A METHOD FOR THE ANALYSIS OF THE MDTF DATA USING NEURAL NETWORKS

MOHAMED IBRAHIM

A METHOD FOR THE ANALYSIS OF THE MDTF DATA USING NEURAL NETWORKS

By

© Ibrahim Mohamed, B.Eng.

A thesis submitted to the School of Graduate Studies in partial

fulfillment of the requirements for the degree of Master of Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May 2000

St. John's Newfoundland Canada

ABSTRACT

Numerical simulation techniques are widely used to investigate the behavior of submarines during the design stage. The accuracy of these techniques depend upon the accurate determination of the hydrodynamic coefficients for the model.

The Marine Dynamic Test Facility (MDTF) is a new-six-degree-of-freedom forced motion testing rig. The rig has the ability to test underwater vehicles in a manner that makes it possible to determine the hydrodynamic coefficients in the equations of motion. Multi-variant linear regression is used to obtain the hydrodynamic coefficients from the experimental data.

In this study a neural network technique to identify the hydrodynamic model from experimental data is investigated. The technique uses the model trajectory (motion history) to predict the hydrodynamic coefficients of the model. A single MDTF generated maneuver was used to train the network. The trained network was then tested using different maneuvers and the network predictions were compared to the actual MDTF measured forces and moments.

Results obtained from the neural network technique indicate that the technique can be used to predict the hydrodynamic model of underwater vehicles. The use of this technique can dramatically cut the running costs to conduct experiments on new models.

ACKNOWLEDGMENTS

I would like to express my endless and inexpressible gratitude to my supervisor Dr. M. R. Haddara for the financial, moral, academic support, and encouragement. I would also like to express my gratitude and appreciation to the members of the advisory committee: Dr. C. D. Williams and Mr. M. Mackay for their help and guidance during this study.

I am grateful to my course instructors Dr. L. Lye, Dr. I. Jordaan, Dr. A. Swamidas, Dr. M. Hinchey, and Dr. Haddara for their dedication and academic proficiency.

I would like to thank the staff of the Institute for Marine Dynamics especially Spence Butt, Kent Brett, Bud Mesh, and Caroline Muselet.

Finally I would like to express my gratitude to my fellow graduate students at Memorial University of Newfoundland Faculty of Engineering for their support and encouragement.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION AND LITERATURE SURVEY

The motion of a submarine is a very complicated process that involves many factors such as the submarine shape and depth of operation. In the early 1940's naval researchers needed to predict the motion of naval submarines in the design stage to evaluate the vehicle's performance and ability to carry out a set of prescribed maneuvers. Numerical simulation techniques were widely used during that time.

Due to the development in naval submarine design, the shape of the vehicle became no longer standard, in addition the type and shape of the hydroplanes that control the motion of the boat changed dramatically from one type to another. These changes introduced new challenges to the process of estimating the hydrodynamic coefficients.

Currently there are three major approaches to investigate the motion of a submarine. These are the theoretical approach, the experimental approach, and the parametric identification approach. Parametric identification methods provide new tools that have been used recently to solve a number of ship motion problems. In this research a new tool for predicting the submarine motion, based on a parametric identification technique, is presented.

1.1. THEORETICAL APPROACH.

Numerous studies to calculate the hydrodynamic coefficients of submarines can be found in the literature. Bohlmann (1990) utilized the three dimensional flow theory to calculate the hydrodynamic coefficients of a submarine in all six degrees of freedom. The study solved the potential flow problem using a singularity distribution and was not limited to slender bodies. Watt (1988) developed an analytical method to estimate the value of the added mass coefficients of the submarine. The method calculates the added mass of the submarine by dividing the submarine into several geometrical components, e.g. sail, tail, fins,.., etc. The added mass was calculated for each component by calculating the added mass for each representative ellipsoid.

Mackay (1988) developed a semi-empirical method for calculating the out-of-plane heave force and pitch moment for a submarine. The method is based on potential flow theory and a first order panel method. The method over-predicts the out-of-plane component due to the absence of the modeling of the cross flow separation in the after-body of the submarine. Mackay and Conway (1990) modeled the cross flow separation by introducing a single doublet sheet along an experimentally determined separation line.

Papoulias and Mckinley (1994) studied the stability of a submarine during its steady state vertical ascent. They considered a free positive ascent case to investigate the response with respect to the stern and the bow plane deflections, excess buoyancy, and relative position between the center of gravity and the center of buoyancy. A nonlinear study of the dynamic stability of a submarine in a dive maneuver was also presented by Papoulias and Papadimitriou (1995). The researchers presented a framework to calculate the critical

speed of the vehicle in terms of its metacentric height, the longitudinal separation of the center of gravity and center of buoyancy, and the dive plane angle.

1.2. Experimental Methods

Captive model tests are used to determine the coefficients in the equations of motion of a model. The use of the rotating arm and planar motion mechanism (PMM) allowed the incorporation of some nonlinear terms and time histories effects, see Feldman (1995) and Kalske (1992). Multi-degree-of-freedom test rigs are being used at present to determine these coefficients, see Williams et al. (1999). Experimental methods are expensive to use and their results suffer from scale effects.

An alternative to the captive model tests is the wind tunnel test. Experiments on a deeply submerged submarine model can be conducted. The advantage of this procedure over conventional captive model tests is that in the wind tunnel the duration of the test can be extended to a much longer time than in the case of towing tank experiments. A wind tunnel experiment can also be used for flow visualization and for studying the effects of propulsion on the submarine model hydrodynamics, see Watt et al. (1993).

1.3. Parametric Identification Approach

Parametric identification techniques provide another way for determining the hydrodynamic model of the submarine. Parametric identification approaches use the relation between the input and the output to estimate the hydrodynamic forces and

moments while in the experimental approach these forces and moments are measured directly. The method assumes a certain structure for the mathematical model. The coefficients of the mathematical model are then estimated such that the model produces an output which matches the measured output. Artificial neural networks (ANN) have been used for many surface ships as a parametric identifier of the vehicle's hydrodynamics.

Neural networks have been shown to yield robust models for ship motions. Haddara (1995) used the technique to analyze the rolling motion of a ship at sea. He also used neural networks to characterize the ship coupled heave and pitch motions, see Haddara and Xu (1999), and to analyze the free roll decay curve, see Haddara and Hinchey (1995). Hinchey also used this technique to fit ship resistance data for an R-Class icebreaker ship model, see Hinchey (1994), and to predict the loads on sub-sea robots, Rivera and Hinchey (1999). Haddara and Wang (1996) used this technique to identify the coupled sway and yaw motion for a ship undergoing a series of standard maneuvers. Lainiotis and Plataniotis (1994) used an adaptive neural network to predict the current as well as future ship position as a space vector.

Multi-linear regression may not yield accurate estimates. The accuracy of the technique decreases as the number of variables increases. Neural networks have been shown to yield a robust result regardless of the number of variables in the functional relation. However the classical division of force into components which are functions of added masses and damping coefficients is usually lost.

1.4. Scope and Objectives.

The objective of this study was to develop a parametric identification tool for predicting the hydrodynamic forces acting on a submarine model using the model motion history. The proposed tool uses a neural network technique to identify the hydrodynamic forces and moments. Data obtained from multi degree-of-freedom tests were used to train the network.

The fully trained network was then used to generate more maneuvers with a limited computational demand. This will reduce the cost associated with experimental testing of submarine models since only a limited number of maneuvers have to be actually performed to obtain the full hydrodynamic model of the submarine.

1.5. Methodology

A neural network model was developed. Digitally generated data and experimental data were used to train and validate the model. The motion of a submarine undergoing a coupled heave and pitch sinusoidal motion was numerically generated at different frequencies using a Runge-Kutta algorithm. Experimental data were obtained from the Marine Dynamic Test Facility (MDTF). The maneuvers considered were horizontal circular arcs, and sway chirp maneuvers. A three-layer feedforward backpropagation neural network was used for system identification. The network prediction was then compared to a multiple linear regression estimate of the model hydrodynamic forces and moments.

1.6. THESIS OUTLINE

Chapter 2 gives background information about neural networks and their architecture; it also discusses some of the training algorithms. Chapter 3 describes the mathematical formulation of the equations of motions, the neural network algorithm, and the multivariate linear regression. Chapter 4 describes the experimental setup and the neural network approach used. Results obtained using experimental data as well as digitally generated data are included in Chapter 5 while Chapter 6 outlines the conclusions. Some recommendations for future work are stated in chapter 7.

2. NEURAL NETWORKS

Tasks that involve pattern recognition or intelligence are very difficult to automate; such tasks are performed easily by animals in their daily activities. Understanding how the biological neural system works allowed researchers to develop artificial systems that mimic the biological brain in performing a variety of computations and intelligent tasks.

2.1. BIOLOGICAL NEURAL SYSTEMS

A biological neural system consists of a number of neurons; each neuron is composed of a cell body, a tubular axon, and a number of dendrites, see Figure 2.1. Data propagates from the dendrites of one neuron to the other through a gap synapse. The magnitude of the signal received by a neuron depends upon the strength "efficiency" of the synaptic connection between that neuron and the one that is sending the signal. A neuron will fire a signal if its excitation exceeds a critical amount, called the threshold. It will then send that signal along its axon. A signal propagates down the axon and reaches the synapses, sending signals of various strengths to other neurons.

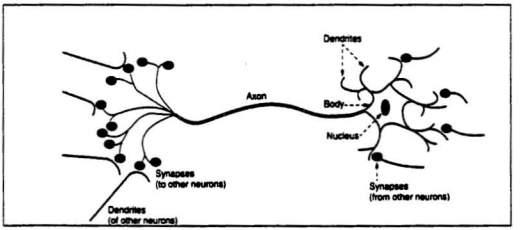


Figure 2.1 Biological neuron

2.2. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) consists of a number of processing elements *(nodes)* equivalent to the neurons in biological systems. Synapses are modeled by links between these nodes while the synaptic efficiencies are replaced by weights. Nodes at the same level form a layer. Each ANN consists of an input layer, output layer, and a number of hidden layers. Inputs to each node are summed up and the node then fires an output according to the assigned activation function.

All ANN models proposed so far are based on the following assumptions

1. The position of the node of the incoming connection is irrelevant.

2. Each node has a single output value, which is distributed to other nodes via connections irrespective of their position.

3. All inputs to the same level come at the same time or remain active long enough for computations to occur, Mehrotra et al (1997).

2.3. ANN SQUASHING FUNCTIONS

Squashing functions can be classified into differentiable and non-differentiable functions. Differentiable and non-differentiable functions can be defined according to the function shape. A differentiable function has a smooth shape and it is needed for some adaptive learning algorithms. A non-differentiable, discontinuous, function gives a true binary output. Discontinuous functions can be used in classification networks.

2.3.1. STEP FUNCTIONS:

A step function is defined as

$$f(net) = \begin{cases} a & \text{if } net < c \\ b & \text{if } net > c \end{cases} \qquad 2.1$$

This function is easy to implement. The net threshold is represented by c, see Figure 2.2, and the function has only two states ON if net > c and OFF otherwise.

The step function is used in class identification where the input belongs to a certain class if its value is greater than a certain value.

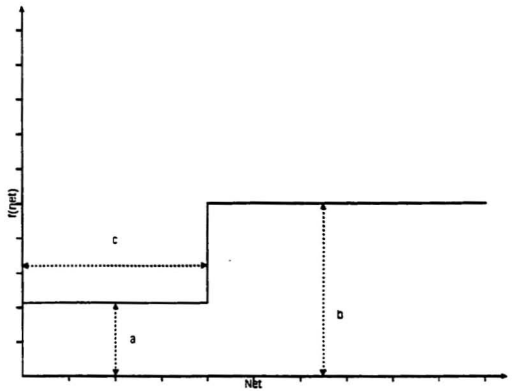


Figure 2.2 Step Function

2.3.2. SIGMOID FUNCTIONS:

This is the most commonly used function, also called the S shaped function; the activation of the node is as follows:

$$f(net) = Z + \frac{1}{1+e^{-a(net)+b}} \qquad 2.3$$

A typical sigmoid function is shown in Figure 2.3. The advantage of such a function is its smoothness.

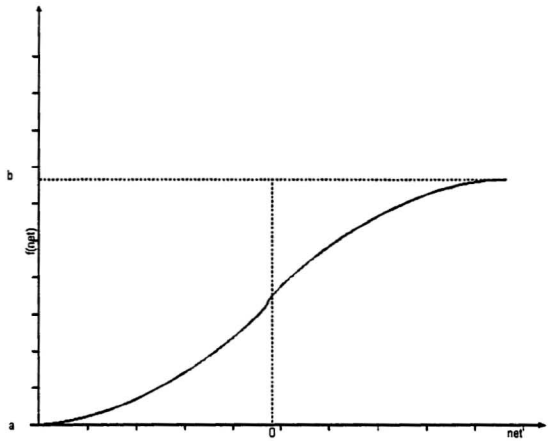


Figure 2.1 Sigmoid Function

2.3.3. GAUSSIAN FUNCTIONS.

Another continuous function that is used in the ANN is known as the bell shaped function, see Figure 2.2

$$f(net) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{net-\mu}{\sigma}\right)^2} \qquad 2.4$$

Such a function requires prior knowledge of certain statistical values such as the mean and the variance of the input, which imposes some limits on its application. The bell shaped function can be used in class identification depending on how close the input is to the mean value μ.

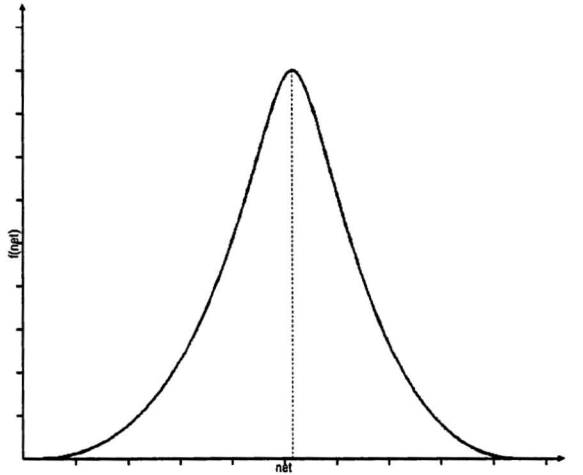


Figure 2.2 Bell shaped function

2.4. Types of Neural Networks:

A single node neural network is insufficient for most practical problems and a multi-node network is normally used. The way those nodes are connected to each other determines how the computational tasks are performed by the network. Neural networks are classified according to their architectural design. This section discusses some of the most popular designs.

2.4.1. LAYERED NETWORK

This type of ANN, also called recurrent neural network, is partitioned into layers. There is no connection between layer k to j if k > j, see Figure 2.3; connections between nodes in the same layer exist except for the input layer.

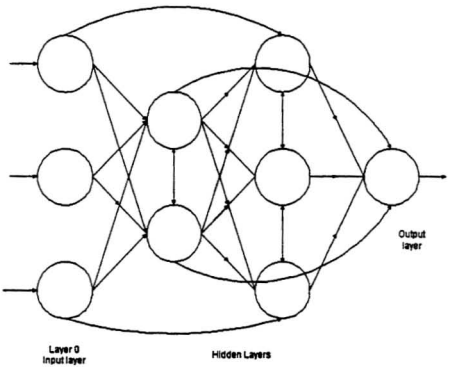


Figure 2.3 Layered neural network

2.4.2. ACYCLIC NETWORK

This is a subclass of the layered network with no intra-layer connections i.e. no connections between nodes in the same layer. These are simpler nets than the recurrent ones with less computational demands, which make them easy to implement. Figure 2.4 shows an acyclic neural network.

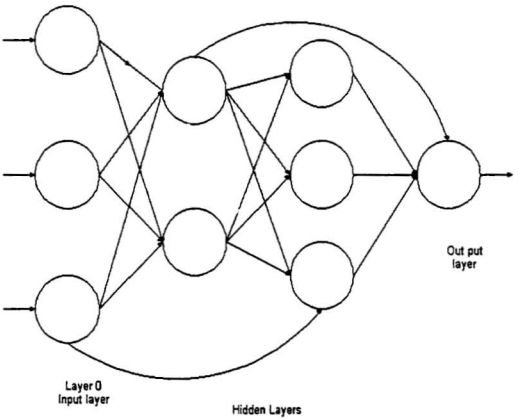


Figure 2.4 Acyclic neural network

2.4.3. FEEDFORWARD NETWORKS

Again this is another subclass of layered networks, where the only connections exist between layer j and k if k = j +1. These are the most in-use networks and they are described by the number of nodes in each layer. For example, Figure 2.5 shows a 3.2.3.2 feedforward net. In general feedforward networks have one or two hidden layers.

From this point forward only feedforward neural networks with one hidden layer will be considered.

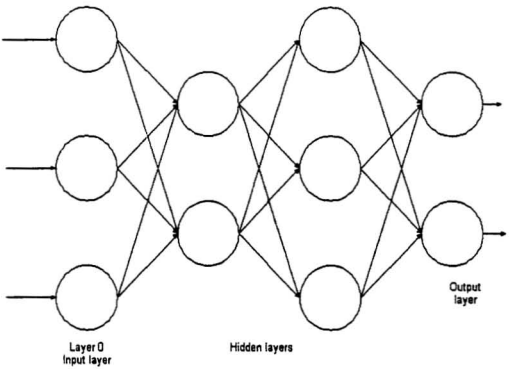


Figure 2.5 Feedforward neural network

2.5. Backpropagation Training Algorithm

The objective of a supervised training algorithm such as the backpropagation algorithm is to minimize the mean square error between the desired output and the network one. To elaborate on this one can consider the framework proposed by Mehrotra et al. (1997).

The input pattern to the network is P. For each input vector x_P there is a desired output

$$d_P=(d_{P,1} ; d_{P,2} ;..... ; d_{P,k}) \qquad 2.5$$

One would expect some error between the desired output and the network output. The algorithm objective is to manipulate the weights so that this error would be minimal. If the network output is:

$$o_P=(o_{P,1} ; o_{P,2} ;..... ; o_{P,k}) \qquad 2.6$$

Then the network error is

$$Error = \sum_{p=1}^{p} Err(\mathbf{o_p}, \mathbf{d_p}) \qquad 2.7$$

The function $Err(\mathbf{o_p}, \mathbf{d_p})$ should be a nonnegative function. One way of computing this function is to express

$Err(\mathbf{o_p}, \mathbf{d_p}) = (\ell_{p,j})^2$ where $\ell_{p,j} = |\mathbf{o_{p,j}} - \mathbf{d_{p,j}}|$

This function is a differentiable function. Moreover it is easy to apply to the backpropagation algorithm since the derivative of the resulting quadratic function is linear and easy to compute.

To discuss the backpropagation in depth one can assume the following case for a three layer network.

Each node receives an input vector of $x_{p,j}$ where 'p' represents the pattern number while 'i' represents the i^{th} node in the layer.

The input to the j^{th} node in the hidden layer is $\text{net}_j = \sum_{i=0}^{n} \omega_{i,j} x_{p,i}$

The output of the j^{th} node $x_{p,j} = \delta(\text{net}_j)$, where δ is a sigmoid function.

The input to the k^{th} node in the output layer is $\text{net}_k = \sum_j \omega_{k,j} x_{p,j}$

The output of the k^{th} node in the output layer is $o_{p,k} = \delta(\text{net}_k)$.

One way to minimize the Mean Square Error (MSE) is to apply the gradient descent method. Since the net output depends on the net weights, the MSE is also a function of the weights w. According to the gradient descent the direction of the weights change is the same as $-\partial E / \partial w$, see Widrow and Lehr (1990). Following this scenario the weight change can be described as follows

$$\Delta\omega_{k,j} \propto \left(-\partial E / \partial \omega_{k,j}\right) \tag{2.8}$$

$$\Delta\omega_{i,j} \propto \left(-\partial E / \partial \omega_{i,j}\right) \tag{2.9}$$

Since the change in the weights results in changing the net output, one can write

$$\frac{\partial E}{\partial \omega_{k,j}} = \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial \omega_{k,j}} \tag{2.10}$$

Equation 2.10 can be simplified into

$$\frac{\partial E}{\partial \omega_{k,j}} = -2(d_k - o_k)\delta'(net_k)x_j \tag{2.11}$$

Now let us consider the weights between the input and hidden layers.

$$\frac{\partial E}{\partial \omega_{i,j}} = \sum_{k=1}^{k}\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial x_j}\frac{\partial x_j}{\partial net_j}\frac{\partial net_j}{\partial \omega_{i,j}} \tag{2.12}$$

$$\frac{\partial E}{\partial \omega_{i,j}} = \sum_{k=1}^{K}\{-2(d_k - o_k)\delta'(net_k)\omega_{k,j}\delta'(net_j)x_i\} \tag{2.13}$$

The change of weight is then calculated for the weight between the output and hidden layer using the actual error while the change of weight for the lower level is calculated using the weighted sum of errors coming to the hidden node from the output node. The error at a node is calculated in the direction opposite to the propagation of the node output. Applying this analogy to equations 2.11 and 2.13 one can write the change of weight for network training using the p^{th} pattern as:

$$\Delta\omega_{k,j} = \eta\delta_{p,k}x_{p,j} \tag{2.14}$$

$$\Delta\omega_{i,j} = \eta\mu_{p,j}x_{p,i} \tag{2.15}$$

where

$$\delta_{p,k} = (d_{p,k} - o_{p,k}) * o_{p,k} * (1 - o_{p,k}) \qquad 2.16$$

and

$$\mu_{p,j} = \sum_{k} \delta_{p,j} \omega_{k,j} * x_{p,j} * (1 - x_{p,j}) \qquad 2.17$$

Training using equation 2.14 and 2.15 results in the manipulation of the network weights to achieve the optimum weights that result in the minimum error. Weights can be updated after each pattern (training per pattern) or after all the training patterns pass through the net (training per epoch). The criteria for terminating the training and the choice of the suitable training rate "η" depends on the user's experience.

2.6. Momentum Update

Backpropagation leads the neural network error to a certain minimum. There is a chance that this minimum is a local minimum different than the global minimum. The network then will stick to this local condition and the algorithm will prevent the error from decreasing further. One can prevent this by forcing the algorithm to update the weights depending on the average gradient of the MSE of a small region rather than at a certain point. Averaging $\partial E / \partial w$ in a small region allows the algorithm to update the weights in the general direction avoiding local minima.

Averages can be very expensive in calculation. Rumelhart et al. (1986) suggested a shortcut using a momentum term as follows:

$$\Delta\omega_{k,j}(t+1) = \eta\delta_k x_j + \alpha\Delta\omega_{k,j}(t) \qquad 2.18$$

The term α is a momentum term yet its optimum value is not known. With the addition of this term the weight change is influenced by the weight change from the previous step as well as the change suggested by the gradient descent.

