Optimal Ship Routing

by

KYRIAKOS AVGOULEAS

Bachelor of Science in Marine Engineering
Hellenic Naval Academy, 1999

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degrees of

NAVAL ENGINEER and
MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2008]
May 2008

Signature of Author .....
Department of Mechanical Engineering
May 20, 2008

Certified by .....
Paul D. Sclavounos
Professor of Mechanical Engineering and Naval Architecture
Thesis Supervisor

Accepted by .....
Lallit Anand
Professor of Mechanical Engineering
Chairman, Committee on Graduate Studies

Optimal Ship Routing

by

Kyriakos Avgouleas

Submitted to the Department of Mechanical Engineering
on May 9, 2008 in Partial Fulfillment of the
Requirements for the Degrees of Naval Engineer and
Master of Science in Mechanical Engineering

ABSTRACT

Fuel savings in ship navigation has always been a popular subject in the maritime industry as well as the world's largest Navies. Oil prices and environmental considerations drive the effort for more fuel-efficient navigation. This thesis addresses the problem of deterministic minimum fuel routing by applying optimal control theory in conjunction with state of the art hydrodynamic and weather forecasting tools.

A fictitious trans-Atlantic route is established and the optimal combination of speed and heading is determined, so that fuel consumption is minimized while certain safety constraints are met. The safety constraints are defined as the probabilities of slamming and deck wetness, both of which are not allowed to exceed prescribed limiting values. The problem formulation adopted in the thesis lies in the framework of Dynamic Programming, which is most suitable for computer implementation.

The hydrodynamic performance of the ship is computed through the use of SWAN1, an advanced frequency domain CFD code. With the aid of SWAN1, ship motions and resistance can be accurately calculated. The latter includes the estimation of mean added resistance in waves, which has a major effect on the fuel consumption of ships sailing in rough seas. Wave and swell forecasts are provided in a deterministic setting by a third generation numerical wave model, the WAM cycle 4, developed at the European Center for Medium-Range Weather Forecasts (ECMWF). Utilizing the hydrodynamic results and the output of the wave model a computer program is developed in MATLAB®, which employs the Iterative Dynamic Programming algorithm to solve the optimal control problem.

Thesis Supervisor: Paul D. Sclavounos

Title: Professor of Mechanical Engineering and Naval Architecture

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my thesis advisor, Professor Paul Sclavounos, for his insightful guidance through the course of this challenging problem.

For their invaluable contribution to my understanding of some aspects of dynamic programming, wave modeling and ship chartering I am indebted to the following individuals:
Professor Stanley B. Gershwin, Department of Mechanical Engineering, MIT
Professor Kerry A. Emanuel, Department of Earth and Planetary Sciences, MIT
Professor Isaac Ginis, University of Rhode Island
Professor Tetsu Hara, University of Rhode Island
Mr Kyriakos Giannopoulos, MEDTANKERS S.A.

For providing the data from the WAM numerical model I am grateful to:
Professor George Kallos, University of Athens
Professor George Galanis, Hellenic Naval Academy

For sponsoring my graduate studies I would like to thank the Hellenic Navy.

On a personal level, I wish to express my gratitude to the Director of Naval Postgraduate Office at MIT, Captain Patrick J. Keenan USN for his mentoring and support.

Above all, I thank my wife Elisavet for her patience and understanding throughout the three years of my intensive studies at MIT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 Motivation and Review of Existing Research

Global maritime transportation relies almost exclusively on fossil fuels as the energy resource for ship powering. This is also true for most of the world's combat fleets, especially the ones not exploiting nuclear power for their propulsion plants. There is a tremendous environmental and economic impact of oil usage in navigation. Authorities around the world have begun to enforce strict regulations regarding the air and sea pollution emanating from the use of bunker fuel in the vicinity of the coastal line. At the same time, the world is witnessing an unprecedented ascent of oil price. The need to find ways to reduce fuel consumption is even more compelling today than it was in the past.

A lot of research is ongoing in the field of hydrodynamics and propulsion. Apart from designing ships with lower drag and propulsors with higher efficiencies, a more direct approach to the problem is to devise strategies for the optimal utilization of propulsion plants on the existing ships. Minimal time and/or minimal cost navigation has been the subject of serious research work for more than five decades.

The pioneering work of James (1957) set the foundation for ship routing research. He studied the effect of waves on impeding the speed of a ship through the water and quantified this effect into empirical curves. These were parametric curves relating speed and wave height with wave direction as a parameter. Haltiner, Hamilton and Arnason (1962) used these performance curves to solve the deterministic minimum time routing problem assuming stationary wave fields. They solved the unconstrained problem using calculus of variations. Bleick and Faulkner (1964) solved the same problem as Haltiner et al. allowing for time varying wave fields. The first rational approach to the solution of stochastic minimum time routing should be attributed to Zoppoli (1972). He formulated the constrained dynamic programming problem treating wave height and direction as

random variables. The solution followed Bellman's recurrence relation applied under the expectation operator $E\{...\}$. Zoppoli also made use of James' relations for speed vs. wave height. In his PhD thesis, Bijlsma (1975) solved the minimal time routing problem using deterministic weather forecasts and Pontryagin's maximum principle. Similarly to the work done before him, he did not account for ship hydrodynamics in detail, but rather resorted to the polar velocity diagrams (i.e. the James results). Later, he extended the same computational method to the minimum fuel and other related problems with different cost functions (Bijlsma, 2001). He also derived the connection of maximum principle with dynamic programming (Bijlsma, 2002) and addressed minimum time/minimum fuel navigation using the concept of limited maneuverability (Bijlsma, 2004).

After Zoppoli's attempt to model the stochastic optimal control problem as an N-stage process, the first comprehensive study of stochastic minimum-cost navigation was conducted by Frankel and Chen (1980) at the Massachusetts Institute of Technology. In their technical report for the National Maritime Research Center, they presented a dynamic programming framework of the stochastic control problem in which the sea-state parameters (significant wave height, peak frequency, and predominant direction) were assumed to be random variables. Forecast data for wind waves and swell were fitted into standard spectral forms (Bretschneider). Besides the insightful formulation of the optimal control aspect, an important contribution of the work of Frankel and Chen was the fact that they abandoned the empirical performance curves introduced by James and attempted to incorporate the actual physics of the problem directly in their model. They used existing hydrodynamic knowledge to carry out rational calculations of ship responses (RAO's), ship resistance and powering in calm water and in waves.

The PhD thesis of Hagiwara (1989) dealt with the deterministic minimum fuel and minimum time routing using the method of isochrones introduced by James (1957). The stochastic counterpart of it was also examined, by providing an estimate for the mean and standard deviation of voyage time and consumed fuel. Perakis and Papadakis (1990) solved the deterministic minimum time routing problem using calculus of variations. The

vessel speed was taken as a function of significant wave height and wave direction. The dependence on engine power setting was also accounted for. Lo and McCord (1998) addressed the stochastic minimum fuel routing under the influence of ocean currents only. They utilized aged global current data (from satellite or radar altimetry) and real-time local data (from ship observations) to estimate state transition probabilities. To solve the problem they derived an adaptive stochastic optimization algorithm. Waves and ship motions were not considered. The fuel consumption was approximated by a cubic function.

Allsopp, Mason and Philpott (2000) tackled the question of minimal sailing time of racing yachts. They used a stochastic dynamic programming algorithm in which the weather uncertainty was modeled as a branching tree. No explicit reference was made to the hydrodynamics of the problem, which was embedded in the software they used to obtain the velocity prediction polar plots as a function of wind speed and angle. Azaron and Kianfar (2003) employed the theory of directed acyclic networks which finds the shortest path from source 1 to sink n. They solved the stochastic shortest-path routing assuming that at each node i (denoting the geographical location) the probability of transition of the environmental variables (i.e. weather conditions) from one state to another is governed by a continuous time Markov process. The arc lengths connecting the nodes represent the actual path lengths. These were assumed to follow an exponential distribution. Their approach involved no ship hydrodynamics. The optimal control problem was modeled with the aid of the network flow formalism used in Operations Research. At each geographical position (node) the decision is whether to stay at that position or leave, and which should be the next position.

Another approach which borrows heavily from Operations Research but pays little attention to the governing physics was proposed by Vlachos (2004). He explored two methods to solve the routing problem independently. The first method involved an iterative optimization algorithm in which successive perturbations of a nominal track were imposed. The second was the simulated annealing method. The objective of both was to calculate almost-optimal routes which minimize a weighted combination of travel

time and safety/comfort. The approach was deterministic and no direct reference was made to hydrodynamic or propulsion modeling. The speed loss in waves, although not explicitly stated in the paper, was described parametrically similarly to the James methodology. Added resistance in waves was not considered, only added resistance induced by steering.

A different methodology from the ones described so far was proposed by Hinnenthal and Saetra (2005). They solved a two-objective optimization problem: simultaneous minimization of arrival time and fuel consumption. Using a genetic algorithm they generated a population of different routes by perturbing a parent route in space and time. Optimization was performed by constructing a Pareto frontier and choosing the Pareto-optimal route. They used only swell forecasts which they modeled using Bretschneider spectrum. They included slamming and vertical/lateral accelerations as safety constraints in the model. Although their approach was mainly deterministic, they used the statistics of ensemble forecasts to assess the robustness of the Pareto solution. A recent paper published by Padhy, Sen and Bhakaran (2007) deals with the deterministic minimal time routing based on WAM forecasts over the Indian Ocean. They use a Dijkstra algorithm to find the shortest path on a weighted directed network. The edges of the graph are assigned a weight which represents the transit time. The sum of the weights on the shortest path will be minimal. Added resistance is explicitly included in determining the transit time from node to node. It is calculated from simple empirical relations.

1.2 Organization of the Thesis

The summary of past and current research presented in the previous section is by no means exhaustive. Yet, it is evident that the ship routing problem is mainly treated as an operations research or optimal control problem with the underlying physical principles either completely neglected or simplified by virtue of heuristic or empirical results. No matter how efficient the solution algorithm, the optimization results will be meaningless unless careful attention is paid to modeling the ship's hydrodynamic behavior and its coupling with propulsion.

This thesis emphasizes on tackling the deterministic minimum fuel problem by carefully modeling the propulsion plant characteristics, the steady and unsteady hydrodynamic responses and their impact on fuel consumption. Once the physics governing the quantity to be optimized is understood and properly captured by the mathematical model, the paper proceeds with the task of numerically optimizing the target metrics.

In particular, the objective is to find the optimal combination of speed and heading to minimize fuel consumption. The nominal speed and voyage time are fixed. In reality, these would be specified and included in the chartering contract of a merchant vessel or the departure order of a naval vessel. Actual forecast information, drawn from the WAM output, is available every three hours over a span of five days. WAM is a third generation wave prediction model developed in the European Center for Medium-Range Weather Forecasts (ECMWF). If the voyage lasts more than five days, the optimization scheme assumes calm seas in the absence of any other information. Of course, the program can be run again with new initial conditions as soon as the next forecast becomes available. The time stages in between the consecutive forecasts are short enough to allow much confidence in the assumption of stationarity for the sea state. In turn, this assumption along with the assumption of linearity, permits the use of frequency domain seakeeping tools for the evaluation of the ship's hydrodynamic performance. SWAN1 is used for this purpose. It is a frequency domain CFD code which uses 3-D Rankine Panel Methods to calculate all the hydrodynamic quantities of interest to the routing problem. These include steady wave resistance, added resistance in waves and ship motions in waves. A fictitious ship based on the Series 60 monohull is used throughout the paper. Propulsion characteristics are drawn from typical low speed marine diesels and a screw propeller is selected from the MARIN B-series. The constraints of bounded slamming and deck wetness probabilities are added to the mathematical model to ensure safe passage through rough seas. For the numerical solution of minimum fuel navigation the Iterative Dynamic Programming (IDP) algorithm is coded in MATLAB.

The present work consists of two distinct parts. The first part deals with ship dynamics, resistance, propulsion and ocean wave modeling. The second part describes the optimal control aspects of the requisite problem. It formulates the mathematical model using the results from the first part, demonstrates a solution technique, and exemplifies it by numerically solving a hypothetical but realistic trans-Atlantic passage.

The exact breakdown structure of the paper is as follows:

Chapter 2 is mainly concerned with hydrodynamics. The various components of resistance and their connection with propulsion and propeller characteristics are discussed. Ideal flow hydrodynamics is briefly described as it is the theoretical basis for numerical simulation using SWAN1.

Chapter 3 deals with the statistics and modeling of ocean waves. The derivation of the energy spectrum is provided and the physics of wave generation is outlined. This chapter concludes the first part.

The second part starts with Chapter 4 which explains the basics of Dynamic Programming formulation in the context of optimal control theory. It also presents the IDP algorithm which is subsequently used in the numerical solution.

Chapter 5 formulates the particular ship routing problem using results from all the previous chapters.

Chapter 6 covers the details and implications of the numerical implementation of the model.

The results are provided in Chapter 7 together with a discussion and some concluding remarks.

Appendix A contains the hydrodynamic output of SWAN1, for a 100-m long series 60 hull. A sample ASCII output of the WAM model is given in Appendix B.

Appendix C accommodates the complete set of MATLAB scripts that were developed for the solution of the optimal control problem.

PART I

HYDRODYNAMICS AND WAVES

CHAPTER

2

SHIP HYDRODYNAMICS

2.1 Calm Water Resistance

The main components of the hydrodynamic drag force experienced by a surface piercing vessel in calm seas are:

(1) Frictional Drag
(2) Form Drag
(3) Wave Resistance

The first two are of viscous origin and sometimes are combined in the term "viscous drag". The third can be calculated using ideal flow theory and it can be thought of as the inviscid component of the total drag. In the context of the famous "Froude hypothesis" (2) and (3) are combined together into what is called "residuary resistance". Figure 1 shows a typical resistance curve for displacement vessels.

In brief, the Froude hypothesis postulates the separate calculation of frictional and residuary resistance. The former can be calculated from the frictional resistance of a smooth plate with surface equal to the ship's wetted surface. The latter can be measured by model experiment and extrapolated to the actual ship using similitude arguments. Details regarding the physical basis and methodologies of resistance calculations can be found in PNA vol II (van Manen and van Oossanen,1988).

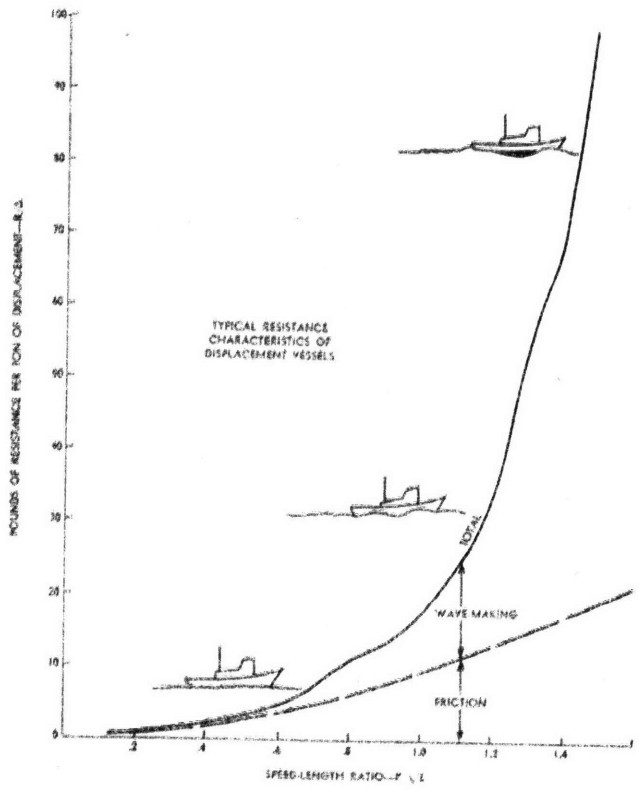


Figure 1 Typical resistance curve of displacement vessels (Gillmer and Johnson, 1982)

2.1.1 Frictional Resistance

It is the component of resistance associated with the skin friction which occurs in the flow of a real fluid around a body. It is assumed to be a function of Reynolds number

$$\mathrm{Re} = \frac{UL}{\nu} \tag{2.1}$$

where U is the flow velocity, L is a characteristic dimension of the body (e.g. ship's length) and ν is the kinematic viscosity of the fluid.

Several methods have been proposed for the calculation of the friction drag coefficient. The one used in this paper is the ITTC 1957 model ship correction line which is the widely used "…for practical engineering purposes (ITTC 1957)".

$$C_F = \frac{0.075}{(\log_{10}\mathrm{Re} - 2)^2} \tag{2.2}$$

2.1.2 Form Drag

In an ideal flow around a solid body the total kinetic energy of the fluid is converted to potential energy (in the form of high pressure) at the forward stagnation point. As the fluid moves around the body it gradually acquires kinetic energy at the expense of its potential energy. As it overcomes the maximum lateral dimension of the solid, the fluid is now moving around the back of it. In this part of the flow the motion is decelerated as a result of an adverse pressure gradient and potential energy (and pressure) starts building up again. In the absence of any loss mechanism the potential energy is fully recovered at the rear stagnation point where the fluid is at rest.

Unlike an inviscid fluid, a real fluid would experience energy losses (in the form of heat) due to viscous dissipation. As a result, the fluid cannot fully regain its potential energy upon reaching the rear stagnation point. Instead, the flow separates earlier leaving a "shadow" of turbulent wake, in which the average pressure is on the same order as the far field (ambient) pressure (Figure 2).

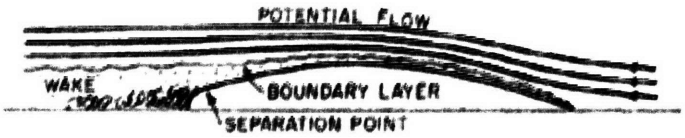


Figure 2 Flow separation and turbulent wake (PNA vol. II)

This imbalance of pressure, namely high stagnation pressure at the front end and lower pressure on the back, manifests itself through an additional drag component called *pressure drag* or *eddy resistance*. Due to its strong dependence on the body geometry it is often called *form drag*. The calculation of pressure drag is an extremely difficult process. In most cases its dependence on Re is neglected and is lumped together with the wave resistance (Gillmer and Johnson, 1982). The calculation of both components (in the form of residuary resistance) can be achieved via model testing. An estimate is possible through statistical analysis of the model data (Holtrop and Mennen, 1982).

2.1.3 Wave Resistance

It is understood that this component of resistance still refers to the steady problem of ship transiting at constant speed U. No implication about the presence of any ambient wave is made. The wave that the name refers to is the flow disturbance generated by the ship due to the existence of the free surface. The energy input from the moving body into the fluid to maintain this disturbance appears as an additional drag component.

In model experiments this drag component is indistinguishable from the pressure drag. The respective combined drag coefficient C_R is assumed to depend only on the Froude number

$$Fn = \frac{U}{\sqrt{gL}} \tag{2.3}$$

Then, the total drag coefficient can be calculated as a sum of the frictional and residuary resistance coefficients:

$$C_T(\mathrm{Re}, Fn) \simeq C_F(\mathrm{Re}) + C_R(Fn) \tag{2.4}$$

This separation of drag components, known as the Froude hypothesis, is only approximate as it neglects cross correlation effects (e.g. viscous effects on the wave making resistance). Still it is invaluable in the estimation of total resistance in a towing tank. Details on this methodology can be found in Gillmer and Johnson (1982), PNA vol. II (van Manen and van Oossanen,1988) and Newman (1977).

Wave resistance (sometimes, more precisely, referred to as wave making resistance) is dominated by gravitational effects (hence the Fn dependence). On this basis, ideal flow theory is often employed in the explicit determination of this force with remarkable success. Nakos and Sclavounos (1994) discuss two equivalent methods for this purpose, *direct pressure integration* and *wave-cut analysis*. The latter method, which is based on momentum flux calculation through transverse cuts in the fluid domain, is utilized in this paper through the use of SWAN1. Details and theoretical justification for the transverse cut method are given in Eggers,Sharma and Ward (1967).

2.1.4 Other Components of Resistance

In addition to those described above, other mechanisms exist that contribute to the total resistance. The most notable are:

(1) Air and wind resistance
(2) Appendage resistance
(3) Resistance due to trim effects
(4) Shallow water effects
(5) Added resistance in waves

Of these components, (5) is of great importance to the routing problem and is treated separately in section 2.4. The rest, namely (1) – (4), are not considered in this work. A fairly detailed description is provided in van Manen and van Oossanen (1988). The optimization problem can be extended to include air and appendage resistance almost trivially. The effect of dynamic sinkage and trim, although captured by SWAN1, is more important for ships in ballast condition (not treated here). Finally, deep water is assumed throughout the paper so (4) is not considered.

2.2 Propulsion

This section provides the mathematical description of propeller action and its connection with the propulsion plant. The equations presented here are to be used later in chapter 5, in the development of the mathematical model for the optimization problem. Once the relation between propeller thrust, ship resistance and prime mover fuel consumption is established, a model for the system's dynamics is set up. This is the fundamental step in the formulation of the routing optimal control problem (Chapter 5).

2.2.1 Propeller

Propellers are by far the most widely used mechanism to provide thrust for the motion of ocean vehicles. Extensive model tests have been conducted with various propeller types and the results have been systematically categorized in methodical series charts. These charts contain curves of certain dimensionless coefficients that uniquely

characterize the performance (and geometry) of a particular screw propeller. (Figure 3). The most important dimensionless quantities are defined below.

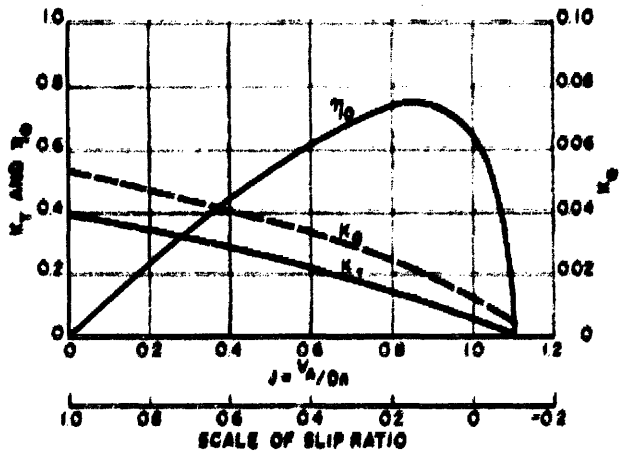


Figure 3 Typical Wageningen B-Series propeller chart (PNA vol. II)

2.2.1.1 Propeller Parameters

(a) Thrust coefficient:

$$K_T = \frac{T}{\rho n^2 D^4} \tag{2.5}$$

where T is the propeller thrust, D is the diameter of the disk, n the rotational speed and ρ the density of the water.

(b) Torque coefficient:

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{2.6}$$

where Q is the propeller torque

(c) Advance ratio:

$$J = \frac{V_A}{nD} \tag{2.7}$$

where V_A is the speed of advance, namely the forward speed of the propeller relative to the water. The speed of advance is lower than the ship's speed. At the location of the

propeller (i.e. inside the ship's wake) the water has a forward speed, tending to follow the ship's motion. As a result the propeller is advancing at a lower speed relative to the water than the rest of the ship.

Although the flow field behind the ship is highly turbulent and the velocity differs from point to point, an average measure of the wake effect, the *wake fraction*, is defined as:

$$w = \frac{V - V_A}{V} \tag{2.8}$$

where V is the ship's speed and V_A the speed of advance. Details on spatial and temporal variations of the hull wake can be found in Breslin and Andersen (1994).

(d) Open water efficiency

$$\eta_o = \frac{TV_A}{2\pi n Q_o} \tag{2.9}$$

where Q_o is the propeller torque in the open water (i.e. in the absence of the ship's hull).

The action of the propeller behind the ship has also an effect on total resistance. If the propeller is visualized as an actuator disk (van Manen and van Oossanen, 1988) application of momentum balance shows that the flow is accelerated downstream from the disk. This increase in velocity results in a decrease in pressure as can be seen from Bernoulli equation. The pressure drop in the back of the ship combined with the stagnation pressure in the bow yields an "augmented" resistance which would otherwise not occur if the ship was being towed. To quantify the additional thrust required to overcome the augmented resistance the *thrust deduction factor* is defined (Woud and Stapesma, 2002):

$$t = \frac{k_p \cdot T - R}{k_p \cdot T} \tag{2.10}$$

where k_p is the number of propellers, T the thrust produced by one propeller and R the towing resistance of the ship.

(e) Expanded Area Ratio

$$EAR = \frac{A_E}{A_o} = \frac{A_E}{\frac{\pi D^2}{4}} \tag{2.11}$$

where A_E is the area of the blades outside the hub, expanded in a plane, and A_o is the area of the circle defined by the tip circumference

(f) Pitch Ratio

$$PR = \frac{P}{D} \tag{2.12}$$

where D is the propeller disk diameter and P is the pitch as defined in Figure 4.

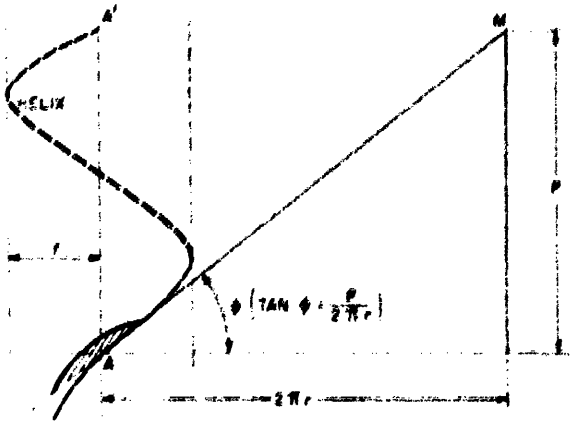


Figure 4 Helical geometry of propeller blade (PNA vol. II)

2.2.1.2 Optimum Propeller Selection

The procedure outlined here is followed later in the paper for the selection of a realistic propeller to be used in the optimization problem. It is based on the Wageningen B-series. Given the ship's nominal speed V and the maximum diameter D of the propeller (dictated by stern geometry constraints) the ratio $\frac{K_T}{J^2}$ is constant:

$$\frac{K_T}{J^2} = \frac{\frac{T}{\rho n^2 D^4}}{\frac{V^2(1-w)^2}{n^2 D^2}} = \frac{\frac{R(V)}{1-t}}{\rho D^2 V^2 (1-w)^2} = const. \tag{2.13}$$

The relation:

$$K_T = const. \cdot J^2 \tag{2.14}$$

defines a load curve on the graph of a specific propeller (identified by its number of blades z, V, and EAR)

The points where (2.14) intersects the K_T curve on the graph are projected straight upwards until they meet the respective η_o curves, as depicted in Figure 5. The point corresponding to maximum η_o uniquely determines K_T, K_Q, J, η_o and the propeller speed n. This procedure can be repeated for various combinations of z and EAR (i.e. different charts) until the absolute maximum is obtained. Usually, a cavitation limit is also imposed in this process, which limits the number of candidate EAR values under consideration.

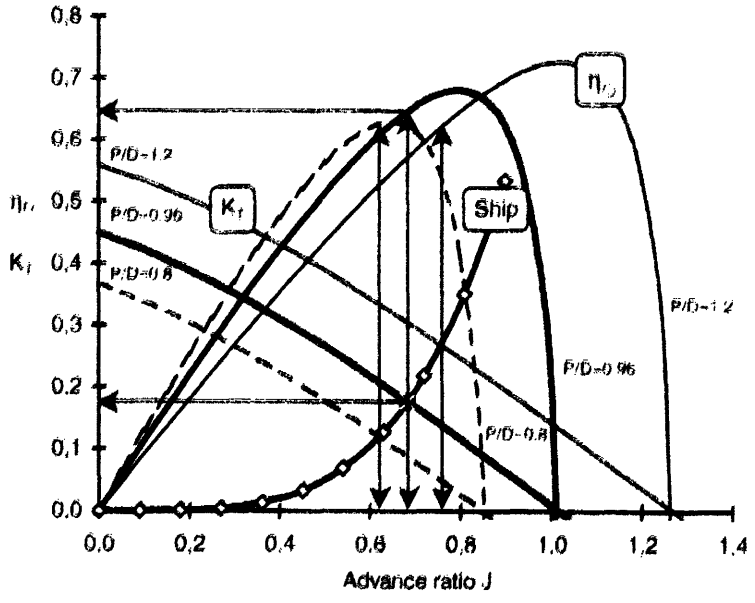


Figure 5 Graphical method for propeller selection (Stapersma and Woud, 2002)

It should be noted that in modern designs more advanced methods are used for the selection of propulsors. Utilization of lifting surface theory and computational methods is today the common practice in computer-based propeller design. Yet, the B-series

approach is a useful method that can at least provide the lower bound in the expected performance of a potential propulsor.

2.2.2 Propulsion Plant

In this section the propulsion chain is described for a typical single-shaft ship with one main engine as the prime mover. Extension to multiple-engine and/or multiple-shaft configuration is trivial.

Knowing the towing resistance R (by model testing or numerically) one can calculate the power required to tow the ship at speed V. This is the *effective power*:

$$P_E = EHP = R \cdot V \tag{2.15}$$

For a shelf-propelled ship the thrust power delivered by the propeller which moves at speed V_A in the water is the *thrust power*:

$$P_T = T \cdot V_A \tag{2.16}$$

The two powers defined above are not the same, owing to the presence of the wake and the augmentation of resistance. Both of these factors are due to the ship's hull. The efficiency introduced thereof is termed *hull efficiency*:

$$\eta_H = \frac{P_E}{P_T} = \frac{R \cdot V}{T \cdot V_A} = \frac{R \cdot V}{\frac{R}{1-t} V(1-w)} = \frac{1-t}{1-w} \tag{2.17}$$

As discussed in 2.2.1.1 the propeller operates with a nominal efficiency η_o in open water conditions. This means that to get power P_T out of a propeller one has to rotate it in the water at a speed n (without the presence of the hull) applying a torque Q_o, such that the ratio of input and output powers defines the open water efficiency of equation (2.9):

$$\eta_o = \frac{P_T}{P_o} = \frac{TV_A}{2\pi n Q_o} \tag{2.18}$$

The fact that a propeller is always behind the ship's hull and not alone in the water implies that the required torque is actually M_P, not Q_o. The input power to the propeller is not the open water power P_o but rather the *propeller power*:

$$P_P = 2\pi n M_P \tag{2.19}$$

The associated efficiency is called *relative rotative efficiency*

$$\eta_R = \frac{P_o}{P_P} = \frac{Q_o}{M_P} \tag{2.20}$$

Although η_H and η_R are loosely defined as "efficiencies", they frequently assume values greater than 1.

In order to deliver the power P_P, the shaft has to overcome losses in struts, bearings and stern tubes. To do that, the shaft has to pick up more power from the propulsion plant, the *shaft power*:

$$P_S = SHP = 2\pi n M_S \tag{2.21}$$

Then, the *shaft efficiency* is defined as

$$\eta_S = \frac{P_P}{P_S} = \frac{M_P}{M_S} \tag{2.22}$$

If a gearbox is present it reduces the rotational speed and increases the output torque from that of the engine (M_B) to that of the shaft (M_S). This function is performed with a *gearbox efficiency*

$$\eta_{GB} = \frac{P_S}{P_B} = \frac{2\pi n M_S}{2\pi n_e M_B} = \frac{1}{\lambda} \cdot \frac{M_S}{M_B} \tag{2.23}$$

where the engine's break power is defined as

$$P_B = BHP = 2\pi n_e M_B \tag{2.24}$$

and the reduction ratio as the ratio of input to output speed:

$$\lambda = \frac{n_e}{n} \tag{2.25}$$

The power flow and the above definitions of various efficiencies are shown schematically in Figure 6.

Apart from what is discussed above, there is also an efficiency associated with the conversion of the fuel's chemical energy into heat and an efficiency in the conversion of heat into mechanical work at the engine shaft. In most cases these efficiencies are incorporated in a parameter called *specific fuel consumption (sfc)*. It is provided by the engine manufacturer usually in g/kWh or lb/HPh and corresponds to the fuel consumption rate per unit of break power. For a diesel engine this parameter is roughly constant in a wide operating range.

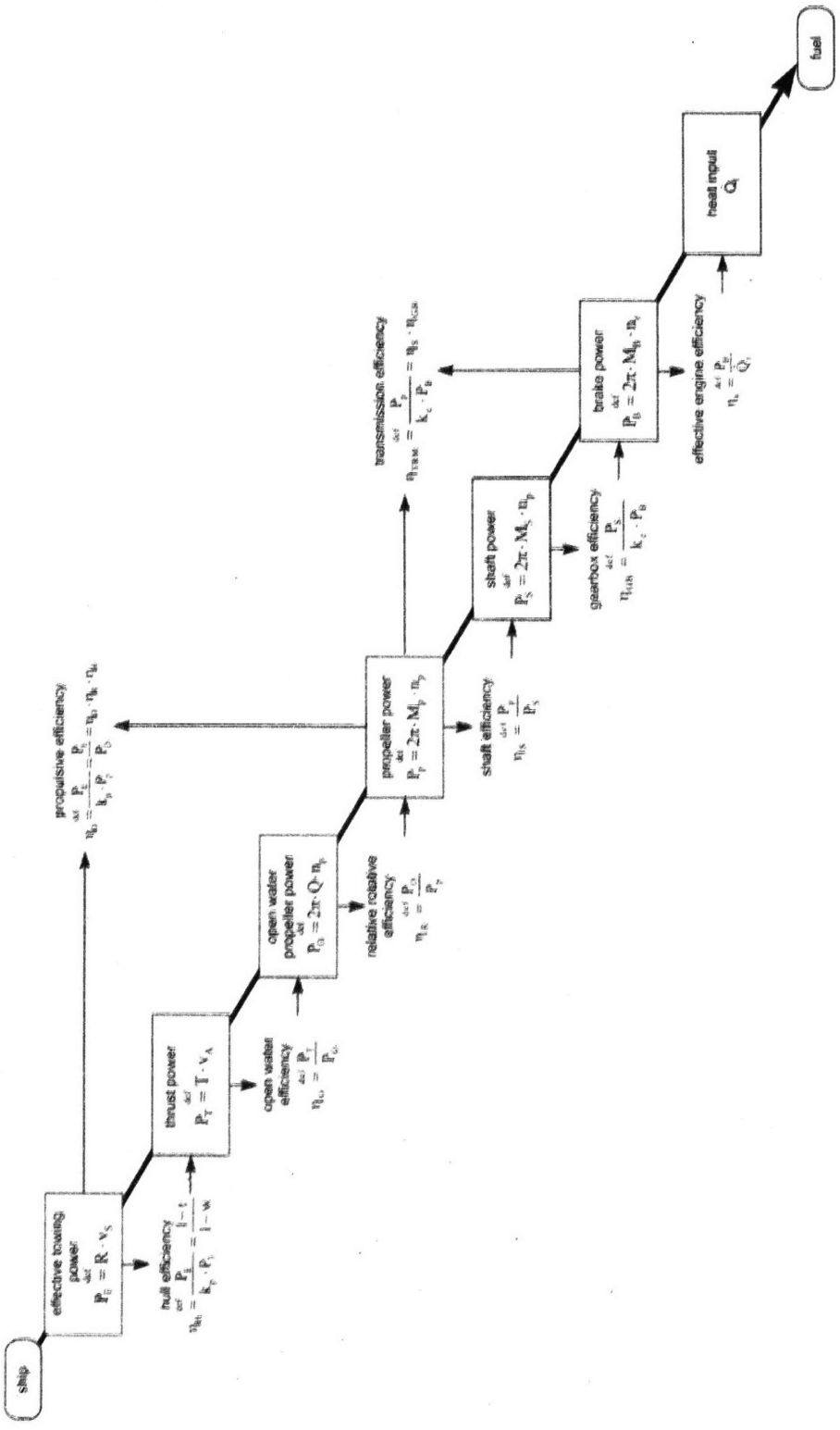


Figure 6 Schematic of the power chain of a marine propulsion plant (Stapersma and Woud, 2002)

2.3 Ship Dynamics in Waves

2.3.1 Plane Progressive Waves

Potential flow theory provides the framework for the mathematical description of regular water waves. Assume an earth fixed coordinate system $Oxyz$, where the plane $z = 0$ coincides with the free surface at rest (Figure 7).

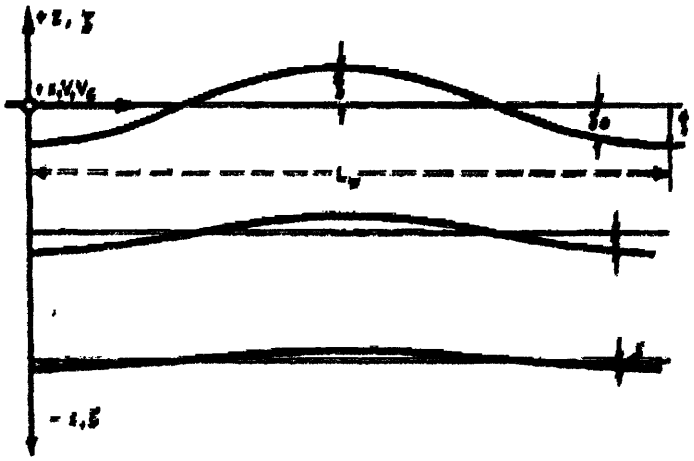


Figure 7 Reference frame and geometry of regular waves (PNA vol. III)

Ideal flow theory postulates the existence of a scalar velocity potential Φ such that

$$\mathbf{v} = \nabla\Phi \tag{2.26}$$

and the absence of vorticity (irrotational flow)

$$\nabla \times \mathbf{v} = \mathbf{0} \tag{2.27}$$

Application of conservation of mass yields Laplace equation in Φ:

$$\nabla \cdot \mathbf{v} = 0 \Rightarrow \nabla \cdot \nabla\Phi = 0 \Rightarrow \nabla^2\Phi = 0 \tag{2.29}$$

Equation of momentum conservation is just the Euler equation for inviscid flow:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla P + \mathbf{g} \tag{2.30}$$

where ρ is the water density, $\mathbf{g}$ the gravitational acceleration and P the fluid pressure. Manipulation of (2.30) using vector calculus yields the well known Bernoulli equation:

$$\frac{\partial \Phi}{\partial t}+\frac{1}{2}\nabla\Phi\cdot\nabla\Phi+\frac{P}{\rho}+gz=const. \tag{2.31}$$

Let $\zeta(x,y)$ be the wave elevation, namely the z position of the free surface. The free surface kinematic boundary condition states that a particle on the free surface stays there at all times (continuum hypothesis). Mathematically, following a fluid particle on $z=\zeta$, the total derivative of $z-\zeta$ must vanish, i.e.:

$$\frac{D}{Dt}(z-\zeta)=0\Rightarrow\left(\frac{\partial}{\partial t}+\mathbf{v}\cdot\nabla\right)(z-\zeta)=0 \quad \text{on } z=\zeta \tag{2.32}$$

The dynamic boundary condition on the free surface requires that the pressure there be atmospheric:

$$P_{z=\zeta}=P_a=0\Rightarrow-\rho\left(\frac{\partial\Phi}{\partial t}+\frac{1}{2}\nabla\Phi\cdot\nabla\Phi+gz\right)_{z=\zeta}=0 \tag{2.33}$$

The field equation (2.29) together with the boundary conditions (2.32) and (2.33) formulate the nonlinear wave boundary value problem (BVP).

Owing to the large gravitational acceleration the wave slopes in most cases are small. This assumption allows for the linearization of the free surface conditions around $z=0$. Retaining only linear terms the BVP becomes:

$$\nabla^2\Phi=0, \qquad z\le 0$$

$$\frac{\partial\Phi}{\partial t}+g\zeta=0, \quad z=0$$

$$\frac{\partial\zeta}{\partial t}=\frac{\partial\Phi}{\partial z}, \qquad z=0$$

$$|\nabla\Phi|\to 0, \qquad |z|\to\infty$$

Assuming deep water the solution of the BVP yields:

$$\Phi(x,y,z,t) = A\Re e\left\{\frac{ig}{\omega}e^{kz-i(kx-\omega t)}\right\} \tag{2.34}$$

The complete proof can be found in Newman (1977). If the direction of propagation is at an arbitrary angle with respect to the x-axis, (2.34) is modified as:

$$\Phi(x,y,z,t) = A\Re e\left\{\frac{ig}{\omega}e^{kz-i[k(x\cos\beta+y\sin\beta)-\omega t]}\right\} \tag{2.35}$$

The dynamic and kinematic boundary conditions can be combined into a single free surface condition

$$\frac{\partial^2\Phi}{\partial t^2}+g\frac{\partial\Phi}{\partial z}=0, \quad z=0 \tag{2.36}$$

from which the dispersion relation is derived:

$$\omega^2 = gk \tag{2.37}$$

where ω is the frequency of the plane wave, λ the wavelength and $k=\frac{2\pi}{\lambda}$ the wavenumber.

The equation for a plane progressive wave that has been assumed in the above solution is

$$\zeta(x,t) = A\cos(kx-\omega t) \tag{2.38}$$

2.3.2 Linear Wave-Body Interaction Theory

Within linear theory the ship can be modeled as a linear time-invariant (LTI) mechanical oscillator with the six degrees of freedom shown in Figure 8.

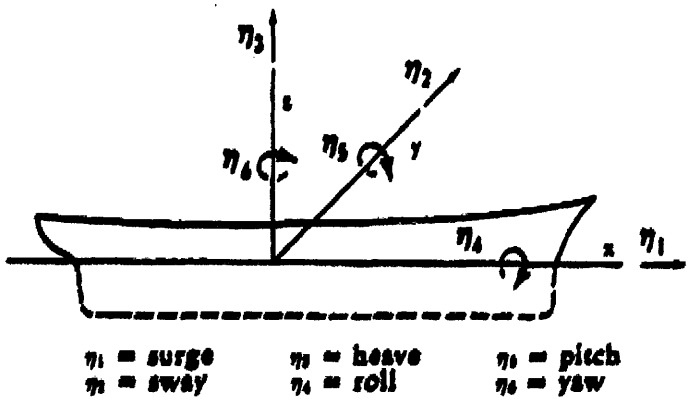


Figure 8 Reference frame and ship modes of motion (PNA vol. III)

In analogy to the simple mass-spring-dashpot dynamical system, the ship is subjected to:

(a) Inertial forces acting on the solid mass and hydrodynamic added mass
(b) Damping forces (arising from the energy dissipation during the generation of waves)
(c) Hydrostatic restoring forces
(d) External excitation forces

The general equations of motion in the time domain can be written as

$$\sum_{j=1}^{6}\left[\left(M_{ij}+A_{ij}\right)\ddot{\eta}_j(t)+B_{ij}\dot{\eta}_j(t)+C_{ij}\eta_j(t)\right]=F_i(t)\quad i=1,\ldots,6 \tag{2.39}$$

where M_{ij} and A_{ij} are the mass and added mass tensors respectively, B_{ij} and C_{ij} the damping and hydrostatic stiffness tensors respectively and $F_i(t)$ the (generalized) excitation force. Note that the modes of motion η_j can be either linear displacements or rotations and F_i can be a force or a moment. Therefore, all the coefficient tensors are dimensionally inhomogeneous.

It should be made clear at this point that (2.39) is not valid for arbitrary time dependence of the response but only for harmonic η (Newman, 1977). It turns out that this is not overly restrictive, within linear theory, as it is possible to linearly superimpose harmonic responses to obtain a general function of time. In fact, this observation

motivates the frequency domain formulation (section 2.3.3) which is the basis of SWAN1.

By virtue of Wiener-Khinchine theorem a harmonic input in a LTI system will result in a harmonic output with a phase shift. In the case of an incident plane wave the input will be a harmonic exciting force and the output will simply be the sinusoidal response of the floating structure. The exciting force consists of two components: one due to the incident wave impacting the structure (Froude-Krylov force) and the other due to the waves scattered away from the structure itself (diffraction force).

The Froude-Krylov force is calculated by direct integration of the linear pressure P around the mean wetted surface of the body:

$$F_j^{FK} = \iint_{\bar{S}_B} P n_j dS \tag{2.40}$$

where n_j is the jth component of the unit normal vector pointing into the body

$$n_j = \begin{cases} n_j & j = 1,2,3 \\ \left|\mathbf{x} \times \mathbf{n}\right|_{j-3} & j = 4,5,6 \end{cases} \tag{2.41}$$

and

$$P = -\rho \frac{\partial \Phi_I}{\partial t} \tag{2.42}$$

with the incidence potential Φ_I given by (2.35).

The force is clearly harmonic and can be expressed as:

$$F_j^{FK} = \Re e\left\{X_j^{FK}(\omega) e^{i\omega t}\right\} \tag{2.42}$$

Similar relations hold for the diffraction force, but now the expression of pressure in (2.42) should involve the diffraction potential Φ_D. To compute Φ_D one has to solve the following BVP:

$$\nabla^2\Phi_D = 0 \qquad z \le 0 \qquad \text{(field equation)}$$
$$\mathbf{n}\cdot\nabla\Phi_D = -\mathbf{n}\cdot\nabla\Phi_I \qquad \text{on } \overline{S}_B \qquad \text{(body boundary condition)}$$
$$\left(\frac{\partial^2}{\partial t^2} + g\frac{\partial}{\partial z}\right)\Phi_D = 0 \qquad \text{on } z = 0 \qquad \text{(free surface condition)}$$

In addition to the above set of equations Φ_D must satisfy the condition that at infinity the waves must be outgoing. The total exciting force will be the sum:

$$F_j = \Re e\left\{X_j(\omega)e^{i\omega t}\right\} = \Re e\left\{[X_j{}^{FK}(\omega) + X_j{}^{D}(\omega)]e^{i\omega t}\right\} \tag{2.43}$$

The response will be of the form

$$\xi_j = \Re e\left\{\Xi_j(\omega)e^{i\omega t}\right\} \tag{2.44}$$

where

$$\Xi_j(\omega) = \left|\Xi_j(\omega)\right|e^{i\theta_j} \tag{2.45}$$

is the complex amplitude of mode j.

Substituting the complex expressions in (2.43), (2.44) into the equation of motion (2.39):

$$\sum_{j=1}^{6}\left[-\omega^2\left(M_{ij} + A_{ij}(\omega)\right) + i\omega B_{ij}(\omega) + C_{ij}\right]\Xi_j(\omega) = X_i(\omega) \qquad i = 1,\ldots,6 \tag{2.46}$$

Added mass A_{ij} and damping coefficients B_{ij} are computed numerically by SWAN1 (Sclavounos, 1996).

2.3.3 Seakeeping with Forward Speed

Assume a ship-fixed coordinate system $Oxyz$. The plane $z = 0$ coincides with the water surface and, together with the ship, the coordinate system is translating with forward speed U in the x-direction as in Figure 9. Harmonic wave of frequency ω_o propagates in a direction β relative to the x-axis. The moving reference frame is related to the earth-fixed $OXYZ$ via the Galilean transformation (Sclavounos, 2007)

$$X = x + Ut$$
$$Y = y \tag{2.47}$$
$$Z = z$$

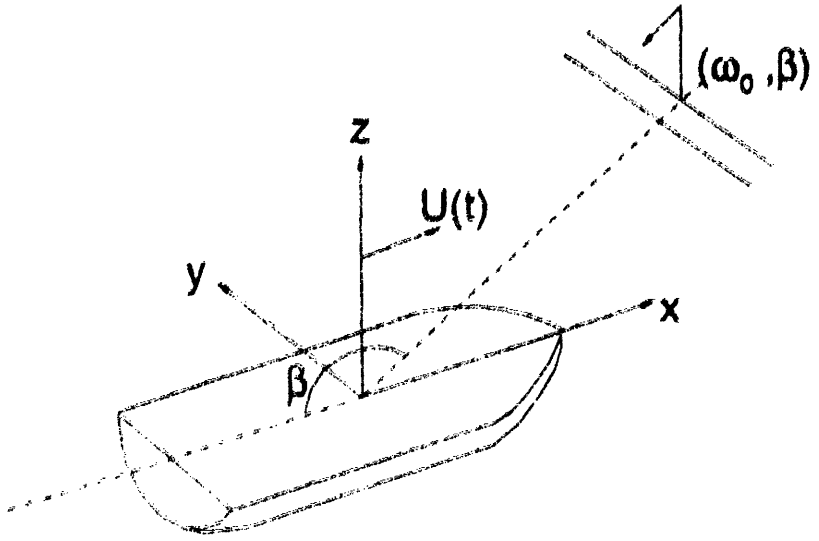


Figure 9 Coordinate system for the seakeeping problem (Sclavounos, 1996)

Substituting (2.47) in (2.35) the wave potential is expressed in the moving frame

$$\Phi(x,y,z,t) = A\Re e\left\{\frac{ig}{\omega_o} e^{kz - ik(x\cos\beta + y\sin\beta)} e^{-i(kU\cos\beta - \omega_o)t}\right\} \tag{2.48}$$

Equation (2.48) assumes the exact same form as (2.35). The difference lies in the frequency of the time harmonic term. It is no longer the absolute wave frequency ω_o, but the *encounter frequency*

$$\omega = \omega_o - kU\cos\beta = \omega_o - \frac{\omega_o^2}{g} U\cos\beta \tag{2.49}$$

It follows that the wave elevation can now be expressed relative to the moving frame as

$$\zeta(x,y,t) = \Re e\left\{Ae^{-i[k(x\cos\beta + y\sin\beta) - \omega t]}\right\} \tag{2.50}$$

The total velocity potential can be decomposed as:

$$\Phi(\mathbf{x},t) = \overline{\Phi}(\mathbf{x},t) + \varphi(\mathbf{x},t) \tag{2.51}$$

The first term on the right-hand side represents the steady flow potential. It can be further decomposed into a basis flow $\overline{\phi}$ and a disturbance flow ψ:

$$\overline{\Phi}(\mathbf{x},t) = \overline{\phi}(\mathbf{x},t) + \psi(\mathbf{x},t) \tag{2.52}$$

The decomposition facilitates the linearization of the nonlinear steady flow around the basis flow, assuming that ψ is small compared to $\overline{\phi}$ (Nakos and Sclavounos, 1990).

