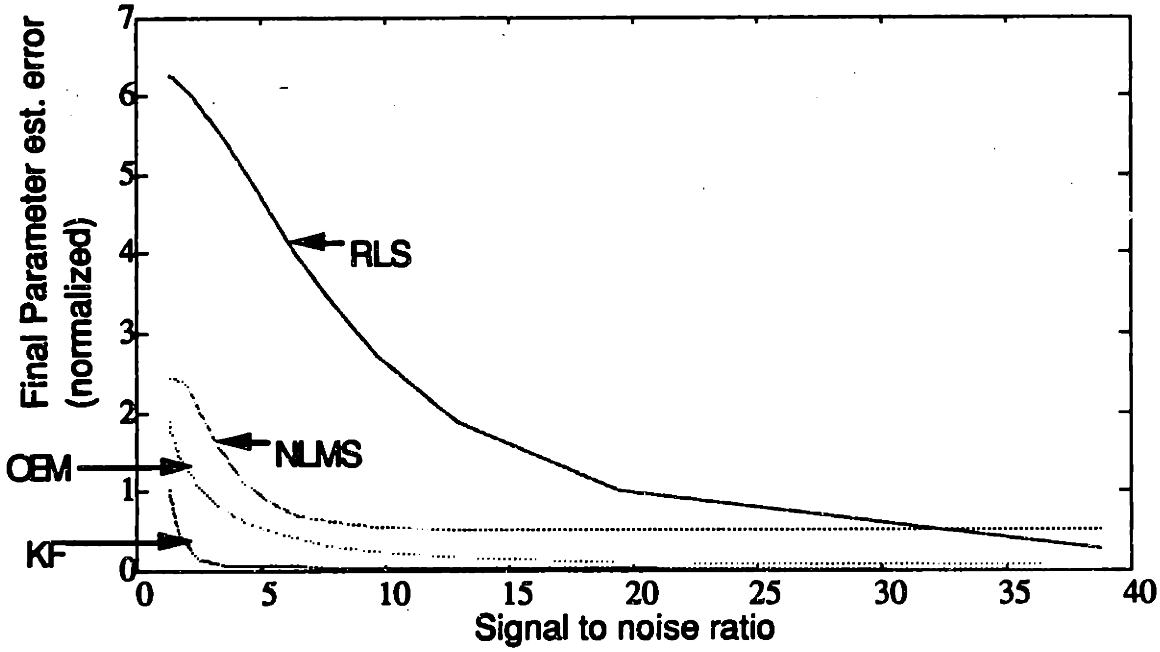


Figure 4.1: Comparison of Identification Methods for a Linear Plant

Although the above comparison is plant dependent, the identification methods were also compared for different stable linear plants and the extended Kalman Filter always performed best relative to the other methods especially for low signal-to-noise ratios. This result should be expected since the Kalman Filter is an optimal mean square linear estimator in the presence of white noise. However, since the identification method will be used on the *Sea Squirt,* the above comparison was repeated using the reduced order nonlinear *Sea Squirt* depth dynamics derived in Chapter 2.

Comparison of the Identification Methods Applied to the *Sea Squirt*

In order to represent the *Sea Squirt* depth dynamics as accurately as possible, the plant parameters (a_1, a_2, b_1, b_2, c) of the reduced order *Sea Squirt* depth dynamics (given in equations (2.2.2)) were determined using the extended Kalman Filter and in-water data. The resulting parameter estimates were a1=.9, a2=-0.2, b1=.002, b2=.02, c=-0.5. Now, this model of the *Sea Squirt* depth dynamics was used to compare the identification methods (RLS, NLMS, OEM, and KF) based on how well they could estimate the parameters given above. The comparison was done using the reduced *Sea Squirt* dynamics and not the actual vehicle since the actual vehicle parameters are not known, making it impossible to determine which method obtained the best parameter estimates.

To make the comparison of the identification methods applied to the *Sea Squirt* depth dynamics as realistic as possible, the depth quantization (0.05 ft) caused by the A/D converters was included in the plant model. Also, instead of using a PRBS sequence as the control input, a square wave of amplitude 1 with randomly changing duty ratio was applied as the commanded depth. Feedback control was used so that the vehicle depth (in simulation) would always move toward the commanded depth. Identifying the plant parameters under feedback control makes the comparison more realistic since the identification method will also have to estimate the parameters under these conditions when implemented on the vehicle. As before, the identification methods will be compared based on how well they can identify the plant parameters (in this case a_1, a_2, b_1, b_2, c) as a function of signal-to-noise ratio. However, as was discussed in Chapter 3, the NLMS and OEM methods require the scaling of the regressor and the parameter vector in order for all parameter estimates to converge at approximately the same rate. This means that NLMS and OEM will minimize the estimation error based on a scaled version of the parameters and, not necessarily the actual parameters, although the actual parameters can be determined. Figure 4.2 shows the Euclidean norm of the final parameter error (based on a_1, a_2, b_1, b_2, c, not the scaled parameters), after the parameters had sufficient time to converge (4000 iterations), as a function of signal-to-noise ratio (Gaussian white noise was added to the depth measurement).

Note that the extended Kalman Filter performs better than the other methods for all signal-to-noise ratios as shown in Figure 4.2. The fact that it performs better than RLS even for high signal-to-noise ratios is partially due to the depth velocity approximation used in the regressor (equation (3.1.13)) which is especially crude with the added quantization noise. Both NLMS and OEM did not perform well because they were unable to correctly identify the drag term, a_2. This can be explained by the fact that neither NLMS nor OEM use a covariance matrix. The correlation between the regressor components corresponding to a_1 and a_2 makes them difficult to discriminate.

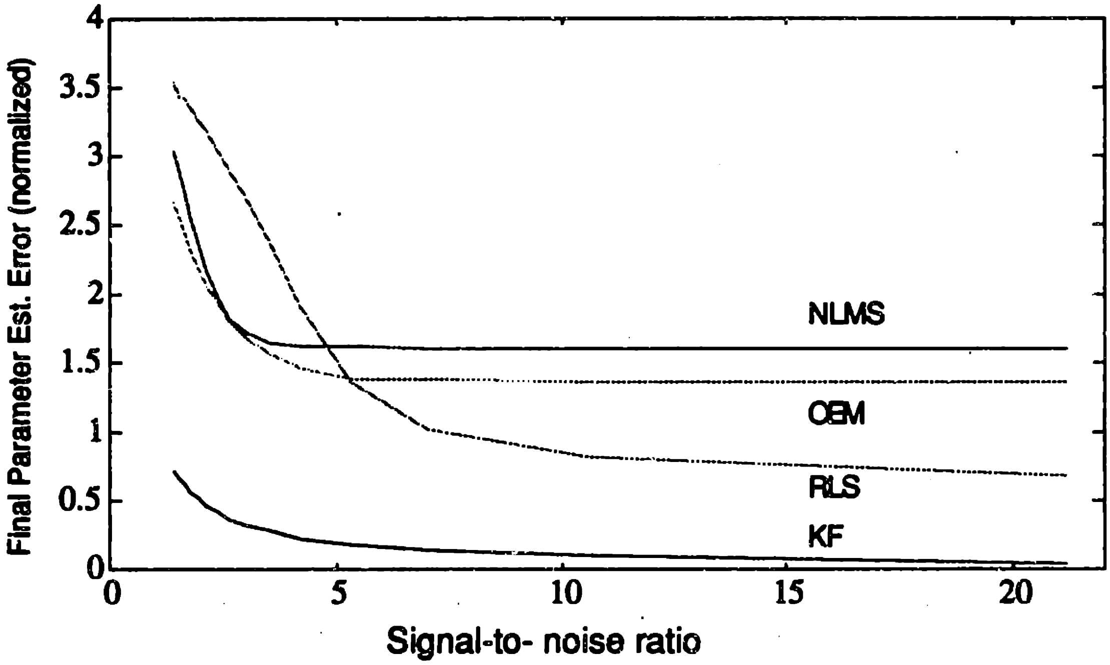


Figure 4.2: Comparison of Identification Methods for a Nonlinear Plant

From Figure 4.2 it should be clear that the KF is the only one of the four identification methods presented which gives reasonably good parameter estimates at low signal-to-noise ratios for the given plant. Another advantage of the Kalman Filter over the other identification methods is that it provides state estimates which are required for full state feedback control. This proved to be an especially important consideration since no depth rate sensor is available on the *Sea Squirt*. However, there is a tradeoff between performance and complexity. The KF requires almost 4 times as many calculations as RLS as was shown in Table 3.1. However, since the available CPU time on the *Sea Squirt* computer was sufficient to implement any of the four identification methods on-line, the complexity of the KF was not a problem. With these considerations in mind, it was decided that the advantages of the extended Kalman Filter by far outway its disadvantages. The remainder of this section will be used to discuss the exact implementation of the KF on the *Sea Squirt*.

Applying the Extended Kalman Filter to the *Sea Squirt*

The general equations necessary to implement the extended Kalman Filter were presented in Section 3.1.4. For the *Sea Squirt* depth dynamics given in Chapter 2, these equations were implemented as follows:

Let the parameter estimate vector be given by

$$\hat{\theta} = [(\hat{b}_1 \;\; \hat{a}_1 \;\; \hat{a}_2 \;\; \hat{b}_2 \;\; \hat{c}]^T \qquad (4.1.3)$$

and the state estimates by

$$\hat{x}(N) = [\hat{d}(N)^T \;\; \hat{w}(N)^T]^T \qquad (4.1.4)$$

The augmented state estimate vector for the KF is then given by

$$\hat{z}(N) = [\hat{\theta}(N)^T \;\; \hat{x}(N)^T]^T \qquad (4.1.5)$$

The state estimates are updated using the equations

$$\hat{d}(N+1)^- = \hat{d}(N)^+ + T\,\hat{w}(N)^+ + \hat{b}_1\,(T_v(N)+ \hat{c}) \qquad (4.1.6a)$$

$$\hat{w}(N+1)^- = \hat{a}_1\,\hat{w}(N)^+ + \hat{a}_2\,\hat{w}(N)^+\,|\hat{w}(N)^+| + \hat{b}_2\,(T_v(N) + \hat{c}) \qquad (4.1.6b)$$

$$\hat{\theta}(N+1)^- = \hat{\theta}(N)^+ \qquad (4.1.6.c)$$

where

$\hat{d}$ = depth estimate

$\hat{w}$ = depth velocity estimate

T = sampling period (.2 seconds)

$\hat{b}_1, \hat{a}_1, \hat{a}_2, \hat{b}_2, \hat{c}$ = parameter estimates

The augmented state estimate vector z(N) is then updated using the equation

$$\hat{z}(N)^+ = \hat{z}(N)^- + K(N)\,[d(N) - \hat{d}(N)] \qquad (4.1.7)$$

where K(N) is the Kalman Gain vector given as follows:

$$K(N) = P(N)^-\,H(N)\left[R + H^T(N)\,P(N)^-\,H(N)\right]^{-1} \qquad (4.1.8)$$

where

$$P(N)^- = F(N-1)\,P(N-1)^+\,F^T(N-1) + Q \qquad (4.1.9)$$

$$P(N)^+ = \left(I - K(N)\,H(N)\right)\,P(N)^- \qquad (4.1.10)$$

Since the depth is equal to the measured output in this case, H(N) is given by

$$H(N) = H = [0\ 0\ 0\ 0\ 0\ 1\ 0]^T \qquad (4.1.11)$$

The Jacobian matrix F(N) is given by

$$F(N) = \begin{bmatrix} I & 0 \\ \frac{\partial f(\hat{x})}{\partial \hat{\theta}} & \frac{\partial f(\hat{x})}{\partial \hat{x}} \end{bmatrix} \qquad (4.1.12)$$

where

$$\frac{\partial f(\hat{x})}{\partial \hat{x}} = \begin{bmatrix} 1 & T \\ 0 & \hat{a}_1 + 2\hat{a}_2 |\hat{w}| \end{bmatrix} \qquad (4.1.13)$$

$$\frac{\partial f(\hat{x})}{\partial \hat{\theta}} = \begin{bmatrix} \frac{\partial f(\hat{d})}{\partial \hat{\theta}} \\ \frac{\partial f(\hat{w})}{\partial \hat{\theta}} \end{bmatrix} = \begin{bmatrix} (T_v+\hat{c}) & 0 & 0 & 0 & \hat{b}_1 \\ 0 & \hat{w} & \hat{w}|\hat{w}| & (T_v+\hat{c}) & \hat{b}_2 \end{bmatrix} \qquad 4.1.14)$$

Although the nonlinear drag term, a_2, was not used in the PP and LQG control designs presented in Chapter 3, including it in the system model helps the identification routine correctly estimate some of the other parameters especially a_1. If a linear model were used, the estimation routine would include the effects of the drag in a_1 and therefore, a_1 would no longer be correct. Furthermore, a linear model could not represent the effects of the drag term for all velocities resulting in a time varying a_1. This is not desirable since all the parameter estimates could be degraded if the identification algorithm was not able to track a_1 correctly. Another disadvantage of identifying a time varying system is that the convergence rate of the estimation routine would have to be fast enough to keep up with the changes in a_1 which might cause less accurate parameter (and state) estimation. A similar argument could motivate the inclusion of higher order terms in the identification model, but Figure A3.1 showed that the magnitude of these terms was small for the *Sea Squirt*.

The weights R and Q in equations (4.1.8) and (4.1.9) are used as design parameters instead of noise covariances to obtain the most favorable performance of the extended Kalman Filter for the given application. The output weight, R, was chosen such that the states (depth and depth velocity) were filtered sufficiently to reduce the effects of measurement and quantization noise. If R is made large, the state estimates will be filtered more than if R is made small and therefore, the variance of the final parameter estimates will be less. The drawback of making R too large is that the convergence rate of the Kalman Filter will be reduced. It takes some experimentation to find the value of R which works best for a given application. For the *Sea Squirt* depth dynamics it was found that a range of output weights work with a value of R around 10 giving the best results.

The state weighting matrix, Q, was chosen to be diagonal so that one weight exists for each parameter and state. Making an element of Q large indicates that there is much uncertainty in the associated parameter or state and causes large and quick changes in this parameter during adaptation. A relatively small value of an element of Q indicates confidence in the dynamics and causes the convergence of the associated parameter to be relatively slow with less sudden changes in its value. The magnitude of the elements of Q also have to be adjusted according to the magnitudes of the parameters or states they apply to. For example, the parameters b_1 and b_2 in equations 4.1.1 were significantly smaller than the parameters a_1 and c and therefore, the weights corresponding to b_1 and b_2 were made smaller than those corresponding to a_1 and c. This could be accomplished by scaling parameters b_1 and b_2, which would have had the added advantage of increased numerical accuracy. However, since the variables used in the C code are accurate to more than 20 digits, scaling was not necessary here.

Although the correct choice of the weights improves the performance of the parameter and state estimation, it takes some patience and simulation time to find the weights that work best. For the *Sea Squirt* depth dynamics, the diagonal elements of Q were chosen to be .01, 1, 0.5, 0.1, 1, 1, 1 corresponding to the parameters b_1, a_1, a_2, b_2, c and the states d and w, respectively. These values approximately correspond to the relative magnitudes of the parameters in the corresponding units. It should be noted that the extended Kalman Filter performed well for a wide range of weights. This is important since the weights were optimized for the *Sea Squirt* model and not for the actual plant.

Fast Buoyancy Identification

Since the buoyancy of the vehicle can change significantly from one mission to another or even during a mission (i.e. if the payload is changed), the corresponding parameter estimate, c, also experiences large changes during these transitions. One problem with the identification methods is that the parameter estimates move as a group and, therefore, a poor buoyancy estimate can slow down the convergence of the other parameter estimates. In order to minimize the effects of buoyancy changes on the performance of the adaptive controller, a method was devised that quickly identifies the buoyancy of the vehicle.

The method uses the fact that the thrust needed to maintain the vehicle at a constant depth must be equal to the net buoyancy. Since the vehicle depth and the thrust commands will never remain exactly constant, the thrust commands were averaged over some time period so that even if the depth changed slightly the average thrust would still be close to the buoyancy. It was decided that a 16 second averaging period would be sufficient to obtain a good buoyancy estimate. Clearly, averaging the thrust over a longer period of time would result in a better buoyancy estimate, but it would also require that the vehicle remained at a constant (or nearly constant) depth longer. Averaging the thrust over 16 seconds means that 80 thrust commands will be averaged since the sampling rate is 5 Hz. Since the thrust average will only be equal to the buoyancy of the vehicle if the depth (d_1) and depth rate (w_1) at the beginning of the averaging period is close to the depth (d_2) and depth rate (w_2) at the end of the averaging period, the buoyancy estimate was adjusted to the thrust average only if $(|d_2-d_1| + |w_1+w_2|) < 0.5$. To avoid changing the buoyancy estimate too often, it was only adjusted using this "fast buoyancy identification" (FBI) method whenever the resulting buoyancy estimate change was more than .2 pounds. It should be noted that this "fast buoyancy identification" method only complements and speeds up the estimation of the extended Kalman Filter. Once the FBI occurs, c is optimized by the Kalman Filter.

To demonstrate the effectiveness of this improvement in the identification method, a simulation is presented which shows the parameter convergence with and without FBI under otherwise identical conditions. Figure 4.3a shows the normal parameter convergence while Figure 4.3b shows the parameter convergence with the FBI method. Both simulations were done with identical reference commands (in this case random step changes in the commanded depth). The simulations were done using a six-degree-of-freedom nonlinear model of the *Sea Squirt.* Note that the parameters with the improved buoyancy identification method (Figure 4.3b) converge to approximately the same value as those in Figure 4.3a in half the time. FBI occurs at time t_1 indicated in Figure 4.3b. Note that the time scale of Figure 4.3a is twice that of Figure 4.3b.

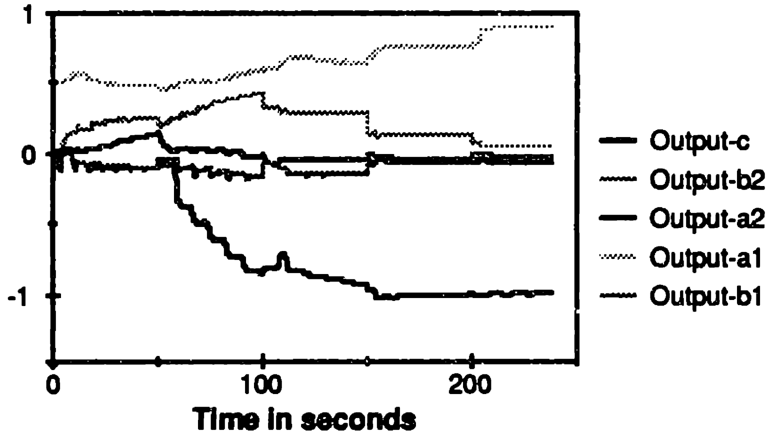


Figure 4.3a: Parameter Convergence Without Fast Buoyancy Id

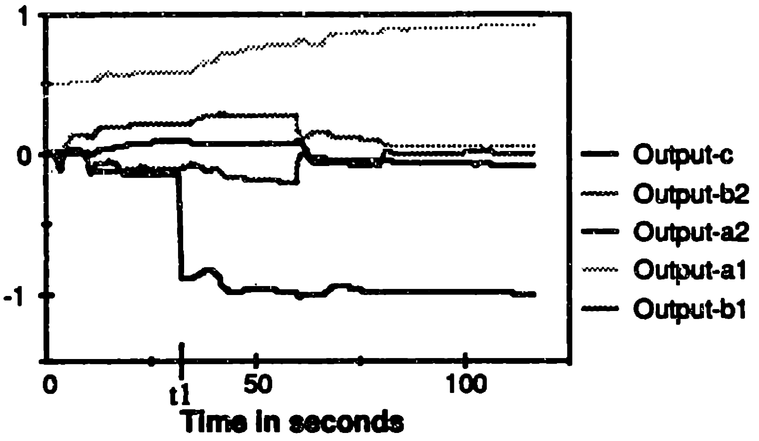


Figure 4.3b: Parameter Convergence with Fast Buoyancy Id

4.2 Simulation Comparison of Control Designs

The controllers that are being compared in this section include the Adaptive Pole-Placement, Adaptive LQG, and Adaptive Sliding Mode. All comparisons done in this section were done using a six-degree-of-freedom nonlinear simulation of the *Sea Squirt*. The extended Kalman Filter was used as the identification method for all simulations and in-water tests to follow. All operating conditions (i.e. initial conditions, noise, disturbances, A/D quantization, reference inputs, etc.) were kept the same for the controllers to make the comparison as fair as possible. The noise added in the simulations to follow was zero-mean gaussian with the variance of the noise being identical for all controllers. Note, however, that the magnitude of the noise at any instance in time is not necessarily the same for all controllers (i.e. only the variance of the noise is the same). To compare the controllers in simulation, the following 5 criterion will be used:

1. Performance without sensor noise after parameter convergence.
2. Performance with noise after parameter convergence
3. Disturbance rejection
4. Initial transient performance while parameter estimates are poor.
5. Ease of implementation (finding weights, computational load, etc.)

