NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TRACKING CONTROL OF AUTONOMOUS UNDERWATER VEHICLES

by

Joseph J. Keller

December 2002

Thesis Advisor: Anthony J. Healey

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Tracking Control for AUV's			**5. FUNDING NUMBERS** N0001401AF00002
6. AUTHOR(S) Joseph J. Keller			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217-5660			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT

Recovery of Autonomous Underwater Vehicles (AUVs) can often be an autonomous operation itself. In the case of an AUV that is launched and recovered at some significant depth below the surface, the recovery platform to which the vehicle will dock is often not a stationary platform. The recovery cage/platform has dynamics associated with it which are induced by wave motion effects on the ship to which the cage is tethered. In order to successfully recover a vehicle into a cage platform it will be preferred for the vehicle to have the capability to compensate for this motion when making its final approach to the cage. Using active compensation, a smaller cage can be utilized for recovery of an AUV.

This research attempts to investigate a means by which a vehicle may be made to track, in depth, dynamic motion with zero phase lag between the vehicle and the recovery platform utilizing an error space controller.

14. SUBJECT TERMS Underwater Vehicle, AUV, Tracking, Control, Error Space Control, AUV Parameter Identification, AUV recovery			**15. NUMBER OF PAGES** 86
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

TRACKING CONTROL OF AUTONOMOUS UNDERWATER VEHICLES

Joseph J. Keller
Lieutenant, United States Navy
B.S.M.E., San Diego State University, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MECHANICAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2002

Author: Joseph J. Keller

Approved by: Anthony J. Healey
Thesis Advisor

Young Kwon
Chairman, Department of Mechanical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Recovery of Autonomous Underwater Vehicles (AUVs) can often be an autonomous operation itself. In the case of an AUV that is launched and recovered at some significant depth below the surface, the recovery platform to which the vehicle will dock is often not a stationary platform. The recovery cage/platform has dynamics associated with it, which are induced by wave motion effects on the ship to which the cage is tethered. In order to successfully recover a vehicle into a cage platform it will be preferred for the vehicle to have the capability to compensate for this motion when making its final approach to the cage. Using active compensation, a smaller cage can be utilized for recovery of an AUV.

This research attempts to investigate a means by which a vehicle may be made to track, in depth, dynamic motion with zero phase lag between the vehicle and the recovery platform utilizing an error space controller.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank all of the people who helped support me through this great learning and growth opportunity. This thesis is the culmination of all of the engineering education I have had the opportunity to receive and has cemented my desire to continue to work in exciting and technically challenging areas the Navy has to offer.

I would like specifically thank the following people for making this thesis possible:

My parents. For their loving support over all the years of my education. Without the values of hard work and determination that they instilled in me, I would not be able to take on such challenges as these.

CDR Bill Marr. I would like to thank for his excellent guidance and leadership in working in the field of AUV research. Without his support the operations and experiments utilizing NPS's AUV vehicle ARIES would not be possible.

Professor Anthony Healey. I would especially like to thank for his incredible vision and techinal mastery in the field of Controls Systems Engineering and AUV operations. It is only with his guidance and mentoring that this project was made possible.

My dog Koa, whose unconditional love and constant companionship through many hours on the computer made working on this thesis a real team effort.

My wife Abby, who is my partner in life, with which I share all experiences, exciting and challenging. For her the love and support through the challenging times and for her shared happiness during the exciting times.

Last of all to God, for the spiritual strength and guidance during the highs and lows of the thesis process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Research in the field of autonomous underwater vehicles (AUVs) at the Naval Postgraduate School (NPS) has progressed steadily since the inception of the Center for AUV Research in 1987. The operational capabilities and sophistication of software and hardware has greatly increased with each new generation of vehicle. From humble beginnings in swimming pools to open ocean operation, these vehicles have been at the forefront of AUV research.

The current generation of NPS AUV is the Acoustic Radio Interactive Exploratory Server (ARIES). While ARIES is designed for the purpose of research into Minewarfare and Acoustic Communications, it remains a valuable platform for controls testing of AUV's. With its dual onboard computers, ARIES is a highly configurable test platform with which various aspects of AUV operations and research can be conducted.

One particular area of interest is the capability to deploy and recover AUV's from a tethered cage at some depth below the surface of the water. These AUV's are utilized for deep water operations where the cage deployment and docking operations become necessary to conserve endurance of the AUVs. The challenge arrives upon recovery of the AUV to the cage platform. The cage platform has dynamics associated with it which are induced by wave motion effects of the ship to which the cage is tethered. In order to successfully recover a vehicle into a cage it will be necessary for the vehicle to have the capability to compensate for this motion when making its final approach to the cage.

B. SCOPE OF THIS WORK

Previous work in the field of AUV control has shown many different techniques for controlling an AUV's trajectory. From simple state feedback methods, to optimal methods like Linear Quadratic Regulator (LQR), to Lyapanov based methods like Sliding Mode Control (SMC), there are many methods which can be successfully utilized, each having particular advantages and disadvantages. In many cases "control" means the AUV's ability to "drive" a pre-planned track of waypoints or to regulate the vehicles depth or altitude. Tracking a moving object is a more complex problem. This is due to the fact the dynamics of the moving object may not be know a priori, and the AUV tracking control must have a way to compensate for the signal dynamics.

While there have been successful applications of recovery of AUV's into recovery platforms such as with MIT's REMUS vehicle [1], this study attempts to show a new method in which tracking and recovery onto a moving platform is enabled.

The focus of this thesis is two-fold:

1. Develop an error space control method to obtain zero error tracking of a dynamic cage system by an AUV in depth control mode.

2. Verify the controller's performance by developing models, simulation, and experimental validation utilizing ARIES as a test platform.

Chapter II will focus on the equations of motion for an AUV and methods for modeling AUV dynamics. Chapter III will discuss tracking control and the use of a general error space method to track a dynamic signal. Chapter IV will discuss application of an error space controller to an AUV attempting to track periodic motion. Chapter V will present simulation and experimental results from the implementation of the error space controller in ARIES.

II. EQUATIONS OF MOTION AND AUV MODELING

A. GENERALIZED EQUATIONS OF MOTION

This section describes the equations of motion for an AUV. It is from these equations of motion that a model can be developed for both simulation of motion as well as construction of model based controllers for AUV's.

Using a Newton-Euler approach, Healey [2] derives the equations of motion for six degrees of freedom as:

SURGE EQUATION OF MOTION

$$m[\dot{u}_r - v_r r + w_r q - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] + (W - B)\mathbf{sin}\,\theta = X_f \tag{1}$$

SWAY EQUATION OF MOTION

$$m[\dot{v}_r + u_r r - w_r p + x_G(pq + \dot{r}) - y_G(p^2 + r^2) + z_G(qr - \dot{p})] - (W - B)\mathbf{cos}\,\theta\,\mathbf{sin}\,\phi = Y_f \tag{2}$$

HEAVE EQUATION OF MOTION

$$m[\dot{w}_r - u_r q + v_r p + x_G(pr - \dot{q}) + y_G(qr + \dot{p}) - z_G(p^2 + q^2)] + (W - B)\,\mathbf{cos}\,\theta\,\mathbf{cos}\,\phi = Z_f \tag{3}$$

ROLL EQUATION OF MOTION

$$\begin{aligned} I_x\dot{p} + (I_z - I_y)qr + I_{xy}(pr - \dot{q}) - I_{yz}(q^2 - r^2) - I_{xz}(pq + \dot{r}) + m[y_G(\dot{w} - u_r q + v_r p) \\ - z_G(\dot{v}_r + u_r r - w_r p)] - (y_G W - y_B B)\mathbf{cos}\,\theta\,\mathbf{cos}\,\phi + (z_G W - z_B B)\mathbf{cos}\,\theta\,\mathbf{sin}\,\phi = K_f \end{aligned} \tag{4}$$

PITCH EQUATION OF MOTION

$$\begin{aligned} I_y\dot{q} + (I_z - I_z)pr - I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) + I_{xz}(p^2 - r^2) - m[x_G(\dot{w} - u_r q + v_r p) \\ - z_G(\dot{u}_r - v_r r + w_r q)] + (x_G W - x_B B)\,\mathbf{cos}\,\theta\,\mathbf{cos}\,\phi + (z_G W - z_B B)\mathbf{sin}\,\theta = M_f \end{aligned} \tag{5}$$

YAW EQUATION OF MOTION

$$\begin{aligned} I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) + m[x_G(\dot{v}_r + u_r r - w_r p) \\ - y_G(\dot{u}_r - v_r r + w_r q)] - (x_G W - x_B B)\,\mathbf{cos}\,\theta\,\mathbf{sin}\,\phi - (y_G W - y_B B)\mathbf{sin}\,\theta = N_f \end{aligned} \tag{6}$$

Where:

u_r, v_r, w_r = component velocities for a body fixed system with respect to the water
p, q, r = component angular velocities for a body fixed system
W = weight
B = buoyancy
I = mass moment of inertia terms
x_B, y_B, z_B = position difference between geometric center of the AUV and center of buoyancy
x_G, y_G, z_G = position difference between geometric center of AUV and center of gravity
X_f, Y_f, Z_f, K_F, M_f, N_f = sums of all external forces and moments acting on an AUV in the particular body fixed direction

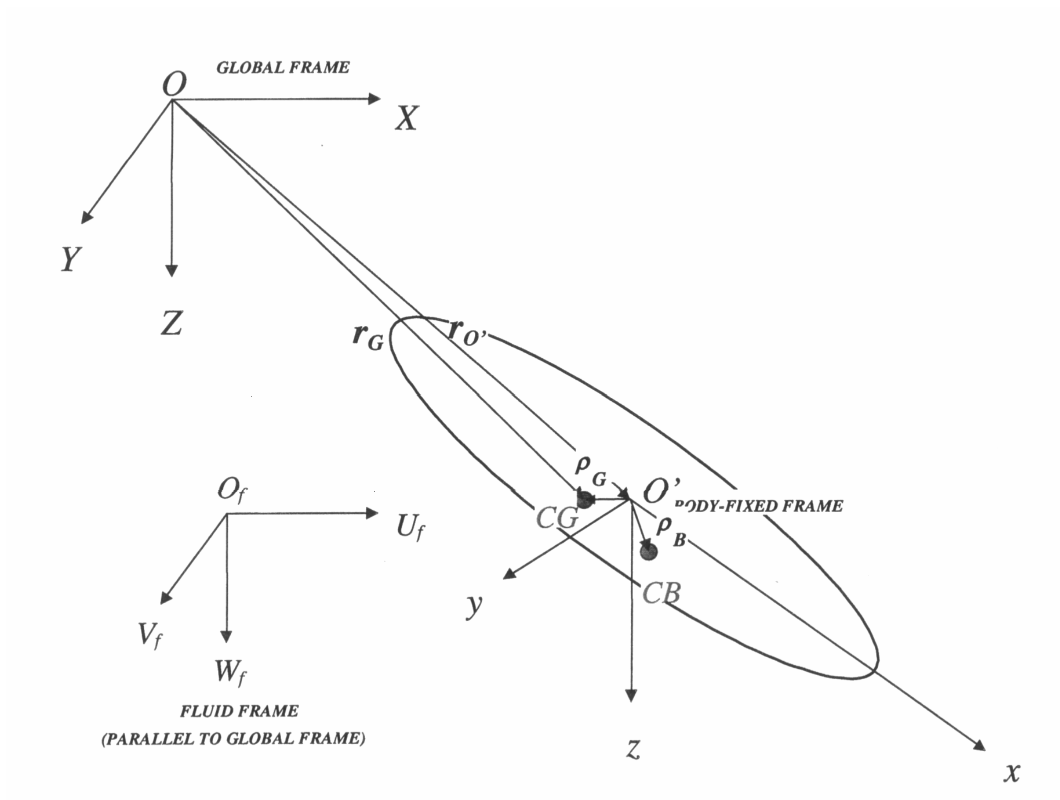

Figure 1 Coordinate System for Vehicle Dynamics from Ref [3]

1. Diving System Model

For the purposes of this study a diving mode controller will be designed, therefore a diving system model will be developed from the above equations of motion. The primary variables of interest are w_r, q, q and z while v_r, r, p, j , y , x, y are neglected. Assuming the vehicle is already in forward motion, under constant forward speed relative to the water, all products of small motions are ignored and the horizontal plane motions coupled to the vertical plane equations can be dropped. Primarily considering the effects of vehicle inertia, hydrostatic and weight terms, and hydrodynamic force components from lift and added mass a set of simplified equations of motion are developed [2].

To handle the force and moment terms, an assumption of "small" motions is made to develop "hydrodynamic coefficients" that can be defined relative to the individual motion components. This will allow the description of the forces and moments as a function of vehicle dynamic states. For heave motion, equation (3), the force in the z direction is:

$$Z_f = Z_{\dot{w}_r}\dot{w}_r + Z_{w_r}w_r + Z_{\dot{q}}\dot{q} + Z_q q \tag{7}$$

and for pitch motion , equation (4) , the rotational moment is:

$$M_f = M_{\dot{w}}\dot{w}_r + M_{w_r}w_r + M_{\dot{q}}\dot{q} + M_q q \tag{8}$$

This leads to:

$$Z_{\dot{w}_r} = \frac{\partial Z_f}{\partial \dot{w}_r};\ Z_{w_r} = \frac{\partial Z_f}{\partial w_r};\ Z_{\dot{q}} = \frac{\partial Z_f}{\partial \dot{q}};\ Z_q = \frac{\partial Z_f}{\partial q};$$

and

$$M_{\dot{w}_r} = \frac{\partial M_f}{\partial \dot{w}_r};\ M_{w_r} = \frac{\partial M_f}{\partial w_r};\ M_{\dot{q}} = \frac{\partial M_f}{\partial \dot{q}};\ M_q = \frac{\partial M_f}{\partial q};$$

Where:

$Z_{\dot{w}_r}$ = added mass due to heave velocity
$M_{\dot{q}}$ = added mass due to pitch rate
Z_{w_r} = coefficient of heave force induced by heave velocity
Z_q = coefficient of heave force induced by pitch rate
$M_{\dot{w}_r}$ = added mass moment of inertia due to heave velocity
$M_{\dot{q}}$ = added mass moment of inertia due to pitch rate
M_{w_r} = coefficient of pitch moment induced by heave velocity
M_q = coefficient of pitch moment induced by pitch rate

In addition, the action of the planes will produce forces that when linearized are: $Z_{d_{pl}}d_{pl}(t)$ and $M_{d_{pl}}d_{pl}(t)$. The dynamics of the vehicle are thus defined as:

$$u_r = U_0 \tag{9}$$

$$m\dot{w}_r = mU_0 q + (W - B)\cos q + Z_{\dot{w}_r}\dot{w}_r + Z_{w_r} w_r + Z_{\dot{q}}\dot{q} + Z_q q + Z_{d_{pl}} d_{pl}(t) \tag{10}$$

$$I_{yy}\dot{q} = (z_B B - z_G W)\sin q + M_{\dot{q}}\dot{q} + M_q q + M_{\dot{w}_r}\dot{w}_r + M_{w_r} w_r \tag{11}$$

$$\dot{q} = q \tag{12}$$

$$\dot{Z} = w_r \cos q - U_0 \sin q \tag{13}$$

Further assumptions of small pitch angle, therefore $\sin q \approx q$ and $\cos q \approx 1$, and small motions in the vertical plane results in a set of linerized equations that can be put in the in matrix form, $\boldsymbol{M}\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \boldsymbol{B}\mathrm{u}$:

$$\begin{pmatrix} (m - Z_{\dot{w}_r}) & -Z_{\dot{q}} & 0 & 0 \\ -Z_{\dot{w}_r} & (I_{yy} - M_{\dot{q}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \dot{w}_r \\ \dot{q} \\ \dot{q} \\ \dot{Z} \end{Bmatrix} = \begin{pmatrix} Z_{w_r} & (mU_o + Z_q) & 0 & 0 \\ M_{w_r} & M_q & (z_B B - z_G W) & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -U_o & 0 \end{pmatrix} \begin{Bmatrix} w_r \\ q \\ q \\ Z \end{Bmatrix} + \begin{Bmatrix} Z_{d_{pl}} \\ M_{d_{pl}} \\ 0 \\ 0 \end{Bmatrix} d_{pl}(t) \tag{14}$$

B. CURRENT ARIES CONTROL LAW FOR DIVING MODE

In 1993 Healey and Lienard proposed utilizing Multivariable Sliding Mode control for Diving and Steering control of AUV's. ARIES current diving mode controller is based on this concept and is presented in reference [4]. The diving controller is based on the linearized dynamics given in equation (14), however the heave velocity equation is ignored. This is primarily because there is no sensor onboard ARIES that directly can measure heave velocity. Since heave velocity affects are small they are handled as a disturbance to the system. This allows equation (14) to be reduced to a simpler third order model:

$$\begin{pmatrix} -Z_{\dot{w}_r} & (I_{yy} - M_{\dot{q}}) & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} \dot{q} \\ \dot{q} \\ \dot{Z} \end{Bmatrix} = \begin{pmatrix} M_{w_r} & M_q & (z_B B - z_G W) \\ 0 & 1 & 0 \\ 1 & 0 & -U_o \end{pmatrix} \begin{Bmatrix} q \\ q \\ Z \end{Bmatrix} + \begin{Bmatrix} M_{d_{pl}} \\ 0 \\ 0 \end{Bmatrix} d_{pl}(t) + \begin{Bmatrix} dM(t) \\ 0 \\ 0 \end{Bmatrix} \tag{15}$$

where $\|dM(t)\|$ is bounded.

For ARIES the control output $d_{pl}(t)$ is a single control output; however the command is sent to the bow and stern planes as equal and opposite signals.

In development of Sliding Mode Controller, a sliding surface is created from a linear combination of the state variable errors, ignoring any nonzero pitch angle and rate commands:

$$s(t) = s'(\mathbf{x} - \mathbf{x}_{com}) \tag{16}$$

Equation (15) is rewritten in the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ where $\mathbf{A} = \mathbf{M}^{-1}\mathbf{A}$ and $\mathbf{B} = \mathbf{M}^{-1}\mathbf{B}$. Now pole placement is utilized to obtain linear state feedback gains **k,** with at least one of the poles placed at zero. The closed loop dynamics matrix can then be calculated where $\mathbf{A_c} = \mathbf{A} - \mathbf{B}\mathbf{k}$ and the sliding surface polynomial (s) is found from the left eigenvector of $\mathbf{A_c}' s = 0$. The resulting control law is then obtained from:

$$d_{pl}(t) = -\mathbf{k}\mathbf{x} - (\mathbf{s}'\mathbf{B})^{-1} h\, sat\, \mathrm{sgn}(\frac{\mathbf{s}'\mathbf{x}}{f}) \tag{17}$$

The resulting controller design for ARIES obtained is the combination of equations (18) and (19) and can be found in reference [5].

$$s(t) = -0.7693(q_{com} - q) - 0.6385(q_{com} - q) + 0.0221(z - z_{com}) \tag{18}$$

where as previously mentioned q_{com}=0, and $q_{com} = 0$.

$$d_{pl}(t) = 0.4994\left(-0.4105q + 0.1086q + h\tanh(\frac{s(t)}{f})\right) \tag{19}$$

where $h = 1.0$ and $f = 0.5$.

C. SIMULATION RESULTS FOR SLIDING MODE CONTROL

The current depth controller is designed primarily to act as a depth regulator, where once a command for depth is received; its job is to approach the commanded depth with desired characteristics and stability. Figure 2 below shows the response of ARIES Depth Controller to a ten meter change in commanded depth.

