NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Nonlinear Study of Open Loop Dynamic Stability of Submersible Vehicles in the Dive Plane

by

Harilaos I. Papadimitriou

March, 1994

Thesis Advisor: Fotis A. Papoulias

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE *March 1994	3.REPORT TYPE AND DATES COVERED Engineers Thesis

4. TITLE AND SUBTITLE * A NONLINEAR STUDY OF OPEN LOOP DYNAMIC STABILITY OF SUBMERSIBLE VEHICLES IN THE DIVE PLANE (U)

5. FUNDING NUMBERS

6. AUTHOR(S) *HARILAOS, PAPADIMITRIOU, I.

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)

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
*A

13. ABSTRACT *(maximum 200 words)*

* This thesis presents a comprehensive nonlinear study of straight line stability of motion of submersibles in the dive plane under open loop conditions. A systematic perturbation analysis demonstrates that the effects of surge on heave/pitch are small and can be neglected. Primary loss of stability is shown to occur in the form of Hopf bifurcations to periodic solutions. Analysis of the periodic solutions that result from these Hopf bifurcations was accomplished through Taylor expansions, up to third order, of the equations of motion. A consistent approximation, utilizing the generalized gradient, is used to study the non-analytic quadratic cross flow integral drag terms. The results indicate that loss of stability occurs always in the form of supercritical Hopf bifurcations with stable limit cycles. It is shown that this is mainly due to the stabilizing effect of the drag forces at high angles of attack.

14. SUBJECT TERMS * Submarine Stabilty, Bifurcations, Periodic Solutions.

15. NUMBER OF PAGES * 105

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

A Nonlinear Study of Open Loop Dynamic Stability of Submersible Vehicles in the Dive Plane

by

Harilaos I. Papadimitriou
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL

March, 1994

Author: _____
Harilaos I. Papadimitriou

Approved by: _____
Fotis A. Papoulias, Thesis Advisor

_____
Matthew D. Kelleher, Chairman
Department of Mechanical Engineering

_____
Richard S. Elster
Dean of Instruction

ABSTRACT

This thesis presents a comprehensive nonlinear study of straight line stability of motion of submersibles in the dive plane under open loop conditions. A systematic perturbation analysis demonstrates that the effects of surge on heave/pitch are small and can be neglected. Primary loss of stability is shown to occur in the form of Hopf bifurcations to periodic solutions. Analysis of the periodic solutions that result from these Hopf bifurcations was accomplished through Taylor expansions, up to third order, of the equations of motion. A consistent approximation, utilizing the generalized gradient, is used to study the non-analytic quadratic cross flow integral drag terms. The results indicate that loss of stability occurs always in the form of supercritical Hopf bifurcations with stable limit cycles. It is shown that this is mainly due to the stabilizing effect of the drag forces at high angles of attack.

TABLE OF CONTENTS

I. INTRODUCTION

A. PROBLEM STATEMENT

The dynamic response of a submersible vehicle operating at the extremes of its operational envelope is becoming increasingly important in order to enhance vehicle operations. Traditionally, dynamic stability of motion is studied using eigenvalue analysis where the equations of motion are linearized around nominal straight line level flight paths (Arentzen & Mandel, 1960), (Clayton & Bishop, 1982), (Feldman, 1987). Under certain simplified assumptions, a simple criterion $G_v > 0$ can be obtained where the stability index G_v is function of the hydrodynamic coefficients in heave and pitch. Values for the stability index can be computed by,

$$G_v = 1 - \frac{M_w(Z_q + m)}{Z_w M_q} . \tag{1}$$

This index is analogous to the familiar stability coefficient for horizontal plane maneuvering (Lewis, 1989) and can be thought of as a high speed approximation where the effect of the metacentric restoring moment is minimal. If the value of G_v is greater than zero, the vehicle is dynamically stable. As we point out in the next chapter though, this is only a sufficient, and rather conservative condition for stability. It is not a necessary condition in the sense that $G_v < 0$ indicates instability at infinite forward speed. It is quite possible that at normal operating speeds the vehicle might be directionally stable. Furthermore, $G_v < 0$ indicates a divergent loss of stability which is

quite uncommon in the vertical plane. Most modern submarines exhibit a flutter–like instability at high speed, which can not be analyzed using the above simplified index. Divergent motions may develop in combined six degrees of freedom (Papoulias *et al* 1993) and their occurrence can not be analyzed by a single stability index.

B. OBJECTIVES AND OUTLINE

In this work we examine the problem of stability of motion with controls fixed in the vertical plane, with particular emphasis on the mechanism of loss of stability of straight line motion. The surge equation is decoupled from heave/pitch through a perturbation series approach (Bender & Orszag, 1978). It is shown that loss of stability occurs in the form of generic bifurcations to periodic solutions (Guckenheimer & Holmes, 1983). Taylor expansions and center manifold approximations are employed in order to isolate the main nonlinear terms that influence system response after the initial loss of stability (Hassard & Wan, 1978). Integral averaging is performed in order to combine the nonlinear terms into a design stability coefficient (Chow & Mallet–Paret, 1977). Special attention is paid to the study of the quadratic drag terms as they constitute some of the main nonlinear terms of the equations of motion. This is a unique feature of the open loop problem. In similar studies of loss of stability under closed loop depth control (Bateman, 1993) it was found that the main damping mechanism is provided through the use

of control susrafces. The difficulty associated with the nonsmoothness of the absolute value nonlinearities of the quadratic drag forces is dealt with by employing the concept of generalized gradient (Clarke, 1983). This has the advantage of keeping the linear terms constant, unlike the linear/cubic approximation typically used in ship roll motion studies (Dalzell, 1978), where the linear damping coefficient is a function of the assumed amplitude of motion.

Vehicle modeling in this work follows standard notation (Gertler & Hagen, 1976), (Smith *et al* 1978), and numerical results are presented for the DARPA SUBOFF model (Roddy, 1990) for which a set of hydrodynamic coefficients and geometric properties is available. Furthermore, the baseline vehicle is marginally stable with controls fixed under normal operating conditions and can serve as a prime example for the techniques described in this work. The model has been experimentally validated for angles of attack on the hull between ± 15 deg., while the constant coefficient approximation introduces very little error in time domain simulations (Tinker, 1978). Unless otherwise mentioned, all results in this work are presented in standard dimensionless form with respect to the vehicle length $\ell = 4.26$ m, and nominal forward speed $U = 2.44$ m/sec. All angular deflections are shown in degrees.

II. PROBLEM FORMULATION

A. EQUATIONS OF MOTION

Assuming that vehicle motion is restricted in the vertical plane, the mathematical model consists of the coupled nonlinear heave and pitch equations of motion. In a moving coordinate frame fixed at the vehicle's geometrical center, Newton's equations of motion for a port/starboard symmetric and neutrally buoyant vehicle are expressed in dimensionless form as follows,

$$m(\dot{w} - uq - z_G q^2 - x_G \dot{q}) = Z_{\dot{q}}\dot{q} + Z_{\dot{w}}\dot{w} + Z_q q + Z_w w$$
$$-C_D \int_{\text{tail}}^{\text{nose}} b(x)(w - xq)|w - xq|\, dx + Z_\delta \delta \,, \tag{2}$$

$$I_y \dot{q} + m z_G(\dot{u} + wq) - m x_G(\dot{w} - uq) = M_{\dot{q}}\dot{q} + M_{\dot{w}}\dot{w} + M_q q + M_w w$$
$$+C_D \int_{\text{tail}}^{\text{nose}} b(x)(w - xq)|w - xq|\, x\, dx$$
$$-x_{GB} W \cos\theta - z_{GB} W \sin\theta + M_\delta \delta \,, \tag{3}$$

where $x_{GB} = x_G - x_B$, $z_{GB} = z_G - z_B$, and the rest of the symbols are based on standard notation and are explained in Table 1. Without loss of generality we can assume that $z_B = x_B = 0$, so that $x_{GB} = x_G$ and $z_{GB} = z_G$. The cross flow integral terms in these equations become very important for high angles of attack maneuvering, where they provide the primary motion damping. The drag coefficient, C_D, is assumed to be constant throughout the vehicle length for simplicity. This does not affect the qualitative properties of the

results that follow. The vehicle pitch rate is,

$$\dot{\theta} = q \, . \tag{4}$$

Dynamic coupling between surge and heave/pitch is present due to coordinate coupling as a result of the nonzero metacentric height. Therefore, pitch and heave motions must be studied together with surge,

$$\begin{aligned} m\dot{u} + mwq - mx_G q^2 + mz_G \dot{q} &= X_{qq} q^2 + X_{\dot{u}} \dot{u} + X_{wq} wq + X_{ww} w^2 \\ &+ X_{uu} u^2 + X_{nn} n^2 + X_{\delta\delta} \delta^2 \, , \end{aligned} \tag{5}$$

where we assume that both resistance and propulsive forces are proportional to the square of the speed or the propeller revolutions, respectively.

In analyzing controls fixed stability of motion, the case $\delta = 0$ is examined first. The steady state solutions of the equations are determined by $\dot{w} = \dot{q} = \dot{u} = \dot{\theta} = q_0 = 0$, where subscript 0 indicates variable value at steady state. Substituting these conditions in (2) we get,

$$Z_w w_0 - C_D A_w w_0 |w_0| = 0 \, , \tag{6}$$

where,

$$A_w = \int_{\text{tail}}^{\text{nose}} b(x) \, dx \, , \tag{7}$$

is the "waterplane" area. Since $Z_w < 0$, equation (7) admits only one solution, namely $w_0 = 0$. Equation (3) then yields,

$$\tan \theta_0 = -\frac{x_{GB}}{z_{GB}} \, , \tag{8}$$

while (5) is used to determine the nominal forward speed, u_0.

TABLE 1: NOMENCLATURE

a	dummy independent variable
a_0	steady state value of a
α_{ij}	expansion coefficients of z_3 in terms of z_1, z_2
$b(x)$	local beam of the hull
C_D	quadratic drag coefficient
γ	regularization parameter
δ	stern plane deflection
ε	perturbation parameter, $\varepsilon = (mz_G^2)$
ϵ	criticality difference, $\epsilon = u - u_c$
I_y	vehicle mass moment of inertia
K	nonlinear stability coefficient
λ	system eigenvalue
m	vehicle mass
M	pitch moment
M_a	derivative of M with respect to a
q	pitch rate
(R, ϕ)	polar coordinates of z_1, z_2
$\mathbf{T}$	transformation matrix of $\mathbf{x}$ to $\mathbf{z}$
θ	pitch angle
u	forward speed
u_c	critical value of u
w	heave velocity
$\mathbf{x}$	state variables vector, $\mathbf{x} = [\theta, w, q]$
X	surge force
X_a	derivative of X with respect to a
(x_B, z_B)	body fixed coordinates of vehicle center of buoyancy
(x_G, z_G)	body fixed coordinates of vehicle center of gravity
x_{GB}	center of gravity/center of buoyancy separation, $x_G - x_B$
z_{GB}	vehicle metacentric height, $z_G - z_B$
$\mathbf{z}$	state variables vector in its normal form
z_1, z_2	critical coordinates of $\mathbf{z}$
z_3	stable coordinate of $\mathbf{z}$
Z	heave force
Z_a	derivative of Z with respect to a
ω_0	imaginary part of critical pair of eigenvalues

B. REDUCTION OF ORDER

The linearized surge, heave, and pitch equations of motion in the vicinity of the nominal point are,

$$(m - X_{\dot{u}})\dot{u} + mz_G\dot{q} = 2X_{uu}u \ , \tag{9}$$

$$(m - Z_{\dot{w}})\dot{w} - (mx_G + Z_{\dot{q}}) = (Z_q + m)q + Z_w w \ , \tag{10}$$

$$(I_y - M_{\dot{q}})\dot{q} + mz_G\dot{u} - (mx_G + M_{\dot{w}})\dot{w} = M_w w + (M_q - mx_G)q + M_\theta \theta \ , \tag{11}$$

where,

$$M_\theta = x_{GB}W\sin\theta_0 - z_{GB}W\cos\theta_0 \ , \tag{12}$$

is the hydrostatic restoring moment coefficient. The characteristic equation of (4), (9), (10), and (11) is obtained as,

$$(-A_1\lambda + B_1)(A_2\lambda^3 + B_2\lambda^2 + C_2\lambda + D_2) + A_3\lambda^4 + B_3\lambda^3 = 0 \ , \tag{13}$$

where,

$$\begin{aligned}
A_1 &= m - X_{\dot{u}} \ , \\
B_1 &= 2X_{uu} \ , \\
A_2 &= (m - Z_{\dot{w}})(I_y - M_{\dot{q}}) - (Z_{\dot{q}} + mx_G)(M_{\dot{w}} + mx_G) \ , \\
B_2 &= -Z_w(I_y - M_{\dot{q}}) - (m - Z_{\dot{w}})(M_q - mx_G) - (Z_q + m)(M_{\dot{w}} + mx_G) \\
&\quad -M_w(Z_{\dot{q}} + mx_G) \ , \\
C_2 &= -M_\theta(m - Z_{\dot{w}}) + Z_w(M_q - mx_G) - M_w(Z_q + m) \ , \\
D_2 &= M_\theta Z_w \ ,
\end{aligned}$$

$$A_3 = (mz_G)^2(m - Z_{\dot{w}}) ,$$

$$B_3 = -(mz_G)^2 Z_w .$$

It can be seen that the parameter (mz_G) is responsible for surge and heave, pitch coupling. For $z_G = 0$, equation (13) decouples into the surge eigenvalue $\lambda = B_1/A_1$ and the classical cubic characteristic equation for the vertical plane. It should be mentioned that the effect of the forward speed u is embedded into the definition for the dimensionless vehicle weight W through,

$$W \longrightarrow \frac{W}{\frac{1}{2}\rho u^2 L^2} . \tag{14}$$

If we introduce a smallness parameter,

$$\epsilon = (mz_G)^2 , \tag{15}$$

we can rewrite (13) in the form,

$$(A + \epsilon\alpha)\lambda^4 + (B + \epsilon\beta)\lambda^3 + C\lambda^2 + D\lambda + E = 0 , \tag{16}$$

where in terms of previously defined coefficients,

$$A = -A_1 a_2 ,$$

$$B = -A_1 B_2 + B_1 A_2 ,$$

$$C = -A_1 C_2 + B_1 B_2 ,$$

$$D = -A_1 D_2 + B_1 C_2 ,$$

$$E = B_1 D_2 ,$$

$$\alpha = m - Z_{\dot{w}} ,$$

$$\beta = -Z_w .$$

Following (Bender & Orszag, 1978) we expand the solutions of (16) in a regular perturbation series,

$$\lambda = \lambda_0 + \lambda_1 \epsilon + O(\epsilon^2) \,, \tag{17}$$

where λ_0 is an eigenvalue for $\epsilon = 0$ (uncoupled surge or heave/pitch), λ_1 is the first order correction due to dynamic coupling, and $O(\epsilon^2)$ contains second and higher order terms in ϵ. Substituting (17) into (16) we can get,

$$\lambda = \lambda_0 - \frac{\lambda_0^3(\alpha\lambda_0 + \beta)}{4A\lambda_0^3 + 3B\lambda_0^2 + 2C\lambda_0 + D}\epsilon + O(\epsilon^2) \,. \tag{18}$$

It can be seen that the correction term is very small compared to the uncoupled root as evidenced by the λ_0^3 term. This is particularly true when λ_0 nears zero; i.e., close to a bifurcation point. Therefore, loss of stability can be studied by analyzing the heave/pitch equations decoupled from surge. The characteristic equation then becomes,

$$A_2\lambda^3 + B_2\lambda^2 + C_2\lambda + D_2 = 0 \,. \tag{19}$$

Plots of the system eigenvalues at nominal speed versus z_G are shown in Figures 1 through 4. The surge eigenvalues is real and negative throughout the range of z_G, while the heave/pitch eigenvalues are real for small values of z_G. The two larger real heave/pitch eigenvalues coalesce into a complex conjugate pair whose real part crosses zero for a certain value of z_G. Within the accuracy of Figures 1 through 3, the eigenvalues λ are identical to those computed by either the coupled or the uncoupled system, or the perturbation equations (18). There is a very small difference for the surge eigenvalue as

shown in Figure 4, but again the agreement between coupled and uncoupled systems is very good.

C. DEGREE OF STABILITY

The eigenvalues of the reduced order system (18) designate stability or instability of motion. A single measure of stability, the "degree of stability", e_v, can be defined as the maximum real part of all three eigenvalues of (18). This measures the slowest exponential convergence to the equilibrium when negative and the fastest exponential divergence from the equilibrium when positive. For all numerical calculations in this work, the degree of stability corresponds to the real part of a complex conjugate pair of eigenvalues. Typical results are presented in Figures 5 through 7. Figure 5 shows the degree of stability versus x_{GB} constant dimensionless speed $u_0 = 0.5$ and different values of z_{GB}. Similar results are shown in Figure 6 for constant $z_{GB} = 0.015$ and different values of u_0. Finally, a three dimensional representation is depicted in Figure 7.

D. CRITICAL SPEED

The parameter value where the real part of the complex conjugate pair of eigenvalues shown in the previous figures crosses zero defines the point where linear stability is lost. This critical point can be computed by considering