All comparisons presented in this Chapter include quantization effects caused by the analog-to-digital converter in order to make the simulations as realistic as possible. Since the controllers have been designed based on the six-degree-of-freedom nonlinear simulation and obtained the desired response time with little or no overshoot, the comparison here will focus on the output variance and thrust variance performance specifications under various operating conditions (e.g. under the influence of noise, disturbances, and parameter uncertainty). The transient behavior of the controllers (i.e. performance with poor parameter estimates) will also be examined since poor transient behavior may limit the utility of the controller.

Section 4.2.1 compares the controllers after parameter convergence. Section 4.2.2 repeats the comparison with sensor noise and disturbances. In Section 4.2.3 the performance of the controllers is compared during the parameter convergence. Finally, Section 4.2.4 compares the controllers based on how easy they are to implement and on the computing power required for each one. A summary of the comparison is given in Section 4.2.5.

4.2.1 Comparison of Controllers After Parameter Convergence

To compare the performance of the controllers after parameter convergence, each simulation was run until the parameter estimates of each simulation converged before starting the comparison. Figure 4.4 shows the response of the 3 adaptive controllers. No noise was added in this simulation so that the best possible performance of each controller could be observed. However, the analog-to-digital quantization of the depth sensor measurement (for this sensor it is .7 inches) was included since it is known a priori and will always be present on the actual vehicle. Since all the adaptive controllers have been designed to meet the same performance specifications, it is not surprising that they all behave in a similar fashion after the parameters have converged. Since large oscillations in the actuators (the vertical thrust in these simulations) would cause unnecessary power consumption, it was thought important to also compare the thrust commands in these simulations.

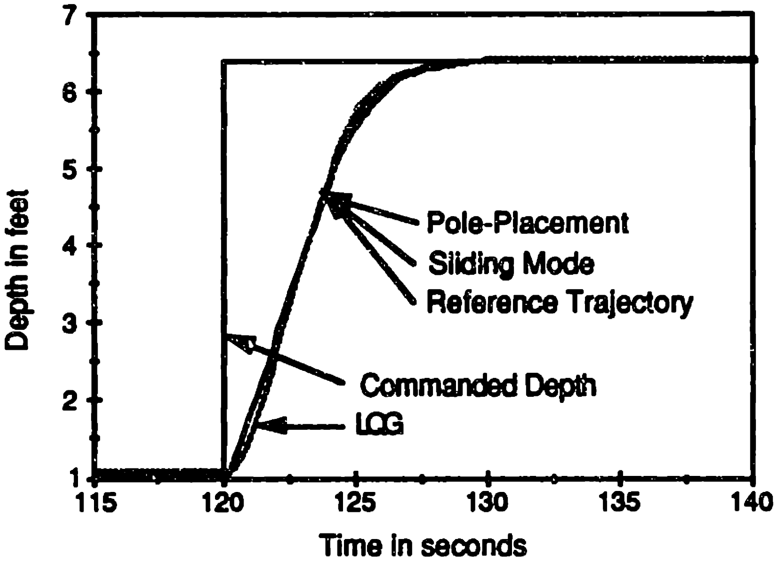


Figure 4.4: Comparison of Controllers After Parameter Convergence (only quantization noise added)

Figure 4.5 shows the vertical thrust for each of the three controllers in response to the commanded depth change in Figure 4.4. In order to demonstrate the benefit of using state estimates, the vertical thrust of the pole-placement controller using the actual states (measured depth and first order difference as depth rate) was also included (Figure 4.5d). Although there is almost no noticeable difference in the control action of the 3 adaptive controllers using the KF state estimates, the vertical thrust control presented in Figure 4.5d shows considerable chattering due to the unfiltered quantization noise in the depth measurement (and therefore, in the depth rate). This demonstrates the effectiveness of the extended Kalman Filter to smooth out the quantization noise in the depth measurement to obtain better depth and depth rate estimates and therefore, smoother control action.

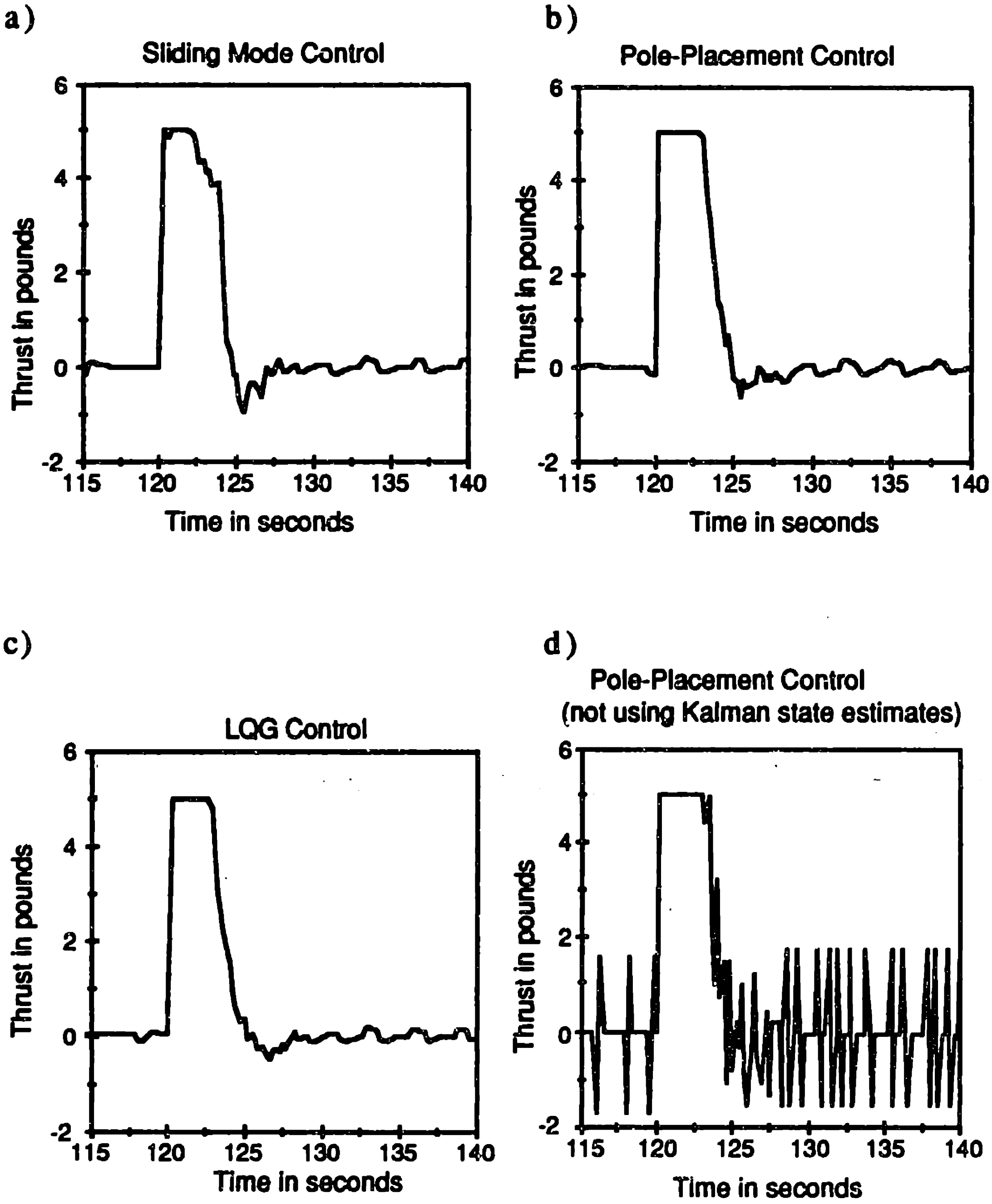


Figure 4.5: Thrust Commands Corresponding to Figure 4.4

4.2.2 Comparison of Controllers with Sensor Noise and Disturbances

Adding Sensor Noise

The simulations in this subsection were done exactly as those in Section 4.2.1 with the exception that sensor noise with a variance of .1 ft was added to the depth measurement. As before, the depth measurement was also quantized (it was quantized after the noise was added, just as it would be in the actual vehicle). Again the parameter estimates were allowed to converge (with the added noise) before the comparison between the controllers was done. Figure 4.6a shows the response of the sliding mode controller to a step change in commanded depth. The reference shown refers to the reference trajectory generated for the sliding mode controller. Figure 4.6b shows the response of the pole-placement and LQG controllers to the same change in commanded depth. Although the performance of the pole-placement and sliding mode is good for the given noise level, the performance of the LQG controller is not as good. As shown in Figure 4.6b the LQG design causes more depth oscillations than the other two methods for the same noise level and, therefore, the performance in terms of low output variance is better for the PP and SM designs than the LQG design. Several simulations were done to confirm that the LQG performance indeed is worse than that of the other controllers for the given noise variance. The thrust commands corresponding to Figures 4.6a and 4.6b are shown in Figure 4.7. The control action of the LQG is more oscillatory than that of the pole-placement and sliding mode controllers making the overall performance of the LQG in the presence of noise worse than that of the other two methods. The SM and PP performance is again almost the same, but the PP actuator commands have better characteristics (less oscillations) as shown in Figure 4.7.

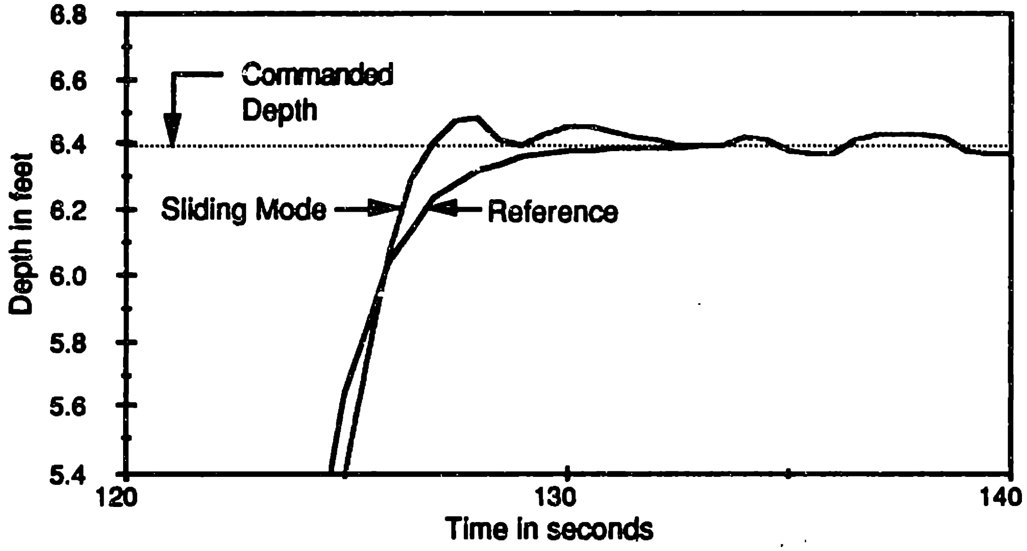


Figure 4.6a: Performance of SM Controllers After Parameter Convergence with Noise Added

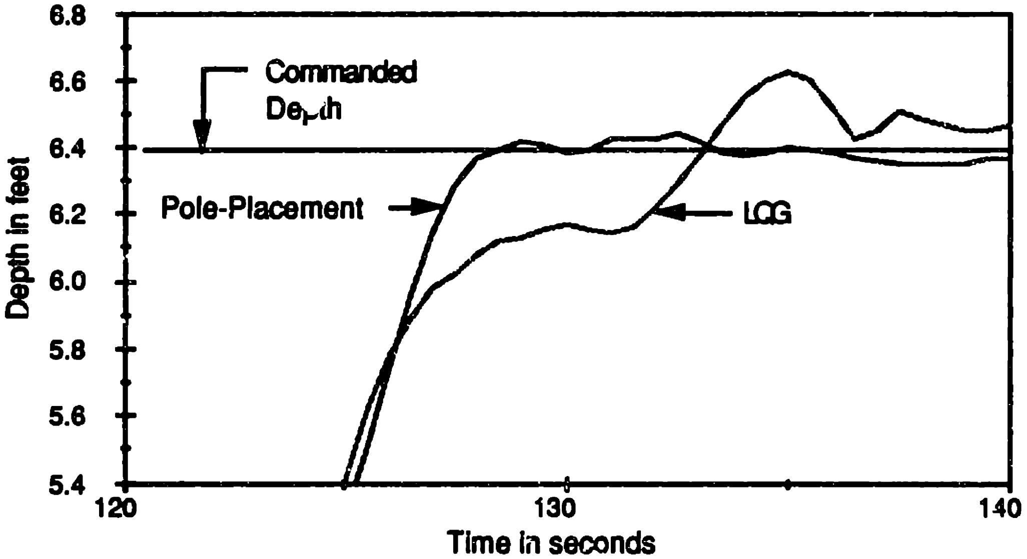


Figure 4.6b: Comparison of PP and LQG Controllers After Parameter Convergence with Noise Added

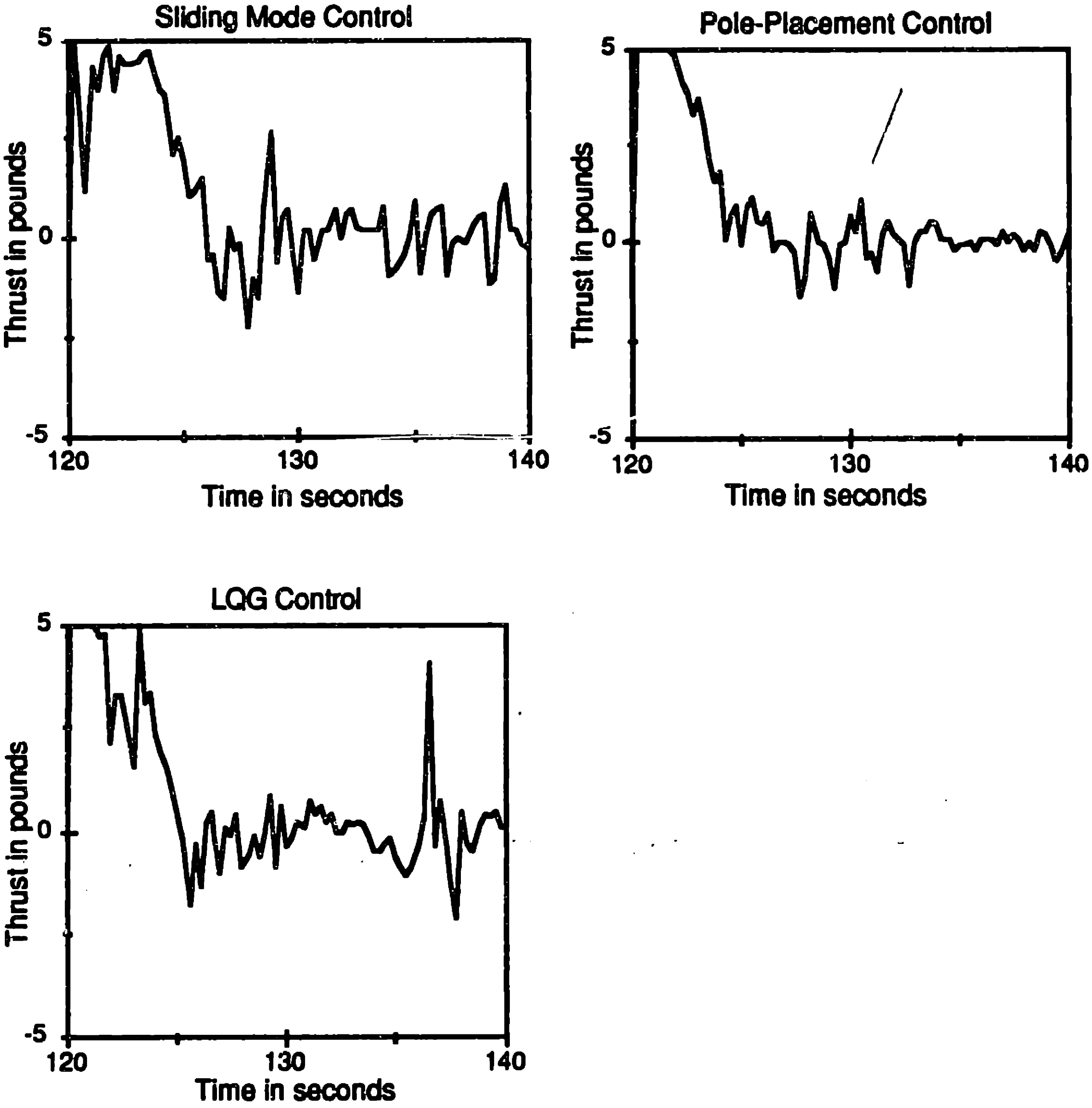


Figure 4.7: Thrust Commands Corresponding to Figures 4.6a and 4.6b

Adding Disturbances

Since the disturbances encountered in an AUV are generally in the form of waves, a sinusoidal disturbance was chosen. In the simulation a sinusoid of amplitude 1 pound was added to the thrust input of all the controllers to check their disturbance rejection capability. Although a wide range of disturbance frequencies were used in simulation, the dynamics of the AUV are slow enough so that only disturbances with considerable wavelength cause oscillations in the depth. In general, all the adaptive controllers exhibited approximately the same disturbance rejection characteristics although the disturbance rejection of the PP and SM controllers was somewhat better than that of the LQG controller. Therefore, the performance of the PP and SM controllers in terms of output variation is better than that of the LQG controller under the influence of disturbances (at least for sinusoidal disturbances). Figure 4.8 shows an example of the disturbance rejection of each of the three controllers for a low frequency sine wave disturbance. The open loop response of the *Sea Squirt* is included to demonstrate the effect a 1 pound sinusoid thrust command would have on the depth if no feedback control was used. The disturbance rejection simulation was done after parameter convergence and with the parameter adaptation stopped. The parameter estimation was stopped by the adaptive controller to avoid adaptation while the control inputs are not sufficiently exciting. The mechanism used in the adaptive controller which determines when adaptation should be stopped is discussed in Chapter 5.

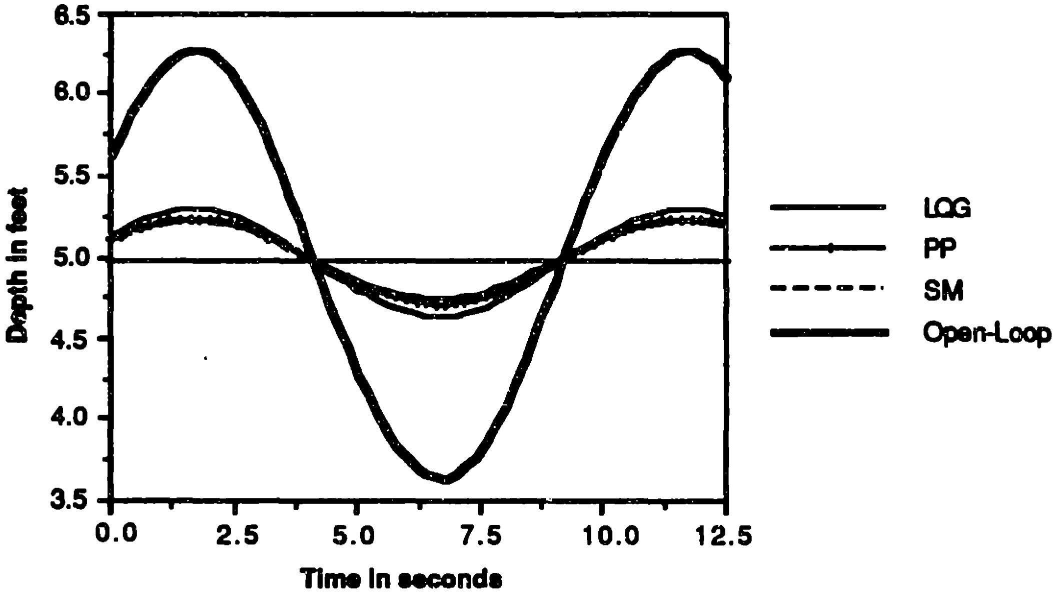


Figure 4.8: Depth Control for a 1 pound Sinusoidal Thrust Disturbance

4.2.3 Comparison of Controllers during Initial Parameter Estimation

Although the adaptive controllers will usually be operating after the parameter estimates have converged, their performance during the initial parameter estimation is also important since poor transient behavior may limit the utility of the algorithm. Therefore, the controller that performs best during this transient has an advantage over the other controllers. However, since the behavior of the adaptive controllers cannot be predicted while the parameter estimates are poor, a safety net approach will be presented in Chapter 5 that guarantees the stability of the closed-loop system during the initial parameter convergence.

In order to compare the performance of the 3 adaptive controllers during the initial parameter convergence, all parameter estimates were initialized to zero except for a_1 which was initialized to .5 (a_1 is known to be between 0 and 1 and therefore was initialized to the midpoint of the two limits). The net buoyancy of the *Sea Squirt* (in simulation) was set to 1 pound so that the buoyancy estimate is also initialized to an incorrect value. Figures 4.9a and 4.9b show the performance of the controllers as the parameter estimates converge to the correct values. A six-degree-of-freedom nonlinear model was used in this simulation as was done in Sections 4.2.1 and 4.2.2. No noise (except for quantization noise) or disturbances were added so that the effects of poor initial parameter estimates on each of the controllers could be compared more easily. The effects of noise and disturbances on the closed-loop system were discussed in Section 4.2.2.

