Forces and Moments Due to Unsteady Motion of an Underwater Vehicle

by

Erik D. Oller
B.S. Mechanical Engineering, University of New Mexico, 1993

Submitted to the Departments of Ocean Engineering and Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degrees of
Naval Engineer
and
Master of Science in Mechanical Engineering
at the
Massachusetts Institute of Technology
June 2003

Author ....
Department of Ocean Engineering
May 17, 2003

Certified by ....
Jerome H. Milgram
Professor of Ocean Engineering
Thesis Supervisor

Certified by ....
Ain A. Sonin
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by...
Ain A. Sonin
Professor of Mechanical Engineering
Chairperson, Committee on ~~Graduate Students~~

Accepted by ....
Michael Triantafyllou
Professor of Ocean Engineering
Chairman, Committee on Graduate Studies

Biographical Note

LT Erik Oller attended Massachusetts Institute of Technology as part of his preparation to become an Engineering Duty Officer for the United States Navy. His previous tours with the Navy include a tour on USS Trepang (SSN-674) and at Navy Personnel Command. Following graduation he will attend the Engineering Duty Officer's Basic Course on his way to report for duty at the Portsmouth Naval Shipyard.

LT Oller is married to the former Grace McCrea of Albuquerque, New Mexico and the couple has two children.

Forces and Moments Due to Unsteady Motion of an Underwater Vehicle

By

Erik D. Oller

Submitted to the Departments of Ocean Engineering and Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degrees of
Naval Engineer and Master of Science in Mechanical Engineering

This research examines the effect of unsteady motion on the forces and moments experienced by an underwater vehicle in shallow water. The test platform is the REMUS Autonomous Underwater Vehicle developed by the Woods Hole Oceanographic Institution, although the results are made non-dimensional to be applicable to a wide range of similar shaped vehicles. The experimental model was moved in sinusoidal motion at various submergences, speeds, frequencies of oscillation, and amplitudes of oscillation.

Thesis Supervisor: Jerome H. Milgram
Title: Professor of Ocean Engineering

Thesis Supervisor: Ain A. Sonin
Title: Professor of Mechanical Engineering

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Unmanned Underwater Vehicles (UUV's) perform many of their missions in shallow water environments subject to the forces of ocean waves and the proximity to the ocean floor. Under these conditions, accurate vertical position control is necessary to prevent broaching or hitting the ocean floor. Accurate horizontal position control is necessary to enable the UUV to conduct its mission with accuracy and return to a predetermined recovery point. Shallow water position control is made more difficult by ocean waves. In deep water the effects of these waves are negligible, but the effects in shallow water are significant. Important shallow water missions include pollution monitoring, marine life sampling, bottom contour mapping, and mine location.

Currently, UUV's are controlled in shallow water by altering empirical control parameters for better shallow water performance and by establishing empirically based operating depth limits on the UUV operations. These operating depth limits are based upon wave conditions. With a thorough understanding of the dynamics of UUV's in shallow water and the forces and moments on vehicles due to sea waves in these waters, improved control systems and vehicle designs can be achieved to allow the UUV to operate in shallower water and in larger waves than is commonly done. This will allow the UUV to be more effectively perform its missions.

This thesis explores the effects of variation in water depth and vehicle submergence on added mass, damping, and restoring forces.

1.2. Historical Background

This work builds on the work that Timothy Prestero performed to build a mathematical simulation of the REMUS behavior in deep water. This work is reported in his Master of Science Dissertation for the Massachusetts Institute of Technology/Woods Hole Oceanographic Institution Joint Program in Oceanography/Applied Ocean Science and Engineering entitled "Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle." Mr. Prestero calculated the hydrodynamic and hydrostatic coefficients based upon deep water performance far from a boundary surface. This thesis extends Mr. Prestero's work by determining those coefficients in shallow water and near the surface. [1]

1.3. Research Platform

The platform for this research is the REMUS (Remote Environmental Monitoring UnitS) AUV (autonomous underwater vehicle) developed by the Oceanographic Systems Laboratory at the Woods Hole Oceanographic Institution. This low-cost, modular AUV was developed for coastal monitoring and multiple vehicle survey operations.[2] REMUS has also been adopted for use in mine-counter measure operations for the United States Navy.[3] REMUS has most recently been used by the United States Navy to hunt for mines from the Iraqi port of Umm Qasr in support of Operation Enduring Freedom.[4]

1.4. Assumptions

To simplify analysis, the author made the following assumptions:

- The vehicle is port-starboard symmetric.
- The vehicle is a rigid body of constant mass.
- There are no significant vehicle dynamics occurring faster than the data sampling frequency of 25 Hz.

2. The Coordinate System

The coordinate system used for this research is shown in Figure 1. This is a body-fixed right-handed coordinate system with the x axis defined along the axial length of the vessel and the z axis defined downward. The origin of the body-fixed coordinate system is at the vessel amidships. The variables shown in Table 1 are defined using the coordinate system shown in Figure 1.

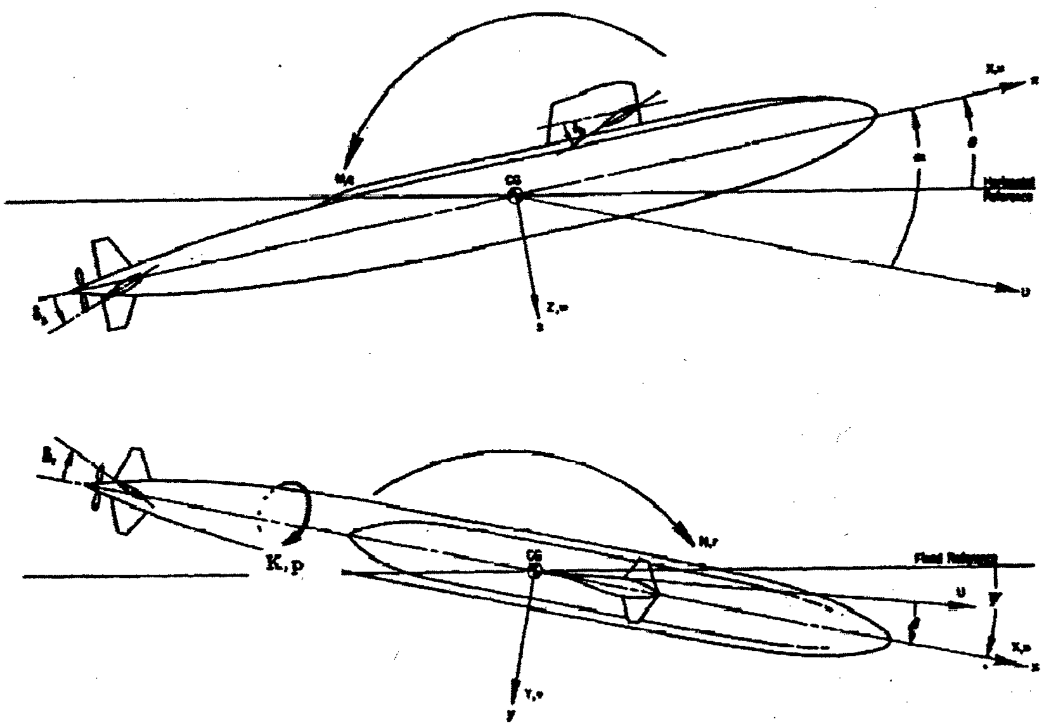

Figure 1. Sketch showing positive directions of axes, angles, velocities, forces and moments. (Feldman, 1979)

Table 1. Coordinate System Variables.

x	Surge position forward.
y	Sway position to the right.
z	Heave position downwards.
u	Velocity in the surge direction.
v	Velocity in the sway direction.
w	Velocity in the heave direction.
p	Rotation about the x axis.
q	Rotation about the y axis.

r Rotation about the z axis.
X Force in the x direction.
Y Force in the y direction.
Z Force in the z direction.
K Moment about the x axis.
M Moment about the y axis.
N Moment about the z axis.

3. The Equations of Motion

This work primarily explores the forces and moments in sway, heave, pitch and yaw due to motion of the vehicle itself in finite depth water. For each series of tests, the vessel was moved in only one plane at a time. The resulting hydrodynamic forces were determined by subtracting inertial forces from the measured forces. Then, the hydrodynamic coefficients were extracted from the hydrodynamic forces. The equations of motion were used to perform the mathematical operations.

The forces and moments associated with surge and roll have not been investigated.

3.1. Vessel Inertial Dynamics

The linearized equations of motion with a body-fixed coordinate system for an unrestrained vessel in water are given by

$$\begin{aligned}
X &= m\left[\dot{u} - vr + wq - x_G(q^2+r^2) + y_G(pq-\dot{r}) + z_G(pr+\dot{q})\right] \\
Y &= m\left[\dot{v} - wp + ur - y_G(r^2+p^2) + z_G(qr-\dot{p}) + x_G(qp+\dot{r})\right] \\
Z &= m\left[\dot{w} - uq + vp - z_G(p^2+r^2) + x_G(rp-\dot{q}) + y_G(rp+\dot{p})\right] \\
K &= I_{xx}\dot{p} + (I_{zz}-I_{yy})qr - I_{xz}(pq+\dot{r}) + I_{yz}(r^2-q^2) + I_{xy}(pr-\dot{q}) + \\
&\quad m[y_G(\dot{w}+pv-qu) - z_G(\dot{v}+ru-pw)] \\
M &= I_{yy}\dot{q} + (I_{xx}-I_{zz})pr - I_{xy}(\dot{p}+qr) + I_{xz}(p^2-r^2) + I_{yz}(qp-\dot{r}) + \\
&\quad m[z_G(\dot{u}+qw-rv) - x_G(\dot{w}+pv-qu)] \\
N &= I_{zz}\dot{r} + (I_{yy}-I_{xx})pq - I_{yz}(\dot{q}+rp) + I_{xy}(q^2-p^2) + I_{xz}(rq-\dot{p}) + \\
&\quad m[x_G(\dot{v}+ru-pw) - y_G(\dot{u}+qw-rv)]
\end{aligned} \tag{1}$$

where

m is the mass of the vessel

(x_G,y_G,z_G) are the coordinates of the center of gravity of the vessel in the body fixed coordinate system.

I_{jk} are the moments of inertia.

These equations can be simplified by fixing the coordinate system at the midship location of the vehicle. The equations can also be simplified by assuming that the lateral distance from the midship location to the center of gravity is negligible, i.e. y_G =0. Further simplification can be obtained by testing and analyzing motions in the vertical and horizontal planes separately. This research does not examine hydrodynamic forces in surge and roll, so the relevant simplified equations are:

$$\begin{aligned}
Y &= m\left[\dot{v} + Ur + x_G\dot{r}\right] \\
Z &= m\left[\dot{w} - Uq - z_G q^2 - x_G\dot{q}\right] \\
M &= I_{yy}\dot{q} + m\left[z_G\left(\dot{u} - vr + wq\right) - x_G\left(\dot{w} - Uq\right)\right] \\
N &= I_{zz}\dot{r} + mx_G\left(\dot{v} + ru\right)
\end{aligned} \tag{2}$$

3.2. Hydrodynamic and Hydrostatic Equations

This thesis explores the hydrodynamic forces and moments due to unsteady motion of an underwater vehicle. For that reason, hydrostatic effects have been removed from the data by subtracting the mean forces and moments from all measured forces during the analysis.

The forces and moments experienced by a ship are assumed to be the forces and moments arising from motions of the ship which in turn have been excited by another source. These forces and moments are computed as functions of speed and acceleration. A mathematically useful form is derived using the Taylor expansion of a function of multiple variables. For example, sway force, Y, and yaw moment, N, are represented functionally as

$$\begin{aligned}
Y &= F_y(u, v, \dot{u}, \dot{v}, r, \dot{r}) \\
N &= F_r(u, v, \dot{u}, \dot{v}, r, \dot{r})
\end{aligned} \tag{3}$$

The Taylor expansion of a single variable states that if the function of a variable, x, and all its derivatives are continuous at a particular value x_1, then the value of the function at a value of x close to x_1 can be expressed as

$$f(x) = f(x_1) + \delta x\frac{df(x)}{dx} + \frac{\delta x^2}{2!}\frac{d^2 f(x)}{dx^2} + \frac{\delta x^3}{3!}\frac{d^3 f(x)}{dx^3} + \ldots + \frac{\delta x^n}{n!}\frac{d^n f(x)}{dx^n} \tag{4}$$

where

f(x) is the value of the function at x close to x_1

$f(x_1)$ is the value of the function at $x = x_1$

$\delta x = x - x_1$

$\frac{d^n f(x)}{dx^n}$ is the *n*th derivative of the function evaluated at $x = x_1$

By making δx sufficiently small, higher order terms can be neglected. Equation (4) reduces to

$$f(x) = f(x_1) + \delta x\frac{df(x)}{dx} \tag{5}$$

and is called the linearized form of the Taylor expansion.

For functions of two variables the linearized form of the Taylor expansion is

$$f(x, y) = f(x1, y1) + \delta x\frac{\partial f(x,y)}{\partial x} + \delta y\frac{\partial f(x,y)}{\partial y} \tag{6}$$

Again, δx and δy must both be small enough that higher order terms can be neglected.

Hydrostatic motion stability typically considers the effect of very small perturbations on the behavior of the ship. Thus, the linearizing assumption for the Taylor expansion can be used to describe the hydrodynamic behavior of a body. Analysis of data from this research indicates that similar non-dimensional results were obtained for tests done at different amplitudes and the Fourier coefficients at the excitation frequencies dominated all others. Because of these facts, the linear terms do indeed predominate and the model based on them is sufficient to describe the relation between vehicle motions and the forces and moments they generate. Using the linearized Taylor expansion, equation (3) can be written as

$$\begin{aligned} Y &= F_y(u_1,v_1,\dot{u}_1,\dot{v}_1,r_1,\dot{r}_1)+(u-u_1)\frac{\partial Y}{\partial u}+(v-v_1)\frac{\partial Y}{\partial v}+\ldots+(\dot{r}-\dot{r}_1)\frac{\partial Y}{\partial \dot{r}} \\ N &= F_r(u_1,v_1,\dot{u}_1,\dot{v}_1,r_1,\dot{r}_1)+(u-u_1)\frac{\partial N}{\partial u}+(v-v_1)\frac{\partial N}{\partial v}+\ldots+(\dot{r}-\dot{r}_1)\frac{\partial N}{\partial \dot{r}} \end{aligned} \tag{7}$$

At this point several simplifying assumptions can be made. The first assumption is that the initial motion is in a straight line at some constant speed. Therefore, $\dot{u}_1=\dot{v}_1=r_1=\dot{r}_1=0$. The ship is symmetrical about the xz-plane, so $v_1 = 0$. Symmetry also leads to the conclusion that $\partial Y/\partial u=\partial Y/\partial \dot{u}=0$ because forward motion will not cause a lateral velocity. Also, a ship traveling forward in equilibrium in straight line motion experiences no sway force, so the term $F_y(u_1,v_1,\dot{u}_1,\dot{v}_1,r_1,\dot{r}_1)$ is also zero. The term u_1 is equal to the straight line velocity U. These assumptions reduce equation (7) to

$$\begin{aligned} Y &= \frac{\partial Y}{\partial v}v+\frac{\partial Y}{\partial \dot{v}}\dot{v}+\frac{\partial Y}{\partial r}r+\frac{\partial Y}{\partial \dot{r}}\dot{r} \\ N &= \frac{\partial N}{\partial v}v+\frac{\partial N}{\partial \dot{v}}\dot{v}+\frac{\partial N}{\partial r}r+\frac{\partial N}{\partial \dot{r}}\dot{r} \end{aligned} \tag{8}$$

In the simplified notation used by the Society of Naval Architects and Marine Engineers and including Pitch and Heave, the simplified linear hydrodynamic equations become[5]

$$\begin{aligned} Y &= Y_v v+Y_{\dot{v}}\dot{v}+Y_r r+Y_{\dot{r}}\dot{r} \\ N &= N_v v+N_{\dot{v}}\dot{v}+N_r r+N_{\dot{r}}\dot{r} \\ Z &= Z_w w+Z_{\dot{w}}\dot{w}+Z_q q+Z_{\dot{q}}\dot{q} \\ M &= M_w w+M_{\dot{w}}\dot{w}+M_q q+M_{\dot{q}}\dot{q} \end{aligned} \tag{9}$$

The simplified notation is interpreted such that $Y_v v$ is the sway force related to sway motion and Y_v is the maneuvering coefficient of sway force due to sway motion.

In accordance with the standard notation the terms of equation (9) include the effect of the rudder and stern planes held at zero degrees. The experiments to extract the coefficients were all performed with no deflection of the control surfaces. Other experiments were performed with control surface deflection. For those experiments, equation (9) has additional terms related to rudder and stern plane angle.[6]

3.3. Added Mass and Damping

The hydrodynamic forces relating to the motion of the body in the fluid can be divided into components in phase with the acceleration and components in phase with the velocity of the body. The hydrodynamic force due to the acceleration of the body in a fluid is known as an added mass force. The hydrodynamic force due to the velocity of the body in the fluid is known as a damping force. These forces are can be discerned by their phases relative to the driving motion. Forces in phase, but opposite in sign, with the driving motion are related to acceleration and are added mass forces. Forces 90 degrees out of phase with the driving motion are related to velocity and are damping forces. In terms of complex notation, the added mass is related to the real component of the measured force and the damping is related to the imaginary component of the measured force.

4. Non-Dimensionalizing

Throughout this thesis several quantities are given in both dimensional and non-dimensional form. Final results are given in non-dimensional form to be readily available for use with other bodies of similar shape. Non-dimensional quantities are denoted by a prime symbol ('). The equations for non-dimensionalizing are:[7]

$$\begin{aligned}
Y' &= \frac{Y}{\frac{1}{2}\rho U^2 L^2} \\
Y_v' &= \frac{Y_v}{\frac{1}{2}\rho U L^2} \\
Y_{\dot{v}}' &= \frac{Y_{\dot{v}}}{\frac{1}{2}\rho L^3} \\
Y_r' &= \frac{Y_r}{\frac{1}{2}\rho U L^3} \\
Y_{\dot{r}}' &= \frac{Y_{\dot{r}}}{\frac{1}{2}\rho L^4} \\
Z' &= \frac{Z}{\frac{1}{2}\rho U^2 L^2} \\
Z_w' &= \frac{Z_{\dot{w}}}{\frac{1}{2}\rho U L^2} \\
Z_{\dot{w}}' &= \frac{Z_{\dot{w}}}{\frac{1}{2}\rho L^3} \\
Z_q' &= \frac{Z_q}{\frac{1}{2}\rho U L^3} \\
Z_{\dot{q}}' &= \frac{Z_{\dot{q}}}{\frac{1}{2}\rho L^4}
\end{aligned} \tag{10}$$

$$
\begin{aligned}
M' &= \frac{M}{\frac{1}{2}\rho U^2 L^3} \\
M_w' &= \frac{M_w}{\frac{1}{2}\rho U L^3} \\
M_{\dot{w}}' &= \frac{M_{\dot{w}}}{\frac{1}{2}\rho L^4} \\
M_q' &= \frac{M_q}{\frac{1}{2}\rho U L^4} \\
M_{\dot{q}}' &= \frac{M_{\dot{q}}}{\frac{1}{2}\rho L^5} \\
N' &= \frac{N}{\frac{1}{2}\rho U^2 L^3} \\
N_v' &= \frac{N_v}{\frac{1}{2}\rho U L^3} \\
N_{\dot{v}}' &= \frac{N_{\dot{v}}}{\frac{1}{2}\rho L^4} \\
N_r' &= \frac{N_r}{\frac{1}{2}\rho U L^4} \\
N_{\dot{r}}' &= \frac{N_{\dot{r}}}{\frac{1}{2}\rho L^5}
\end{aligned}
\tag{11}
$$

Other non-dimensional equations include

$$m' = \frac{m}{\frac{1}{2}\rho L^3}$$

$$I_{zz}' = \frac{I_{zz}}{\frac{1}{2}\rho L^5}$$

$$x_G' = \frac{x_G}{L}$$

$$U' = \frac{U}{U} = 1$$

$$v' = \frac{1}{U} v$$

$$\dot{v}' = \frac{L}{U^2} \dot{v}$$

$$w' = \frac{1}{U} w$$

$$\dot{w}' = \frac{L}{U^2} \dot{w}$$

$$q = \frac{L}{U} q$$

$$\dot{q} = \frac{L^2}{U^2} \dot{q}$$

$$r = \frac{L}{U} r$$

$$\dot{r} = \frac{L^2}{U^2} \dot{r}$$

$$Fr = \frac{U}{\sqrt{gL}}$$

$$\omega' = \frac{\omega}{\sqrt{\frac{g}{L}}}$$

$$Submergence' = \frac{Length}{Submergence} \tag{12}$$

Using non-dimensional coefficient, the equations of motion have the form

$$Y' = Y_v' v' + Y_{\dot{v}}' \dot{v}' + Y_r' r' + Y_{\dot{r}}' \dot{r}' \tag{13}$$

5. Experimental Procedure

The determination of the maneuvering coefficients was conducted using both full scale and model scale experiments. Full scale experiments were used to determine the body lift and control surface effects. Model scale experiments were used to determine the unsteady motion effects.

5.1. Experiment Apparatus

5.1.1. Model Geometry

Figure 2 shows the geometry of the full scale model. The small scale model is geometrically similar at a scale of 0.4334. This scale was selected to provide the smallest model that would contain the transducer discussed in Section 5.1.2 without incidental contact between the transducer and the model.

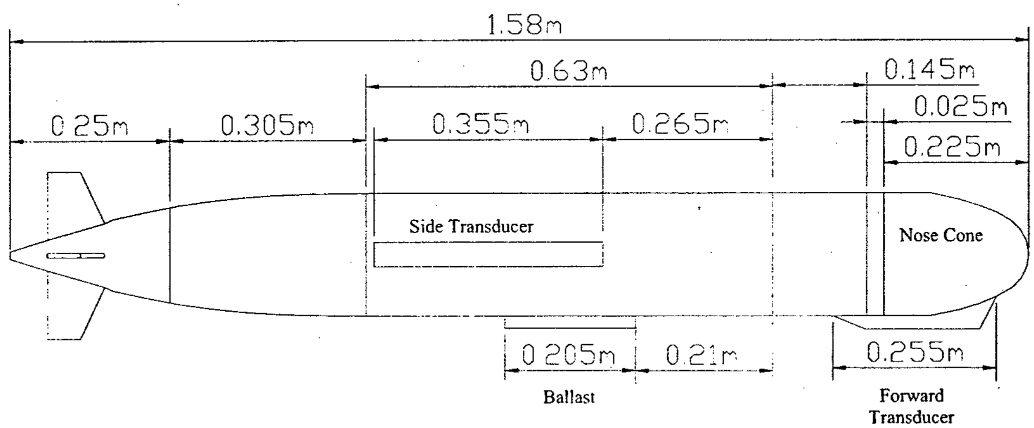

Figure 2. Full Scale Model Geometry

Figure 3 shows the full scale model mounted in the United States Naval Academy Towing Tank. Figure 4 shows the 0.4334 scale model mounted in the Massachusetts Institute of Technology Marine Computation and Instrumentation Laboratory.

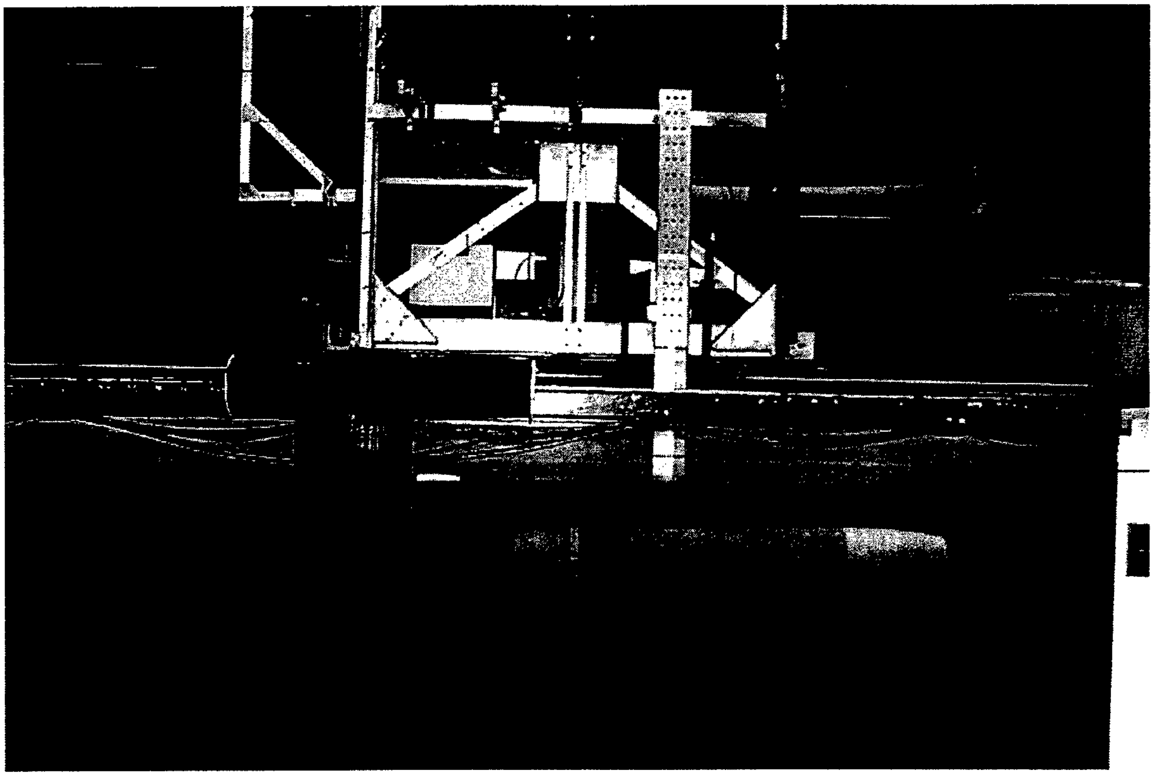

Figure 3. Full Scale Model Mounted in United States Naval Academy Towing Tank

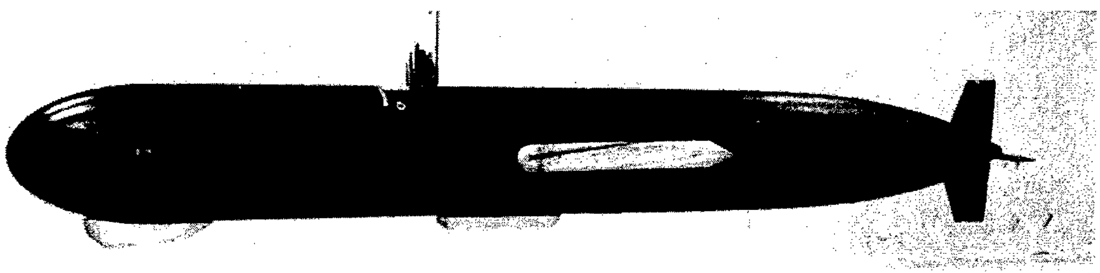

Figure 4. 0.4334 Scale Model Mounted at the MIT Marine Instrumentation and Computation Laboratory

5.1.2. Force and Moment Measurement

The forces and moments were measured using a UDW3 underwater transducer manufactured by Advanced Mechanical Technology, Inc. The transducer, shown in Figure 5, is able to simultaneously measure forces and moments in all of the three orthogonal directions (making six measurements of forces and moments) and is suitable for underwater applications. A pressure compensating bladder in the transducer equalizes internal and external pressures to allow underwater operation with little effect of hydrostatic pressure. The capacities and general specifications of the dynamometer are shown in Table 2.

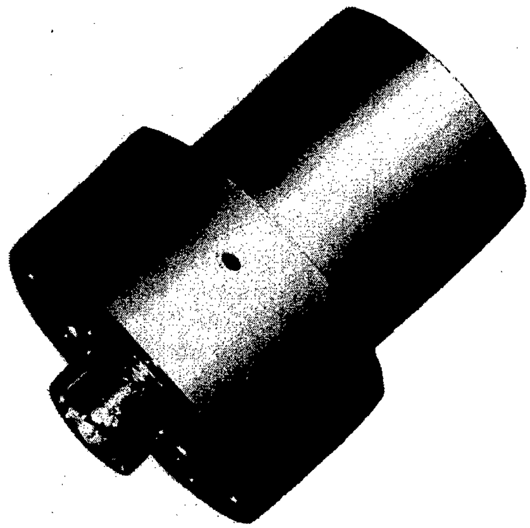

Figure 5. UDW3 Underwater Sensor.[8]

The transducer was mounted to a bulkhead within the volume of the vehicle. The strut was attached to the end of the transducer not attached to the vehicle. Sufficient clearance was provided to ensure the transducer output was not compromised by contact with the sides of the vehicle.

Table 2. Dynamometer Capacity and Specifications[9]

Vertical and Lateral Force Capacity	556 N
Axial Force Capacity	1112.1 N
Pitch and Yaw Moment Capacity	28.2 N-m
Roll Moment Capacity	14.1 N-m
Vertical and Lateral Force Sensitivity	2.7 $\mu V/(V*N)$
Axial Force Sensitivity	.67 $\mu V/(V*N)$
Pitch and Yaw Moment Sensitivity	137.2 $\mu V/(V*N-m)$
Roll Moment Sensitivity	97.4 $\mu V/(V*N-m)$
Vertical and Lateral Force Stiffness	5.3 x 10^6 N/m
Axial Force Stiffness	7.88 x 10^7 N/m
Roll Moment Stiffness	5.7 x 10^3 N-m/radian
Weight	2 kg
Recommended Excitation	10 V or less
Crosstalk	< 2% on all channels
Temperature Range	-17 to 52° C
Force Channel Hysteresis	± 0.2% Full Scale Output
Force Channel Non-Linearity	± 0.2% Full Scale Output

Excitation for the transducer and amplification for the output were provided by a MSA-6 Mini-Amplifier also developed by AMTI. This amplifier, shown in Figure 6, provides excitation and amplification for up to six channels. The excitation is selected by individual jumpers for each channel and ranges from 2.5 to 10 volts. The gain for each channel is also selectable by jumpers and ranges from 1000 to 4000. The output of the amplifier is ±10 VDC. The amplifier contains an auto-zero feature that allows for push button zeroing of the output of the load cell.

Figure 6. MSA-6 Mini Amplifier[10]

The output of the amplifier was connected to an analog-to-digital converter installed in a notebook computer. The system control software sampled the six channels of output of the load cell and the six positions of the gantry system at an operator selected frequency of 25 Hz.

5.1.3. United States Naval Academy Tests

Full scale model testing was performed at the United States Naval Academy Hydromechanics Laboratory shown in Figure 7. This set of tests included determining the forces and moments resulting from body angles in pitch and yaw and control surface angles. The towing tank used was 120 ft long, 8 ft wide, and 5 ft deep. The towing tank included a wave making machine, a wave absorbing beach and a moving carriage.[11]

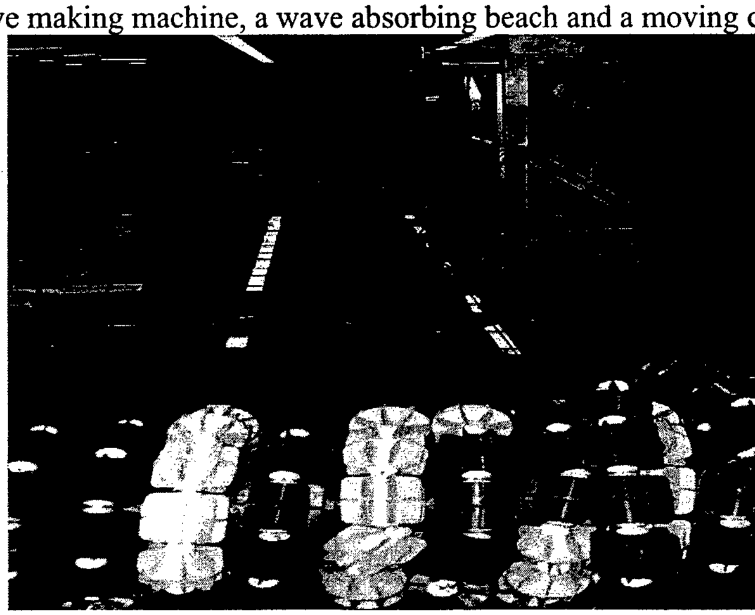

Figure 7. United States Naval Academy Hydromechanics Laboratory Towing Tank

5.1.4. Massachusetts Institute of Technology Tests

Small scale model testing was performed at the Marine Instrumentation and Computation Laboratory at the Massachusetts Institute of Technology. These experiments were performed in order to determine the forces and moments associated with unsteady motion. The model was moved in prescribed sinusoidal motions and the resultant forces and moments were measured.

The laboratory contains a tank and a gantry system capable of simultaneous motion in five degrees of freedom. The experimental tank is 10 m long, 4 m wide, and 1 m deep. The gantry system consists of five different motors and several gear assemblies to ensure smooth operation at the speeds and frequencies required for the experiments. The gantry system is computer controlled for precise positioning.

The testing and correction of the very sophisticated gantry system and control software occupied a significant amount of the time allocated for the performance of this research. The gantry and control software was designed and assembled by D'Ambra Technologies, the only firm known to the research supervisor to be capable of developing the system and the software. The original contract called for completion of the gantry system by January 2002.

Testing began in June 2002 and significant problems with the system and control software were soon identified. Correction of the problems introduced several weeks of delay. The cycle of problem identification and correction continued until very early in 2003. In this process the vertical axis controls were completely redesigned. The original stepper motors were found to be inadequate and were replaced by servo motors. The pitch mechanism was strengthened three times to be able to provide the desired frequency and amplitudes of oscillation. The gantry system and control software were believed to be reliable and accurate in early April 2003.

The research team also encountered problems related to unidentified faults in the force measurement system. AMTI conducted significant troubleshooting of the load cell and connections on several occasions to determine the cause of the abnormal readings. Some of the abnormal readings were attributed to a fault in the cabling and others were attributed to the high level of electronic noise in the long cables of the system. Eventually, all of the issues with the force measurement system were corrected.

One difficult issue with the force measurement system that was discovered late in the process is that the measured forces and moments often represent a very small fraction of the capability of the transducer. The manufacturer states that the transducer provides accurate results at very small fractions of its capability, but this needs experimental verification.

Several hundred experiments were conducted during the process of identifying and correcting system problems. Experiments were performed by the author, by an independent contractor, and by several undergraduate students acting with supervision. The data from these experiments needs to be closely examined to determine if the experiments are valid. The first future work that will be done is to predict what those tests should show and check for agreement. If the data are found to be valid, they will contribute to a more complete data set with less variance that will allow for better modeling of the vehicle behavior.

5.2. Design of Experiments

The variables that affect the behavior of an underwater vessel include the depth of the water, the submergence of the vessel, the forward velocity, and the frequency and amplitude of oscillation. The Central Composite Method was used for Design of Experiments in order to reduce the total number of experiments required.

The Central Composite, or Box-Wilson, Design is a three- or five-level design that includes the corner, center, and axial points of the design space. The three-factor Central Composite Design space is shown in Figure 8.

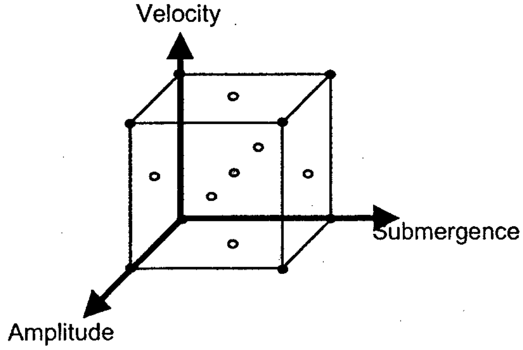

Figure 8. Central Composite Method

The three factor design space is developed from 15 point designs: a center point design, eight corner point designs, and 6 axial point designs. This model represents the response surface more accurately than most other methods since the corner points are included. Corner points represent the limits of the experimental space. However, attempting to reach these corner point designs may strain the engineering model[12].