The effect of the use of a momentum term in a weight update depends upon the value chosen for α. A well-chosen value can significantly reduce the time needed for training the network. A value of 0 implies that the past step does not affect the current update while a value closer to 1 suggests that the current error does not affect the training. Since the weight update in this algorithm depends upon the direction of the previous weight update, the direction chosen in an early stage of the training procedure can strongly bias future weight updates and can restrict the network to a certain region preventing the algorithm from exploring other regions.

3. MATHEMATICAL FORMULATION

3.1. Equations of Motion

The equations describing submarine motion can be obtained using Newton's second law.

$$M\ddot{X} = F \qquad 3.1A$$

where **M** is the mass matrix and **X** and **F** are the displacement and the force vectors respectively.

One can expand on equation 3.1A to describe the motion of an underwater vehicle as:

$$(\mathbf{I}+\mathbf{A})\dot{\mathbf{x}} = \mathbf{F(x)}+\mathbf{X(x)}+\mathbf{S(x)} \qquad 3.1B$$

where **x** is the extended state vector containing the vehicle's translation and rotational velocities, Euler angle displacements, and control deflections, see Figure 3.1. **I** is a matrix representing the inertial properties of the vehicle, **A** is the added mass matrix, **F** is the hydrodynamic force vector, **X** represents the force vector arising from axis rotation, buoyancy, gravity, and, propulsion, and **S** represents other external force vectors.

For small amplitude motion of a submarine model attached to a forced motion testing rig with port - starboard symmetry, and using the coordinate system shown in Figure 3.1, equation 3.1B can be expressed in a component form as:

$$\ddot{x}_3 + b_{33}\dot{x}_3 + b_{35}\dot{x}_5 + c_{35}x_5 = F_{3o}\sin(\omega t) \qquad 3.2$$

$$\ddot{x}_5 + b_{53}\dot{x}_3 + b_{55}\dot{x}_5 + c_{55}x_5 = F_{5o}\sin(\omega t) \qquad 3.3$$

where x_3 and x_5 are the heave and pitch displacements respectively, $\dot{x}_i$ and $\ddot{x}_i$ are the first and second derivative of the displacement, F_{3o} and F_{5o} are the heave force and pitch moment amplitudes, ω is the exciting frequency, and t is time.

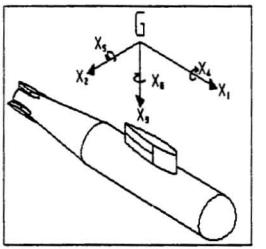


Figure 3.1 Model coordinates system

The hydrodynamic coefficients b_{ij}, where i , j = 3, 5, represent the submarine damping coefficients. The coefficients $c_{i,j}$ are the pitch restoring coefficients which act to return the submarine to the position of vertical equilibrium. The right hand sides of equations 3.2 and 3.3 represent the heave force and pitch moment that are imposed on the submarine model by the forced-motion apparatus as the submarine follows the path. For

the validation of the proposed technique a generic submarine with the following arbitrary hydrodynamic coefficients was used to generate the numerical motion data.

b_{33}	b_{35}	b_{55}	b_{53}	c_{55}	c_{35}	F_{3o}	F_{5o}
0.234	0.158	0.222	0.58	30.78	0.238	3	2

Table 3.1 Numerical model coefficients

Equations 3.2 and 3.3 were solved numerically using a Runge-Kutta algorithm to obtain the time series of the heave and pitch displacements, velocities, and accelerations.

3.2. NEURAL NETWORK MODEL

The neural network model used in this study consists of an input layer, hidden layer, and an output layer. Figure 3.2 shows the details of the neural network structure.

The input to the j^{th} node in the hidden layer consists of the weighted sum of the i^{th} components in the input vector:

$$M_j = \sum_{i=1}^{6} w_{ij} h_i \qquad 3.6$$

where w_{ij} is the weight between the i^{th} node in the input layer and the j^{th} node in the hidden layer, $h_1 = \dot{x}_3$, $h_2 = \ddot{x}_3$, $h_3 = \dot{x}_5$, $h_4 = \ddot{x}_5$, $h_5 = x_5$, and $h_6 = 1$. The input to the j^{th} node in the hidden layer is transformed via a squashing function as:

$$G(M_j) = \frac{1}{1 + e^{-M_j}} \qquad 3.7$$

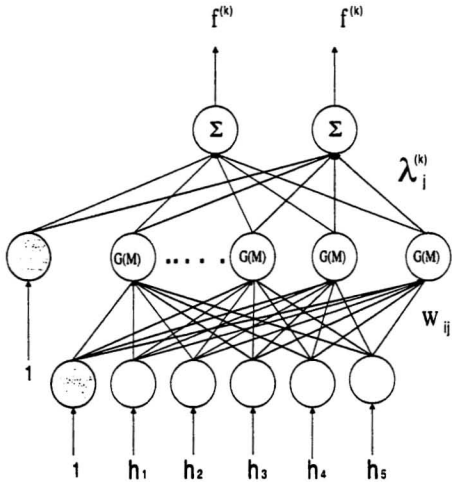


Figure 3.2 Neural network structure

The output of the k^{th} node in the output layer is given as:

$$f^{(k)} = \sum_{j=1}^{16} \lambda^{(K)}{}_j \, G(M_j)$$

3.8

where k is either 3 or 5, and $\lambda^{(k)}{}_j$ is the network weight between the j^{th} node in the hidden layer and the k^{th} node in the output layer.

3.3. LINEAR REGRESSION MODEL

Multiple linear regression is often called "model fitting". It depends upon the assumption of a mathematical model that fits the data at hand. The adequacy of the technique depends upon the accuracy of the assumed model.

Considering the model described in Equation 3.9 there are k regressors and i is the number of observations, i= 1,2,…,n.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \qquad 3.9$$

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \qquad 3.10$$

One can express equation 3.10 in a matrix form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \qquad 3.11$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

$$\text{and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

The least squares method determines the value of the parameters β_k so that the sum of the squares of the error is minimized. The least squares function is

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \qquad 3.12$$

equation 3.12 can be rewritten as

$$L = \mathbf{y}'\mathbf{y} - 2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \qquad 3.13$$

in order for the function L to be minimized with respect to $\boldsymbol{\beta}$, the least squares estimator $\hat{\boldsymbol{\beta}}$ must satisfy.

$$\left.\frac{\partial L}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = 0 \qquad 3.14$$

From equation 3.14 the least squares estimator is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \qquad 3.15$$

Montgomery (1997) presented this framework of least squares as an unbiased estimator of the parameter $\boldsymbol{\beta}$ in the linear regression model.

3.3.1. UNCOUPLED HEAVE MODEL

Given n points, the linear regression model for the uncoupled heave force estimate is.

$$F_i = \beta_0 + \beta_1 \dot{x}_i + \beta_2 \ddot{x}_i \qquad i = 1,2,3,\ldots,n \qquad 3.16$$

3.3.2. UNCOUPLED PITCH MODEL

Using the least squares method the pitch moment is assumed to follow the mathematical model shown:

$$F_i = \beta_0 + \beta_1 x_i + \beta_2 \dot{x}_i + \beta_3 \ddot{x}_i \qquad i = 1,2,\ldots,n \qquad 3.17$$

3.3.3. COUPLED HEAVE AND PITCH MODEL

The coupled heave and pitch digitally generated motion can be expressed in the following linear equations:

$$F_{3i} = \beta_{03} + \beta_{31}\dot{x}_{3i} + \beta_{32}\ddot{x}_{3i} + \beta_{33}x_{5i} + \beta_{34}\dot{x}_{5i} \qquad 3.18$$

$$F_{5i} = \beta_{05} + \beta_{51}\dot{x}_{3i} + \beta_{52}x_{5i} + \beta_{53}\dot{x}_{5i} + \beta_{54}\ddot{x}_{5i} \qquad i=1,2,\ldots,n \qquad 3.19$$

Rearranging equations 3.23 and 3.24 into matrix form yields:

$$\mathbf{F}_3 = \mathbf{X}_3\boldsymbol{\beta}_3 \qquad 3.20$$

$$\mathbf{F}_5 = \mathbf{X}_5\boldsymbol{\beta}_5 \qquad 3.21$$

where

$$\boldsymbol{\beta}_i = \begin{bmatrix} \beta_{i0} \\ \beta_{i1} \\ \beta_{i2} \\ \beta_{i3} \\ \beta_{i4} \end{bmatrix} \qquad 3.22$$

$$\mathbf{X}_3 = \begin{bmatrix} 1 & \dot{x}_{31} & \ddot{x}_{31} & x_{51} & \dot{x}_{51} \\ 1 & \dot{x}_{32} & \ddot{x}_{32} & x_{52} & \dot{x}_{52} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \dot{x}_{3n} & \ddot{x}_{3n} & x_{5n} & \dot{x}_{5n} \end{bmatrix} \qquad 3.23$$

and

$$\mathbf{X}_5 = \begin{bmatrix} 1 & \dot{x}_{31} & x_{51} & \dot{x}_{51} & \ddot{x}_{51} \\ 1 & \dot{x}_{32} & x_{52} & \dot{x}_{52} & \ddot{x}_{52} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \dot{x}_{3n} & x_{5n} & \dot{x}_{5n} & \ddot{x}_{5n} \end{bmatrix} \qquad 3.24$$

Results obtained from the neural network technique and the linear regression calculations are shown in Chapter 5. Least squares calculations are shown in Appendix A.

4. FRESH WATER TOWING TANK EXPERIMENTS

4.1. MARINE DYNAMIC TEST FACILITY (MDTF)

The MDTF is a multi-degree-of-freedom forced-motion test rig. It has the capability of forcing the test model in an arbitrary motion trajectory, within the rig and the towing tank limits. The motion parameters as well as the forces acting on the model are measured. The MDTF is part of the testing facilities at the Institute for Marine Dynamics (IMD). The MDTF was commissioned in early 1999 after its performance had been evaluated through a prototype rig (1993 to 1996).

During the period from October to December 1999 maneuvers were conducted using a standard submarine model. The MDTF description, experiment setup, and the maneuvers performed will be briefly presented in the following sections.

4.1.1. MDTF SETUP

The general arrangement of the MDTF with the model mounted on it, is shown in Figure 4.1. The whole configuration is mounted on the carriage of the towing tank with the model deeply submerged. The MDTF rig consists of two lateral railed carriages supporting two vertical telescopic struts; a picture of the sub-carriage is shown in Figure 4.2. Two lateral servomotors, shown in Figure 4.3, drive the lateral struts sideways parallel to the carriage transverse centerline while two other servomotors, Figure 4.4, drive the telescopic struts in the vertical direction. A universal joint, see Figure 4.5, connects the sting to the struts. The sting is connected to the model via a six component internal balance. The six components of the reaction loads were measured at the model balance-resolving center (BRC). For detailed information on the MDTF and its setup see Williams et al. (1999).

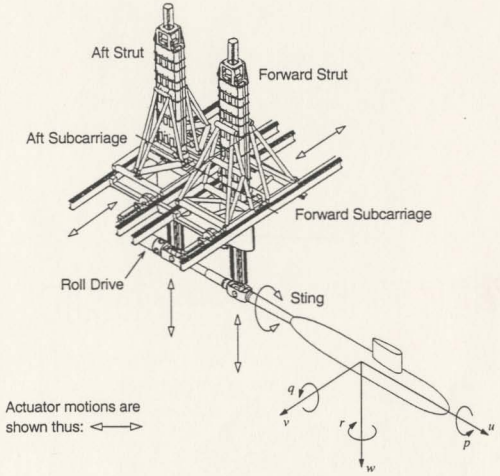


Figure 4.1 MDTF General Arrangement

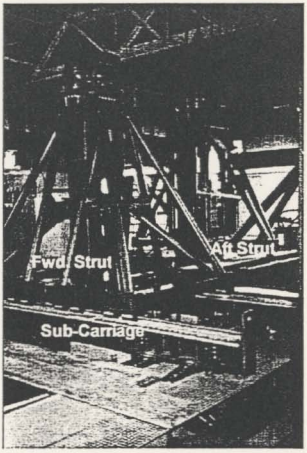


Figure 4.2 MDTF Sub-Carriage

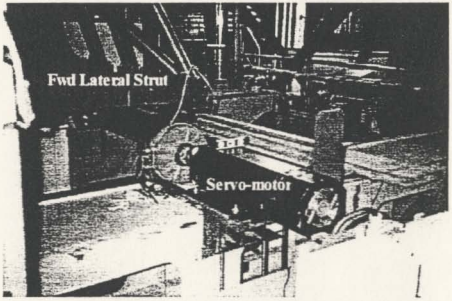


Figure 4.3 Lateral Motor Arrangement

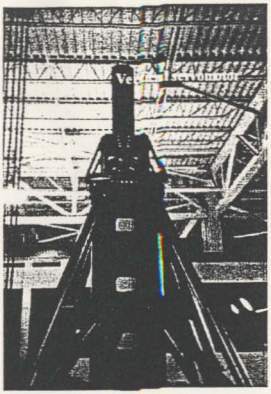


Figure 4.4 Vertical Motor Arrangement

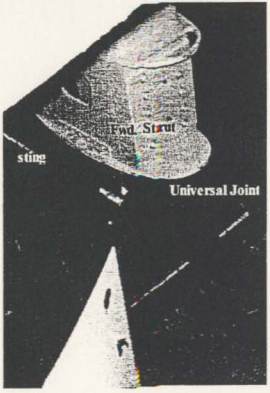


Figure 4.5 Forward Strut Universal Joint

4.1.2. EXPERIMENTAL SETUP

The model, IMD 497 Standard submarine model, was 4.445 m in length and 0.5 m in diameter at the mid-body. The BRC is located 5.453 m from the aft strut. The forward and afterward struts are 2 m apart. Model dimensions and coordinate system are shown in Figure 4.1 to Figure 4.8. The model displacements were calculated from the strut displacements as shown in equations 4.1 to 4.4.

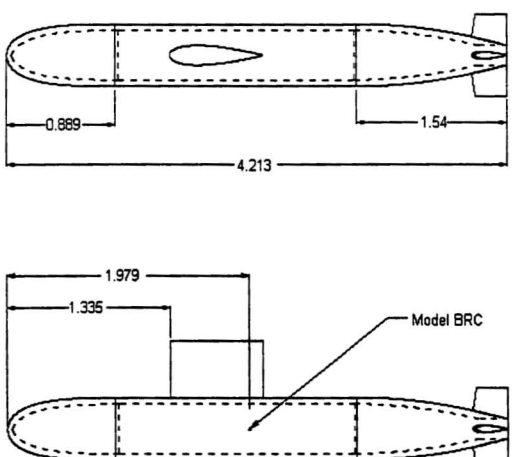


Figure 4.1 DREA Standard submarine model dimensions

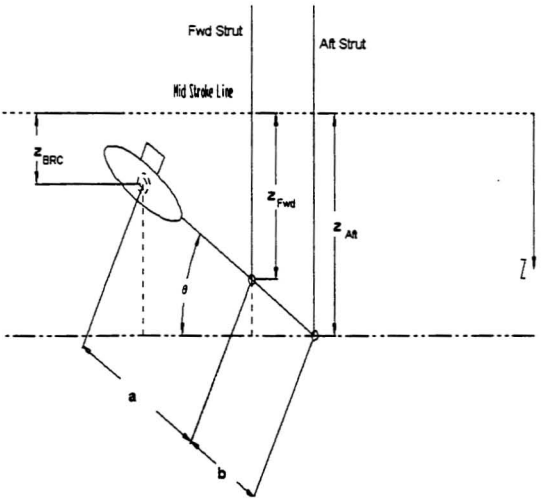


Figure 4.7 Side view of the MDTF setup

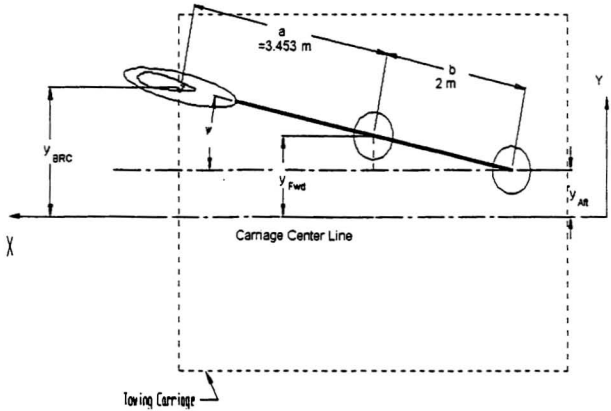


Figure 4.8 Top view of the MDTF setup

$$z_{BRC} = z_{aft} - \left(\left[\frac{a+b}{b} \right] \left[z_{aft} - z_{fwd} \right] \right) \tag{4.1}$$

$$\theta = \sin^{-1} \left(\frac{\left[z_{aft} - z_{fwd} \right]}{b} \right) \tag{4.2}$$

$$y_{BRC} = y_{aft} + \left(\left[\frac{a+b}{b} \right] \left[y_{fwd} - y_{aft} \right] \right) \tag{4.3}$$

$$\psi = \sin^{-1} \left(\frac{\left[y_{fwd} - y_{aft} \right]}{b} \right) \tag{4.4}$$

The above equations are based on a "rigid" sting assumption and they describe the model displacements relative to a set of axes which are fixed to the towing carriage but which translate longitudinally relative to a fixed earth reference point.

4.2. MANEUVERS

The design of the MDTF allows the model to perform non-conventional maneuvers. The maneuvers that have been conducted can be divided into four categories. These are described as follows:

4.2.1. CIRCULAR ARC MANEUVERS.

In these maneuvers the model is forced to move in a series of arcs separated by transition regions. The model has a constant speed, u, and a constant pitch rate, q, during the vertical arc maneuvers and constant yaw rate, r, during the horizontal arc maneuvers. Heave and sway displacements may have nonzero values. The arcs have a non-dimensional turning rate, t', defined as.

$$t' = L/R \qquad 4.5$$

where L is the model length, R is the turning radius. $R = \frac{u}{r}$ in the horizontal plane and $R = \frac{u}{q}$ in the vertical maneuvers. t' represents the non-dimensional turning rate (r' in the horizontal plane and q' in the vertical plane). Figure 4.9 shows a plot of the aft strut displacement, forward strut displacement, and model trajectory for an arc maneuver in the horizontal plane.

4.2.2. LINEAR CHIRP MANEUVERS

Chirps are harmonic maneuvers with constant amplitude and varying frequency. The frequency ranged from 0.1 up to 0.5 Hz. These frequencies were swept in the form of low-high-low sawtooth shape. Figure 4.10 shows a chirp maneuver in the vertical plane.

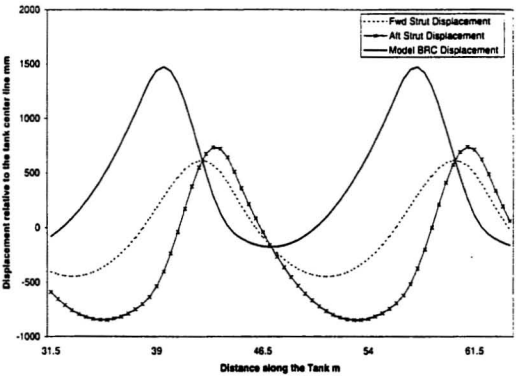


Figure 4.9 Arc maneuver trajectory in the lateral plane r' = 0.2

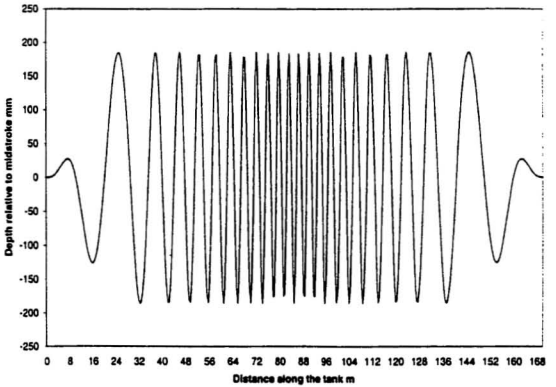


Figure 4.10 Pure heave chirp, frequencies 0.1 to 0.5 Hz

4.2.3. PURE HARMONICS

A series of harmonic runs were performed during the testing period. The frequency ranged from 0.15 to 0.25 Hz with amplitude range of 0.07 to 0.4 m.

4.2.4. MULTI-DEGREE-OF-FREEDOM MANEUVERS

A limited number of maneuvers were tried with the model moving in three, four, and five degrees of freedom; most of these were chirps. During the testing period a set of non-conventional maneuvers such as cone helical maneuvers were also tried. More information on these can be found in Williams et al (1999).

The above mentioned maneuvers were performed with four different configurations of the submarine. These configurations, shown in Figure 4.11, were bare hull sub; hull and sail; hull and tail; and hull sail and tail (full configuration). A list of all maneuvers performed is given in Appendix C.

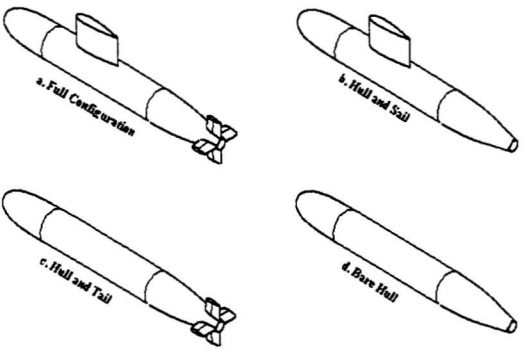


Figure 4.11 Model configurations

4.3. Data Acquisition

Data from load cells, accelerometers, and potentiometers were sampled at 50 points/sec frequency. The GEDAP® software stored all measured time series in binary format. A split routine was then used to split each run file into separate run channels and preliminary statistical calculations and corrections were performed. An export routine was then applied to transfer the run at hand to ASCII format. The exported file had the format shown in Table 4. 1.

4.3.1. TIME TARE AND COORDINATES TRANSFORMATION

During testing, data collection started before the actual maneuver started, moreover the maneuvers started some time before the carriage reached its constant speed. Time before the steady speed is considered tare time and all data collected during that time segment had to be removed.

The six force components were measured at the model BRC located at 5.453 m from the aft strut. A simple transformation of the struts displacements, based on equations 4.1 to 4.4 and assuming a rigid sting, was needed to obtain the model linear and rotational displacements at the BRC.