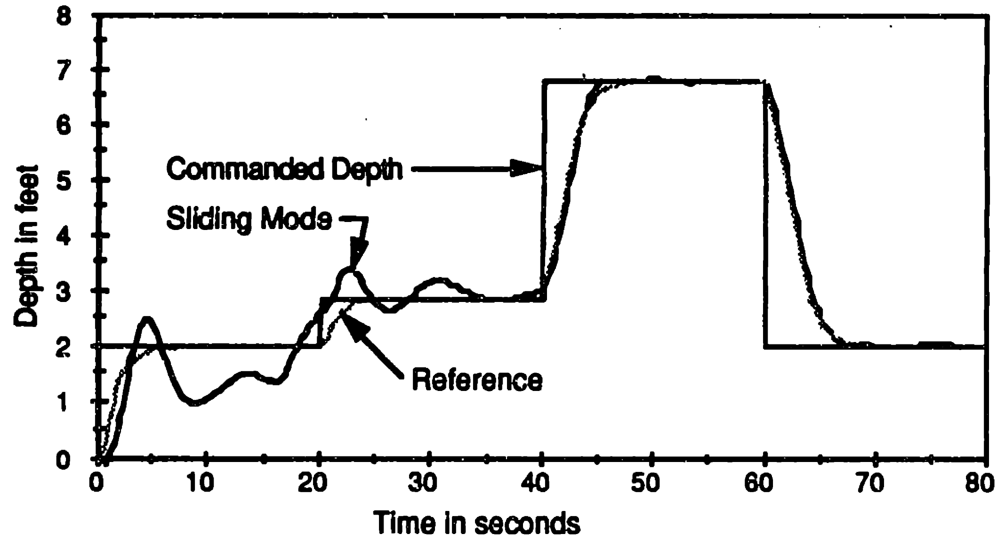


Figure 4.9a: Sliding Mode Controller Performance During Initial Parameter Convergence

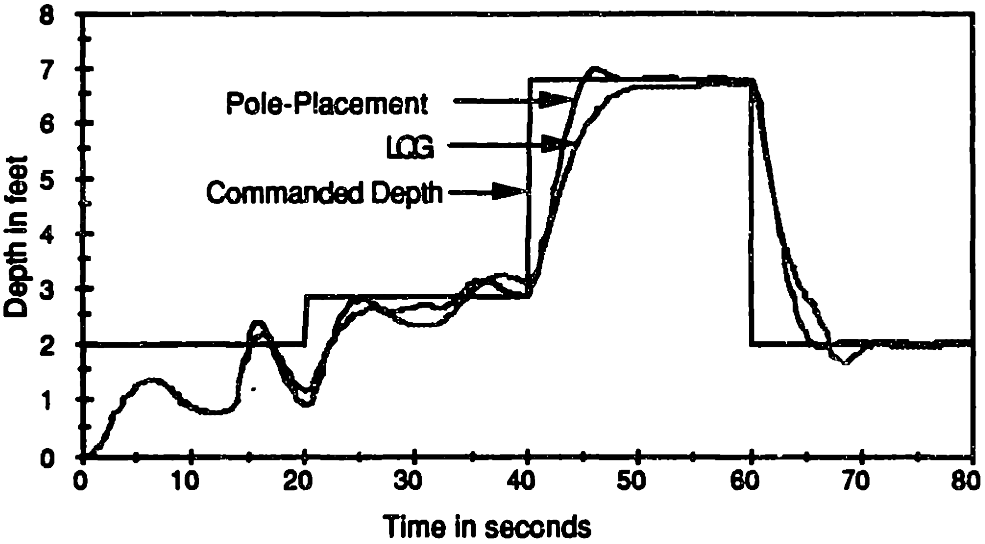


Figure 4.9b: PP and LQG Controller Performance During Initial Parameter Convergence

The performance of all of the adaptive controllers in this simulation is almost the same with the sliding mode performing slightly better than the others. The parameter estimates corresponding to Figures 4.9a and 4.9b are shown in Figures 4.10a-c. The large correction in the buoyancy estimate, c, (at time t1) apparent in these plots demonstrates the effectiveness of the "fast buoyancy identification" (FBI) discussed earlier. The FBI occurs at a different time for the SM controller than for the PP and LQG controllers because the SM controller caused less depth oscillations during the first step command than the other methods and therefore, allowed FBI to obtain a good buoyancy estimate earlier in time. For details on how the FBI was implemented see Section 4.1. It should be noted that the simulation shown in Figures 4.9a and 4.9b was continued (beyond 80 seconds) to make sure that the parameter estimates presented in Figure 4.10a and 4.10b indeed converged to the same values. It should also be noted that the response of the controllers after parameter convergence shown in Figure 4.4 was obtained by continuing the simulation in Figures 4.9a and 4.9b. Only the first 80 seconds of this simulation are shown in the Figures 4.9a and 4.9b so that the details of the initial parameter convergence and the initial transient performance of the controllers can be seen more clearly. Since a second order system was used to represent the model of the six-degree-of-freedom nonlinear simulation, it was not possible to determine the theoretical values of the parameters being identified. However, the fact that the responses shown in Figures 4.9a and 4.9b converge to the reference trajectory (see Figure 4.4) and that all parameter estimates converge to the same values gives some indication that the parameter estimates are converging to the correct values.

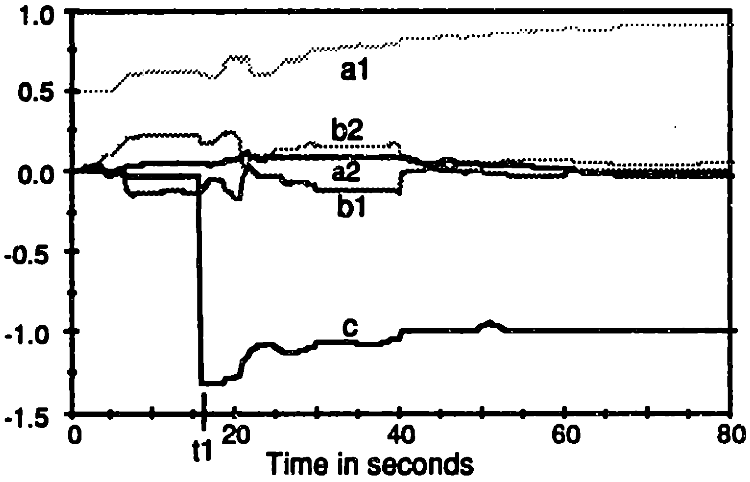


Figure 4.10a: Parameter Estimates Corresponding to the SM Controller

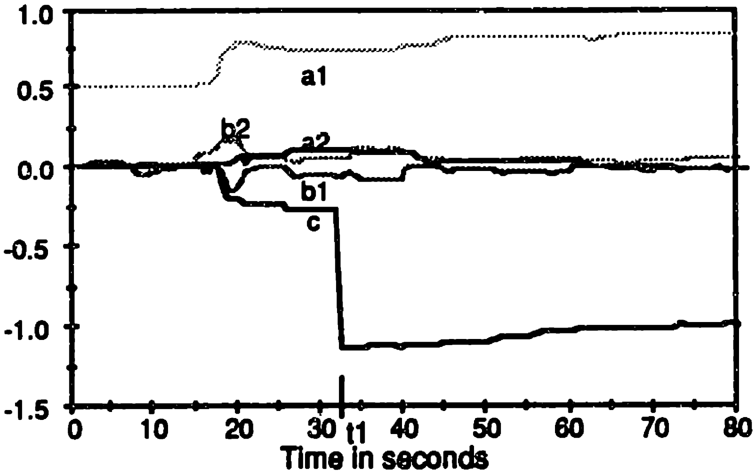


Figure 4.10b: Parameter Estimates Corresponding to the PP Controller

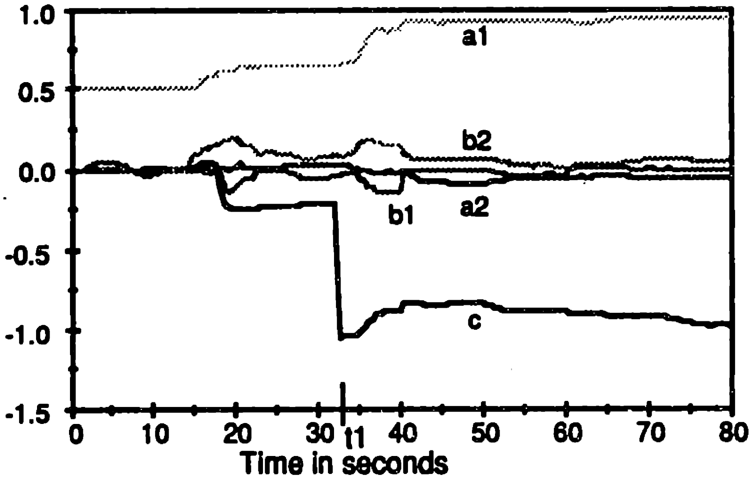


Figure 4.10c: Parameter Estimates Corresponding to the LQG Controller

4.2.4 Comparison of Controllers Based on Implementation Issues

Although the issues discussed in Sections 4.2.1 to 4.2.3 are very important, implementation issues (such as ease of implementation and required computer time) have to be considered as well. For example, it was found that the LQG algorithm uses approximately 10 times more CPU time than the pole-placement or sliding mode algorithms, even if the associated Riccati equation is solved analytically as was done here. The sliding mode and pole-placement algorithms both use approximately the same amount of CPU time.

It was found that choosing the desired closed-loop poles, as is done in pole-placement, is simpler than determining the desired weights, Q and R, in the LQG cost function. Clearly, this finding applies only to the given performance specifications. The sliding mode design requires one to determine the constants μ, λ, η, and Φ (see Section 3.2.3 for the definitions of these constants) in addition to the desired trajectory, while the pole-placement technique requires only the choice of the closed-loop poles. These parameters, although they have physical significance, do not have the clear interpretation in terms of closed-loop behavior that the poles have and therefore, make the SM design more complicated for the application and desired performance specifications given here.

4.2.5 Summary and Conclusion of Off-Line Control Design Comparison

From the comparisons done in Sections 4.2.1 to 4.2.4, it should be clear that the PP and SM designs have advantages over the LQG design. In fact, LQG method does not perform better than either one of the other methods in any of the comparisons done. The overall performance of the pole-placement and sliding mode designs is almost the same. The PP is slightly better in the presence of noise (Section 4.2.2) and the SM performed slightly better during the initial parameter convergence. Even the computational load on the microprocessor is approximately the same. However, one additional advantage of the pole-placement design is that it is simpler to implement. It does not require the generation of a reference trajectory while SM does. Also, SM requires one to choose several parameters related to the sliding surface and boundary layer which is not necessary in the pole-placement design. Overall, the off-line simulations suggest that for the given performance specifications the pole-placement design is slightly better than the sliding mode design with the LQG method being the least desirable.

CHAPTER 5

IMPLEMENTATION ISSUES

In order to implement the indirect adaptive controller developed in this thesis, several safety issues had to be addressed. One important consideration is closed-loop stability. Since the parameter estimates could be poor initially, one has to guarantee boundedness of the output to within some limits of the desired trajectory during the estimation transient. As is discussed in [14], boundedness alone is not sufficient for practical implementation unless the value of the bound is known to be reasonable a priori. In Section 5.1 a safety net approach is presented which guarantees that the actual path of the vehicle will stay within some limits of the desired trajectory. Section 5.2 addresses the problem of identification during periods of insufficient excitation in the presence of noise and/or disturbances. A solution is presented which prevents good parameter estimates from being degraded by stopping adaptation. Since this solution prevents adaptation if the control inputs are not sufficiently exciting, it is not acceptable when the parameter estimates are poor. Section 5.3 presents a method which solves this problem by adding excitation to the control inputs if necessary to ensure sufficient excitation when the parameter estimates are deemed to be poor.

5.1 Transient Safety

The adaptive controller developed in this thesis assumes a very limited knowledge of the plant. Since the initial parameter estimates could be poor, the initial control may be poor as well. If the adaptive controller is to be used immediately[2], stability of the control design must be guaranteed. In fact, some performance criterion should also be met to ensure that the cutput is bounded to within some limits of the desired trajectory. Since the initial behavior of the adaptive controller cannot be anticipated, the robustness will be guaranteed by switching from the adaptive controller to a robust controller if the trajectory of the vehicle differs from the desired trajectory by more than some set bound. This places an envelope around the desired trajectory of the vehicle inside of which the adaptive controller is used and outside of which a robust controller is used. The desired trajectory along with the safety net limits are shown in Figure 5.1 for a typical depth change. The region between the safety net limits, including the reference trajectory, will be referred to as the desired trajectory envelope.

[2] In normal operation the last parameter estimates from one run should be used as the initial estimates for the next run. Therefore, the parameter estimates should only be poor the first time the adaptive controller is used and any time the vehicle dynamics change drastically.

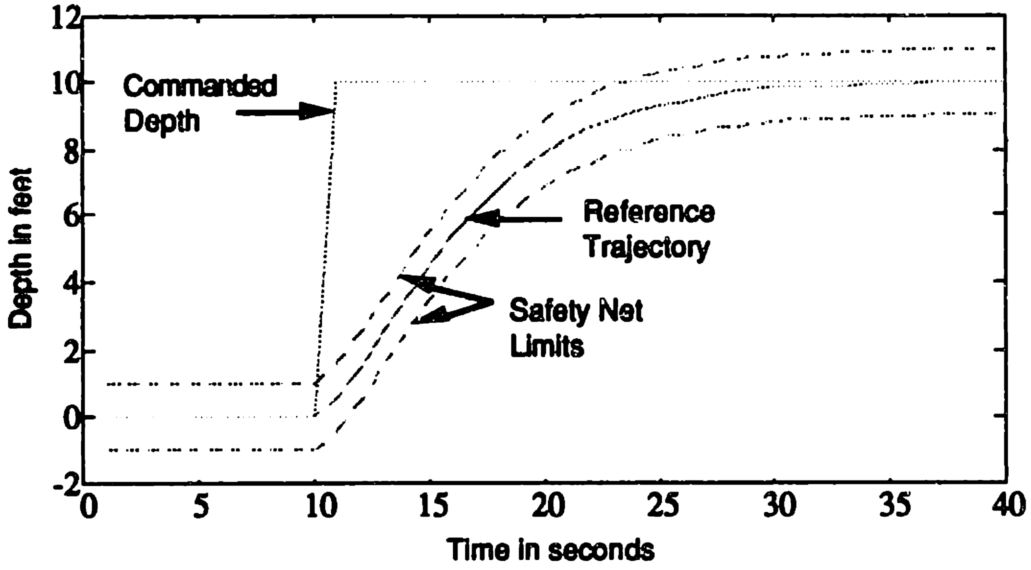


Figure 5.1: Safety Net for Adaptive Controller

It is assumed that a robust controller can be found with the available a priori knowledge of the plant. This is reasonable since the sole purpose of the robust controller is to return the vehicle to the desired trajectory envelope in the case that the adaptive controller should cause the vehicle to leave this envelope (due to poor parameter estimates). For a stable plant such as the *Sea Squirt*, it is easy to find a robust controller with the above property without the aid of an optimal design technique such as LQG or H_∞. However, since the performance of the robust controller depends on the a priori knowledge of the vehicle, the safety net limits have to be matched to the robust controller available. It is also necessary to make the safety net limits large enough to prevent the robust controller from forcing the vehicle into one side of the desired trajectory envelope and out the other in a single time step without ever switching to the adaptive controller. This is important since making the safety net limits in Figure 5.1 too small would cause the robust controller to always control the vehicle even if the parameter estimates are correct.

To clarify the operation of the safety net design presented above, let's use the *Sea Squirt* depth control as an example. Assuming that the initial parameter estimates are poor, the initial control will also be poor with fairly large depth oscillations similar to those shown in Figure 4.8b. If safety net limits shown in Figure 5.1 (1 foot limits) are used, then some of these oscillations might cause the vehicle depth trajectory to leave the desired trajectory envelope. However, as soon as this occurs, the robust controller takes over and forces the vehicle back into the desired trajectory envelope. Since the parameter identification continues no matter which controller (adaptive or robust) is used, the parameter estimates will improve over time. Although control might switch several times between the adaptive and robust controllers initially, as soon as the parameter estimates have improved enough for the adaptive controller to keep the vehicle path within the desired trajectory envelope, the robust controller will not be used again (until the dynamics of the system change).

5.2 Preventing Parameter Drift

As was discussed in Chapter 4, the parameter estimates of the extended Kalman Filter will converge only if the control input (thrust in the case of the *Sea Squirt*) is sufficiently exciting. However, if the input does not change for an extended period of time, parameter drift may occur under certain conditions. With a constant input and no noise or disturbances, the parameter estimates would stay at their original value indefinitely causing no drift since the residual would approach zero. If noise or disturbances are present, the identification routine will try to obtain the mapping from the control input, which is constant (approximately) here, to the output, which is not constant due to the noise and/or disturbances that are present. The result is that the effective signal-to-noise ratio approaches zero which translates to large parameter estimation errors as was shown in Figure 4.1. Therefore, even if the parameter estimates are good, they will drift from their conect values under the above conditions. Since the residual between the measured output and the estimated output is kept small in the extended Kalman Filter, the parameter drift is limited. However, assuming that the parameter estimates are good, it is desirable to prevent any parameter movement when the signal-to-noise ratio is low and therefore, parameter estimation should be stopped in this case. The problem lies in determining when the signal-to-noise ratio is too low. One method is to check only for sufficient excitation and assume that the disturbances and noise are negligible. However, this method would turn adaptation on whenever the cortrol input was sufficiently exciting even if the excitation was caused by large disturbances. The result would be adaptation in the presence of low signal-to-noise ratio which we were trying to avoid. If bounds were known on the disturbances, the approach discussed in [15] could be used to prevent parameter drift. However, since these bounds are not known for the *Sea Squirt* a different method had to be used here. The commanded input was found to be more useful in preventing parameter drift (than the control input) since it gives an indication of the control input excitation without being affected by noise or disturbances. For the *Sea Squirt* it was decided that checking to see if the commanded input (depth or heading) had changed over the last *t* seconds would be sufficient to decide whether or not the control input was sufficiently exciting to prevent parameter drift. The time interval *t* was selected according to the response time of the system and was chosen to be 20 seconds for the *Sea Squirt*. The reasoning behind this method is that it takes about 20 seconds for the *Sea Squirt* to settle to a new depth or heading after a command change and hence, adaptation should continue since valuable input/output information will be available during this time. If the commanded input has not changed in the last, *t*, seconds, it can be assumed that any changes in the control inputs must be due to noise and disturbances and therefore, adaptation should be stopped in this case. Although this method prevents good parameter estimates from drifting, it does not address the problem of what to do if the parameter estimates are poor and the control inputs are not sufficiently exciting. This problem will be addressed in Section 5.3.

5.3 Adding Excitation

Although stopping the parameter estimation whenever the commanded input has not changed in the last *t* seconds would solve the problem of parameter drift due to noise or disturbances, it would also prevent adaptation when it is desired. For example, the vehicle dynamics might change while the commanded input is constant causing poor tracking of the controller[3]. Since the control inputs will probably not be sufficiently exciting in this case, excitation must be added in order to adapt to the new dynamics. One solution to the problem would be to add excitation whenever the commanded input is constant. This would allow parameter estimation even if the commanded input is constant, but it would also reduce the performance of the controller if the parameter estimates are correct. Also, disturbances larger than the added excitation signal would still cause incorrect parameter estimation. An alternative strategy is to add excitation only under special circumstances which are delineated in Table 5.1.

Table 5.1: Conditions for Adaptation and Addition of Excitation

	commanded input changed in last t seconds	commanded input did not change in last t seconds	
		$\lvert$residual avg.$\rvert$ > limit*	$\lvert$residual avg.$\rvert$ < limit *
adapt?	Yes	Yes	No
add excitation?	No	Yes	No

* residual average calculated over a period of p seconds

[3] This is assuming a worst case situation where the dynamics change unexpectedly. If the time of change is known (i.e. payload deployment) then it could be used to add excitation and change the covariance P at the correct instance in time.

As Table 5.1 indicates, adaptation will remain on if the commanded input has changed in the last t seconds and no excitation will be added. Since no parameter drift will occur as long as there is excitation, this condition is reasonable as long as the command changes are greater than the noise level or any disturbances that might be present. As discussed in Section 5.2, the AUV will have some time constant that determines the response time to command changes and therefore, the adaptation will be left on for t seconds after a command change to allow continuous parameter adaptation during normal operation with sufficient excitation.

