NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THREE DIMENSIONAL PURSUIT GUIDANCE AND
CONTROL OF SUBMERSIBLE VEHICLES

by

Evangelos G. Papasotiriou

September, 1991

Thesis Advisor: Fotis A. Papoulias

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 69	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No. / Work Unit Accession Number

11. TITLE (Include Security Classification)
THREE DIMENSIONAL PURSUIT GUIDANCE AND CONTROL OF SUBMERSIBLE VEHICLES

12. PERSONAL AUTHOR(S) Evangelos G. Papasotiriou

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Master's Thesis	From To	September 1991	112

16. SUPPLEMENTARY NOTATION
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.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)
Submersible, autopilot, path, stability, guidance.

19. ABSTRACT (continue on reverse if necessary and identify by block number)

A pure pursuit guidance law is combined with a heading autopilot to provide accurate path keeping of submersible vehicles. The scheme is implemented and analyzed in both the horizontal and vertical planes. A complete stability analysis is performed in order to evaluate regions of stable vehicle operations. Numerical integrations support the analytic predictions. Two distinct stability boundaries are established. In the first, the vehicle loss of stability is accompanied by the generation of oscillatory motions around the commanded path. In the second, loss of stability occurs with linearly increasing path deviation. The horizontal and vertical plane schemes are combined with a propulsion control law in order to achieve path tracking of a general commanded route composed of several straight line segments in three dimensional space.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [X] UNCLASSIFIED/UNLIMITED [ ] SAME AS REPORT [ ] DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Fotis A. Papoulias	22b. TELEPHONE (Include Area code) (408) 646-3381	22c. OFFICE SYMBOL 69Pa

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

Three dimensional pursuit guidance and control of submersible vehicles .

by

Evangelos G. Papasotiriou
Lieutenant, Hellenic Navy

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Author: ______________________________
Evangelos G. Papasotiriou

Approved by: ______________________________
Fotis A. Papoulias, Thesis Advisor

____
Anthony J. Healey, Chairman
Department of Mechanical Engineering

ABSTRACT

A pure pursuit guidance law is combined with a heading autopilot to provide accurate path keeping of submersible vehicles. The scheme is implemented and analyzed in both the horizontal and vertical planes. A complete stability analysis is performed in order to evaluate regions of stable vehicle operations. Numerical integrations support the analytic predictions. Two distinct stability boundaries are established. In the first, the vehicle loss of stability is accompanied by the generation of oscillatory motions around the commanded path. In the second, loss of stability occurs with linearly increasing path deviation. The horizontal and vertical plane schemes are combined with a propulsion control law in order to achieve path tracking of a general commanded route composed of several straight line segments in three dimensional space.

TABLE OF CONTENTS

LIST OF FIGURES

I INTRODUCTION

One of the most significant functions of an underwater vehicle is accurate path control for transiting along prescribed routes in three dimensional space. The commanded path is usually described by a series of way points in space and time either by the commander or by a path planner function in the case of an unmanned vehicle. Without significant loss of generality we can assume that the commanded path can be approximated by straight line segments between consecutive way points. This assumption does not alter the important features of the path keeping problem since every smooth path can be approximated arbitrarily closely by a series of straight line segments. Once a desired straight line path has been generated, the vehicle guidance and autopilot functions are called upon to ensure satisfactory path keeping through the use of the vehicle actuators.

One way to ensure that the vehicle goes through a specified sequence of way points is by using a heading autopilot coupled with a line of sight guidance scheme [1]. The scheme proved to be robust enough so that when coupled with an independently developed depth autopilot [2], accurate depth control was maintained while transiting between way points in the horizontal plane. The disadvantage associated with this technique is that the actual vehicle path between two consecutive way points differ significantly from the corresponding straight line segment.

In order to overcome this problem and achieve accurate path control in the presence of obstacles and underwater currents, a cross track error autopilot was developed for the horizontal [3] as well as the combined horizontal and vertical planes [4]. A cross track error autopilot incorporates the deviation of the assumed straight line path into the control law design. This requires the introduction of additional kinematic relations in the control design and, as a result, the controller tends to be more sensitive to actual system / mathematical model mismatch.

The main drawback of a cross track error autopilot is that it represents a combined guidance / control scheme with no clear distinction between these two functions. Thus it is very vehicle specific and offers little flexibility in the design. Path control is limited to cross track error only and analysis of alternate schemes [5] is not possible unless the combined scheme is redesigned. For this reason we decide to separate once more the guidance and autopilot functions of the vehicle. An orientation controller is designed in order to provide accurate vehicle headings in response to guidance commands. The controller is, thus, based on the vehicle dynamical equations and Euler angle rates. A guidance scheme is used to provide appropriate heading commands through the kinematic equations of inertial position rates. A line of sight guidance command law is employed as in [6] and [7]. We consider a reference point that is moving ahead of the vehicle at a constant distance on the desired straight line path. We refer to this distance as the lookahead distance. The commanded heading is then equal to the line of sight angle between the center of the vehicle and the lookahead point. By suitably selecting the lookahead distance the degree of convergence of the guidance law can be varied from very slow to very rapid onto the straight line path.

Although the above scheme appears to be trouble free on the surface, a significant complication arises in the case of underwater vehicles. Since the actual vehicle response is relatively slow as dominated by the existence of important dynamical lags there is the possibility of instability when the guidance and control functions are combined. High values of the lookahead distance result in very slow vehicle response. The problem is then to evaluate these regions of stable and unstable vehicle response. Chapter II of this thesis summarizes the stability analysis results for the horizontal plane. In Chapter III we proceed with the analysis of motions under the guidance and control scheme for the vertical plane. It is shown that the existence of hydrostatic restoring moments here due to the nonzero (positive) metacentric height brings in an additional form of instability not present in the horizontal plane. Finally in Chapter IV the previous two guidance and control schemes for the horizontal and vertical planes are combined and with a speed autopilot, accurate path tracking in three dimensional space is achieved. The main conclusion of this work is that guidance and control laws for underwater vehicles must be designed together even if they are kept separated, in order to ensure stable and satisfactory path keeping. All computations in this work are performed for the Swimmer Delivery Vehicle [8] for which a complete set of hydrodynamic coefficients and geometric properties is available.

II. HORIZONTAL PLANE

In this section the vehicle equations of motion for the horizontal plane (x,y), the design of a heading autopilot and simulations and stability results are presented.

A. EQUATIONS OF MOTION

For the horizontal plane the mathematical model consists of the nonlinear sway and yaw differential equations shown below:

$$m(\dot{v}+ur+x_G\dot{r}-y_Gr^2)=Y \tag{2.1}$$

$$I_z\dot{r}+mx_G(\dot{v}+ur)-my_Gur=N \tag{2.2}$$

Equations (2.1),(2.2) can be easily derived from the general six degrees of freedom equations for a vehicle by assuming all terms off the horizontal plane to be zero. The equations for the sway force Y and yaw moment N are presented below:

$$Y=Y_{\dot{r}}\dot{r}+(Y_{\dot{v}}\dot{v}+Y_rur)+Y_vuv-\frac{\rho}{2}\int[C_{D_y}h(\xi)\frac{(v+\xi r)^3}{|v+\xi r|}]\,d\xi+Y_\delta u^2\delta$$

$$N=N_{\dot{r}}\dot{r}+(N_{\dot{v}}\dot{v}+N_rur)+N_vuv-\frac{\rho}{2}\int[C_{D_y}h(\xi)\frac{(v+\xi r)^3}{|v+\xi r|}\xi]\,d\xi+N_\delta u^2\delta$$

To complete the model, expressions of the inertial position rates and yaw rate are required. These are the kinematic equations:

$$\dot{\psi}=r \tag{2.3}$$

$$\dot{x}=u\cos\psi-v\sin\psi \tag{2.4}$$

$$\dot{y}=u\sin\psi-v\cos\psi \tag{2.5}$$

B. CONTROL LAW

It is more convenient for the design of a linear state space heading controller to represent the above equations (2.1),(2.2),(2.3) in the following form (with $y_G=0$):

$$\psi=r \tag{2.6}$$

$$\dot{v}=a_{11}uv+a_{12}ur+b_1u^2\delta+d_v(v,r) \tag{2.7}$$

$$\dot{r}=a_{21}uv+a_{22}ur+b_2u^2\delta+d_r(v,r) \tag{2.8}$$

where:

$$D=(I_z-N_{\dot{r}})(m-Y_{\dot{v}})-(mx_G-Y_{\dot{r}})(mx_G-N_{\dot{v}})$$

$$a_{11}=\frac{1}{D}[(I_z-N_{\dot{r}})Y_v-(mx_G-Y_{\dot{r}})N_v]$$

$$a_{12}=\frac{1}{D}\left[(I_z-N_{\dot{r}})(m-Y_r)-(mx_G-Y_{\dot{r}})(-mx_G+N_r)\right]$$

$$a_{21}=\frac{1}{D}\left[(m-Y_{\dot{v}})N_v-(mx_G-N_{\dot{v}})Y_v\right]$$

$$a_{22}=\frac{1}{D}\left[(m-Y_{\dot{v}})(-mx_G+N_r)-(mx_G-N_{\dot{v}})(-m+Y_r)\right]$$

$$b_1=\frac{1}{D}\left[(I_z-N_{\dot{r}})Y_\delta-(mx_G-Y_{\dot{r}})N_\delta\right]$$

$$b_2=\frac{1}{D}\left[(m-Y_{\dot{v}})Y_\delta-(mx_G-N_{\dot{v}})Y_\delta\right]$$

$$d_v(v,r)=-\frac{1}{D}\frac{1}{2}\rho C_{D_y}\left[(I_z-N_{\dot{r}})I_1+Y_{\dot{r}}I_2\right]$$

$$d_r(v,r)=-\frac{1}{D}\frac{1}{2}\rho C_{D_y}\left[(m-Y_{\dot{v}})I_1+N_{\dot{v}}I_2\right]$$

$$I_1=\int\left[h(\xi)(v+\xi r)\,|(v+\xi r)|\right]d\xi$$

$$I_2=\int\left[h(\xi)(v+\xi r)\,|(v+\xi r)|\,\xi\right]d\xi$$

The nonlinear terms $d_v(v,r), d_r(u,r)$ are small and can be neglected for control law design. They are kept, however, in all numerical simulations that follow.

1. ZERO YAW ANGLE

When the commanded yaw angle of the vehicle is zero the control law has the following form:

$$\delta=k_1\psi+k_2v+k_3r \tag{2.9}$$

where k_1, k_2, k_3 are computed so the system will have the desired dynamics. The closed loop characteristic equation has the following form:

$$\lambda^3+a_1\lambda^2+a_2\lambda+a_3=0 \tag{2.10}$$

where:

$$a_1=a_{11}u+a_{22}u+b_1u^2k_2+b_2u^2k_3$$

$$a_2=(a_{11}a_{22}+a_{11}b_2uk_3+b_1a_{22}uk_2-a_{12}a_{21}-a_{12}b_2uk_2-a_{21}b_1uk_3-b_2uk_1)\,u^2$$

$$a_3=(b_2a_{11}-b_1a_{21})\,u^3k_1$$

The characteristic equation is specified in the following way. It can be chosen to satisfy the minimum ITAE criterion where it assumes the form:

$$\lambda^3+\alpha_1\lambda^2+\alpha_2\lambda+\alpha_3=0 \tag{2.11}$$

where:

$$\alpha_1=1.75\omega_0$$

$$\alpha_2=2.15\omega_0^2$$

$$\alpha_3=\omega_0^3$$

$$\omega_c = \frac{10u}{t_H l}$$

and t_H represents the dimensionless settling time for the system. Equating the coefficients of equation (2.10) with the desired equation (2.11) and after some algebra we find:

$$k_1 = \frac{\alpha_3}{(b_2 a_{11} - b_1 a_{21})\, u^3} \tag{2.12}$$

$$k_2 (b_1 a_{22} - b_2 a_{12})\, u^3 + k_3 (b_2 a_{11} - b_1 a_{21})\, u^3 = \alpha_2 + b_2 u^2 k_1 \tag{2.13}$$

$$k_2 b_1 u^2 + k_3 b_2 u^2 = -\alpha_1 - (a_{11} + a_{22})\, u \tag{2.14}$$

Selecting a value for t_H according to the ITAE criterion, dictates complex conjugate dominant poles with oscillatory transient response. It was found that other poles selections (for example real negative) do not change significantly the nature of the results and the stability boundaries that are presented later.

2. NON ZERO YAW ANGLE

If the commanded yaw angle is non zero and equal to ψ_c then the control law (2.9) is simply modified to:

$$\delta = k_1 (\psi - \psi_c) + k_2 v + k_3 r \tag{2.15}$$

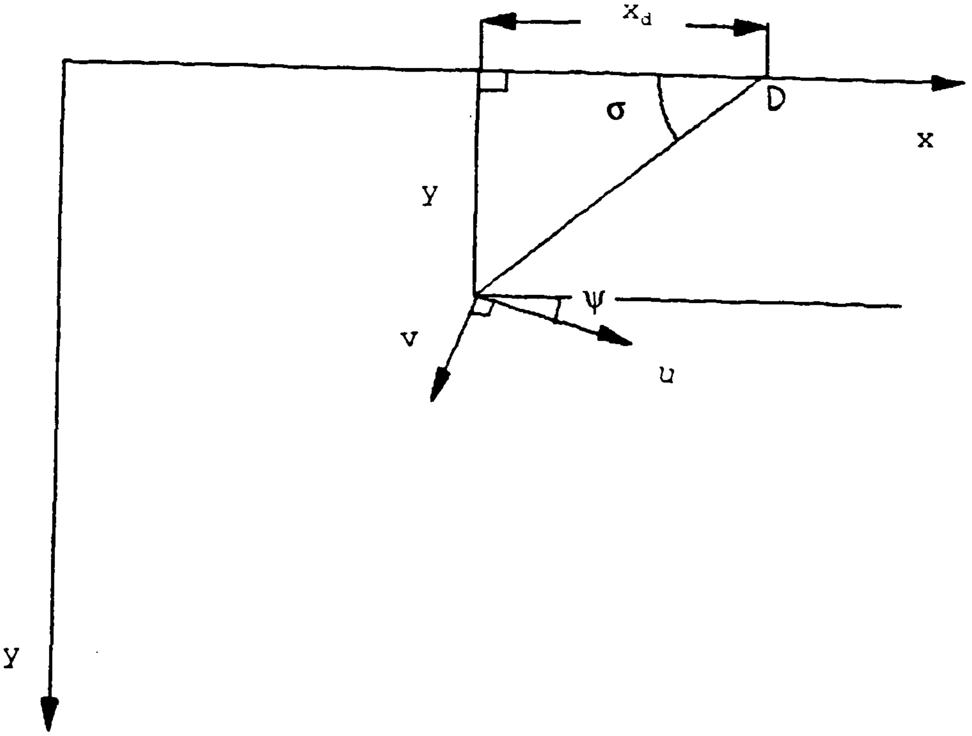

Figure 1. Horizontal plane geometry

No feedforward term is necessary in (2.15) since no rudder angle is required to keep the vehicle to a constant non zero heading angle at steady state.

C. GUIDANCE

The heading autopilot that was designed in the previous section is called upon now to provide vehicle path in the sense of passing through a series of way points in the horizontal plane. In order to achieve it without changing the previously designed heading autopilot we have to couple it with a suitable navigation scheme such as line of sight guidance.

The simplest such guidance law is a pure pursuit navigation which is accomplished as follows. The autopilot attempts to point the longitudinal axis of the vehicle towards a point D which is located ahead to the vehicle on the nominal straight line path at a fixed distance x_d as shown in Figure 1. This target distance x_d to as the visibility, lookahead, or preview distance. The line of sight angle σ is defined by:

$$\tan\sigma = -\frac{y}{x_d} \tag{2.16}$$

Pure pursuit navigation then corresponds to taking:

$$\psi_\sigma = \sigma \tag{2.17}$$

as the commanded heading angle in the control law (2.15).

It can be seen now that the commanded vehicle heading angle is not constant but it is function of the vehicle position y. This introduces the lateral deviation equation (2.5) into the problem, and since the control law was based on equations (2.6),(2.7) and (2.8) only, stability of the combined autopilot-guidance scheme is no longer guaranteed. Therefore, we need to develop conditions which will guarantee stability and ensure satisfactory path keeping.

D. STABILITY

The complete system is given by the differential equations (2.6),(2.7),(2.8), the control law (2.15), and the guidance equations (2.16),(2.17). The trivial equilibrium state corresponding to a straight line motion is characterized by:

$$\psi = v = r = y = 0$$

Linearization of the state equations gives the following linear system:

$$\dot{X} = AX$$

where the complete state vector is:

$$X = [\psi, v, r, y]$$

Local stability properties are established by the eigenvalues of [A] The characteristic equation is found to be:

$$A\lambda^4+B\lambda^3+C\lambda^2+D\lambda+E=0 \tag{2.18}$$

where:

$$A=1$$

$$B=-B_1-C_1$$

$$C=-D_1+B_1C_2-C_1B_2-A_2$$

$$D=-C_1D_2+D_1C_2-uD_2-A_1B_2+A_2B_1$$

and

$$A_1=b_1u^2k_1$$

$$A_2=b_2u^2k_1$$

$$B_1=a_{11}u+b_1u^2k_2$$

$$B_2=a_{21}u+b_2u^2k_2$$

$$C_1=a_{12}u+b_1u^2k_3$$

$$C_2=a_{22}u+b_2u^2k_3$$

$$D_1=b_1u^2k_1\frac{1}{x_d}$$

$$D_2=b_2u^2k_1\frac{1}{x_d}$$

Loss of stability occurs when:

$$BCD-B^2E-AD^2=0 \tag{2.19}$$

Equation (2.19) is derived from Rooth's criterion for (2.18), and it corresponds to a pair of complex conjugate roots crossing the imaginary axis. After some algebra equation (2.19) is simplified to:

$$a_1x_d^2+a_2x_d+a_3=0 \tag{2.20}$$

where:

$$a_1=\alpha_1\alpha_2-\alpha_3$$

$$a_2=\frac{(\alpha_1\alpha_2-2\alpha_3)\,(b_1a_{22}-b_2a_{12}-b_2)}{b_2a_{11}-b_1a_{21}}-\frac{b_1\alpha_1\alpha_3}{(b_2a_{11}-b_1a_{21})\,u}-\alpha_1^2u$$

$$a_3=\frac{-(b_1a_{22}-b_2a_{12}-b_2)\,[b_1\alpha_1+(b_1a_{22}-b_2a_{12}-b_2)\,u]\,\alpha_3}{(b_2a_{11}-b_1a_{21})^2u}$$

The positive root of equation (2.20) determines the critical value of x_d for stability. For every $x_d > x_{d\ critical}$ the system is stable which means that the vehicle will follow the path. In the opposite case where $x_d < x_{d\ critical}$ the system becomes unstable and the motion of the vehicle becomes oscillatory as a result of a complex conjugate pair of eigenvalues with positive real parts.

Results for the dimensionless critical visibility versus settling time t_H are presented in Figure 2. These results are independent of the forward speed since gains k_1, k_2, k_3 are functions of u. It can be seen from Figure 2 that for higher t_H (softer controller) higher lookahead distance x_d is required in order for the system to remain stable. It is obvious that very high values of x_d correspond to a very slow navigator with a loss in speed of response and navigational accuracy. The results of this section establish analytically the minimum required lookahead distance that is required for stability based on linear approximations.

It should be mentioned that all results in this work are presented in dimensionless form unless otherwise mentioned. Nondimensionalizations are performed by using the vehicle length and the vehicle forward speed.

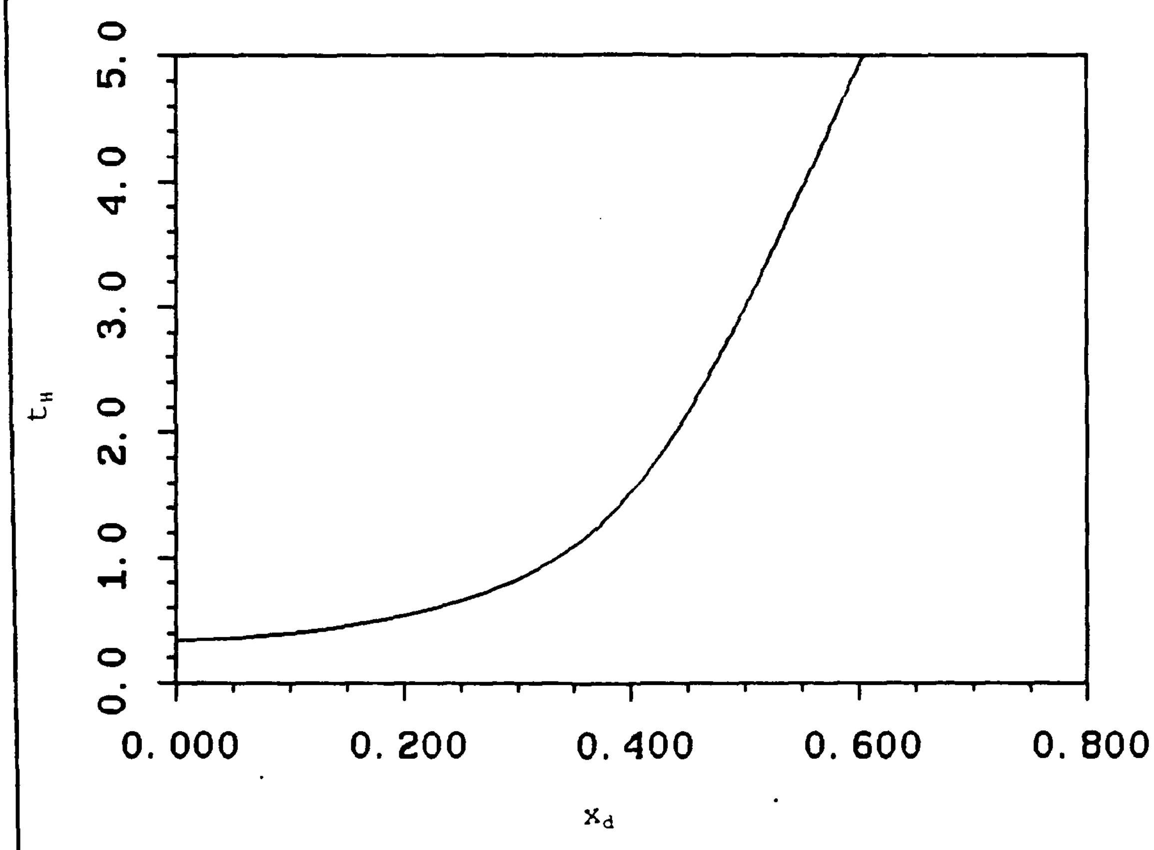

Figure 2. Regions of stability in the horizontal plane

E. SIMULATIONS

Numerical simulations confirm the results of the stability analysis of Figure 2. The simulated lateral distance y (in vehicle lengths) versus time t (in dimensionless seconds) is shown in Figure 3 for two cases. The nominal straight line path is y=0 Case 1 is located in Region 1 of Figure 2 and it can be seen that the vehicle response is unstable. Case 2 corresponds to a stable (t_H, x_d) combination and the vehicle converges to the desired path.