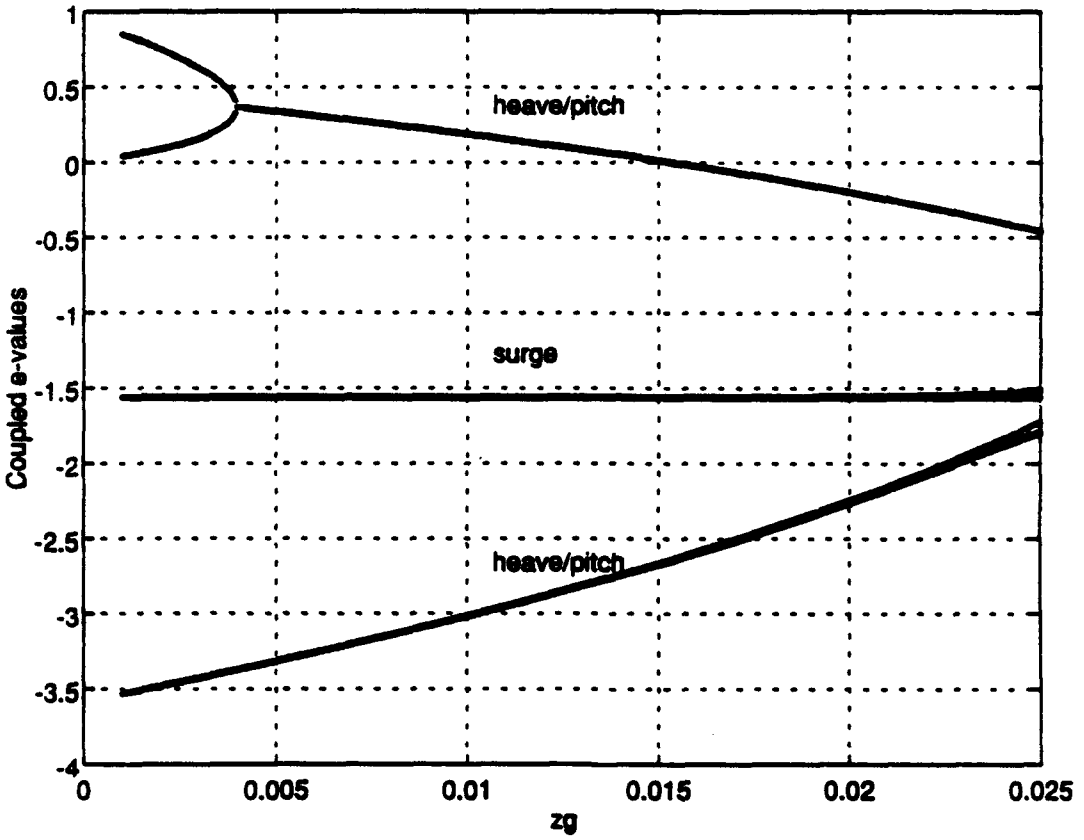

Figure 1: System eigenvalues versus z_G

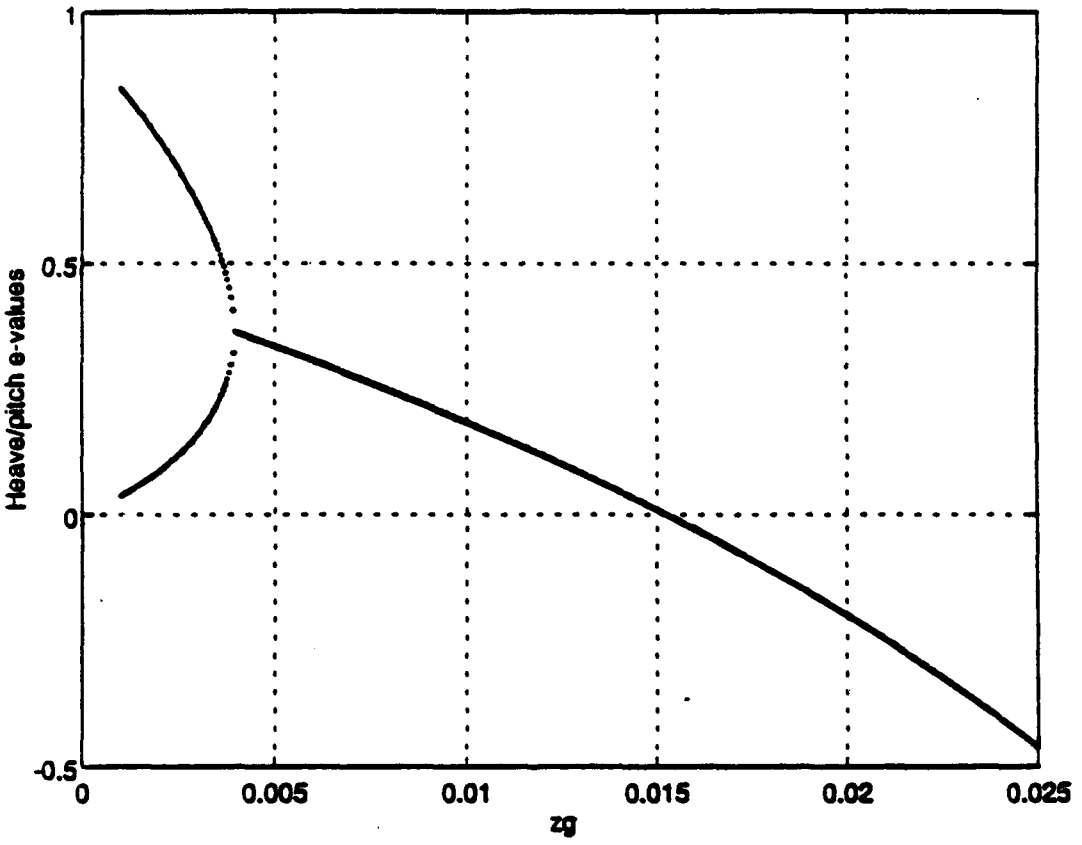

Figure 2: Complex conjugate heave/pitch eigenvalues versus z_G

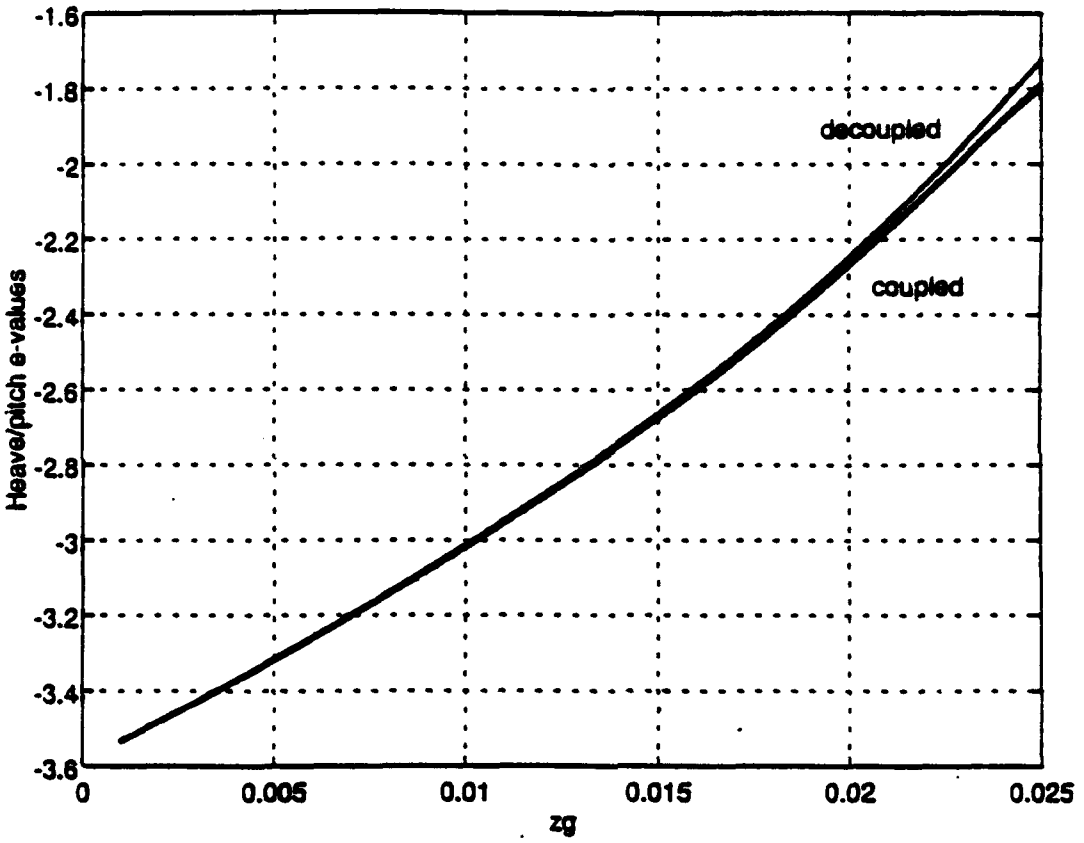

Figure 3: Real heave/pitch eigenvalues versus z_G

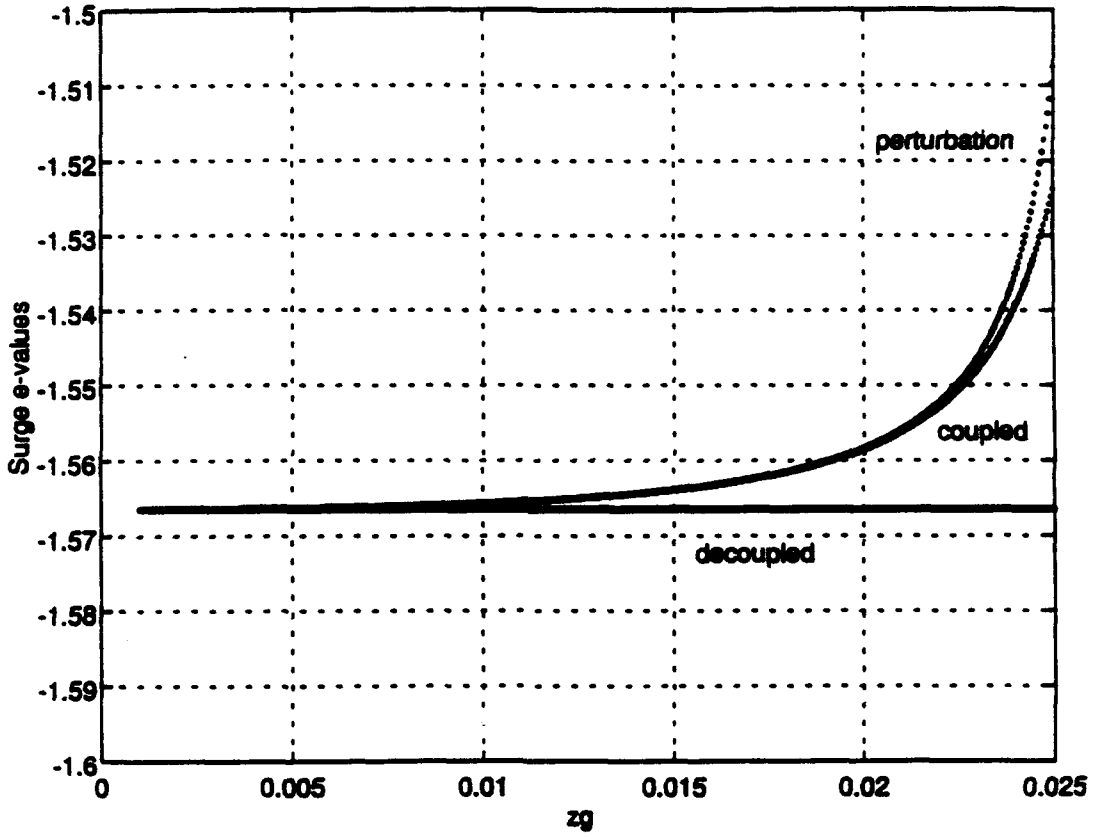

Figure 4: Surge eigenvalue versus z_G

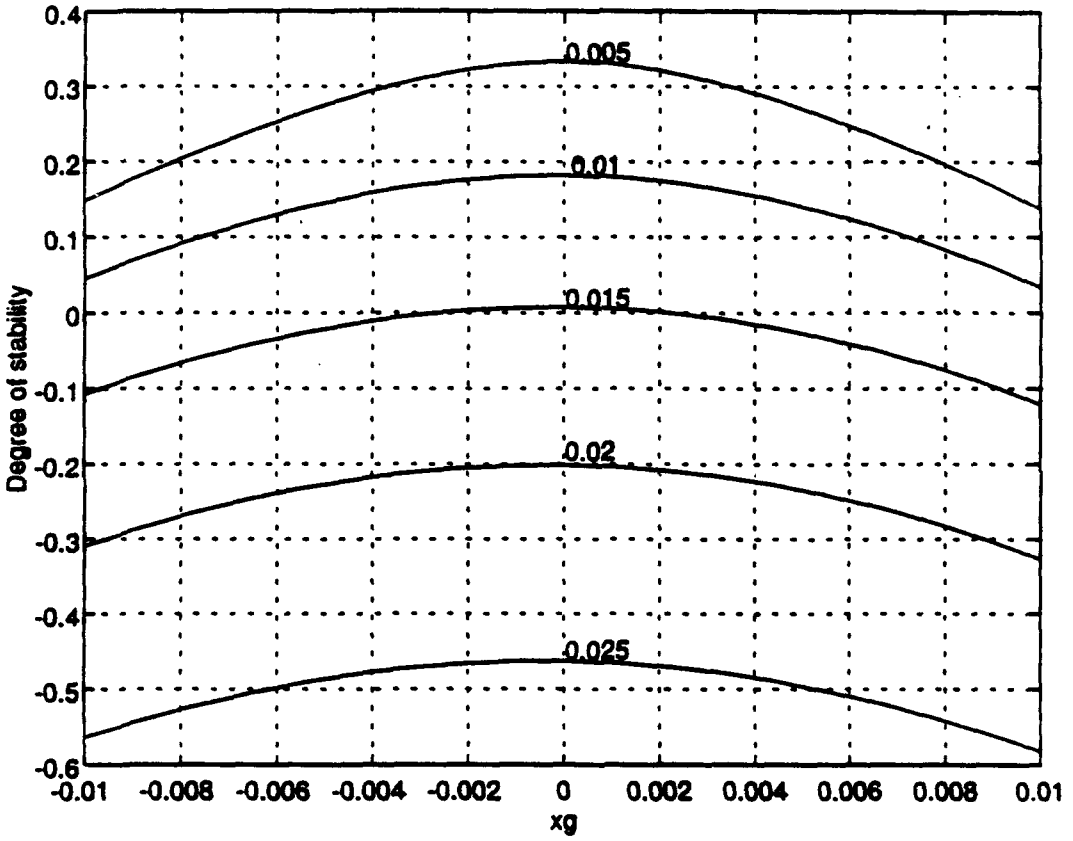

Figure 5: Degree of stability versus z_{GB} for $u_0 = 0.5$ and different values of z_{GB}

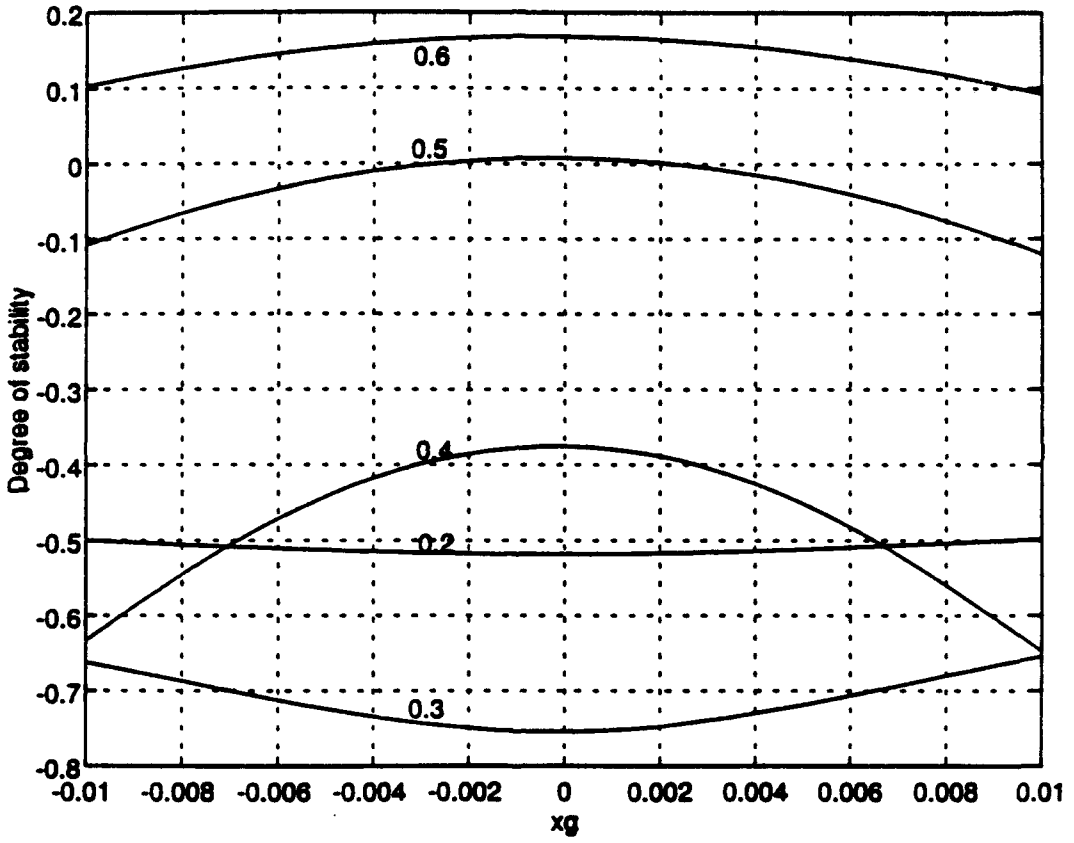

Figure 6: Degree of stability versus x_{GB} for $z_{GB} = 0.015$ and different values of u_0

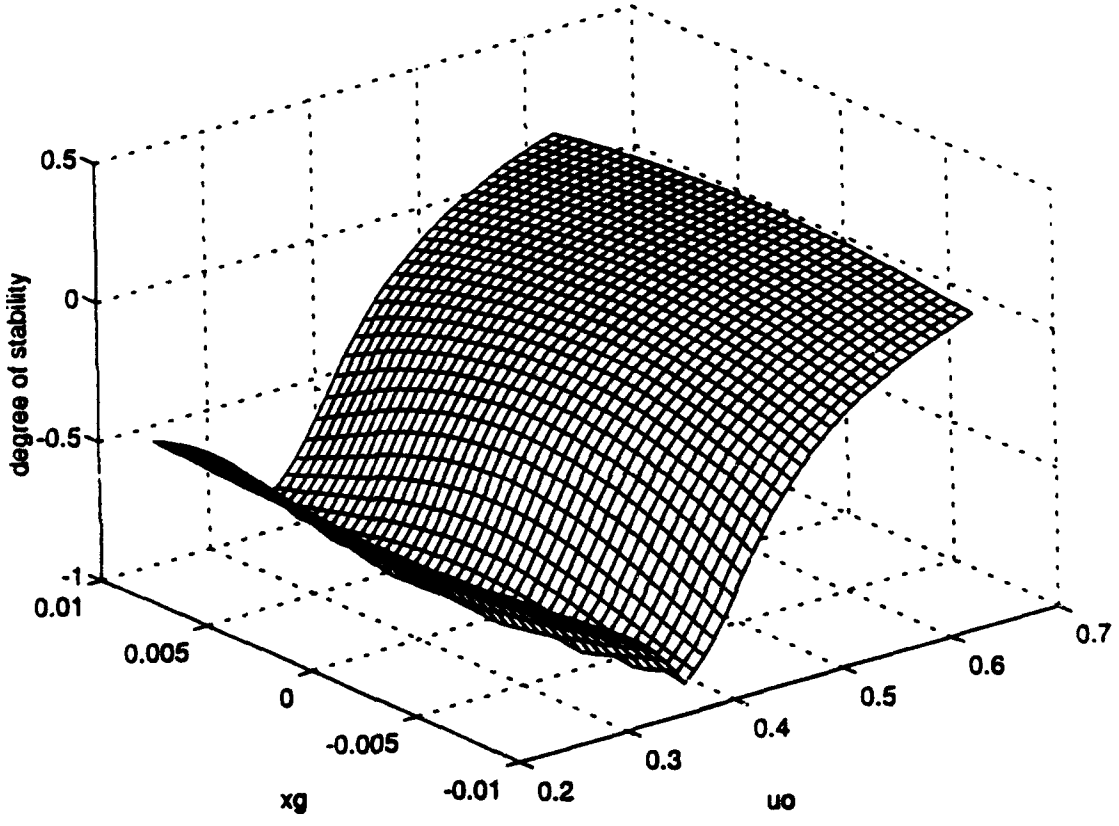

Figure 7: Degree of stability versus x_{GB} and u_0 for $z_{GB} = 0.015$

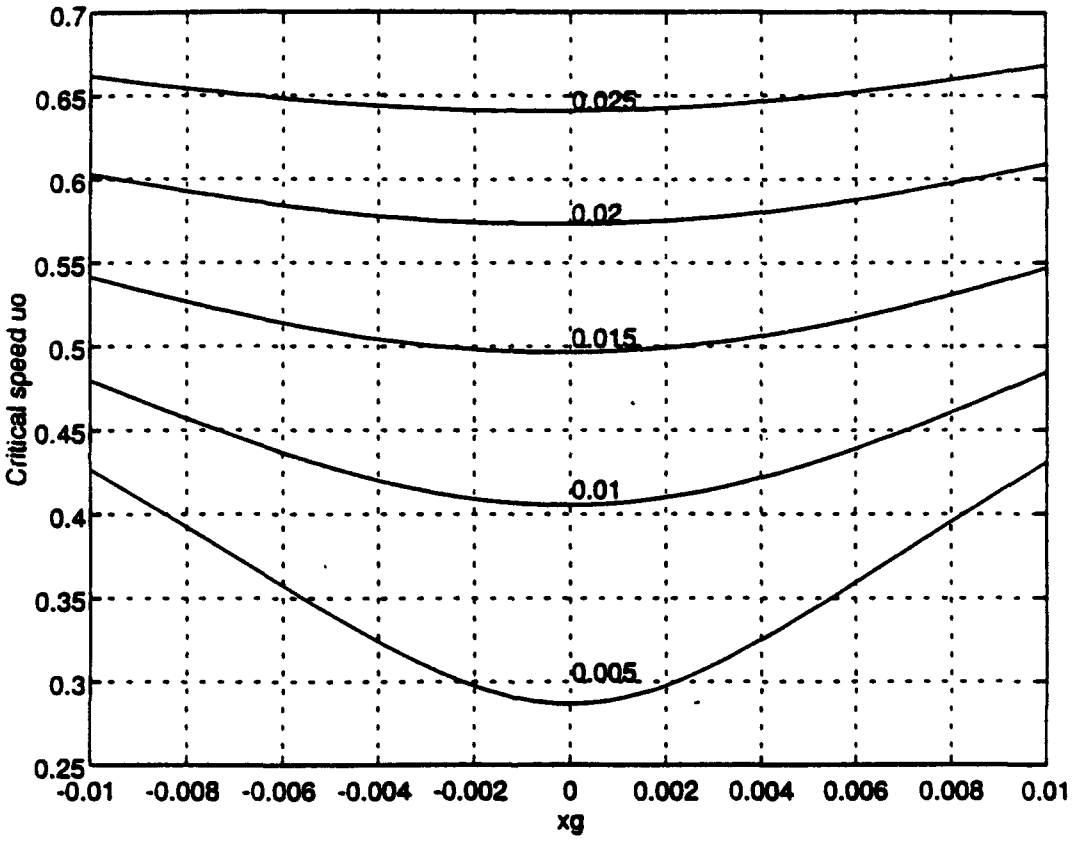

Figure 8: Critical speed u_c versus x_G for different values of z_{GB}

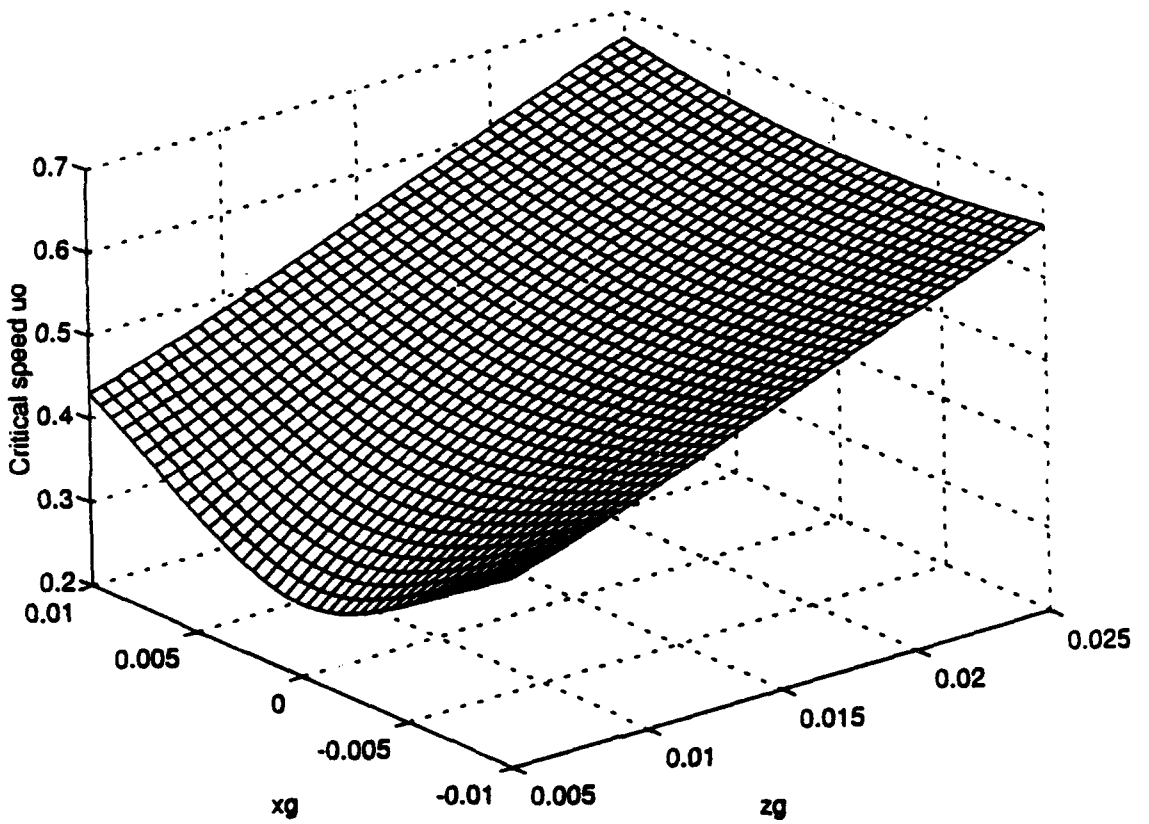

Figure 9: Critical speed u_c versus x_G and z_G

equation (19). Routh's criterion applied to this cubic yields $A_2 D_2 = B_2 C_2$ which can be solved for the dimensionless weight,

$$W = \frac{B_2 C_{2,0}}{A_2 D_{2,1} - B_2 C_{2,1}} , \tag{20}$$

where,

$$\begin{aligned} C_{2,0} &= Z_w(M_q - m x_G) - M_w(Z_q + m) , \\ C_{2,1} &= (m - Z_{\dot{w}})(z_{GB} \cos\theta_0 - x_{GB} \sin\theta_0) , \\ D_{2,1} &= Z_w(x_{GB} \sin\theta_0 - z_{GB} \cos\theta_0) . \end{aligned}$$

The value of the critical speed u_c can then be evaluated from (20) and (14). Typical results are presented in Figures 8 and 9, nondimensionalized with respect to nominal vehicle speed 2.44 m/sec and length 4.26 m. Vertical plane motions are stable for forward speeds less than the critical speed. It can be seen that stability is increasing with increasing z_G while $x_G = 0$ is the most conservative condition for stability. Therefore, a vehicle which is stable when properly trimmed will remain stable for off-trim conditions. For comparison, we note that the simple stability coefficient G_v, defined in equation (1), is monotonically decreasing and becomes more negative for decreasing x_G, as shown in Figure 10. Thus it would have predicted unstable motions for the entire range of parameters shown in Figures 8 and 9. For completeness, the value of the steady state pitch angle, θ_0 (in degrees), is shown in Figure 11 versus x_{GB} using z_{GB} as the parameter. This pitch angle is computed from equation (8).