The selection of test points in an incomplete matrix on the basis of orthogonal numeric functions is fine when the dependent variable depends linearly on the input variables. However, for things like a nonlinear relation between force coefficient and excitation frequency, it is better to be sure that all corners in the test space are tested so that the mathematical model will interpolate rather than extrapolate.

Several experiments that had been planned were not performed due to limitation of the gantry system at higher speeds and higher frequencies of oscillation. Other experiments were not performed due to physical constraints of the gantry system. Appendix A lists the full scale experiments that were performed. Appendix B lists the model scale experiments that were used for analysis. All submergences listed in the test matrices are to the center of the body.

A large number of other experiments were performed earlier in the test program, but uncertainty in the equipment behavior resulted in uncertainty in the quality of the data and data from those experiments was not used. That data will be reevaluated as part of future work.

5.3. Analysis of Experimental Data to Extract Measured Forces and Moments

For testing performed at MIT, the conversion from raw force, moment, and position data was performed using MATLAB routines developed by the author. The transducer provided data on forces and moments in the form of voltages for each channel that had to be converted to the MKS system. The gantry system provided data on the position of the system in a numerical format that had to be converted to the MKS system for analysis.

For testing performed at the United States Naval Academy, raw force and moment data were converted using routines developed the author and by LTJG Greg Sabra, USCG. The steady force results were determined in a manner very similar to the method used to compute steady force test results for MIT tests. The method for computing MIT test results will be discussed in a later section. A complete discussion of LTJG Sabra's code is contained in his thesis entitled ""Wave Effects on Underwater Vehicles in Shallow Water."

All of the MIT test conditions were listed in a common Excel file called "MIT Test Plan.xls". This file contains separate worksheets for each of the many series of tests that were performed. These worksheets look very similar to the table contained in Appendix B, with the addition of two columns at the left to record date and time information for each experiment. The worksheets in the MIT Test Plan file were used by the MATLAB routines to determine which data files to analyze and what some of the test conditions were for each experiment. The test conditions obtained from the test plan were the water depth and submergence of the vehicle during the test. All other test conditions were extracted directly from the test data file.

5.3.1. User Interface and Data File Management

The user interface and the file management were performed by a MATLAB routine called "AutoanalyzeXls.m". This file is contained as Appendix C. After the user starts this program, the user selects the series of experiments to be analyzed by entering the number corresponding to the desired series. All series that have been performed are listed, even those that are suspected to be of little value. After the test series has been selected, the program imports the list of experiments and the depth and submergence information. The program then calls other MATLAB routines to analyze the data files. For steady force tests, the analysis program is "AnalyzemodXlsSF.m". For all other tests, the analysis program is "AnalyzemodXls.m".

The analyses were performed using the data files recorded by the notebook computer in Excel format. An example of a data file is included as Appendix D.

5.3.2. Steady Force Data Analysis

Steady force tests at MIT involved towing the vehicle down the tank with a steady angle of yaw or pitch. These tests were analyzed using "AnalyzemodXlsSF.m", contained in Appendix E. "AnalyzemodXlsSF.m" starts by importing the data file identified by "AutoanalyzeXls.m". The program determines the ordered parameters and the date and time at which the experiment occurred. Then, the file eliminates the first 1.2 seconds of data to allow for gantry acceleration and any data recorded after the gantry velocity returns to zero at the end of the test. The remaining position data is converted to the MKS system using a conversion factor. The remaining force and moment data undergo a more detailed analysis.

The voltage output of the transducer is converted to forces and moments using

$$F = \frac{V_{out}}{Gain \times V_{exc} \times S \times 10^{-6}} \tag{14}$$

where

F is the calculated force or moment

V_{out} is the output voltage recorded by the computer

Gain is the gain of that channel in the amplifier

V_{exc} is the excitation voltage of the channel, and

S is the sensitivity of the channel.

The calculation of equation (14) is performed using matrices so that the effects of cross-talk in the transducer can be accounted for.

The mean force is calculated by taking the mean of the forces measured in the data interval and shifting the origin of the mean force from the origin of the load cell to the origin of the vessel coordinate system, defined to be at the midships of the vessel. The origin shift was done using

$$\begin{aligned} M_{midships} &= M_{transducer} - Zx_{transducer} \\ N_{midships} &= N_{transducer} + Yx_{transducer} \end{aligned} \tag{15}$$

The origin-shifted mean forces and moments were written to a common output file that contained the mean force and moment data for all analyzed experiments. This output file and explanatory notes are included as Appendix F.

5.3.3. Analysis of Experiments Involving Unsteady Motion

The analysis of experiments involving unsteady motion was performed using "AnalyzemodXls.m" called by "AutoanalyzeXls.m". The code is included as Appendix G. This is the most complicated of the codes used for this research and, as a result, is the most heavily commented.

The first section of the code identifies and defines most of the variables used in the code. Next, the code initializes by reading the data file and gathering some basic information about the parameters of the experiment. The last four lines of the data file contain information about the ordered frequency and amplitudes of oscillation as well as sample frequency, velocity, and travel duration and distance. Then, the actual sample frequency is calculated by taking the inverse of the average interval between data points according to

$$f_{sample} = \frac{1}{mean(\Delta t)} \tag{16}$$

The ordered sample frequency was always 25 Hz, but for a certain period of time during the research errors in the control software resulted in data being taken at other frequencies.

The code drops the first 1.2 seconds of data to allow for the acceleration of the gantry. The time interval to drop was chosen short enough to allow sufficient time remaining to have several periods of oscillation remaining but long enough to remove the majority of the acceleration period. The code also drops data recorded after the vessel completed its travel along the tank. This was necessary because the control software continued to collect data until the ordered time period of the experiment was completed, whether or not the travel distance had been accomplished. Setting the time period of travel too short resulted in sudden stops of the gantry causing large accelerations on both the vehicle and the gantry system. Ordered durations were made longer than absolutely necessary to prevent this mechanical shock to the system and prolong the life of the apparatus.

The analysis code determines the frequency of oscillation by checking the input parameters in the data file to determine the ordered frequency of oscillation. This information is used to ensure that the data to be analyzed consisted of an integer number of wavelengths of the oscillation. This feature was absolutely necessary to get highly accurate results from the Fourier analysis that takes place later in the program. The period of a cycle is given by

$$period = \frac{1}{frequency}$$

The duration of data recorded was found by taking the difference in time between the first and last remaining data points. The number of periods recorded is

$$\text{number of periods} = \frac{duration}{period}$$

The number of data points retained for Fourier analysis is found by rounding down to the next integer the product of sample frequency, period, and the number of periods according to

$$\text{\# of data points} = round(f_{sample} * period * floor(\text{number of periods}))$$

"round" is a MATLAB function that rounds the element to the nearest integer. "floor" is a MATLAB function that round the element to the next lower integer.

Once an integer number of data points is established it the mean force and moments are subtracted from all measured forces and moments in order to remove steady effects.

Next, the voltages from the transducer are converted to forces and moments using equation (14) and the numeric position data is converted to metric system position data using known relationships between the controller data and gantry motion.

The force, moment and location data are conditioned by the program in preparation for the Fourier analysis. With the position data in metric format, the program translates the position data from the location of the strut to the vehicle midships. For linear motion, the motion of the strut forward of midships represents the motion of midships. For angular motion, this is not the case. The effect of angular motion on the x,y,and z position of midships is calculated by

$$\begin{aligned} y_{midships} &= y_{strut} - \text{distance}\sin\left(\psi\frac{\pi}{180}\right) \\ x_{midships} &= x_{strut} + L_{strut}\sin\left(\theta\frac{\pi}{180}\right) \\ z_{midships} &= z_{strut} - L_{strut}\left(1-\cos\left(\theta\frac{\pi}{180}\right)\right) \end{aligned} \tag{17}$$

where

θ is the pitch angle

ψ is the yaw angle

distance is the distance along the x axis from the strut to midships.

L_{strut} is the length of the strut arm from its pivot point to the vehicle.

The forces and moments are shifted from having their origin at the transducer to having their origin at midships using equation (15). Also, the data is interpolated into even intervals of exactly 0.4 seconds. The mean value of position for each channel except the x position is subtracted to remove any bias in the position data. Then, the data matrix is padded with zeros to obtain exactly 2048 data points.

The most precise Fourier transformation requires the data to have an integer number of periods of the waveform and the frequency of signal to be analyzed must be a multiple of the fundamental frequency of the data sample. For an interval of 2048 data points being sampled at 25 Hz, the fundamental frequency is

$$f_{fundamental} = \frac{f_{sample}}{\text{number of points}} = \frac{25Hz}{2048} = 0.012207 \quad (18)$$

The frequencies of oscillation used for this research are 0.402831, 0.79346, and 1.19629 Hz representing 33, 65 and 98 times the fundamental frequency of the analysis. The sample frequency is assured by interpolating the data into exact time intervals between data points. As a result, the force, moment, and position data relating to oscillation are readily extracted using Fourier analysis.

The first step in the Fourier analysis is to begin to build the data matrix by constructing the frequency column. The first row is assigned a frequency of zero Hz and each successive row is assigned a frequency of the row number multiplied by the fundamental frequency. The fast Fourier transformation is applied to the force, moment, and position data. To compensate for the zero padding added earlier, the value of each of the Fourier coefficients is multiplied by the ratio of the padded size to the unpadded size.

After the Fourier transformation occurs, each coefficient has both a magnitude and phase associated with it. The phase of each of the coefficients is changed to make it relative to the phase of the motion that produced the force. This is done by multiplying every coefficient by $e^{-i\varphi}$ where φ is the phase angle of the driving motion at that frequency.

Low pass filters were installed on all of the force data collection channels to reduce the effects of electronic noise in the system. The effects of these filters is removed from each channel of force and moment using

$$\eta_{unfiltered} = \eta_{filtered} e^{-i\tan^{-1}(\omega CR)} \quad (19)$$

where $\eta_{unfiltered}$ is the amplitude of the signal with filtering effects removed

$\eta_{filtered}$ is the amplitude of the signal after filtering

ω is the frequency of oscillation

C is the capacitance of the filter, and

R is the resistance of the filter.

The effects of filtering at the frequencies of oscillation altered the magnitude by less than one percent and the phase angle by less than five degrees.

The program determines the frequency of motion closest to the ordered frequency of oscillation for all motions. The actual amplitudes of motion and forces are converted back into the time domain from the frequency domain using

$$\eta(t) = \sqrt{2 * 2\left(\frac{|\eta(\omega)|}{2048}\right)^2} \quad (20)$$

where $\eta(t)$ is the amplitude in the time domain

$\eta(\omega)$ is the amplitude in the frequency domain, and

2048 is the number of data points used in the analysis.

The actual amplitudes of motion are used to calculate the inertial forces experienced by the vehicle using equation (2). The inertial forces are then subtracted from the measured forces and moments to leave only the hydrostatic forces and moments remaining. The mass, inertia, and center of gravity terms used to calculate the inertial forces and moments were derived by performing oscillation tests in air. The model was filled with water to ensure that the effective mass of the vehicle was the same as it would be if the vehicle were in water. This guaranteed that the only significant forces measured were the inertial forces. The validity of the process was checked by performing oscillation tests in air and subtracting the calculated inertial components. The results of these inertial calculation checks are included as Appendix H.

At this point in the program all of the hydrodynamic forces and moments at the frequencies of oscillation and their phases relative to the driving motion are determined. The results are conditioned in order to have positive amplitudes and have magnitudes of the phase angles less than 180 degrees.

The program has also determined the forces and moments at twice and three times the oscillation frequency. In all cases the forces and moments at multiples of the oscillation frequency have magnitudes of approximately 10% or less of the forces and moments at the oscillation frequency. This indicated the response of the complete hydrodynamic system is linear and that non-linear forces and moments can safely be neglected in analyzing vehicle dynamics or in designing control systems. This also indicates that problems such as hysteresis in the load cell o-rings were unlikely.

The results are written to an output file and contain all of the test information as well as the actual amplitudes and frequencies of oscillation and the amplitudes and frequencies of all six forces and moments. The phase angle between the force or motion and the driving is also listed. The output also includes the frequency, amplitude, and phase information for the second and third harmonics forces and moments. A partial example of the output file is contained in Appendix I. A complete output file consists of one row of 103 columns for each file analyzed.

5.3.4. Curve Fitting the Experimental Results

The following sections describe how the force and moment results were analyzed and present the resulting maneuvering coefficients. The effects of submergence and speed on the maneuvering coefficients will be analyzed. Part of that analysis will involve curve fitting the data to determine the functional relationships involved. The program used to do the curve fitting is a MATLAB routine generated by the author called "CoeffSolver.m". The code is included as Appendix J.

"CoeffSolver.m" uses user-coded values of the coefficients and the test parameters of Length/Submergence and Froude Number to perform a least squares regression of the data. When sufficient test data is present a second order equation is derived. When there is not sufficient data for a second order equation a first order equation is used.

In general, the set of equations is of the form

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
$$

which can be written in matrix form

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

or

$$\overline{\overline{A}}\overline{x} = \overline{B}$$

where

A is an m x n matrix
x is an array of size m
B is an array of size m.

Matrix A represents the known parameters of the equations such as the speed and submergence of the vehicle for the experiment. Also, the terms a_{i1} usually equal 1 to allow for some constant to be built into the equation. The B array represents the measured or calculated data points. The array x represents the coefficients that are calculated to best represent the data using a least squares linear regression. The least squares regression calculates the elements of array x that will minimize the sum of the squares of the errors between the predicted and the calculated values. This manipulation is performed easily in MATLAB using $x = A \backslash B$.

To measure the quality of fit, the program finds the root mean square of the difference between the predicted and the measured coefficients. The code uses

$$Difference = \frac{(Predicted - Actual)}{Actual}$$

and

$$Difference_{rms} = \frac{norm(Difference)}{sqrt(\#\text{ of elements})}$$

where the norm of the Difference array is the largest singular value in the Difference array.

The output of the Coefficient Solver Program is included as

5.4. Analysis of Experimental Results

5.4.1. Rudder and Stern Planes Effects

The effects of the rudder and stern planes were examined by performing tests at the United States Naval Academy with the control surfaces at no angle and with the control surfaces deflected to seven degrees. The results were normalized by subtracting the lift and moment at zero degrees from the lift and moment measured with control surface deflection to remove any imbalance in loading. The lift coefficient per degree of fin deflection was calculated according to

$$C_{L\delta} = \frac{Lift}{\frac{1}{2}\rho U^2 A_{fin}\delta} \tag{21}$$

where $C_{L\delta}$ is the lift coefficient per degree, Lift is the measured lift force, U is the vehicle forward velocity, A_{fin} is the effective area of the rudder, and δ is the control surface angle. In order to provide more data points from the same experiments, the measured moments were converted into pseudo-lift forces by dividing the moments by the length of the control surface moment arm, 0.7 m. These were then converted into coefficients using equation (21).

For high aspect ratio wings, with aspect ratios greater than five, the theoretical approximation found by Hoerner[13] is

$$C_{L\alpha} = \frac{dC_L}{d\alpha} = \left[10 + \frac{20}{AR_e}\right]^{-1} \tag{22}$$

where AR_e is the aspect ratio found by

$$AR_e = \left(\frac{Span^2}{Area}\right) \tag{23}$$

The area was estimated by combining the calculated area of each of the fins with the estimated effective area provided by the body between the fins. Figure 9 shows the control surface geometry. Using an area of 0.02 m^2 and span of 0.254 m, the aspect ratio is 3.23, and C_L is 0.0618/Degree.

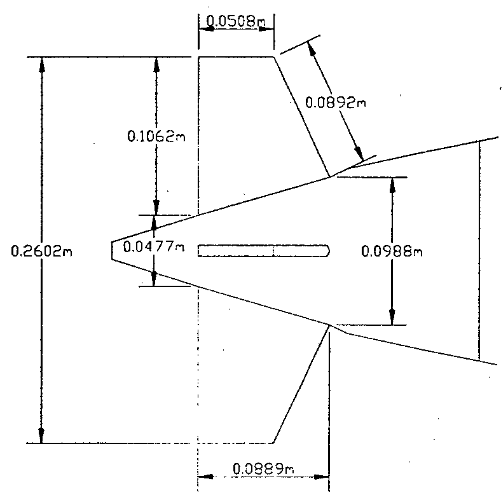

Figure 9. Control Surface Geometry

For aspect ratios between three and five Hoerner recommends using

$$C_{L\alpha} = \frac{dC_L}{d\alpha} = \left[10 + \frac{10}{AR_e^2} + \frac{26}{AR_e}\right]^{-1} \tag{24}$$

which yields C_L of 0.053/Degree.[14]

Figure 10 shows the measured rudder lift coefficients plotted against the ratio of body length to submergence. The figure also shows the linear approximation to the data and the theoretical value for intermediate aspect ratio fins in an infinite fluid. The data for Figure 10 are presented in Table 3.

The linear approximation to the data is

$$C_{L\delta} = -0.0013\tfrac{Length}{Submergence} + 0.0518 \tag{25}$$

The value of the linear approximation at deep submergence is 0.0518/Degree, very near the theoretical value of 0.053/Degree calculated using equation (24).

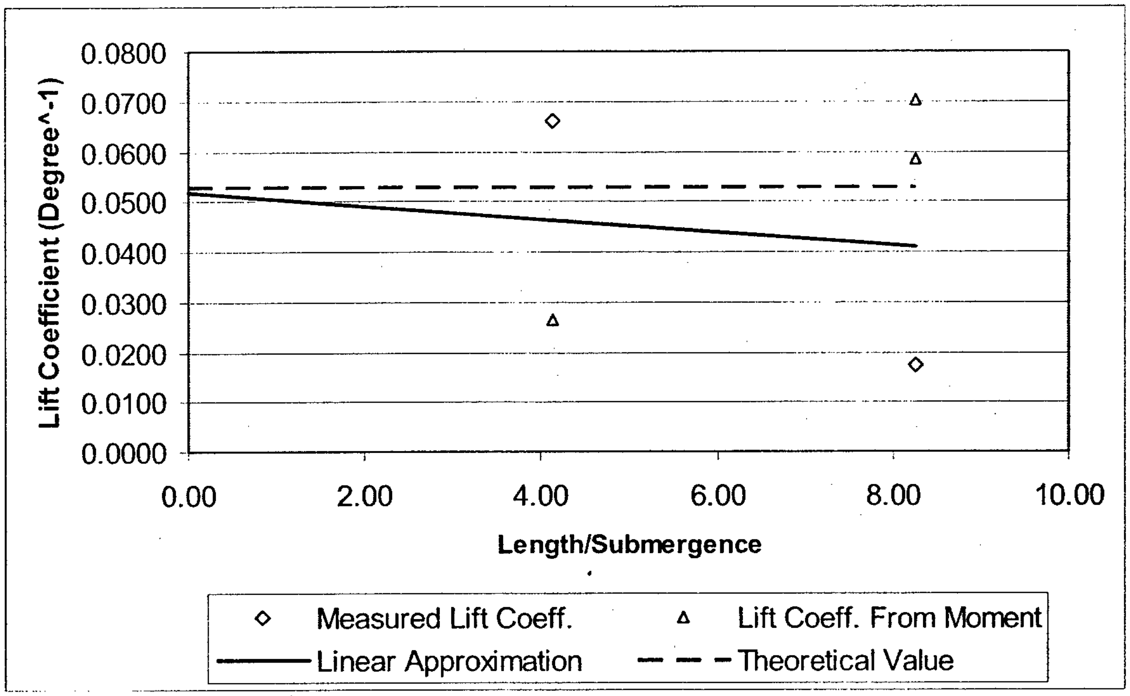

Figure 10. Rudder Lift Coefficient for Various Values of Length/Submergence

Table 3. Rudder Lift Coefficient Data

	Test	Speed	L/Subm.	Rudder Angle	Rudder Lift	Normalized Rudder Lift	Rudder Lift Coeff	Yaw Moment	Normalized Yaw Moment	Yaw Moment Coeff
		m/s		Degrees	N	N	1/Degree	N-m	N-m	1/Degree
Measured Lift	N1	2.06	2.07	0	-0.915	0.000		0.333	0	
	N6	0.515	4.14	0	-0.259	0.000		0.302	0.000	
	N7	1.03	4.14	0	-1.729	0.000		1.336	0.000	
	N8	2.06	4.14	0	-7.349	0.000		6.402	0.000	
	N9	2.06	4.14	7	12.267	19.616	0.0662	-10.914	-17.316	0.0825
	N15	0.515	8.26	0	0.280	0.000		0.196	0.000	
	N16	2.06	8.26	0	2.810	0.000		1.512	0.000	
	N22	2.06	8.26	7	8.001	5.191	0.0175	-10.594	-12.106	0.0577
	N24	2.06	8.26	7	7.922	5.112	0.0172	-13.083	-14.595	0.0695
Lift Calculated from Moment	N9m	2.06	4.14	7		-7.832	0.0264			
	N22m	2.06	8.26	7		-17.294	0.0583			
	N24m	2.06	8.26	7		-20.850	0.0703			

The effects of stern planes were determined in a manner similar to those of the rudder with similar results. Figure 11 shows the measured Stern Planes Lift Coefficients plotted against the ratio of body length to submergence. Figure 11 also shows the linear approximation to the data and the calculated theoretical value. The data for Figure 11 are included as Table 4.

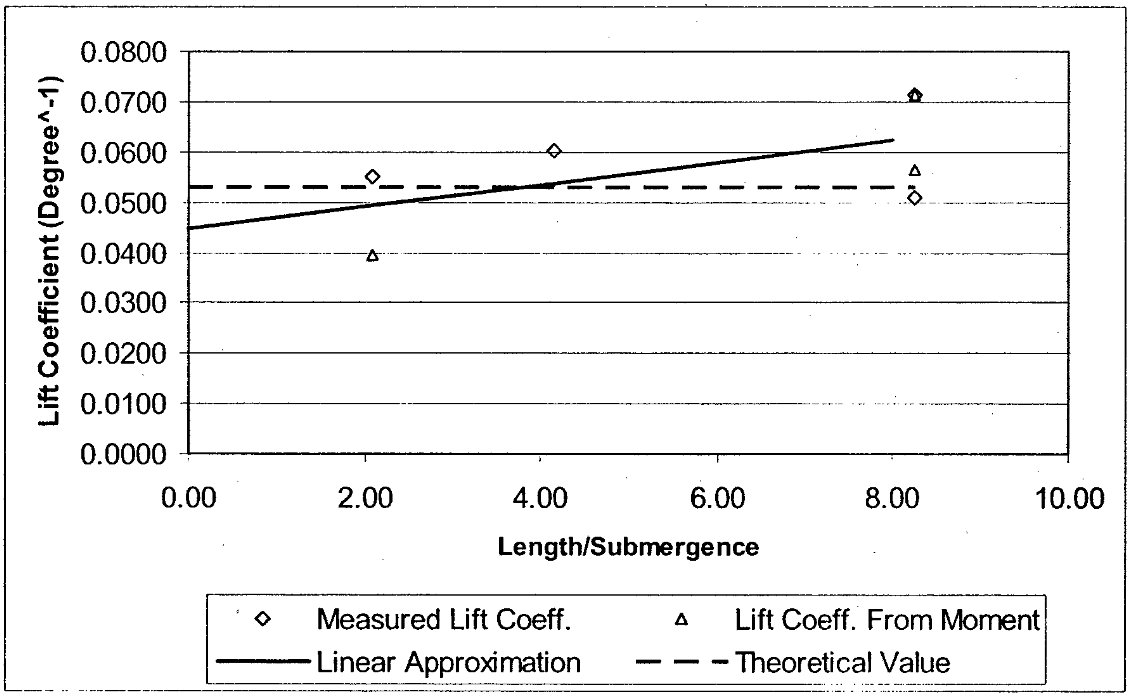

Figure 11. Stern Planes Lift Coefficient for Various Values of Length/Submergence

Table 4. Stern Planes Lift Coefficient Data

	Test	Speed	L/Subm.	Stern Planes Angle	Stern Planes Lift	Normalized Stern Planes Lift	Stern Planes Lift Coeff	Pitch Moment	Normalized Pitch Moment	Pitch Moment Coeff
		m/s		Degrees	N	N	1/Degree	N-m	N-m	1/Degree
Measured Lift	N1	2.06	2.07	0	2.850	0.000		-3.227		
	N4	2.06	2.07	7	19.131	16.281	0.0549	4.982	8.209	0.0391
	N6	0.515	4.14	0	0.210	0.000		-0.157		
	N7	1.03	4.14	0	0.394	0.000		-0.778		
	N8	2.06	4.14	0	4.556			-6.761		
	N10	2.06	4.14	-7	-13.273	-17.829	0.0601	-8.460	-1.699	0.0081
	N15	0.515	8.26	0	0.337	0.000		-0.162		
	N16	2.06	8.26	0	4.034	0.000		-8.868		
	N20	2.06	8.26	-7	-11.085	-15.119	0.0510	-20.558	-11.690	0.0557
	N24	2.06	8.26	-7	-17.080	-21.114	0.0712	-23.650	-14.782	0.0704
Lift Calculated from Moment	N4m	2.06	2.07	7		11.727	0.0396			
	N20m	2.06	8.26	-7		-16.700	0.0563			
	N24m	2.06	8.26	-7		-21.117	0.0712			

These results seem to show that the rudder has a slightly reduced effect when near the surface and the stern planes have a slightly greater effect when near the surface. In both cases, the linear approximation of the data closely approaches the theoretical value for a submerged body in an infinite fluid. The opposite slopes of the linear approximations are not explainable at this time. The small variations shown in the coefficients over the range of submergences monitored show that the effect of submergence is very small for submergences greater than 10% of body length.

5.4.2. Sway Force and Yaw Moment Due to Sway Motion

Experiments to determine the effects of sway motion were performed in various combinations of submergence, velocity, frequency and amplitude of oscillation. Table 5 contains the test conditions and key results.

Table 5. Test Conditions and Results for Sway Force and Yaw Moment Due to Sway Motion

Sway Motion				Hydrodynamic Sway Force, Y							
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(Y)	Yvdot	Yvdot'	Im(Y)	Yv	Yv'
m	m/s	Hz	m	N	Degrees	N	kg		N	kg/s	
0.543	0.333	0.40283	0.1	2.217	-32.6	1.8676	-2.9063	-0.0175	-1.1944	-4.7044	-0.0589
0.252	1.000	0.40283	0.1	2.641	-39.9	2.0257	-3.1557	-0.0190	-1.6937	-6.6784	-0.0278
0.398	0.667	0.79346	0.1	8.644	-35.7	7.0193	-2.7725	-0.0167	-5.0439	-9.9323	-0.0621
0.543	1.000	0.40283	0.1	2.677	-38.1	2.1065	-3.2812	-0.0197	-1.6517	-6.5120	-0.0271
0.252	0.333	0.40283	0.1	2.216	-32.3	1.8730	-2.9156	-0.0175	-1.1841	-4.6652	-0.0584

Sway Motion				Hydrodynamic Yaw Moment, N							
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(N)	Nvdot	Nvdot'	Im(N)	Nv	Nv'
m	m/s	Hz	m	N-m	Degrees	N-m	kg		N-m	kg/s	
0.543	0.333	0.40283	0.1	0.085	-45.4	0.0600	-0.09331	-0.00081	-0.06081	-0.23950	-0.00432
0.252	1.000	0.40283	0.1	0.351	-81.6	0.0513	-0.07986	-0.00069	-0.34714	-1.36874	-0.00823
0.398	0.667	0.79346	0.1	0.352	-55.8	0.1976	-0.07806	-0.00068	-0.29080	-0.57264	-0.00516
0.543	1.000	0.40283	0.1	0.371	-80.1	0.0638	-0.09941	-0.00086	-0.36567	-1.44166	-0.00867
0.252	0.333	0.40283	0.1	0.085	-43.3	0.0617	-0.09607	-0.00083	-0.05816	-0.22914	-0.00414

The hydrodynamic forces and moments are determined by subtracting the inertial forces and moments from the measured forces and moments for each experiment. For pure sway motion, the applicable equations of motion from equation (9) are

$$\begin{aligned} Y &= Y_v v + Y_{\dot{v}} \dot{v} \\ N &= N_v v + N_{\dot{v}} \dot{v} \end{aligned} \tag{26}$$

The notation of complex equations is used to separate the components of the measured force and moment into the component related to velocity, a damping component, and the component related to acceleration, an added mass component. This is done using

$$\begin{aligned} Y &= \mathrm{Re}(Y) + i\,\mathrm{Im}(Y) = Y\cos(\phi) + iY\sin(\phi) \\ N &= \mathrm{Re}(N) + i\,\mathrm{Im}(N) = N\cos(\phi) + iN\sin(\phi) \end{aligned} \tag{27}$$

and understanding that

$$\begin{aligned} \mathrm{Re}(Y) &= Y\cos(\phi) = Y_{\dot{v}}\dot{v} \\ \mathrm{Im}(Y) &= Y\sin(\phi) = Y_v v \\ \mathrm{Re}(N) &= N\cos(\phi) = Y_{\dot{v}}\dot{v} \\ \mathrm{Im}(N) &= N\sin(\phi) = Y_v v \end{aligned} \tag{28}$$

where ϕ is the phase angle taken by subtracting the phase angle of the force from the phase angle of the driving motion.

Sinusoidal motions can be described by

$$\eta = \overline{\eta} e^{i\omega t} \tag{29}$$

where η is a time varying position

$\overline{\eta}$ is the amplitude of the sinusoidal oscillation, and

ω is the frequency of oscillation.

The velocity component is

$$\frac{\partial \eta}{\partial t} = \omega A e^{i\omega t} \tag{30}$$

and the acceleration component is

$$\frac{\partial^2 \eta}{\partial t^2} = -\omega^2 A e^{i\omega t} \tag{31}$$

In the standard method of notation the exponential term is understood and velocity is represented by ωA and the acceleration term is represented by $-\omega^2 A$.
Therefore,

$$\begin{aligned} v &= \omega A \\ \dot{v} &= -\omega^2 A \end{aligned} \tag{32}$$

By combining equations (28) and (32), we obtain the necessary equations to compute the hydrodynamic forces and moments due to sway according to

$$\begin{aligned} Y_v &= \frac{\mathrm{Im}(Y)}{\omega A} \\ Y_{\dot{v}} &= \frac{\mathrm{Re}(Y)}{-\omega^2 A} \\ N_v &= \frac{\mathrm{Im}(N)}{\omega A} \\ N_{\dot{v}} &= \frac{\mathrm{Re}(N)}{-\omega^2 A} \end{aligned} \tag{33}$$

The results are non-dimensionalized according to the method of Section 4.

5.4.3. Heave Force and Pitch Moment Due to Heave Motion

Experiments to determine the effects of heave motion were also performed in various combinations of submergence, velocity, frequency and amplitude of oscillation. Table 6 contains the test conditions and key results.

Table 6. Test Conditions and Results for Heave Force and Pitch Moment Due to Heave Motion

Heave Motion				Hydrodynamic Heave Force, Z							
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(Z)	Zwdot	Zwdot'	Im(Z)	Zw	Zw'
m	m/s	Hz	m	N	Degrees	N	kg		N	kg/s	
0.488	0.333	0.40283	0.1	2.406	-31.8	2.0452	-3.1878	-0.0192	-1.2681	-5.0026	-0.0626
0.272	1.000	0.40283	0.1	3.005	-35.8	2.4370	-3.8243	-0.0230	-1.7576	-6.9812	-0.0291
0.398	0.667	0.79346	0.1	8.722	-36.9	6.9747	-2.7856	-0.0167	-5.2367	-10.4271	-0.0652
0.488	1.000	0.40283	0.1	2.687	-34.2	2.2220	-3.4543	-0.0208	-1.5100	-5.9418	-0.0248
0.272	0.333	0.40283	0.1	2.633	-28.9	2.3054	-3.6077	-0.0217	-1.2727	-5.0408	-0.0631

Heave Motion				Hydrodynamic Pitch Moment, M							
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(M)	Mwdot	Mwdot'	Im(M)	Mw	Mw'
m	m/s	Hz	m	N-m	Degrees	N-m	kg		N-m	kg/s	
0.488	0.333	0.40283	0.1	0.333	129.000	-0.2098	0.32703	0.00284	0.25910	1.02217	0.01845
0.272	1.000	0.40283	0.1	0.395	111.900	-0.1472	0.23097	0.00200	0.36612	1.45422	0.00874
0.398	0.667	0.79346	0.1	0.966	148.400	-0.8223	0.32843	0.00285	0.50591	1.00733	0.00908
0.488	1.000	0.40283	0.1	0.450	97.400	-0.0579	0.09008	0.00078	0.44615	1.75554	0.01055
0.272	0.333	0.40283	0.1	0.293	144.200	-0.2372	0.37125	0.00322	0.17110	0.67769	0.01223

The analysis of the results in heave motion follows the same train of reasoning as described previously for sway motion. The equations for heave motion become

$$\begin{aligned} Z &= Z_w w + Z_{\dot{w}} \dot{w} \\ M &= M_w w + M_{\dot{w}} \dot{w} \end{aligned} \tag{34}$$

and

$$
\begin{aligned}
Z_w &= \frac{\mathrm{Im}(Z)}{\omega A} \\
Z_{\dot{w}} &= \frac{\mathrm{Re}(Z)}{-\omega^2 A} \\
M_w &= \frac{\mathrm{Im}(M)}{\omega A} \\
M_{\dot{w}} &= \frac{\mathrm{Re}(M)}{-\omega^2 A}
\end{aligned} \tag{35}
$$

5.4.4. Discussion of Heave and Sway Motion Results

The results of these experiments are in agreement with the expected results. Because the sway and heave motions with forward velocity create effective angles of attack of the body, the bow and stern both experience lift force opposed to v and w, therefore Y_v and Z_w are always negative. The terms $Y_{\dot{v}}$ and $Z_{\dot{w}}$ are always negative and have a magnitude approximately equal to the displacement of the vessel. N_v and M_w are usually negative, but can become positive if the rudder or stern planes are very large. $N_{\dot{v}}$ and $M_{\dot{w}}$ are usually relatively small quantities of uncertain sign.[15] The results shown in Table 5 and Table 6 match the expected values.

The results contained in Table 5 and Table 6 are graphically presented in Figure 12 through Figure 27. The results have been made non-dimensional according to the method described in section 4.