In SWAN1 the basis flow potential is implemented using the double-body flow representation. In this setting, the floating body is augmented by its image about the horizontal plane (free surface). The free surface is replaced by a rigid wall, on which:

$$\frac{\partial \overline{\phi}}{\partial z} = 0 \ , \qquad z = 0 \tag{2.53}$$

Application of the boundary condition yields:

$$\frac{\partial \psi}{\partial n} = 0 \Rightarrow \nabla \psi \cdot \mathbf{n} = 0 \qquad \text{on } \overline{S}_B \tag{2.54}$$

Together with the free surface condition, the field equation and the radiation condition the formulated BVP can be numerically solved. The solution $\overline{\Phi}$ represents the potential of a steady, outgoing wave pattern known as the Kelvin wake. The *Theory Manual* of SWAN (Sclavounos, 1996) provides information on the double-body linearization and its ramifications.

The second term on the right-hand side of (2.51) is expressed as:

$$\varphi(\mathbf{x},t) = \Re e\left\{\tilde{\varphi}(\mathbf{x}) e^{i\omega t}\right\} \tag{2.55}$$

where ω is the frequency of encounter. This expression is the cornerstone of the frequency domain formulation of the linear seakeeping problem. It is assumed that under the incidence of monochromatic plane waves of frequency ω_o the ship undergoes small oscillations at frequency ω. The complex potential $\tilde{\varphi}$ is a superposition of the incident wave potential $\tilde{\varphi}_I$, the diffraction potential $\tilde{\varphi}_D$ and the radiation potentials $\tilde{\varphi}_j$, j=1,...,6 for all modes of motion

$$\tilde{\varphi} = \tilde{\varphi}_I + \tilde{\varphi}_D + \sum_{j=1}^{6} \tilde{\varphi}_j \tag{2.56}$$

Upon solving the relevant BVP's for all the (complex) potentials in (2.56), pressure can be calculated from Bernoulli equation. Then, by integration over the hull mean wetted surface $\overline{S}_B$, the forces can be obtained. An outline of the procedure follows. It starts with the calculation of the exciting forces (both Froude-Krylov and diffraction components).

$$\tilde{P} = -\rho\left(i\omega - U\frac{\partial}{\partial x}\right)\left(\tilde{\varphi}_I + \tilde{\varphi}_D\right) \tag{2.57}$$

$$X_i = \iint_{\overline{S}_B} \tilde{P} n_i dS \tag{2.58}$$

These forces are balanced by inertial, damping and hydrostatic restoring forces by virtue of Newton's law. The equations of motion are similar to those in the case of zero forward speed:

$$\sum_{j=1}^{6}\left[-\omega^2\left(M_{ij} + A_{ij}(\omega)\right) + i\omega B_{ij}(\omega) + C_{ij}\right]\Xi_j(\omega) = X_i(\omega) \qquad i = 1,...,6 \tag{2.59}$$

In this case, however, ω stands for the encounter frequency, not the absolute frequency of the ambient wave.

The added mass and damping tensors A_{ij} and B_{ij} are computed after solving the radiation BVP. Once the radiation potentials are known, the complex amplitudes of the respective forces are calculated:

$$\tilde{F}_i = \iint_{S_B} \tilde{P} n_i dS = \iint_{S_B} \hat{p}_j \Xi_j n_i dS = \Xi_j \left(\iint_{S_B} \hat{p}_j n_i dS \right) \tag{2.60}$$

where

$$\hat{p}_j = -\rho \left(i\omega - U \frac{\partial}{\partial x} \right) \hat{\varphi}_j \tag{2.61}$$

and

$$\tilde{\varphi}_j = \Xi_j \hat{\varphi}_j \tag{2.62}$$

Added mass and damping components are obtained by

$$\iint_{S_B} \hat{p}_i n_j dS = \omega^2 A_{ij}(\omega) - i\omega B_{ij}(\omega) \tag{2.63}$$

2.3.4 Derived Responses

A stationary sea state can be represented as a superposition of many monochromatic plane waves. In the frequency domain this representation is condensed into a wave spectrum $S(\omega)$. Properties and statistics of waves are the subject of chapter 3. For now, the spectral density $S(\omega)$ should be visualized as a continuous distribution of energy over the frequencies of the individual wave components contained in a sea state.

Linear Systems Theory predicts that the response of a LTI system excited by a harmonic input signal will also be harmonic. For random signals, if the input is normally distributed (Gaussian) the output signal will also be Gaussian. This very powerful property of linear systems is extremely useful in predicting ship responses in waves. This is so, because waves are often modeled as a Gaussian, zero-mean, random process. As a

result, ship responses can be predicted with very good accuracy (when the assumptions of linearity hold).

Suppose that by using the methods of sections 2.4.2, 2.4.3 (frequency domain methods) the complex modes of motion given by (2.45) are known. For a given description of a sea state $S(\omega)$ the output seakeeping quantities of interest are calculated from the Wiener-Khinchine spectral relations:

$$\sigma_j^2 = \int_0^\infty S_\zeta(\omega)\left|\Xi_j(\omega)\right|^2 d\omega \tag{2.64}$$

where σ_j is the variance of the j^{th} mode of motion. In problems with forward speed $S_\zeta(\omega)$ is the encounter spectrum and $\Xi_j(\omega)$ is given at the encounter frequency

$$\omega = \left|\omega_o - \frac{\omega_o^2}{g} U\cos\beta\right| \tag{2.65}$$

For dimensional consistency the modes $\Xi_j(\omega)$ need to be non-dimensional, namely divided by the appropriate quantity, e.g. the wave amplitude or slope (equation (2.71) below). The integral (2.64) should be evaluated over the absolute frequency ω_o. For this, the Jacobian of the transformation is needed:

$$\frac{d\omega}{d\omega_o} = \begin{cases} 1 - 2U\dfrac{\omega_o}{g}\cos\beta & \omega_o < \dfrac{g}{U\cos\beta} \\ 2U\dfrac{\omega_o}{g}\cos\beta - 1 & \omega_o > \dfrac{g}{U\cos\beta} \end{cases} \tag{2.66}$$

and

$$\sigma_j^2 = \int_0^\infty \frac{d\omega}{d\omega_o} S_\zeta\left(\left|\omega_o - \frac{\omega_o^2}{g}U\cos\beta\right|\right)\left|\Xi_j\left(\left|\omega_o - \frac{\omega_o^2}{g}U\cos\beta\right|\right)\right|^2 d\omega_o \tag{2.67}$$

The seakeeping responses mentioned above refer to simple events, namely the 6-degrees-of-freedom motion of the origin (in the ship–fixed frame). It is often desirable to obtain

the response of complex seakeeping events, called composite events. An example of such an event is the relative motion of a point $P(x_P, y_P, z_P)$ in the bow of the ship (Figure 10). The remainder of this section is devoted to the calculation of the variance of this composite event. All the quantities cited henceforth are complex, unless otherwise stated.

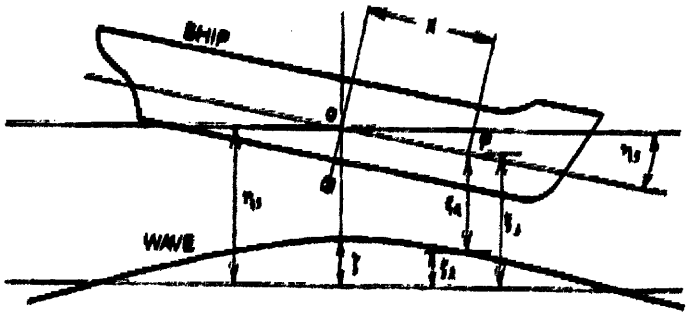


Figure 10 Relative bow motion (PNA vol. III)

Incident monochromatic wave of amplitude A and frequency ω_o (absolute) impacts the ship coming from direction β. The ship is moving in the x-direction with constant speed U. The wave elevation at point P in the bow is

$$\tilde{\zeta}(x_P, y_P, t) = Ae^{i\omega t - ik(x_p \cos\beta + y_p \sin\beta)} \tag{2.68}$$

where ω is the encounter frequency and

$$k = \frac{\omega_o^{\,2}}{g} \tag{2.69}$$

The longitudinal motions at the origin are

$$\tilde{\xi}_j = \Xi_j(\omega)e^{i\omega t} \quad , \quad j = 1,3,5 \tag{2.70}$$

SWAN1 computes Ξ_j in the form of Response Amplitude Operators:

$$RAO_j(\omega) = \frac{\Xi_j(\omega)}{A} \tag{2.71}$$

The relative motion at point P is:

$$\tilde{\xi}_r = \tilde{\xi}_3 - x_P\tilde{\xi}_5 - \tilde{\zeta} \Rightarrow \Xi_r = \Xi_3 - x_P\Xi_5 - Ae^{-ik(x_p\cos\beta + y_p\sin\beta)} \tag{2.72}$$

Normalizing with the wave amplitude:

$$RAO_r(\omega) = RAO_3(\omega) - x_P RAO_5(\omega) - e^{-i\frac{\omega_o^2}{g}(x_p\cos\beta + y_p\sin\beta)} \tag{2.73}$$

The quantities on the right-hand side are all known, so $RAO_r(\omega)$ can be calculated. Some caution should be exercised here as this operator is expressed as a function of the encounter frequency. Establishing a relationship:

$$RAO^*_r(\omega_o) = RAO_r\left(\left|\omega_o - \frac{\omega_o^2}{g}U\cos\beta\right|\right) \tag{2.74}$$

the requisite variance can be obtained as an integral over ω_o:

$$\sigma_r^2 = \int_0^\infty S_\zeta(\omega_o)\left|RAO^*_r(\omega_o)\right|^2 d\omega_o \tag{2.75}$$

2.4 Added Resistance in Waves

This section deals with the additional drag experienced by a ship sailing in waves. This phenomenon is of vital importance in the minimum-fuel optimization problem. The dependence of added resistance on the sea state and its effect on fuel consumption render it a decisive factor in the determination of the optimal route.

Added resistance is a second order effect. It is the mean value of the total resistance in waves minus the calm water resistance (Sclavounos and Nakos, 1993). It can be calculated by direct integration of the pressure around the hull. The pressure is obtained by the methods described in the previous sections only with the linear and second order terms retained in the expansion. So far, all the integrations have been performed over the mean position of the submerged hull. As explained in Faltinsen (1990), a second order expansion simply attempts to approximate the instantaneous position of the hull more accurately than the linear approach.

2.4.1 Calculation of Added resistance

Gerristma and Beukelman (1972) derived a method to determine added resistance by calculating the radiated energy of the pitching and heaving ship (PNA vol.III: Beck et al.,1989). They demonstrated good agreement of their theory with experimental results in regular head waves. In follow-up experiments Journée (report 0428, 1976) verified the validity of the theory for head waves. However, significant deviations were observed in following waves (Journée, report 0440, 1976). Faltinsen et al. (1980) derived a formula valid for all headings, but limited to small Froude numbers ($F_n \leq 0.2$) and blunt ship forms.

Clearly, reasonable accuracy is required for all headings and a range of speeds in order for the minimum-fuel optimization results to be meaningful. In this paper, added resistance results are obtained from SWAN1 which implements the direct pressure integration method. Details are provided in Sclavounos and Nakos (1993), a brief outline follows below.

The total velocity potential is given by (2.51). The hydrodynamic pressure is

$$P = -\rho\left(\frac{\partial \Phi}{\partial t} + \frac{1}{2}\nabla\Phi \cdot \nabla\Phi + gz\right) \tag{2.76}$$

The total resistance is the x-component of the hydrodynamic force obtained by integrating (2.76) around the (instantaneous) submerged hull:

$$R_T = -\iint_{S_B} P n_1 dS_B \tag{2.77}$$

Replacing Φ in (2.76) by the steady flow potential $\overline{\Phi}$ and integrating over the mean surface of the submerged body the effect of waves is eliminated.

$$\overline{P} = -\rho\left(\frac{\partial \overline{\Phi}}{\partial t} + \frac{1}{2}\nabla\overline{\Phi} \cdot \nabla\overline{\Phi} + gz\right) \tag{2.78}$$

$$R_C = -\iint_{\overline{S}_B} \overline{P}\overline{n}_1 dS_B \tag{2.79}$$

Obviously, (2.79) is the clam water resistance.

The added resistance is defined as the mean of the difference $R_T - R_C$. This difference is oscillatory, time dependent, and can be written as the sum of three components:

$$R_T - R_C = R_1 + R_2 + R_3 \tag{2.80}$$

$$R_1 = -\iint_{\bar{S}_B} \delta P \bar{n}_1 dS_B \tag{2.81}$$

$$R_2 = -\iint_{\bar{S}_B} \bar{P} \delta n_1 dS_B \tag{2.82}$$

$$R_3 = -\iint_{\delta S_B} \bar{P} \bar{n}_1 dS_B \tag{2.83}$$

In the above relations, δS_B is the correction to the mean wetted surface $\bar{S}_B$ to account for the ship's motions, δP is the correction to the steady pressure P at some rigid point on the hull and δn_1 is the difference between the x-component of the unit normal vector at the instantaneous and mean positions. By expanding (2.80) in a Taylor series the quantities δS_B, δP, δn_1 appear explicitly in terms of $\bar{\Phi}$, φ and their gradients evaluated at the mean positions of the hull and waterline. Since the latter quantities are available from the solution of the linear seakeeping problem, added resistance can be calculated

$$R_w(\omega, \beta) = \overline{R_1 + R_2 + R_3} \tag{2.84}$$

SWAN1 gives the added resistance in the form of an operator normalized by the square of the (monochromatic) wave amplitude:

$$ARO(\omega, \beta) = \frac{R_w(\omega, \beta)}{A^2} \tag{2.85}$$

For irregular seas characterized by a spectrum $S(\omega,\theta)$ the mean added resistance is given by

$$\overline{R}_w = 2\int_0^{\infty} d\omega \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} d\theta S(\omega,\theta)\cdot ARO \tag{2.86}$$

which follows from the definition (section 3.1)

$$S(\omega_o,\theta)\delta\omega_o\delta\theta = \frac{1}{2}A^2 \tag{2.87}$$

CHAPTER

3

STATISTICS & MODELING OF OCEAN WAVES

The main topic of this chapter is the wave spectrum. Section 3.1 defines it and presents its statistical properties and physical implications. Section 3.2 deals with the physics of wave generation and the numerical models which provide forecasts for the evolution of the spectrum. The final section introduces the application of wave prediction in the routing problem.

3.1 Wave Statistics

3.1.1 Statistical Representation of Ocean Waves

The description of a monochromatic wave of frequency ω and amplitude A traveling in the direction θ is given by

$$\zeta(x,y,t) = A\cos[k(x\cos\theta + y\sin\theta) - \omega t + \varepsilon] \tag{3.1}$$

where ε is the phase and k the wavenumber. A superposition of infinitely many such waves yields a description for the sea surface

$$\zeta(x,y,t) = \sum_j A_j \cos[k_j(x\cos\theta_j + y\sin\theta_j) - \omega_j t + \varepsilon_j] \tag{3.2}$$

where the frequencies and wavenumbers are bound together by the (deep water) dispersion relation

$$k_j = \frac{\omega_j^2}{g} \tag{3.3}$$

If the amplitudes A_j, the frequencies ω_j, the directions θ_j and the phases ε_j are random variables then the wave profile obtained by (3.2) is essentially a random process. This

representation was developed by Rice in communication engineering (Ochi, 1998). Two different wave records of the form (3.2) are shown in Figure 11. The wave records are taken at different fixed points in the ocean (e.g. a probe at a particular location). So, they can be considered as two different realizations ${}^{1}\zeta(t), {}^{2}\zeta(t)$ of the random process.

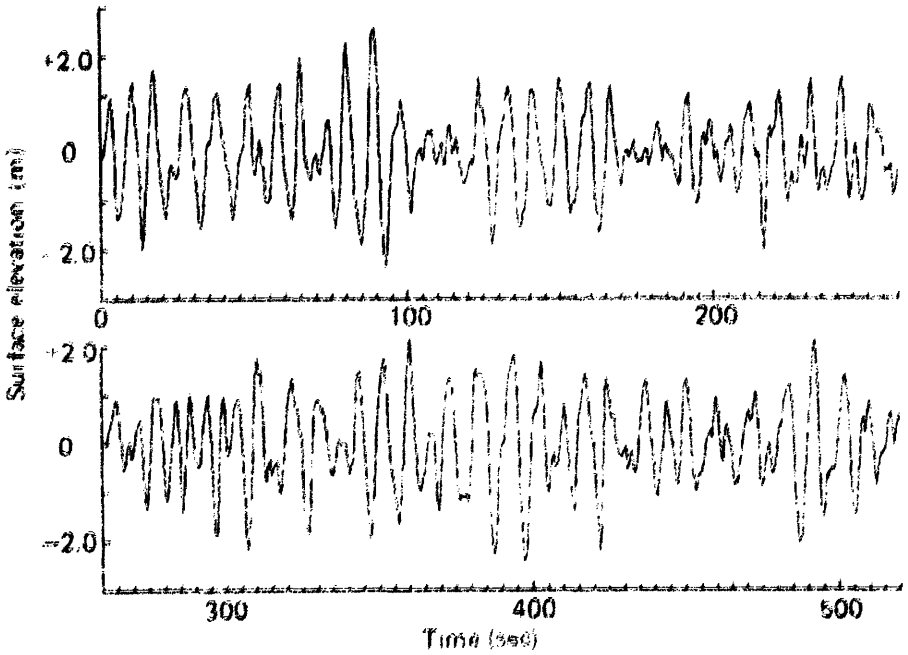


Figure 11 Example of two different wave records (Tucker and Pitt, 2001)

Many different realizations (like the two shown in the above figure) comprise the ensemble of the process. Waves possess the ergodic property, which allows the extraction of statistics not from the ensemble (as should be formally done) but from a single realization by time averages (Ochi, 1998). If all the statistics of a process are time invariant then the process is *stationary* or *steady-state*. Ocean waves are only *weakly stationary* or *second order stationary* as the first two moments (namely the mean and the covariance) are time invariant. Hence, the covariance of the wave record, defined by

$$Cov[\zeta(t_k),\zeta(t_k+\tau)] = \frac{1}{N}\sum_{j=1}^{N}[{}^{j}\zeta(t_k)-\bar{\zeta}(t_k)][{}^{j}\zeta(t_k+\tau)-\bar{\zeta}(t_k+\tau)] \qquad (3.4)$$

is time invariant (i.e. depends only on the time shift τ) and equals the autocorrelation function, defined by

$$R(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \zeta(t)\zeta(t+\tau)dt \tag{3.5}$$

Central Limit Theorem guarantees that the sum of many identically distributed, independent random variables (or processes) is itself a Gaussian random variable (or process) with mean and variance which are the sums of the means and variances of the individual constituent variables. In this sense, the process defined in (3.2) can be treated as normally distributed with zero mean. From equations (3.4) and (3.5), for $\tau = 0$ the autocorrelation represents the variance of the wave record (since $\overline{\zeta} = 0$). It also represents the time average of the wave energy

$$R(0) = \overline{W} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \zeta^2(t)dt \tag{3.6}$$

Making use of Parseval's theorem for Fourier transform pairs, Ochi (1998) defines the spectral density function of random waves $\zeta(t)$ as:

$$S(\omega) = \lim_{T\to\infty} \frac{1}{4\pi T} \int_{-T}^{T} |Z(\omega)|^2 d\omega \tag{3.7}$$

where $Z(\omega)$ is the Fourier transform of $\zeta(t)$:

$$Z(\omega) = \int_{-\infty}^{\infty} \zeta(t)e^{-i\omega t}dt \tag{3.8}$$

The Wiener-Khinchine theorem states that, under the assumption of weak stationarity of random waves, the autocorrelation and spectral density are Fourier transform pairs, namely:

$$S(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} R(\tau)e^{-i\omega\tau}d\tau \tag{3.9}$$

$$R(\tau) = \frac{1}{2} \int_{-\infty}^{\infty} S(\omega)e^{i\omega\tau}d\omega \tag{3.10}$$

$R(\tau)$ is by definition a real function. It is also an even function (Grimmet and Stirzaker, 2001), so that (3.10) can be written as

$$R(\tau) = 2\,\mathrm{Re}\left\{\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)e^{i\omega\tau}d\omega\right\} = \int_{0}^{\infty} S(\omega)\cos\omega\tau d\omega \tag{3.11}$$

which implies that the area under the spectrum is equal to the average energy and also the variance of the wave record

$$R(0) = \int_{0}^{\infty} S(\omega)d\omega = \bar{W} = Var[\zeta(t)] \tag{3.12}$$

The variance of the instantaneous wave elevation at a fixed point in space is by definition

$$Var[\zeta(t)] = \mathrm{E}[\zeta^2(t)] = \overline{\zeta^2(t)} \tag{3.13}$$

In the above relation the properties of ergodicity and zero mean ($\bar{\zeta} = 0$) have been invoked. Substituting (3.2) for a given point in space (e.g. the origin $x = y = 0$) it can be easily proven (Tucker and Pitt, 2001) that

$$\overline{\zeta^2(t)} = \sum_j \frac{1}{2}A_j^2 \tag{3.14}$$

The energy density of a water column with upper surface raised a distance ζ above the mean free surface level is given by

$$E = \frac{1}{2}\rho g\zeta^2 \tag{3.15}$$

The mean energy density is therefore

$$\bar{E} = \frac{1}{2}\rho g\overline{\zeta^2} \tag{3.16}$$

Combining equations (3.12)-(3.16) it can be deduced that

$$\bar{E} = \frac{\bar{E}}{\rho g} = \int_0^\infty S(\omega) d\omega = \sum_j \frac{1}{2} A_j^2 \tag{3.17}$$

Accounting for the directionality of the spectrum, (3.17) can be written as

$$\int_0^\infty \int_{-\pi}^{\pi} S(\omega, \theta) d\theta d\omega = \sum_j \frac{1}{2} A_j^2 \tag{3.18}$$

which is the integral of equation (2.87). The superposition of spectral components is depicted in Figure 12.

The n^{th} spectral moment is defined as

$$m_n = \int_0^\infty \omega^n S(\omega) d\omega \tag{3.19}$$

From this definition two useful spectral characteristics are extracted, which are later used extensively in the routing problem. Their definitions are provided below (Massel, 1996).

Significant Wave Height:

$$H_s = 4\sqrt{m_0} \tag{3.20}$$

where:

$$m_0 = \int_0^\infty S(\omega) d\omega = \overline{\zeta^2} \tag{3.21}$$

Mean Frequency:

$$\bar{\omega} = \frac{2\pi}{\bar{T}} = \frac{m_1}{m_0} \tag{3.22}$$

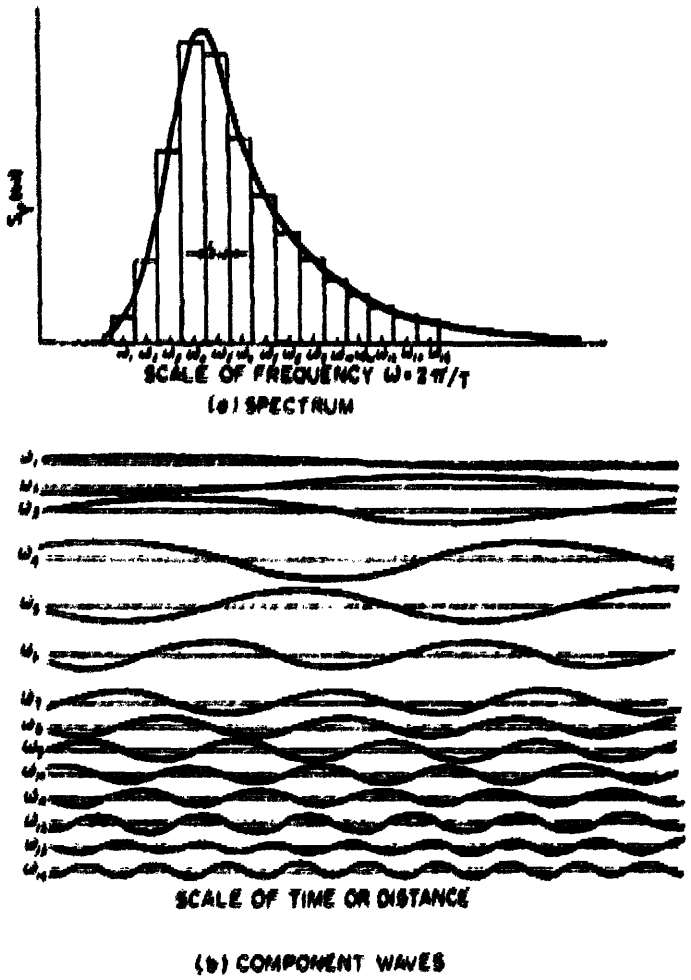


Figure 12 Synthesis of harmonic waves to construct wave spectrum (PNA vol. III)

3.2 Wave Modeling and Forecasting

3.2.1 Physics of Spectrum Evolution

The evolution of a wave spectrum in space and time is governed by the *energy balance equation:*

$$\frac{DS(\omega,\theta,\mathbf{x},t)}{Dt} = Q \tag{3.23}$$

In the above relation the operator D/Dt denotes the substantial derivative following a wave packet, and the right-hand side Q is the forcing term. The physics governing the evolution of the directional spectral density $S(\omega,\theta,\mathbf{x},t)$ is described in Massel (1996) and

Tucker and Pitt (2001). Derivations and discussion of numerical solutions can be found in Komen et al. (1994). The energy balance equation is the mathematical idealization of the physical law that describes the generation, evolution and spreading of wind-generated ocean waves.

Neglecting the influence of the bottom, the source term on the right can be decomposed into three separate terms, each representing a different forcing mechanism. These mechanisms are:

(1) Wind energy input
(2) Non-linear interaction among the frequency components
(3) Dissipation of energy due to wave breaking

The first mechanism manifests itself in the governing equation through the term Q_{in} which stands for the rate of energy transfer from the wind to the ocean. The second mechanism is captured by the term Q_{nl}, which is the rate of non-linear energy transfer among the wave components of different frequencies. Note that the representation (3.2) is strictly linear. Inclusion of cross terms in (3.2) (i.e. products of single components) results in steeper waves. The third mechanism participates through the term Q_{dis} which stands for the rate of energy dissipation. Energy losses are mostly due to viscosity and whitecapping (i.e. breaking of waves).

Neglecting the effect of currents, the energy balance equation (3.23) can be expressed, for deep water, as:

$$\left(\frac{\partial}{\partial t}+\mathbf{c_g}\cdot\nabla\right)S(\omega,\theta,\mathbf{x},t)=Q_{in}+Q_{nl}+Q_{dis} \tag{3.24}$$

where $\mathbf{c_g}$ is the group velocity.

In spherical coordinates the position on the spherical globe is described by the latitude ϕ and longitude λ. Equation (3.24) expressed in these coordinates becomes:

$$\frac{\partial S}{\partial t}+(\cos\phi)^{-1}\frac{\partial}{\partial\phi}\left(\dot{\phi}\cos\phi S\right)+\frac{\partial}{\partial\lambda}\left(\dot{\lambda}S\right)+\frac{\partial}{\partial\theta}\left(\dot{\theta}S\right)=Q_{in}+Q_{nl}+Q_{dis} \qquad (3.25)$$

in which:

$$\dot{\phi}=\frac{d\phi}{dt}=c_g R^{-1}\cos\theta \qquad (3.26)$$

$$\dot{\lambda}=\frac{d\lambda}{dt}=c_g \sin\theta\left(R\cos\phi\right)^{-1} \qquad (3.27)$$

$$\dot{\theta}=\frac{d\theta}{dt}=c_g \sin\theta\tan\phi R^{-1} \qquad (3.28)$$

where c_g is the group speed and R the radius of the earth.

3.2.2 Numerical Models and Wave Prediction

The solution of equation (3.25) with the associated boundary conditions is a challenging task. Historically, many theoretical models have been introduced to describe the solution, each one bearing certain assumptions. These models make extensive use of similarity arguments to simplify the analysis. Today, numerical models are available that solve the requisite boundary value problem directly. The state of the art in the field is the third generation models which are based on a better understanding of the underlying physics while at the same time take advantage of the computational power that was not available in the earlier models. The effort of developing an effective third generation model started with the WAM (Wave Modeling) project. All the relevant information about this work was published by Komen et al. in 1994.

One of the major contributions of the WAM model is that it captures the physics of nonlinear energy transfer better than the previous models. It can solve a more involved

version of equation (3.25) which accounts for bottom losses and current effects as well. Since the model is spectrum-based, the spectrum is readily available in the program output. This fact resolves the difficulties encountered in the past when wave data (like wave height, frequency and direction) had to be fitted to a standard spectrum representing the actual sea-state. The implementation of the model in global version was carried out by the European Center for Medium-Range Weather Forecasts (ECMWF).

This paper uses results from the ECMWF version of the model, the WAM cycle 4 (Emmanouil et al., 2007), courtesy of the University of Athens, Greece. The model solves the wave transport equation without any assumption on the spectral shape. Output is provided at 28 different frequencies and 24 wave directions. The spectral information is further processed and the final commercial output contains significant wave height, maximum expected wave height, mean and peak frequencies and mean wave direction. For swell, the output contains swell height, mean direction and mean frequency. These results are given as forecasts in 3-hour time intervals at each grid point on the global map. The spatial resolution of 0.5×0.5 degrees corresponds to an area of roughly 55×55 km. Sample ASCII outputs of this model can be found in Appendix B. The forecast is uploaded daily at 02:00 UTC and covers a total period of 108 hours.

3.3 Application to the Routing Problem

Accurate representation of the sea state is of foremost importance to minimum fuel navigation. The calculation of added resistance and ship responses in waves relies heavily on the wave spectrum, as shown in chapter 2. Although a state-of-the-art wave model such as WAM4 primarily calculates the spectrum, this output is not practical for commercial use. The vast size of the data files make it impossible to process as part of the optimization program. This predicament essentially eliminates the enormous advantage that third generation models offer. That is, the fact that they make no assumptions about the shape of the spectrum, they provide it explicitly and fully accounting for phenomena like stage of growth, directionality, remote storms etc.

In order to solve the routing problem, the data of Appendix B need to be fitted to separate spectra, one for the local storm and another for the swell. Which of the standard spectra is most suitable for each case is not at all straightforward. Tucker and Pitt (2001) argue in favor of "the universality of JONSWAP spectrum". Massel (1996) gives a swell spectrum proposed by Davidan in 1969. The latter exhibits good agreement with experimental measurements for

$$\frac{\bar{H}}{g\bar{T}^2} \leq 0.00125 \tag{3.29}$$

where $\bar{H}$ is the mean swell height and $\bar{T}$ the mean swell period. However, for most of the data corresponding to the North Atlantic forecast of April 5, 2008 (which is the sample forecast used in this paper for the optimal routing solution) the relation (3.29) is not satisfied.

For reasons of simplicity in coding and computational efficiency, the computer program developed in the present work fits the wave data to the two-parameter spectrum known as *the Bretschneider Spectrum* (Ochi, 1998):

$$S(\omega) = 0.278 \frac{\bar{\omega}^4}{\omega^5} \bar{H}^2 e^{-0.437(\bar{\omega}/\omega)^4} \tag{3.30}$$

where $\bar{\omega}$ is the mean frequency defined in (3.22) and the mean wave height $\bar{H}$ is related to the significant wave height in (3.20) via the relation

$$\frac{H_s}{\bar{H}} = 1.6 \tag{3.31}$$

PART II

OPTIMAL CONTROL

CHAPTER

4

DYNAMIC PROGRAMMING & OPTIMAL CONTROL

The fundamentals of hydrodynamics and wave modeling were covered in the previous chapters. To complete the background material needed for the routing problem, this chapter deals with the application of Dynamic Programming (DP) to Optimal Control Theory. Section 4.1 introduces the basic concepts and 4.2 presents the standard algorithm of discrete DP. Section 4.3 describes a more advanced numerical technique which is used in chapter 6 to solve the routing problem defined in chapter 5.

4.1 Dynamic Programming Formulation

4.1.1 Definitions

Consider a physical system the dynamics of which is modeled by the following differential equation:

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ &\mathbf{x}(t) \in X, \quad \mathbf{u}(t) \in U \\ &\mathbf{x}(t_0) = \mathbf{x}_0 \end{aligned} \tag{4.1}$$

In (4.1), the $n \times 1$ vector $\mathbf{x}(t)$ denotes the *state* of the system at time t and the $m \times 1$ vector $\mathbf{u}(t)$ denotes the *control history* up to time t, namely, the action that needs to be applied to the system to bring it from the initial state $\mathbf{x}_0$ to the current state $\mathbf{x}(t)$. The sets X and U are the *state space* and *control space* respectively. They represent the sets of all admissible states that the system can find itself at, and all the controls that are allowed to act on the system, respectively. Now consider a scalar quantity of the form

$$I = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt \tag{4.2}$$

where t_0, t_f are the (fixed) initial and final time respectively. The scalar I is a measure of an attribute of the system that will be optimized. It is known as the *performance index* or *cost function*. Solving an optimal control problem is synonymous to finding the control history $\mathbf{u}^*(t)$ which will minimize (or maximize) the index I. The resulting state history $\mathbf{x}^*(t)$ is the *optimal trajectory*.

The optimization of functionals like (4.2) is the subject of classical calculus of variation. In certain cases, some theoretical conditions of optimality guarantee a unique solution of the optimal control problem. These are discussed in Stengel (1994) and Kirk (2004). The most common one is the case when equation (4.1) is linear in the state $\mathbf{x}(t)$ and control $\mathbf{u}(t)$, and the performance index (4.2) is quadratic in the state and control. The problem has, then, a closed-form solution which is a linear feedback control law. This class of problems is called LQ (Linear – Quadratic) and is addressed by Anderson and Moore (2007).

Unfortunately, many problems do not admit an analytical solution. In fact, a great many of them do not even satisfy the optimality or existence and uniqueness conditions. This does not automatically mean that no optimal solution exists. In those cases, where the classical calculus of variations formulation falls short, the Dynamic Programming (DP) formulation introduced by Bellman in the 50's, appears more promising.

4.1.2 The Recurrence Relation and the HJB Equation

Borrowing the derivation from Kirk (2004), this section will arrive at the recurrence relation of DP. Consider the following constrained optimization problem:

$$\text{minimize} \qquad I = h(\mathbf{x}(t_f), t_f) + \int_0^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t)dt \qquad (4.3)$$

subject to the constraints

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \qquad (4.4)$$

$$\mathbf{x}(t) \in X \qquad (4.5)$$

$$\mathbf{u}(t) \in U \qquad (4.6)$$

with t_f fixed and $\mathbf{x}(0)$ prescribed.

This is a continuous-time optimal control problem. It will be converted to discrete to comply with the DP formulation. The time interval $0 \le t \le t_f$ is discretized into a sequence $0, \Delta t, 2\Delta t, ..., N\Delta t$. This is done by dividing the interval into N equally spaced subintervals of length Δt. The instants of time are now discrete:

$$0 \le t_k = k\Delta t \le t_f \ , \qquad k = 0,1,...,N \tag{4.7}$$

For simplicity, the actual time instants can be replaced by the integer k which signifies the beginning of the corresponding stage. The differential equation (4.4) is approximated by the difference equation

$$\frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t)}{\Delta t} \simeq \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{4.8}$$

$$\mathbf{x}(t+\Delta t) \simeq \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t \tag{4.9}$$

In shorthand, (4.9) can be rewritten as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k)\Delta t \tag{4.10}$$

or

$$\mathbf{x}(k+1) \triangleq \mathbf{a}(\mathbf{x}(k), \mathbf{u}(k), k) \tag{4.11}$$

The cost function is split into discrete intervals as well:

$$I = h(\mathbf{x}(N\Delta t), t_N) + \int_0^{\Delta t} \mathcal{L} dt + \int_{\Delta t}^{2\Delta t} \mathcal{L} dt + ... + \int_{(N-1)\Delta t}^{N\Delta t} \mathcal{L} dt \tag{4.12}$$

For small Δt each integrand is assumed constant throughout the intervals defined by the integral limits, so the cost function can be approximated:

$$I \simeq h(\mathbf{x}(N), N) + \Delta t \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}(k), \mathbf{u}(k), k) \tag{4.13}$$

or, more compactly

$$I \triangleq h(\mathbf{x}(N), N) + \sum_{k=0}^{N-1} \mathcal{L}_D(\mathbf{x}(k), \mathbf{u}(k), k) \tag{4.14}$$

The cost of arriving at the final state $\mathbf{x}(N)$ is:

$$I_{NN} \triangleq h(\mathbf{x}(N), N) \tag{4.15}$$

If the system is at stage $N-1$, the cost associated with the corresponding state is the cost of transition from $N-1$ to N, plus the cost of reaching the terminal state at time N, namely

$$I_{N-1,N} \triangleq \mathcal{L}_D(\mathbf{x}(N-1), \mathbf{u}(N-1), N-1) + I_{NN}(\mathbf{x}(N), N) \tag{4.16}$$

Using (4.11) the terminal state $\mathbf{x}(N)$ can be eliminated from $I_{N-1,N}$, so the latter is only dependent on the state and control at stage $N-1$:

$$I_{N-1,N}(\mathbf{x}(N-1), \mathbf{u}(N-1), N-1) = \mathcal{L}_D(\mathbf{x}(N-1), \mathbf{u}(N-1), N-1) + I_{NN}(\mathbf{a}(\mathbf{x}(N-1), \mathbf{u}(N-1), N-1)) \tag{4.17}$$

Equation (4.17) is also the cost of a one-stage process with initial state $\mathbf{x}(N-1)$. The optimal cost is the minimum of (4.17) over all admissible controls at stage $N-1$, that is

$$I^*_{N-1,N} \triangleq \min_{\mathbf{u}(N-1)} \{\mathcal{L}_D(\mathbf{x}(N-1), \mathbf{u}(N-1), N-1) + I_{NN}\} \tag{4.18}$$

The control which minimizes (4.17) belongs to the optimal control history and is denoted by $\mathbf{u}^*(\mathbf{x}(N-1), N-1)$.

Bellman's principle of optimality states that if the transition from stage $N-1$ to stage N is performed in optimal manner, that portion of the trajectory will always lie on an optimal trajectory, regardless of what has happened prior to reaching stage $N-1$. In Bellman's own words (Bellman, 1957):

"*An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision*".

A proof by contradiction is given in Kirk (2004).

Moving one time step back, to stage $N-2$, the optimal cost of operation over the last two intervals can be derived in a similar fashion. The result is

$$I^*_{N-2,N} \triangleq \min_{\mathbf{u}(N-2),\mathbf{u}(N-1)} \left\{ \mathcal{L}_D(\mathbf{x}(N-2),\mathbf{u}(N-2),N-2) + I_{N-1,N} \right\} \tag{4.19}$$

Invoking the principle of optimality, (4.19) can be expressed as:

$$I^*_{N-2,N} \triangleq \min_{\mathbf{u}(N-2)} \left\{ \mathcal{L}_D(\mathbf{x}(N-2),\mathbf{u}(N-2),N-2) + I^*_{N-1,N} \right\} \tag{4.20}$$

Equation (4.20) means that optimization of this two stage process need only be performed over $\mathbf{u}(N-2)$, while at stage $N-2$. The final-stage portion of the process is already optimal by virtue of the optimality principle.

Continuing backwards in time, exactly as demonstrated above, the recurrence formula for a K-stage process is obtained:

$$I^*_{N-K,N}(\mathbf{x}(N-K),N-K) \triangleq \min_{\mathbf{u}(N-K)} \left\{ \mathcal{L}_D(\mathbf{x}(N-K),\mathbf{u}(N-K),N-K) + I^*_{N-(K-1),N} \right\} \tag{4.21}$$

Equation (4.21) is known as the "Bellman equation". It is a discrete functional equation, thus suitable for computer implementation. The continuous analogue of this equation is the nonlinear PDE, known as the "Hamilton-Jacobi-Bellman or HJB equation":

$$\frac{\partial I^*(\mathbf{x}(t),t)}{\partial t} = -\min_{\mathbf{u}(t)} \left\{ \mathcal{L}(\mathbf{x}(t),\mathbf{u}(t),t) + \frac{\partial I^*(\mathbf{x}(t),t)}{\partial \mathbf{x}} \cdot \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t) \right\} \tag{4.22}$$

The equivalence of (4.21) and (4.22) is demonstrated in Kirk (2004).

4.2 Standard Computational Procedure of DP

A standard algorithm is given here, utilizing the recurrence relation (4.21) to solve optimization problems. The algorithm is straightforward and relatively easy to program. However, there are some shortcomings associated with accuracy and computational efficiency. These are addressed in section 4.3. Regardless, the logic of the procedure is both useful and instructive, and provides the basis for more advanced algorithms that can be used to solve many practical problems (an example is the IDP algorithm of section 4.3).

1. Discretize time into N stages of equal length.
2. Discretize each component j of the $n\times 1$ state vector into s_j admissible values as follows:

$$s_j = \frac{x_{j_{\max}} - x_{j_{\min}}}{\Delta x_j} + 1 \qquad j = 1,2,...,n \tag{4.23}$$

In (4.23), Δx_j is chosen so that the fraction is an integer. The state discretization results in a grid of size $s_1 \times s_2 ... \times s_n$, namely of a total of $S = s_1 \cdot s_2 \cdot .... \cdot s_n$ grid points.

3. Discretize each component of the $m\times 1$ control vector into c_r admissible values:

$$c_r = \frac{u_{r_{\max}} - u_{r_{\min}}}{\Delta u_r} + 1 \qquad r = 1,2,...,m \tag{4.24}$$

Again, Δu_r divides the range of admissible controls $\left(u_{r_{\max}} - u_{r_{\min}}\right)$ to an integer number. The total number of admissible controls is $C = c_1 \cdot c_2 \cdot .... \cdot c_m$.

At each stage k $(k = 0,1,...,N)$ the state of the system can be any one of the vectors $\mathbf{x}^i(k)$ $(i = 1,2,...,S)$ and the value of the control can be any one of the vectors $\mathbf{u}^j(k)$ $(j = 1,2,...,C)$.

4. Calculate the terminal cost $I^*_{NN}(\mathbf{x}^i(N))$ for all grid points i.

5. Move one stage backwards, setting $K=1$. Start with the grid point corresponding to $i=1$, i.e. state $\mathbf{x}^1(N-K)$. Apply all admissible controls $\mathbf{u}^j(N-K)$ $(j=1,2,...,C)$ one by one and record the reached state $\mathbf{x}^{1,j}(N-K+1)$ each time. This will possibly not coincide exactly with some of the existing grid points $\mathbf{x}^i(N)$ $(i=1,2,...,S)$, as shown in Figure 13. In that case the value of the optimal terminal cost $I^*_{N-(K-1),N}(\mathbf{x}^{i,j}(N-K+1))$, $(i=1, j=1,...,C)$, should be interpolated among the known values $I^*_{NN}(\mathbf{x}^i(N))$ for all i.

6. Calculate the total cost at state $i=1$ and stage $N-K$ for all controls $j=1,...,C$:

$$I_{N-K,N}(\mathbf{x}^i(N-K),\mathbf{u}^j(N-K)) = \mathcal{L}_D(\mathbf{x}^i(N-K),\mathbf{u}^j(N-K)) + I^*_{N-(K-1),N}(\mathbf{x}^{i,j}(N-K+1)) \tag{4.25}$$

In (4.25) the explicit dependence of the costs on the stage $N-K$ has been suppressed for brevity.

7. Minimize (4.25) over all controls and store the optimal value $I^*_{N-K,N} = \min_{\mathbf{u}^j(N-K)} I_{N-K,N}$ and the optimal control $\mathbf{u}^{i*}(N-K)$. This notation for the control means the optimal decision at state (grid point) i and at stage $N-K$.

8. Repeat steps 5 through 7 above for all the grid points i at stage $N-K$.

9. Repeat step 8 above for all the stages moving backwards one stage at a time, i.e. increasing the integer K.

Upon completion of the algorithm for all states and stages, a value for the optimal cost and optimal control is available at any point of the $n+1$ dimensional state-time grid. A solution can be readily obtained starting with any initial condition (i.e. initial state) and integrating forward using the stored optimal controls at each reached state. Evidently, the DP algorithm provides a solution not only for the problem with the particular initial condition but for the whole family of problems with different initial conditions, those corresponding to the states (i.e. grid points) of stage 1.

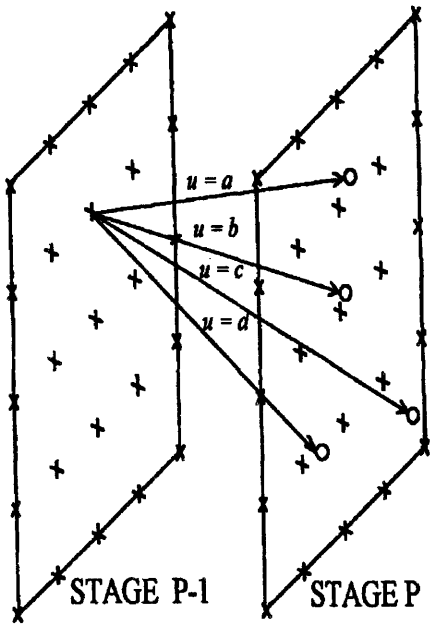


Figure 13 Application of control may reach states off the grid (Luus, 2000)

4.3 An Improved Approach: Iterative Dynamic Programming

4.3.1 Introduction

The computational procedure outlined in the previous section may look appealing for a number of reasons (Larson and Casti, 1982). It is simple and easy to code. It bypasses the issues of existence and uniqueness of solution, continuity or existence of the HJB derivatives and optimality conditions. It can handle strong nonlinearities and any type of constraint easily. It is less prone to be trapped in a local minimum (or maximum) than a numerical algorithm stemming from the classical calculus of variations (like the steepest descent algorithm). The solution is a feedback control law, since the control is determined for any state and stage.

However, to ensure convergence to the global optimum a fine grid is required. This makes the feasibility of the algorithm extremely sensitive to the dimensionality of the problem in hand (i.e. the number of elements of the state vector). Even for problems of moderately low dimension, memory storage requirement and computational time are

such that make it at least impractical, if not completely infeasible, to run even with the current computational capacity. This shortcoming is known as the "curse of dimensionality" and it has provided the impetus for developing new techniques that circumvent the problem while at the same time retain the advantages of Dynamic Programming formulation.

One of these advanced techniques is the Iterative Dynamic Programming (IDP) methodology developed by professor Rein Luus in the 90's. This method is utilized in this paper in the solution of the deterministic minimum-fuel navigation problem.

4.3.2 IDP Algorithm

The algorithm described in this section is a special case of the general IDP structure. It entails the use of piecewise constant control throughout the length of each stage, for the minimization of a performance index. Instead of a complete grid of admissible states it uses a single grid point. Starting with an initial guess for the optimal control, a procedure fairly similar to the standard DP algorithm is carried out iteratively. An optimal control policy is obtained in each iteration. The next iteration improves the solution by refining the granularity of the quantized allowable controls around the control policy of the previous iteration. The procedure outlined here is adapted from Luus (2000).

1. Discretize the time interval $[0, t_f]$ into N time stages of equal length Δt.

2. Make an initial "guess" for the entire control sequence $\mathbf{u}^*(k) \quad (k = 1,...,N)$. In the iteration following the first, this will be the optimal control obtained in the previous iteration. Using $\mathbf{u}^*(k)$ integrate (4.4) to generate an initial nominal trajectory $\mathbf{x}^*(k)$, where $k = 1,...,N+1$.

3. Choose the number of controls $c_r \quad (r = 1,2,...,m)$ for each element of the control vector $\mathbf{u}$. The total number of controls is $C = c_1 \cdot c_2 \cdot ... \cdot c_m$. Choose the control region contraction factor γ.

4. Choose an increment $\Delta\mathbf{u}$ that determines the span of control region around the central value $\mathbf{u}^*$, i.e.

$$ {}^{j}\mathbf{u}^* \pm \Delta\mathbf{u} \cdot {}^{j}R \tag{4.26} $$

The left superscript j denotes iteration number. The quantity ${}^{j}R$ is a scaling factor applied to $\Delta\mathbf{u}$. It is responsible for narrowing the control region around ${}^{j}\mathbf{u}^*$ in every iteration. For the first iteration ${}^{1}R = 1$.