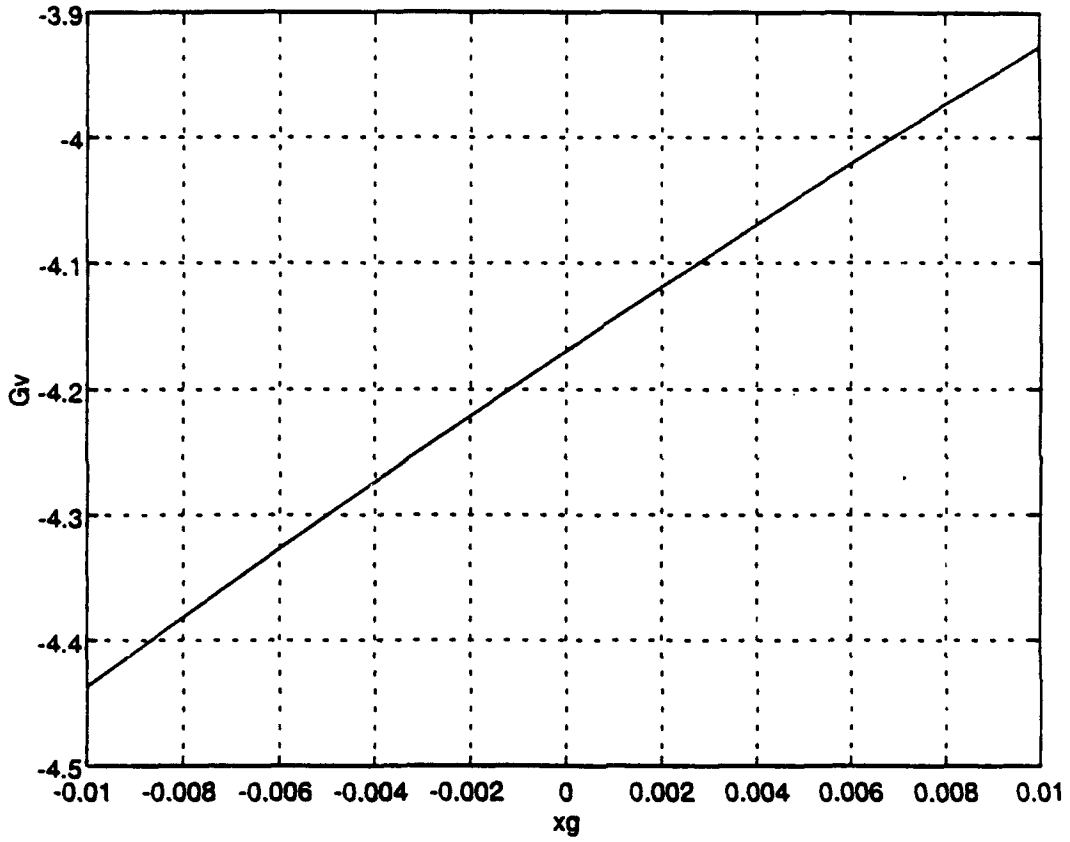

Figure 10: Stability coefficient G_v versus x_{GB}

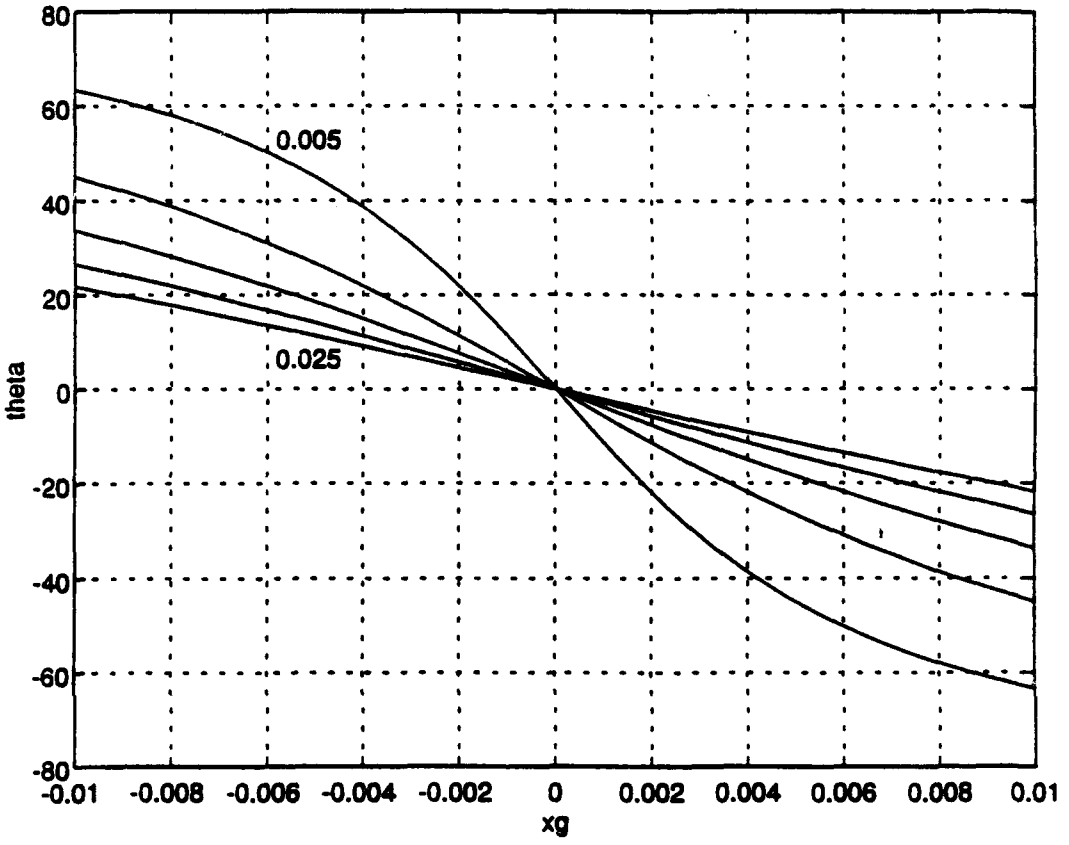

Figure 11: Steady state pitch angle θ_0 versus x_{GB} for z_{GB} from 0.005 to 0.025 with increments of 0.005

III. BIFURCATION ANALYSIS

A. INTRODUCTION

In all cases of stability loss of the previous chapter, one pair of complex conjugate eigenvalues of the corresponding eigenvalue problem crosses transversally the imaginary axis. A situation like this in which a certain parameter is varied such that the real part of one pair of complex conjugate eigenvalues of the linearized system matrix crosses zero, results in the system leaving its steady state in an oscillatory manner. This loss of stability is called Hopf bifurcation and generically occurs in either supercritical or subcritical form. In the supercritical case, stable limit cycles are generated after the nominal straight line motion loses its stability. The amplitudes of these limit cycles are continuously increasing as the parameter distance from its critical value is increased. For small values of this criticality distance the resulting limit cycle is of small amplitude and differs little from the initial nominal state. In the subcritical case, however, stable limit cycles are generated before the nominal state loses its stability. Therefore, depending on the initial conditions it is possible to diverge away from the nominal straight line path and converge towards a limit cycle even before the nominal motion loses its stability. This means that in the subcritical Hopf bifurcation case the domain of attraction of the nominal state is decreasing and in fact it shrinks to zero as the critical point is approached. Random external disturbances of

sufficient magnitude can throw the vehicle off to an oscillatory steady state even though the nominal state may still remain stable. After the nominal state becomes unstable, a discontinuous increase in the magnitude of motions is observed as there exist no simple stable nearby attractors for the vehicle to converge to. Distinction between these two qualitatively different types of bifurcation is, therefore, essential in design. The computational procedure requires higher order approximations in the equations of motion and is the subject of this chapter.

B. THIRD ORDER EXPANSIONS

The nonlinear heave/pitch equations of motion (2), (3), and (4) are written in the form,

$$\dot{\theta} = q\,, \tag{21}$$

$$\dot{w} = a_{11}w + a_{12}q + a_{13}(x_{GB}\cos\theta + z_{GB}\sin\theta) + d_w(w,q) + c_1(w,q)\,, \tag{22}$$

$$\dot{q} = a_{21}w + a_{22}q + a_{23}(x_{GB}\cos\theta + z_{GB}\sin\theta) + d_q(w,q) + c_2(w,q)\,, \tag{23}$$

where

$$\begin{aligned} D_v &= (m - Z_{\dot{w}})(I_y - M_{\dot{q}}) - (mx_G + Z_{\dot{q}})(mx_G + M_{\dot{w}})\,, \\ a_{11}D_v &= (I_y - M_{\dot{q}})Z_w + (mx_G + Z_{\dot{q}})M_w\,, \\ a_{12}D_v &= (I_y - M_{\dot{q}})(m + Z_q) + (mx_G + Z_{\dot{q}})(M_q - mx_G)\,, \end{aligned}$$

$$
\begin{aligned}
a_{12}D_v &= -(mx_G + Z_{\dot{q}})W \ , \\
a_{21}D_v &= (m - Z_{\dot{w}})M_w + (mx_G + M_{\dot{w}})Z_w \ , \\
a_{22}D_v &= (m - Z_{\dot{w}})(M_q - mx_G) + (mx_G + M_{\dot{w}})(m + Z_q) \ , \\
a_{23}D_v &= -(m - Z_{\dot{w}})W \ , \\
d_w(w,q)D_v &= (I_y - M_{\dot{q}})I_w + (mx_G + Z_{\dot{q}})I_q \ , \\
d_q(w,q)D_v &= (m - Z_{\dot{w}})I_q + (mx_G + M_{\dot{w}})I_w \ , \\
c_1(w,q)D_v &= (I_y - M_{\dot{q}})mz_G q^2 - (mx_G + Z_{\dot{q}})mz_G wq \ , \\
c_2(w,q)D_v &= -(m - Z_{\dot{w}})mz_G wq + (mx_G + M_{\dot{w}})mz_G q^2 \ ,
\end{aligned}
$$

and I_w, I_q are the cross flow integrals

$$
I_w = C_D \int_{\text{tail}}^{\text{nose}} b(x)(w - xq)|w - xq| \, dx \ , \tag{24}
$$

$$
I_q = C_D \int_{\text{tail}}^{\text{nose}} b(x)(w - xq)|w - xq| x \, dx \ . \tag{25}
$$

The system of equations (21) through (23) is written in the compact form

$$
\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{g}(\mathbf{x}) \ , \tag{26}
$$

where

$$
\mathbf{x} = [\theta, w, q] \ , \tag{27}
$$

is the three state variables vector, and **A** is the linearized sytem matrix evaluated at the nominal point $\mathbf{x}_0$. The term $\mathbf{g}(\mathbf{x})$ contains all nonlinear terms of the equations. Hopf bifurcation analysis can be performed by isolating the primary nonlinear terms in $\mathbf{g}(\mathbf{x})$. Keeping terms up to third order, we can write

$$
\mathbf{g}(\mathbf{x}) = \mathbf{g}^{(2)}(\mathbf{x}) + \mathbf{g}^{(3)}(\mathbf{x}) \ . \tag{28}
$$

Using equations (21) through (25), the various terms in (28) can be written as,

$$\begin{aligned} g_1^{(2)} &= 0\,, \\ g_2^{(2)} &= (I_y - M_{\dot{q}})mz_G q^2 - (mx_G + Z_{\dot{q}})mz_G wq + d_w^{(2)}(w,q)\,, \\ g_3^{(2)} &= -(m - Z_{\dot{w}})mz_G wq + (mx_G + M_{\dot{w}})mz_G q^2 + d_q^{(2)}(w,q)\,, \end{aligned} \tag{29}$$

and

$$\begin{aligned} g_1^{(3)} &= 0\,, \\ g_2^{(3)} &= d_w^{(3)}(w,q) + \tfrac{1}{6}a_{13}(x_{GB}\sin\theta_0 - z_{GB}\cos\theta_0)\theta^3\,, \\ g_3^{(3)} &= d_q^{(3)}(w,q) + \tfrac{1}{6}a_{23}(x_{GB}\sin\theta_0 - z_{GB}\cos\theta_0)\theta^3\,. \end{aligned} \tag{30}$$

Expansion in Taylor series of d_w, d_q requires expansion of the cross flow integrals I_w, I_q, which require the Taylor series of

$$f(\xi) = \xi|\xi|\,. \tag{31}$$

This expression can be converted into an analytic function using Dalzell's approximation (Dalzell, 1978),

$$\xi|\xi| \approx \frac{5}{16}\xi_c\xi + \frac{35}{48}\frac{\xi^3}{\xi_c}\,, \tag{32}$$

which is derived by a least squares fit of an odd series over some assumed range of ξ, namely $-\xi_c < \xi < \xi_c$. This approximation, which is shown in Figure 12, has been extensively used in ship roll motion studies and is very useful for its intended purpose. However, in the present problem it suffers from several major drawbacks:

- It introduces a linear term which depends on the assumed range of motion, and it renders the critical speed function of the vehicle motions.
- The cubic term, which is ultimately responsible for the Hopf bifurcation analysis, is a function of the assumed range of vehicle motions which can not be known in advance.
- As Figure 12 demonstrates, the slope of the actual curve at the origin is significanly different than the approximation, which would make the bifurcation results unreliable.

Instead of Dalzell's approximation, we employ the concept of generalized gradient (Clarke, 1983), which is used in the study of control systems involving discontinuous or non–smooth functions. In this way we approximate the gradient of a non–smooth function at a discontinuity by a map equal to the convex closure of the limiting gradients near the discontinuity. In our problem we write,

$$f(\xi) = \xi_0|\xi_0| + 2|\xi_0|(\xi - \xi_0) + \text{sign}(\xi_0)(\xi - \xi_0)^2 + f^{(3)}(\xi)\,, \qquad (33)$$

as the Taylor series epansion of $f(\xi)$ near ξ_0. The sign function in (33) can be approximated by,

$$\text{sign}(\xi_0) = \lim_{\gamma \to 0} \tanh\left(\frac{\xi_0}{\gamma}\right)\,. \qquad (34)$$

A graphical representation of the approximation (34) is shown in Figure 13. The quantity γ is a small regularization parameter and is used in the

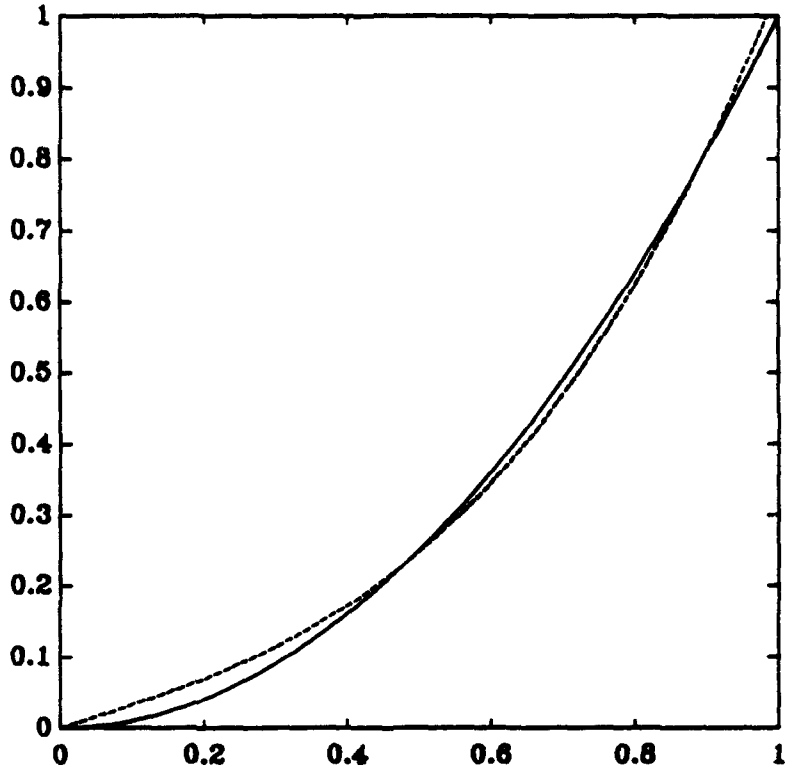

Figure 12: Graphical representation of Dalzell's approximation of $\xi|\xi|$ versus ξ/ξ_c. Solid curve is the exact expression and dotted curve is the approximation (32)

next section for the proper normalization of the results. Using (34), we can approximate $f(\xi)$ in the vicinity of $\xi_0 = 0$ by,

$$\xi|\xi| \approx \frac{1}{6\gamma}\xi^3 \,. \tag{35}$$

Since

$$\xi \mapsto w - xq \,, \tag{36}$$

we can express the non-smooth cross flow integral terms by,

$$I_w = \frac{C_D}{6\gamma}(E_0 w^3 - 3E_1 w^2 q + 3E_2 w q^2 - E_3 q^3) \,, \tag{37}$$

$$I_q = \frac{C_D}{6\gamma}(E_1 w^3 - 3E_2 w^2 q + 3E_3 w q^2 - E_4 q^3) \,, \tag{38}$$

where

$$E_i = \int_{\text{tail}}^{\text{nose}} x^i b(x)\, dx \,, \tag{39}$$

are the moments of the vehicle "waterplane" area.

C. COORDINATE TRANSFORMATIONS

Using the previous second and third order Taylor series expansions, equation (26) is written in the form,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{g}^{(2)}(\mathbf{x}) + \mathbf{g}^{(3)}(\mathbf{x}) \,. \tag{40}$$

If $\mathbf{T}$ is the matrix of eigenvectors of $\mathbf{A}$ evaluated at the critical point $u = u_c$, the linear change of coordinates,

$$\mathbf{x} = \mathbf{T}\mathbf{z} \,, \quad \mathbf{z} = \mathbf{T}^{-1}\mathbf{x} \,, \tag{41}$$

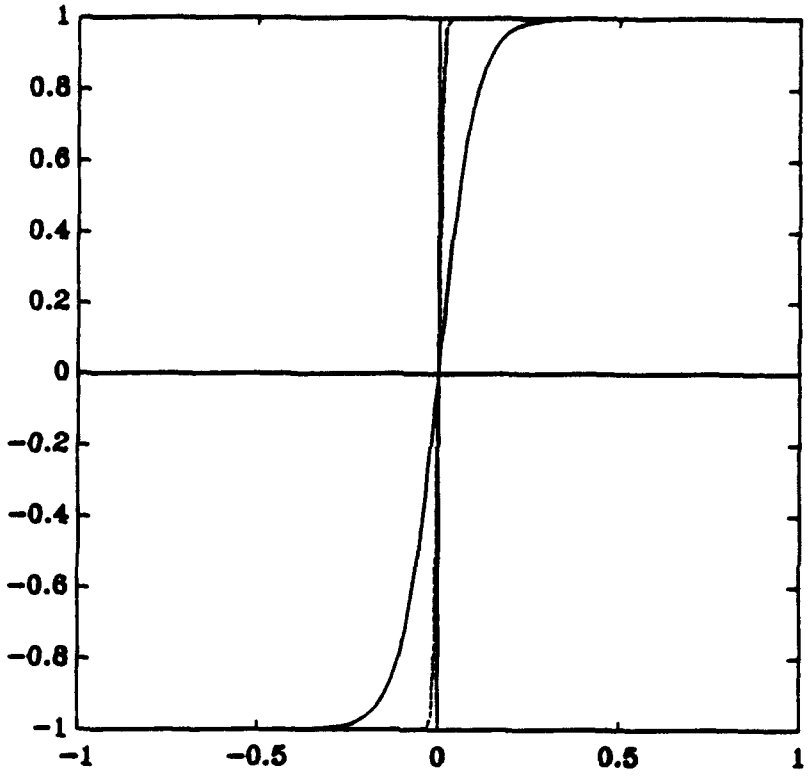

Figure 13: Graphical representation of the sign function and its hyperbolic tangent approximation (34). Solid curve corresponds to $\gamma = 0.1$ and dotted curve corresponds to $\gamma = 0.01$

transforms system (40) into its normal coordinate form,

$$\dot{\mathbf{s}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{s} + \mathbf{T}^{-1}\mathbf{g}^{(2)}(\mathbf{T}\mathbf{s}) + \mathbf{T}^{-1}\mathbf{g}^{(3)}(\mathbf{T}\mathbf{s})\,. \tag{42}$$

At the Hopf bifurcation point, matrix $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ takes the form,

$$\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{bmatrix} 0 & -\omega_0 & 0 \\ \omega_0 & 0 & 0 \\ 0 & 0 & p \end{bmatrix},$$

where ω_0 is the imaginary part of the critical pair of eigenvalues, and the remaining eigenvalue p is negative. For values of u close to the bifurcation poit u_c, matrix $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ becomes,

$$\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{bmatrix} \alpha'\epsilon & -(\omega_0 + \omega'\epsilon) & 0 \\ (\omega_0 + \omega'\epsilon) & \alpha'\epsilon & 0 \\ 0 & 0 & p + p'\epsilon \end{bmatrix},$$

where ϵ denotes the criticality difference

$$\epsilon = u - u_c\,, \tag{43}$$

and

α' = derivative of the real part of the critical eigenvalue with respect to ϵ,

ω' = derivative of the imaginary part of the critical eigenvalue with respect to ϵ,

p' = derivative of p with respect to ϵ.

D. CENTER MANIFOLD EXPANSIONS

Due to continuity, the eigevalue $p+p'\epsilon$ remains negative for small nonzero values of ϵ. Therefore, the coordinate z_3 corresponds to a negative eigenvalue and is asymptotically stable. Center manifold theory predicts that the relationship between the critical coordinates z_1, z_2 and the stable coordinate z_3 is at least of quadratic order. We can then write z_3 as,

$$z_3 = \alpha_{11} z_1^2 + \alpha_{12} z_1 z_2 + \alpha_{22} z_2^2 \,, \tag{44}$$

where the coefficients, α_{ij}, in the quadratic center manifold expansion (44) need to be determined. By differentiating equation (44) we obtain,

$$\dot{z}_3 = 2\alpha_{11} z_1 \dot{z}_1 + \alpha_{12}(\dot{z}_1 z_2 + z_1 \dot{z}_2) + 2\alpha_{22} z_2 \dot{z}_2 \,. \tag{45}$$

We substitute $\dot{z}_1 = -\omega_0 z_2$ and $\dot{z}_2 = \omega_0 z_1$ from equation (42) into (45), and we obtain

$$\dot{z}_3 = \alpha_{12}\omega_0 z_1^2 + 2(\alpha_{22} - \alpha_{11})\omega_0 z_1 z_2 - \alpha_{12}\omega_0 z_2^2 \,. \tag{46}$$

The third equation of (42) is written as,

$$\dot{z}_3 = p z_3 + \left[\mathbf{T}^{-1}\mathbf{g}^{(2)}(\mathbf{T}\mathbf{z})\right]_{(3,3)} \,, \tag{47}$$

where terms up to second order have been kept. If we denote the elements of $\mathbf{T}$ and $\mathbf{T}^{-1}$ by,

$$\mathbf{T} = [m_{ij}] \,, \quad \mathbf{T}^{-1} = [n_{ij}] \,, \tag{48}$$

then

$$\mathbf{T}^{-1}\mathbf{g}^{(2)}(\mathbf{T}\mathbf{z}) = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \,,$$

where

$$d_1 = n_{12}(\ell_{25}z_1^2 + \ell_{26}z_1z_2 + \ell_{27}z_2^2) + n_{13}(\ell_{35}z_1^2 + \ell_{36}z_1z_2 + \ell_{37}z_2^2)\,, \quad (49)$$

$$d_2 = n_{22}(\ell_{25}z_1^2 + \ell_{26}z_1z_2 + \ell_{27}z_2^2) + n_{23}(\ell_{35}z_1^2 + \ell_{36}z_1z_2 + \ell_{37}z_2^2)\,, \quad (50)$$

$$d_3 = n_{32}(\ell_{25}z_1^2 + \ell_{26}z_1z_2 + \ell_{27}z_2^2) + n_{33}(\ell_{35}z_1^2 + \ell_{36}z_1z_2 + \ell_{37}z_2^2)\,. \quad (51)$$

Expressions for the coefficients ℓ_{ij} are given in the Fortran programs in the appendix.