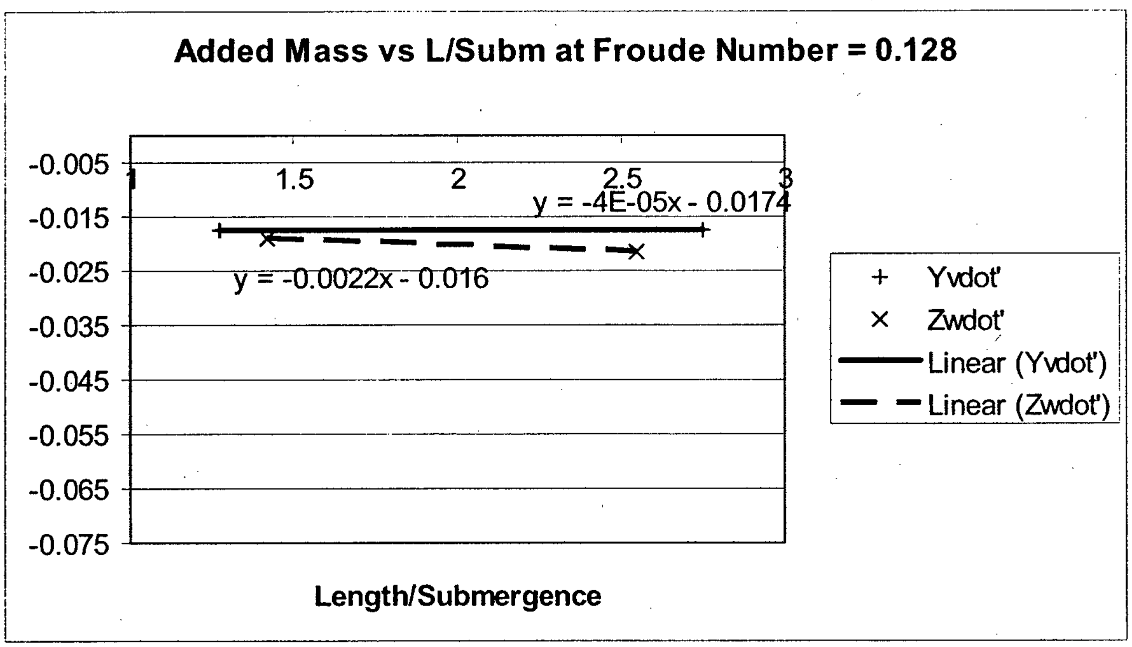

Figure 12. $Y_{\dot{v}}'$ and $Z_{\dot{w}}'$ vs L/Subm at Froude Number = 0.128

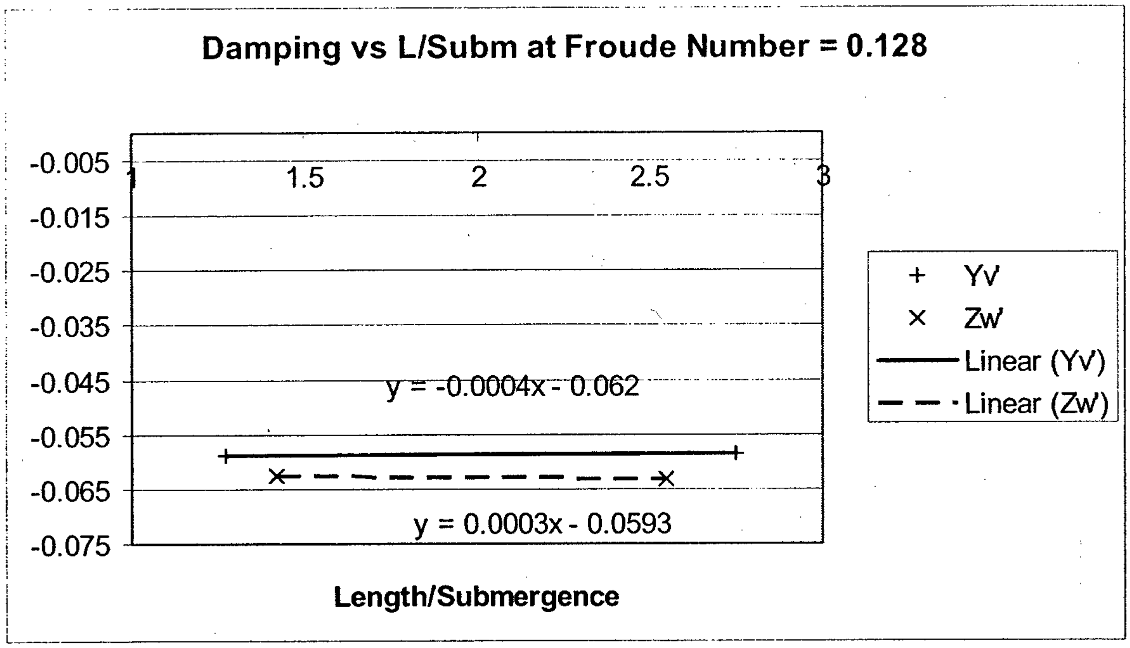

Figure 13. Y_v' **and** Z_w' **vs L/Subm at Froude Number = 0.128**

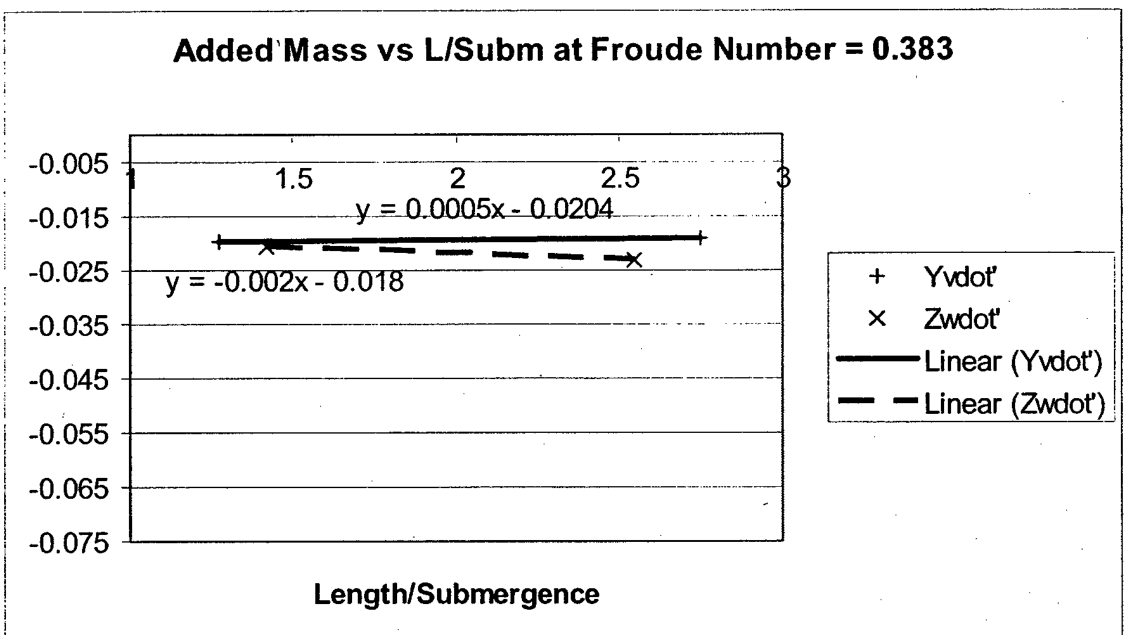

Figure 14. $Y_{\dot{v}}'$ **and** $Z_{\dot{w}}'$ **vs L/Subm at Froude Number = 0.383**

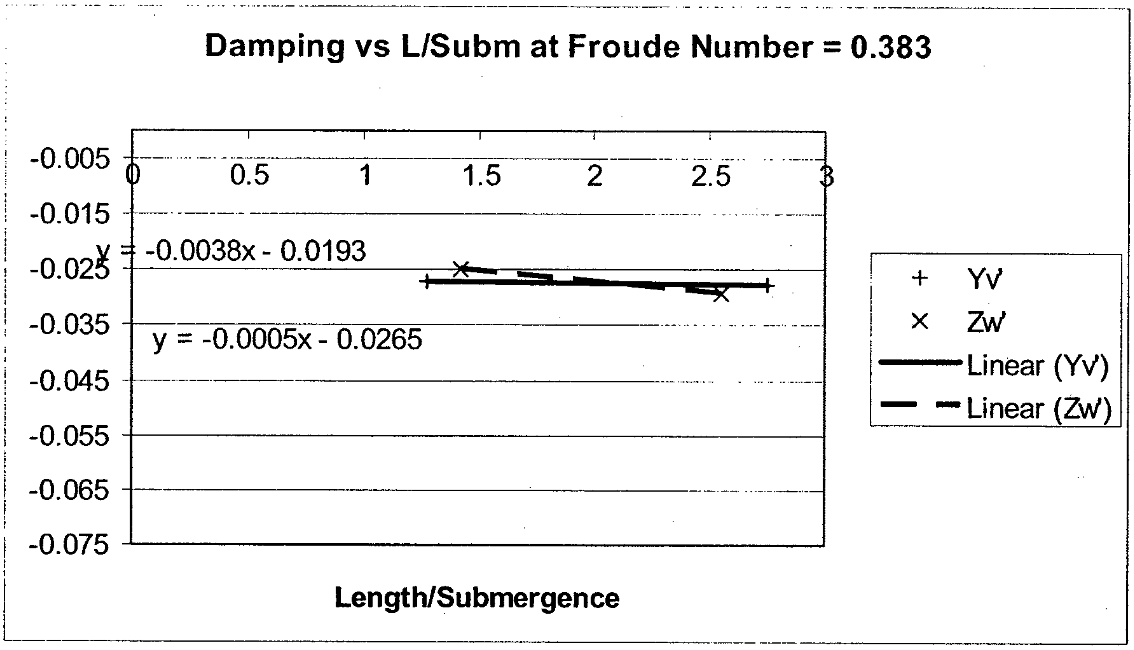

Figure 15. Y_v' and Z_w' vs L/Subm at Froude Number = 0.383

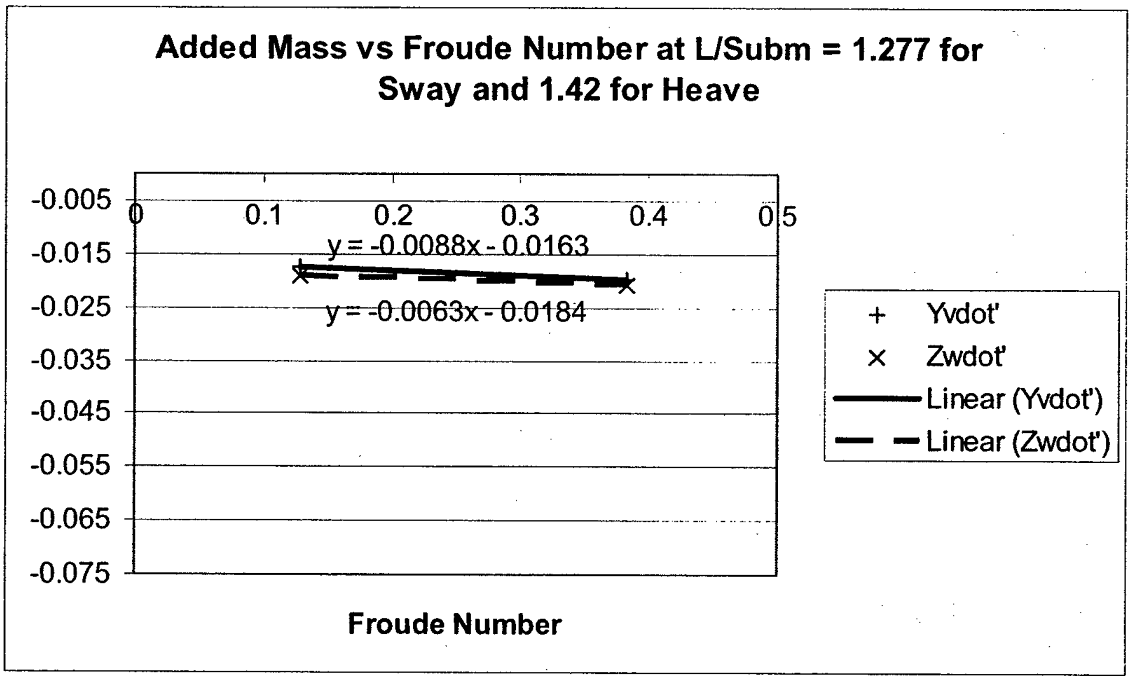

Figure 16. $Y_{\dot{v}}'$ and $Z_{\dot{w}}'$ vs Froude Number at L/Subm = 1.277 for Sway and 1.42 for Heave

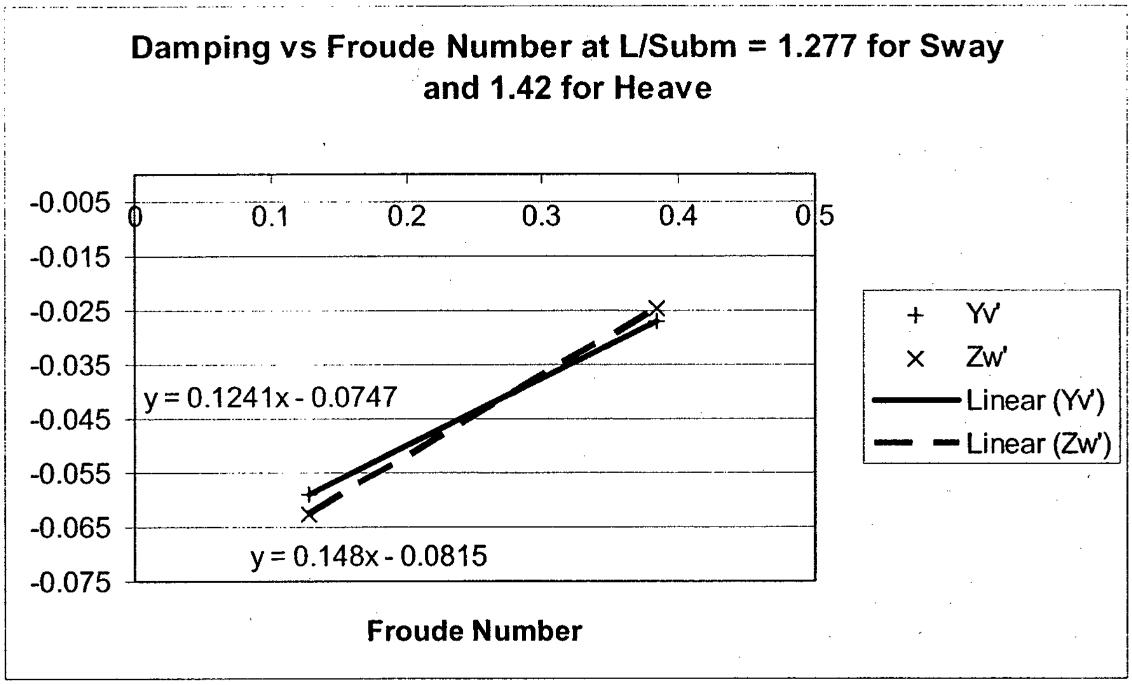

Figure 17. Y_v' and Z_w' vs Froude Number at L/Subm = 1.277 for Sway and 1.42 for Heave

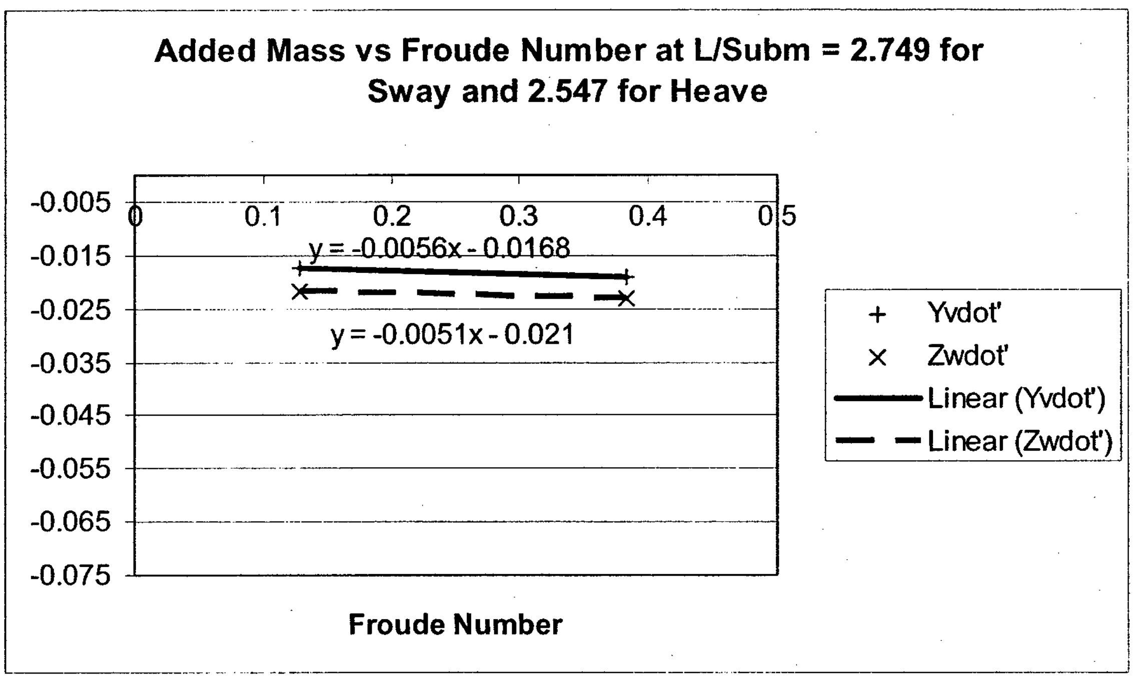

Figure 18. $Y_{\dot{v}}'$ and $Z_{\dot{w}}'$ vs Froude Number for L/Subm = 2.749 for Sway and 2.547 for Heave

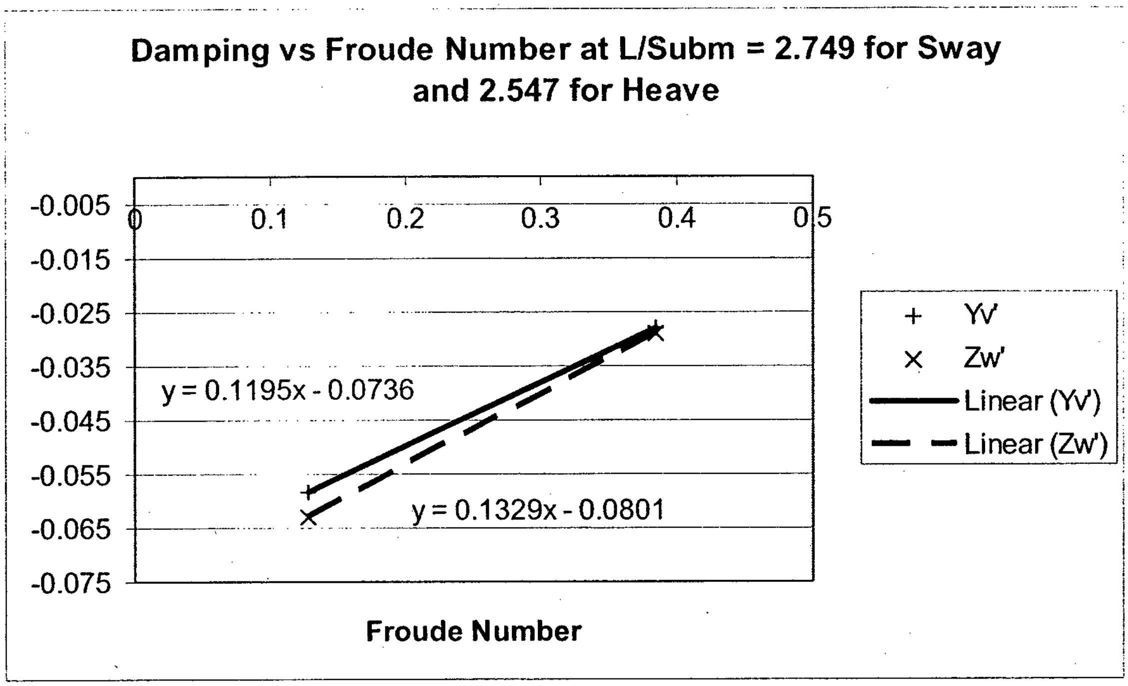

Figure 19. Y_v' **and** Z_w' **vs Froude Number for L/Subm = 2.749 for Sway and 2.547 for Heave**

Figure 12 through Figure 15 show that the effect of submergence on the direct added mass and damping terms for sway and heave motion is nearly negligible. Figure 16 through Figure 19 show that there is a significant effect of speed on these terms. These figures also show that the direct terms in sway and heave behave very similarly. The cross-term added mass and damping coefficients shown in Figure 20 through Figure 27 have different patterns of behavior

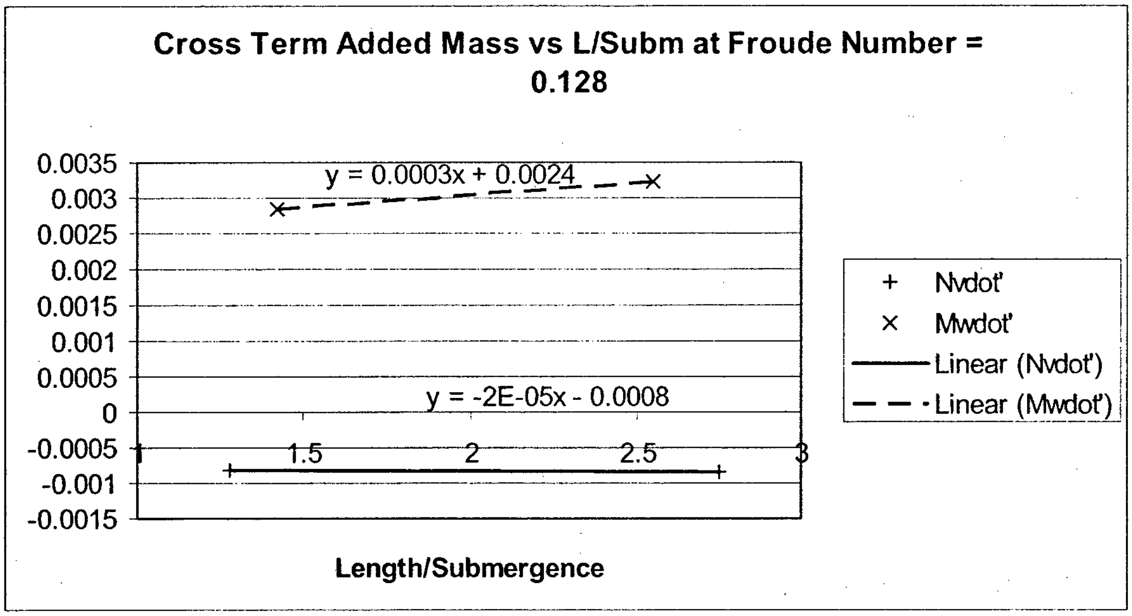

Figure 20. $N_{\dot{v}}'$ **and** $M_{\dot{w}}'$ **vs L/Subm for Froude Number = 0.128**

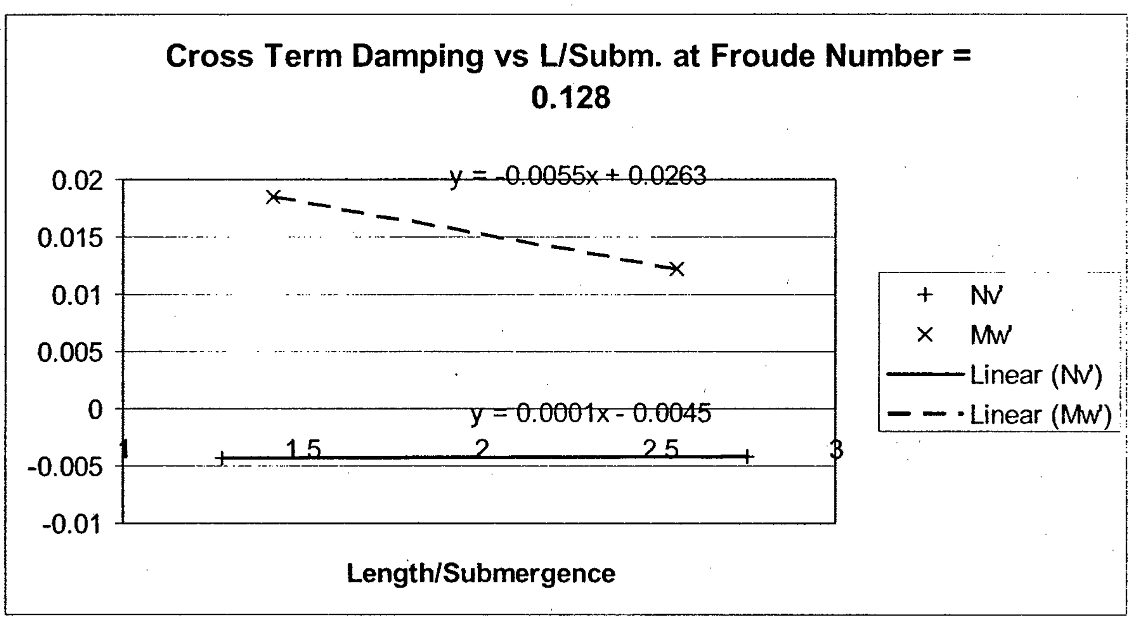

Figure 21. N_v' **and** M_w' **vs L/Subm for Froude Number = 0.128**

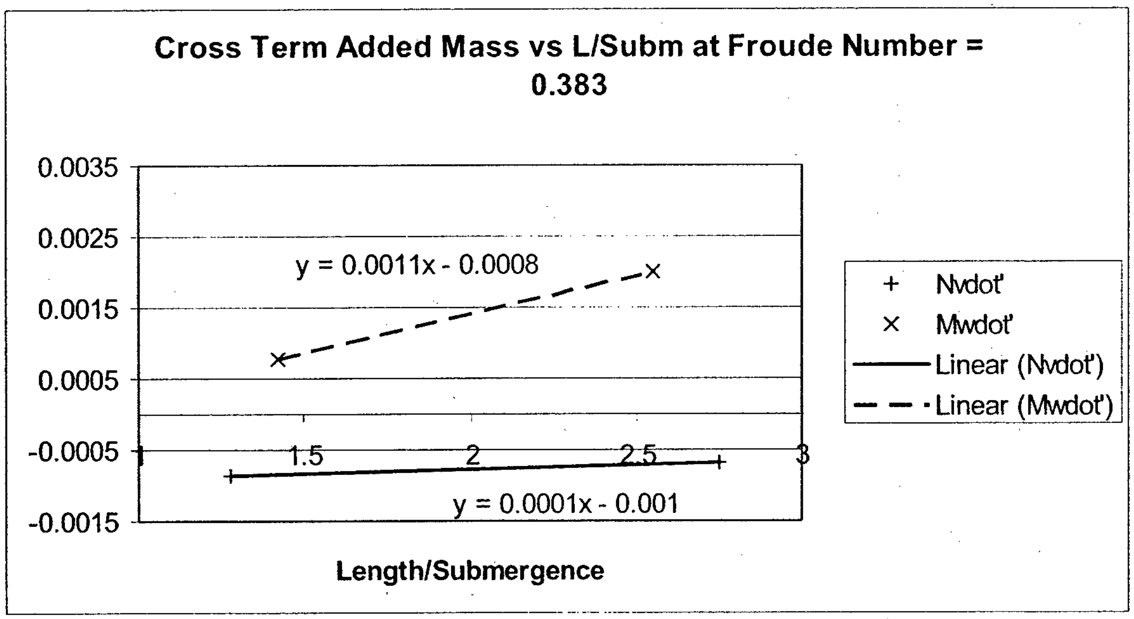

Figure 22. $N_{\dot{v}}{}'$ **and** $M_{\dot{w}}{}'$ **vs L/Subm for Froude Number = 0.383**

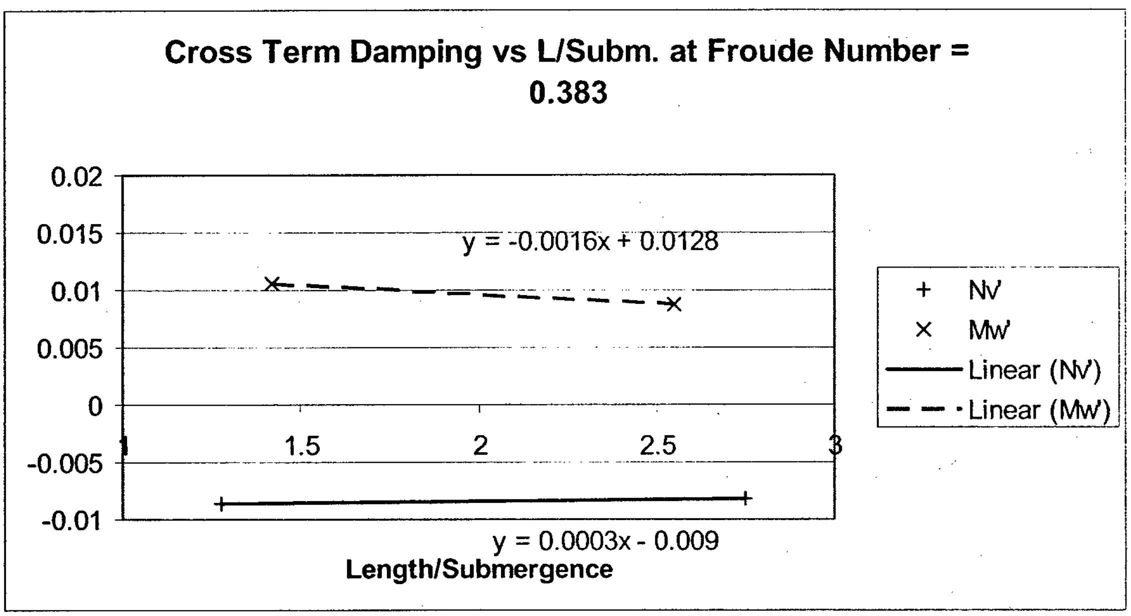

Figure 23. $N_{v}{}'$ **and** $M_{w}{}'$ **vs L/Subm for Froude Number = 0.383**

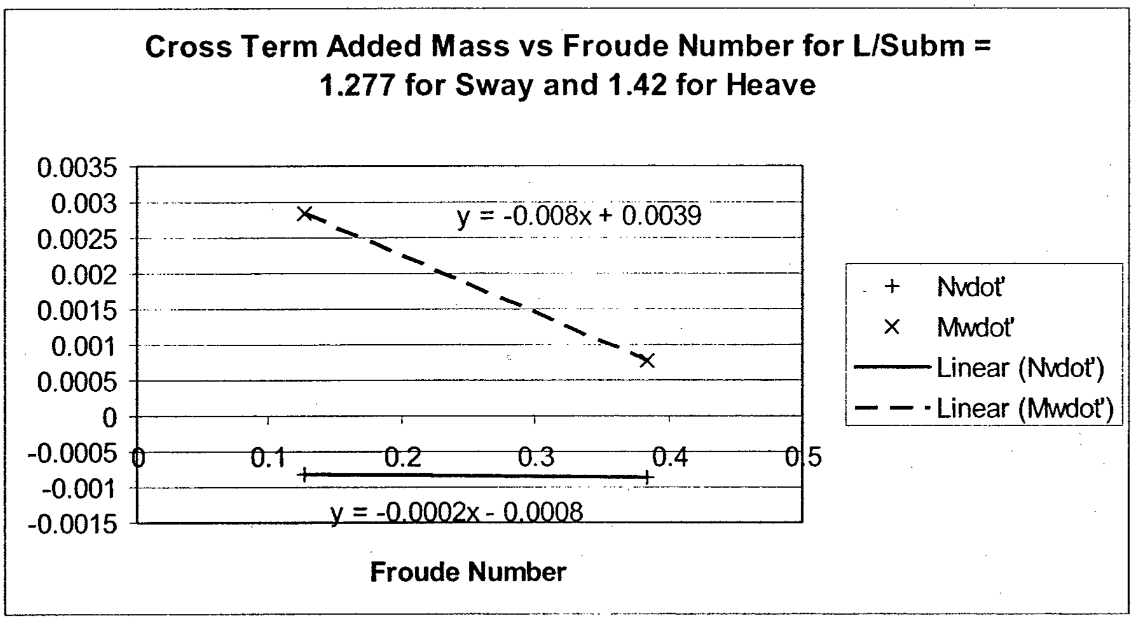

Figure 24. $N_{\dot{v}}'$ **and** $M_{\dot{w}}'$ **vs Froude Number for L/Subm = 1.277 for Sway and 1.42 for Heave**

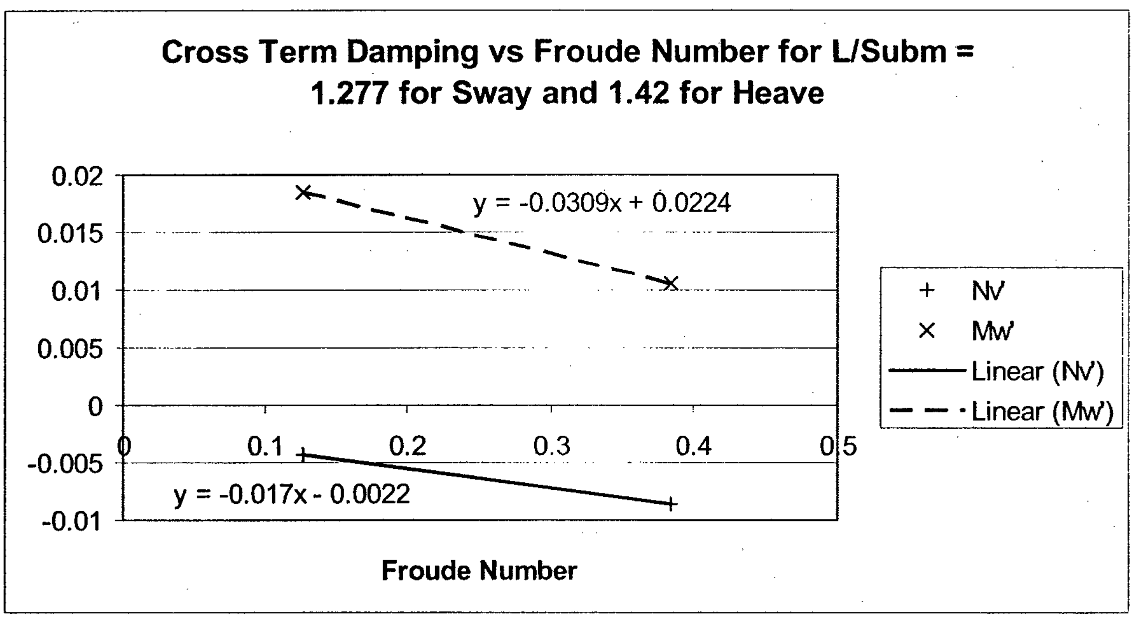

Figure 25. N_{v}' **and** M_{w}' **vs Froude Number for L/Subm = 1.277 for Sway and 1.42 for Heave**

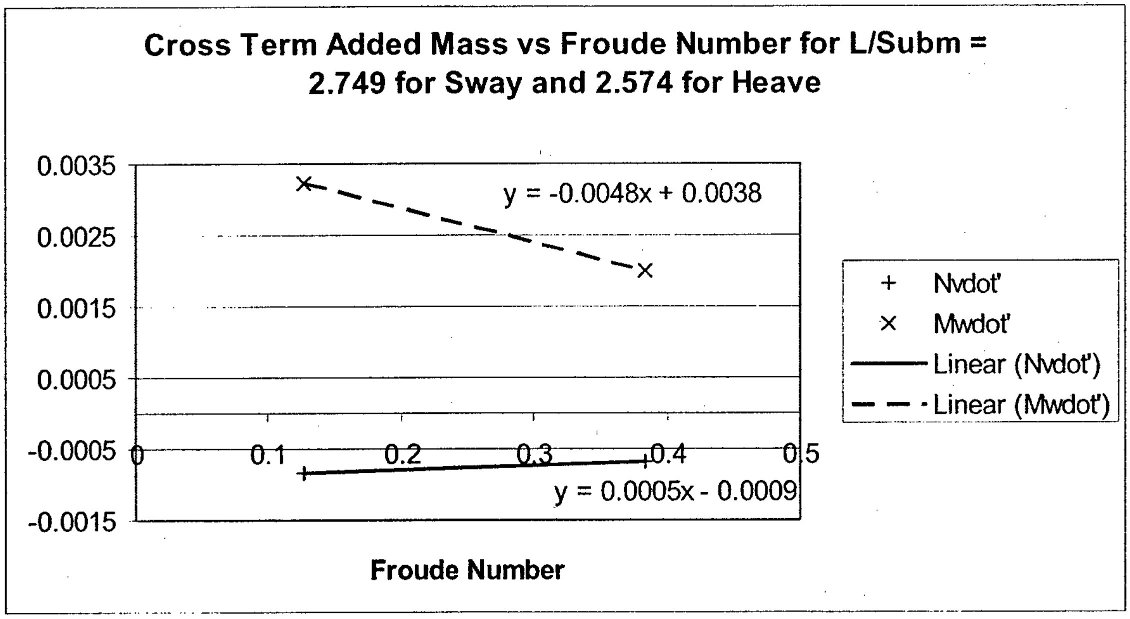

Figure 26. $N_{\dot{v}}{}'$ **and** $M_{\dot{w}}{}'$ **vs Froude Number for L/Subm = 2.749 for Sway and 2.574 for Heave**

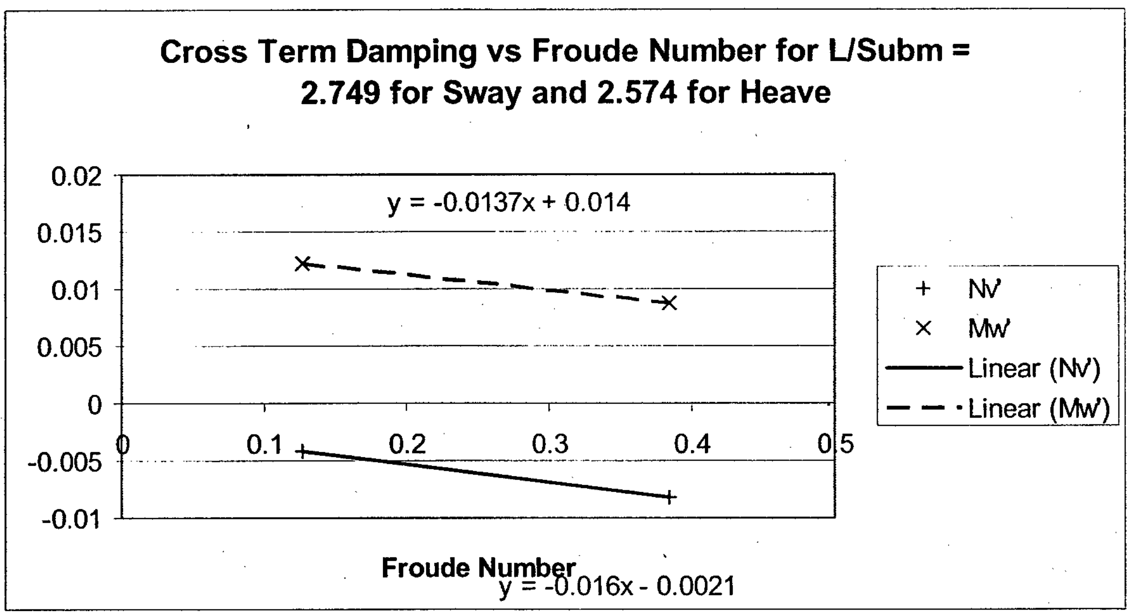

Figure 27. $N_{v}{}'$ **and** $M_{w}{}'$ **vs Froude Number for L/Subm = 2.749 for Sway and 2.574 for Heave**

The cross-terms due to sway and heave motion are expected to be very similar to each other and have similar dependencies, however Figure 20 through Figure 27 show that this may not be the case. Examination of the measured moments listed in Table 5 and Table 6 show that the measured yaw moments due to sway force range from 0.085 to 0.352 N-m and the measured pitch moments due to heave force range from 0.293 to 0.966 N-m. The transducer is rated for up to 28 N-m in yaw and pitch. This means that the maximum measured moments represent less than 4% of the range of the transducer. The minimum moment represents only 0.3% of the range of the transducer. The abnormal behavior of the moment coefficients may be due to insufficient moment being present to properly deflect the transducer.