Column #	GEDAP Channel #	Variable	Units
1	Implicit Variable	Time	sec
2	ch.031	Carriage speed	m/s
3	ch-sname.001	Fx	N
4	ch-sname.002	Fy	N
5	ch-sname.003	Fz	N
6	ch-sname.004	Mx	N-m
7	ch-sname.005	My	N-m
8	ch-sname.006	Mz	N-m
9	ch.022	FV. Pos.	m
10	ch.024	AV Pos.	m
11	ch.026	FL Pos.	m
12	ch.028	AL Pos.	m
13	ch.007	Fwd. Ax	g
14	ch.008	Fwd Ay	g
15	ch.009	Fwd Az	g
16	ch.010	Aft Ax	g
17	ch.011	Aft Ay	g
18	ch.012	Aft Az	g

Table 4. 1 ASCII file format "HST_xxxxx. asc"

MDTF Prepshop is a Windows application written by the author to perform the above mentioned steps, tare removal and coordinate transformation. It allows the user, through a graphic interface, to select the ASCII run file, the time segment, and the output file name. The program will then run the scenario mentioned earlier. The program also calculates the model's heave, pitch, yaw, and sway rates at the BRC. Prepshop was written in Visual Basic® 6 and its code is shown in Appendix D.

4.4. Filtering

The data were then filtered using the Matlab® Wavelet toolbox. A wavelet filter allows the use of a variable-sized windowing technique where a long time interval is used when

we want more precise low frequency information and shorter interval where high frequency information is needed.

4.4.1. WHAT IS A WAVELET ?

A wavelet is a waveform with a limited duration that has a zero average value. While Fourier analysis breaks the signal into sine waves with different frequencies, wavelet analysis breaks the signal into shifted and scaled versions of the original wavelet.

A continuous wavelet is the sum over time of the original signal multiplied by a scaled, shifted version of the chosen wavelet function φ.

$$C(scale, shift) = \int_{-\infty}^{\infty} f(t)\varphi(scale, shift)dt \qquad 4.6$$

Equation 4.6 results in many wavelet coefficients C. These coefficients are a function of the shift and scale of the function.

Matlab® decomposes the wavelet signal into two components, a high frequency and a low frequency component, by removing the high frequency component. Each wavelet performs a single level filtering. To obtain a smoother signal the filtering process can be repeated a number of times on the low frequency component. This is called multilevel decomposition. Matlab® includes a number of wavelet families, see the Wavelet Toolbox user guide (1997).

4.4.2. FILTERING PROCEDURE

The wavelet used in these runs was a Daubechies 'db5' with a decomposition level 6. This filter worked fairly well in eliminating most of the noise of the signal while preserving the general form of the data. The filtering script, which is called *clean.m*, is a Matlab M-file that reads all the channels from the Prepshop file; filters them, then exports the output to a *filename*.flt file. The procedure is to split the file into individual channels and perform the following steps.

1- Decompose the signal using Daubechies wavelets with six levels and calculate the scale and position.

2- Determine the default values for de-noising the signal.

3- De-noise and compress the signal using a global positive threshold.

The filtering script *clean.m* as well as a number of selected runs with filtering are shown in Appendix E.

4.5. Neural Network Approach

Tow different neural networks were used to describe the horizontal and the vertical maneuvers. In the case of horizontal maneuvers the input consisted of the sway velocity, $\dot{x}_2$, the sway displacement, x_2, the yaw rate, $\dot{x}_6$, and the yaw angle, x_6. The input to the j[th] node in the hidden layer, M_j, is given by:

$$M_j = \sum_{i=1}^{5} w_{ij} h_i \qquad 4.7$$

where , $h_1 = \dot{x}_2$, $h_2 = x_2$, $h_3 = \dot{x}_6$, $h_4 = x_6$, $h_5 = 1$.

The output from the j^{th} node in the hidden layer, $G(M_j)$ is given by

$$G(M_j) = \frac{1}{1+e^{-M_j}} \qquad j= 1,2,\ldots,17 \qquad 4.8$$

finally, the network output, $F^{(k)}$, is given by.

$$F^{(k)} = \sum_{j=1}^{18} \lambda^{(k)}{}_j G(M_j) \qquad k=2,3,4,5,6 \qquad 4.9$$

In the vertical plane there would be no out of plane forces and moment. The reason for that is the sub is symmetrical around the vertical axis and asymmetrical around the horizontal axis. The presence of the sail in the horizontal plane, see Mackay (1988), Mackay and Conway (1990 and 1991), while the sub is turning, produces a drift angle, which results in a positive heave force on the after body of the sub and a positive pitch moment at the nose which attempts to lift the nose up.

After some preliminary trials it has been found that a three-layer feedforward net with 15 hidden nodes would be sufficient in the case of predicting the forces in the arc maneuver. On the other hand more hidden processing elements, namely 17 nodes, were needed to obtain the relation in the case of chirp maneuvers.

5. RESULTS AND DISCUSSION

5.1.1. UNCOUPLED HEAVE MOTION GENERATION

The equation of uncoupled heave motion for the model can be written as follows.

$$\ddot{x}_{3t} + b_{33}\dot{x}_{3t} = F_{30}\sin(\omega t_i) \qquad 5.1$$

Using this equation, 20 time histories were generated at the frequencies: 0.5, 0.6, 0.7,..., 2.4 Hz. The heave force is plotted as a function of the displacement, velocity, and acceleration for different frequencies. The results are shown in Figure 5.1 to Figure 5.6

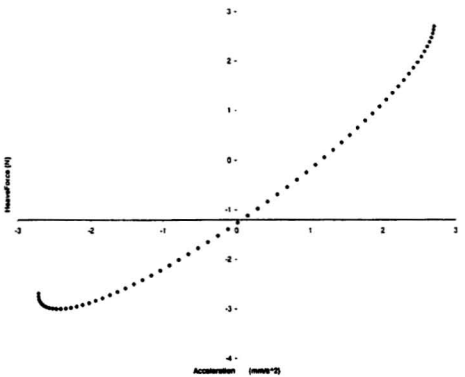


Figure 5.1 Hydrodynamic force vs. acceleration at 0.5 Hz uncoupled heave

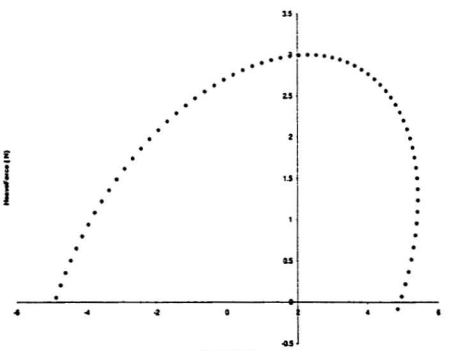


Figure 5.2 Hydrodynamic force vs. velocity at 0.5 Hz uncoupled heave

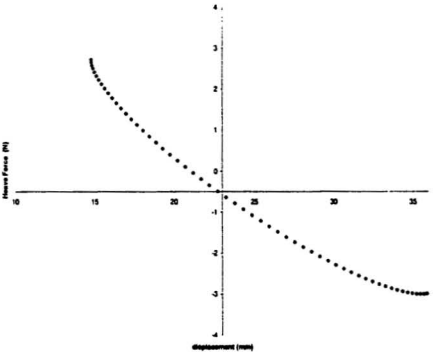


Figure 5.3 Hydrodynamic force vs. displacement at 0.5 Hz uncoupled heave

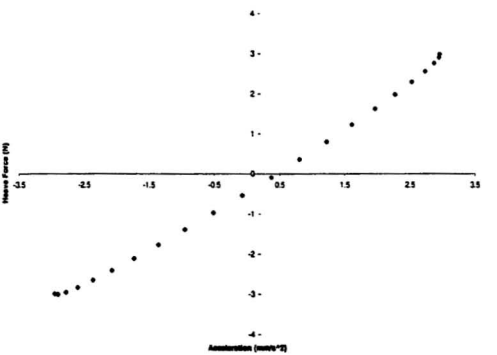


Figure 5.4 Hydrodynamic force vs. acceleration at 1.5 Hz uncoupled heave

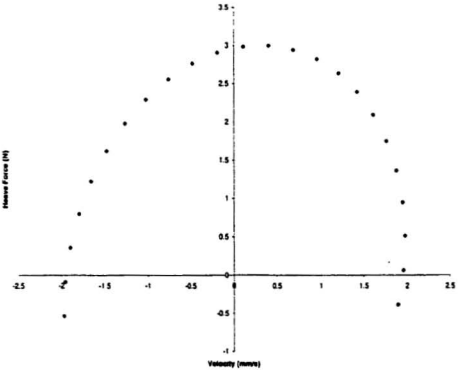


Figure 5.5 Hydrodynamic force vs. velocity at 1.5 Hz uncoupled heave

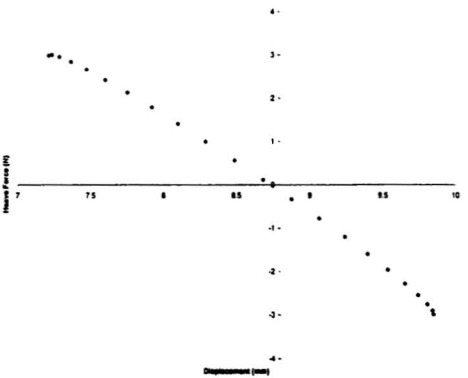


Figure 5.6 Hydrodynamic force vs. displacement at 1.5 Hz uncoupled heave

5.1.2. UNCOUPLED PITCH MOTION GENERATION

As in uncoupled heave, the equation of motion for a submarine model undergoing pitch motion can be written as follows

$$\ddot{x}_{5i} + b_{55}\dot{x}_{5i} + c_{55}x_{5i} = F_{5o}\sin(\omega t_i) \qquad 5.2$$

The model motions were generated at frequencies of 0.5, 0.6, 0.7,…, 2.4 Hz. The pitch moment is plotted as a function the motion. Plots are shown in Figure 5.7 Pitch moment vs. acceleration at 0.5 Hz uncoupled pitch to Figure 5.12.

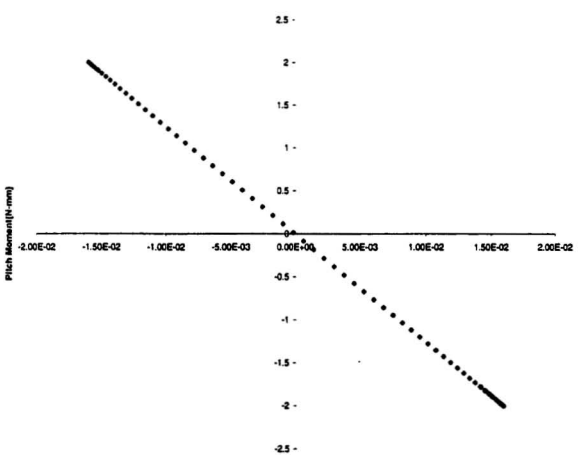


Figure 5.7 Pitch moment vs. acceleration at 0.5 Hz uncoupled pitch

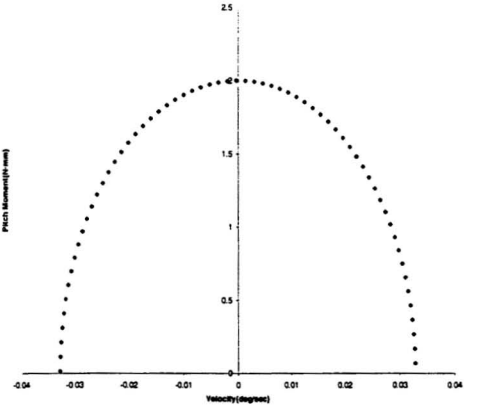


Figure 5.8 Pitch moment vs. velocity at 0.5 Hz uncoupled pitch

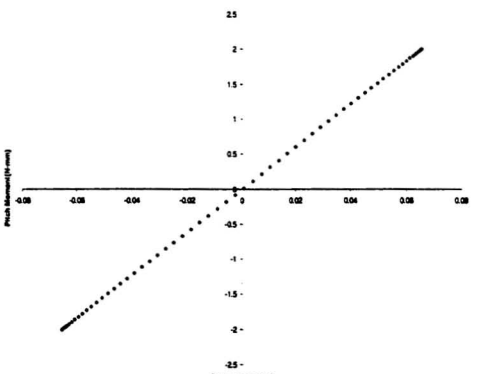


Figure 5.9 Pitch moment vs. displacement at 0.5 Hz uncoupled pitch

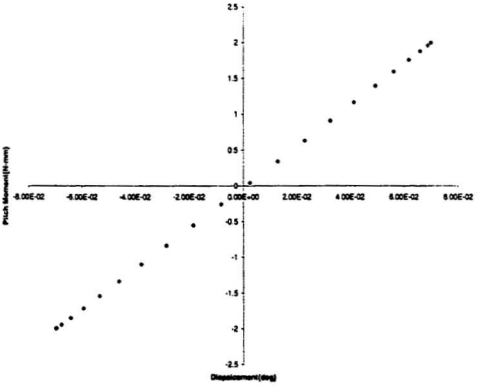


Figure 5.10 Pitch moment vs. acceleration at 1.5 Hz uncoupled pitch

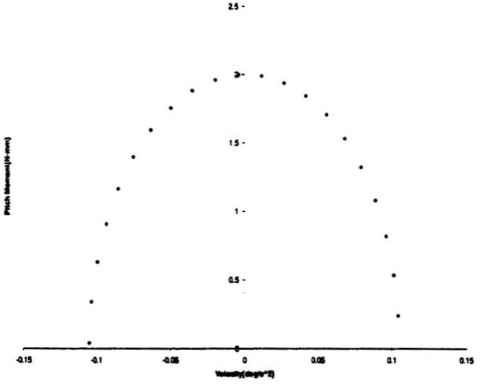


Figure 5.11 Pitch moment vs. velocity at 1.5 Hz uncoupled pitch

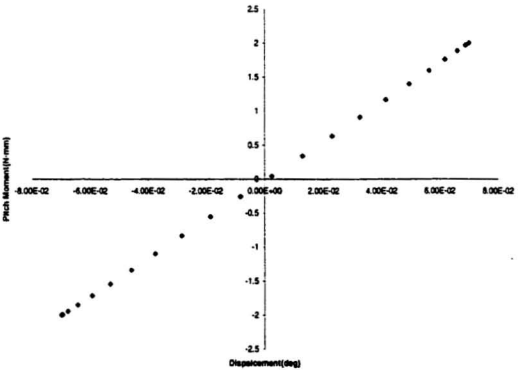


Figure 5.12 Pitch moment vs. displacement at 1.5 Hz uncoupled pitch

5.1.3. COUPLED HEAVE AND PITCH MOTION

Uncoupled heave and uncoupled pitch motions are theoretical motions. In this section the more realistic coupled motion will be considered. The motion for the same generic submarine model was generated undergoing regular coupled heave and pitch. The model motion is described by equations 3.2 and 3.3.

As in the previous sections these equations can be solved using a Runge-Kutta method. Simulated data were obtained for 0.5, 0.6,……., 2.0 Hz for the coupled motion. The displacement, velocity, and acceleration time series were obtained at these frequencies.

To validate the proposed technique the numerically generated force was compared to the predicted and least square estimated force.

5.1.4. EFFECTS OF RANDOM NOISE

Direct measured values, experimental values, often contain a noise component either from instrumental errors or mechanical noise from the experimental apparatus. In the following section the coupled heave pitch motion is generated as in the previous section but with the addition of a 20 % random noise term to the motion displacement, velocity and acceleration.

The motion of the submarine was modeled using equations 3.2 and 3.3. Applying the same procedures as before the numerically generated data were obtained for the same range of frequencies. The random noise was then added to the time series.

The reason for adding the noise after the time series has been generated is that the noise simulated here was due to instrumentation errors. Once again the generated hydrodynamic force and moment force were compared to the estimated force from the least squares method and the predicted one using a neural network.

5.2. RESULTS FROM DIGITALLY GENERATED DATA

Figure 5.13 and Figure 5.14 show a comparison between a least squares estimate and a neural network prediction for the heave force for the uncoupled heave case. Figure 5.15 shows a plot of actual force vs. the neural predicted force (uncoupled heave at 0.7 Hz) while Figure 5.20 is a plot of actual pitching moment vs. least squares estimated pitch moment (uncoupled pitch at 0.7 Hz). Figure 5.17 and Figure 5.18 show the comparison between a least squares estimate and a neural network prediction of the pitch moment for the uncoupled pitch motion. Figure 5.19 is a plot of actual pitching moment vs. a neural predicted pitch moment (uncoupled pitch at 0.7 Hz). Figure

5.20 is a plot of actual pitching moment vs. a least squares estimated pitch moment (uncoupled pitch at 0.7 Hz).

Figure 5.21 and Figure 5.22 show a comparison between a least squares estimate and a neural network prediction of the coupled heave-pitch motion. The effect of randomly generated noise on the least squares and the neural prediction is show in Figure 5.23 and Figure 5.24.

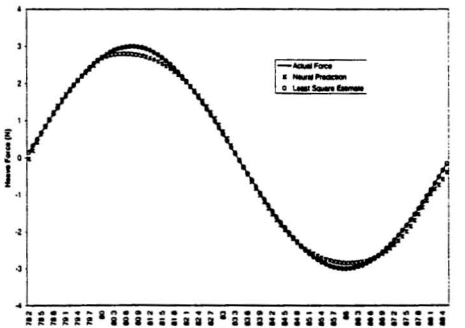


Figure 5.13 Comparison between least square estimation and neural net prediction (uncoupled heave at 0.6 Hz)

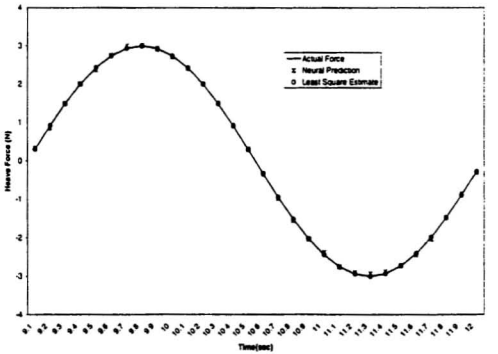


Figure 5.14 Comparison between least square estimate and neural net prediction (uncoupled heave at 1.5 Hz)

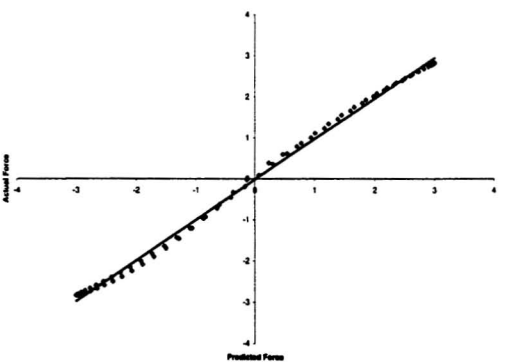


Figure 5.15 Plot of actual force vs. neural predicted force (uncoupled heave at 0.7 Hz)

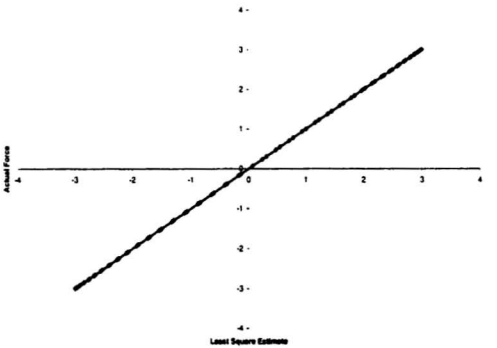


Figure 5.16 Plot of actual force vs. least squares estimated force (uncoupled heave at 0.7 Hz)

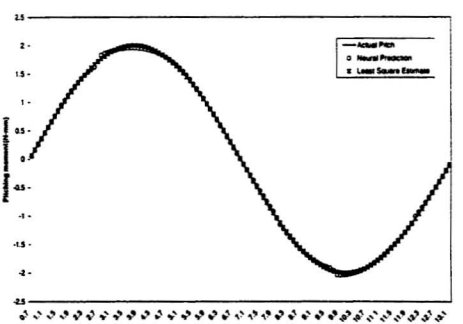


Figure 5.17 Comparison between least square estimation and neural network prediction (uncoupled pitch at 0.7 Hz)

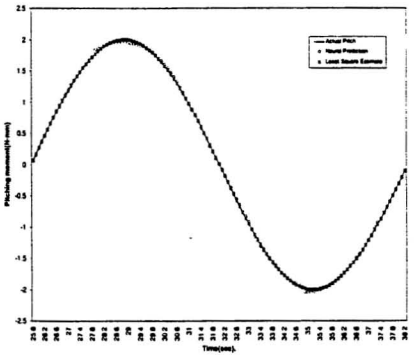


Figure 5.18 Comparison between least square estimation and neural network prediction (uncoupled pitch at 1.2 Hz)

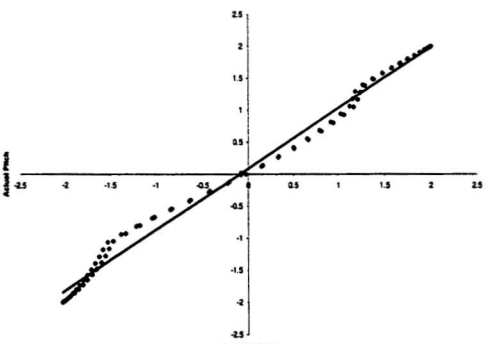


Figure 5.19 Plot of actual pitching moment vs. neural predicted pitch moment (uncoupled pitch at 0.7 Hz)

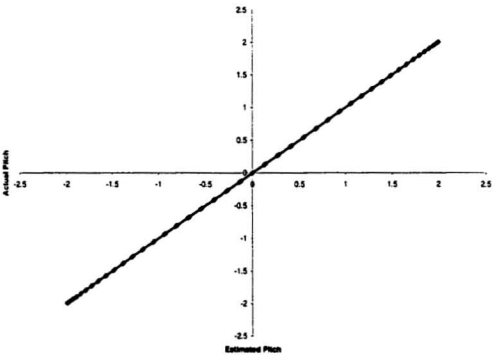


Figure 5.20 Plot of actual pitching moment vs. least squares estimated pitch moment (uncoupled pitch at 0.7 Hz)

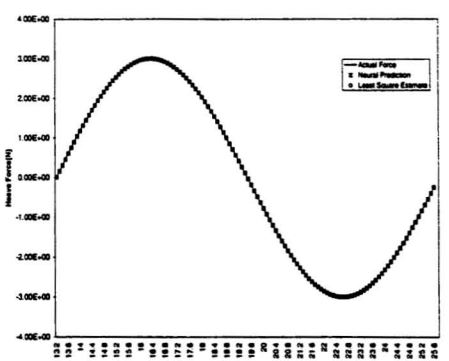


Figure 5.21 Comparison between least squares estimation and neural network prediction (coupled heave-pitch at 0.5 Hz).