This response is excellent when the mode of the depth controller is acting as regulator, responding to step changes in depth. The study for this thesis is more interested in tracking a depth command that changes continuously with time. Figure 3 shows the response of ARIES Depth Controller to a sinusoidal commanded depth of amplitude one meter and period of twenty seconds.

It can be seen there is a significant phase lag in the depth achieved and even an inability to match amplitude of the commanded signal. In all fairness, this is not a deficiency in the controller or sliding mode control in general. The controller in ARIES was designed for regulation and not tracking control. A sliding mode control could be developed and tuned to better handle this tracking control problem. This example will merely set a standard for which to compare later control law developments so qualitative and quantitative comparisons can be made.

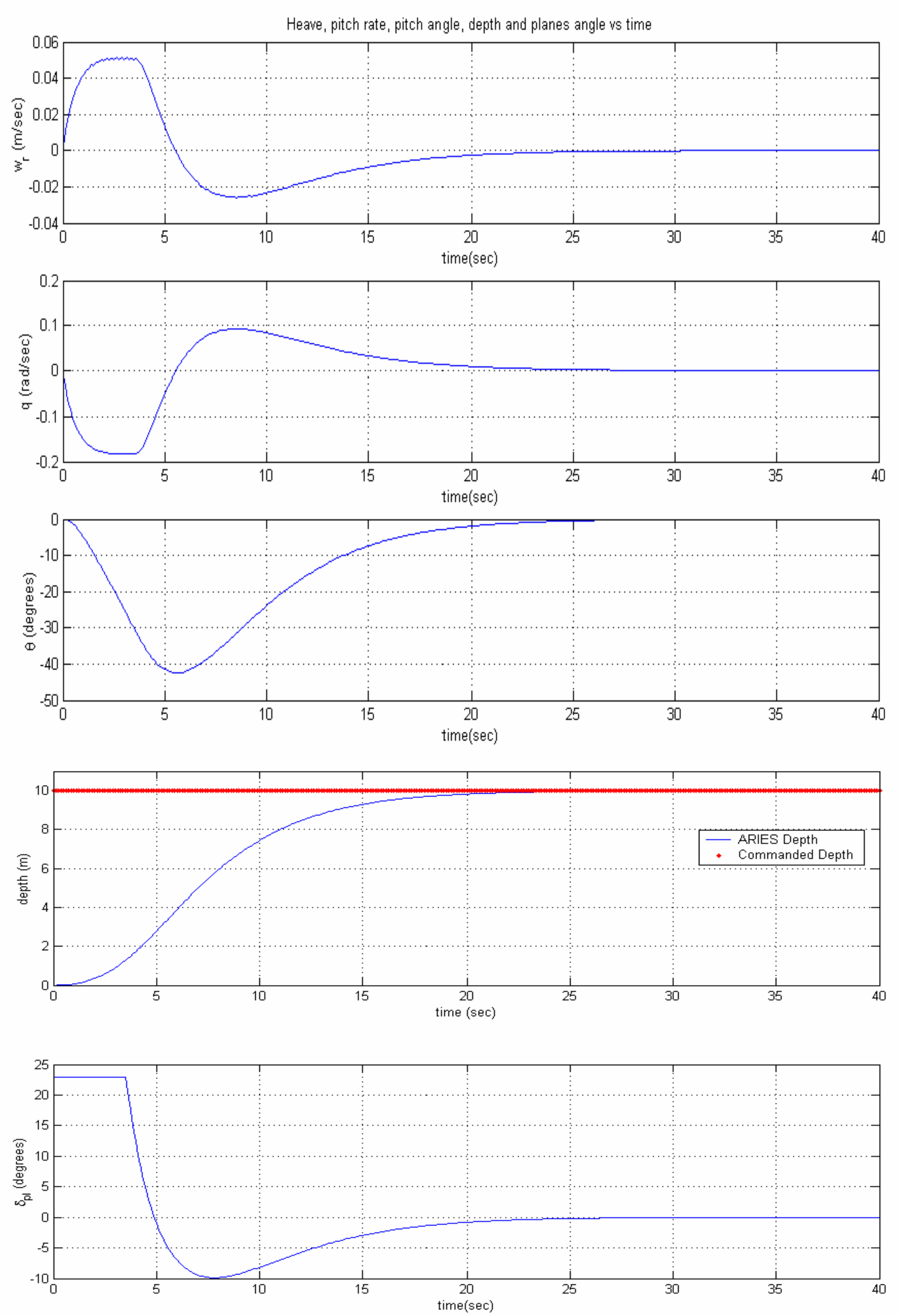

Figure 2 ARIES Depth Controller response to 10 m change in depth

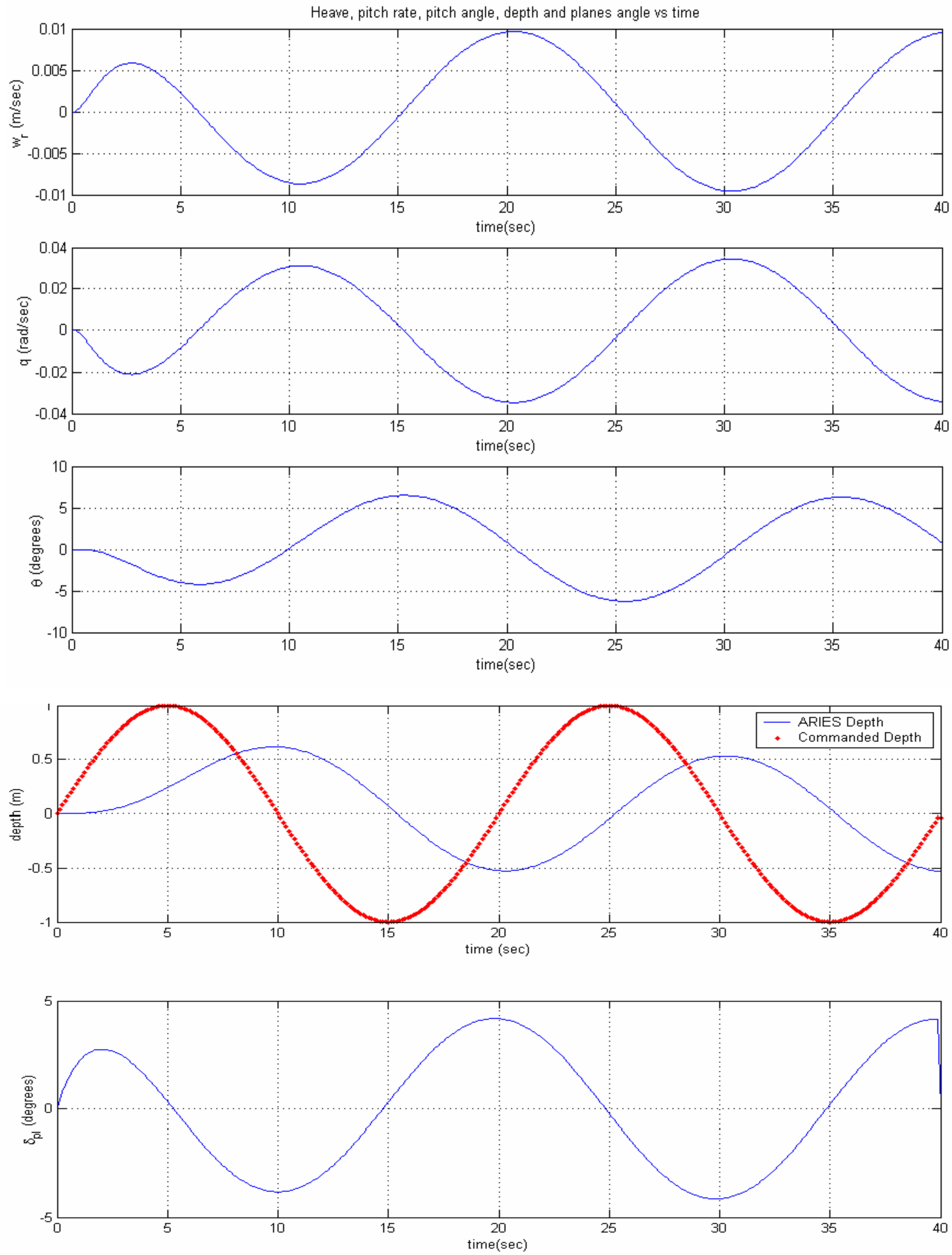

Figure 3 ARIES Depth controller response to sinusoidal depth command

III. TRACKING CONTROL

A. INTRODUCTION AND BACKGROUND

In controls there are two broad categories of control, regulation and tracking. For controls that are designed for regulation, the concept is to maintain a system parameter or series of parameters at a defined steady state value in the presence of disturbances and changes in overall system parameters. With regulation, a controller can often be designed, optimized and tested around a particular setpoint which allows for linearization assumptions, ample analysis of system stability, and ability for shortened testing of control schemes. For tracking control, the objective is to have a parameter or series of parameters track a given time varying input. In tracking control the precise input signal dynamics may or may not be know at the time of controller design. This leads to concerns that linearization assumptions may not be valid, testing over the broad range of possible input signal dynamics may be required, and a more complex control design is often required to achieve acceptable performance.

The design of a controller for an AUV that is to dock onto a cage system whose position varies with time can be approached as a tracking control problem. The goal is to have the AUV track the motion of the cage and match the cage's change in depth as it makes the final approach. Many model based approaches can be implemented successfully in order to solve this tracking control problem. In the next section the theory for an approach utilizing an error space control will be shown.

B. TRACKING A DYNAMIC SIGNAL WITH ZERO ERROR

1. Theory

The control design method presented in this section utilizes error space control to attain zero error tracking. Essential for implementation of this controller is a shift from the state space control of a system to an error space control. The error space is a coupled system including the dynamics of the error signal with the system dynamics. Below is the formulation for the control law that was inspired by an Integral Error Space approach presented in reference [6].

To start off, the system being controlled in state space is:

$$\dot{x}_s = [A_s]\, x_s + [B_s] u \text{ with } y = [C_s]\, x_s \tag{20}$$

The reference signal that is being tracked has the following dynamics:

$$\dot{x}_r = [A_r]\, x_r + [B_r] u_r \quad \text{with} \quad r = [C_r]\, x_r \tag{21}$$

in which the order of the reference dynamics is not the same as the system model, and u_r may be considered to be zero mean white noise.

The tracking error is then defined as:

$$e = y - r \tag{22}$$

The goal of the control law design is to design a compensator system that will compensate for the signal dynamics of the input and allow for near zero error tracking. One way to do this is to utilize a compensator system, which has the same dynamics as the input signal and is driven by the error between the output of the system and the input signal.

The compensator system will therefore have dynamics given by :

$$\dot{x}_c = [A_r]\, x_c + [B_r]\, e \tag{23}$$

A combined system is then developed such that:

$$\begin{Bmatrix} \dot{x}_c \\ \dot{x}_s \end{Bmatrix} = \begin{pmatrix} A_r & 0 \\ 0 & A_s \end{pmatrix} \begin{Bmatrix} x_c \\ x_s \end{Bmatrix} + \begin{pmatrix} B_r \\ B_s \end{pmatrix} \{ e \quad u \} \tag{24}$$

knowing that $e = [C_s]\, x_s - r$ the error space system can be rewritten as

$$\begin{Bmatrix} \dot{x}_c \\ \dot{x}_s \end{Bmatrix} = \begin{pmatrix} A_r & B_r C_s \\ 0 & A_s \end{pmatrix} \begin{Bmatrix} x_c \\ x_s \end{Bmatrix} + \begin{pmatrix} 0 \\ B_s \end{pmatrix} u + \begin{pmatrix} -B_r \\ 0 \end{pmatrix} r \tag{25}$$

As can be seen in equation (25), a cross coupled system is obtained, in which the system that is being controlled is being driven both by the input signal, r, and the control force, u.

Now pole placement or Linear Quadratic Regulator (LQR) methods can be utilized to stabilize the entire error space system with a feedback control law given by:

$$u = -\begin{bmatrix} K_c & K_s \end{bmatrix} \begin{Bmatrix} x_c \\ \tilde{x}_s \end{Bmatrix} \tag{26}$$

where $\tilde{x}_s$ is used for the system state feedback based on command error, i.e.

$$\tilde{x}_s = x_s - rC_s^T \tag{27}$$

therefore the closed loop system becomes:

$$\begin{Bmatrix} \dot{x}_c \\ \dot{x}_s \end{Bmatrix} = \begin{pmatrix} A_r & B_r C_s \\ -B_s K_r & A_s - B_s K_s \end{pmatrix} \begin{Bmatrix} x_c \\ x_s \end{Bmatrix} + \begin{pmatrix} -B_r \\ B_s K_s C_s^T \end{pmatrix} r \tag{28}$$

and is assumed to be fully controllable.

Note: The model reference portion is controllable in this case as opposed to other model reference tracking systems.

The new error space system can be written now as

$$\dot{z} = [E]z + [F]r \tag{29}$$

By designing an error space system that fits the model of equation (29) it can be seen that a system of desired characteristics can be obtained which is being driven by the input command signal, r.

The error dynamics are now given by equation (29) which is stable, with an error output equation given by:

$$e = \begin{bmatrix} 0 & C_s \end{bmatrix} z - r$$

It remains to show that output y tracks the input r as $t \to \infty$.

For the steady state portion of $z(t)$ and $r(t)$:

$$\underset{t \to \infty}{z(t)} = -E^{-1}F \cdot r(t) \tag{30}$$

$$\therefore \underset{t \to \infty}{e(t)} = \left\{ -\begin{bmatrix} 0 & C_s \end{bmatrix} E^{-1} F - I \right\} r(t) \tag{31}$$

It can be shown the error space controller as designed is able to achieve zero error tracking is due to $-\begin{bmatrix} 0 & C_s \end{bmatrix} E^{-1} F = I$

Figure 4 below shows the resulting MATLAB SIMULINK model of the controller. This shows in particular how the controller can be implemented. A compensator control input signal is developed based on the input error signal. The system is stabilized by control input obtained from the state feedback of the error states, i.e., the difference between the commanded states and the actual states of the system.

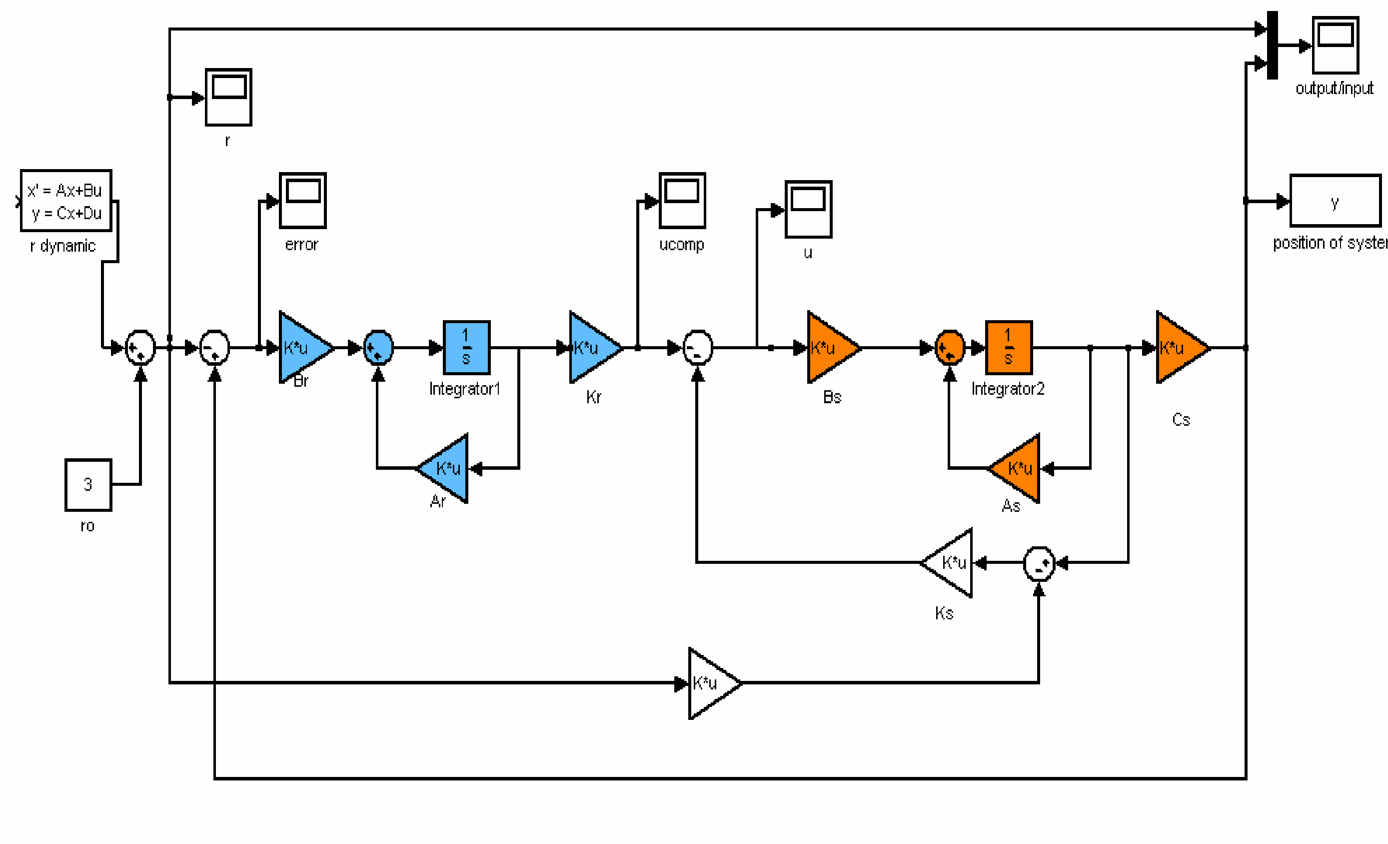

Figure 4 SIMULINK Model of an Error Space Controller

2. Evaluation

As a test platform for the Error Space Controller a simple sping-mass system was developed in which a mass with control force acting on it attempts to track a vibrating spring mass system as pictured in Figure 5. Figure 6 shows the resulting controller response to a second order reference signal from the modeled spring-mass system. As can be seen the controller is able to track the reference signal with zero error once the initial error between the system position and reference mass position is overcome. Appendix B has the MATLAB code used to model the system and design and simulate the controller and system response.

This controller was chosen as the focus for the thesis study because it offers tracking with zero error and displays the best tracking performance of all the controllers studied. This is a very robust design, but one that has limits when converted to discrete form that will be discussed. In addition, there are limitations on the gains that can be utilized when there is a limited control authority available.