If no command change has occurred in the last t seconds, adaptation may or may not be on depending on the residual average. The residual average is the average of the difference between the measured output and the estimated output over some time period, p. Since the absolute value of the residual average gives a measure of parameter correctness, comparing it to a predetermined limit allows one to add excitation and start adaptation if the parameter estimates are thought to be inaccurate and to stop adaptation otherwise. Again, if excitation is added and adaptation is started, it will be continued for at least t seconds for the same reason as discussed above. It is assumed that if a parameter estimate is incorrect, it will cause a bias in the residual and hence, cause the residual average to be nonzero. The residual average was used instead of the residual itself in the hope that some zero mean disturbances in the output (such as waves in the case of AUV's) which affect the residual would be averaged out and therefore, would not cause adaptation to occur. Clearly, if adaptation is allowed to occur during significant disturbances, the parameter estimates will be degraded. In simulation the above method effectively distinguished parameter errors from zero mean disturbances and hence, allowed one to adapt in the former case and not in the latter. Even if the disturbances are not zero mean (or have a period longer than p) and cause adaptation to occur, the added excitation will improve the chances of correct parameter adaptation.

For the *Sea Squirt* the parameters t and p were both chosen to be 20 seconds and the average residual *limit* in Table 5.1 was chosen to be .05 ft. Although this value might seem small, it should be noted that the residual is averaged over 20 seconds or 100 time steps and therefore, most of the noise and disturbances will be averaged out. The numerical value for the residual limit should be chosen such that the absolute value of the residual average will always be less than the limit when the parameter estimates are correct, but it should exceed the limit if the controller cannot maintain the desired trajectory due to incorrect parameter estimates. Some experimentation will be necessary to find the most favorable average residual limit.

Deciding What Type of Excitation Should be Added

Although we have determined when excitation should be added and when not (see Table 5.1), we still need to decide what type of excitation should be added and how it should be added. In order to bound the output, it was decided that any excitation should be added to the commanded input and not the control input. This enables one to place a bound on the output since the amplitude of the added excitation will be known a priori. Since the control inputs will respond to changes in the commanded input, adequate excitation can be achieved by ensuring that the commanded input changes sufficiently. This was done by adding excitation in the form of a square wave with "randomly" changing period. The amplitude of the square wave was fixed. The period was changed randomly within some limits based on the time constant of the system dynamics. For the *Sea Squirt* depth dynamics the limits for the period were chosen to be 1 and 10 seconds and the amplitude was chosen to be +/- 1 ft. An example of the excitation added by the controller to the commanded input is shown in Figure 6.11.

CHAPTER 6
ON-LINE CONTROL IMPLEMENTATION

In Section 6.1 the three adaptive controllers (SM, PP, and LQG) that were compared via simulation in Section 4.2 are compared via in-water tests. The on-line comparison will be similar to the off-line comparison in Chapter 4.

In Section 6.2 a heading controller is presented for the *Sea Squirt* using the control design which performed best in the depth control comparisons. All implementation issues involved in the heading control are discussed. Autonomous in-water tests are presented to demonstrate the operation of the heading and depth controller at sea.

6.1 In-Water Depth Control Comparison

To minimize the differences in the operating conditions of the three controllers (SM, PP, LQG) being compared, they were tested on the same day one right after the other. However, it should be noted that disturbances and sensor noise cannot be controlled and therefore, will be slightly different for each controller. One major source of disturbances was the fact that these tests were performed with the tether connected (to avoid the possibility of losing the vehicle in the Charles River). Not only does the tether significantly change the drag of the vehicle depending on how far it is stretched, it also exerts an upward force on the aft section of the vehicle because floats are attached to the tether every 5 to 20 feet to make it slightly buoyant. Although the AUV is capable of pulling the floats underwater, the tether changes the apparent buoyancy of the vehicle as a function of depth and as a function of how far the tether is stretched. With the tether attached to the aft section of the vehicle, it causes the pitch dynamics to be dependent on depth. In order to minimize the effects of the tether (especially those caused by stretching the tether to its limit), the depth control tests in this section were done with the horizontal thrusters off. However, due to water currents and surface wind acting on the tether, the actual vehicle position (tether effects) were different for each comparison. These considerations should be kept in mind when looking at the control design comparisons presented in this section. All in-water comparisons were done using the extended Kalman Filter with the safety features discussed in Chapter 5 implemented.

6.1.1 Comparison of Controllers After Parameter Convergence

One basis for comparison is the depth response to a step change in commanded input after parameter convergence; thus, the parameter estimates were allowed to converge to their final values for this first comparison. To ensure adequate excitation, 10 different step changes in depth were commanded over an 8 minute time period. The same 10 waypoints were used for each of the three controllers (SM, PP, and LQG) and the last step change in depth was selected for the comparison shown in Figure 6.1.

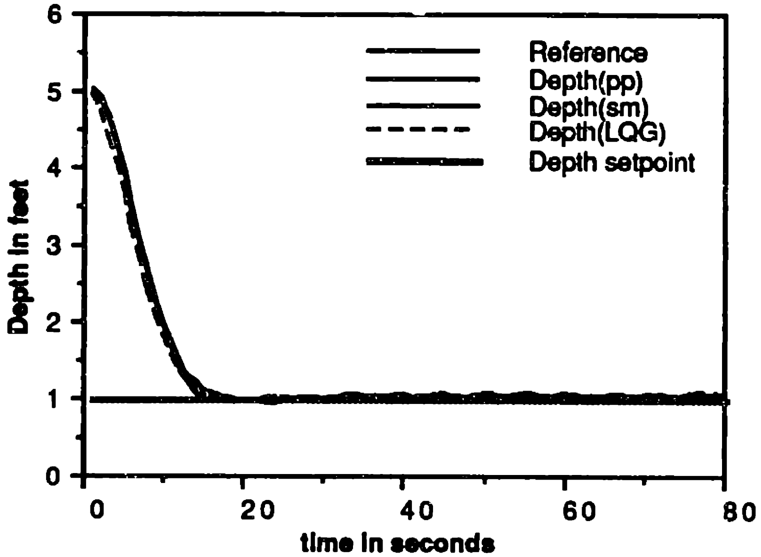
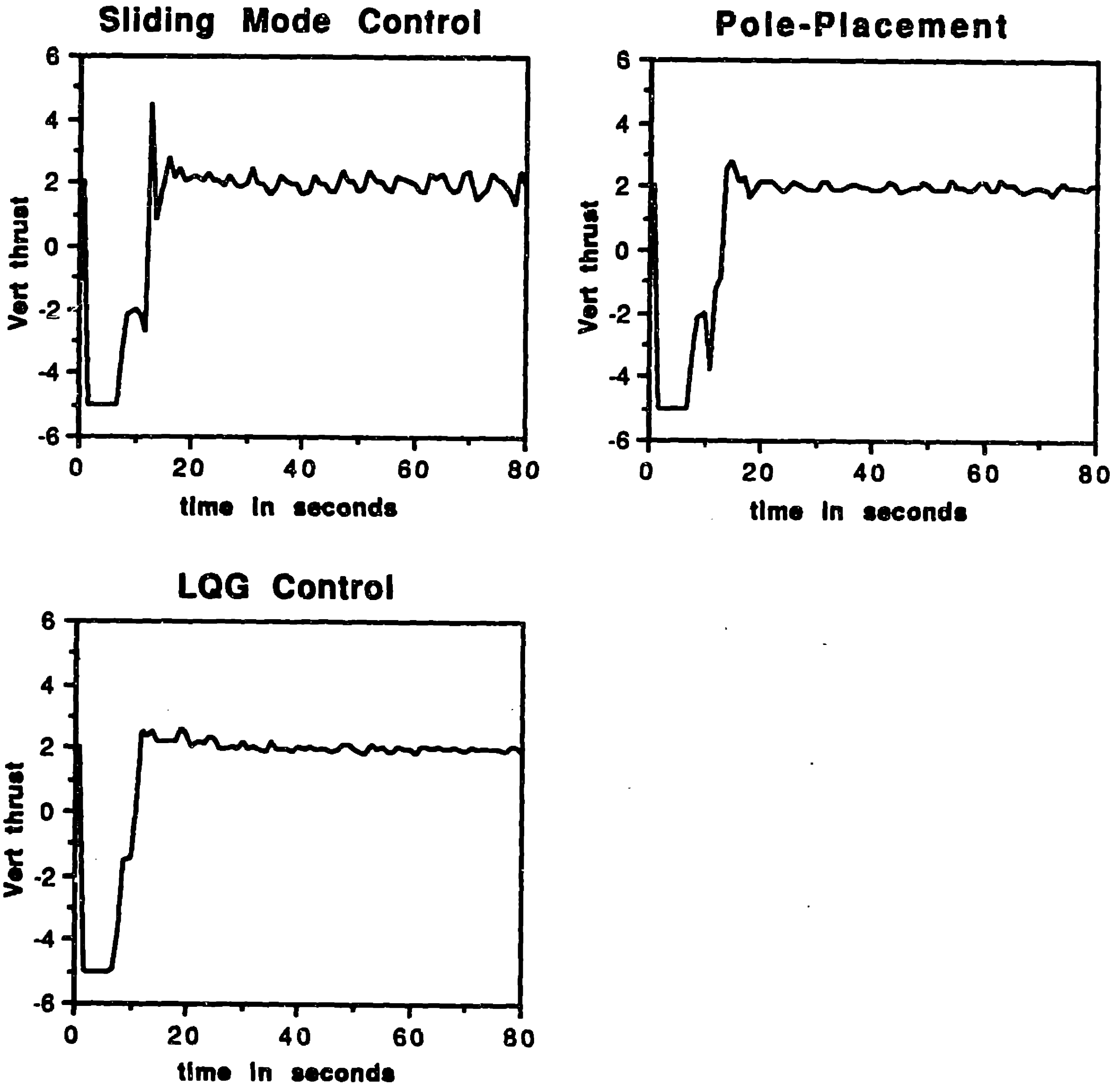


Figure 6.1: In-Water Comparison of Controllers After Parameter Convergence

The plot in Figure 6.1 shows the response of the three adaptive controllers (SM, PP, LQG) as the commanded depth is changed from 5 feet to 1 foot. All the controllers performed almost identically in this test as the simulation in Section 4.2.1 predicted. The thrust commands corresponding to Figure 6.1 are shown in Figure 6.2.

Again, the three controllers behave in a similar fashion and compare well to the off-line simulation in Figure 4.3. It should be noted that the thrusts shown in Figure 6.2 settle to 2 pounds instead of zero even though the vehicle's net buoyancy was only about 0.25 pounds. The 1.75 pound difference was due to a miscalibration of the vertical thruster forward setpoint (see Figure 2.2) which was off by about 1.8 pounds. The reverse setpoint was close to the correct value. Since the adaptive controllers incorporated the 2 pound offset into the buoyancy term, c, the overall performance was not significantly degraded. This demonstrates one of the advantages of the adaptive controller over conventional non-adaptive designs which would not have been able to correct for the miscalibrated thruster.


Sliding Mode Control

Pole-Placement

LQG Control

Figure 6.2: Thrust Commands Corresponding to Figure 6.1

It should be noted that all data obtained through in-water tests was logged only every second, not every sampling period. This means that some high frequency behavior might be unobservable. Since the depth and yaw accelerations are fairly slow for the *Sea*

Squirt, the depth and heading data should accurately reflect the vehicles behavior. However, the thrust commands can change significantly from one sampling point to the other and hence, some nigh frequency changes might be unobservable in the commanded thrust plots.

6.1.2 Comparison of controllers during Initial Parameter Estimation

In order to compare the controllers during the initial parameter convergence, all parameter estimates were set to zero except for a_1 which was initialized to 0.5 as was done in the off-line comparison in Section 4.2.3. The feature discussed in Section 5.3, which allows the controllers to add excitation to the commanded input (if necessary), was disabled for this test to ensure that the commanded input would be the same for all 3 controllers. Although disabling this feature makes the test less realistic and probably slowed the parameter identification, it was the only way to make a fair comparison between the controllers during the initial parameter convergence. Another test will be presented in Section 6.2.2 to demonstrates that the adaptive controller correctly adds excitation if necessary.

Figure 6.3a through 6.3c show the performance of the pole-placement, sliding mode, and LQG controllers, respectively. As can be seen from the figures the response of each of the controllers improves as the parameter estimates improve. The safety net limits were set to 1 foot for this test as shown in Figure 5.1. Only the sliding mode controller actually reached this limit at approximately 25 seconds as shown in Figure 6.3b. Both the PP and LQG controllers stayed within the desired trajectory envelope (i.e. within 1 foot of the desired trajectory) throughout the test. Surprisingly, the SM controller shows more oscillations than the other two controllers with the pole-placement oscillating the least during the initial parameter estimation. However, the SM recovers quickly from these early oscillations and performs well after the second depth command change. The steady state tracking error apparent in Figures 6.3a through 6.3c during the first depth command is eliminated at time t_1 when the buoyancy estimate is adjusted using the "fast buoyancy identification" developed in Chapter 4.

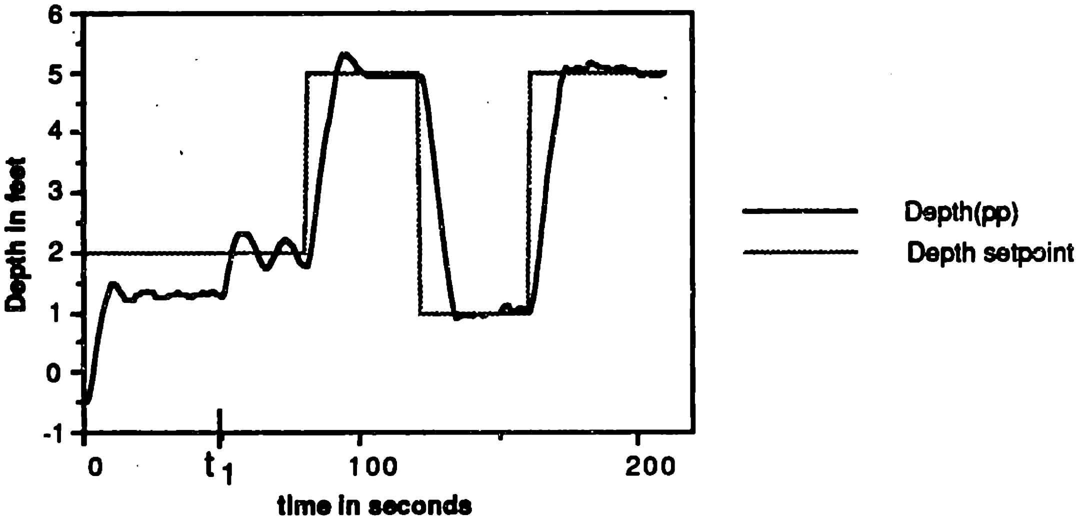


Figure 6.3a: PP Controller Performance During Initial Parameter Convergence

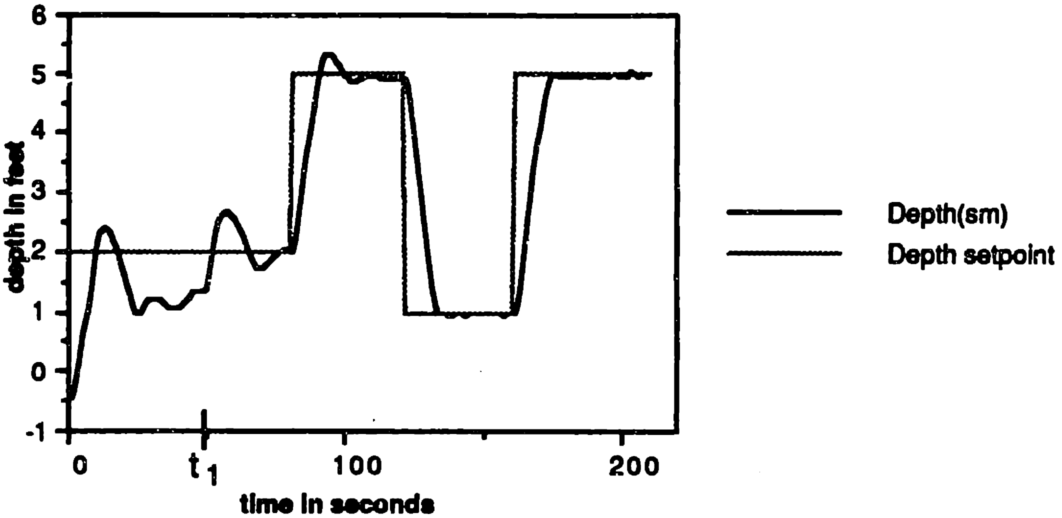


Figure 6.3b: SM Controller Performance During Initial Parameter Convergence

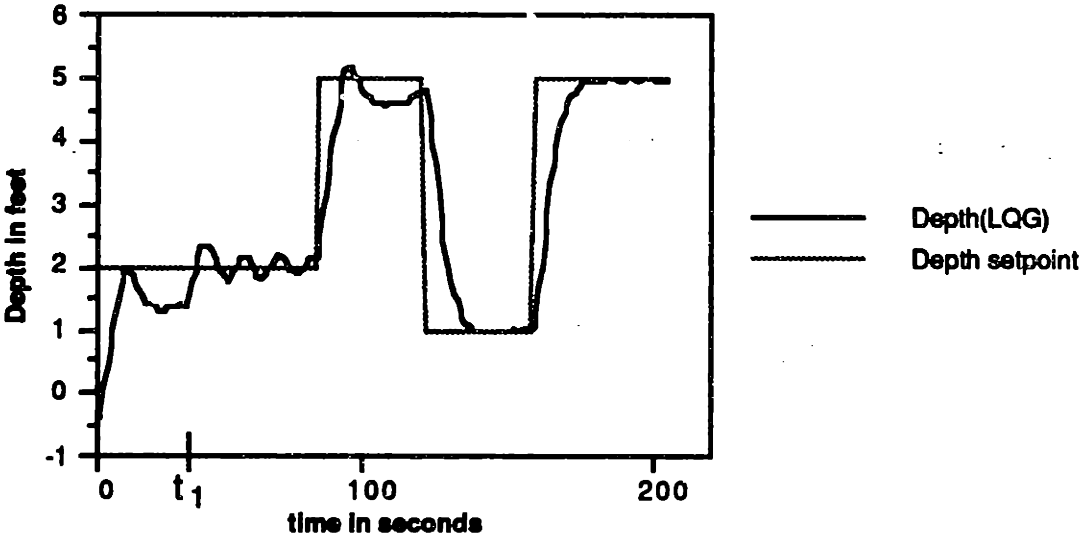


Figure 6.3c: LQG Controller Performance During Initial Parameter Convergence

Figures 6.4a-c show the parameter estimates corresponding to Figures 6.3a-c, respectively. The large correction in the buoyancy estimate at time t_1 demonstrates the effectiveness of the "fast buoyancy identification" discussed in Chapter 4. The changes in the buoyancy estimate after this initial correction are in part due to the floats attached to the tether. Every time the depth was increased, more tether was pulled underwater and with it more of the floats were submerged causing changes in the net upward force experienced by the vehicle. The changes do not always occur at the same depth because the tether was not pulled down vertically, but rather at some angle depending on how much the tether was stretched. Figures 6.4a-c also show that parameter estimation is turned off (flat regions) during periods where the commanded depth does not change for an extended period of time (20 seconds). It should be noted that the parameter estimates were not at their final value at the end of the 200 second test run, but were close enough to show the initial "learning" period of the adaptive controllers.