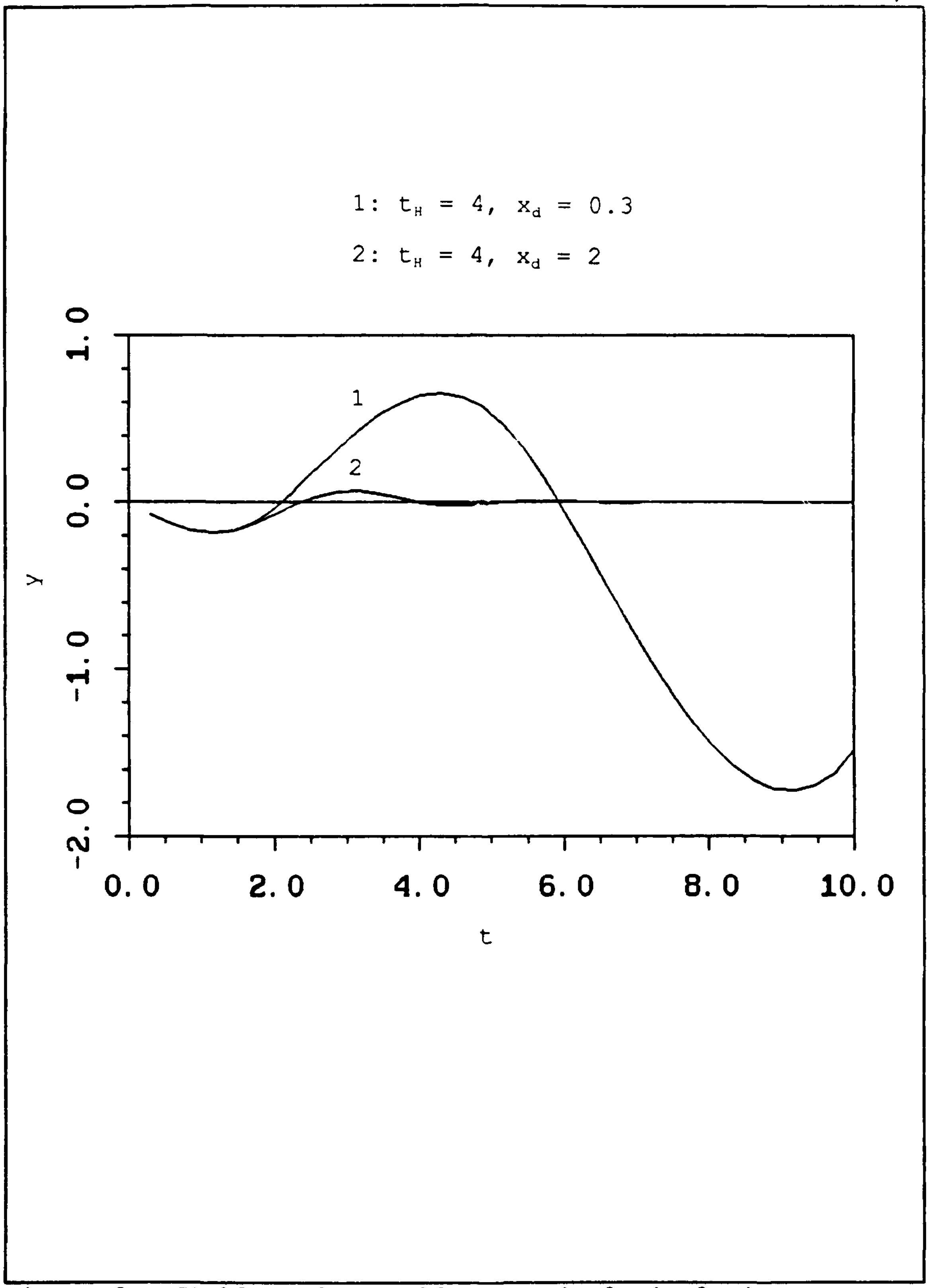

Figure 3. Stable and unstable numerical simulations

III. VERTICAL PLANE

In this section the vehicle equations of motion for the vertical plane (x,z), the design of a vertical heading autopilot and simulations and stability results are presented.

A. EQUATIONS OF MOTION

Restricting our attention to the vertical plane the mathematical model consists of the nonlinear heave and pitch differential equations shown below:

$$m(\dot{w}-uq-x_G\dot{q}-z_Gq^2)=Z \tag{3.1}$$

$$I_y\dot{q}-mx_G(\dot{w}-uq)+mz_Gwq=M \tag{3.2}$$

where only vertical plane related terms have been kept. The heave force Z and pitch moment M are written as:

$$Z=Z_{\dot{q}}\dot{q}+(Z_{\dot{w}}\dot{w}+Z_quq)+Z_wuw-\frac{\rho}{2}\int c_{D_z}b(x)\frac{(w-xq)^3}{|w-xq|}dx+(W-B)\cos\theta+ u^2(Z_{\delta_s}\delta_s+Z_{\delta_b}\delta_b)$$

$$M=M_{\dot{q}}\dot{q}+(M_{\dot{w}}\dot{w}+M_quq)+M_wuw+\frac{\rho}{2}\int c_{D_z}b(x)\frac{(w-xq)^3}{|w-xq|}xdx-(x_GW-x_BB)\cos\theta -(z_GW-z_BB)\sin\theta+u^2(M_{\delta_s}\delta_s+M_{\delta_b}\delta_b)$$

In the above equations is the vehicle weight, B the buoyancy, (x_G,z_G) the coordinates of the center of gravity, and (x_B,z_B)

the coordinates of the center of buoyancy. Also, provision for two sets of control surfaces (stern and bow planes) is made. The kinematic equations are:

$$\dot{x}=u\cos\theta+w\sin\theta \tag{3.3}$$

$$\dot{z}=-u\sin\theta+w\cos\theta \tag{3.4}$$

$$\dot{\theta}=q \tag{3.5}$$

B. CONTROL LAW

The linearized state space form of equations (3.1),(3.2) and (3.5) is used for vertical plane heading control:

$$\dot{w}=a_{11}uw+a_{12}uq+a_{13}\theta+b_{11}u^2\delta_s+b_{12}u^2\delta_b \tag{3.6}$$

$$\dot{q}=a_{21}uw+a_{22}uq+a_{23}\theta+b_{21}u^2\delta_s+b_{22}u^2\delta_b \tag{3.7}$$

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

where:

$$D_v=(m-z_{\dot{w}})\,(I_y-M_{\dot{q}})-(mx_G+Z_{\dot{q}})\,(mx_G+M_{\dot{w}})$$

$$a_{11}=\frac{1}{D_v}\left[(I_y-M_{\dot{q}})\,Z_w+(mx_G+Z_{\dot{q}})\,M_w\right]$$

$$a_{12}=\frac{1}{D_v}\left[(I_y-M_{\dot{q}})\,(m+Z_q)+(mx_G+Z_{\dot{q}})\,(M_q-m)\right]$$

$$a_{13}=-\frac{1}{D_v}[(z_G-z_B)(mx_G+Z_{\dot{q}})W]$$

$$b_{11}=\frac{1}{D_v}[(I_y-M_{\dot{q}})z_{\delta s}+(mx_G+Z_{\dot{q}})M_{\delta s}]$$

$$b_{12}=\frac{1}{D_v}[(I_y-M_{\dot{q}})z_{\delta b}+(mx_G+Z_{\dot{q}})M_{\delta b}]$$

$$a_{21}=\frac{1}{D_v}[(m-Z_{\dot{w}})M_w+(mx_G+M_{\dot{w}})Z_w]$$

$$a_{22}=\frac{1}{D_v}[(m-Z_{\dot{w}})(M_q-m)+(mx_G+M_{\dot{w}})(m+Z_q)]$$

$$a_{23}=-\frac{1}{D_v}[(m-Z_{\dot{w}})(z_G-z_B)W]$$

$$b_{21}=\frac{1}{D_v}[(m-Z_{\dot{w}})M_{\delta s}+(mx_G+M_{\dot{w}})Z_{\delta s}]$$

$$b_{22}=\frac{1}{D_v}[(m-Z_{\dot{w}})M_{\delta b}+(mx_G+M_{\dot{w}})Z_{\delta b}]$$

In these W=B and $x_G=x_B$ have been assumed. Considering that the effect of the bow and the stern planes is the same we have:

$$\delta_s=\delta$$

$$\delta_b=-\delta$$

so

$$b_1=b_{11}-b_{12}$$

$$b_2=b_{21}-b_{22}$$

From the above the final form of the equations of motion is:

$$\theta=q$$

$$\dot{w}=a_{11}uw+a_{12}uq+a_{13}\theta+b_1u^2\delta \qquad \textbf{(3.8)}$$

$$\dot{q}=a_{21}uw+a_{22}uq+a_{23}\theta+b_2u^2\delta \qquad \textbf{(3.9)}$$

1. ZERO PITCH ANGLE

When the commanded direction of the underwater vehicle is horizontal the control law has the following form:

$$\delta=k_1\theta+k_2w+k_3q \qquad \textbf{(3.10)}$$

where k_1, k_2, k_3 are calculated below. From the system of the three differential equations (3.5),(3.8),(3.9) the closed loop characteristic equation has the following form:

$$\lambda^3+a_1\lambda^2+a_2\lambda+a_3=0 \qquad \textbf{(3.11)}$$

where:

$$a_1=-a_{11}u-b_1u^2k_2-a_{22}u-b_2u^2k_3$$

$$a_2=a_{11}a_{22}u^2+a_{11}b_2u^3k_3+a_{22}b_1u^2k_2-a_{12}a_{21}u^2-b_2a_{12}u^3k_2-b_1a_{21}u^3k_3-a_{23}-b_2u$$

$$a_3=a_{13}a_{21}u-a_{13}b_2u^2k_2-b_1a_{21}u^3k_1+a_{11}a_{23}u+a_{11}b_2u^3k_1+a_{23}b_1u^2k_2$$

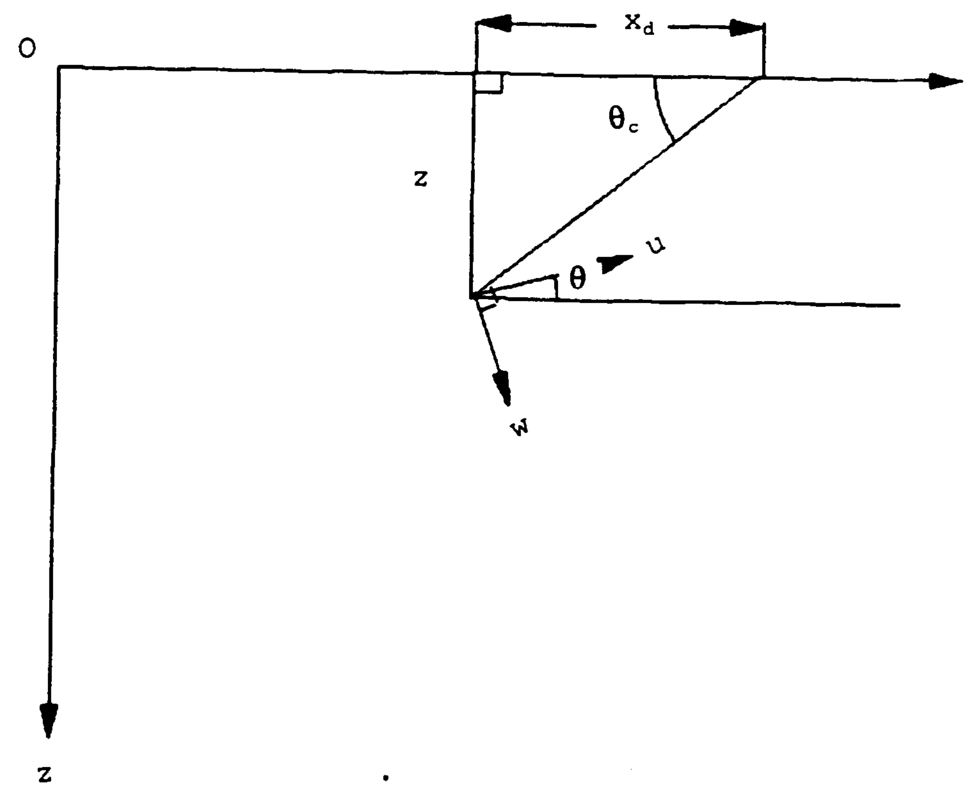

Figure 4. Vertical plane geometry: Horizontal commanded path

The desired characteristic polynomial according to the ITAE criterion is:

$$\lambda^3+\alpha_1\lambda^2+\alpha_2\lambda+\alpha_3=0 \tag{3.12}$$

where:

$$\alpha_1=1.75\omega_0$$

$$\alpha_2=2.15\omega_0^2$$

$$\alpha_3=\omega_0^3$$

$$\omega_0=\frac{10u}{t_v l}$$

and t_v represents the dimensionless settling time for the vertical plane autopilot. Equating the coefficients of equation (3.11) with equation (3.12) we get:

$$b_1u^2k_2+b_2u^2k_3=-\alpha_1-(a_{11}+a_{22})u \tag{3.13}$$

$$\begin{aligned}(b_1a_{22}-b_2a_{12})u^3k_2+(b_2a_{11}-b_1a_{21})u^3k_3=\alpha_2+b_2u^2k_1\\+a_{23}+(a_{12}a_{21}-a_{11}a_{22})u^2\end{aligned} \tag{3.14}$$

$$\begin{aligned}(b_2a_{11}-b_1a_{21})u^3k_1+(a_{23}b_1-a_{13}b_2)u^2k_2=\\\alpha_3+(a_{13}a_{21}-a_{11}a_{23})u\end{aligned} \tag{3.15}$$

To simplify notations, equations (3.13),(3.14) and (3.15) are

written as:

$$A_2k_2+A_3k_3=D_1 \qquad (3.16)$$

$$B_1k_1+B_2k_2+B_3k_3=D_2 \qquad (3.17)$$

$$C_1k_1+C_2k_2=D_3 \qquad (3.18)$$

where:

$$A_2=b_1u^2$$

$$A_3=b_2u^2$$

$$B_1=b_2u^2$$

$$B_2=(b_1a_{22}-b_2a_{12})\,u^3$$

$$B_3=(b_2a_{11}-b_1a_{21})\,u^3$$

$$C_1=(b_2a_{11}-b_1a_{21})\,u^3$$

$$C_2=(a_{23}b_1-a_{13}b_2)\,u^2$$

$$D_1=-\alpha_1-(-a_{11}+a_{22})\,u$$

$$D_2=\alpha_2+a_{23}+(a_{12}a_{21}-a_{11}a_{22})\,u^2$$

$$D_3=\alpha_3+(a_{13}a_{21}-a_{11}a_{23})\,u$$

From the above system of equations (3.16),(3.17),(3.18) we can find expressions for the gains k_1,k_2,k_3

$$k_1 = \frac{D_3 - C_2 k_2}{C_1} \tag{3.19}$$

$$k_2 = \frac{A_3 B_1 D_3 + C_1 B_3 D_1 - D_2 C_1 A_3}{A_3 B_1 C_2 + C_1 B_3 A_2 - C_1 A_3 B_2} \tag{3.20}$$

$$k_3 = \frac{D_1 - A_2 k_2}{A_3} \tag{3.21}$$

2. NON ZERO PITCH ANGLE

When the commanded pitch angle of the vehicle is not equal to zero, we have:

$$\theta = a_V + \theta' \tag{3.22}$$

where: a_v is the commanded pitch angle

θ' is the deviation from the commanded angle

Then

$$\sin\theta = \sin a_v \cos\theta' + \cos a_v \sin\theta' = \sin a_v + \theta' \cos a_v$$

for small deviations θ'. The system of equations of motion (3.5),(3.8),(3.9) takes the form:

$$\theta' = q \tag{3.23}$$

$$\dot{w} = a_{11} uw + a_{12} uq + a_{13} \cos a_v \theta' + b_1 u^2 \delta + a_{13} \sin a_v \tag{3.24}$$

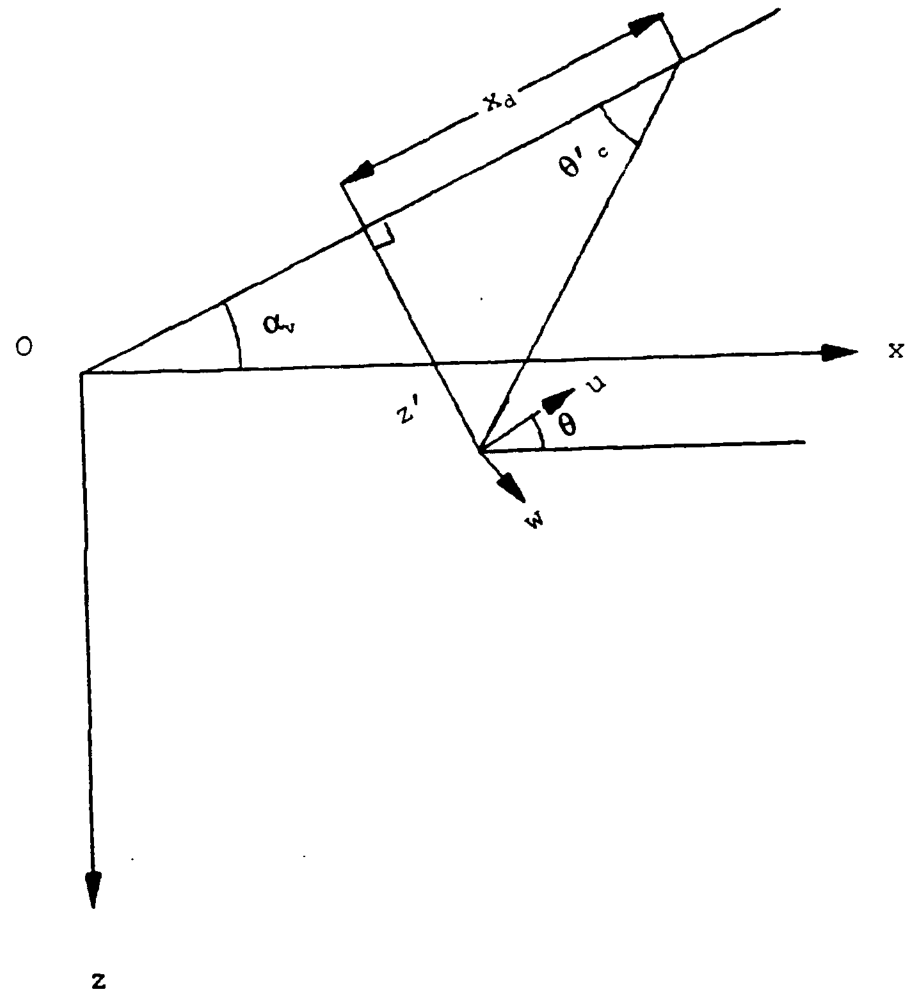

Figure 5. Vertical plane geometry: Inclined commanded path

$$\dot{q}=a_{21}uw+a_{22}uq+a_{23}\cos a_v\theta'+b_2u^2\delta+a_{23}\sin a_v \tag{3.25}$$

The control law now takes the form:

$$\delta=k_1(\theta-a_v)+k_2w+k_3q+k_4 \tag{3.26}$$

where k_1,k_2,k_3 can be calculated with the some procedure as before, and the feedforward gain k_4 is calculated from the desired steady state accuracy. At steady state we have:

$$q=0$$

$$\theta=a_v$$

$$\theta'=0$$

so that the system of the equations of motion (3.23),(3.24),(3.25) yields:

$$a_{11}uw+b_1u^2\delta+a_{13}\sin a_v=0 \tag{3.27}$$

$$a_{21}uw+b_2u^2\delta+a_{23}\sin a_v=0 \tag{3.28}$$

Equations (3.27),(3.28) can be solved for the steady state values of δ and w, and by substitution into equation (3.26), after some calculations k_4 is found to be:

$$k_4=-\frac{a_{13}(a_{21}+b_2uk_2)-a_{23}(a_{11}+b_1uk_2)}{(b_1a_{21}-b_2a_{11})u^2}\sin a_v \qquad \textbf{(3.29)}$$

Note that if $a_v=0$ or $z_G=z_B$ then $k_4=0$.

C. GUIDANCE LAW

A similar to the horizontal plane case guidance law can be used here to allow path keeping in the vertical plane. To the previous system of differential equations (3.1),(3.2),(3.5) one more equation is added, the kinematic equation (3.4). The new system is now going to be examined for two different cases.

a) Horizontal path (no change in depth)

b) Inclined path (change in depth)

1. HORIZONTAL PATH

In this case where the commanded depth remains the same the control law is:

$$\delta=k_1(\theta-\theta_c)+k_2w+k_3q \qquad \textbf{(3.30)}$$

where θ_c is the commanded line of sight (pitch angle)

$$\theta_c=\tan^{-1}\frac{z}{x_d} \qquad \textbf{(3.31)}$$

where k_1, k_2, k_3 are already known from the previous section, and x_d is the visibility distance similar to the horizontal case, shown in Figure 4.

2. INCLINED PATH

Here the commanded depth changes linearly so that the angle δ is given by:

$$\delta = k_1 (\theta - a_v - \theta'_c) + k_2 w + k_3 q + k_4 \tag{3.32}$$

where k_1, k_2, k_3, k_4 are the same as previously determined. The k_4 term exists here because an angle $\delta \neq 0$ has to remain when the underwater vehicle changes depth to equalize the restoring moment due to the pitch angle. The commanded pitch angle is:

$$\theta_c' = \tan^{-1} \frac{z'}{x_d} \tag{3.33}$$

where z' is the cross track error off the inclined path as shown in Figure 5.

D. STABILITY

The complete system is given by the equations of motion (3.1), (3.2), (3.4), (3.5), the control law (3.30), and the guidance law (3.31). Horizontal motion at the commanded depth is characterized by:

$$\theta = w = q = z = 0$$

Linearization of the above equations produces the linear system:

$$\dot{X}=AX$$

where:

$$X=[\theta, w, q, z]$$

and

$$A=\begin{Vmatrix} 0 & 0 & 1 & 0 \\ a_{13}z_{GB}+b_1u^2K_1 & a_{11}u+b_1u^2K_2 & a_{12}u+b_1u^2K_3 & -b_1u^2\dfrac{K_1}{x_d} \\ a_{23}z_{GB}+b_2u^2K_1 & a_{21}u+b_2u^2K_2 & a_{22}u+b_2u^2K_3 & -b_2u^2\dfrac{K_1}{x_d} \\ -u & 1 & 0 & 0 \end{Vmatrix} \qquad \textbf{(3.34)}$$

where:

$$z_{GB}=z_G-z_B \qquad \textbf{(3.35)}$$

is the metacentric height. Stability properties of the straight line motion are established by the eigenvalues of matrix [A]. It should be mentioned that from now until the end of this chapter a_{13}, a_{23} have been redefined to show explicitly the metacentric height Z_{GB}.

A program is written to compute the eigenvalues of matrix (3.34) over a range of (t_v, x_d) values, and detect whether one or more eigenvalues become unstable. Typical results are shown in Figure 6 for u=5 ft/sec and z_{GB}=0.1.

1. REGIONS OF STABILITY

It can be seen that the stability boundary of Figure 6 separates the parameter space (x_d,t_v) into three regions:

1: Unstable region, one pair of complex conjugate eigenvalues of [A] has positive real parts.

2: Stable region, all eigevalues of [A] have negative real parts.

3: Unstable region, one real positive eigenvalue of [A].

Obviously, stable vehicle response is not possible unless the parameters (x_d,t_v) are chosen in region 2.

1: One pair of complex conjugate eigevalues with positive real parts.

2: Region of stability.

3: One real positive eigenvalue.