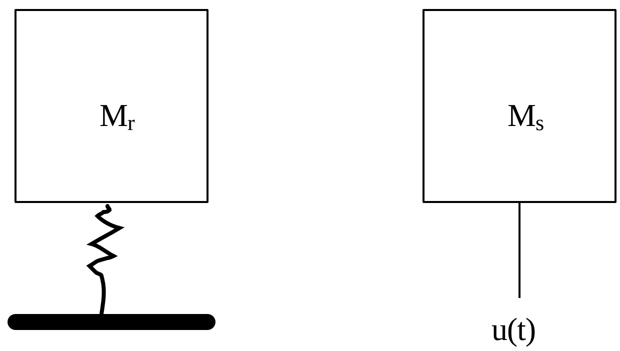

Figure 5 Second order system to be modeled and controlled by Error Space Controller

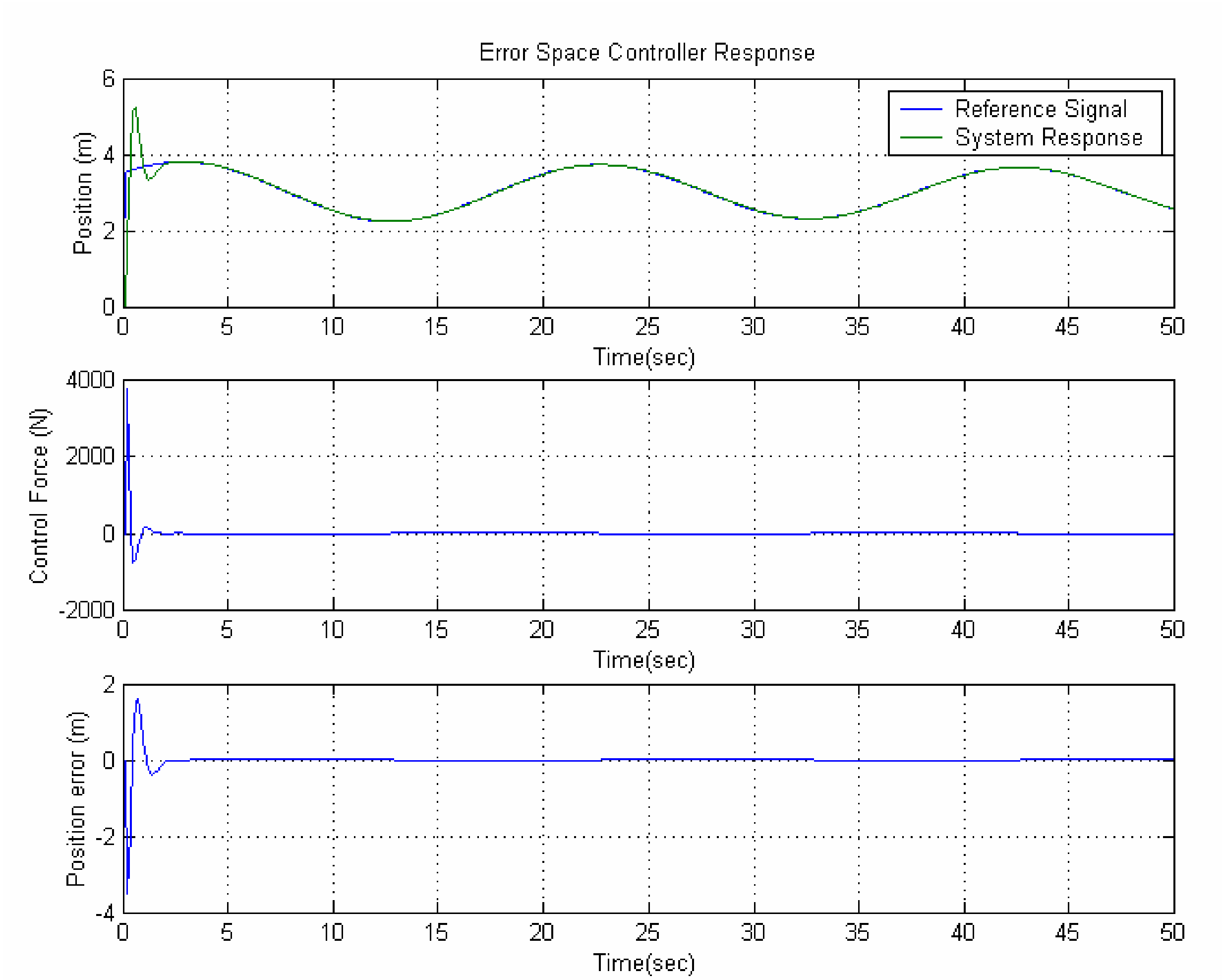

Figure 6 Error Space Controller Response

3. Robustness

As with the many other control laws this method relies heavily on the knowledge of the model of the signal being tracked as well as a model of the system being controlled. This modeling requirement can be a limitation for the controller depending on the ability to accurately model both systems. This particular design however, is robust in that it can handle errors in modeling the dynamics of the input signal that is being tracked.

As a test of the robustness, performance of the controller was examined when errors in modeling the reference signal were present. For this case, once the gains for the controller were chosen based on an assumed model of the reference signal, the controller was able to track a signal of different set of model parameters with little degradation in tracking performance. Figure 7 shows the results with 100% error in the modeling of the coefficients of the reference signal, and shows the tracking error to be within 4% the maximum amplitude of motion.

The controller has the same robustness properties as any LQR state feedback controller having infinite gain margin and is tolerable to signals that are off-design frequency.

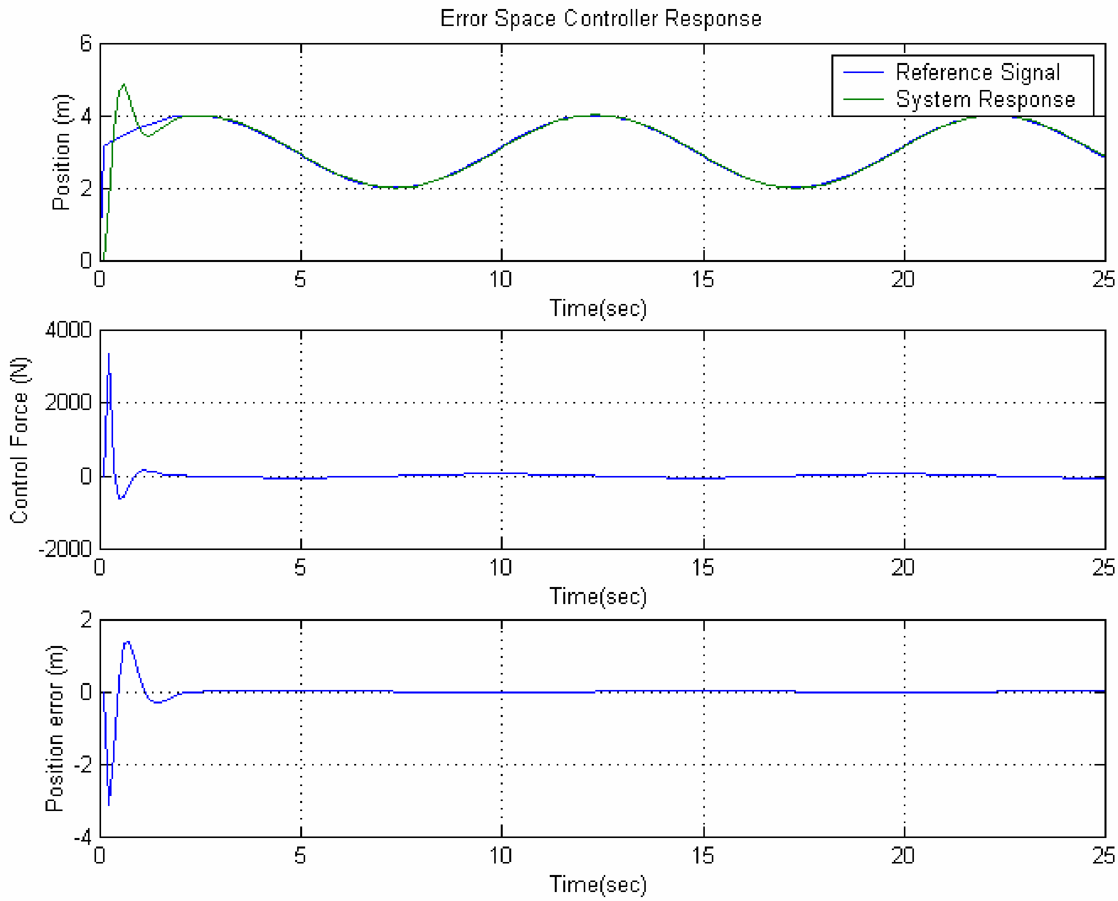

Figure 7 Error Space Controller Response with errors in modeling the reference signal

A look at Figure 8, the bode plot of the error compared to reference signal, shows the controller is indeed robust. For the given example the controller is designed for a 0.316 rad/sec signal which is where the peak of -80.6 dB (0.009% error) occurs. At 0.1 rad/sec the magnitude of error compared to signal is -51.6 dB (0.26% error), and out to 1 rad/sec , -35.9 dB (1.6% error).

An improvement in controller design could be made by making the controller an adaptive controller. If the coefficients of signal dynamics matrix, A_r, are determined recursively utilizing a least squares method, the controller could adapt to the precise signal dynamics [7]. This would allow for a large range of signal input as well as time varying signal dynamics (i.e. $A_r(t)$). The cost of this of course is a more complex controller design, and will not be studied in this thesis, but is recommended for future studies.

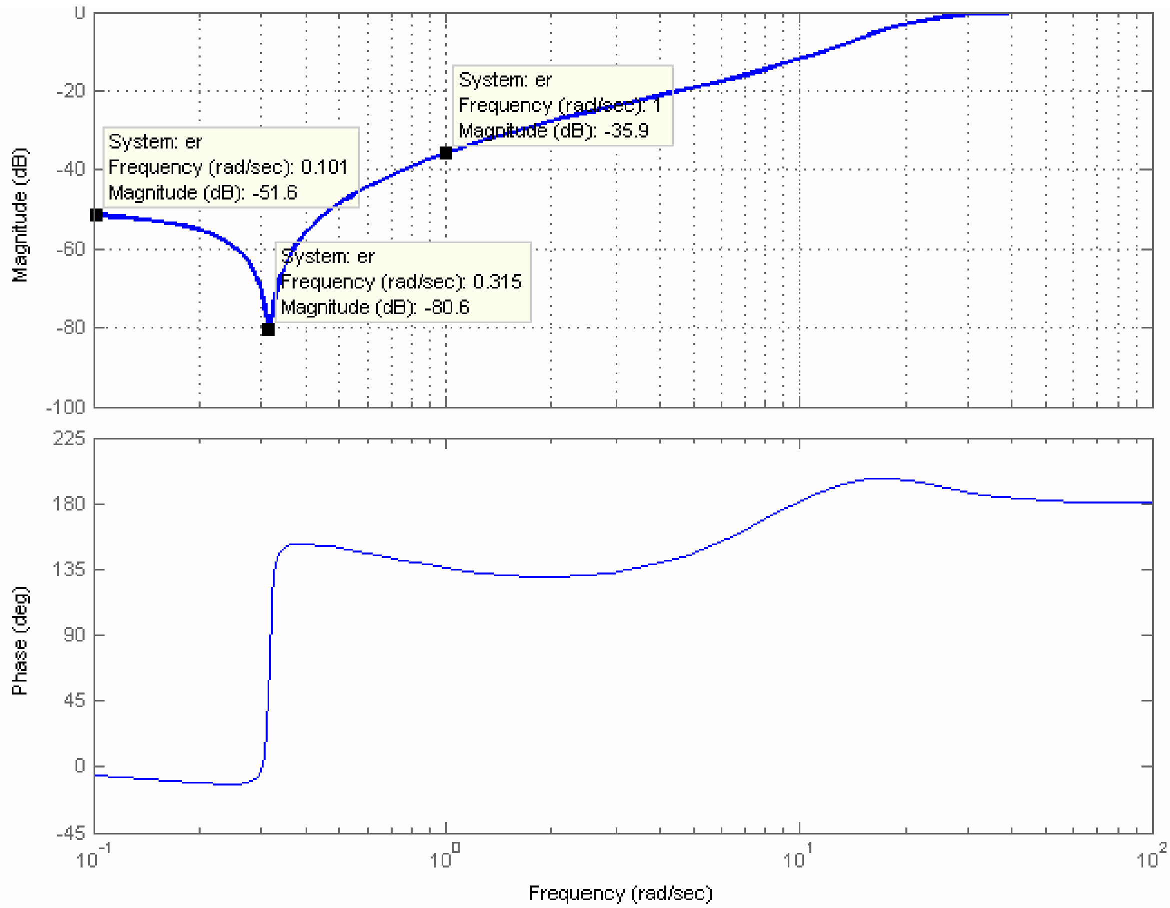

Figure 8 Bode plot of error to reference signal (e/r)

4. Stability

The stability of a controller over a wide range of operating conditions is a desirable trait. While this design shows great robustness there are a few areas that need to be carefully considered when designing the controller, to assure stability of the controller and system.

a. Saturation

One important design point to carefully consider is the initial control force required to get the system tracking. This can be large in the presence of large position error, and if saturation of the controller occurs, due to limits in the control authority available, the result can be unstable control. This can be tuned in the controller by lowering the frequency at which the poles of the system are placed, yielding smaller gains and therefore smaller initial control force commanded. However, decreasing gains will decrease the tracking capability and require a longer time period to bring the tracking error to zero. Another option is to use a different controller to get the system "close" and then allow the Error Space Controller to do the tracking of the time varying signal.

b. Discrete Controller Form

As a continuous controller this design is very robust. However, when converting the controller to a discrete form there are some additional limitations. There is a relationship between the discrete time step used and the frequency at which the poles of the system can be placed. As shown in discrete control theory, the highest system frequency should be *at most*, one half the sampling frequency [8]. Since the assumption of an 8 Hz sampling frequency (as is found in ARIES) was used, the discrete time steps will be 0.125 sec, and therefore, the controller design has a limited frequency at which the system poles can be placed. The highest pole that can be placed is estimated to be $w = \frac{1}{2}\left(\frac{2p}{T}\right) = \frac{1}{2}\left(\frac{2p}{0.125\sec}\right) = 25.2 rad/\sec$. As discovered when implementing this controller design, the actual limitation appears to be 1/6th of the sampling frequency or around four rad/sec for our example case.

5. Tunability

The gains of this controller can easily be adjusted by moving the poles selected for pole placement. The higher the frequency of the poles, the faster the error will go to zero (faster settling time) and the larger the intial overshoot. All this comes at the cost of higher gains, which yield large initial control forces that may be unrealistic for any available actuators. In addition these large gains have an amplifying effect on any noise found in the system.

6. Conclusion

This controller proves to be very capable of tracking a dynamic signal with near zero error. Application of its use in an AUV tracking a dynamic cage system will be discussed in the Chapter IV.

IV. DESIGN OF AN ERROR SPACE TRACKING CONTROLLER FOR ARIES

A. MODELING ARIES

Many controllers use model based techniques to develop the control law. The error space controller relies heavily on the knowledge of both the system parameters of the vehicle being controlled, as well of the parameters of the object that is being tracked. This section will show a method in which a combination of least squares and first principles can be utilized to create a reasonable model for the vehicle and its parameters for utilization in the control law design.

1. Vehicle Model

As previously discussed in Chapter II, a series of differential equations can be formulated to form a state space model of an AUV in the diving mode.

Given the following equations of motion for heave and pitch

<u>Heave</u>

$$m\dot{w} = mU_o q + (W - B)\cos\theta + Z_{\dot{w}}\dot{w} + Z_w w + Z_{\dot{q}}\dot{q} + Z_q q + Z_{\delta}\delta_{pl} \tag{32}$$

<u>Pitch</u>

$$I_{yy}\dot{q} = (z_B B - z_G W)\sin\theta + M_{\dot{q}}\dot{q} + M_q q + M_{\dot{w}}\dot{w} + M_w w + M_{pl}\delta_{pl} \tag{33}$$

$$\dot{\theta} = q \tag{34}$$

<u>Depth</u>

$$\dot{z} = w_r \cos\theta - U_0 \sin\theta \tag{35}$$

The following simplifications are made:

$\sin\theta \approx \theta$, $\cos\theta \approx 1$, W-B is small, and the cross coupling terms ($\dot{q}$ in heave, and $\dot{w}$ in pitch) are ignored.

As a result the system can be put in state space form:

$$\begin{pmatrix} (m-Z_{\dot{w}_r}) & -Z_{\dot{q}} & 0 & 0 \\ -Z_{\dot{w}_r} & (I_{yy}-M_{\dot{q}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \dot{w}_r \\ \dot{q} \\ \dot{q} \\ \dot{Z} \end{Bmatrix} = \begin{pmatrix} Z_{w_r} & (mU_o+Z_q) & 0 & 0 \\ M_{w_r} & M_q & (z_B B - z_G W) & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -U_o & 0 \end{pmatrix} \begin{Bmatrix} w_r \\ q \\ q \\ Z \end{Bmatrix} + \begin{Bmatrix} Z_{d_{pl}} \\ M_{d_{pl}} \\ 0 \\ 0 \end{Bmatrix} d_{pl}(t) \quad (36)$$

B. PARAMETER IDENTIFICATION

Parameter or System Identification is a topic that is treated in many areas of controls; in particular controls that utilize the system model parameters to formulate the control law. One such technique is an adaptive control that utilizes recursive parameter identification as part of the control law development [7]. For the error space controller, a good estimate of the parameters of the vehicle only needs to be made once and need not be part of a recursive algorithm. The parameters can be estimated from calculation of first principles of dynamics of marine vehicles, from statistical least squares estimation of experimental data, or a combination of the two. The identification of the parameters of the item being tracked can be approached in many different ways. The parameters can be determined once, from a model of the expected input signal. It also could be done adaptively utilizing a recursive method to identify the parameters of the incoming signal. The focus of the discussion for this section will be on the AUV parameter identification methods. The input signal dynamics will be assumed to be known at the time of controller design. Further studies into the identification of the input signal dynamics parameters are recommended for future studies.

1. Least Squares Estimation

The heave and pitch equations (32), (33) can be cast into form: $y(t) = H(t)\Theta(t)$.

Where **y**(t) is the next time step state history for the equation of interest, **H**(t) is a matrix containing a time history of the states affecting a particular equation of motion and Θ is a vector of the yet to be determined parameters in the equation of motion.

For heave:

Utilizing an Euler transformation

$\dot{w} = \dfrac{w(t+1) - w(t)}{\Delta t}$, therefore equation (32) can be written

$$w(t+1)=\left(\frac{Z_w \Delta t}{M_{totw}}+1\right)w+\left(\frac{(mU+Z_q)\Delta t}{M_{totw}}\right)q+\left(\frac{Z_{d\,pl}\Delta t}{M_{totw}}\right)d_{pl} \tag{37}$$

where $M_{totw}=m-Z_{\dot{w}}$

Therefore $y(t)=\begin{bmatrix} w_{t+1} \\ w_t \\ w_{t-1} \\ \vdots \\ w_{t-n+1} \end{bmatrix}$, $H(t)=\begin{bmatrix} w_t & q_t & d_t \\ w_{t-1} & q_{t-1} & d_{t-1} \\ w_{t-2} & q_{t-2} & d_{t-2} \\ \vdots & \vdots & \vdots \\ w_{t-n} & q_{t-n} & d_{t-n} \end{bmatrix}$ and $\Theta=\begin{bmatrix} \left(\frac{Z_w \Delta t}{M_{totw}}+1\right) \\ \left(\frac{(mU+Z_q)\Delta t}{M_{totw}}\right) \\ \left(\frac{Z_{d\,pl}\Delta t}{M_{totw}}\right) \end{bmatrix}$

where n is the batch length.

Using a least squares method to estimate the parameters Θ [9], an error in output estimate is defined as:

$$e(t)=(y(t)-H(t)\hat{\Theta}(t)) \tag{38}$$

where $\hat{\Theta}(t)$ is the estimate of $\Theta(t)$.