Equation (47) then becomes

$$\dot{z}_3 = pz_3 + d_3\,, \quad (52)$$

and substituting (44) and (51) into (52) we get,

$$\begin{aligned}\dot{z}_3 &= (p\alpha_{11} + n_{32}\ell_{25} + n_{33}\ell_{35})z_1^2 + (p\alpha_{12} + n_{32}\ell_{26} + n_{33}\ell_{36})z_1z_2 \\ &\quad +(p\alpha_{22} + n_{32}\ell_{27} + n_{33}\ell_{37})z_2^2\,. \quad (53)\end{aligned}$$

Comparing coefficients of (46) and (53) we get,

$$-p\alpha_{11} + \omega_0\alpha_{12} = n_{32}\ell_{25} + n_{33}\ell_{35}\,, \quad (54)$$

$$-2\omega_0\alpha_{11} - p\alpha_{12} + 2\omega_0\alpha_{22} = n_{32}\ell_{26} + n_{33}\ell_{36}\,, \quad (55)$$

$$-\omega_0\alpha_{12} - p\alpha_{22} = n_{32}\ell_{27} + n_{33}\ell_{37}\,. \quad (56)$$

Solution of the system of linear equations (54) through (56) yields the coefficients in the center manifold expansion (44).

E. AVERAGING

Using the previous Taylor expansions and center manifold approximations, we can write the reduced two–dimensional system that describes the center manifold flow of (42) in the form,

$$\dot{z}_1 = \alpha'\epsilon z_1 - (\omega_0 + \omega'\epsilon)z_2 + F_1(z_1, z_2)\,, \tag{57}$$

$$\dot{z}_2 = (\omega_0 + \omega'\epsilon)z_1 + \alpha'\epsilon z_2 + F_2(z_1, z_2)\,, \tag{58}$$

where

$$\begin{aligned} F_1(z_1, z_2) &= r_{11}z_1^3 + r_{12}z_1^2 z_2 + r_{13}z_1 z_2^2 + r_{14}z_2^3 \\ &\quad + p_{11}z_1^2 + p_{12}z_1 z_2 + p_{13}z_2^2\,, \end{aligned} \tag{59}$$

$$\begin{aligned} F_2(z_1, z_2) &= r_{21}z_1^3 + r_{22}z_1^2 z_2 + r_{23}z_1 z_2^2 + r_{24}z_2^3 \\ &\quad + p_{21}z_1^2 + p_{22}z_1 z_2 + p_{23}z_2^2\,, \end{aligned} \tag{60}$$

and

$$r_{ij} = n_{i2}\ell_{2j} + n_{i3}\ell_{3j}\,, \quad i = 1,2\,, \quad j = 1,\ldots,4\,, \tag{61}$$

$$p_{ij} = n_{i2}\ell_{2k} + n_{i3}\ell_{3k}\,, \quad i = 1,2\,, \quad j = 1,2,3\,, \quad k = j+4\,. \tag{62}$$

If we introduce polar coordinates in the form,

$$z_1 = R\cos\phi\,, \quad z_2 = R\sin\phi\,, \tag{63}$$

we can use (57) and (58) to produce an equation describing the rate of change of the radial coordinate R,

$$\dot{R} = \alpha'\epsilon R + P(\phi)R^3 + Q(\phi)R^2\,. \tag{64}$$

This equation contains one variable, R, which is slowly varying in time, and another variable, ϕ, which is a fast variable. Therefore, equation (64) can be averaged over one complete cycle in ϕ to produce an equation with constant coefficients and similar stability properties,

$$\dot{R} = \alpha' \epsilon R + K R^3 + L R^2 \,, \tag{65}$$

where

$$K = \frac{1}{2\pi} \int_0^{2\pi} P(\phi)\, d\phi = \tfrac{1}{8}(3r_{11} + r_{13} + r_{22} + 3r_{24}) \,, \tag{66}$$

$$L = \frac{1}{2\pi} \int_0^{2\pi} Q(\phi)\, d\phi = 0 \,. \tag{67}$$

Therefore, the averaged equation (65) becomes

$$\dot{R} = \alpha' \epsilon R + K R^3 \,. \tag{68}$$

F. LIMIT CYCLE ANALYSIS

Equation (68) admits two steady state solutions, one at $R = 0$ which corresponds to the trivial equilibrium solution at zero, and one at

$$R_0 = \sqrt{-\frac{\alpha'}{K}\epsilon} \,. \tag{69}$$

This equilibrium solution corresponds to a periodic solution or limit cycle in the cartesian coordinates z_1, z_2. For this limit cycle to exist, the quantity R_0 must be a real number. In our case α' is always positive, since the system loses its stability; i.e., the real part of the critical pair of eigenvalues changes

from negative to positive, for increasing u. Therefore, existence of these periodic solutions depends on the value of K. Specifically,

- if $K < 0$, periodic solutions exist for $\epsilon > 0$ or $u > u_c$, and
- if $K > 0$, periodic solutions exist for $\epsilon < 0$ or $u < u_c$.

The characteristic root of (68) in the vicinity of (69) is

$$\beta = -2\alpha'\epsilon\,, \tag{70}$$

and we can see that

- if periodic solutions exist for $u > u_c$ they are stable, and
- if periodic solutions exist for $u < u_c$ they are unstable.

The period of these periodic solutions can be estimated as follows. Equations (57), (58), and (63) produce an equation in $\dot{\phi}$ similar to (64),

$$\dot{\phi} = \omega_0 + \omega'\epsilon + F(\phi)R^2 + G(\phi)R\,. \tag{71}$$

The averaged form of (71) is

$$\dot{\Phi} = \omega_0 + \omega'\epsilon + MR^2\,, \tag{72}$$

where

$$M = \tfrac{1}{2\pi}\int_0^{2\pi} F(\phi)\,d\phi = \tfrac{1}{8}(3r_{11} + r_{23} - r_{12} - 3r_{14})\,. \tag{73}$$

The limit cycle period can be computed by substituting (69) into (72),

$$T = \frac{2\pi}{\omega_0 + \omega'\epsilon + MR_0^2} = \frac{2\pi}{\omega_0}\left(1 - \frac{\omega'K - \alpha' M}{\omega_0 K}\epsilon\right) + O\left(\epsilon^2\right)\,. \tag{74}$$

G. RESULTS AND DISCUSSION

Typical results of the nonlinear stability coefficient K are shown in Figures 14 and 15. Figure 14 presents a plot of $K \cdot \gamma$ versus x_G for $z_G = 0.015$ and for different values of C_D. It should be emphasized that the use of $K \cdot \gamma$ is more meaningful than the use of K, since it properly accounts for the use of the regularization parameter γ as seen from equations (35) and (68). Numerical evidence demonstrates that all curves $K \cdot \gamma$ versus x_G converge for $\gamma \to 0$. For practical purposes, values of γ smaller than 0.001 produce identical results. The results of Figure 14 demonstrate the profound effect that the quadratic drag coefficient C_D has on stability of limit cycles. All Hopf bifurcations are supercritical ($K < 0$), and they become stronger supercritical as C_D is increased. It is worth noting that results for $C_D = 0$ produce subcritical behavior, $K > 0$, which is clearly incorrect. Thus, neglecting the effects of C_D would have produce entirely wrong results in the present problem. Figure 15 shows a plot of $K \cdot \gamma$ versus x_G for $C_D = 0.5$ and different values of the metacentric height z_G. It can be seen that, the bifurcations become stronger supercritical as initial stability z_G is increased.

The bifurcation analysis results are verified by direct numerical simulations shown in Figures 16 through 18. Figure 16 shows the results of two numerical simulations for two values of nominal speed u_0 in terms of the vehicle pitch angle θ (in degrees) versus time (in seconds). The critical value of speed, u_c, is about 0.495 as can be seen from Figure 8, while the other parameters in the simulations were $C_D = 0.5$, $z_G = 0.015$, and $x_G = 0$. It

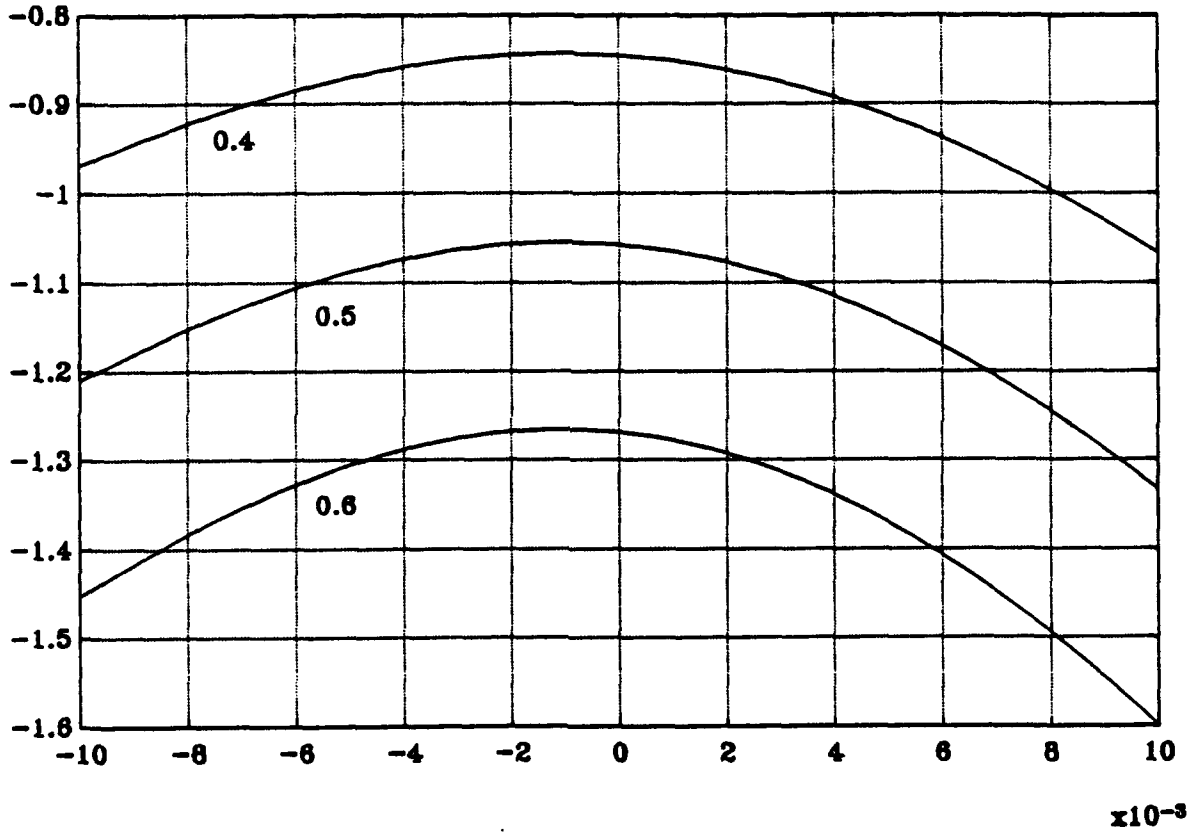

Figure 14: $K \cdot \gamma$ versus x_G for $z_G = 0.015$ and different values of the drag coefficient C_D

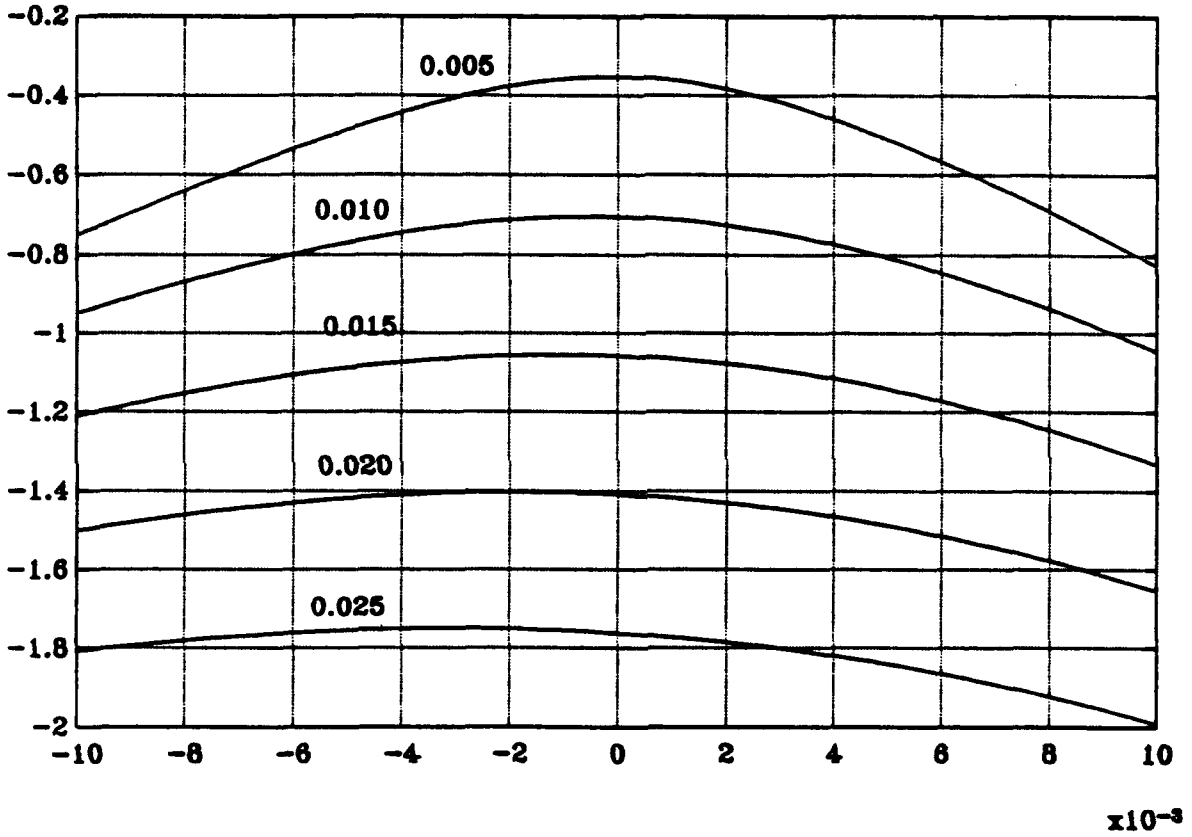

Figure 15: $K \cdot \gamma$ **versus** x_G **for** $C_D = 0.5$ **and different values of the metacentric height** z_G

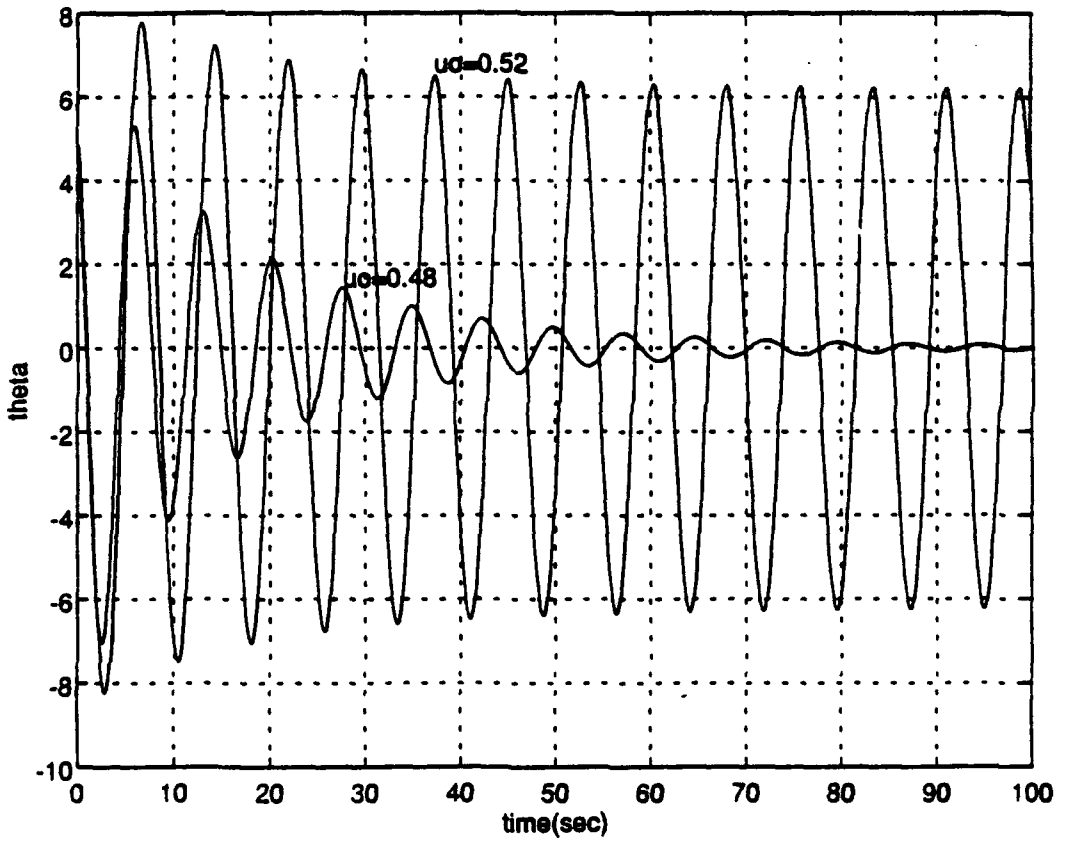

Figure 16: Time histories (θ, t) for $C_D = 0.5$, $z_G = 0.015$, $x_G = 0$, and two different values of nominal speed

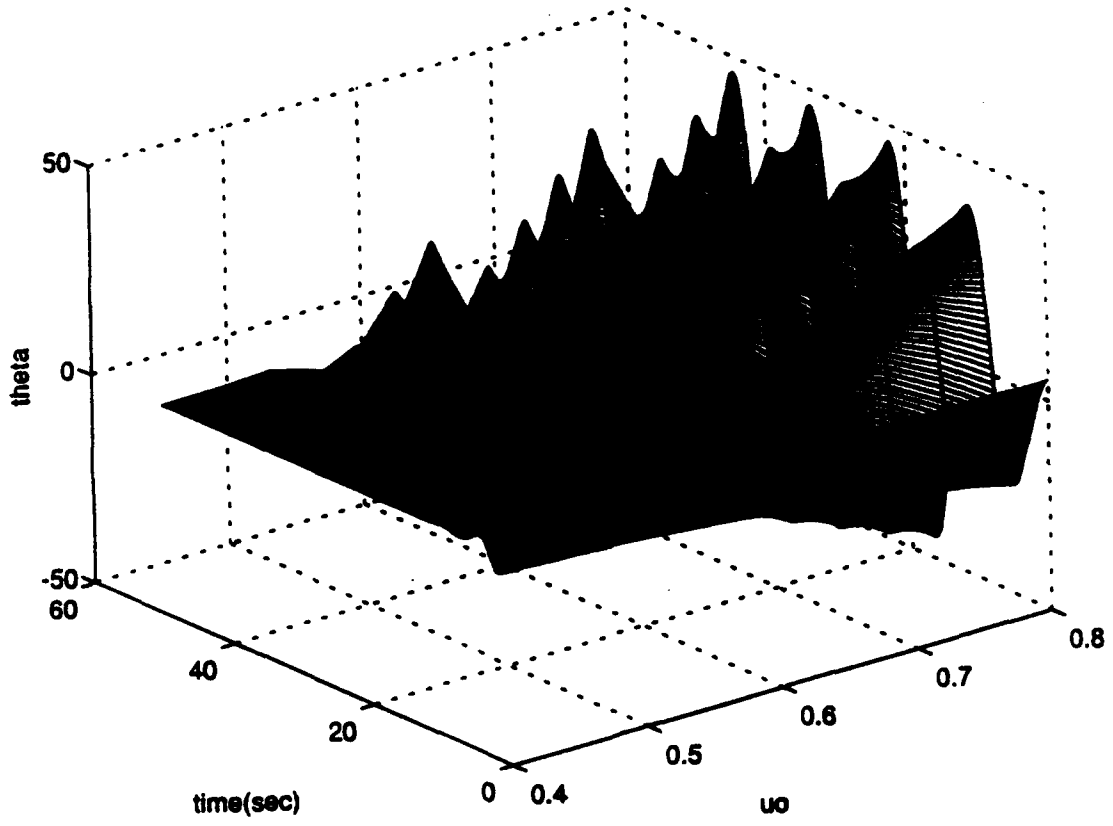

Figure 17: Time histories (θ, t) for $C_D = 0.5$, $x_G = 0.015$, $z_G = 0$, and a range of nominal forward speeds

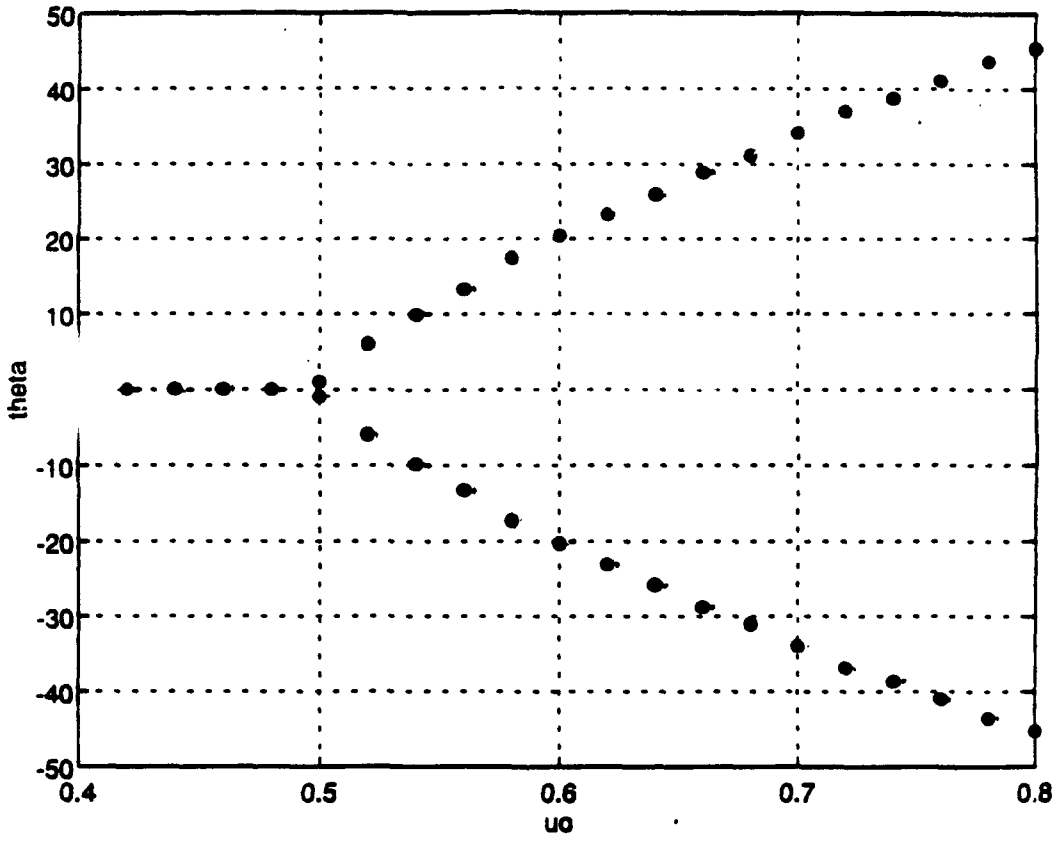

Figure 18: Limit cycle amplitudes from the results of Figure 17

can be seen that convergence to zero is ensured for $u_0 < u_c$ and convergence to a limit cycle occurs for $u_0 > u_c$. This indicates supercritical behavior as shown before. A selection of time histories is shown in Figure 17 for a range of forward speeds and the same parameters as in Figure 16. The same initial disturbance, $\theta = 5$ degrees, was introduced at $t = 0$ for all simulations. It can be seen that the amplitude of limit cycles increases as the distance of u from u_c is increasing. The rate of convergence of solutions to their limit cycles is also increasing, while their period remains essentially constant. These results are summarized in Figure 18, where the amplitudes of the numerically computed limit cycles are plotted versus u_0. The behavior is clearly supercritical, which agrees with our findings of the bifurcation analysis.