5. Discretize each element r of $\Delta\mathbf{u}$ into $\frac{c_r}{2}$ quantized values.

6. Start the iterations by setting $j = 1$, ${}^{j}R = 1$.

7. Move one step back form the terminal state to the beginning of stage N. This corresponds to time $t_f - \Delta t$. Integrate forward along the last stage, that is from time $t_f - \Delta t$ to time t_f, C times using C different values of $\mathbf{u}$:

$$ \mathbf{u}(N) = {}^{j}\mathbf{u}^*(N) \pm \Delta\mathbf{u} \cdot {}^{j}R \tag{4.27} $$

If any of the elements of $\mathbf{u}(N)$ exceeds the allowable values for control, clip them at the upper or lower bound as appropriate. The integration starts from the known state ${}^{j}\mathbf{x}^*(N)$ of the current optimal path. Calculate the reached state $\mathbf{x}^r(N+1)$ $(r = 1,2,...,C)$ and the associated performance index I for each of the C different controls. Find the r for which I is minimized and store the corresponding $\mathbf{u}^r(N)$ as ${}^{j+1}\mathbf{u}^*(N)$.

8. Move one step further back, to the beginning of stage $N-1$ (corresponding to time $t_f - 2\Delta t$). Starting from state ${}^{j}\mathbf{x}^*(N-1)$ integrate one step forward, up to time $t_f - \Delta t$, using C different values of $\mathbf{u}$:

$$ \mathbf{u}(N-1) = {}^{j}\mathbf{u}^*(N-1) \pm \Delta\mathbf{u} \cdot {}^{j}R \tag{4.28} $$

Calculate the reached state $\mathbf{x}^r(N)$. For every r, carry out the integration of the remaining trajectory up to the terminal state. Use the optimal control ${}^{j+1}\mathbf{u}^*(N)$ derived in the previous step to integrate. Calculate the performance index for this segment of the

path (i.e. from time $t_f - 2\Delta t$ to t_f). Among the C different values, store the minimum I and the corresponding control as ${}^{j+1}\mathbf{u}^*(N-1)$.

9. Repeat steps 7 and 8 until the initial state is reached (initial condition). Integrate forward along the complete path using all available combinations of control as before and determine the best one for the initial stage, i.e. ${}^{j+1}\mathbf{u}^*(1)$. This concludes the first iteration. A complete control sequence ${}^{j+1}\mathbf{u}^*(k)$ $(k = 1,2,...,N)$ and optimal trajectory ${}^{j+1}\mathbf{x}^*(l)$, $l = 1,2,...,N+1$ are now available for the next iteration.

10. Reduce the size of the control region by setting ${}^{j+1}R = \gamma^j R$. Increase the iteration index j by 1 and repeat the algorithm starting from step 6.

4.3.3 The Benefits of IDP

The IDP algorithm combines most of the advantages of the standard DP algorithm, while at the same time eliminates the "curse of dimensionality". In fact, problems of very high dimension are still tractable as long as the total number of controls C remains low. Since derivatives are not required in this formulation, IDP is applicable to a broader scope of problems than variational methods based on Pontryagin's maximum principle. Luus (2000) points out that IDP has proven very reliable in finding the global optimum in very difficult problems with many local optima.

Modifications of the single-grid-point method described above can be adopted depending on the problem structure. Such modifications, proposed by Luus (2000), are the addition of more grid points (e.g. 3 or 5) and/or the augmentation of the algorithm cycles into a multipass approach. In this approach, after the iterations are completed, the control region is restored to a fraction of its original size. Then, the whole scheme is repeated in a second pass, where the iterations gradually contract the region size as before. If more passes are desired, the region size is again (partially) restored and the cycles continue.

In the solution of the routing problem with which this paper is concerned, the IDP methodology was opted over variational optimization methods (e.g. gradient methods).

The reason for this choice, apart from the benefits discussed above, was the particular structure of the minimum-fuel problem. It is a highly non-linear problem for which convexity or other optimality conditions cannot be established either locally or globally. Prior to applying the algorithm to the routing problem, it was tested on benchmark problems with known solutions. Those were the Zermelo problem of a ship traveling in calm water with varying current (Bryson, 1999), the Continuous Stirred Tank Reactor problem of coolant flow into a chemical reactor (Kirk 2004 and Luus 2000), and the trivial problem of determining the shortest path of a ship traveling between to points in calm water. In all cases the algorithm yielded satisfactory results, which built up some confidence regarding the proper utilization of the method.

CHAPTER

5

MINIMUM-FUEL SHIP ROUTING

This chapter lays the foundation for the solution of the problem of interest, which is minimum-fuel navigation. Section 5.1 derives the system model. In 5.2 the cost function is defined and in 5.3 the constraints. Section 5.4 assembles the individual results of the previous sections into the complete formulation of the routing problem in continuous and discrete time. The final section discusses the underlying assumptions of the approach followed in this paper to solve the problem in hand.

5.1 System Dynamics

5.1.1 Definition of States and Controls

Consider a ship moving in the ocean. Each point of the ocean is characterized by certain properties describing the severity of the sea state at that particular location and time. Clearly, the motion is affected by the position of the ship in the sea because the mean added resistance will be a function of space and time. This is not the case in calm water, where the problem is time and space invariant. Recognizing this, a natural choice for the "state" of the system is the ship's location on the sea surface. This location should be referenced to an appropriate coordinate system. This is worked out in section 5.1.2. By definition, a state should have the property of describing the current condition and prior history of a process in sufficient detail to allow evaluation of current alternatives (Denardo, 2003). The position vector $\mathbf{x}(t) = \begin{pmatrix} x(t) & y(t) \end{pmatrix}^T$ satisfies the above property and therefore qualifies for a state. The state variables will be the elements of the position vector, i.e. the coordinates $x(t)$ and $y(t)$.

The choice of controls comes naturally as the answer to the simple question: What controls the route of a ship in the sea? Speed and heading, of course. Instead of speed,

one might pick shaft rotational speed or engine power setting as a possible control. These are completely equivalent since there is a one-to-one correspondence between shaft speed, throttle setting and ship speed. This paper uses the actual speed of advance U referenced to an inertial frame. The other control will be the course p measured from the true North. The control vector can be written as $\mathbf{u}(t) = \begin{pmatrix} U(t) & p(t) \end{pmatrix}^T$. The mathematical model of the system will simply be the equations of motion in a plane:

$$\begin{aligned} x(t) &= U(t)\cos p(t) \\ y(t) &= U(t)\sin p(t) \end{aligned} \tag{5.1}$$

Equation (5.1) is sometimes referred to as the dynamic constraint of the optimization problem.

5.1.2 Maps and Projections

The mathematical model is chosen to refer to a Cartesian planar coordinate system. In such a coordinate system the minimum distance between two points is the straight line that connects them. However, the surface of the earth is not planar. In order for the optimization scheme to be meaningful, the curvilinear coordinate system of longitude λ and latitude ϕ used to identify true position on the spherical globe needs to be mapped to a plane through appropriate transformation. A number of such transformations exist, known as *projections.* A brief discussion about projections follows, but it is useful to be preceded by two basic navigational definitions:

Loxodrome

It is the projection of a curve intersecting the meridians at a constant angle. It is also called a *rhumb line* and is essentially the path of constant heading navigation. The loxodrome is the most common route followed by the ships because it is plotted on Mercator maps as a straight line. Therefore, the ships need only maintain a steady course along this route.

Orthodrome

It is the projection of the shortest curve between two points on a sphere. The shortest connection between two points is part of a *great circle*. The orthodrome is the projection of a great circle segment into the plane. Although it represents the shortest distance curve it is not a straight line in all but the gnomonic projection. Since navigational routes are commonly plotted on Mercator maps, a ship would have to constantly change course to follow the great circle path. This is a serious practical shortfall which has rendered the use of great circle navigation very limited.

In this paper, the Mercator projection is adopted. The transformation equations are:

$$X = \bar{R}_{earth} \ln\left[\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right)\right] \tag{5.2}$$

$$\begin{aligned} Y &= \bar{R}_{earth}\lambda \\ \lambda &\in [0,2\pi) \end{aligned} \tag{5.3}$$

where the latitude $\phi \in \left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ is measured from the equator (positive north), the longitude $\lambda \in [0,2\pi)$ is measured from the Greenwich meridian (increasing eastbound) and $\bar{R}_{earth}$ is the mean radius of the earth.

The main advantage of Mercator projection, apart from being abundantly used in navigation, is the fact that it is conformal. The wave forecast from the numerical model WAM is given in spherical coordinates (latitude, longitude) and each point carries the information of mean wave (and swell) direction. It is therefore very important to employ a mapping that preserves angles, in order to easily incorporate the forecast information in the optimization scheme. A central projection, like the gnomonic, is not conformal. This makes the mathematical representation of wave direction and ship route very cumbersome.

Figure 14 shows the loxodrome and orthodrome paths in both Mercator and gnomonic projections. An extensive treatment of map projections can be found in Richardus and Adler (1972).

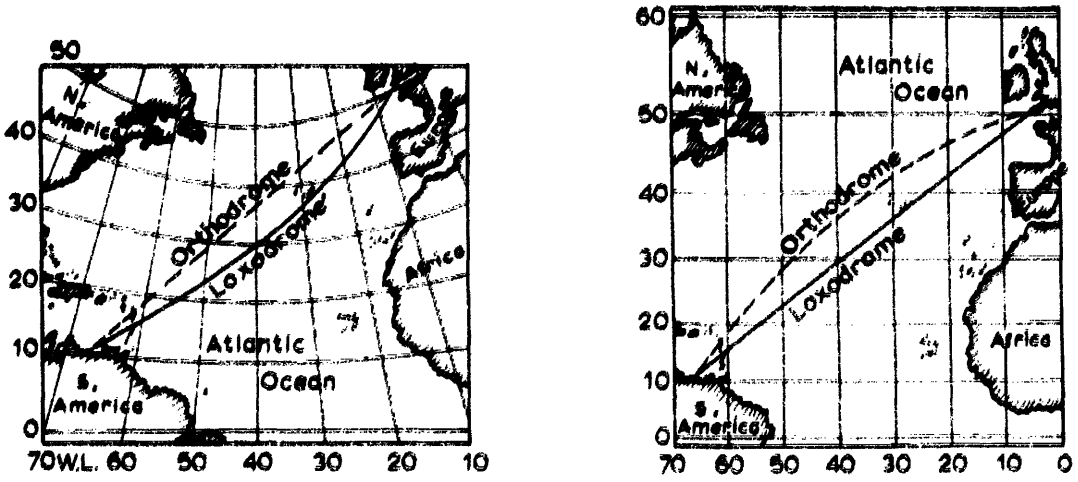


Figure 14 Example of gnomonic (left) and Mercator (right) projections (Richardus and Adler, 1972)

5.2 Performance Index

5.2.1 Resistance and Powering in Rough Seas

Assuming quasi-steady conditions (see section 5.5) the thrust delivered by the propulsor is balanced by the total resistance:

$$T(U,p,x,y,\tilde{t}) = \frac{R_{calm}(U) + \bar{R}_w(U,p,x,y,\tilde{t})}{1-t} \tag{5.4}$$

where $\tilde{t}$ denotes time to distinguish it from the thrust deduction factor t. Owing to the presence of waves, the thrust T is a function of the control settings as well as the position of the ship in space and time. This dependence on state, control and time will be conveyed to the fuel consumption as will be shown subsequently. It is, therefore, the

waves that remove the time and space invariance that the performance index would normally enjoy in the calm water version of the problem.

Substituting equations (2.5), (2.7) and (2.8) into (5.4) we obtain the load curve (2.14):

$$K_T = \frac{R_{calm}(U) + \bar{R}_w(\mathbf{u}, \mathbf{x}, \tilde{t})}{\rho(1-t)(1-w)^2 U^2} J^2 \tag{5.5}$$

The load curve together with the $K_T = f_1(J)$ curve of the B-series chart define a nonlinear system of equations in the unknown K_T and J. The system can be solved numerically to yield the operating point of the propeller at that particular load. Substituting the solution J into the $K_Q = f_2(J)$ curve of the B-series chart, the torque coefficient is computed. Finally, the open water efficiency at that particular operating condition (i.e. external load) is calculated from the relation

$$\eta_o = \frac{K_T J}{2\pi K_Q} \tag{5.6}$$

The last equation results from substitution of (2.5), (2.6) and (2.7) in (2.9).

5.2.2 Fuel Consumption

The rate of fuel mass flow in the engine is given by the product of the specific fuel consumption and the break power:

$$q = sfc \cdot P_B \tag{5.7}$$

Making use of equations (2.15) through (2.24) the break power can be expressed as:

$$P_B = \frac{R_{total} U}{\eta_D \eta_{TRM}} \tag{5.8}$$

where

$$R_{total} = R_{calm} + \bar{R}_w \tag{5.9}$$

and the *quasi-propulsive coefficient* η_D and *transmission efficiency* η_{TRM} are defined as:

$$\eta_D = \eta_R \eta_o \eta_H \tag{5.10}$$

$$\eta_{TRM} = \eta_S \eta_{GB} \tag{5.11}$$

In light of equations (5.8) through (5.11) the fuel rate can be formally expressed as:

$$q(\mathbf{u}, \mathbf{x}, \tilde{t}) = sfc \cdot \frac{R_{calm}(U) + \bar{R}_w(\mathbf{u}, \mathbf{x}, \tilde{t})}{\eta_R \cdot \eta_o(\mathbf{u}, \mathbf{x}, \tilde{t}) \cdot \eta_H \cdot \eta_S \cdot \eta_{GB}} \tag{5.12}$$

where the control and state vectors have already been defined, respectively, by the following relations:

$$\mathbf{u} = \mathbf{u}(\tilde{t}) = \begin{pmatrix} U(\tilde{t}) \\ p(\tilde{t}) \end{pmatrix} \tag{5.13}$$

$$\mathbf{x} = \mathbf{x}(\tilde{t}) = \begin{pmatrix} x(\tilde{t}) \\ y(\tilde{t}) \end{pmatrix} \tag{5.14}$$

5.2.3 Terminal State Penalty

In chapter 4 the performance index was defined to consist of a functional in the form of an integral and a function $h(\mathbf{x}(t_f), t_f)$ which depends on the terminal state and final time. This function stands for the penalty (in performance or cost) incurred by the system finding itself at a particular state for a particular final time. For problems with fixed final time, like the routing problem dealt with here, the explicit dependence on t_f can be dropped and the terminal cost can be written as $h(\mathbf{x}(t_f))$. There are certain terminal states that will incur a large terminal cost and others with smaller or none at all.

In the routing problem, since the final time is fixed, there has to be a way to penalize any voyage track that will result in a terminal location other than the predefined destination point. The farther the ship finds herself from the destination, when the voyage time expires, the higher the penalty. A reasonable way to quantify this penalization could

be to incorporate the final distance from destination in the function $h(\mathbf{x}(t_f))$. The terminal cost would then be of the form

$$h(\mathbf{x}(t_f)) = \tilde{c} \cdot d(\mathbf{x}(t_f)) \tag{5.15}$$

where the distance from destination point (x_{dest}, y_{dest}) is defined:

$$d(\mathbf{x}(t_f)) = \sqrt{\left(x(t_f) - x_{dest}\right)^2 + \left(y(t_f) - y_{dest}\right)^2} \tag{5.16}$$

and $\tilde{c}$ is a weighting constant which also carries the unit conversion of distance (meters) to fuel weight (metric tons). The performance index for the routing problem would have the form:

$$I = \tilde{c} \cdot \sqrt{\left(x(t_f) - x_{dest}\right)^2 + \left(y(t_f) - y_{dest}\right)^2} + \int_0^{t_f} q(\mathbf{u}, \mathbf{x}, \tilde{t}) dt \tag{5.17}$$

Although the above representation of the terminal state penalty is perfectly legitimate, it turns out that it encumbers the computer program with unnecessary computations which increase the execution time and make the detection of the global optimum a quite challenging task. In a more elegant representation the terminal location is fixed and the ship is forced to be at that location when the final time is reached. By treating the terminal cost as a "hard constraint", as discussed later in chapter 6, the routing problem is much more straightforward and the results come out faster and with greater accuracy. In this paper, the latter representation is used and the performance index in (5.17) is free of the first term. In a more general "minimum-cost" formulation (not just minimum fuel), the terminal cost may be selected to reflect actual cost incurred by the contract demurrage or dispatch clauses.

5.3 Inequality Constraints

5.3.1 State Constraints

With state defined as the position on the map, possible state constraints would represent regions of forbidden navigation, such as shallow waters or land. In the present work, trans-Atlantic passage is addressed, thus assuming open ocean navigation with no state constraints.

5.3.2 Control Constraints

The ship's course should be free to take any value between 0 and 360 degrees. Consequently, no constraint is imposed on this control. In practice, as it will be shown in chapter 6, when implementing the IDP algorithm, there is a heading allowance of ±90 degrees relative to the nominal course. In other words, the ship is allowed to deviate only up to 90 degrees on each side relative to her current heading. Turn-around or heading back towards the destination is not permitted. This is not overly restrictive, as ship captains almost never turn around to avoid bad weather (and when they do it is because of being oblivious of more "optimized" alternatives). At the same time, it cuts down the computational cost enormously, as the discretization of heading space spans only 180 degrees instead of a full circle of 360 degrees.

The speed is constrained not to exceed 16 knots, for the example-ship used in this paper. This upper bound is imposed by the propulsion plant capabilities. Apart from that, a lower bound of 7 knots is imposed as well. This speed corresponds to a very low Froude number ($Fn \sim 0.1$). The responses at this and lower speeds are practically identical to the zero speed responses (as verified by SWAN). This simply means that by further reducing speed below 7 knots to alleviate the effect of severe sea state the captain will not gain anything at all. If he makes no course alteration the motions will remain practically the same at any speed from 7 knots to a standstill. Thus, applying a lower bound to speed is justifiable from a practical standpoint. Moreover, it offers the advantage of narrowing the size of allowable control space in favor of computational speed.

In general, the control constraints proposed in the current work are:

$$0 \le p \le 2\pi \tag{5.18}$$

$$U_{\min} \le U \le U_{\max} \tag{5.19}$$

5.3.3 Safety Constraints

While seeking the trajectory and control policy that will minimize fuel consumption, the optimizing program should also ensure that certain safety limits will not be exceeded. Here, the composite events of relative motion and relative velocity in the bow are examined in order to assess the frequency of occurrence of deck wetness and slamming.

Consider a point Π_1 in the uppermost part of the bow. Deck wetness, or green water on deck, occurs when the relative motion at point Π_1 exceeds the freeboard f at that point. The variance σ_r^2 of relative motion of any point on the ship is calculated in equation (2.75) where the respective RAO is given by (2.73), (2.74). It can be shown (Ochi, 1998) that the maxima of this event follow the Rayleigh distribution. The probability of relative motion exceeding the freeboard and thus the probability of green water on deck is

$$P(\text{water on deck}) = e^{-\frac{f^2}{2\sigma_r^2}} \tag{5.20}$$

Now consider a point Π_2 in the keel at a longitudinal position 10% of the ship's LBP aft of FP (Journée and Meijers, 1980). Slamming occurs when the relative motion at point Π_2 exceeds the draft at that point and when the relative velocity exceeds a critical value defined by

$$V_{cr} = 0.093\sqrt{g \cdot LBP} \tag{5.21}$$

where the length between perpendiculars is measured in (m), the acceleration of gravity in (m/sec^2) and the critical speed in (m/sec). Assuming that the two events are statistically independent, the Rayleigh distribution gives the probability of slamming:

$$P(\text{slamming}) = e^{-\left(\frac{H^2}{2\sigma_r^2}+\frac{V_{cr}^2}{2\sigma_V^2}\right)} \tag{5.22}$$

where H is the draft, σ_r^2 the variance of relative motion at Π_2 and σ_V^2 the variance of relative velocity at Π_2. Note that σ_r^2 is not the same as the one which appears in (5.20), because it refers to a different point. The calculation of σ_V^2 follows the logic of equations (2.73) through (2.75), using the result:

$$RAO_V(\omega) = i\omega RAO_r(\omega) \tag{5.23}$$

Above certain limits of deck wetness and slamming frequency of occurrence, the motions are so severe that the ship's safety is compromised and voluntary speed reduction is required. These limits are set to the following values (Faltinsen, 1990):

$$P(\text{slamming}) \le 0.03 \tag{5.24}$$

$$P(\text{water on deck}) \le 0.07 \tag{5.25}$$

5.4 Formulation of the Routing Problem

5.4.1 Continuous Time Formulation

The complete formulation of the ship routing problem can be summarized in a standard format for constrained optimization problems:

$$\min \quad I = \int_0^{t_f} sfc \cdot \frac{R_{calm}(U) + \overline{R}_w(\mathbf{u}, \mathbf{x}, \tilde{t})}{\eta_R \cdot \eta_o(\mathbf{u}, \mathbf{x}, \tilde{t}) \cdot \eta_H \cdot \eta_S \cdot \eta_{GB}} dt \tag{5.26}$$

subject to:

a. dynamic constraint

$$x(t) = U(t)\cos p(t)$$

$$y(t) = U(t)\sin p(t)$$

b. control bounds

$$0 \le p \le 2\pi$$

$$U_{\min} \le U \le U_{\max}$$

c. safety constraints

$$P(\text{slamming}) = e^{-\left(\frac{H^2}{2\sigma_r^2}+\frac{V_{cr}^2}{2\sigma_V^2}\right)} \le 0.03$$

$$P(\text{water on deck}) = e^{-\frac{f^2}{2\sigma_r^2}} \le 0.07$$

d. initial conditions and final time

$$\mathbf{x}(0) = \begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

t_f prescribed

5.4.2 Discrete Time Formulation

The discrete counterparts of the cost function and dynamic constraint are needed in order to solve the problem numerically. The final time is divided into N stages of length

$$\Delta t = \frac{t_f}{N} \tag{5.27}$$

The equations of motion are approximated as

$$x(k+1) = x(k) + \Delta t U(k)\cos p(k) \tag{5.28}$$

$$y(k+1) = y(k) + \Delta t U(k)\sin p(k) \tag{5.29}$$

with $x(1) = x_0$, $y(1) = y_0$ and $k = 1, 2, ..., N$.

The performance index in discrete form is:

$$I = \sum_{k=1}^{N} q\big(U(k), p(k), x(k), y(k), k\big)\Delta t \tag{5.30}$$

5.5 Assumptions

The formulation derived in the preceding sections relies on certain assumptions, cited here. Some of these assumptions are revisited in chapter 7 with recommendations for future work.

1. The framework of the optimal control problem is entirely deterministic. It is assumed that state measurements are perfect and so are the measurements of control commands. This is a valid assumption given the accuracy of speed measurement as well as course and position tracking offered by GPS and contemporary navigational instruments (INS, electronic gyros etc.). In reality, what forces the formulation to be deterministic is that the state of the art in wave forecasting is deterministic. The problem is solved as if the 5-day forecast available were not an estimate but rather a perfect knowledge of the future weather. This is of course not true because any forecast is inherently stochastic.

2. The length of the time stages is chosen to coincide with the period of the forecast information, namely 3 hours. In the time window of those 3 hours a stationary sea state is assumed with spatial variability as given by the forecast. Stationarity is a legitimate assumption for short-term description of a sea state up to perhaps 10 hours (Faltinsen, 1990). Furthermore, it permits the use of frequency domain methods for the determination of ship responses within that time window. This is extremely convenient because a CFD program, like SWAN1, can be used, which utilizes frequency domain methods and predicts all the quantities of interest with notable accuracy.

3. The control policy is piecewise constant over the time windows mentioned above. No accelerations are taken into account and the transients associated with changing the control input (e.g. engine settling to a new rpm) are very short relative to the problem's

time scale and therefore neglected. This assumption has practical value as well, since it is realistic for the operator to change the speed and/or course every 3 hours.

4. As noted in chapter 2, air resistance and additional drag due to appendages, trim or steering are not included in the analysis. These are either steady components, which can be easily incorporated in the model, or small in magnitude and thus can be neglected, to a first approximation.

5. Currently, only slamming and deck wetness safety constraints are supported by the optimization program developed in this paper. Depending on the type of ship and cargo (or mission) more constraints can be added at the expense of computational speed.

CHAPTER

6

NUMERICAL IMPLEMENTATION

The solution to the problem formulated in chapter 5 can only be obtained numerically. This chapter covers the techniques used in the numerical solution. Section 6.1 defines the example-ship which provides the platform for all the computations. Section 6.2 describes how the calm water resistance is estimated. Section 6.3 deals with the computation of added resistance and ship motions, tasks which are routinely taken on by SWAN1. Section 6.4 involves propeller selection and the coupling of thrust with fuel consumption. A description of the Iterative Dynamic Programming solution of the optimal routing problem is reserved for section 6.5. The closing section contains some remarks on the optimization code structure. All the computer programs were developed in MATLAB. The respective scripts are not referenced in the text, but are given in Appendix C. Each script is self-contained, documented in detail by the appropriate commentary.

6.1 Example-Ship

The computation of flow parameters and the solution to the routing optimization problem are carried out for a baseline Series 60 hull form. The principal characteristics of the platform used in this paper are tabulated in Table 1 and Table 2. The top four values of table 2 are estimated by proper interpolation of the Series 60 parent models data from Table 3 (Todd, 1963). Note that reduction ratio equal to unity simply indicates absence of a gearbox. The transmission efficiency in that case is just equal to the shaft efficiency as defined in chapter 2.

Table 1 Principal dimensions of Series 60 example-ship

Description	Symbol	Unit	Value
Length at DWL	L	m	101.7
Beam at DWL	B	m	14.28
Draft at DWL	T	m	5.7
Freeboard at DWL (bow)	H	m	2.85
Displacement	Δ	*tons*	5700
Wetted Surface Area	S	m^2	1950
Bilge Keel Total Surface Area	S_{bk}	m^2	17.64
Block Coefficient	C_B	-	0.676
Prismatic Coefficient	C_P	-	0.688
Midship Section Coefficient	C_M	-	0.982

Table 2 Propulsion factors of Series 60 example-hull

Description	Symbol	Value
Wake Fraction	w	0.249
Thrust Deduction Factor	t	0.176
Hull Efficiency	η_H	1.0972
Relative Rotative Efficiency	η_R	1.035
Number of Shafts	k_S	1
Number of Main Engines	k_E	1
Type of Main Engine	Low-Speed Diesel	
Gear Reduction Ratio	λ	1
Transmission Efficiency	η_{TRM}	0.98

Table 3 Propulsion factors for parent models of Series 60 (Todd, 1963)

C_B	0.6	0.65	0.7	0.75	0.8
F_n	0.229	0.236	0.212	0.19	0.167
L/B	7.5	7.25	7	6.75	6.5
B/T	2.5	2.5	2.5	2.5	2.5
$L/\nabla^{1/3}$	6.166	5.869	5.593	5.335	5.092
LCB % L_{PP} from	1.50A	0.50A	0.50F	1.50F	2.50F
w	0.249	0.268	0.277	0.307	0.352
t	0.176	0.167	0.161	0.171	0.2
η_H	1.097	1.138	1.16	1.196	1.235
η_0	0.669	0.659	0.666	0.653	0.624
η_R	1.035	1.026	1.01	1.014	1.014
η_D	0.759	0.769	0.781	0.792	0.783

6.2 Resistance Calculation

6.2.1 Wave Resistance

Steady wave resistance is computed in SWAN. The program uses potential flow theory and momentum flux methods to calculate the wave-making drag experienced by the ship in calm water. The computation is performed over the range of speeds that are to be used in the optimization solution, namely from 7 to 16 knots. For the sake of comparison, steady wave resistance is also calculated independently using the method derived by Holtrop and Mennen (1982). Their method relies on statistical analysis of model tests conducted with 191 ships of various types. The regression coefficients are valid for the low Froude number regime (≤ 0.4), hence suitable for the current analysis. Figure 15 shows the results obtained from the two independent calculations.

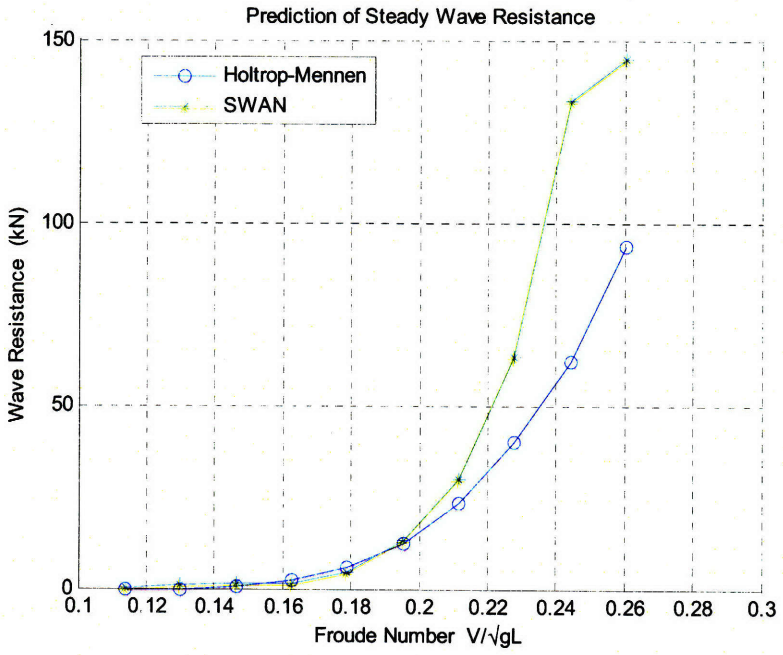


Figure 15 Wave resistance of example ship

Apparently, for all but two of the calculated Froude numbers (those corresponding to 10 and 11 knots) the Holtrop-Mennen method yields a wave resistance which is 5% to 90% lower than the SWAN prediction. Given the size, shape and draft of the ship the

Holtrop prediction seems to underestimate the actual wave drag that would normally be anticipated. This paper uses the SWAN results for steady wave resistance.

6.2.2 Viscous and Appendage Resistance

As an ideal-flow computational tool, SWAN cannot handle viscous phenomena. Consequently, to calculate the friction and form drag as well as the drag caused by the bilge keels one should resort to other methods. In this paper, the Holtrop (1984) method is applied for this purpose. It is an improvement to the Holtrop-Mennen (1982) regression analysis based on tests from 334 models. Friction drag is estimated from the ITTC formula given by (2.2). The form drag coefficient, the contribution of the bilge keels and the ship-model correlation allowance are all calculated from the Hotrop method. Figure 16 shows the two major components of resistance and their sum.

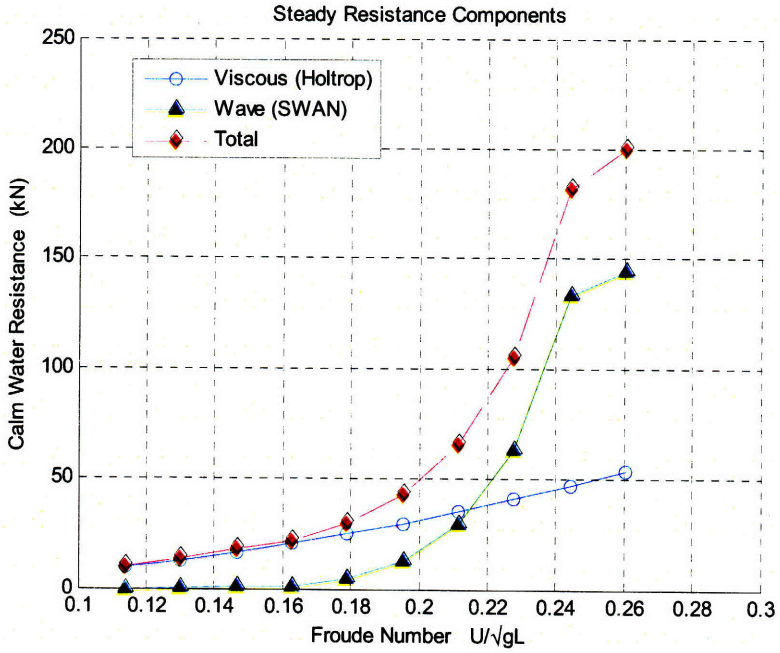


Figure 16 Resistance curves of example ship

The effective form factor for the bilge keels is read off the list of Table 5. The range of applicability of the Holtrop method is displayed in Table 4

Table 4 Range of applicability of Holtrop method (PNA vol.II)

Ship Type	Max. F_n	C_P		L/B		B/T	
		min	max	min	max	min	max
Tankers, bulk carriers (ocean)	0.24	0.73	0.85	5.1	7.1	2.4	3.2
Trawlers, coasters, tugs	0.38	0.55	0.65	3.9	6.3	2.1	3
Containerships, destroyer types	0.45	0.55	0.67	6	9.5	3	4
Cargoliners	0.3	0.56	0.75	5.3	8	2.4	4
Roll-on, roll-off ships; car-ferries	0.35	0.55	0.67	5.3	8	3.2	4

Table 5 Effective form factor values for different appendages (PNA vol. II)

Type of appendage	Value of $1+k_2$
Rudder of single-screw ship	1.3 to 1.5
Spade-type rudders of twin-screw ships	2.8
Skeg-rudders of twin-screw ships	1.5 to 2.0
Shaft brackets	3
Bossings	2
Bilge keels	1.4
Stabilizer fins	2.8
Shafts	2
Sonar dome	2.7

6.3 Ship Motions and Added Resistance Computation

Added Resistance Operators (ARO) and Response Amplitude Operators (RAO) are computed at 10 different speeds, from 7 to 16 knots (in 1-knot increments) and for 5 different wave directions (relative to the ship): head seas (180°), bow seas (135°), beam seas (90°), quartering seas (45°) and following seas (0°). The complete set of output plots is given in Appendix A. Note that there are combinations of speed and wave direction which are not admissible in SWAN1. These combinations correspond to a reduced frequency $\tau = \frac{\omega_e U}{g}$ which is lower than the critical value $\frac{1}{4}$. The radiation condition of vanishing disturbance upstream in the far field is not valid in this case, as there is a wave system propagating upstream (Sclavounos, 1996). Numerical simulation using frequency domain methods is prone to inaccuracies in the cases where $\tau < \frac{1}{4}$ and for low Froude numbers. SWAN1 safeguards against poor accuracy by skipping the runs that are more

susceptible to it. Data corresponding to this range was interpolated with cubic splines. The output is displayed in terms of the encounter wave period. A code was developed in order to reference the output to the absolute wave period, extract the ARO and RAO data and calculate the transfer functions of the composite events defined in chapter 5.

Although roll is eventually not included in the routing code (for reasons of computational efficiency), the information of roll RAO's and the related transfer functions is provided. With minor modification of the optimization code, roll constraints can easily be added.

6.4 Propulsion Calculations

The method described in section 2.2.1.2 was used for the selection of a suitable propeller for the example-ship. Setting the maximum diameter to 3.2 m and the cavitation limit to 10% the optimal propeller from the MARIN B-Series has the characteristics shown in Table 6.

Table 6 Propeller characteristics of example-ship

Description	Symbol	Value
Diameter (m)	D	3.2
Pitch Ratio	P/D	1.2
Expanded Area Ratio	EAR	0.75
Number of Blades	z	5

The B-series charts were essentially transformed into numerical tables using the Troost regression coefficients. As described in section 5.2.1, for any given external loading (thrust) there is a unique operating point for the propeller, defined by the quadruplet $\left(K_T, J, K_Q, \eta_o\right)$. Variation in the external load of the propeller is caused by the variation of ship's speed and the sea state (the latter via added resistance). As the Dynamic Programming recurrence progresses, at any reached point different conditions of the ocean are encountered. Therefore, the propeller operating point needs to be calculated for any reached state.

To obtain the operating point one has to solve the nonlinear system defined by the K_T curve of the B-series chart and the loading curve defined in equation (5.5). The graphical analogue of solving the nonlinear system is to find the intersection of the two curves on the chart, as illustrated in Figure 17. The K_T curve in the graph is for a Newton-Rader propeller, but the logic is the same.

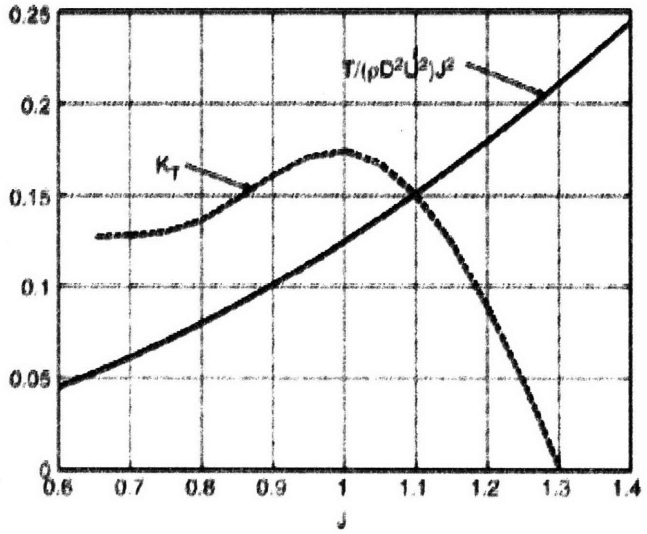


Figure 17 Example of determining operating point from curve intersection (Faltinsen, 2005)

The solution to the nonlinear system is obtained by a "solver" script developed for this purpose. It uses the MATLAB built-in command **fsolve**.

6.5 Numerical Optimization

6.5.1 Preliminaries

A fictitious route is constructed from a departure point off of the coast of Madeira (Azores) to a destination point off of the coast of New York City. The nominal speed is chosen to be 13 knots and the nominal course is the slope of the rhumb line (loxodrome) connecting the two points. Table 7 contains all the nominal route information.

Table 7 Nominal (calm water) route information

Description	Symbol	Value
Start Point – Latitude (deg)	ϕ_1	40.5
Start Point – Longitude (deg)	λ_1	330
End Point – Latitude (deg)	ϕ_2	40
End Point – Longitude (deg)	λ_2	287
Nominal Speed (knots)	u_o	13
Nominal Course (deg)	p_o	269.13
Sailing Time (days)	t_v	8.31
Sailing Distance (n.m.)	d_v	2593.8
Nominal Fuel Cost (tons)	I_{calm}	21

Forecast data is downloaded from the FTP server of the University of Athens. Before importing the data to the optimization program, the ASCII content is stored in a MATLAB compatible **.mat** file, after appropriately converting units and applying Mercator projection. The information is also filtered so that only the region of interest (i.e. North Atlantic) is covered. This is done to avoid overloading the computer memory with unnecessary data. It is understood that the preset departure and destination points need not represent the actual starting and ending locations, but rather the end points of the actual *navigation* route. This term describes the open ocean operation, in which the ship would normally follow a constant heading track in the absence of obstacles, land, heavy traffic or any kind of unexpected obstruction. The route outside these two points may be thought of as corresponding to the *piloting* route. Piloting refers to the operation of the ship under conditions of enhanced caution, as in restricted waters or when pulling in and out of port. In these situations the operator focuses on the safe passage exclusively, optimal routing being rather meaningless for this part of the voyage.

Calm water resistance, propeller curves, hydrodynamic responses and ship particulars are loaded together with the aforementioned processed weather data prior to

the execution of the optimization routine. All this information will be subsequently used to interpolate the relevant data that will be created when running the program.

6.5.2 Optimization Routine

As discussed in chapter 4, the Iterative Dynamic Programming (IDP) scheme initializes the backward recursion with an initial guess of the solution. The nominal trajectory, namely the calm water solution, appears to be a good candidate for the initial guess. Once the start and end points are specified, the details of the nominal solution are easily calculated.

A critical parameter is certainly the number R of different control settings. Choosing a large number of allowable speeds and headings introduces a fine grid of controls which works in favor of detecting the global optimum, but imposes a gigantic penalty on computational time. Tied to that is the choice of the contraction factor γ. In every iteration the control region size is reduced by the factor γ. For example, if $\gamma = 0.9$, after 20 iterations the R values of control will span a range of size $0.9^{20} r \simeq 0.12r$ as opposed to r for the first pass. It is obvious that the R values are now more tightly packed in the control domain, and this enhancement in granularity facilitates the convergence of the IDP algorithm to the optimum solution.

Having established all the required parameters, the iterations start from the $N-2$ stage stepping backwards in time, up to stage 1 as described in section 4.3.2. At each stage the R controls are screened to check if they violate the constraints. If not, the integration of the equations of motion is carried out forward in time from the current stage to the end. If the constraints are not satisfied, the particular set of controls that violates them is skipped. For any reached state, the weather information is extracted from the closest point of the forecast grid (i.e. the closest point for which weather information is known). At the end, the cost function is calculated for the given set of controls and stored in memory. Having swept the entire control space, the pair of (u, p) which yields the minimum cost is stored in memory to be used in the next loop. Upon completing the backtracking in time, there is an optimal control policy covering the entire time line of

the problem (i.e. there is a pair (u^*, p^*) at each stage). This will constitute the nominal control policy of the next iteration.

At stage $N-1$ the only possible control combination is the one which brings the ship to the final destination along a straight line (since no forecast is available and calm water is assumed). The program checks if the final state is reachable (i.e. if the speed required to transit from $N-1$ to N is no greater than the maximum ship speed). If it is not, the particular pair of (u, p) that led to this $N-1$ state is skipped as before. The program sees the terminal location as a "hard constraint", namely a state at which the ship must end up, regardless of the intermediate route. It is treated exactly like the initial state (all routes start from the same point). If the duration of the voyage is shorter than the extent of forecast coverage then the final stage (i.e. from $N-1$ to N) is traversed as explained in the previous paragraph.

After concluding the last cycle, the optimal trajectory, control policy, number of iterations and minimum cost are stored in a data file. It should be noted here that the optimization scheme is structured in a way that the first few iterations (e.g. 4 – 5) may yield an infeasible solution. This is a consequence of the fact that constraint violation is monitored only at the beginning of every stage, during the backtracking of time. The forward integrations initiated there are conducted without further verification of feasibility (in terms of constraints).

This is a crucial coding "trick" to ensure that the program will not overlook some of the feasible solutions and will actually locate the optimum. Put simply, the algorithm is allowed to "infringe" through the domain of infeasible solutions only to guarantee that the true optimum is obtained. Numerous trial runs were conducted doing exactly the opposite, namely enforcing satisfaction of constraints at each step of forward integration. In all cases the optimization process broke down. The program was terminated after ruling out all control combinations as infeasible. By applying the technique described above, this artifact is eliminated and a solution is guaranteed to be found (if it exists).

Furthermore, the running time is boosted significantly. The only required provision for this technique to work is a reasonable number of iterations (e.g. >5).

6.6 Some Comments on the Optimization Code

It is customary to write code in MATLAB exploiting the software's ability to handle vectors and matrices. This *vectorization* produces compact code avoiding unnecessary loops, which in an interpreted language like MATLAB are synonymous to low computational speed. The script "WAVEROUTE.m" is written in a different programming style, better suited to a compiled language, like FORTRAN or C. This style was opted in order to benefit from the JIT-accelerator feature of MATLAB, which compiles whole sections of contiguous code which are JIT-compatible.

In order for the code to be JIT compatible, external function callbacks are eliminated. The same holds for non-scalar **if** structures. The JIT feature only supports MATLAB built-in commands, so any potential user-defined function call would interrupt the compiler and slow down the execution. Therefore, all the computations that can be carried out off-line are performed externally to the optimization code and the results are stored in memory. Such an example is the SWAN RAO's and ARO's and the derived responses. Moreover, calculations that would normally be done live (i.e. while the program is running) were also completed externally and stored in memory. An example of this is the determination of the propeller operating point at each loading condition.

Having the results pre-calculated at discrete input values and stored in memory, the main program needs to only load those datasets and interpolate them accordingly, during runtime. This way, the compilation is not interrupted because interpolation is a procedure supported by JIT. The schematic of this programming logic is depicted in Figure 18.

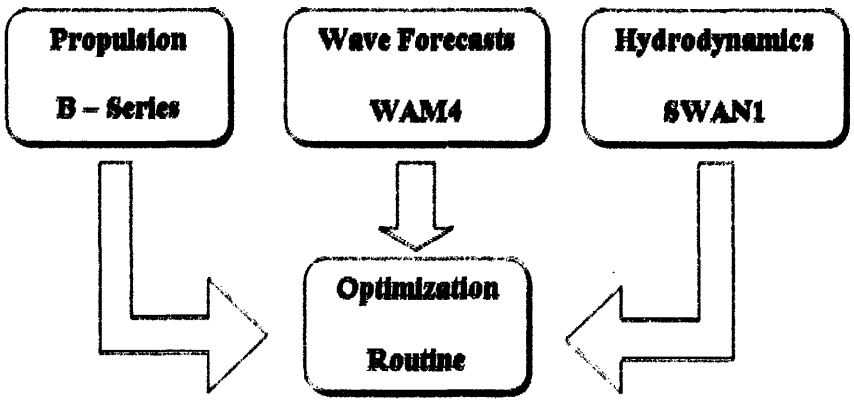


Figure 18 Optimization code logic

Although WAVEROUTE.m by no means claims merit for supreme computational efficiency, it is considerably faster than its vectorized counterpart, which was originally developed for the analysis. In particular, it exhibits a two-fold improvement in execution speed.

CHAPTER

7

RESULTS & CONCLUSIONS

7.1 Optimization Results

The program is run using the global forecast promulgated on April 5 2008, at 21:00 EDT. The forecast covers the time period from midnight of April 6 until noon of April 10. For simplicity, it is assumed that the ship is at the "start" point at 21:00, namely at the time when the forecast becomes available. The optimal route, evading bad weather, is shown in Figure 19 by the solid red line. The rhumb line route is depicted by the blue dashed line. The deviation of the proposed route from the rhumb line is greatly exaggerated in Figure 19. This is only to demonstrate pictorially the details of the route and is attributed to the different latitude (y-axis) scale. Plotted on the same scale for longitude and latitude the route looks like that of Figure 21. The optimization summary (Table 8) shows a deviation from the nominal route that adds 58 more nautical miles to the total trip distance. To obtain the results a total of 10 speed settings and 16 course settings are used in the program. That gives a number of control combinations equal to $R = 160$. The number of iterations is set to 30 and γ is chosen to be 0.7. Convergence is illustrated in Figure 20.

Table 8 Summary of optimal solution

Description	Unit	Value
Nominal Voyage Distance	naut.miles	2594
Nominal Voyage Time	days	8.31
Actual Voyage Distance	naut.miles	2652
Optimal Cost	tons	27.5878
Theoretical (loxodrome) Cost	tons	25.2426

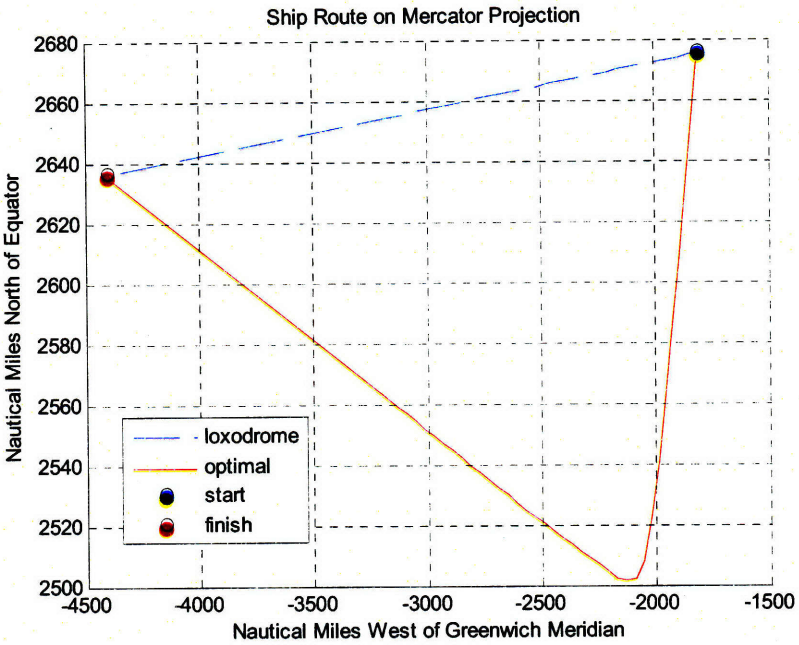


Figure 19 Minimum fuel optimal route

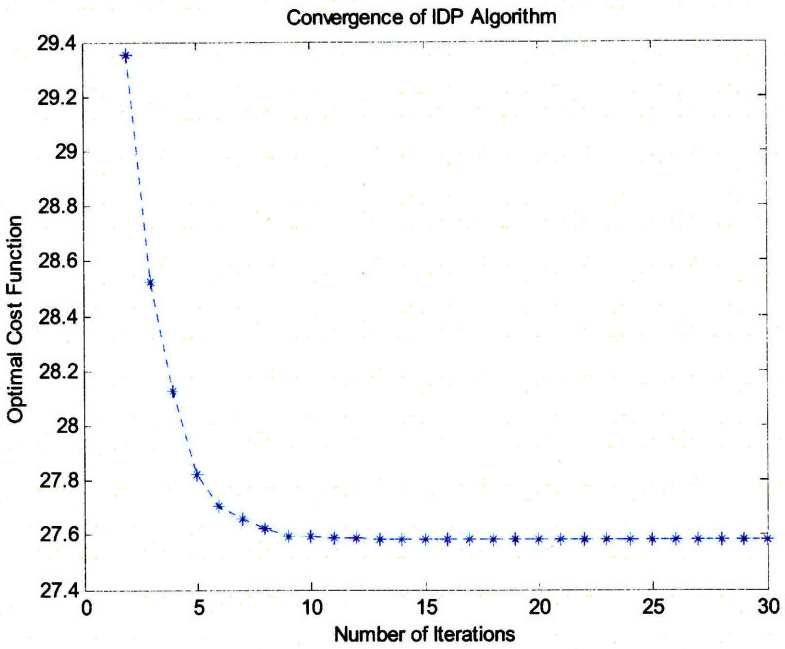


Figure 20 Convergence of the optimal solution using γ=0.7

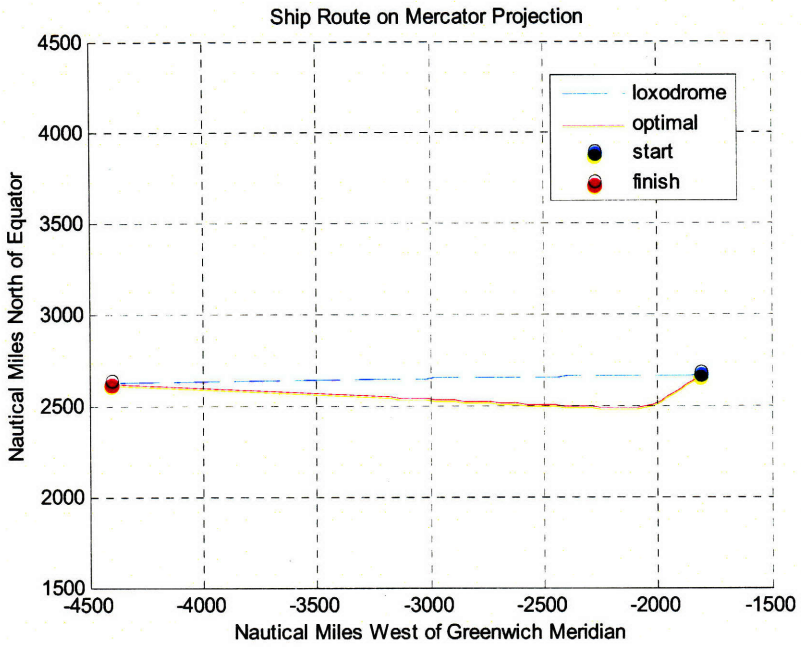


Figure 21 Solution plotted on equal x-y scales

The cost of the optimal route is 27.59 tons of fuel whereas if the ship had been allowed to follow the loxodrome the cost would have been 25.24 tons of fuel. However, the latter route violates the safety constraints over the 22% of its length. Had the weather been completely calm throughout the whole trip, the cost of the straight line route (loxodrome) would have been 21 tons. In this particular case, the additional cost incurred by the added resistance in waves is on the order of 20%. Another scenario was run with a more severe weather profile (the forecast of April 16, 2008 was used). The difference in loxodrome route cost for the two cases, namely calm and rough seas, is now computed to be on the order of 80% and the safety constraints are violated over the 72% of the total length of the route.