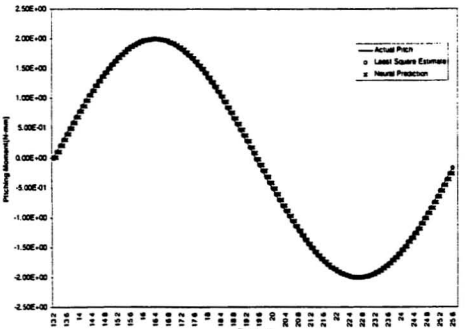


Figure 5.22 Comparison between least squares estimation and neural network prediction (coupled heave-pitch at 0.5 Hz)

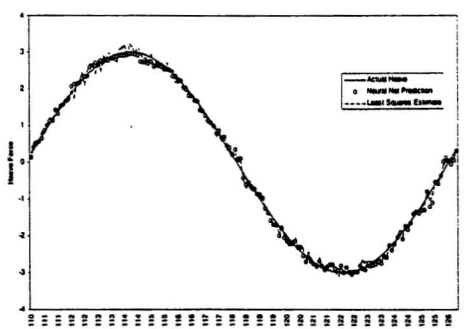


Figure 5.23 Comparison between least squares estimate and neural prediction (coupled heave and pitch at 0.4 Hz with noise)

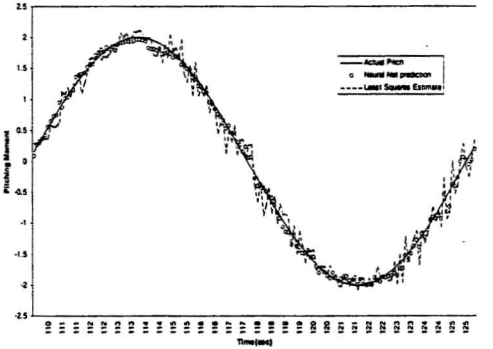


Figure 5.24 Comparison between least squares estimate and neural prediction (coupled heave and pitch at 0.4 Hz with noise)

5.3. RESULTS FROM EXPERIMENTAL DATA

A number of neural networks were trained to identify the hydrodynamic forces acting on the submarine model during different types of maneuvers. Two of the categories described in section 4.2 and the model in its full configuration were considered. The two maneuvers were the horizontal circular arc and sway chirp maneuver.

A single run from each maneuver was used in training. The number of data points used in the training process was enormous. 2501 x 4 data points were used in training the chirp networks and 1920 x 4 data points were used in training the circular arc networks. After a number of preliminary trials the author found that training the network in a modular architecture gives better results. Each module has three layers and 15 hidden nodes in the case of the arc maneuvers

and 17 hidden nodes in the chirp maneuvers. Figure 5.25 shows the proposed network architecture. The network consists of small modules each module was trained separately then the final net was then assembled.

Figure 5.26 to Figure 5.30 show plots of the hydrodynamic loads computed by the net vs. the actual component measured for the training data set. The deviation around the diagonal line represents the network error. Figure 5.31 to Figure 5.35 show the same results in a different way. In these set of plots the neural network training and the actual force were plotted vs. time. The network was trained with the horizontal circular arc $r' = 0.2$, and $\beta=0.0$ file name "HST_SV2000H_001.DAC; 1".

Figure 5.36 through Figure 5.45 show the same results described above for the trained network applied to a new data file. The run chosen to apply the net to was a horizontal circular arc $r' = 0.1$, and $\beta=0.0$ file name "HST_SVA1000H_001.DAC; 1".

The sway chirp network was trained using sway run " HST_F1t3V500S_001.DAC;1" where the sway frequency ranged from 0.1 to 0.3 Hz with strut velocity 500 mm/sec and sway amplitude of 1.85 m. Figure 5.46 to Figure 5.50 show plots of the neural training verses the actual hydrodynamic loads on the model while Figure 5.51 to Figure 5.55 show the same training results and actual hydrodynamic loads plotted in a time series format.

Figure 5.57 to Figure 5.73 show a comparison between results obtained by neural networks and multi-linear regression for the circular arc maneuvers.

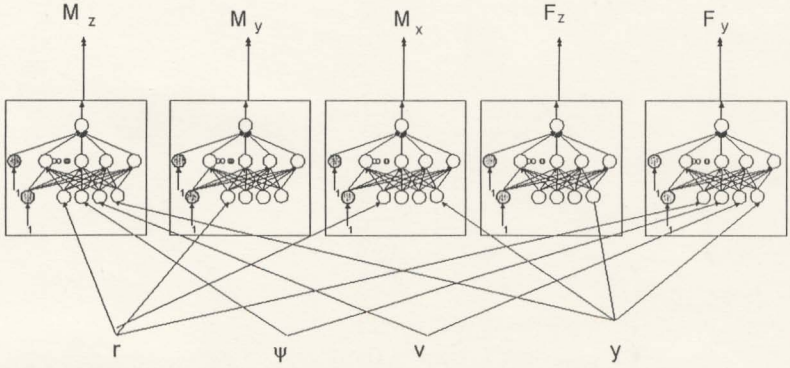


Figure 5.25 Modular neural network architecture

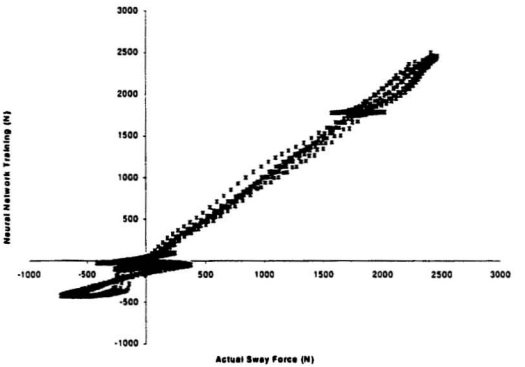


Figure 5.26 Plot of sway force vs. training (horizontal circular arc r' = 0.2)

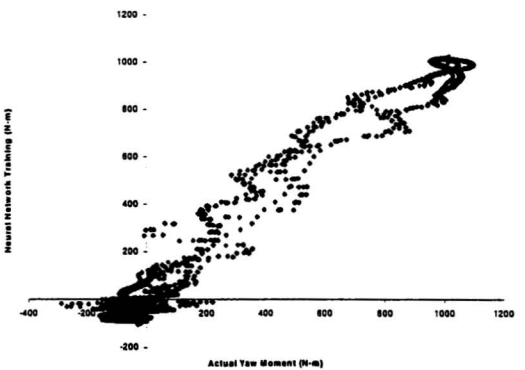


Figure 5.27 Plot of yaw moment vs. training (horizontal circular arc r' = 0.2)

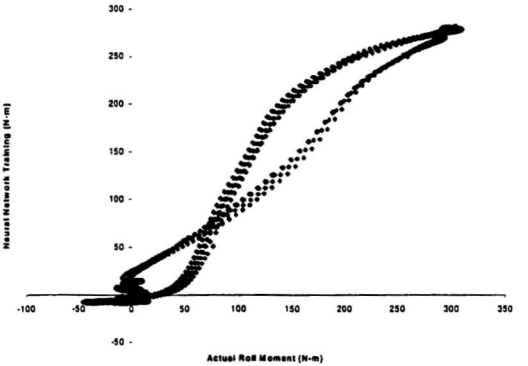


Figure 5.28 Plot of roll moment vs. training (horizontal circular arc r' = 0.2)

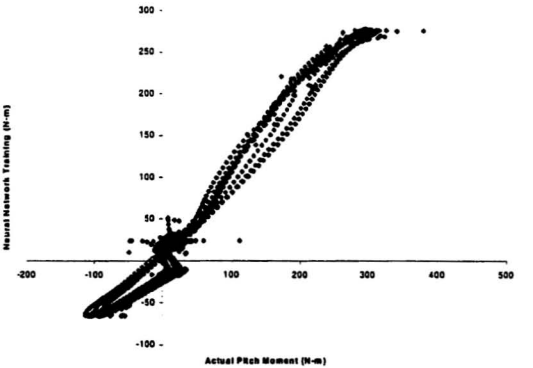


Figure 5.29 Plot of pitch moment vs. training (horizontal circular arc r' = 0.2)

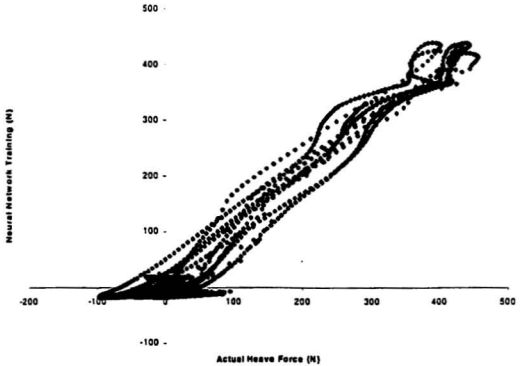


Figure 5.30 Plot of heave force vs. training (horizontal circular arc r' = 0.2)

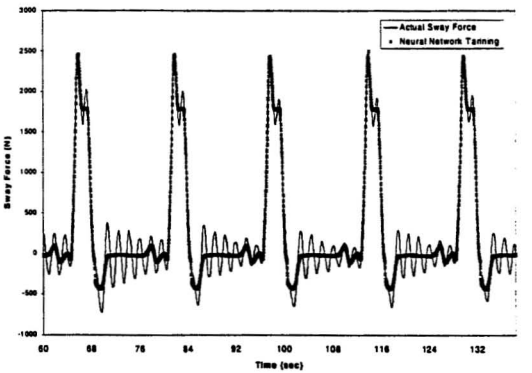


Figure 5.31 Neural network training of the sway force (horizontal circular arc r' = 0.2)

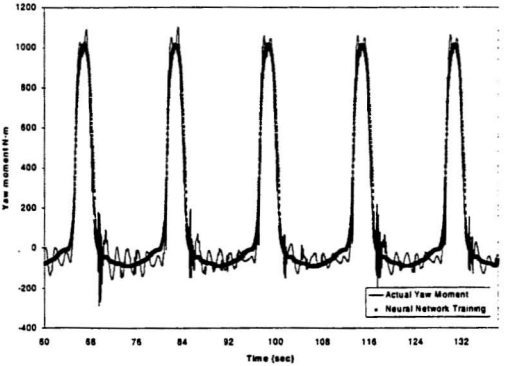


Figure 5.32 Neural network training of the yaw moment (horizontal circular arc r' = 0.2)

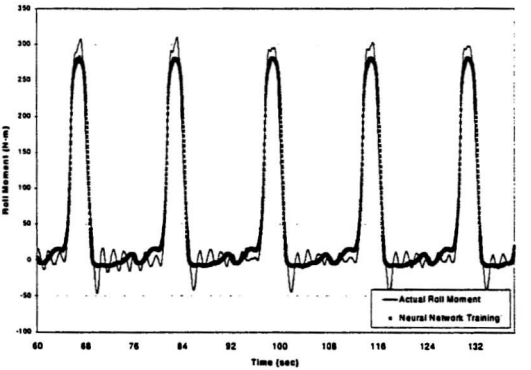


Figure 5.33 Neural network training of the roll moment (horizontal circular arc r' = 0.2)

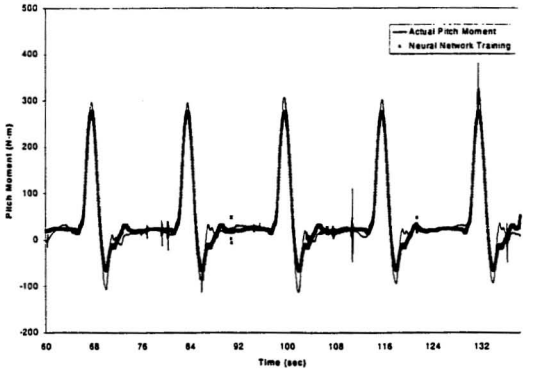


Figure 5.34 Neural network training of the pitch moment (horizontal circular arc r' = 0.2)

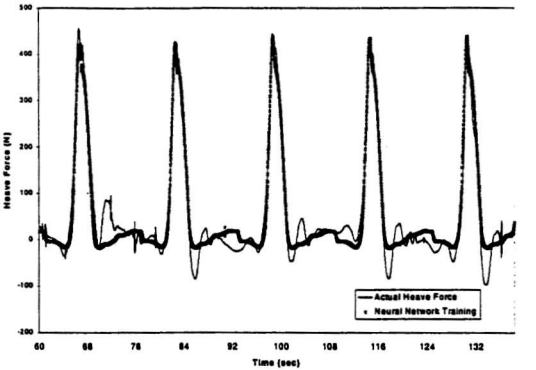


Figure 5.35 Neural network training of the heave force (horizontal circular arc r' = 0.2)

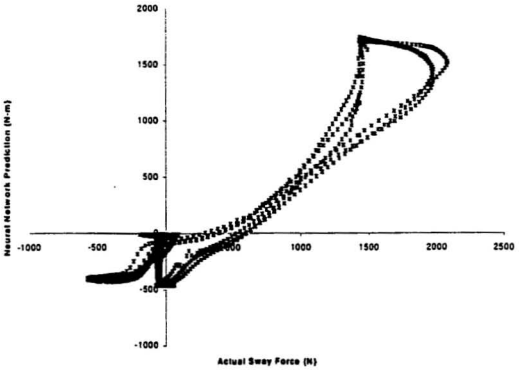


Figure 5.36 Plot of sway force vs. prediction (horizontal circular arc r' = 0.1)

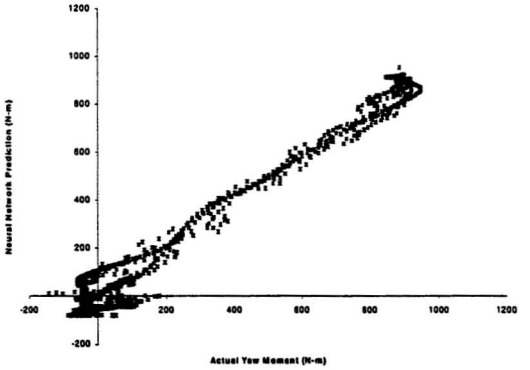


Figure 5.37 Plot of yaw moment vs. prediction (horizontal circular arc r' = 0.1)

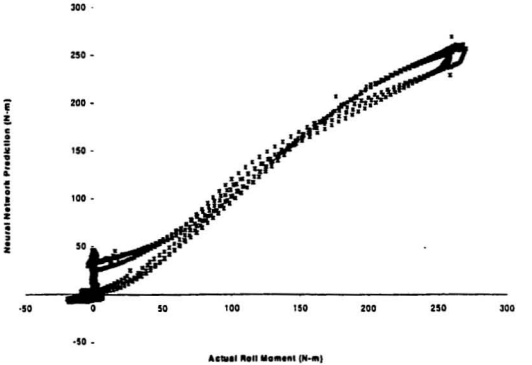


Figure 5.38 Plot of roll moment vs. prediction (horizontal circular arc r' = 0.1)

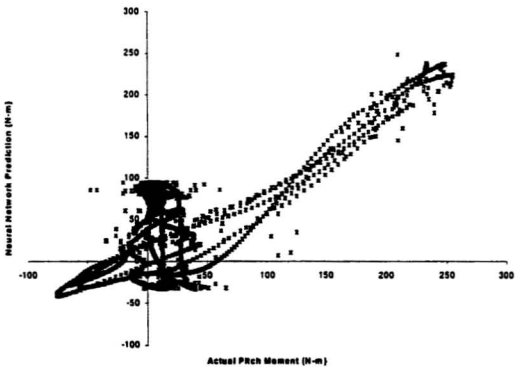


Figure 5.39 Plot of pitch moment vs. prediction (horizontal circular arc r' = 0.1)

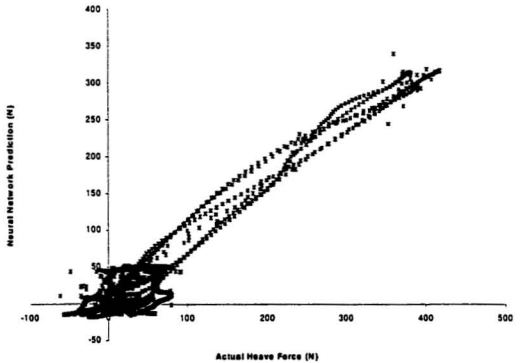


Figure 5.40 Plot of heave force vs. prediction (horizontal circular arc r' = 0.1)

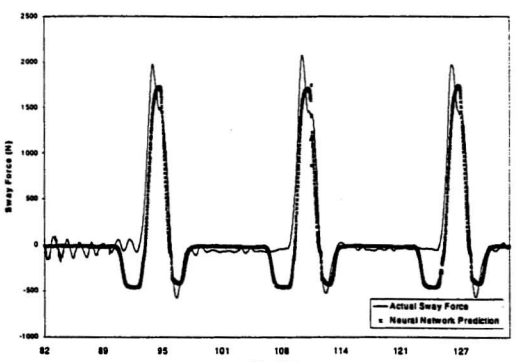


Figure 5.41 Neural network prediction of the sway force (horizontal circular arc r' = 0.1)

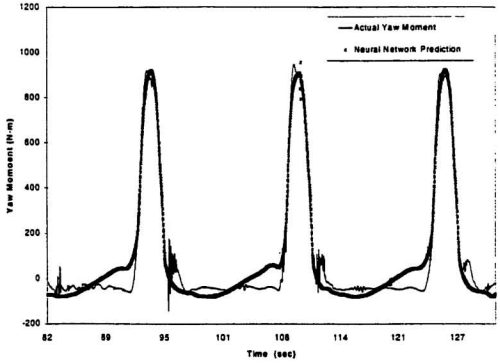


Figure 5.42 Neural network prediction of the yaw moment (horizontal circular arc r' = 0.1)

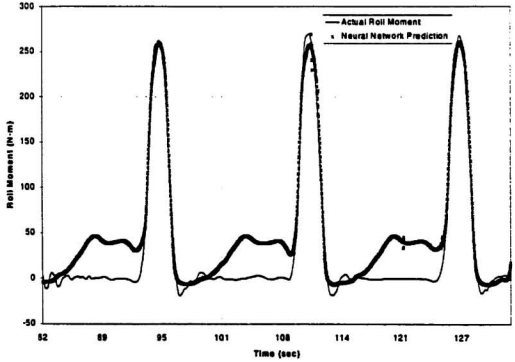


Figure 5.43 Neural network prediction of the roll moment (horizontal circular arc r' = 0.1)

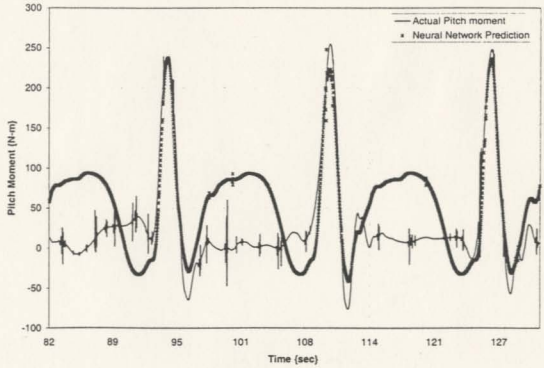


Figure 5.44 Neural network prediction of the pitch moment (horizontal circular arc r' = 0.1)

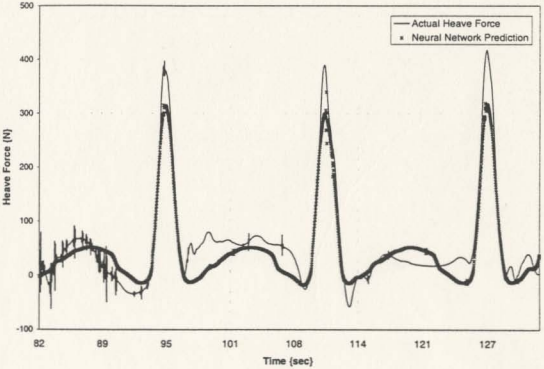


Figure 5.45 Neural network prediction of the heave force (horizontal circular arc r' = 0.1)

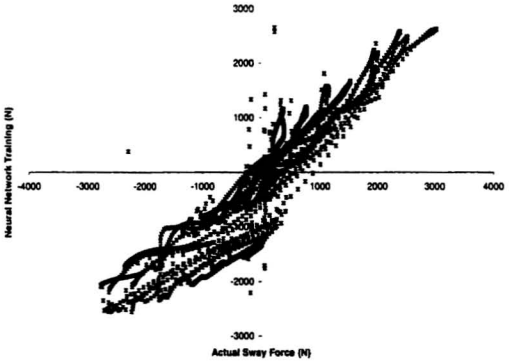


Figure 5.46 Plot of sway force vs. training (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

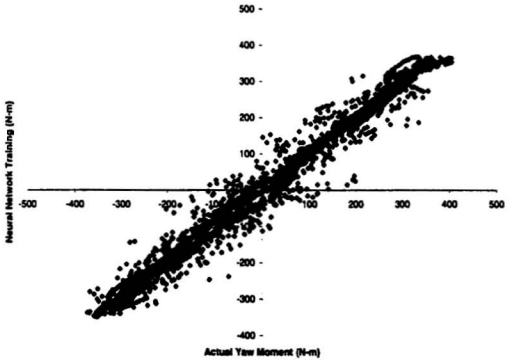


Figure 5.47 Plot of yaw moment vs. training (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

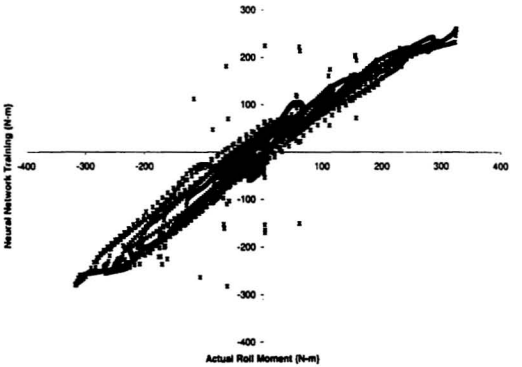


Figure 5.48 Plot of roll moment vs. training (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

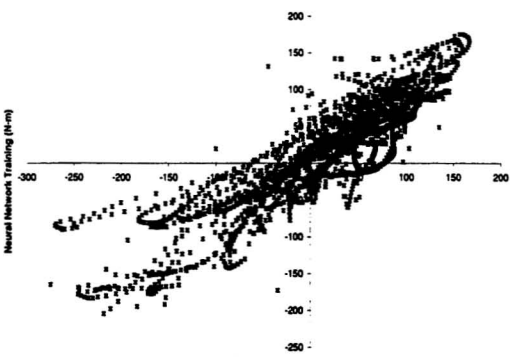


Figure 5.49 Plot of pitch moment vs. training (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