The effects of submergence and speed were determined numerically by performing a least squares fit to the data using linear regression. This procedure is described in section 5.3.4. The resulting equations for the coefficients are:

$$Y_v' = -0.081458 + 0.001774\frac{Length}{Submergence} + 0.121754Fr$$

$$Y_{\dot{v}}' = -0.016345 + 0.000061\frac{Length}{Submergence} - 0.007224Fr$$

$$Z_w' = -0.086685 + 0.00091\frac{Length}{Submergence} + 0.140751Fr$$

$$Z_{\dot{w}}' = -0.013633 - 0.002679\frac{Length}{Submergence} - 0.005671Fr$$

$$M_w' = 0.023224 - 0.002943\frac{Length}{Submergence} - 0.022359Fr$$

$$M_{\dot{w}}' = 0.002825 + 0.000594\frac{Length}{Submergence} - 0.006407Fr$$

$$N_v' = -0.00207 + 0.000093\frac{Length}{Submergence} - 0.016489Fr$$

$$N_{\dot{v}}' = -0.00089 + 0.000037\frac{Length}{Submergence} + 0.000172Fr \tag{36}$$

The quality of the fit for these equations is expressed in terms of the root-mean-square value of the percent difference between the predicted and the empirical coefficients. Table 7 contains the values for the quality of fit.

Table 7. Quality of Fit for Heave and Sway Motion Coefficients

Coefficient	Fit
Y_v'	15%
$Y_{\dot{v}}'$	4%
Z_w'	17%
$Z_{\dot{w}}'$	9%
M_w'	19%
$M_{\dot{w}}'$	27%
N_v'	9%
$N_{\dot{v}}'$	9%

5.4.5. Forces and Moments Due to Body Angle

The forces and moments due to body angle were determined using data from experiments performed on the small scale model at MIT. Appendix L contains the list of experiments performed.

The equations for the coefficients due to body angle are[16]

$$
\begin{aligned}
Y_{uv} &= \frac{Y}{-\frac{1}{2}\rho U^2 D^2 \alpha} \\
Z_{uw} &= \frac{Z}{-\frac{1}{2}\rho U^2 D^2 \alpha} \\
M_{uw} &= \frac{M}{-\frac{1}{2}\rho U^2 D^2 L\alpha} \\
N_{uv} &= \frac{N}{-\frac{1}{2}\rho U^2 D^2 L\alpha}
\end{aligned}
\qquad (37)
$$

where

D is the diameter of the vehicle
L is the length of the vehicle
α is the angle of attack.

The results of these experiments are included as Appendix M. The effect of the angles is assumed to be linear for small angles, so the coefficients were calculated for each experiment and then averaged for each combination of speed and submergence. Pitch and Yaw angle coefficients were averaged separately. Any data point with a calculated coefficient more than 1.15 standard deviations from the mean was removed from consideration as unreliable data. 1.15 standard deviations was chosen as the discrimination point in order to remove the clearly bad data while retaining as much of the possibly good data as possible. The mean coefficients for each combination of submergence and velocity are shown in Table 8. Figure 28 through Figure 31 illustrate the dependency of the restoring forces on submergence and speed.

It is very important to note at this point that the magnitudes of the yaw and pitch moments measured during these tests are very small, on the order of less than 1% of the capacity of the load cell. The data is analyzed and presented here, but further work is required to determine the accuracy of the equations with a more appropriate transducer.

Table 8. Mean Coefficients for Force and Moment Due to Body Angle at Various Submergences and Velocities

Yuv		
Submergence	Velocity	Yuv
m	m/s	
0.252	0.75	-5.36887
0.398	0.75	-4.85345
0.543	0.75	-2.32929
0.252	1	-3.51475
0.398	1	-2.8487
0.543	1	-1.61492

Nuv		
Submergence	Velocity	Nuv
m	m/s	
0.252	0.75	-1.02061
0.398	0.75	-0.99671
0.543	0.75	-0.68264
0.252	1	-0.80028
0.398	1	-0.73773
0.543	1	-0.58644

Zuw		
Submergence	Velocity	Zuw
m	m/s	
0.252	0.75	1.023955
0.398	0.75	2.269136
0.543	0.75	1.141877
0.252	1	1.220249
0.398	1	2.612357
0.543	1	1.801033

Muw		
Submergence	Velocity	Muw
m	m/s	
0.252	0.75	-0.626
0.398	0.75	-0.59447
0.543	0.75	-0.71149
0.252	1	-0.4012
0.398	1	-0.61916
0.543	1	-0.59916

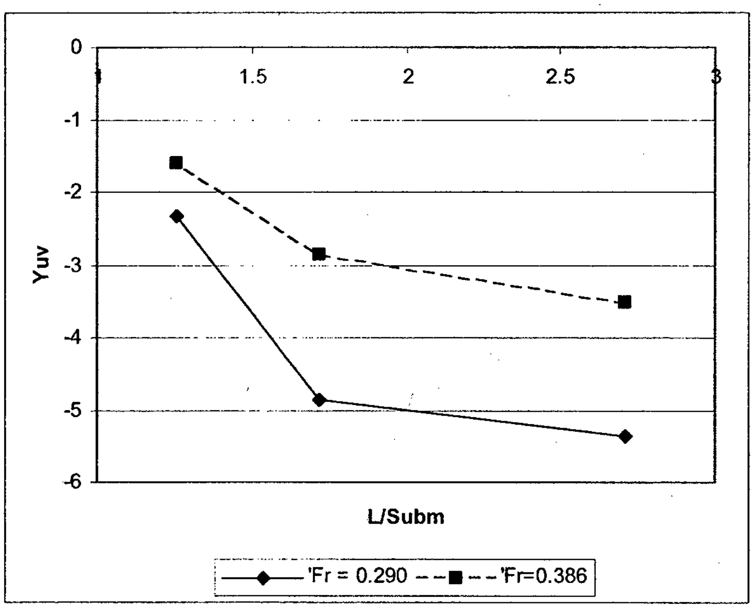

Figure 28. Yuv as a Function of Submergence for Two Speeds

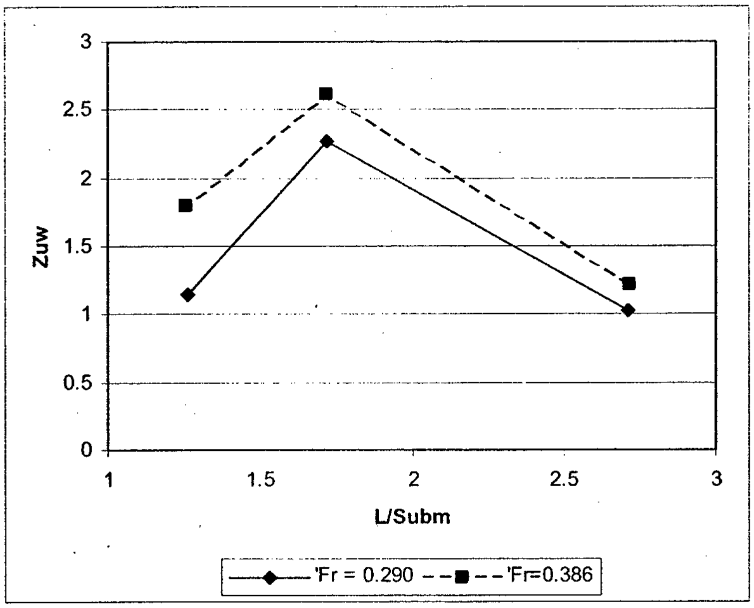

Figure 29. Zuw as a Function of Submergence for Two Speeds

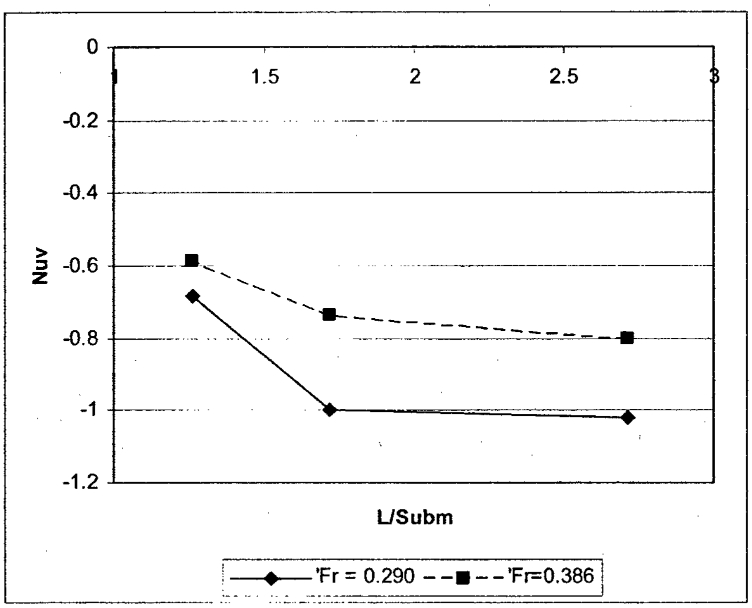

Figure 30. Nuv as a Function of Submergence for Two Speeds

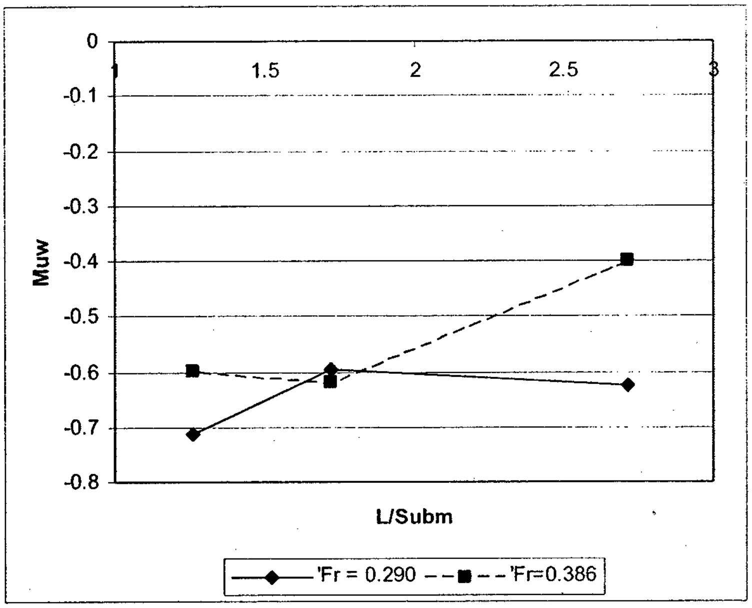

Figure 31. Muw as a Function of Submergence for Two Speeds

The data from the restoring force and moment experiments was put into a linear regression model to determine the dependency of those forces and moments on submergence and speed. The resulting equations are

$$
\begin{aligned}
Y_{uv} &= 3.035943 - 11.232476\frac{Length}{Submergence} + 2.399695\left(\frac{Length}{Submergence}\right)^2 + 15.795188Fr \\
Z_{uw} &= -7.723764 + 9.156976\frac{Length}{Submergence} - 2.365368\left(\frac{Length}{Submergence}\right)^2 + 4.182974Fr \\
N_{uv} &= 0.020017 - 1.452965\frac{Length}{Submergence} + 0.317964\left(\frac{Length}{Submergence}\right)^2 + 1.987686Fr \\
M_{uw} &= -1.168727 + 0.127681\frac{Length}{Submergence} - 0.007568\left(\frac{Length}{Submergence}\right)^2 + 1.079278Fr
\end{aligned}
\tag{38}
$$

Table 9. Quality of Fit for Restoring Force Coefficients

Coefficient	Quality of Fit
Y_{uv}	13%
Z_{uw}	8%
M_{uw}	5%
N_{uv}	10%

5.4.6. Sway Force and Yaw Moment Due to Yaw Motion

Experiments were performed with several combinations of submergence, velocity, and frequency of oscillation to determine the effect of yaw motion on sway force and yaw moment. The experiments performed are listed in Appendix B. The method of analysis was similar to that used for the sway and heave motion tests, but in this case there is another term due to the angle of the body as it moves forward and oscillates in yaw.

The equations are

$$\begin{aligned} Y &= Y_r r + Y_{\dot{r}} \dot{r} + Y_{uv}(-\tfrac{1}{2}\rho U^2 D^2 \alpha) \\ N &= N_r r + N_{\dot{r}} \dot{r} + N_{uv}(-\tfrac{1}{2}\rho U^2 D^2 L\alpha) \end{aligned} \tag{39}$$

and

$$\begin{aligned} Y_r &= \frac{\mathrm{Im}(Y)}{\omega A} \\ Y_{\dot{r}} &= \frac{\mathrm{Re}(Y) - Y_{uv}(-\frac{1}{2}\rho U^2 D^2 \alpha)}{-\omega^2 A} \\ N_r &= \frac{\mathrm{Im}(N)}{\omega A} \\ N_{\dot{r}} &= \frac{\mathrm{Re}(N) - Y_{uv}(-\frac{1}{2}\rho U^2 D^2 L\alpha)}{-\omega^2 A} \end{aligned} \tag{40}$$

The restoring force is a real force and must be subtracted from the measured real hydrodynamic force to calculate the force due to the yaw motion.

The test conditions and results are presented in Table 10.

Table 10 Test Conditions and Results for Sway Force and Yaw Moment Due to Yaw Motion

Yaw Motion				Hydrodynamic Sway Force, Y								
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(Y)	Re(Y)-Yuv	Yrdot	Yrdot'	Im(Y)	Yr	Yr'
m	m/s	Hz	Degrees	N	Degrees	N	N	kg		N	kg/s	
0.252	0.333	1.19629	10	1.841	124.400	-1.0402	-1.5127	0.1543	0.0013	1.5192	1.1650	0.0210
0.252	1.000	1.19629	10	3.134	97.200	-0.3928	-2.4287	0.2462	0.0021	3.1095	2.3690	0.0142
0.543	0.333	0.40283	10	0.420	101.700	-0.0851	-0.4163	0.3739	0.0032	0.4111	0.9343	0.0169
0.543	0.333	1.19629	10	1.760	124.600	-0.9994	-1.3294	0.1359	0.0012	1.4487	1.1130	0.0201
0.398	0.667	0.40283	10	1.761	57.100	0.9565	-0.3364	0.3010	0.0026	1.4785	3.3486	0.0302
0.543	1.000	0.79346	10	1.535	101.400	-0.3034	-1.0481	0.2425	0.0021	1.5049	1.7359	0.0104

Yaw Motion				Hydrodynamic Yaw Moment, N								
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(N)	Re(N)-Nuv	Nrdot	Nrdot'	Im(N)	Nr	Nr'
m	m/s	Hz	Degrees	N-m	Degrees	N-m	N	kg		N-m	kg/s	
0.252	0.333	1.196	10	0.814	-31.6	0.6932	0.6371	-0.0650	-0.0008	-0.42647	-0.32704	-0.00851
0.252	1.000	1.196	10	1.114	-34.2	0.9215	0.6086	-0.0617	-0.0008	-0.62622	-0.47710	-0.00414
0.543	0.333	0.403	10	0.130	-21.2	0.1212	0.0758	-0.0681	-0.0009	-0.04701	-0.10685	-0.00278
0.543	0.333	1.196	10	0.828	-32.2	0.7008	0.6556	-0.0670	-0.0008	-0.44133	-0.33905	-0.00883
0.398	0.667	0.403	10	0.350	-22.7	0.3229	0.1444	-0.1292	-0.0016	-0.13507	-0.30591	-0.00398
0.543	1.000	0.793	10	0.497	-27.8	0.4397	0.2260	-0.0523	-0.0007	-0.23184	-0.26742	-0.00232

5.4.7. Heave Force and Pitch Moment Due to Pitch Motion

The analysis for heave force and pitch moment due to pitch motion closely mirrors the analysis for sway force and yaw moment due to yaw motion. For pitch motion, the equations are

$$\begin{aligned} Z &= Z_q q + Z_{\dot{q}} \dot{q} + Z_{uw}(-\tfrac{1}{2}\rho U^2 D^2 \alpha) \\ M &= M_q q + M_{\dot{q}} \dot{q} + M_{uw}(-\tfrac{1}{2}\rho U^2 D^2 L\alpha) \end{aligned} \tag{41}$$

and

$$
\begin{aligned}
Z_q &= \frac{\mathrm{Im}(Z)}{\omega A} \\
Z_{\dot{q}} &= \frac{\mathrm{Re}(Z) - Z_{uw}(-\frac{1}{2}\rho U^2 D^2 \alpha)}{-\omega^2 A} \\
M_q &= \frac{\mathrm{Im}(M)}{\omega A} \\
M_{\dot{q}} &= \frac{\mathrm{Re}(M) - M_{uw}(-\frac{1}{2}\rho U^2 D^2 L \alpha)}{-\omega^2 A}
\end{aligned}
\tag{42}
$$

The test conditions and results are presented in Table 11. During these tests special consideration was given to removing the effects of surge motion caused by the long pitch arm of the test apparatus. In order to remove these effects, the test apparatus was oscillated in surge at the same time pitch oscillations occurred. The surge oscillations were 180° out of phase with the pitch oscillations and of such a magnitude as to cancel the surge due to pitch.

Table 11. Test Conditions and Results for Heave Force and Pitch Moment due to Pitch Motion

Pitch Motion				Hydrodynamic Heave Force, Z								
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(Z)	Re(Z)-Zuw	Zqdot	Zqdot'	Im(Z)	Zq	Zq'
m	m/s	Hz	Degrees	N	Degrees	N	N	kg		N	kg/s	
0.252	0.333	1.19629	9	3.173	-61.600	1.5092	1.5034	-0.1727	-0.0015	-2.7912	-2.4102	-0.0435
0.252	1.000	0.40283	9	0.677	-74.300	0.1833	-0.4326	0.4118	0.0036	-0.6520	-1.5709	-0.0094
0.543	0.333	1.19629	9	3.146	-62.300	1.4622	1.4267	-0.1634	-0.0014	-2.7850	-2.3970	-0.0433
0.543	0.333	0.79346	9	2.415	-88.400	0.0674	0.0303	-0.0075	-0.0001	-2.4143	-2.9939	-0.0540
0.252	1.000	0.40283	9	0.653	-76.000	0.1579	-0.4579	0.4360	0.0038	-0.6331	-1.5255	-0.0092

Pitch Motion				Hydrodynamic Pitch Moment, M								
Submergence	Velocity	Frequency	Amplitude	Amplitude	Phase	Re(M)	Re(M)-Muw	Mqdot	Mqdot'	Im(M)	Mq	Mq'
m	m/s	Hz	Degrees	N-m	Degrees	N-m	N	kg		N-m	kg/s	
0.252	0.333	1.196	9	0.665	-34.700	0.5470	0.5749	-0.0660	-0.0008	-0.37874	-0.32704	-0.00851
0.252	1.000	0.403	9	0.080	-21.300	0.0745	0.2420	-0.2303	-0.0029	-0.02906	-0.07001	-0.00061
0.543	0.333	1.196	9	0.718	-35.600	0.5841	0.6175	-0.0707	-0.0009	-0.41820	-0.35994	-0.00937
0.543	0.333	0.793	9	0.357	-29.000	0.3125	0.3475	-0.0864	-0.0011	-0.17322	-0.21481	-0.00559
0.252	1.000	0.403	9	0.111	-17.000	0.1062	0.3203	-0.3049	-0.0038	-0.03248	-0.07827	-0.00068

5.4.8. Discussion of Yaw and Pitch Motion Results

The results of yaw and pitch motion testing are graphically illustrated in Figure 32 through Figure 47. Both submergence and speed were found to be significant factors.

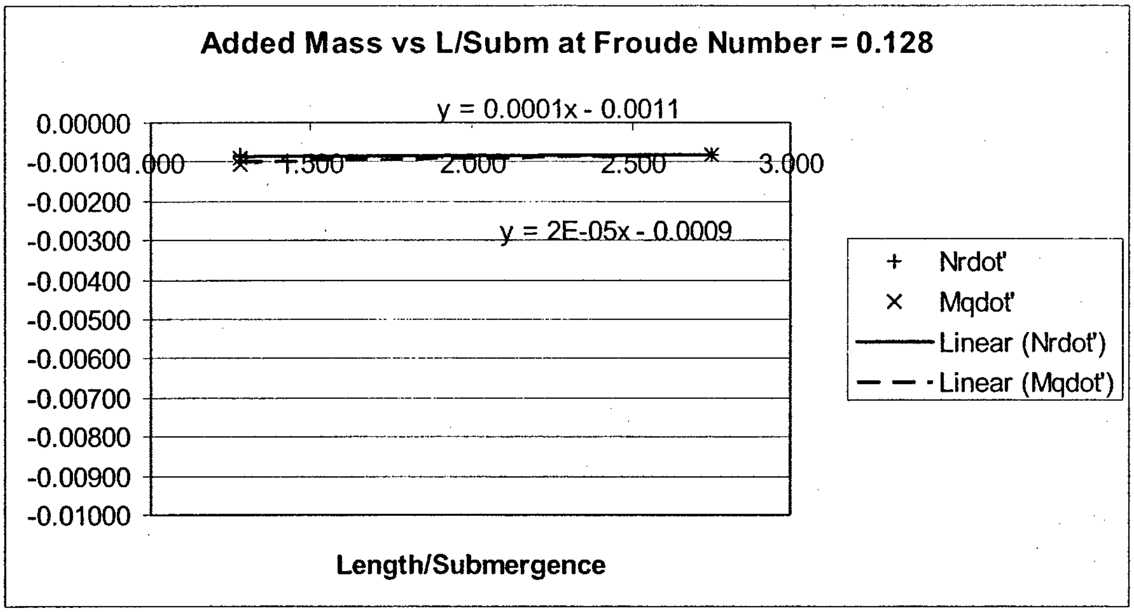

Figure 32. $N_{\dot{r}}'$ **and** $M_{\dot{q}}'$ **vs L/Subm at Froude Number = 0.128**

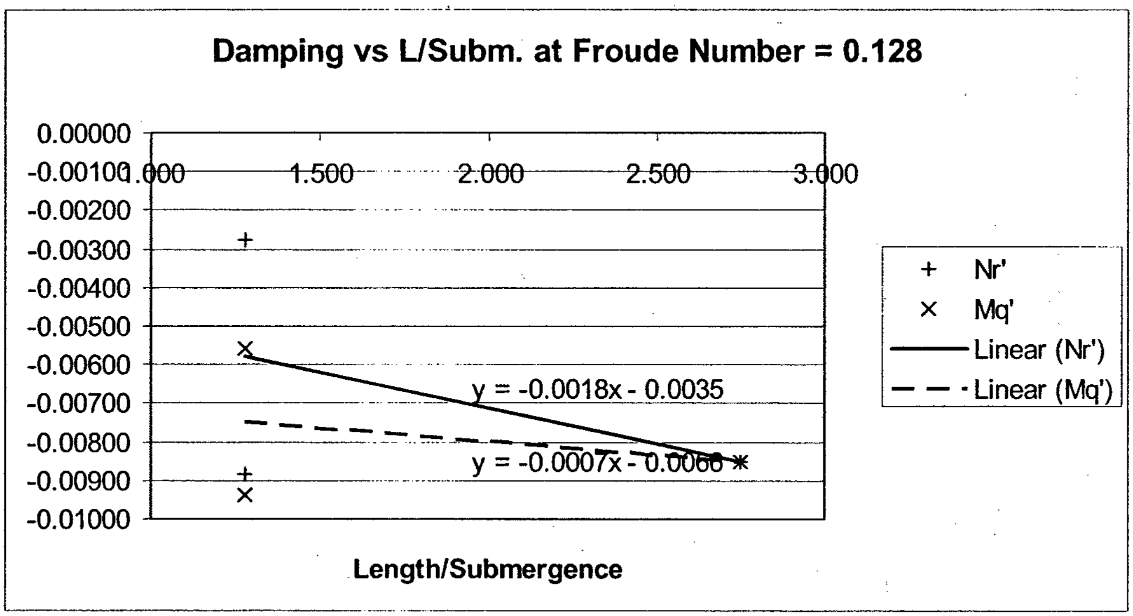

Figure 33. N_r' **and** M_q' **vs L/Subm at Froude Number = 0.128**

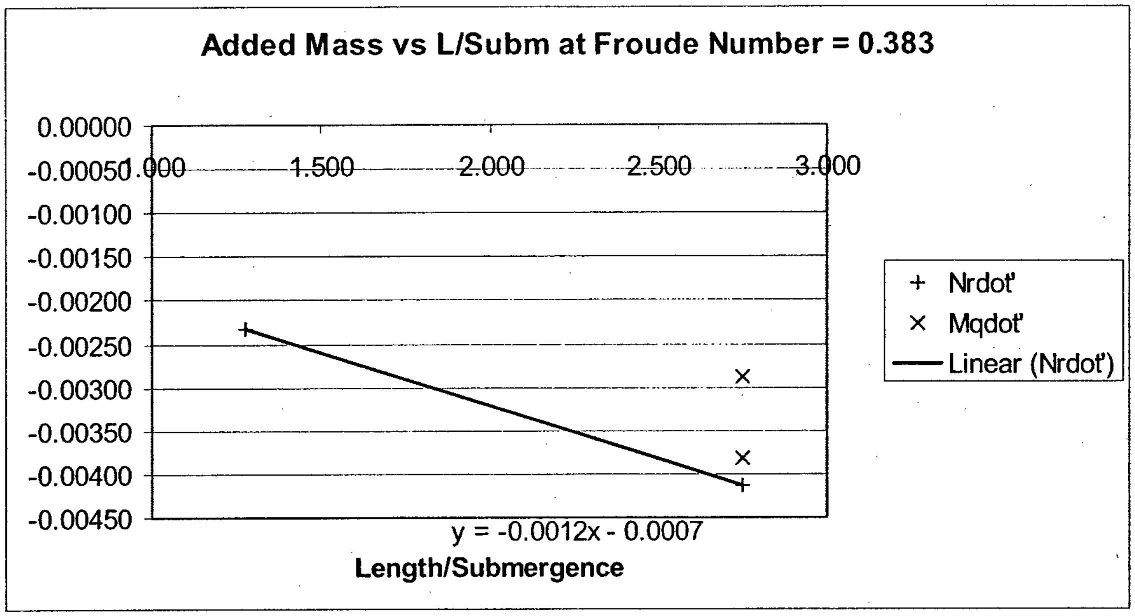

Figure 34. $N_{\dot{r}}'$ **and** $M_{\dot{q}}'$ **vs L/Subm at Froude Number = 0.383**