In order to minimize the error, a scalar positive squared error measure is defined,

$$J(n)=\sum_{t=1}^{n} e^{'}(t)e(t) \tag{39}$$

and then the minimization of J is given by:

$$\frac{dJ}{d\hat{\Theta}}=0=-\sum_{t=1}^{n} H^{'}(t)e(t)$$

and substituting $y(t)=H(t)\Theta(t)$ produces:

$$0=-\sum_{t=1}^{n} H^{'}(t)(y(t)-H(t)\hat{\Theta}(t))$$

Rearranging this into matrix form and solving for $\hat{\Theta}(t)$ gives:

$$\hat{\Theta}=[H^{'}H]^{-1}H^{'}y \tag{40}$$

The known states in equation (37) are w,q,d_{pl} which can be obtained from data collected from onboard sensors and planes output during an experimental run. The parameters Z_w, Z_q, and Z_{pl} are the parameters to be estimated. The mass (m), added mass in heave($Z_{\dot{w}}$), are known or calculated properties of the vehicle, and the Δt is the discrete processing time of the sensor data available. For ARIES the discrete sampling time is based on the 8 Hz sampling frequency of the sensors yielding a Δt = 0.125 sec. The mass of ARIES is 222 kg and the added mass in heave is estimated to be 234 kg [10].

An experiment can be run which excites the heave mode of motion by making continuous depth changing maneuvers. It should be noted that the regression matrix, $[\sum_{t=1}^{n} H^{'}(t)H(t)]$, must be positive and strong (with no singularity) otherwise its inverse does not exist. This means that the system must be perpetually excited by its input. From the data obtained from onboard sensors $\hat{\Theta}$ is estimated from equation (40) and from this estimate the unknown parameters can be calculated.

$$Z_w = \frac{(\Theta_{(1)} - 1)M_{totw}}{\Delta t} \tag{41}$$

$$Z_q = \frac{\Theta_{(2)}M_{totw}}{\Delta t} - mU \tag{42}$$

$$Z_{pl} = \frac{\Theta_{(3)}M_{totw}}{\Delta t} \tag{43}$$

At this point the parameters that will be estimated will minimize the least squares error for the data obtained during the experiment. The parameters that result may not necessarily be the actual parameters for the vehicle but should be a good estimate. Errors or simplifications in modeling the equations of motion, errors in estimates of parameters like added mass, all can lead to errors in the parameter identification. Astrom [7] recommends providing the parameter estimation as many know parameters as possible to allow for the most realistic model to be identified. For the case of an AUV, a reasonable estimation of the planes coefficients can be made which help to obtain good parameter estimation.

a. First Principles Calculation

The force in the z direction due to the planes comes from the lift force provided by each of the planes.

The lift force coefficient for one plane is:

$$Z_{1\,plane} = \frac{1}{2} r U^2 A_{fin} C_L \tag{44}$$

For ARIES , A_{fin}=0.02163 m^2, $C_L = \frac{1.2}{0.4 rad}$, U was taken to be 1.41 m/s.

Since on ARIES the bow and stern planes act in opposite directions, the lift forces act in opposite directions. Also an assumption is made that the bow planes are about 80% as effective as the stern planes. Therefore :

$$Z_{d\,pl} = 2Z_{1\,plane} - 2(0.8Z_{1\,plane}) = 0.4Z_{1plane} \tag{45}$$

$$Z_{d\,pl} = 0.4\left[\frac{1}{2}(1024\frac{kg}{m^3})(1.4\frac{m}{s})^2(0.02163m^2)\frac{1.2}{0.4rad}\right] = 26.1\frac{N}{rad}$$

b. Least Squares Estimate with First Principles Parameter

Now the number of parameters that are estimated in the heave equation can be reduced by one. Since the remaining parameters Z_w and Z_q are not easily or accurately calculated from first principles, the least squares parameter identification is utilized where:

$$y(t) = w(t+1) - \frac{Z_{pl}\Delta t}{M_{totw}} d_{pl}\ , H(t) = [w(t) \quad q(t)] \text{ and } \Theta(t) = \begin{bmatrix} \left(\frac{Z_w \Delta t}{M_{totw}} + 1\right) \\ \left(\frac{(mU + Z_q)\Delta t}{M_{totw}}\right) \end{bmatrix} \tag{46}$$

and the parameters are calculated from equations (41) and (42).

c. *Pitch Equation Parameter Identification*

The same procedure of least squares estimation with first principle parameter calculation can then be done for the pitch equation (33).

The known parameters for ARIES are I_{yy}=119.1 kg m^2 and M_q=93.13 kg m^2 [10].

The parameter $M_q = z_B B - z_G W$ can be easily estimated since the ARIES body is symmetric, the center of buoyancy located at the center therefore z_B=0. The center of gravity is estimated to be about 0.5 in below the center of the body therefore

$$M_\Theta = -0.5in \cdot \frac{0.026m}{in} \cdot 222kg \cdot 9.81\frac{m}{s^2} = -28.3Nm$$

First principles are used to calculate $M_{1plane} = \frac{1}{2} rU^2 A_{fin} C_L L$ where L is the distance from the fin to the center of rotation. The center of rotation was taken to be the midpoint between the bow planes and the stern planes. For the planes, since they act in opposite direction, they provide the same direction moment and therefore

$$M_{d\,pl} = 2M_{1\,plane} + 2(0.8M_{1plane}) = 3.6M_{1\,plane} \tag{47}$$

$$M_{d\,pl} = 3.6\left[\frac{1}{2}(1024\frac{kg}{m^3})(1.4\frac{m}{s})^2(0.02163m^2)\frac{1.2}{0.4rad}1.013m\right] = 237.9\frac{Nm}{rad}$$

The pitch equation is now written as:

$$q(t+1) = \left(\frac{M_w \Delta t}{M_{totp}}\right)w + \left(\frac{M_q \Delta t}{M_{totp}} + 1\right)q + \left(\frac{M_q \Delta t}{M_{totp}}\right)q + \left(\frac{M_{d\,pl} \Delta t}{M_{totp}}\right)d_{pl} \tag{48}$$

Now the first principles parameters are included:

$$y(t) = q(t+1) - \frac{M_q \Delta t}{M_{totp}}q(t) - \frac{M_{pl} \Delta t}{M_{totp}}d_{pl}(t) \;, H(t) = \left[w(t) \quad q(t)\right], \Theta(t) = \begin{bmatrix} \left(\frac{M_w \Delta t}{M_{totp}}\right) \\ \left(\frac{M_q \Delta t}{M_{totp}} + 1\right) \end{bmatrix} \tag{49}$$

2. Parameter Identification Results

An experimental run was conducted with ARIES. A commanded depth change between 4 and 10 meters was utilized. The run was designed to have near continuous changing of all the states to ensure that the batch least squares estimation will be sufficiently excited to allow for good parameter identification. Figure 9 shows the results which include the data collected from the run, and simulation of the model utilizing the parameters identified from the experimental data.

It can be seen that the data for heave has a great deal of noise. This is because there is no sensor that directly measures the heave velocity. An onboard acoustic doppler sensor measures depth rate ($\dot{z}$), and a gyro measures pitch angle (q). From these two sensor data streams, heave velocity is calculated from $w = \dfrac{\dot{z} - U\sin q}{\cos q}$ and therefore has a combination of the noise from two different sensors. The noise is not a problem for the parameter identification as long as the noise is zero mean white noise. All sensor data was de-meaned prior to least squares estimation to ensure zero mean data.

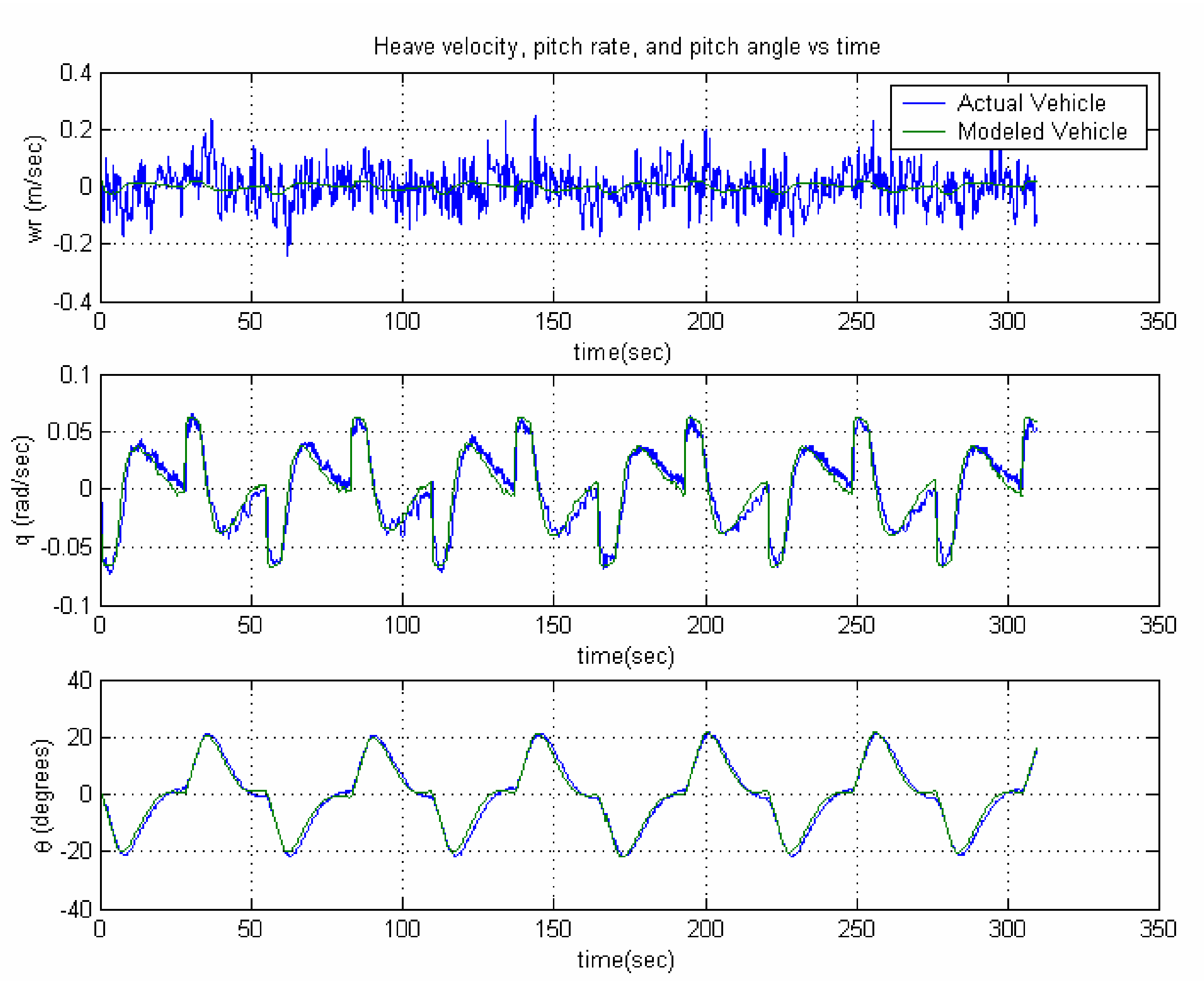

Figure 9 Parameter Identification data and results

A summary of all the parameters, known, calculated and estimated are shown in Table 1 and Table 2 below.

Known/Calculated Parameters	**m (kg)**	**Zw_dot (kg)**	**U(m/s)**	**dt (sec)**
	222	234	1.41	0.125
First Principles Calculations	**Zdpl (N/rad)**			
	26.1			
Least Squares Estimation	**Zw (Nsec/m)**	**Zq(Nsec)**		
Identified using Parameter ID	-764	120		

Table 1 Heave motion parameter results

Known/Calculated Parameters	**Iyy (kg m^2)**	**Mq_dot (kg m^2)**	**U(m/s)**	**dt (sec)**
	119.1	93.3	1.41	0.125
First Principles Calculations	**Mdpl (Nm/rad)**	**Mtheta(Nm)**		
	237.5	-28.3		
Least Squares Estimation	**Mw (Nsec)**	**Mq(Nmsec)**		
Identified using Parameter ID	45.5	-1147.8		

Table 2 Pitch motion parameter results

C. THE TRACKING CONTROLLER

Now that a model with all of the parameters for ARIES has been obtained, all that remains is the model of the signal dynamics. For the remainder of this thesis study the model of the signal is assumed to be a known second order signal with known frequency. As mentioned previously the signal can be and is most likely more complex. Analysis of the signal dynamics of the cage is recommended for future studies. A step by step solution to the design of an Error Space Controller to be implemented in ARIES is as follows:

1. Signal Dynamics

Since the cage motion is modeled as a simple sinusoid the resulting dynamics matrix becomes:

$$A_r = \begin{pmatrix} 1 & 0 \\ -w^2 & 0 \end{pmatrix} \tag{50}$$

where, for the model, a period of 20 seconds was chosen as a nominal value about which to test the controller. The resulting frequency is 0.3142 rad/sec therefore :

$$A_r = \begin{pmatrix} 1 & 0 \\ -0.0987 & 0 \end{pmatrix}$$

2. ARIES Dynamics

As previously shown ARIES dynamics are given by equation (36). Placing this equation in the form $\dot{\mathbf{x}}_s = \mathbf{A}_s\mathbf{x}_s + \mathbf{B}_s u$ where $\mathbf{A}_s = \mathbf{M}^{-1}\mathbf{A}$ and $\mathbf{B}_s = \mathbf{M}^{-1}\mathbf{B}$ and utilizing the coefficients calculated from first principles and parameter identification in Chapter III, ARIES dynamics and control force matrices become:

$$A_s = \begin{pmatrix} -2.0466 & 3.2363 & 0.0421 & 0 \\ 1.3892 & -8.5590 & -0.1575 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1.4100 & 0 \end{pmatrix} \tag{51}$$

$$B_s = \begin{pmatrix} 0.4207 \\ -1.3606 \\ 0 \\ 0 \end{pmatrix} \tag{52}$$

These matrices are utilized in the simulation of the response of ARIES in the diving mode.

For design of the controller the heave mode is ignored. As previously discussed, this is because of the uncertainity of ability to measure or estimate the heave velocity. The controller proves to be still valid with this assumption. Therefore the model becomes:

$$A_{sm} = \begin{pmatrix} -6.7012 & -0.1333 & 0 \\ 1 & 0 & 0 \\ 0 & -1.4100 & 0 \end{pmatrix} \tag{53}$$

$$B_{sm} = \begin{pmatrix} -1.1191 \\ 0 \\ 0 \end{pmatrix} \tag{54}$$

It is these matrices that will be utilized in the controller design.

3. Compensator Design

Now the error space matrices are formed such that

$$\begin{Bmatrix} \dot{x}_c \\ \dot{x}_s \end{Bmatrix} = \begin{pmatrix} A_r & B_r C_s \\ 0 & A_s \end{pmatrix} \begin{Bmatrix} x_c \\ x_s \end{Bmatrix} + \begin{pmatrix} 0 \\ B_s \end{pmatrix} u + \begin{pmatrix} -B_r \\ 0 \end{pmatrix} r$$

$$\text{where} \quad E = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -0.0987 & 0 & 0 & 0 & 1 \\ 0 & 0 & -6.7012 & -0.133 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1.41 & 0 \end{pmatrix}, \; F = \begin{pmatrix} 0 \\ 0 \\ -1.1191 \\ 0 \\ 0 \end{pmatrix}$$

The poles are then placed at :

$$poles = \begin{pmatrix} -0.4400 & -0.4620 & -0.4730 & -0.4840 & -0.4950 \end{pmatrix}$$

Pole selection was determined through iterative tuning process in which the higher the frequency the poles were placed the better the tracking performance. However, at a certain point in the iterative process placing the poles any higher yielded plane saturation that did not allow for stable tracking of the signal.

The resulting gains chosen are:

$$k = \begin{pmatrix} -0.0361 & 0.0229 & 3.8847 & -1.7725 & 0.5133 \end{pmatrix}$$

4. C code Implementation in ARIES

The resulting design of the Error Space Controller was implemented as another mode of the Flight Depth Controller in ARIES. The originally designed Sliding Mode Controller would be kept as the mode for achieving normal depth regulation and changes in ordered depth. The new mode could then be implemented as a change in the mode of the Flight Depth Controller in which this new mode of the controller is used to track a dynamic signal utilizing the error space design.

The following is general description how the code is implemented inside the vehicle. The actual code for ARIES Flight Depth Controller is found in Appendix E.

First the depth error is calculated from the commanded depth, which theoretically comes from the current dynamic depth state of the cage, but for experimental validation a sinusoidal signal of given amplitude and period about a given mean depth was utilized. The Depth is found as measured by ARIES on board sensor:

Deptherror = Depth - Depth_com

Then the compensator input to the control is calculated from:

ucomp = xcom1

where xcom1 is the first state of the compensator.

Then the control input from state feedback is calculated from:

ufb = - 3.8847*q + 1.7725*theta - 0.5133*Deptherror

note : that the default commanded pitch rate and pitch angle are zero, but the depth state is feedback from Deptherror to allow for tracking about a commanded depth not a zero depth.

The resulting input to the planes is then:

delta_sp = ucomp + ufb

Last the compensator states are updated:

xcom1(i+1) = 0.9992*xcom1 + 0.1250*xcom2 - 0.0026*Deptherror

xcom2(i+1) = -0.0123*xcom1 + 0.9992*xcom2 + 0.0045*Deptherror

note: in actual implementation of this controller it is important not to allow the compensator to update its state from the zero state until you are within a predetermined range of the commanded depth. This is because if there is a large intial depth error the integration of that error will cause the compensator states to grow too large. Results of not limiting the conditions when the compensator is activated will be shown and discussed in Chapter V.

The resulting controller design and simulation code are found APPENDIX D and simulation results are shown and discussed in Chapter V.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

A. SIMULATION OF ARIES VEHICLE RESPONSE TO SINUSOIDAL TRACKING SIGNAL

The resulting controller and full state vehicle response can be simulated under various intial conditions, and with differing input signal characteristics. The controller proves to be fairly robust, and able to track a signal with zero error. In an effort to understand the controller's strengths and weaknesses, a number of cases are presented below that highlight the Error Space Controller's features.

The resulting vehicle response to tracking a half meter amplitude sine wave of period 20 seconds is shown in Figure 10 and Figure 11 .