The last example is mentioned only to illustrate the significance of added resistance in the routing problem. The ensuing sensitivity analysis and discussion pertain to the original example presented at the beginning of the chapter.

7.2 Discussion and Conclusions

Unless otherwise stated, this section utilizes a "datum" solution obtained after 25 iterations with $R = 48$ control combinations, for faster results. The optimal route with this control space is shown in Figure 22. The black 'x' markers designate the 3-hour steps in changing the decision variables (speed and course). The straight line from the last 'x' to the destination is the optimal path in calm seas, since there is no forecast available for this part of the route (as explained in the previous chapter).

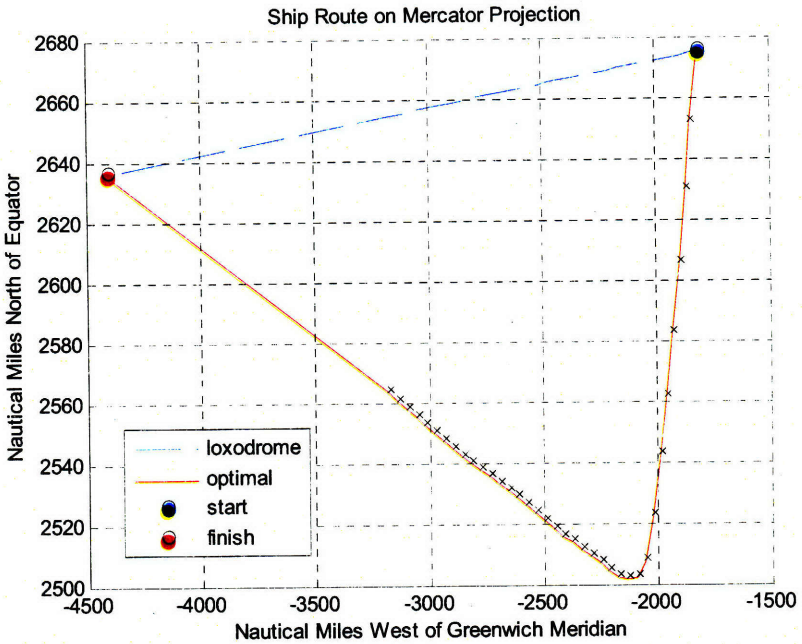


Figure 22 Optimal solution with reduced controls R=48

7.2.1 Convergence and Accuracy

Speed of convergence is dominated by the control space R and contraction factor γ. With $R = 160$ and $\gamma = 0.7$ the optimal result comes after the 27^{th} iteration, while using $\gamma = 0.85$ the final result is higher than the minimum (by 0.74%) even after as many as 60 iterations (Figure 23). Further reducing the contraction factor to 0.6 yields a faster convergence to an answer very close to the minimum (higher only by 0.016%). Setting $R = 48$ the picture is almost analogous. With $\gamma = 0.7$ we get an answer only 0.043% off

of the minimum after 25 iterations. With 0.85 the convergence is so slow that after 50 iterations the result is off by 0.22%. Choosing $\gamma = 0.6$ the minimum is missed by 1.22% but the result comes extremely fast (after 18 iterations). The loss of accuracy in the last example is due to premature collapse of the control region. The optimal control policy in the aforementioned trial runs differs from the true optimal policy at most by 4%.

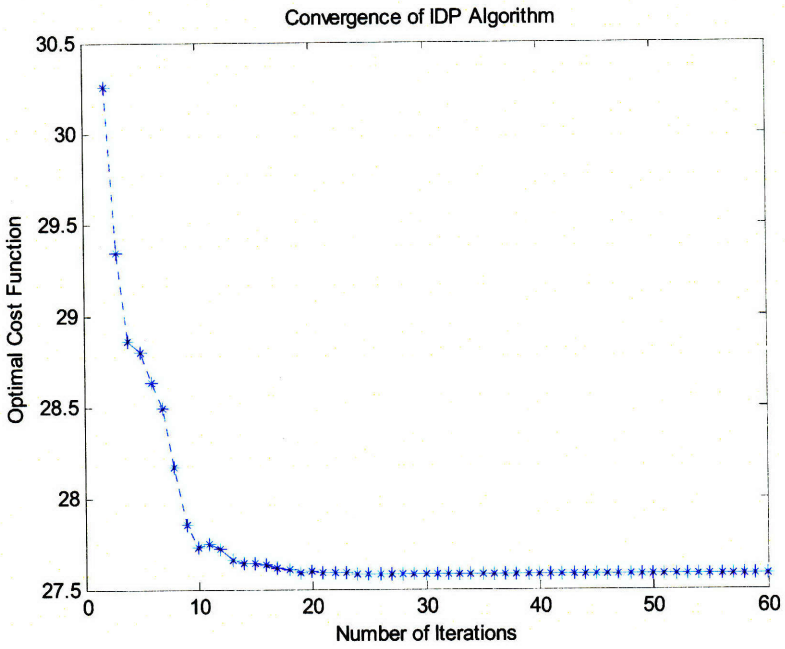


Figure 23 Convergence with R=160 and γ=0.85

Tuning the factor γ is not a trivial task, especially since the solution is not known beforehand (e.g. from analytical calculation) and thus comparisons cannot be made. There are problems fairly sensitive to the choice of γ and others that are not. A typical range is $0.75 \le \gamma \le 0.95$ and Luus (2000) suggests that when a sufficiently large R is used then γ can be safely reduced to as low as 0.7. In the present work it was found that this parameter can be further reduced to 0.6 without great compromise in the optimal outcome. Nevertheless, it is concluded that for the particular problem the best balance in the trade-off between speed and accuracy is achieved using a value of 0.7.

The evolution of the optimal solution as the iterations progress is demonstrated in Figures 24 through 27.

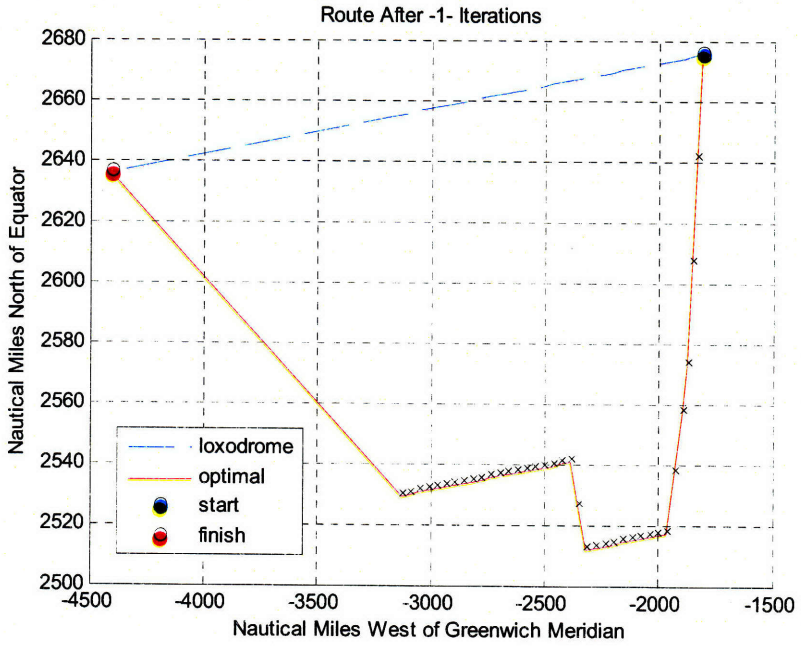


Figure 24 Output of 1st iteration

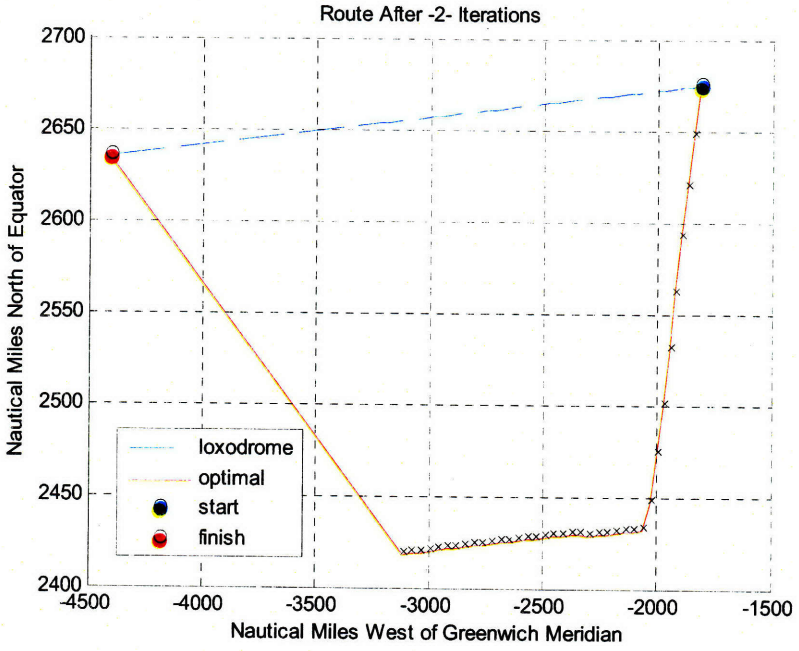


Figure 25 Output of 2nd iteration

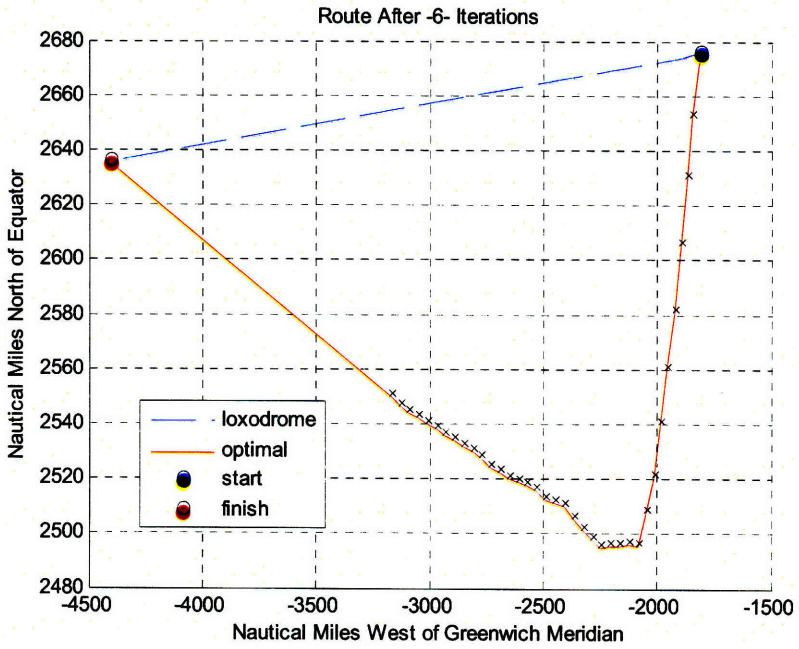


Figure 26 Output of 6th iteration

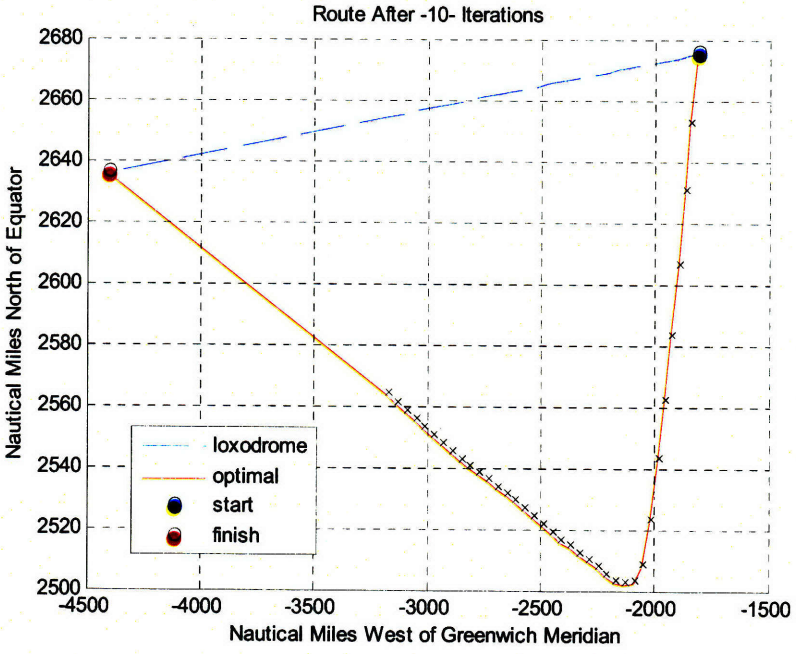


Figure 27 Output of 10th iteration

7.2.2 Sensitivity Analysis

Sensitivity analysis examines how changes in constraints affect the structure of the solution. For start, suppose the two safety constraints are lifted. This assumption redefines the problem as that of seeking the minimum-fuel route in rough seas, without any concerns about safety. Figure 28 shows the optimal route of the unconstrained problem.

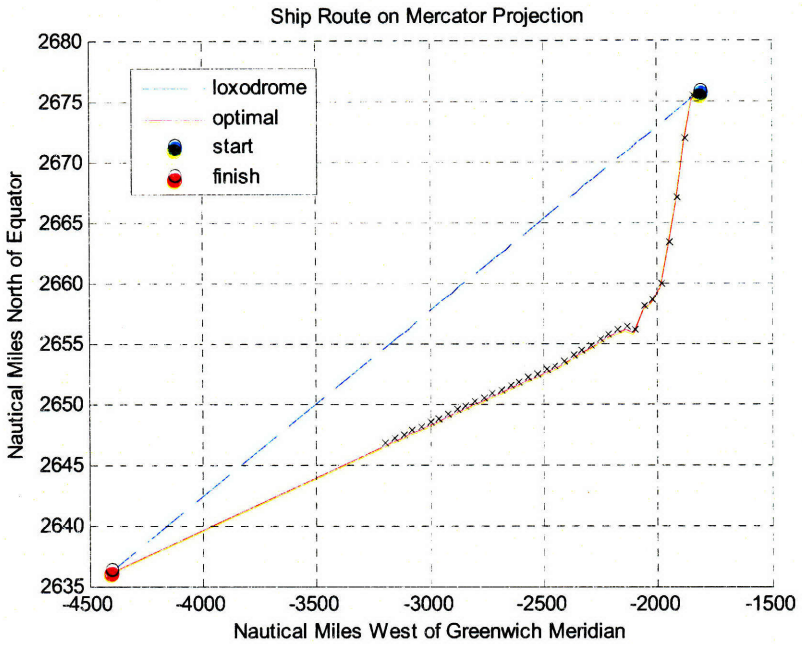


Figure 28 Optimal solution of the unconstrained problem

The optimal cost is 24.44 tons which, compared to the nominal loxodrome cost of 25.24 tons, indicates fuel savings of 3.2%. Even for this ideal case when safety is a priori guaranteed, the value of an optimal decision making tool is evident from this result. It shows that there is great potential in cost reduction because the program "knows" the effect of added resistance and can direct the ship towards the least expensive track. Without it, the captain would just follow the rhumb line route. Section 7.2.3 demonstrates the benefit of optimal routing in the more realistic case where constraints are imposed.

Next, the slamming constraint is relaxed and the code is run again accounting for deck wetness only. The solution is given in Figure 29.

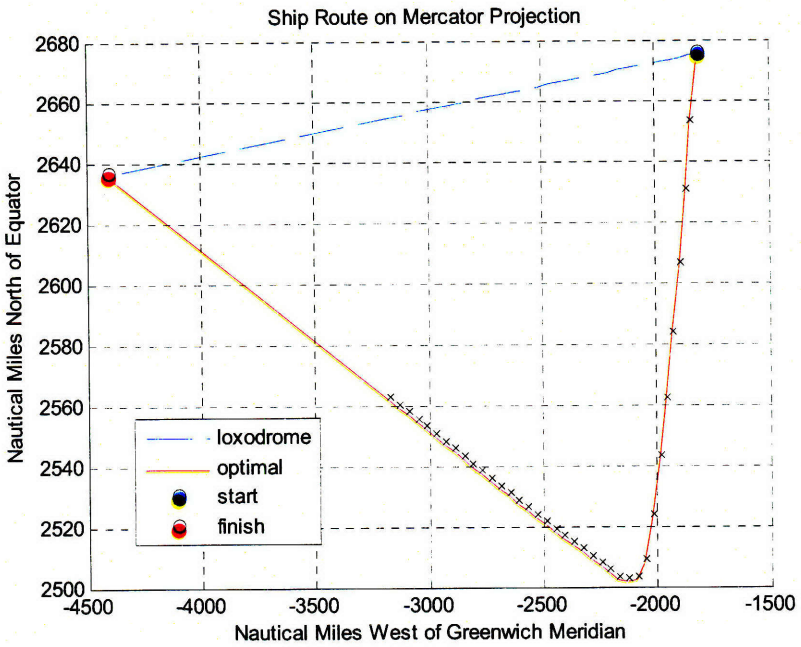


Figure 29 Optimal route without slamming

It is the same as the one shown in Figure 22, which implies that the slamming constraint is never enforced in the optimization process. Relaxing the deck wetness constraint and letting only slamming in the formulation results in the solution of Figure 30, which is precisely that of the unconstrained problem.

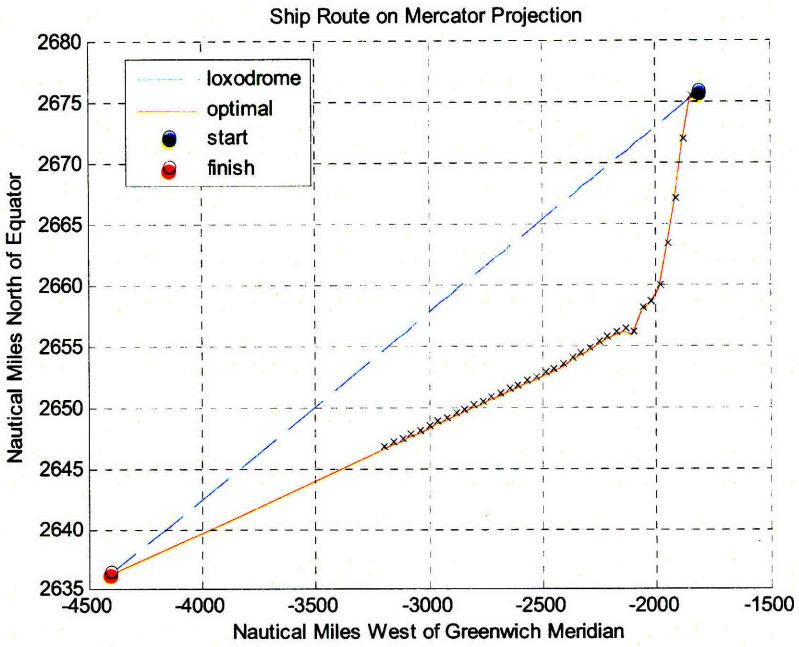


Figure 30 Optimal route without deck wetness

7.2.3 Fuel Savings

The favorable effect of optimal navigation is straightforward when no constraints are present. When safety constraints are added to the problem, it is not easy to figure out what would be the route that the captain would follow to avoid rogue waves (with no optimization in mind). If this route could be predicted then it would be easily compared with its proposed optimal counterpart and the cost difference would be determined.

In an attempt to quantify the difference between an optimal and a baseline feasible navigation track, in terms of fuel cost, we need to make some judicious assumptions. Suppose the captain knows nothing about the optimal route but wants to avoid the weather that will cause excessive slamming or bow submergence. From the standard maritime forecast he knows which areas exhibit most severe weather conditions. Furthermore, at any instant of time he can adjust his course and speed so that the ship's motions are confined to levels acceptable for the safety of the crew and cargo. It is reasonable to assume (and in fact this is exactly what happens in practice) that the captain will deviate only as much from the nominal course as necessary for the "constraints" to be satisfied. Similarly, he will adjust the speed (by tentatively reducing it) to a level as close to the nominal speed as possible, but such that the severity of motions is within the allowable limits. Once the storm dies down he will immediately re-plot his route to bring the ship to destination, following the straight line that connects the current position with the end location (assuming open seas and no obstacles).

In the subsequent analysis we establish a baseline-feasible solution which will represent the hypothetical route followed by the captain without any decision support system (i.e. optimal router). We then compare this solution to the optimal solution given by the computer code.

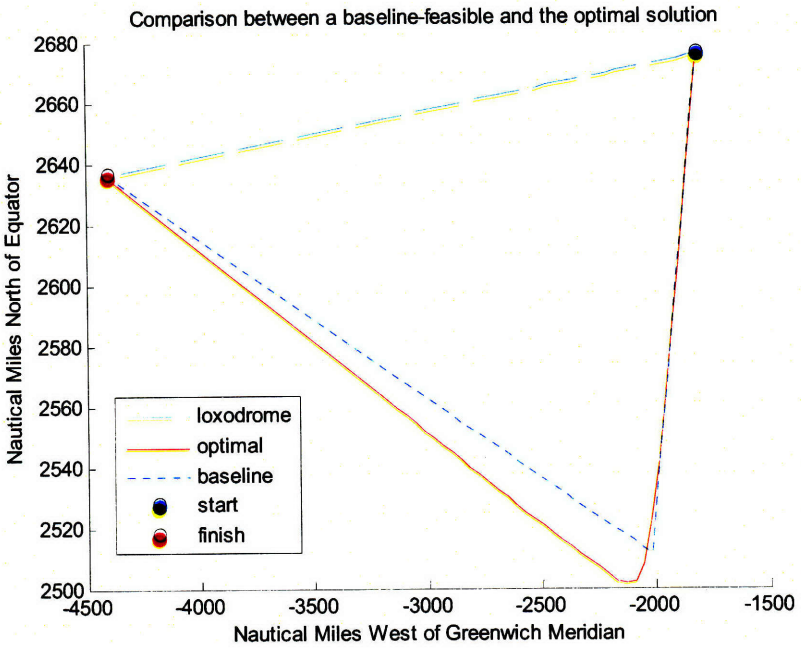


Figure 31 A baseline - feasible route close to the optimal

In the baseline solution the initial course is taken to be exactly that of the optimal one, as shown in Figure 31. Along this course the captain finds (by trial and error) that the speed which causes the least violent motions to the ship is 9.79 knots. He maintains this course up to the point where the appeased weather allows him to turn around and close in on the original rhumb line, maintaining a constant speed of 14 knots. Figure 31 shows how close the assumed baseline is to the optimal solution. Yet, the fuel cost of the former is 32.15 tons which is 14% higher than the optimal. Even if the shrewd captain finds exactly the right heading (coincident with the optimal), there is no way for him to know the optimal speed profile along this course, so he will just settle to a constant speed as discussed above. The optimal router, on the other hand, is fully cognizant of what the best speed policy should be at each time step, and this fact alone may result in significant savings.

In this particular example the feasible route is neighboring to the optimal. In reality, it is not at all certain that the route chosen by the operator will be nearly as close. It can therefore be anticipated that in some realistic scenarios the fuel savings may exceed 14%. On the opposite extreme, let us define a different feasible solution exploiting the

iterative nature of the IDP solver. Figure 26 depicts the output of the 6th iteration which, in fact, is the first optimal route computed by the program that satisfies all the constraints (as explained in chapter 6). Figure 32 shows its proximity to the true optimum. We consider it a "near-optimal" solution. The cost function is 2.8% higher than the optimal, which signifies a notable margin of improvement. In the unlikely event that a captain was able to guess the near-optimal solution, the additional savings corresponding to the 2.8% margin underlines the unique advantage of optimal decision making in ship routing. Table 9 provides a summary of this section's findings.

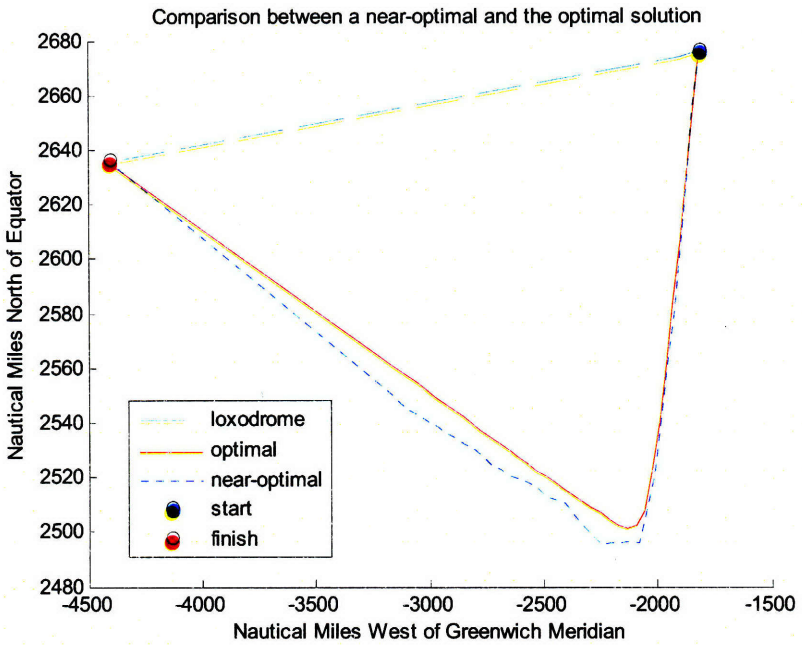


Figure 32 A near-optimal route from early iterations

Table 9 Fuel savings summary

	Description	Value
unconstrained	Cost of rhumb line route (tons)	25.2426
	Cost of optimal route (tons)	24.44
	Savings (%)	3.2
constrained	Cost of baseline feasible route (tons)	32.1533
	Cost of optimal route (tons)	27.5878
	Savings (%)	14

7.3 Conclusions

The fundamental result of this work is an understanding of the deterministic minimum-fuel navigation problem and a proposed approach to its solution. This paper stresses the importance of constructing a mathematical model which best encompasses the governing physics of the real problem. Several algorithms exist for the solution of optimal control problems. Regardless of which particular method is employed, the algorithm should be scrutinized and the results carefully checked for sanity.

Apart from the above general remarks, a few noteworthy observations can be summarized here with regard to the findings of this paper.

(1) The importance of added resistance is paramount to optimal ship routing. Its effect should be explicitly modeled in the mathematical formulation and its magnitude should be computed using appropriate hydrodynamic tools.

(2) Dynamic Programming appears to offer an attractive methodology for the solution of the problem. In particular, it circumvents the thorny questions of existence and uniqueness of solution and conditions of optimality. At the same time it is easy to implement on a personal computer.

(3) The accuracy of the forecast determines the accuracy of the final solution. Spatial resolution is also a driving factor since the calculations for any given location are based on the closest neighboring grid point.

(4) Back-fitting the forecast data to a standard spectrum is an undesirable inevitability. Ideally the spectrum computed by the WAM numerical model should be used, but this has no practical value with the current computational power available.

(5) The iterative nature of Iterative Dynamic Programming makes it more fit to be coded in a compiled language such as FORTRAN or C, rather than MATLAB. It is expected that the code will run faster in these languages.

(6) The overarching conclusion of the entire work presented in this thesis is the indisputable contribution of optimal routing software to the reduction of fuel cost of a ship in a seaway.

7.4 Recommendations for Future Work

Future extension to the current work could include validation of the deterministic minimum fuel routing results using the framework of Linear Quadratic control theory. Although the problem in hand is nonlinear, an LQR feedback control law can be derived for small perturbations of the nominal solution. The convergence of successive such perturbations can be explored and the solution can be compared with the current results.

Great circle navigation follows the true shortest path. Although route planning and plotting is predominantly based on the loxodrome, it would be interesting to explore the savings in fuel consumption following an optimal route along the orthodrome. For this to be of practical significance, an onboard automatic optimal controller could be designed that would eliminate the major shortfall of great circle routing, which is the need to constantly change the ship's course. Furthermore, a system of this type could deliver optimal rudder action (active steering) that would minimize the drag continuously throughout the elapsed time between two consecutive forecasts (and hence between two consecutive speed and course changes).

The effect of other components of resistance can also be examined. The existing model may be adapted to include these components, as well as additional safety constraints important for special applications (e.g. rolling accelerations of container ships). Propulsion phenomena that are not accounted for in the current model may be incorporated. One example is the change in the propulsion coefficients and propeller efficiency due to the presence of waves (Faltinsen et al., 1980).

Most importantly, optimal routing is intrinsically a stochastic rather than a deterministic problem. It is the present structure of the numerical wave forecasts that dictates a deterministic formulation of the problem. Statistical information that can be derived from the forecast models, either in their current or future form, can lead to the revision of the routing problem as a stochastic diffusion process. An interesting extension of the work presented in this paper would entail the formulation and solution of such a stochastic optimal control problem.

REFERENCES

1. Allsopp, T., A. Mason, and A. Philpott *Optimal sailing routes with uncertain weather.* in *35th Annual Conference of the Operational Research Society of New Zealand.* 2000. Victoria University of Wellington.

2. Anderson, B.D.O. and J.B. Moore, *Optimal control : linear quadratic methods.* 2007, Mineola, N.Y.: Dover Publications.

3. Azaron, A. and F. Kianfar, *Dynamic shortest path in stochastic dynamic networks: Ship routing problem.* EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, 2003. **144**(1): p. 138-156.

4. Beck, R.F., W. E. Cummins, J. F. Dalzell, P. Mandel, and W. C. Webster, *Motions in Waves*, in *Principles of Naval Architecture* E.V. Lewis, Editor. 1989, Society of Naval Architects and Marine Engineers: Jersey City.

5. Bellman, R.E., *Dynamic programming.* 2003, Mineola, N.Y.: Dover Publications.

6. Bijlsma, S.J., *On minimal-time ship routing.* 1975, Staatsuitgeverij: 's-Gravenhage.

7. Bijlsma, S.J., *A computational method for the solution of optimal control problems in ship routing.* Navigation., 2001. **48**(3): p. 145-154.

8. Bijlsma, S.J., *On the applications of optimal control theory and dynamic programming in ship routing.* Navigation., 2002. **49**(2): p. 71-80.

9. Bijlsma, S.J., *A computational method in ship routing using the concept of limited manoeuvrability.* Journal of Navigation, 2004. **57**(3): p. 357-369.

10. Bleick, W.E., and F.D. Faulkner, *Minimal-Time Ship Routing.* Journal of Applied Meteorology, 1965. **4**(2): p. 217-221.

11. Breslin, J.P. and P. Andersen, *Hydrodynamics of ship propellers.* Cambridge ocean technology series, 3. 1994, Cambridge [England]; New York: Cambridge University Press.

12. Bryson, A.E., *Dynamic optimization.* 1999, Menlo Park, CA: Addison Wesley Longman.

13. Bryson, A.E. and Y.-C. Ho, *Applied optimal control : optimization, estimation, and control.* 1975, New York: Wiley.

14. Denardo, E.V., *Dynamic programming : models and applications*. 2003, Mineola, N.Y.: Dover Publications.

15. Eggers, K.W.H., S.D.Sharma, and L.W.Ward, *An assessment of some experimental methods for determining the wavemaking characteristics of a ship form.* Transactions, Society of Naval Architects and Marine Engineers, 1967. **75**: p. 112-157.

16. Emmanouil, G., G. Galanis, and G. Kallos, *Assimilation of radar altimeter data in numerical wave models: an impact study in two different wave climate regions.* Annales geophysicae /, 2007. **25**(3): p. 581-596.

17. Faltinsen, O.M., K. J. Minsaas, N. Liapis, and S. O. Skørdal, *Prediction of Resistance and Propulsion of a Ship in a Seaway*, in *13th Symposioum on Naval Hydrodynamics*, T. Inui, Editor. 1980, The Shipbuilding Research Association of Japan: Tokyo. p. 505-530.

18. Faltinsen, O.M., *Sea loads on ships and offshore structures*. Cambridge ocean technology series, [1]. 1990, Cambridge; New York: Cambridge University Press.

19. Faltinsen, O.M., *Hydrodynamics of high-speed marine vehicles*. 2005, Cambridge; New York: Cambridge University Press.

20. Frankel, E.G., H. Chen, and C. National Maritime Research, *Optimization of ship routing*. 1980, Kings Point, N.Y.: National Maritime Research Center.

21. Gerritsma, J., and W. Beukelman, *Analysis of the resistance in waves of a fast cargo ship.* International Shipbuilding Progress, 1972. **19**(217): p. 285-293.

22. Grimmett, G. and D. Stirzaker, *Probability and random processes*. 2004, Oxford, U.K.; New York, N.Y.: Oxford University Press.

23. Hagiwara, H., *Weather routing of (sail-assisted) motor vessels*. 1985.

24. Haltiner, G.J., H.D. Hamilton, and G.'Arnason, *Minimal-Time Ship Routing.* Journal of Applied Meteorology, 1962. **1**(1): p. 1-7.

25. Hinnenthal, J., and Ø. Saetra, *Robust Pareto-Optimal Routing of Ships Utilizing ensemble weather forecasts*, in *Maritime Transportation and Exploration of Ocean and Coastal Resources*. 2005, Taylor & Francis Group: London.

26. Holtrop, J., and G.G.J. Mennen *An approximate power prediction method.* International Shipbuilding Progress, 1982. **29**(335): p. 166-170.

27. Hotrop, J., *A statistical re-analysis of resistance and propulsion data.* International Shipbuilding Progress, 1984. **31**(363): p. 272-276.

28. James, R.W., *Application of wave forecasts to marine navigation.* 1957, Navy Hydr. Office: Washington D.C.

29. Journée, J.M.J., *Motions, Resistance and Propulsion of a Ship in Regular Head Waves.* 1976, Delft University of Technology.

30. Journée, J.M.J., *Motions and Resistance of a Ship in Regular Following Waves.* 1976, Delft University of Technology.

31. Journée, J.M.J., and J.H.C. Meijers, *Ship routeing for optimum performance.* 1980, Delft University of Technology.

32. Kirk, D.E., *Optimal control theory : an introduction.* 2004, Mineola, N.Y.: Dover Publications.

33. Komen, G.J., *Dynamics and modelling of ocean waves.* 1994, Cambridge; New York, NY, USA: Cambridge University Press.

34. Larson, R.E. and J.L. Casti, *Principles of dynamic programming.* Control and systems theory, v. 7. 1978, New York: M. Dekker.

35. Lo, H.K. and M.R. McCord, *Adaptive ship routing through stochastic ocean currents: general formulations and empirical results.* Transportation research. Part A, Policy and practice., 1998. **32**(7): p. 547.

36. Luus, R., *Iterative dynamic programming.* 2000, Boca Raton: Chapman & Hall/CRC.

37. Massel, S.R., *Ocean surface waves : their physics and prediction.* Advanced series on ocean engineering, v. 11. 1996, Singapore; River Edge, NJ: World Scientific.

38. Nakos, D.E., and P. D. Sclavounos. *Ship motions by a three dimensional Rankine panel method.* in *18th Symposium on Naval Hydrodynamics.* 1990. Ann Arbor, Michigan.

39. Nakos, D.E. and P.D. Sclavounos, *Kelvin Wakes and Wave Resistance of Cruiser- and Transom-Stern Ships.* Journal of ship research., 1994. **38**(1): p. 9.

40. Newman, J.N., *Marine hydrodynamics.* 1977, Cambridge, Mass.: MIT Press.

41. Ochi, M.K., *Ocean waves : the stochastic approach*. Cambridge ocean technology series, 6. 1998, Cambridge, U.K.; New York: Cambridge University Press.

42. Padhy, C.P., D. Sen, and P.K. Bhaskaran, *Application of wave model for weather routing of ships in the North Indian Ocean.* Natural Hazards, 2008. **44**(3): p. 373-385.

43. Perakis, A.N., and N.A. Papadakis, *deterministic minimal time vessel routing.* Operations Research, 1990. **38**(3): p. 426-438.

44. Richardus, P. and R.K. Adler, *Map projections : for geodesists, cartographers and geographers*. 1972, Amsterdam: North-Holland.

45. Sclavounos, P.D., and D.E.Nakos. *Seakeeping and added resistance of I.A.C.C. yachts by Three-dimensional rankine panel methods*. in *11th Chesapeake Sailing Yacht Symposium*. 1993. Annapolis MD.

46. Sclavounos, P.D., *Computation of wave ship interactions*, in *Advances in Marine Hydrodynamics*, M.Ohkusu, Editor. 1996, Computational Mechanics Publications: Southampton UK. p. 233-278.

47. Sclavounos, P.D., *Ocean Wave Interactions with Ships: Lecture Notes*. 2007, Massachusetts Institute of Technology: Cambridge MA.

48. Stengel, R.F., *Optimal control and estimation*. Dover books on advanced mathematics. 1994, New York: Dover Publications.

49. Todd, F.H., *Series 60 methodical experiments with models of single-screw merchant ships*. 1963, Washington: For sale by the Supt. of Docs., U.S. Govt. Print. Off.

50. Tucker, M.J. and E.G. Pitt, *Waves in ocean engineering*. Elsevier ocean engineering book series, v. 5. 2001, Amsterdam; New York: Elsevier.

51. Turabian, K.L., *A manual for writers of research papers, theses, and dissertations : Chicago style for students and researchers*. Chicago guides to writing, editing, and publishing. 2007, Chicago: University of Chicago Press.

52. Van Manen, J.D., and P. Van Oossanen, *Resistance*, in *Principles of Naval Architecture* E.V. Lewis, Editor. 1988, Society of Naval Architects and Marine Engineers: Jersey City.

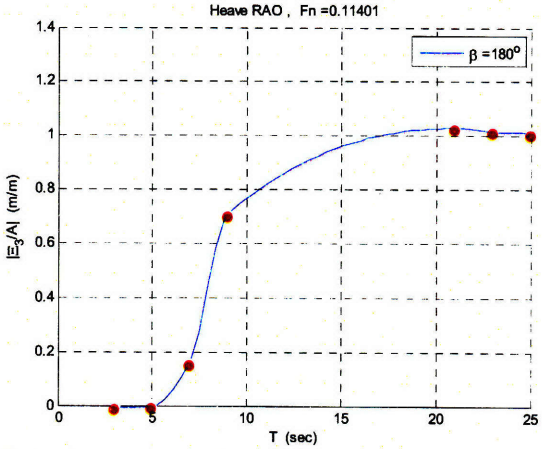
53. Van Manen, J.D., and P. Van Oossanen, *Propulsion*, in *Principles of Naval Architecture* E.V. Lewis, Editor. 1988, Society of Naval Architects and Marine Engineers: Jersey City.

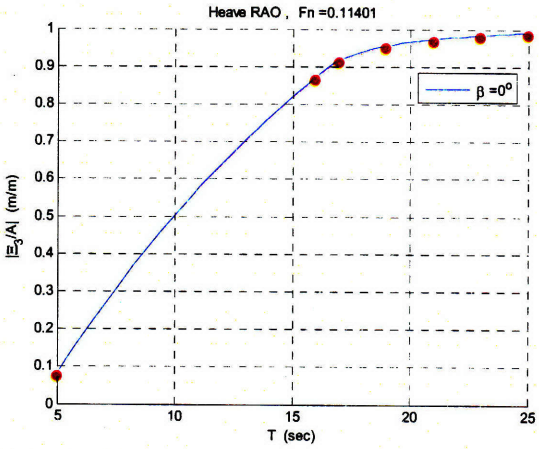
54. Woud, H.K. and D. Stapersma, *Design of propulsion and electric power generation systems*. 2002, London: IMarEST, Institute of Marine Engineering, Science and Technology.

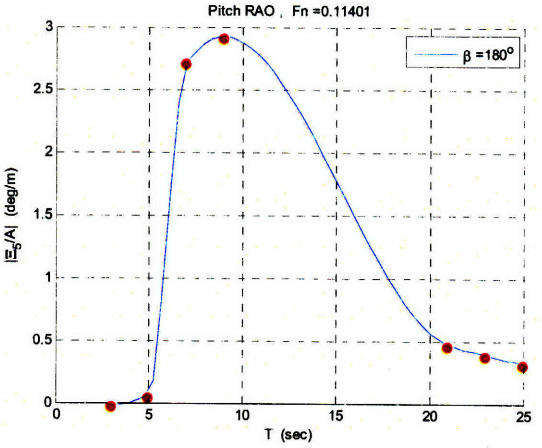
55. Zoppoli, R., *Minimum-Time routing as an N-Stage Decision Process.* Journal of Applied Meteorology, 1972. **11**(3): p. 429-435.