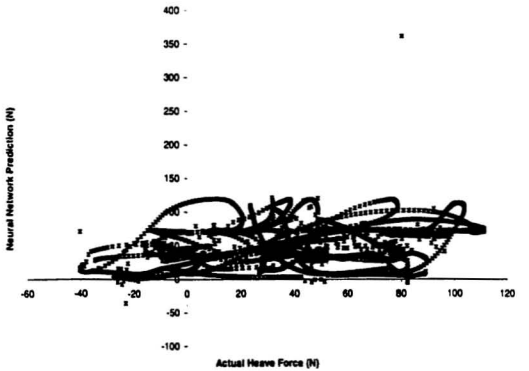


Figure 5.50 Plot of heave force vs. training (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

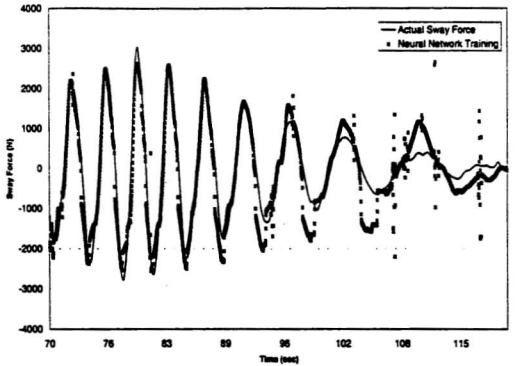


Figure 5.51 Neural network training of the sway force (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

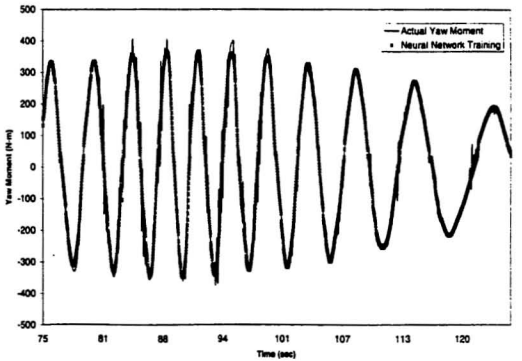


Figure 5.52 Neural network training of the yaw moment (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

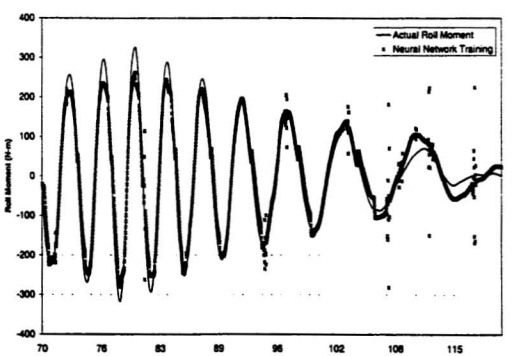


Figure 5.53 Neural network training of the roll moment (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

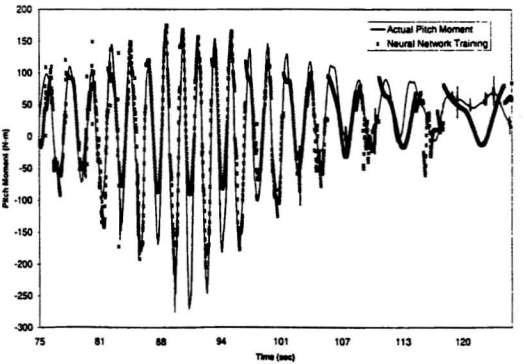


Figure 5.54 Neural network training of the pitch moment (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

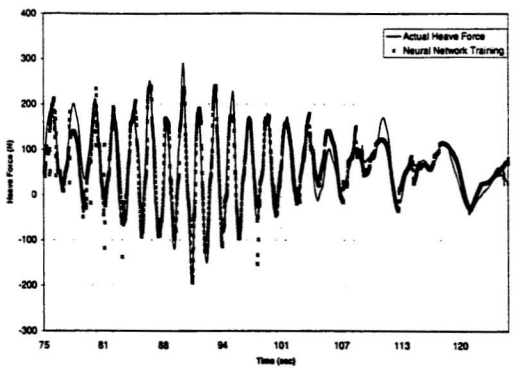


Figure 5.55 Neural network training of the heave force (sway chirp F= 0.1 to 0.3 Hz, V= 500 mm/s)

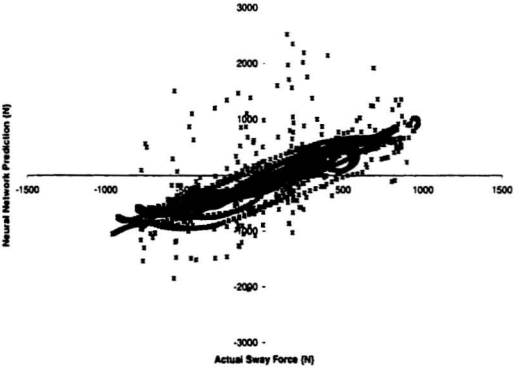


Figure 5.56 Plot of sway force vs. prediction (sway chirp F= 0.1 to 0.3 Hz, V= 250 mm/s)

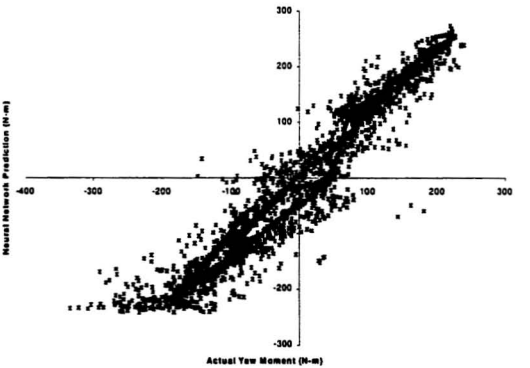


Figure 5.57 Plot of yaw moment vs. prediction (sway chirp F= 0.1 to 0.3 Hz, V= 250 mm/s)

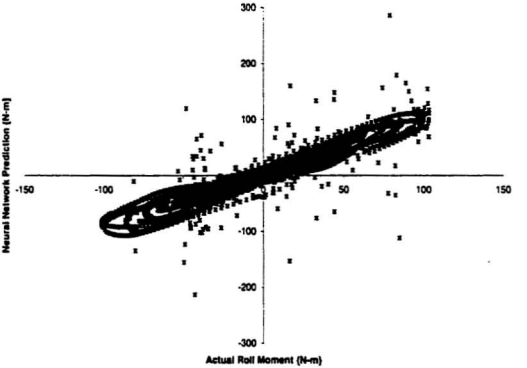


Figure 5.58 Plot of roll moment vs. prediction (sway chirp F= 0.1 to 0.3 Hz, V= 250 mm/s)

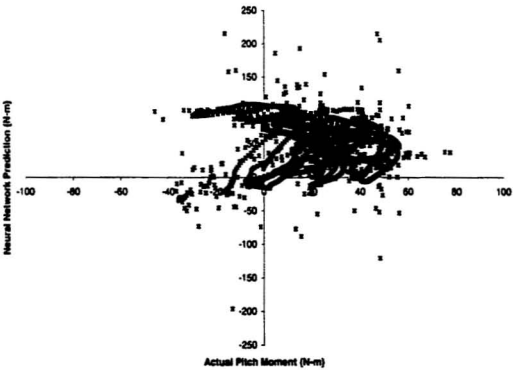


Figure 5.59 Plot of pitch moment vs. prediction (sway chirp F= 0.1 to 0.3 Hz, V= 250 mm/s)

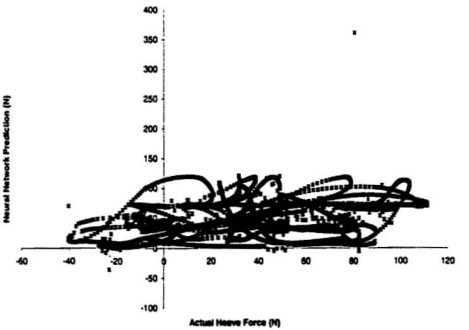


Figure 5.60 Plot of heave force vs. prediction (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

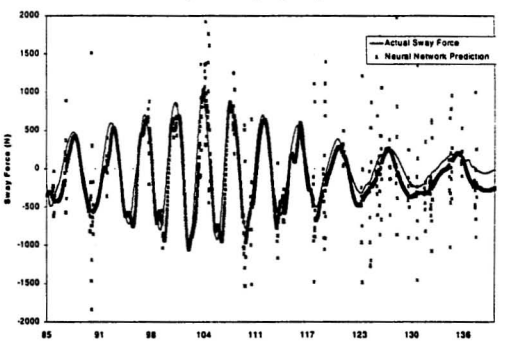


Figure 5.61 Neural network prediction of the sway force (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

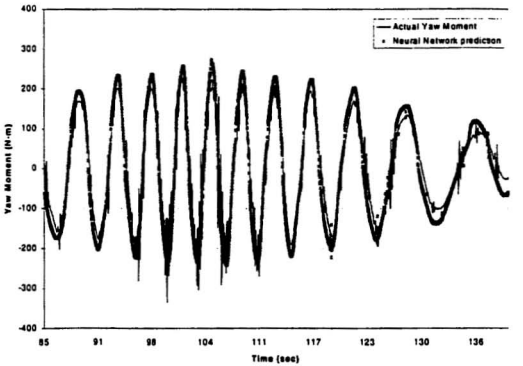


Figure 5.62 Neural network prediction of the yaw moment (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

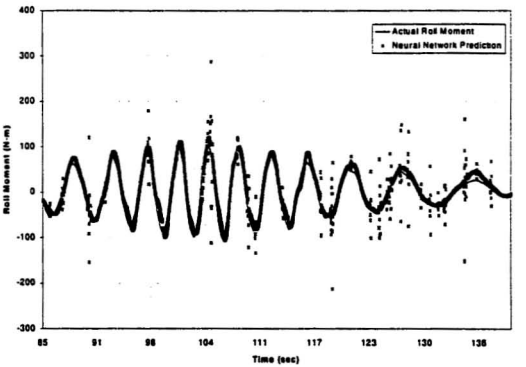


Figure 5.63 Neural network prediction of the roll moment (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

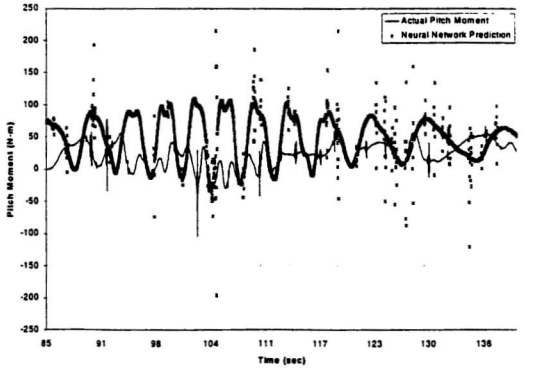


Figure 5.64 Neural network prediction of the pitch moment (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

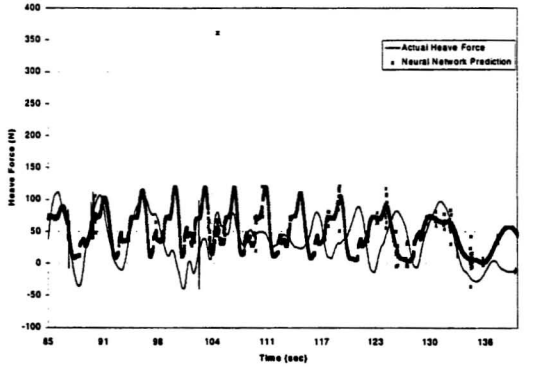


Figure 5.65 Neural network prediction of the heave force (sway chirp F= 0.1 to 0.3 Hz V= 250 mm/s)

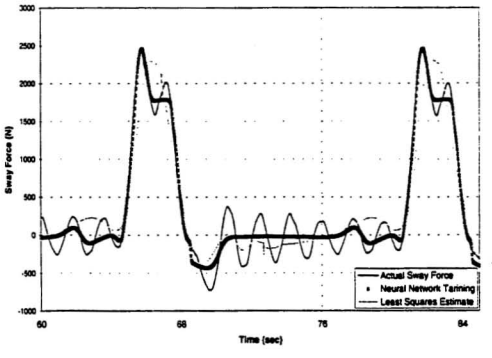


Figure 5. 66 Comparison between neural network training and least squares estimate of the sway force (horizontal circular arc r' = 0.2)

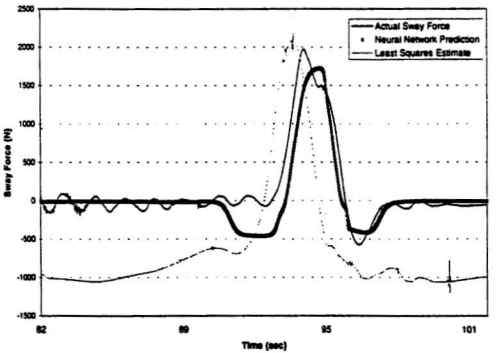


Figure 5. 67 Comparison between neural network prediction and least squares estimate of the sway force (horizontal circular arc r' = 0.1)

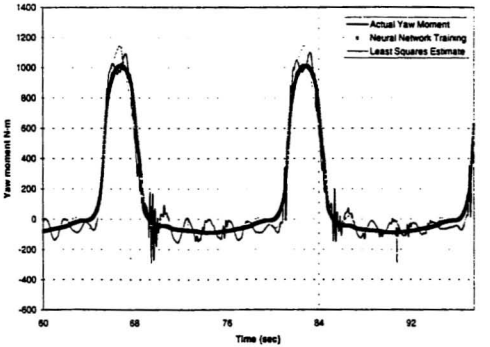


Figure 5. 68 Comparison between neural network training and least squares estimate of the yaw moment (horizontal circular arc r' = 0.2)

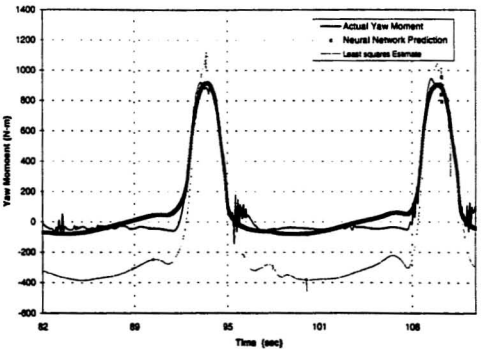


Figure 5. 69 Comparison between neural network prediction and least squares estimate of the yaw moment (horizontal circular arc r' = 0.1)

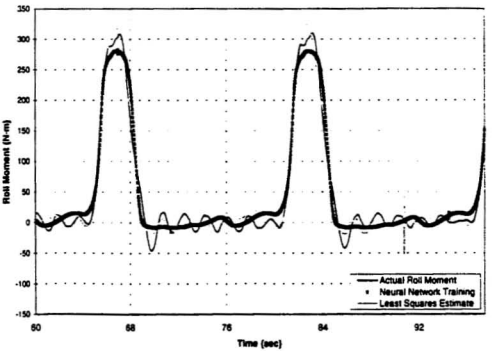


Figure 5. 70 Comparison between neural network training and least squares estimate of the roll moment (horizontal circular arc r' = 0.2)

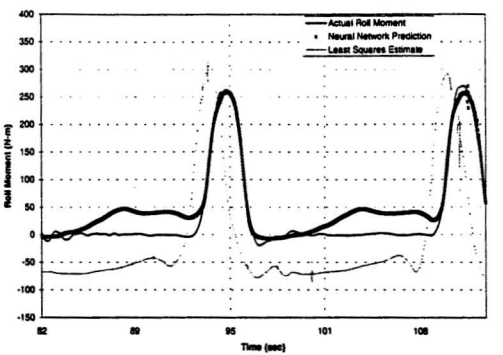


Figure 5. 71 Comparison between neural network prediction and least squares estimate of the roll moment (horizontal circular arc r' = 0.1)

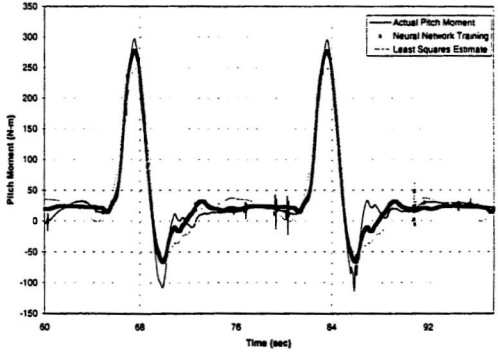


Figure 5. 72 Comparison between neural network training and least squares estimate of the pitch moment (horizontal circular arc r' = 0.2)

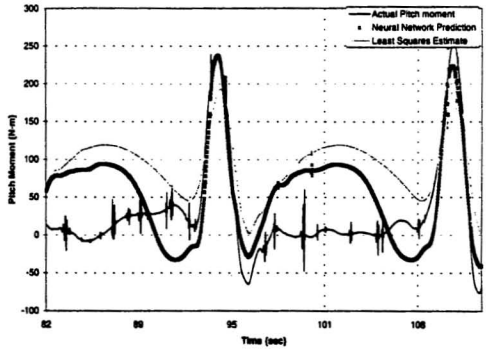


Figure 5. 73 Comparison between neural network prediction and least squares estimate of the pitch moment (horizontal circular arc r' = 0.1)

5.4. Discussion

The horizontal arc network consisted of five modules. Table 5.1 shows the training criteria for the network.

Module	Heave	Sway	Yaw	Roll	Pitch
Training events	240768	403788	87780	135432	393756
Training epochs	95	160	34	53	156
Training time, min.	89	154	132	50	148

Table 5. 1 Training criteria for horizontal circular arc network

From Figures 5.31 and 5.41 it can been seen that the neural network prediction for the sway force work fairly well. Moreover the network prediction from the steady state segment of the arc is shown to be of a constant value which agrees with the fact that the model is turning with a constant yaw rate. On the other hand the fluctuation on the actual (measured by MDTF) force is due to model vibration and rocking. The same discussion is valid for the rest of the hydrodynamics components. In the case of out of plane force and moment, Figures 5.43, 5.44 and 5.45, the network slightly over-predicts the steady state force.

The sway chirp maneuvers network has the same architecture as the horizontal arc network. The training criteria for the sway network are shown in Table 5.2. The prediction of the " in plane " components was good but once again the network slightly over-predicts the out-of-plane components. It is notable to say at this point that the-out-of plane components were very small in this case since it is a pure sway maneuver. The author feels that the presence of these forces, heave and pitch, in the sway chirp maneuver was mainly due to the flexibility of the sting and resulting model vibration. A

closer look at the strut displacements showed that there was a 0.2 to 0.5 degree yaw angle which might have been enlarged by the sting flexibility and resulted in the shown forces. Changing the architecture of the net reduced the training time significantly. Moreover the net can be disassembled and one of the modules can be retrained and assembled into the net. This gives a great advantage to this architecture.

Module	Heave	Sway	Yaw	Roll	Pitch
Training events	14385189	5622810	5802900	2187093	11515755
Training epochs	7188	2809	2899	1092	5754
Training time, min.	104	40	42	19	87

Table 5. 2 Training criteria for sway chirp network

Neuroshell® 2 , a commercial software available from Ward Inc., was used in building the modules. The network was then exported to Visual Basic code and a Windows® version of the net was assembled together. A C ** code can also be exported to be included with some other code that can be run in any environment (i.e. Unix, VAX, Windows).

6. CONCLUSIONS

In this work a new tool for predicting the hydrodynamic model for a submarine was presented. A neural network technique was used to predict the hydrodynamic forces acting on the submarine. The approach was validated using digitally generated data for coupled heave and pitch motions as well as experimental, see Figures 5.12 to Figure 5.24, data for circular arc maneuvers and sway chirp maneuvers, Figures 5.31 to Figure 5.65. The motion time series was numerically generated and both neural network predictions and least squares estimates were obtained and compared to the actual force. Results indicate that both the least squares method and the neural technique can provide accurate predictions for the hydrodynamic forces and moments as long as the data are not contaminated with noise. The neural network approach is shown to be superior to the least squares method when there is noise in the data. The network maximum error when applied to the noisy data was 2.67 % while the least squares produced 8.067 % error.

In section 5.3, the tool was used to predict the hydrodynamics of a submarine model undergoing certain maneuvers. The maneuvers considered were horizontal arc maneuvers and sway chirp maneuvers. A single run from each maneuver was used to train the network. The fully trained networks were then used to predict the hydrodynamic forces acting on the model at different conditions. For both maneuvers the force and moment

prediction agree fairly well with the original measurement although there were some cases where the network over predicted the force. During the experimental tests visual observation showed that there were some vibrations in the model as it followed the desired trajectory and in some cases the vibration exceeded the MDTF limitation and the maneuvers were terminated by the control loop. We feel that this vibration affected the measured forces and since the motion history used for training the network were taken from the MDTF struts motion, the relation between the force and the motion changed to some extent. Filtering was used to eliminate the effect of noise on the data.

A comparison between the neural network and the least squares estimate is also shown in Section 5.3. The comparison showed that the neural network approach is more superior than the classical technique.

Vertical arc maneuvers and vertical chirp maneuvers were limited in number and they were not considered within this study.

The combination between the proposed tool and the MDTF represents a new and reliable way to investigate the hydrodynamics of a submarine. With the new technique only a limited number of maneuvers have to actually be performed to train the network. The network can then be used to generate the force and moment for the remaining needed maneuvers. By doing this we combined the capabilities of multi-degrees-of-freedom testing with the affordability and convenience of computational techniques.

7. RECOMMENDATIONS

Future work should include maneuvers that were not considered within this study due to the lack of time and the limited number of tests conducted. Maneuvers such as vertical arcs, combined chirps, and five-degree-of freedom maneuvers such as cork-screw maneuvers should also be investigated.

As stated in the conclusions due to model vibration the use of the rig struts displacements introduces some uncertainty to the relation obtained by the network. Further work should use the estimate of the actual motion obtained from the onboard accelerometers.

REFERENCES

Bhattacharyya, R., 1978, "Dynamics of Marine Vehicles", John Wiley & Sons. New York, NY.

Bohlmann, H., 1990, "Calculation of hydrodynamic coefficients of submarines for the prediction of motion patterns" Ph.D. Thesis, Institute of Naval Architecture University of Hamburg, Hamburg, Germany.

Feldman, J., 1995, " Method of performing captive model experiments to predict the stability and the control characteristics of submarines", CRDKNSWC-HD-0393-25.

Haddara, M. R., 1995, "On the use of neural network techniques for the identification of ship stability parameters at sea", Proceedings of the 14th International Conference on Offshore Mechanics and Arctic Engineering, vol. II, pp.127-135.

Haddara, M. R., and Hinchey, M., 1995, "On the use of neural network techniques in the analysis of free roll decay curves", International Shipbuilding Progress, vol. 42, no. 430, pp. 166-178.

Haddara, M. R., and Wang, Y. 1996, "Parametric identification of coupled sway and yaw motions". Proceedings of the 15th International Conference on Offshore Mechanics and Arctic Engineering, vol. I, pp. 267-273

Haddara, M. R, and Xu, J., 1999, "On the identification of ship coupled heave-pitch motions using neural networks", Ocean Engineering Journal, no. 26, pp. 381-400.