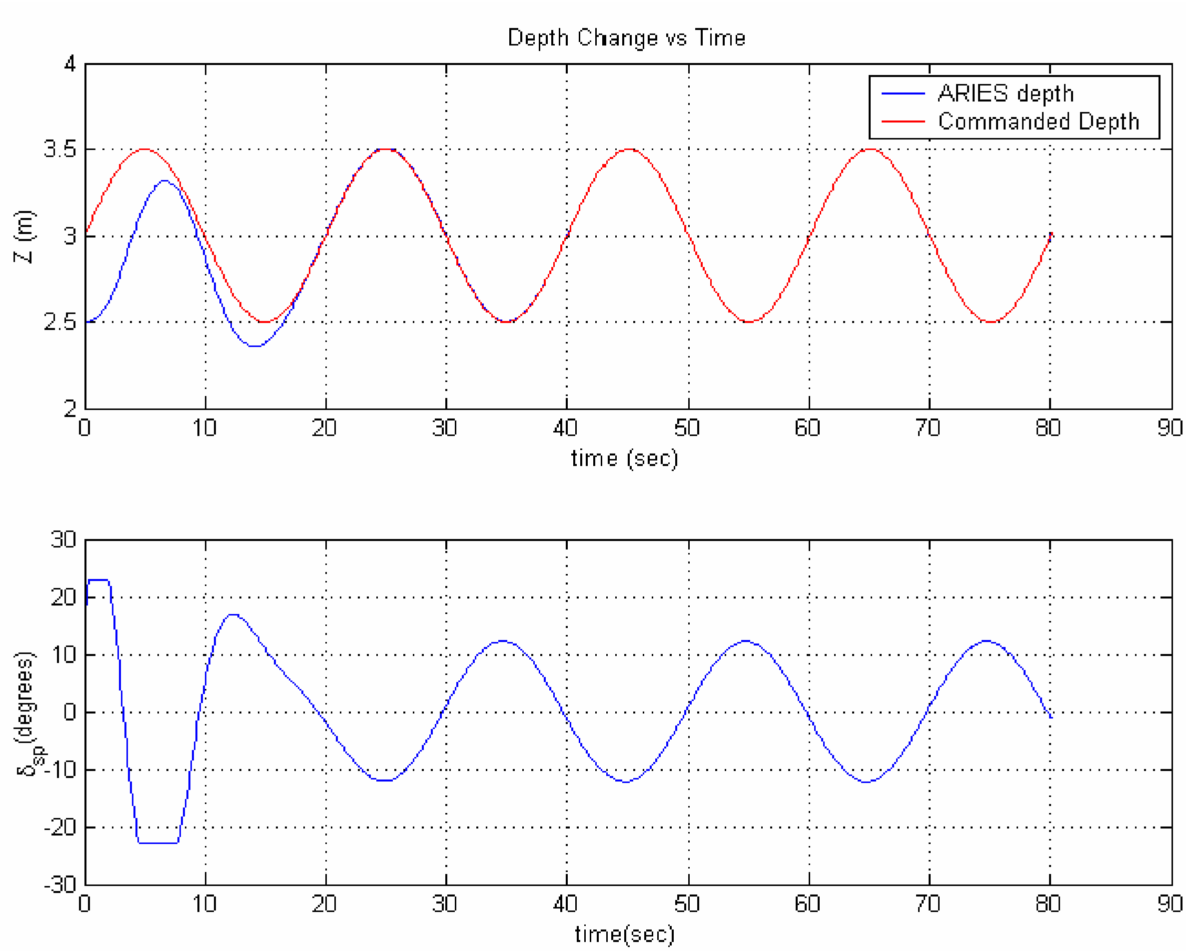

Figure 10 Error Space Control of ARIES with input signal T=20 secs, Amp=0.5m

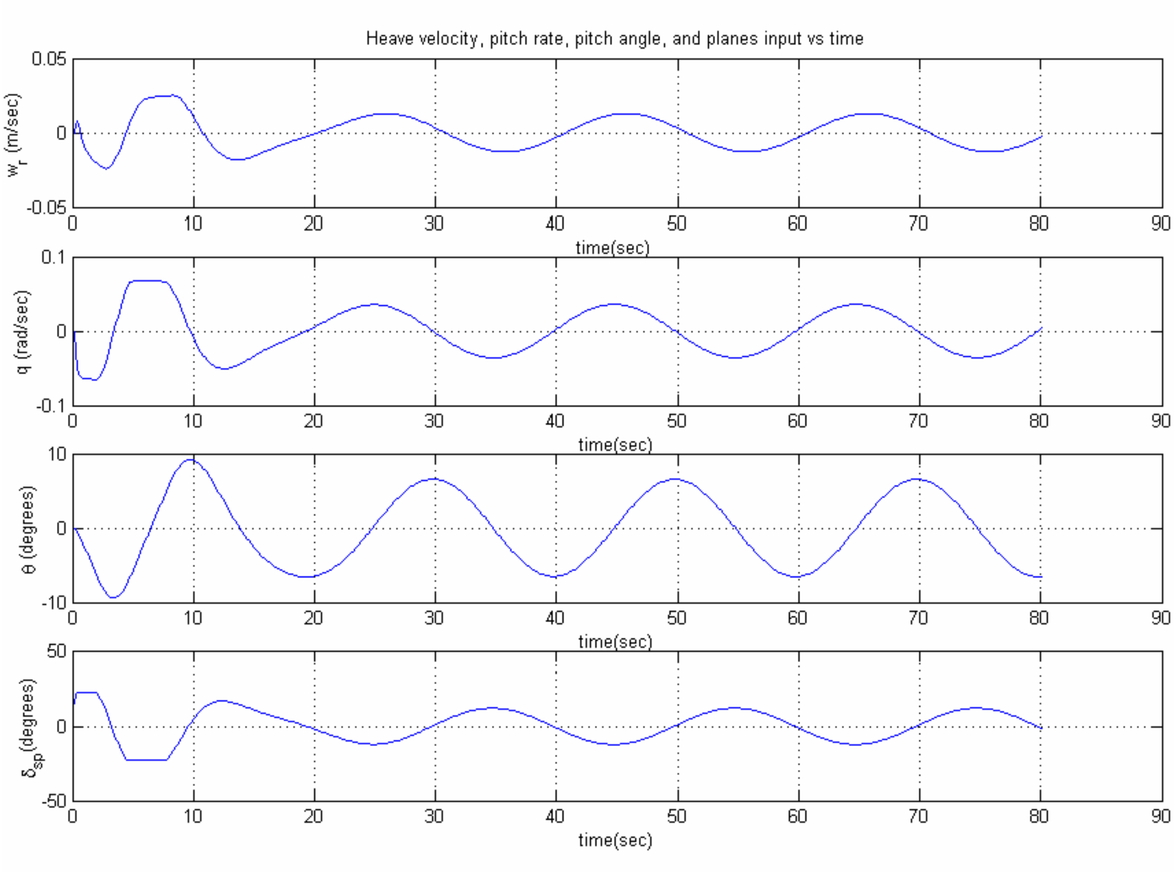

Figure 11 ARIES state responses to tracking a sinusoidal wave

As is shown in Figure 10, ARIES is able to track a half-meter amplitude signal of period twenty seconds with zero error. The intial conditions for this case show a half-meter position error between the depth of ARIES and the commanded depth. The depth error intial condition is an important requirement to be aware of to ensure the controller will have the ability to show accurate tracking. The compensator in the Error Space Controller should not be started until the vehicle is within a specified depth error band. It was found through iterative simulation that for ARIES, this depth band was around one meter. The initial depth error limitation is handled by turning on the compensator only after ARIES is within the one-meter depth error band. The result is the controller is simply a state feedback controller that will stabilize the vehicle about the ordered depth until the ARIES is within the depth error band at which time the compensator portion of the controller is turned on. Figure 12 shows the controller and vehicle response when the vehicle position starts at zero meters, and the sinusoidal command is the same as the previous example.

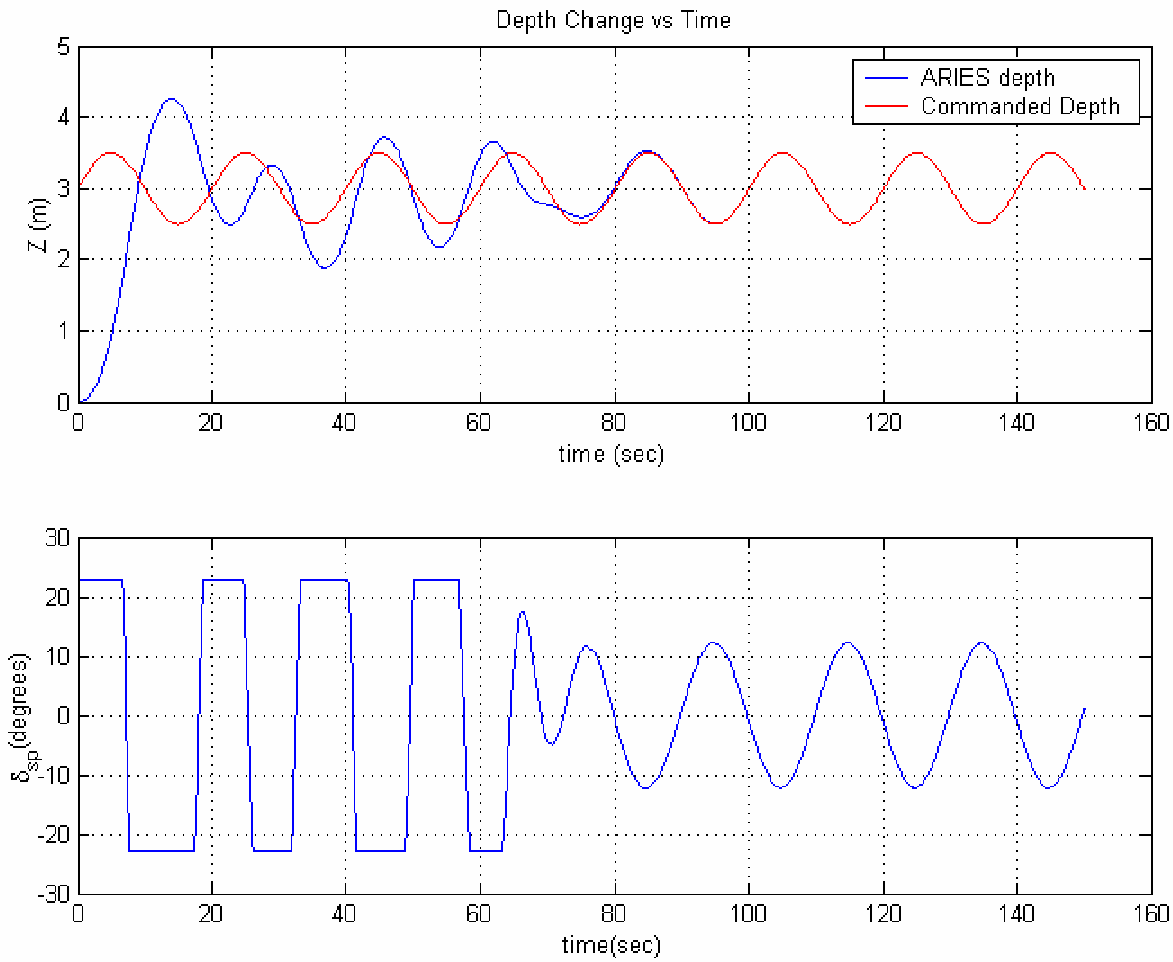

Figure 12 ARIES response to large intial depth error

As can be seen in Figure 12, the controller still has the ability to track the signal with zero error. In this case it takes nearly 80 seconds to reach good tracking of the signal vice the 20 seconds it took when the vehicle started only a half meters off the commanded signal. It it obvious though that this is mainly due to the intial planes saturation period that occurs when the controller is trying to achieve the large depth change manuever. This shows the stability of the controller design even to large errors in depth. The controller's performance however is still not ideal, in that it doesn't approach the mean depth with the most desired characteristics, i.e. there is a large overshoot of mean depth initially. It is for this reason that when implemented in ARIES a Sliding

Mode Depth Control will be utilized to achieve the ordered depth change. This will allow the vehicle to get within a one-meter depth band of the signal to be tracked, and then the Error Space Controller will be activated to track the time varying signal.

As an alternative to using the Sliding Mode Controller, the Error Space controller could be tuned for use in depth changing maneuvers, where the poles of the controller are now placed to allow a more damped response, and therefore less overshoot of the ordered depth. This is good for approaching ordered depth, but results in a longer time to reach zero error tracking. The resulting performance is seen in Figure 13, where the new poles of the system are places at: [-0.32,-0.2584+0.188i, -0.2584 - 0.188i, -0.0984 + 0.3040i, -0.0984 - 0.3040i]

The resulting compensator and feedback states are as follows:

ucomp = xcom1

ufb = - 5.0646*q + 0.2701*theta - 0.0436*Deptherror

delta_sp = ucomp + ufb

xcom1(i+1) = 0.9992*xcom1 + 0.1250*xcom2 + 0.0007*Deptherror

xcom2(i+1) = -0.0123*xcom1 + 0.9992*xcom2 + 0.0003*Deptherror

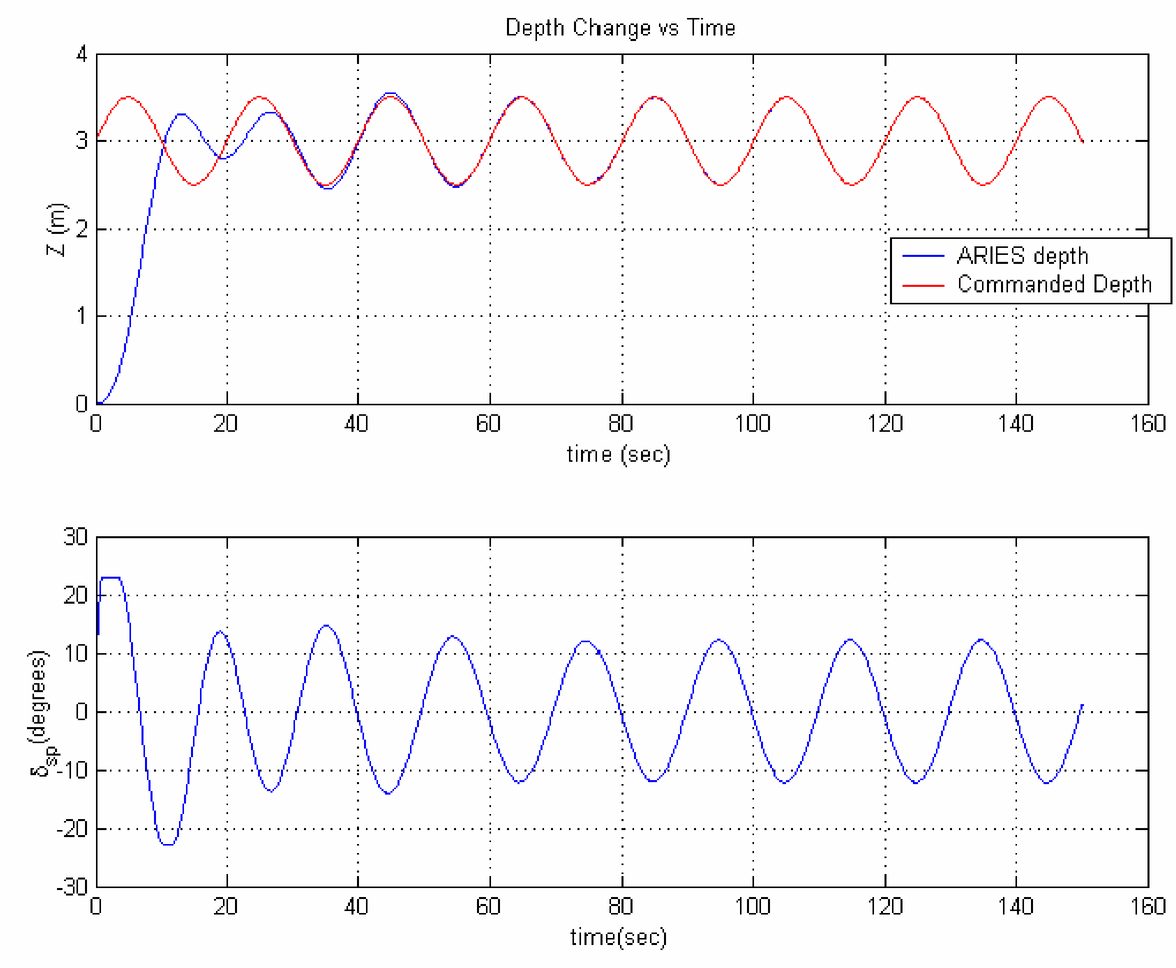

Figure 13 ARIES responses to large depth error with more damped poles

There are other limitations that have to be addressed with the controller. These limits are primarily associated with the limited control authority available to the vehicle and the characteristics of the dynamics matrix of the vehicle.

For a given period signal there is an associated maximum amplitude signal that can be tracked. As can be seen in Figure 14, a signal of one meter amplitude is not able be tracked because the planes are continually being saturated, therefore preventing zero error tracking of the signal.

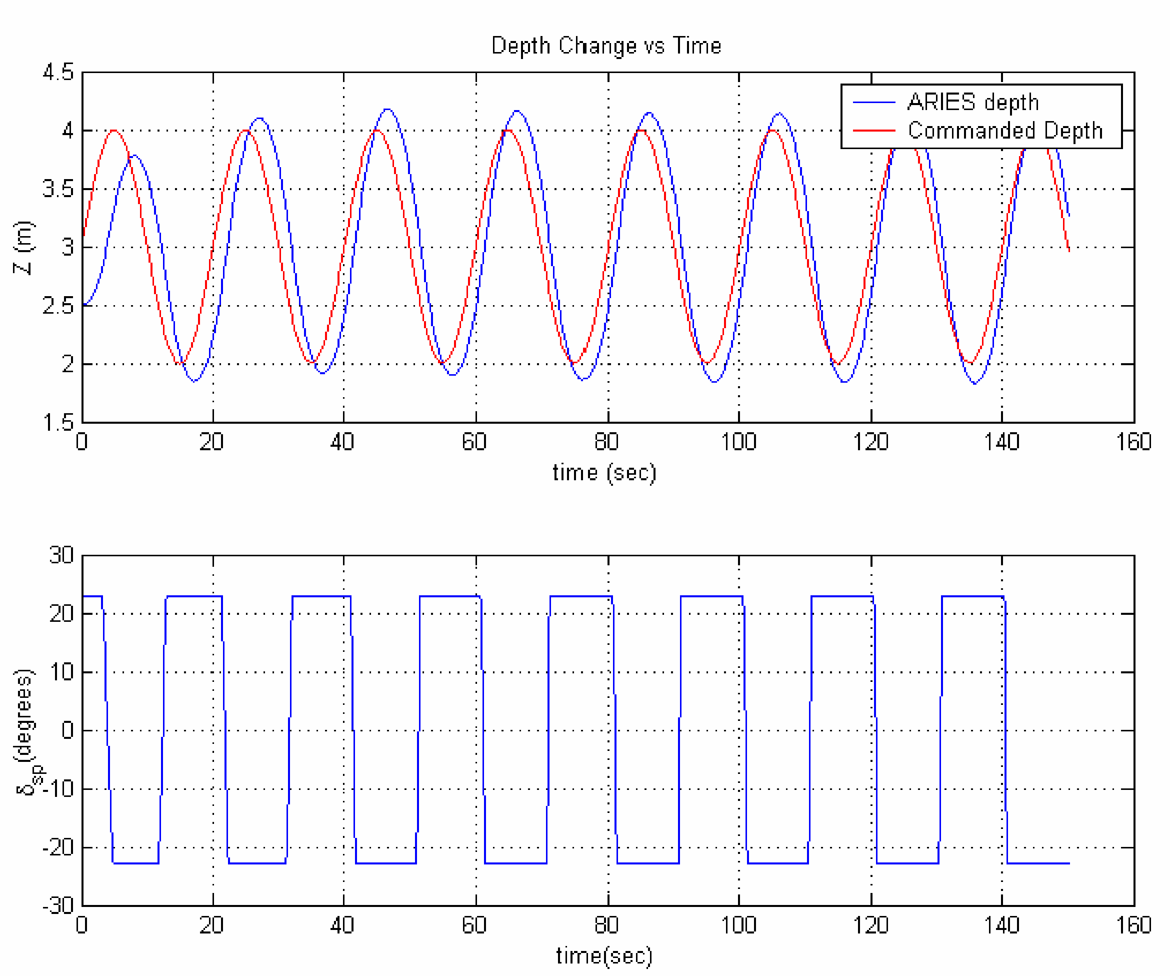

Figure 14 Response to 1 meter amplitude, 20 second period signal

The solution to this problem lies in the limits on control authority in the pitch and heave mode of the vehicle. An increase in control authority in heave and pitch mode would allow for better tracking of higher amplitude signals as well as shorter period signals. For the case of ARIES, vertical thrusters could be added to provide more control authority if deemed necessary to track faster or higher amplitude signal. In case of a completely new vehicle design, consideration could be made for heave and pitch mode control authorities as well as general shape and mass of the vehicle that allows for more favorable hydrodynamic coefficients.