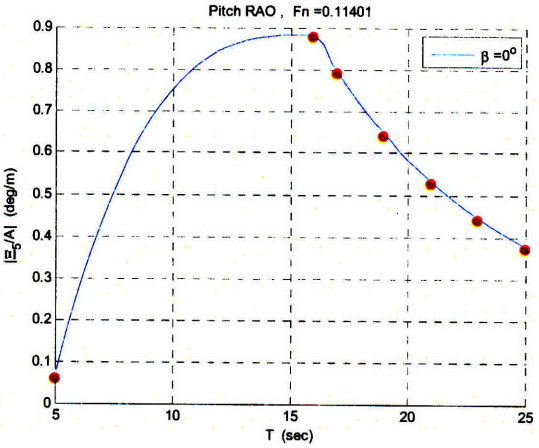
APPENDICES

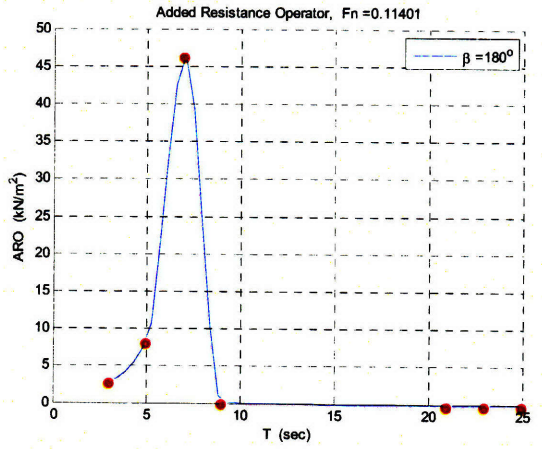
Appendix A: SWAN1 Output

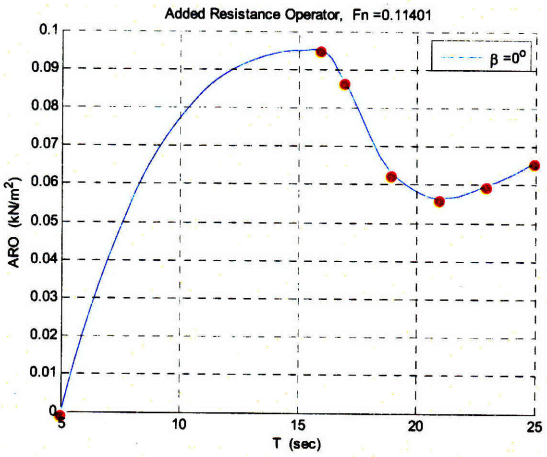


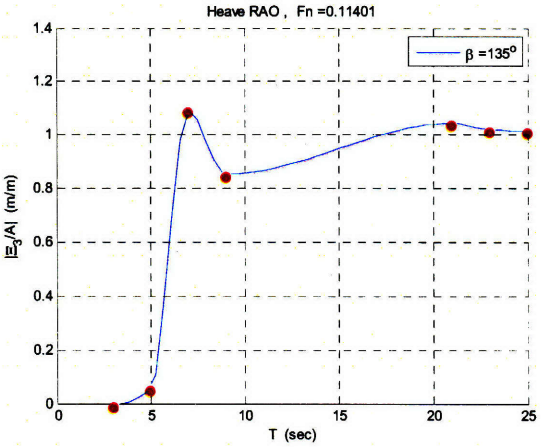


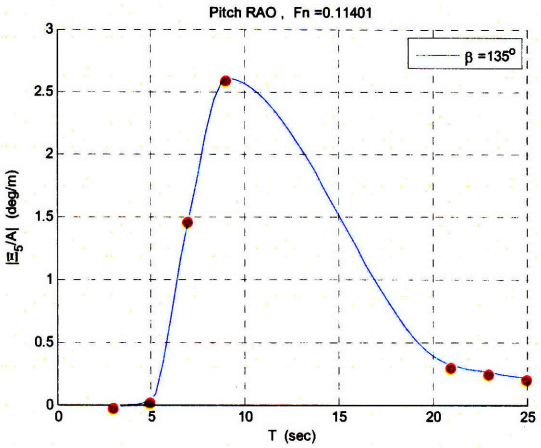


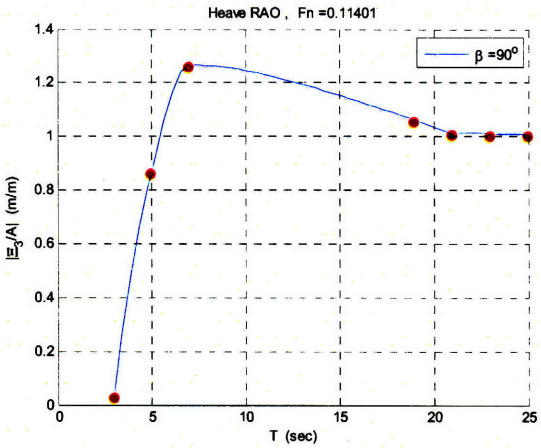


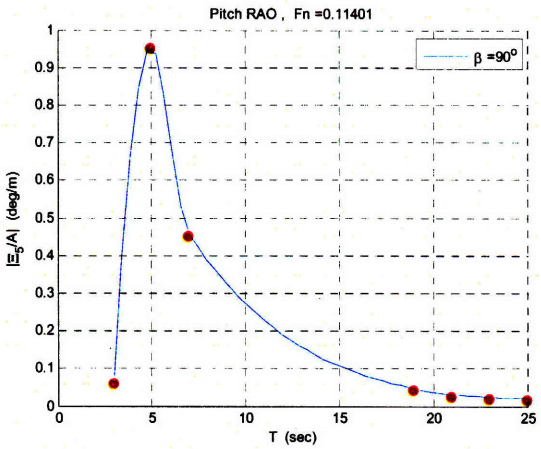


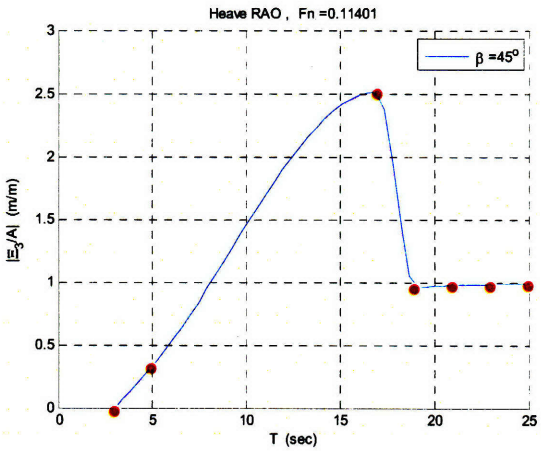


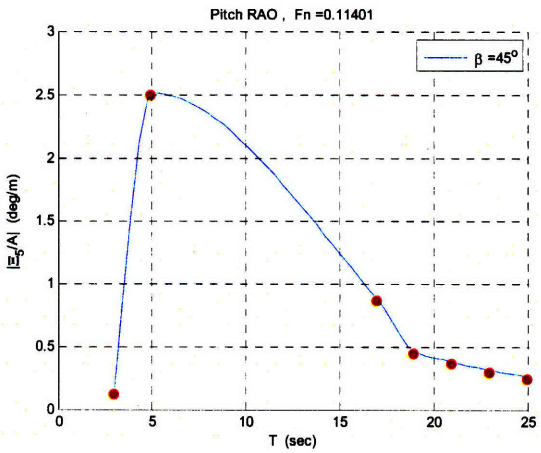

Heave RAO , Fn =0.11401
β =135°
|Ξ3/A| (m/m)
T (sec)

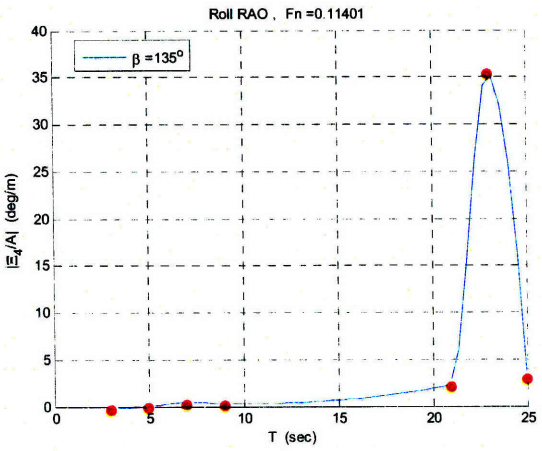

Pitch RAO , Fn =0.11401
β =135°
|Ξ5/A| (deg/m)
T (sec)

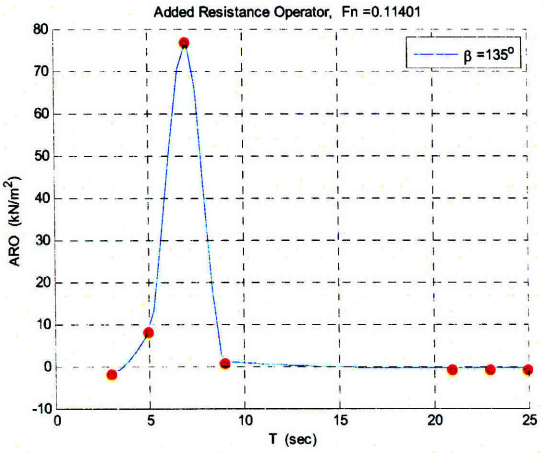

Heave RAO , Fn =0.11401
β =90°
|Ξ3/A| (m/m)
T (sec)

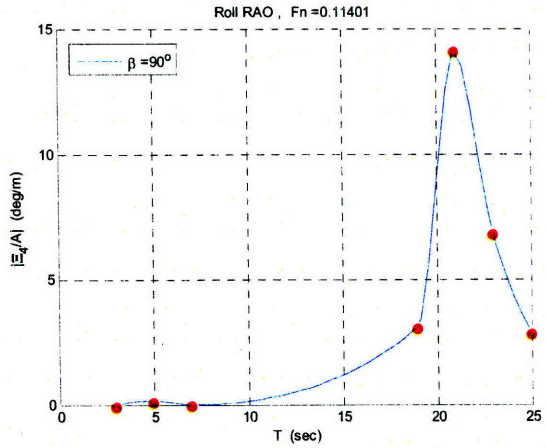

Pitch RAO , Fn =0.11401
β =90°
|Ξ5/A| (deg/m)
T (sec)

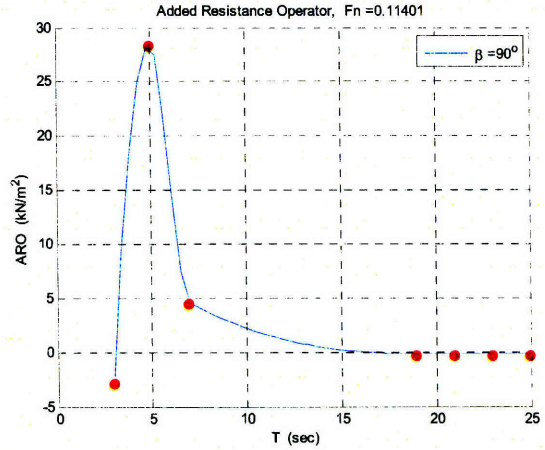

Heave RAO , Fn =0.11401
β =45°
|Ξ3/A| (m/m)
T (sec)

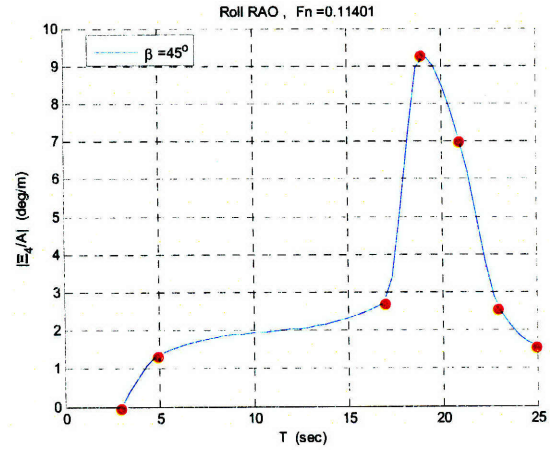

Pitch RAO , Fn =0.11401
β =45°
|Ξ5/A| (deg/m)
T (sec)

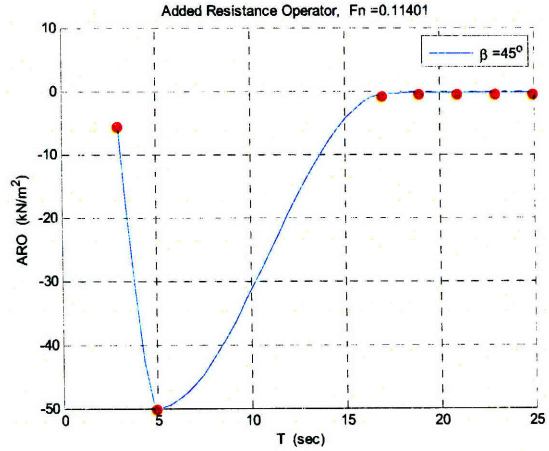

Roll RAO , Fn =0.11401
β =135°
|Ξ4/A| (deg/m)
T (sec)

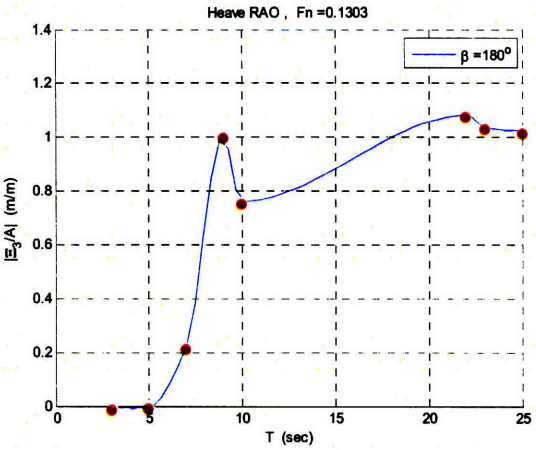

Added Resistance Operator, Fn =0.11401
β =135°
ARO (kN/m²)
T (sec)

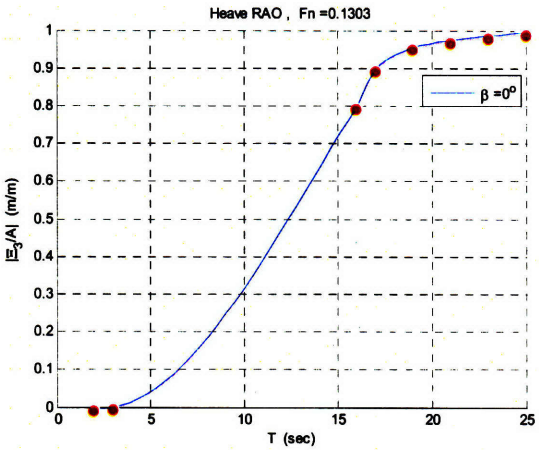

Roll RAO , Fn =0.11401
β =90°
|Ξ4/A| (deg/m)
T (sec)

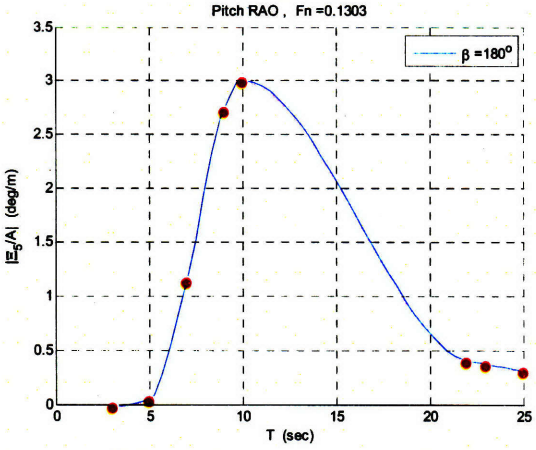

Added Resistance Operator, Fn =0.11401
β =90°
ARO (kN/m²)
T (sec)

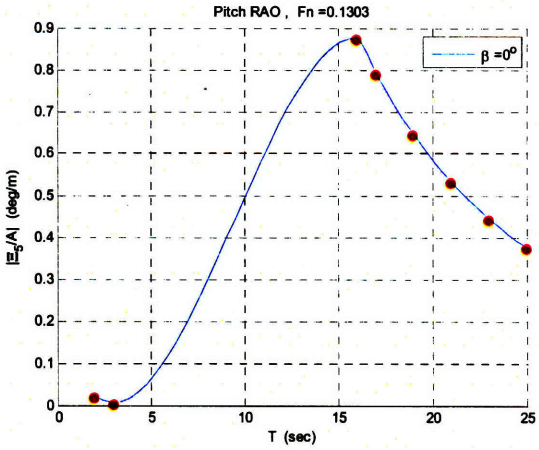

Roll RAO , Fn =0.11401
β =45°
|Ξ4/A| (deg/m)
T (sec)

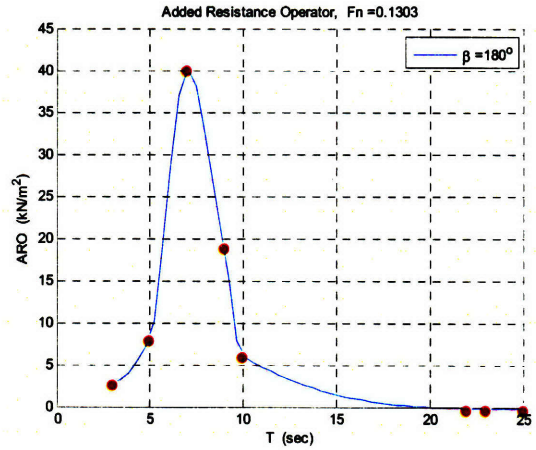

Added Resistance Operator, Fn =0.11401
β =45°
ARO (kN/m²)
T (sec)

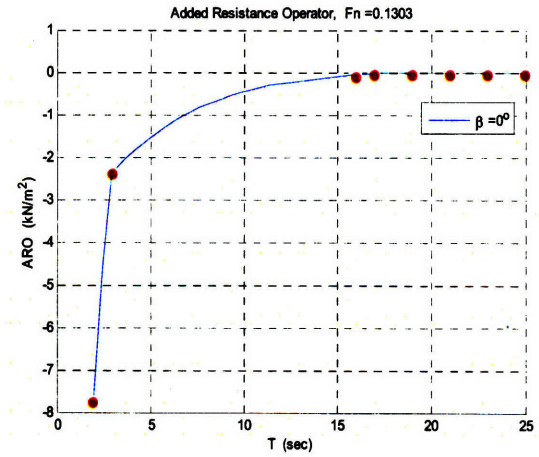

Heave RAO , Fn =0.1303
β =180°
|Ξ3/A| (m/m)
T (sec)

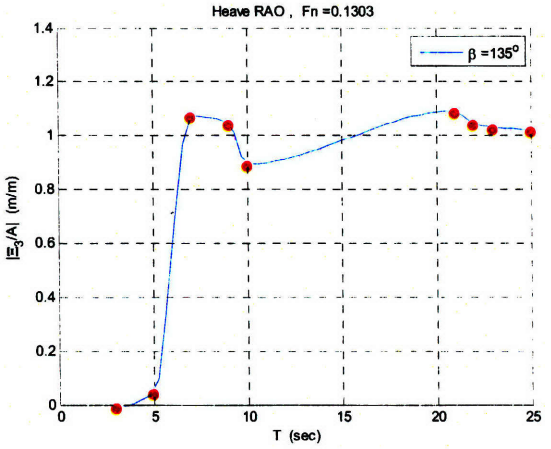

Heave RAO , Fn =0.1303
β =0°
|Ξ3/A| (m/m)
T (sec)

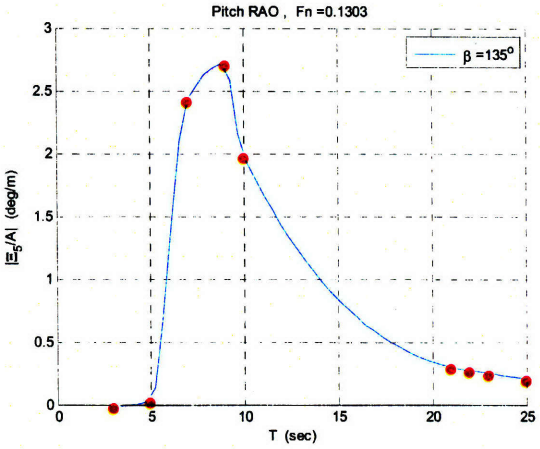

Pitch RAO , Fn =0.1303
β =180°
|Ξ5/A| (deg/m)
T (sec)

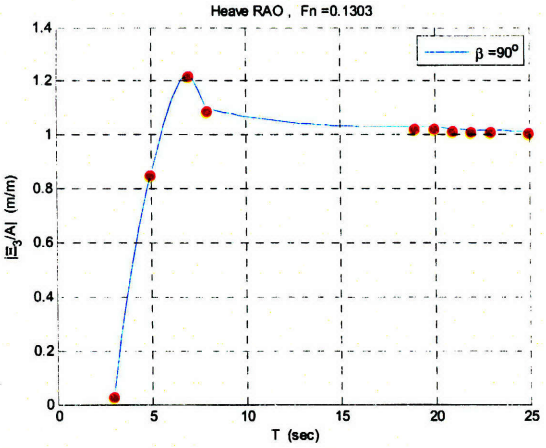

Pitch RAO , Fn =0.1303
β =0°
|Ξ5/A| (deg/m)
T (sec)

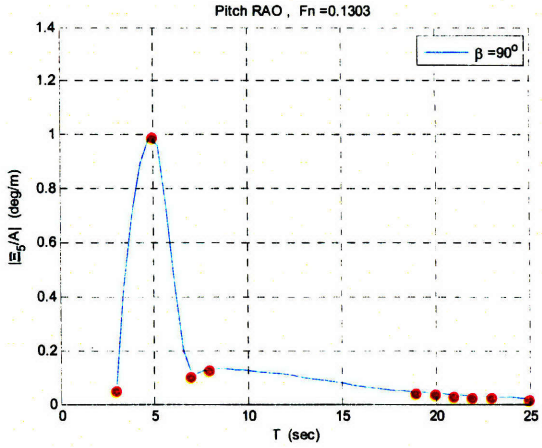

Added Resistance Operator, Fn =0.1303
β =180°
ARO (kN/m2)
T (sec)

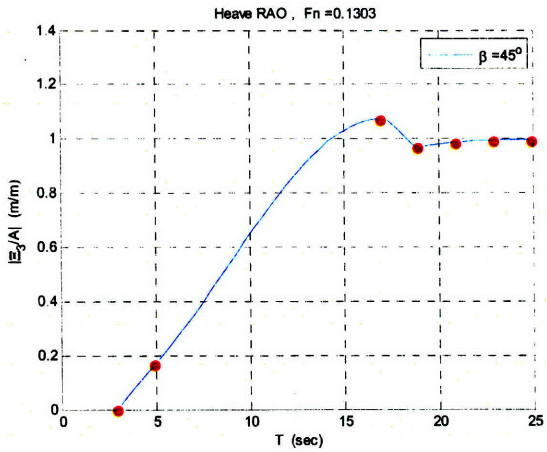

Added Resistance Operator, Fn =0.1303
β =0°
ARO (kN/m2)
T (sec)

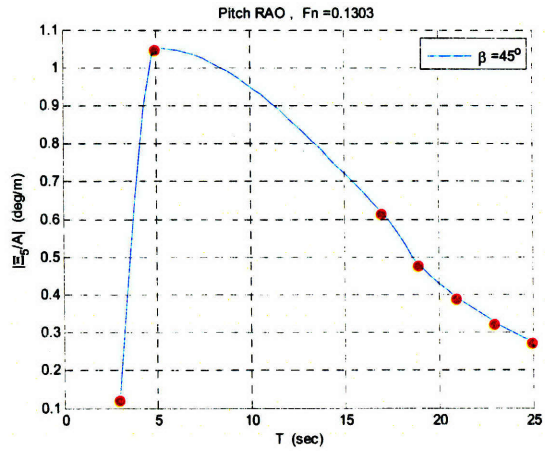

Heave RAO , Fn =0.1303
β =135°
|Ξ3/A| (m/m)
T (sec)

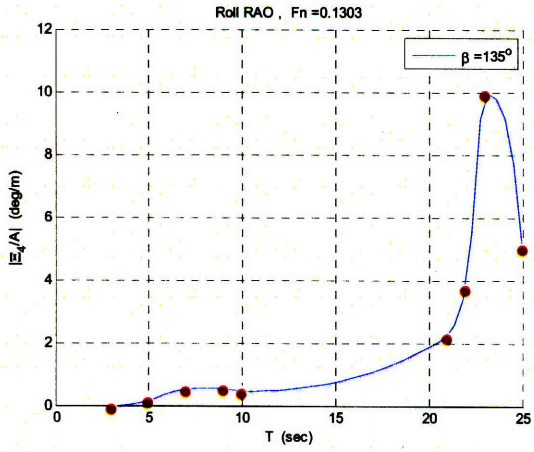

Pitch RAO , Fn =0.1303
β =135°
|Ξ5/A| (deg/m)
T (sec)

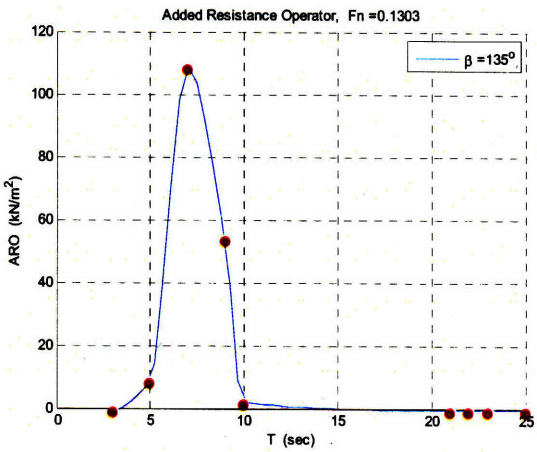

Heave RAO , Fn =0.1303
β =90°
|Ξ3/A| (m/m)
T (sec)

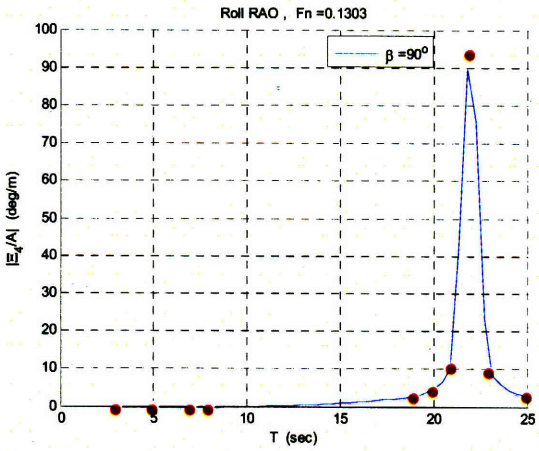

Pitch RAO , Fn =0.1303
β =90°
|Ξ5/A| (deg/m)
T (sec)

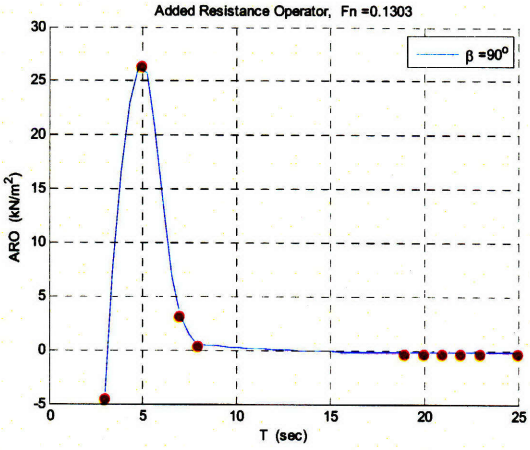

Heave RAO , Fn =0.1303
β =45°
|Ξ3/A| (m/m)
T (sec)

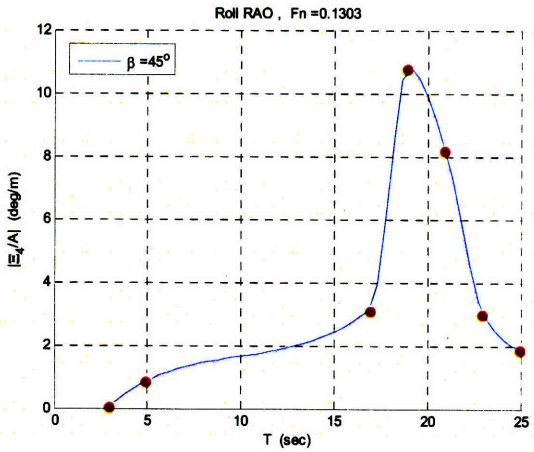

Pitch RAO , Fn =0.1303
β =45°
|Ξ5/A| (deg/m)
T (sec)

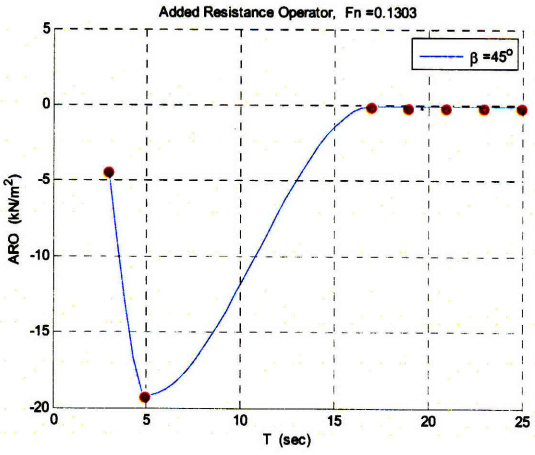

Roll RAO , Fn =0.1303
β =135°
|Ξ4/A| (deg/m)
T (sec)

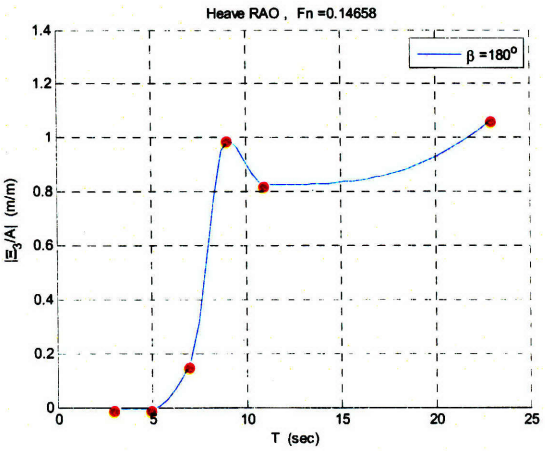

Added Resistance Operator, Fn =0.1303
β =135°
ARO (kN/m2)
T (sec)

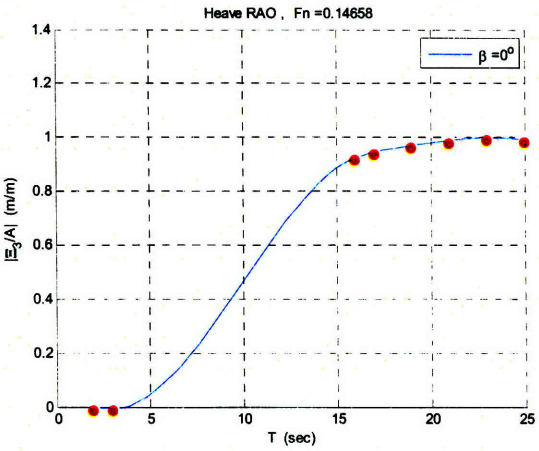

Roll RAO , Fn =0.1303
β =90°
|Ξ4/A| (deg/m)
T (sec)

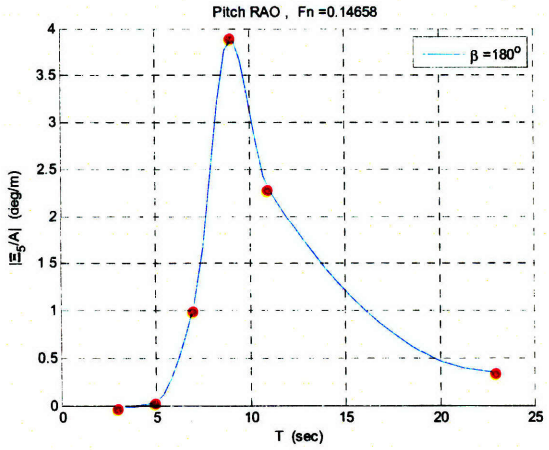

Added Resistance Operator, Fn =0.1303
β =90°
ARO (kN/m2)
T (sec)

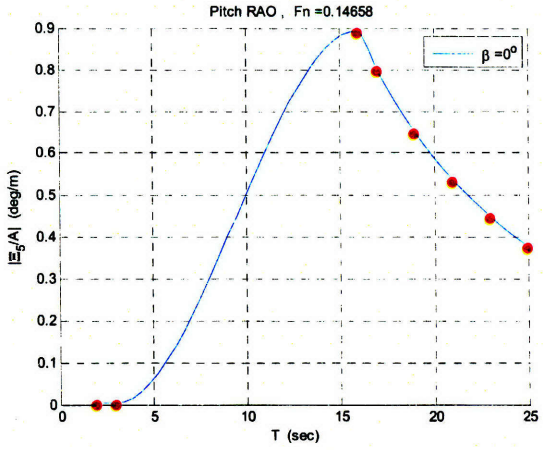

Roll RAO , Fn =0.1303
β =45°
|Ξ4/A| (deg/m)
T (sec)

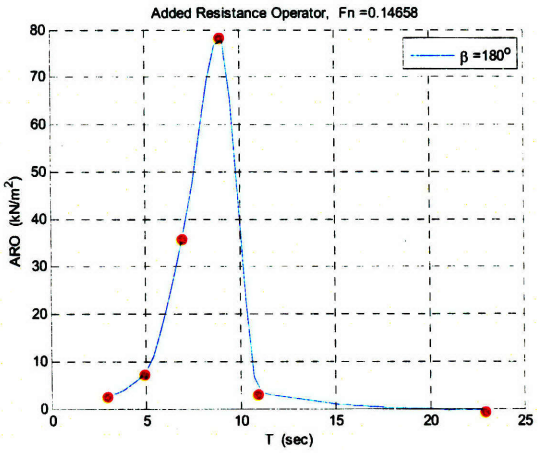

Added Resistance Operator, Fn =0.1303
β =45°
ARO (kN/m2)
T (sec)

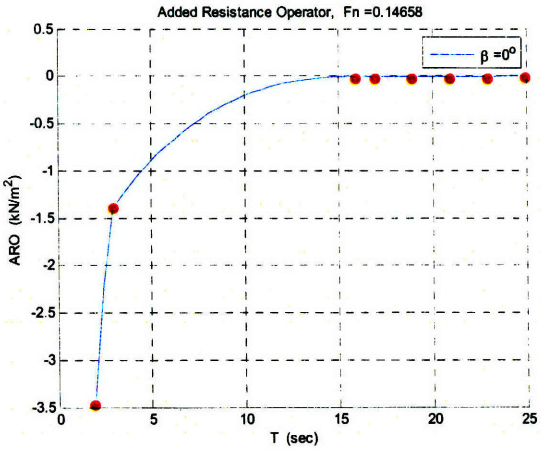

Heave RAO , Fn =0.14658
β =180°
|Ξ3/A| (m/m)
T (sec)

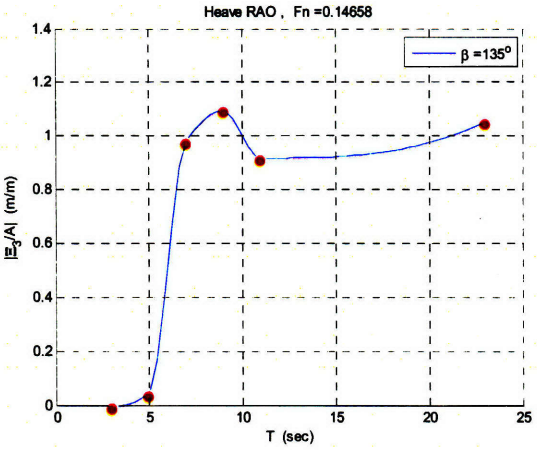

Heave RAO , Fn =0.14658
β =0°
|Ξ3/A| (m/m)
T (sec)

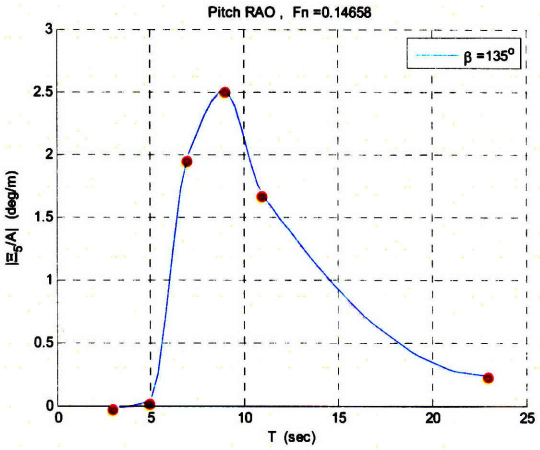

Pitch RAO , Fn =0.14658
β =180°
|Ξ5/A| (deg/m)
T (sec)

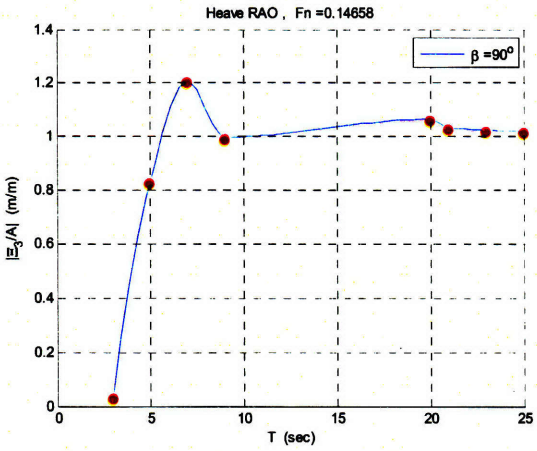

Pitch RAO , Fn =0.14658
β =0°
|Ξ5/A| (deg/m)
T (sec)

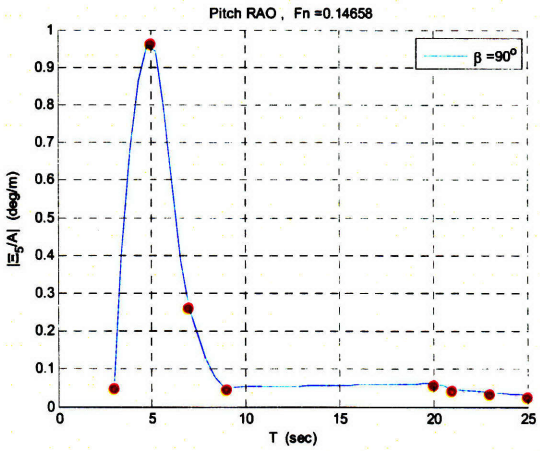

Added Resistance Operator, Fn =0.14658
β =180°
ARO (kN/m2)
T (sec)

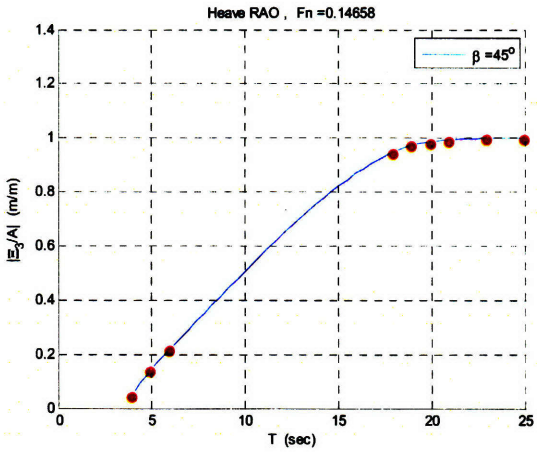

Added Resistance Operator, Fn =0.14658
β =0°
ARO (kN/m2)
T (sec)

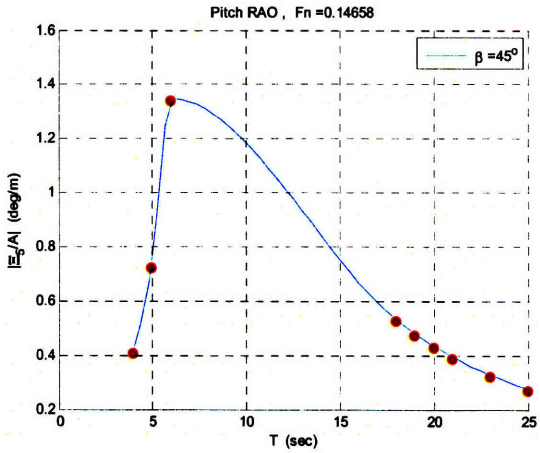

Heave RAO , Fn =0.14658
β =135°
|Ξ3/A| (m/m)
T (sec)

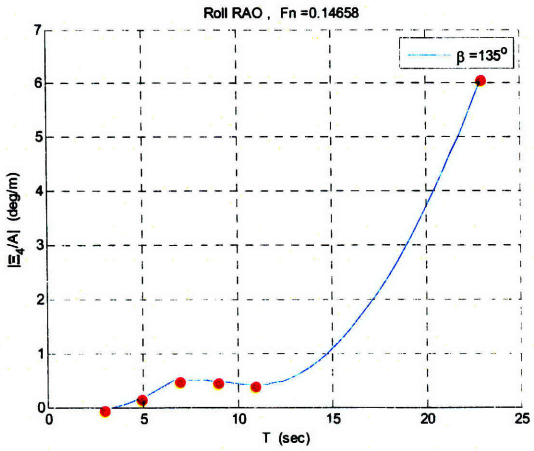

Pitch RAO , Fn =0.14658
β =135°
|Ξ5/A| (deg/m)
T (sec)

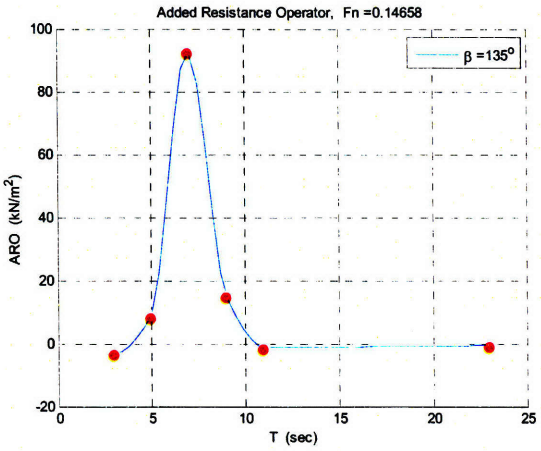

Heave RAO , Fn =0.14658
β =90°
|Ξ3/A| (m/m)
T (sec)

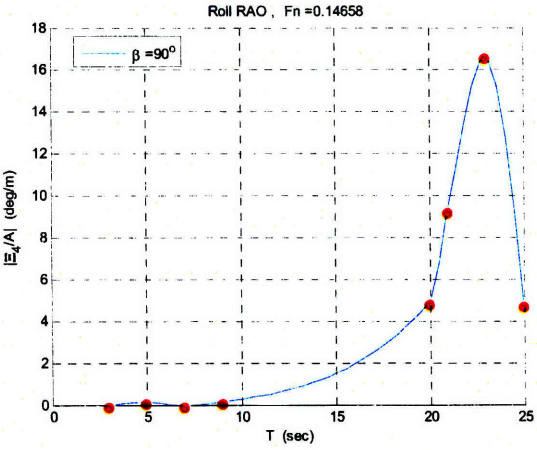

Pitch RAO , Fn =0.14658
β =90°
|Ξ5/A| (deg/m)
T (sec)

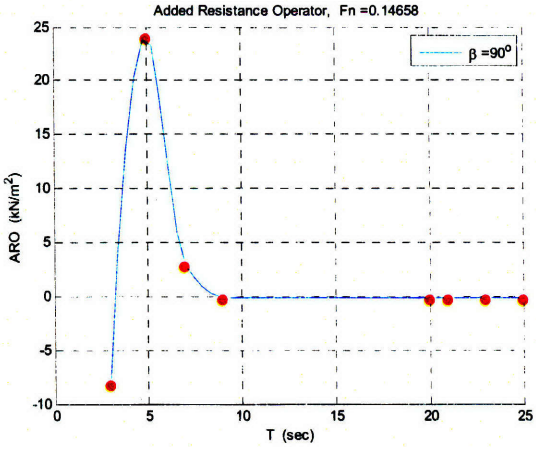

Heave RAO , Fn =0.14658
β =45°
|Ξ3/A| (m/m)
T (sec)

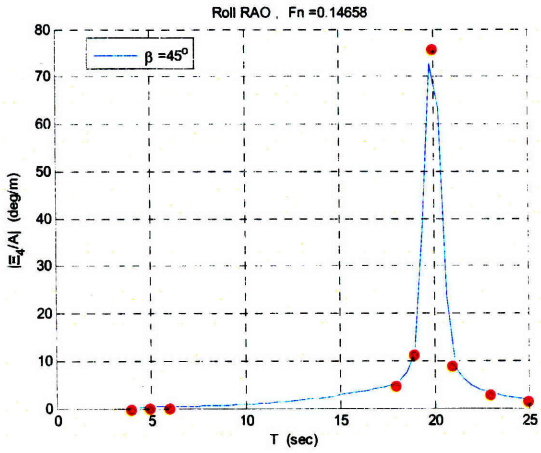

Pitch RAO , Fn =0.14658
β =45°
|Ξ5/A| (deg/m)
T (sec)

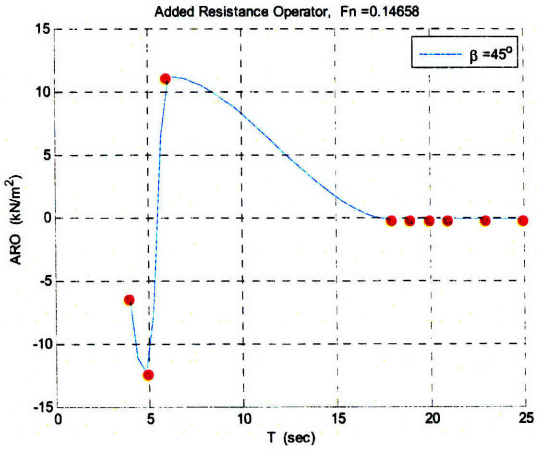

Roll RAO , Fn =0.14658
β =135°
|Ξ4/A| (deg/m)
T (sec)

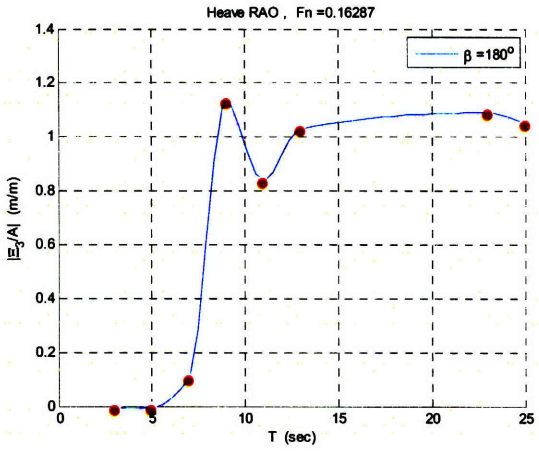

Added Resistance Operator, Fn =0.14658
β =135°
ARO (kN/m²)
T (sec)

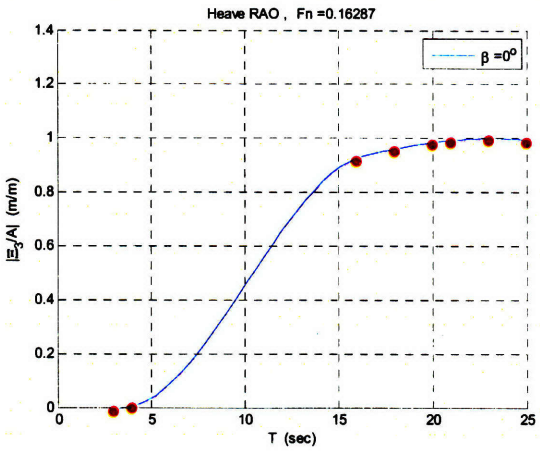

Roll RAO , Fn =0.14658
β =90°
|Ξ4/A| (deg/m)
T (sec)

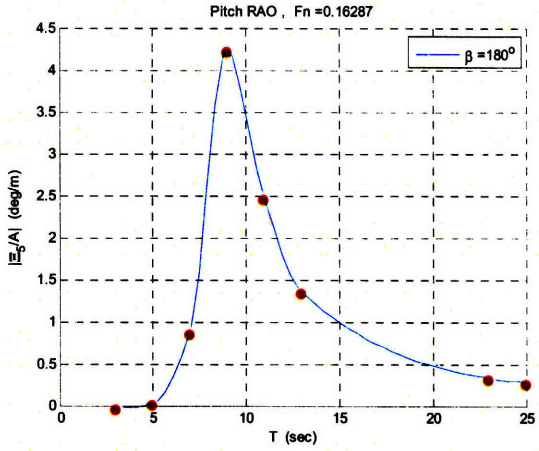

Added Resistance Operator, Fn =0.14658
β =90°
ARO (kN/m²)
T (sec)

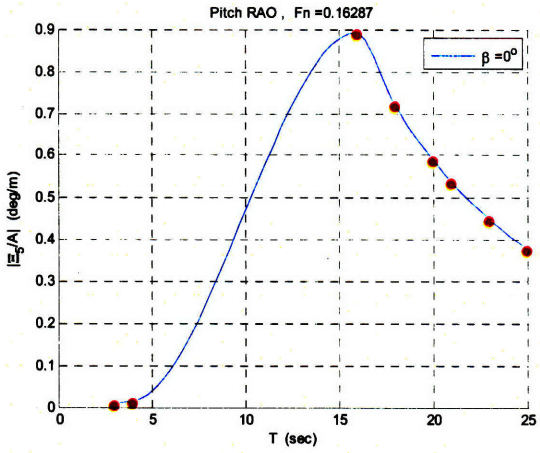

Roll RAO , Fn =0.14658
β =45°
|Ξ4/A| (deg/m)
T (sec)

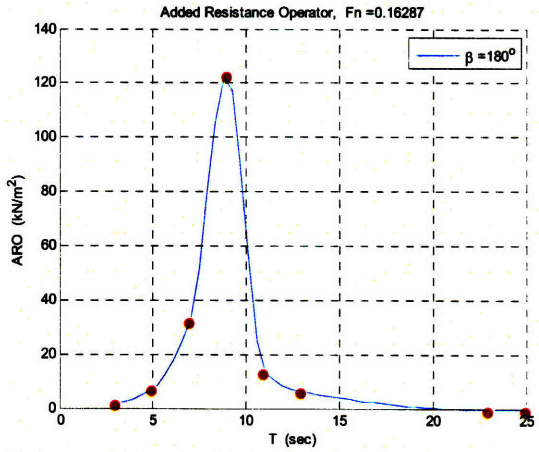

Added Resistance Operator, Fn =0.14658
β =45°
ARO (kN/m²)
T (sec)

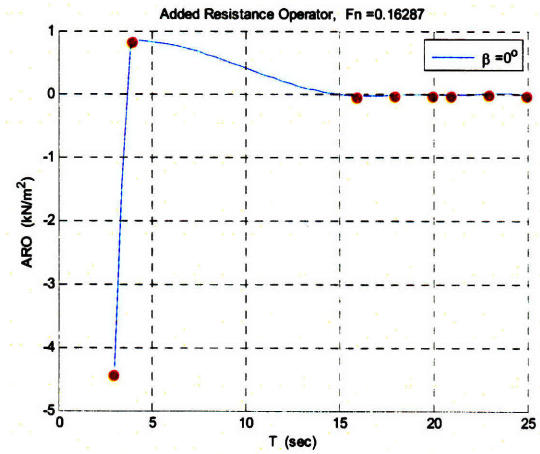

Heave RAO , Fn =0.16287
β =180°
|Ξ3/A| (m/m)
T (sec)

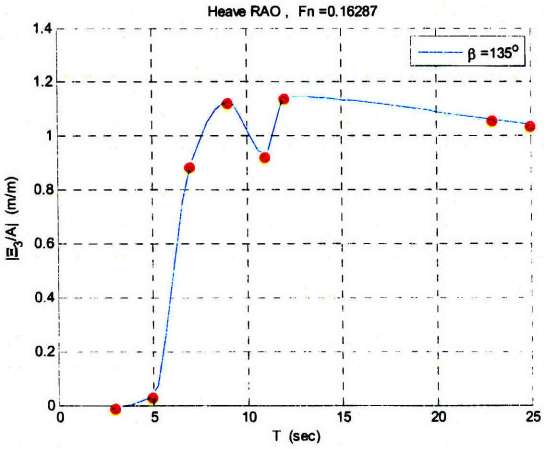

Heave RAO , Fn =0.16287
β =0°
|Ξ3/A| (m/m)
T (sec)

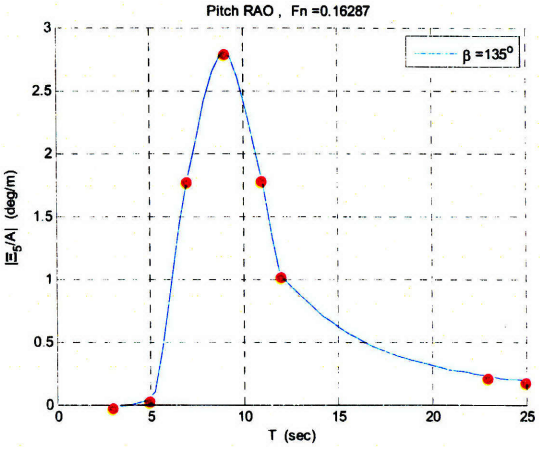

Pitch RAO , Fn =0.16287
β =180°
|Ξ5/A| (deg/m)
T (sec)

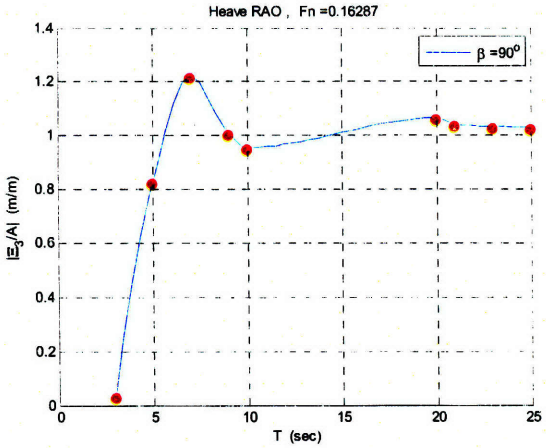

Pitch RAO , Fn =0.16287
β =0°
|Ξ5/A| (deg/m)
T (sec)

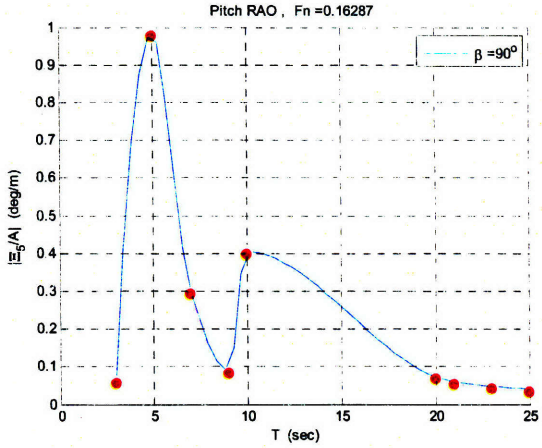

Added Resistance Operator, Fn =0.16287
β =180°
ARO (kN/m²)
T (sec)