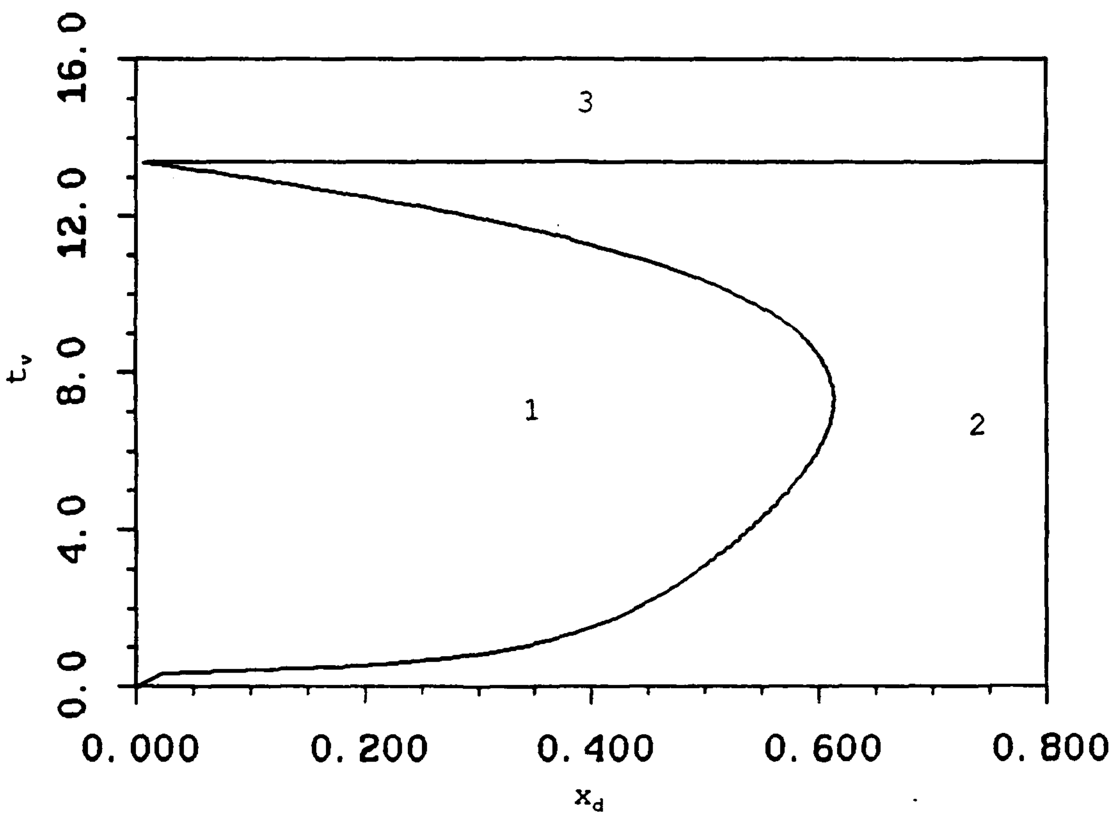

Figure 6. Regions of stability for u=5 ft/sec and z_{CB}=0.1 ft

2. SIMULATIONS

Before proceeding further with the stability analysis, numerical integrations are first performed in order to examine the response of the vehicle in each of the above three regions of stability of Figure 6. The same parameters u=5 ft/sec and z_{GB}=0.1 ft are used. Simulations for the pitch angle θ and the commanded line of sight angle θ_c for the case t_v=5 and x_d=4 are shown in Figure 7. This corresponds to region 2 of Figure 6 which is the region of stability. The simulation results show that the actual vehicle pitch angle approaches the commanded angle, after some oscillations, and the depth reaches its commanded value at zero as predicted.

When the visibility distance is x_d=0.4 with the same t_v, the vehicle moves into the unstable region 1 of Figure 6. The simulated response is shown in Figure 8 where oscillatory characteristics are exhibited. If we keep the same value for x_d=4 and we change the controller time constant t_v=15 we enter the unstable region 3 of Figure 6. The simulated vehicle response is shown in Figure 9 where it appears that θ and θ_c diverge and they both reach nonzero steady state values. As a result the vehicle depth is now a linear function of time, without ever stabilizing. These results require a more detailed analysis of the regions of stability of the controller / guidance combination.

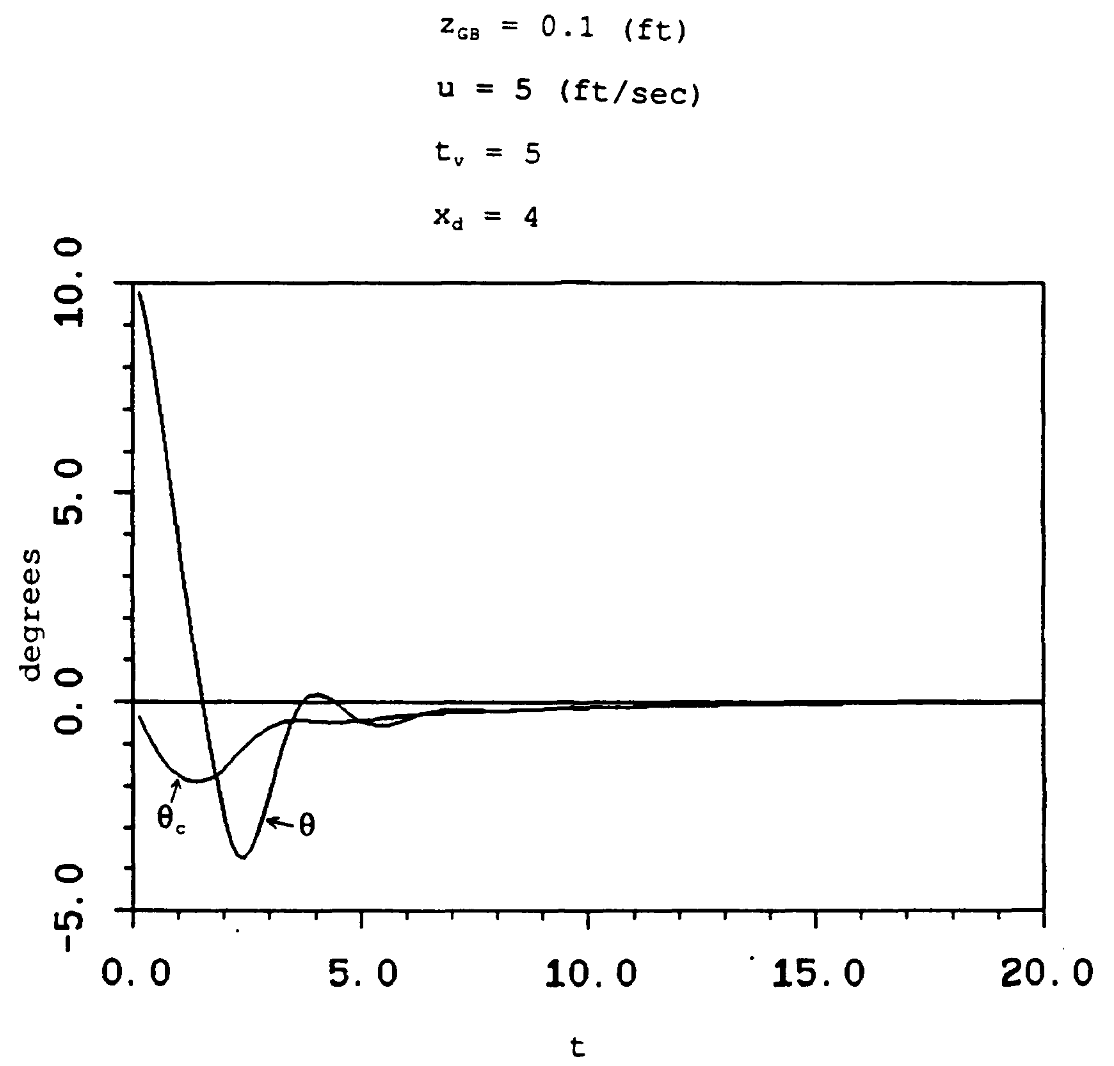

Figure 7. Numerical simulations in region 2

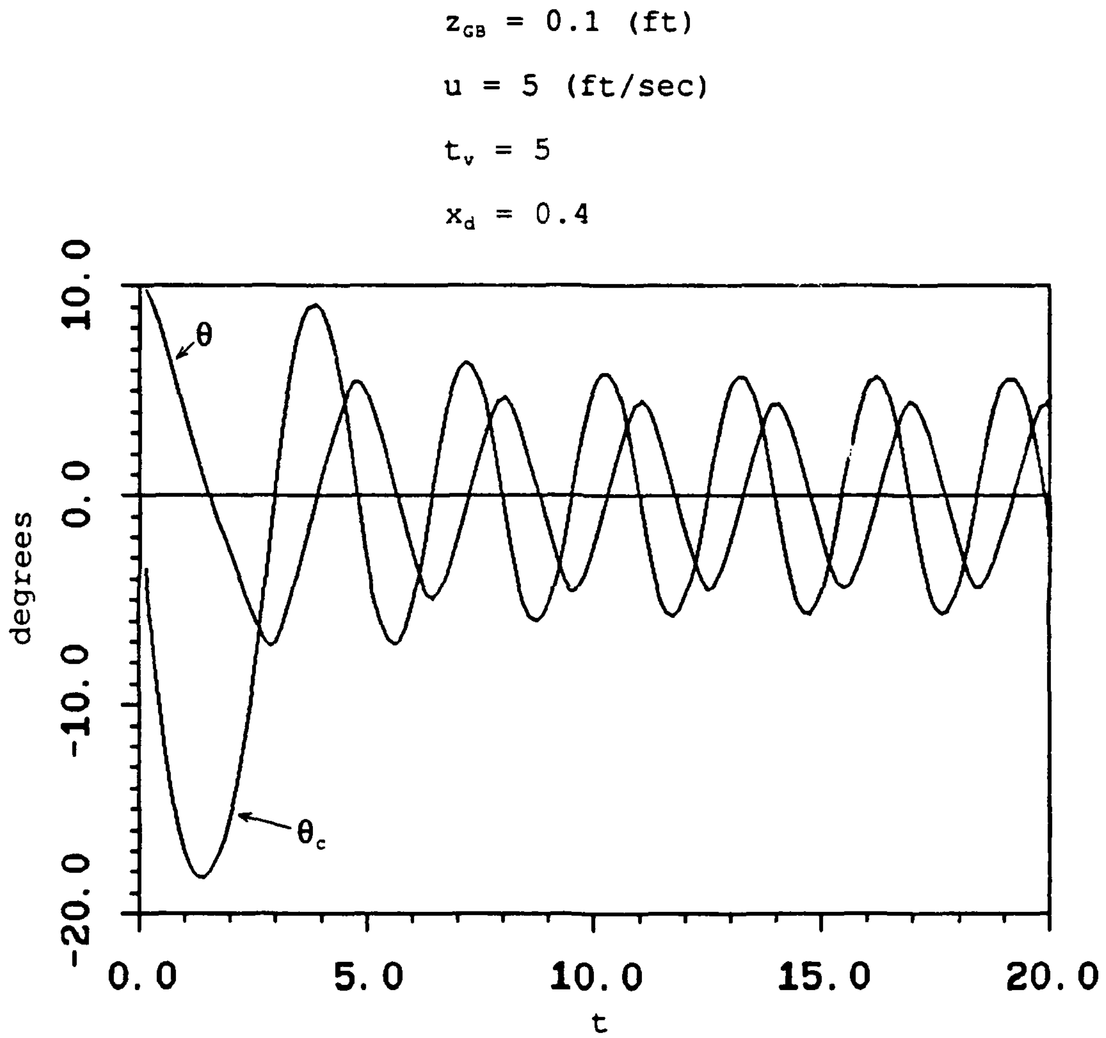

Figure 8. Numerical simulations in region 1

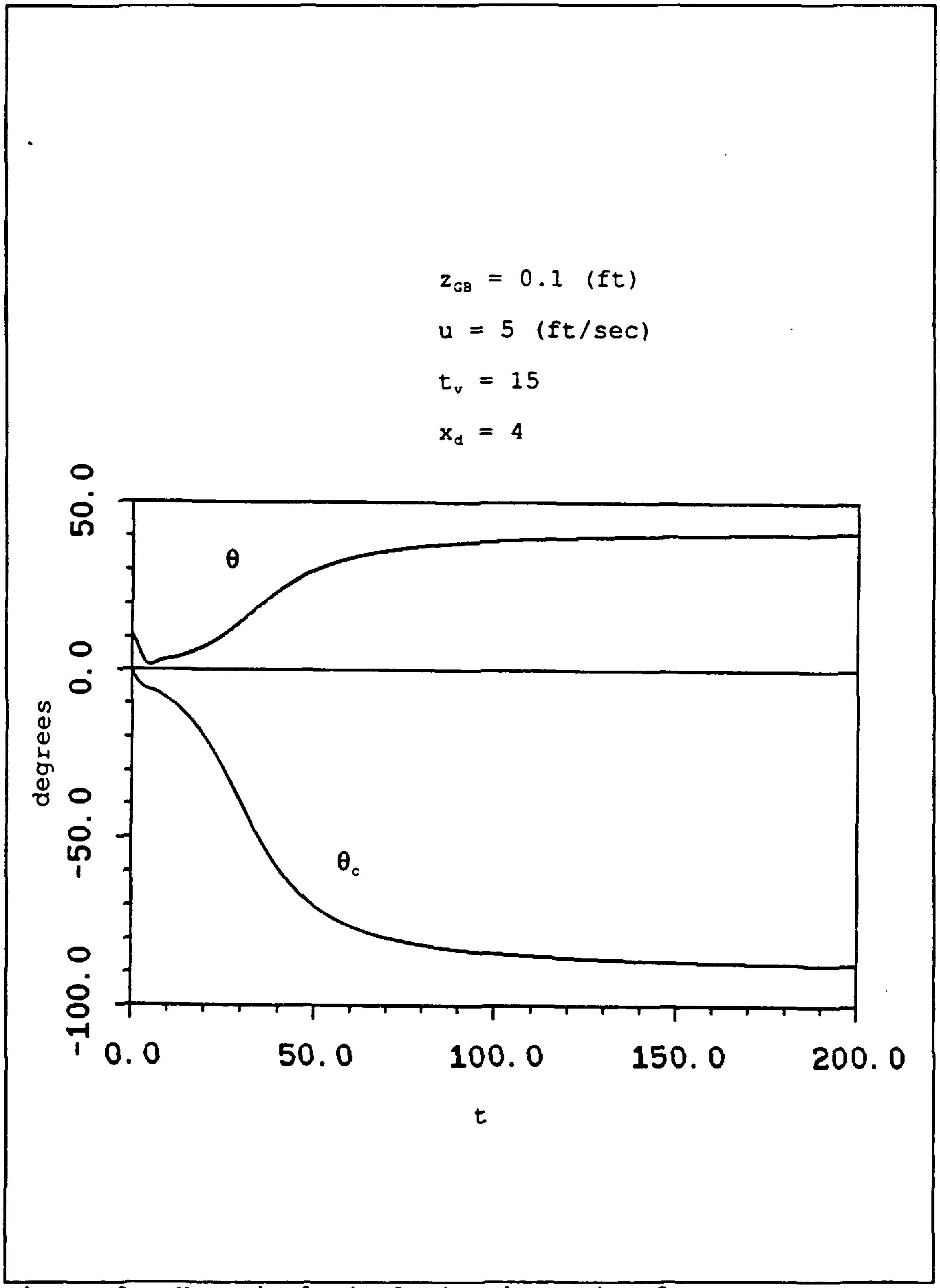

Figure 9. Numerical simulation in region 3

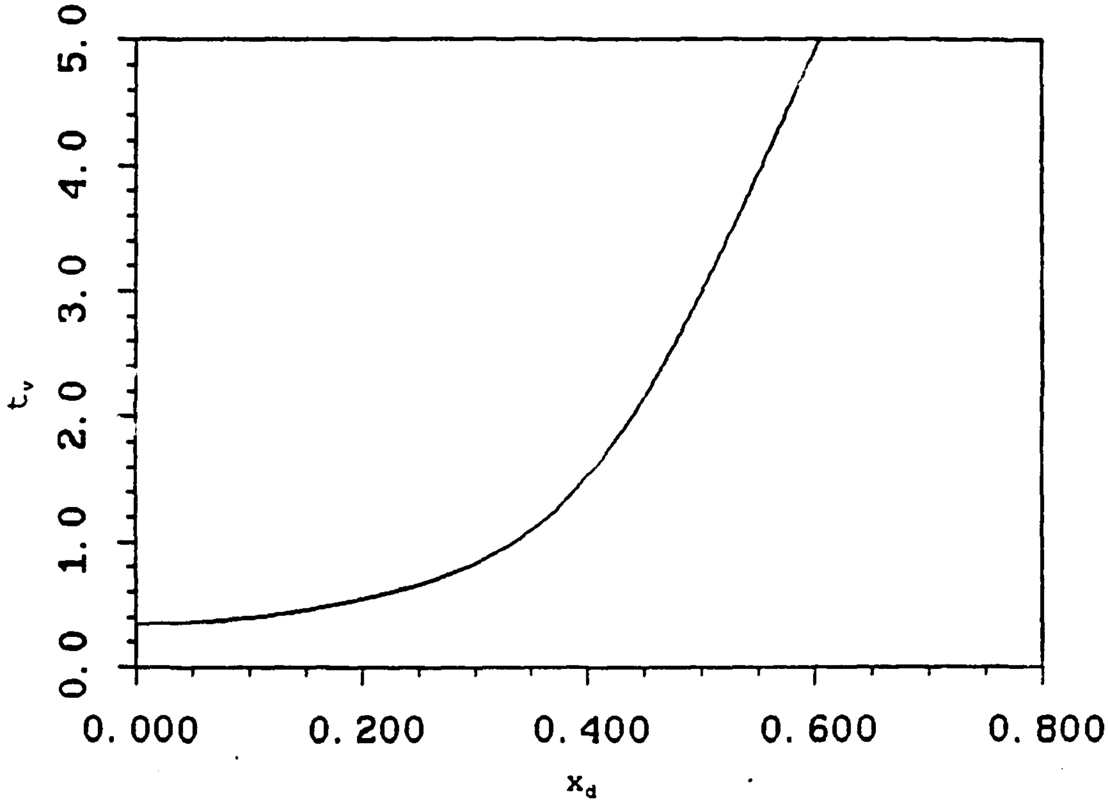

Figure 10. Regions of stability for $z_{CB}=0$ and for any speed u

E. ANALYSIS

The stability properties of the system are characterized by the eigenvalues of the linearized matrix [A], given by equation (3.34). The characteristic equation of [A] has the form:

$$A\lambda^4+B\lambda^3+C\lambda^2+D\lambda+E=0 \tag{3.36}$$

where:

$$A=1$$

$$B=\alpha_1$$

$$C=\alpha_2+b_1u^2k_1\frac{1}{x_d}$$

$$D=\alpha_3+(b_2a_{12}-b_1a_{22})\,u^3k_1\frac{1}{x_d}-b_2u^3k_1\frac{1}{x_d}$$

$$E=z_{GB}(b_2a_{13}-b_1a_{23})\,u^2k_1\frac{1}{x_d}+(b_2a_{11}-b_1a_{21})\,u^4k_1\frac{1}{x_d}$$

According to Routh's criterion (3.36) has one pair of complex conjugate roots crossing the imaginary axis when:

$$BCD-AD^2-B^2E=0 \tag{3.37}$$

After some algebra , equation (3.37) can be written as :

$$\alpha_3(\alpha_1\alpha_2-\alpha_3)x_d^2+[d_1(\alpha_1\alpha_2-\alpha_3)+\alpha_1 c_1\alpha_3-\alpha_3 d_1-\alpha_1^2 e_1] \quad \textbf{(3.38)}$$
$$x_d+d_1(\alpha_1 c_1-d_1)=0$$

where:

$$c_1=A_2 k_1$$

$$d_1=(-B_2-A_3 u)k_1$$

$$e_1=(-C_2+C_1 u)k_1$$

and A_2,A_3,B_2,C_1,C_2 were defined previously following equations (3.16),(3.17) and (3.18).

The positive root of equation (3.38) provides the critical value of x_d for stability. This produces the curve separating the regions 1 and 2 of Figure 6. As the (x_d,t_v) combinations cross into region 1, the response of the system becomes oscillatory as a result of the pair of complex conjugate eigenvalues with positive real part. This explains the simulations observed in Figures 7 and 8.

A different kind of instability occurs when one real root of (3.36) crosses zero. For this to happen the condition is:

$$E=0 \quad \textbf{(3.39)}$$

and using the previous definition of E, this happens when

$$k_1 = 0 \tag{3.40}$$

Equations (3.40) and (3.19) yield

$$k_2 = \frac{D_3}{C_2} \tag{3.41}$$

Equations (3.41),(3.20) define then the critical condition for stability. In our case this can be simplified as follows. The expression for C_2 is reproduced here.

$$C_2 = (a_{23}b_1 - a_{13}b_2)\, z_{GB}u^2 \tag{3.42}$$

Since we have assumed that bow and stern planes have the same strength

$$Z_{\delta s} = Z_{\delta b} < 0$$

$$M_{\delta s} = -M_{\delta b} < 0$$

and substituting the expressions for a_{23}, b_1, a_{13}, b_2 we can find that

$$C_2 = 0 \tag{3.43}$$

Equations (3.43) and (3.41) then require that

$$D_3=0 \tag{3.44}$$

and using the definition for D_3 we get

$$\alpha_3+(a_{13}a_{21}-a_{11}a_{23})\,z_{GB}u=0$$

or

$$\left(\frac{10u^3}{t_v l}\right)^3+(a_{13}a_{21}-a_{11}a_{23})\,z_{GB}=0$$

and we can find the critical value of t_v as

$$t_{v_{critical}}=\frac{10u}{l\,[(a_{11}a_{23}-a_{13}a_{21})\,z_{GB}]^{\frac{1}{3}}} \tag{3.45}$$

Condition (3.45) shows that the critical value of t_v is independent of x_d which is demonstrated in Figure 6 as a straight line parallel to the x_d axis. Furthermore, the other stability curve, equation (3.38), intersects the t_v axis at $x_d=0$ when $k_1=0$ which is the same condition as (3.45). This means that the stability conditions (3.38) and (3.45) separate the (x_d,t_v) parameter space into three regions of stability, as shown in Figure 6.

Results of the stability regions for $z_{GB}=0$ are shown in Figure 10. These are independent of the forward speed u just as in the horizontal plane case. It should be mentioned that for $z_{GB}=0$, $t_{vcritical} \rightarrow \infty$ and therefore ,region 3 of figure 6 never appears.

For $z_{GB} > 0$, the stability regions depend heavily on the forward speed u. This is demonstrated in Figure 11 for $z_{GB}=0.1$ (ft) and various values of u in (ft/sec). As the speed is decreased the critical value of t_v from (3.45) also decreases with the effect of reducing region 1 and enlarging region 2 and 3.

The effect of varying the metacentric height z_{GB} while keeping u constant is evaluated in Figure 12 for u=2. Similar conclusions can be drawn for this case as previously.

The critical value of t_v as given by (3.45) is shown in Figure 13 for different values of the forward speed u and the metacentric height z_{GB}. The surface shown in the figure separates the stability regions 2 and 3.

The final task of this section is to explain the simulations observed in Figure 9 when the vehicle operates in region 3.