Hinchey, M., 1994, "A neural network fit to icebreaker resistance data", Proceedings of the 12th International Conference on Offshore Mechanics and Arctic Engineering, vol. I, pp. 263-266.

Kalske, S., 1992, "Motion simulation of underwater vehicles", Technical Research Center of Finland (VTT) , VTT publication 97, ESPOO 1992.

Lainiotis, D., and Plataniotis, K., 1994, "Neural network estimators: application to ship position estimation", Proceedings of IEEE conference, pp. 4710-4717.

Math Works Inc., 1994, "MATLAB High performance numeric computation and visualization software", User's guide.

Math Works Inc., 1997, "Wavelet toolbox for the use with MATLAB ", User's guide.

Mackay, M., 1988, " The prediction of submarine out of plane force and moment using panel method", RINA Warship 88 Symposium on Conventional Naval Submarines, London.

Mackay, M., Conway, J., 1990, " Prediction of the effects of body separation vortices on submarine configurations using the CANERO panel method", Proceedings of AIAA 28th Aerospace Science Meeting, AIAA 90-0302.

Mackay, M, Conway, J., 1991, "Modeling the cross flow body separation on a submarine using panel method", Warship 91 Symposium on Conventional Naval Submarines, London

Mehrotra, K., Mohan, C., and Ranka, S., 1997, "Elements of Artificial Neural Networks", MIT Press, Cambridge, MA.

Montgomery, D., 1997, " Design and Analysis of Experiments", John Wiley & Sons. New York, NY.

Papoulias, A., and Mckinley, D., 1994, " Inverted pendulum stabilization of submarines in free positive buoyancy ascent", Journal of Ship Research, Vol. 38, No. 1, pp. 71-82.

Papoulias, A., and Papadimitriou, H., 1995, " Nonlinear studies of dynamic stability of submarines in the dive plane", Journal of Ship Research, Vol. 39, No. 4, pp. 347-356.

Rivera, C., and Hinchey, M., 1999, "Hydrodynamic loads on subsea robots", Ocean Engineering Journal, no. 26, pp. 805-812.

Rumelhart, D., Hinton, G., Williams, R., 1986, "Learning internal representation by error propagation", Parallel distributed processing: Exploration in the microstructure of cognition. vol. 1: Foundation, MIT Press.

Ward System Group, Inc. 1995," Neural Shell 2: User Manual".

Watt, G., 1988, " Estimates for the added mass of a multi-component, deeply submerged vehicle", Defence Research Establishment Atlantic, Technical Memorandum 88/213, Dartmouth, NS.

Watt, G., Nguyen, V., Cooper, K., and Tanguay, B., 1993, "Wind tunnel investigations of submarine hydrodynamics", Canadian Aerodynamics and Space Journal, vol. 39, no. 3, pp. 119-126.

Widrow, B., and Lehr, M., 1990, "30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation", Proceeding. IEEE, vol. 78, no. 9, pp. 1415-1442.

Williams, C., Mackay, M., Perron, C., and Muselet, C., 1999, " The NRC-IMD Marine Dynamic Test Facility: a six-degree-of-freedom forced motion apparatus for underwater vehicle testing", NRCC/IMD report IR-1999-28, St. John's, NF.

APPENDIX A NUMERICAL RESULTS

The following are the calculations for the multiple linear regression in the case of uncoupled heave, uncoupled pitch and coupled heave and pitch as well as the effect of random noise.

Uncoupled Heave.

The mathematical model for the heave force can be expressed as

$$F_i = \beta_0 + \beta_1 \dot{x}_i + \beta_2 \ddot{x}_i \qquad \text{A.1}$$

Using the least squares method one can obtain the coefficients in the equation as follows:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 229 & 6.4 & -1.5 \\ 6.4 & 2960.5 & 7 \\ -1.5 & 7 & 871.3 \end{bmatrix} \qquad \text{A.2}$$

$$\mathbf{X}^T\mathbf{F} = \begin{bmatrix} 0.0215 \\ 699.791 \\ 872.975 \end{bmatrix} \qquad \text{A.3}$$

$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} 0.0044 & 0 & 0 \\ 0 & 0.0003 & 0 \\ 0 & 0 & 0.00011 \end{bmatrix} \qquad \text{A.4}$$

$$\hat{\beta} = \begin{bmatrix} 0.0044 & 0 & 0 \\ 0 & 0.0003 & 0 \\ 0 & 0 & 0.00011 \end{bmatrix} \begin{bmatrix} 0.0215 \\ 699.791 \\ 872.975 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.234 \\ 1.0 \end{bmatrix} \quad \text{A.5}$$

so the expected heave force would be

$$\hat{F}_i = 0.0 + 0.234\dot{x}_i + 1.0\ddot{x}_i \quad \text{A.6}$$

the actual Heave force was

$$F_i = 0.0 + 0.234\dot{x}_i + 1.0\ddot{x}_i$$

Uncoupled Pitch

Using the least squares method the pitch moment is assumed to follow the mathematical model shown in equation A.7

$$F_i = \beta_0 + \beta_1 x_i + \beta_2 \dot{x}_i + \beta_3 \ddot{x}_i \quad \text{A.7}$$

The coefficients are estimated as follows

$$\mathbf{X^T X} = \begin{bmatrix} 586 & 0.0014 & -0.1 & -0.0007 \\ 0.0014 & 1.291 & 0.001 & -0.761 \\ -0.1 & 0.001 & 0.776 & 0.0001 \\ -0.0007 & -0.761 & 0.0001 & 0.569 \end{bmatrix} \quad \text{A.8}$$

$$\mathbf{X^T F} = \begin{bmatrix} 0.0109 \\ 38.9769 \\ 0.1756 \\ -22.8582 \end{bmatrix} \quad \text{A.9}$$

$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} 0.0017 & 0 & 0.0002 & 0 \\ 0 & 3.6537 & -0.0011 & 4.8836 \\ 0.0002 & -0.0011 & 1.2887 & -0.0016 \\ 0 & 4.8836 & -0.0016 & 8.2836 \end{bmatrix} \quad \text{A.10}$$

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} 0.0017 & 0 & 0.0002 & 0 \\ 0 & 3.6537 & -0.0011 & 4.8836 \\ 0.0002 & -0.0011 & 1.2887 & -0.0016 \\ 0 & 4.8836 & -0.0016 & 8.2836 \end{bmatrix} \begin{bmatrix} 0.0109 \\ 38.9769 \\ 0.1756 \\ -22.8582 \end{bmatrix} = \quad \text{A.11}$$

$$\begin{bmatrix} 0 \\ 30.7804 \\ 0.2222 \\ 1.0005 \end{bmatrix}$$

So the expected pitch moment would be

$$\hat{F}_i = 0.00 + 30.7804\, x_i + 0.2222\, \dot{x}_i + 1.0005\, \ddot{x}_i \quad \text{A.12}$$

the actual pitch moment is

$$F_i = 0.00 + 30.78\, x_i + 0.222\, \dot{x}_i + 1.0\, \ddot{x}_i$$

Coupled Heave-Pitch

Applying the least squares method one can obtain the following

$$\mathbf{X}_3^T\mathbf{X}_3 = 10^3 * \begin{bmatrix} 0.465 & 0.0137 & -0.0033 & -0.0033 & -0.0001 \\ 0.0137 & 4.5464 & 0.0136 & -0.0633 & -0.0407 \\ -0.0033 & 0.0136 & 1.8516 & 0.0408 & -0.0261 \\ -0.0033 & -0.0633 & 0.0408 & 0.0018 & 0 \\ -0.0001 & -0.0407 & -0.0261 & 0 & 0.0008 \end{bmatrix} \quad \text{A.13}$$

$$\mathbf{X_3}^T\mathbf{X_3} = 10^3 * \begin{bmatrix} 0.466 & 0.0106 & -0.0002 & -0.0001 & 0.0001 \\ 0.0106 & 4.5558 & -0.0635 & -0.0409 & -0.0259 \\ -0.0002 & -0.0635 & 0.0018 & 0.0 & -0.0008 \\ -0.0001 & -0.0409 & 0.0 & 0.0008 & 0 \\ 0.0001 & 0.0259 & -0.0008 & 0 & 0.0004 \end{bmatrix} \quad \text{A.14}$$

$$\mathbf{X_3}^T\mathbf{F_3} = 10^3 * \begin{bmatrix} -0.0002 \\ 1.0559 \\ 1.8604 \\ 0.0265 \\ -0.0355 \end{bmatrix} \quad \text{A.15}$$

$$\mathbf{X_5}^T\mathbf{F_5} = \begin{bmatrix} 0.0091 \\ 703.506 \\ 17.643 \\ -23.6649 \\ -8.8516 \end{bmatrix} \quad \text{A.16}$$

$$\hat{\boldsymbol{\beta}}_3 = \begin{bmatrix} 0.0 \\ 0.2340 \\ 1.0001 \\ 0.2355 \\ 0.1581 \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_5 = \begin{bmatrix} 0.0 \\ 0.5801 \\ 30.7843 \\ 0.2247 \\ 1.0044 \end{bmatrix} \quad 1.17$$

So the expected heave force and pitch moment would be

$$\hat{F}_{3i} = 0.0 + 0.234\dot{x}_{3i} + 1.0001\ddot{x}_{3i} + 0.2355x_{5i} + 0.1581\dot{x}_{5i} \quad \text{A.18}$$

$$\hat{F}_{5i} = 0.0 + 0.5801\dot{x}_{3i} + 30.7843x_{5i} + 0.2247\dot{x}_{5i} + 1.0044\ddot{x}_{5i} \quad \text{A.19}$$

the actual heave and pitch moment were

$$F_{3i} = 0.0 + 0.234\dot{x}_{3i} + 1.0001\ddot{x}_{3i} + 0.238x_{5i} + 0.1581\dot{x}_{5i}$$

$$F_{5i} = 0.0 + 0.58\dot{x}_{3i} + 30.78x_{5i} + 0.222\dot{x}_{5i} + 1.0\ddot{x}_{5i}$$

APPENDIX B VISUAL BASIC PROGRAM

"SIMULATION"

The following Visual Basic program was written to generate the sub motion time series used in the validation of the method.

```
Dim delta As Double, omega As Double, k(4) As Double, t As Double
Dim L(4) As Double, a3(4000), a5(4000), M(4) As Double, N(4) As Double, v3(4000)
As Double, v5(4000) As Double
Dim x3(4000) As Double, x5(4000) As Double, F1(4000) As Double, F2(4000) As
Double
Dim B33, c33, B35, c35, B53, c53, B55, c55
Dim i As Integer, j As Integer, f30 As Integer, f50 As Integer
Dim G1 As Double, G2 As Double, ne As Integer, Nois As Double
Dim workfile As String
-----------------------------------------------------------------------------------------
Private Sub Command1_Click()
' reading data from input form
Dim Counter As Integer
    Counter = 0
    ProgressBar1.Min = Counter
  ProgressBar1.Max = 14001
  ProgressBar1.Visible = True
-----------------------------------------------------------------------------------------
'Set the Progress's Value to Min.
-----------------------------------------------------------------------------------------
  ProgressBar1.Value = ProgressBar1.Min
    ProgressBar1.Value = Counter
B33 = b11(0).Text
B35 = b21(1).Text
B55 = b31(2).Text
B53 = b41(3).Text
c33 = Me_c33(4).Text
c35 = Me_c35(5).Text
c55 = Me_c55(6).Text
```

```
c53 = Me_c53(7).Text
f30 = Me_f30(8).Text
f50 = Me_f50(9).Text
omega = Me_omeg(10).Text
ne = Noise.Value
Nois = Noise_value(0).Text
  ProgressBar1.Value = Counter
-------------------------------------------------------------------------------
' creating a temp. working  file
-------------------------------------------------------------------------------
Open "c:\simulation\temp" For Output As #1
Print #1, B33, B35, B55, B53
Print #1, c33, c35, c55, c53
Print #1, f30, f50, omega, ne, Nois
Close #1
delta = 0.1
x3(1) = 0
v3(1) = 0
x5(1) = 0
v5(1) = 0
t = 0
-------------------------------------------------------------------------------
' the following generates the heave and pitch force
-------------------------------------------------------------------------------
For i = 1 To 3000
Counter = Counter + 1
F1(i) = f30 * Sin(omega * t)
F2(i) = f50 * Sin(omega * t)
t = t + delta
  ProgressBar1.Value = Counter
Next i
-------------------------------------------------------------------------------
'Numerical integration of the ship's equations of motion
-------------------------------------------------------------------------------
For i = 1 To 3000
Counter = Counter + 1
Call equation(B33, c33, B35, c35, B53, c53, B55, c55, x3(i), x5(i), v3(i), v5(i))
k(1) = delta * (v3(i))
L(1) = delta * (F1(i) - G1)
M(1) = delta * (v5(i))
N(1) = delta * (F2(i) - G2)
Call equation(B33, c33, B35, c35, B53, c53, B55, c55, x3(i) + 0.5 * k(1), x5(i) + 0.5 *
M(1), v3(i) + 0.5 * L(1), v5(i) + 0.5 * N(1))
k(2) = delta * (v3(i) + 0.5 * L(1))
L(2) = delta * (((F1(i) + F1(i + 1)) / 2) - G1)
M(2) = delta * (v5(i) + 0.5 * N(1))
```

```
N(2) = delta * (((F2(i) + F2(i + 1)) / 2) - G2)
Call equation(B33, c33, B35, c35, B53, c53, B55, c55, x3(i) + 0.5 * k(2), x5(i) + 0.5 *
M(2), v3(i) + 0.5 * L(2), v5(i) + 0.5 * N(2))
k(3) = delta * (v3(i) + 0.5 * L(2))
L(3) = delta * (((F1(i) + F1(i + 1)) / 2) - G1)
M(3) = delta * (v5(i) + 0.5 * N(2))
N(3) = delta * (((F2(i) + F2(i + 1)) / 2) - G2)
Call equation(B33, c33, B35, c35, B53, c53, B55, c55, x3(i) + k(3), x5(i) + M(3), v3(i) +
L(3), v5(i) + N(3))
k(4) = delta * (v3(i) + L(3))
L(4) = delta * (F1(i + 1) - G1)
M(4) = delta * (v5(i) + N(3))
N(4) = delta * (F2(i + 1) - G2)
x3(i + 1) = x3(i) + ((k(1) + 2 * k(2) + 2 * k(3) + k(4)) / 6)
x5(i + 1) = x5(i) + ((M(1) + 2 * M(2) + 2 * M(3) + M(4)) / 6)
v3(i + 1) = v3(i) + ((L(1) + 2 * L(2) + 2 * L(3) + L(4)) / 6)
v5(i + 1) = v5(i) + ((N(1) + 2 * N(2) + 2 * N(3) + N(4)) / 6)
ProgressBar1.Value = Counter
Next i
For i = 1 To 3000
Counter = Counter + 1
Call equation(B33, c33, B35, c35, B53, c53, B55, c55, x3(i), x5(i), v3(i), v5(i))
a3(i) = F1(i) - G1
a5(i) = F2(i) - G2
ProgressBar1.Value = Counter
Next i

 For i = 1 To 3000
 Counter = Counter + 1
 x3(i) = x3(i) + (x3(i) * Rnd * Nois * ne)
x5(i) = x5(i) + (x5(i) * Rnd * Nois * ne)
v3(i) = v3(i) + (v3(i) * Rnd * Nois * ne)
v5(i) = v5(i) + (v5(i) * Rnd * Nois * ne)
a3(i) = a3(i) + (a3(i) * Rnd * Nois * ne)
a5(i) = a5(i) + (a5(i) * Rnd * Nois * ne)
ProgressBar1.Value = Counter
Next i
t = 100
-------------------------------------------------------------------------------------------
' Writing out the results
-------------------------------------------------------------------------------------------
file = CStr(omega)
file = Trim(file)
Open "c:\simulation\hevpitch" + file + ".out" For Output As #4
Print #4, "t"; ","; "x3"; ","; "v3"; ","; "a3"; ","; "F30"; ","; "x5"; ","; "v5"; ","; "a5"; ",";
"f50"
```

```
For j = 1000 To 3000
Counter = Counter + 1
Print #4, t; ","; x3(j); ","; v3(j); ","; a3(j); ","; F1(j); ","; x5(j); ","; v5(j); ","; a5(j); ",";
F2(j)
t = t + delta
ProgressBar1.Value = Counter
Next j
Close #4
Timer1.Interval = 2000
End Sub
```

```
Public Sub equation(B33, c33, B35, c35, B53, c53, B55, c55, x3 As Double, x5 As
Double, v3 As Double, v5 As Double)
G1 = (B33 * v3) + (c33 * x3) + (B35 * v5) + (c35 * x5)
G2 = (B53 * v3) + (c53 * x3) + (B55 * v5) + (c55 * x5)
End Sub
```

```
Private Sub Command2_Click()
```

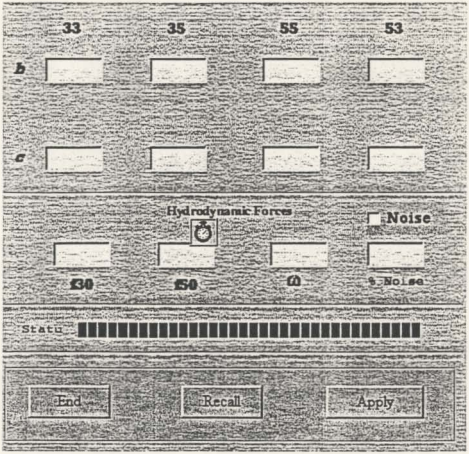
```
'recalling the previous inputs
```

```
Open "c:\simulation\temp" For Input As #1
Input #1, B33, B35, B55, B53
Input #1, c33, c35, c55, c53
Input #1, f30, f50, omega, ne, Nois
Close #1
b11(0).Text = B33
b21(1).Text = B35
b31(2).Text = B55
b41(3).Text = B53
Me_c33(4).Text = c33
Me_c35(5).Text = c35
Me_c55(6).Text = c55
Me_c53(7).Text = c53
Me_f30(8).Text = f30
Me_f50(9).Text = f50
Me_omeg(10).Text = omega
Noise_value(0).Text = Nois
If ne = 1 Then Noise.Value = 1
End Sub
```

```
Private Sub Command3_Click()
Unload Me
```

```
End Sub
```

```
Private Sub Timer1_Timer()
ProgressBar1.Value = ProgressBar1.Min
End Sub
```



APPENDIX C LIST OF RUNS

DREA_standard_sub_runs_13oct_to_17nov.xls
I. Mohamed, C.D.Williams, 23 Dec 99 - revised 19 Jan 2000

Dynamic runs for DREA standard sub

Dir: TOW:[TEST_PJ9868.OCT99]	VMS blocks	Date acquired	Time acquired	File type	Maneuver Type	Drift Angle	Status	Time segments	Cs	Amplitude	Remarks
SVA0500H_001.DAC;1	2086	08-Nov-99	9:52:08	Dynamic	Horizontal Arc 0.05	0	Successful	80- 200 sec	0		
SVA1000H_001.DAC;1	902	08-Nov-99	11:10:53	Dynamic	Horizontal Arc 0.10	0	Premature termination	25-80 sec	0		High Following error- Aft vertical
SVA0500H_002.DAC;1	1124	08-Nov-99	11:47:03	Dynamic	Horizontal Arc 0.05	0	Successful	10-135 sec	0		
SVA1000H_002.DAC;1	754	08-Nov-99	11:50:09	Dynamic	Horizontal Arc 0.10	0	Premature termination	13-63 sec	0		High Following error- Aft vertical
SVA1000H_003.DAC;1	1420	08-Nov-99	11:56:34	Dynamic	Horizontal Arc 0.10	0	Successful	52-130sec	1.5		
SVA1500H_001.DAC;1	1346	08-Nov-99	12:09:46	Dynamic	Horizontal Arc 0.15	0	Premature termination	80-150 sec	1.5		High Following error- Aft vertical
SVA0500H_004.DAC;1	1346	08-Nov-99	12:22:45	Dynamic	Horizontal Arc 0.05	0	Successful	60-140 sec	1.5		
SVA2000H_001.DAC;1	902	08-Nov-99	12:32:18	Dynamic	Horizontal Arc 0.20	0	Premature termination	65-95 sec	1.5		High Following error- Aft vertical
SVA2500H_001.DAC;1	1272	08-Nov-99	14:52:39	Dynamic	Horizontal Arc 0.25	0	Premature termination	80-130 sec	1.5		High Following error- Aft vertical
SVA3000H_001.DAC;1	1346	08-Nov-99	15:03:56	Dynamic	Horizontal Arc 0.30	0	Premature termination	90-125 sec	1.5		High Following error- Aft vertical
SVA0500V_001.DAC;1	2678	08-Nov-99	15:55:16	Dynamic	Vertical Arc 0.05	0	Failed		1.5		High Following error- Aft vertical
SVA0500V_002.DAC;1	2678	08-Nov-99	16:06:54	Dynamic	Vertical Arc 0.05	0	Failed		1.5		High Following error- Aft vertical
Hull + Sail + Tail											
HST_SVA0500H_001.DAC;1	1346	10-Nov-99	15:53:05	Dynamic	Horizontal Arc 0.05	0	Successful	75-150 sec	1.5		
HST_SVA1000H_001.DAC;1	1494	10-Nov-99	16:03:24	Dynamic	Horizontal Arc 0.10	0	Successful	85-160sec	1.5		
HST_SVA1500H_001.DAC;1	1346	10-Nov-99	16:12:57	Dynamic	Horizontal Arc 0.15	0	Successful	75-140 sec	1.5		
HST_SV2000H_001.DAC;1	1346	10-Nov-99	16:23:32	Dynamic	Horizontal Arc 0.20	0	Successful	65-140 sec	1.5		
HST_SV2500H_001.DAC;1	1198	10-Nov-99	16:32:53	Dynamic	Horizontal Arc 0.25	0	Premature termination	60-130 sec	1.5		High Following error- Aft vertical
HST_SV3000H_001.DAC;1	1272	10-Nov-99	16:42:50	Dynamic	Horizontal Arc 0.30	0	Premature termination	70-140 sec	1.5		see foot note 1
HST_SV3000V_TEST_001.DAC;1	1642	10-Nov-99	16:54:12	Dynamic	Vertical Arc 0.30	0	Successful	50-175 sec	0		
Hull + Sail											
HS_SVA0500H_001.DAC;1	1346	12-Nov-99	13:07:09	Dynamic	Horizontal Arc 0.05	0	Successful	75-150 sec	1.5		
HS_SVA1000H_001.DAC;1	1272	12-Nov-99	13:16:50	Dynamic	Horizontal Arc 0.10	0	Successful	60-130 sec	1.5		
HS_SVA1500H_001.DAC;1	1272	12-Nov-99	13:28:18	Dynamic	Horizontal Arc 0.15	0	Successful	60-130 sec	1.5		
HS_SVA2000H_001.DAC;1	1198	12-Nov-99	13:39:34	Dynamic	Horizontal Arc 0.20	0	Successful	60-130 sec	1.5		
HS_SVA2500H_001.DAC;1	1198	12-Nov-99	13:49:54	Dynamic	Horizontal Arc 0.25	0	Successful	60-125 sec	1.5		
HS_SVA3000H_001.DAC;1	606	12-Nov-99	13:58:29	Dynamic	Horizontal Arc 0.30	0	Failed		1.5		Following error-Fwd Lateral
HS_SVA3000H_002.DAC;1	1272	12-Nov-99	16:34:30	Dynamic	Horizontal Arc 0.30	0	Successful	70-140 sec	1.5		
Hull + Sail + Tail											
HST_SVA0505H_001.DAC;1	1346	15-Nov-99	12:21:02	Static	Horizontal Arc 0.05	5	Successful	50- 150 sec	1.5		
HST_SVA0505H_002.DAC;1	1346	15-Nov-99	12:28:42	Dynamic	Horizontal Arc 0.05	5	Successful	80-150 sec	1.5		
HST_SVA0510H_001.DAC;1	1272	15-Nov-99	12:38:24	Dynamic	Horizontal Arc 0.05	10	Successful	75-150 sec	1.5		
HST_SVA1005H_001.DAC;1	1642	15-Nov-99	14:41:56	Dynamic	Horizontal Arc 0.10	5	Successful	120-190 sec	1.5		
HST_SVA1010H_001.DAC;1	1568	15-Nov-99	14:54:02	Dynamic	Horizontal Arc 0.10	10	Successful	90-160 sec	1.5		
HST_SVA1505H_001.DAC;1	1198	15-Nov-99	15:04:38	Dynamic	Horizontal Arc 0.15	5	Successful	65-130 sec	1.5		
HST_SVA2005H_001.DAC;1	1198	15-Nov-99	15:15:05	Dynamic	Horizontal Arc 0.20	5	Successful	65-130 sec	1.5		
HST_SVA2505H_001.DAC;1	1198	15-Nov-99	15:24:44	Dynamic	Horizontal Arc 0.25	5	Successful	60-125 sec	1.5		
HST_SVA1510H_001.DAC;1	1124	15-Nov-99	15:35:17	Dynamic	Horizontal Arc 0.15	10	Successful	60-125 sec	1.5		
HST_SVA2010H_001.DAC;1	1124	15-Nov-99	15:46:11	Dynamic	Horizontal Arc 0.20	10	Successful	60- 125 sec	1.5		
HST_SVA2510H_001.DAC;1	1346	15-Nov-99	15:55:59	Dynamic	Horizontal Arc 0.25	10	Successful	80-155 sec	1.5		
HST_SVA0500V_001.DAC;1	1050	15-Nov-99	16:20:41	Dynamic	Vertical Arc 0.05	0	Successful	45 - 120 sec	1.5		
HST_SVA1000V_001.DAC;1	532	15-Nov-99	17:02:58	Dynamic	Vertical Arc 0.05	0	Failed				see footnote 2