As is shown in Figure 15, if the signal where of longer period of 30 seconds, ARIES could track the signal of one meter amplitude.

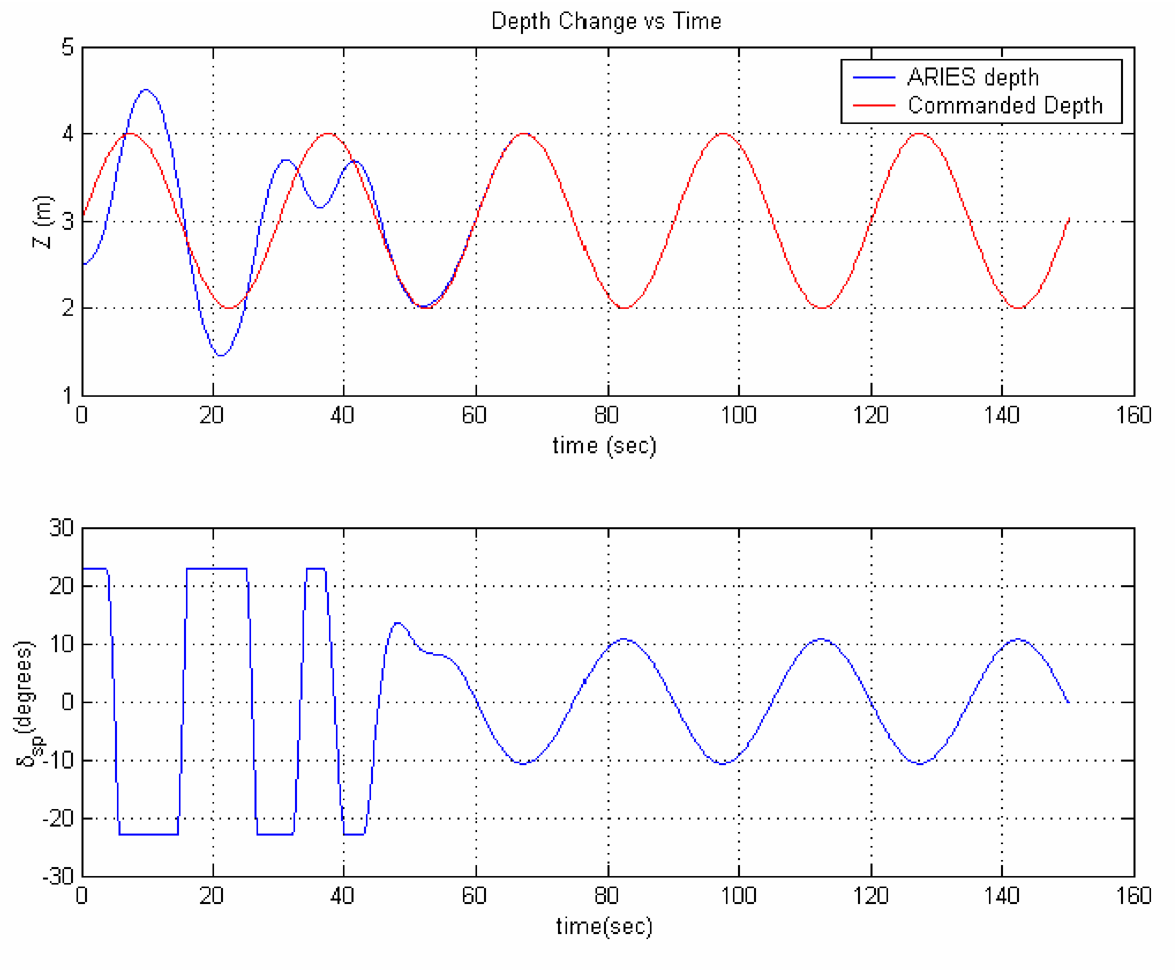

Figure 15 Response to 1 meter amplitude, 30 second period signal

The next limitation to investigate is the effects of a mismatch between the estimated period of the signal being tracked with that of the actual period of the signal. Figure 16 shows the results of a modeled 20 second period signal, and the actual signal is 22 second period.

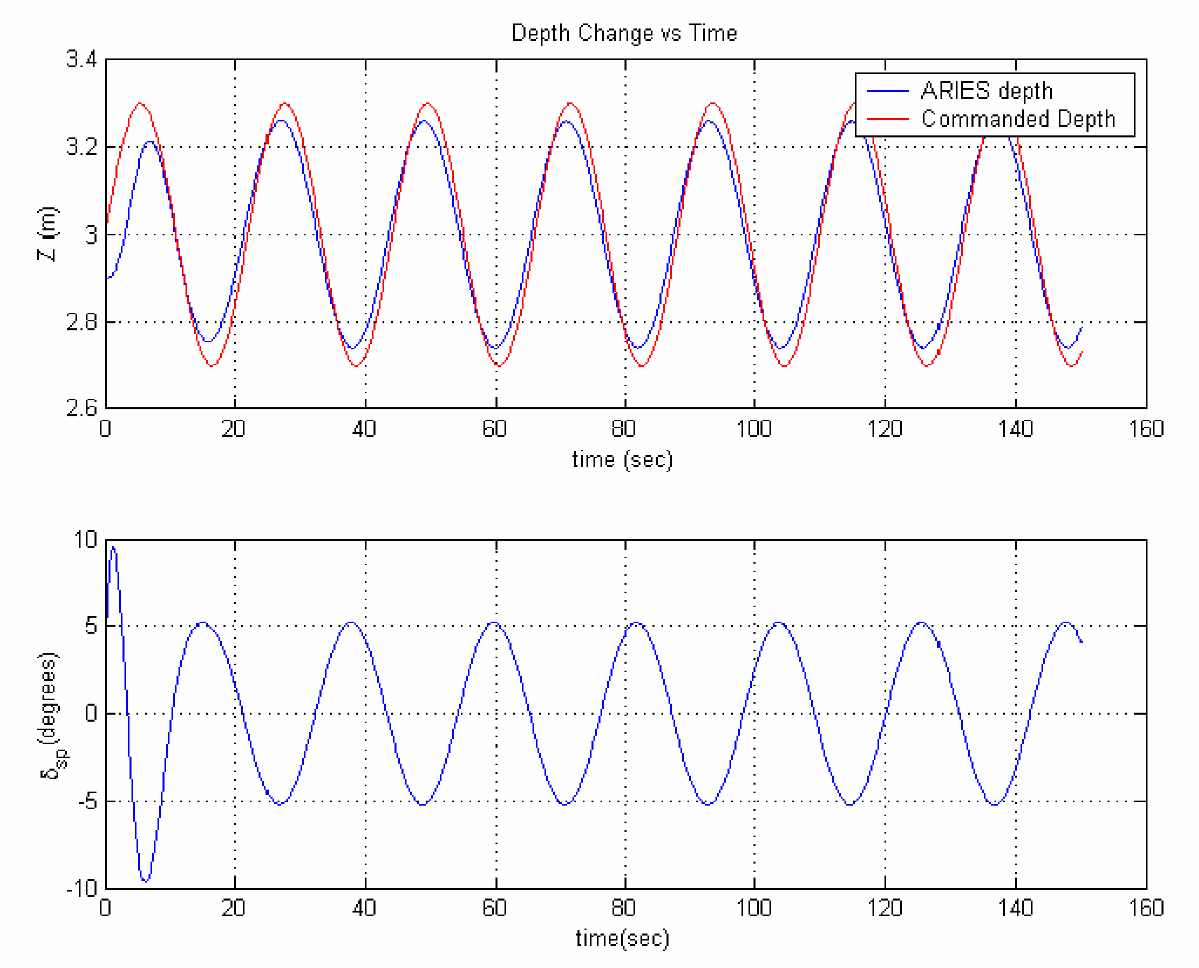

Figure 16 Response to mismatch in modeled input signal dynamics

As can be seen, the controller is not able to achieve zero error tracking and thus highlights the neccesity to have identified the input signal characterstics somewhat accurately.

The results shown here show the feasibility of the use of the Error Space Controller for tracking a time time varying depth command. The controller has shown to be robust, stable, and readily tunable.

B. EXPERIMENTAL RESULTS FOR NEW TRACKING CONTROL IMPLEMENTED IN ARIES

The next step is experimental validation of the error space code inside ARIES. Since experimentally no cage system was available for which to conduct the test with, the control signal for the depth of the cage was generated inside ARIES during the experimental runs.

For the experimental runs ARIES was command to change depth to three meters using mode 0 of the Flight Depth Controller (see Appendix E), which is the original Sliding Mode Control design. ARIES is given 40 seconds to conduct this maneuver after which time the Flight Depth Control mode is changed to mode 2, the error space control. Once inside the error space control code the command for depth is generated from the mean commanded depth of three meters and a sinusoidal componet of period 20 seconds and amplitude 0.5 meters. Two runs were conducted in Monterey Bay on November 14th, 2002 to validate the controller's performance; the results are presented in Figures 17-20.

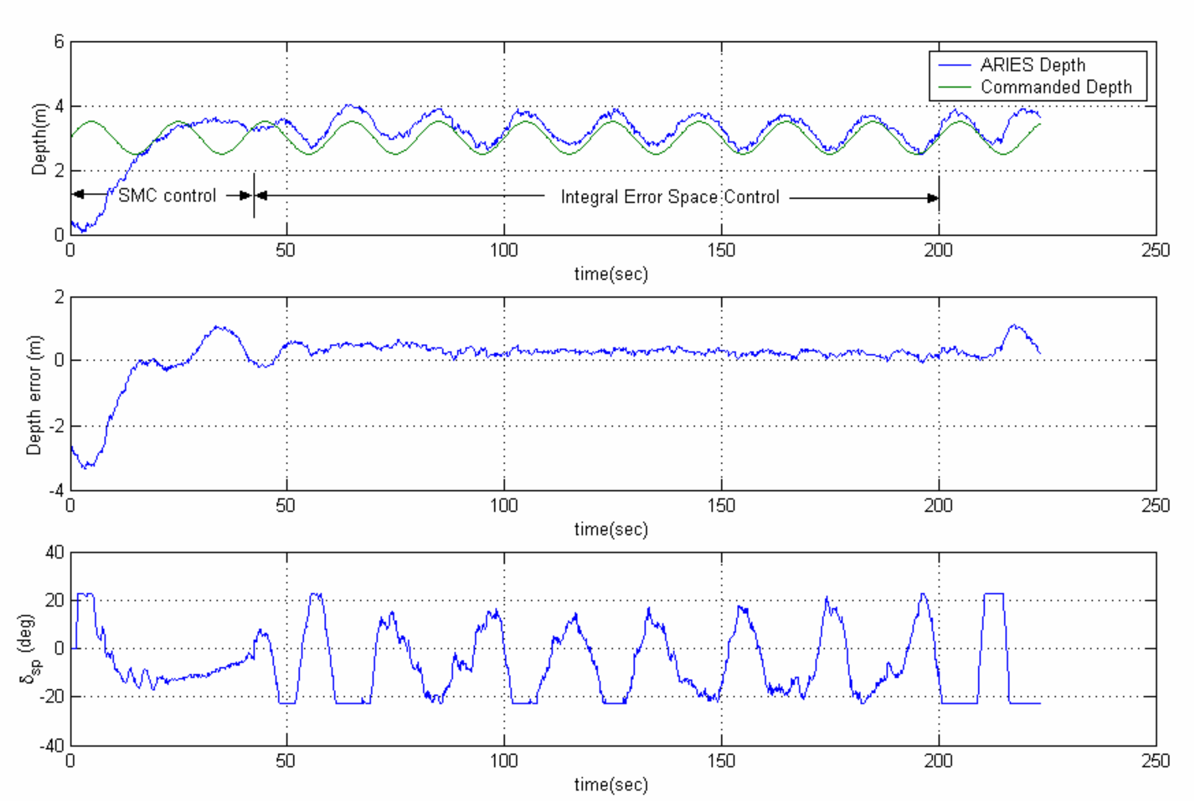

Figure 17 Experimental Run 1 – Tracking a 0.5 m amplitude, 20 second sinusoidal depth command

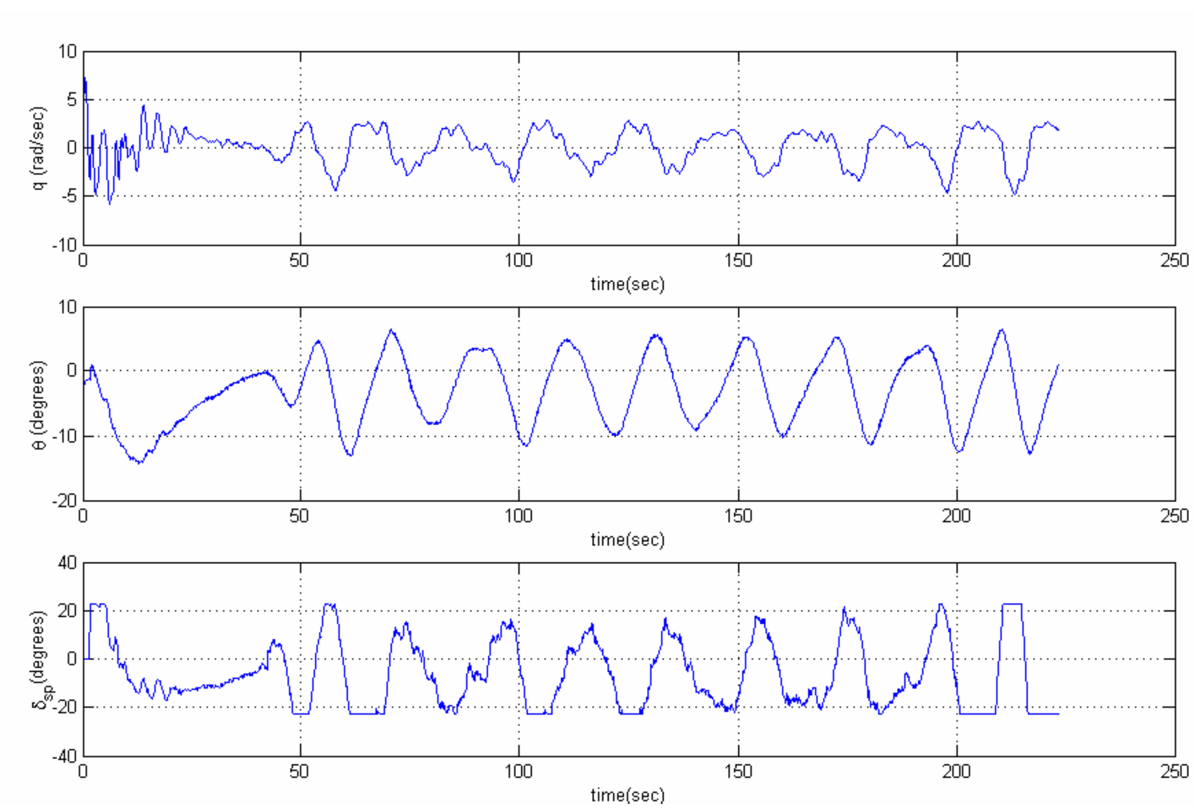

Figure 18 State response to tracking sinusoidal depth command – Run 1

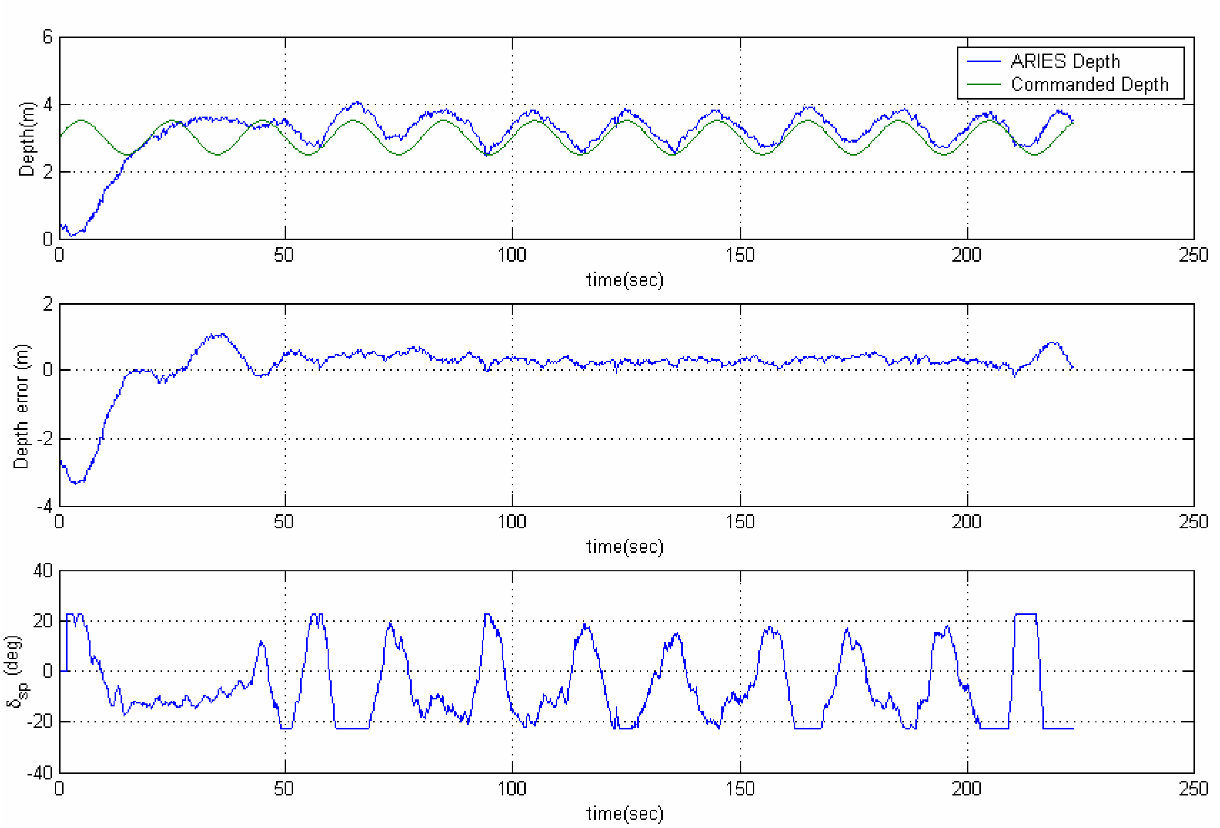

Figure 19 Experimental Run 2 – Tracking a 0.5 m amplitude, 20 second sinusoidal depth command

A few observations can be made from the experimental results shown in Figures 17-19. First off, there appears to be a 0.25 m offset from the signal being tracked and the actual position of ARIES. As was shown previously in Figure 10, the simulated response of the controller shows no offset. This is due to the fact in the simulation, effects of a buoyancy mismatch were ignored as well as any other external disturbance effects such as currents. For ARIES there is an actual buoyancy mismatch whose value depends on the operating depth and conditions of the water (temperature and salinity). During the experimental runs was the water conditions at the operating depth were such that a resulting buoyancy mismatch occurred. This leads to the vehicle tracking at a depth deeper than the commanded depth. A conclusion can be made that the vehicle appears to be running slightly heavy at the operating depth of three meters and therefore tracks at a depth slightly below the commanded. However, as can be seen in Figure 17 and Figure 19 the error in depth due to this mismatch in weight and buoyancy is kept relatively small, with the error on average beign about 0.25 meter. This degree of offset depending on the operationally requirements of the docking cage should be acceptable.