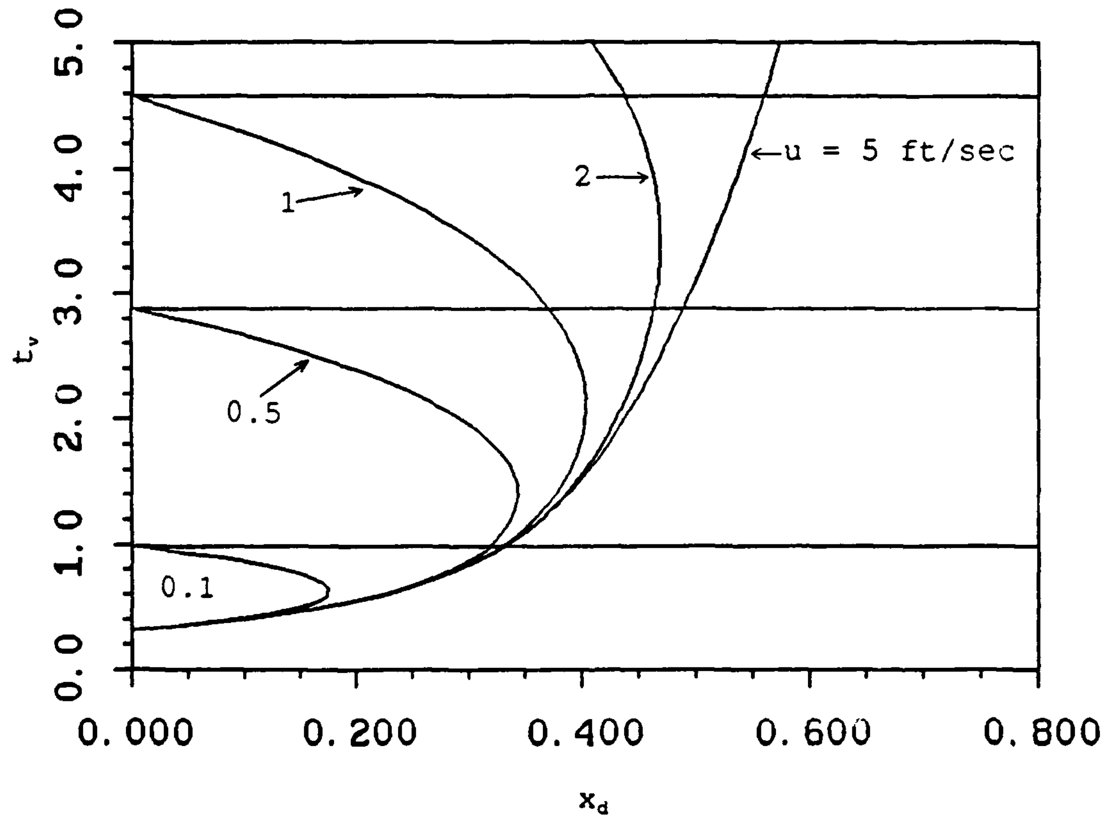

Figure 11. Regions of stability for z_{GB}=0.1 ft

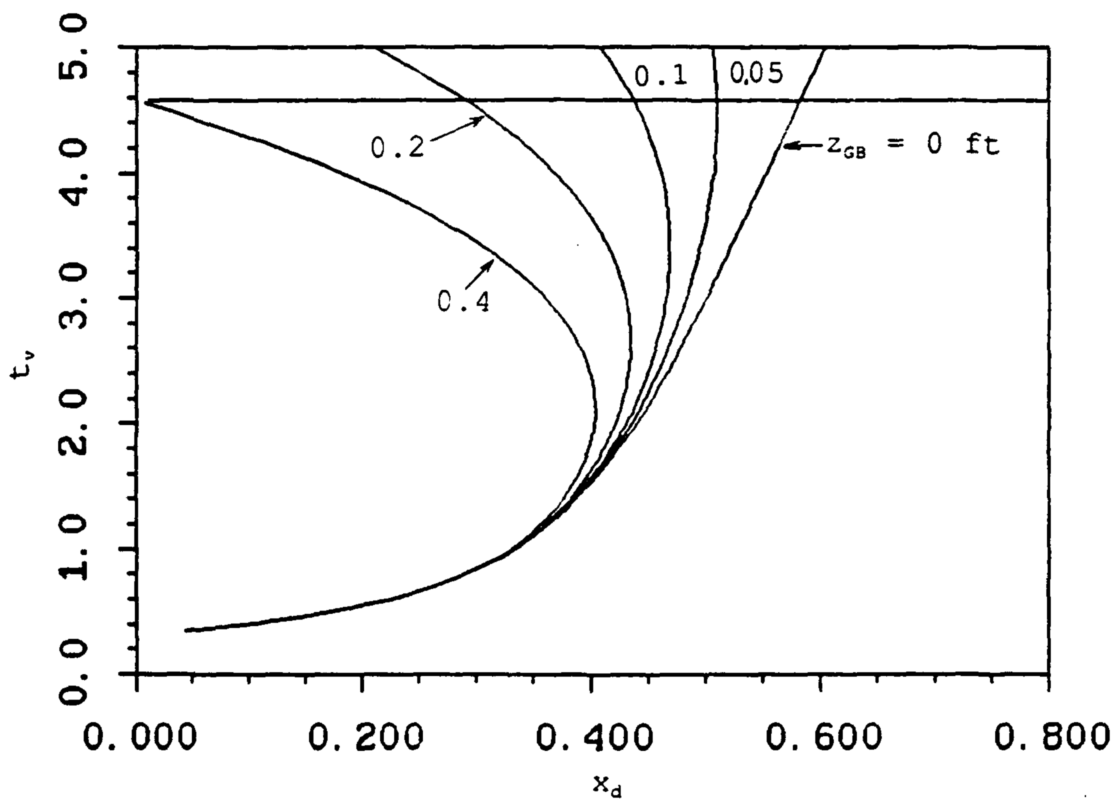

Figure 12. Regions of stability for u = 2 ft/sec

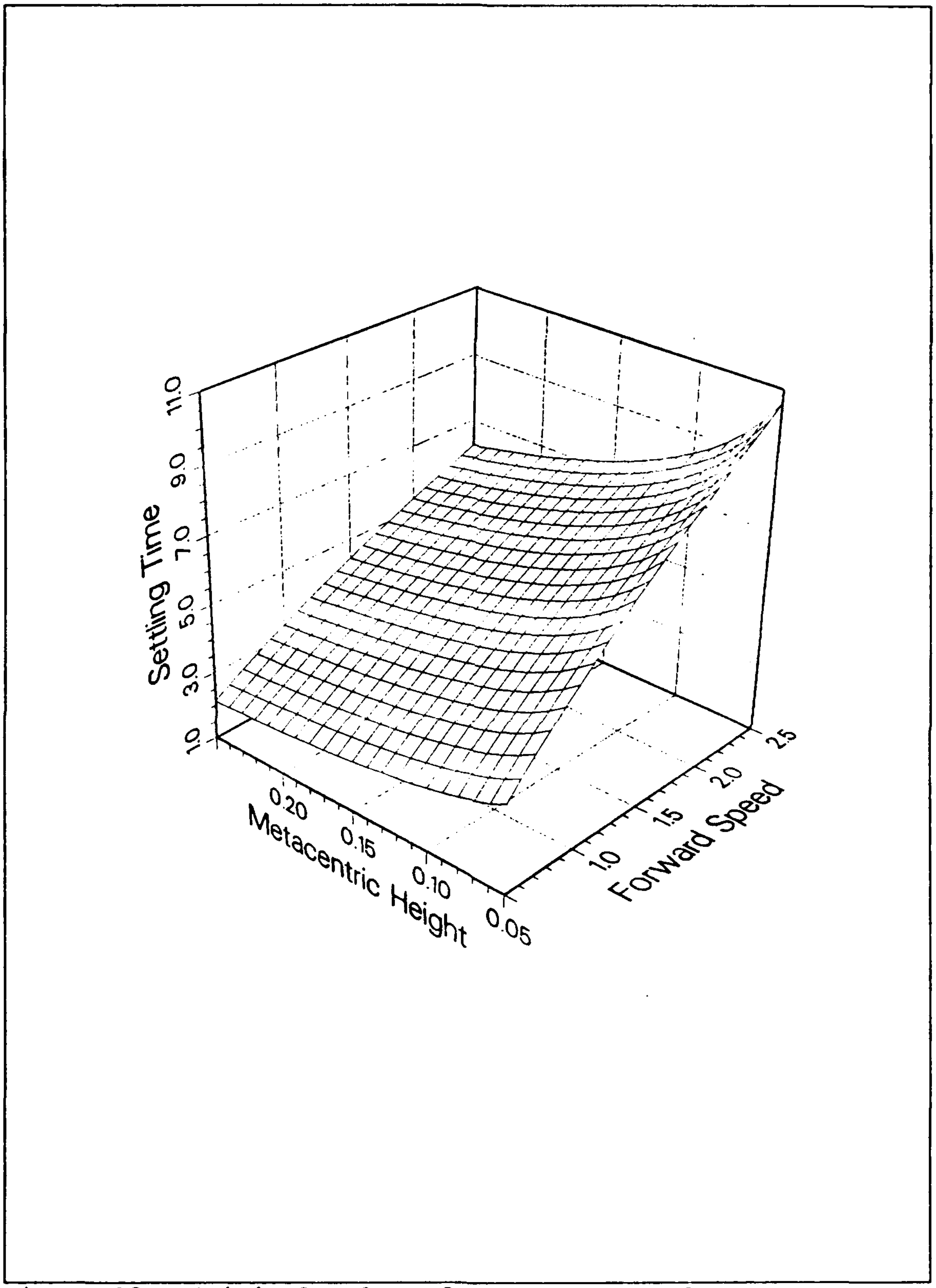

Figure 13. Critical value of t_v versus u and z_{GB}

F. STEADY STATE SOLUTIONS

It was shown in the previous section that transition between Regions 2 and 3 is associated with a real eigenvalue crossing zero. Usually when such a loss of stability occurs and the primary equilibrium solution becomes unstable, additional stable equilibrium solutions appear. To evaluate these new steady state solutions we consider the complete system given by equations (3.4), (3.5), (3.6), (3.7), (3.30), and (3.31). At steady state the time derivatives vanish and we get

$$q=0 \tag{3.46}$$

$$w=\frac{C_2}{C_1}\sin\theta \tag{3.47}$$

$$\delta=\frac{D_3-\alpha_3}{C_1}\sin\theta \tag{3.48}$$

Substituting equations (3.46),(3.47), and (3.48) into equation (3.4) we get:

$$\left(-u+\frac{C_2}{C_1}\cos\theta\right)\sin\theta=0 \tag{3.49}$$

Equation (3.49) may accept besides the normal solution $\theta=0$, another solution given by:

$$\cos\theta = \frac{C_1}{C_2}u = \frac{(b_2a_{11}-b_1a_{21})\,u^2}{(b_1a_{23}-b_2a_{13})\,z_{GB}} \tag{3.50}$$

Equation (3.50) is valid provided $\cos\Theta \leq 1$ which means:

$$u^2 \leq \frac{(b_1a_{23}-b_2a_{13})\,z_{GB}}{b_2a_{11}-b_1a_{21}} \tag{3.51}$$

If (3.51) is satisfied the equilibrium angle Θ can be determined from (3.50) provided:

$$\frac{D_3-\alpha_3}{C_1}\sin\theta \leq \delta_{sat} \tag{3.52}$$

where δ_{sat} is the maximum dive plane angle typically set at 0.4 radians.

In our case conditions (3.51) and (3.52) are not satisfied, which means that the non zero equilibrium pitch angle cannot be computed from (3.50). Furthermore $z \neq 0$ at steady state, which means that $\dot{z}$=constant. Therefore, z is linearly increasing with time, and

$$\tan^{-1}\frac{z}{x_d} \to \frac{\pi}{2},\ as\ldots t\to\infty \tag{3.53}$$

Substituting equations (3.46), (3.47), (3.48), and (3.53) into the control law (3.30) and (3.31) we can find the equation for the unknown steady state pitch angle.

$$(D_3-\alpha_3)\sin\theta = k_1 C_1 (\theta - \frac{\pi}{2}) + k_2 C_2 \sin\theta \tag{3.54}$$

If we call $\theta - \pi/2 = \theta'$, equation (3.54) reduces to:

$$k_1 C_1 \theta' + (\alpha_3 - k_1 C_1)\cos\theta' = 0 \tag{3.55}$$

It can now be seen that equation (3.55) has a solution when k_1 crosses zero which is the same condition for transition between regions (2) and (3) found in the previous section. The steady state solution is then computed from (3.55) if:

$$\frac{D_3-\alpha_3}{C_1}\sin\theta \le \delta_{sat} \tag{3.56}$$

or from:

$$\sin\theta = \frac{C_1}{D_3-\alpha_3}\delta_{sat} \tag{3.57}$$

otherwise.

Results for the steady values of θ and δ are presented in Figures 14 and 15 versus z_{GB} for $u=5$ and $t_v=15$.

Solid lines correspond to stable and dashed lines to unstable equilibrium positions. It can be seen that the simulation results for Z_{GB} =0.1 of Figure 9 are verified.

The steady state pitch angle θ=0 loses its stability at Z_{GB}=0.07 and begins to increase together with the dive plane angle δ. This is up to Z_{GB}=0.12 where δ reaches its maximum value. For increasing Z_{GB} beyond this point, the pitch angle θ begins to decrease since δ remains constant. These results are for fixed t_v and u. Results for different values of the controller settling time t_v and vehicle speed u are shown in Figures 16 and 17 respectively.

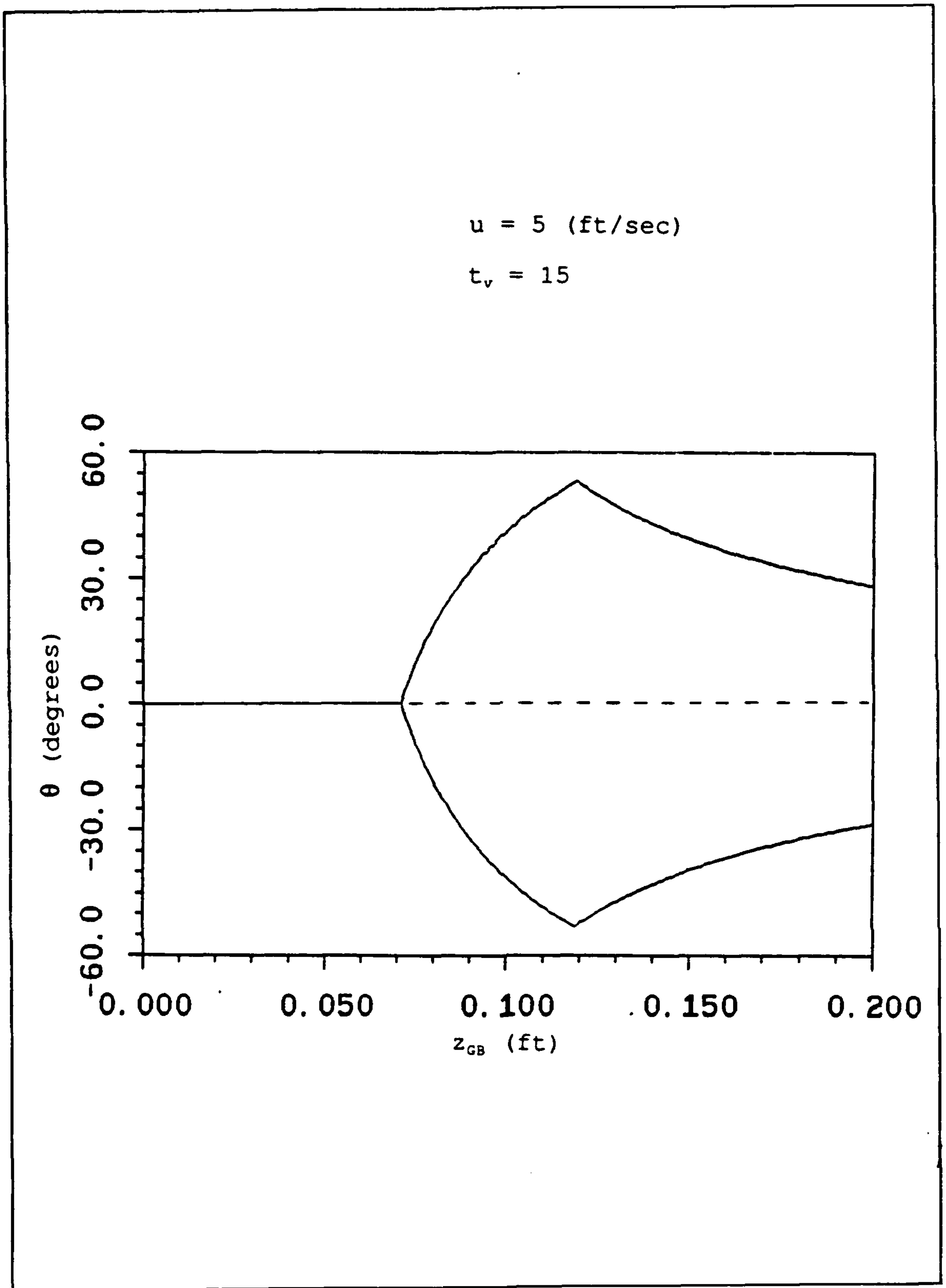

Figure 14. Steady state pitch angle θ versus z_{GB}

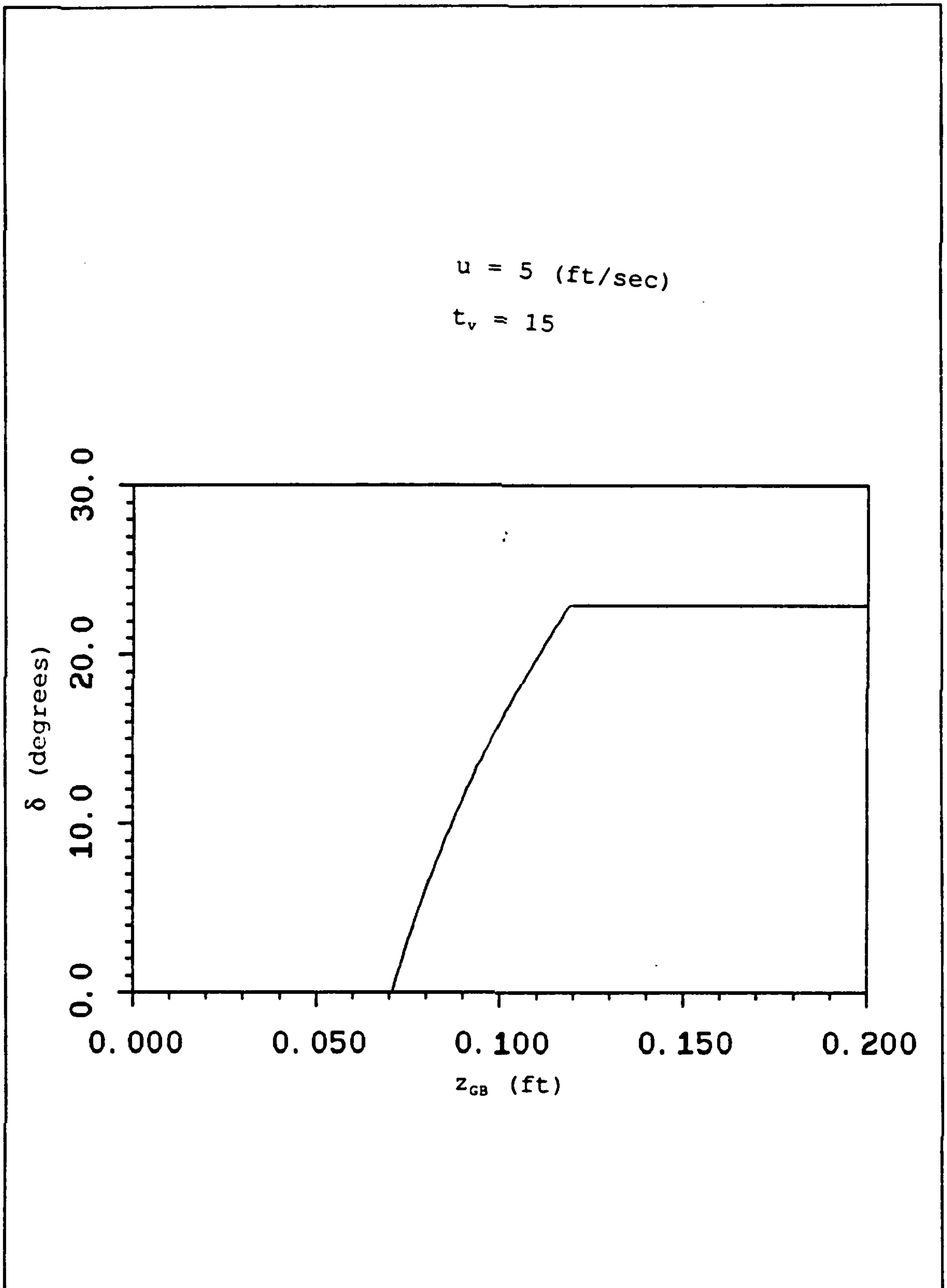

Figure 15. Steady state dive plane angle δ versus z_{GB}

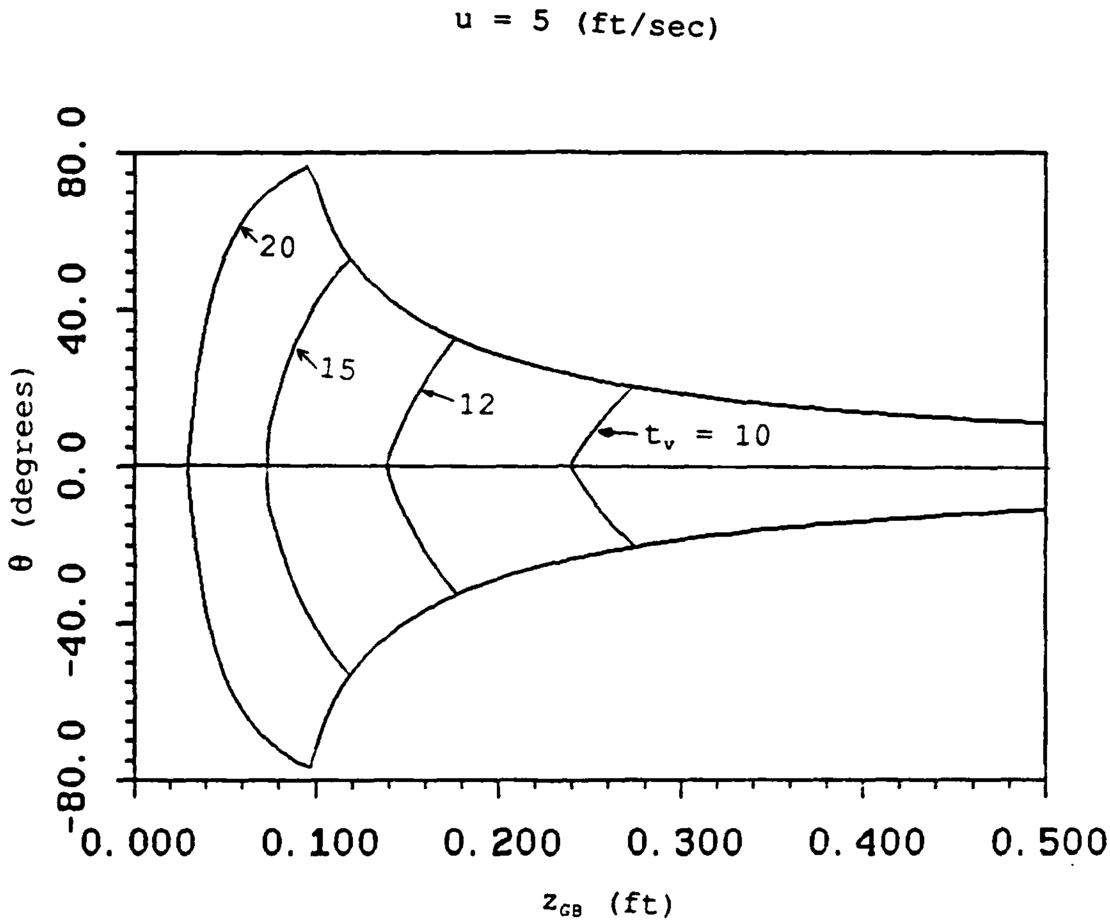

Figure 16. Steady state θ versus z_{GB} for several values of t_v

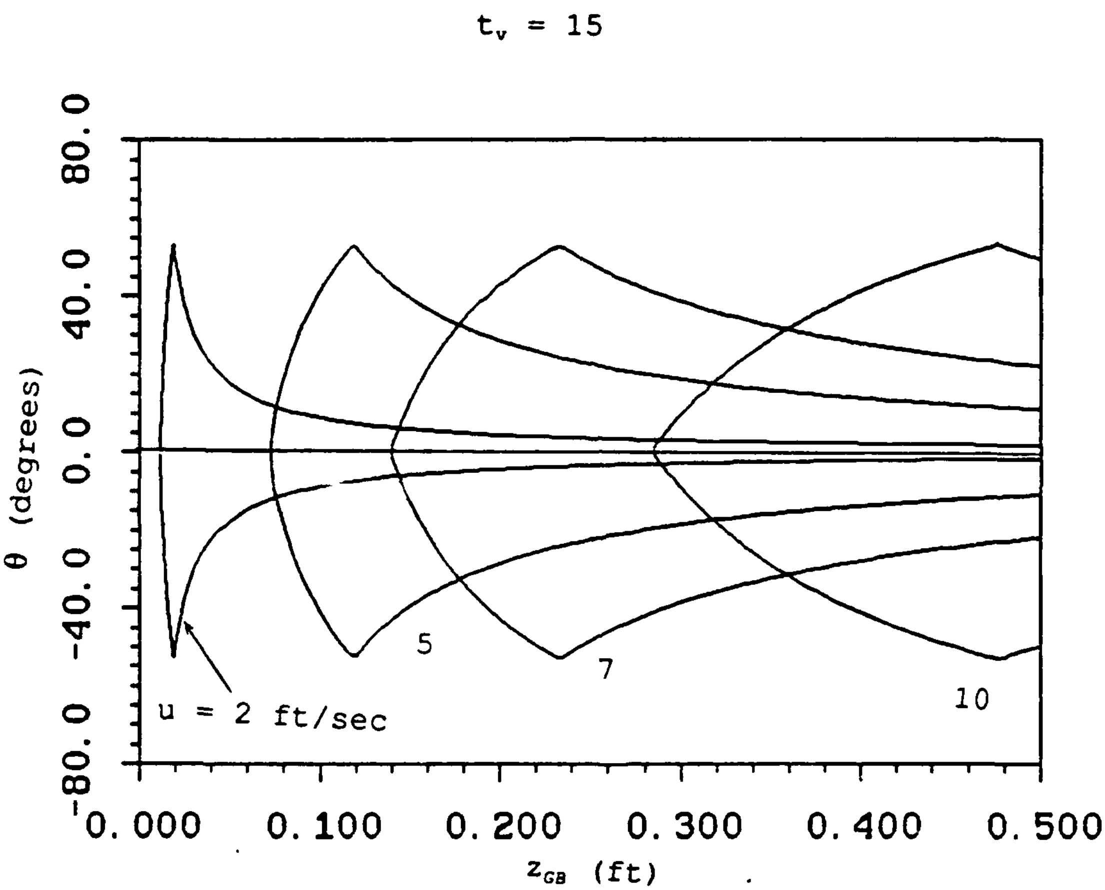

Figure 17. Steady state θ versus z_{GB} for several values of u

IV THREE DIMENSIONAL GUIDANCE CONTROL

The horizontal and vertical plane guidance and control laws that were developed in the previous two chapters are combined here to provide accurate path keeping in three dimensional space. The other requirement is that the forward speed along the path should be constant and equal to a commanded value. This will enable path tracking instead of simply path keeping.