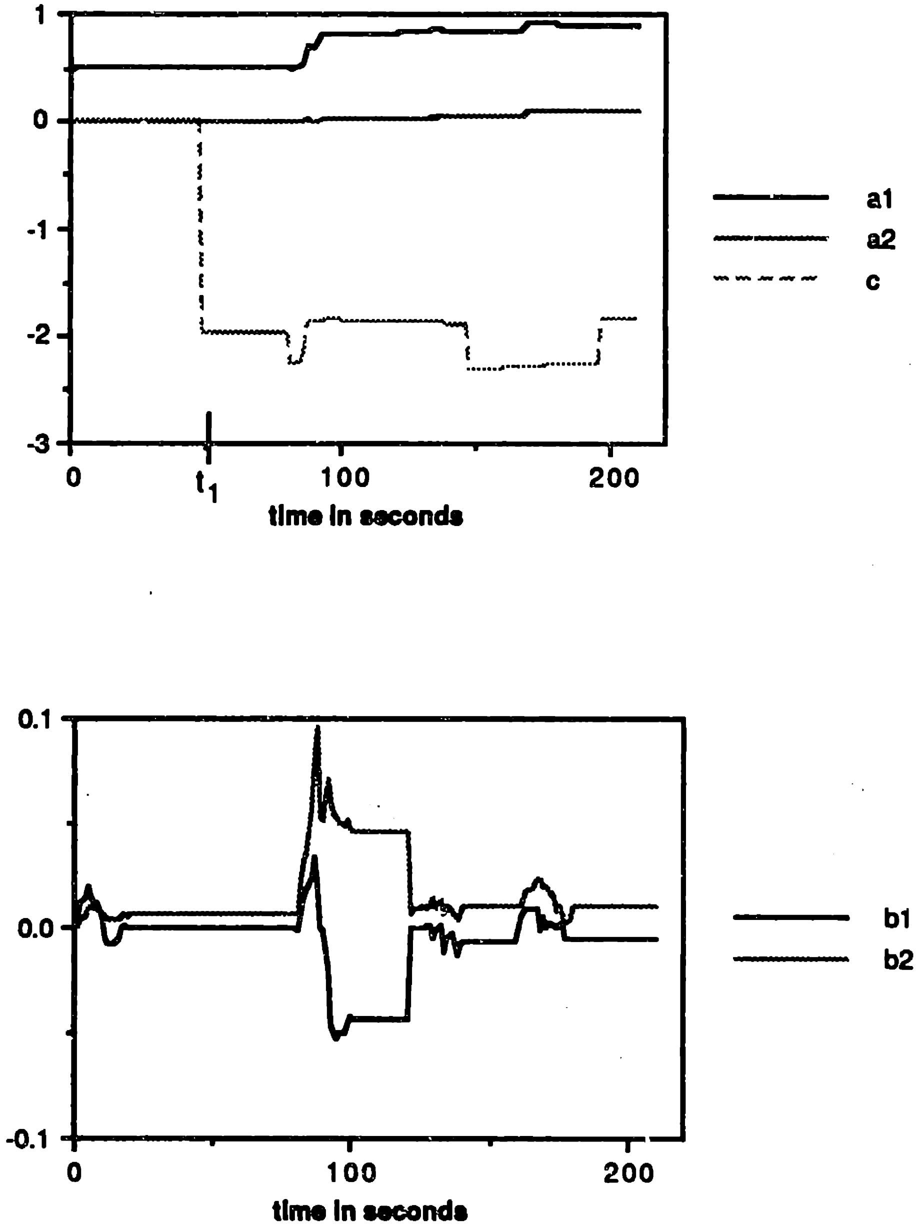


Figure 6.4a: Parameter Estimates Corresponding to the PP Controller

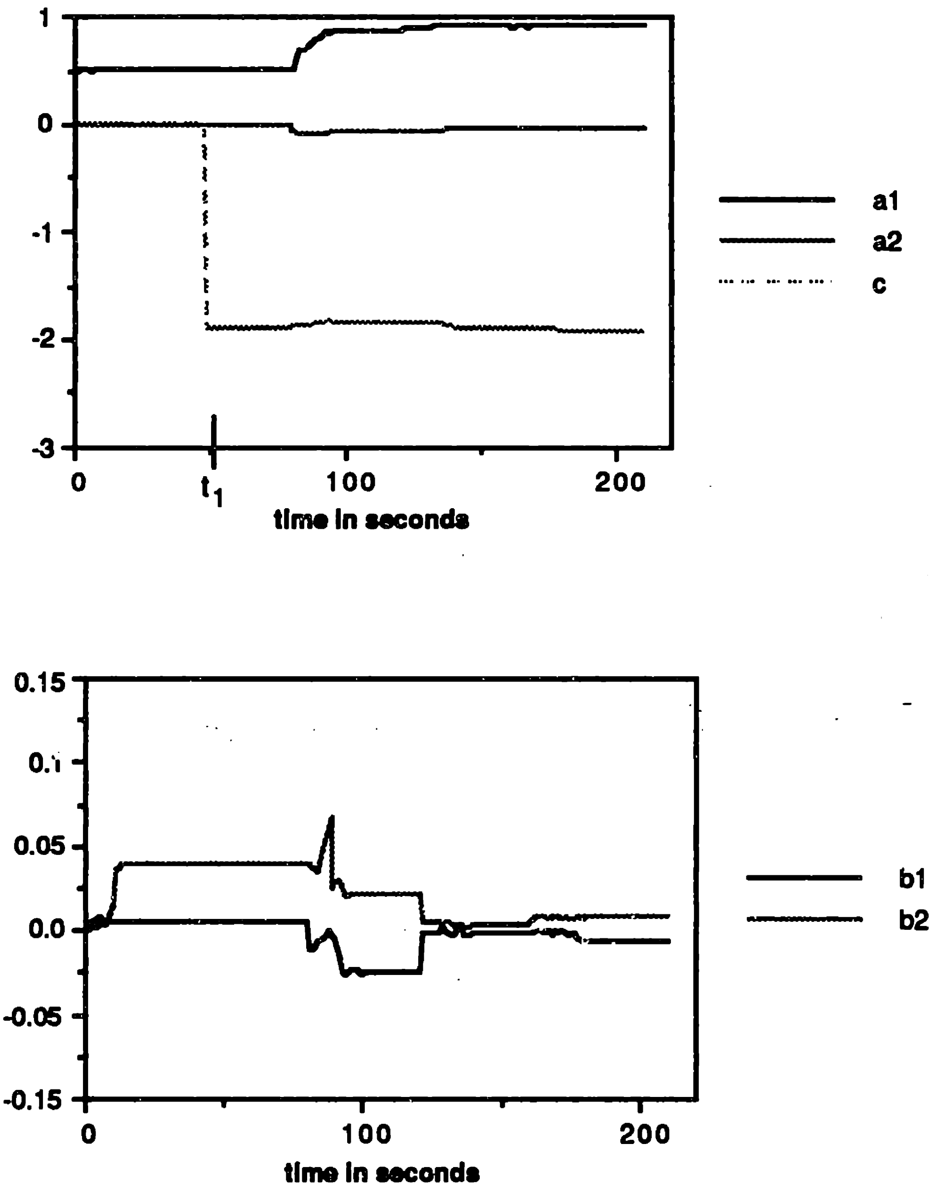


Figure 6.4b: Parameter Estimates Corresponding to the SM Controller

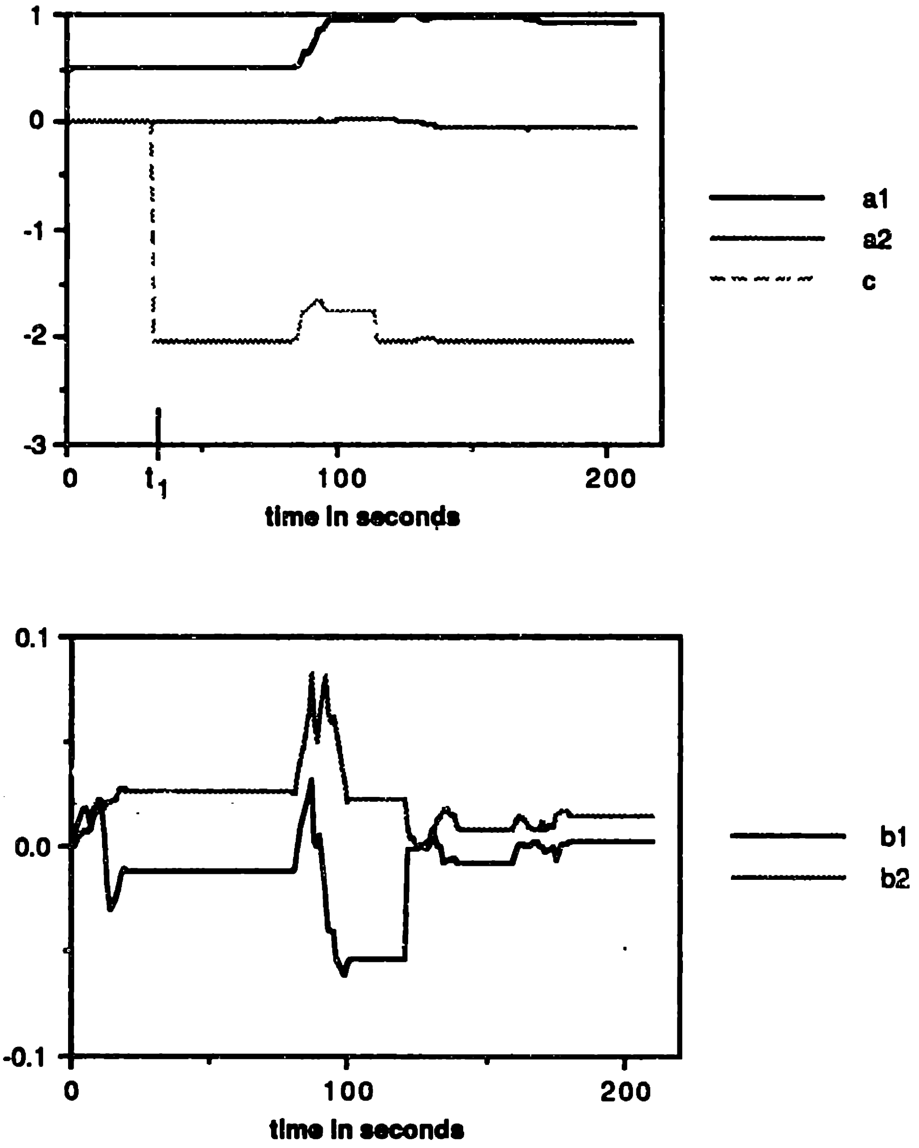


Figure 6.4c: Parameter Estimates Corresponding to the LQG Controller

Altogether, the three adaptive controllers performed comparably and no one was clearly superior to the others. They all achieved the desired response as shown in Figure 6.1 after parameter convergence, and performed nearly the same during the initial parameter learning. However, the SM controller did leave the desired trajectory envelope briefly while the PP and LQG controllers did not.

6.1.3 Conclusions of Depth Control Comparison

In the in-water depth control comparison done in the previous sections, all controllers performed approximately equally after parameter convergence as was expected from the simulations done in Chapter 4. During the initial parameter convergence period, the SM controller caused more depth oscillations than the other two controllers even though it performed better in the corresponding off-line simulation in Section 4.2.3.

Although the SM and PP controllers both performed equally well in the on-line and off-line comparisons, the PP design was chosen as the preferred method since it was simpler in terms of implementation with no apparent loss in performance. The LQG design was rejected because it is difficult to implement in an on-line fashion (as discussed in Chapter 4) and its performance was at best equal to that of the PP design. It should be noted that all the controller comparisons were done using the *Sea Squirt* or a model of the *Sea Squirt* and that the outcome of the comparison may be different for other plants.

6.2 Adaptive Heading Control

To demonstrate a complete closed-loop depth and heading adaptive control system for the *Sea Squirt*, a heading controller for the vehicle was developed using the adaptive pole-placement technique. The development of the heading controller is presented in the following section using the same general design. All the steps involved in developing the heading controller will be discussed in Section 6.2.1.

The operation of both the heading and depth controllers is presented in a full-scale autonomous test in Section 6.2.2. Several runs will be presented which show the effectiveness of the adaptive controllers under different operating conditions.

6.2.1 Heading Control Design

As discussed in Chapter 2, the heading dynamics of the *Sea Squirt* are very similar to the depth dynamics so that the same discrete time model can be used with depth measurements replaced by heading measurements (yaw). To distinguish the heading and depth parameters, an *h* will be appended to the heading parameters as shown in the following equation.

The simplified discrete time heading dynamics are given by:

$$\Psi(N+1) = \Psi(N) + T\,r(N) + hb_1\,(T_p - T_s + hc) \qquad (6.2.1a)$$

$$r(N+1) = ha_1\,r(N) + ha_2\,r(N)\,|r(N)| + hb_2\,(T_p - T_s + hc) \qquad (6.2.1b)$$

where

Ψ = yaw
r = yaw rate
T_p = port thrust
T_s = starboard thrust
T = sampling period (.2 seconds)
hb_1, ha_1, ha_2, hb_2, hc = parameters to be identified

Although there are two thrusters (port and starboard) which control heading, only their difference, ΔT, is important in the heading control so that a SISO system results. The added degree of freedom will be used in open-loop speed control of the vehicle. The parameter *hc* shown in equation 6.2.1 was included to accommodate any constant torque applied to the vehicle (due to differences in the port and starboard thrusters or cross flow currents) which might otherwise cause steady state heading errors and identification problems. If the two horizontal thrusters are mechanically identical and are calibrated correctly, *hc* should be zero assuming that there are no water currents.

Since the form of the heading dynamics model is identical to that of the depth dynamics with T_v replaced by ΔT, the PP control law given in Section 3.2.1 still holds so that

$$\Delta T(N) = K_1 \, (\mathrm{ref}(N) - \hat{\Psi}(N)) - K_2 \, \hat{r}(N) - hc \qquad (6.2.2)$$

where K_1 and K_2 are the state feedback gains obtained from the on-line pole-placement algorithm using the heading parameters. As was done in the depth control, the Kalman state estimates were used in the control law instead of the actual values and the thrust command,

ΔT, was limited to 5 pounds to avoid actuator saturation. Although a yaw rate sensor was available and could have been used in the Kalman filter, it was not used because it may generate rate biases after a certain operating time. Although these biases could have been eliminated, this would have added unnecessary complexity to the code since the controller proved to work well without the rate measurement.

The port and starboard thrust commands were obtained from the differential thrust given in equation 6.2.2 as follows:

$$T_p = \Delta T/2 + \text{speed} \qquad (6.2.3a)$$

$$T_s = -\Delta T/2 + \text{speed} \qquad (6.2.3b)$$

where *speed* is the desired net forward force of the two thrusters which determines the forward velocity of the vehicle. For example, if 50% of the maximum forward velocity is desired, *speed* would be set to 50% of the maximum thrust or 2.5 pounds (the maximum thrust limit per thruster was set to 5 pounds) assuming linearity. Since the speed sensor of the vehicle was not currently operational and therefore no feedback could be used to control speed, the speed achieved by the vehicle was dependent on the drag of the vehicle and the battery condition. This means that the vehicle will be slower tethered than untethered even if the same speed is commanded.

Although the speed and heading control will be independent in the ideal case, they couple when actuator saturation is considered. For example, if a large percentage of the maximum speed is desired and the thruster commands are limited to 5 pounds, a heading change could easily result in the saturation of one of thrusters causing either the thrust differential ΔT to be less than the value commanded by the controller, or the forward speed to be less than the desired value. Since this problem cannot be avoided, it was decided to give the heading control priority over the speed control. This means that the speed offset of the thrust command will be reduced as much as necessary to obtain the desired ΔT. Whether to give heading control priority over speed control or vice versa depends on the application at hand. With the vehicle being used as a testbed for controllers, either choice would have been acceptable as long as the identification algorithm uses the appropriate thrusts in the parameter estimation.

One problem that arises in heading control and not in depth control is that the heading error is bounded. Since a heading of 180 degrees and -180 degrees are identical, the heading error can never be more than 180 degrees (assuming that only the heading itself is important and not the number of turns made by the vehicle). It was assumed that the controller should always turn in the direction closest to the desired heading. For example, if the current heading is 170 degrees and the commanded heading is -170 degrees, then the desired path would be to turn clockwise. This means that the heading will change sign at 180 degrees. To ensure that the heading error seen by the controller is always between -180 and 180 degrees, 360 degrees are added to the error if it is less than -180 degrees and 360 degrees are subtracted from the error if it is more than 180 degrees. The same idea also applies to the residual (heading - estimated heading) calculated in the Kalman Filter and to the safety net controller discussed in Chapter 5.

6.2.2 In-Water Autonomous Depth and Heading Control of the *Sea Squirt*

With the depth and heading control design complete, it is now possible to test the vehicle in an autonomous (untethered) mission. One advantage of running the vehicle untethered instead of tethered is that the vehicle is no longer influenced by the tether dynamics. This improves the performance of the adaptive controller, in terms of tracking and parameter identification, because none of the tether dynamics were included in the vehicle model. Another advantage of operating untethered is that simultaneous heading and depth control can be done with significant forward speed. This was not possible in tethered mode since the tether is only about 80 feet long which means that it would run out after only 40 seconds at a speed of 2 ft/sec. Furthermore, the normal operating mode of the vehicle is untethered and therefore, the controller must work in this mode.

To demonstrate the operation of all the features of the adaptive controller developed in this thesis several autonomous tests had to be done. All the test runs presented here were done with speeds ranging from 2 ft/s to 4 ft/s (this corresponds to 30 to 60 percent of maximum speed) since this is the normal operating speed range of the vehicle. All speeds given in this chapter were approximated using an experimental positioning system developed for the *Sea Squirt*. Although the software for position control was not finished in time for the autonomous tests done here, the positioning system provided an effective way of determining the approximate speed of the vehicle for a given forward thrust command. Note, however, that the speed was determined from a tethered test and therefore, the speeds indicated for the untethered control tests are only approximate.

The first autonomous test run was done to demonstrate parameter convergence and the adaptive controller performance during parameter convergence. Figure 6.5 shows the adaptive pole-placement heading control starting with poor initial parameter estimates. To make the test more realistic, heading and depth control was done simultaneously with a forward speed of 2 ft/sec.

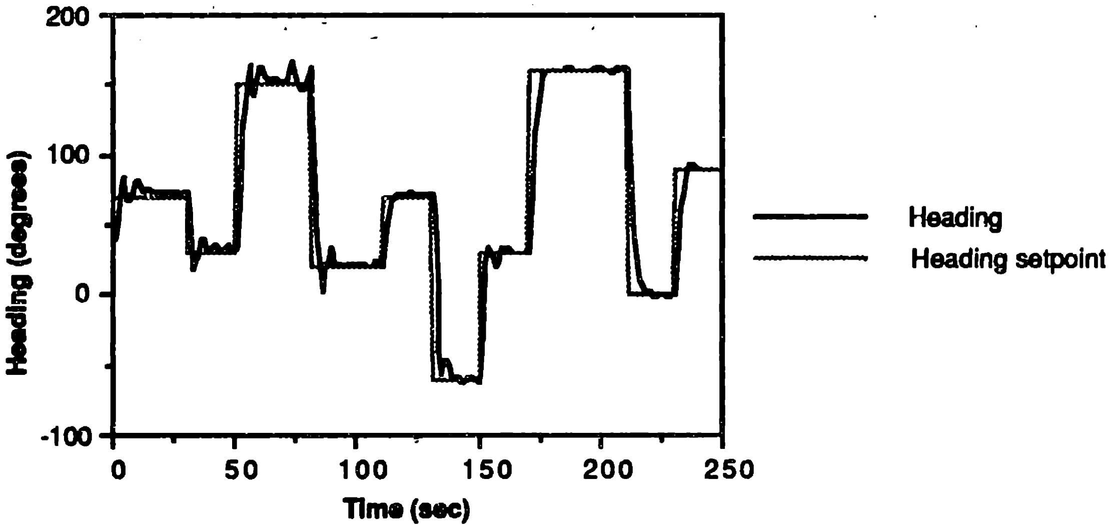


Figure 6.5: Autonomous Pole-Placement Heading Control

As can be seen from Figure 6.5, the response to step changes in the heading setpoint improves significantly over time with the overshoots and oscillations being almost eliminated at the end of the run. Also, the steady-state error apparent in the first few step changes (Figure 6.5) is eliminated. This error was due to a miscalibration of one or both of the horizontal thrusters which caused them to produce different thrusts for the same commanded thrust. As will be shown later, the *hc* parameter was adjusted correctly by the extended Kalman Filter, thereby eliminating the steady-state error.

Figure 6.6 shows the port and starboard thrust commands corresponding to Figure 6.5. As the control response in Figure 6.5 improves over time, so does the control action shown in Figure 6.6. The thrust oscillations are minimal during the last 100 seconds of the run. Note that the starboard thrust settles to about 1.75 pounds while the port thrust settles to about 1.25 pounds for constant heading commands. This means that the two thrusters were miscalibrated by a total of about .5 pounds.

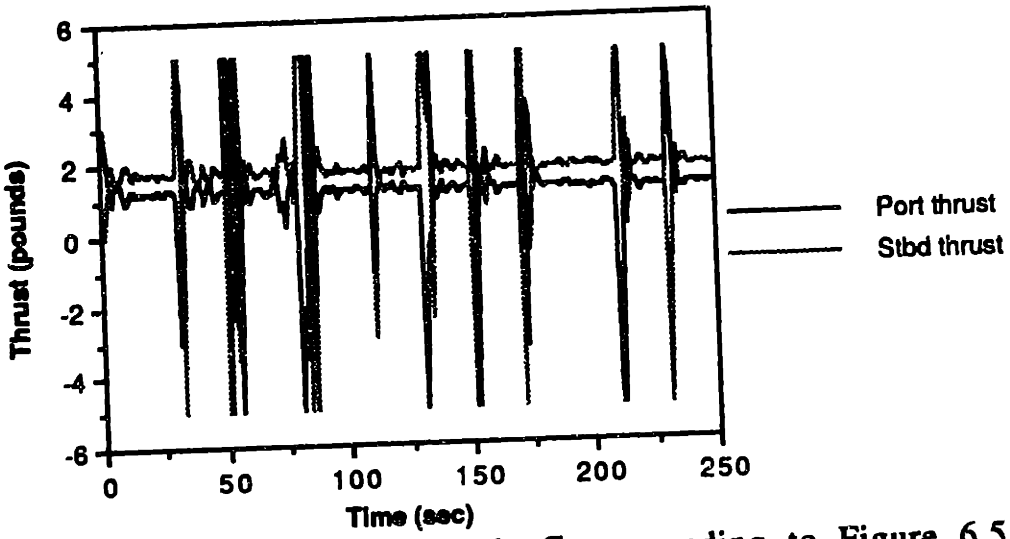


Figure 6.6: Thrust Commands Corresponding to Figure 6.5

Figure 6.7a and 6.7b show the parameter estimates for the heading control shown in Figure 6.5. The figure shows that *hc* converges to a value close to .5 pounds which correctly accounts for the miscalibration in the thrusters. The other parameter estimates shown in Figures 6.7a and 6.7b also converge to reasonable values. Although the actual values for these parameters are not known, the improvement (over time) in the heading response indicates that the parameter estimates must be converging to the correct values.