The compensation for this buoyancy mismatch is readily solvable by through disturbance compensation techniques and is recommended for future refinement of this controller. The results of this experiment show the valuable need for experimental validation of model based control codes and the value of having a test platform such as ARIES to conduct these tests. Often in modeling and simulation, assumptions are made and models formed to attempt to effectively simulate an AUV; however, any mismatch in model parameters or invalid assumptions can cause a design to not completely meet the design requirements.

Removing the effects of the buoyancy mismatch Figure 20 shows that ARIES tracks the signal the commanded sinusoidal signal with near zero error.

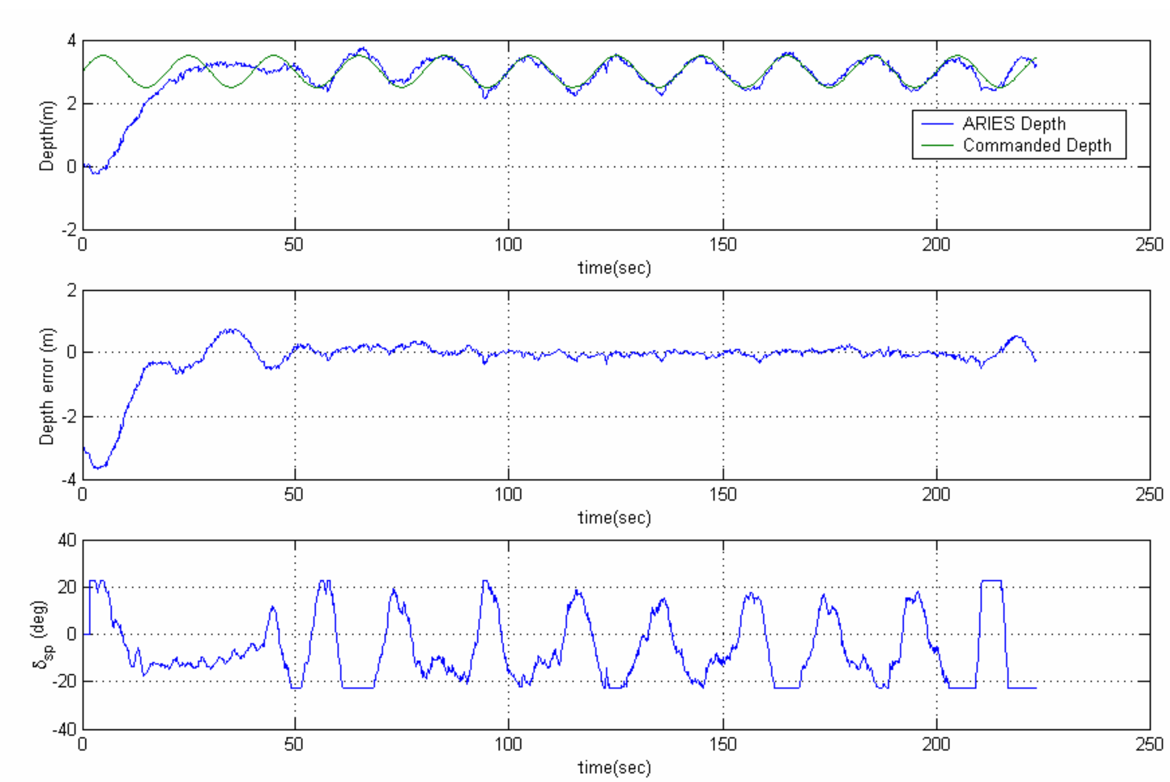

Figure 20 Results removing effect of buoyancy mismatch

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis has shown that utilizing a model based, Error Space Controller, an AUV can track a time varying depth command with zero error. The Error Space Controller has proven to be fairly robust and readily tunable. However, there are limitations, in particular its sensitivity to errors in modeling the input signal. This controller design provides a means by which an AUV could possibly compensate for the time varying depth of a cage system to which it is attempting to dock.

In addition this thesis has shown a good technique for developing the hydrodynamic coefficients of an AUV through a combination of first principles and least squares parameter identification. With these parameters, the engineer is able to readily calculate many new model based control laws and simulate the expected response of the vehicle to various conditions. Finally it has been shown that once satisfactory simulation of the new control law is completed, Naval Postgraduate School's ARIES vehicle provides a valuable test platform for testing new controls code in a real time ocean environment.

B. RECOMMENDATIONS

It was shown that the Error Space Controller was readily able to track a dynamic signal with zero error. However, in order to focus this thesis study on the controller development an assumed input signal was utilized. In reality the input signal characteristics will not be of a known constant value but rather an input into the controller at the time of operation. It is expected that the dynamic parameters of a moving cage system may change with time and that estimation of the signal parameter may be necessary as a real time calculation. Since the the Error Space Controller relies on the model of the input signal to develop the control law, an adaptive control technique may be required to allow for time varying input signal parameters. Further research in to methods by which the controller can obtain the time varying dynamics parameters for the signal and implement them into the control code is recommended for future studies.

In addition to the development of a technique for identifying the cage dynamics, research into the area of expected dynamics of a tethered cage sytem would also be useful. With this data the range of the input signals could be bound. Therefore the limits of the controller/vehicle combination could more accurately be simulated and a better tuned and tested controller could be developed.

This thesis did not attempt to solve the entire problem of how to dock an AUV onto a moving cage system. The focus was on the compensation for dynamics in the vertical plane. In order for a real time docking to take place, consideration to the x-y plane approach to the cage needs to be considered. The horizontal plane portion of the design problem is much less of a dynamic tracking problem, and more of formulating a steering controller to head into the cage within some predefined heading band. The problem of designing a controller for horizontal plane approach during the docking procedure is recommended for future studies.

APPENDIX A - ARIES SLIDING MODE DEPTH CONTROL

```
% ARIES current sliding mode depth control model
% Note this utilizes original model parameters used to design
controller
% Code developed by Dr. Dave Marco and modified by Joe Keller

clear; clc;

% ARIES  HEAVE AND PITCH COEFFICIENTS (calculated and estimated)
W   = 500.0;   %lbs
rho = 1.9903;  %density of seawater slugs/ft^3
Boy = 505.0;   %lbs
g = 32.174;    %ft/s^2

zg = 0.5/12.0; %ft
U = 5;       %ft/s (=3.55 kts or 1.8 m/s)
m = W/g;
a = m;       %check this assumption
M = m+a; % Mass + Added Mass in Heave
L = 10.0;  %ft
Iy =100 ; % lbsec^2/ft % Scaled to 500 lb & Longer from Jay Johnson
87.82
Mq_dot = -0.00625*(rho/2)*(L^5);
Mw_dot = -0.00253*(rho/2)*(L^4);
Zq_dot = -0.00253*(rho/2)*(L^4);
Zw_dot = -0.009340*(L^3);   % looks wrong
Mq    =  -0.01530*(rho/2)*(L^4);%these are all the non-dimensional
coeff*rhoL^5/2 but not times U
Mth = -zg*W;
Mw  =  0.05122*(rho/2)*(L^3);
Zw  = -0.78440*(rho/2)*(L^2);
Zq  = -0.07013*(rho/2)*(L^3);
%Msp = -2.6496; % Stern Plane Moment Effectiveness Approx for Aires
%Mbp =  1.989; % Bow Plane Moment Effectiveness Approx for Aires
Msp = -0.02110*(rho/2)*(L^2)*(L/2); % Stern Plane Moment Effectiveness
(but not time U^2)
Mbp   =      0.02110*(rho/2)*(L^2)*(L/2)*0.8;   %   Bow   Plane   Moment
Effectiveness
Zds = -0.02110*(rho/2)*(L^2); % Stern Plane Force Effectiveness Approx
for Aires
Zdb =  -0.02110*(rho/2)*(L^2); % Bow Plane Force Effectiveness Approx
for Aires

% With cross coupled terms
MM = [(Iy-Mq_dot) 0   -Mw_dot       0;...
           0      1       0         0;...
      -Zq_dot     0    (m-Zw_dot)   0;...
           0      0       0         1];


% [q theta w z]

AA = [Mq*U  Mth  Mw*U   0;...
     1       0      0   0;...
     Zq*U  0      Zw*U  0;...
     0      -U      1   0];
% Unaugmented A Matrix
Aua=inv(MM)*AA;
```

```
% Combining Stern & Bow Plane Effectivness

Mpl = Msp - Mbp;
Zpl = 0.0; % Since Bow and Stern Planes operate equal and opposite
BB = [Mpl*U^2; 0; Zpl*U^2; 0];

% Unaugmented B Matrix
Bua=inv(MM)*BB;


E = ([0;0;0;0]);   %  no disturbances


Dist = inv(MM)*E;
% Augmented A Matrix
A = zeros(5,5);
A([1:4],[1:4]) = Aua;
A(5,4) = -1;

% Augmented B Matrix
Bua(5,1) = 0;
B=Bua;

% Controller Model Has no Heave Velocity FeedBack
I = Iy - Mq_dot;
eta = 1.0;
phi = 0.5;
dt = 0.125;
time=40;
t = [0:dt:time]';
q = 0.0*ones(size(t)); theta = q; z = q; sigma = q;
w=q; dpl=q;
q_com     = 0.0;  q(1)     = 0.0;
theta_com = 0.0;  theta(1) = 0.0;
w_com     = 0.0;  w(1)     = 0.0;

Depth(1) = 0.0;
z(1) = 0;
Ize(1) = 0.0;
C = eye(4);
D = zeros(4,1);

x(:,1) = [q(1);theta(1);w(1);z(1);0]; % Full State
xcm(:,1) = [q(1);theta(1);z(1);0];    % Controller State

     INT = 0;
     omega=2*pi/20;
     ampl=1;
     Depth_com= z(1) + ampl*sin(omega*t);
     %Depth_com=10*ones(1,length(t));

for i=1:length(t)-1,
    %control signal in execf.c

    sigma(i)=-0.7693*(q_com-q(i))-0.6385*(theta_com-theta(i))
+0.0221*3.28*(Depth_com(i)-Depth(i));
    dpl(i)=1.2801*(-0.4105*q(i)+0.1086*theta(i)+
eta*tanh(sigma(i)/phi));
```

```
   if(abs(dpl(i)) > 0.4)
      dpl(i) = 0.4*sign(dpl(i));
   end;

% True State
   x_dot(:,i) = [A(1,1)*q(i) + A(1,2)*theta(i) + A(1,3)*w(i) + ...
                 A(1,4)*z(i) + B(1,1)*dpl(i) + Dist(1); ...
                 A(2,1)*q(i) + A(2,2)*theta(i) + A(2,3)*w(i) + ...
                 A(2,4)*z(i) + B(2,1)*dpl(i);...
                 A(3,1)*q(i) + A(3,2)*theta(i) + A(3,3)*w(i) + ...
                 A(3,4)*z(i) + B(3,1)*dpl(i) + Dist(3);...
                 A(4,1)*q(i) + A(4,2)*theta(i) + A(4,3)*w(i) + ...
                 A(4,4)*z(i) + B(4,1)*dpl(i);...
                 INT*((Depth_com(i)-z(i)))];

% Measurements of the State
   x(:,i+1)   = x(:,i) + dt*x_dot(:,i);

   e(i)=0*rand(1);     % add noise to states
   sigmad=0.15; sigmaq=0.02; sigmath=0.04;sigmaw=0.15;
   q(i+1)     = x(1,i+1)+sigmaq*e(i);
   theta(i+1) = x(2,i+1)+sigmath*e(i);
   w(i+1)     = x(3,i+1)+sigmaw*e(i);
   z(i+1)     = x(4,i+1)+sigmad*e(i);
   Ize(i+1)   = x(5,i+1)+e(i);
   Depth(i+1)   = z(i+1)/3.28 ; % Measurement of z in meters

end;

%%%%% Plot Results
clf;
orient tall;
figure(1);
Depth_com(i+1)=Depth_com(i);
hold off;
subplot(3,1,1);plot(t,(w/3.208)),xlabel('time(sec)'),ylabel('w_r
(m/sec)'),grid on;
title('Heave,  pitch  rate,  pitch  angle,  depth  and  planes  angle  vs
time');
subplot(3,1,2);plot(t,q),xlabel('time(sec)'),ylabel('q  (rad/sec)'),grid
on;
subplot(3,1,3);plot
(t,theta.*180/pi),xlabel('time(sec)'),ylabel('\theta(degrees)'),grid
on;
figure(2);
subplot(2,1,1);plot(t,Depth);grid;
hold;plot(t,Depth_com,'r .');%axis([0 40 0 11]);
xlabel('time (sec)');ylabel('depth(m)');legend('ARIES Depth','Commanded
Depth')
subplot(2,1,2);plot(t,dpl*(180/pi));grid;
xlabel('time(sec)');ylabel('\delta_p_l (degrees)');
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B - SPRING MASS SYSTEM SIMULATION FOR ERROR SPACE CONTROL

```
%Discrete spring mass damper control system
% By LT Joe Keller Naval Postgraduate School for Thesis work

% Integral error space control - (Feedback control textbook version pg
595 Franklin)

%  See intererr2rev2.mdl for simulink response
clear;

k=1;
m=10;
c=0.1;

Ar=[0 1;-k/m -c/m];        %Modeled reference signal dynamics
Br=[0; 1/m];
Cr=[1 0];
D=0;

As=[0 1;0 0];              %Modeled system dynamics
Bs=[0; 1/m];
Cs=[1 0];
Ds=0;

Ara=[0 1;-100*k/m -100*c/m]; % Actual reference signal dynamics;
Bra=[0;1/m];              % This is useful for showing robustness under
modeling error

%Initial Conditions
dt=0.125;
tt=25;

T=0:dt:tt;

%WORKER MASS MOTION
 x(:,1)=[0.5;0.2];
 [p,g]=c2d(Ara,Bra,dt);

  for i=1:length(T)
   x(:,i+1)=p*x(:,i);
   y(:,i)=0.7*sin(sqrt(k/m)*i*dt) + 3;
  end

% SERVER CONTROLLER
%Form error space matrices
E=[Ar Br*Cs;zeros(2) As];
F=[0;0;Bs];
poles=1.0*[-3.707+3.707i,-3.707-3.707i,-4.1,-4.2];
K=place(E,F,poles);

Ko=[K(3) K(4)];  %Plant feedback gains

Ec=Ar;             %compensator dynamics
Fc=[K(1) K(2)];  %compensator error feedback gains
Gc=[1 0];
Hc=0;
```

```
Acom=[Ar Br*Cs;-Bs*Fc As-Bs*Ko];
Bcom=[-Br;0;0];
Ccom=-K;
Bcom=[-Br;Bs*Ko*Cs'];

mu1=-[0 0 Cs]*inv(Acom)*Bcom;

%Frequency response of controller and system
[num,den]=ss2tf(Ar,Fc',Gc,Hc);
Gcomp=tf(num,den);                  %compensator
[num2,den2]=ss2tf(As-Bs*Ko,Bs,Cs,Ds);  %plant with feedback
Gsys=tf(num2,den2);
Gtot=Gcomp*Gsys/(Gcomp*Gsys-1);  %output to input transfer function
er=Gtot-1;                %error to input transfer function

figure(2);
margin(er);grid;

%Change to disrcete

[Ard,Brd]=c2d(Ar,Br,dt);
[Asd,Bsd]=c2d(As,Bs,dt);
[Adcom,Bdcom]=c2d(Acom,Bcom,dt);

%Initial Conditions

y(1)=[0];
xcom(:,1)=[0;0;0;0];

for i=1:(length(T)-1)
   xcom(:,i+1)=Adcom*xcom(:,i)+Bdcom*y(i);
   u(i)=-K*xcom(:,i)+Ko*Cs'*y(i);
   e(i)=xcom(3,i)-y(i);
end

%Plot Results
figure(1);
subplot(3,1,1)
plot(T,y,T,xcom(3,:));title('Error Space Controller Response');grid on;
ylabel('Position             (m)');xlabel('Time(sec)');legend('Reference
Signal','System Response');
subplot(3,1,2);
plot((1:i)*dt,u);grid;
ylabel('Control Force (N)');xlabel('Time(sec)')
subplot(3,1,3);
plot((1:i)*dt,e);grid;
ylabel('Position error (m)');xlabel('Time(sec)')
```

APPENDIX C-PARAMETER IDENTIFICATION OF ARIES

```
%  Parameter ID for pitch and heave modes of ARIES
%
%  code by LT Joe Keller, Naval Postgraduate School
%  for completion of Masters Thesis

clear;
clc;

load experdatam  % loads only need states from Nav_d and d081602_02.d
files
load wtemp

% %De-mean all values
w=w-mean(w);
q=q-mean(q);
theta=theta-mean(theta);
dsp=dsp-mean(dsp);
umean=mean(u);


%Insert Lag into planes (see Johnson pg 28,53)
lagdsp(1)=dsp(1);
tau=0.5;    % see rudder vs pitch rate response for a lag of 0.5 sec
for i=2:length(dsp);
    lagdsp(i)=(1-tau)*lagdsp(i-1)+tau*dsp(i-1);
end
lagdsp=lagdsp';

% takes data up to first turn
start=692;      %before turn
finish=3169;
% start2=4480;             %after turn
% finish2=6936;
start2=3170;
finish2=3170;

w=[w(start:finish);w(start2:finish2)];
q=[q(start:finish);q(start2:finish2)];
theta=[theta(start:finish);theta(start2:finish2)];
dsp=[dsp(start:finish);dsp(start2:finish2)];
lagdsp=[lagdsp(start:finish);lagdsp(start2:finish2)];
Depth=[Depth(start:finish);Depth(start2:finish2)];

%%%  Pitch Equation Parameter ID  %%%%%%

dt=0.125; % sample time for Aries
U=1.41;  % m/s from umean
L=10/3.208;  % ft
rho = 1020;  %density of seawater kg/m^3
Iyy=119.1; %kg*m^2  (Johnson thesis pg 20)
Mq_dot = -93.13; % kg*m^2 = Nr_dot Johnson pg 51
Mw_dot = -0.00253*(rho/2)*(L^4);   %check this
Mtotp=Iyy-Mq_dot;

% Asssume Mdsp known from first principle calculations
Mdsp=-237.5 ; % Nm/rad
theta4=Mdsp*dt/Mtotp;
```

```
Mtheta=-28.3; %Nm
theta3=Mtheta*dt/Mtotp;

for ii=2:length(q)
    yp(ii-1)=q(ii)-theta3*theta(ii-1)-theta4*lagdsp(ii-1);
end
yp(ii)=yp(ii-1);
yp=yp';

H=[w q];

thetahat=inv(H'*H)*H'*yp;