A. PROPULSION CONTROL

Just as the horizontal (vertical) plane path control design was based on the linearized lateral (vertical) equations of motion, the propulsion control law will be based on the linearized form of the surge equation only. The surge equation is:

$$m\dot{u}=X_{\dot{u}}\dot{u}+X_{ww}w^2+uw(X_{w\delta_s}-X_{w\delta_b})\delta+u^2(X_{\delta_s\delta b})\delta^2+C_{Do}(\alpha^2n^2-u^2) \quad \textbf{(4.1)}$$

where:

$$\alpha=\frac{u_{max}}{n_{max}} \quad \textbf{(4.2)}$$

n is the propeller revolutions, and δ the dive plane angle. Only w and δ terms remain in equation (4.1) because only these terms are nonzero at steady state for a constant commanded dive or rise angle. A propulsion control law is introduced of the form:

$$n=n_0+k_n(u-u_c) \tag{4.3}$$

The feedback gain k_n is computed from stability requirements whereas the feedforward term n_0 is computed from steady state accuracy. When $n=n_0$ the forward speed u must equal the commanded speed u_c. Therefore, (4.1) becomes:

$$f(u_c)+C_{D_0}\alpha^2 n_0^2=0 \tag{4.4}$$

where we defined

$$f(u)=X_{ww}w^2+uw(X_{w\delta_s}-X_{w\delta_b})\delta+u^2(X_{\delta_s\delta_s}+X_{\delta_b\delta_b})\delta^2-C_{D_0}u^2 \tag{4.5}$$

The terms w and δ are given as functions of u_c and the commanded pitch angle a_v by

$$w=\frac{(b_1a_{23}-b_2a_{13})Z_{GB}\sin a_v}{(a_{11}b_2-a_{21}b_1)u_c} \tag{4.6}$$

(4.7)

Solving (4.4) for n_0 we get

$$n_0^2 = -\frac{f(u_c)}{C_{D_0}\alpha^2} \tag{4.8}$$

This term n_0 guarantees the required steady state accuracy. To evaluate k_n we substitute (4.3) and (4.8) into (4.1) and we get:

$$(m - X_{\dot{u}})\dot{u} - 2C_{D_0}\alpha^2 n_0 k_n (u - u_c) = 0 \tag{4.9}$$

The characteristic equation of (4.9) is

$$s - \frac{2C_{D_0}\alpha^2 n_0 k_n}{m - X_{\dot{u}}} = 0 \tag{4.10}$$

The desired characteristic equation is

$$s + \omega_0 = 0, \ldots \omega_0 = \frac{10u_c}{t_n l} \tag{4.11}$$

where t_n is the desired dimensionless settling time for the speed control. Comparing (4.10) with (4.11) we can solve for the control gain.

$$k_n = -\frac{5u_c(m - X_{\dot{u}})}{C_{D_0}\alpha^2 n_0 t_n l} \tag{4.12}$$

With the choice of gains (4.12) and (4.8), the propulsion control law (4.3) is complete.

B. THREE LIMENSIONAL PATH KEEPING

Suppose the commanded path is a general straight line in three dimensions, from point O to point F as shown in Figure 18. The vehicle position is at point A. With respect to the inertial coordinate frame (x,y,z) the commanded path is characterized with the two angles a_H and a_V as shown in the Figure. In order to achieve the commanded path, a coordinate frame rotation by an angle a_H is performed first as shown in Figure 19. The necessary geometric relations are:

$$\alpha_H=\tan^{-1}\frac{y_F-y_o}{x_F-x_o} \tag{4.13}$$

$$x'=(y-y_o)\sin a_H+(x-x_o)\cos a_H \tag{4.14}$$

$$y'=(y-y_o)\cos a_H-(x-x_o)\sin a_H \tag{4.15}$$

The rudder control law is then of the form:

$$\delta=k_1(\psi-\alpha_H-\sigma_H)+k_2 v+k_3 r \tag{4.16}$$

where the line of sight angle for horizontal plane control σ_H is defined by:

$$\tan\sigma_H = -\frac{y'}{x_{d_H}} \qquad (4.17)$$

x_{dH} is the lookahead distance determined according to the stability analysis of Chapter II, and k_1, k_2, k_3 are the horizontal plane control gains from Chapter II.

Another rotation by an angle a_v is conducted next as shown in Figure 20. The geometric relations here are:

$$\alpha_v = \tan^{-1}\frac{z_F - z_o}{x'_F} \qquad (4.18)$$

$$x'_F = (y_F - y_o)\sin\alpha_H + (x_F - x_o)\cos\alpha_H \qquad (4.19)$$

$$x'' = -(z - z_o)\sin a_v + x'\cos a_v \qquad (4.20)$$

$$z' = (z - z_o)\cos\alpha_v + x'\sin\alpha_v \qquad (4.21)$$

The dive plane control law is:

$$\delta = k_1(\theta - \alpha_v - \sigma_v) + k_2 w + k_3 q + k_4 \qquad (4.22)$$

where the line of sight angle for vertical plane control σ_v is defined by:

$$\tan\sigma_v = \frac{z'}{x_{d_v}}$$

x_{dv} is the lookahead distance determined according to the stability analysis of Chapter III, and k_1, k_2, k_3, k_4 are the vertical plane control gains as computed in Chapter III. The existence of two distinct distances x_{dH}, x_{dv} is for maximum flexibility in the design and to allow for the possibly different stability conditions for horizontal and vertical plane, as analyzed in the previous two chapters.

Results are presented for a typical three dimensional commanded route that consists of the following way points (x, y, z) = (20, 0, 5), (40, 5, 5), (60, -5, -3), (100, 0, -5) vehicle lengths with individual straight line paths connecting them. Switch from one to the next straight line path was initiated when the vehicle position, measured along the current commanded path, was within a specified target distance (TD) from the way point. Parameters used for the simulation were the following: t_H=7, t_V=5, z_G=0.1, t_N=0.2, x_{dH}=3, x_{dv}=2.5 commanded speeds u=(4, 4, 5, 5) for the four straight line segments respectively, and TD=1. Simulation results are presented in Figure 21 through 25. It can be seen from Figures 21 that accurate path control is maintained in both the horizontal and vertical planes. Speed control is also very accurate, see Figure 22, despite the course changes and nonzero dive and rise angles. The speed controller revolutions per minute are shown in Figure 23, where the maximum saturation limit is set 500 rpm. Rudder response is shown in Figure 25 where the steady state nonzero values occur during a nonzero commanded pitch angle. Comparing Figures 24 and 25 with 22, it can be observed that the vehicle slows down momentarily when the control surfaces become active, a situation which is quickly corrected by the speed controller.