IV. BIAS EFFECTS

A. LOSS OF STABILITY

Stability analysis of motions at a nonsero angle of attack can be performed by first introducing some bias into the steady state solution and its perturbations. This can be achieved by maintaining a nonsero dive plane angle at nominal. In this case the steady state solutions are $q_0 = 0$, and w_0, θ_0 are computed from,

$$Z_w w_0 - C_D E_0 w_0 |w_0| + Z_\delta \delta = 0 \,, \tag{75}$$

$$M_w w_0 + C_D E_1 w_0 |w_0| - W(x_{GB} \cos\theta_0 + z_{GB} \sin\theta_0) + M_\delta \delta = 0 \,. \tag{76}$$

The coefficients E_0, E_1 are computed from (39). In order to solve (75) we observe that when $\delta > 0$, then $w_0 < 0$,

$$w_0 = \frac{-Z_w - \sqrt{Z_w^2 - 4C_D E_0 Z_\delta \delta}}{-2C_D E_0} \,, \tag{77}$$

and when $\delta < 0$, then $w_0 > 0$,

$$w_0 = \frac{-Z_w - \sqrt{Z_w^2 + 4C_D E_0 Z_\delta \delta}}{-2C_D E_0} \,. \tag{78}$$

The angle θ_0 can then be computed from equation (76).

Linearization of the equations of motion in the neighborhood of the above equilibrium point produces the linear system,

$$(m - Z_{\dot w})\dot w - (mx_G + Z_{\dot q})\dot q = (Z_w - 2C_D E_0 |w_0|)w$$

$$+(Z_q + m + 2C_D E_1 |w_0|)q \ , \tag{79}$$

$$-(M_{\dot{w}} + mx_G)\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_w + 2C_D E_1 |w_0|)w$$
$$+(M_q - mx_G - mz_G w_0 - 2C_D E_2 |w_0|)q$$
$$+W(x_{GB} \sin\theta_0 - z_{GB} \cos\theta_0)\theta \ , \tag{80}$$

$$\dot{\theta} = q \ , \tag{81}$$

where the variables w, θ are understood as small deviations from their equilibrium values. Numerical solution of the generalised eigenvalue problem (79) through (81) yields the critical speed values where the nominal equilibrium solution becomes unstable.

B. ANALYSIS OF HOPF BIFURCATIONS

It can be numerically verified that the above calculations for the new critical speed result in a loss of stability in the form of Hopf bifurcations, as for the $\delta = 0$ case. These Hopf bifurcations can be analyzed using the same general methodology that was developed in the previous chapter. The nonlinear expansions are like equations (21) to (23) with the following changes: The substitutions

$$\begin{aligned}
Z_w &\rightarrow Z_w - 2C_D E_0 w_0 \ , \\
M_w &\rightarrow M_w + 2C_D E_1 w_0 \ , \\
Z_q + m &\rightarrow Z_q + m + 2C_D E_1 w_0 \ , \\
M_q - mx_G &\rightarrow M_q - mx_G - mz_G w_0 - 2C_D E_2 w_0 \ ,
\end{aligned}$$

are assumed in the definition of coefficients a_{ij}. Furthermore, equation (33) prduces 2nd order contributions due to $w_0 \neq 0$, as well as 3rd order. Using (33) we can compute the 2nd order expansions of $d_w^{(2)}$ and $d_q^{(2)}$ in (29) using,

$$I_w^{(2)} = C_D(E_0 w^2 - 2E_1 wq + E_2 q^2)\,, \tag{82}$$

$$I_q^{(2)} = C_D(E_1 w^2 - 2E_2 wq + E_3 q^2)\,. \tag{83}$$

These equations are valid for $w_0 > 0$ or $\delta < 0$. For $\delta > 0$ the signs of E_i must be switched. The third order expansions $I_w^{(3)}$ and $I_q^{(3)}$ are the same as in equations (37) and (38). Using these additional terms, the nonlinear stability coefficient K can be computed in the same way as in the previous chapter.

C. RESULTS AND DISCUSSION

Typical results for $\delta \neq 0$ are shown in Figures 19 through 24. Figure 19 shows the equilibrium pitch angle θ_0 (in degrees) versus x_G for different values of δ from -5 to 5 degrees with increments of 1 degree. Solid curves correspond to positive δ and dashed curves to negative. Figure 20 shows the degree of stability for the equilibrium points of Figure 19, while Figure 21 presents the degree of stability in a three dimensional view. It can be seen that positive and negative values of δ have almost identical stability characteristics. Furthermore, the degree of stability becomes more negative as the absolute value of δ is increased, which means that we expect a wider domain of stability in this case. This is verified by the critical speed plots shown in

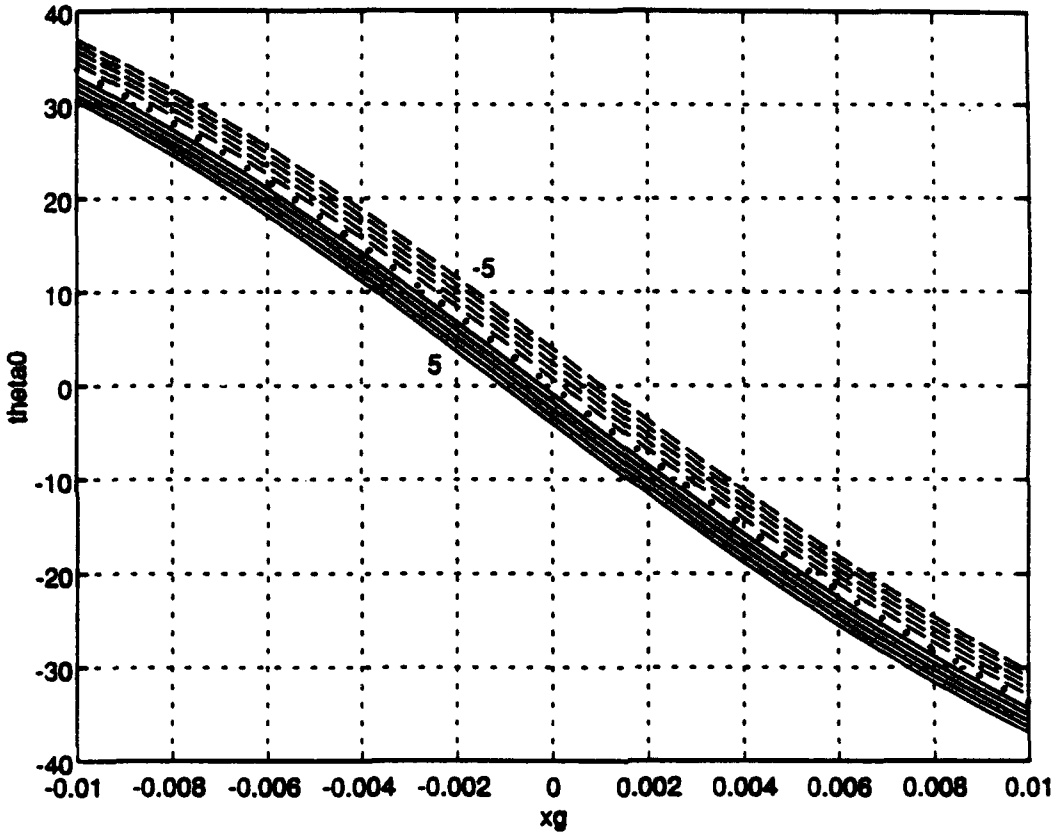

Figure 19: Pitch angle θ_0 versus x_G for $z_G = 0.015$, $u_0 = 0.5$, and different values of δ

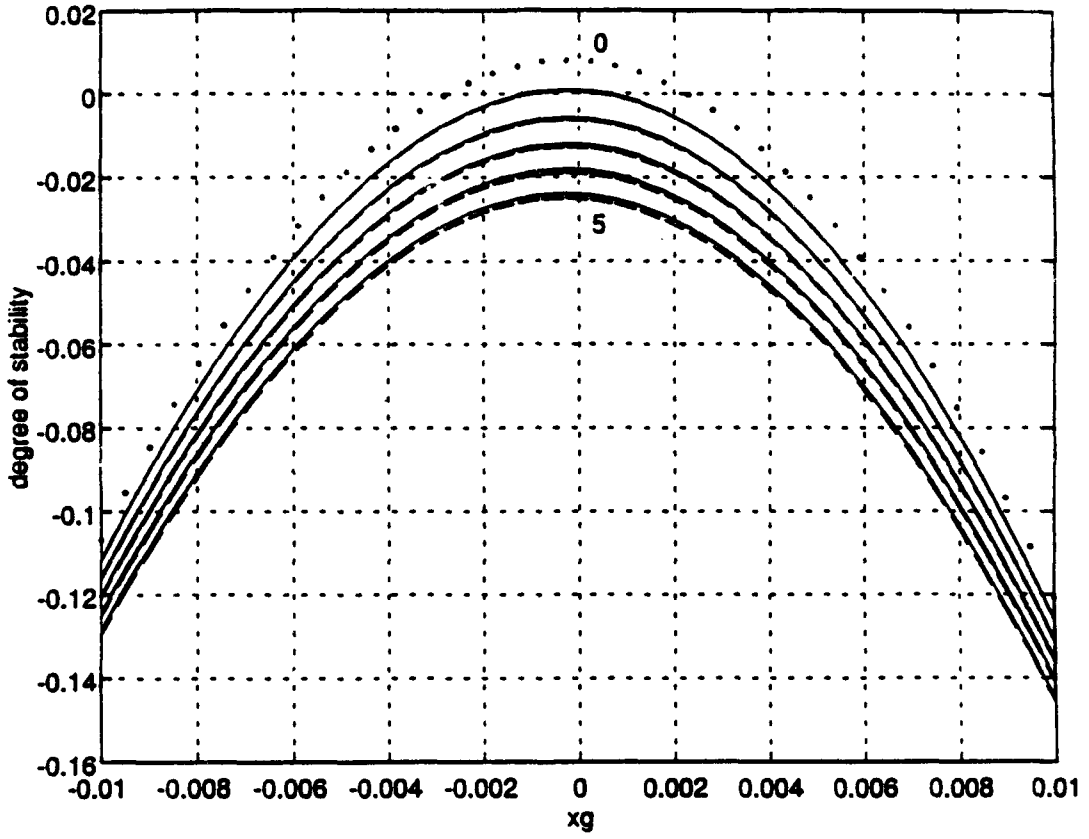

Figure 20: Degree of stability versus x_G for $z_G = 0.015$, $u_0 = 0.5$, and different values of δ

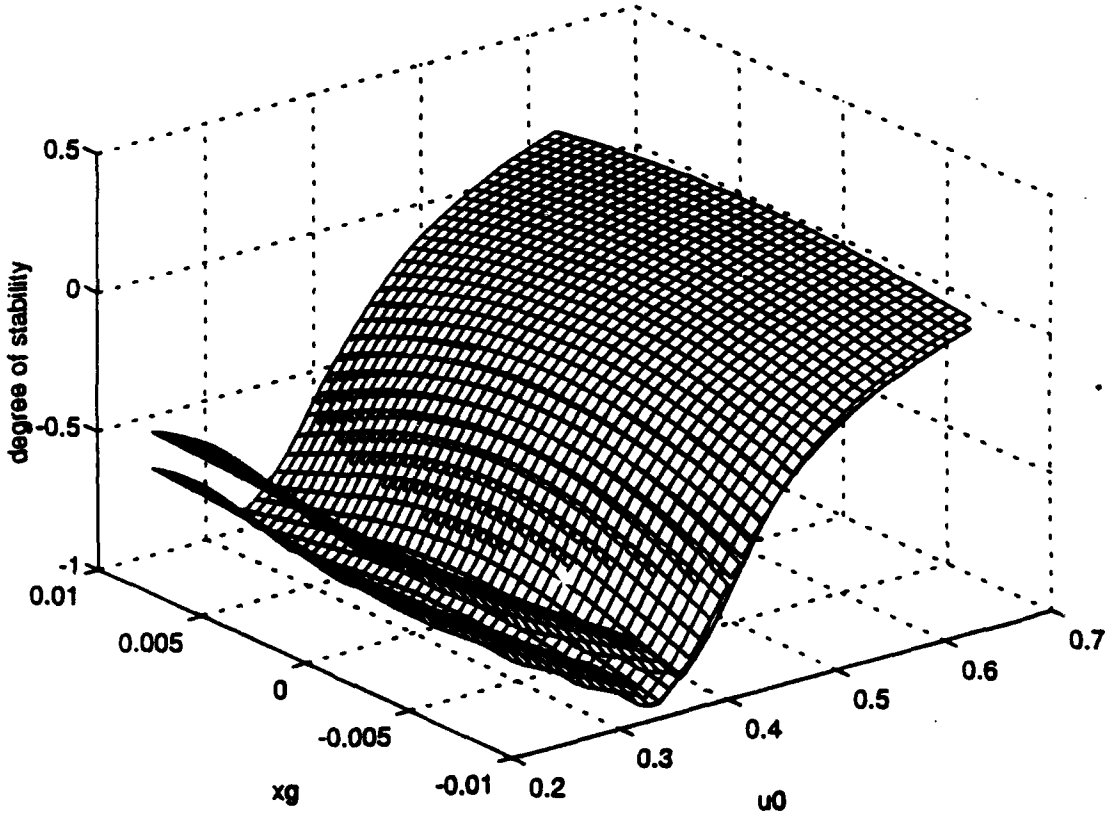

Figure 21: Degree of stability versus x_G and u_0 for $z_G = 0.015$ and two values of δ

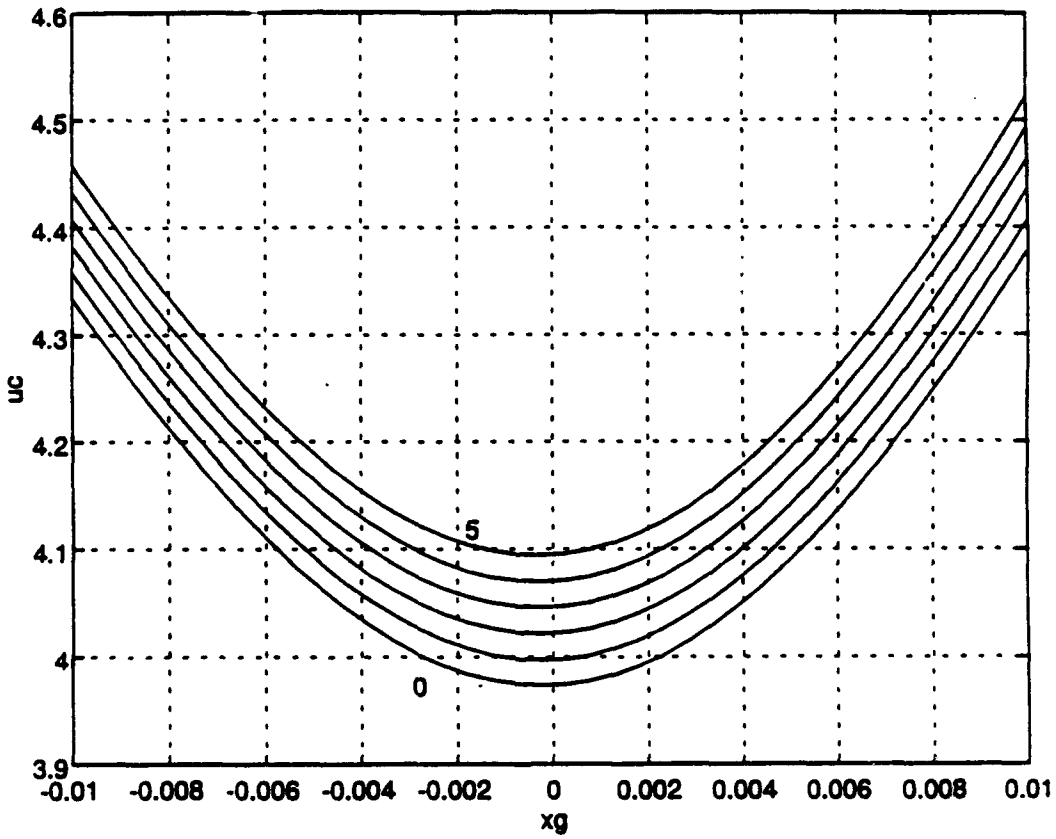

Figure 22: Critical speed u_c versus x_G for $z_G = 0.015$ and different values of δ

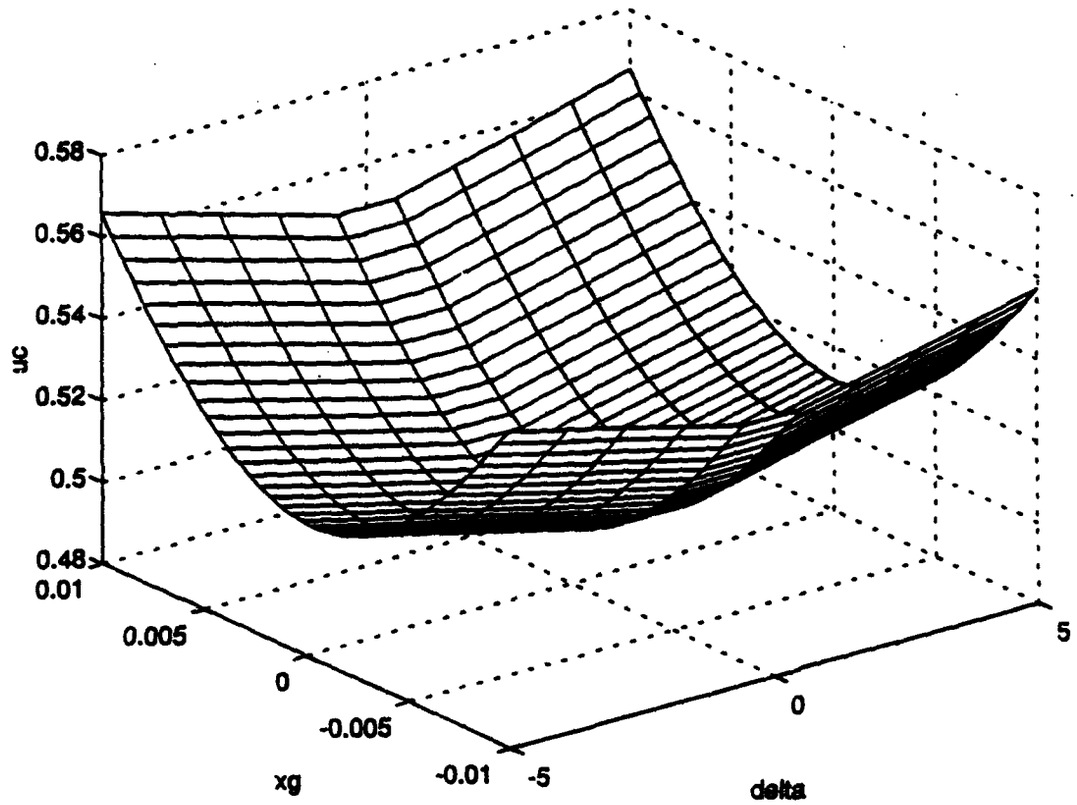

Figure 23: Critical speed u_c versus x_G and δ for $x_G = 0.015$

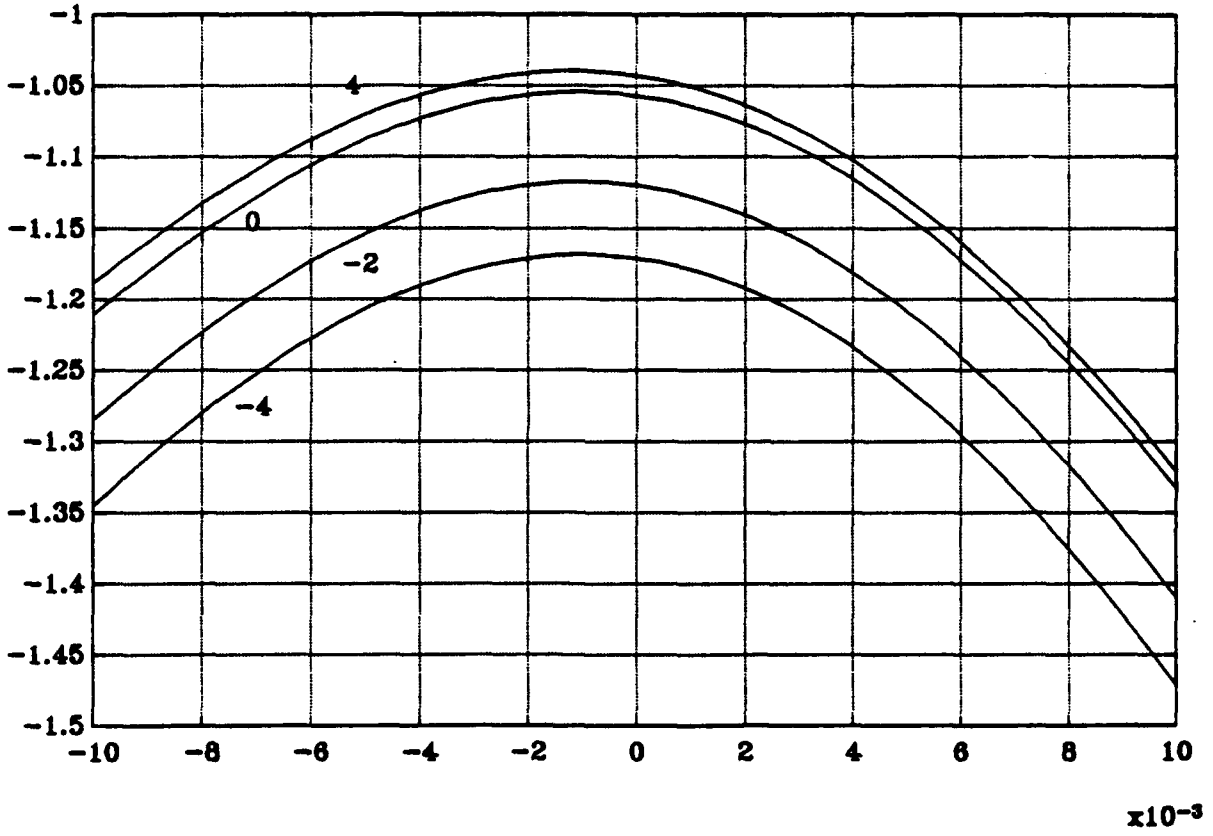

Figure 24: Nonlinear stability coefficient K versus x_G for $C_D = 0.5$, $z_G = 0.015$, and different values of δ

Figures 22 and 23. The critical speed is minimum for $\delta = 0$ and it increases monotonically with increasing absolute value of δ. This stabilizing effect of asymmetry (bias) remains approximately true for the nonlinear analysis, as demonstrated by the results of Figure 24. It can be seen that the nonlinear stability coefficient K becomes more negative as δ is decreasing from zero. For increasing δ, K becomes less negative but the difference from the $\delta = 0$ calculations appears to be very small. Therefore, limit cycle stability is not significantly affected by the bias effects that are induced by small nonzero dive plane angles.