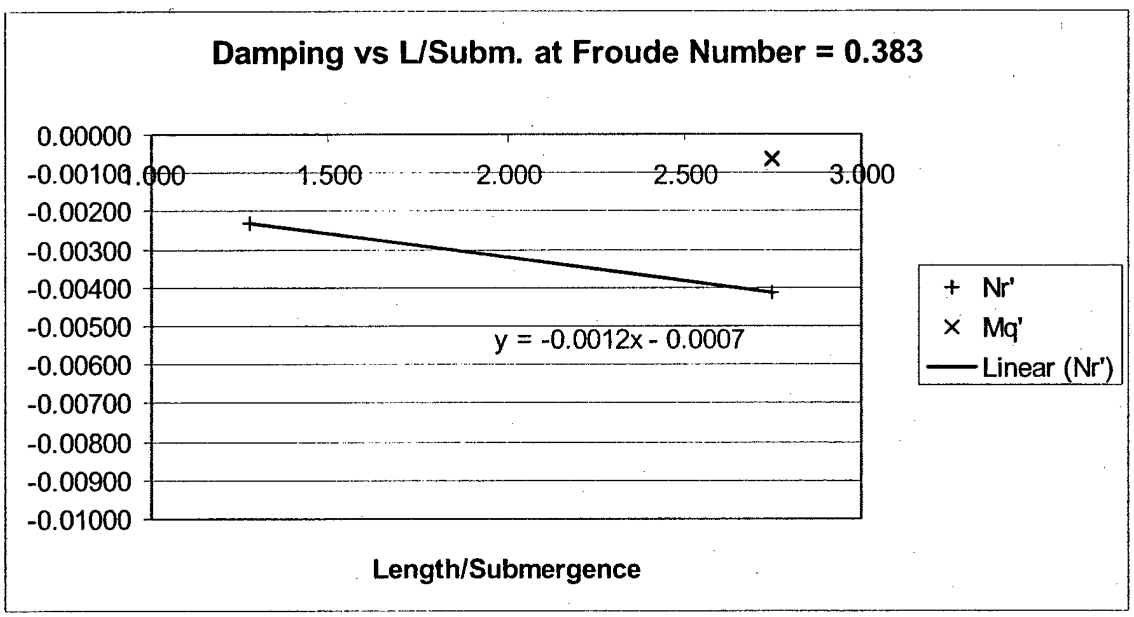

Figure 35. N_{r}' **and** M_{q}' **vs L/Subm at Froude Number = 0.383**

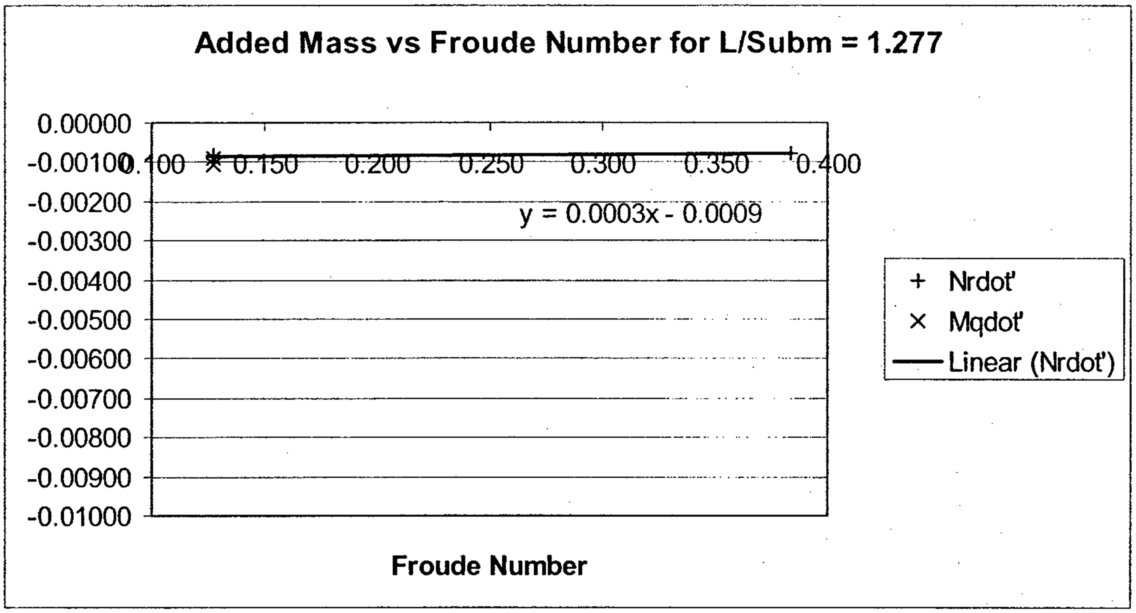

Figure 36. $N_{\dot{r}}'$ and $M_{\dot{q}}'$ vs Froude Number at L/Subm= 1.277

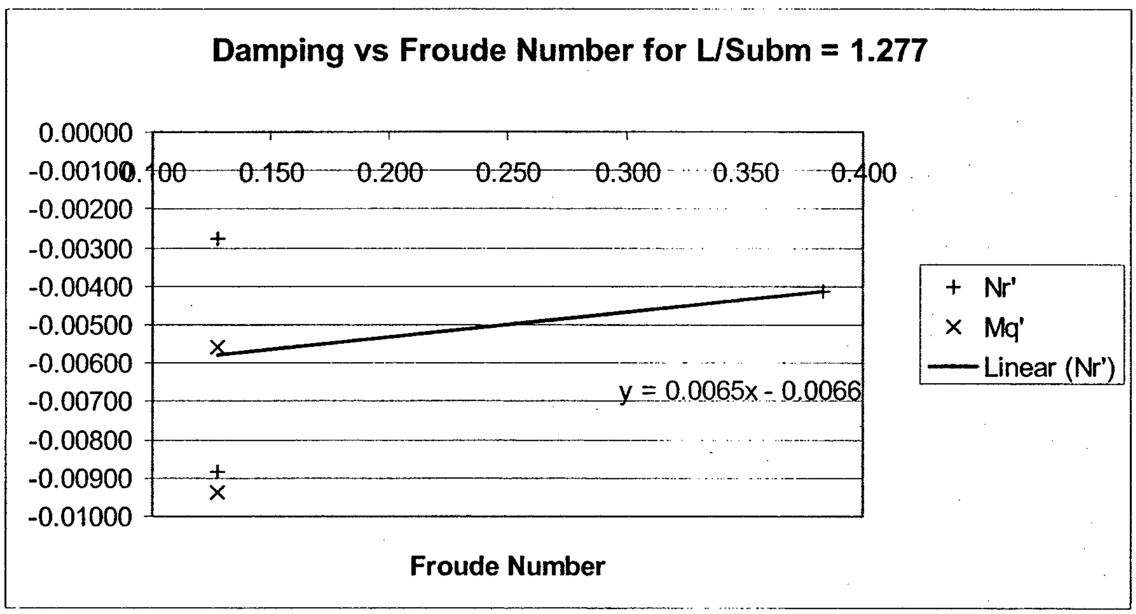

Figure 37. N_r' and M_q' vs Froude Number at L/Subm= 1.277

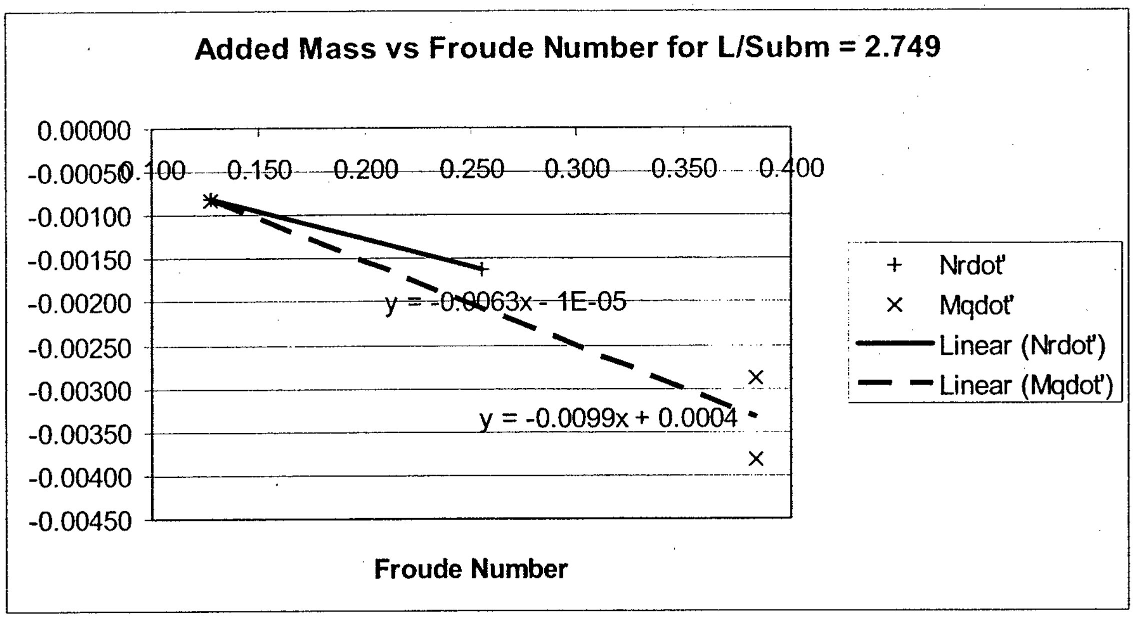

Figure 38. $N_{\dot{r}}'$ and $M_{\dot{q}}'$ vs Froude Number at L/Subm= 2.749

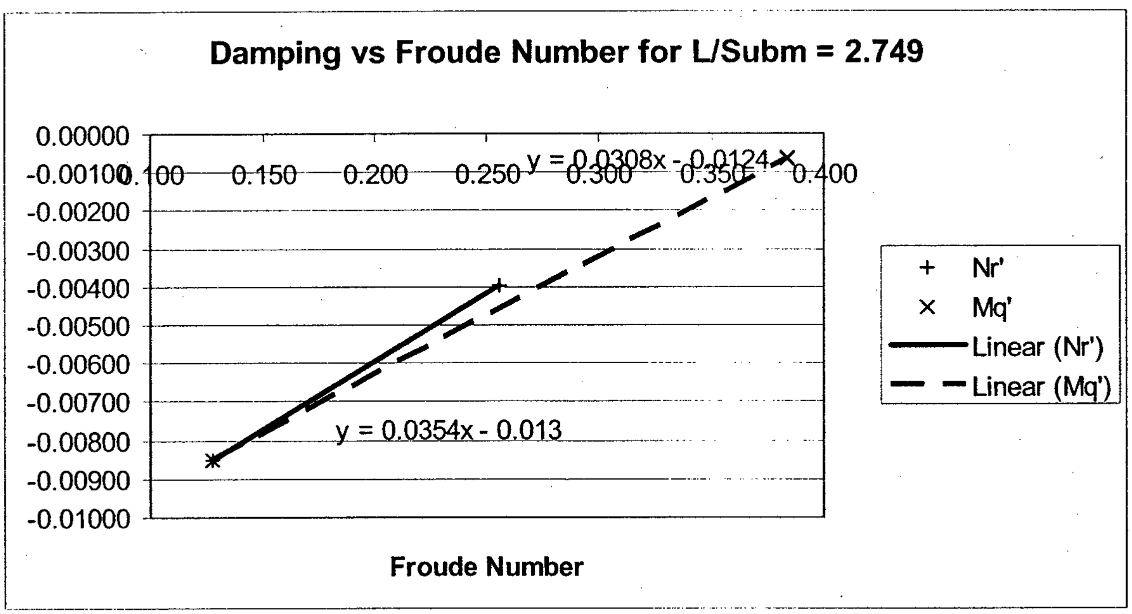

Figure 39. N_r' and M_q' vs Froude Number at L/Subm= 2.749

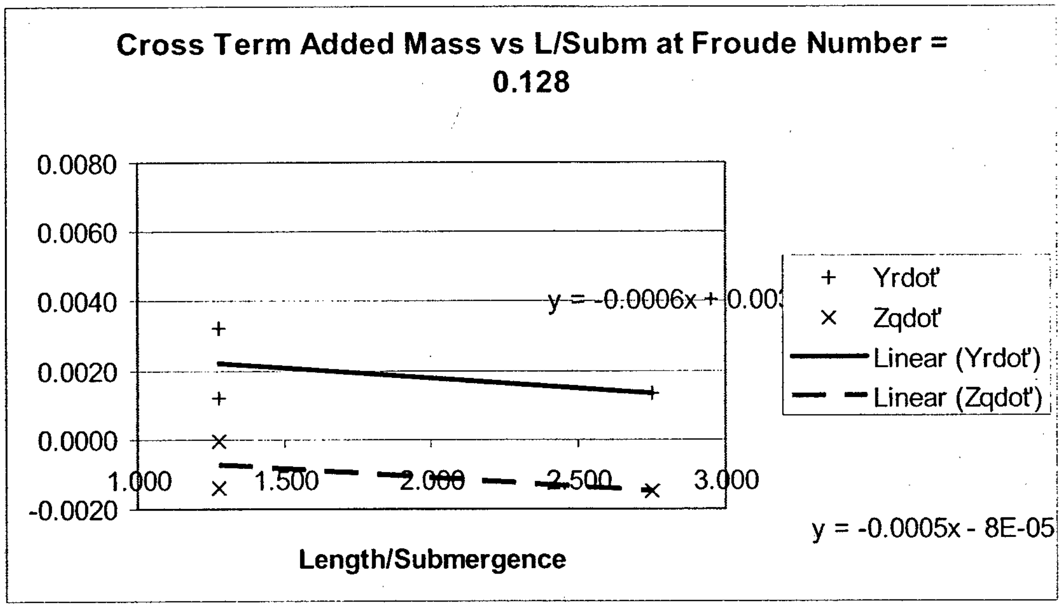

Figure 40. $Y_{\dot{r}}'$ and $Z_{\dot{q}}'$ vs L/Subm at Froude Number = 0.128

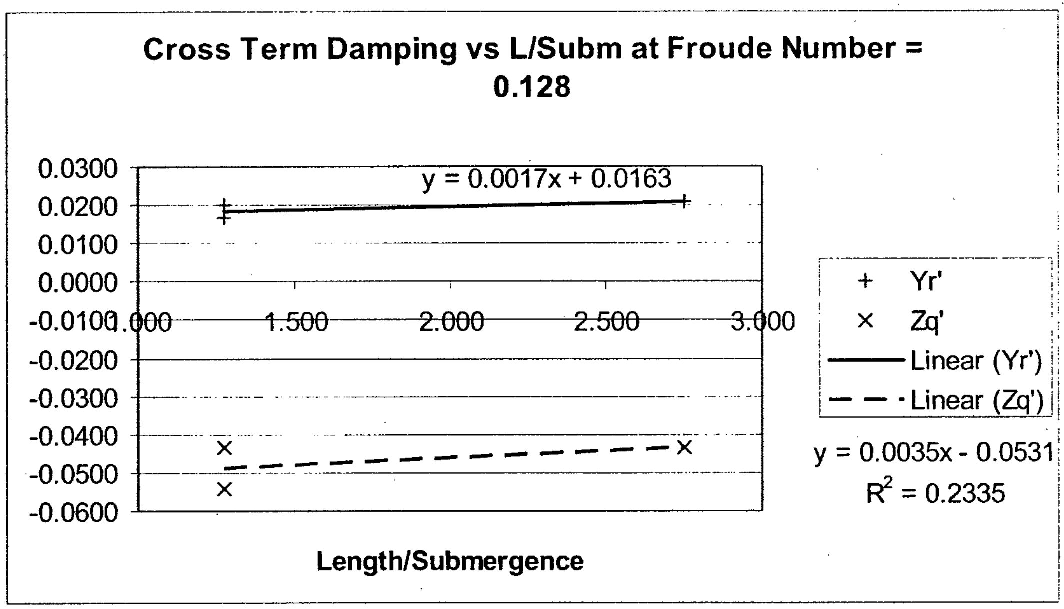

Figure 41. Y_r' and Z_q' vs L/Subm at Froude Number = 0.128

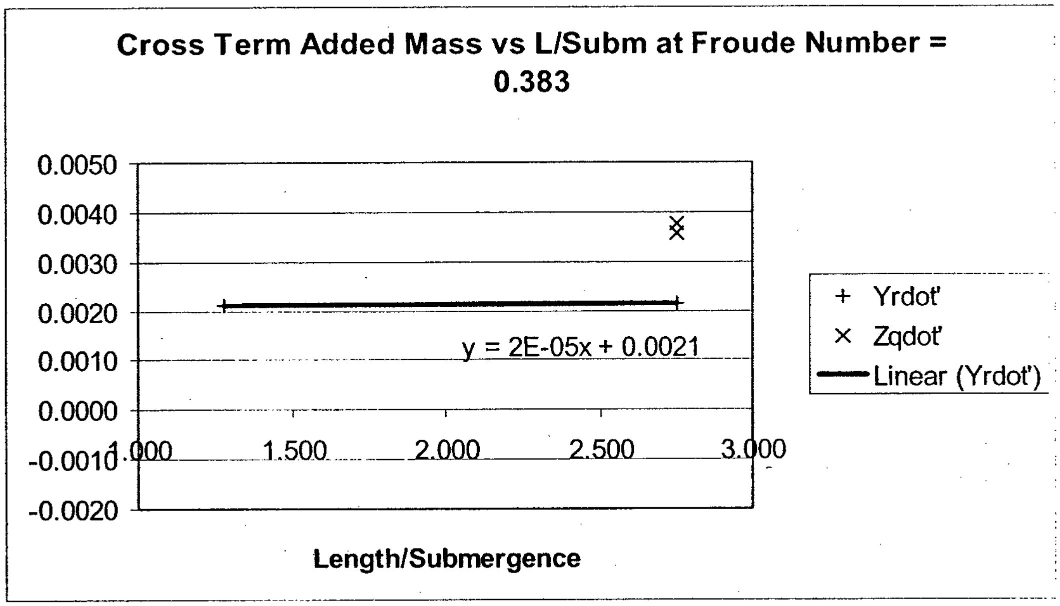

Figure 42. $Y_{\dot{r}}'$ **and** $Z_{\dot{q}}'$ **vs L/Subm at Froude Number = 0.383**

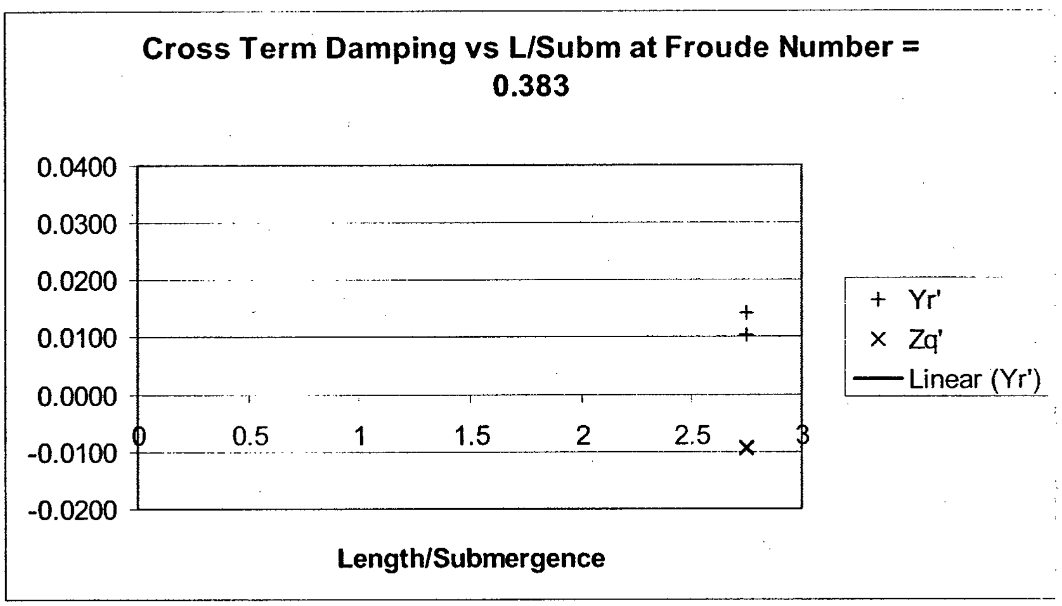

Figure 43. Y_{r}' **and** Z_{q}' **vs L/Subm at Froude Number = 0.383**

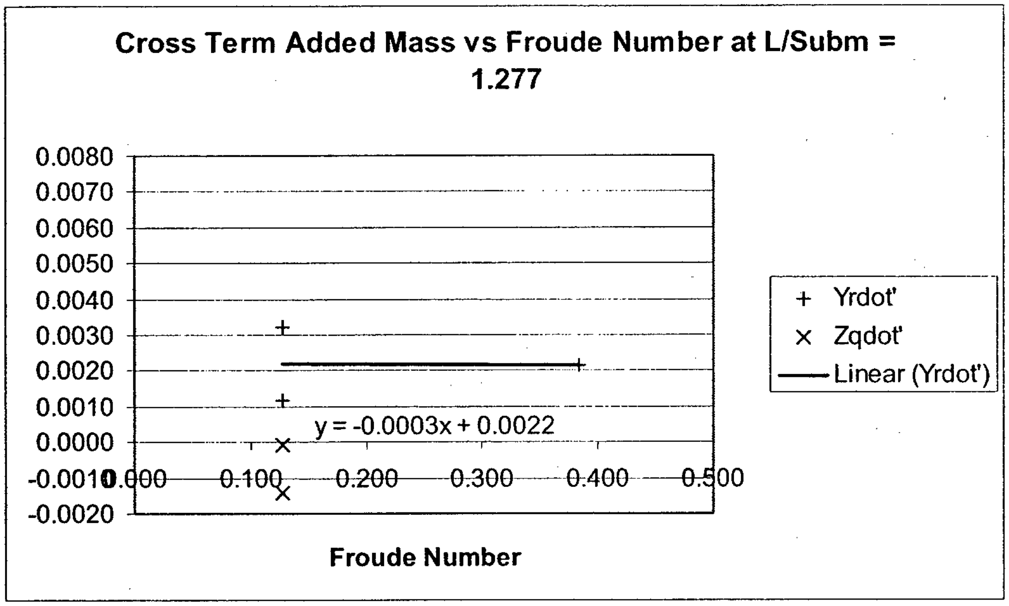

Figure 44. $Y_{\dot{r}}'$ and $Z_{\dot{q}}'$ vs Froude Number at L/Subm= 1.277

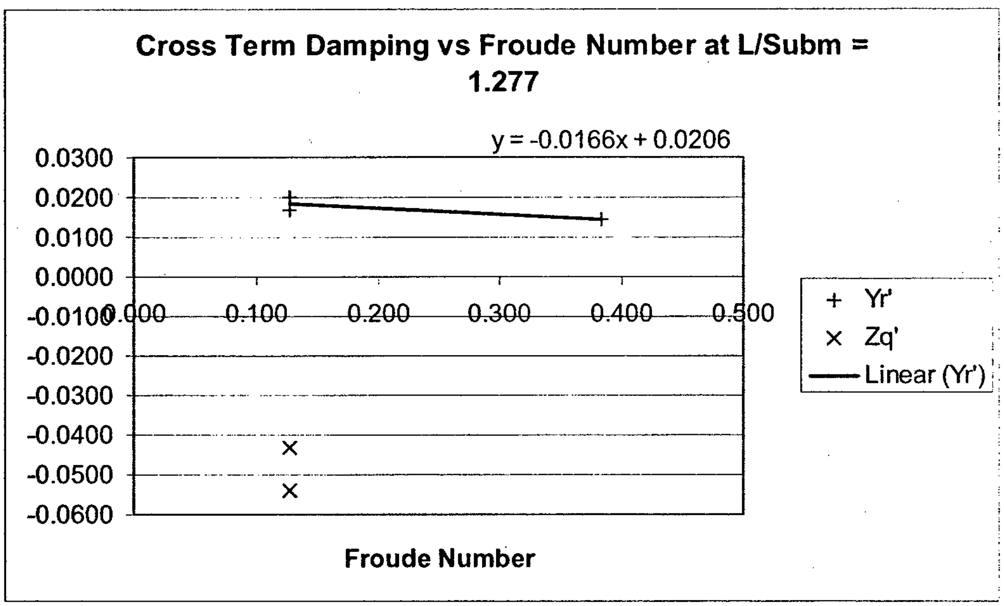

Figure 45. Y_r' and Z_q' vs Froude Number at L/Subm= 1.277

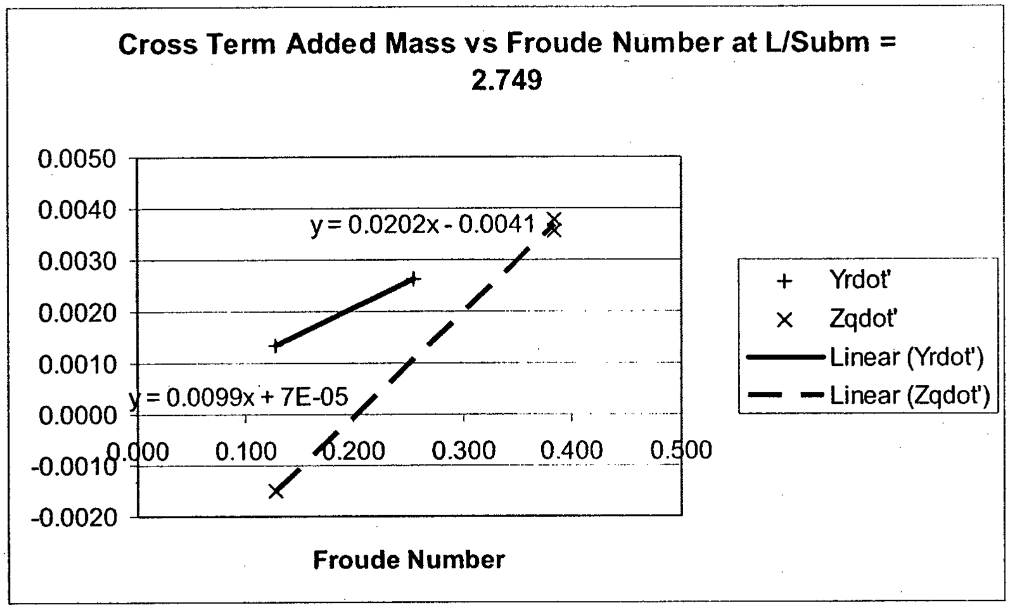

Figure 46. $Y_{\dot{r}}'$ **and** $Z_{\dot{q}}'$ **vs Froude Number at L/Subm= 2.749**

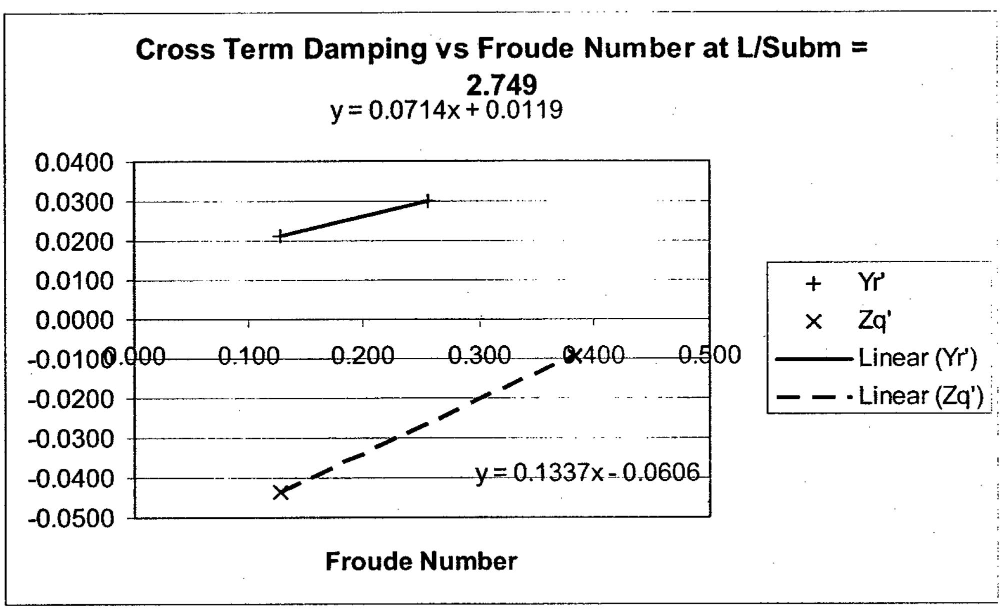

Figure 47. Y_r' **and** Z_q' **vs Froude Number at L/Subm= 2.749**

The data from pitch and yaw motion was also analyzed by linear regression to yield equations for the coefficients. Those equations are

$$Y_r' = 0.020906 + 0.000403\frac{Length}{Submergence} - 0.017979Fr$$

$$Y'_{\dot{r}} = 0.002324 - 0.000087\frac{Length}{Submergence} - 0.000632Fr$$

$$Z_q' = -0.070199 + 0.003498\frac{Length}{Submergence} + 0.133729Fr$$

$$Z_{\dot{q}}' = -0.002666 - 0.000514\frac{Length}{Submergence} + 0.020222Fr$$

$$M_q' = -0.010515 - 0.000701\frac{Length}{Submergence} + 0.030777Fr$$

$$M_{\dot{q}}' = 0.00014 + 0.000106\frac{Length}{Submergence} - 0.009854Fr$$

$$N_r' = -0.006013 - 0.001173\frac{Length}{Submergence} + 0.012297Fr$$

$$N_{\dot{r}}' = -0.000786 - 0.000019\frac{Length}{Submergence} - 0.000582Fr \qquad (43)$$

Table 12 contains the values for the quality of fit.

Table 12. Quality of Fit for Pitch and Yaw Motion Coefficients

Coefficient	Quality of Fit
Y_r'	7%
$Y_{\dot{r}}'$	46%
Z_q'	0%
$Z_{\dot{q}}'$	0%
M_q'	0%
$M_{\dot{q}}'$	0%
N_r'	23%
$N_{\dot{r}}'$	8%

6. Computational Analysis

Numerical method results to accompany experimental results are very important. To that end the validated free surface linear code for surface ships, SWAN, was revised by one of its developers to include submerged objects. Unfortunately, the results from the submerged vehicle version of SWAN were too unreasonable to be either valid or included here. For example, in some conditions, the added mass was predicted to be 16 times the actual displaced mass. That cannot be correct. There was not enough time to properly revise, test, and validate the new version of this numerical method. Zero forward speed evaluations of added mass and damping were performed with the validated code WAMIT and those results were reasonable. However, the interest for this research is for an underwater vehicle with forward speed.

7. Conclusion

The analyses performed for this research have drawn upon a limited number of experiments with a great deal of uncertainty in the quality of data. The limited number of experiments is a result of very significant delays experienced in designing and testing the test apparatus. Although limited data exists that is known to be valid, a very large amount of data exists from earlier experiments. These earlier experiments may have had interaction between the transducer and the vehicle shell. Although the geometry of the model indicates no interference, there may have been interference due to shell distortion and corrosion products. Also, some of the uncertainty in the quality of early data results from the measurement of very small forces and moments that represent a very small fraction of the range of the transducer. The method of research and analysis is sound and, at the very least, provides a roadmap for conducting the research in the future.

The results associated with the sway force due sway motion and heave force due to heave motion are very similar and indicate that the quality of that data is very good. The effects of variation in submergence and speed on those forces can also be expected to be good results.

The proximity to the free surface has relatively little effect on forces due to motions or control surface deflections. This indicates that with minor control system alterations, operation in shallow water is probably feasible. Of course further work on proximity to the bottom is necessary.

The causes for the variation in the results with Froude number and submergence are not well understood. There are various theories that discuss the interaction of the flow stream with the free surface and the bottom that may be useful, but true understanding of the results of these analyses must be saved for future work.

8. Future Work

In this thesis we have found the effects of shallow water and submergence on many of the dominant terms in the maneuvering equations. The reason for these variations is also yet to be determined. In the very near future the data from the experiments conducted early in this research and not known to be reliable will be analyzed. If that data is found to be reliable, the results reported here will be expanded and modified to include that data and provide a better description of the effects of submergence and speed.

This work indicates that the proximity to the free surface has relatively little effect on forces due to motions or control surface deflections. This indicates that with minor control system alterations, operation in shallow water is probably feasible. Of course further work on proximity to the bottom is necessary. Much of this is available from the work of William Ramsey.[17]

Additional future work will involve finding a numerical method that can accurately predict the phenomena studied here. The results of this and similar research will be used to validate that method for widespread use in predicting this kind of behavior.

9. Acknowledgements

The author wishes to express his gratitude to those who assisted and supported this research. Professor Jerome Milgram provided tireless assistance and teaching. LTJG Greg Sabra, USCG assisted in the performance of the research as part of his related research for a thesis entitled "Wave Effects on Underwater Vehicles in Shallow Water." John Hill and John Zseleczky of the United States Naval Academy Marine Hydromechanics Laboratory provided much appreciated support and education during the research conducted there.

Most importantly, the author wishes to thank Grace, Dorothea, and Mark Oller for their patience and unending devotion during the conduct of this research.

10. Endnotes

[1] Prestero, Timothy, "Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle," Master of Science Dissertation, Massachusetts Institute of Technology, 2001.

[2] www.whoi.edu/home/marine/remus_main.html

[3] http://www.nwdc.navy.mil/Conference/FBEJ/7-28-release.asp

[4] Coleman, Jack, "Undersea Drones Pull Duty in Iraq Hunting Mines", Cape Cod Times Archives, April 2, 2003.

[5] "Nomenclature for Treating the Motion of a Submerged Body through a Fluid," SNAME Technical and Research Bulletin 1-5, 1952.

[6] Lewis, Edward V., ed., Principles of Naval Architecture, 3rd ed., SNAME, 1989.

[7] Triantafyllou, Michael S., and Franz S. Hover, Maneuvering and Control of Marine Vehicles, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[8] www.amtiweb.com

[9] "Dynamometer Instructions," Advanced Mechanical Technology, Inc., Watertown, Massachusetts, 2000.

[10] www.amtiweb.com

[11] http://www.usna.edu/AcResearch/HydromechanicsLaboratory.pdf

[12] Schmidt, Stephen R. and Robert G. Launsby, Understanding Industrial Experiments, Air Academy Press and Associates, 4th edition, 1988.

[13] Hoerner, Sigard F., Henry V. Borst, Fluid-Dynamic Lift, Second Edition, Mrs. Lisolette A. Hoerner, 1985.

[14] Hoerner

[15] Lewis

[16] Hoerner

[17] Ramsey, William Durand, "Boundary Integral Methods for Lifting Bodies with Vortex Wakes.", Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1996.

Appendices

Appendix A. Full Scale Experiments Performed at the United States Naval Academy to Determine the Effects of Body Angle and Control Surface Deflection

NAVAL ACADEMY REMUS NO WAVE STEADY FORCE TEST RUNS
(depth is measured to top of vehicle at strut center)

Run #	Rudder Angle (deg)	Stern Plane Angle (deg)	Pitch Angle (deg)	Yaw Angle (deg)	Speed (ft/s)	Depth (meters)	V/ sqrt(gL)	depth/L
N1	0	0	0	0	6.8	0.762	0.526	0.481
N2	0	0	0	8	3.4	0.762	0.263	0.481
N3	0	0	8	0	6.8	0.762	0.526	0.481
N4	0	7	0	0	6.8	0.762	0.526	0.481
N5	7	0	0	0	3.4	0.762	0.263	0.481
N6	0	0	0	0	1.69	0.381	0.131	0.240
N7	0	0	0	0	3.4	0.381	0.263	0.240
N8	0	0	0	0	6.8	0.381	0.526	0.240
N9	7	0	0	0	6.8	0.381	0.526	0.240
N10	0	7	0	0	6.8	0.381	0.526	0.240
N11	0	0	4	0	6.8	0.381	0.526	0.240
N12	0	0	8	0	6.8	0.381	0.526	0.240
N13	0	0	0	4	6.8	0.381	0.526	0.240
N14	0	0	0	8	6.8	0.381	0.526	0.240
N15	0	0	0	0	1.69	0.1905	0.131	0.120
N16	0	0	0	0	6.8	0.1905	0.526	0.120
N17	0	0	0	8	6.8	0.1905	0.526	0.120
N18	0	0	8	0	6.8	0.1905	0.526	0.120
N19	0	0	-8	0	6.8	0.1905	0.526	0.120
N20	0	7	0	0	6.8	0.1905	0.526	0.120
N21	0	7	-8	0	6.8	0.1905	0.526	0.120
N22	7	0	0	0	6.8	0.1905	0.526	0.120
N23	7	0	-8	0	6.8	0.1905	0.526	0.120
N24	7	7	0	0	6.8	0.1905	0.526	0.120

Appendix B. Model Scale Experiments Performed at MIT to Determine the Forces and Moments Due to Unsteady Motion

Test #	Water Depth	Subm.	Yamp	Yfreq	Zamp	Zfreq	Yaw		Pitch		Velocity
	(m)	(m)	Ampl. (m)	Freq.	Ampl. (m)	Freq.	Ampl.(Deg)	Freq.	Ampl.(Deg)	Freq.	(m/sec)
OSC07	0.795	0.543	0.1	0.4028							0.333
OSC15	0.795	0.252	0.1	0.4028							1
OSC17	0.795	0.398	0.1	0.7935							0.667
OSC24	0.795	0.543	0.1	0.4028							1
OSC25	0.795	0.252	0.1	0.4028							0.333
OSC32	0.795	0.488			0.1	0.402831					0.333
OSC40	0.795	0.272			0.1	0.402831					1
OSC42	0.795	0.398			0.1	0.793455					0.667
OSC49	0.795	0.488			0.1	0.402831					1
OSC50	0.795	0.272			0.1	0.402831					0.333
OSC54	0.795	0.252					10	1.19625			0.333
OSC55	0.795	0.252					10	1.19625			1
OSC57	0.795	0.543					10	0.402831			0.333
OSC58	0.795	0.543					10	1.19625			0.333
OSC65	0.795	0.252					10	0.402831			1
OSC67	0.795	0.398					10	0.793455			0.667
OSC71	0.795	0.543					10	1.19625			1
OSC79	0.795	0.252							10	1.1963	0.333
OSC82	0.795	0.543							10	0.4028	0.333
OSC83	0.795	0.543							10	1.1963	0.333
OSC92	0.795	0.398							10	0.7935	0.667
OSC100	0.795	0.252							10	0.4028	0.333

Appendix C. AutoanalyzeXls.m

```
% AutoanalyzeXls.m

% Erik Oller, 2003

% This program uses other programs to automatically analyze all the data
%    files.
% Operational Overview
% 1. User starts this program to perform analysis of experimental data
%    files.
% 2. User selects the series of experiments to be analyzed.
% 3. The program opens an excel file called "MIT Test Plan.xls" and
%    imports the data from the worksheet for the selected series. The
%    matrix seriesnum contain the numerical data from the worksheet
%    and the matrix seriestext contains the text data from the
%    worksheet.
% 4. The program runs the appropriate analysis program for each test
%    series. For steady force tests, the analysis program is
%    "AnalyzemodXlsSF.m". For all other tests, the analysis program
%    is "AnalyzemodXls.m".
% 5. The called analysis program analyzes the raw data and writes the
%    will display that the data file does not exist.
% 6. When all data files listed in the test plan have been analyzed,
%    file.
% 7. The following files must be in the same directory:
%    "AutoanalyzeXls.m"
%    "AnalyzemodXlsSF.m"
%    "AnalyzemodXls.m"
%    "MIT Test Plan.xls"
%    All Data Files to be analyzed.

% Initialize the workspace by clearing all variables and closing all
%   windows.
close all;
clear all;


% Determine which set of tests to analyze.
fprintf(' 1: Horizontal Plane \n')
fprintf(' 2: Vertical Plane \n')
fprintf(' 3: Pure Sway \n')
fprintf(' 4: Pure Heave \n')
fprintf(' 5: Pure Pitch \n')
fprintf(' 6: Pure Yaw \n')
fprintf(' 7: Mass Matrix \n')
```

```
fprintf(' 8: Steady Force \n')
fprintf(' 9: Inertial Calculation Checks \n')
fprintf('10: Miscellaneous \n')
fprintf('11: Oscillation Tests \n')

series = input('Which Test Series? \n');

% Select the worksheet in "MIT Test Plan.xls" and the output file based
%   upon the test series.
switch series
case 1
   worksheet = 'Hor Plane';
   outputfile = 'HorPlaneOut.txt';
case 2
   worksheet = 'Compensating Vert Plane';
   outputfile = 'VertPlaneOut.txt';
case 3
   worksheet = 'Pure Sway';
   outputfile = 'SwayOut.txt';
case 4
   worksheet = 'Pure Heave';
   outputfile = 'HeaveOut.txt';
case 5
   worksheet = 'Pure Pitch';
   outputfile = 'PitchOut.txt';
case 6
   worksheet = 'Pure Yaw';
   outputfile = 'YawOut.txt';
case 7
   worksheet = 'Mass Matrix Tests';
   outputfile = 'MassTestsOut.txt';
case 8
   worksheet = 'Steady Force';
   outputfile = 'SteadyForceOut.txt';
case 9
   worksheet = 'Mass Matrix Tests';
   outputfile = 'InertiaCalcTestsOut.txt';
case 10
   worksheet = 'Miscellaneous';
   outputfile = 'MiscellaneousOut.txt';
case 11
   worksheet = 'Oscillation';
   outputfile = 'OscillationOut.txt';
end

% Import the test filenames and test conditions from the selected
```

```
% worksheet.
[seriesnum,seriestext] = xlsread('MIT Test Plan',worksheet);
numtests = size(seriesnum,1)-2;

if series ==11 % For an unknown reason, the oscillation test worksheet
   % imports differently from the other worksheets.
   numtests =size(seriesnum,1);
   for testindex = 1:numtests
      run(testindex) = (seriestext(testindex+2,3));
      DandS(testindex,1) = seriesnum(testindex,1);
      DandS(testindex,2) = seriesnum(testindex,2);
   end
else
   for testindex = 1:numtests
      run(testindex) = (seriestext(testindex+2,3));
      DandS(testindex,1) = seriesnum(testindex+2,1);
      DandS(testindex,2) = seriesnum(testindex+2,2);
   end
end

% Display the first and last files to be analyzed.
firsttest=char(run(1));
lasttest=char(run(numtests));
fprintf('Autoanalyze will proceed from %s to %s \n', firsttest,lasttest)

% Open the output file.
warning off
delete(outputfile)
warning on
manyrowsfid=fopen(outputfile,'a');

% Display which file is being processed and process the test file. Send
% the test condition data that can not be extracted for the data file.
for fileindex = 1:numtests
   fname = char(run(fileindex));
   fprintf('Processing %s\n',fname)

   Depth = DandS(fileindex,1);
   Submergence = DandS(fileindex,2);
   if series == 8 % Steady force tests.
      YawAngle = seriesnum(fileindex+2,3);
      PitchAngle = seriesnum(fileindex+2,4);
      Rudder = 0;
      SternPlanes = 0;
      analyzemodxlssf;
   else % All unsteady motion tests.
```

```
        YawAngle = 0;
        PitchAngle = 0;
        Rudder = 0;
        SternPlanes = 0;
        analyzemodxls
    end


end

% Display that processing is complete and close the output file.
fprintf('Processing Complete');
status = fclose(manyrowsfid);
```

Appendix D. Sample Data File

This is the first portion of the data from experiment OSC07. OSC07 involved oscillations in sway with forward velocity. The two rows of text at the top were added by the author for clarification and are not actually part of the data file. Not shown are the additional rows at the end of the data file which contain the ordered parameters for the experiment.