HST_HSPY_F1T3V700_001.DAC;	606	16-Nov-99	18:11:35	Dynamic	4 DOF Chirp 0.1 to 0.3	Premature termination	45- 65 sec	2		Following error- Aft vertical
HST_YP_F1T3V700_001.DAC;1	606	16-Nov-99	18:21:55	Dynamic	Combined Yaw & Pitch Chirp	Premature termination	50-60 sec	2		Following error- Aft vertical
Hull + Sail										
HS_F1T3V250S_001.DAC;1	902	16-Nov-99	19:10:41	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	45- 110 sec	2		
HS_F1T3V500S_001.DAC;1	1198	16-Nov-99	19:24:08	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	85-140 sec	2		
HS_F1T3V600S_001.DAC;1	754	16-Nov-99	19:32:49	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Premature termination	63-85 sec	2		High following error- Fwd Lateral
HS_F1T3V700S_001.DAC;1	902	16-Nov-99	19:43:01	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	45-105 sec	2		
HS_F1T3V800S_001.DAC;1	680	16-Nov-99	19:58:34	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Premature termination	50- 65 sec	2		High following error- Fwd Lateral
HS_F1T3V600S_002.DAC;1	976	16-Nov-99	20:27:53	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	50- 105 sec	2		
HS_F1T3V800S_002.DAC;1	680	16-Nov-99	20:39:28	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Premature termination	45- 75 sec	2		High following error- Fwd Lateral
HS_F1T3V250H_001.DAC;1	976	16-Nov-99	20:48:58	Dynamic	Heave Chirp 0.1 to 0.3 Hz	Successful	50- 105 sec	2		
HS_F1T3V500Y_001.DAC;1	902	16-Nov-99	20:59:49	Dynamic	Yaw Chirp 0.1 to 0.3 Hz	Successful	40- 105 sec	2		
HS_F1T5V250Y_001.DAC;1	976	16-Nov-99	21:10:52	Dynamic	Yaw Chirp 0.1 to 0.5 Hz	Successful	60- 114 sec	2		
HS_F1T3V700Y_001.DAC;1	902	16-Nov-99	21:21:14	Dynamic	Yaw Chirp 0.1 to 0.3 Hz	Successful	50- 105 sec	2		
HS_F1T3V250YS_001.DAC;1	1124	16-Nov-99	21:31:58	Dynamic	Combined Yaw & Sway Chirp	Successful	75- 130 sec	2		
HS_F1T3V500YS_001.DAC;1	976	16-Nov-99	21:41:32	Dynamic	Combined Yaw & Sway Chirp	Successful	60- 114 sec	2		
HS_F1T5V400YS_001.DAC;1	976	16-Nov-99	21:50:56	Dynamic	Combined Yaw & Sway Chirp	Successful	57- 110 sec	2		
HS_F1T5V500YS_001.DAC;1	754	16-Nov-99	21:59:40	Dynamic	Combined Yaw & Sway Chirp	Premature termination	60- 75 sec	2		Following error- Fwd Lateral
HS_HSPY_F1T3V700_001.DAC;1	606	16-Nov-99	22:15:12	Dynamic	4 DOF Chirp 0.1 to 0.3	Premature termination	40- 62 sec	2		Following error-Aft vertical
HS_YP_F1T3V700_001.DAC;1	902	16-Nov-99	22:24:33	Dynamic	Combined Yaw & Pitch Chirp	Failed		2		Following error
Hull + Fins										
HT_F1T3V250S_001.DAC;1	1420	17-Nov-99	10:46:50	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	110- 165 sec	2		
HT_F1T3V500S_001.DAC;1	1346	17-Nov-99	10:56:23	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	95- 150 sec	2		
HT_F1T3V600S_001.DAC;1	1050	17-Nov-99	11:05:57	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	60- 110 sec	2		
HT_F1T3V700S_001.DAC;1	1198	17-Nov-99	11:24:39	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Successful	80- 130 sec	2		
HT_F1T3V800S_001.DAC;1	1198	17-Nov-99	11:36:30	Dynamic	Sway Chirp 0.1 to 0.3 Hz	Premature termination	110- 130 sec	2		High following error- Fwd lateral
HT_F1T3V250H_001.DAC;1	1124	17-Nov-99	11:48:17	Dynamic	Heave Chirp 0.1 to 0.3 Hz	Successful	70- 125 sec	2		
HT_F1T3V500Y_001.DAC;1	1494	17-Nov-99	12:11:12	Dynamic	Yaw Chirp 0.1 to 0.3 Hz	Successful	80- 130 sec	2		
HT_F1T5V250Y_001.DAC;1	1124	17-Nov-99	12:20:33	Dynamic	Yaw Chirp 0.1 to 0.5 Hz	Successful	60-115 sec	2		
HT_F1T3V700Y_001.DAC;1	1272	17-Nov-99	12:29:58	Dynamic	Yaw Chirp 0.1 to 0.3 Hz	Successful	80- 140 sec	2		
HT_F1T3V250YS_001.DAC;1	1272	17-Nov-99	13:47:55	Dynamic	Combined Yaw & Sway Chirp	Successful	85- 140 sec	2		
HT_F1T3V500YS_001.DAC;1	1124	17-Nov-99	14:17:31	Dynamic	Combined Yaw & Sway Chirp	Successful	60- 120 sec	2		
HT_F1T5V400YS_001.DAC;1	1124	17-Nov-99	14:26:51	Dynamic	Combined Yaw & Sway Chirp	Successful	70- 125 sec	2		
HT_F1T5V500YS_001.DAC;1	1124	17-Nov-99	14:36:42	Dynamic	Combined Yaw & Sway Chirp	Successful	70- 125 sec	2		
HT_HSPY_F1T3V700_001.DAC;1	754	17-Nov-99	14:50:03	Dynamic	4 DOF Chirp 0.1 to 0.3	Premature termination	60- 75 sec	2		Following error- Aft vertical
HT_YH_F1T3V700_001.DAC;1	828	17-Nov-99	15:07:41	Dynamic	Combined Yaw & Heave Chirp	Premature termination	80- 92 sec	2		Following error- Aft vertical
HT_SCREW5DEG_001.DAC;1	1124	17-Nov-99	15:24:23	Dynamic	Screw Maneuver 5 Deg Rotation	Successful	70- 125 sec	2		
HT_SCREW_NOROT_001.DAC;1	1272	17-Nov-99	15:37:07	Dynamic	Screw Maneuver no Rotation	Premature termination	85- 140 sec	2		Following error
WATER_WT_001.DAC;1	384	17-Nov-99	15:46:17	Weight						
AIR_WT_001.DAC;1	384	17-Nov-99	16:14:05	Weight						

125 dynamic runs for the DREA standard sub

1 Aft Strut shows no activity MDTF had to be shutt down and restarted
2 System Crashed - MDTF Had to be shutt down and restated
3 Following error- Fwd Lateral- model start to rock
4 Following error- Fwd Lateral- excessive rocking in the model

DREA_STD_SUB_DYNAMIC_RUNS _APR_99
I. Mohamed, C.D.Williams, 10 Sep 99.,Revised 20 January 1999

Dir TOW:[TEST_PJ98683.APR99.RAW]	VMS blocks	Date acquired	Time acquired	File type	Maneuver Type	Status	Time segments	Cs	Frequency	Amplitude	Remarks
					Hull + Sail + Tail						
SCREW_5DEG_001_001.DAC;1	1796	11-May-99	15:27:42	Dynamic	Corkscrew	Successful	30 - 240 sec	0			
SCREW_5DEG_001.DAC;1	1796	11-May-99	15:28:24	Dynamic	Corkscrew	Successful	30 - 240 sec	0			
SCREW_NO_ROT_001.DAC;1	1004	11-May-99	15:31:05	Dynamic	Corkscrew	Successful	20 - 120 sec	0			
SCREW_5DEG_002.DAC;1	1928	11-May-99	15:37:18	Dynamic	Corkscrew	Successful	10 -150 sec	0			
HSF_V3_HA200_HF025_001.DAC;1	740	12-May-99	13:19:47	Dynamic	Harmonic Heave	Successful	60 75 sec	3	0.25	0.2	
HSF_V3_HA333_HF015_001.DAC;1	806	12-May-99	13:32:14	Dynamic	Harmonic Heave	Successful	60 - 90 sec	3	0.15	0.333	
HSF_V3_HA333_HF015_002.DAC;1	608	12-May-99	13:43:24	Dynamic	Harmonic Heave	Successful	40 - 75 sec	3	0.15	0.333	
HSF_V3_HA111_HF045_001.DAC;1	740	12-May-99	13:54:20	Dynamic	Harmonic Heave	Successful	63 - 88 sec	3	0.45	0.111	
HSF_V3_HA077_HF065_001.DAC;1	608	12-May-99	14:45:32	Dynamic	Harmonic Heave	Failed		3	0.65	0.077	
HSF_V3_SA333_SF015_001.DAC;1	608	14-May-99	8:16:35	Dynamic	Harmonic Heave	Failed		3	0.15	0.333	
HSF_V3_SA333_SF015_002.DAC;1	806	14-May-99	8:29:15	Dynamic	Harmonic Sway	Successful	60 - 95 sec	3	0.15	0.333	
HSF_V3_SA200_SF025_001.DAC;1	674	14-May-99	8:44:15	Dynamic	Harmonic Sway	Successful	50 - 80	3	0.25	0.2	
HSF_V3_SA666_SF015_001.DAC;1	740	14-May-99	8:56:06	Dynamic	Harmonic Sway	Successful	60 - 88 sec	3	0.15	0.666	
HSF_V3_SA400_SF025_001.DAC;1	740	14-May-99	9:07:43	Dynamic	Harmonic Sway	Successful	55 - 88 sec	3	0.25	0.444	
HSF_V3_YA667_YF015_001.DAC;1	806	14-May-99	9:19:09	Dynamic	Harmonic Sway						

Total of 15 files

DREA_STD_SUB_DYNAMIC_RUNS_NOV_98
I. Mohamed, C.D.Williams, 10 Sep 99. Revised 20 January 1999

Dir TOW:[TEST_PJ98683.RAW	VMS blocks	Date acquired	Time acquired	File type	Maneuver Type	Status	Time segments	Cs	Frequency	Amplitude	Remarks
Hull + Sail + Tail											
V0_SWAYTEST_001.DAC;1	482	05-Nov-98	14:53:43	Dynamic	Harmonic Sway	Successful	10 - 30 sec	0	0.05		
V0_HEAVETEST_001.DAC;1	638	05-Nov-98	15:01:15	Dynamic	Harmonic Heave	Successful	10- 60 sec	0	0.12		
V0_HEAVETEST_002.DAC;1	2588	05-Nov-98	17:10:32	Dynamic	Harmonic Heave	Successful	50 - 300 sec	0	0.015		
V0_HEAVETEST_003.DAC;1	1340	05-Nov-98	17:14:27	Dynamic	Harmonic Heave	Successful	0 - 140 sec	0	0.03		
V0_HEAVETEST_004.DAC;1	404	05-Nov-98	17:15:37	Dynamic	Harmonic Heave	Successful	0 - 20 sec	0	0.145		
V0_HEAVETEST_005.DAC;1	404	05-Nov-98	17:16:56	Dynamic	Harmonic Heave	Successful	0 - 25 sec	0	0.145		
V0_HR314_HF15_001.DAC;1	482	05-Nov-98	17:20:46	Dynamic	Harmonic Heave	Successful	0 - 25 sec	0	0.15		
V0_HR314_HF25_001.DAC;1	482	05-Nov-98	18:11:56	Dynamic	Harmonic Heave	Successful	0 - 30 sec	0	0.25		
V0_HR314_HF45_001.DAC;1	560	05-Nov-98	18:22:12	Dynamic	Harmonic Heave	Failed		0	0.45		
V0_HR628_HF15_001.DAC;1	638	05-Nov-98	18:32:29	Dynamic	Harmonic Heave	Failed		0	0.15		
V0_SR314_SF15_001.DAC;1	794	05-Nov-98	18:46:03	Dynamic	Harmonic Sway	Failed		0	0.15		
V0_SR314_SF15_002.DAC;1	1340	08-Nov-98	13:55:59	Dynamic	Harmonic Sway	Successful	80 - 140 sec	0	0.15		
V0_SR400_SF15_001.DAC;1	794	08-Nov-98	14:04:33	Dynamic	Harmonic Sway	Successful	13 - 62 sec	0	0.15		
V0_SR314_SF25_001.DAC;1	950	08-Nov-98	14:09:57	Dynamic	Harmonic Sway	Successful	40 - 75 sec	0	0.25		
V0_SR628_SF15_001.DAC;1	638	08-Nov-98	14:13:08	Dynamic	Harmonic Sway	Failed		0	0.15		
V0_SR628_SF15_002.DAC;1	1418	08-Nov-98	16:11:09	Dynamic	Harmonic Sway	Failed		0	0.15		Actuator plot shows that amplitude is 0.325
V0_SR628_SF15_003.DAC;1	794	08-Nov-98	16:18:41	Dynamic	Harmonic Sway	Successful	12 - 75 sec	0	0.15		
V1_SR628_SF15_001.DAC;1	1106	08-Nov-98	16:26:13	Dynamic	Harmonic Sway	Successful	45 - 110 sec	1	0.15		
V35_SR628_SF15_001.DAC;1	872	08-Nov-98	16:34:33	Dynamic	Harmonic Sway	Successful	25 - 50 sec	3.5	0.15		
V35_SR314_SF25_001.DAC;1	638	08-Nov-98	16:42:59	Dynamic	Harmonic Sway	Successful	30 - 50 sec	3.5	0.25		
V0_SR314_YF15_001.DAC;1	1808	08-Nov-98	17:24:59	Dynamic	Combined Sway & Yaw	Successful	80 - 140 sec	0	0.15		
V0_SR400_YF15_001.DAC;1	794	08-Nov-98	17:31:28	Dynamic	Combined Sway & Yaw	Successful	12 - 62 sec	0	0.15		
V0_SR628_YF15_001.DAC;1	638	08-Nov-98	17:36:13	Dynamic	Combined Sway & Yaw	Successful	20 - 60 sec	0	0.15		
V35_SR628_YF15_001.DAC;1	716	08-Nov-98	17:40:15	Dynamic	Combined Sway & Yaw	Successful	30 - 60	3.5	0.15		
V0_SF15_YF15_001.DAC;1	716	08-Nov-98	17:58:10	Dynamic	Combined Sway & Yaw	Successful	10 - 50 sec	0	0.15		
V35_SF15_YF15_001.DAC;1	794	08-Nov-98	18:03:01	Dynamic	Combined Sway & Yaw	Successful	40 - 70 sec	3.5	0.15		
V0_SF15_YF15_002.DAC;1	716	08-Nov-98	18:28:03	Dynamic	Combined Sway & Yaw	Successful	20 - 75 sec	0	0.15		
V0_SF25_YF25_001.DAC;1	716	08-Nov-98	18:30:25	Dynamic	Combined Sway & Yaw	Successful	13 - 62 sec	0	0.25		
V35_SF25_YF25_001.DAC;1	638	08-Nov-98	18:33:23	Dynamic	Combined Sway & Yaw	Successful	30 - 45 sec	3.5	0.25		
V0_SF25_YF25_002.DAC;1	3914	08-Nov-98	18:46:40	Dynamic	Combined Sway & Yaw	Successful	10 - 50 sec	0	0.25		

Total of 30 files

APPENDIX D VISUAL BASIC PROGRAM

"PREPSHOP"

This appendix contains the code for the MDTF Prepshop transformation of axis. It reads the MDTF exported file and perform the steps described in chapter 4

```
Private Sub Form_Load()
Timemin.Caption = HScroll1(0).Value
Timemax.Caption = HScroll2(1).Value
End Sub
---
Private Sub HScroll1_Change(Index As Integer)
Timemin.Caption = HScroll1(0).Value
End Sub
---
Private Sub HScroll2_Change(Index As Integer)
Timemax.Caption = HScroll2(1).Value
End Sub
---
Private Sub OK_Click()
t3 = HScroll1(0).Value
t4 = HScroll2(1).Value
Timefrm.Hide
End Sub
---
---
Private Sub Command1_Click()
Sting_length = HScroll1.Value / 1000
stnig.Hide
End Sub
---
Private Sub Form_Load()
Stinglen.Caption = HScroll1.Value / 1000
End Sub
---
Private Sub HScroll1_Change()
Stinglen.Caption = HScroll1.Value / 1000
```

```
End Sub
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
Dim t() As Double, c() As Double, Fx() As Double, Fy() As Double, Fz() As Double,
Mx() As Double, My() As Double, Mz() As Double, FV() As Double, Av() As Double,
Fl() As Double, Al() As Double
Dim fax, fay, faz, aax, aay, aaz As Variant
Dim Sway() As Double, Heave() As Double, Yaw() As Double, Pitch() As Double
Dim Sway_vel() As Double, Heave_vel() As Double, Yaw_vel() As Double, Pitch_vel()
As Double
Dim i As Integer, y As Double, Sting_length As Double, t3 As Integer, t4 As Integer
Dim X As Single, pi As Double, Lower_value As Single, Upper_value As Single
Dim File_name As String, SaveFile As String, File_out As String
-----------------------------------------------------------------------------------------------------------
Private Sub Picture1_Click()
CommonDialog1.CancelError = True
   On Error GoTo ErrHandler
CommonDialog1.Filter = "Ascii(*.asc)|*.asc|Prepshop (*.pmd)|*.pmd|All Files (*.*)|*.*|"
CommonDialog1.ShowOpen
File_name = CommonDialog1.filename
FileCharNo = InStr(File_name, ".")
If FileCharNo = Len(File_name) Then
SaveFile = (File_name)
Else
SaveFile = Left(File_name, FileCharNo - 1)
End If
ErrHandler:
Exit Sub
End Sub
-----------------------------------------------------------------------------------------------------------
Private Sub Picture2_Click()
stnig.Show
End Sub
-----------------------------------------------------------------------------------------------------------
Private Sub Picture3_Click()
Timefrm.Show
End Sub
-----------------------------------------------------------------------------------------------------------
Private Sub Picture5_Click()
Static garb As Double
Open File_name For Input As #1
pi = 3.141592654
y = 1
StatusBar1.SimpleText = "Loading File  " + File_name
Sting_length = stnig.HScroll1.Value / 1000
t3 = Timefrm.HScroll1(0).Value
```

```
t4 = Timefrm.HScroll2(1).Value
Do While Not EOF(1)
Input #1, garb
y = y + 1
Loop
Close #1
i = y / 18
j = i
ProgressBar1.Min = 0
ProgressBar1.Max = j
ProgressBar1.Visible = True
ProgressBar1.Value = Min
ReDim t(j), c(j), Fx(j), Fy(j), Fz(j), Mx(j), My(j), Mz(j), FV(j), Av(j), Fl(j), Al(j)
ReDim Sway(j) As Double, Heave(j) As Double, Yaw(j) As Double, Pitch(j) As Double
ReDim Sway_vel(j) As Double, Heave_vel(j) As Double, Yaw_vel(j) As Double,
Pitch_vel(j) As Double
Open File_name For Input As #1
For i = 1 To j - 1
ProgressBar1.Value = i
Input #1, t(i), c(i), Fx(i), Fy(i), Fz(i), Mx(i), My(i), Mz(i), FV(i), Av(i), Fl(i), Al(i), fax,
fay, faz, aax, aay, aaz
Next i
Close #1
ProgressBar1.Visible = False
StatusBar1.SimpleText = "computing motion at BRC"
For i = 1 To j
Sway(i) = Al(i) + ((Sting_length / Sting_length - 2) * (Fl(i) - Al(i)))
X = (Fl(i) - Al(i)) / 2
Yaw(i) = Atn(X / Sqr(-X * X + 1))
Yaw(i) = Yaw(i) * 180 / pi
Heave(i) = Av(i) - ((Sting_length / Sting_length - 2) * (Av(i) - FV(i)))
X = (Av(i) - FV(i)) / 2
Pitch(i) = Atn(X / Sqr(-X * X + 1))
Pitch(i) = Pitch(i) * 180 / pi
Next i
Prep.Refresh
StatusBar1.SimpleText = "Differentiating motions to obtain velocities at BRC"
h = t(3) - t(2)
For i = 3 To j - 3
Sway_vel(i) = (1 / (12 * h)) * (Sway(i - 2) - 8 * Sway(i - 1) + 8 * Sway(i + 1) - Sway(i +
2))
Heave_vel(i) = (1 / (12 * h)) * (Heave(i - 2) - 8 * Heave(i - 1) + 8 * Heave(i + 1) -
Heave(i + 1))
Yaw_vel(i) = (1 / (12 * h)) * (Yaw(i - 2) - 8 * Yaw(i - 1) + 8 * Yaw(i + 1) - Yaw(i + 2))
Pitch_vel(i) = (1 / (12 * h)) * (Pitch(i - 2) - 8 * Pitch(i - 1) + 8 * Pitch(i + 1) - Pitch(i +
2))
```

```
Next i
StatusBar1.SimpleText = "Organizing time history for selected time segment"
Lower_value = t3 / h
Upper_value = t4 / h
If Upper_value > i Then Upper_value = i
StatusBar1.SimpleText = "Done"
End Sub

Private Sub Picture4_Click()
CommonDialog1.CancelError = True
   On Error GoTo ErrHandler2
CommonDialog1.filename = SaveFile + ".pmd"
CommonDialog1.ShowSave
CommonDialog1.Filter = "PrepShop(*.pmd)|*.pmd|"
CommonDialog1.DefaultExt = txt
File_out = CommonDialog1.filename
StatusBar1.SimpleText = "Writing Output file " + File_out
ProgressBar1.Min = 0
ProgressBar1.Max = Upper_value
ProgressBar1.Value = 0
ProgressBar1.Visible = True

Open File_out For Output As #2
Prep.Refresh
'Write #2, "Time", "Carriage Speed", "Heave", "Pich", "Sway", "Yaw", "Heave vel.",
"Pitch vel.", "Sway vel.", "Yaw vel.", "Fx", "Fy", "Fz", "Mx", "My", "Mz"
For i = Lower_value To Upper_value
ProgressBar1.Value = i
Print #2, t(i), c(i), Heave(i), Pitch(i), Sway(i), Yaw(i), Heave_vel(i), Pitch_vel(i),
Sway_vel(i), Yaw_vel(i), Fx(i), Fy(i), Fz(i), Mx(i); My(i), Mz(i)
Next i
Close #2
StatusBar1.SimpleText = "Finished"
ProgressBar1.Visible = False
ErrHandler2:

End Sub
```

```
Option Explicit

Private Sub Form_KeyPress(KeyAscii As Integer)
   Prep.Show
      Unload Me
End Sub
```

```
Private Sub Form_Load()
   lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
   lblProductName.Caption = App.Title
   Timer1.Interval = 3000
End Sub

Private Sub Frame1_Click()
Prep.Show
   Unload Me
End Sub

Private Sub Timer1_Timer()
Prep.Show
Unload Me
End Sub
```