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis presented a comprehensive nonlinear study of straight line stability of motion of submersibles in the dive plane under open loop conditions. A systematic perturbation analysis demonstrated that the effects of surge in heave/pitch are small and can be neglected. Primary loss of stability was shown to occur in the form of Hopf bifurcations to periodic solutions. The critical speed were instability occurs was computed in terms of metacentric height, longitudinal separation of the centers of buoyancy and gravity, and the dive plane angle. Analysis of the periodic solutions that resulted from the Hopf bifurcations was accomplished through Taylor expansions, up to third order, of the equations of motion. A consistent approximation, utilizing the generalized gradient, was used to study the non-analytic quadratic cross flow integral drag terms. The results indicated that loss of stability occurs always in the form of supercritical Hopf bifurcations with stable limit cycles. It was shown that this is mainly due to the stabilizing effect of the drag forces at high angles of attack. As a recommendation for further research, we suggest a nonlinear analysis of coupled motions in six degrees of freedom.

APPENDIX

The following is a list and description of the computer programs used in this thesis. The programs are written in FORTRAN or MATLAB. Complete printouts of the programs follow after the list.

- **PERTURB.M**

  MATLAB program for performing surge/heave/pitch perturbation analysis.

- **CRIT_0.M**

  MATLAB program for calculating the critical speed for $\delta = 0$.

- **CRIT_DELTA.M**

  MATLAB program for calculating the critical speed for $\delta \neq 0$.

- **HOPF_0.FOR**

  FORTRAN program for calculating the nonlinear stability coefficient K for $\delta = 0$.

- **HOPF_DELTA.FOR**

  FORTRAN program for calculating the nonlinear stability coefficient K for $\delta \neq 0$.