Forces and Moments						Wave Height		Position Data in Controller Counts					Time Data			Alternate Time Data	
Sway (V)	Heave (V)	Surge (V)	Pitch (V)	Yaw (V)	Roll (V)	Channels (Not Used)		X	Y	Z	Roll	Pitch	Time Data	Date Data	DeltaT	Time Date	DeltaT
0.817	-0.031	0.086	0.086	0.102	-0.115	0.002	0.001	165	316744	798231	503551	13916	37142.76	37725	0	0.027	0
0.322	-0.099	0.005	0.021	-0.456	-0.026	0	0.001	166	319080	798231	503551	13916	37142.8	37725	0.039063	0.073	0.046
0.667	-0.057	0.005	0.072	-0.796	0.002	-0.001	0.001	227	321424	798231	503551	13916	37142.84	37725	0.039063	0.116	0.043
0.361	-0.006	0.014	-0.091	0.784	-0.079	0.002	0.001	416	323714	798231	503551	13916	37142.88	37725	0.042969	0.156	0.04
-0.052	-0.016	0.005	0.149	-0.492	-0.045	0.001	0.002	836	326137	798230	503551	13916	37142.92	37725	0.039063	0.198	0.042
0.413	-0.036	0	0.584	-0.238	0.012	0.001	0.003	1350	327779	798231	503551	13916	37142.96	37725	0.039063	0.241	0.043
0.638	-0.074	0.003	0.347	-0.371	-0.079	0.001	0.001	2548	330343	798230	503551	13916	37143	37725	0.039063	0.269	0.028
0.294	-0.086	-0.005	0.489	0.129	-0.111	0	0.001	4159	332879	798232	503551	13916	37143.04	37725	0.039063	0.31	0.041
-0.027	-0.046	0.005	0.295	-0.067	-0.049	0.001	0	6398	335523	798230	503551	13916	37143.08	37725	0.042969	0.352	0.042
0.229	-0.101	0.008	-1.175	0.169	-0.065	0.001	0.001	9262	338054	798231	503551	13916	37143.12	37725	0.039063	0.395	0.043
-0.008	0.011	0.013	-1.092	0.119	-0.004	0	0	12747	340486	798230	503551	13916	37143.16	37725	0.039063	0.437	0.042
0.286	0.024	0.004	0.519	-0.023	0.033	0	0.001	16652	342728	798231	503551	13916	37143.2	37725	0.039063	0.477	0.04
0.345	-0.059	-0.005	0.794	-0.127	-0.118	-0.006	0	21057	344921	798230	503551	13916	37143.24	37725	0.042969	0.518	0.041
0.043	-0.167	0.004	0.646	0.224	-0.166	0	0.001	25744	347005	798231	503551	13916	37143.28	37725	0.039063	0.558	0.04
-0.057	0.042	0.011	0.361	-0.055	0.07	0	0.002	29136	348342	798231	503551	13916	37143.32	37725	0.039063	0.587	0.029
-0.341	-0.067	0.015	-0.454	0.676	0.085	0	0.002	34368	350165	798230	503552	13916	37143.36	37725	0.042969	0.629	0.042
-0.234	0.013	0.011	-0.067	0.159	0.026	0.001	0.007	39725	351745	798232	503551	13916	37143.4	37725	0.039063	0.672	0.043
-0.069	0.184	0.015	0.009	-0.448	-0.033	0	0.002	45221	353150	798231	503551	13916	37143.44	37725	0.039063	0.714	0.042
0.208	-0.013	0.031	-0.441	0.257	-0.029	0.001	0.001	50493	354288	798231	503551	13916	37143.48	37725	0.039063	0.754	0.04
-0.189	-0.152	0.012	0.088	-0.165	0.004	0.001	0.001	55941	355252	798232	503551	13916	37143.52	37725	0.042969	0.797	0.043
-0.843	-0.053	0.027	0.165	0.834	0.084	-0.001	0	61351	355897	798232	503551	13916	37143.56	37725	0.039063	0.839	0.042
-0.409	-0.003	0.017	0.167	-0.171	0.021	0.001	0	64685	356137	798231	503551	13916	37143.6	37725	0.039063	0.88	0.041
-0.109	0.034	0.014	0.562	0.097	0.039	0.001	0.001	70086	356319	798231	503551	13916	37143.64	37725	0.039063	0.908	0.028
-0.173	-0.03	0.01	-0.12	0.203	-0.003	0.001	0	75293	356303	798231	503551	13916	37143.68	37725	0.042969	0.949	0.041
-0.205	-0.018	0.007	-0.403	-0.553	0.049	0	0	80797	356089	798231	503551	13916	37143.72	37725	0.039063	0.991	0.042
-0.698	-0.089	0.013	0.284	0.863	0.061	0.001	0.001	86001	355606	798232	503551	13916	37143.76	37725	0.039063	1.031	0.04
-0.97	0.019	0.015	0.106	0.113	-0.007	0.001	0.001	91428	354722	798232	503551	13916	37143.8	37725	0.039063	1.074	0.043
-0.219	-0.052	0.012	0.192	-0.204	0.035	0	0	96569	353570	798231	503551	13916	37143.84	37725	0.042969	1.114	0.04
-0.267	0.033	0.014	0.26	0.544	0.066	0	0.002	101985	352184	798231	503551	13916	37143.88	37725	0.039063	1.157	0.043
-0.382	0.05	0.065	-0.104	-0.105	-0.024	0	0	107466	350626	798231	503551	13916	37143.92	37725	0.039063	1.199	0.042
-0.141	-0.035	0.012	-0.241	-0.123	0.036	0.001	0	111116	349500	798231	503551	13916	37143.96	37725	0.039063	1.228	0.029
-0.668	0.002	0.019	0.342	0.669	0.02	0.001	0	116324	347693	798231	503551	13916	37144	37725	0.042969	1.268	0.04
-0.688	0.077	0.014	0.321	-0.271	0.059	0	0.001	121748	345560	798231	503551	13916	37144.04	37725	0.039063	1.311	0.043
-0.136	-0.047	-0.004	0.218	0.261	-0.003	0.001	0.001	127162	343249	798231	503551	13916	37144.08	37725	0.039063	1.353	0.042
-0.19	0.063	0.003	0.39	-0.264	0.069	0.001	0.001	132580	340876	798231	503551	13916	37144.12	37725	0.039063	1.395	0.042
-0.062	-0.066	0.008	-0.456	-0.179	0.039	0.001	0.001	137793	338568	798231	503551	13916	37144.16	37725	0.042969	1.436	0.041
-0.208	-0.041	0.011	-0.45	-0.086	-0.018	0	0.001	143271	336076	798231	503551	13916	37144.2	37725	0.039063	1.478	0.042
-0.6	-0.094	0.013	0.581	0.243	0.072	0.001	0.001	147161	334225	798232	503551	13916	37144.24	37725	0.039063	1.509	0.031
-0.333	0.107	0.009	-0.048	0.169	-0.048	0	0	152338	331709	798232	503551	13916	37144.28	37725	0.039063	1.563	0.054
-0.061	-0.154	0.01	-0.065	-0.001	-0.025	-0.003	0.001	157760	329060	798231	503551	13916	37144.32	37725	0.042969	1.592	0.029
0.509	0.079	0.001	0.307	-0.365	-0.025	0.002	0	163177	326519	798231	503551	13916	37144.36	37725	0.039063	1.634	0.042
0.11	-0.038	0.001	-0.117	-0.32	0.117	0	0.002	168907	323971	798231	503551	13916	37144.4	37725	0.039063	1.679	0.045
-0.183	-0.019	0.015	-0.508	-0.109	0.009	0.001	-0.001	174105	321717	798232	503551	13916	37144.44	37725	0.039063	1.719	0.04
-0.264	-0.197	0.014	0.302	0.189	-0.049	0.001	0.001	179560	319427	798232	503551	13916	37144.48	37725	0.042969	1.761	0.042
-0.124	-0.011	0.008	0.177	0.261	-0.152	0.001	0	183179	317973	798232	503551	13916	37144.52	37725	0.039063	1.79	0.029
0.584	-0.259	0.022	0.136	-0.581	0.012	0.001	0	188340	316043	798231	503551	13916	37144.56	37725	0.039063	1.83	0.04
0.397	0.078	0.005	0.792	0.021	0.092	0	0.001	193773	314203	798231	503551	13916	37144.6	37725	0.039063	1.873	0.043
0.073	-0.127	0.005	-0.214	-0.107	-0.1	0	0.001	198972	312694	798231	503551	13916	37144.64	37725	0.042969	1.913	0.04
0.153	-0.039	0.013	-0.529	0.065	-0.12	-0.001	0.002	204427	311307	798232	503551	13916	37144.68	37725	0.039063	1.956	0.043
0.091	-0.033	0.012	-0.027	-0.299	0.073	0.001	0.001	209887	310112	798232	503551	13916	37144.72	37725	0.039063	1.998	0.042
0.572	-0.008	0.023	-1.172	-0.164	-0.008	0.002	0.001	215065	309219	798232	503551	13916	37144.76	37725	0.042969	2.038	0.04
0.474	-0.261	0.019	-0.152	0.118	-0.047	0.001	0	220485	308554	798232	503551	13916	37144.8	37725	0.039063	2.081	0.043
0.308	0.067	0.012	0.431	-0.427	0.049	0.001	0	223845	308301	798232	503551	13916	37144.84	37725	0.039063	2.107	0.026
0.676	-0.112	0.01	0.179	-0.08	-0.019	0.001	0	229289	308112	798231	503551	13916	37144.88	37725	0.039063	2.15	0.043
0.12	0.135	0.017	0.102	0.201	-0.069	0.001	0	234754	308141	798231	503551	13916	37144.92	37725	0.042969	2.192	0.042
0.345	-0.029	0.014	0.504	-0.113	-0.03	0.002	0.001	239953	308375	798231	503551	13916	37144.96	37725	0.039063	2.233	0.041
0.592	-0.043	0.011	-0.115	-0.098	-0.062	0.001	0.001	245392	308866	798231	503551	13916	37145	37725	0.039063	2.275	0.042
0.997	-0.122	0.026	0.062	-0.601	-0.046	-0.001	0	250823	309761	798231	503551	13916	37145.04	37725	0.039063	2.317	0.042
0.852	0.007	0.019	0.266	0.224	-0.083	-0.001	0.001	256250	310980	798231	503551	13916	37145.08	37725	0.039063	2.36	0.043
0.216	-0.065	0.013	-0.168	-0.435	0.005	0.001	0.002	261434	312375	798231	503551	13916	37145.12	37725	0.042969	2.4	0.04
0.158	-0.005	0.02	-0.38	0.164	-0.049	-0.021	0.003	265073	313395	798231	503551	13916	37145.16	37725	0.039063	2.429	0.029
0.509	-0.084	0.013	0.315	-0.457	-0.018	-0.001	0.001	270532	315070	798231	503551	13916	37145.2	37725	0.039063	2.483	0.054
0.697	-0.043	0.016	0.169	0.255	-0.115	0.001	0.003	275725	316843	798232	503551	13916	37145.24	37725	0.039063	2.512	0.029
0.476	-0.116	0.016	-0.246	-0.355	-0.014	0	0	280904	318874	798232	503551	13916	37145.28	37725	0.042969	2.552	0.04
0.748	-0.047	0.021	-0.064	-0.639	0.021	0.001	0	286336	321203	798232	503551	13916	37145.32	37725	0.039063	2.594	0.042
0.305	-0.001	0.051	-0.252	0.825	-0.066	0	-0.001	291770	323613	798232	503551	13916	37145.36	37725	0.039063	2.637	0.043
-0.091	0.014	0.012	-0.507	-0.449	-0.05	0.001	0.001	297219	326056	798232	503551	13916	37145.4	37725	0.042969	2.679	0.042
0.382	-0.016	0.021	0.265	-0.11	-0.031	0.001	0.001	300600	327567	798232	503551	13916	37145.44	37725	0.039063	2.72	0.041
0.648	-0.076	0.012	0.397	-0.193	-0.112	0.001	0	306050	330104	798232	503551	13916	37145.48	37725	0.039063	2.762	0.042
0.426	-0.126	0.025	0.019	-0.047	-0.095	0	0.001	311488	332753	798232	503551	13916	37145.52	37725	0.039063	2.791	0.029
0.012	-0.131	0.016	0.711	-0.134	-0.025	0	0.001	316927	335402	798232	503550	13916	37145.56	37725	0.042969	2.833	0.042
0.175	0.014	0.007	0.311	0.11	-0.062	0.001	0.001	322098	337845	798232	503551	13916	37145.6	37725	0.039063	2.874	0.041
-0.033	-0.119	-0.018	-0.181	0.614	-0.076	0.001	0.001	327555	340282	798232	503551	13916	37145.64	37725	0.039063	2.916	0.042

Appendix E. AnalyzemodXlsSF.m

```
% AnalyzemodXlsSF.m

% Erik Oller, 2003

% Performs analyses for Steady Force Tests only

% This program is designed to be called by AutoAnalyzeXls.m and requires
%no manual intervention.

% Initialize matrices.
[Data] = 0;
[Datain] =0;

% Get the input data and find the length of the file.
infname = strcat(fname,'.xls');
existencecheck = exist(infname); % Determines if the input file exists.
if existencecheck ==0
   fprintf('%s does not exist.\n',infname)
   return
end

Datain = xlsread(infname); % Reads the input file.
NumLines = size(Datain,1)-4; % Gets the number of data samples in
                   %   the input file.

% Gets the date and converts from EXCEL to MATLAB format.
Date=datestr(Datain(1,15)-36525,2);

% Build a date-time string to ensure the correct gains are applied.
% Add 693960 to the date to get number of days from 0000.
% Divide the number of minutes by 86400 to get fractional days.
% MATLAB date serial numbers are in the form:
%    days since 0000.fraction of a day
DateNum=Datain(1,15)+693960+Datain(1,14)/86400;

% Deterimine the ordered velocity.
XSpd = Datain(NumLines+3,10);
if isnan(XSpd)==1
   XSpd = 0;
end

% Determine the ordered distance of travel.
XDist = Datain(NumLines+3,11);
if isnan(XDist)==1
```

```
   XDist = 0;
end

% Determine the ordered length of time of the experiment.
Time = Datain(NumLines+4,5);

% Determine actual sample frequency based upon the measured interval
% between data points.  Necessary because the control software did not
% always sample at the ordered sample rate.
SampleFreq = 1/mean(Datain(1:NumLines,16));


% Drop the first 1.2 seconds of data to account for acceleration.  At
% the sample frequency of 25 Hz, drop the first 30 points.  Shift the
% other points up in the array.
dropgap = round(1.2*SampleFreq);
for I = 1:NumLines-dropgap;
        Datain(I,1:16)=Datain(I+dropgap,1:16);
end

NumLines = NumLines - dropgap;

% Drop data recorded after the model stopped moving along the X-axis
%   for non-Inertial tests.
   NumLines2=NumLines;
   for I = NumLines-1:-1:75
      if (Datain(I,9)==Datain(I+1,9)) %Looks for constant X position.
         NumLines2 = I;
      end
   end
   NumLines = NumLines2;

% Determine the time of the first data point.  Used for converting
%   from time past midnight to time of run.
TimeStart = Datain(1,14);

% The inverse sensitivity matrix (B).  The sensitivity matrix used
% depends on the when the experiment was performed.  Ealry experiments
% used a different load cell than later ones.

if((datenum(Date)>=datenum(2002,07,17))&...
      (datenum(Date)<datenum(2002,10,31)))
   [B] = [ 0.3901  0.0029 -0.0071 -0.0010 -0.0024 -0.0016;
          0.0023  0.3887  0.0016  0.0025 -0.0039  0.0019;
          0.0139  0.0129  1.5006 -0.0002 -0.0215 -0.0018;
          0.0000 -0.0001  0.0016  0.0069  0.0000 -0.0001;
```

```
        -0.0001  0.0000  0.0018  0.0000  0.0070  0.0000;
        -0.0003 -0.0002  0.0000  0.0000  0.0000  0.0108];
else %valid on and after Oct 31, 2002
   [B] = [ 0.3856  0.0020 -0.0016 -0.0008  0.0001 -0.0030;
         0.0023  0.3811 -0.0040  0.0012 -0.0034  0.0031;
         0.0093  0.0024  1.5109  0.0068 -0.0231 -0.0014;
         0.0001  0.0000  0.0013  0.0069  0.0000  0.0000;
         0.0000  0.0000  0.0007  0.0000  0.0069 -0.0001;
        -0.0003  0.0001  0.0000  0.0000  0.0000  0.0106];
end

% Establish Gains and Excitation Voltage
if (DateNum>731434.61458) % August 6, 2002 1445.
   [Gain] = [4000;4000;1000;1000;1000;4000]; % y z x pitch yaw roll
   [Vexc] = [10; 10; 2.5; 10; 10; 10];
end

% Calculate the Conversion Factors (CF) between voltage and force
%   or moment
CFtemp = Gain.*Vexc*10^-6;
CF = zeros(6,6);
CF(1,1)= CFtemp(1);
CF(2,2)= CFtemp(2);
CF(3,3)= CFtemp(3);
CF(4,4)= CFtemp(4);
CF(5,5)= CFtemp(5);
CF(6,6)= CFtemp(6);

% Establish multipliers to convert from controller counts to MKS units.
% Multipliers are different for different time intervals due to system
% upgrades.

%  Between 0800 Oct 24, 2002 and 0800 Dec 1, 2002.
if ((DateNum>731513.33333)&(DateNum<731551.33333))
   xfactor = 3850;
   yfactor = 2410;
   zfactor = 2114;
   pitchfactor = 972;
   yawfactor = 1818;
elseif DateNum>731551.33333 % After 0800 Dec 1, 2002
   xfactor = 3850;
   yfactor = 2410;
   zfactor = 2114;
   pitchfactor = 155;
   yawfactor = 1818;
else % Before Oct 24, 2003
```

```
    xfactor = 3850;
    yfactor = 2410;
    zfactor = 4921;
    pitchfactor = 1111;
    yawfactor = 1025;
end

% Convert from voltages to forces
for I = 1 : NumLines
    Datain(I,1:6) = (CF^-1*B*Datain(I,1:6)')';
    Datain(I,9)  = Datain(I,9)  / xfactor;
    Datain(I,10) = Datain(I,10) / yfactor;
    Datain(I,11) = Datain(I,11) / zfactor;
    Datain(I,12) = Datain(I,12) / yawfactor;
    Datain(I,13) = Datain(I,13) / pitchfactor;
    Datain(I,14) = Datain(I,14) - TimeStart;
end

% Find the mean force
for J=1:6
    MeanForce(J) = real(mean(Datain(1:NumLines,J)));
end

% PROCESSING STAGE
% Shift the origin of the coordinate system from the origin of the load
%   cell to vessel amidships.
shiftlength = 0.0522; % meters
MeanForce(4) = MeanForce(4) - MeanForce(2) * shiftlength; % Pitch
MeanForce(5) = MeanForce(5) + MeanForce(1) * shiftlength; % Yaw

% Print the results to a common row output file.
fprintf(manyrowsfid,'%s \t %s\t %6.2f',fname,Date,Depth);
fprintf(manyrowsfid,'\t%6.2f \t%5.1f \t %5.2f\t%5.1f',...
    Submergence,XSpd,XDist,Time);
fprintf(manyrowsfid,' \t %2.0f \t%2.0f\t %3.1f\t %3.1f',...
    PitchAngle,YawAngle,SternPlanes,Rudder);
fprintf(manyrowsfid,' \t %7.4f',...
    MeanForce(3),MeanForce(1),MeanForce(2),MeanForce(6),MeanForce(4),...
    MeanForce(5));
fprintf(manyrowsfid,'\n');
```

Appendix F. Output File from AnalyzemodXlsSF.m called by AutoanalyzeXls.m for Steady Force Tests

This is part of the output file from AnalyzemodXlsSF.m when called by AutoanalyzeXls.m to analyze steady force tests. The two rows of text at the top were added by the author for clarification and are not actually part of the data file. When using this table please note the following:

SF67-72, 80-90, 144-185 were performed with control surfaces removed.
SF73 was a test to determine the effect of system noise with no motion.
SF74 was performed with no motion but with a 4 oz weight attached near midships.
SF75 was performed with no motion but with a 2 oz weight attached near midships.

Test #	Date	Depth	Subm.	Spd	Xdist	Time	Pitch	Yaw	SP	Rudder	X	Y	Z	Roll	Pitch	Yaw
		cm	mm	cm/s	cm	s	Degrees	Degrees	Degrees	Degrees	N	N	N	N-m	N-m	N-m
SF61	03/05/03	79.5	356.2	34	700	22	0	0	0	0	6.7506	-0.5651	0.736	-0.001	-0.0746	-0.0205
SF62	03/05/03	79.5	356.2	34	700	22	0	4	0	0	10.7151	-0.1987	0.3827	0.0051	-0.032	0.0089
SF63	03/05/03	79.5	356.2	34	700	22	0	8	0	0	10.9515	-0.1364	0.4721	0.0067	-0.0244	0.0176
SF64	03/05/03	79.5	356.2	68	700	11	0	0	0	0	5.3257	-0.5732	0.7378	-0.0013	-0.0782	-0.0175
SF65	03/05/03	79.5	356.2	68	700	11	0	4	0	0	7.3517	0.0361	0.7981	-0.0072	-0.0851	0.0612
SF66	03/05/03	79.5	356.2	68	700	11	0	8	0	0	6.7879	0.3516	0.6998	-0.0046	-0.0372	0.1184
SF67	03/05/03	79.5	356.2	34	700	22	0	0	0	0	19.8251	0.6296	-0.2936	-0.0133	-0.0042	0.0361
SF68	03/05/03	79.5	356.2	34	700	22	0	4	0	0	20.801	0.3637	-0.0698	-0.0069	-0.0256	0.0391
SF69	03/05/03	79.5	356.2	34	700	22	0	8	0	0	17.9008	0.6628	-0.175	-0.0165	-0.0003	0.0694
SF70	03/05/03	79.5	356.2	68	700	11	0	0	0	0	18.9798	0.5325	0.1027	-0.0176	-0.0432	0.0359
SF71	03/05/03	79.5	356.2	68	700	11	0	4	0	0	17.655	0.5197	-0.1675	-0.0142	-0.0002	0.1008
SF72	03/05/03	79.5	356.2	68	700	11	0	8	0	0	17.5917	0.9081	-0.0402	-0.017	0.0217	0.1842
SF73	03/05/03	79.5	356.2	0	0	20	0	0	0	0	-3.8807	0.1229	-0.1337	-0.0056	0.0123	0.0093
SF74	03/07/03	79.5	356.2	0	0	20	0	0	0	0	17.5033	-0.1331	1.0175	0.0292	-0.0721	-0.014
SF75	03/07/03	79.5	356.2	0	0	20	0	0	0	0	18.2672	-0.3689	0.961	0.0215	-0.054	-0.0237
SF76	04/04/03	79.5	356.2	100	700	10	0	0	0	0	14.4912	-0.0623	0.3053	0.0178	-0.098	-0.0183
SF77	04/04/03	79.5	356.2	100	700	10	0	4	0	0	7.3252	0.2929	0.257	0.0059	-0.0875	0.0822
SF78	04/04/03	79.5	356.2	100	700	10	0	8	0	0	7.8204	0.5732	0.469	0.0035	-0.0797	0.1728
SF78B	04/04/03	79.5	356.2	100	700	10	0	12	0	0	6.3188	1.2356	0.8194	0.0007	-0.0858	0.2656
SF79	04/04/03	79.5	356.2	100	700	10	0	16	0	0	5.8774	2.1367	0.0009	-0.0019	-0.1104	0.3389
SF80	04/04/03	79.5	356.2	100	700	10	0	0	0	0	8.6834	0.0818	0.5544	-0.0087	-0.0791	-0.0039
SF81	04/04/03	79.5	356.2	100	700	10	0	4	0	0	12.1557	0.4921	0.4522	-0.0008	-0.0802	0.1348
SF82	04/04/03	79.5	356.2	100	700	10	0	8	0	0	7.9449	1.0495	0.0418	-0.0179	-0.0306	0.2852
SF82B	04/04/03	79.5	356.2	100	700	10	0	12	0	0	10.9136	1.0141	0.4823	0	-0.1804	0.3774
SF83	04/04/03	79.5	356.2	100	700	10	0	16	0	0	8.0716	2.2319	-0.2717	-0.0166	-0.0772	0.5317
SF84	04/04/03	79.5	356.2	100	700	10	0	0	0	0	23.3779	-0.1231	0.3215	0.0153	-0.0857	-0.0266
SF85	04/04/03	79.5	356.2	100	700	10	0	4	0	0	26.1843	0.1264	0.1717	0.016	-0.0563	0.1104
SF86	04/04/03	79.5	356.2	100	700	10	0	8	0	0	22.1398	0.4683	0.5599	0.0112	-0.0638	0.2422
SF87	04/04/03	79.5	356.2	100	700	10	0	12	0	0	20.9844	0.9228	0.5964	0.0115	-0.0872	0.3704
SF88	04/04/03	79.5	356.2	100	700	10	4	0	0	0	21.7807	-0.3638	0.1393	0.0182	-0.0021	-0.0355
SF89	04/04/03	79.5	356.2	100	700	10	8	0	0	0	23.6841	-0.569	-0.1426	0.0206	0.1086	-0.0427
SF90	04/04/03	79.5	356.2	100	700	10	12	0	0	0	21.4243	-0.475	-0.5479	0.0202	0.2067	-0.036
SF91	04/04/03	79.5	356.2	100	700	10	0	0	0	0	7.5073	0.0728	-0.015	-0.0008	-0.1033	-0.0076
SF92	04/04/03	79.5	356.2	100	700	10	0	4	0	0	10.8973	0.7809	-0.1784	-0.0129	-0.0639	0.1026
SF93	04/04/03	79.5	356.2	100	700	10	0	8	0	0	6.0916	1.2924	0.3105	-0.0203	-0.0868	0.2104
SF94	04/04/03	79.5	356.2	100	700	10	0	12	0	0	8.3035	1.6519	0.2402	-0.0108	-0.078	0.2797
SF95	04/04/03	79.5	356.2	100	700	10	4	0	0	0	6.013	-0.0121	0.0331	-0.0065	-0.0191	-0.0095
SF96	04/04/03	79.5	356.2	100	700	10	8	0	0	0	8.3074	0.0053	-0.3269	-0.0063	0.0113	-0.0062
SF97	04/04/03	79.5	356.2	100	700	10	12	0	0	0	7.1199	0.0485	-0.993	-0.0044	0.0543	-0.0039
SF98	04/07/03	79.5	356.2	100	700	10	0	0	0	0	14.3126	-0.6259	0.5736	0.0084	-0.1263	-0.0457
SF99	04/07/03	79.5	356.2	75	700	13	0	0	0	0	17.1046	-0.6789	0.2246	0.0136	-0.0526	-0.0474
SF100	04/07/03	79.5	210.9	100	700	10	0	0	0	0	28.6384	-0.3125	-0.3894	0.0441	-0.0554	-0.0408
SF101	04/07/03	79.5	210.9	75	700	13	0	0	0	0	21.3003	-0.3711	-0.4691	0.0403	-0.0416	-0.0368
SF102	04/07/03	79.5	501.5	100	700	10	0	0	0	0	8.4768	0.2046	0.2218	-0.0085	-0.1139	0
SF103	04/07/03	79.5	501.5	75	700	13	0	0	0	0	8.104	0.2389	-0.1237	-0.0059	-0.0566	0.0065
SF104	04/07/03	79.5	356.2	100	700	10	0	4	0	0	0.0083	-0.1356	0.7407	-0.0239	-0.1217	0.0659
SF105	04/07/03	79.5	356.2	75	700	13	0	4	0	0	-5.3657	0.1665	0.4406	-0.0349	-0.0367	0.0572
SF106	04/07/03	79.5	210.9	100	700	10	0	4	0	0	7.9829	0.5966	-0.2151	-0.0043	-0.0736	0.0959
SF107	04/07/03	79.5	210.9	75	700	13	0	4	0	0	11.5173	0.4629	-0.9813	0.0007	-0.0326	0.0647
SF108	04/07/03	79.5	501.5	100	700	10	0	4	0	0	13.1946	0.6244	-0.1566	0.0198	-0.0778	0.0915
SF109	04/07/03	79.5	501.5	75	700	13	0	4	0	0	17.8258	0.6527	-0.3395	0.0142	-0.0294	0.0722
SF110	04/07/03	79.5	356.2	100	700	10	0	8	0	0	-7.1228	0.7253	0.6748	-0.0415	-0.1104	0.1874
SF111	04/07/03	79.5	356.2	75	700	13	0	8	0	0	-4.6664	0.2552	0.8404	-0.0352	-0.044	0.1106
SF112	04/07/03	79.5	210.9	100	700	10	0	8	0	0	5.9871	1.1071	0.4481	-0.0174	-0.0857	0.199
SF113	04/07/03	79.5	210.9	75	700	13	0	8	0	0	9.4164	0.8015	0.934	-0.0123	-0.033	0.1305
SF114	04/07/03	79.5	501.5	100	700	10	0	8	0	0	-5.3702	0.4664	0.8559	-0.0415	-0.1143	0.1744
SF115	04/07/03	79.5	501.5	75	700	13	0	8	0	0	-5.1133	0.0701	0.808	-0.0382	-0.0898	0.1038
SF116	04/07/03	79.5	356.2	100	700	10	0	12	0	0	-4.4508	1.5926	0.1168	-0.0049	-0.0714	0.2824
SF117	04/07/03	79.5	356.2	75	700	13	0	12	0	0	-1.2336	0.8619	0.0513	0.0016	-0.0279	0.1713
SF118	04/07/03	79.5	210.9	100	700	10	0	12	0	0	7.9814	1.8835	-0.1084	0.0196	-0.0386	0.2878
SF119	04/07/03	79.5	210.9	75	700	13	0	12	0	0	9.5864	1.4279	0.1144	0.0221	0.0022	0.1957
SF120	04/07/03	79.5	501.5	100	700	10	0	12	0	0	-7.2312	1.2275	0.894	-0.0446	-0.1012	0.2688
SF121	04/07/03	79.5	501.5	75	700	13	0	12	0	0	1.4662	0.8562	0.0813	0.0017	-0.0339	0.1749
SF122	04/07/03	79.5	356.2	100	700	10	4	0	0	0	1.5423	-0.272	-0.0287	0.0086	-0.0199	-0.0161
SF123	04/07/03	79.5	356.2	75	700	13	4	0	0	0	-1.8708	-0.1491	0.0406	0.0037	-0.003	-0.0077
SF124	04/07/03	79.5	210.9	100	700	10	4	0	0	0	11.5445	0.3353	-1.0374	0.0309	0.0335	0.0038
SF125	04/07/03	79.5	210.9	75	700	13	4	0	0	0	14.0924	0.2787	-0.1434	0.0275	0.0337	0.0086