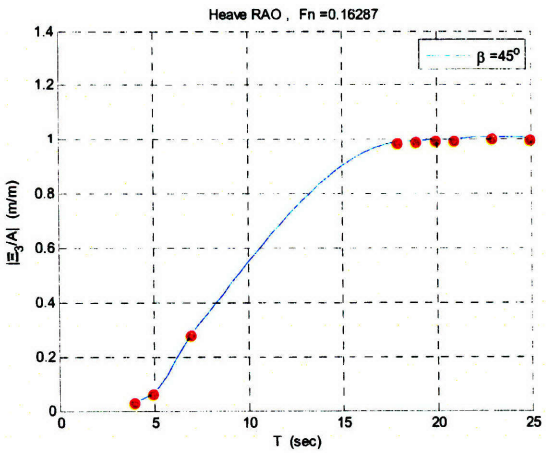

Added Resistance Operator, Fn =0.16287
β =0°
ARO (kN/m²)
T (sec)

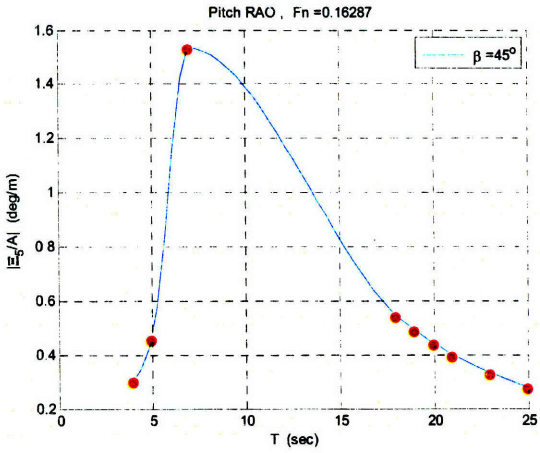

Heave RAO , Fn =0.16287
β =135°
|Ξ3/A| (m/m)
T (sec)

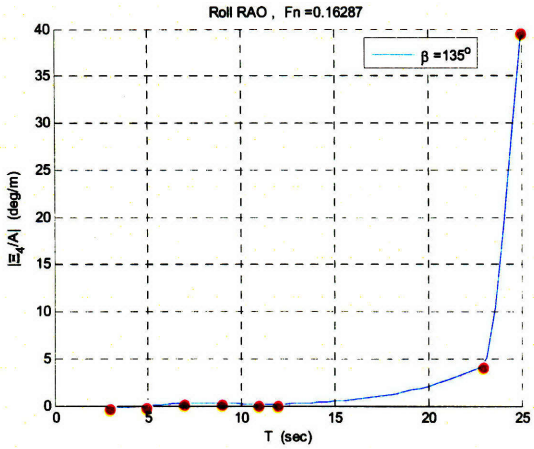

Pitch RAO , Fn =0.16287
β =135°
|Ξ5/A| (deg/m)
T (sec)

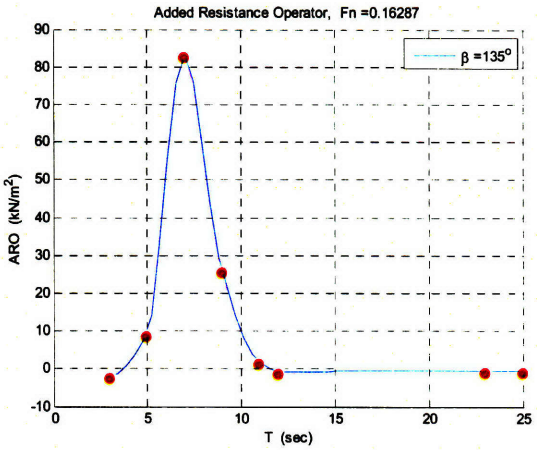

Heave RAO , Fn =0.16287
β =90°
|Ξ3/A| (m/m)
T (sec)

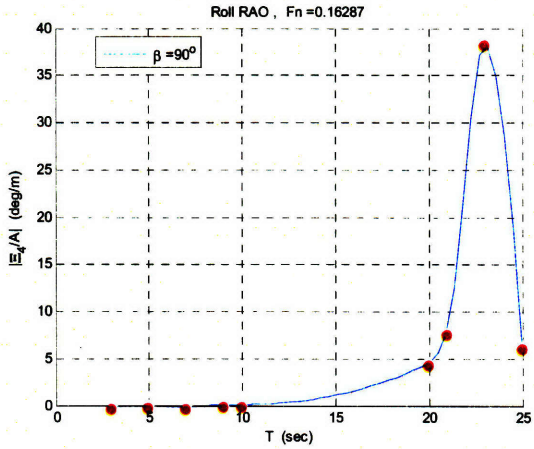

Pitch RAO , Fn =0.16287
β =90°
|Ξ5/A| (deg/m)
T (sec)

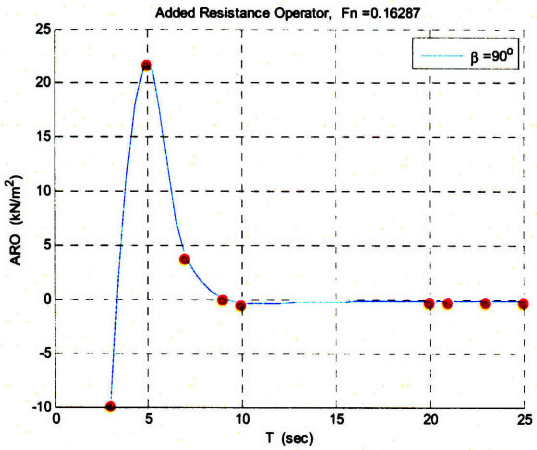

Heave RAO , Fn =0.16287
β =45°
|Ξ3/A| (m/m)
T (sec)

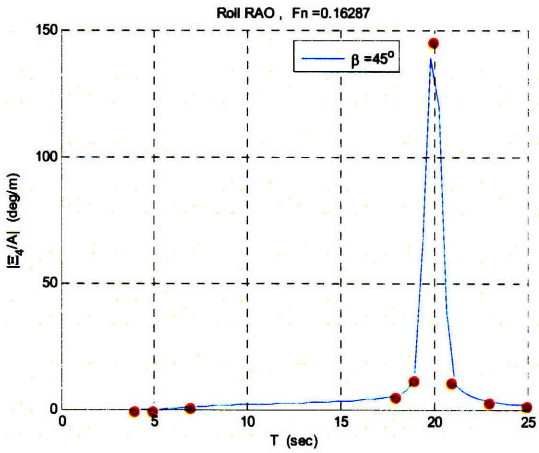

Pitch RAO , Fn =0.16287
β =45°
|Ξ5/A| (deg/m)
T (sec)

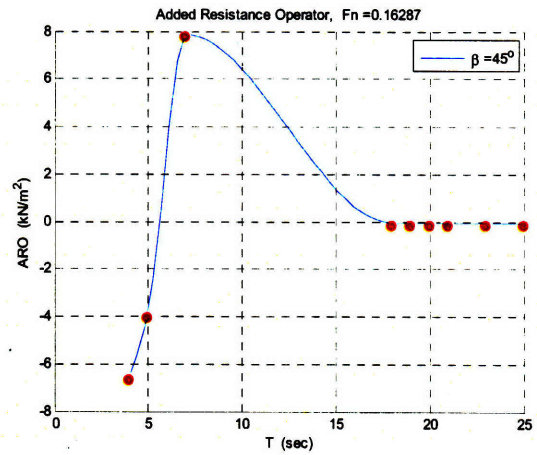

Roll RAO , Fn =0.16287
β =135°
|Ξ4/A| (deg/m)
T (sec)

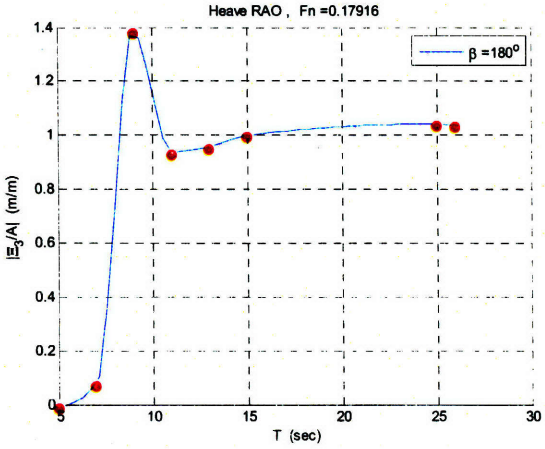

Added Resistance Operator, Fn =0.16287
β =135°
ARO (kN/m2)
T (sec)

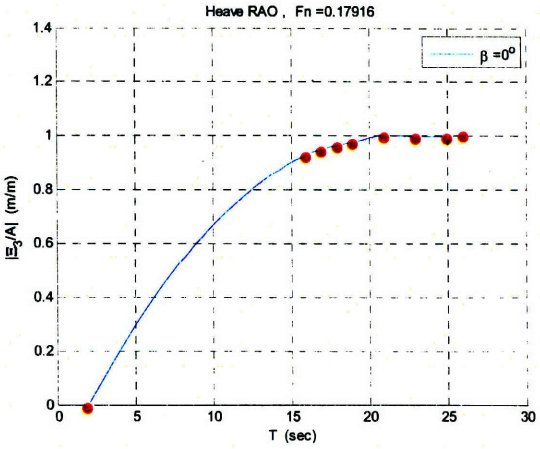

Roll RAO , Fn =0.16287
β =90°
|Ξ4/A| (deg/m)
T (sec)

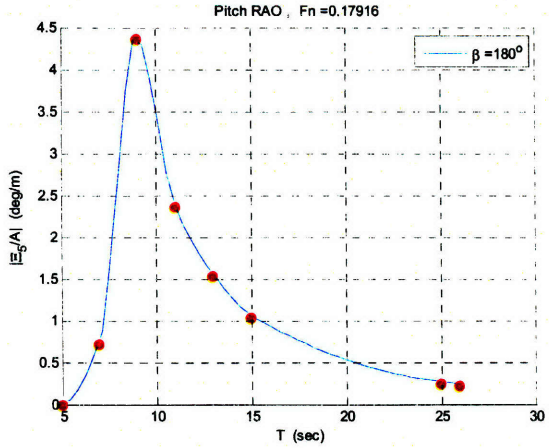

Added Resistance Operator, Fn =0.16287
β =90°
ARO (kN/m2)
T (sec)

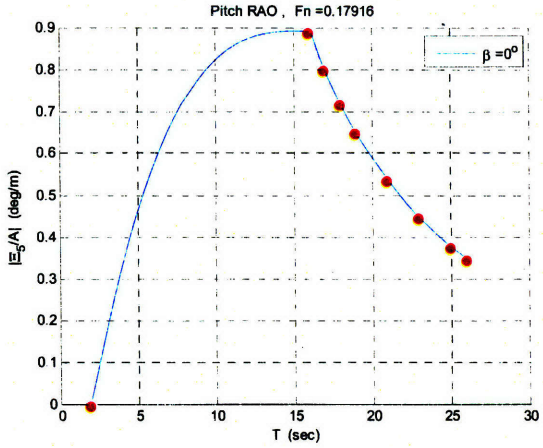

Roll RAO , Fn =0.16287
β =45°
|Ξ4/A| (deg/m)
T (sec)

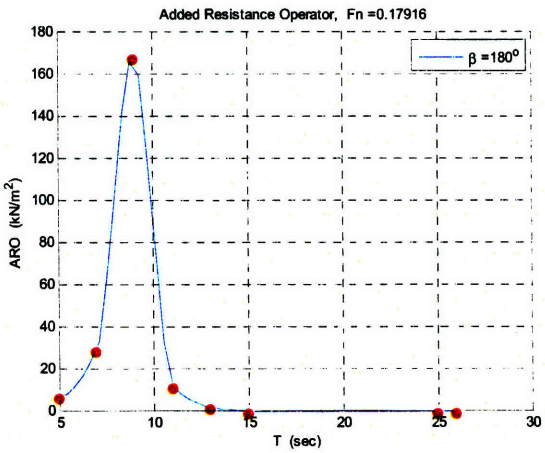

Added Resistance Operator, Fn =0.16287
β =45°
ARO (kN/m2)
T (sec)

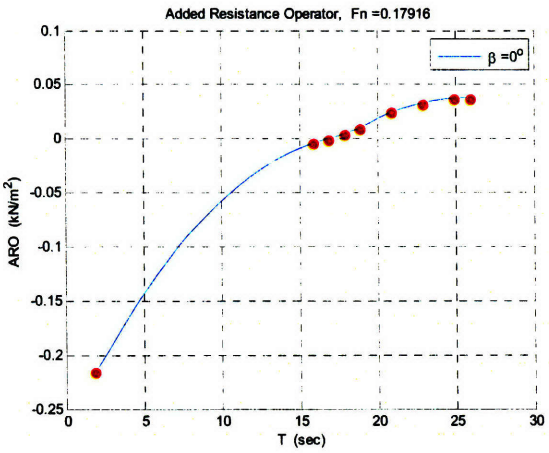

Heave RAO , Fn =0.17916
β =180°
|Ξ3/A| (m/m)
T (sec)

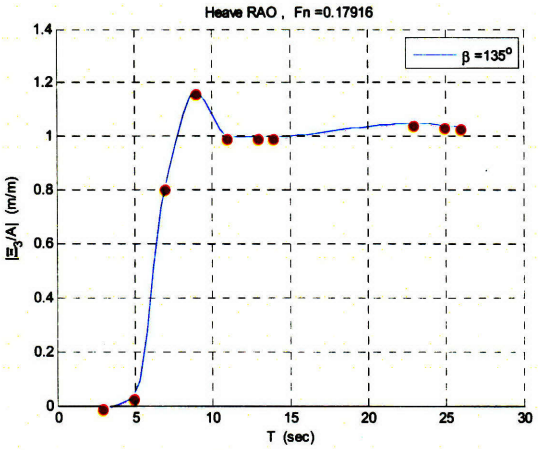

Heave RAO , Fn =0.17916
β =0°
|Ξ3/A| (m/m)
T (sec)

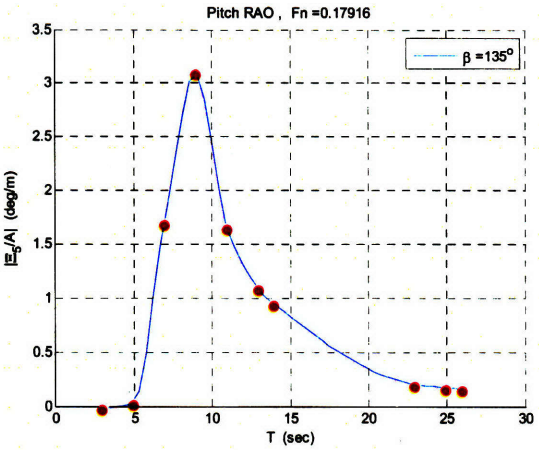

Pitch RAO , Fn =0.17916
β =180°
|Ξ5/A| (deg/m)
T (sec)

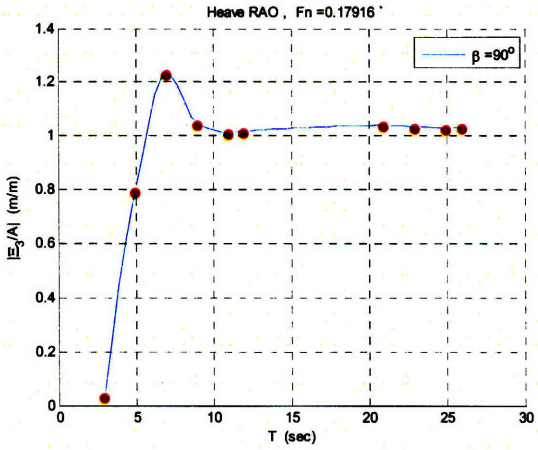

Pitch RAO , Fn =0.17916
β =0°
|Ξ5/A| (deg/m)
T (sec)

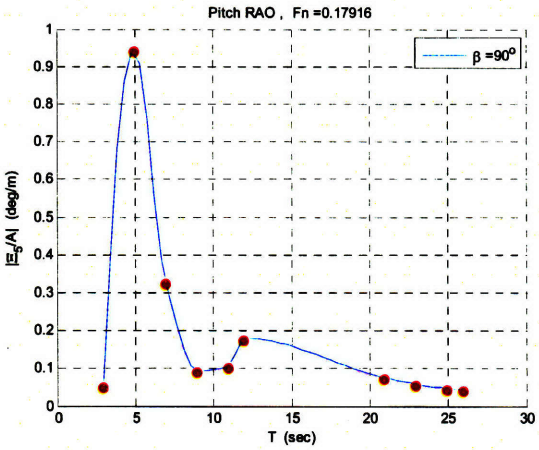

Added Resistance Operator, Fn =0.17916
β =180°
ARO (kN/m2)
T (sec)

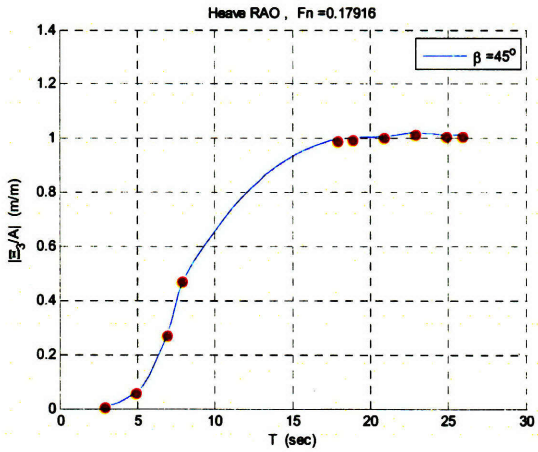

Added Resistance Operator, Fn =0.17916
β =0°
ARO (kN/m2)
T (sec)

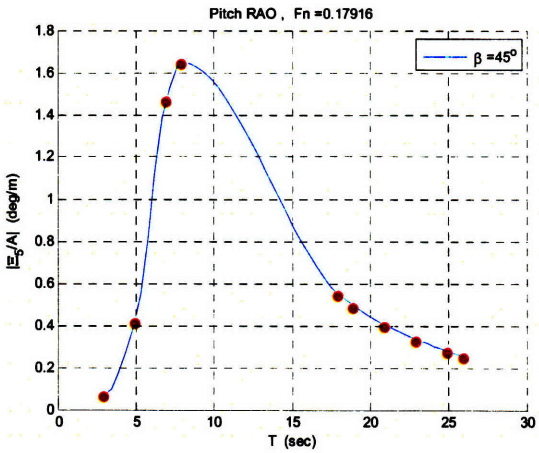

Heave RAO , Fn =0.17916
β =135°
|Ξ3/A| (m/m)
T (sec)

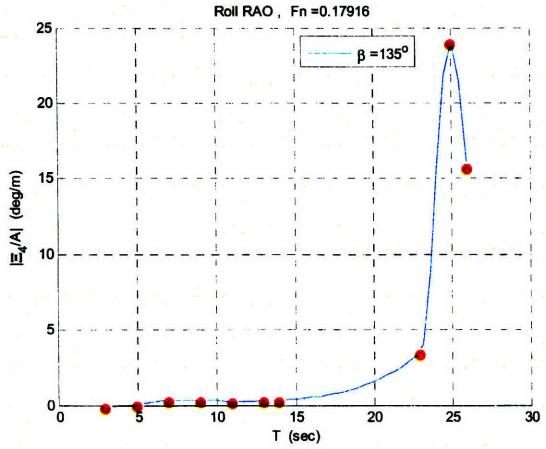

Pitch RAO , Fn =0.17916
β =135°
|Ξ5/A| (deg/m)
T (sec)

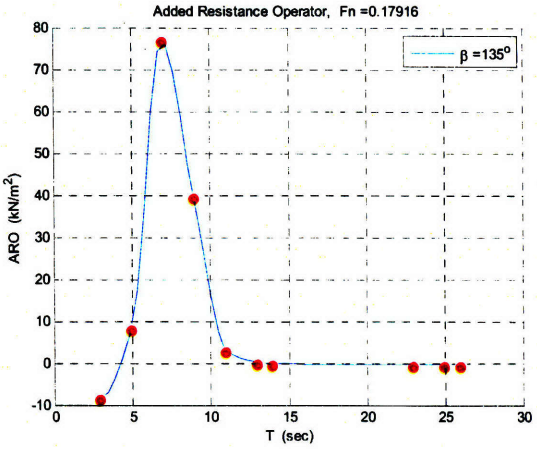

Heave RAO , Fn =0.17916
β =90°
|Ξ3/A| (m/m)
T (sec)

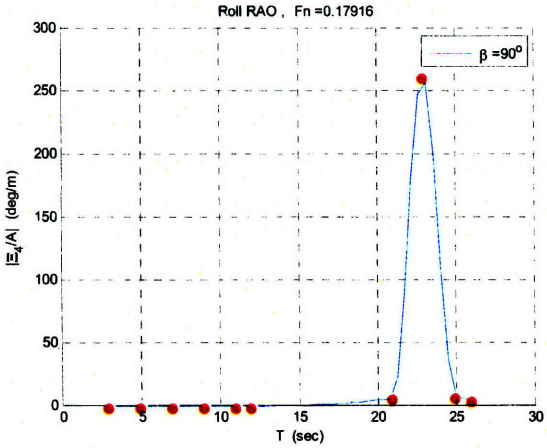

Pitch RAO , Fn =0.17916
β =90°
|Ξ5/A| (deg/m)
T (sec)

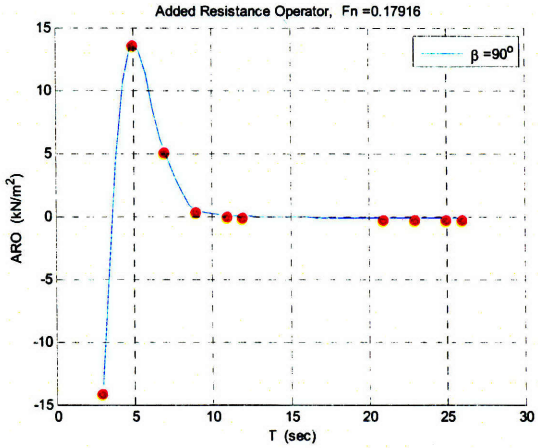

Heave RAO , Fn =0.17916
β =45°
|Ξ3/A| (m/m)
T (sec)

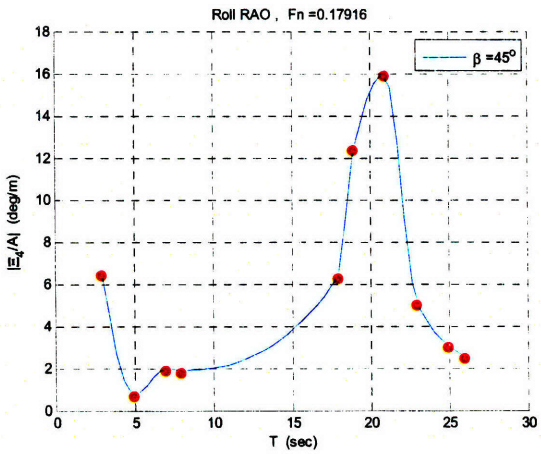

Pitch RAO , Fn =0.17916
β =45°
|Ξ5/A| (deg/m)
T (sec)

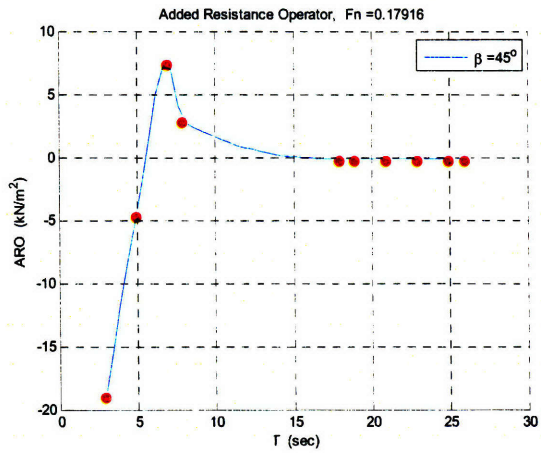

Roll RAO , Fn =0.17916
β =135°
|Ξ4/A| (deg/m)
T (sec)

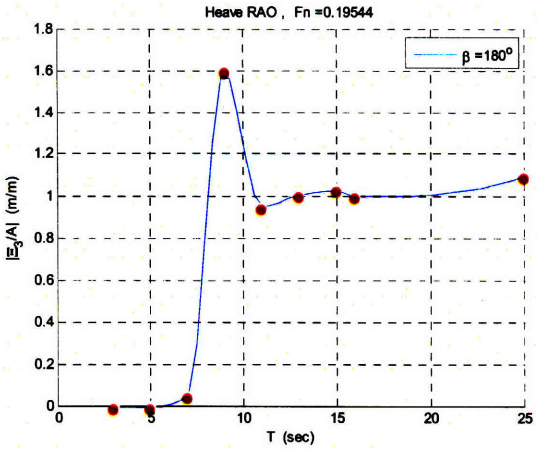

Added Resistance Operator, Fn =0.17916
β =135°
ARO (kN/m²)
T (sec)

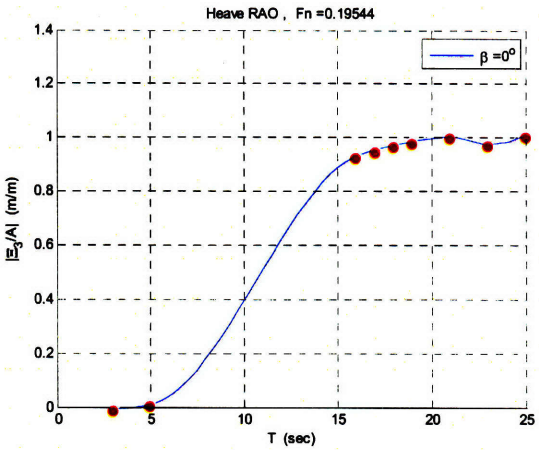

Roll RAO , Fn =0.17916
β =90°
|Ξ4/A| (deg/m)
T (sec)

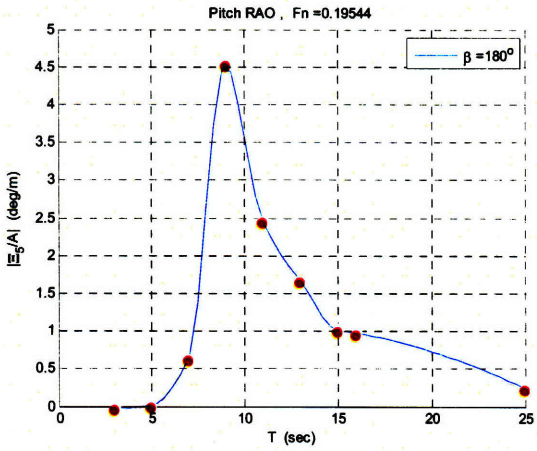

Added Resistance Operator, Fn =0.17916
β =90°
ARO (kN/m²)
T (sec)

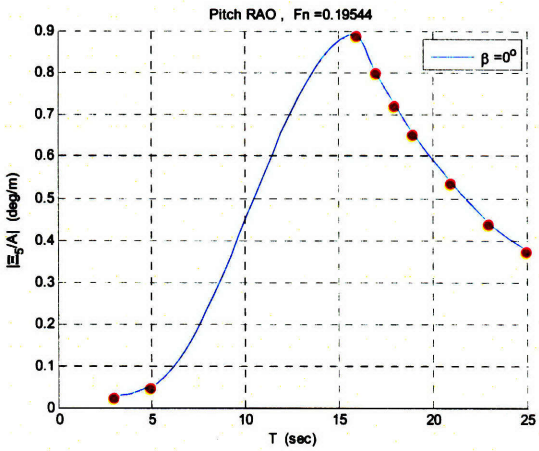

Roll RAO , Fn =0.17916
β =45°
|Ξ4/A| (deg/m)
T (sec)

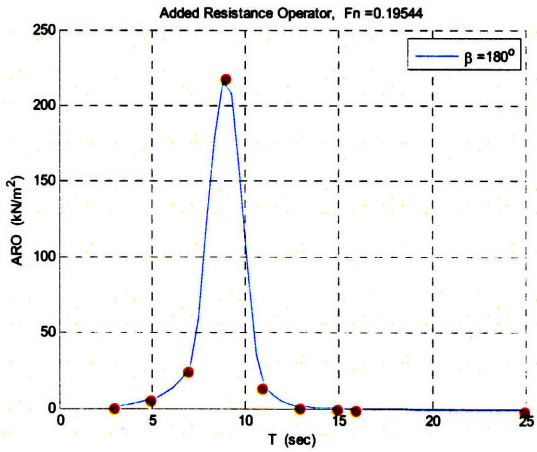

Added Resistance Operator, Fn =0.17916
β =45°
ARO (kN/m²)
T (sec)

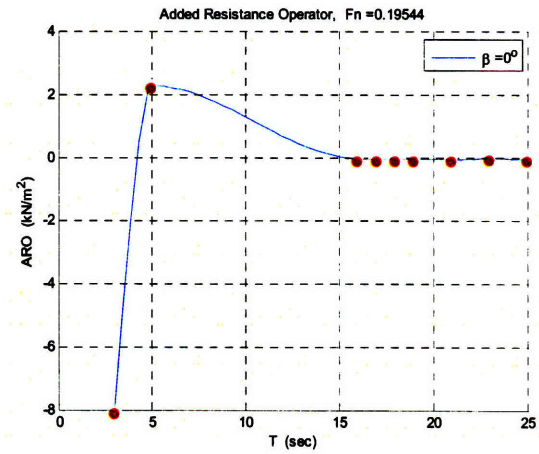

Heave RAO , Fn =0.19544
β =180°
|Ξ3/A| (m/m)
T (sec)

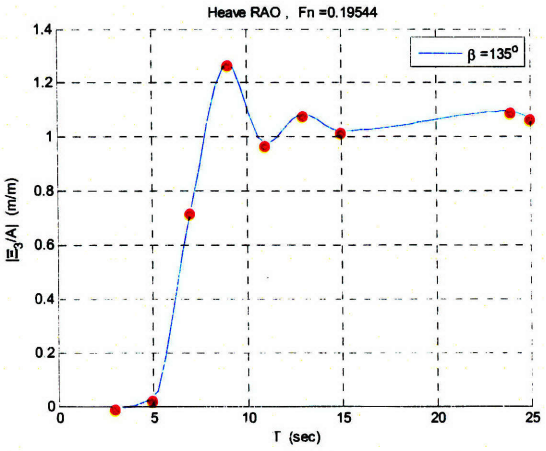

Heave RAO , Fn =0.19544
β =0°
|Ξ3/A| (m/m)
T (sec)

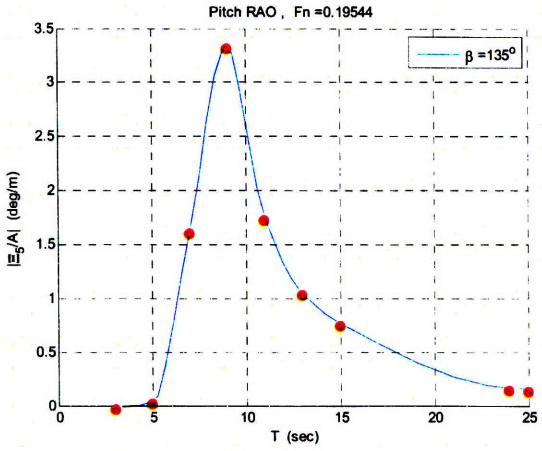

Pitch RAO , Fn =0.19544
β =180°
|Ξ5/A| (deg/m)
T (sec)

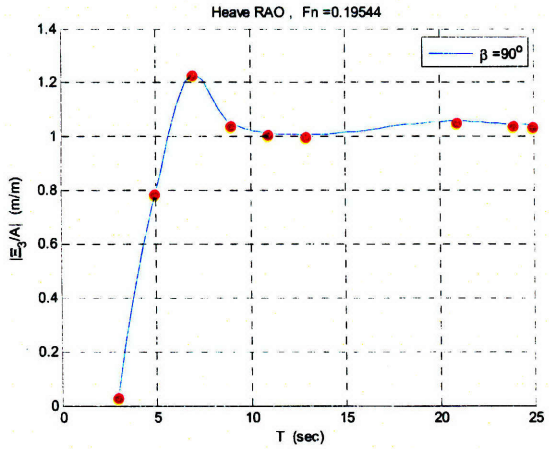

Pitch RAO , Fn =0.19544
β =0°
|Ξ5/A| (deg/m)
T (sec)

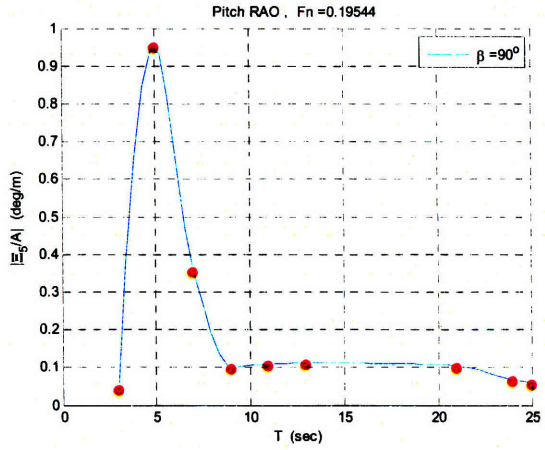

Added Resistance Operator, Fn =0.19544
β =180°
ARO (kN/m2)
T (sec)

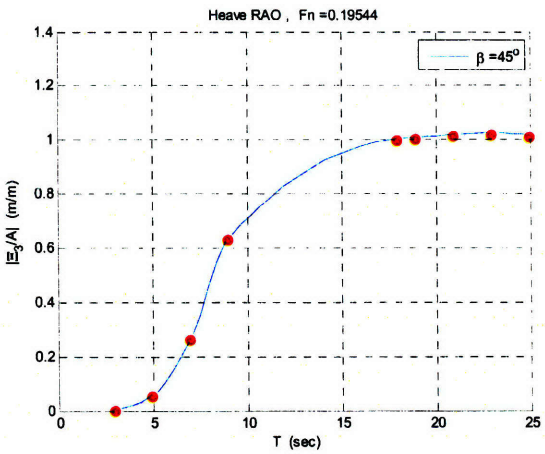

Added Resistance Operator, Fn =0.19544
β =0°
ARO (kN/m2)
T (sec)

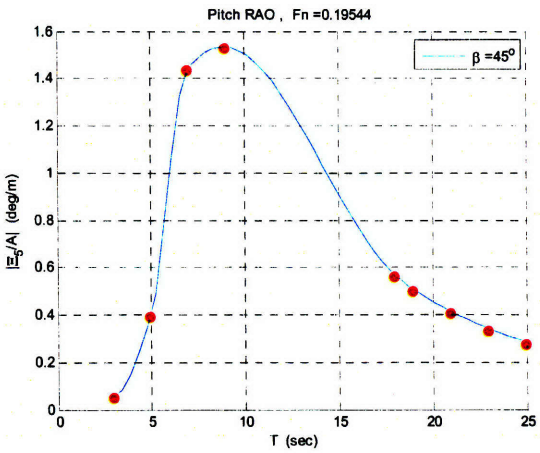

Heave RAO , Fn =0.19544
β =135°
|Ξ3/A| (m/m)
T (sec)

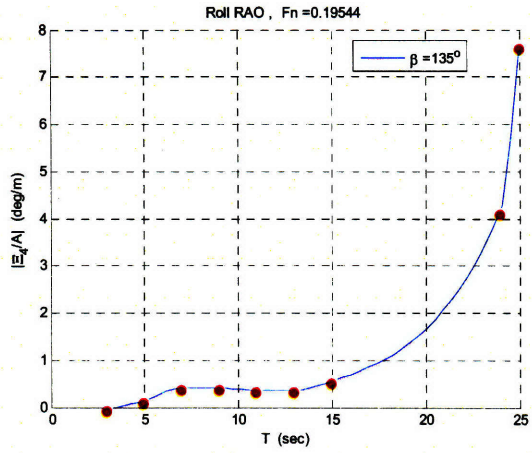

Pitch RAO , Fn =0.19544
β =135°
|Ξ5/A| (deg/m)
T (sec)

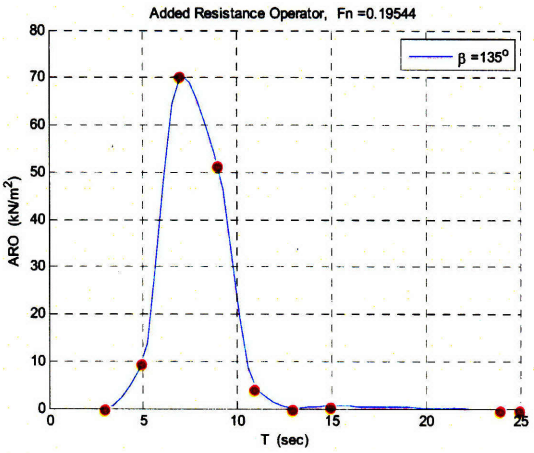

Heave RAO , Fn =0.19544
β =90°
|Ξ3/A| (m/m)
T (sec)

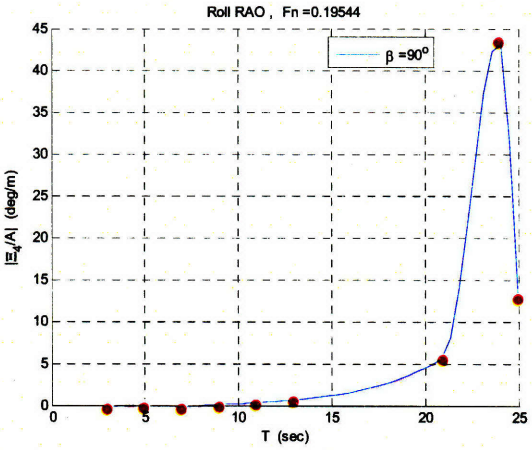

Pitch RAO , Fn =0.19544
β =90°
|Ξ5/A| (deg/m)
T (sec)

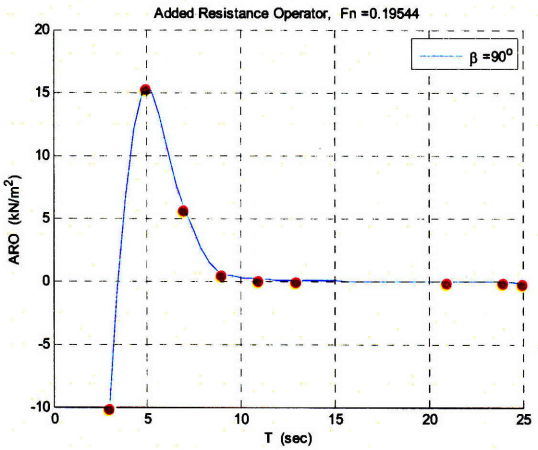

Heave RAO , Fn =0.19544
β =45°
|Ξ3/A| (m/m)
T (sec)

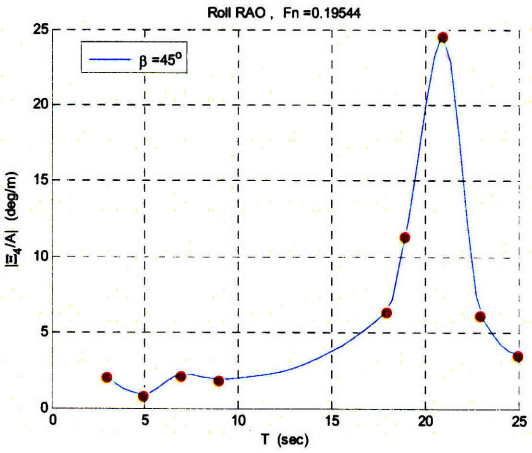

Pitch RAO , Fn =0.19544
β =45°
|Ξ5/A| (deg/m)
T (sec)

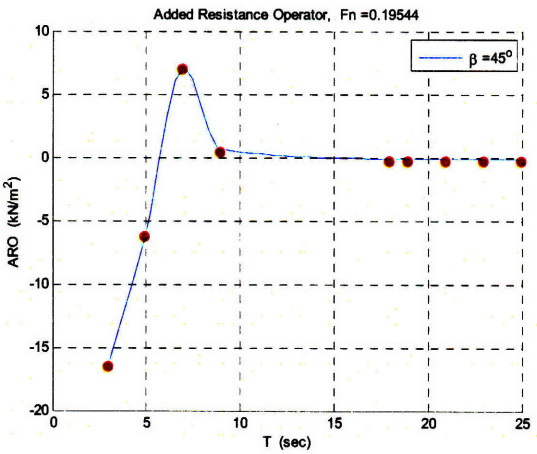

Roll RAO , Fn =0.19544
β =135°
|Ξ4/A| (deg/m)
T (sec)

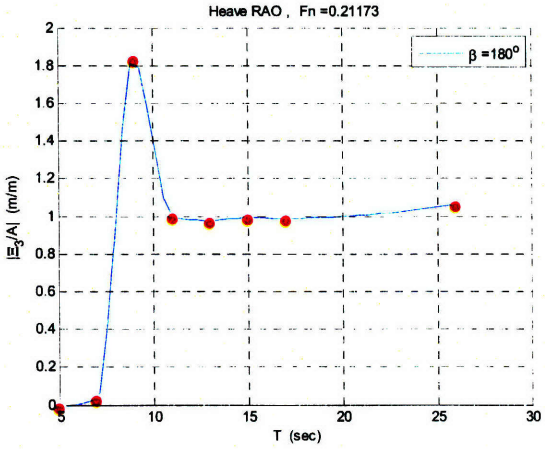

Added Resistance Operator, Fn =0.19544
β =135°
ARO (kN/m2)
T (sec)

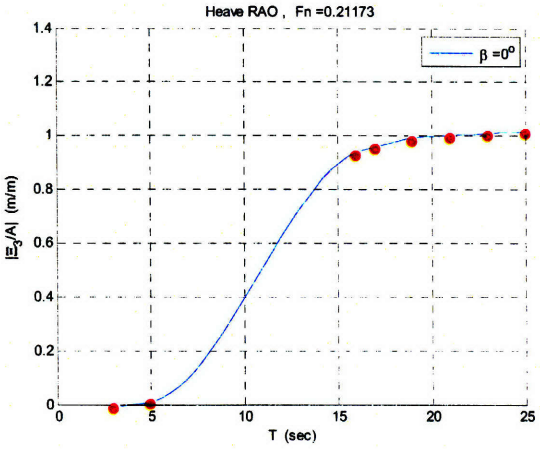

Roll RAO , Fn =0.19544
β =90°
|Ξ4/A| (deg/m)
T (sec)

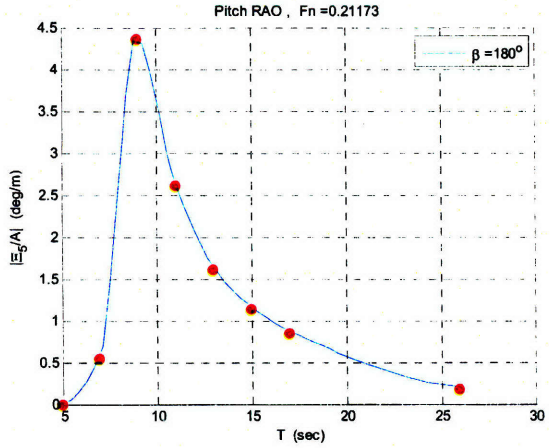

Added Resistance Operator, Fn =0.19544
β =90°
ARO (kN/m2)
T (sec)

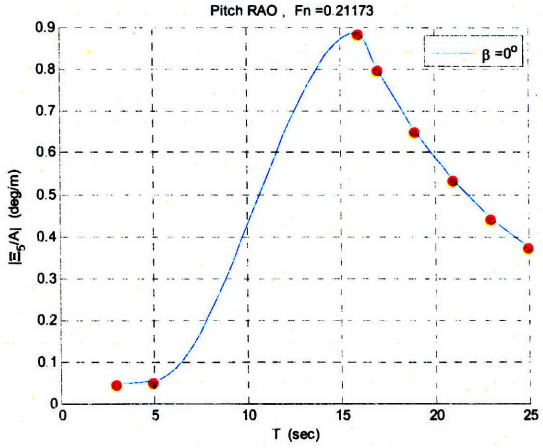

Roll RAO , Fn =0.19544
β =45°
|Ξ4/A| (deg/m)
T (sec)

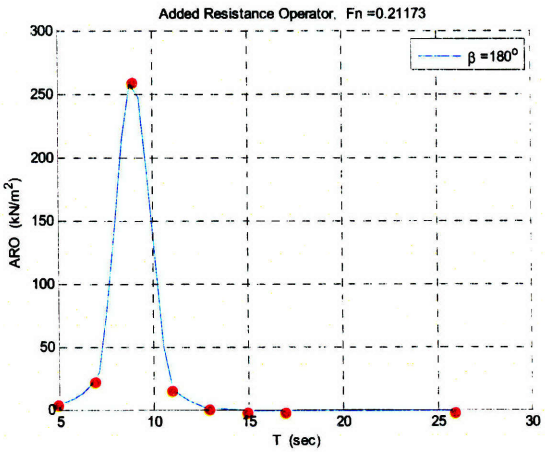

Added Resistance Operator, Fn =0.19544
β =45°
ARO (kN/m2)
T (sec)

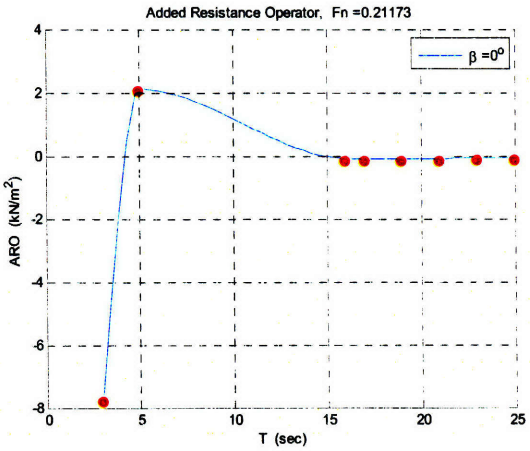

Heave RAO , Fn =0.21173
β =180°
|Ξ3/A| (m/m)
T (sec)

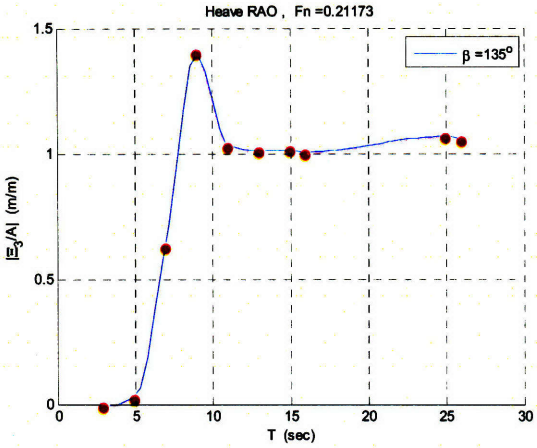

Heave RAO , Fn =0.21173
β =0°
|Ξ3/A| (m/m)
T (sec)

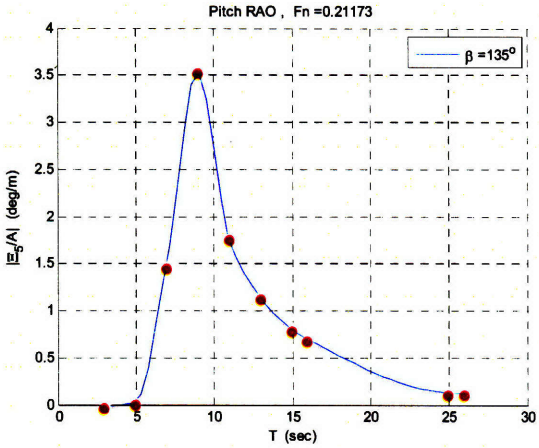

Pitch RAO , Fn =0.21173
β =180°
|Ξ5/A| (deg/m)
T (sec)

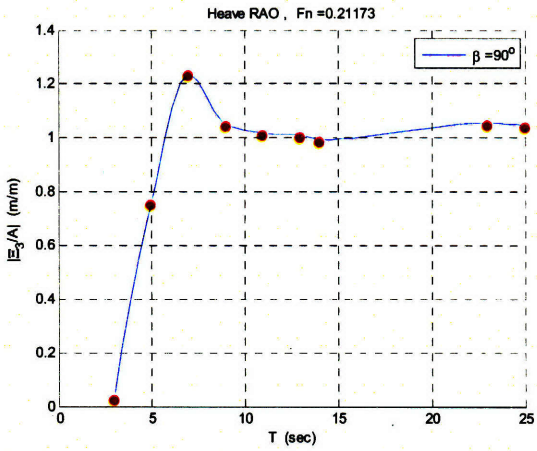

Pitch RAO , Fn =0.21173
β =0°
|Ξ5/A| (deg/m)
T (sec)