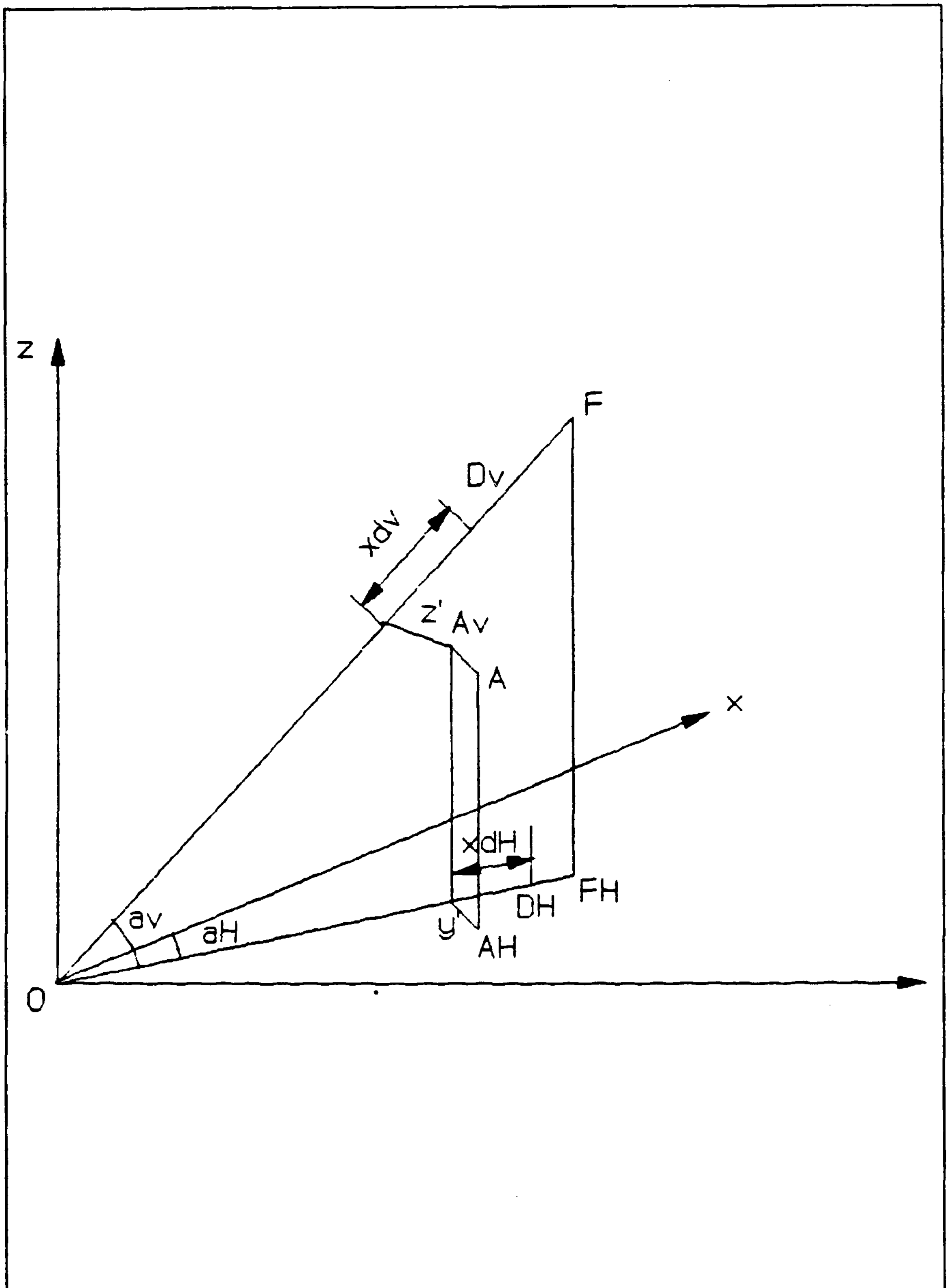

Figure 18. Coordinate transformation for 3-D path keeping

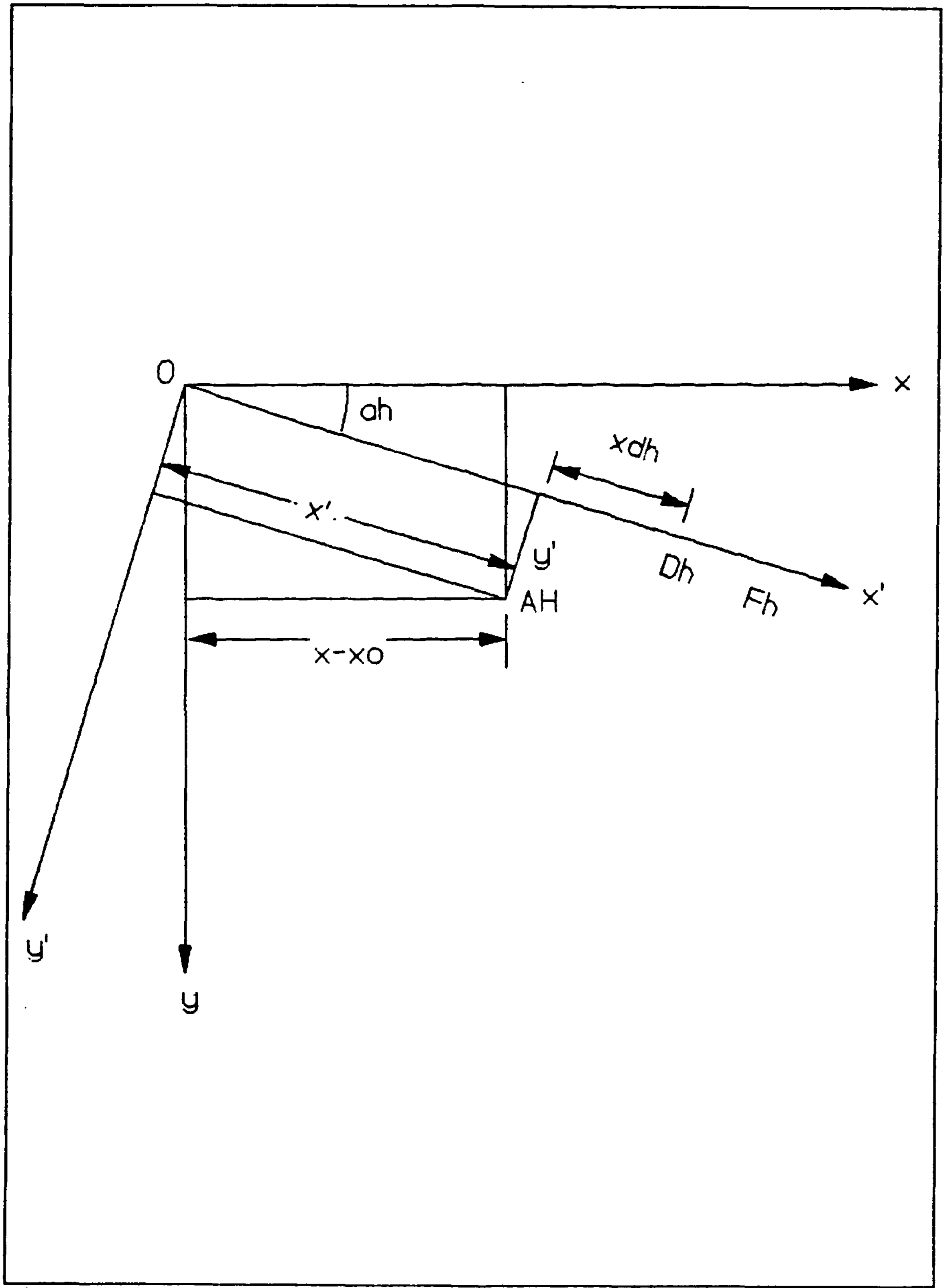

Figure 19. Horizontal plane rotation

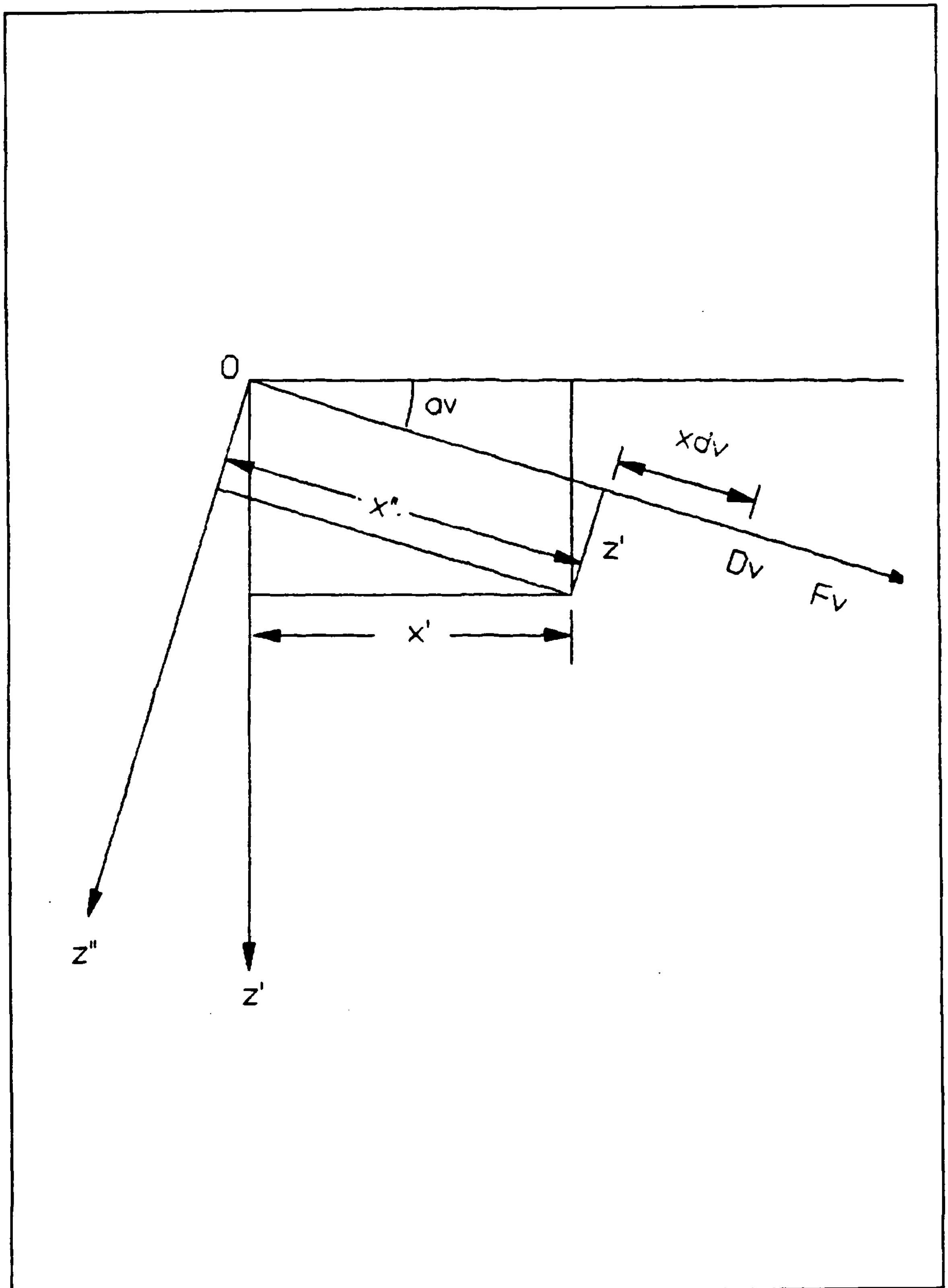

Figure 20. Vertical plane rotation

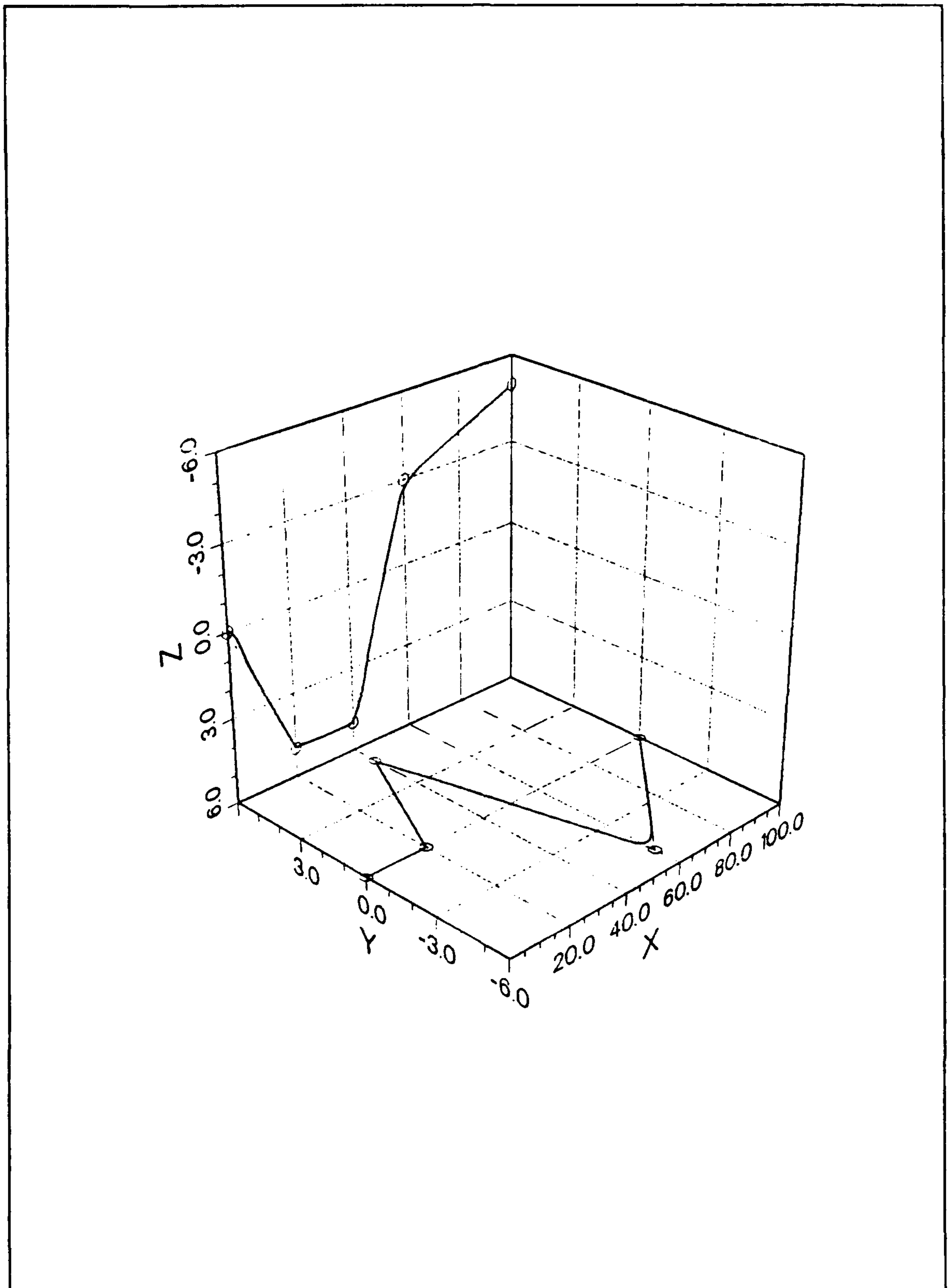

Figure 21. Numerical simulation for 3-D path keeping

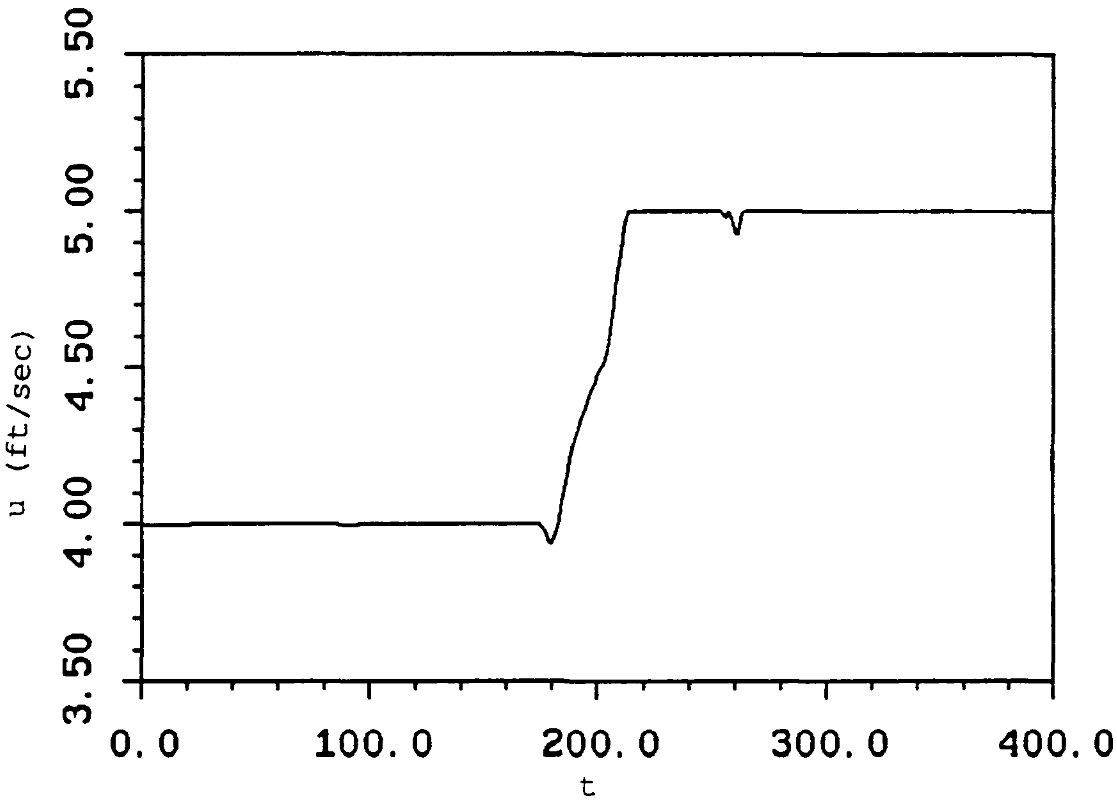

Figure 22. Time history of vehicle speed u

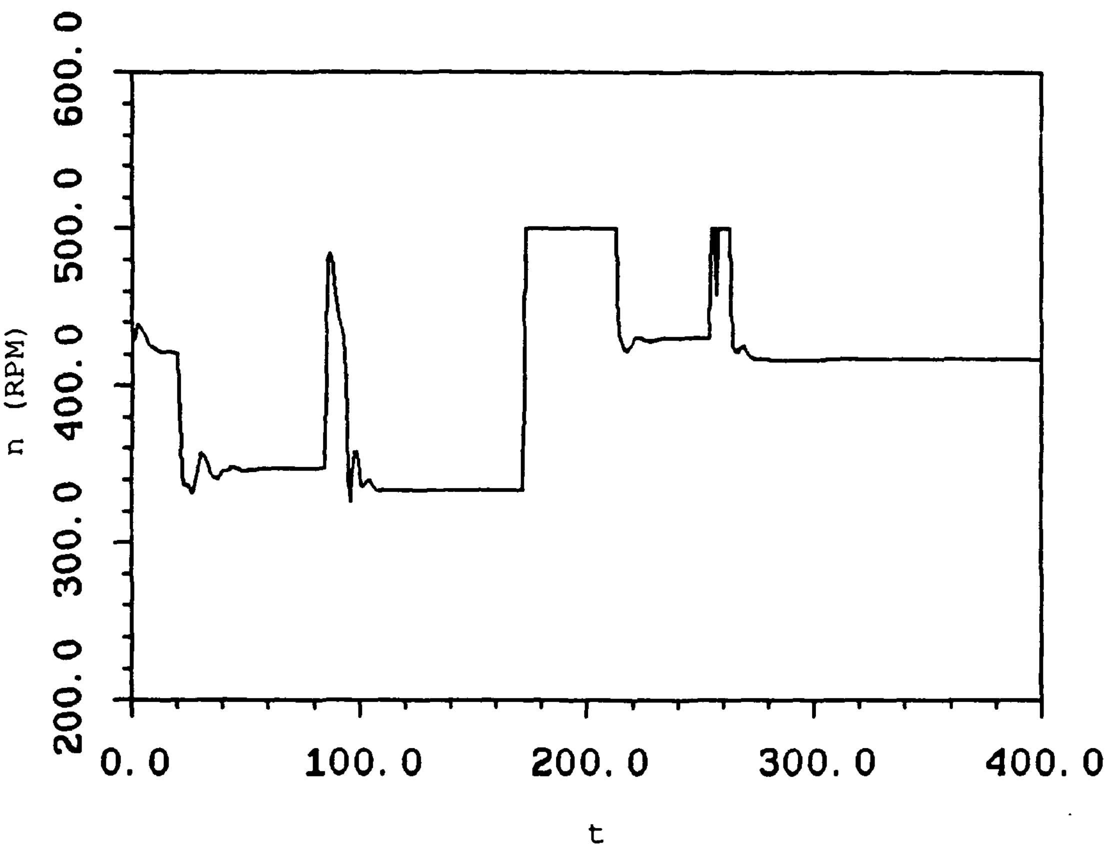

Figure 23. Time history of propeller revolutions per minute

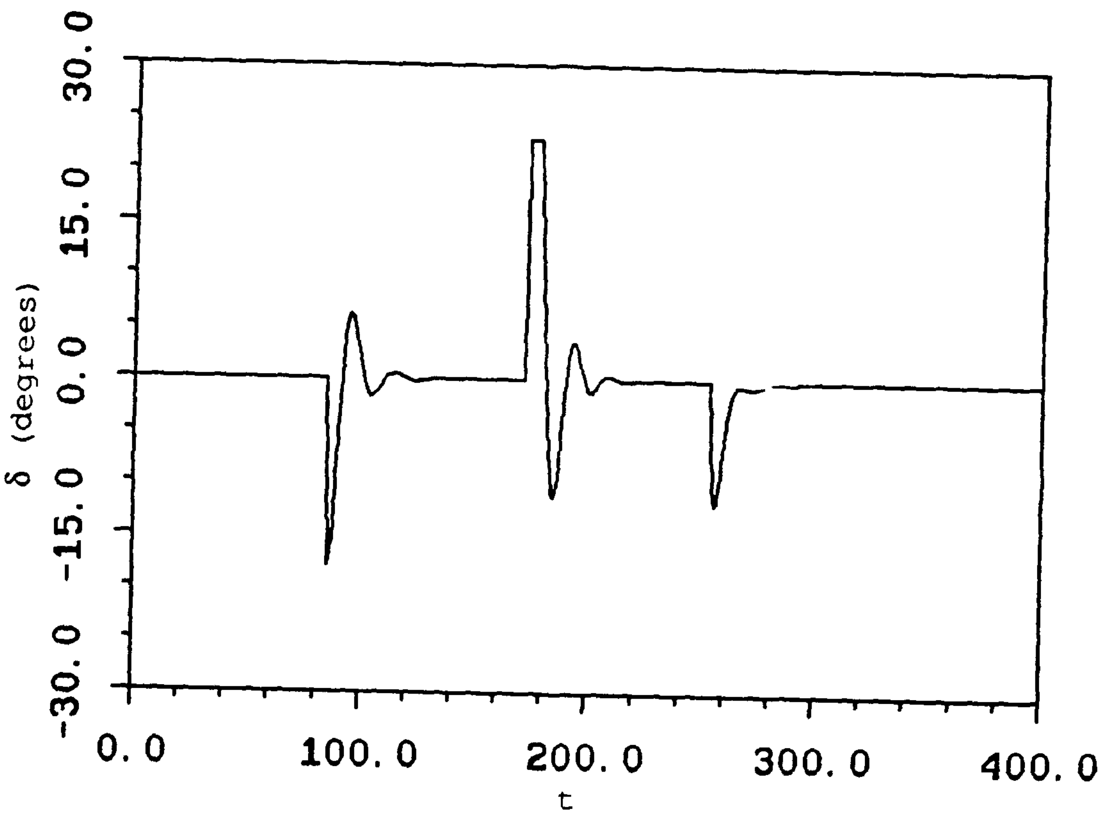

Figure 24. Time history of rudder angle

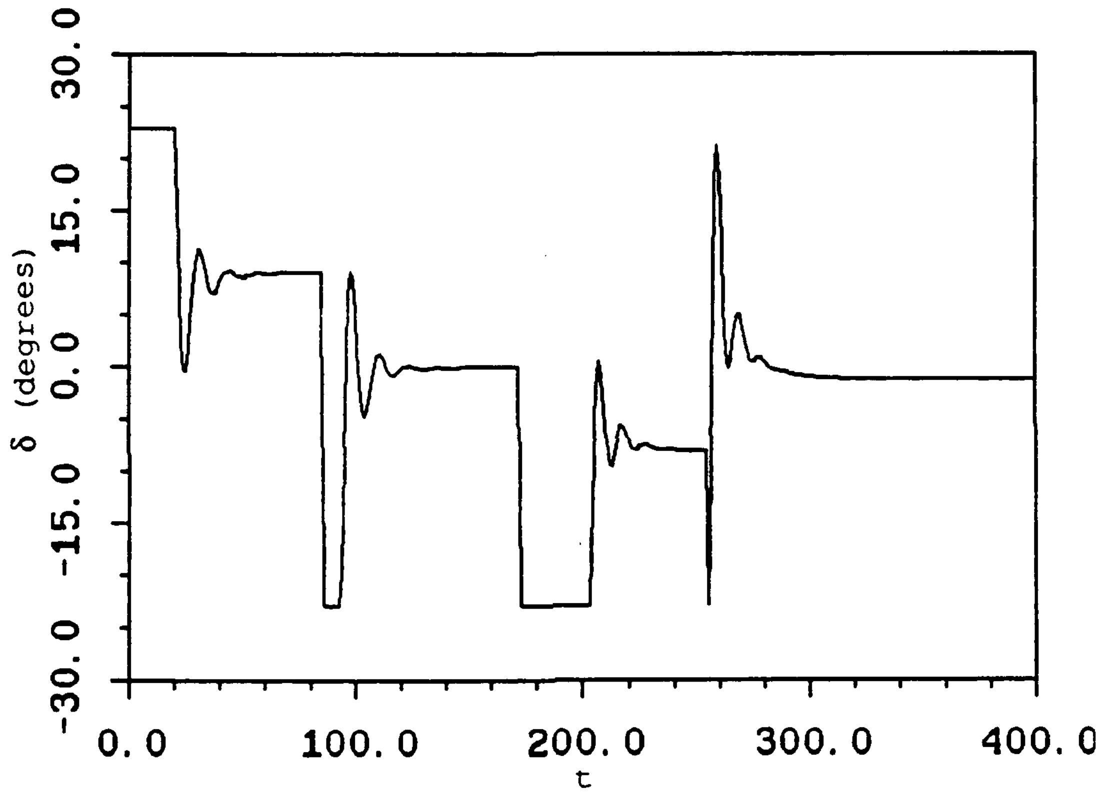

Figure 25. Time history of dive plane angle

CONCLUSIONS AND RECOMMENDATIONS

The main conclusions and contributions of this work can be summarized as follows:

1. Pursuit guidance law and decoupled horizontal and vertical plane orientation controllers were shown to provide accurate vehicle path keeping in three dimensions.

2. The scheme proved to be robust enough so that it could handle the nonlinear coupling between speed response, and horizontal and vertical plane motions without performance degradation.

3. It was shown that the guidance and control schemes must be designed together in order to avoid loss of stability or excessive oscillatory response.

4. Analytic conditions for stability were derived. The conditions were expressed explicitly in terms of the vehicle hydrodynamic characteristics and the guidance and control law design specifications.

5. An extensive study of the mechanism of loss of stability was undertaken for the vertical plane motions. Two distinct possibilities were discovered and analyzed. In the first one pair of complex conjugate eigenvalues crosses the imaginary axis and results in an oscillatory vehicle behavior around the commanded path. In the second , one real eigenvalue crosses zero and the vehicle drifts off to a steady state path

with its deviation from the commanded path linearly increasing with time. This new path was computed and explicit conditions to avoid such a undesirable situation were given.

Some recommendations for further research include the following:

1. Comparisons from the point of view of path keeping response under physical system / mathematical model mismatch.

2. State estimators must be included in the analysis to evaluate performance under partial state knowledge and sensor noise.