Test #	Date	Depth	Subm.	Spd	Xdist	Time	Pitch	Yaw	SP	Rudder	X	Y	Z	Roll	Pitch	Yaw
		cm	mm	cm/s	cm	s	Degrees	Degrees	Degrees	Degrees	N	N	N	N-m	N-m	N-m
SF126	04/07/03	79.5	501.5	100	700	10	4	0	0	0	1.5582	-0.3684	-0.1921	0.0073	0.0015	-0.0231
SF127	04/07/03	79.5	501.5	75	700	13	4	0	0	0	1.7731	-0.0883	-0.192	0.0027	0.0111	-0.0039
SF128	04/07/03	79.5	356.2	100	700	10	8	0	0	0	-4.1236	-0.1895	-0.6621	0.0044	0.0651	-0.0037
SF129	04/07/03	79.5	356.2	75	700	13	8	0	0	0	-2.5511	-0.2603	-0.4944	0.0051	0.0613	-0.0105
SF130	04/07/03	79.5	210.9	100	700	10	8	0	0	0	5.4903	0.0547	-0.8964	0.0277	0.0674	0.0062
SF131	04/07/03	79.5	210.9	75	700	13	8	0	0	0	8.7811	0.239	-0.7814	0.026	0.0676	0.0145
SF132	04/07/03	79.5	501.5	100	700	10	8	0	0	0	-1.9805	-0.1559	-0.6271	-0.0014	0.0606	-0.0079
SF133	04/07/03	79.5	501.5	75	700	13	8	0	0	0	1.3509	-0.0493	-0.4612	0.0006	0.0524	-0.0012
SF134	04/07/03	79.5	356.2	100	700	10	12	0	0	0	2.9365	-0.4546	-1.0079	0.0183	0.1172	-0.0088
SF135	04/07/03	79.5	356.2	75	700	13	12	0	0	0	0.1206	-0.2504	-0.7061	0.0112	0.0867	-0.0135
SF136	04/07/03	79.5	210.9	100	700	10	12	0	0	0	10.111	0.2624	-1.6039	0.027	0.1288	0.0077
SF137	04/07/03	79.5	210.9	75	700	13	12	0	0	0	13.6603	0.2164	-1.1318	0.0324	0.1141	0.0176
SF138	04/07/03	79.5	501.5	100	700	10	12	0	0	0	1.0817	-0.526	-0.8771	0.0042	0.1005	-0.0276
SF139	04/07/03	79.5	501.5	75	700	13	12	0	0	0	0.7022	-0.1462	-0.7014	0.0016	0.0688	-0.004
SF140	04/07/03	79.5	501.5	100	700	10	0	8	0	0	8.2723	0.8434	0.7545	-0.0361	-0.0988	0.194
SF141	04/07/03	79.5	501.5	75	700	13	0	8	0	0	6.0705	0.7525	0.0117	-0.006	-0.0367	0.134
SF142	04/07/03	79.5	210.9	100	700	10	8	0	0	0	14.5032	0.172	-0.7051	0.0024	0.0661	0.0052
SF143	04/07/03	79.5	210.9	75	700	13	4	0	0	0	19.6126	0.2523	-0.4773	0.005	0.013	0.0121
SF144	04/07/03	79.5	356.2	100	700	10	0	0	0	0	11.9977	-0.3885	0.1854	0.0021	-0.0888	-0.0355
SF145	04/07/03	79.5	356.2	75	700	13	0	0	0	0	12.6364	0.0928	-0.0909	-0.0048	-0.0267	-0.0034
SF146	04/07/03	79.5	210.9	100	700	10	0	0	0	0	11.2486	0.0564	-0.3442	0.0016	-0.0728	-0.0127
SF147	04/07/03	79.5	210.9	75	700	13	0	0	0	0	12.6827	0.0567	-0.2317	0.004	-0.0304	-0.008
SF148	04/07/03	79.5	501.5	100	700	10	0	0	0	0	7.8708	-0.3903	0.2886	-0.0143	-0.0798	-0.0368
SF149	04/07/03	79.5	501.5	75	700	13	0	0	0	0	7.3999	-0.2902	0.1767	-0.0129	-0.0526	-0.0216
SF150	04/07/03	79.5	356.2	100	700	10	0	4	0	0	-0.8426	0.0721	0.4963	-0.0322	-0.0894	0.1153
SF151	04/07/03	79.5	356.2	75	700	13	0	4	0	0	2.0863	-0.1088	0.6661	-0.0248	-0.0585	0.0663
SF152	04/07/03	79.5	210.9	100	700	10	0	4	0	0	8.6837	0.2023	0.3061	-0.0066	-0.082	0.1125
SF153	04/07/03	79.5	210.9	75	700	13	0	4	0	0	11.8808	0.2309	-0.4769	-0.0046	-0.003	0.0779
SF154	04/07/03	79.5	501.5	100	700	10	0	4	0	0	4.0575	0.1175	0.3141	-0.0248	-0.0839	0.1181
SF155	04/07/03	79.5	501.5	75	700	13	0	4	0	0	11.2072	0.0949	0.188	-0.0183	-0.0416	0.0777
SF156	04/07/03	79.5	356.2	100	700	10	0	8	0	0	1.8573	0.3133	0.6259	-0.0283	-0.0693	0.2394
SF157	04/07/03	79.5	356.2	75	700	13	0	8	0	0	10.5516	0.6441	-0.2885	0.0098	0.0026	0.1728
SF158	04/07/03	79.5	210.9	100	700	10	0	8	0	0	12.9229	1.0197	0.0742	-0.0048	-0.0546	0.2681
SF159	04/07/03	79.5	210.9	75	700	13	0	8	0	0	11.4304	0.4812	0.8981	-0.0097	-0.0339	0.1663
SF160	04/07/03	79.5	501.5	100	700	10	0	8	0	0	-2.0303	0.4854	0.2242	-0.0035	-0.0645	0.253
SF161	04/07/03	79.5	501.5	75	700	13	0	8	0	0	4.6134	0.7048	0.1588	-0.0186	-0.0291	0.1856
SF162	04/07/03	79.5	356.2	100	700	10	0	12	0	0	-5.1568	1.2942	0.4418	-0.0165	-0.0632	0.4016
SF163	04/07/03	79.5	356.2	75	700	13	0	12	0	0	3.8291	0.9787	-0.1571	0.0085	0.0067	0.2613
SF164	04/07/03	79.5	210.9	100	700	10	0	12	0	0	10.272	1.5353	0.1432	-0.0012	-0.038	0.4029
SF165	04/07/03	79.5	210.9	75	700	13	0	12	0	0	13.0686	1.065	-0.8274	0.0037	-0.0076	0.2577
SF166	04/07/03	79.5	501.5	100	700	10	0	12	0	0	-7.5024	1.4947	0.18	-0.017	-0.0564	0.4133
SF167	04/07/03	79.5	501.5	75	700	13	0	12	0	0	6.2886	0.9665	-0.0814	0.0012	-0.0278	0.2643
SF168	04/07/03	79.5	356.2	100	700	10	4	0	0	0	7.7978	-0.6456	0.5741	-0.0051	0.0123	-0.0521
SF169	04/07/03	79.5	356.2	75	700	13	4	0	0	0	9.444	-0.1815	0.1395	-0.0051	0.0376	-0.0216
SF170	04/07/03	79.5	210.9	100	700	10	4	0	0	0	3.7602	0.8067	-0.3934	-0.025	0.0252	0.0285
SF171	04/07/03	79.5	210.9	75	700	13	4	0	0	0	3.9603	0.7468	0.0696	-0.0282	0.0341	0.04
SF172	04/07/03	79.5	501.5	100	700	10	4	0	0	0	-3.5378	-0.1568	-0.0781	0.0035	0.0379	-0.027
SF173	04/07/03	79.5	501.5	75	700	13	4	0	0	0	-3.392	-0.1565	0.047	0.0039	0.0278	-0.0167
SF174	04/07/03	79.5	356.2	100	700	10	8	0	0	0	8.2774	0.2489	-0.6666	0.009	0.1331	-0.0013
SF175	04/07/03	79.5	356.2	75	700	13	8	0	0	0	15.9178	0.3335	-0.6135	0.0198	0.1163	0.0029
SF176	04/07/03	79.5	210.9	100	700	10	8	0	0	0	14.9037	0.3619	-0.6767	0.0128	0.1374	0.0167
SF177	04/07/03	79.5	210.9	75	700	13	8	0	0	0	19.9008	0.7155	-0.5469	0.0154	0.1086	0.0298
SF178	04/07/03	79.5	501.5	100	700	10	8	0	0	0	13.905	-0.3111	-0.2381	0.0055	0.1145	-0.0191
SF179	04/07/03	79.5	501.5	75	700	13	8	0	0	0	12.3352	-0.1277	-0.1732	0.002	0.0754	-0.0116
SF180	04/07/03	79.5	356.2	100	700	10	12	0	0	0	18.163	0.2045	-1.0493	0.0147	0.2235	-0.003
SF181	04/07/03	79.5	356.2	75	700	13	12	0	0	0	17.4979	0.1598	-0.6601	0.0119	0.1675	0.008
SF182	04/07/03	79.5	210.9	100	700	10	12	0	0	0	14.9671	-0.0355	-0.5625	0.0014	0.1805	-0.0027
SF183	04/07/03	79.5	210.9	75	700	13	12	0	0	0	16.0427	-0.0323	-0.2541	0.003	0.1056	-0.0015
SF184	04/07/03	79.5	501.5	100	700	10	12	0	0	0	2.9632	-0.1106	-0.8265	0.0007	0.2131	-0.0102
SF185	04/07/03	79.5	501.5	75	700	13	12	0	0	0	0.0831	-0.2192	-0.3446	-0.005	0.1249	-0.0096

Appendix G. AnalyzemodXls.m

```
% AnalyzemodXls.m
% Erik Oller, 2003

% Series Specific Treatment
% Series 7: Inertial Results are not subtracted for Mass Matrix Tests.

% Variables
% ActualPhasef1,f2,f3:   The actual phase of the forces at the
%          oscillation frequency.
% Ampl:      The amplitude of the data point.
% avg(13):            The mean value of the force and motions for each
%          experiment.
% B(6,6):             The inverse sensitivity matrix for the load cell.
% C(6):      The values of capacitances in the electrical filters in
%          microfarads.
% CF(6,6):            The conversion factor matrix for the load cell.
% CFtemp(6):   The conversion factor vector for the load cell data.
% Data(NumPoints,15):   The Datain matrix after interpolation for
%          constant time intervals.
% Date:      The date the experiment was performed.
% DateNum:            The numeric MATLAB code for the date and time the
%  experiment was
%               performed.
% Datain:    Holds all of the input data as read from the xls file.
%               Column 1:       Y force
%                       2:      Z Force
%                       3:      X Force
%                       4:      Pitch Moment
%                       5:      Yaw Moment
%                       6:      Roll Moment
%                       7:      Reserved for wave data
%                       8:      Reserved for wave data
%                       9:      X position
%                       10:     Y position
%                       11:     Z Position
%                       12:     Yaw Position
%                       13:     Pitch Position
%                       14:     Time
%                       15:     Date
%                       16:     Time Interval between samples
% DeltaT:             The average time interval between samples in seconds.
% Depth:              The depth of the experiment in cm.
% dropgap:            The number of data samples dropped from the start of
%          the data in order to allow for acceleration.
% DrivingFreq:   The frequency of driving force oscillation in Hz.
%          This is omega converted into Hz.
% Duration:           The total length of time of the experiment.
% existencecheck:  Holds the results of the check for the input file.
% f1:        The ordered frequency of oscillation in Hz.
% f2:        The first harmonic of the oscillation frequency in Hz.
% f3:        The second harmonic of the oscillation frequency in Hz.
% fname:     The name of the file to be analyzed.
% Fpitch:             Calculated inertial force in pitch in Newton-meters.
```

% Fx: Calculated inertial force in the X direction in Newtons.
% Fy: Calculated inertial force in the Y direction in Newtons.
% Fyaw: Calculated inertial force in yaw in Newton-meters.
% Fz: Calculated inertial force in the Y direction in Newtons.
% Gain: The gain of the load cell amplifier.
% infname: The name of the input file with the .xls extension added.
% indexhigh: The index of the row of data representing the frequency
% to begin filtering.
% InertialForce: The Calculated Inertial Force in N or N-m.
% Interval: The time duration of the fft data.
% Iyy: First moment of inertia abouthte Y-axis.
% Izz: First moment of inertia about the Z-axis.
% k: An index used in filtering.
% LocalMax: The local maximum of data. Used in finding the peak
% frequencies.
% LocalMaxCounter: An index used to identify the lines of data for the
% local maxima.
% look: The size of the gap between frequency intervals.
% manyrowsfid: The id number of the series row output file.
% From the calling file.
% mass: The mass of the model full of water in kg.
% MotionPhi: The phase of the driving force motion at the frequency
% of the local maxima.
% nonzeroplanes: From the calling file. Indicates nonzero stern
% planes or rudder.
% NumLines: The number of lines in the input file containing force
% and motion data.
% NumLines2: Used to determine when the model stopped traveling in
% the X direction.
% NumPeriods: The number of periods that occur over the duration of
% the experiment.
% NumPoints: The number of points that will be used for the FFT.
% omega: The frequency of oscillation in radians per second.
% Period: The time in seconds of one cycle of motion.
% PhaseIndexf1,f2,f3: The line of Data containing the actual phase of
% the motion at the oscillation frequency.
% phi: The phase shift caused by dropping the first 1.2 seconds of
% data.
% PitchAmp: Amplitude of Pitch oscillation in degrees.
% PitchAngle: Steady pitch angle. Positive is nose up.
% PitchDist: Ordered distance of travel in the Pitch direction in
% degrees.
% pitchfactor: The control factor for gantry motion in pitch.
% PitchFreq: Frequency of Pitch oscillation in Hz.
% PitchPhase: Ordered phase of Pitch oscillation in degrees.
% PitchRadius: The distance from the center of gravity to the axis of
% pitch in cm.
% PitchSpd: Speed in the Pitch direction in degreees/sec.
% PitchSupSpd: Superimposed speed in the Pitch direction in deg/sec.
% PitchSupDir: Direction of superimposed motion.
% R(6): The values of resistances in the electrical filters in
% ohms.
% Results(3,24): Holds all of the peaks of the FFT.
% rowfname: The name of the row output file.
% rowed: The file id number of the row output file.
% Rudder: The angle of the rudder in degrees.

```
% series:          The series of data to be analyzed.
%        Comes from the calling program.
%              1: Horizontal Plane
%              2: Vertical Plane
%              3: Pure Sway
%              4: Pure Heave
%              5: Pure Pitch
%              6: Pure Yaw
%              7: Mass Matrix
%              8: Steady Force
%        9: Check of Inertial Calculations
%       10: Miscellaneous
%       11: Oscillation Tests
% SampleFreq:   Calculated sample frequency based on average interval
%        between datapoints.
% shiftlength:   The distance between the center of gravity and midships
%        in m.
% SternPlanes: The angle of the stern planes in degrees.
% Submergence: The submergence of the model to the top of the hull in
%        mm. From the calling file.
% Summary(3,24):   Holds data from the FFT peak closest to the driving
%        frequency.
% Time:     Ordered length of time of the experiment in seconds.
% TimeStart:     The first recorded time of the experiment.
% Vexc:     The excitation voltage of the amplifier.
% write:   A switch for whether or not to write the individual row
%        output file.
% XAmp:               Amplitude of X oscillation in cm.
% XDist:            Ordered distance of travel in the X direction in cm.
% xfactor:          The control factor for gantry motion in the X direction.
% XFreq:            Frequency of X oscillation in Hz.
% Xg:     The distance from the model center of gravity to the
%        zero point of the load cell.
% xhigh:            The cutoff frequency for low pass filtering.
% XPhase:           Ordered phase of X oscillation in degrees.
% XSpd:               Speed in the X direction in cm/sec.
% XSupSpd:          Superimposed speed in the X direction in cm/sec.
% XSupDir:          Direction of superimposed motion.
% YAmp:               Amplitude of Y oscillation in cm.
% YawAmp:           Amplitude of Yaw oscillation in degrees.
% YawAngle:         The steady yaw angle in degrees.  Positive is nose to
%        stbd.
% YawDist:          Ordered distance of travel in the Yaw direction in
%        degrees.
% yawfactor:    The control factor for gantry motion in yaw.
% YawFreq:          Frequency of Yaw oscillation in Hz.
% YawPhase:         Ordered phase of Yaw oscillation in degrees.
% YawRadius:    The distance from the center of gravity to the strut in
%        cm.
% YawSpd:           Speed in the Yaw direction in degreees/sec.
% YawSupSpd:    Superimposed speed in the Yaw direction in deg/sec.
% YawSupDir:    Direction of superimposed motion.
% yfactor:          The control factor for gantry motion in the Y direction.
% YFreq:            Frequency of Y oscillation in Hz.
% YPhase:           Ordered phase of Y oscillation in degrees.
% YSpd:               Ordered speed of travel in the Y direction in cm/sec.
```

```
% YSupSpd:        Superimposed speed in the Y direction in cm/sec.
% YSupDir:        Direction of superimposed motion.
% ZAmp:              Amplitude of Z oscillation in mm.
% ZDist:          Ordered distance of travel in the Z direction in mm.
% zfactor:        The control factor for gantry motion in the Z direction.
% ZFreq:          Frequency of Z oscillation in Hz.
% ZPhase:         Ordered phase of Z oscillation in degrees.
% ZSpd:     Ordered speed of travel in the Z direction in mm/sec.
% ZSupSpd:        Superimposed speed in the Z direction in mm/sec.
% ZSupDir:        Direction of superimposed motion.

% Initialize certain variables.
NumPoints = 2048;
Results = zeros(3,24);  % Will hold all of the Peaks of the FFT.
Summary = zeros(3,24);  % Will hold the FFT peak closest to the driving
            % frequency.
[Data] = 0;
[Datain]=0;

% Get the input data and find the length of the file.
infname = strcat(fname,'.xls');
existencecheck = exist(infname); % Determines if the input file exists.
if existencecheck ==0
  fprintf('%s does not exist.\n',infname)
  return
end
Datain = xlsread(infname); % Reads the input file.
NumLines = size(Datain,1)-4; % Gets the number of data samples in the
              % input file.

% Get the date from the input file and convert from EXCEL to
%       MATLAB format.
Date=datestr(Datain(1,15)-36525,2);
% Build a date-time string to ensure the correct gains are applied.
% Add 693960 to the date to get number of days from 0000.
% Divide the number of minutes by 86400 to get fractional days.
% MATLAB date serial numbers are in the form:
%    days since 0000.fraction of a day
DateNum=Datain(1,15)+693960+Datain(1,14)/86400;

% Record input motion parameters.
XAmp = Datain(NumLines+2,1);
if isnan(XAmp)==0
  XFreq = Datain(NumLines+2,2);
  XPhase = Datain(NumLines+2,3);
  XSupSpd = Datain(NumLines+2,4);
  XSupDir = Datain(NumLines+2,5);
else
  XAmp=0;
  XFreq = 0;
  XPhase = 0;
  XSupSpd = 0;
  XSupDir = 0;
end

YAmp = Datain(NumLines+2,6);
```

```
if isnan(YAmp)==0
   YFreq = Datain(NumLines+2,7);
   YPhase = Datain(NumLines+2,8);
   YSupSpd = Datain(NumLines+2,9);
   YSupDir = Datain(NumLines+2,10);
else
   YAmp=0;
   YFreq = 0;
   YPhase = 0;
   YSupSpd = 0;
   YSupDir = 0;
end

ZAmp = Datain(NumLines+2,11);
if isnan(ZAmp)==0
   ZFreq = Datain(NumLines+2,12);
   ZPhase = Datain(NumLines+2,13);
   ZSupSpd = Datain(NumLines+2,14);
   ZSupDir = Datain(NumLines+2,15);
else
   ZAmp=0;
   ZFreq = 0;
   ZPhase = 0;
   ZSupSpd = 0;
   ZSupDir = 0;
end

YawAmp = Datain(NumLines+2,16);
if isnan(YawAmp)==0
   YawFreq = Datain(NumLines+3,1);
   YawPhase = Datain(NumLines+3,2);
   YawSupSpd = Datain(NumLines+3,3);
   YawSupDir = Datain(NumLines+3,4);
else
   YawAmp=0;
   YawFreq = 0;
   YawPhase = 0;
   YawSupSpd = 0;
   YawSupDir = 0;
end

PitchAmp = Datain(NumLines+3,5);
if isnan(PitchAmp)==0
   PitchFreq = Datain(NumLines+3,6);
   PitchPhase = Datain(NumLines+3,7);
   PitchSupSpd = Datain(NumLines+3,8);
   PitchSupDir = Datain(NumLines+3,9);
else
   PitchAmp=0;
   PitchFreq = 0;
   PitchPhase = 0;
   PitchSupSpd = 0;
   PitchSupDir = 0;
end

XSpd = Datain(NumLines+3,10);
```

```
if isnan(XSpd)==1
   XSpd = 0;
   if ((isnan(XSupSpd)==0)&(XSupDir==1))
      XSpd = XSupSpd;
   end
end

XDist = Datain(NumLines+3,11);
if isnan(XDist)==1
   XDist = 0;
end

YSpd = Datain(NumLines+3,12);
YDist = Datain(NumLines+3,13);
ZSpd = Datain(NumLines+3,14);
ZDist = Datain(NumLines+3,15);

YawSpd = Datain(NumLines+3,16);
YawDist = Datain(NumLines+4,1);
PitchSpd = Datain(NumLines+4,2);
PitchDist = Datain(NumLines+4,3);

Time = Datain(NumLines+4,5);

% Determine actual sample frequency based upon the measured interval
% between data points. Necessary because the control software did not
% always sample at the ordered sample rate.
SampleFreq = 1/mean(Datain(1:NumLines,16));

% Determine the oscillation frequency.
f1=0;
f2=0;
f3=0;

if (XFreq ~= 0)&(isnan(XFreq)==0)
   f1 = XFreq;
   f2 = 2* XFreq;
   f3 = 3* XFreq;
end

if ((YFreq ~= 0)&(isnan(YFreq)==0))
   f1 = YFreq;
   f2 = 2* YFreq;
   f3 = 3* YFreq;
end

if (ZFreq ~= 0)&(isnan(ZFreq)==0)
   f1 = ZFreq;
   f2 = 2* ZFreq;
   f3 = 3* ZFreq;
end

if (PitchFreq ~= 0)&(isnan(PitchFreq)==0)
   f1 = PitchFreq;
   f2 = 2* PitchFreq;
   f3 = 3* PitchFreq;
```

```
end

if (YawFreq ~= 0)&(isnan(YawFreq)==0)
   f1 = YawFreq;
   f2 = 2* YawFreq;
   f3 = 3* YawFreq;
end

omega=f1*2*pi; % The frequency of oscillation in radians per second.

% Drop the first 1.2 seconds of data to account for acceleration.  At
% the sample frequency of 25 Hz, drop the first 30 points.  Shift the
% other points up in the array.

dropgap = round(1.2*SampleFreq);
for I = 1:NumLines-dropgap;
   Datain(I,1:16)=Datain(I+dropgap,1:16);
end
NumLines = NumLines - dropgap;

% Drop data recorded after the model stopped moving along the X-axis
%   for non-Inertial tests.
if(XSpd~=0)
   NumLines2=NumLines;
   for I = NumLines-1:-1:10
      if (Datain(I,9)==Datain(I+1,9)) % Looks for constant X position.
         NumLines2 = I;
      end
   end
   NumLines = NumLines2;
end

% Make the time of good data be an integer number of wavelengths.
Period = 1/f1;
Duration = Datain(NumLines,14)-Datain(1,14);
NumPeriods = Duration/Period;
NumLines = round(SampleFreq*Period*floor(NumPeriods));

% PREPROCESSING STAGE
% Preprocess the input file for analysis.  This includes:
%  1)  Accounting for the phase shift caused by dropping the first 1.2
%       seconds.
%  2)  Determining the mean force for each column and removing that mean
%       from the column data to get dynamic response,
%  3)  Converting force data from voltages to forces and moments,
%  4)  Converting position data to centimeters for x and y, millimeters
%       for z, and degrees for the angles,
%  5)  Converting the time past midnight to time of data run,
%  6)  Finding the Mean DeltaT,
%  7)  Interpolating the data and time to produce even time intervals
%        and the associated data.

% 1,2) Account for the phase shift and determine the mean of the forces
% columns.

for J = 1:6
```

```
  avg(J) = mean(Datain(1:NumLines,J));
  for I = 1:NumLines
    Datain(I,J) = Datain(I,J) - avg(J);
  end
end

% 3,4) Convert force data from voltages to forces and moments.
%     Convert position data to centimeters for x and y, millimeters for
%      z, and degrees for the angles.

TimeStart = Datain(1,14); % Used for converting from time past midnight
                 % to time of run.

% The inverse sensitivity matrix (B)
if ((datenum(Date)>=datenum(2002,07,17))&(datenum(Date)<datenum(2002,10,31)))
  [B] = [ 0.3901  0.0029 -0.0071 -0.0010 -0.0024 -0.0016;
    0.0023  0.3887  0.0016  0.0025 -0.0039  0.0019;
    0.0139  0.0129  1.5006 -0.0002 -0.0215 -0.0018;
    0.0000 -0.0001  0.0016  0.0069  0.0000 -0.0001;
    -0.0001  0.0000  0.0018  0.0000  0.0070  0.0000;
    -0.0003 -0.0002  0.0000  0.0000  0.0000  0.0108];
else %valid on and after Oct 31, 2002
  [B] = [ 0.3856  0.0020 -0.0016 -0.0008  0.0001 -0.0030;
    0.0023  0.3811 -0.0040  0.0012 -0.0034  0.0031;
    0.0093  0.0024  1.5109  0.0068 -0.0231 -0.0014;
    0.0001  0.0000  0.0013  0.0069  0.0000  0.0000;
    0.0000  0.0000  0.0007  0.0000  0.0069 -0.0001;
    -0.0003  0.0001  0.0000  0.0000  0.0000  0.0106];
end

% Establish Gains and Excitation Voltage
if (DateNum>731434.61458) % August 6, 2002 1445.
  [Gain] = [4000;4000;1000;1000;1000;4000]; % y z x pitch yaw roll
  [Vexc] = [10; 10; 2.5; 10; 10; 10];
end

% Calculate the Conversion Factors (CF)

CFtemp = Gain.*Vexc*10^-6;
CF = zeros(6,6);
CF(1,1)= CFtemp(1);
CF(2,2)= CFtemp(2);
CF(3,3)= CFtemp(3);
CF(4,4)= CFtemp(4);
CF(5,5)= CFtemp(5);
CF(6,6)= CFtemp(6);

% Establish multipliers
% yfactor has a negative multipler to convert from the gantry y positive
% to the load cell y positive direction.
if ((DateNum>731513.33333)&(DateNum<731551.33333))
   % Between Oct 24, 0800 and Dec 1, 0800.
  xfactor = 3850;
  yfactor = -2410;
  zfactor = 2114;
  pitchfactor = 972;
```

```
    yawfactor = 1818;
elseif DateNum>731551.33333 % Dec 1, 0800
    xfactor = 3850;
    yfactor = -2410;
    zfactor = 2114;
    pitchfactor = 155;
    yawfactor = 1818;
else
    xfactor = 3850;
    yfactor = -2410;
    zfactor = 4921;
    pitchfactor = 1111;
    yawfactor = 1025;
end

% Convert voltages to forces and position data to the metric system.
for I = 1 : NumLines
    Datain(I,1:6) = (CF^-1*B*Datain(I,1:6)')';
    Datain(I,9)  = Datain(I,9)  / xfactor;
    Datain(I,10) = Datain(I,10) / yfactor;
    Datain(I,11) = Datain(I,11) / zfactor;
    Datain(I,12) = Datain(I,12) / yawfactor;
    Datain(I,13) = Datain(I,13) / pitchfactor;
    Datain(I,14) = Datain(I,14) - TimeStart;
end

% Convert the position data to position of midships vice position of
% the strut.
distance = .1093; % The distance from the strut to midships.
PitchRadius = .2576; % Length of pitch arm
for I = 1: NumLines
    % Effect of yaw on y position.
    Datain(I,10) = Datain(I,10)-distance*sin(Datain(I,12)*pi/180);
    % Effect of pitch on x position.
    Datain(I,9) = Datain(I,9) + PitchRadius*sin(Datain(I,11)*pi/180);
    % Effect of pitch on z position.
    Datain(I,11) =Datain(I,11)-PitchRadius*(1-cos(Datain(I,11)*pi/180));
end

% Account for the transducer being forward of midships.
% Reference: Marine Hydrodynamics by J.N. Newman, Sect 6.2
% waveheight = Acos(kx-wt+arbitrary phase shift) (eq 7 page 240)
% Let x = 0 at the midships location.
% The wave is travelling in the positive x direction.
% The transducer is at a negative x direction from midships.
% Assume the wave is travelling in the positive x-direction.
% Use midships as the zero reference location.

% Shift the origin of the coordinate system from the origin of the load
% cell to vessel amidships.
shiftlength = 0.0522;

wavenum = omega^2/9.81;
for I =1:NumLines
    Datain(I,4) = Datain(I,4) - Datain(I,2) * shiftlength; % pitch
    Datain(I,5) = Datain(I,5) + Datain(I,1) * shiftlength; % yaw
```

```
end

% 6) Determine the DeltaT from the input file.
DeltaT = 1/SampleFreq;

% 7) Interpolate data points to produce even time intervals and
% associated data.  By this process the input datain matrix will be
% interpolated into the data matrix.
% Start by building even time intervals and the date column.
for I = 1:NumPoints;
   % Interpolate as if the sample freq was 25 hz.
   Data(I,14) = (I-1)* .04;
   Data(I,15) = Datain(1,15);
end

% Interpolate each column in turn.
for J = 1:13
   Data(1,J) = Datain(1,J);
   for I = 2:NumLines;
      for K = 1:NumLines;
         if (Datain(K,14)>Data(I,14))
            break
         end
      end
      KS  = K - 1;
      if (Data(I,14)<=Datain(NumLines,14))
         Data(I,J) = Datain(KS,J) + (Data(I,14)-Datain(KS,14))*...
            (Datain(K,J)-Datain(KS,J))/(Datain(K,14)-Datain(KS,14));
      else
         Data(I,J) = 0.0;
      end
   end
end


% Identify the driving force.
if YFreq ~= 0 & isnan(YFreq)==0
   DrivingForce = 10;
   omega = YFreq*2*pi;
elseif ZFreq~= 0 & isnan(ZFreq)==0
   DrivingForce = 11;
   omega = ZFreq*2*pi;
elseif YawFreq~= 0 & isnan(YawFreq)==0
   DrivingForce = 12;
   omega = YawFreq*2*pi;
elseif PitchFreq~=0 & isnan(PitchFreq)==0
   DrivingForce = 13 ;
   omega = PitchFreq*2*pi;
elseif XFreq ~= 0 & isnan(XFreq)==0
   DrivingForce = 9;
   omega = XFreq*2*pi;
else
   DrivingForce=9;
   omega=0;
end
```

```
% PROCESSING STAGE
for J = 10:13
  Data(1:NumLines,J)=Data(1:NumLines,J)-mean(Data(1:NumLines,J));
end


% Round all data up to 2048 points. Pad the excess with zero's.
Data(NumLines+1:NumPoints,1:13) = 0.0;

% Convert the time column to a frequency column.
Interval = .04 * NumPoints;

for I = 1: NumPoints
  Data(I,14)=(I-1)/Interval;
end

% For each column, perform Fast Fourier Transformation, low pass
% filtering, and identify the frequencies and amplitudes of the
% forces.

% Perform the Fast Fourier Transformation.
for J = 1: 13
  Data(:,J) = fft(Data(:,J));
  % Transform the FFT Coefficient to account for padding.
  for I = 1:NumPoints
    Data(I,J) = Data(I,J)* NumPoints/NumLines;
  end
end


% Change the phase of all data such that the phase is relative to
% the driving motion.


for I = 1:NumLines
  for J = 1:13
    Data(I,J) = Data(I,J)*exp(-i*angle(Data(I,DrivingForce)));
  end
end

% Account for ELECTRICAL FILTERS.
[R] = [145500 144100 147400 149200 191600 146600];
[C] = [.073 .114 .081 .099 .089 .075 ]*10^-6;

for J= 1:6
  for I = 1:NumLines
    Data(I,J) = Data(I,J)*exp(-i*atan(omega*C(J)*R(J))) ;
  end
end


% Determine the actual phase of the motion at the oscillation frequency
% Find the index of the frequency of oscillation.
look = .5*1/Interval;
PhaseIndexf1 = find(((f1-look)<=Data(:,14))&(Data(:,14)<(f1+look)));
PhaseIndexf2 = find(((f2-look)<=Data(:,14))&(Data(:,14)<(f2+look)));
```

```
PhaseIndexf3 = find(((f3-look)<=Data(:,14))&(Data(:,14)<(f3+look)));
for J=9:13
   ActualFreq(J)=Data(PhaseIndexf1,14);
   ActualPhasef1(J)= angle(Data(PhaseIndexf1,J))*180/pi;
   ActualPhasef2(J)= angle(Data(PhaseIndexf2,J))*180/pi;
   ActualPhasef3(J)= angle(Data(PhaseIndexf3,J))*180/pi;
  ActualMotion(J)=sqrt(2.*2.*((abs(Data(PhaseIndexf1,J))/NumPoints)^2));
end

omega = ActualFreq(DrivingForce)*2*pi;


% Find forces at the frequencies of oscillation.
for J = 1:6
Summaryf1(3*J-2)=Data(PhaseIndexf1,14); % freq
Summaryf1(3*J-1)=sqrt(2.*2.*((abs(Data(PhaseIndexf1,J))/NumPoints)^2)); % ampl
Summaryf1(3*J)=angle(Data(PhaseIndexf1,J))*180/pi-ActualPhasef1(DrivingForce); %phase

Summaryf2(3*J-2)=Data(PhaseIndexf2,14); % freq
Summaryf2(3*J-1)=sqrt(2.*2.*((abs(Data(PhaseIndexf2,J))/NumPoints)^2)); % ampl
Summaryf2(3*J)=angle(Data(PhaseIndexf2,J))*180/pi-ActualPhasef2(DrivingForce); %phase

Summaryf3(3*J-2)=Data(PhaseIndexf3,14); % freq
Summaryf3(3*J-1)=sqrt(2.*2.*((abs(Data(PhaseIndexf3,J))/NumPoints)^2)); % ampl
Summaryf3(3*J)=angle(Data(PhaseIndexf3,J))*180/pi-ActualPhasef3(DrivingForce); %phase
end

% Calculate the inertial forces in mks units for all series except
%   mass matrix evaluation (7).
mass= 3.67;
Xg=0.03;
Gyradius = 0.0764;
YawRadius = 0.07;
if series ~= 7 % Calculates inertial force for all except mass matrix
   % (inertial) tests.
   Iyy = 0.092509;
   Izz = 0.1006;
   XAmp = ActualMotion(9);
   YAmp = ActualMotion(10);
   ZAmp = ActualMotion(11);
   PitchAmp = ActualMotion(13);
   YawAmp = ActualMotion(12);
   Fx = -omega^2*mass*(XAmp/100 + PitchRadius*(PitchAmp*pi/180));
   Fy = -omega^2*(mass*(YAmp/100 + (YawAmp*pi/180)*YawRadius) + mass*Xg*(YawAmp*pi/180));
   Fz = -omega^2*(mass*(ZAmp/1000 + (PitchAmp*pi/180) * (YawRadius))-
mass*Xg*(PitchAmp*pi/180));
   Fpitch = -omega^2*(-mass*Xg*(ZAmp/1000 + (PitchAmp*pi/180) * (YawRadius-Xg)) + Iyy *
(PitchAmp*pi/180));
   Fyaw = -omega^2 * (mass*Xg*(YAmp/100 - (YawAmp*pi/180)*YawRadius)+Izz*(YawAmp*pi/180));
else
   Iyy = 0;
   Izz=0;
   Fx=0;
   Fy=0;
   Fz=0;
   Fpitch = 0;
```

```
    Fyaw=0;
end

% Calculate the hydrodynamic response at the oscillation frequency and
%   its harmonics.
DrivingFreq = omega/(2*pi);

for J = 1:6
    switch J
    case 1
        InertialForce = Fy;
    case 2
        InertialForce = Fz;
    case 3
        InertialForce = Fx;
    case 4
        InertialForce = Fpitch;
    case 5
        InertialForce = Fyaw;
    case 6
        InertialForce = 0;
    end

    Results(1,4*J-3) = Summaryf1(3*J-2);  % oscillation freq
    Results(1,4*J-2) = Summaryf1(3*J-1); % total force
    Results(1,4*J-1) = imag(Summaryf1(3*J-1))+real(Summaryf1(3*J-1)) + InertialForce; % force without
inertial component
    Results(1,4*J) = Summaryf1(3*J); % phase


    Results(2,4*J-3) = Summaryf2(3*J-2);  % first harmonic of oscillation freq
    Results(2,4*J-2) = Summaryf2(3*J-1); % total force
    Results(2,4*J-1) = Summaryf2(3*J-1);%
    Results(2,4*J) = Summaryf2(3*J); % phase


    Results(3,4*J-3) = Summaryf3(3*J-2);  % second harmonic of oscillation freq
    Results(3,4*J-2) = Summaryf3(3*J-1); % total force
    Results(3,4*J-1) = Summaryf3(3*J-1); % force without inertial component
    Results(3,4*J) = Summaryf3(3*J); % phase

end

% Condition the Results array.
% Ensure the phase angles are +- 180 degrees.
% Ensure the amplitudes are all positive.
for I = 1:3
    for J = 1:6
        if Results(I,4*J-1) <0
            Results(I,4*J-1) = abs(Results(I,4*J-1));
            Results(I,4*J) = Results(I,4*J) + 180;
        end
        if Results(I,4*J) > 180
            Results(I,4*J) = Results(I,4*J) - 360;
        elseif Results(I,4*J) <-180
            Results(I,4*J) = Results(I,4*J) + 360;
```

```
    end
  end
end

% Print to a common row output file.
fprintf(manyrowsfid,'%s \t %s\t %6.2f',fname,Date,Depth);
fprintf(manyrowsfid,'\t%6.4f \t%5.1f \t %5.2f\t%5.1f',Submergence,XSpd,XDist,Duration);
fprintf(manyrowsfid,' \t %2.0f \t%2.0f\t %3.1f\t %3.1f',PitchAngle,YawAngle,SternPlanes,Rudder);
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %4.1f \t %4.1f \t%8.5f \t %7.4f \t %7.4f \t
%4.1f',ActualFreq(9),ActualMotion(9),XPhase,ActualPhasefl(9),Results(1,9:12));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 9:12));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(3, 9:12));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %4.1f \t %4.1f \t %8.5f \t %7.4f \t %7.4f \t
%4.1f',ActualFreq(10),ActualMotion(10),YPhase,ActualPhasefl(10),Results(1,1:4));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 1:4));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(3, 1:4));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %4.1f \t %4.1f \t %8.5f \t %7.4f \t %7.4f \t
%4.1f',ActualFreq(11),ActualMotion(11),ZPhase,ActualPhasefl(11),Results(1,5:8));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 5:8));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(3, 5:8));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(1,21:24));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 21:24));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(3, 21:24));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %4.1f \t %4.1f \t %8.5f \t %7.4f \t %7.4f \t
%4.1f',ActualFreq(13),ActualMotion(13),PitchPhase,ActualPhasefl(13),Results(1,13:16));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 13:16));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(3, 13:16));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %4.1f \t %4.1f \t %8.5f \t %7.4f \t %7.4f \t
%4.1f',ActualFreq(12),ActualMotion(12),YawPhase,ActualPhasefl(12),Results(1,17:20));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f',Results(2, 17:20));
fprintf(manyrowsfid,' \t %8.5f \t %7.4f \t %7.4f \t %4.1f \n',Results(3, 17:20));
```

Appendix H. Results of Inertial Force Calculation Checks

Experiment Parameters						Y Motion				Y Force			
Test #	Date	Depth	Subm.	X Spd	Travel	Freq	Ampl	Ph	Act Ph	Freq	Ampl	Ampl-I	Ph
M25	03/03/03	0	0	0	0	0.40283	2.0694	0	0	0.40283	0.4287	0.0578	176.3
M26	03/03/03	0	0	66.7	700	0.40283	2.0624	0	0	0.40283	0.4485	0.0364	178.8
M27	03/03/03	0	0	0	0	0.40283	4.1458	0	0	0.40283	1.0518	0.077	-5.5
M28	03/03/03	0	0	66.7	700	0.40283	4.1399	0	0	0.40283	1.0795	0.1062	-6.2
M29	03/03/03	0	0	0	0	0.40283	8.2823	0	0	0.40283	2.0185	0.0712	-4.2
M30	03/03/03	0	0	66.7	700	0.40283	8.2847	0	0	0.40283	2.0635	0.1157	-3.9

Experiment Parameters						Yaw Motion				Yaw Moment			
Test #	Date	Depth	Subm.	X Spd	Travel	Freq	Ampl	Ph	Act Ph	Freq	Ampl	Ampl-I	Ph
M31	03/03/03	0	0	0	0	0.40283	9.9801	0	0	0.40283	0.1049	0.0012	-16.7
M32	03/03/03	0	0	66.7	700	0.40283	9.9761	0	0	0.40283	0.1172	0.0135	-1
M33	03/03/03	0	0	0	0	0.40283	15.0002	0	0	0.40283	0.1717	0.0159	-3.9
M34	03/03/03	0	0	66.7	700	0.40283	14.562	0	0	0.40283	0.1458	0.0055	-161.5
M35	03/03/03	0	0	0	0	0.40283	19.7336	0	0	0.40283	0.2228	0.0178	-6
M36	03/03/03	0	0	66.7	700	0.40283	19.9864	0	0	0.40283	0.2263	0.0187	-8.5

Experiment Parameters						Z Motion				Z Force			
Test #	Date	Depth	Subm.	X Spd	Travel	Freq	Ampl	Ph	Act Ph	Freq	Ampl	Ampl-I	Ph
M37	03/03/03	0	0	0	0	0.40283	20.5998	0	0	0.40283	0.503	0.0187	-23.9
M38	03/03/03	0	0	66.7	700	0.40283	20.5418	0	0	0.40283	0.6571	0.1741	-55.6
M39	03/03/03	0	0	0	0	0.40283	41.204	0	0	0.40283	0.9937	0.025	-9.3
M40	03/03/03	0	0	66.7	700	0.40283	41.0487	0	0	0.40283	1.2129	0.2478	-22.8
M41	03/03/03	0	0	0	0	0.40283	82.4881	0	0	0.40283	1.943	0.0036	-14.8
M42	03/03/03	0	0	66.7	700	0.40283	82.2366	0	0	0.40283	1.7104	0.2231	178.3

Experiment Parameters						Pitch Motion				Pitch Moment			
Test #	Date	Depth	Subm.	X Spd	Travel	Freq	Ampl	Ph	Act Ph	Freq	Ampl	Ampl-I	Ph
M43	03/03/03	0	0	0	0	0.40283	9.0085	0	0	0.40283	0.0789	0.0099	172.3
M44	03/03/03	0	0	66.7	0	0.40283	9.0958	0	0	0.40283	0.09	0.0004	-30.2
M45	03/03/03	0	0	0	0	0.40283	13.9231	0	0	0.40283	0.1701	0.0329	2.4
M46	03/03/03	0	0	66.7	0	0.40283	13.9589	0	0	0.40283	0.1607	0.0232	-6.1
M47	03/03/03	0	0	0	0	0.40283	18.7394	0	0	0.40283	0.2065	0.0219	-7.8
M48	03/03/03	0	0	66.7	0	0.40283	18.8035	0	0	0.40283	0.2121	0.0268	-6

Appendix I. Selected Portion of the Output File from AnalyzemodXls.m for the Analysis of the Oscillation Test Series

The first two rows have been added by the author for clarity.