- **SIM.FOR**

  FORTRAN program for simulating the equations of motion.

```
% Program perturb.m
% This file applies concepts of perturbation analysis theory,
% in order to prove that the 3 by 3 system describing motion
% in the vertical plane, decouples from the 4 by 4 one.

rho = 1.94;
g   = 32.2;
L   = 13.9792;
nd1 = 0.5*rho*L^2;
nd2 = 0.5*rho*L^3;
nd3 = 0.5*rho*L^4;
nd4 = 0.5*rho*L^5;
iy  = 561.32/nd4;
zqdot = -6.33e-4;
zwdot = -1.4529e-2;
zq    =  7.545e-3;
zw    = -1.391e-2;
mqdot = -8.8e-4;
mwdot = -5.61e-4;
mq    = -3.702e-3;
mw    =  1.0324e-2;
cd    =  0.015;
zb    =  0/L;
m     =  1556.2363/(g*nd2);
xudot = -0.05*m;
xb    =  0/L;
uo    =  4;
w     =  1556.2363./(nd1.*uo.^2);
b     =  w;

xg    =  0;
zg    =  linspace(0.001,0.025,500);

xgb = xg-xb;
zgb = zg-zb;
theta = atan(-xgb./zgb);

alpha = m-zwdot;
beta  = -zw;
A1 = m-xudot;
B1 = -2*cd;
```

```
A  = (m-zwdot)*(iy-mqdot)-(m*xg+zqdot)*(mwdot+m*xg);
B  = -zw*(iy-mqdot)-(m-zwdot)*(mq-m*xg)-(zq+m)*(mwdot+m*xg)...
     -(m*xg+zqdot)*mw;

for i = 1:length(zg)
  deltax(i) = xgb*w*sin(theta(i))-zgb(i)*w*cos(theta(i));
  C(i)  = -deltax(i)*(m-zwdot)+zw*(mq-m*xg)-(zq+m)*mw;
  D(i)  = deltax(i)*zw;
  A2(i) = (m*zg(i))^2*(m-zwdot);
  B2(i) = -(m*zg(i))^2*zw;

  CA = -A1*A;
  CB = -A1*B+B1*A;
  CC = -A1*C(i)+B1*B;
  CD = -A1*D(i)+B1*C(i);
  CE =  B1*D(i);

  p  = [CA CB CC CD CE];
  r(1:4,i)  = roots(p);

  lamda(1:4,i) = r(1:4,i)-(r(1:4,i).^3.*(alpha.*r(1:4,i)+beta))...
                 .*(m.*zg(i)).^2./(4.*CA.*r(1:4,i).^3+...
                 3.*CB.*r(1:4,i).^2+2.*CC.*r(1:4,i)+CD);

  PA = -A1*A+A2(i);
  PB = -A1*B+B1*A+B2(i);
  PC = -A1*C(i)+B1*B;
  PD = -A1*D(i)+B1*C(i);
  PE =  B1*D(i);

  p4  = [PA PB PC PD PE];     % e-values of 4 by 4 system.
  r4(1:4,i) = roots(p4);
end
```

```
% Program crit_0.m
% Evaluation of critical speed for delta=0

rho = 1.94;
g   = 32.2;
L   = 13.9792;
nd1 = 0.5*rho*L^2;
nd2 = 0.5*rho*L^3;
nd3 = 0.5*rho*L^4;
nd4 = 0.5*rho*L^5;
iy  = 561.32/nd4;
zqdot = -6.33e-4;
zwdot = -1.4529e-2;
zq    =  7.545e-3;
zw    = -1.391e-2;
mqdot = -8.8e-4;
mwdot = -5.61e-4;
mq    = -3.702e-3;
mw    =  1.0324e-2;
cd    =  0.015;
zb    =  0/L;
m     =  1556.2363/(g*nd2);
xudot = -0.05*m;
xb    =  0/L;

xg = linspace(-0.01,0.01,40);
zg = linspace(0.005,0.025,40);

Gv = 1 - mw.*(zq+m)./(zw.*(mq-m.*xg));
lambda = -2*cd/m-xudot;   %  lambda = -1.6439

xgb = xg-xb;
zgb = zg-zb;

for j=1:length(zg)
for i=1:length(xg)

theta(i,j) = atan(-xgb(i)/zgb(j));
a0 =  (m-zwdot)*(iy-mqdot)-(mwdot+m*xg(i))*(zqdot+m*xg(i));
b0 =  (-zwdot*m-m*mw-zq*m)*xg(i)+(-m*mq+zwdot*mq-zqdot*mw...
      -zq*mwdot-m*mwdot-iy*zw+mqdot*zw);
```

```
c0 = -m*zw*xg(i)+mq*zw-zq*mw-m*mw;
c1 =  (-m*xg(i)+zwdot*xg(i)+m*xb-zwdot*xb)*sin(theta(i,j))...
     +(-m*zb-zwdot*zg(j)+zwdot*zb+m*zg(j))*cos(theta(i,j));
d1 =  (zw*xg(i)-zw*xb)*sin(theta(i,j))+...
      (zw*zb-zw*zg(j))*cos(theta(i,j));

% After applying AD-BC...

w(i,j) = b0*c0/(a0*d1-b0*c1);
u0(i,j)     = (1556.2363/(nd1*w(i,j)))^.5;

end
end
```

```
% File crit_delta.m
% Evaluation of critical speed for nonzero stern plane angle.

rho = 1.94;
g   = 32.2;
L   = 13.9792;
nd1 = 0.5*rho*L^2;
nd2 = 0.5*rho*L^3;
nd3 = 0.5*rho*L^4;
nd4 = 0.5*rho*L^5;
iy  = 561.32/nd4;
zqdot = -6.33e-4;
zwdot = -1.4529e-2;
zq    =  7.545e-3;
zw    = -1.391e-2;
mqdot = -8.8e-4;
mwdot = -5.61e-4;
mq    = -3.702e-3;
mw    =  1.0324e-2;
m     =  1556.2363/(g*nd2);
xudot = -0.05*m;
xb    =  0/L;
zb    =  0/L;
zg    =  0.015;

cdz   =  0.5;              % if cdz=0, division by zero provides NaN...
zd    = -5.603e-03;
md    = -2.409e-03;
E0    =  0.1036509;        % E0 == Integral of b(x)*dx
E1    = -2.3629982e-03;    % E1 == Integral of x*b(x)*dx
E2    =  7.3672147e-03;    % E2 == Integral of x^2*b(x)*dx

xg = linspace(-0.01,0.01,80);    % NON-DIMENSIONAL
u0 = linspace(2,5,40);

xgb = xg-xb;
zgb = zg-zb;
w1  = 1556.2363./(nd1.*u0.^2);

delta = input('Enter the value of delta: delta = linspace( , , )');
delta = delta*pi/180;
```

```
for k=1:length(delta)
for j=1:length(u0)
for i=1:length(xg)

if delta(k) > 0
 w0 = -(zw+sqrt(zw^2-4*cdz*E0*zd*delta(k)))/(2*cdz*E0);
 coeff1 = (mw*w0-cdz*w0^2*E1+md*delta(k))/w1(j);
 poly1  = [(xgb(i)^2+zgb^2) (-2*coeff1*zgb) (coeff1^2-xgb(i)^2)];
 theta0 = asin(roots(poly1));
 check1 = xgb(i)*cos(theta0(1))+zgb*sin(theta0(1));
 check2 = xgb(i)*cos(theta0(2))+zgb*sin(theta0(2));
 if abs(check2-coeff1) < abs(check1-coeff1)
  theta0 = theta0(2);
 else
  theta0 = theta0(1);
 end

   B=[ (m-zwdot)    -(m*xg(i)+zqdot) 0
      -(mwdot+m*xg(i))  iy-mqdot     0
       0             0            1];

   A=[(zw+2*cdz*E0*w0)      (zq+m)-2*cdz*E1*w0          0
      mw-2*cdz*E1*w0  (mq-m*xg(i))-m*zg*w0+2*cdz*E2*w0 ...
            (xgb(i)*sin(theta0)-zgb*cos(theta0))*w1(j)
      0                        1                       0];
elseif delta(k) < 0
 w0 = -(zw+sqrt(zw^2+4*cdz*E0*zd*delta(k)))/(-2*cdz*E0);
 coeff1 = (mw*w0+cdz*w0^2*E1+md*delta(k))/w1(j);
 poly1  = [(xgb(i)^2+zgb^2) (-2*coeff1*zgb) (coeff1^2-xgb(i)^2)];
 theta0 = asin(roots(poly1));
 check1 = xgb(i)*cos(theta0(1))+zgb*sin(theta0(1));
 check2 = xgb(i)*cos(theta0(2))+zgb*sin(theta0(2));
 if abs(check2-coeff1) < abs(check1-coeff1)
  theta0 = theta0(2);
 else
  theta0 = theta0(1);
 end

   B=[ (m-zwdot)    -(m*xg(i)+zqdot) 0
      -(mwdot+m*xg(i))  iy-mqdot     0
       0             0            1];
```

```
   A=[(zw-2*cdz*E0*w0)       (zq+m)+2*cdz*E1*w0          0
      mw+2*cdz*E1*w0  (mq-m*xg(i))-m*zg*w0-2*cdz*E2*w0 ...
                (xgb(i)*sin(theta0)-zgb*cos(theta0))*w1(j)
      0                          1                      0];
end

evals = eig(A,B);
degstab(i,j) = max(real(evals));

end
end

% Evaluation of the critical speed, ucrit:

for i=1:length(xg)
for j=1:(length(u0))-1
  flag1 = sign(degstab(i,j));
  flag2 = sign(degstab(i,j+1));
  if flag1 ~= flag2
     ucrit(i,k) = (u0(j+1)*degstab(i,j)-u0(j)*degstab(i,j+i))/...
                  (degstab(i,j)-degstab(i,j+1));
  end
end
end
end

% Revaluation of the nominal angle, 'thetacr'
% at the critical speed, 'ucrit':
% Note that in this case, at a certain 'xg',
% corresponds a specific 'ucrit'.
% That's why we use the same index 'i':

w2  = 1556.2363./(nd1.*ucrit.^2);
for i=1:length(xg)

if delta(1) > 0
 w0 = -(zw+sqrt(zw^2-4*cdz*E0*zd*delta(1)))/(2*cdz*E0);
 coeff2 = (mw*w0-cdz*w0^2*E1+md*delta(1))/w2(i);
 poly2  = [(xgb(i)^2+zgb^2) (-2*coeff2*zgb) (coeff2^2-xgb(i)^2)];
 theta0 = asin(roots(poly2));
 check1 = xgb(i)*cos(theta0(1))+zgb*sin(theta0(1));
```

```
 check2 = xgb(i)*cos(theta0(2))+zgb*sin(theta0(2));
 if abs(check2-coeff2) < abs(check1-coeff2)
  thetacr(i) = theta0(2);
 else
  thetacr(i) = theta0(1);
 end
elseif delta(1) < 0
 w0 = -(zw+sqrt(zw^2+4*cdz*E0*zd*delta(1)))/(-2*cdz*E0);
 coeff2 = (mw*w0+cdz*w0^2*E1+md*delta(1))/w2(i);
 poly2  = [(xgb(i)^2+zgb^2) (-2*coeff2*zgb) (coeff2^2-xgb(i)^2)];
 theta0 = asin(roots(poly2));
 check1 = xgb(i)*cos(theta0(1))+zgb*sin(theta0(1));
 check2 = xgb(i)*cos(theta0(2))+zgb*sin(theta0(2));
 if abs(check2-coeff2) < abs(check1-coeff2)
  thetacr(i) = theta0(2);
 else
  thetacr(i) = theta0(1);
 end
end

end

results = [xg' ucrit(:,1) w0*ones(length(xg),1) thetacr']
save ucrit0.dat results -ascii
```

```
C     PROGRAM HOPF_0.FOR
C     EVALUATION OF HOPF BIFURCATION FORMULAS
C     ZERO DIVE PLANE ANGLE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION L,IY,MASS,MQDOT,MWDOT,ND1,THETA,
     1                 MQ,MW,MD,MDS,MDB,K1,K2,
     2                 ALFA,BETA,GAMA,DELTA,
     3                 E0,E1,E2,E3,E4,
     4                 DW1,DW2,DW3,DW4,
     5                 DQ1,DQ2,DQ3,DQ4
      DOUBLE PRECISION M11,M12,M13,M21,M22,M23,
     1                 M31,M32,M33,
     2                 N11,N12,N13,N21,N22,N23,
     3                 N31,N32,N33,
     4                 L21,L22,L23,L24,L31,L32,L33,L34,
     5                 L25,L26,L27,L35,L36,L37,
     6                 L21A,L22A,L23A,L24A,L31A,
     7                 L32A,L33A,L34A
C
      DIMENSION A(3,3),T(3,3),TINV(3,3),FV1(3),IV1(3),YYY(3,3)
      DIMENSION WR(3),WI(3),TSAVE(3,3),TLUD(3,3),IVLUD(3),SVLUD(3)
      DIMENSION ASAVE(3,3),A1(3,3),A2(3,3),XL(25),BR(25)
      DIMENSION VEC0(25),VEC1(25),VEC2(25),VEC3(25),VEC4(25)
C
      OPEN (20,FILE='HOPF.RES',STATUS='NEW')
C
      WEIGHT=1556.2363
      IY    = 561.32
      L     =  13.9792
      RHO   =   1.94
      WRITE (*,*) ' ENTER CD'
      READ  (*,*) CD
      CD    = 0.5*CD*RHO
      G     =  32.2
      XB    =   0.0
      WRITE (*,*) ' ENTER ZG/L'
      READ  (*,*) ZG
      ZG    =ZG*L
      ZB    =   0.0
      MASS  = WEIGHT/G
```

```
      BOY    = WEIGHT
      MD1    = 0.5*RHO*L**2
      ZQDOT =-6.3300E-04*0.5*RHO*L**4
      ZWDOT =-1.4529E-02*0.5*RHO*L**3
      ZQ     = 7.5450E-03*0.5*RHO*L**3
      ZW     =-1.3910E-02*0.5*RHO*L**2
      MQDOT =-8.8000E-04*0.5*RHO*L**5
      MWDOT =-5.6100E-04*0.5*RHO*L**4
      MQ     =-3.7020E-03*0.5*RHO*L**4
      MW     = 1.0324E-02*0.5*RHO*L**3
C
      ZGB=ZG-ZB
C
C     DEFINE THE LENGTH X AND BREADTH B TERMS FOR THE INTEGRATION
C
      XL( 1)=  0.0000
      XL( 2)=  0.1000
      XL( 3)=  0.2000
      XL( 4)=  0.3000
      XL( 5)=  0.4000
      XL( 6)=  0.5000
      XL( 7)=  0.6000
      XL( 8)=  0.7000
      XL( 9)=  1.0000
      XL(10)=  2.0000
      XL(11)=  3.0000
      XL(12)=  4.0000
      XL(13)=  7.7143
      XL(14)= 10.0000
      XL(15)= 15.1429
      XL(16)= 16.0000
      XL(17)= 17.0000
      XL(18)= 18.0000
      XL(19)= 19.0000
      XL(20)= 20.0000
      XL(21)= 20.1000
      XL(22)= 20.2000
      XL(23)= 20.3000
      XL(24)= 20.4000
      XL(25)= 20.4167

      DO 102 N = 1,25
```

```
        XL(N) = (L/20.)*XL(N) - L/2.
  102   CONTINUE

C
      BR( 1)= 0.000
      BR( 2)= 0.485
      BR( 3)= 0.658
      BR( 4)= 0.778
      BR( 5)= 0.871
      BR( 6)= 0.945
      BR( 7)= 1.010
      BR( 8)= 1.060
      BR( 9)= 1.180
      BR(10)= 1.410
      BR(11)= 1.570
      BR(12)= 1.660
      BR(13)= 1.670
      BR(14)= 1.670
      BR(15)= 1.670
      BR(16)= 1.630
      BR(17)= 1.370
      BR(18)= 0.919
      BR(19)= 0.448
      BR(20)= 0.195
      BR(21)= 0.188
      BR(22)= 0.168
      BR(23)= 0.132
      BR(24)= 0.053
      BR(25)= 0.000
C
      DO 104 K = 1,25
        VEC0(K)=BR(K)
        VEC1(K)=XL(K)*BR(K)
        VEC2(K)=XL(K)*XL(K)*BR(K)
        VEC3(K)=XL(K)*XL(K)*XL(K)*BR(K)
        VEC4(K)=XL(K)*XL(K)*XL(K)*XL(K)*BR(K)
  104 CONTINUE
      CALL TRAP(25,VEC0,XL,E0)
      CALL TRAP(25,VEC1,XL,E1)
      CALL TRAP(25,VEC2,XL,E2)
      CALL TRAP(25,VEC3,XL,E3)
      CALL TRAP(25,VEC4,XL,E4)
```

```
      WRITE (*,*) ' ENTER GAMMA'
      READ  (*,*) EPSILON
      XGMIN=-0.01
      XGMAX=+0.01
      IXG=80
      XGMIN=XGMIN*L
      XGMAX=XGMAX*L
C=====================================================================
      DO 1 IT=1,IXG
        WRITE (*,3001) IT,IXG
        XG=XGMIN+(XGMAX-XGMIN)*(IT-1)/(IXG-1)
        XGB=XG-XB
        DV=(MASS-ZWDOT)*(IY-MQDOT)-(MASS*XG+ZQDOT)*(MASS*XG+MWDOT)
        CD6=CD/(3.DO*EPSILON*DV)
        DW1=CD6*((IY-MQDOT)*(-E0)+(MASS*XG+ZQDOT)*E1)
        DW2=CD6*((IY-MQDOT)*(3*E1)-(MASS*XG+ZQDOT)*3*E2)
        DW3=CD6*((IY-MQDOT)*(-3*E2)+(MASS*XG+ZQDOT)*3*E3)
        DW4=CD6*((IY-MQDOT)*(E3)-(MASS*XG+ZQDOT)*E4)
        DQ1=CD6*((MASS-ZWDOT)*(E1)+(MASS*XG+MWDOT)*(-E0))
        DQ2=CD6*((MASS-ZWDOT)*(-3*E2)+(MASS*XG+MWDOT)*(3*E1))
        DQ3=CD6*((MASS-ZWDOT)*(3*E3)+(MASS*XG+MWDOT)*(-3*E2))
        DQ4=CD6*((MASS-ZWDOT)*(-E4)+(MASS*XG+MWDOT)*(E3))
        THETA0=ATAN(-XGB/ZGB)
        AA0=(MASS-ZWDOT)*(IY-MQDOT)-(MWDOT+MASS*XG)*(ZQDOT+MASS*XG)
        BB0=(-ZWDOT*MASS-MASS*MW-ZQ*MASS)*XG
     &     +(-MASS*MQ+ZWDOT*MQ-ZQDOT*MW
     &     -ZQ*MWDOT-MASS*MWDOT-IY*ZW+MQDOT*ZW)
        CC0=-MASS*ZW*XG+MQ*ZW-ZQ*MW-MASS*MW
        CC1=(-MASS*XG+ZWDOT*XG+MASS*XB-ZWDOT*XB)*SIN(THETA0)
     &     +(-MASS*ZB-ZWDOT*ZG+ZWDOT*ZB+MASS*ZG)*COS(THETA0)
        DD1=(ZW*XG-ZW*XB)*SIN(THETA0)+(ZW*ZB-ZW*ZG)
     &      *COS(THETA0)
C
        WEI=BB0*CC0/(AA0*DD1-BB0*CC1)
        U0=DSQRT(1556.2363/WEI)
        U=U0
C
C       DETERMINE [A] AND [B] COEFFICIENTS
C
        A11DV=(IY-MQDOT)*ZW+(MASS*XG+ZQDOT)*MW
        A12DV=(IY-MQDOT)*(MASS+ZQ)+(MASS*XG+ZQDOT)*(MQ-MASS*XG)
        A13DV=-(MASS*XG+ZQDOT)*WEIGHT
```

```
      A21DV=(MASS-ZWDOT)*MW+(MASS*XG+MWDOT)*ZW
      A22DV=(MASS-ZWDOT)*(MQ-MASS*XG)+(MASS*XG+MWDOT)*(MASS+ZQ)
      A23DV=-(MASS-ZWDOT)*WEIGHT
C
      A11=A11DV/DV
      A12=A12DV/DV
      A13=A13DV/DV
      A21=A21DV/DV
      A22=A22DV/DV
      A23=A23DV/DV
C
      C11DV=(IY-MQDOT)*MASS*ZG
      C12DV=-(MASS*XG+ZQDOT)*MASS*ZG
      C21DV=-(MASS-ZWDOT)*MASS*ZG
      C22DV=(MASS*XG+MWDOT)*MASS*ZG
C
      C11=C11DV/DV
      C12=C12DV/DV
      C21=C21DV/DV
      C22=C22DV/DV
C=====================================================================
C     EVALUATE TRANSFORMATION MATRIX OF EIGENVECTORS
C
      K1=-(XGB*SIN(THETA0)-ZGB*COS(THETA0))
      K2=-(1./6.)*(ZGB*COS(THETA0)-XGB*SIN(THETA0))
C
      A(1,1)=0.0
      A(1,2)=0.0
      A(1,3)=1.0
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
      DO 11 I=1,3
        DO 12 J=1,3
          ASAVE(I,J)=A(I,J)
12      CONTINUE
11    CONTINUE
      CALL RG(3,3,A,WR,WI,1,YYY,IV1,FV1,IERR)
      CALL DSOMEG(IEV,WR,WI,OMEGA,CHECK)
```

```
      OMEGAO=OMEGA
      DO 5 I=1,3
        T(I,1)= YYY(I,IEV)
        T(I,2)=-YYY(I,IEV+1)
 5    CONTINUE
      IF (IEV.EQ.1) GO TO 13
      IF (IEV.EQ.2) GO TO 14
      STOP 3004
14    DO 6 I=1,3
        T(I,3)=YYY(I,1)
 6    CONTINUE
      GO TO 17
13    DO 16 I=1,3
        T(I,3)=YYY(I,3)
16    CONTINUE
17    CONTINUE
C
C     NORMALIZATION OF THE CRITICAL EIGENVECTOR
C
      INORM=1
      IF (INORM.NE.0) CALL NORMAL(T)
C
C     INVERT TRANSFORMATION MATRIX
C
      DO 2 I=1,3
        DO 3 J=1,3
          TINV(I,J)=0.0
          TSAVE(I,J)=T(I,J)
3       CONTINUE
2     CONTINUE
      CALL DLUD(3,3,TSAVE,3,TLUD,IVLUD)
      DO 4 I=1,3
        IF (IVLUD(I).EQ.0) STOP 3003
4     CONTINUE
      CALL DILU(3,3,TLUD,IVLUD,SVLUD)
      DO 8 I=1,3
        DO 9 J=1,3
          TINV(I,J)=TLUD(I,J)
9       CONTINUE
8     CONTINUE
C
C     CHECK Inv(T)*A*T
```

```
C
      IMULT=1
      IF (IMULT.EQ.1) CALL MULT(TINV,ASAVE,T,A2)
      IF (IMULT.EQ.0) STOP
      P=A2(3,3)
      PEIG=P
C
C     DEFINITION OF Nij
C
      N11=TINV(1,1)
      N21=TINV(2,1)
      N31=TINV(3,1)
      N12=TINV(1,2)
      N22=TINV(2,2)
      N32=TINV(3,2)
      N13=TINV(1,3)
      N23=TINV(2,3)
      N33=TINV(3,3)
C
C     DEFINITION OF Mij
C
      M11=T(1,1)
      M21=T(2,1)
      M31=T(3,1)
      M12=T(1,2)
      M22=T(2,2)
      M32=T(3,2)
      M13=T(1,3)
      M23=T(2,3)
      M33=T(3,3)
C
C     DEFINITION OF Lij
C
      L25=C11*M31*M31+C12*M21*M31
      L26=2*C11*M31*M32+C12*(M21*M32+M22*M31)
      L27=C11*M32*M32+C12*M22*M33
      L35=C22*M31*M31+C21*M21*M31
      L36=2*C22*M31*M32+C21*(M21*M32+M22*M31)
      L37=C22*M32*M32+C21*M33*M22
C
C     DEFINITION OF ALFA, BETA, GAMA
C
```

```
      D1 =M32*L25 + M33*L35
      D2 =M32*L26 + M33*L36
      D3 =M32*L27 + M33*L37
C
      D11=-P
      D12=OMEGAO
      D21=-2*OMEGAO
      D22=-P
      D23=2*OMEGAO
      D32=-OMEGAO
      D33=-P
C
      BETA=(D2-D21*D1/D11-D23*D3/D33)/(D22-D21*D12/D11-D23*D32/D33)
      ALFA=(D1-D12*BETA)/D11
      GAMA=(D3-D32*BETA)/D33
C
      L21A=2*C11*ALFA*M31*M33+C12*ALFA*(M21*M33+M23*M31)
C
      L22A=2*C11*ALFA*M32*M33 + 2*C11*BETA*M31*M33
     &     + C12*ALFA*(M22*M33+M32*M23)
     &     + C12*BETA*(M21*M33+M23*M31)
C
      L23A=2*C11*GAMA*M31*M33+2*C11*BETA*M32*M33
     &     + C12*GAMA*(M21*M33+M23*M31)
     &     + C12*BETA*(M22*M33+M23*M32)
C
      L24A=2*C11*GAMA*M32*M33+C12*GAMA*(M22*M33+M23*M32)
C
      L31A=2*C22*ALFA*M31*M33+C21*ALFA*(M21*M33+M23*M31)
C
      L32A=2*C22*ALFA*M32*M33+2*C22*BETA*M31*M33
     &     + C21*ALFA*(M22*M33+M32*M23)
     &     + C21*BETA*(M21*M33+M23*M31)
C
      L33A=2*C22*GAMA*M31*M33+2*C22*BETA*M32*M33
     &     + C21*GAMA*(M21*M33+M23*M31)
     &     + C21*BETA*(M22*M33+M23*M32)
C
      L34A=2*C22*GAMA*M32*M33+C21*GAMA*(M22*M33+M23*M32)
C
      L21=L21A+A13*K2*M11**3+DW1*M21**3+DW2*M31*M21**2
     &    +     DW3*M21*M31**2+DW4*M31**3
```

```
      L22=L22A+3*A13*K2*M12*M11**2+3*DW1*M22*M21**2
     &    +      DW2*(2*M21*M22*M31+M32*M21**2)
     &    +      DW3*(2*M21*M31*M32+M22*M31**2)
     &    +      3*DW4*M32*M31**2
      L23=L23A+3*A13*K2*M11*M12**2+3*DW1*M21*M22**2
     &    +      DW2*(M31*M22**2+2*M21*M22*M32)
     &    +      DW3*(M21*M32**2+2*M22*M31*M32)
     &    +      3*DW4*M31*M32**2
      L24=L24A+A13*K2*M12**3+DW1*M22**3+DW2*M32*M22**2
     &    +      DW3*M22*M32**2+DW4*M32**3
      L31=L31A+A23*K2*M11**3+DQ1*M21**3+DQ2*M31*M21**2
     &    +      DQ3*M21*M31**2+DQ4*M31**3
      L32=L32A+3*A23*K2*M12*M11**2+3*DQ1*M22*M21**2
     &    +      DQ2*(2*M21*M22*M31+M32*M21**2)
     &    +      DQ3*(2*M21*M31*M32+M22*M31**2)
     &    +      3*DQ4*M32*M31**2
      L33=L33A+3*A23*K2*M11*M12**2+3*DQ1*M21*M22**2
     &    +      DQ2*(M31*M22**2+2*M21*M22*M32)
     &    +      DQ3*(M21*M32**2+2*M22*M31*M32)
     &    +      3*DQ4*M31*M32**2
      L34=L34A+A23*K2*M12**3+DQ1*M22**3+DQ2*M32*M22**2
     &    +      DQ3*M22*M32**2+DQ4*M32**3
C
      R11=N12*L21+N13*L31
      R12=N12*L22+N13*L32
      R13=N12*L23+N13*L33
      R14=N12*L24+N13*L34
      R21=N22*L21+N23*L31
      R22=N22*L22+N23*L32
      R23=N22*L23+N23*L33
      R24=N22*L24+N23*L34
C
C     EVALUATE DALPHA AND DOMEGA
C
      UINC=0.001
      UR =U+UINC
      UL =U-UINC
      U  =UR
C
      A(1,1)=0.0
      A(1,2)=0.0
      A(1,3)=1.0
```

```
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
C
      CALL RG(3,3,A,WR,WI,0,YYY,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
      ALPHR=DEOS
      OMEGR=FREQ
C
      U=UL
C
      A(1,1)=0.0
      A(1,2)=0.0
      A(1,3)=1.0
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
C
      CALL RG(3,3,A,WR,WI,0,YYY,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
      ALPHL=DEOS
      OMEGL=FREQ
C
      DALPHA=(ALPHR-ALPHL)/(UR-UL)
      DOMEGA=(OMEGR-OMEGL)/(UR-UL)
C
C     EVALUATION OF HOPF BIFURCATION COEFFICIENTS
C
      COEF1=3.0*R11+R13+R22+3.0*R24
      COEF2=3.0*R21+R23-R12-3.0*R14
      AMU2 =-COEF1/(8.0*DALPHA)
      BETA2=0.25*COEF1
C     TAU2 =-(COEF2-DOMEGA*COEF1/DALPHA)/(8.0*OMEGA0)
C     PER  =2.0*3.1415927/OMEGA0
C     PER  =PER*U/L
      WRITE (20,2001) XG/L,U,COEF1,DALPHA,OMEGA0,PEIG
```

```
    1 CONTINUE
      STOP
 1001 FORMAT (' ENTER NUMBER OF DATA LINES')
 1002 FORMAT (' ENTER UO, ZG, AND DSAT')
 1003 FORMAT (' ENTER BOW PLANE TO STERN PLANE RATIO')
 1004 FORMAT (' ENTER ZG')
 2001 FORMAT (6E14.5)
 4001 FORMAT (3F15.5)
 3001 FORMAT (2I5)
      END
C=======================================================================
      SUBROUTINE DSOMEG(IJK,WR,WI,OMEGA,CHECK)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WR(3),WI(3)
      CHECK=-1.0D+25
      DO 1 I=1,3
        IF (WR(I).LT.CHECK) GO TO 1
        CHECK=WR(I)
        IJ=I
    1 CONTINUE
      OMEGA=DABS(WI(IJ))
      IF (WI(IJ).GT.0.D0) IJK=IJ
      IF (WI(IJ).LT.0.D0) IJK=IJ-1
      RETURN
      END
C=======================================================================
      SUBROUTINE DSTABL(DEOS,WR,WI,OMEGA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WR(3),WI(3)
      DEOS=-1.0D+20
      DO 1 I=1,3
        IF (WR(I).LT.DEOS) GO TO 1
        DEOS=WR(I)
        IJ=I
    1 CONTINUE
      OMEGA=WI(IJ)
      OMEGA=DABS(OMEGA)
      RETURN
      END
C=======================================================================
      SUBROUTINE NORMAL(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      DIMENSION T(3,3),TNOR(3,3)
      CFAC=T(1,1)**2+T(1,2)**2
      IF (DABS(CFAC).LE.(1.D-10)) STOP 4001
      TNOR(1,1)=1.D0
      TNOR(2,1)=(T(1,1)*T(2,1)+T(2,2)*T(1,2))/CFAC
      TNOR(3,1)=(T(1,1)*T(3,1)+T(3,2)*T(1,2))/CFAC
      TNOR(1,2)=0.D0
      TNOR(2,2)=(T(2,2)*T(1,1)-T(2,1)*T(1,2))/CFAC
      TNOR(3,2)=(T(3,2)*T(1,1)-T(3,1)*T(1,2))/CFAC
      DO 1 I=1,3
        DO 2 J=1,2
          T(I,J)=TNOR(I,J)
    2   CONTINUE
    1 CONTINUE
      RETURN
      END
C=======================================================================
      SUBROUTINE MULT(TINV,A,T,A2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION TINV(3,3),A(3,3),T(3,3),A1(3,3),A2(3,3)
      DO 1 I=1,3
        DO 2 J=1,3
          A1(I,J)=0.D0
          A2(I,J)=0.D0
    2   CONTINUE
    1 CONTINUE
      DO 3 I=1,3
        DO 4 J=1,3
          DO 5 K=1,3
            A1(I,J)=A(I,K)*T(K,J)+A1(I,J)
    5     CONTINUE
    4   CONTINUE
    3 CONTINUE
      DO 6 I=1,3
        DO 7 J=1,3
          DO 8 K=1,3
            A2(I,J)=TINV(I,K)*A1(K,J)+A2(I,J)
    8     CONTINUE
    7   CONTINUE
    6 CONTINUE
      DO 11 I=1,3
C       WRITE (*,101) (A(I,J),J=1,3)
```

```
   11 CONTINUE
      DO 12 I=1,3
C        WRITE (*,101) (T(I,J),J=1,3)
   12 CONTINUE
      DO 10 I=1,3
C       WRITE (*,101) (A2(I,J),J=1,3)
   10 CONTINUE
C       WRITE (*,101) A2(1,1)
      RETURN
  101 FORMAT (3E15.5)
      END
C=============================================================
      SUBROUTINE TRAP(N,A,B,OUT)
C
C     NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      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     PROGRAM HOPF_DELTA.FOR
C     EVALUATION OF HOPF BIFURCATION FORMULAS
C     STERN PLANE ANGLE IS NON ZERO
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION L,IY,MASS,MQDOT,MWDOT,ND1,THETA,
     1                 MQ,MW,MD,ZD,MDS,MDB,K1,K2,
     2                 ALFA,BETA,GAMA,DELTA,
     3                 E0,E1,E2,E3,E4,
     4                 DW1,DW2,DW3,DW4,
     5                 DQ1,DQ2,DQ3,DQ4
      DOUBLE PRECISION M11,M12,M13,M21,M22,M23,
     1                 M31,M32,M33,
     2                 N11,N12,N13,N21,N22,N23,
     3                 N31,N32,N33,
     4                 L21,L22,L23,L24,L31,L32,L33,L34,
     5                 L25,L26,L27,L35,L36,L37,
     6                 L21A,L22A,L23A,L24A,L31A,
     7                 L32A,L33A,L34A
C
      DIMENSION A(3,3),T(3,3),TINV(3,3),FV1(3),IV1(3),YYY(3,3)
      DIMENSION WR(3),WI(3),TSAVE(3,3),TLUD(3,3),IVLUD(3),SVLUD(3)
      DIMENSION ASAVE(3,3),A1(3,3),A2(3,3),XL(25),BR(25)
      DIMENSION VEC0(25),VEC1(25),VEC2(25),VEC3(25),VEC4(25)
C
      OPEN (10,FILE='HOPF.DAT',STATUS='OLD')
      OPEN (20,FILE='HOPF.RES',STATUS='NEW')
C
      WEIGHT=1556.2363
      IY    = 561.32
      L     =  13.9792
      RHO   =   1.94
      WRITE (*,*) ' ENTER CD'
      READ  (*,*) CD
      CD    = 0.5*CD*RHO
      G     =  32.2
      XB    =   0.0
      WRITE (*,*) ' ENTER ZG/L'
      READ  (*,*) ZG
      ZG    =ZG*L
      ZB    =   0.0
```

```
      MASS   = WEIGHT/G
      BOY    = WEIGHT
      ND1    = 0.5*RHO*L**2
      ZQDOT =-6.3300E-04*0.5*RHO*L**4
      ZWDOT =-1.4529E-02*0.5*RHO*L**3
      ZQ     = 7.5450E-03*0.5*RHO*L**3
      ZW     =-1.3910E-02*0.5*RHO*L**2
      MQDOT =-8.8000E-04*0.5*RHO*L**5
      MWDOT =-5.6100E-04*0.5*RHO*L**4
      MQ     =-3.7020E-03*0.5*RHO*L**4
      MW     = 1.0324E-02*0.5*RHO*L**3
C
      ZGB=ZG-ZB
C
C     DEFINE THE LENGTH X AND BREADTH B TERMS FOR THE INTEGRATION
C
      XL( 1)=  0.0000
      XL( 2)=  0.1000
      XL( 3)=  0.2000
      XL( 4)=  0.3000
      XL( 5)=  0.4000
      XL( 6)=  0.5000
      XL( 7)=  0.6000
      XL( 8)=  0.7000
      XL( 9)=  1.0000
      XL(10)=  2.0000
      XL(11)=  3.0000
      XL(12)=  4.0000
      XL(13)=  7.7143
      XL(14)= 10.0000
      XL(15)= 15.1429
      XL(16)= 16.0000
      XL(17)= 17.0000
      XL(18)= 18.0000
      XL(19)= 19.0000
      XL(20)= 20.0000
      XL(21)= 20.1000
      XL(22)= 20.2000
      XL(23)= 20.3000
      XL(24)= 20.4000
      XL(25)= 20.4167
```

```
      DO 102 N = 1,25
        XL(N) = (L/20.)*XL(N) - L/2.
  102 CONTINUE

C
      BR( 1)= 0.000
      BR( 2)= 0.485
      BR( 3)= 0.658
      BR( 4)= 0.778
      BR( 5)= 0.871
      BR( 6)= 0.945
      BR( 7)= 1.010
      BR( 8)= 1.060
      BR( 9)= 1.180
      BR(10)= 1.410
      BR(11)= 1.570
      BR(12)= 1.660
      BR(13)= 1.670
      BR(14)= 1.670
      BR(15)= 1.670
      BR(16)= 1.630
      BR(17)= 1.370
      BR(18)= 0.919
      BR(19)= 0.448
      BR(20)= 0.195
      BR(21)= 0.188
      BR(22)= 0.168
      BR(23)= 0.132
      BR(24)= 0.053
      BR(25)= 0.000
C
      DO 104 K = 1,25
        VEC0(K)=BR(K)
        VEC1(K)=XL(K)*BR(K)
        VEC2(K)=XL(K)*XL(K)*BR(K)
        VEC3(K)=XL(K)*XL(K)*XL(K)*BR(K)
        VEC4(K)=XL(K)*XL(K)*XL(K)*XL(K)*BR(K)
  104 CONTINUE
      CALL TRAP(25,VEC0,XL,E0)
      CALL TRAP(25,VEC1,XL,E1)
      CALL TRAP(25,VEC2,XL,E2)
      CALL TRAP(25,VEC3,XL,E3)
```

```
      CALL TRAP(25,VEC4,XL,E4)
      WRITE (*,*) ' ENTER GAMMA'
      READ  (*,*) EPSILON
      IXG=80
C=====================================================================
      DO 1 IT=1,IXG
        READ (10,*) XG,U,WO,THETAO
        WRITE (*,3001) IT,IXG
        XG=XG*L
        XGB=XG-XB
        DV=(MASS-ZWDOT)*(IY-MQDOT)-(MASS*XG+ZQDOT)*(MASS*XG+MWDOT)
        CONST2=CD/(3.0*DV*EPSILON)
        TDW1=CONST2*((IY-MQDOT)*(-E0)+(MASS*XG+ZQDOT)*(E1))
        TDW2=CONST2*((IY-MQDOT)*(3.0*E1)+(MASS*XG+ZQDOT)*(-3.0*E2))
        TDW3=CONST2*((IY-MQDOT)*(-3.0*E2)+(MASS*XG+ZQDOT)*(3.0*E3))
        TDW4=CONST2*((IY-MQDOT)*(E3)+(MASS*XG+ZQDOT)*(-E4))
        TDQ1=CONST2*((MASS-ZWDOT)*(E1)+(MASS*XG+MWDOT)*(-E0))
        TDQ2=CONST2*((MASS-ZWDOT)*(-3.0*E2)+(MASS*XG+MWDOT)*(3.0*E1))
        TDQ3=CONST2*((MASS-ZWDOT)*(3.0*E3)+(MASS*XG+MWDOT)*(-3.0*E2))
        TDQ4=CONST2*((MASS-ZWDOT)*(-E4)+(MASS*XG+MWDOT)*(E3))
C
        CONST=TANH(WO/EPSILON)*CD/DV
        DW1=CONST*( (IY-MQDOT)*(-E0)     +(MASS*XG+ZQDOT)*E1)
        DW2=CONST*( (IY-MQDOT)*(2*E1)    -(MASS*XG+ZQDOT)*2*E2)
        DW3=CONST*( (IY-MQDOT)*(-E2)     +(MASS*XG+ZQDOT)*E3)
        DQ1=CONST*( (MASS-ZWDOT)*(E1)    +(MASS*XG+MWDOT)*(-E0))
        DQ2=CONST*( (MASS-ZWDOT)*(-2*E2)+(MASS*XG+MWDOT)*2*E1)
        DQ3=CONST*( (MASS-ZWDOT)*(E3)    +(MASS*XG+MWDOT)*(-E2))
C
C       DETERMINE [A] AND [B] COEFFICIENTS
C
        A11DV=(IY-MQDOT)*(ZW-2*CD*E0*WO)
     &        +(MASS*XG+ZQDOT)*(MW+2*CD*E1*WO)
        A12DV=(IY-MQDOT)*(MASS+ZQ+2*CD*E1*WO)+(MASS*XG+ZQDOT)
     &        *(MQ-MASS*XG-MASS*ZG*WO-2*CD*E2*WO)
        A13DV=-(MASS*XG+ZQDOT)*WEIGHT
        A21DV=(MASS-ZWDOT)*(MW+2*CD*E1*WO)
     &        +(MASS*XG+MWDOT)*(ZW-2*CD*E0*WO)
        A22DV=(MASS-ZWDOT)*(MQ-MASS*XG-MASS*ZG*WO-2*CD*E2*WO)
     &        +(MASS*XG+MWDOT)*(MASS+ZQ+2*CD*E1*WO)
        A23DV=-(MASS-ZWDOT)*WEIGHT
C
```

```
      A11=A11DV/DV
      A12=A12DV/DV
      A13=A13DV/DV
      A21=A21DV/DV
      A22=A22DV/DV
      A23=A23DV/DV
C
      C11DV=(IY-MQDOT)*MASS*ZG
      C12DV=-(MASS*XG+ZQDOT)*MASS*ZG
      C21DV=-(MASS-ZWDOT)*MASS*ZG
      C22DV=(MASS*XG+MWDOT)*MASS*ZG
C
      C11=C11DV/DV
      C12=C12DV/DV
      C21=C21DV/DV
      C22=C22DV/DV
C=====================================================================
C     EVALUATE TRANSFORMATION MATRIX OF EIGENVECTORS
C
      K1=-(XGB*SIN(THETA0)-ZGB*COS(THETA0))
      K2=-(1./6.)*(ZGB*COS(THETA0)-XGB*SIN(THETA0))
C
      A(1,1)=0.0
      A(1,2)=0.0
      A(1,3)=1.0
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
      DO 11 I=1,3
        DO 12 J=1,3
          ASAVE(I,J)=A(I,J)
12      CONTINUE
11    CONTINUE
      CALL RG(3,3,A,WR,WI,1,YYY,IV1,FV1,IERR)
      CALL DSOMEG(IEV,WR,WI,OMEGA,CHECK)
      OMEGA0=OMEGA
      DO 5 I=1,3
        T(I,1)= YYY(I,IEV)
        T(I,2)=-YYY(I,IEV+1)
```

```
    5   CONTINUE
        IF (IEV.EQ.1) GO TO 13
        IF (IEV.EQ.2) GO TO 14
        STOP 3004
   14   DO 6 I=1,3
          T(I,3)=YYY(I,1)
    6   CONTINUE
        GO TO 17
   13   DO 16 I=1,3
          T(I,3)=YYY(I,3)
   16   CONTINUE
   17   CONTINUE
C
C       NORMALIZATION OF THE CRITICAL EIGENVECTOR
C
        INORM=1
        IF (INORM.NE.0) CALL NORMAL(T)
C
C       INVERT TRANSFORMATION MATRIX
C
        DO 2 I=1,3
          DO 3 J=1,3
            TINV(I,J)=0.0
            TSAVE(I,J)=T(I,J)
    3     CONTINUE
    2   CONTINUE
        CALL DLUD(3,3,TSAVE,3,TLUD,IVLUD)
        DO 4 I=1,3
          IF (IVLUD(I).EQ.0) STOP 3003
    4   CONTINUE
        CALL DILU(3,3,TLUD,IVLUD,SVLUD)
        DO 8 I=1,3
          DO 9 J=1,3
            TINV(I,J)=TLUD(I,J)
    9     CONTINUE
    8   CONTINUE
C
C       CHECK Inv(T)*A*T
C
        IMULT=1
        IF (IMULT.EQ.1) CALL MULT(TINV,ASAVE,T,A2)
        IF (IMULT.EQ.0) STOP
```

```
      P=A2(3,3)
      PEIG=P
C
C     DEFINITION OF Nij
C
      N11=TINV(1,1)
      N21=TINV(2,1)
      N31=TINV(3,1)
      N12=TINV(1,2)
      N22=TINV(2,2)
      N32=TINV(3,2)
      N13=TINV(1,3)
      N23=TINV(2,3)
      N33=TINV(3,3)
C
C     DEFINITION OF Mij
C
      M11=T(1,1)
      M21=T(2,1)
      M31=T(3,1)
      M12=T(1,2)
      M22=T(2,2)
      M32=T(3,2)
      M13=T(1,3)
      M23=T(2,3)
      M33=T(3,3)
C
C     DEFINITION OF Lij
C
      L25=C11*M31*M31+C12*M21*M31
      L26=2*C11*M31*M32+C12*(M21*M32+M22*M31)
      L27=C11*M32*M32+C12*M22*M33
      L35=C22*M31*M31+C21*M21*M31
      L36=2*C22*M31*M32+C21*(M21*M32+M22*M31)
      L37=C22*M32*M32+C21*M33*M22
C
C     DEFINITION OF ALFA, BETA, GAMA
C
      D1 =N32*L25 + N33*L35
      D2 =N32*L26 + N33*L36
      D3 =N32*L27 + N33*L37
C
```

```
      D11=-P
      D12=OMEGAO
      D21=-2*OMEGAO
      D22=-P
      D23=2*OMEGAO
      D32=-OMEGAO
      D33=-P
C
      BETA=(D2-D21*D1/D11-D23*D3/D33)/(D22-D21*D12/D11-D23*D32/D33)
      ALFA=(D1-D12*BETA)/D11
      GAMA=(D3-D32*BETA)/D33
C
      L21A=2*(C11+DW3)*ALFA*M31*M33+(C12+DW2)*ALFA*(M21*M33+M23*M31)
     &     +2*DW1*M21*M23*ALFA
C
      L22A=2*(C11+DW3)*ALFA*M32*M33 + 2*(C11+DW3)*BETA*M31*M33
     &     +  (C12+DW2)*ALFA*(M22*M33+M32*M23)
     &     +  (C12+DW2)*BETA*(M21*M33+M23*M31)
     &     +  2*DW1*(M21*M23*BETA+M22*M23*ALFA)
C
      L23A=2*(C11+DW3)*GAMA*M31*M33+2*(C11+DW3)*BETA*M32*M33
     &     +  (C12+DW2)*GAMA*(M21*M33+M23*M31)
     &     +  (C12+DW2)*BETA*(M22*M33+M23*M32)
     &     +  2*DW1*(M21*M23*GAMA+M22*M23*BETA)
C
      L24A=2*(C11+DW3)*GAMA*M32*M33+(C12+DW2)*GAMA*(M22*M33+M23*M32)
     &     +2*DW1*M22*M23*GAMA
C
      L31A=2*(C22+DQ3)*ALFA*M31*M33+(C21+DQ2)*ALFA*(M21*M33+M23*M31)
     &     +2*DQ1*M21*M23*ALFA
C
      L32A=2*(C22+DQ3)*ALFA*M32*M33 + 2*(C22+DQ3)*BETA*M31*M33
     &     +  (C21+DQ2)*ALFA*(M22*M33+M32*M23)
     &     +  (C21+DQ2)*BETA*(M21*M33+M23*M31)
     &     +  2*DQ1*(M21*M23*BETA+M22*M23*ALFA)
C
      L33A=2*(C22+DQ3)*GAMA*M31*M33+2*(C22+DQ3)*BETA*M32*M33
     &     +  (C21+DQ2)*GAMA*(M21*M33+M23*M31)
     &     +  (C21+DQ2)*BETA*(M22*M33+M23*M32)
     &     +  2*DQ1*(M21*M23*GAMA+M22*M23*BETA)
C
      L34A=2*(C22+DQ3)*GAMA*M32*M33+(C21+DQ2)*GAMA*(M22*M33+M23*M32)
```

```
     &      +2*DQ1*M22*M23*GAMA
C
       L21=L21A+A13*K2*M11**3
       L22=L22A+3*A13*K2*M12*M11**2
       L23=L23A+3*A13*K2*M11*M12**2
       L24=L24A+A13*K2*M12**3
       L31=L31A+A23*K2*M11**3
       L32=L32A+3*A23*K2*M12*M11**2
       L33=L33A+3*A23*K2*M11*M12**2
       L34=L34A+A23*K2*M12**3
C
       L21=L21+TDW1*M21**3+TDW2*M31*M21**2+TDW3*M21*M31**2+TDW4*M31**3
       L22=L22+TDW1*3.0*M22*M21**2+TDW2*(2.0*M21*M31*M22+M32*M21**2)
     &    +TDW3*(2.0*M31*M32*M21+M22*M31**2)+TDW4*3.0*M32*M31**2
       L23=L23+TDW1*3.0*M21*M22**2+TDW2*(M31*M22**2+2.0*M21*M22*M32)
     &    +TDW3*(M21*M32**2+2.0*M31*M32*M22)+TDW4*3.0*M31*M32**2
       L24=L24+TDW1*M22**3+TDW2*M32*M22**2+TDW3*M22*M32**2+TDW4*M32**3
       L31=L31+TDQ1*M21**3+TDQ2*M31*M21**2+TDQ3*M21*M31**2+TDQ4*M31**3
       L32=L32+TDQ1*3.0*M22*M21**2+TDQ2*(2.0*M21*M31*M22+M32*M21**2)
     &    +TDQ3*(2.0*M31*M32*M21+M22*M31**2)+TDQ4*3.0*M32*M31**2
       L33=L33+TDQ1*3.0*M21*M22**2+TDQ2*(M31*M22**2+2.0*M21*M22*M32)
     &    +TDQ3*(M21*M32**2+2.0*M31*M32*M22)+TDQ4*3.0*M31*M32**2
       L34=L34+TDQ1*M22**3+TDQ2*M32*M22**2+TDQ3*M22*M32**2+TDQ4*M32**3
C
       R11=N12*L21+N13*L31
       R12=N12*L22+N13*L32
       R13=N12*L23+N13*L33
       R14=N12*L24+N13*L34
       R21=N22*L21+N23*L31
       R22=N22*L22+N23*L32
       R23=N22*L23+N23*L33
       R24=N22*L24+N23*L34
C
C      EVALUATE DALPHA AND DOMEGA
C
       UINC=0.001
       UR =U+UINC
       UL =U-UINC
       U  =UR
C
       A(1,1)=0.0
       A(1,2)=0.0
```

```
      A(1,3)=1.0
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
C
      CALL RG(3,3,A,WR,WI,0,YYY,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
      ALPHR=DEOS
      OMEGR=FREQ
C
      U=UL
C
      A(1,1)=0.0
      A(1,2)=0.0
      A(1,3)=1.0
      A(2,1)=A13*K1
      A(2,2)=A11*U
      A(2,3)=A12*U
      A(3,1)=A23*K1
      A(3,2)=A21*U
      A(3,3)=A22*U
C
      CALL RG(3,3,A,WR,WI,0,YYY,IV1,FV1,IERR)
      CALL DSTABL(DEOS,WR,WI,FREQ)
      ALPHL=DEOS
      OMEGL=FREQ
C
      DALPHA=(ALPHR-ALPHL)/(UR-UL)
      DOMEGA=(OMEGR-OMEGL)/(UR-UL)
C
C     EVALUATION OF HOPF BIFURCATION COEFFICIENTS
C
      COEF1=3.0*R11+R13+R22+3.0*R24
      COEF2=3.0*R21+R23-R12-3.0*R14
      AMU2 =-COEF1/(8.0*DALPHA)
      BETA2=0.25*COEF1
C     TAU2 =-(COEF2-DOMEGA*COEF1/DALPHA)/(8.0*OMEGA0)
C     PER  =2.0*3.1415927/OMEGA0
C     PER  =PER*U/L
```

```
        WRITE (20,2001) XG/L,U,COEF1,DALPHA,OMEGAO,PEIG
    1 CONTINUE
      STOP
 1001 FORMAT (' ENTER NUMBER OF DATA LINES')
 1002 FORMAT (' ENTER UO, ZG, AND DSAT')
 1003 FORMAT (' ENTER BOW PLANE TO STERN PLANE RATIO')
 1004 FORMAT (' ENTER ZG')
 2001 FORMAT (6E14.5)
 4001 FORMAT (3F15.5)
 3001 FORMAT (2I5)
      END
C=======================================================================
      SUBROUTINE DSOMEG(IJK,WR,WI,OMEGA,CHECK)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WR(3),WI(3)
      CHECK=-1.0D+25
      DO 1 I=1,3
        IF (WR(I).LT.CHECK) GO TO 1
        CHECK=WR(I)
        IJ=I
    1 CONTINUE
      OMEGA=DABS(WI(IJ))
      IF (WI(IJ).GT.0.D0) IJK=IJ
      IF (WI(IJ).LT.0.D0) IJK=IJ-1
      RETURN
      END
C=======================================================================
      SUBROUTINE DSTABL(DEOS,WR,WI,OMEGA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WR(3),WI(3)
      DEOS=-1.0D+20
      DO 1 I=1,3
        IF (WR(I).LT.DEOS) GO TO 1
        DEOS=WR(I)
        IJ=I
    1 CONTINUE
      OMEGA=WI(IJ)
      OMEGA=DABS(OMEGA)
      RETURN
      END
C=======================================================================
      SUBROUTINE NORMAL(T)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION T(3,3),TNOR(3,3)
      CFAC=T(1,1)**2+T(1,2)**2
      IF (DABS(CFAC).LE.(1.D-10)) STOP 4001
      TNOR(1,1)=1.DO
      TNOR(2,1)=(T(1,1)*T(2,1)+T(2,2)*T(1,2))/CFAC
      TNOR(3,1)=(T(1,1)*T(3,1)+T(3,2)*T(1,2))/CFAC
      TNOR(1,2)=0.DO
      TNOR(2,2)=(T(2,2)*T(1,1)-T(2,1)*T(1,2))/CFAC
      TNOR(3,2)=(T(3,2)*T(1,1)-T(3,1)*T(1,2))/CFAC
      DO 1 I=1,3
        DO 2 J=1,2
          T(I,J)=TNOR(I,J)
    2   CONTINUE
    1 CONTINUE
      RETURN
      END
C=======================================================================
      SUBROUTINE MULT(TINV,A,T,A2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION TINV(3,3),A(3,3),T(3,3),A1(3,3),A2(3,3)
      DO 1 I=1,3
        DO 2 J=1,3
          A1(I,J)=0.DO
          A2(I,J)=0.DO
    2   CONTINUE
    1 CONTINUE
      DO 3 I=1,3
        DO 4 J=1,3
          DO 5 K=1,3
            A1(I,J)=A(I,K)*T(K,J)+A1(I,J)
    5     CONTINUE
    4   CONTINUE
    3 CONTINUE
      DO 6 I=1,3
        DO 7 J=1,3
          DO 8 K=1,3
            A2(I,J)=TINV(I,K)*A1(K,J)+A2(I,J)
    8     CONTINUE
    7   CONTINUE
    6 CONTINUE
      DO 11 I=1,3
```

```
C        WRITE (*,101) (A(I,J),J=1,3)
   11 CONTINUE
      DO 12 I=1,3
C        WRITE (*,101) (T(I,J),J=1,3)
   12 CONTINUE
      DO 10 I=1,3
C       WRITE (*,101) (A2(I,J),J=1,3)
   10 CONTINUE
C      WRITE (*,101) A2(1,1)
      RETURN
  101 FORMAT (3E15.5)
      END
C=======================================================================
      SUBROUTINE TRAP(N,A,B,OUT)
C
C     NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      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     PROGRAM SIM.FOR
C
C     SIMULATION PROGRAM USING A FOURTH ORDER ADAMS-BASHFORTH
C     INTEGRATION SCHEME
C
      REAL L,MW,MASS,IY,MQDOT,MWDOT,MQ,
     *     U,TIME,ND1,ND2,ND3,ND4,
     *     F1(4),F2(4),F3(4)

C
      DIMENSION XL(25),BR(25),VEC1(25),VEC2(25)
C
      OPEN (10,FILE='SIM.DAT',STATUS='OLD')
      OPEN (20,FILE='SIM.RES',STATUS='NEW')
C
C     ENTER SPEEDS AND TIME DATA
C
      READ(10,*) U,TSIM,DELT,IPRNT,
     *           ZG,XG,THETA,W,Q
C
C     GEOMETRIC PROPERTIES AND HYDRODYNAMIC COEFFICIENTS
C
      PI    =4.0*ATAN(1.0)
      THETA =THETA*PI/180
      RHO   =1.94
      CDZ   =0.50
      L     =13.9792
      ND1   =0.5*RHO*L**2
      ND2   =0.5*RHO*L**3
      ND3   =0.5*RHO*L**4
      ND4   =0.5*RHO*L**5
      WEIGHT=1556.2363/(ND1*U**2)
      MASS  =1556.2363/(32.2*ND2)
      IY    =561.32/ND4
      ZQDOT=-6.3300E-04
      ZWDOT=-1.4529E-02
      ZQ    = 7.5450E-03
      ZW    =-1.3910E-02
      MQDOT=-8.8000E-04
      MWDOT=-5.6100E-04
      MQ    =-3.7020E-03
```

```
      MW    = 1.0324E-02
C
      ZB    =0.0
      XB    =0.0
      ZGB   =ZG-ZB
      XGB   =XG-XB
C
C     DETERMINE [A] AND [B] COEFFICIENTS
C
      DV    =(MASS-ZWDOT)*(IY-MQDOT)-(MASS*XG+ZQDOT)*(MASS*XG+MWDOT)
      A11DV=(IY-MQDOT)*ZW+(MASS*XG+ZQDOT)*MW
      A12DV=(IY-MQDOT)*(MASS+ZQ)+(MASS*XG+ZQDOT)*(MQ-MASS*XG)
      A13DV=-(MASS*XG+ZQDOT)*WEIGHT
      A21DV=(MASS-ZWDOT)*MW+(MASS*XG+MWDOT)*ZW
      A22DV=(MASS-ZWDOT)*(MQ-MASS*XG)+(MASS*XG+MWDOT)*(MASS+ZQ)
      A23DV=-(MASS-ZWDOT)*WEIGHT
C
      A11=A11DV/DV
      A12=A12DV/DV
      A13=A13DV/DV
      A21=A21DV/DV
      A22=A22DV/DV
      A23=A23DV/DV
C
C     DEFINE THE LENGTH X AND BREADTH B TERMS FOR THE INTEGRATION
C
      XL( 1)=  0.0000
      XL( 2)=  0.1000
      XL( 3)=  0.2000
      XL( 4)=  0.3000
      XL( 5)=  0.4000
      XL( 6)=  0.5000
      XL( 7)=  0.6000
      XL( 8)=  0.7000
      XL( 9)=  1.0000
      XL(10)=  2.0000
      XL(11)=  3.0000
      XL(12)=  4.0000
      XL(13)=  7.7143
      XL(14)= 10.0000
      XL(15)= 15.1429
      XL(16)= 16.0000
```

```
      XL(17)= 17.0000
      XL(18)= 18.0000
      XL(19)= 19.0000
      XL(20)= 20.0000
      XL(21)= 20.1000
      XL(22)= 20.2000
      XL(23)= 20.3000
      XL(24)= 20.4000
      XL(25)= 20.4167

      DO 102 N = 1,25
            XL(N) = (L/20)*XL(N) - L/2
 102  CONTINUE

C
      BR( 1)= 0.000
      BR( 2)= 0.485
      BR( 3)= 0.658
      BR( 4)= 0.778
      BR( 5)= 0.871
      BR( 6)= 0.945
      BR( 7)= 1.010
      BR( 8)= 1.060
      BR( 9)= 1.180
      BR(10)= 1.410
      BR(11)= 1.570
      BR(12)= 1.660
      BR(13)= 1.670
      BR(14)= 1.670
      BR(15)= 1.670
      BR(16)= 1.630
      BR(17)= 1.370
      BR(18)= 0.919
      BR(19)= 0.448
      BR(20)= 0.195
      BR(21)= 0.188
      BR(22)= 0.168
      BR(23)= 0.132
      BR(24)= 0.053
      BR(25)= 0.000

      DO 103 M = 1,25
```

```
            XL(M) = XL(M)/L
            BR(M) = BR(M)/L
 103  CONTINUE

C
      PISIM =TSIM/DELT
      ISIM  =PISIM
C
C     SIMULATION BEGINS
C
      DO 100 I=1,ISIM
C
C       CALCULATE THE DRAG FORCE, INTEGRATE THE DRAG OVER THE VEHICLE
C
        DO 101 K=1,25
          UCF=W-XL(K)*Q
          VEC1(K)=BR(K)*UCF*ABS(UCF)
          VEC2(K)=BR(K)*UCF*ABS(UCF)*XL(K)
  101   CONTINUE
        CALL TRAP(25,VEC1,XL,HEAVE)
        CALL TRAP(25,VEC2,XL,PITCH)
        HEAVE=CDZ*HEAVE
        PITCH=CDZ*PITCH
C
C       COUPLING EQUATIONS
C
        DWDV=(IY-MQDOT)*(-HEAVE)+(MASS*XG+ZQDOT)*PITCH
        DQDV=(MASS-ZWDOT)*PITCH+(MASS*XG+MWDOT)*(-HEAVE)
        C1DV=(IY-MQDOT)*MASS*ZG*Q**2-(MASS*XG+ZQDOT)*MASS*ZG*W*Q
        C2DV=-(MASS-ZWDOT)*MASS*ZG*W*Q+(MASS*XG+MWDOT)*MASS*ZG*Q**2
C
        DW=DWDV/DV
        DQ=DQDV/DV
        C1=C1DV/DV
        C2=C2DV/DV
C
        WDOT  =A11*W+A12*Q+A13*(XGB*COS(THETA)
     *          +ZGB*SIN(THETA))+DW+C1
        QDOT  =A21*W+A22*Q+A23*(XGB*COS(THETA)
     *          +ZGB*SIN(THETA))+DQ+C2
        THEDOT=Q
        IF (I.GT.3) GO TO 150
```

```
C
C       INITIAL FIRST ORDER INTEGRATION
C
        W     = W     + DELT*WDOT
        Q     = Q     + DELT*QDOT
        THETA = THETA + DELT*THEDOT
C
        F1(I)=WDOT
        F2(I)=QDOT
        F3(I)=THEDOT
C
C       ADAMS-BASHFORTH INTEGRATION
C
  150   F1(4)=A11*W+A12*Q+A13*(XGB*COS(THETA)+ZGB*SIN(THETA))+DW+C1
        F2(4)=A21*W+A22*Q+A23*(XGB*COS(THETA)+ZGB*SIN(THETA))+DQ+C2
        F3(4)=Q
C
        W    =W    +(DELT/24.0)*(55.0*F1(4)-59.0*F1(3)+37.0*F1(2)
     *             -9.0*F1(1))
        Q    =Q    +(DELT/24.0)*(55.0*F2(4)-59.0*F2(3)+37.0*F2(2)
     *             -9.0*F2(1))
        THETA=THETA+(DELT/24.0)*(55.0*F3(4)-59.0*F3(3)+37.0*F3(2)
     *             -9.0*F3(1))
C
        F1(1)=F1(2)
        F1(2)=F1(3)
        F1(3)=F1(4)
        F2(1)=F2(2)
        F2(2)=F2(3)
        F2(3)=F2(4)
        F3(1)=F3(2)
        F3(2)=F3(3)
        F3(3)=F3(4)
C
        TIME=I*DELT
        J=J+1
        IF (J.NE.IPRNT) GO TO 100
        ALFA=W
        ALFA=(-ATAN(ALFA))*180/PI
        WRITE (20,991) TIME,THETA*180/PI,Q,ALFA
        J=0
C
```

```
  100 CONTINUE
      STOP
  991 FORMAT (4E15.5)
      END
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
```