APPENDIX A

```
C       PROGRAM SIM_3D.FOR
C
C       FOTIS A. PAPOULIAS/ANGELLOS G. PAPASOTIRIOU
C       NAVAL POSTGRADUATE SCHOOL
C       AUGUST 1991
C
C       VEHICLE THREE DIMENSIONAL PATH KEEPING
C       HEADING AUTOPILOTS
C       PURE PURSUIT NAVIGATION
C       SIMULTANEOUS RUDDER/DIVE PLANE SWITCHINGS
C
C       DECLARATIONS
C
        REAL L,MASS,IX,IY,IZ,IXZ,IYZ,IXY
        REAL K1H,K2H,K3H,K1V,K2V,K3V,K4V,KN
        REAL KPDOT,KRDOT,KPQ,KQR,KVDOT,KP,KR,KVQ,KWP,KWR,KV,KVW,
     &      KPN,KDB
        REAL MQDOT,MPP,MPR,MRR,MWDOT,MQ,MVP,MVR,MW,MVV,
     &      MDS,MDB,NDRB
        REAL NPDOT,NRDOT,NPQ,NQR,NVDOT,NP,NR,NVQ,NWP,NWR,
     &      NV,NVW,NDRS
        REAL MM(6,6),INDX(100)
        DIMENSION X(9),BR(9),HH(9),VECH1(9),VECH2(9),XMMINV(6,6)
        DIMENSION VECV1(9),VECV2(9),F(12),FP(6),DISV(100)
        DIMENSION XDES(100),YDES(100),ZDES(100),UDES(100),
     &           DISH(100)
C
C       GEOMETRIC PROPERTIES
C
        WEIGHT=12000.0
        IX     = 1760.0
        IY     = 9450.0
        IZ     =10700.0
        IXY    =    0.0
        IYZ    =    0.0
        IXZ    =    0.0
        L      =   17.425
        RHO    =    1.94
        G      =   32.2
        XG     =    0.0
        YG     =    0.0
        XB     =    0.0
        YB     =    0.0
        ZB     =    0.0
```

```
      A0     =     2.0
      CD0    =     0.0057
      MASS   =WEIGHT/G
      BOY    =WEIGHT
      RPMMAX=  500.0
      RPMMIN= -500.0
      UMAX   =     6.0
      UMIN   =     0.1
      ALPHA  =UMAX/RPMMAX
C
C     SURGE HYDRODYNAMIC COEFFICIENTS
C
      XPP    = 7.030E-03*0.5*RHO*L**4
      XQQ    =-1.470E-02*0.5*RHO*L**4
      XRR    = 4.010E-03*0.5*RHO*L**4
      XPR    = 7.640E-04*0.5*RHO*L**4
      XUDOT  =-7.580E-03*0.5*RHO*L**3
      XWQ    =-1.920E-01*0.5*RHO*L**3
      XVP    =-3.240E-03*0.5*RHO*L**3
      XVR    = 1.890E-02*0.5*RHO*L**3
      XQDS   = 2.610E-02*0.5*RHO*L**3
      XQDB   =-2.600E-03*0.5*RHO*L**3
      XRDR   =-8.180E-04*0.5*RHO*L**3
      XVV    = 5.290E-02*0.5*RHO*L**2
      XWW    = 1.710E-01*0.5*RHO*L**2
      XVDR   = 1.730E-03*0.5*RHO*L**2
      XWDS   = 4.600E-02*0.5*RHO*L**2
      XWDB   = 9.660E-03*0.5*RHO*L**2
      XDSDS  =-1.160E-02*0.5*RHO*L**2
      XDBDB  =-8.070E-03*0.5*RHO*L**2
      XDRDR  =-1.010E-02*0.5*RHO*L**2
      XRES   =       CD0*0.5*RHO*L**2
      XPROP  = XRES*ALPHA**2
C
C     SWAY HYDRODYNAMIC COEFFICIENTS
C
      YPDOT  = 1.270E-04*0.5*RHO*L**4
      YRDOT  = 1.240E-03*0.5*RHO*L**4
      YPQ    = 4.125E-03*0.5*RHO*L**4
      YQR    =-6.510E-03*0.5*RHO*L**4
      YVDOT  =-5.550E-02*0.5*RHO*L**3
      YP     = 3.055E-03*0.5*RHO*L**3
      YR     = 2.970E-02*0.5*RHO*L**3
      YVQ    = 2.360E-02*0.5*RHO*L**3
      YWP    = 2.350E-01*0.5*RHO*L**3
      YWR    =-1.880E-02*0.5*RHO*L**3
      YV     =-9.310E-02*0.5*RHO*L**2
      YVW    = 6.840E-02*0.5*RHO*L**2
      YDRS   =+2.270E-02*0.5*RHO*L**2
      YDRB   =+2.270E-02*0.5*RHO*L**2
C
```

```
C     HEAVE HYDRODYNAMIC COEFFICIENTS
C
      ZQDOT =-6.810E-03*0.5*RHO*L**4
      ZPP   = 1.270E-04*0.5*RHO*L**4
      ZPR   = 6.670E-03*0.5*RHO*L**4
      ZRR   =-7.350E-03*0.5*RHO*L**4
      ZWDOT =-2.430E-01*0.5*RHO*L**3
      ZQ    =-1.350E-01*0.5*RHO*L**3
      ZVP   =-4.810E-02*0.5*RHO*L**3
      ZVR   = 4.550E-02*0.5*RHO*L**3
      ZW    =-3.020E-01*0.5*RHO*L**2
      ZVV   =-6.840E-02*0.5*RHO*L**2
      ZDS   =-2.270E-02*0.5*RHO*L**2
      ZDB   =-2.270E-02*0.5*RHO*L**2
C
C     ROLL HYDRODYNAMIC COEFFICIENTS
C
      KPDOT =-1.010E-03*0.5*RHO*L**5
      KRDOT =-3.370E-05*0.5*RHO*L**5
      KPQ   =-6.930E-05*0.5*RHO*L**5
      KQR   = 1.680E-02*0.5*RHO*L**5
      KVDOT = 1.270E-04*0.5*RHO*L**4
      KP    =-1.100E-02*0.5*RHO*L**4
      KR    =-8.410E-04*0.5*RHO*L**4
      KVQ   =-5.115E-03*0.5*RHO*L**4
      KWP   =-1.270E-04*0.5*RHO*L**4
      KWR   = 1.390E-02*0.5*RHO*L**4
      KV    = 3.055E-03*0.5*RHO*L**3
      KVW   =-1.870E-01*0.5*RHO*L**3
C
C     PITCH HYDRODYNAMIC COEFFICIENTS
C
      MQDOT =-1.680E-02*0.5*RHO*L**5
      MPP   = 5.260E-05*0.5*RHO*L**5
      MPR   = 5.040E-03*0.5*RHO*L**5
      MRR   =-2.860E-03*0.5*RHO*L**5
      MWDOT =-6.810E-02*0.5*RHO*L**4
      MQ    =-6.860E-02*0.5*RHO*L**4
      MVP   = 1.180E-03*0.5*RHO*L**4
      MVR   = 1.730E-02*0.5*RHO*L**4
      MW    = 9.860E-02*0.5*RHO*L**3
      MVV   =-2.510E-02*0.5*RHO*L**3
      MDS   =-1.113E-02*0.5*RHO*L**3
      MDB   = 1.113E-02*0.5*RHO*L**3
C
C     YAW HYDRODYNAMIC COEFFICIENTS
C
      NPDOT =-3.370E-05*0.5*RHO*L**5
      NRDOT =-3.400E-03*0.5*RHO*L**5
      NPQ   =-2.110E-02*0.5*RHO*L**5
      NQR   = 2.750E-03*0.5*RHO*L**5
```

```
      NVDOT = 1.240E-03*0.5*RHO*L**4
      NP    =-8.405E-04*0.5*RHO*L**4
      NR    =-1.640E-02*0.5*RHO*L**4
      NVQ   =-9.990E-03*0.5*RHO*L**4
      NWP   =-1.750E-02*0.5*RHO*L**4
      NWR   = 7.350E-03*0.5*RHO*L**4
      NV    =-7.420E-03*0.5*RHO*L**3
      NVW   =-2.670E-02*0.5*RHO*L**3
      NDRS  =-1.113E-02*0.5*RHO*L**3
      NDRB  =+1.113E-02*0.5*RHO*L**3
C
C     OPEN DATA AND RESULTS FILES
C
      OPEN (10,FILE='PATH_3D.DAT',STATUS='OLD')
      OPEN (11,FILE='XY.RES',STATUS='NEW')
      OPEN (12,FILE='XZ.RES',STATUS='NEW')
      OPEN (13,FILE='DRS.RES',STATUS='NEW')
      OPEN (14,FILE='DS.RES',STATUS='NEW')
      OPEN (15,FILE='YCTE.RES',STATUS='NEW')
      OPEN (16,FILE='ZCTE.RES',STATUS='NEW')
      OPEN (17,FILE='XYZ.RES',STATUS='NEW')
      OPEN (18,FILE='U.RES',STATUS='NEW')
      OPEN (19,FILE='RPM.RES',STATUS='NEW')
      OPEN (20,FILE='PHI.RES',STATUS='NEW')
      OPEN (21,FILE='THETA.RES',STATUS='NEW')
      OPEN (22,FILE='PSI.RES',STATUS='NEW')
      OPEN (23,FILE='V.RES',STATUS='NEW')
      OPEN (24,FILE='R.RES',STATUS='NEW')
      OPEN (25,FILE='W.RES',STATUS='NEW')
      OPEN (26,FILE='Q.RES',STATUS='NEW')
      OPEN (27,FILE='YZ.RES',STATUS='NEW')
C
C     READ DATA FILE
C
      READ (10,*) TSIM,DELTA,IPRNT
      READ (10,*) IPTS,TARGET
      READ (10,*) TN,TH,TV,ZG
      IF (IPTS.GT.100) IPTS=100
      DO 1 I=1,IPTS
        READ (10,*) XD,YD,ZD,XDH,XDV,U0
        XDES(I)=XD*L
        YDES(I)=YD*L
        ZDES(I)=ZD*L
        UDES(I)=U0
        DISH(I)=XDH*L
        DISV(I)=XDV*L
    1 CONTINUE
C
C     MASS MATRIX INITIALIZATION AND DEFINITION
C
      DO 15 J=1,6
```

```
          DO 10 K=1,6
            XMMINV(J,K)=0.0
            MM(J,K)=0.0
   10     CONTINUE
   15   CONTINUE
C
        MM(1,1)= MASS-XUDOT
        MM(1,5)= MASS*ZG
        MM(1,6)=-MASS*YG
C
        MM(2,2)= MASS-YVDOT
        MM(2,4)=-MASS*ZG-YPDOT
        MM(2,6)= MASS*XG-YRDOT
C
        MM(3,3)= MASS-ZWDOT
        MM(3,4)= MASS*YG
        MM(3,5)=-MASS*XG-ZQDOT
C
        MM(4,2)=-MASS*ZG-KVDOT
        MM(4,3)= MASS*YG
        MM(4,4)= IX-KPDOT
        MM(4,5)=-IXY
        MM(4,6)=-IXZ-KRDOT
C
        MM(5,1)= MASS*ZG
        MM(5,3)=-MASS*XG-MWDOT
        MM(5,4)=-IXY
        MM(5,5)= IY-MQDOT
        MM(5,6)=-IYZ
C
        MM(6,1)=-MASS*YG
        MM(6,2)= MASS*XG-NVDOT
        MM(6,4)=-IXZ-NPDOT
        MM(6,5)=-IYZ
        MM(6,6)= IZ-NRDOT
C
C       MASS MATRIX INVERSION
C
        DO 12 I=1,6
          DO 11 J=1,6
            XMMINV(I,J)=0.0
   11     CONTINUE
          XMMINV(I,I)=1.0
   12   CONTINUE
        CALL INVTA(MM,6,INDX,D)
        DO 13 J=1,6
          CALL INVTB(MM,6,INDX,XMMINV(1,J))
   13   CONTINUE
C
C       VARIABLES INITIALIZATION
C
```

```
      PISIM =TSIM/DELTA
      ISIM  =PISIM
      ECHO  =1.0/DELTA
      IECHO =ECHO
      YAW   =0.0
      SWAY  =0.0
      PITCH =0.0
      HEAVE =0.0
      U     =UDES(1)
      RPM   =UDES(1)/ALPHA
      V     =0.0
      W     =0.0
      P     =0.0
      Q     =0.0
      R     =0.0
      DS    =0.0
      DB    =0.0
      DR    =0.0
      TWOPI =8.0*ATAN(1.0)
      PI    =0.5*TWOPI
      PHI   =0.0
      ISTART=1
      TARGET=TARGET*L
      XPOS  =0.0
      YPOS  =0.0
      ZPOS  =0.0
      CDY   =0.5
      CDZ   =0.5
      JPRNT =0
      IJK   =0
      JE    =0
      DRS   =0.0
      DRB   =0.0
      DS    =0.0
      DB    =0.0
C
C     DEFINE THE LENGTH X, BREADTH BR, AND HEIGHT HH TERMS
C
      X(1)  =  -105.9/12.0
      X(2)  =   -99.3/12.0
      X(3)  =   -87.3/12.0
      X(4)  =   -66.3/12.0
      X(5)  =    72.7/12.0
      X(6)  =    83.2/12.0
      X(7)  =    91.2/12.0
      X(8)  =    99.2/12.0
      X(9)  =   103.2/12.0
C
      HH(1) =    0.00/12.0
      HH(2) =    8.24/12.0
      HH(3) =   19.76/12.0
```

```
      HH(4) =   29.36/12.0
      HH(5) =   31.85/12.0
      HH(6) =   27.84/12.0
      HH(7) =   21.44/12.0
      HH(8) =   12.00/12.0
      HH(9) =    0.00/12.0
C
      BR(1) =    0.00/12.0
      BR(2) =    8.24/12.0
      BR(3) =   19.76/12.0
      BR(4) =   29.36/12.0
      BR(5) =   31.85/12.0
      BR(6) =   27.84/12.0
      BR(7) =   21.44/12.0
      BR(8) =   12.00/12.0
      BR(9) =    0.00/12.0
C
C     AUXILLIARY VARIABLES FOR HORIZONTAL PLANE CONTROL
C
      DH  =(IZ-NRDOT)*(MASS-YVDOT)-
     &     (MASS*XG-YRDOT)*(MASS*XG-NVDOT)
      A11H=((IZ-NRDOT)*YV-(MASS*XG-YRDOT)*NV)/DH
      A12H=((IZ-NRDOT)*(-MASS+YR)-
     &     (MASS*XG-YRDOT)*(-MASS*XG+NR))/DH
      A21H=((MASS-YVDOT)*NV-(MASS*XG-NVDOT)*YV)/DH
      A22H=((MASS-YVDOT)*(-MASS*XG+NR)-
     &     (MASS*XG-NVDOT)*(-MASS+YR))/DH
      B11H=((IZ-NRDOT)*YDRS-(MASS*XG-YRDOT)*NDRS)/DH
      B12H=((IZ-NRDOT)*YDRB-(MASS*XG-YRDOT)*NDRB)/DH
      B21H=((MASS-YVDOT)*NDRS-(MASS*XG-NVDOT)*YDRS)/DH
      B22H=((MASS-YVDOT)*NDRB-(MASS*XG-NVDOT)*YDRB)/DH
      B1H =B11H-B12H
      B2H =B21H-B22H
C
C     AUXILLIARY VARIABLES FOR VERTICAL PLANE CONTROL
C
      DV  =(MASS-ZWDOT)*(IY-MQDOT)-ZQDOT*MWDOT
      A11V=((IY-MQDOT)*ZW+ZQDOT*MW)/DV
      A12V=((IY-MQDOT)*(ZQ+MASS)+ZQDOT*MQ)/DV
      A13V=-(ZG-ZB)*(MASS*XG+ZQDOT)*WEIGHT/DV
      A21V=(MWDOT*ZW+(MASS-ZWDOT)*MW)/DV
      A22V=(MWDOT*(ZQ+MASS)+(MASS-ZWDOT)*MQ)/DV
      A23V=-(ZG-ZB)*(MASS-ZWDOT)*WEIGHT/DV
      B11V=((IY-MQDOT)*ZDS+ZQDOT*MDS)/DV
      B12V=((IY-MQDOT)*ZDB+ZQDOT*MDB)/DV
      B21V=(MWDOT*ZDS+(MASS-ZWDOT)*MDS)/DV
      B22V=(MWDOT*ZDB+(MASS-ZWDOT)*MDB)/DV
      B1V =B11V-B12V
      B2V =B21V-B22V
C
C
```

```
C     SIMULATION BEGINS
C
C     LOOP OVER WAY POINTS
C
      DO 200 IP=1,IPTS
        IF (IP.GE.2) GO TO 210
        XDH=DISH(1)
        XDV=DISV(1)
        U0 =UDES(1)
        XD =XDES(1)
        YD =YDES(1)
        ZD =ZDES(1)
        XD1=0.0
        YD1=0.0
        ZD1=0.0
        XD2=XD
        YD2=YD
        ZD2=ZD
        GO TO 211
 210    XDH=DISH(IP)
        XDV=DISV(IP)
        U0 =UDES(IP)
        XD =XDES(IP)
        YD =YDES(IP)
        ZD =ZDES(IP)
        XD1=XD2
        YD1=YD2
        ZD1=ZD2
        XD2=XD
        YD2=YD
        ZD2=ZD
 211    ZD12=ZD2-ZD1
        XD12=XD2-XD1
        YD12=YD2-YD1
C
C       HORIZONTAL HEADING CONTROL GAINS
C
        OMEGAH=(10.0*U0)/(TH*L)
        AD1H=1.75*OMEGAH
        AD2H=2.15*OMEGAH**2
        AD3H=OMEGAH**3
        A1=B1H*U0*U0
        B1=B2H*U0*U0
        C1=-AD1H-(A11H+A22H)*U0
        A2=(B1H*A22H-B2H*A12H)*U0**3
        B2=(B2H*A11H-B1H*A21H)*U0**3
        K1H=AD3H/((B2H*A11H-B1H*A21H)*U0**3)
        C2=AD2H-(A11H*A22H-A12H*A21H)*U0**2+B2H*U0*U0*K1H
        K2H=(C1*B2-C2*B1)/(A1*B2-A2*B1)
        K3H=(C2*A1-C1*A2)/(A1*B2-A2*B1)
C
```

```
C        VERTICAL HEADING CONTROL GAINS
C
         OMEGAV=(10.0*U0)/(TV*L)
         AD1V=1.75*OMEGAV
         AD2V=2.15*OMEGAV**2
         AD3V=OMEGAV**3
         A2=B1V*U0*U0
         A3=B2V*U0*U0
         D1=-AD1V-(A11V+A22V)*U0
         B1=-B2V*U0*U0
         B2=(B1V*A22V-B2V*A12V)*U0**3
         B3=(B2V*A11V-B1V*A21V)*U0**3
         D2=AD2V+A23V+(A12V*A21V-A11V*A22V)*U0**2
         C1=(B2V*A11V-B1V*A21V)*U0**3
         C2=(A23V*B1V-A13V*B2V)*U0**2
         D3=AD3V+(A13V*A21V-A11V*A23V)*U0
         K2V=(A3*B1*D3+C1*B3*D1-D2*C1*A3)
         K2V=K2V/(A3*B1*C2+C1*B3*A2-C1*A3*B2)
         K1V=(D3-C2*K2V)/C1
         K3V=(D1-A2*K2V)/A3
C
         ALPHAH=ATAN(YD12/XD12)
         ALPHAH=ABS(ALPHAH)
         IF ((XD12.GE.0.0).AND.(YD12.GE.0.0)) ALPHAH=   ALPHAH
         IF ((XD12.GE.0.0).AND.(YD12.LT.0.0)) ALPHAH=  -ALPHAH
         IF ((XD12.LT.0.0).AND.(YD12.GE.0.0)) ALPHAH=PI-ALPHAH
         IF ((XD12.LT.0.0).AND.(YD12.LT.0.0)) ALPHAH=PI+ALPHAH
         XCTEH=(YPOS-YD1)*SIN(ALPHAH)+(XPOS-XD1)*COS(ALPHAH)
         YCTE =(YPOS-YD1)*COS(ALPHAH)-(XPOS-XD1)*SIN(ALPHAH)
         X1P  =YD12*SIN(ALPHAH)+XD12*COS(ALPHAH)
         ALPHAV=ATAN(ZD12/X1P)
         ALPHAV=ABS(ALPHAV)
         IF (ZD12.GE.0.0) ALPHAV=-ALPHAV
         K4V=-(A13V*(A21V+B2V*U0*K2V)-A23V*(A11V+B1V*U0*K2V))
         K4V=K4V*SIN(ALPHAV)/((B1V*A21V-B2V*A11V)*U0*U0)
         ZCTE = (ZPOS-ZD1)*COS(ALPHAV)+XCTEH*SIN(ALPHAV)
         XCTEV=-(ZPOS-ZD1)*SIN(ALPHAV)+XCTEH*COS(ALPHAV)
C
C        PROPULSION CONTROL GAIN
C
         WSS=(B1V*A23V-B2V*A13V)*SIN(ALPHAV)
         WSS=WSS/((A11V*B2V-A21V*B1V)*U0)
         DSS=(A21V*A13V-A11V*A23V)*SIN(ALPHAV)
         DSS=DSS/((A11V*B2V-A21V*B1V)*U0*U0)
         FUC=XWW*WSS**2+U0*WSS*(XWDS-XWDB)*DSS
     &      +U0*U0*(XDSDS+XDBDB)*DSS**2-XRES*U0**2
         RPM0=-FUC/(XRES*ALPHA**2)
         RPM0=SQRT(RPM0)
         WRITE (*,*) RPM0,U0/ALPHA
         KN=-5.0*U0*(MASS-XUDOT)/(XRES*ALPHA*ALPHA*RPM0*TN*L)
C
```

```
      WRITE (*,201) XD/L,YD/L,ZD/L
C
C     SIMULATION FOR EACH WAY POINT
C
      DO 100 I=ISTART,ISIM
        ICOUNT=I
C
        IF (U.LT.UMIN) U=UMIN
C
C       CALCULATE THE DRAG FORCE, INTEGRATE THE DRAG OVER
C                        THE VEHICLE
C
        DO 600 K=1,9
          UCF=(V+X(K)*R)**2+(W-X(K)*Q)**2
          UCF=SQRT(UCF)
          IF (UCF.LT.1.E-6) GO TO 601
          CFLOW=CDY*HH(K)*(V+X(K)*R)**2+CDZ*BR(K)*
     &          (W-X(K)*Q)**2
          VECH1(K)=CFLOW*(V+X(K)*R)/UCF
          VECH2(K)=CFLOW*(V+X(K)*R)*X(K)/UCF
          VECV1(K)=CFLOW*(W-X(K)*Q)/UCF
          VECV2(K)=CFLOW*(W-X(K)*Q)*X(K)/UCF
  600   CONTINUE
        CALL TRAP(9,VECV1,X,HEAVE)
        CALL TRAP(9,VECV2,X,PITCH)
        CALL TRAP(9,VECH1,X,SWAY )
        CALL TRAP(9,VECH2,X,YAW  )
        HEAVE=-0.5*RHO*HEAVE
        PITCH=+0.5*RHO*PITCH
        SWAY =-0.5*RHO*SWAY
        YAW  =-0.5*RHO*YAW
        GO TO 602
  601   HEAVE=0.0
        PITCH=0.0
        SWAY =0.0
        YAW  =0.0
  602   CONTINUE
C
C       FORCE EQUATIONS
C
C
C       SURGE FORCE
C
      FP(1) = MASS*V*R-MASS*W*Q+MASS*XG*Q**2+MASS*XG*R**2-
     &        MASS*YG*P*Q-MASS*ZG*P*R+XPP*P**2+XQQ*
     &        Q**2+XRR*R**2+XPR*P*R+XWQ*W*Q+XVP*V*P+
     &        XVR*V*R+U*Q*(XQDS*DS+XQDB*DB)+
     &        U*R*(XRDRS*DRS+XRDRB*DRB)+XVV*V**2+XWW*
     &        W**2+U*V*(XVDRS*DRS+XDRB*DRB)+U*W*
     &       (XWDS*DS+XWDB*DB)+(XDSDS*DS**2+XDBDB*DB**2+
     &        XDRDR*(DRS**2+DRB**2))*U**2-(WEIGHT-BOY)*
```

```
     &           SIN(THETA)+XPROP*RPM*RPM-XRES*U*U
C
C        SWAY FORCE
C
      FP(2) =-MASS*U*R-MASS*XG*P*Q+MASS*YG*R**2-MASS*ZG*Q*R+
     &          YPQ*P*Q+YQR*Q*R+YP*U*P+YR*U*R+YVQ*V*Q+
     &          YWP*W*P+YWR*W*R+YV*U*V+YVW*V*W+YDRS*U**2*DRS+
     &          YDRB*U**2*DRB+(WEIGHT-BOY)*
     &         COS(THETA)*SIN(PHI)+MASS*W*P+MASS*YG*P**2+SWAY
C
C        HEAVE FORCE
C
        FP(3) = MASS*U*Q-MASS*V*P-MASS*XG*P*R-MASS*YG*Q*R+
     &            MASS*ZG*P**2+MASS*ZG*Q**2+ZPP*P**2+
     &            ZPR*P*R+ZRR*R**2+ZQ*
     &            U*Q+ZVP*V*P+ZVR*V*R+ZW*U*W+ZVV*V**2+HEAVE+
     &            U**2*(ZDS*DS+ZDB*DB)+(WEIGHT-BOY)*
     &            COS(THETA)*COS(PHI)
C
C        ROLL MOMENT
C
        FP(4) = -IZ*Q*R+IY*Q*R-IXY*P*R+IYZ*Q**2-
     &            IYZ*R**2+IXZ*P*Q+MASS*YG*U*Q-MASS*
     &            YG*V*P-MASS*ZG*W*P+KPQ*P*Q+KQR*Q*R+
     &            KP*U*P+KR*U*R+KVQ*V*Q+KWP*W*P+
     &            KWR*W*R+KV*U*V+KVW*V*W+(YG*WEIGHT-YB*BOY)*
     &            COS(THETA)*COS(PHI)-(ZG*WEIGHT-
     &            ZB*BOY)*COS(THETA)*SIN(PHI)+MASS*ZG*U*R
C
C        PITCH MOMENT
C
        FP(5) = -IX*P*R+IZ*P*R+IXY*Q*R-IYZ*P*Q-
     &            IXZ*P**2+IXZ*R**2-MASS*XG*U*Q+
     &            MASS*XG*V*P+MASS*ZG*V*R-
     &            MASS*ZG*W*Q+MPP*P**2+
     &            MPR*P*R+MRR*R**2+MQ*
     &            U*Q+MVP*V*P+MVR*V*R+MW*U*W+
     &            MVV*V**2+U**2*(MDS*DS+MDB*DB)-(XG*WEIGHT-
     &            XB*BOY)*COS(THETA)*COS(PHI)-
     &            (ZG*WEIGHT-ZB*BOY)*SIN(THETA)+PITCH
C
C        YAW MOMENT
C
        FP(6) = -IY*P*Q+IX*P*Q+IXY*P**2-IXY*Q**2+IYZ*P*R-
     &            IXZ*Q*R-MASS*XG*U*R+MASS*XG*W*P-MASS*YG*
     &            V*R+MASS*YG*W*Q+NPQ*P*Q+NQR*Q*R+NP*U*P+NR*
     &            U*R+NVQ*V*Q+NWP*W*P+NWR*W*R+NV*U*V+
     &            NVW*V*W+NDRS*U**2*DRS+NDRB*U**2*DRB+
     &            (XG*WEIGHT-XB*BOY)*COS(THETA)*SIN(PHI)+
     &            (YG*WEIGHT-YB*BOY)*SIN(THETA)+YAW
C
```

```
C
C         COMPUTE THE RIGHT HAND SIDE OF XDOT=F(X)
C
          DO 610 J = 1,6
            F(J) = 0.0
            DO 611 K = 1,6
              F(J) = XMMINV(J,K)*FP(K) + F(J)
  611       CONTINUE
  610     CONTINUE
C
C         INERTIAL POSITION RATES
C
          F(7) = U*COS(PSI)*COS(THETA)+V*(COS(PSI)*SIN(THETA)*
     &           SIN(PHI)-SIN(PSI)*COS(PHI))+W*(COS(PSI)*
     &           SIN(THETA)*COS(PHI)+SIN(PSI)*SIN(PHI))
C
          F(8) = U*SIN(PSI)*COS(THETA)+V*(SIN(PSI)*SIN(THETA)*
     &           SIN(PHI)+COS(PSI)*COS(PHI))+W*(SIN(PSI)*
     &           SIN(THETA)*COS(PHI)-COS(PSI)*SIN(PHI))
C
          F(9) = -U*SIN(THETA)+V*COS(THETA)*SIN(PHI)+
     &           W*COS(THETA)*COS(PHI)
C
C         EULER ANGLE RATES
C
          F(10)= P+Q*SIN(PHI)*TAN(THETA)+R*COS(PHI)*TAN(THETA)
C
          F(11)= Q*COS(PHI)-R*SIN(PHI)
C
          F(12)= Q*SIN(PHI)/COS(THETA)+R*COS(PHI)/COS(THETA)
C
C         ASSIGN XDOT VECTOR
C
          UDOT   = F(1)
          VDOT   = F(2)
          WDOT   = F(3)
          PDOT   = F(4)
          QDOT   = F(5)
          RDOT   = F(6)
          XDOT   = F(7)
          YDOT   = F(8)
          ZDOT   = F(9)
          PHIDOT = F(10)
          THEDOT = F(11)
          PSIDOT = F(12)
C
C         FIRST ORDER INTEGRATION
C
          U      = U      + DELTA*UDOT
          V      = V      + DELTA*VDOT
          W      = W      + DELTA*WDOT
```

```
      P     = P     + DELTA*PDOT
      Q     = Q     + DELTA*QDOT
      R     = R     + DELTA*RDOT
      XPOS  = XPOS  + DELTA*XDOT
      YPOS  = YPOS  + DELTA*YDOT
      ZPOS  = ZPOS  + DELTA*ZDOT
      PHI   = PHI   + DELTA*PHIDOT
      THETA = THETA + DELTA*THEDOT
      PSI   = PSI   + DELTA*PSIDOT
C
C     VELOCITY INPUT CALCULATION
C
      UC=U0
      IF (UC.GE.UMAX) UC=UMAX
      IF (UC.LE.UMIN) UC=UMIN
C
C     RPM INPUT CALCULATION
C
      RPM0=UC/ALPHA
      RPM=RPM0+KN*(U-UC)
      IF (RPM.GE.RPMMAX) RPM=RPMMAX
      IF (RPM.LE.RPMMIN) RPM=RPMMIN
C
C     COORDINATE TRANSFORMATIONS
C
     XCTEH= (YPOS-YD1)*SIN(ALPHAH)+(XPOS-XD1)*COS(ALPHAH)
     YCTE = (YPOS-YD1)*COS(ALPHAH)-(XPOS-XD1)*SIN(ALPHAH)
      ZCTE = (ZPOS-ZD1)*COS(ALPHAV)+XCTEH*SIN(ALPHAV)
      XCTEV=-(ZPOS-ZD1)*SIN(ALPHAV)+XCTEH*COS(ALPHAV)
C
C     HIT CRITERIA
C
      VTOTAL=(XD2-XD1)**2+(ZD2-ZD1)**2
      VTOTAL=SQRT(VTOTAL)
      HTOTAL=(XD2-XD1)**2+(YD2-YD1)**2
      HTOTAL=SQRT(HTOTAL)
      VAWAY =VTOTAL-XCTEV
      VAWAY =ABS(VAWAY)
      HAWAY =HTOTAL-XCTEH
      HAWAY =ABS(HAWAY)
      IF ((VAWAY.LT.TARGET).OR.(HAWAY.LT.TARGET)) GO TO
     &   101
C
C     DIVE PLANE INPUT CALCULATION
C
      ZPHI=ZCTE
      SIGV=ATAN(ZPHI/XDV)
      DS=K1V*(THETA-ALPHAV-SIGV)+K2V*W+K3V*Q+K4V
C
      IF (DS.GE. 0.4) DS= 0.4
      IF (DS.LE.-0.4) DS=-0.4
```

```
C
          DB=-DS
C
C         RUDDER INPUT CALCULATION
C
          YPHI=YCTE
          SIGH=-ATAN(YPHI/XDH)
          DRS=K1H*(PSI-ALPHAH-SIGH)+K2H*V+K3H*R
C
          IF (DRS.GE. 0.4) DRS= 0.4
          IF (DRS.LE.-0.4) DRS=-0.4
C
          DRB=-DRS
C
C         PRINT RESULTS
C
          TIME=I*DELTA
          JE=JE+1
          IF (JE.NE.IECHO) GO TO 99
          JE=0
   99     JPRNT=JPRNT+1
          IF (JPRNT.NE.IPRNT) GO TO 100
          IJK=IJK+1
          TIME=I*DELTA
          WRITE (11,*) XPOS/L,YPOS/L
          WRITE (12,*) XPOS/L,ZPOS/L
          WRITE (13,*) TIME,DRS*180.0/PI
          WRITE (14,*) TIME,DS*180.0/PI
          WRITE (15,*) TIME,YCTE/L
          WRITE (16,*) TIME,ZCTE/L
          WRITE (17,*) XPOS/L,YPOS/L,ZPOS/L
          WRITE (18,*) TIME,U
          WRITE (19,*) TIME,RPM
          WRITE (20,*) TIME,PHI*180.0/PI
          WRITE (21,*) TIME,(THETA-ALPHAV)*180.0/PI
          WRITE (22,*) TIME,(PSI-ALPHAH)*180.0/PI
          WRITE (23,*) TIME,V
          WRITE (24,*) TIME,R
          WRITE (25,*) TIME,W
          WRITE (26,*) TIME,Q
          WRITE (27,*) YPOS/L,ZPOS/L
          JPRNT=0
C
  100   CONTINUE
        GO TO 500
  101   ISTART=ICOUNT
  200 CONTINUE
  500 STOP
  201 FORMAT (' HEADING FOR (X,Y,Z) = ( ',F9.3,' , ',F9.3,' ,
     &          ',F9.3'  )')
      END
```

```
C
C=============================================================
C
      SUBROUTINE TRAP(N,A,B,OUT)
C
C      NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
        DIMENSION A(1),B(1)
        N1=N-1
        OUT=0.0
        DO 1 I=1,N1
          OUT1=0.5*(A(I)+A(I+1))*(B(I+1)-B(I))
          OUT =OUT+OUT1
      1 CONTINUE
        RETURN
        END
C
C=============================================================
C
       SUBROUTINE INVTA(MM,N,INDX,D)
        PARAMETER (NMAX=100,TINY=1.0E-20)
        DIMENSION INDX(6),VV(NMAX)
        REAL MM(6,6)
        D=1
        DO 12 I=1,N
          AAMAX=0.
          DO 11 J=1,N
            IF(ABS(MM(I,J)).GT.AAMAX) AAMAX=ABS(MM(I,J))
11        CONTINUE
          IF (AAMAX.EQ.0.) PAUSE 'SINGULAR MATRIX'
          VV(I)=1./AAMAX
12      CONTINUE
        DO 19 J=1,N
          DO 14 I=1,J-1
            SUM=MM(I,J)
            DO 13 K=1,I-1
              SUM=SUM-MM(I,K)*MM(K,J)
13          CONTINUE
            MM(I,J)=SUM
14        CONTINUE
          AAMAX=0.
          DO 16 I=J,N
            SUM=MM(I,J)
            DO 15  K=1,J-1
              SUM=SUM-MM(I,K)*MM(K,J)
15          CONTINUE
          MM(I,J)=SUM
          DUM=VV(I)*ABS(SUM)
          IF (DUM.GE.AAMAX) THEN
            IMAX=I
            AAMAX=DUM
```

```
        ENDIF
16      CONTINUE
        IF (J.NE.IMAX)THEN
           DO 17 K=1,N
             DUM=MM(IMAX,K)
             MM(IMAX,K)=MM(J,K)
             MM(J,K)=DUM
17         CONTINUE
           D=-D
           VV(IMAX)=VV(J)
        ENDIF
        INDX(J)=IMAX
        IF(MM(J,J).EQ.0.)MM(J,J)=TINY
        IF(J.NE.N)THEN
           DUM=1./MM(J,J)
           DO 18 I=J+1,N
             MM(I,J)=MM(I,J)*DUM
18         CONTINUE
        ENDIF
19    CONTINUE
      RETURN
      END
C
C=============================================================
C
      SUBROUTINE INVTB(MM,N,INDX,B)
      DIMENSION INDX(N),B(N)
      REAL MM(6,6)
      II=0.
      DO 12 I=1,N
        LL=INDX(I)
        SUM=B(LL)
        B(LL)=B(I)
        IF (II.NE.0)THEN
            DO 11 J=II,I-1
              SUM=SUM-MM(I,J)*B(J)
11          CONTINUE
          ELSE IF (SUM.NE.0) THEN
            II=I
          ENDIF
        B(I)=SUM
12    CONTINUE
      DO 14 I=N,1,-1
        SUM=B(I)
        IF (I.LT.N)THEN
          DO 13 J=I+1,N
            SUM=SUM-MM(I,J)*B(J)
13        CONTINUE
        ENDIF
        B(I)=SUM/MM(I,I)
14    CONTINUE
```

```
RETURN
END
```