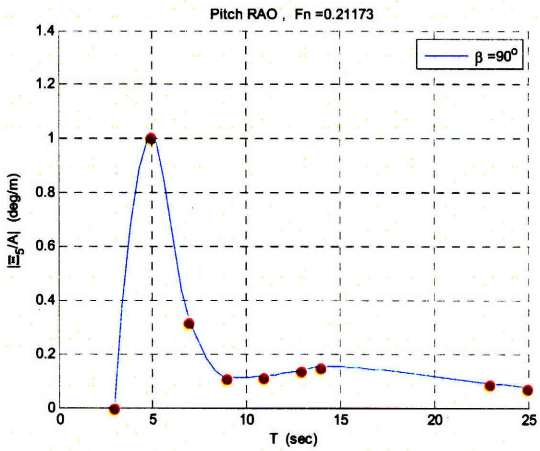

Added Resistance Operator, Fn =0.21173
β =180°
ARO (kN/m2)
T (sec)

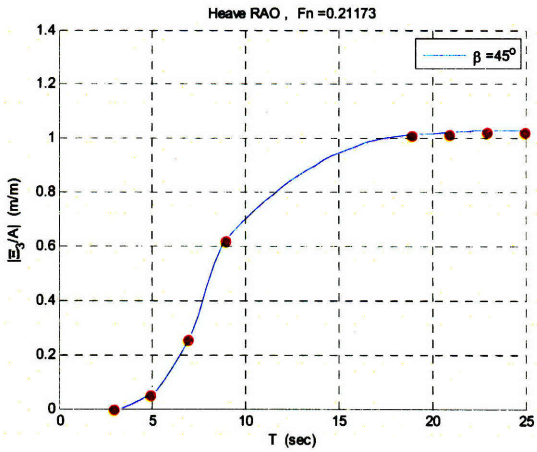

Added Resistance Operator, Fn =0.21173
β =0°
ARO (kN/m2)
T (sec)

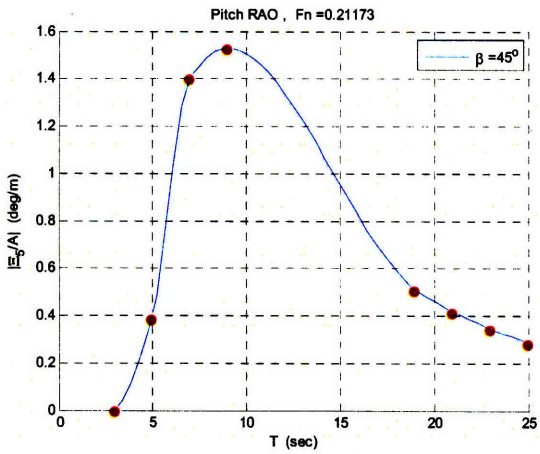

Heave RAO , Fn =0.21173
β =135°
|Ξ3/A| (m/m)
T (sec)

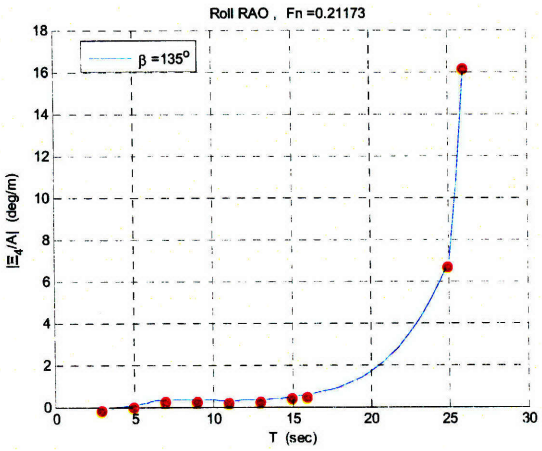

Pitch RAO , Fn =0.21173
β =135°
|Ξ5/A| (deg/m)
T (sec)

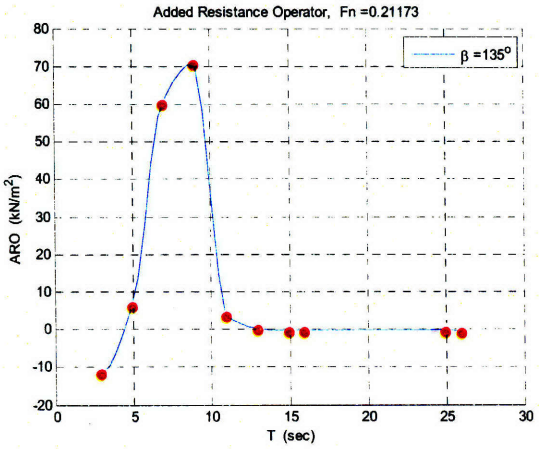

Heave RAO , Fn =0.21173
β =90°
|Ξ3/A| (m/m)
T (sec)

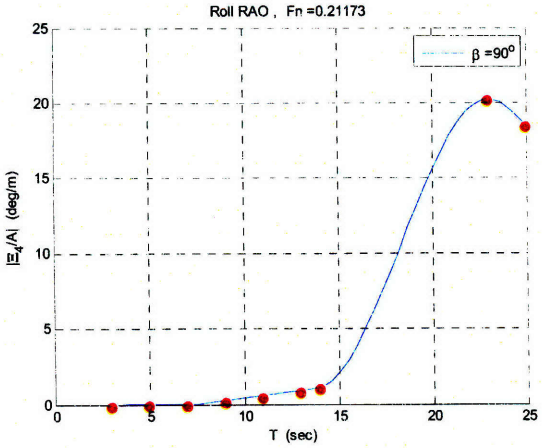

Pitch RAO , Fn =0.21173
β =90°
|Ξ5/A| (deg/m)
T (sec)

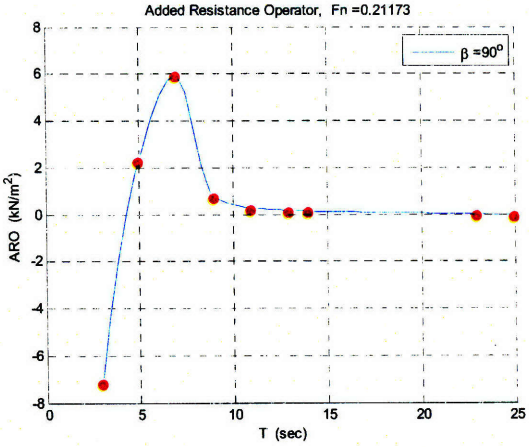

Heave RAO , Fn =0.21173
β =45°
|Ξ3/A| (m/m)
T (sec)

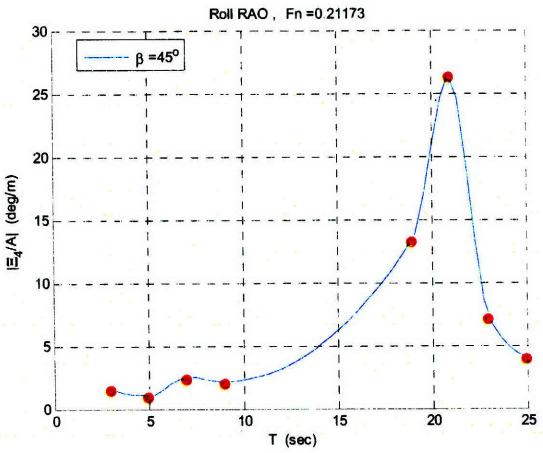

Pitch RAO , Fn =0.21173
β =45°
|Ξ5/A| (deg/m)
T (sec)

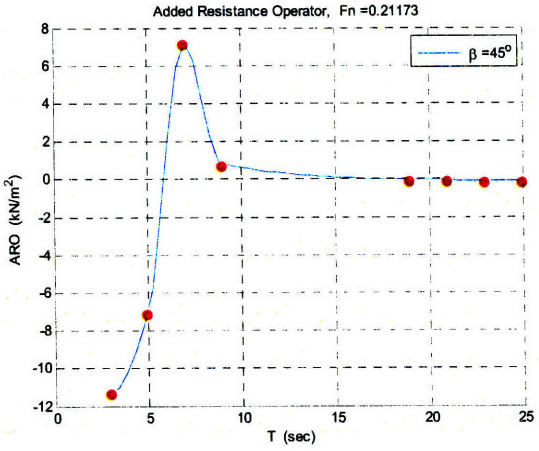

Roll RAO , Fn =0.21173
β =135°
|Ξ4/A| (deg/m)
T (sec)

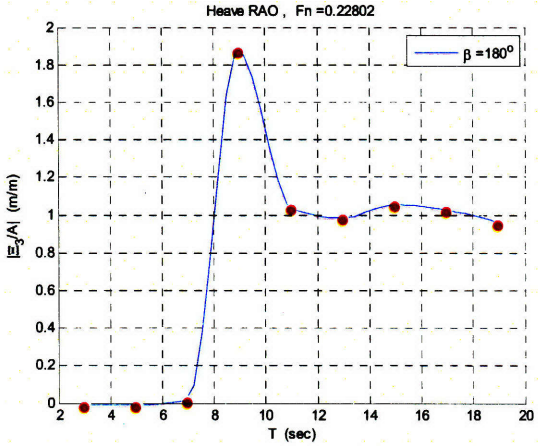

Added Resistance Operator, Fn =0.21173
β =135°
ARO (kN/m²)
T (sec)

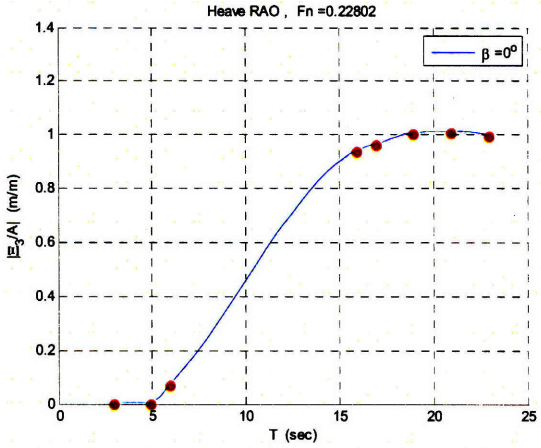

Roll RAO , Fn =0.21173
β =90°
|Ξ4/A| (deg/m)
T (sec)

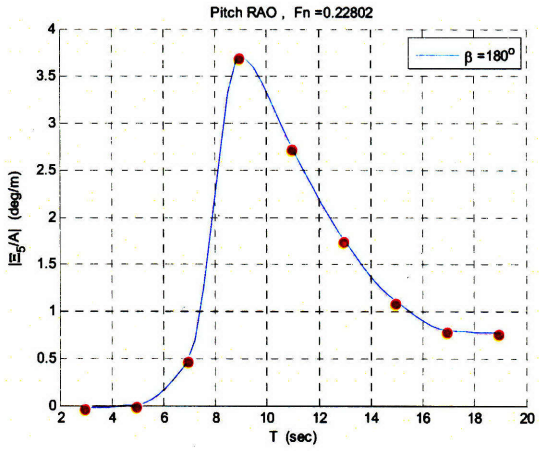

Added Resistance Operator, Fn =0.21173
β =90°
ARO (kN/m²)
T (sec)

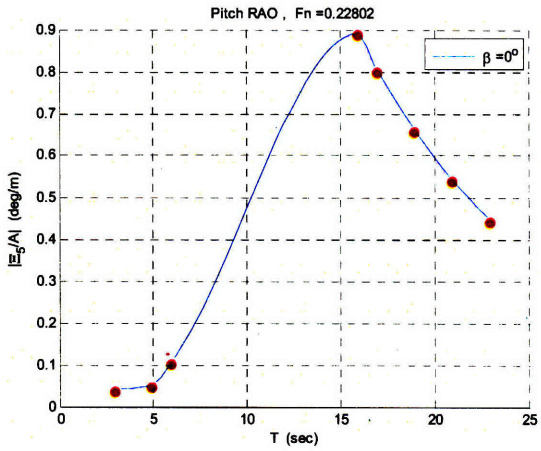

Roll RAO , Fn =0.21173
β =45°
|Ξ4/A| (deg/m)
T (sec)

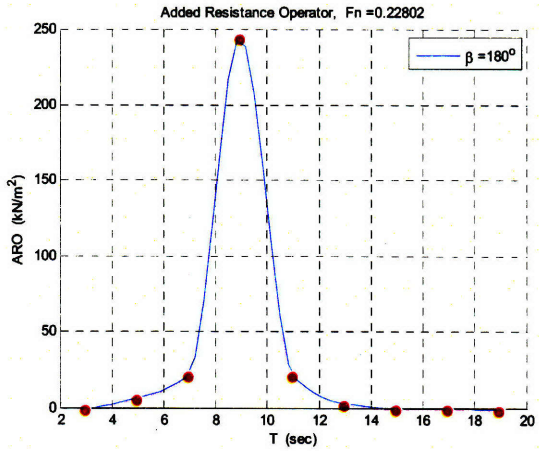

Added Resistance Operator, Fn =0.21173
β =45°
ARO (kN/m²)
T (sec)

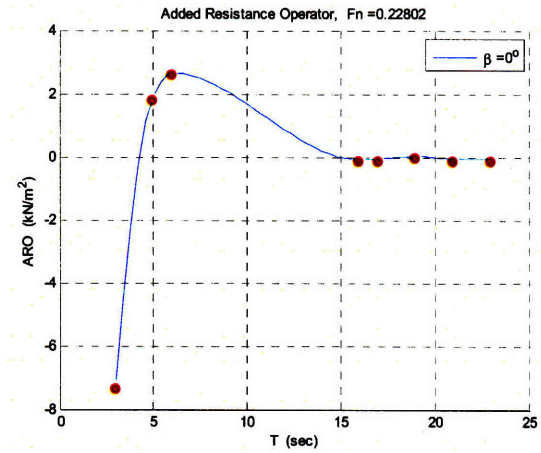

Heave RAO , Fn =0.22802
β =180°
|Ξ3/A| (m/m)
T (sec)

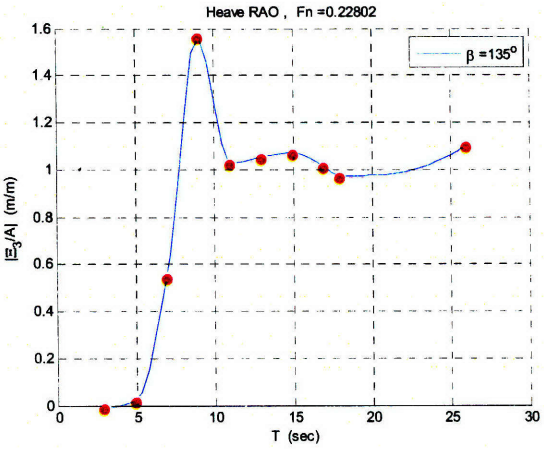

Heave RAO , Fn =0.22802
β =0°
|Ξ3/A| (m/m)
T (sec)

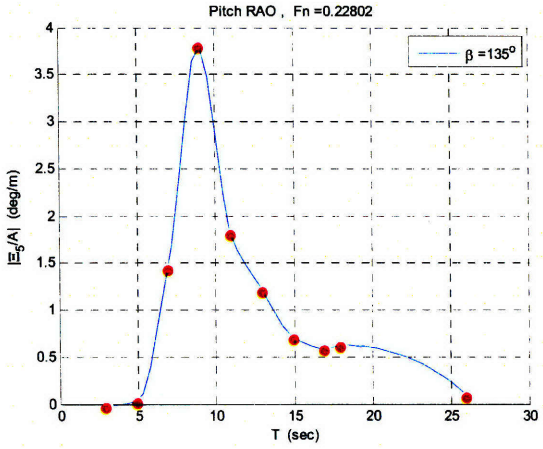

Pitch RAO , Fn =0.22802
β =180°
|Ξ5/A| (deg/m)
T (sec)

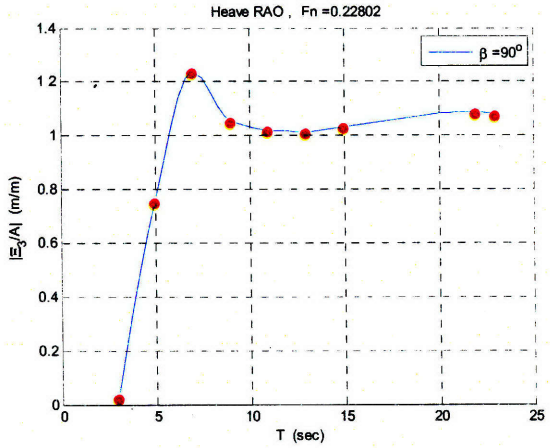

Pitch RAO , Fn =0.22802
β =0°
|Ξ5/A| (deg/m)
T (sec)

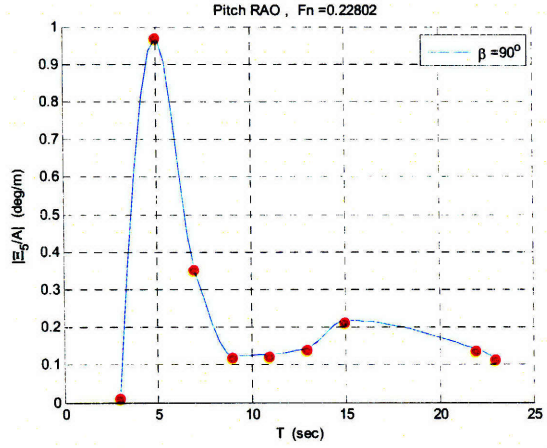

Added Resistance Operator, Fn =0.22802
β =180°
ARO (kN/m²)
T (sec)

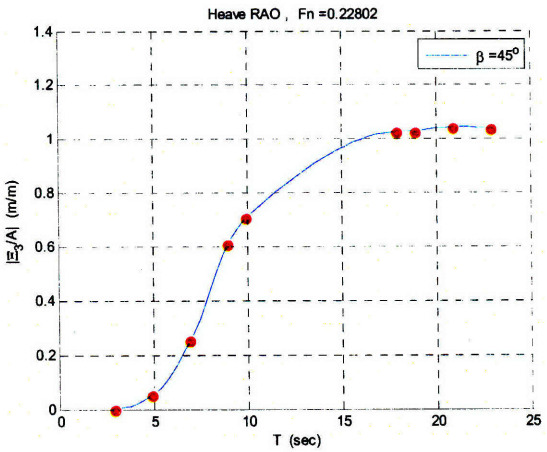

Added Resistance Operator, Fn =0.22802
β =0°
ARO (kN/m²)
T (sec)

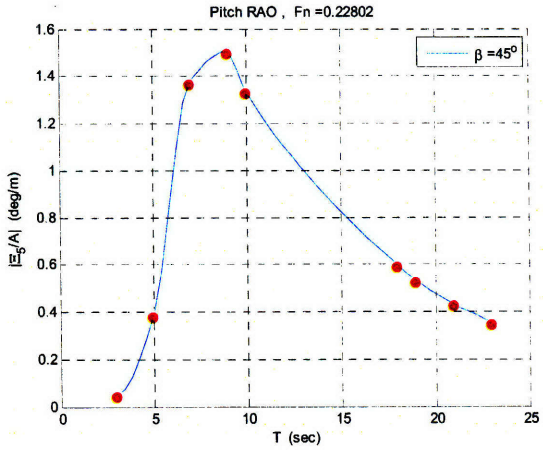

Heave RAO , Fn =0.22802
β =135°
|Ξ3/A| (m/m)
T (sec)

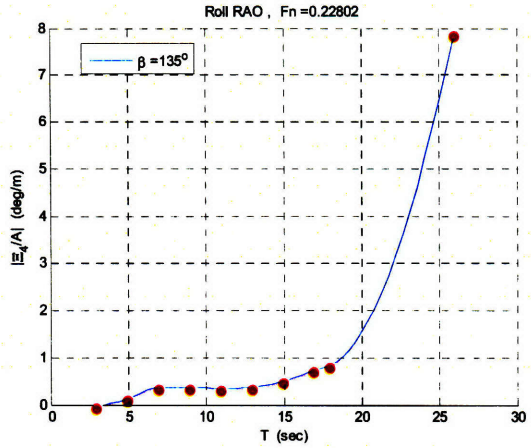

Pitch RAO , Fn =0.22802
β =135°
|Ξ5/A| (deg/m)
T (sec)

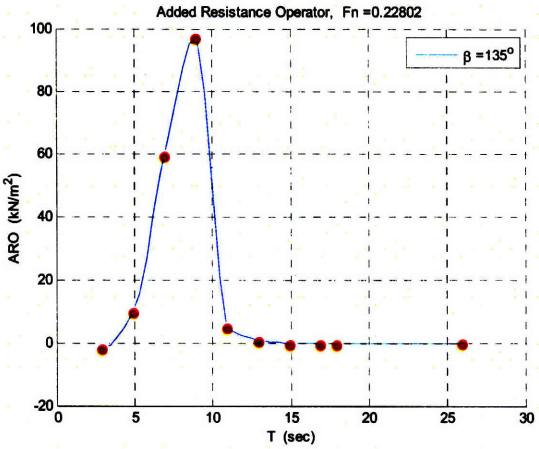

Heave RAO , Fn =0.22802
β =90°
|Ξ3/A| (m/m)
T (sec)

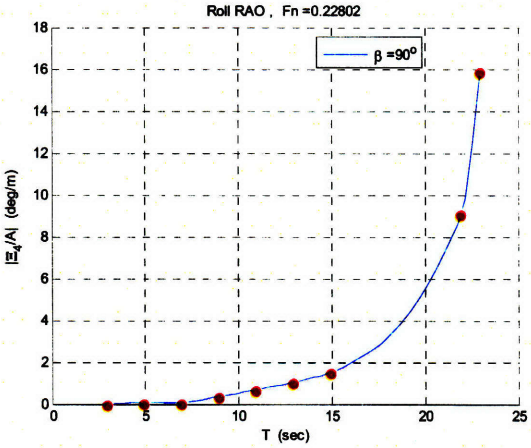

Pitch RAO , Fn =0.22802
β =90°
|Ξ5/A| (deg/m)
T (sec)

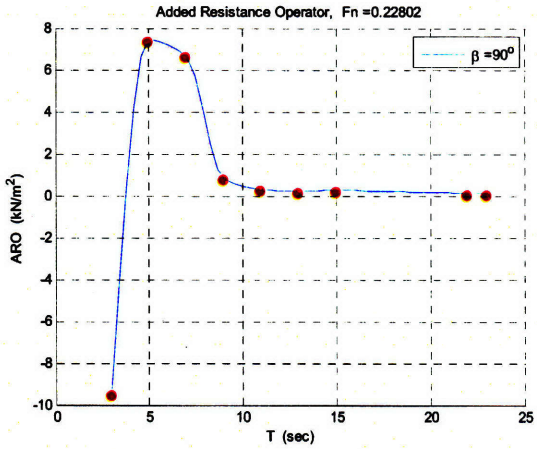

Heave RAO , Fn =0.22802
β =45°
|Ξ3/A| (m/m)
T (sec)

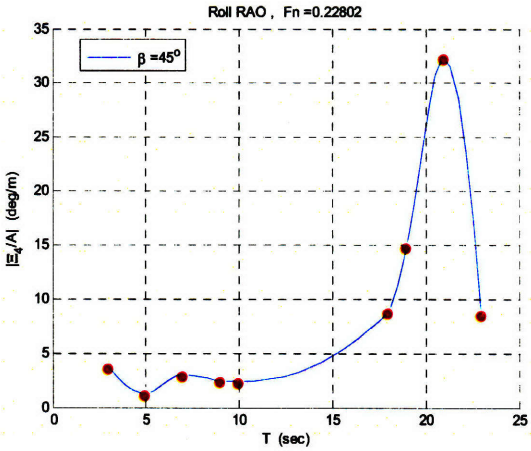

Pitch RAO , Fn =0.22802
β =45°
|Ξ5/A| (deg/m)
T (sec)

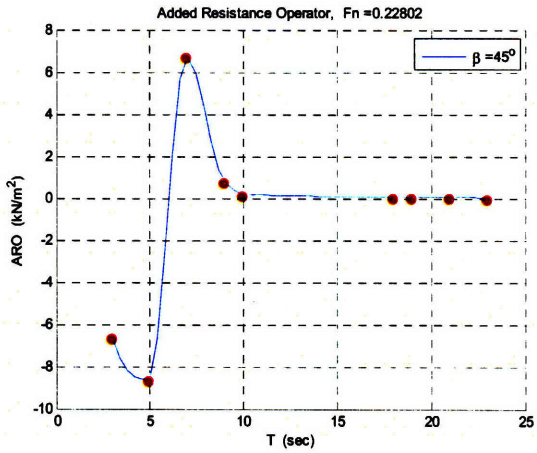

Roll RAO , Fn =0.22802
β =135°
|Ξ4/A| (deg/m)
T (sec)

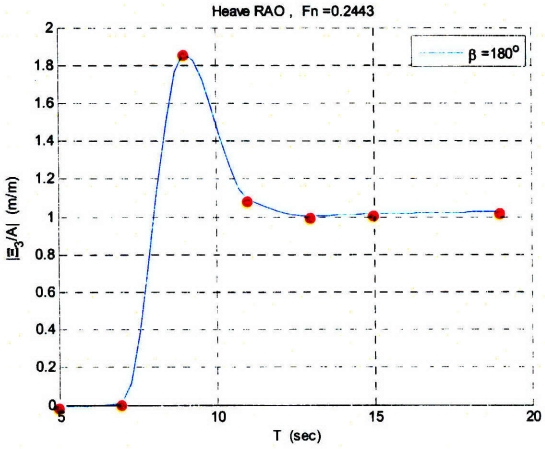

Added Resistance Operator, Fn =0.22802
β =135°
ARO (kN/m2)
T (sec)

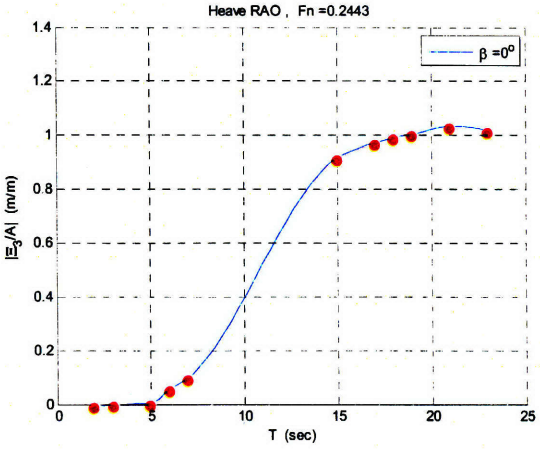

Roll RAO , Fn =0.22802
β =90°
|Ξ4/A| (deg/m)
T (sec)

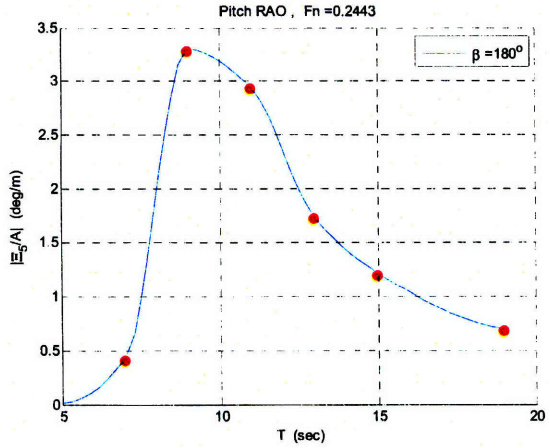

Added Resistance Operator, Fn =0.22802
β =90°
ARO (kN/m2)
T (sec)

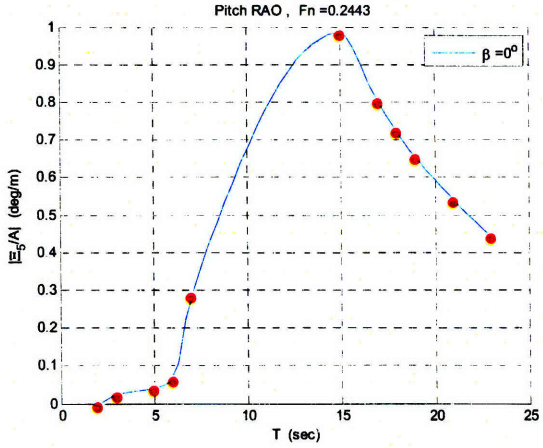

Roll RAO , Fn =0.22802
β =45°
|Ξ4/A| (deg/m)
T (sec)

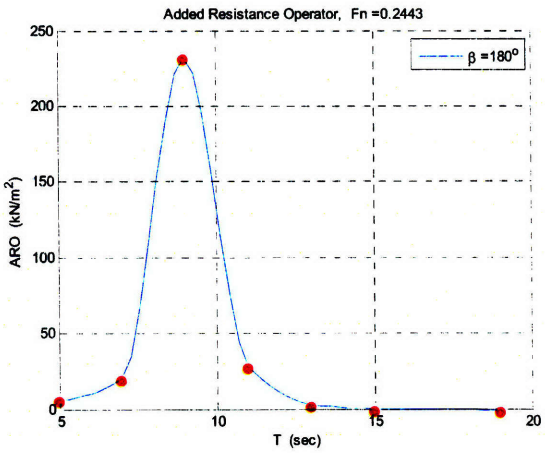

Added Resistance Operator, Fn =0.22802
β =45°
ARO (kN/m2)
T (sec)

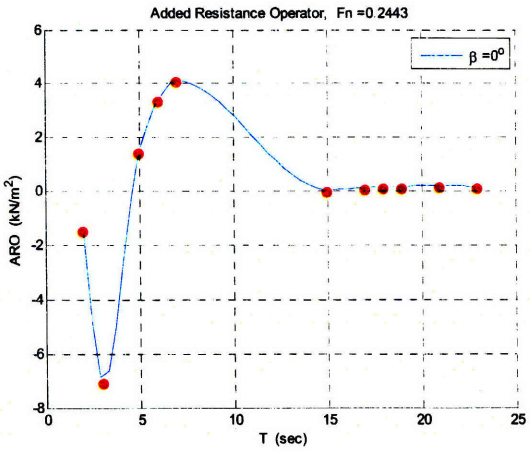

Heave RAO , Fn =0.2443
β =180°
|Ξ3/A| (m/m)
T (sec)

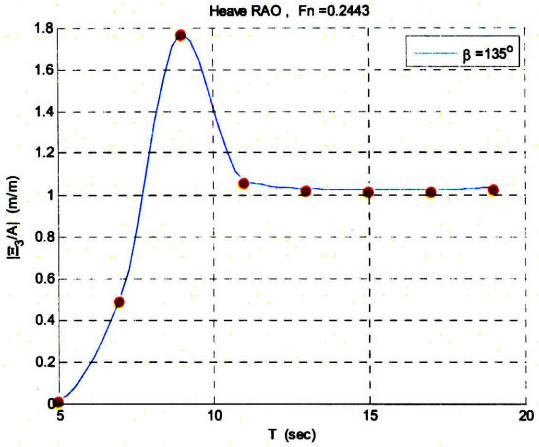

Heave RAO , Fn =0.2443
β =0°
|Ξ3/A| (m/m)
T (sec)

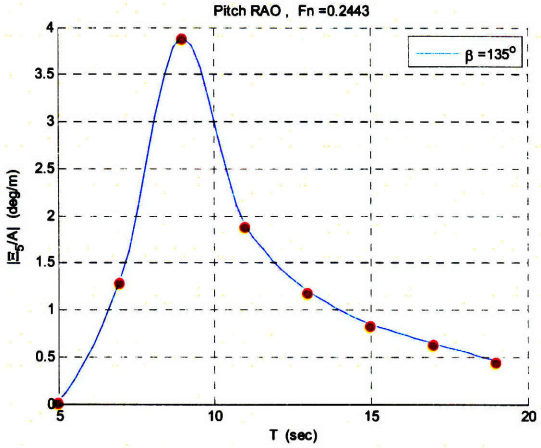

Pitch RAO , Fn =0.2443
β =180°
|Ξ5/A| (deg/m)
T (sec)

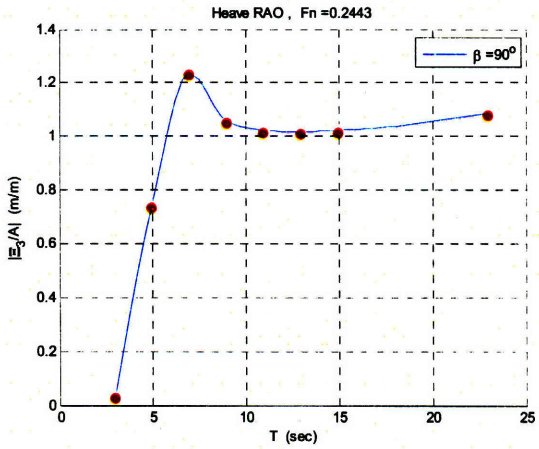

Pitch RAO , Fn =0.2443
β =0°
|Ξ5/A| (deg/m)
T (sec)

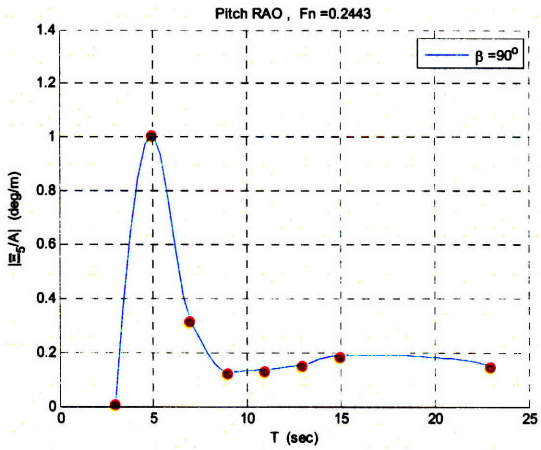

Added Resistance Operator, Fn =0.2443
β =180°
ARO (kN/m2)
T (sec)

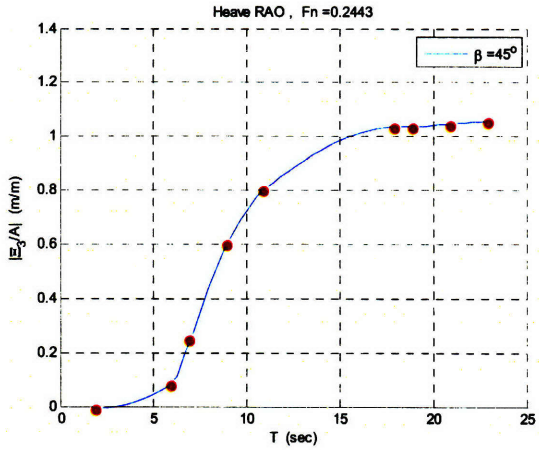

Added Resistance Operator, Fn =0.2443
β =0°
ARO (kN/m2)
T (sec)

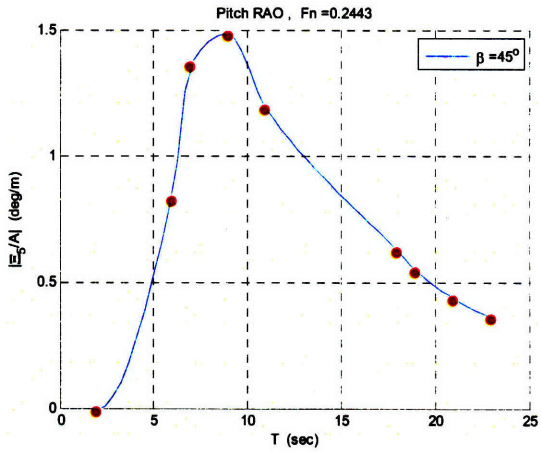

Heave RAO , Fn =0.2443
β =135°
|Ξ3/A| (m/m)
T (sec)

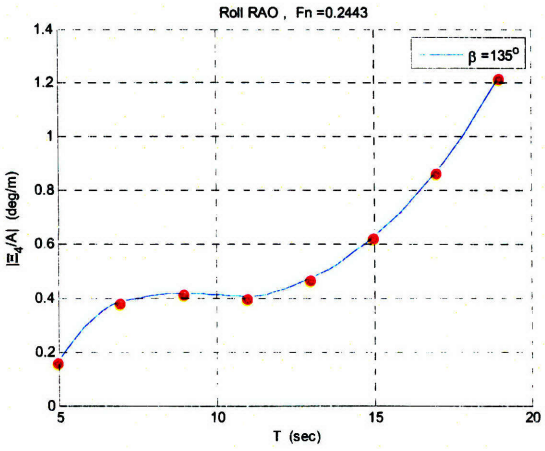

Pitch RAO , Fn =0.2443
β =135°
|Ξ5/A| (deg/m)
T (sec)

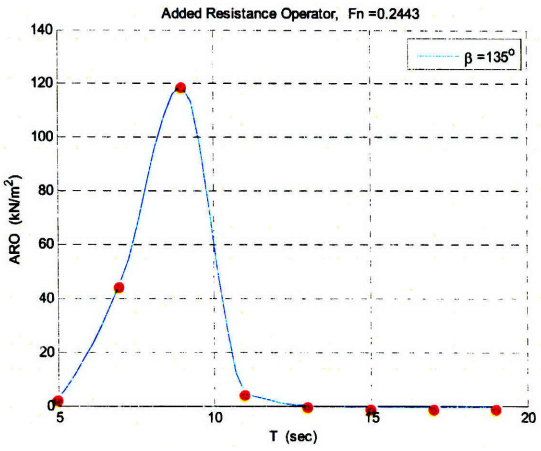

Heave RAO , Fn =0.2443
β =90°
|Ξ3/A| (m/m)
T (sec)

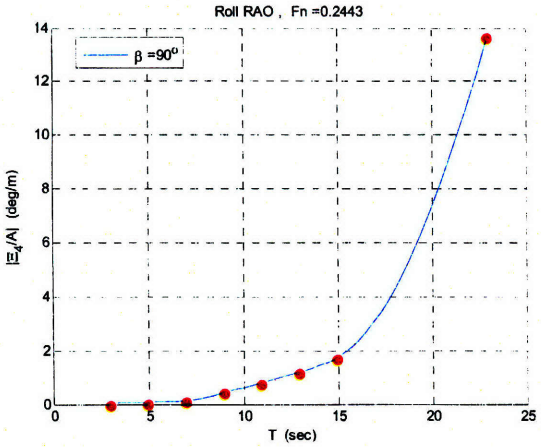

Pitch RAO , Fn =0.2443
β =90°
|Ξ5/A| (deg/m)
T (sec)

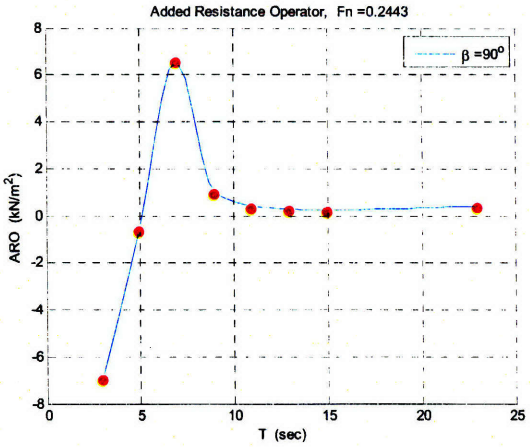

Heave RAO , Fn =0.2443
β =45°
|Ξ3/A| (m/m)
T (sec)

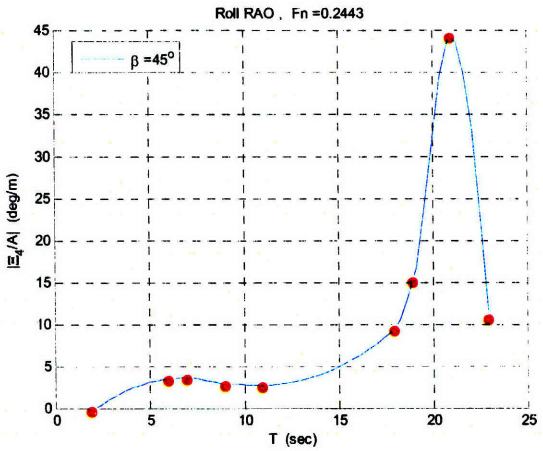

Pitch RAO , Fn =0.2443
β =45°
|Ξ5/A| (deg/m)
T (sec)

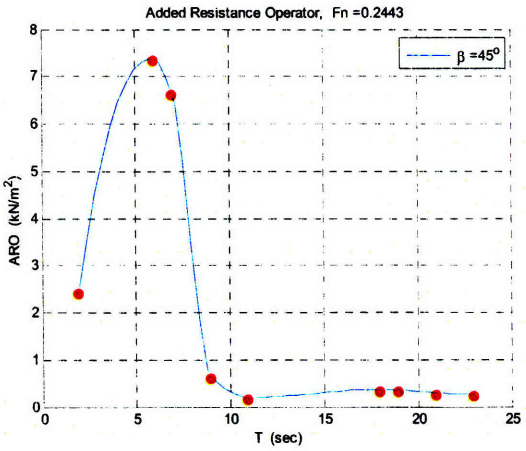

Roll RAO , Fn =0.2443
β =135°
|Ξ4/A| (deg/m)
T (sec)

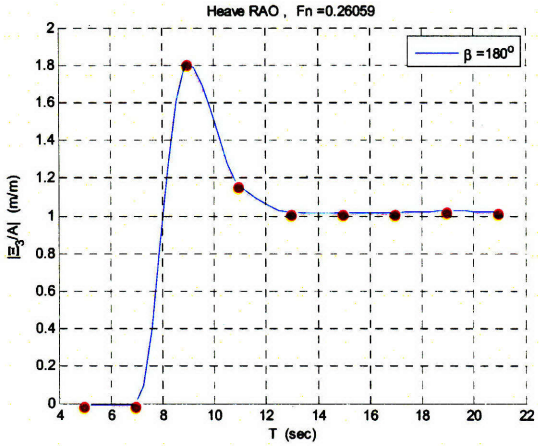

Added Resistance Operator, Fn =0.2443
β =135°
ARO (kN/m²)
T (sec)

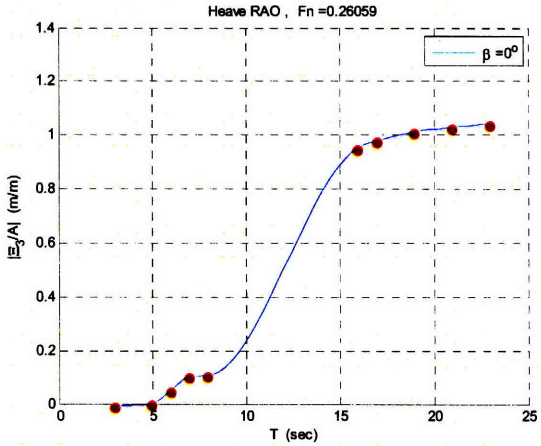

Roll RAO , Fn =0.2443
β =90°
|Ξ4/A| (deg/m)
T (sec)

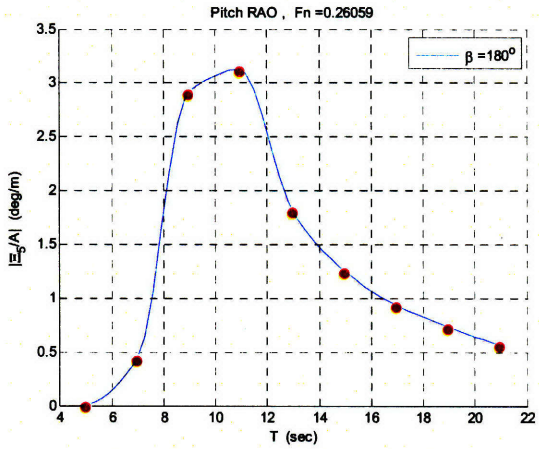

Added Resistance Operator, Fn =0.2443
β =90°
ARO (kN/m²)
T (sec)

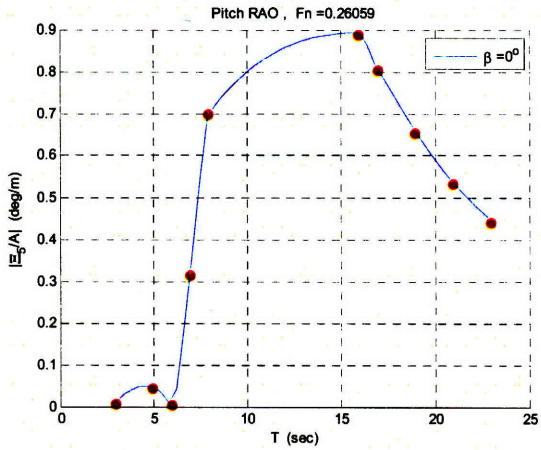

Roll RAO , Fn =0.2443
β =45°
|Ξ4/A| (deg/m)
T (sec)

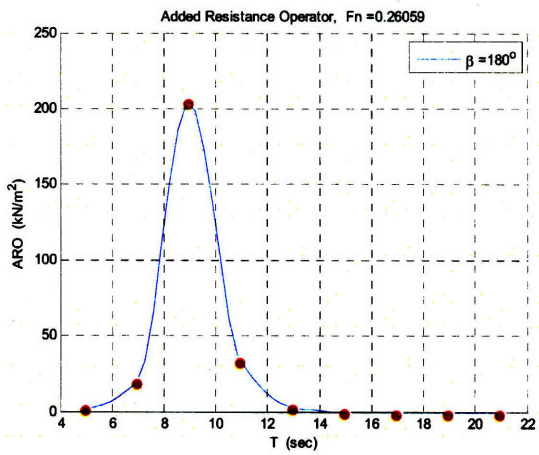

Added Resistance Operator, Fn =0.2443
β =45°
ARO (kN/m²)
T (sec)

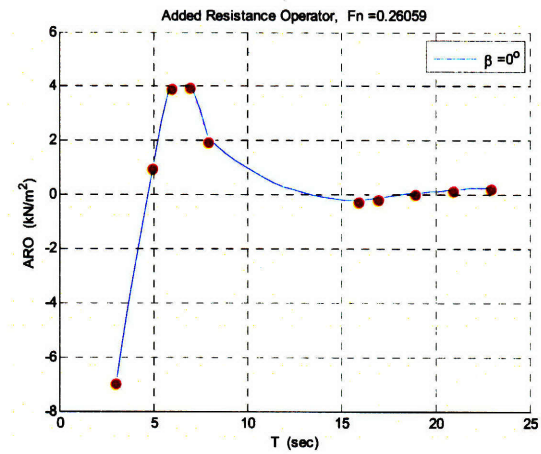

Heave RAO , Fn =0.26059
β =180°
|Ξ3/A| (m/m)
T (sec)

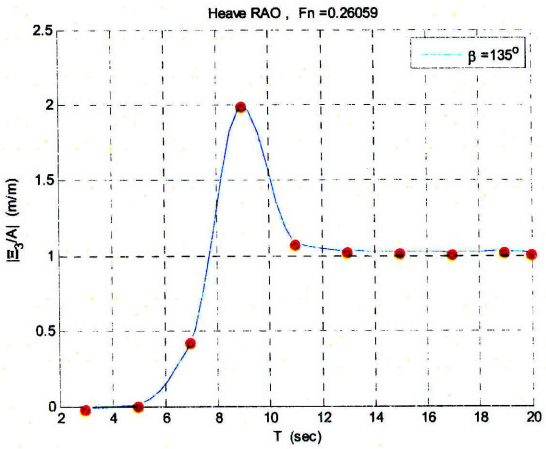

Heave RAO , Fn =0.26059
β =0°
|Ξ3/A| (m/m)
T (sec)

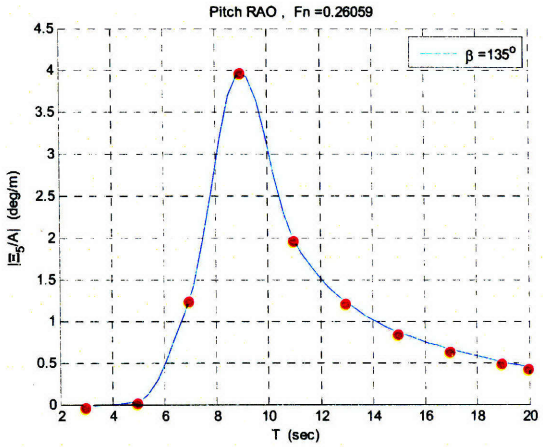

Pitch RAO , Fn =0.26059
β =180°
|Ξ5/A| (deg/m)
T (sec)

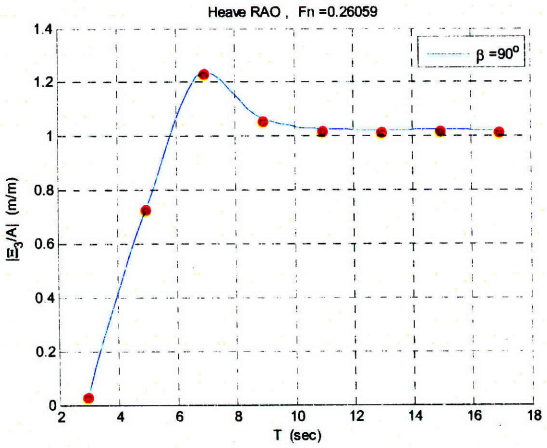

Pitch RAO , Fn =0.26059
β =0°
|Ξ5/A| (deg/m)
T (sec)