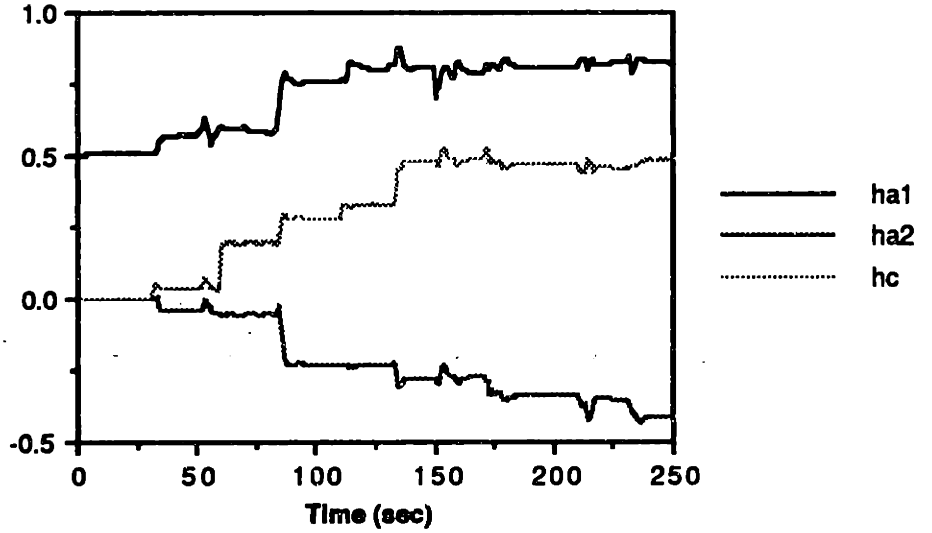


Figure 6.7a: Heading parameters Corresponding to Figure 6.5

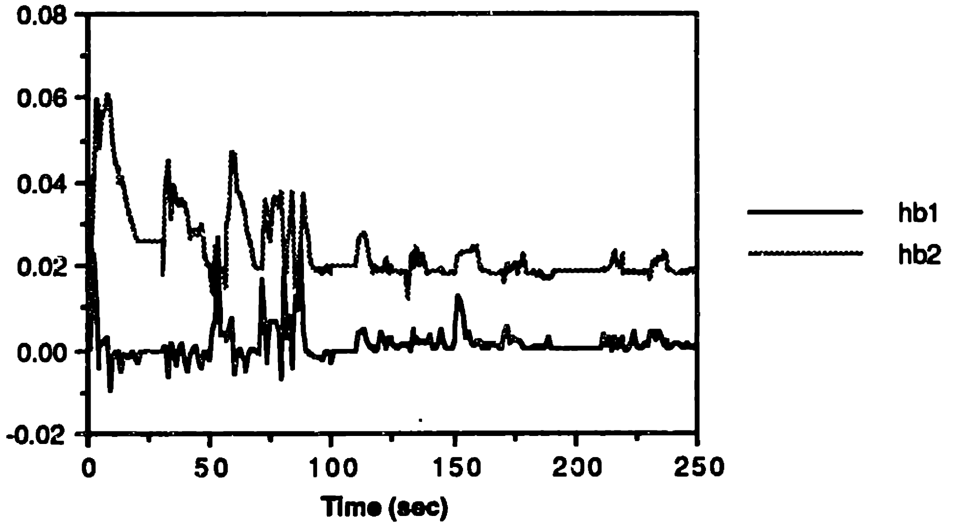


Figure 6.7b: Heading parameters Corresponding to Figure 6.5

Figure 6.8 shows the depth response corresponding to the heading control presented in Figure 6.5. Again, the controller was initialized with poor parameter estimates.

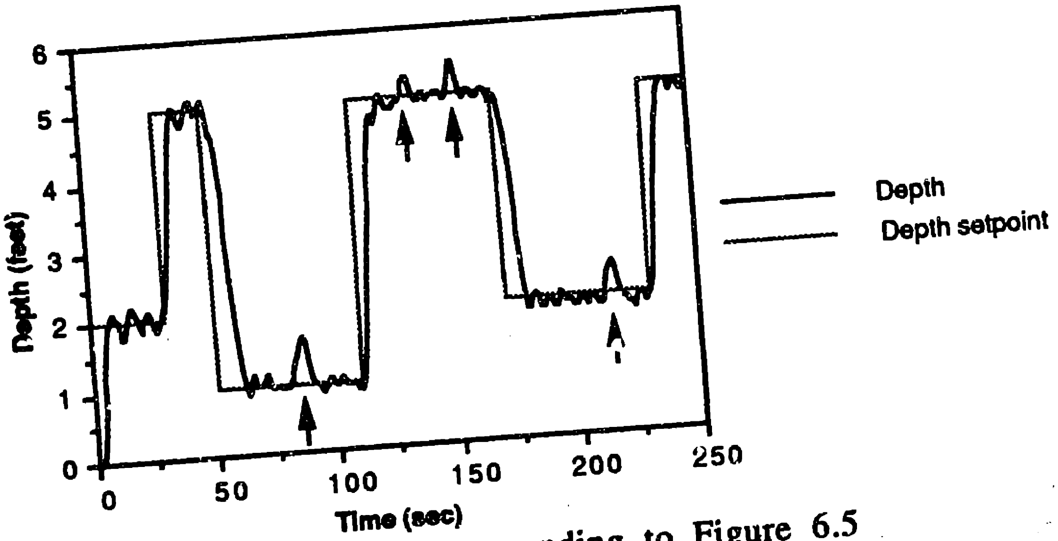


Figure 6.8: Depth Control Corresponding to Figure 6.5

The arrows shown in Figure 6.8 indicate the points in time where heading changes occurred. Although the heading and depth dynamics were assumed to be decoupled, some coupling clearly does occur during fast heading changes. The fact that the depth always increased during turns indicates that the vehicle pitched downward which caused the horizontal thrusters to exert a downward force on the vehicle. To reduce the effects of the heading changes on the depth control, the heading controller bandwidth could be reduced so that the vehicle would turn slower. The heading changes shown in Figure 6.5 were done at turn rates of up to 50 deg/sec. The oscillations around the desired depth setpoint shown in Figure 6.8 which are not caused by heading changes are significantly reduced after the parameter estimates converged to the correct values. Some small oscillations persist and are partially due to the large quantization step size of .7 inches and the non-zero forward speed which induces non-zero pitch angles. Note that the safety net feature discussed in Chapter 5 is never used as the vehicle path shown in Figure 6.8 never leaves the desired trajectory envelope. The thrust commands corresponding to the depth control of Figure 6.8 are shown in Figure 6.9.

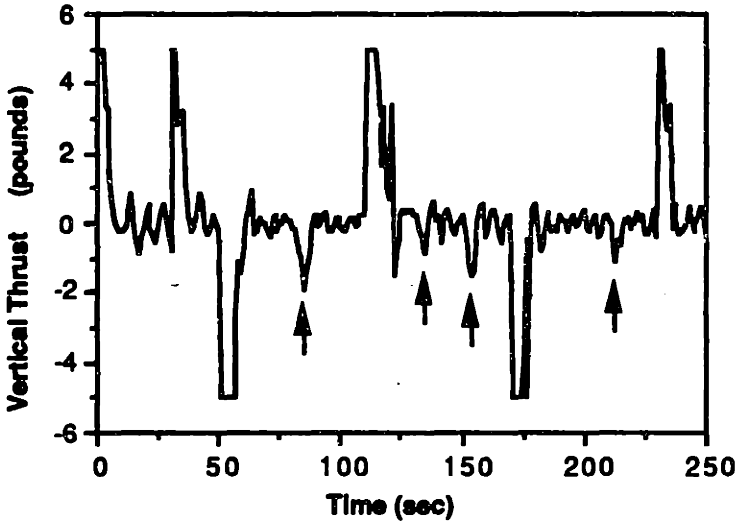


Figure 6.9: Thrust Commands Corresponding to Figure 6.8

The arrows in Figure 6.9 show the thrust commands in response to the depth disturbances caused by the heading changes. During constant depth control the thrust commands settle to within about .3 pounds towards the end of the run (i.e. after parameter convergence). The buoyancy of the vehicle was very close to zero for all the autonomous tests and therefore, the thrust values shown in Figure 6.9 settle to a value close to zero when a constant depth is being maintained.

The parameter estimates corresponding to the depth control shown in Figure 6.8 are shown in Figure 6.10a and 6.10b. As expected the buoyancy term, c, settles to a value close to zero. All the other parameters also settle to reasonable values (the actual parameter values are not known). Again, several runs were done to confirm that the parameter estimates settle to the same values independent of their initial value.

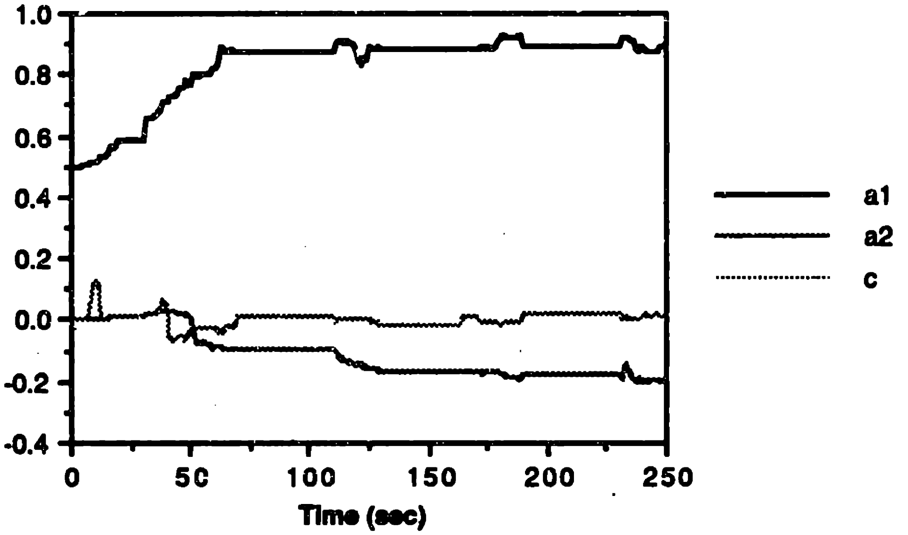


Figure 6.10a: Parameter Estimates Corresponding to Figure 6.8

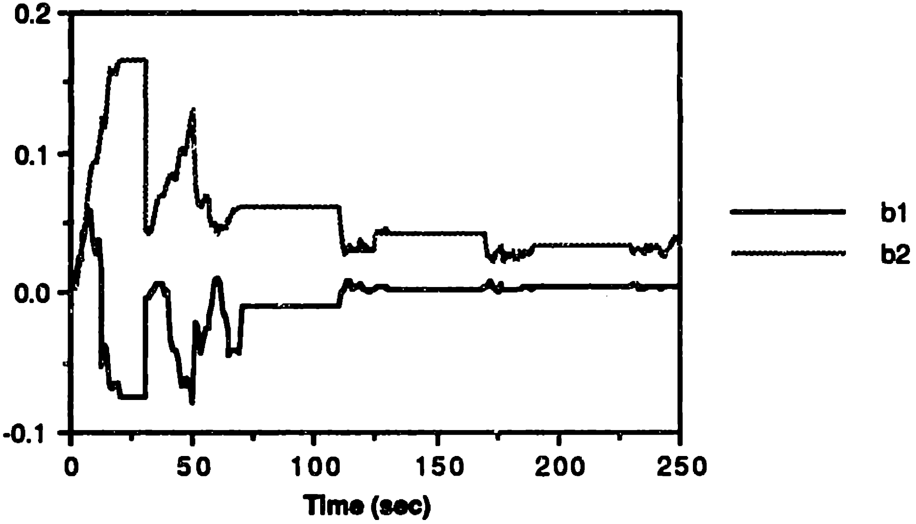


Figure 6.10b: Parameter Estimates Corresponding to Figure 6.8

Adaptation Without Sufficiently Exciting Command Inputs

To demonstrate that the feature discussed in Section 5.3, which ensures adequate excitation if the parameter estimates are poor, works on the vehicle, the following in-water test was done. The parameter estimates for the depth controller were initialized to good initial values (obtained from previous tests) and a constant depth command was given (a speed of 2 ft/sec was used) and the vehicle and all commands were allowed to reach steady-state. Since no excitation will be added by the controller unless the parameter estimates are incorrect, it was necessary to either change the hydrodynamics of the vehicle or change the parameter estimates during the test. It would have been very difficult to actually change the hydrodynamics of the AUV during a run and, therefore, the parameter estimates were changed instead. By changing the parameter estimates to incorrect values (b1=0, a1=0.5, a2=0, b2=0.01, c=-1) at a time of 150 seconds (see Figure 6.11), the adaptive controller has to re-adapt in order to maintain the desired depth. However, since the commanded input is constant and has been constant for some time the excitation will not be adequate for adaptation; the controller has to add excitation and re-start adaptation (remember that adaptation is stopped if the commanded input has not changed in the last 20 seconds to avoid parameter drift as discussed in Section 5.2). Note that the buoyancy estimate, c, was changed to -1 which directly affects the vertical thrust command and therefore, causes a steady-state depth error. Figure 6.11 shows the depth and commanded depth of the vehicle and clearly shows the degraded performance of the controller when the parameter estimates are changed (at a time of 150 seconds). The arrows in Figure 6.11 indicate the depth changes caused by heading command changes. The figure shows that the controller maintains the depth of the vehicle to within 1 to 2 inches, except for the times when the heading changes occurred, during the first 150 seconds and during the last 40 seconds of the run (i.e. when the parameter estimates were close to the actual values). As was desired the controller correctly added excitation to the depth command to allow adaptation when it was necessary. The time delay between when the parameter estimates were changed and when the excitation was added is due to the fact that the residual average was used instead of the residual itself in determining when excitation should be added (i.e. it takes some time for the average to exceed the threshold even though individual points may). The mechanism used to determine when excitation should be added by the controller is discussed in Chapter 5.

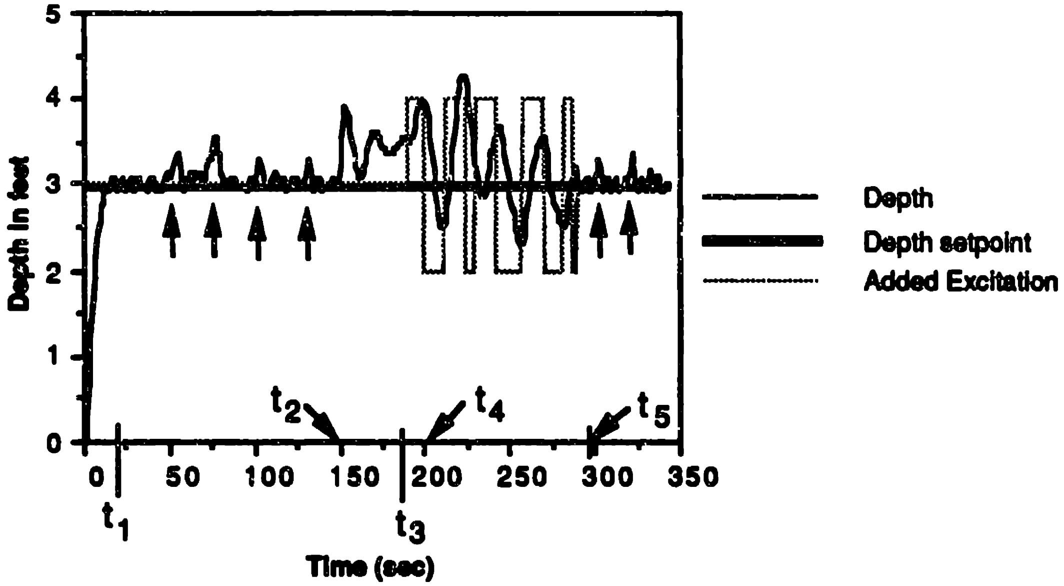


Figure 6.11: Depth Control With Parameter Change at 150 seconds

Figures 6.12a and 6.12b show the parameter estimates corresponding to Figure 6.11 and indicates that adaptation was turned off (flat region) at time t_1 (i.e. 20 seconds after the last command change). At time t_2 the parameter estimates were changed to the incorrect values causing poor tracking of the controller (see Figure 6.11). At time t_3 the residual average exceeded the threshold (given in Section 5.3) indicating that the parameter estimates are incorrect. Since excitation was not adequate, the controller added excitation and re-started adaptation. As shown in Figure 6.12a, the "fast buoyancy identification" (discussed in Chapter 4) re-adjusts the buoyancy estimate, c, to nearly the correct value at time t_4. At time t_5 the parameter estimates had converged sufficiently close to the actual values (i.e. the absolute value of the residual average was below the threshold) and adaptation and excitation was stopped. An indication that the parameter estimates were nearly correct at time t_5 is given by the fact that the tracking performance of the controller shown in Figure 6.11 is approximately the same after t_5 as it was before the parameter estimates were changed at time t_2. This demonstrates that the adaptive controller correctly adds excitation to the commanded input to ensure that the estimated depth and thereby, the parameter estimates, remain close to their actual values.

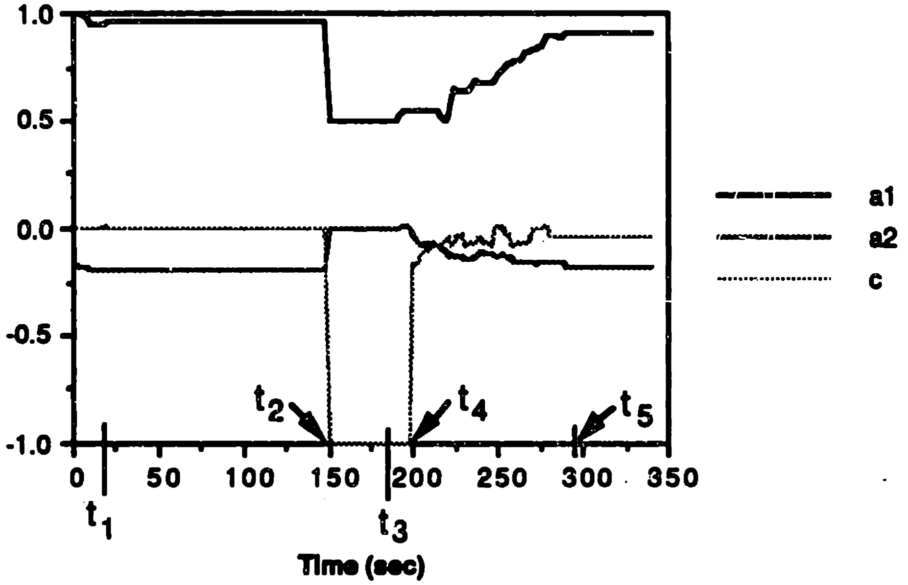


Figure 6.12a: Parameter Estimates Corresponding to Figure 6.11

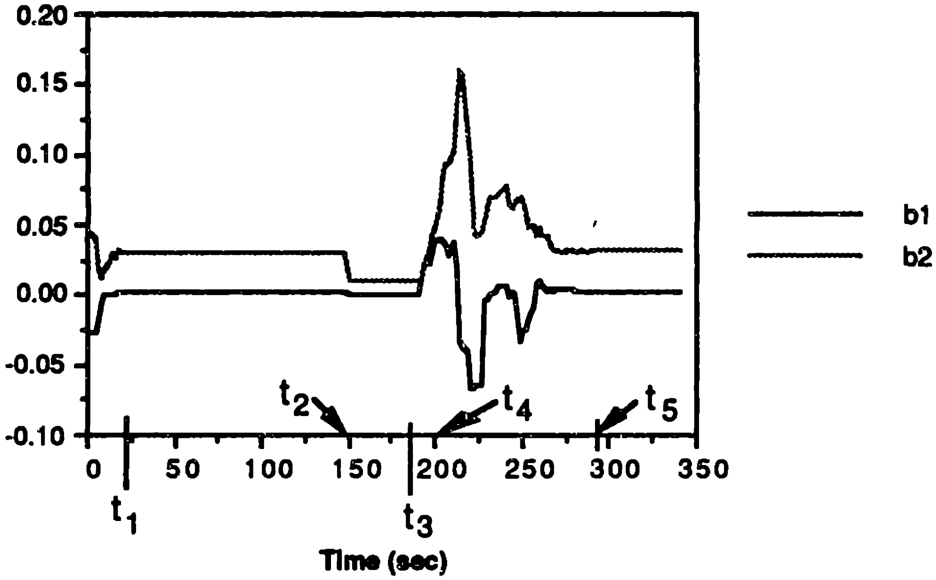


Figure 6.12b: Parameter Estimates Corresponding to Figure 6.11

Changing the Safety Net Limits

To demonstrate the robustness of the safety net design presented in Chapter 5, the heading control corresponding to the depth control in Figure 6.11 was done by changing the safety net limits and resetting the parameter estimates for the heading 4 times in 60 second increments. In this section resetting the parameters means to reinitialize them to 0, 0.5, 0, 0, 0 for the parameters hb1, ha1, ha2, hb2, hc, respectively. The safety net limits were changed according to Table 6.1. Figure 6.13 shows the resulting heading control.