LIST OF REFERENCES

1. Arentzen, E. S. and Mandel, P. [1960] "Naval architectural aspects of submarine design", *Trans. Soc. of Naval Archit. & Marine Engrs.*, **68**, pp. 662–692.

2. Bender, C. M. and Orszag, S. A. [1978] *Advanced Mathematical Methods for Scientists and Engineers* (McGraw–Hill, New York).

3. Chow, S.–N. and Mallet–Paret, J. [1977] "Integral averaging and bifurcation", *Journal of Differential Equations*, **26**, pp. 112–159.

4. Clayton, B. R. and Bishop, R. E. D. [1982] *Mechanics of Marine Vehicles* (Gulf Publishing Company, Houston).

5. Clarke, F. [1983] *Optimization and Nonsmooth Analysis* (Wiley and Sons, New York).

6. Dalzell, J. F. [1978] "A note on the form of ship roll damping", *Journal of Ship Research*, **22**, 3.

7. Feldman, J. [1987] Straightline and rotating arm captive–model experiments to investigate the stability and control characteristics of submarines and other submerged vehicles. Carderock Division, Naval Surface Warfare Center, Report DTRC/SHD–0303–20.

8. Gertler, M. and Hagen, G. R. [1967] Standard equations of motion for submarine simulation. David Taylor Research Center, Report 2510.

9. Guckenheimer, J. and Holmes, P. [1983] *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer–Verlag, New York).

10. Hassard, B. and Wan, Y.H. [1978] "Bifurcation formulae derived from center manifold theory", *Journal of Mathematical Analysis and Applications*, **63**, pp. 297–312.

11. Papoulias F. A., Aydin, I., and McKinley, B. D. [1993] "Characterization of Steady State Solutions of Submarines Under Casualty Conditions", in *Nonlinear Dynamics of Marine Vehicles* (J. M. Falzarano, F. A. Papoulias, eds.), (ASME, New York).

12. Roddy, R. F. [1990] Investigation of the stability and control characteristics of several configurations of the DARPS SUBOFF model (DTRC

model 5470) from captive-model experiments. Carderock Division, Naval Surface Warfare Center, Report DTRC/SHD-1298-08.

13. Smith, N. S., Crane, J. W., and Summey, D. C. [1978] SDV simulator hydrodynamic coefficients. Naval Coastal Systems Center, Report NCSC-TM231-78.

14. Tinker, S. J. [1978] "Fluid memory effects on the trajectory of a submersible", *International Shipbuilding Progress*, **25**, 290.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304–6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943–5002	2
3. Chairman, Code ME Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943	1
4. Professor Fotis A. Papoulias, Code ME/Pa Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943	3
5. Embassy of Greece Naval Attache 2228 Massachusetts Ave., N.W. Washington, D.C. 20008	3
6. Harilaos I. Papadimitriou 38 Argolidos Str. Ampelokipoi, Athens 11523 Greece	3
7. Naval Engineering Curricular Office, Code 34 Naval Postgraduate School Monterey, California 93943	1