Experiment Parameters											Y Motion					Y Force				2nd Y Force			
Test #	Date	Depth	Subm.	X Spd	Travel	Time	Pitch	Yaw	St Pl	Rudder	Freq	Ampl	Ph	Act Ph	Freq	Ampl	Ampl-I	Ph	Freq	Ampl	Ampl-I	Ph
OSC07	04/14/03	0.8	0.543	33.3	700	20.5	0	0	0	0	0.40283	10.031	0	0	0.40283	4.5753	2.2169	-32.6	0.80566	0.2087	0.2087	100.4
OSC15	04/14/03	0.8	0.252	100	700	7.8	0	0	0	0	0.40283	10.0202	0	0	0.40283	4.9963	2.6405	-39.9	0.80566	0.3432	0.3432	-9
OSC17	04/14/03	0.8	0.398	66.7	700	10.7	0	0	0	0	0.79346	10.1862	0	0	0.79346	17.9352	8.6436	-35.7	1.58691	0.2721	0.2721	115
OSC24	04/14/03	0.8	0.543	100	700	7.8	0	0	0	0	0.40283	10.0214	0	0	0.40283	5.033	2.6769	-38.1	0.80566	0.2017	0.2017	-159.4
OSC25	04/14/03	0.8	0.252	33.3	700	20.5	0	0	0	0	0.40283	10.0279	0	0	0.40283	4.5736	2.2159	-32.3	0.80566	0.3405	0.3405	-12
OSC32	04/14/03	0.8	0.488	33.3	700	20.6	0	0	0	0	0.40283	0	0	17.6	0.40283	0.0809	0.0809	80.9	0.80566	0.0437	0.0437	-148.5
OSC40	04/14/03	0.8	0.272	100	700	7.8	0	0	0	0	0.40283	0	0	88.1	0.40283	0.2653	0.2653	82.7	0.80566	0.1509	0.1509	149.7
OSC42	04/14/03	0.8	0.398	66.7	700	10.7	0	0	0	0	0.79346	0	0	-18.9	0.79346	0.2085	0.2085	24.1	1.58691	0.1883	0.1883	102.8
OSC49	04/14/03	0.8	0.488	100	700	7.8	0	0	0	0	0.40283	0	0	-49.1	0.40283	0.2812	0.2812	46	0.80566	0.1458	0.1458	4.8
OSC50	04/14/03	0.8	0.272	33.3	700	20.5	0	0	0	0	0.40283	0	0	-154.3	0.40283	0.1127	0.1127	149.6	0.80566	0.1874	0.1874	178
OSC54	04/14/03	0.8	0.252	33.3	700	20.6	0	0	0	0	1.19629	0.0023	0	-179.6	1.19629	5.4432	1.8412	124.4	2.39258	0.1766	0.1766	128.9
OSC55	04/14/03	0.8	0.252	100	700	7.8	0	0	0	0	1.19629	0.0023	0	-179.4	1.19629	6.7597	3.1342	97.2	2.39258	0.2889	0.2889	11.1
OSC57	04/14/03	0.8	0.543	33.3	700	20.6	0	0	0	0	0.40283	0.0023	0	-179.3	0.40283	0.8291	0.4198	101.7	0.80566	0.0524	0.0524	35.9
OSC58	04/14/03	0.8	0.543	33.3	700	20.6	0	0	0	0	1.19629	0.0022	0	-179.1	1.19629	5.3553	1.76	124.6	2.39258	0.0957	0.0957	105.4
OSC65	04/14/03	0.8	0.252	100	700	7.8	0	0	0	0	0.40283	0.0023	0	180	0.40283	2.1715	1.7609	57.1	0.80566	0.4104	0.4104	-82.5
OSC67	04/14/03	0.8	0.398	66.7	700	10.7	0	0	0	0	0.79346	0.0023	0	180	0.79346	3.1234	1.5352	101.4	1.58691	0.0712	0.0712	0.5
OSC71	04/14/03	0.8	0.543	100	700	7.8	0	0	0	0	1.19629	0.0023	0	-179.5	1.19629	7.1373	3.5312	97.9	2.39258	0.327	0.327	126.7
OSC79	04/29/03	0.8	0.252	33.3	0	20.8	0	0	0	0	1.19629	0	0	-177.1	1.19629	0.3187	0.3186	-27.4	2.39258	0.4632	0.4632	-160.5
OSC82	04/29/03	0.8	0.543	33.3	0	18.8	0	0	0	0	0.40283	0	0	-16.8	0.40283	0.1159	0.1158	40.7	0.80566	0.1177	0.1177	-158.5
OSC83	04/29/03	0.8	0.543	33.3	0	19.8	0	0	0	0	1.19629	0	0	94.1	1.19629	0.2508	0.2507	-29.9	2.39258	0.3689	0.3689	-143.7
OSC92	04/29/03	0.8	0.398	66.7	0	8.8	0	0	0	0	0.79346	0	0	26.7	0.79346	0.1875	0.1875	35.9	1.58691	0.1958	0.1958	-56.5
OSC100	04/29/03	0.8	0.252	33.3	0	19.8	0	0	0	0	0.40283	0	0	50.9	0.40283	0.1533	0.1533	44.6	0.80566	0.0767	0.0767	-174.6

Units for Interpreting the Results of Model Scale Experiments Performed at MIT to Determine the Restoring Forces and Moments due to Body Angle

Quantity	Units
Depth	cm
Submergence	mm
Speed	cm/s
Phase	Degrees
Forces	Newtons
Moments	N-m

Appendix J. CoeffSolver.m

```
%CoeffSolver.m

% Erik Oller
% Spring 2003

% Solves for coefficients using linear regression.
% Added mass and damping coefficients are of the form
% Coeff = a0 + a1*L/subm + a3*Fr
% Restoring forces equations are of the form
% Coefficient=a0 + a1*L/subm + a2*L/subm^2 +a3*Fr
clear all;
close all;

% Solve for Restoring Force Coefficients
% Solve for Yaw Induced Restoring Force (Yuv)
YuvMatrix = [-5.368865729
-4.853454974
-2.329293369
-3.514754719
-2.848702047
-1.614922323];

SteadyYawMatrix=[1      2.711810649     7.353916996     0.289533426
1       1.720624859     2.960549904     0.289533426
1       1.260063003     1.587758772     0.289533426
1       2.711810649     7.353916996     0.386044568
1       1.720624859     2.960549904     0.386044568
1       1.260063003     1.587758772     0.386044568];

YuvCoeffMatrix=SteadyYawMatrix\YuvMatrix;
PredYuvMatrix=SteadyYawMatrix*YuvCoeffMatrix;
for i = 1:size(PredYuvMatrix,1)
   YuvDiff(i) = abs((PredYuvMatrix(i)-YuvMatrix(i)))/YuvMatrix(i);
end
Yuvrms=norm(YuvDiff)/sqrt(size(YuvMatrix,1));


% Solve for Pitch Induced Restoring Force (Zuw), C53
ZuwMatrix = [1.023955274
2.269136274
1.141877357
1.220249361
2.612356757
1.801033337];

SteadyPitchMatrix=[1    2.711810649     7.353916996     0.289533426
1       1.720624859     2.960549904     0.289533426
1       1.260063003     1.587758772     0.289533426
1       2.714285714     7.367346939     0.386044568
1       1.718592965     2.953561779     0.386044568
1       1.259668508     1.586764751     0.386044568];

ZuwCoeffMatrix=SteadyPitchMatrix\ZuwMatrix;
```

```
PredZuwMatrix=SteadyPitchMatrix*ZuwCoeffMatrix;
for i = 1:size(PredZuwMatrix,1)
  ZuwDiff(i) = abs((PredZuwMatrix(i)-ZuwMatrix(i)))/ZuwMatrix(i);
end
Zuwrms=norm(ZuwDiff)/sqrt(size(ZuwMatrix,1));


% Solve for Yaw Induced Restoring Moment (Nuv)
NuvMatrix = [-1.020609587
-0.996711818
-0.682640735
-0.800283816
-0.737734321
-0.586442502];

SteadyYawMomMatrix=[1	2.711810649	7.353916996	0.289533426
1	1.720624859	2.960549904	0.289533426
1	1.260063003	1.587758772	0.289533426
1	2.711810649	7.353916996	0.386044568
1	1.720624859	2.960549904	0.386044568
1	1.260063003	1.587758772	0.386044568];

NuvCoeffMatrix=SteadyYawMomMatrix\NuvMatrix;
PredNuvMatrix=SteadyYawMomMatrix*NuvCoeffMatrix;
for i = 1:size(PredNuvMatrix,1)
  NuvDiff(i) = abs((PredNuvMatrix(i)-NuvMatrix(i)))/NuvMatrix(i);
end
Nuvrms=norm(NuvDiff)/sqrt(size(NuvMatrix,1));


% Solve for Pitch Induced Restoring Moment (Muw)
MuwMatrix = [-0.626004966
-0.594465093
-0.71149292
-0.401198364
-0.619156216
-0.599156406];

SteadyPitchMomMatrix=[1	2.711810649	7.353916996	0.289533426
1	1.720624859	2.960549904	0.289533426
1	1.260063003	1.587758772	0.289533426
1	2.714285714	7.367346939	0.386044568
1	1.718592965	2.953561779	0.386044568
1	1.259668508	1.586764751	0.386044568];

MuwCoeffMatrix=SteadyPitchMomMatrix\MuwMatrix;
PredMuwMatrix=SteadyPitchMomMatrix*MuwCoeffMatrix;
for i = 1:size(PredMuwMatrix,1)
  MuwDiff(i) = abs((PredMuwMatrix(i)-MuwMatrix(i)))/MuwMatrix(i);
end
Muwrms=norm(MuwDiff)/sqrt(size(MuwMatrix,1));

% Solve for Sway induced added masses

% Solve for Yvdot
```

```
YvdotMatrix=[-0.017470
-0.018969
-0.016666
-0.019724
-0.017526];

SwayMatrix=[1   1.277450258       0.127678512
1          2.749405234       0.383418954
1          1.744402516       0.255740442
1          1.277450258       0.383418954
1          2.749405234       0.127678512];

YvdotCoeffMatrix = SwayMatrix\YvdotMatrix;
PredYvdotMatrix=SwayMatrix*YvdotCoeffMatrix;

sumofsquares=0;

for i = 1:size(PredYvdotMatrix,1)
   YvdotDiff(i) = (PredYvdotMatrix(i)-YvdotMatrix(i))/YvdotMatrix(i);
end
Yvdotrms=norm(YvdotDiff)/sqrt(size(YvdotMatrix,1));

% Solve for Yv
YvMatrix=[-0.058883
-0.027836
-0.062066
-0.027142
-0.058392];

YvCoeffMatrix = SwayMatrix\YvMatrix;
PredYvMatrix=SwayMatrix*YvCoeffMatrix;
sumofsquares=0;
for i = 1:size(PredYvMatrix,1)
   YvDiff(i) = abs((PredYvMatrix(i)-YvMatrix(i)))/YvMatrix(i);
end
Yvrms=norm(YvDiff)/sqrt(size(YvMatrix,1));


% Solve for Nvdot
NvdotMatrix=[-0.000809
-0.000692
-0.000677
-0.000862
-0.000833];
NvdotCoeffMatrix = SwayMatrix\NvdotMatrix;
PredNvdotMatrix=SwayMatrix*NvdotCoeffMatrix;

for i = 1:size(PredNvdotMatrix,1)
   NvdotDiff(i)=abs(PredNvdotMatrix(i)-NvdotMatrix(i))/NvdotMatrix(i);
end
Nvdotrms=norm(NvdotDiff)/sqrt(size(NvdotMatrix,1));

% Solve for Nv

NvMatrix=[-0.004323
-0.008228
```

```
-0.005161
-0.008666
-0.004136];

NvCoeffMatrix = SwayMatrix\NvMatrix;
PredNvMatrix=SwayMatrix*NvCoeffMatrix;

for i = 1:size(PredNvMatrix,1)
   NvDiff(i) = (PredNvMatrix(i)-NvMatrix(i))/NvMatrix(i);

end
Nvrms=norm(NvDiff)/sqrt(size(NvMatrix,1));

% Solve for coefficients in heave
%  Solve for Zwdot

ZwdotMatrix=[-0.019162
-0.022988
-0.016744
-0.020764
-0.021686];

HeaveMatrix=[1  1.421     0.128
1         2.547    0.383
1         1.744    0.256
1         1.421    0.383
1         2.547    0.128];

ZwdotCoeffMatrix = HeaveMatrix\ZwdotMatrix;
PredZwdotMatrix=HeaveMatrix*ZwdotCoeffMatrix;

for i = 1:size(PredZwdotMatrix,1)
  ZwdotDiff(i)=abs((PredZwdotMatrix(i)-ZwdotMatrix(i)))/ZwdotMatrix(i);
end
Zwdotrms=norm(ZwdotDiff)/sqrt(size(ZwdotMatrix,1));

% Solve for Zw
ZwMatrix=[-0.062616
-0.029098
-0.065158
-0.024766
-0.063094];

ZwCoeffMatrix = HeaveMatrix\ZwMatrix;
PredZwMatrix=HeaveMatrix*ZwCoeffMatrix;

for i = 1:size(PredZwMatrix,1)
   ZwDiff(i) = abs((PredZwMatrix(i)-ZwMatrix(i)))/ZwMatrix(i);
end
Zwrms=norm(ZwDiff)/sqrt(size(ZwMatrix,1));


% Solve for Mwdot
MwdotMatrix=[0.002835
0.002002
0.002847
```

```
0.000781
0.003218];
MwdotCoeffMatrix = HeaveMatrix\MwdotMatrix;
PredMwdotMatrix=HeaveMatrix*MwdotCoeffMatrix;

for i = 1:size(PredMwdotMatrix,1)
   MwdotDiff(i) = abs((PredMwdotMatrix(i)-MwdotMatrix(i)))/MwdotMatrix(i);
end
Mwdotrms=norm(MwdotDiff)/sqrt(size(MwdotMatrix,1));

% Solve for Mw

MwMatrix=[0.018451
0.008741
0.009078
0.010553
0.012233];

MwCoeffMatrix = HeaveMatrix\MwMatrix;
PredMwMatrix=HeaveMatrix*MwCoeffMatrix;

for i = 1:size(PredMwMatrix,1)
   MwDiff(i) = abs((PredMwMatrix(i)-MwMatrix(i)))/MwMatrix(i);
end
Mwrms=norm(MwDiff)/sqrt(size(MwMatrix,1));


% Solve for coefficients in yaw
%  Solve for Yrdot

YrdotMatrix=[0.001338
0.002209
0.003194
0.002134
0.001086];

YawMatrix=[1    2.749    0.128
1         1.277    0.128
1         1.744    0.256
1         2.749    0.383
1         1.277    0.383];

YrdotCoeffMatrix = YawMatrix\YrdotMatrix;
PredYrdotMatrix=YawMatrix*YrdotCoeffMatrix;

for i = 1:size(PredYrdotMatrix,1)
  YrdotDiff(i)=abs((PredYrdotMatrix(i)-YrdotMatrix(i)))/YrdotMatrix(i);
end
Yrdotrms=norm(YrdotDiff)/sqrt(size(YrdotMatrix,1));


% Solve for Yr
YrMatrix=[0.021030
0.018478
0.015644
0.014240
```

```
0.016104];

YrCoeffMatrix = YawMatrix\YrMatrix;
PredYrMatrix=YawMatrix*YrCoeffMatrix;

for i = 1:size(PredYrMatrix,1)
  YrDiff(i) = abs((PredYrMatrix(i)-YrMatrix(i)))/YrMatrix(i);
end
Yrrms=norm(YrDiff)/sqrt(size(YrMatrix,1));

% Solve for Nrdot
NrdotMatrix=[-0.000871
-0.000844
-0.00113666
-0.001051
-0.000960];
NrdotCoeffMatrix = YawMatrix\NrdotMatrix;
PredNrdotMatrix=YawMatrix*NrdotCoeffMatrix;

for i = 1:size(PredNrdotMatrix,1)
  NrdotDiff(i)=abs((PredNrdotMatrix(i)-NrdotMatrix(i)))/NrdotMatrix(i);
end
Nrdotrms=norm(NrdotDiff)/sqrt(size(NrdotMatrix,1));

% Solve for Nr

NrMatrix=[-0.008514
-0.005804
-0.003476
-0.004136
-0.003916];

NrCoeffMatrix = YawMatrix\NrMatrix;
PredNrMatrix=YawMatrix*NrCoeffMatrix;

for i = 1:size(PredNrMatrix,1)
  NrDiff(i) = abs((PredNrMatrix(i)-NrMatrix(i)))/NrMatrix(i);

end
Nrrms=norm(NrDiff)/sqrt(size(NrMatrix,1));

% Solve for coefficients in Pitch
%  Solve for Zqdot

ZqdotMatrix=[-0.001497196
-0.000740791
0.0036744];

PitchMatrix=[1   2.749405234     0.127678512
1         1.277450258     0.127678512
1         2.749405234     0.383418954];

ZqdotCoeffMatrix = PitchMatrix\ZqdotMatrix;
PredZqdotMatrix=PitchMatrix*ZqdotCoeffMatrix;
for i = 1:size(PredZqdotMatrix,1)
 ZqdotDiff(i)=abs((PredZqdotMatrix(i)-ZqdotMatrix(i)))/ZqdotMatrix(i);
```

```
end
Zqdotrms=norm(ZqdotDiff)/sqrt(size(ZqdotMatrix,1));

% Solve for Zq
ZqMatrix=[-0.043506132
-0.048655759
-0.009306293];


ZqCoeffMatrix = PitchMatrix\ZqMatrix;
PredZqMatrix=PitchMatrix*ZqCoeffMatrix;

for i = 1:size(PredZqMatrix,1)
   ZqDiff(i) = abs((PredZqMatrix(i)-ZqMatrix(i)))/ZqMatrix(i);

end
Zqrms=norm(ZqDiff)/sqrt(size(ZqMatrix,1));

% Solve for Mqdot
MqdotMatrix=[-0.000825618
-0.00098228
-0.003345578];
MqdotCoeffMatrix = PitchMatrix\MqdotMatrix;
PredMqdotMatrix=PitchMatrix*MqdotCoeffMatrix;

for i = 1:size(PredMqdotMatrix,1)
  MqdotDiff(i)=abs((PredMqdotMatrix(i)-MqdotMatrix(i)))/MqdotMatrix(i);
end
Mqdotrms=norm(MqdotDiff)/sqrt(size(MqdotMatrix,1));


% Solve for Mq

MqMatrix=[-0.008513669
-0.007481107
-0.000642713];

MqCoeffMatrix = PitchMatrix\MqMatrix;
PredMqMatrix=PitchMatrix*MqCoeffMatrix;

for i = 1:size(PredMqMatrix,1)
   MqDiff(i) = abs((PredMqMatrix(i)-MqMatrix(i)))/MqMatrix(i);

end
Mqrms=norm(MqDiff)/sqrt(size(MqMatrix,1));

% Build Output File
warning off
delete('CoeffSolver.txt')
warning on
fileid = fopen('CoeffSolver.txt','a');

fprintf(fileid,'%s\t%s\t%s\t%s\t%s\t%s\t\n','Coeff','1','L/Subm','(L/Subm)^2','Fr','rms');
fprintf(fileid,'%s\t%8.6f\t%8.6f\t %8.6f \t%8.6f\t%g \n','Yuv',YuvCoeffMatrix,Yuvrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %8.6f\t%g \n','Zuw',ZuwCoeffMatrix,Zuwrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f \t %8.6f \t%g \n','Nuv',NuvCoeffMatrix,Nuvrms);
```

```
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f \t%8.6f \t %g \n','Muw',MuwCoeffMatrix,Muwrms);
fprintf(fileid,'%s \t%s \t%s \t%s \t%s  \t\n','Coeff','1','L/Subm','Fr','rms');
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Yrdot',YrdotCoeffMatrix,Yrdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Yr',YrCoeffMatrix,Yrrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Yvdot',YvdotCoeffMatrix,Yvdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Yv',YvCoeffMatrix,Yvrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Zqdot',ZqdotCoeffMatrix,Zqdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Zq',ZqCoeffMatrix, Zqrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Zwdot',ZwdotCoeffMatrix,Zwdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Zw',ZwCoeffMatrix, Zwrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Mqdot',MqdotCoeffMatrix,Mqdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Mq',MqCoeffMatrix, Mqrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Mwdot',MwdotCoeffMatrix,Mwdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Mw',MwCoeffMatrix, Mwrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Nrdot',NrdotCoeffMatrix, Nrdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Nr',NrCoeffMatrix, Nrrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Nvdot',NvdotCoeffMatrix, Nvdotrms);
fprintf(fileid,'%s \t %8.6f \t %8.6f\t %8.6f\t  %g \n','Nv',NvCoeffMatrix, Nvrms);
status = fclose(fileid);
```

Appendix K. Output of CoeffSolver.m

This file shows the output of the file CoeffSolver.m. To derive the equation for coefficient A', find the sum of the products of the elements in the row labeled A and the value of the header for the column of the element, excepting the last column. In mathematical form, the equation looks is

$$A' = \sum_{j=1}^{n} (A_j Header_j) \tag{44}$$

where n =4 for restoring force coefficients and n=3 for all other coefficients. The root mean square of the difference between the predicted and measured values are shown in the final column.

Coeff	1	L/Subm	(L/Subm)^2	Fr	rms
Yuv	3.035943	-11.2325	2.39970	15.79519	0.132812
Zuw	-7.723764	9.156976	-2.36537	4.182974	0.0752513
Nuv	0.020017	-1.45297	0.31796	1.987686	0.050452
Muw	-1.168727	0.127681	-0.00757	1.079278	0.0951314
Coeff	1	L/Subm	Fr	rms	
Yrdot	0.002324	-8.7E-05	-0.00063	0.462378	
Yr	0.020906	0.000403	-0.01798	0.072234	
Yvdot	-0.016345	0.000061	-0.00722	0.043182	
Yv	-0.081458	0.001774	0.12175	0.148568	
Zqdot	-0.002666	-0.00051	0.02022	2.00E-16	
Zq	-0.070199	0.003498	0.13373	8.66E-16	
Zwdot	-0.013633	-0.00268	-0.00567	0.088254	
Zw	-0.086685	0.00091	0.14075	0.170412	
Mqdot	0.00014	0.000106	-0.00985	1.66E-16	
Mq	-0.010515	-0.0007	0.03078	1.18E-15	
Mwdot	0.002825	0.000594	-0.00641	0.271656	
Mw	0.023224	-0.00294	-0.02236	0.190629	
Nrdot	-0.000786	-1.9E-05	-0.00058	0.080731	
Nr	-0.006013	-0.00117	0.01230	0.232831	
Nvdot	-0.00089	0.000037	0.00017	0.092017	
Nv	-0.00207	0.000093	-0.01649	0.093355	

Appendix L. Model Scale Experiments Performed at MIT to Determine the Restoring Forces and Moments due to Body Angle

Test #	Water Depth	Subm.	Yaw	Pitch	Velocity
	m	m	Degrees	Degrees	m/sec
SF98	0.795	0.3962	0	0	1
SF99	0.795	0.3962	0	0	0.75
SF100	0.795	0.2509	0	0	1
SF101	0.795	0.2509	0	0	0.75
SF102	0.795	0.5415	0	0	1
SF103	0.795	0.5415	0	0	0.75
SF104	0.795	0.3962	4	0	1
SF105	0.795	0.3962	4	0	0.75
SF106	0.795	0.2509	4	0	1
SF107	0.795	0.2509	4	0	0.75
SF108	0.795	0.5415	4	0	1
SF109	0.795	0.5415	4	0	0.75
SF110	0.795	0.3962	8	0	1
SF111	0.795	0.3962	8	0	0.75
SF112	0.795	0.2509	8	0	1
SF113	0.795	0.2509	8	0	0.75
SF114	0.795	0.5415	8	0	1
SF115	0.795	0.5415	8	0	0.75
SF116	0.795	0.3962	12	0	1
SF117	0.795	0.3962	12	0	0.75
SF118	0.795	0.2509	12	0	1
SF119	0.795	0.2509	12	0	0.75
SF120	0.795	0.5415	12	0	1
SF121	0.795	0.5415	12	0	0.75
SF122	0.795	0.3962	0	4	1
SF123	0.795	0.3962	0	4	0.75
SF124	0.795	0.2509	0	4	1
SF125	0.795	0.2509	0	4	0.75
SF126	0.795	0.5415	0	4	1
SF127	0.795	0.5415	0	4	0.75
SF128	0.795	0.3962	0	8	1
SF129	0.795	0.3962	0	8	0.75
SF130	0.795	0.2509	0	8	1
SF131	0.795	0.2509	0	8	0.75
SF132	0.795	0.5415	0	8	1
SF133	0.795	0.5415	0	8	0.75
SF134	0.795	0.3962	0	12	1
SF135	0.795	0.3962	0	12	0.75
SF136	0.795	0.2509	0	12	1
SF137	0.795	0.2509	0	12	0.75
SF138	0.795	0.5415	0	12	1
SF139	0.795	0.5415	0	12	0.75

Appendix M. Results of Model Scale Experiments Performed at MIT to Determine the Restoring Forces and Moments due to Body Angle

Shaded cells represent values greater than 1.15 standard deviations away from the mean for that speed and submergence.

Test #	Subm.	Spd	Pitch	Yaw	X	Y	Z	Roll	Pitch	Yaw	Normalized Sway Force	Normalized Heave Force	Normalized Yaw Moment	Normalized Pitch Moment
SF101	252	75	0	0	21.3003	-0.3711	-0.4691	0.0403	-0.0416	-0.0368	0	0	0	0
SF107	252	75	0	4	11.5173	0.4629	-0.9813	0.0007	-0.0326	0.0647	0.834		0.1015	
SF113	252	75	0	8	9.4164	0.8015	0.934	-0.0123	-0.033	0.1305	1.1726		0.1673	
SF119	252	75	0	12	9.5864	1.4279	0.1144	0.0221	0.0022	0.1957	1.799		0.2325	
SF125	252	75	4	0	14.0924	0.2787	-0.1434	0.0275	0.0337	0.0086		0.3257		0.0753
SF143	252	75	4	0	19.6126	0.2523	-0.4773	0.005	0.013	0.0121		-0.0082		0.0546
SF131	252	75	8	0	8.7811	0.239	-0.7814	0.026	0.0676	0.0145		-0.3123		0.1092
SF137	252	75	12	0	13.6603	0.2164	-1.1318	0.0324	0.1141	0.0176		-0.6627		0.1557
SF99	398	75	0	0	17.1046	-0.6789	0.2246	0.0136	-0.0526	-0.0474	0	0	0	0
SF105	398	75	0	4	-5.3657	0.1665	0.4406	-0.0349	-0.0367	0.0572	0.8454		0.1046	
SF111	398	75	0	8	-4.6664	0.2552	0.8404	-0.0352	-0.044	0.1106	0.9341		0.158	
SF117	398	75	0	12	-1.2336	0.8619	0.0513	0.0016	-0.0279	0.1713	1.5408		0.2187	
SF123	398	75	4	0	-1.8708	-0.1491	0.0406	0.0037	-0.003	-0.0077		-0.184		0.0496
SF129	398	75	8	0	-2.5511	-0.2603	-0.4944	0.0051	0.0613	-0.0105		-0.719		0.1139
SF135	398	75	12	0	0.1206	-0.2504	-0.7061	0.0112	0.0867	-0.0135		-0.9307		0.1393
SF103	543	75	0	0	8.104	0.2389	-0.1237	-0.0059	-0.0566	0.0065	0	0	0	0
SF109	543	75	0	4	17.8258	0.6527	-0.3395	0.0142	-0.0294	0.0722	0.4138		0.0657	
SF115	543	75	0	8	-5.1133	0.0701	0.808	-0.0382	-0.0898	0.1038	-0.1688		0.0973	
SF141	543	75	0	8	6.0705	0.7525	0.0117	-0.006	-0.0367	0.134	0.5136		0.1275	
SF121	543	75	0	12	1.4662	0.8562	0.0813	0.0017	-0.0339	0.1749	0.6173		0.1684	
SF127	543	75	4	0	1.7731	-0.0883	-0.192	0.0027	0.0111	-0.0039		-0.0683		0.0677
SF133	543	75	8	0	1.3509	-0.0493	-0.4612	0.0006	0.0524	-0.0012		-0.3375		0.109
SF139	543	75	12	0	0.7022	-0.1462	-0.7014	0.0016	0.0688	-0.004		-0.5777		0.1827
SF100	252	100	0	0	28.6384	-0.3125	-0.3894	0.0441	-0.0554	-0.0408	0	0	0	0
SF106	252	100	0	4	7.9829	0.5966	-0.2151	-0.0043	-0.0736	0.0959	0.9091		0.1367	
SF112	252	100	0	8	5.9871	1.1071	0.4481	-0.0174	-0.0857	0.199	1.4196		0.2398	
SF118	252	100	0	12	7.9814	1.8835	-0.1084	0.0196	-0.0386	0.2878	2.196		0.3286	
SF124	252	100	4	0	11.5445	0.3353	-1.0374	0.0309	0.0335	0.0038		-0.648		0.0889
SF130	252	100	8	0	5.4903	0.0547	-0.8964	0.0277	0.0674	0.0062		-0.507		0.1228
SF142	252	100	8	0	14.5032	0.172	-0.7051	0.0024	0.0661	0.0052		-0.3157		0.1215
SF136	252	100	12	0	10.111	0.2624	-1.6039	0.027	0.1288	0.0077		-1.2145		0.1842
SF98	398	100	0	0	14.3126	-0.6259	0.5736	0.0084	-0.1263	-0.0457	0	0	0	0
SF104	398	100	0	4	0.0083	-0.1356	0.7407	-0.0239	-0.1217	0.0659	0.4903		0.1116	
SF110	398	100	0	8	-7.1228	0.7253	0.6748	-0.0415	-0.1104	0.1874	1.3512		0.2331	
SF116	398	100	0	12	-4.4508	1.5926	0.1168	-0.0049	-0.0714	0.2824	2.2185		0.3281	
SF122	398	100	4	0	1.5423	-0.272	-0.0287	0.0086	-0.0199	-0.0161		-0.6023		0.1064
SF128	398	100	8	0	-4.1236	-0.1895	-0.6621	0.0044	0.0651	-0.0037		-1.2357		0.1914
SF134	398	100	12	0	2.9365	-0.4546	-1.0079	0.0183	0.1172	-0.0088		-1.5815		0.2435
SF102	543	100	0	0	8.4768	0.2046	0.2218	-0.0085	-0.1139	0	0	0	0	0
SF108	543	100	0	4	13.1946	0.6244	-0.1566	0.0198	-0.0778	0.0915	0.4198		0.0915	
SF114	543	100	0	8	-5.3702	0.4664	0.8559	-0.0415	-0.1143	0.1744	0.2618		0.1744	
SF140	543	100	0	8	8.2723	0.8434	0.7545	-0.0361	-0.0988	0.194	0.6388		0.194	
SF120	543	100	0	12	-7.2312	1.2275	0.894	-0.0446	-0.1012	0.2688	1.0229		0.2688	
SF126	543	100	4	0	1.5582	-0.3684	-0.1921	0.0073	0.0015	-0.0231		-0.4139		0.1154
SF132	543	100	8	0	-1.9805	-0.1559	-0.6271	-0.0014	0.0606	-0.0079		-0.8489		0.1745
SF138	543	100	12	0	1.0817	-0.526	-0.8771	0.0042	0.1005	-0.0276		-1.0989		0.2144

Units for Interpreting the Results of Model Scale Experiments Performed at MIT to Determine the Restoring Forces and Moments due to Body Angle

Quantity	Units
Depth	cm
Submergence	mm
Speed	Cm/s
Angle	Degrees
Forces	Newtons
Moments	N-m