Table 6.1: Safety Net Limits for the Heading Control in Figure 6.13

Time (sec)	Safety Net Limit
0 to 60	0.05 deg
60 to 120	0.2 deg
120 to 180	3 deg
180 to 240	6 deg
240 to 340	10 deg

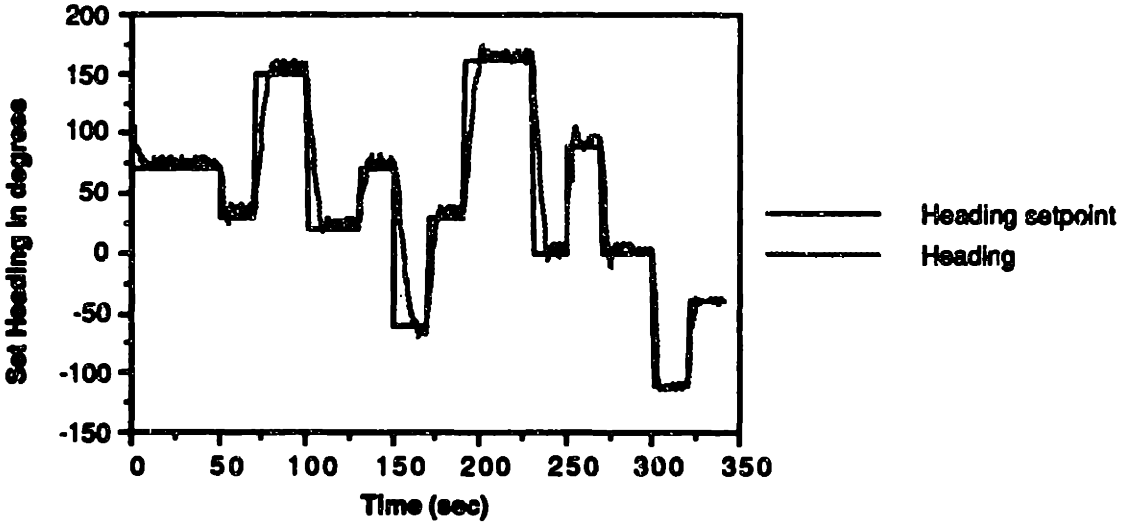


Figure 6.13: Heading Control Corresponding to Figure 6.11

Since the safety net limits were set to values less than or equal to 3 degrees for the first 180 seconds of the run shown in Figure 6.13, the response for this time period is almost completely determined by the robust controller. This is due to the fact that the robust controller is unable to control the vehicle to within 3 degrees of the desired trajectory especially since the two thrusters were miscalibrated. The miscalibration of the thrusters caused the robust controller to have a steady-state error of about 5 to 10 degrees which means that the adaptive controller never got a chance to control the AUV for any length of time during the first 180 seconds. This emphasizes the need to match the safety net limits to the performance of the robust controller as discussed in Chapter 5.

From Table 6.1 it can be seen that the parameter estimates and safety net limits were not changed after 240 seconds. Since four command changes occurred during these last 100 seconds, the parameter estimates had enough time to converge reasonably close to their actual values and the response of the vehicle improved considerably during this period. Although this was the only time period in this run in which the heading parameter estimates had enough time to converge before being reset, the parameter estimates always started to converge in the correct direction as can be seen in Figure 6.14a and 6.14b. The vertical dashed lines in Figure 6.14 indicate where the parameter estimates were reset and the safety net limits were changed. Due to memory limitations of the on-board computer and due to the fact that the vehicle position could not be measured on-line, it was not possible to continue any one test run for more than 7 or 8 minutes. This is why the parameter estimates were reset at 60 second intervals and not at longer intervals which would have allowed the parameter estimates to converge for each value of the safety net limits shown in Table 6.1.

As can be seen from Figure 6.13, the heading response greatly improves for the last two step command changes when the parameter estimates had converged sufficiently to allow adaptive control. The heading oscillations shown in Figure 6.13 during the first 180 (or even 240) seconds are partially due to the fact that the robust controller uses the actual heading measurement (as opposed to the Kalman estimate) and its derivative in the feedback control. The response time of the robust controller is much longer than that of the adaptive controllers. This can be seen by comparing the heading response during the first 180 seconds to the heading response after 300 seconds when the adaptive controller was used.

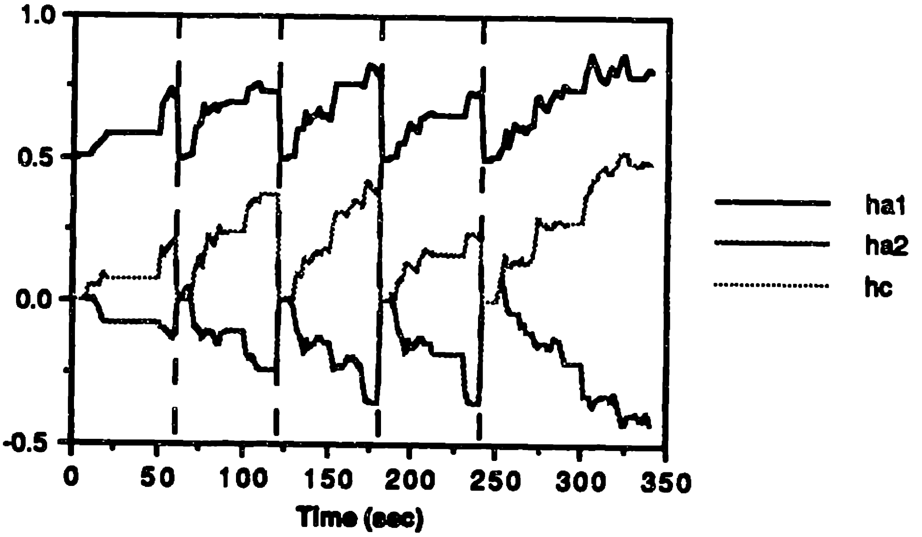


Figure 6.14a: Parameter Estimates Corresponding to Figure 6.13

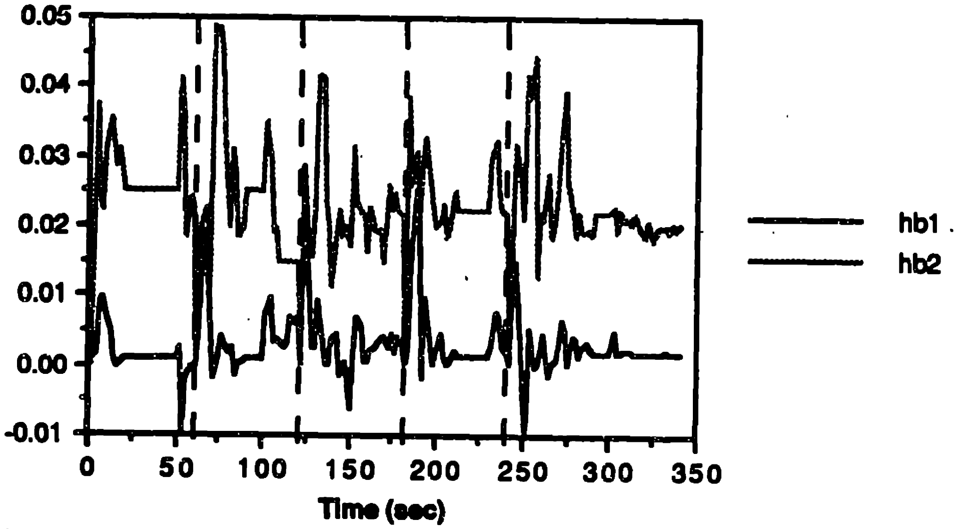


Figure 6.14b: Parameter Estimates Corresponding to Figure 6.13

From the previous discussion on the in-water tests using different safety net limits, it should be clear that not all limits used in Table 6.1 are acceptable for normal operation. Since the robust controller was not able to return the vehicle to the desired trajectory envelope for limits less than about 6 to 10 degrees, the limit should be chosen larger than 10 degrees. The normal value used for the safety net limits (i.e. the value used in all in-water heading tests other than the one shown in Figure 6.13) was 18 degrees. This value was determined through simulation prior to any in-water heading control tests and worked well for the actual vehicle.

CHAPTER 7

CONCLUSIONS AND FUTURE APPLICATIONS

7.1 Conclusions

The main goal of this thesis was to develop an indirect adaptive depth and heading controller for the *Sea Squirt* and demonstrate its performance in autonomous testing. This goal was accomplished as was discussed in Chapter 6. During the development of the adaptive controller, several identification methods and control designs were researched and compared. In comparing the estimation methods it was found that the extended Kalman Filter (KF) performed better than Recursive Least Square, Normalized Least Mean Square, and Output Error Minimization especially in situations with low signal-to-noise ratios. The KF also provided state estimates which were essential for state feedback control of the *Sea Squirt*. Although the KF was the best choice for the *Sea Squirt*, one of the other identification methods might be better suited for applications where the signal-to-noise ratio is high and the computational power available is insufficient for the KF.

The adaptive controllers which were compared in this thesis using the Kalman Filter identification method included adaptive Pole-Placement (PP), adaptive Sliding Mode (SM), and adaptive Linear Quadratic Gaussian (LQG). Both off-line and on-line comparisons were done. The PP design proved to perform just as well as the SM design with the LQG design performing worse than either of the other methods. Since the PP controller is simpler to implement than the SM controller and works just as well it was chosen as the preferred control design. It should be noted, however, that the controller comparisons were done based on the *Sea Squirt* or a model of the *Sea Squirt* and that the outcome of the comparison could be different for other plants.

7.2 Future Improvements and Applications

Although the indirect adaptive controller developed in this thesis worked well, several improvements could be made. For example, the autonomous in-water test results shown in Chapter 6 indicate that there is significant coupling between the heading and depth dynamics especially during quick heading changes. To reduce or even eliminate the depth transients caused by heading changes a multi-input-multi-output (MIMO) controller could be considered. For the *Sea Squirt* an adaptive MIMO controller would probably be too complex for on-line implementation, but future developments in computers might make such a design feasible. Even if an on-line implementation of the adaptive controller is not possible for the MIMO case, it might be possible to use the identification method to obtain a model of the vehicle and then do an off-line MIMO design. The depth transients caused by heading changes could also be reduced by including the pitch dynamics and measurements in the depth control design.

To improve the controllers robustness to thruster modeling errors and nonlinearities, a closed-loop thruster controller could be designed. Although this would imply adding a sensor that can measure either the force applied by the thruster or at least some measurement from which the force could be approximated such as propellor rpm, it would ensure that the thrust commanded by the vehicle controller is actually achieved by the thrusters. Knowing the thrust applied to the vehicle with greater accuracy would also improve the performance of the identification routine since it relies on the thrust commands and output measurements to obtain the parameter estimates. However, the adaptive controller developed in this thesis performed well even without closed-loop thrust control. Therefore, the addition of a sensor for the purpose of thrust measurement should only be considered if the desired performance cannot be achieved otherwise.

In the case where it is impossible to add a sensor which could approximate thrust, including the dynamics of the thrusters in the plant model could be considered. Although this would increase the system model dimension and the number of parameters that have to be identified, it would enable the adaptive controller to determine the dynamics of the thrusters on-line. The result would be improved performance of the adaptive controller (after parameter convergence) with less a priori knowledge of the plant. The influence of thruster dynamics on the behavior of underwater vehicles is discussed in [16].

APPENDIX 1

DEFINITION OF VEHICLE RELATED VARIABLES

Tc establish a coordinate frame, the *Sea Squirt* diagram was shown in Figure 2.1 while the corresponding euler angle representation is given in Figure A1.1. Small letters represent body relative directions while capital letters represent inertial (fixed) directions.

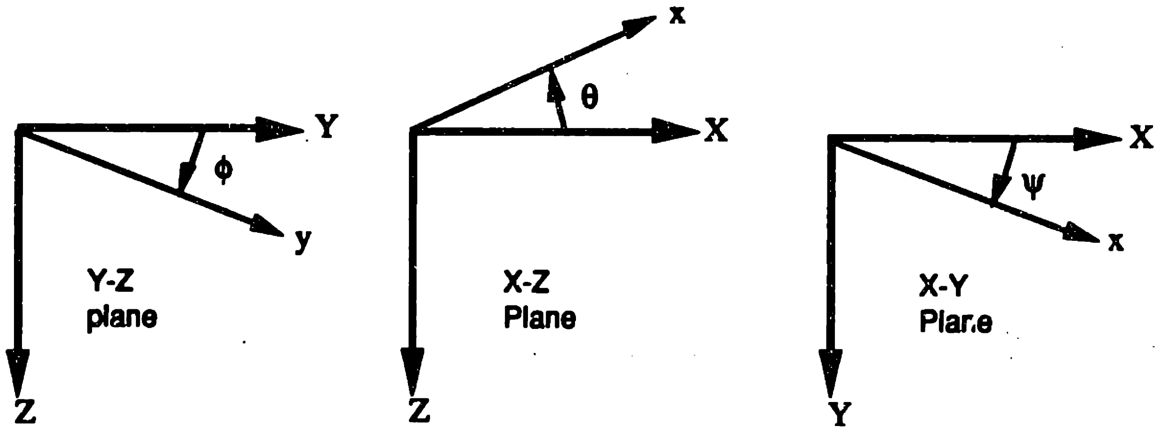


Figure A1.1: Representation of Euler Angles

Definition of Variables

T_v	=	vertical thrust
T_p	=	port thrust
T_s	=	starboard thrust
m	=	vehicle mass
B	=	buoyancy
W	=	weight

Fixed reference frame positions

X = position (X)
Y = position (Y)
Z = position (Z)
ϕ = roll angle
θ = pitch angle
ψ = yaw angle

Body relative velocities

u = forward velocity
v = starboard velocity
w = downward velocity
p = roll rate
q = pitch rate
r = yaw rate

The relations between the body relative and fixed reference frame variables are as follows:

$$\dot{\phi} = p + \dot{\psi}\ \sin(\theta)$$

$$\dot{\theta} = \cos(\phi) + r\ \sin(\phi)$$

$$\dot{\psi} = \frac{r\ \cos(\phi) + q\ \sin(\phi)}{\cos(\theta)}$$

$$\dot{X} = u\ \cos(\theta)\ \cos(\psi) + v\ \{\sin(\phi)\ \sin(\theta)\ \cos(\psi) - \cos(\phi)\ \sin(\psi)\} + w\{\sin(\phi)\ \sin(\psi) + \cos(\phi)\ \sin(\theta)\ \cos(\psi)\}$$

$$\dot{Y} = u\ \cos(\theta)\ \sin(\psi) + v\ \{\cos(\phi)\ \cos(\psi) + \cos(\phi)\ \sin(\theta)\ \sin(\psi)\} + w\{\cos(\phi)\ \sin(\theta)\ \sin(\psi) - \cos(\phi)\ \sin(\phi)\ \cos(\psi)\}$$

$$\dot{Z} = -u\ \sin(\theta) + v\ \cos(\theta)\ \sin(\phi) + w\ \cos(\theta)\ \cos(\phi)$$

APPENDIX 2

AUV EQUATIONS OF MOTION

Axial Forces:

$$m[\dot{u} - vr + wq - X_g(q^2 + r^2) + Y_g(pq - \dot{r}) + Z_g(pr + \dot{q})]$$
$$= X_{\dot{u}}\dot{u} + X_{u|u|}u|u| - (W - B)\sin(\theta) + T_p + T_s$$

Lateral Forces:

$$m[\dot{v} - wp + ur - Y_g(r^2 + p^2) + Z_g(qr - \dot{p}) + X_g(qp + \dot{r})]$$
$$= Y_{\dot{v}}\dot{v} + Y_{\dot{p}}\dot{p} + Y_{\dot{r}}\dot{r} + Y_{v|v|}v|v| + Y_{ur}ur + Y_{uv}uv + (W\text{-}B)\cos(\theta)\sin(\phi)$$

Vertical Forces:

$$m[\dot{w} - uq + vp - Z_g(p^2 + q^2) + X_g(rp - \dot{q}) + Y_g(rq + \dot{p})]$$
$$= Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{w|w|}w|w| + Z_{uw}uw + Z_{uq}uq + (W\text{-}B)\cos(\theta)\cos(\phi) + T_v$$

Rolling Moment:

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - (\dot{r} + pq)I_{zx} + (r^2 - q^2)I_{yz}$$
$$+ (pr - \dot{q})I_{xy} + m[Y_g(\dot{w} - uq + vp) - Z_g(\dot{v} - wp + ur)]$$
$$= K_{\dot{p}}\dot{p} + K_{\dot{v}}\dot{v} + (Y_gW - Y_bB)\cos(\theta)\cos(\phi) - (Z_gW - Z_bB)\cos(\theta)\sin(\phi) + T_vY_g$$

Pitching Moment:

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})qpr - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx}$$
$$+ (qp - \dot{r})I_{yz} + m[Z_g(\dot{u} - vr + wq) - X_g(\dot{w} - uq + vp)]$$
$$= M_{\dot{q}}\dot{q} + M_{\dot{w}}\dot{w} + M_{uw}uw + M_{uq}uq - (X_gW - X_bB)\cos(\theta)\cos(\phi)$$
$$- (Z_gW - Z_bB)\sin(\theta) - (T_p + T_s)Z_g + T_vX_g$$

Yawing Moment:

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy}$$
$$+ (rq - \dot{p})I_{zx} + m[X_g(\dot{v} - wp + ur) - Y_g(\dot{u} - vr + wq)]$$
$$= N_{\dot{r}}\dot{r} + N_{\dot{v}}\dot{v} + N_{ur}ur + N_{uv}uv + N_{r|r|}r|r| - (X_gW - X_bB)\cos(\theta)\sin(\phi)$$
$$- (Y_gW - Y_bB)\sin(\theta) + Y_T(T_p - T_s)$$

The approximate numerical values (in standard English units) for the constants used in the previous equations are shown in Table A2.1 below. These are obtained through hydrodynamic analysis of the actual vehicle and are, in general, only fair (or even poor) approximations to the actual values. The values were obtained from [2].

Table A2.1: Numerical Values For the *Sea Squirt* Constants (approximate)

B	80
I_{xx}	.3
I_{yy}	2.3
I_{zz}	2.3
$K_{p\|p\|}$	-.8
$K_{p'}$	-.025
$K_{v'}$	.0079
mass	1.97
$M_{q\|q\|}$	-6.5
$M_{q'}$	-.41
M_{uq}	-.32
M_{uw}	1.57
$M_{w\|w\|}$	0.0
$M_{w'}$	-.02
$N_{r\|r\|}$	-5.6
$N_{r'}$	-.38
N_{ur}	-.32
N_{uv}	-1.57
$N_{v'}$	0.0
W	80

X_b	.12
X_g	.12
$X_{u\|u\|}$	-.84
$X_{u'}$	-.35
Y_b	0.0
Y_g	0.0
$Y_{p'}$	.0079
$Y_{r'}$	0.0
Y_T	.56
Y_{ur}	.042
Y_{uv}	-.3
$Y_{v\|v\|}$	-3.00
$Y_{v'}$	-1.48
Z_b	0.0
Z_g	.091
$Z_{q'}$	-.02
Z_{uq}	-.042
Z_{uw}	-.28
$Z_{w\|w\|}$	-3.49
$Z_{w'}$	-1.32

Moments of Inertia

I_{xx} = product of inertia about X-axis

I_{yy} = product of inertia about Y-axis

I_{zz} = product of inertia about Z-axis

Moment Products (assumed to be zero for the *Sea Squirt*)

I_{xy} = product of inertia about XY-axis

I_{zx} = product of inertia about ZX-axis

I_{yz} = product of inertia about YZ-axis

Roll Related Coefficients

K = hydrodynamic force component along Y-axis (pitch)

$K_{p'}$ = partial of $K(p')$ with respect to p'

$K_{v'}$ = partial of $K(v')$ with respect to v'

$K_{p|p|}$ = partial of $K(p|p|)$ with respect to $p|p|$

Pitch Related Coefficients

M = hydrodynamic force component along Y-axis (pitch)

$M_{q'}$ = partial of $M(q')$ with respect to q'

$M_{w'}$ = partial of $M(w')$ with respect to w'

$M_{q|q|}$ = partial of $M(q|q|)$ with respect to $q|q|$

Yaw Related Coefficients

N = hydrodynamic moment component along Y-axis (yaw)

$N_{r'}$ = partial of N(r') with respect to r'

$N_{v'}$ = partial of N(v') with respect to v'

$N_{r|r|}$ = partial of N(r|r|) with respect to r|r|

Axial Coefficients

X = hydrodynamic force component along X-axis

X_b = X coordinate of c.b

X_g = X coordinate of c.g.