 Mw=thetahat(1)*Mtotp/dt;
 Mq=(thetahat(2) - 1)*Mtotp/dt;
%  Mdsp=thetahat(3)*Mtotp/dt;


 %%% Check Condition and error

 Conditionp=cond(H'*H);
 errorp=yp-H*thetahat;
 Jp=1/max(size(H))*(errorp'*errorp);
 relerrorp=sqrt(Jp)/std(yp)*100;


  sprintf('Mw %5.1f    Mq %5.1f    Mtheta %5.1f    Mdsp %5.1f   Relative
Error %3.1f%',Mw,Mq,Mtheta,Mdsp,relerrorp)

%%%%%%  Heave equation Parameter ID  %%%%%%%%%%%%%%%%%%%%%%%

m   = 222;   %kg from Jay Johnson Thesis
Zw_dot = -234 ;  %kg  =Yv_dot=-234 kg from Johnson pg 51
Zq_dot = -0.00253*(rho/2)*(L^4); %check this
Mtotw=m-Zw_dot;

% Provide coeff's from first principles
Zdsp=26.1 ; % N
thetaw3=Zdsp*dt/Mtotw;


for ii=2:length(w)
    yw(ii-1)=w(ii) - thetaw3*lagdsp(ii-1);
end
yw(ii)=yw(ii-1);
yw=yw';

Hw=[w q];

thetahatw=inv(Hw'*Hw)*Hw'*yw;

 Zw=(thetahatw(1)-1)*Mtotw/dt;
 Zq=thetahatw(2)*Mtotw/dt-m*U;
%  Zdsp=thetahatw(3)*Mtotw/dt;

  %%% Check Condition and error
 Conditionw=cond(Hw'*Hw);
```

```
 errorw=yw-Hw*thetahatw;
 Jw=1/max(size(Hw))*(errorw'*errorw);
 relerrorw=sqrt(Jw)/std(yw)*100;

 sprintf('Zw%5.1f         Zq%5.1f         Zdsp%5.1f         Relativeerror
%3.1f%',Zw,Zq,Zdsp,relerrorw)

 %%%  Simulate vehicle response with new coefficients  %%%%%%

 %Coefficient matrices


% Actual Vehicle Matrices
M=[m-Zw_dot -Zq_dot 0 0;
   -Mw_dot Iyy-Mq_dot 0 0;
    0 0 1 0;
    0 0 0 1];
A=[Zw m*U+Zq 0 0;
   Mw Mq Mtheta 0;
   0 1 0 0;
   1 0 -U 0];
B=[Zdsp Mdsp 0 0]';

As=inv(M)*A;
Bs=inv(M)*B;

Cs=[0 0 0 1];
Ds=0;


[Asd,Bsd]=c2d(As,Bs,dt);


ys(:,1)=[w(1);q(1);theta(1);Depth(1)];

%States ys(:,1)=w; ys(:,2)=q; ys(:,3)=theta; ys(:,4)=Depth

T=0:dt:(length(q)-1)*dt;

for i=1:(length(T)-1)
     ys(:,i+1)=Asd*ys(:,i)+Bsd*lagdsp(i);             %Aries full state
motion
end

for i=2:(length(T)-1)
    ys(4,i)=ys(4,i)  +  (0.0092)*i*dt;     %  takes  linear  trend  out  of
depth
end


% Plot Results

%  figure(1);subplot(3,1,1);plot(T,w);
xlabel('time(sec) '),ylabel('wr (m/sec)'),grid on;
figure(1);subplot(3,1,1);plot(T,w,T,ys(1,:)),xlabel('time(sec)'),
ylabel('wr (m/sec)'),grid on;
title('Heave velocity, pitch rate, and pitch angle vs time');
legend('Actual Vehicle','Modeled Vehicle');
```

```
subplot(3,1,2);plot(T,q,T,ys(2,:)),xlabel('time(sec)');
ylabel('q (rad/sec)'),grid on;
%subplot(3,1,2);plot(T,q),xlabel('time(sec)')
ylabel('q (rad/sec)'),grid on;
subplot(3,1,3);
plot(T,theta*180/pi,T,ys(3,:)*180/pi),xlabel('time(sec)'),ylabel('\thet
a (degrees)'),grid on;
%subplot(3,1,3);
plot(T,theta*180/pi);
xlabel('time(sec)'),ylabel('\theta (degrees)'),grid on;
figure(2);subplot(2,1,1);plot(T,Depth,T,ys(4,:));
xlabel('time (sec)'),ylabel('Z (m)'),grid on;
% figure(2);subplot(2,1,1);
plot(T,Depth),xlabel('time (sec)'),ylabel('Z (m)'),grid on;
title('Depth Change vs Time');
legend('Actual Vehicle','Modeled Vehicle');
subplot(2,1,2);plot(T,lagdsp*180/pi);
xlabel('time(sec)'),ylabel('dsp(degrees)');grid;
```

APPENDIX D – ARIES ERROR SPACE CONTROL CODE

```
%  Aries diving control Using Error Space Control
%
%  Code by LT Joe Keller, Naval Postgraduate School
%  for completion of Masters Thesis
%
clear; clc;

%Vehicle Characteristics

% ARIES  HEAVE AND PITCH COEFFICIENTS

% All units metric
U=1.41;  % m/s from umean
L=10/3.208;  % m
rho = 1020;  %density of seawater kg/m^3
Iyy=119.1; %kg*m^2  (Johnson thesis pg 20)
Mq_dot = -93.13; % kg*m^2 = Nr_dot Johnson pg 51
Mw_dot = -0.00253*(rho/2)*(L^4);   %check this
Mtotp=Iyy-Mq_dot;
m   = 222;   %kg from Jay Johnson Thesis
Zw_dot = -234 ;  %kg  =Yv_dot=-234 kg from Johnson pg 51
Zq_dot = -0.00253*(rho/2)*(L^4); %check this
Mtotw=m-Zw_dot;


%% From parameter id
Mw=45.5;
Mq=-1422.2;
Mtheta=-28.3;
Mdsp = -237.5;

Zw=-764;
Zq=120;
Zdsp=26.1;

%Set time scale
dt=0.125;
tt=150;
T=0:dt:tt;

%Recovery Cage motion
Tc=20; % cage period (seconds)
w=2*pi/Tc;
Amp=0.5;
Ar=[0 1;-(w^2) 0];
Br=[0;0];
Cr=[1 0];

Mean_depth = 3;

x(:,1)=[0;Amp];

 [p,g]=c2d(Ar,Br,dt);

  for i=1:length(T)
   x(:,i+1)=p*x(:,i);
```

```
   y(:,i)= Mean_depth + Amp*sin(2*pi*i*dt/(1*Tc));    %note this change
  end

%Coefficient matrices

% Actual Vehicle Matrices
M=[m-Zw_dot -Zq_dot 0 0;
   -Mw_dot Iyy-Mq_dot 0 0;
    0 0 1 0;
    0 0 0 1];
A=[Zw m*U+Zq 0 0;
   Mw Mq Mtheta 0;
   0 1 0 0;
   1 0 -U 0];
B=[Zdsp Mdsp 0 0]';

As=inv(M)*A;
Bs=inv(M)*B;
Cs=[0 0 0 1];
Ds=0;

% For controller design model of Aries assuming no influence of heave
and use 3 state model
% As well as utilizing values from Parameter identification
Mm=[Iyy-Mq_dot 0 0;
    0 1 0;
    0 0 1];

Am=[Mq Mtheta 0;
   1 0 0;
   0 -U 0];

Bm=[Mdsp 0 0]';

Asm=inv(Mm)*Am;
Bsm=inv(Mm)*Bm;
Csm=[0 0 1];
Dsm=0;



%Form error space matrices for controller design
E=[Ar Br*Csm;[zeros(3,2)] Asm];
F=[0;0;Bsm];
poles=1.1*[-0.4,-0.42,-0.43,-0.44,-0.45]; % poles to use if controller
initialized with small depth error , faster tracking
%poles=0.8*[-0.4,-0.323+0.235i,-0.323-0.235i,-0.123+0.38i,-0.123-
0.38i]; % poles to use if using with large intial depth error

K=place(E,F,poles);

Ko=[0 K(3) K(4) K(5)];  %Plant feedback gains

Ec=Ar;              %controller dynamics
Fc=-[K(1);K(2)];    %controller error feedback gains
Gc=[1 0];
Hc=0;
```

```
%Change to disrcete

[Ecd,Fcd]=c2d(Ec,Fc,dt);
[Asd,Bsd]=c2d(As,Bs,dt);
[Asdm,Bsdm]=c2d(Asm,Bsm,dt);



% Aries response with error space controller
% Coded as it will be used in vehicle C code

%Initial values

Depth_com=y;

xcom1(1)=0;
xcom2(1)=0;
w(1)=0;
q(1)=0;
theta(1)=0;
Depth(1)=2.5;


for i=1:(length(T)-1)

    Deptherror(i)=Depth(i)-Depth_com(i);   %error in depth

    %Compensator input to control (u)
    ucomp(i)= Gc(1)*xcom1(i)+Gc(2)*xcom1(i);

    %State Feedback input to control
    ufb(i)= -Ko(2)*q(i) - Ko(3)*theta(i) - Ko(4)*Deptherror(i);

    %Total control input to dive planes
    delta_sp(i)= ucomp(i) + ufb(i);


    %compensator dynamics

     % this prevents compensator from turning on until within reasonable
depth error
    if abs(Deptherror(i))>1.0
        xcom1(i+1)=0;        % is achieved
        xcom2(i+1)=0;
    else
       xcom1(i+1)=Ecd(1,1)*xcom1(i)+Ecd(1,2)*xcom2(i)+
Fcd(1,1)*Deptherror(i);
       xcom2(i+1)=Ecd(2,1)*xcom1(i)+Ecd(2,2)*xcom2(i)+
Fcd(2,1)*Deptherror(i);
   end



    %limits the bow/stern planes to 0.4 radians
    if abs(delta_sp(i))>0.4
        delta_sp(i)=0.4*sign(delta_sp(i));
    end
```

```
    %Simulate Aries full state motion

    ys(:,i+1)=Asd*[w(i);q(i);theta(i);Depth(i)]+Bsd*delta_sp(i);

    w(i+1)=ys(1,i+1);
    q(i+1)=ys(2,i+1);
    theta(i+1)=ys(3,i+1);
    Depth(i+1)=ys(4,i+1);

end

%%%%%%%%%%%%%%%%%%%%
% Plot the results %
%%%%%%%%%%%%%%%%%%%%

figure(1)
subplot(2,1,1);plot(ufb);ylabel('ufb');subplot(2,1,2);plot(ucomp);ylabe
l('ucomp');

 T=(1:(i+1))*dt;
 delta_sp(i+1)=delta_sp(i);
 figure(2);subplot(4,1,1);plot(T,w);
 xlabel('time(sec) '),ylabel('w_r (m/sec)'),grid on;
  title('Heave velocity, pitch rate, pitch angle, and planes input vs
time');
 subplot(4,1,2);plot(T, q);
 xlabel('time(sec)'),ylabel('q (rad/sec)'),grid on;
 subplot(4,1,3);plot (T, theta*180/pi);
 xlabel('time(sec)'),ylabel('\theta (degrees)'),grid on;
 subplot(4,1,4);plot(T,delta_sp*180/pi);
 xlabel('time(sec)'),ylabel('\delta_s_p(degrees)');grid;
 figure(3);subplot(2,1,1);plot(T,Depth);
 xlabel('time (sec)'),ylabel('Z (m)'),grid on;
 title('Depth Change vs Time');
 hold on;
 plot(T,Depth_com,'r'); hold off;
 legend('ARIES depth','Commanded Depth');

;subplot(2,1,2);plot(T,delta_sp*180/pi);xlabel('time(sec)'),ylabel('\de
lta_s_p(degrees)');grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%  Output lines of code for insertion into ARIES  %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sprintf('Deptherror = Depth - Depth_com')
sprintf('ucomp = %8.4f*xcom1 + %8.4f*xcom2',Gc(1),Gc(2))
sprintf('ufb     =     -%8.4f*q          -     %8.4f*theta          -
%8.4f*Deptherror',Ko(2),Ko(3),Ko(4))
sprintf('delta_sp = ucomp + ufb')
sprintf('xcom1(i+1)     =     %8.4f*xcom1     +          %8.4f*xcom2     +
%8.4f*Deptherror',Ecd(1,1),Ecd(1,2),Fcd(1,1))
sprintf('xcom2(i+1)     =     %8.4f*xcom1     +          %8.4f*xcom2     +
%8.4f*Deptherror',Ecd(2,1),Ecd(2,2),Fcd(2,1))
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Create Data file for vrml display %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%  Aries [t,X,Y,Z]  note vrml y is ARIES -z
xaries=1.4*T;   %  U=1.4 m/s
aries=[T',xaries',-ys(4,:)',0*T']; format short;
save ariesdata.d aries -ASCII -DOUBLE -TABS

%%%%%  Cage [t,X,Y,Z]
xcage=(1.41*max(T)*ones([1,length(T)]));
cage=[T',xcage',-y',0*T'];
save cagedata.d cage -ASCII -DOUBLE –TABS
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E – MODIFIED ARIES C CODE INCLUDING MODE 2 – ERROR SPACE CONTROLLER

```
FlightDepthControl(Depth_com,Mode)
   double Depth_com; /* Meters */
   int  Mode;          /*  Mode  1  =  Int  Cont,  0  =  No  Int  Cont  ,
                               Mode 2 = Int Err Space*/
{
   double q_com,theta_com;
   double Sigma_FlightDepth,DepthError;
   double ufb, ucomp, Depth_com_es, xcom1_new, xcom2_new;


   q_com     = 0.0;
   theta_com = 0.0;

 switch(Mode)
{

 /* Sliding Mode depth control*/
 case 0:

    /* From /vault3/marco/ocean_test/MODEL/dive_designAires.m */
    /* For Poles = [0.0 -0.41 -0.42 ]                           */
    /* Eta_FlightDepth = 1.0; Phi_FlightDepth = 0.5;           */

    DepthError = Depth_com - Depth;

    /* Saturate the Error at 2.0 meters */
    /*if(fabs(DepthError) > 2.0)
    {
       DepthError = 2.0*dsign(DepthError);
    }*/

    Sigma_FlightDepth = -0.7693*(q_com - q) - 0.6385*(theta_com -
                              theta)+ 0.0221*(3.28*DepthError);

    delta_sp = 1.2801*( -0.4105*q + 0.1086*theta
              +
      Eta_FlightDepth*dtanh(Sigma_FlightDepth/Phi_FlightDepth) );


    break;/*End Case 0*/


  /* Integral error space control with sinusoidal Depth_com*/
  case 2:

     Depth_com_es = Depth_com + 0.5*sin(0.3142*t);
     /* printf("%f\n",Depth_com_es); */
     DepthError = Depth - Depth_com_es;

/* Update compensator state*/

     if(DepthError > 1.0)
     {
```

```
        xcom1_new = 0.0;
        xcom2_new = 0.0;
       }
       else
       {

       xcom1_new = 0.9992*xcom1 + 0.1250*xcom2 - 0.0026*DepthError;
       xcom2_new = -0.0123*xcom1 + 0.9992*xcom2+0.0045*DepthError;

       }

       xcom1 = xcom1_new;
       xcom2 = xcom2_new;

       ucomp = xcom1;
       ufb = -3.8847*q + 1.7725*theta -0.5133*DepthError;
       delta_sp = ucomp + ufb;


       break;  /*End Case 2*/

     }  /* End Switch */


     if(fabs(delta_sp) > 0.4)
     {
        delta_sp = 0.4*delta_sp/fabs(delta_sp);
     }

     /* Depth Below Which Suction Force is Negligible */
     if( (Depth < DepthSuck) && (Depth_com > DepthSuck))
     {
        Planes(0.4,-0.4); /* Give the Max Deflection */
     }
     else
     {
        Planes(delta_sp,delta_sp);
     }
  }
```

LIST OF REFERENCES

[1] Stokey, R., M. Purcell, N. Forrester, T. Austin, R. Goldsborough, B. Allen and C. von Alt. 1997. "A Docking System for REMUS, an Autonomous Underwater Vehicle." Proceedings from the IEEE Oceans '97 Conference, Vol. II. Halifax, Nov Scotia, Canada, October 6-9, 1997, pp 1132-1136.

[2] Healey, A.J., Dynamics of Marine Vehicles (ME-4823), Class Notes, Naval Postgraduate School, Monterey, CA, 1995.

[3] Reidel, Jeffrey, "Seaway Learning and Motion Compensation in Shallow Waters for Small AUVs," Doctoral Dissertation, Naval Postgraduate School, Monterey, CA, June 1999.

[4] Lienard, D. and A.J. Healey, "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles, Journal of Oceanic Engineering, vol. 18, pp 327-339, July 1993.

[5] Marco, D.B and A.J. Healey, "Command, Control and Navigation Experimental Results With the NPS ARIES AUV," IEEE Journal of Oceanic Engineering – Special Issue, 2001.

[6] Franklin G.F, Powell, Emami-Naeini. Feedback Control of Dynamic Systems.4th Edition New Jersey:Prentice Hall 2002.

[7] Astrom K.J., Wittnemark. Adaptive Control 2nd Edition California:Addison-Wesley 1995.

[8] Franklin G.F., Powell. Digital Control of Dynamic Systems Massachusetts:Addisson Wesley 1980.

[9] Gelb, A., [and others], Applied Optimal Estimation, MIT Press, Cambridge, MA 1974

[10] Johnson, Jay, "Parameter Identification of the ARIES AUV," M.S. Thesis Naval Postgraduate School, Monterey, CA, June 2001.

[11] Healey, A. J., "Command and Control Demonstrations with Cooperating Vehicles," ONR Research Proposal in response to ONR BAA 01-012 "Demonstration of Undersea Autonomous Operation Capabilities and Related Technology Development", August 2001.

[12] Lewis, E.V. (ed.), Principles of Naval Architecture, vol. II, second revision, Society of Naval Architects and Marine Engineers (SNAME), Jersey City, NJ, 1988.

[13] Lienard, David, "Autopilot Design for Autonomous Underwater Vehicles Based on Sliding Mode Control," M.S. Thesis, Naval Postgraduate School, Monterey, CA, June 1990.

[14] Marco, D.B., “Procedure to Run Missions with the ARIES,” Personal Notes, 2001.

[15] Marco, D.B. and A.J. Healey, “Current Developments in Underwater Vehicle Control and Navigation,” Proceedings of IEEE Oceans, 2000.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Mechanical Engineering Department Chairman, Code ME
   Naval Postgraduate School
   Monterey, CA

4. Naval/Mechanical Engineering Curriculum Code 34
   Naval Postgraduate School
   Monterey, CA

5. Professor Anthony J. Healey, Code ME/HY
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA

6. Dr. Donald Brutzman, Code UW/Br
   Undersea Warfare Group
   Naval Postgraduate School
   Monterey, CA

7. Dr. T. B. Curtin, Code 322OM
   Office of Naval Research
   Arlington, VA

8. Dr. T. Swean, Code 32OE
   Office of Naval Research
   Arlington, VA

9. L.C. Van Alt
   WoodsHole Oceanographic Institute
   Boston, MA

10. D. Richard Blidberg
    AUSI
    Durham, NH

11. Dr. Jeng Yuh
    National Science Foundation
    Arlington, VA

12. Dr. J. Sur
Korean Naval Academy
Chinhae, Korea

13. LT Joe Keller
Naval Postgraduate School
Monterey, CA