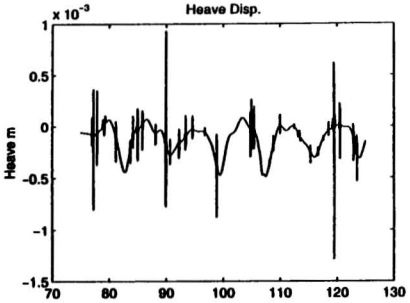
APPENDIX E MATLAB FILTERING SCRIPT

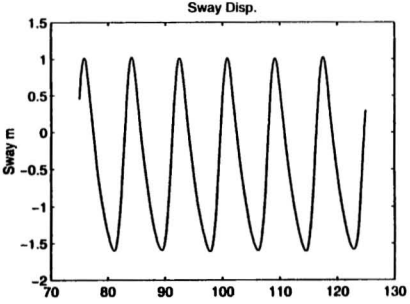
The following is the MatLab script used for filtering followed by a number of plots for filtered data vs. noisy data for a selected number of runs.

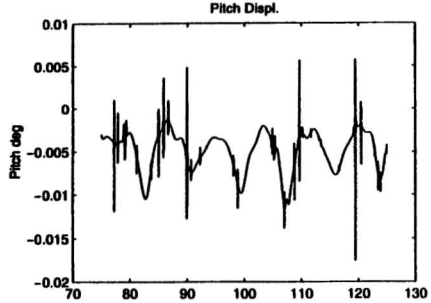
```
p1=input('enter file name without extension pmd >' ,'s');
file_load= [p1,'.pmd'];
eval(['load ',file_load]);
eval(['[s,co]=size(',p1,');']);
ti=eval([p1,'(:,1)']);
%figure(1)
ch=eval([p1,'(:,7)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_c7=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,1);plot(ti,ch);title('Heave Velocity');ylabel(' m/s')
subplot(4,1,2);plot(ti,ch_c7);title('Filtered Heave Velocity
'),ylabel('m/s')

%figure(2)
ch1=eval([p1,'(:,8)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch1);
ch_c8=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);

subplot(4,1,3);plot(ti,ch1);title('Pitch Velocity');ylabel(' deg/s')
subplot(4,1,4);plot(ti,ch_c8);title('Filtered Pich Velocity
'),;ylabel('deg/s');xlabel('Time sec')
text(-2,-2,p1)
orient landscape
eval(['print ',p1,'1 -dps'])
%pause

%figure(3)
ch=eval([p1,'(:,9)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_c9=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,1);plot(ti,ch);title('Sway Velocity');ylabel(' m/s')
subplot(4,1,2);plot(ti,ch_c9);title('Filtered Sway Velocity ');ylabel('
m/s')


%figure(4)
ch1=eval([p1,'(:,10)']);
```

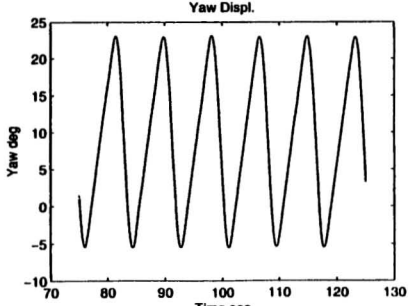
```
[c,l]=wavedec(ch1,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch1);
ch_cl0=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,3);plot(ti,ch1);title('Yaw Velocity');ylabel('deg/s')

subplot(4,1,4);plot(ti,ch_cl0);title('Filtered Yaw Velocity ');ylabel('
deg/s');xlabel('Time sec')
text(0,0,p1)
orient landscape
eval(['print ',p1,'2 -dps'])

%pause

ch=eval([p1,'(:,11)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_cl1=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,1);plot(ti,ch);title('Fx');ylabel('  N')

subplot(4,1,2);plot(ti,ch_cl1);title('Filtered Fx ');ylabel('  N')


ch1=eval([p1,'(:,14)']);
[c,l]=wavedec(ch1,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch1);
ch_cl4=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,3);plot(ti,ch1);title('Mx');ylabel(' N-m')
subplot(4,1,4);plot(ti,ch_cl4);title('Filtered Mx ');ylabel(' N-
m');xlabel('Time sec')
text(0,0,p1)
orient landscape
eval(['print ',p1,'3 -dps'])
%pause


ch=eval([p1,'(:,12)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_cl2=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,1);plot(ti,ch);title('Fy');ylabel('N')
subplot(4,1,2);plot(ti,ch_cl2);title('Filtered Fy ');ylabel(' N')

ch1=eval([p1,'(:,16)']);
[c,l]=wavedec(ch1,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch1);
ch_cl6=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,3);plot(ti,ch1);title('Mz');ylabel(' N-m')

subplot(4,1,4);plot(ti,ch_cl6);title('Filtered Mz');ylabel(' N-
m');xlabel('Time sec')

text(1,1,p1)
orient landscape
eval(['print ',p1,'4 -dps'])

ch=eval([p1,'(:,13)']);
[c,l]=wavedec(ch,10,'sym8');
```

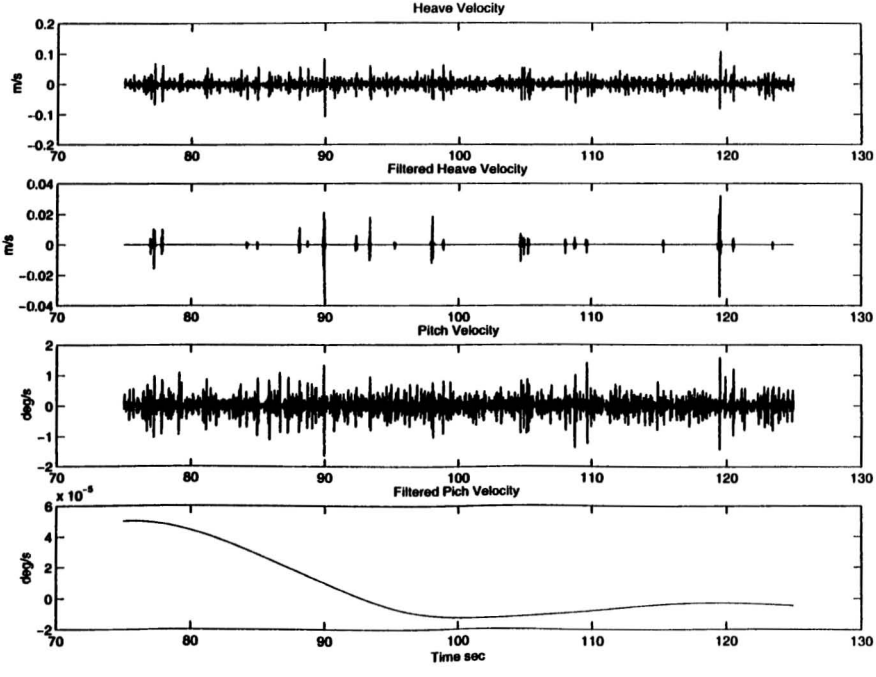
```
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_c13=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,1);plot(ti,ch);title('Fz');ylabel('N')
subplot(4,1,2);plot(ti,ch_c13);title('Filtered Fz');ylabel(' N')


ch1=eval([p1,'(:,15)']);
[c,l]=wavedec(ch1,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch1);
ch_c15=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(4,1,3);plot(ti,ch1);title('My');ylabel(' N-m')
subplot(4,1,4);plot(ti,ch_c15);title('Filtered My');ylabel(' N-
m');xlabel('Time sec')

text(1,1,p1)
orient landscape
eval(['print ',p1,'5 -dps'])
%%pause

ch=eval([p1,'(:,2)']);
[c,l]=wavedec(ch,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch);
ch_c2=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
%%pause
ch_c1=eval([p1,'(:,1)']);
ch3=eval([p1,'(:,3)']);
[c,l]=wavedec(ch3,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch3);
ch_c3=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);

ch3=eval([p1,'(:,4)']);
[c,l]=wavedec(ch3,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch3);
ch_c4=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
ch3=eval([p1,'(:,5)']);
[c,l]=wavedec(ch3,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch3);
ch_c5=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
ch3=eval([p1,'(:,6)']);
[c,l]=wavedec(ch3,10,'sym8');
[thr,sorh,keepapp]=ddencmp('den','wv',ch3);
ch_c6=wdencmp('gbl',c,l,'sym8',10,thr,sorh,keepapp);
subplot(2,2,1);plot(ti,ch_c3);title('Heave Disp.');ylabel('Heave m')
subplot(2,2,3);plot(ti,ch_c4);title('Pitch Displ.');ylabel('Pitch deg')
subplot(2,2,2);plot(ti,ch_c5);title('Sway Disp.');ylabel('Sway m')
subplot(2,2,4);plot(ti,ch_c6);title('Yaw Displ.');ylabel('Yaw
deg');xlabel('Time sec')
text(1,.5,p1)
orient landscape
eval(['print ',p1,'6 -dps'])

Filesave=[p1,'.flt'];
matrix=[p1,'a']
eval([matrix,'=[ch_c1 ch_c2 ch_c3 ch_c4 ch_c5 ch_c6 ch_c7 ch_c8 ch_c9
ch_c10 ch_c11 ch_c12 ch_c13 ch_c14 ch_c15 ch_c16];']);
eval(['save ',Filesave,' ',matrix,' -ascii -double -tabs ;']);
```

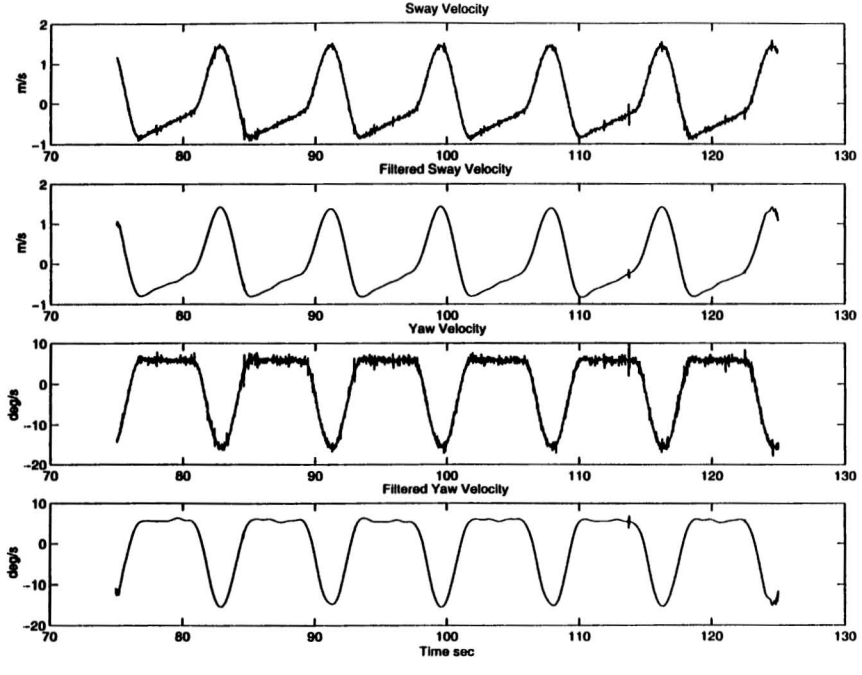

Heave Disp.
x 10^-3
Heave m
1
0.5
0
-0.5
-1
-1.5
70
80
90
100
110
120
130

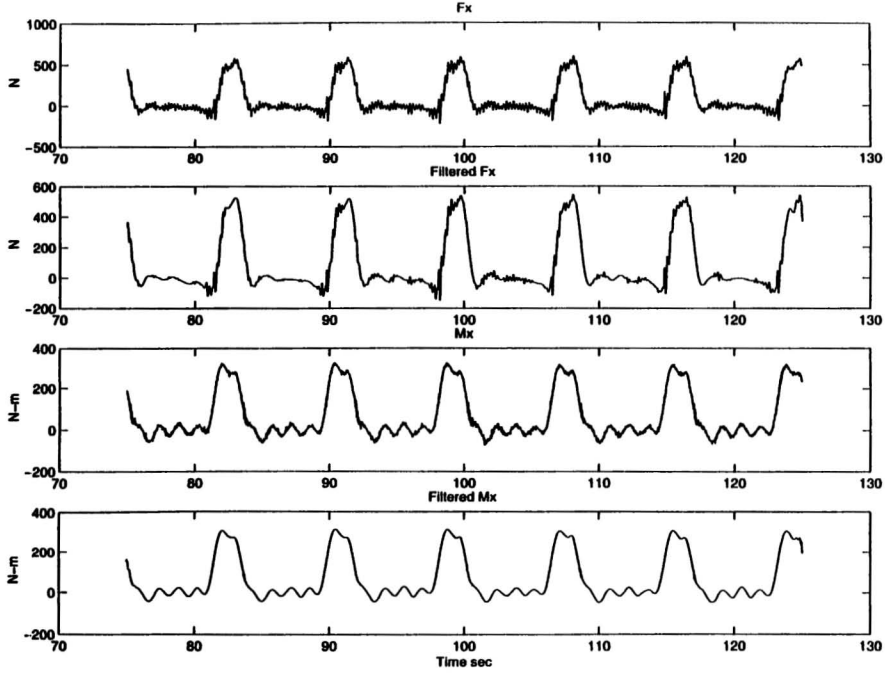

Sway Disp.
Sway m
1.5
1
0.5
0
-0.5
-1
-1.5
-2
70
80
90
100
110
120
130

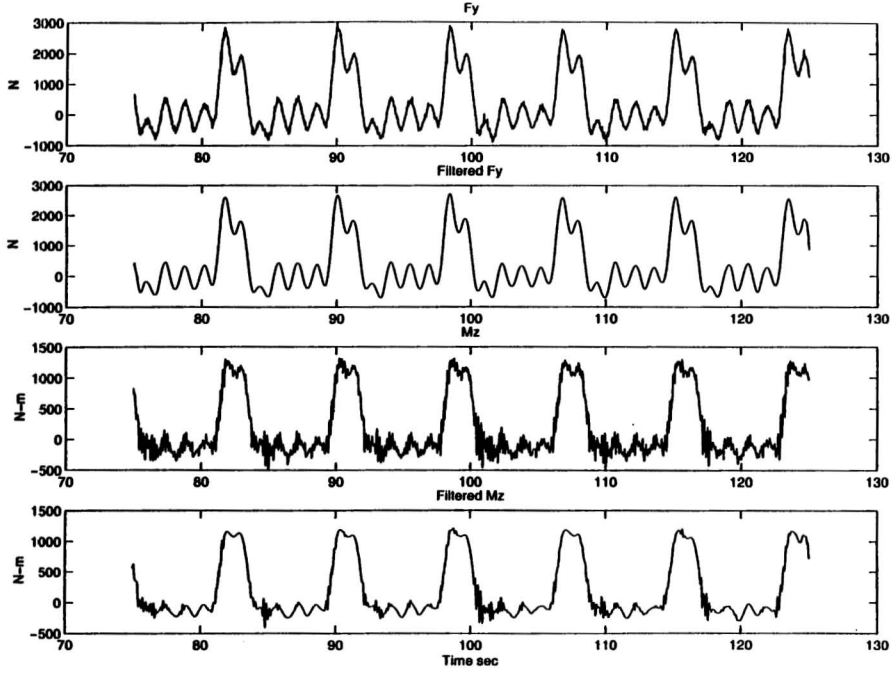

Pitch Displ.
Pitch deg
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
70
80
90
100
110
120
130

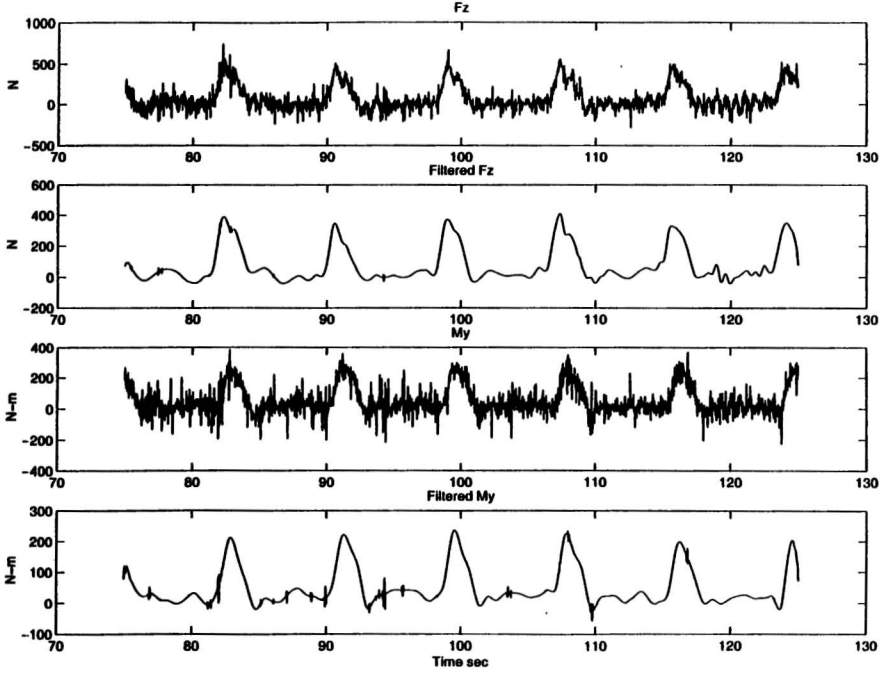

Yaw Displ.
Yaw deg
25
20
15
10
5
0
-5
-10
70
80
90
100
110
120
130
Time sec


Heave Velocity
m/s
0.2
0.1
0
-0.1
-0.2
70
80
90
100
110
120
130
Filtered Heave Velocity
m/s
0.04
0.02
0
-0.02
-0.04
70
80
90
100
110
120
130
Pitch Velocity
deg/s
2
1
0
-1
-2
70
80
90
100
110
120
130
Filtered Pich Velocity
x 10^-5
deg/s
6
4
2
0
-2
70
80
90
100
110
120
130
Time sec


Sway Velocity
m/s
2
1
0
−1
70
80
90
100
110
120
130
Filtered Sway Velocity
m/s
2
1
0
−1
70
80
90
100
110
120
130
Yaw Velocity
deg/s
10
0
−10
−20
70
80
90
100
110
120
130
Filtered Yaw Velocity
deg/s
10
0
−10
−20
70
80
90
100
110
120
130
Time sec


Fx
N
1000
500
0
-500
70
80
90
100
110
120
130
Filtered Fx
N
600
400
200
0
-200
Mx
N-m
400
200
0
-200
Filtered Mx
Time sec


Fy
N
3000
2000
1000
0
-1000
70
80
90
100
110
120
130
Filtered Fy
N
3000
2000
1000
0
-1000
70
80
90
100
110
120
130
Mz
N-m
1500
1000
500
0
-500
70
80
90
100
110
120
130
Filtered Mz
N-m
1500
1000
500
0
-500
70
80
90
100
110
120
130
Time sec


Fz
N
1000
500
0
-500
70
80
90
100
110
120
130
Filtered Fz
N
600
400
200
0
-200
70
80
90
100
110
120
130
My
N-m
400
200
0
-200
-400
70
80
90
100
110
120
130
Filtered My
N-m
300
200
100
0
-100
70
80
90
100
110
120
130
Time sec