X_T = X distance from vehicle c.g. to the horizontal thruster c.g.

$X_{u'}$ = partial of X(u') with respect to u'

$X_{u|u|}$ = partial of X(u|u|) with respect to u|u|

Lateral Coefficients

Z = hydrodynamic force component along Y-axis

Y_b = Y coordinate of c.b

Y_g = Y coordinate of c.g.

Y_T = Y distance from vehicle c.g. to the horizontal thruster c.g.

$Y_{p'}$ = partial of Y(p') with respect to p'

Y_{ur} = partial of Y(ur) with respect to ur

Y_{uv} = partial of Y(uv) with respect to uv

$Y_{v'}$ = partial of Y(v') with respect to v'

$Y_{v|v|}$ = partial of Y(v|v|) with respect to v|v|

Vertical Coefficients

Z = hydrodynamic force component along Z-axis

Z_b = Z coordinate of c.b.

Z_g = Z coordinate of c.g.

$Z_{q'}$ = partial of Z(q') with respect to q'

Z_{uq} = partial of Z(uq) with respect to uq

Z_{uw} = partial of Z(uw) with respect to uw

$Z_{w'}$ = partial of Z(w') with respect to w'

$Z_{w|w|}$ = partial of Z(w|w|) with respect to w|w|

APPENDIX 3

DERIVATION OF DISCRETE TIME DEPTH DYNAMICS

The following derivation converts the continuous time depth equations (A3.1) to the discrete time depth equations (A3.2).

$$\frac{d\ depth}{dt} = w \qquad \text{(A3.1a)}$$

$$\frac{dw}{dt} = a\,w + b\,w\,|w| + \frac{c + T_V}{mass\text{-}Z_{w'}} \qquad \text{(A3.1b)}$$

where

$$a = \frac{1}{mass\text{-}Z_{w'}} Z_{uw}\ u_o$$

$$b = \frac{1}{mass\text{-}Z_{w'}} Z_{w|w|}$$

u_o= some nominal forward velocity
w = depth velocity as a function of time
T_V= vertical thrust (pounds)
c = W-B

$$d(N+1) = d(N) + T\ w(N) + b_1\ (T_V(N) + c) \qquad \text{(A3.2a)}$$

$$w(N+1) = a_1\ w(N) + a_2\ w(N)\ |w(N)| + b_2\ (T_V(N) + c) \qquad \text{(A3.2b)}$$

where

d = depth
w = depth velocity
T = sampling period (.2 seconds)
b_1, a_1, a_2, b_2, c = parameters to be identified

Since the thrust, T_V, and the buoyancy of the vehicle are constant over one sampling period, equation (A3.1b) can be rewritten as follows:

$$\frac{dw}{dt} = a\,w + b\,w^2 + D \qquad \text{(A3.3)}$$

for w>0 and as

$$\frac{dw}{dt} = a\,w - b\,w^2 + D \qquad \text{(A3.4)}$$

for w<0.

D is a constant (over one sampling period) and is given by

$$D = \frac{c + T_V}{\text{mass-}Z_{w'}} \qquad \text{(A3.5)}$$

Since the solution to eq(A3.3) and eq(A3.4) are almost identical, the detailed math will be shown for eq(A3.3) only (w>0). The D term in equations (A3.3) and (A3.4) will be dealt with later and is omitted from the following equations to clarify the math.

Equation (A3.3) can be written in the following form

$$\int \frac{dw}{a\,w + b\,w^2} = \int dt \qquad \text{(A3.6)}$$

Integration yields

$$\frac{-1}{a} \ln\left|\frac{a + b\,w}{w}\right| = t + C \qquad \text{(A3.7)}$$

where C is the constant of integration.

If a + b w > 0,

$$w(t) = \frac{a}{e^{-at}\,e^{-aC} - b} \qquad \text{(A3.8)}$$

To evaluate e^{-aC}, let's define an initial condition

$$w_o = w(t{=}0) = \frac{a}{e^{-aC} - b} \tag{A3.9}$$

Now equation (A3.8) becomes

$$w(t) = \frac{a}{e^{-at}\left(\frac{a}{w_o} + b\right) - b} \tag{A3.10}$$

To check the math, one can evaluate eq(A3.10) for b=0. This results in the familiar linear solution

$$w(t) = e^{at}\, w_o \tag{A3.11}$$

Also, it is known that the depth dynamics are stable, and hence equation (A3.10) should approach zero as t approaches infinity. Indeed this is the case because 'a' is negative and thus the denominator of equation (A3.10) approaches infinity as t approaches infinity.

If $a + b\,w < 0$, eq(A3.8) becomes

$$w = \frac{a}{-\,e^{-at}\,e^{-aC} - b} \tag{A3.12}$$

and eq(A3.9) becomes

$$w_o = \frac{a}{-e^{-aC} - b} \tag{A3.13}$$

Combining eq(A3.12) and eq(A3.13) results in eq(A3.10) again. Therefore, eq(A3.10) holds, in general, for w>0.

Starting with eq(A3.4) instead of eq(A3.3) and repeating the above steps, results in

$$w(t) = \frac{a}{e^{-at}\left(\frac{a}{w_o} + b\right) + b} \qquad \text{(A3.14)}$$

for w<0.

To obtain the desired discrete time equivalent of eq(A3.1b), let's define the following discrete variables:

$$w_N = w_o \qquad \text{(A3.15)}$$

$$w_{N+1} = w(T) \qquad \text{(A3.16)}$$

where T is the sampling period.

We can now write w_{N+1} in terms of the previous time step w_N as follows

$$w_{N+1} = \frac{A\ w_N}{B\ w_N + C} \qquad \text{(A3.17)}$$

for w>0, and

$$w_{N+1} = \frac{A\ w_N}{-B\ w_N + C} \qquad \text{(A3.18)}$$

for w<0
where

$$A = -a$$
$$B = b\ (1 - \exp(-aT))$$
$$C = -a\ \exp(-aT)$$

Eq(A3.17) and eq(A3.18) are not ideally suited for identification purposes since they would require the identification of the three parameters A, B, and C which appear nonlinearly; identification is simplified if the parameters appear linearly in the model. Also, the sign difference between the two equations might not be easily dealt with.

To find a more suitable model representation we use the Taylor's Series expansion of eq(A3.17) and eq(A3.18). Since w can take on both positive and negative values, the expansion will be done around w=0 as follows:

Define $w_{N+1} = f(w_N)$

then

$$f(w_N) = f(0) + f'(0)\, w_N + \frac{f''(0)}{2!} w_N^2 + \frac{f'''(0)}{3!} w_N^3 + \dots \qquad \text{(A3.19)}$$

For w>0,

$$f(0) = 0$$

$$f'(0) = \left.\frac{AC}{(Bw_N + C)^2}\right|_{w_N=0} = \frac{A}{C}$$

$$f''(0) = \left.\frac{-2ABC}{(Bw_N + C)^3}\right|_{w_N=0} = \frac{-2AB}{C^2}$$

$$f'''(0) = \left.\frac{6AB^2C}{(Bw_N + C)^4}\right|_{w_N=0} = \frac{6AB^2}{C^3}$$

For w<0,

$$f(0) = 0$$

$$f'(0) = \left.\frac{AC}{(-Bw_N + C)^2}\right|_{w_N=0} = \frac{A}{C}$$

$$f''(0) = \left.\frac{-2ABC}{(-Bw_N + C)^3}\right|_{w_N=0} = \frac{-2AB}{C^2}$$

$$f'''(0) = \left.\frac{6AB^2C}{(-Bw_N + C)^4}\right|_{w_N=0} = \frac{6AB^2}{C^3}$$

The previous equations for w>0 and w<0 can combined to give the following result:

$$w_{N+1} = \frac{A}{C} w_N - \frac{AB}{C^2} w_N|w_N| + \frac{AB^2}{C^3} w_N^3 - \frac{AB^3}{C^4} w_N|w_N^3| + \ldots \qquad \text{(A3.20)}$$

where

$$A = \frac{-1}{\text{mass-}Z_{w'}} Z_{uw}\, u_o$$

$$B = b(1 - e^{-aT})$$

$$b = \frac{1}{\text{mass-}Z_{w'}} Z_{w|w|}$$

$$C = -a\, e^{-aT}$$

$$a = -A$$

One now has to evaluate the coefficients of eq(A3.19) to determine how many terms are needed for a good approximation of $f(w_N)$. From above,

$$\frac{A}{C} = e^{aT} \qquad \text{(A3.21)}$$

$$\frac{B}{C} = \frac{b(1 - e^{-aT})}{-a\, e^{-aT}} \qquad \text{(A3.22)}$$

Substituting for 'b' and 'a' in equation (A3.22) yields

$$\frac{B}{C} = \frac{Z_{w|w|} - Z_{w|w|}\, e^{-aT}}{Z_{uw}\, e^{-aT}\, u_o} \qquad \text{(A3.23)}$$

If $u_o = 0$, then $a = 0$ and

$$\frac{B}{C} = \frac{0}{0} \tag{A3.24}$$

To evaluate eq(A3.23) for $u_o = 0$, we apply L'Hôpital's Rule which states

$$\lim_{u \to 0} \frac{f(u)}{g(u)} = \lim_{u \to 0} \frac{f'(u)}{g'(u)} \tag{A3.25}$$

Differentiating the numerator and denominator of eq(A3.23) with respect to u_o and taking the limit gives

$$\lim_{u_o \to 0} \frac{B'}{C} = \frac{Z_{w|w|}\, T}{\text{mass-}Z_{w'}} \tag{A3.26}$$

Note: 'a' is a function of u_o as defined earlier.

For the *Sea Squirt* eq(A3.26) equals approximately -.21. Clearly, the terms in eq(A3.20) will decrease in magnitude as long as $\frac{B}{C} < 1$ for all reasonable values of u_o. Assuming that the *Sea Squirt* has a maximum forward velocity of 10 ft/sec (actually the maximum forward velocity is close to 5 ft/sec), the coefficients of eq(A3.20) can be plotted as a function of u_o as follows:

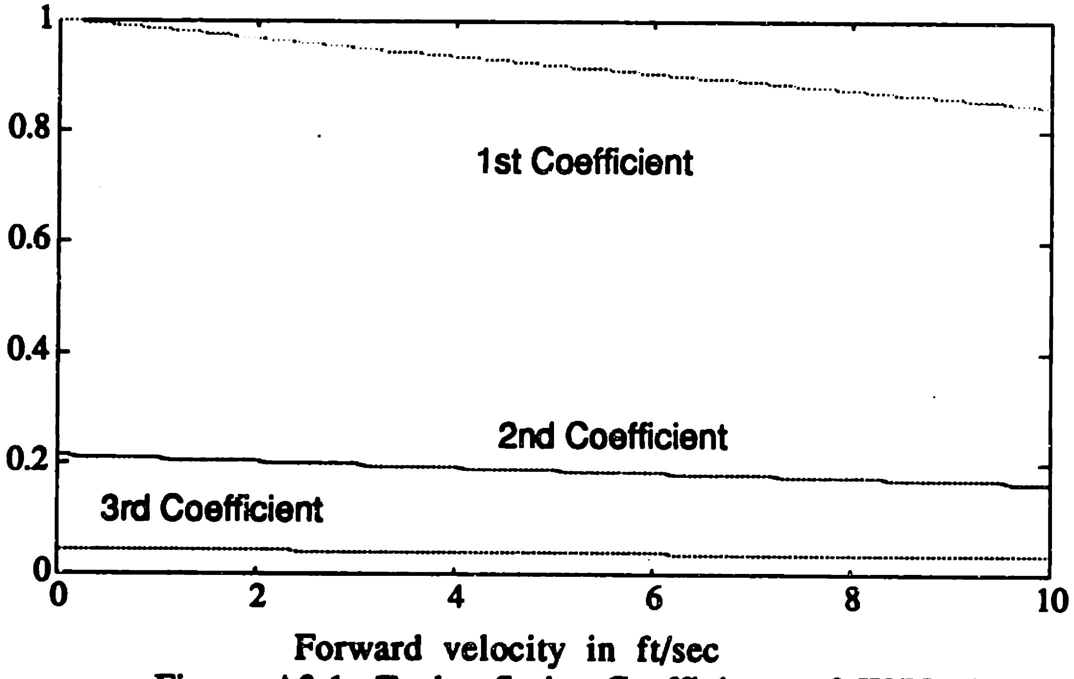


Figure A3.1: Taylor Series Coefficients of W(N+1)

If we limit the depth rate, w, to ±1 ft/sec (which is reasonable for the *Sea Squirt*), it is sufficient to consider only the first two terms of eq(A3.20) to obtain a good estimate of w_{N+1}.

Since in eq. (A3.1) the vertical thrust, T_v, and the buoyancy, c, are constant over one sampling period, repetition of this derivation including the constant D results in eq. A3.2 where T_v and c appear linearly. This results in the constant b_2 in equation (A3.27b). Integrating the depth velocity yields the desired discrete time depth dynamics (A3.27).

$$d(N+1) = d(N) + T\,w(N) + b_1\,(T_v(N) + c) \tag{A3.27a}$$

$$w(N+1) = a_1\,w(N) + a_2\,w(N)\,|w(N)| + b_2\,(T_v(N) + c) \tag{A3.27b}$$

where

d = depth
w = depth velocity
T = sampling period
= .2 seconds

The thrust and buoyancy also appear in equation (A3.27a) (not only in equation (A3.27b) because the depth at time N+1 is partially due to the force applied at time N. Parameters b_1 and b_2 are both dependent on the sampling period, T. The parameters to be identified are b_1, a_1, a_2, b_2, and c. Any net buoyancy of the vehicle will be captured by parameter c.

APPENDIX 4

ANALYTIC SOLUTION OF THE DISCRETE TIME RICCATI EQUATION

The second order *Sea Squirt* depth equations (derived in Section 2.2) are given by

$$d(N+1) = d(N) + T\,w(N) + b_1\,(T_v + c) \qquad (3.2.4a)$$

$$w(N+1) = a_1\,w(N) + a_2\,w(N)\,|w(N)| + b_2\,(T_v + c) \qquad (3.2.4b)$$

where

d	=	depth
w	=	depth velocity
T	=	sampling period (.2 seconds)
b_1, a_1, a_2, b_2, c	=	parameters to be identified

Representing the above equations in matrix form and neglecting the nonlinear term yields

$$x(N+1) = A\,x(N) + B\,u(N)$$

where

$$A = \begin{bmatrix} 1 & T \\ 0 & a_1 \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$x(N) = \begin{bmatrix} d(N) \\ w(N) \end{bmatrix}$$

$$u(N) = T_v + c$$

Since the Riccati equation can not be easily solved analytically for the A and B matrix given, the system will be transformed to controllable form as follows (the subscript "c" will be used to refer to the matrices in controllable form):

Define

$$z(N) = H\, x(N)$$

such that

$$A_c = \begin{bmatrix} 0 & 1 \\ -a_1 & 1+a_1 \end{bmatrix} \qquad B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where $A_c = H\, A\, H^{-1}$, $B_c = H\, B$

$$H = \begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix}$$

After some algebraic manipulations, it can be seen that H is given by

$$h_1 = h_3 = \frac{1}{b_1 + T\, b_2 - b_1\, a_1}$$

$$h_2 = \frac{-\, h_1\, b_1}{b_2}$$

$$h_4 = \frac{1\, -\, h_1\, b_1}{b_2}$$

To solve the Riccati equation in controllable form, the state weighting matrix, Q, also has to be converted to controllable form as follows:

Define positive semidefinite weighting matrices Q and Q_c

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \qquad Q_c = \begin{bmatrix} q_{c1} & q_c \\ q_c & q_{c2} \end{bmatrix}$$

such that

$$Q_c = (H^T)^{-1}\, Q\, H^{-1}.$$

Note that Q was chosen to be diagonal so that there is one weight for each state. This choice of Q will not simplify the solution of the Riccati equation since Q_c is not diagonal, but it does simplify the choice of q_1 and q_2. The following equations give the relation between the elements of Q_c and those of Q.

Define auxiliary variables

$$aux_1 = (h_1 h_4 + h_1 h_2)^2$$

$$aux_2 = h_1^2 q_2$$

then

$$q_{c1} = \frac{aux_2 + h_4^2 q_1}{aux_1}$$

$$q_{c2} = \frac{aux_2 + h_2^2 q_1}{aux_1}$$

$$q_c = -\frac{aux_2 + h_2 h_4 q_1}{aux_1}$$

Since we are dealing with a single input system, the control weighting, R, will be a scalar, r. The matrix Riccati equation can now be solved using A_c, B_c, Q_c, and r define above. To avoid writing several pages of tedious algebra, only the final analytic solution will be given here. Some of the algebra involved was done on a Macintosh using *Mathematica* software.

Let the solution matrix of the Riccati equation be given by

$$X = \begin{bmatrix} x_1 & x \\ x & x_2 \end{bmatrix}$$

To simplify the solution let us define the following variables.

$a = 1+a_1$
$t_0 = a\ r$
$t_1 = q_{c1} + q_{c2}$
$t_2 = t_1/2 + t_0\ (a - 2)$
$t_3 = q_c^2 - a\ t_0\ (2\ q_c + t_1)$
$t_4 = q_c^2 - 2\ t_0\ (t_1 + a\ q_c) + t_3$

$$t_5 = \sqrt{\frac{t_1^2}{4} - t_3}$$

$$t_6 = \frac{t_1\ t_2 - t_4}{2\ t_5} + t_2$$

$t_7 = r + x_2$

Then the solution to the Riccati equation is given by

$$x_2 = \frac{t_6 + \sqrt{4\ r\ \left(\frac{t_1}{2} + t_5\right) + t_6^2}}{2}$$

$$x = \frac{-a_1\ t_0\ x_2 + q_c\ t_7}{t_0 + x_2}$$

$$x_1 = q_{c1} + \frac{r\ a_1^2\ x_2}{t_7}$$

The feedback gain vector, K, is given by

$$K = K_c\, H = [k_1\ k_2]$$

where

$$k_1 = \frac{h_1(x_2 + x)}{t_7}$$

$$k_2 = \frac{h_4\ x + x_2(h_2 + a\,(h_4 - h_2))}{t_7}$$

REFERENCES

[1] J. Feldman, "DTNSRDC Revised Standard Submarine Equations of Motion," DTNSRDC/SPD-0303-09, June 1979.

[2] Eric Wallar, Design and Full-Scale Testing of a Sliding Mode Controller for a Small Underwater Vehicle. Master's Thesis, MIT, Cambridge, Massachusetts, July 1989.

[3] C. Richard Johnson, Jr. (1988): Lectures on Adaptive Parameter Estimation, Prentice-Hall, Inc., Englewood Cliffs, N. J.

[4] Arthur Gelb (1974): Applied Optimal Estimation. The MIT Press, Cambridge, MA.

[5] David M. DeLonga, A Control System Design Technique for Nonlinear Discrete Time Systems. PhD. Thesis, MIT, Cambridge, MA, February 1989.

[6] Kwakernaak and Sivan (1972): Linear Optimal Control Systems. John Wiley and Sons Inc.

[7] Rafel Santos-Mendes and Joseph Aguilar-Martin, "Robust Pole-Placement Design," Int. J. Control, 1989, vol. 50, No. 1, 113-128.

[8] T. Pappas, A. J. Laub and N. R. Sandell, "On the Numerical Solution of the Discrete-Time Algebraic Riccati Equation," IEEE Trans. on Automatic Control, vol. 25, No. 4, 1980.

[9] P. M. Mäkilä and H. T. Toivonen, "Computational Methods for Parametric LQ Problems--A Survey," IEEE Trans. on Automatic Control, vol. 32, No. 8, 1987.

[10] J.-J. E. Slotine and J. A. Coetsee, "Adaptive Sliding Controller Synthesis for Non-linear Systems," Int. J. Control, vol. 43, No. 6, pp. 1631-1651,1986.

[11] R. Cristi, F. A. Papoulias, and A. J. Healey, "Adaptive Sliding Mode Control of Autonomous Underwater Vehicles in the Dive Plane," IEEE Oceanic Engineering, vol. 15, No. 3, pp. 152-60, 1990.

[12] Rogelio Lozano-Leal, "Robust Adaptive Regulation Without Persistent Excitation", IEEE Trans. on Automatic Control, vol. 34, No. 12, pp. 1260-1267,1989.

[13] K. J. Åstrom and B. Wittenmark (1984): Computer Controlled Systems . Prentice-Hall, Inc., Englewood Cliffs, N. J.

[14] Charles E. Rohrs, Rethinking Adaptive Control for the 90's, Proceedings of the 29th Conference on Decision and Control, Dec. 1990, Honolulu, Hawaii.

[15] B. B. Peterson and K. S. Narendra, "Bounded Error Adaptive Control", IEEE Trans. on Automatic Control, vol. 27, No. 6, 1982.

[16] Dana R. Yoerger, "The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation Into Control System Design," IEEE Oceanic Engineering, vol. 15, No. 3, pp. 167-78, 1990.