APPENDIX B

```
C       PROGRAM VERT_STAB.FOR
C
C       REGIONS OF STABILITY - VERTICAL PLANE
C       PARAMETERS ARE: XD AND TV
C       NUMERICAL OR ANALYTIC COMPUTATION
C       IT NEEDS FILE "SUBRTNS.FOR" OR ANY STANDARD EIGENVALUE
C                             SOLVER
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DOUBLE PRECISION K1V,K2V,K3V,L
        DOUBLE PRECISION MQDOT,MQ,MW,MWDOT,MDS,MDB,MASS,IY
        DIMENSION A(4,4),FV1(4),IV1(4),ZZZ(4,4),WR(4),WI(4)
C
        OPEN (10,FILE='BIF0.RES',STATUS='NEW')
        OPEN (11,FILE='BIF1.RES',STATUS='NEW')
        OPEN (12,FILE='BIF2.RES',STATUS='NEW')
        OPEN (13,FILE='BIF3.RES',STATUS='NEW')
C
        WEIGHT=12000.0
        IY     = 9450.0
        L      =   17.425
        RHO    =    1.94
        G      =   32.2
        XG     =    0.0
        ZB     =    0.0
        MASS   =WEIGHT/G
        BOY    =WEIGHT
        ZQDOT  =-6.810E-03*0.5*RHO*L**4
        ZWDOT  =-2.430E-01*0.5*RHO*L**3
        ZQ     =-1.350E-01*0.5*RHO*L**3
        ZW     =-3.020E-01*0.5*RHO*L**2
        ZDS    =-2.270E-02*0.5*RHO*L**2
        ZDB    =-2.270E-02*0.5*RHO*L**2
        MQDOT  =-1.680E-02*0.5*RHO*L**5
        MWDOT  =-6.810E-02*0.5*RHO*L**4
        MQ     =-6.860E-02*0.5*RHO*L**4
        MW     = 9.860E-02*0.5*RHO*L**3
        MDS    =-1.113E-02*0.5*RHO*L**3
        MDB    = 1.113E-02*0.5*RHO*L**3
C
        WRITE (*,1001)
        READ  (*,*)     TVMIN,TVMAX,ITV
        WRITE (*,1002)
        READ  (*,*)     XDMIN,XDMAX,IXD
```

```
      XDMIN=XDMIN*L
      XDMAX=XDMAX*L
      WRITE (*,1003)
      READ  (*,*)    U,ZG
      WRITE (*,1004)
      READ  (*,*)    ISOL
C
C     AUXILIARY VARIABLES
C
      DV   =(MASS-ZWDOT)*(IY-MQDOT)-ZQDOT*MWDOT
      A11V=((IY-MQDOT)*ZW+ZQDOT*MW)/DV
      A12V=((IY-MQDOT)*(ZQ+MASS)+ZQDOT*MQ)/DV
      A13V=-(ZG-ZB)*(MASS*XG+ZQDOT)*WEIGHT/DV
      A21V=(MWDOT*ZW+(MASS-ZWDOT)*MW)/DV
      A22V=(MWDOT*(ZQ+MASS)+(MASS-ZWDOT)*MQ)/DV
      A23V=-(ZG-ZB)*(MASS-ZWDOT)*WEIGHT/DV
      B11V=((IY-MQDOT)*ZDS+ZQDOT*MDS)/DV
      B12V=((IY-MQDOT)*ZDB+ZQDOT*MDB)/DV
      B21V=(MWDOT*ZDS+(MASS-ZWDOT)*MDS)/DV
      B22V=(MWDOT*ZDB+(MASS-ZWDOT)*MDB)/DV
      B1V =B11V-B12V
      B2V =B21V-B22V
C
      EPS  =1.D-5
      ILMAX=1500
C
C     LOOP OVER TV
C
      DO 1 I=1,ITV
        WRITE (*,2001) I,ITV
        TV=TVMIN+(I-1)*(TVMAX-TVMIN)/(ITV-1)
        OMEGAV=(10.0*U)/(TV*L)
        AD1V=1.75*OMEGAV
        AD2V=2.15*OMEGAV**2
        AD3V=OMEGAV**3
        A2=B1V*U*U
        A3=B2V*U*U
        D1=-AD1V-(A11V+A22V)*U
        B1=-B2V*U*U
        B2=(B1V*A22V-B2V*A12V)*U**3
        B3=(B2V*A11V-B1V*A21V)*U**3
        D2=AD2V+A23V+(A12V*A21V-A11V*A22V)*U**2
        C1=(B2V*A11V-B1V*A21V)*U**3
        C2=(A23V*B1V-A13V*B2V)*U**2
        D3=AD3V+(A13V*A21V-A11V*A23V)*U
        K2V=(A3*B1*D3+C1*B3*D1-D2*C1*A3)
        K2V=K2V/(A3*B1*C2+C1*B3*A2-C1*A3*B2)
        K1V=(D3-C2*K2V)/C1
        K3V=(D1-A2*K2V)/A3
        D333=(A13V*A21V-A11V*A23V)*U
        XAAA=CBRT(-D333)
```

```
      D334=(D3*B2*C1*A3+B3*C1*A2*A3-B3*C1*D1*C2-D1*C1*C2*A3)
      D335=B3*C1*A2+B2*C1*A3-B1*C2*A3
      D336=D334/D335
      XBBB=CBRT(D336)
C
C     ANALYTICAL COMPUTATION
C
      IF (ZG.NE.0.0) TVCR1=(10.*U)/(XAAA*L)
      IF (ZG.NE.0.0) TVCR2=(10.*U)/(XAAA*L)
      IF (ISOL.EQ.0) GO TO 22
      CXD2=AD3V*(AD1V*AD2V-AD3V)
      CXD1=-(B2+A3*U)*K1V*(AD1V*AD2V-AD3V)+AD3V*K1V*
     &     (A2*AD1V+B2+A3*U)-AD1V*AD1V*K1V*(-C2+C1*U)
      CXD0=-(B2+A3*U)*K1V*K1V*(AD1V*A2+B2+A3*U)
      DET=CXD1*CXD1-4.0*CXD2*CXD0
      IF (DET.LT.0.0) GO TO 1
      XD1=(-CXD1+DSQRT(DET))/(2.0*CXD2)
      XD2=(-CXD1-DSQRT(DET))/(2.0*CXD2)
      IF (XD1.NE.0.0)
     &  VAL1=AD3V+((B2V*A12V-B1V*A22V-B2V)*K1V*U**3)/XD1
      IF (XD2.NE.0.0)
     &  VAL2=AD3V+((B2V*A12V-B1V*A22V-B2V)*K1V*U**3)/XD2
      GO TO 23
C
C     NUMERICAL COMPUTATION
C     LOOP OVER XD
C
   22 DO 2 J=1,IXD
        XD=XDMIN+(J-1)*(XDMAX-XDMIN)/(IXD-1)
        THETA=0.0D0
        CT=DCOS(THETA)
        ST=DSIN(THETA)
        W=0.0D0
        A(1,1)=0.0D0
        A(1,2)=0.0D0
        A(1,3)=1.0D0
        A(1,4)=0.0D0
        A(2,1)=B1V*U*U*K1V+A13V*CT
        A(2,2)=B1V*U*U*K2V+A11V*U
        A(2,3)=B1V*U*U*K3V+A12V*U
        A(2,4)=-B1V*U*U*K1V/XD
        A(3,1)=B2V*U*U*K1V+A23V*CT
        A(3,2)=B2V*U*U*K2V+A21V*U
        A(3,3)=B2V*U*U*K3V+A22V*U
        A(3,4)=-B2V*U*U*K1V/XD
        A(4,1)=-U*CT-W*ST
        A(4,2)=CT
        A(4,3)=0.0D0
        A(4,4)=0.0D0
C
C       COMPUTE EIGENVALUES
```

```
C
      CALL RG(4,4,A,WR,WI,0,ZZZ,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
C
      IF (J.GT.1) GO TO 10
      DEOSOO=DEOS
      XDOO  =XD
      LL=0
      GO TO 2
10    DEOSNN=DEOS
      XDNN  =XD
      PR=DEOSNN*DEOSOO
      IF (PR.GT.0.D0) GO TO 3
      LL=LL+1
      IF (LL.GT.3) STOP 1000
      IL=0
      XDO=XDOO
      XDN=XDNN
      DEOSO=DEOSOO
      DEOSN=DEOSNN
 6    XDL=XDO
      XDR=XDN
      DEOSL=DEOSO
      DEOSR=DEOSN
      XD=(XDL+XDR)/2.D0
      A(1,1)=0.0D0
      A(1,2)=0.0D0
      A(1,3)=1.0D0
      A(1,4)=0.0D0
      A(2,1)=B1V*U*U*K1V+A13V*CT
      A(2,2)=B1V*U*U*K2V+A11V*U
      A(2,3)=B1V*U*U*K3V+A12V*U
      A(2,4)=-B1V*U*U*K1V/XD
      A(3,1)=B2V*U*U*K1V+A23V*CT
      A(3,2)=B2V*U*U*K2V+A21V*U
      A(3,3)=B2V*U*U*K3V+A22V*U
      A(3,4)=-B2V*U*U*K1V/XD
      A(4,1)=-U*CT-W*ST
      A(4,2)=CT
      A(4,3)=0.0D0
      A(4,4)=0.0D0
C
      CALL RG(4,4,A,WR,WI,0,ZZZ,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
C
      DEOSM=DEOS
      XDM=XD
      PRL=DEOSL*DEOSM
      PRR=DEOSR*DEOSM
      IF (PRL.GT.0.D0) GO TO 5
      XDO=XDL
```

```
          XDN=XDM
          DEOSO=DEOSL
          DEOSN=DEOSM
          IL=IL+1
          IF (IL.GT.ILMAX) STOP 3100
          DIF=DABS(XDL-XDM)
          IF (DIF.GT.EPS) GO TO 6
          XD=XDM
          GO TO 4
    5     IF (PRR.GT.0.D0) STOP 3200
          XDO=XDM
          XDN=XDR
          DEOSO=DEOSM
          DEOSN=DEOSR
          IL=IL+1
          IF (IL.GT.ILMAX) STOP 3100
          DIF=DABS(XDM-XDR)
          IF (DIF.GT.EPS) GO TO 6
          XD=XDM
    4     LLL=10+LL
          WRITE (LLL,*) XD/L,TV
    3     XDOO=XDNN
          DEOSOO=DEOSNN
    2   CONTINUE
        GO TO 1
   23   IF (VAL1.GT.0.0) WRITE (11,*) XD1/L,TV
        IF (VAL2.GT.0.0) WRITE (12,*) XD2/L,TV
    1 CONTINUE
      IF (ZG.NE.0.0) WRITE (10,*) XDMIN/L,TVCR1
      IF (ZG.NE.0.0) WRITE (10,*) XDMAX/L,TVCR1
C
 1001 FORMAT (' ENTER MIN, MAX, AND INCREMENTS OF TV')
 1002 FORMAT (' ENTER MIN, MAX, AND INCREMENTS OF XD')
 1003 FORMAT (' ENTER U AND ZG')
 1004 FORMAT (' ENTER 0 : NUMERICAL',/,
     &        '       1 : ANALYTICAL')
 2001 FORMAT (2I5)
      END
C
      SUBROUTINE DSTABL(DEOS,WR,WI,OMEGA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WR(4),WI(4)
      DEOS=-1.0D+20
      DO 1 I=1,4
        IF (WR(I).LT.DEOS) GO TO 1
        DEOS=WR(I)
        IJ=I
    1 CONTINUE
      OMEGA=WI(IJ)
      OMEGA=DABS(OMEGA)
      RETURN
```

```
      END
C
      FUNCTION CBRT(A)
      IF (A.GT.0.0) CBRT=   A **(1./3.)
      IF (A.LE.0.0) CBRT=-(-A)**(1./3.)
      RETURN
      END
```

APPENDIX C

```
C       PROGRAM VERT_STEADY.FOR
C
C        COMPUTATION OF STEADY STATE SOLUTIONS IN THE VERTICAL
C                                PLANE
C       (CHAPTER III, PARAGRAPH F)
C
        REAL K1V,K2V,K3V,L,MQDOT,MQ,MW,MWDOT,MDS,MDB,MASS,IY
C
        WEIGHT=12000.0
        IY     = 9450.0
        L      =   17.425
        RHO    =    1.94
        G      =   32.2
        XG     =    0.0
        ZB     =    0.0
        MASS   =WEIGHT/G
        BOY    =WEIGHT
C
        ZQDOT =-6.810E-03*0.5*RHO*L**4
        ZWDOT =-2.430E-01*0.5*RHO*L**3
        ZQ    =-1.350E-01*0.5*RHO*L**3
        ZW    =-3.020E-01*0.5*RHO*L**2
        ZDS   =-2.270E-02*0.5*RHO*L**2
        ZDB   =-2.270E-02*0.5*RHO*L**2
C
        MQDOT =-1.680E-02*0.5*RHO*L**5
        MWDOT =-6.810E-02*0.5*RHO*L**4
        MQ    =-6.860E-02*0.5*RHO*L**4
        MW    = 9.860E-02*0.5*RHO*L**3
        MDS   =-1.113E-02*0.5*RHO*L**3
        MDB   = 1.113E-02*0.5*RHO*L**3
C
        OPEN (11,FILE='THETA1.RES',STATUS='NEW')
        OPEN (12,FILE='THETA2.RES',STATUS='NEW')
        OPEN (13,FILE='THETA3.RES',STATUS='NEW')
        OPEN (14,FILE='THETA4.RES',STATUS='NEW')
        OPEN (21,FILE='DELTA1.RES',STATUS='NEW')
        OPEN (22,FILE='DELTA2.RES',STATUS='NEW')
C
        SAT =0.4
        SATP= SAT
        SATM=-SAT
        PI  =4.0*ATAN(1.0)
C
```

```
      WRITE (*,1001)
      READ  (*,*)    IVAR
      GO TO (10,20,30), IVAR
   10 WRITE (*,1002)
      READ  (*,*)    UMIN,UMAX,IU
      INCR=IU
      WRITE (*,1003)
      READ  (*,*)    ZG
      WRITE (*,1006)
      READ  (*,*)    TV
      GO TO 15
   20 WRITE (*,1004)
      READ  (*,*)    ZGMIN,ZGMAX,IZG
      INCR=IZG
      WRITE (*,1005)
      READ  (*,*)    U
      WRITE (*,1006)
      READ  (*,*)    TV
      GO TO 15
   30 WRITE (*,1007)
      READ  (*,*)    TVMIN,TVMAX,ITV
      INCR=ITV
      WRITE (*,1003)
      READ  (*,*)    ZG
      WRITE (*,1005)
      READ  (*,*)    U
C
   15 DO 1 I=1,INCR
        IF (IVAR.EQ.1) U =UMIN +(UMAX -UMIN )*(I-1)/(INCR-1)
        IF (IVAR.EQ.2) ZG=ZGMIN+(ZGMAX-ZGMIN)*(I-1)/(INCR-1)
        IF (IVAR.EQ.3) TV=TVMIN+(TVMAX-TVMIN)*(I-1)/(INCR-1)
        DV  =(MASS-ZWDOT)*(IY-MQDOT)-ZQDOT*MWDOT
        A11V=((IY-MQDOT)*ZW+ZQDOT*MW)/DV
        A12V=((IY-MQDOT)*(ZQ+MASS)+ZQDOT*MQ)/DV
        A13V=-(ZG-ZB)*(MASS*XG+ZQDOT)*WEIGHT/DV
        A21V=(MWDOT*ZW+(MASS-ZWDOT)*MW)/DV
        A22V=(MWDOT*(ZQ+MASS)+(MASS-ZWDOT)*MQ)/DV
        A23V=-(ZG-ZB)*(MASS-ZWDOT)*WEIGHT/DV
        B11V=((IY-MQDOT)*ZDS+ZQDOT*MDS)/DV
        B12V=((IY-MQDOT)*ZDB+ZQDOT*MDB)/DV
        B21V=(MWDOT*ZDS+(MASS-ZWDOT)*MDS)/DV
        B22V=(MWDOT*ZDB+(MASS-ZWDOT)*MDB)/DV
        B1V =B11V-B12V
        B2V =B21V-B22V
C
        OMEGAV=(10.0*U)/(TV*L)
        AD1V=1.75*OMEGAV
        AD2V=2.15*OMEGAV**2
        AD3V=OMEGAV**3
        A2=B1V*U*U
        A3=B2V*U*U
```

```
      D1=-AD1V-(A11V+A22V)*U
      B1=-B2V*U*U
      B2=(B1V*A22V-B2V*A12V)*U**3
      B3=(B2V*A11V-B1V*A21V)*U**3
      D2=AD2V+A23V+(A12V*A21V-A11V*A22V)*U**2
      C1=(B2V*A11V-B1V*A21V)*U**3
      C2=(A23V*B1V-A13V*B2V)*U**2
      D3=AD3V+(A13V*A21V-A11V*A23V)*U
      K2V=(A3*B1*D3+C1*B3*D1-D2*C1*A3)
      K2V=K2V/(A3*B1*C2+C1*B3*A2-C1*A3*B2)
      K1V=(D3-C2*K2V)/C1
      K3V=(D1-A2*K2V)/A3
C
      IF (IVAR.EQ.1) OUT=U
      IF (IVAR.EQ.2) OUT=ZG
      IF (IVAR.EQ.3) OUT=TV
      D3P=(A13V*A21V-A11V*A23V)*U
      XAAA=CBRT(-D3P)
      TVCR=(10.*U)/(XAAA*L)
      IF (TV.LT.TVCR) GO TO 1
C
      CALL SOLSET(INUM,THSOLS,K1V,C1,AD3V,SSTH)
      ICHECK=0
      DO 2 III=1,INUM
        THCH=2.0*(THSOLS-0.5*PI)
        CHECK=SIN(THCH)*(D3-AD3V)/C1
        IF (ABS(CHECK).GT.SATP) GO TO 2
        WRITE (13,*) OUT, THCH*180.0/PI
        WRITE (14,*) OUT,-THCH*180.0/PI
        WRITE (21,*) OUT, ABS(CHECK)*180.0/PI
        ICHECK=1
    2 CONTINUE
      IF (ICHECK.EQ.0) GO TO 3
      GO TO 1
C
    3 STHETA=SATP*C1/(D3-AD3V)
      SSTH=ASIN(STHETA)
      THETA1= SSTH*180.0/PI
      THETA2=-SSTH*180.0/PI
      WRITE (11,*) OUT,THETA1
      WRITE (12,*) OUT,THETA2
      WRITE (22,*) OUT, SATP*180.0/PI
    1 CONTINUE
      STOP
 1001 FORMAT (' ENTER 1 : U  VARIATION',/,
     &       '       2 : ZG VARIATION',/,
     &       '       3 : TV VARIATION')
 1002 FORMAT (' ENTER MIN, MAX, AND INCREMENTS IN U')
 1003 FORMAT (' ENTER ZG')
 1004 FORMAT (' ENTER MIN, MAX, AND INCREMENTS IN ZG')
 1005 FORMAT (' ENTER U')
```

```
 1006 FORMAT (' ENTER TV')
 1007 FORMAT (' ENTER MIN, MAX, AND INCREMENTS IN TV')
           END
C
      FUNCTION CBRT(A)
      IF (A.GT.0.0) CBRT=   A **(1./3.)
      IF (A.LE.0.0) CBRT=-(-A)**(1./3.)
      RETURN
      END
C
      SUBROUTINE SOLSET(L,ANS,K1V,C1,AD3V,SSTH)
      REAL K1V
      DIMENSION VF(1,2)
C
      PI=4.0*ATAN(1.0)
C
C     FIND FIRST ESTIMATE OF THE SOLUTIONS
      L=0
      VMIN=  0.0
      VMAX=+90.0
      IV=100
      VA=VMIN*PI/180.0
      VAO=VA
      VO=THETEQ(1,VA,K1V,C1,AD3V)
      DO 10 I=2,IV
        VA=VMIN+(VMAX-VMIN)*(I-1)/(IV-1)
        VA=VA*PI/180.0
        VAN=VA
        VN=THETEQ(1,VA,K1V,C1,AD3V)
        VP=VO*VN
        IF (VP.GE.0.0) GO TO 11
        L=L+1
        VF(L,1)=VAO
        VF(L,2)=VAN
        GO TO 12
   11   VO=VN
        VAO=VAN
   10 CONTINUE
C
C     EXACT COMPUTATION OF SOLUTIONS VIA NEWTON'S METHOD
   12 E=1.E-5
      IEND=500
      DO 20 J=1,L
        X=(VF(J,1)+VF(J,2))/2.0
        F=THETEQ(1,X,K1V,C1,AD3V)
        FDER=THETEQ(2,X,K1V,C1,AD3V)
        DO 30 K=1,IEND
          IF (FDER.EQ.0.0) STOP 1001
          DX=F/FDER
          X1=X-DX
          F=THETEQ(1,X1,K1V,C1,AD3V)
```

```
      FDER=THETEQ(2,X1,K1V,C1,AD3V)
      IF (F.EQ.0.) GO TO 35
      A=ABS(X1-X)
      IF (A-E) 35,35,40
40    X=X1
30  CONTINUE
    GO TO 20
35  ANS=X1
20 CONTINUE
   RETURN
   END
C
   FUNCTION THETEQ(K,THETA,K1V,C1,AD3V)
   REAL K1V
   GO TO (10,20), K
10 THETEQ=K1V*C1*THETA+(AD3V-K1V*C1)*COS(THETA)
   GO TO 50
20 THETEQ=K1V*C1-(AD3V-K1V*C1)*SIN(THETA)
50 RETURN
   END
```

LIST OF REFERENCES

1. Lienard, D.L. (1990) "Autopilot design for autonomous underwater vehicles based on sliding mode control", M.E. Thesis, Mechanical Engineering, Naval Postgraduate School, Monterey, California.

2. Sur, J.-N. (1989) "Design and investigation of a dive plane sliding mode compensator for an autonomous underwater vehicle", M.S. Thesis, Naval Postgraduate School, Monterey, California.

3. Chism, S. (1990) "Robust path tracking of autonomous underwater vehicles using sliding modes", M.E. Thesis, Naval Postgraduate School, Monterey, California.

4. Hawkinson, T. (1990) "Multiple input sliding mode control for autonomous diving and steering of underwater vehicles", M.E. Thesis Naval Postgraduate School, Monterey, California.

5. Kanyama, Y, and Hartman, B.I. (1989) "Smooth local path planning for autonomous vehicles", Proceedings, IEEE international Conference on Robotics and Automation, Scottsdale, Arizona.

6. Suwandee, P. (1991) "Orientation guidance and control for marine vehicles in the horizontal plane", M.S. Thesis, Naval Postgraduate School.

7. Papoulias, F.A. (1991) "Stability considerations of guidance and control laws for autonomous underwater vehicles in the horizontal plane", Proceedings, 7th International Symposium on Unmanned Underwater Submersible Technology, Durham, New Hampshire.

8. Smith, N.S., Crane, J.W., and Summery,D.C. (1978) " SDV Simulator hydrodynamic coefficients", Naval Coastal Systems Center, Panama City, Florida, Report No. NCSC - TM231 - 78.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Dr. Fotis A. Papoulias,Code 69Pa Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	2
4.	Chairman, Code 69Hy Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1
5.	Embassy of Greece Naval Attache 2228 Massachusetts Ave., N.W. Washington, D.C. 20008	5
6.	Agelos G. Papasotiriou Polemi 3 Patisia Athens, Greece	3