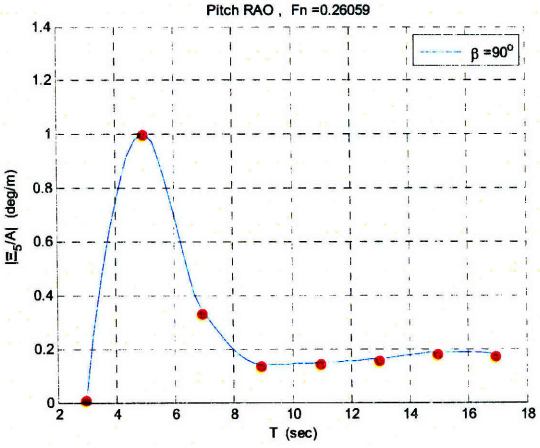

Added Resistance Operator, Fn =0.26059
β =180°
ARO (kN/m2)
T (sec)

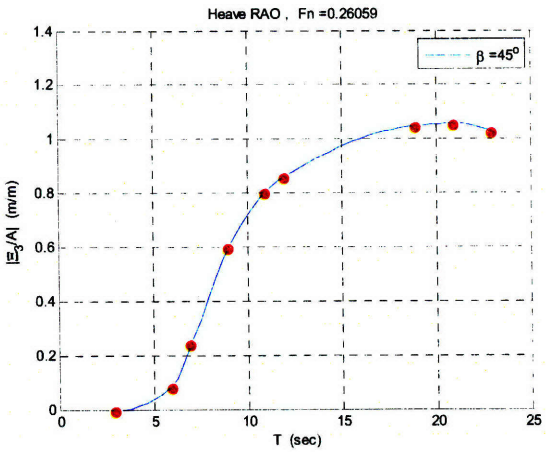

Added Resistance Operator, Fn =0.26059
β =0°
ARO (kN/m2)
T (sec)

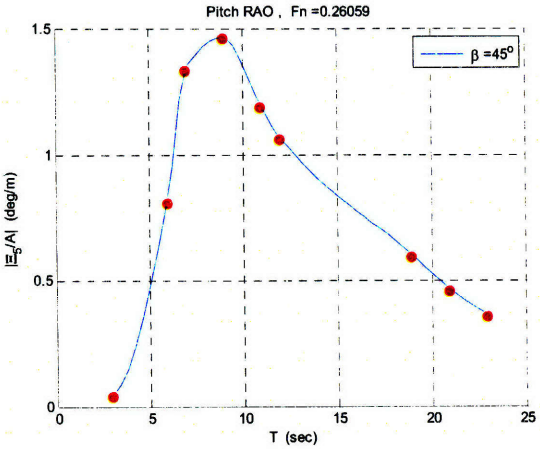

Heave RAO , Fn =0.26059
β =135°
|Ξ3/A| (m/m)
T (sec)

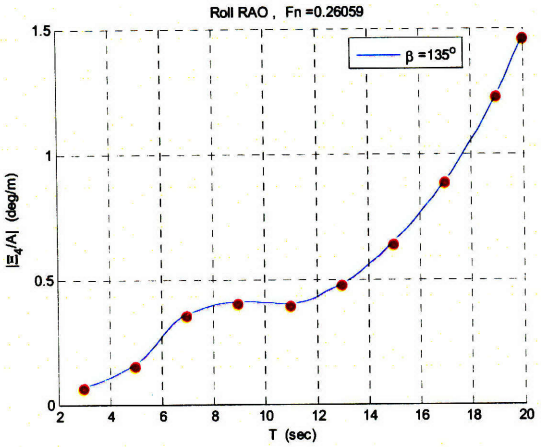

Pitch RAO , Fn =0.26059
β =135°
|Ξ5/A| (deg/m)
T (sec)

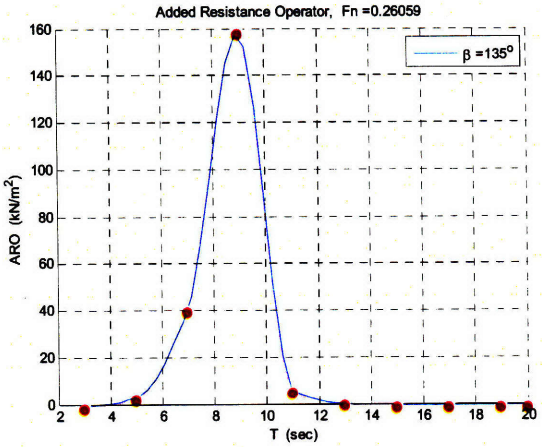

Heave RAO , Fn =0.26059
β =90°
|Ξ3/A| (m/m)
T (sec)

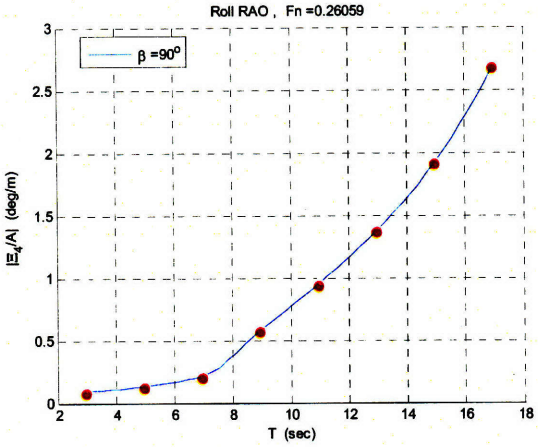

Pitch RAO , Fn =0.26059
β =90°
|Ξ5/A| (deg/m)
T (sec)

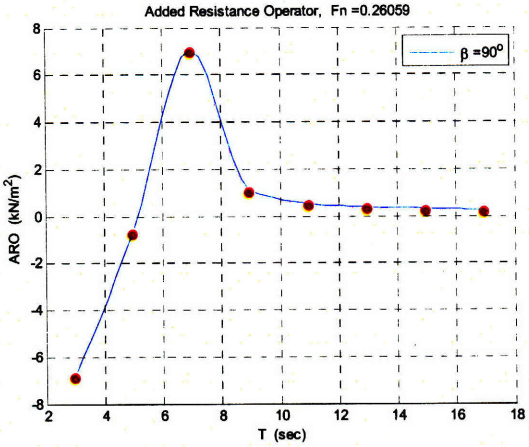

Heave RAO , Fn =0.26059
β =45°
|Ξ3/A| (m/m)
T (sec)

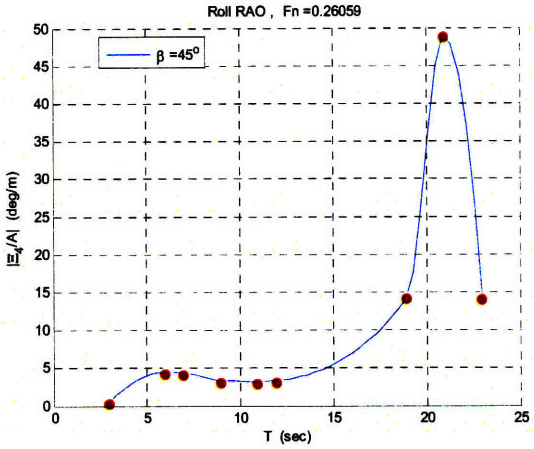

Pitch RAO , Fn =0.26059
β =45°
|Ξ5/A| (deg/m)
T (sec)

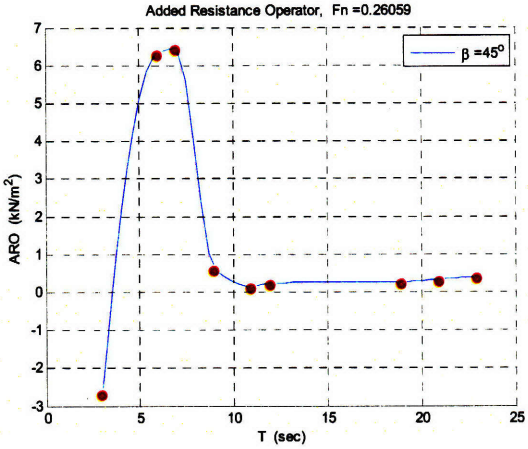

Roll RAO , Fn =0.26059
β =135°
|Ξ4/A| (deg/m)
T (sec)


Added Resistance Operator, Fn =0.26059
β =135°
ARO (kN/m2)
T (sec)


Roll RAO , Fn =0.26059
β =90°
|Ξ4/A| (deg/m)
T (sec)


Added Resistance Operator, Fn =0.26059
β =90°
ARO (kN/m2)
T (sec)


Roll RAO , Fn =0.26059
β =45°
|Ξ4/A| (deg/m)
T (sec)


Added Resistance Operator, Fn =0.26059
β =45°
ARO (kN/m2)
T (sec)

Appendix B: WAM4 Sample Output

Results for the area :
Lon = 0.0000000E+00 to 359.5000
Lat = -80.00000 to 80.00000
LON = Longidute
LAT = Latidute
Forecasted Parameters :
S_WHT = Significant Wave Height in m
MEANWDIR = Mean Wave Direction in deg (meteorological convention)
MAX_WHT = Maximum Expected Wave Height in m
PEAK_FR = Peak Frequency in Hz
MEAN_FR = Mean Frequency in Hz
Resolution = 0.5 degrees

LON	LAT	S_WHT	MEANWDIR	MAX_WHT	PEAK_FR	MEAN_FR
236.500	-69.500	1.70	217.836	3.06	0.174	0.203
237.000	-69.500	1.99	221.364	3.582	0.174	0.185
237.500	-69.500	2.13	225.195	3.834	0.159	0.173
238.000	-69.500	2.24	229.355	4.032	0.159	0.163
238.500	-69.500	2.36	233.960	4.248	0.159	0.155
239.000	-69.500	2.47	239.458	4.446	0.144	0.147
287.500	-69.500	1.12	37.134	2.016	0.211	0.239
288.000	-69.500	0.91	37.205	1.638	0.211	0.260
236.500	-69.000	1.92	214.161	3.456	0.174	0.188
237.000	-69.000	2.26	217.708	4.068	0.159	0.171
237.500	-69.000	2.44	221.602	4.392	0.159	0.160
238.000	-69.000	2.60	225.939	4.68	0.144	0.152
238.500	-69.000	2.73	230.521	4.914	0.144	0.145
239.000	-69.000	2.86	235.675	5.148	0.144	0.139
293.000	-69.000	0.90	74.378	1.62	0.255	0.290
235.500	-68.500	1.85	218.593	3.33	0.174	0.200
236.000	-68.500	2.15	220.530	3.87	0.174	0.183
236.500	-68.500	2.42	218.865	4.356	0.159	0.168
237.000	-68.500	2.61	220.309	4.698	0.144	0.158
237.500	-68.500	2.76	223.278	4.968	0.144	0.149
238.000	-68.500	2.91	227.131	5.238	0.144	0.142
238.500	-68.500	3.04	231.561	5.472	0.131	0.136
239.000	-68.500	3.18	236.498	5.724	0.131	0.131
239.500	-68.500	3.13	246.191	5.634	0.131	0.125
240.000	-68.500	3.21	252.171	5.778	0.131	0.121
240.500	-68.500	3.32	256.938	5.976	0.131	0.118
241.000	-68.500	3.42	261.454	6.156	0.131	0.114
241.500	-68.500	3.50	266.104	6.3	0.067	0.110
242.000	-68.500	3.53	271.143	6.354	0.067	0.106
242.500	-68.500	3.53	276.469	6.354	0.067	0.101
243.000	-68.500	3.54	280.895	6.372	0.067	0.098
243.500	-68.500	3.56	284.717	6.408	0.067	0.094
244.000	-68.500	3.58	287.676	6.444	0.067	0.091
244.500	-68.500	3.62	289.631	6.516	0.067	0.089
246.500	-68.500	3.59	297.673	6.462	0.067	0.080
260.500	-68.500	0.90	333.920	1.62	0.074	0.090
261.000	-68.500	1.10	326.861	1.98	0.074	0.090

220.000	36.000	2.67	282.256	4.806	0.056	0.070
220.500	36.000	2.65	281.484	4.77	0.051	0.070
221.000	36.000	2.62	280.505	4.716	0.051	0.070
221.500	36.000	2.59	279.432	4.662	0.051	0.070
222.000	36.000	2.57	278.188	4.626	0.051	0.070
222.500	36.000	2.55	276.844	4.59	0.051	0.071
223.000	36.000	2.53	275.412	4.554	0.051	0.071
223.500	36.000	2.52	273.915	4.536	0.051	0.071
224.000	36.000	2.51	272.365	4.518	0.051	0.072
224.500	36.000	2.49	270.767	4.482	0.051	0.072
225.000	36.000	2.48	269.132	4.464	0.051	0.072
225.500	36.000	2.47	267.479	4.446	0.051	0.073
226.000	36.000	2.46	265.835	4.428	0.051	0.073
226.500	36.000	2.45	264.227	4.41	0.051	0.073
227.000	36.000	2.44	262.677	4.392	0.074	0.073
227.500	36.000	2.44	261.198	4.392	0.074	0.074
228.000	36.000	2.43	259.801	4.374	0.074	0.074
228.500	36.000	2.43	258.490	4.374	0.074	0.074
229.000	36.000	2.43	257.265	4.374	0.074	0.075
229.500	36.000	2.42	256.115	4.356	0.074	0.075
230.000	36.000	2.42	255.018	4.356	0.074	0.075
230.500	36.000	2.42	253.946	4.356	0.074	0.075
231.000	36.000	2.42	252.883	4.356	0.074	0.076
231.500	36.000	2.42	251.834	4.356	0.074	0.076
232.000	36.000	2.41	250.822	4.338	0.074	0.076
232.500	36.000	2.41	249.880	4.338	0.074	0.076
233.000	36.000	2.41	249.038	4.338	0.074	0.076
233.500	36.000	2.42	248.336	4.356	0.074	0.076
234.000	36.000	2.42	248.004	4.356	0.074	0.077
234.500	36.000	2.44	248.329	4.392	0.074	0.077
235.000	36.000	2.47	250.390	4.446	0.074	0.079
235.500	36.000	2.55	256.325	4.59	0.067	0.082
236.000	36.000	2.59	259.751	4.662	0.067	0.083
236.500	36.000	2.65	263.540	4.77	0.067	0.085
237.000	36.000	2.75	270.165	4.95	0.067	0.089
237.500	36.000	2.94	278.321	5.292	0.067	0.095
238.000	36.000	2.83	270.982	5.094	0.067	0.094
283.500	36.000	0.28	39.495	0.504	0.453	0.348
284.000	36.000	0.76	132.045	1.368	0.232	0.201
284.500	36.000	1.41	183.364	2.538	0.131	0.162
285.000	36.000	1.81	192.245	3.258	0.131	0.145
285.500	36.000	2.18	197.596	3.924	0.131	0.138
286.000	36.000	2.38	204.837	4.284	0.119	0.140
286.500	36.000	2.60	209.890	4.68	0.119	0.141
287.000	36.000	2.78	212.792	5.004	0.119	0.141
287.500	36.000	2.89	214.998	5.202	0.119	0.139
288.000	36.000	2.92	217.362	5.256	0.119	0.138
288.500	36.000	2.92	220.126	5.256	0.119	0.135
289.000	36.000	2.93	223.123	5.274	0.119	0.134
289.500	36.000	2.94	226.106	5.292	0.119	0.132
290.000	36.000	2.94	228.948	5.292	0.119	0.131
290.500	36.000	2.93	231.574	5.274	0.119	0.130
291.000	36.000	2.91	233.927	5.238	0.119	0.130
291.500	36.000	2.87	235.933	5.166	0.119	0.130
292.000	36.000	2.83	237.640	5.094	0.119	0.130
292.500	36.000	2.77	239.032	4.986	0.119	0.131

Swell Results for the area :
Lon = 0.0000000E+00 to 359.5000
Lat = -80.00000 to 80.00000
LON = Longidute
LAT = Latidute
Forecasted Parameters :
SWE_H = Swell Wave Height in m
MEANSDIR = Mean Swell Direction in deg (meteorologcal convention)
MEANSFR = Mean Swell Frequency in Hz
Resolution = 0.5 degrees

LON	LAT	SWE_H	MEANSDIR	MEANSFR
236.500	-69.500	0.34	87.448	0.113
237.000	-69.500	0.42	339.841	0.101
237.500	-69.500	0.56	316.223	0.097
238.000	-69.500	0.68	313.207	0.094
238.500	-69.500	0.83	310.888	0.092
239.000	-69.500	0.98	309.833	0.090
287.500	-69.500	0.21	341.821	0.197
288.000	-69.500	0.27	343.724	0.213
236.500	-69.000	0.50	139.998	0.114
237.000	-69.000	0.61	268.679	0.102
237.500	-69.000	0.80	263.771	0.100
238.000	-69.000	0.89	301.975	0.093
238.500	-69.000	1.03	303.944	0.090
239.000	-69.000	1.17	306.707	0.087
293.000	-69.000	0.08	275.885	0.215
235.500	-68.500	0.37	129.482	0.128
236.000	-68.500	0.51	137.981	0.113
236.500	-68.500	0.70	170.874	0.106
237.000	-68.500	0.92	209.009	0.103
237.500	-68.500	1.03	272.840	0.096
238.000	-68.500	1.14	298.217	0.091
238.500	-68.500	1.29	303.053	0.088
239.000	-68.500	1.46	302.131	0.086
239.500	-68.500	1.57	306.815	0.083
240.000	-68.500	1.75	306.458	0.082
240.500	-68.500	1.94	305.752	0.082
241.000	-68.500	2.12	304.952	0.081
241.500	-68.500	2.34	303.597	0.082
242.000	-68.500	2.52	303.234	0.081
242.500	-68.500	2.73	301.689	0.082
243.000	-68.500	2.90	300.908	0.082
243.500	-68.500	3.02	301.080	0.081
244.000	-68.500	3.13	301.258	0.080
244.500	-68.500	3.22	301.321	0.079
245.000	-68.500	3.30	301.214	0.079
245.500	-68.500	3.39	300.756	0.078
246.000	-68.500	3.47	300.035	0.078
246.500	-68.500	3.55	298.774	0.079
260.500	-68.500	0.90	334.114	0.090
261.000	-68.500	1.09	327.093	0.090
261.500	-68.500	1.21	323.881	0.090
262.000	-68.500	1.30	322.034	0.090

220.000	45.000	3.56	277.079	0.076
220.500	45.000	3.45	275.998	0.075
221.000	45.000	3.45	275.898	0.075
221.500	45.000	3.39	273.982	0.075
222.000	45.000	3.21	272.947	0.073
222.500	45.000	3.21	272.265	0.073
223.000	45.000	3.16	272.134	0.073
223.500	45.000	3.19	272.710	0.074
224.000	45.000	3.17	272.474	0.074
224.500	45.000	3.31	276.448	0.075
225.000	45.000	3.30	275.780	0.075
225.500	45.000	3.22	276.593	0.075
226.000	45.000	3.34	278.538	0.077
226.500	45.000	3.46	278.795	0.078
227.000	45.000	3.47	278.842	0.079
227.500	45.000	3.55	278.894	0.080
228.000	45.000	3.69	279.191	0.082
228.500	45.000	3.80	279.414	0.084
229.000	45.000	3.82	278.355	0.084
229.500	45.000	3.87	277.539	0.086
230.000	45.000	3.83	276.305	0.087
230.500	45.000	3.74	273.646	0.087
231.000	45.000	3.58	270.152	0.086
231.500	45.000	3.35	265.768	0.084
232.000	45.000	3.20	261.895	0.084
232.500	45.000	3.07	257.743	0.084
233.000	45.000	2.91	252.968	0.082
233.500	45.000	2.79	247.621	0.080
234.000	45.000	2.76	240.529	0.081
234.500	45.000	2.70	241.729	0.079
235.000	45.000	2.57	246.929	0.076
235.500	45.000	2.46	249.181	0.074
236.000	45.000	1.79	271.566	0.079
293.000	45.000	1.18	177.349	0.120
293.500	45.000	1.30	186.789	0.118
294.000	45.000	0.63	196.047	0.131
294.500	45.000	0.32	201.746	0.159
298.000	45.000	1.85	115.307	0.122
298.500	45.000	2.48	119.776	0.117
299.000	45.000	2.51	124.757	0.114
299.500	45.000	2.45	133.722	0.112
300.000	45.000	2.38	141.588	0.111
300.500	45.000	2.34	147.234	0.110
301.000	45.000	2.12	163.315	0.109
301.500	45.000	1.76	163.342	0.109
302.000	45.000	1.75	173.609	0.109
302.500	45.000	1.77	175.993	0.109
303.000	45.000	1.74	178.791	0.108
303.500	45.000	1.68	181.670	0.108
304.000	45.000	1.61	183.599	0.109
304.500	45.000	1.63	179.585	0.110
305.000	45.000	1.64	175.602	0.111
305.500	45.000	1.70	167.196	0.109
306.000	45.000	1.76	158.486	0.109
306.500	45.000	1.82	147.857	0.109
307.000	45.000	1.88	135.181	0.109
307.500	45.000	1.92	107.516	0.109

Appendix C: MATLAB® Scripts

```
% WAVEROUTE.m
% Optimization code for minimum fuel navigation.
% Solves the constrained problem, accounting for added resistance
% in rough seas. Uses real forecast data from the WAM4 model.
% Safety constraints considered: slamming and deck wetness.
% For the part of the trip beyond forecast coverage calm seas assumed.

clear all
warning off % disable "division by zero" warning
tic
load ship_particulars % load L,T,B,f etc.
load TransferFunctions % load RAO_r1,ARO,omega,RAO_r2,RAO_v,RAO_a,
                       % RAO_roll,W,U,C
load PropData % load EAR,P_D,z,RR,UU,KT,JJ,nTRM,nH,nR
load NorthAtlantic5Apr08_with_swell % load wave and swell data
load CalmRes % load calm water resistance vector Rt
Vcr=0.093*sqrt(L*g);  % critical speed in slamming criterion (m/sec)
Limit_dw=0.07;  % probability limit for water on deck
Limit_sl=0.03;  % probability limit for slamming
sfc=170e-6/3600; % typical specific fuel consumption of a low speed
                 % diesel engine in tonne/kWsec
P=36;       % number of stages covered by forecast
IT=30;       % no of iterations
R_speed=10; % no of speed settings
R_course=16;% no of course settings
R=R_speed*R_course; % total no of admissible controls
nominal_speed=13;   % nominal cruising speed in knots

% set departure and destination location
lambda1=330;  % longitude of point 1 (deg)
phi1=40.5;    % latitude of point 1 (deg)
lambda2=287; % longitude of point 2 (deg)
phi2=40;      % latitude of point 2 (deg)

% map spherical coordinates onto a plane using Mercator projection
[X1,Y1]=MercProj(lambda1,phi1);  % departure point
[X2,Y2]=MercProj(lambda2,phi2);  % destination point
% longitude and latitude columns converted to coordinates
LON=swell(:,1,1);
LAT=swell(:,2,1);
% determine slope of loxodrome (rhumb line)
slope=atan2((Y2-Y1),(X2-X1));
if slope<0
    % westbound routes
    slope=slope+2*pi;
end

VoyageDist=sqrt((X2-X1)^2+(Y2-Y1)^2)/1852; % distance in n.m.
VoyageHours=VoyageDist/nominal_speed;  % nominal voyage duration
ForecastHours=108;     % period of forecast coverage
dt=ForecastHours/P;    % time increment for weather forecast (3h)
dt=3600*dt;            % convert to sec
tf=ForecastHours*3600;  % forecast time in sec
```

```
tv=VoyageHours*3600;    % voyage time in sec
% t1: time period not covered by forecast (calm seas assumed);
% if forecast exceeds voyage time t1 is set to 0
t1=(VoyageHours-ForecastHours)*3600*(VoyageHours>ForecastHours);
% if voyage shorter than forecast coverage adjust final stage length
dt1
if ForecastHours>VoyageHours
    P=floor(tv/dt);
    tf=P*dt;
    dt1=tv-tf;
else
    dt1=t1;
end
N=P+2;
tt=0:dt:tf;

% initial guess for controls (nominal)
p=slope*ones(1,N-1);
u=0.51444*nominal_speed*ones(1,N-1);
% nominal (rhumb line) trajectory
x=zeros(1,N);
y=zeros(1,N);
x(1:N-1)=X1'+u(1)*cos(p(1))*tt; x(N)=x(N-1)+u(1)*cos(p(1))*dt1;
y(1:N-1)=Y1+u(1)*sin(p(1))*tt; y(N)=y(N-1)+u(1)*sin(p(1))*dt1;
x0=[X1+0.51444*nominal_speed*cos(slope)*tt X2];
y0=[Y1+0.51444*nominal_speed*sin(slope)*tt Y2];
Imin=zeros(1,IT); % initialize optimal value function for each
iteration
r=1;              % control region size
gamma=0.7;  % control region contraction factor
% define admissible controls; ensure that 0 is included in the
% constructed control increments
dU=zeros(1,R);
dC=zeros(1,R);
deltaU1=0.51444*linspace(-6,0,R_speed/2);
deltaU2=0.51444*linspace(6/(R_speed/2),6,R_speed/2);
deltaU=[deltaU1 deltaU2]; % speed increments in m/s
deltaC1=linspace(-pi/2,0,R_course/2);
deltaC2=linspace(pi/R_course,pi/2,R_course/2);
deltaC=[deltaC1 deltaC2]; % heading increments in radians
[dU,dC]=meshgrid(deltaU,deltaC); % create all control pairs
% convert admissible controls to vectors
dU=reshape(dU,1,R);
dC=reshape(dC,1,R);
% set speed range for steady resistance interpolation
speed=0.51444*linspace(7,16,10);

% start iterations
for j=1:IT

    I=zeros(1,R);    % value function for each set of controls
    Iopt=zeros(1,N); % optimal value function at each stage
    % initialize trajectory buffer for each pair of controls
    X=ndgrid(x,1:R);
    Y=ndgrid(y,1:R);
```

```
    % preallocate memory for variables to be used below
    ind=0; counter=0; meter=0; ind1=0; index1_p=0; index2_p=0;
    uu=zeros(1,R); u_buffer=zeros(1,R);
    pp=zeros(1,R); dfd=zeros(1,R); u_buffer=zeros(1,R);
    aro_wave=zeros(length(omega),1);
    aro_swell=zeros(length(omega),1);
    RAO1_w=zeros(length(omega),1);
    RAO2_w=zeros(length(omega),1);
    RAO3_w=zeros(length(omega),1);
    RAO1_s=zeros(length(omega),1);
    RAO2_s=zeros(length(omega),1);
    RAO3_s=zeros(length(omega),1);
    WaveSpectr=zeros(length(omega),1);
    SwellSpectr=zeros(length(omega),1);

    % begin to step backwards in time
    for k=2:N-1
        % store all possible control settings
        uu=u(N-k)+dU*r;
        pp=p(N-k)+dC*r;

        % loop through available controls
        for m=1:R

            % check if controls are out of bounds
            if uu(m)<=3.6011
                uu(m)=3.60111;
            elseif uu(m)>=8.231
                uu(m)=8.231;
            end
            if pp(m)<0
                pp(m)=pp(m)+2*pi;
            elseif pp(m)>2*pi
                pp(m)=pp(m)-2*pi;
            end

            Rcalm=pchip(speed,Rt,uu(m)); % interpolate steady
resistance

            % integrate equations one step forward
            X(N-k+1,m)=x(N-k)+dt*uu(m)*cos(pp(m));
            Y(N-k+1,m)=y(N-k)+dt*uu(m)*sin(pp(m));
            % find point on map closest to reached state
(X(N-k+1,m),Y(N-k+1,m))
            d_square=(X(N-k+1,m)-LAT).^2+(Y(N-k+1,m)-LON).^2;
            closest=find(d_square==min(d_square)); %row index of
closest point
            closest=closest(1); % keep first entry if more than one
            % get wave and swell data at this point
            Hwave=wave(closest,3,N-k);
            theta_wave=wave(closest,4,N-k);
            freq_wave=wave(closest,5,N-k);
            Hswell=swell(closest,3,N-k);
            theta_swell=swell(closest,4,N-k);
            freq_swell=swell(closest,5,N-k);
            % calculate wave direction relative to ship's course;
```

```
            % the result should be port-starboard symmetric
            relative_wave_direction=abs(pp(m)-theta_wave);
            if relative_wave_direction>pi
                relative_wave_direction=2*pi-relative_wave_direction;
            end
            % an additional correction is needed because 180 deg
            % means head waves in seakeeping (opposite to forecast
            % output)
            relative_wave_direction=abs(relative_wave_direction-pi);
            relative_swell_direction=abs(pp(m)-theta_swell);
            if relative_swell_direction>pi
                relative_swell_direction=2*pi-relative_swell_direction;
            end
            relative_swell_direction=abs(relative_swell_direction-pi);

            % interpolate to get response amplitude
            % operators as a function of frequency omega

RAO1_w=interpn(W,U,C,RAO_r1,omega,uu(m),relative_wave_direction);
RAO2_w=interpn(W,U,C,RAO_r2,omega,uu(m),relative_wave_direction);
RAO3_w=interpn(W,U,C,RAO_v,omega,uu(m),relative_wave_direction);
RAO1_s=interpn(W,U,C,RAO_r1,omega,uu(m),relative_swell_direction);
RAO2_s=interpn(W,U,C,RAO_r2,omega,uu(m),relative_swell_direction);
RAO3_s=interpn(W,U,C,RAO_v,omega,uu(m),relative_swell_direction);

% calculate two separate Bretschneider spectra for wave and swell

WaveSpectr=0.278*freq_wave^4./omega.^5.*(Hwave/1.6)^2.*exp(-
0.437*(freq_wave./omega).^4);
            SwellSpectr=0.278*freq_swell^4./omega.^5.*(Hswell/1.6)^2.*
exp(-0.437*(freq_swell./omega).^4);

% integrate to get response variances
m0_r1=trapz(omega,(WaveSpectr.*RAO1_w.^2+SwellSpectr.*RAO1_s.^2)); %
variance of relative bow motion

m0_r2=trapz(omega,(WaveSpectr.*RAO2_w.^2+SwellSpectr.*RAO2_s.^2)); %
variance of relative bow motion (keel)

m0_v=trapz(omega,(WaveSpectr.*RAO3_w.^2+SwellSpectr.*RAO3_s.^2)); %
variance of relative bow velocity
            % define probability of occurence of composite events
            Prob1=exp(-f^2./(2*m0_r1)); % probability of deck wetness
(i.e. freeboard exceedance)
            Prob2=exp(-(T^2./(2*m0_r2)+Vcr^2./(2*m0_v))); % probability
of slamming
            % check if constraints are met, otherwise skip controls
            if Prob1>Limit_dw||Prob2>Limit_sl
                I(m)=1e200;
                counter=counter+1;
                if counter==R
                    disp('all controls inadmissible')
                    return
                end
                continue
            end
            % interpolate to get added resistance operators
```

```
aro_wave=interpn(W,U,C,ARO,omega,uu(m),relative_wave_direction);

aro_swell=interpn(W,U,C,ARO,omega,uu(m),relative_swell_direction);
            % integrate to get mean added resistance
            Rwave=2*trapz(omega,WaveSpectr.*aro_wave);
            Rswell=2*trapz(omega,SwellSpectr.*aro_swell);
            Rtotal=Rwave+Rswell+Rcalm;
            % extract data from propeller curves
            Kt_coef1=interp2(RR,UU,KT,Rtotal,uu(m),'cubic'); % thrust
coefficient
            Jcoef1=interp2(RR,UU,JJ,Rtotal,uu(m),'cubic');    % advance
ratio
            Kq_coef1=pchip(J1,KQ,Jcoef1);         % torque coefficient
            nO=Kt_coef1*Jcoef1/Kq_coef1/(2*pi);  % open water
efficiency
            nD=nH*nO*nR;    % quasi-propulsive coefficient
            q1=sfc*Rtotal*uu(m)/nD/nTRM;    % fuel consumption rate
            I(m)=q1*dt; % fuel cost of this step

            for s=1:k-1
                if (N-k+s+1)<N
                    delta_t=dt;
                    X(N-k+s+1,m)=X(N-k+s,m)+delta_t*u(N-k+s)*
cos(p(N-k+s));
                    Y(N-k+s+1,m)=Y(N-k+s,m)+delta_t*u(N-k+s)*
sin(p(N-k+s));
                    Rcalm=pchip(speed,Rt,u(N-k+s));
                    % find point on map closest to (X(N-k+s+1,m),
Y(N-k+s+1,m))
                    d_square=(X(N-k+s+1,m)-LAT).^2+(Y(N-k+s+1,m)-
LON).^2;
                    closest=find(d_square==min(d_square)); %row index
of closest point
                    closest=closest(1); % keep first entry if more than
one
                    % get wave and swell data at this point
                    Hwave=wave(closest,3,N-k+s);
                    theta_wave=wave(closest,4,N-k+s);
                    freq_wave=wave(closest,5,N-k+s);
                    Hswell=swell(closest,3,N-k+s);
                    theta_swell=swell(closest,4,N-k+s);
                    freq_swell=swell(closest,5,N-k+s);
                    % calculate wave direction relative to ship's
course;
                    relative_wave_direction=abs(p(N-k+s)-theta_wave);
                    if relative_wave_direction>pi
                        relative_wave_direction=2*pi-
relative_wave_direction;
                    end
relative_wave_direction=abs(relative_wave_direction-pi);
                    relative_swell_direction=abs(p(N-k+s)-theta_swell);
                    if relative_swell_direction>pi
                        relative_swell_direction=2*pi-
relative_swell_direction;
                    end
```

```
relative_swell_direction=abs(relative_swell_direction-pi);
% calculate two separate Bretschneider spectra for wave and swell

WaveSpectr=0.278*freq_wave^4./omega.^5.*(Hwave/1.6)^2.*exp(-
0.437*(freq_wave./omega).^4);

SwellSpectr=0.278*freq_swell^4./omega.^5.*(Hswell/1.6)^2.*exp(-
0.437*(freq_swell./omega).^4);
                    % interpolate to get added resistance operator as a
function
                    % of frequency omega
                    aro_wave=interpn(W,U,C,ARO,omega,u(N-
k+s),relative_wave_direction);
                    aro_swell=interpn(W,U,C,ARO,omega,u(N-
k+s),relative_swell_direction);
                    % integrate to get mean added resistance
                    Rwave=2*trapz(omega,WaveSpectr.*aro_wave);
                    Rswell=2*trapz(omega,SwellSpectr.*aro_swell);
                    Rtotal=Rwave+Rswell+Rcalm;
                else

                    delta_t=dt1;
                    dfd(m)=sqrt((X2-X(N-1,m)).^2+(Y2-Y(N-1,m)).^2);
                    u_buffer(m)=dfd(m)/delta_t;

                    if u_buffer>8.231
                        I(m)=1e200;
                        counter=counter+1;
                        if counter==R
                            disp('all controls inadmissible')
                            return
                        end
                        continue
                    end
                    p_buffer(m)=atan2((Y2-Y(N-1,m)),(X2-X(N-1,m)));
                    if p_buffer(m)<0
                        p_buffer(m)=p_buffer(m)+2*pi;
                    elseif p_buffer(m)>2*pi
                        p_buffer(m)=p_buffer(m)-2*pi;
                    end
                    Rcalm=pchip(speed,Rt,u_buffer(m));
                    X(N-k+s+1,m)=X(N-k+s,m)
+delta_t*u_buffer(m)*cos(p_buffer(m));
                    Y(N-k+s+1,m)=Y(N-k+s,m)
+delta_t*u_buffer(m)*sin(p_buffer(m));
                    if t1==0
                        % find point on map closest to (X(N-
k+s+1,m),Y(N-k+s+1,m))
                        d_square=(X(N-k+s+1,m)-LAT).^2+(Y(N-k+s+1,m)-
LON).^2;
                        closest=find(d_square==min(d_square)); %row
index of closest point
                        closest=closest(1); % keep first entry if more
than one
                        % get wave and swell data at this point
```

```
                        Hwave=wave(closest,3,N-k+s);
                        theta_wave=wave(closest,4,end);
                        freq_wave=wave(closest,5,end);
                        Hswell=swell(closest,3,end);
                        theta_swell=swell(closest,4,end);
                        freq_swell=swell(closest,5,end);
                        % calculate wave direction relative to ship's
course;
                        relative_wave_direction=abs(p(N-k+s)-
theta_wave);
                        if relative_wave_direction>pi
                            relative_wave_direction=2*pi-
relative_wave_direction;
                        end

relative_wave_direction=abs(relative_wave_direction-pi);
                        relative_swell_direction=abs(p(N-k+s)-
theta_swell);
                        if relative_swell_direction>pi
                            relative_swell_direction=2*pi-
relative_swell_direction;
                        end

relative_swell_direction=abs(relative_swell_direction-pi);
                        % calculate two separate Bretschneider spectra
for wave and swell

WaveSpectr=0.278*freq_wave^4./omega.^5.*(Hwave/1.6)^2.*exp(-
0.437*(freq_wave./omega).^4);

SwellSpectr=0.278*freq_swell^4./omega.^5.*(Hswell/1.6)^2.*exp(-
0.437*(freq_swell./omega).^4);
                        % interpolate to get added resistance operator
as a function
                        % of frequency omega
                        aro_wave=interpn(W,U,C,ARO,omega,
u(N-k+s),relative_wave_direction);
                        aro_swell=interpn(W,U,C,ARO,omega,
u(N-k+s),relative_swell_direction);
                        % integrate to get mean added resistance
                        Rwave=2*trapz(omega,WaveSpectr.*aro_wave);
                        Rswell=2*trapz(omega,SwellSpectr.*aro_swell);
                        Rtotal=Rwave+Rswell+Rcalm;
                    else
                        Rtotal=Rcalm;
                    end


                end
                Kt_coef1=interp2(RR,UU,KT,Rtotal,u(N-k+s),'cubic');
                Jcoef1=interp2(RR,UU,JJ,Rtotal,u(N-k+s),'cubic');
                Kq_coef1=pchip(J1,KQ,Jcoef1);
                nO=Kt_coef1*Jcoef1/Kq_coef1/(2*pi);
                nD=nH*nO*nR;
                q1=sfc*Rtotal*u(N-k+s)/nD/nTRM;
                I(m)=I(m)+q1*delta_t;
```

```
            end
        end
        % find index for which value function is minimized
        ind=find(I==min(I)); ind=ind(1); % if more than one, choose
first
        % store optimal control history
        u(N-k)=uu(ind);
        p(N-k)=pp(ind);
        u(N-1)=u_buffer(ind);
        p(N-1)=p_buffer(ind);
        % store optimal value
        Iopt(N-k)=I(ind);
        % reset parameters
        I=zeros(1,R);
        counter=0;

    end
    % generate the optimal trajectory for this iteration
    % (to be used as initial trajectory in the next iteration)
    x=X(:,ind);  y=Y(:,ind);

    % store minimum value for each iteration
    Imin(j)=Iopt(1);
    r=gamma*r; % contract control region
end

% verify that optimal route satisfies constraints
for k=1:P
    d_square=(x(k+1)-LAT).^2+(y(k+1)-LON).^2;
    closest=find(d_square==min(d_square));
    closest=closest(1);
    Hwave=wave(closest,3,k);
    theta_wave=wave(closest,4,k);
    freq_wave=wave(closest,5,k);
    Hswell=swell(closest,3,k);
    theta_swell=swell(closest,4,k);
    freq_swell=swell(closest,5,k);
    relative_wave_direction=abs(p(k)-theta_wave);
    if relative_wave_direction>pi
        relative_wave_direction=2*pi-relative_wave_direction;
    end
    relative_wave_direction=abs(relative_wave_direction-pi);
    relative_swell_direction=abs(p(k)-theta_swell);
    if relative_swell_direction>pi
        relative_swell_direction=2*pi-relative_swell_direction;
    end
    relative_swell_direction=abs(relative_swell_direction-pi);
    RAO1_w=interpn(W,U,C,RAO_r1,omega,u(k),relative_wave_direction);
    RAO2_w=interpn(W,U,C,RAO_r2,omega,u(k),relative_wave_direction);
    RAO3_w=interpn(W,U,C,RAO_v,omega,u(k),relative_wave_direction);
    RAO1_s=interpn(W,U,C,RAO_r1,omega,u(k),relative_swell_direction);
    RAO2_s=interpn(W,U,C,RAO_r2,omega,u(k),relative_swell_direction);
    RAO3_s=interpn(W,U,C,RAO_v,omega,u(k),relative_swell_direction);
    WaveSpectr=0.278*freq_wave^4./omega.^5.*(Hwave/1.6)^2.*
exp(-0.437*(freq_wave./omega).^4);
    SwellSpectr=0.278*freq_swell^4./omega.^5.*(Hswell/1.6)^2.*
```

```
exp(-0.437*(freq_swell./omega).^4);
    m0_r1=trapz(omega,(WaveSpectr.*RAO1_w.^2+SwellSpectr.*RAO1_s.^2));
    m0_r2=trapz(omega,(WaveSpectr.*RAO2_w.^2+SwellSpectr.*RAO2_s.^2));
    m0_v=trapz(omega,(WaveSpectr.*RAO3_w.^2+SwellSpectr.*RAO3_s.^2));
    Prob1=exp(-f^2./(2*m0_r1));
    Prob2=exp(-(T^2./(2*m0_r2)+Vcr^2./(2*m0_v)));
    % check if constraints are met; if not record how many times they
are violated
    if Prob1>Limit_dw||Prob2>Limit_sl
        meter=meter+1;
    end
end
if meter==0
    disp('Optimal solution found. All constraints are satisfied.')
else
    disp(strcat('Constraints are violated   ',num2str(meter),'
times.'))
end
TIME=toc
% total distance covered:
Ltotal=(sum(u(1:end-1))*dt+u(end)*delta_t)/1852;
Lmax=16*VoyageHours; % max possible distance for given voyage time
MINCOST=Imin(end);
% Plot results
figure
hold on
plot(y0/1852,x0/1852,'--b',y/1852,x/1852,'r')
plot(Y1/1852,X1/1852,'o','MarkerEdgeColor','k','MarkerFaceColor','g')
plot(Y2/1852,X2/1852,'o','MarkerEdgeColor','k','MarkerFaceColor','r')
plot(y(2:end-1)/1852,x(2:end-1)/1852,'xk')
legend('loxodrome','optimal','start','finish')
xlabel('Nautical Miles West of Greenwich Meridian')
ylabel('Nautical Miles North of Equator')
title('Ship Route on Mercator Projection')
grid on

figure
plot(2:IT,Imin(2:end),':*')
xlabel('Number of Iterations')
ylabel('Optimal Cost Function')
title('Convergence of IDP Algorithm')

save Res5Apr_gamma7_R_high x y x0 y0 u p MINCOST Imin IT Ltotal X1 X2
Y1 Y2 TIME meter
```

```
% loxodrome.m
% Program to calculate cost of rhumb line routing in rough seas
% assuming no constraints apply. The ship is following the loxodrome
% (rhumb line) which is the route of constant heading.
% The path is represented by a straight line on a Mercator map.
% WAM forecast is used. The fraction of forecast time for which
constraintsare violated is calculated.

clear all
load ship_particulars % load L,T,B,f etc.
load TransferFunctions % load RAO_r1,ARO,omega,RAO_r2,RAO_v,RAO_a,
% RAO_roll,W,U,C
load PropData % load EAR,P_D,z,RR,UU,KT,JJ,nTRM,nH,nR
load NorthAtlantic5Apr08_with_swell % load wave and swell data
load CalmRes % load calm water resistance vector Rt
sfc=170e-6/3600; % specific fuel consumption of a low speed Wartsila
% diesel in tonne/kWsec
Limit_dw=0.07;  % probability limit for water on deck
Limit_sl=0.03;  % probability limit for slamming
Vcr=0.093*sqrt(L*g);  % critical speed in slamming criterion (m/sec)
tf=108;  % final time (in hours)
P=36;     % number of stages covered by forecast
dt=tf/P;   % time step for forecast stages
N=P+2;   % no of stations dividing time axis
tt=0:dt:tf;  % time discretization
tt=3600*tt;    dt=3600*dt; % convert time to sec
nominal_speed=13*0.51444;  % nominal speed in m/s
% set departure and destination location
lambda1=330;  % longitude of point 1 (deg)
phi1=40.5;   % latitude of point 1 (deg)
lambda2=287; % longitude of point 2 (deg)
phi2=40;     % latitude of point 2 (deg)

% map spherical coordinates onto a plane using Mercator projection
[X1,Y1]=MercProj(lambda1,phi1);
[X2,Y2]=MercProj(lambda2,phi2);
% longitude and latitude columns converted to coordinates
LON=wave(:,1,1);
LAT=wave(:,2,1);
% determine slope of loxodrome (rhumb line)
slope=atan2((Y2-Y1),(X2-X1));
if slope<0
    % westbound routes
    slope=slope+2*pi;
end

VoyageDist=sqrt((X2-X1)^2+(Y2-Y1)^2)/1852; % distance in n.m.
VoyageHours=VoyageDist/13;  % nominal voyage duration
ForecastHours=108;          % period of forecast coverage

% time step of final stage, not covered by forecast (calm seas assumed)
dt1=(VoyageHours-ForecastHours)*3600*(VoyageHours>ForecastHours);

% nominal controls
p=slope;
u=nominal_speed;
```

```
% nominal (rhumb line) trajectory
x=zeros(1,N);
y=zeros(1,N);
x(1:end-1)=X1+u(1)*cos(p(1))*tt;
y(1:end-1)=Y1+u(1)*sin(p(1))*tt;
x(end)=x(end-1)+u(end)*cos(p(end))*dt1;
y(end)=y(end-1)+u(end)*sin(p(end))*dt1;

% initialize variables
RAO1_w=zeros(length(omega),1);
RAO2_w=zeros(length(omega),1);
RAO3_w=zeros(length(omega),1);
RAO1_s=zeros(length(omega),1);
RAO2_s=zeros(length(omega),1);
RAO3_s=zeros(length(omega),1);
aro_wave=zeros(length(omega),1);
aro_swell=zeros(length(omega),1);
WaveSpectr=zeros(length(omega),1);
SwellSpectr=zeros(length(omega),1);
I=zeros(1,N); counter=zeros(1,N-1);

% set speed range for resistance interpolation
speed=0.51444*linspace(7,16,10);
Rcalm=pchip(speed,Rt,u(1));  % interpolate steady resistance
for k=2:N-1
    d_square=(x(k)-LAT).^2+(y(k)-LON).^2;
    closest=find(d_square==min(d_square)); %row index of closest point
    closest=closest(1); % keep first entry if more than one

    % get wave and swell data at this point
    Hwave=wave(closest,3,k-1);
    theta_wave=wave(closest,4,k-1);
    freq_wave=wave(closest,5,k-1);
    Hswell=swell(closest,3,k-1);
    theta_swell=swell(closest,4,k-1);
    freq_swell=swell(closest,5,k-1);
    % calculate wave direction relative to ship's course;
    % the result should be port-starboard symmetric
    relative_wave_direction=abs(p(1)-theta_wave);
    if relative_wave_direction>pi
        relative_wave_direction=2*pi-relative_wave_direction;
    end
    relative_wave_direction=abs(relative_wave_direction-pi);
    relative_swell_direction=abs(p(1)-theta_swell);
    if relative_swell_direction>pi
        relative_swell_direction=2*pi-relative_swell_direction;
    end
    relative_swell_direction=abs(relative_swell_direction-pi);
    % interpolate to get added resistance operator as a function
    % of frequency omega
    aro_wave=interpn(W,U,C,ARO,omega,u(1),relative_wave_direction);
    aro_swell=interpn(W,U,C,ARO,omega,u(1),relative_swell_direction);
    % calculate two separate Bretschneider spectra for wave and swell
    WaveSpectr=0.278*freq_wave^4./omega.^5.*(Hwave/1.6)^2.*
exp(-0.437*(freq_wave./omega).^4);
```

```
    SwellSpectr=0.278*freq_swell^4./omega.^5.*(Hswell/1.6)^2.*
exp(-0.437*(freq_swell./omega).^4);
    % integrate to get mean added resistance
    Rwave=2*trapz(omega,WaveSpectr.*aro_wave);
    Rswell=2*trapz(omega,SwellSpectr.*aro_swell);
    Rtotal=Rwave+Rswell+Rcalm;
    RAO1_w=interpn(W,U,C,RAO_r1,omega,u(1),relative_wave_direction);
    RAO2_w=interpn(W,U,C,RAO_r2,omega,u(1),relative_wave_direction);
    RAO3_w=interpn(W,U,C,RAO_v,omega,u(1),relative_wave_direction);
    RAO1_s=interpn(W,U,C,RAO_r1,omega,u(1),relative_swell_direction);
    RAO2_s=interpn(W,U,C,RAO_r2,omega,u(1),relative_swell_direction);
    RAO3_s=interpn(W,U,C,RAO_v,omega,u(1),relative_swell_direction);
    m0_r1=trapz(omega,(WaveSpectr.*RAO1_w.^2+SwellSpectr.*RAO1_s.^2));
    m0_r2=trapz(omega,(WaveSpectr.*RAO2_w.^2+SwellSpectr.*RAO2_s.^2));
    m0_v=trapz(omega,(WaveSpectr.*RAO3_w.^2+SwellSpectr.*RAO3_s.^2));
    Prob1=exp(-f^2./(2*m0_r1)); % probability of deck wetness (i.e.
freeboard exceedance)
    Prob2=exp(-(T^2./(2*m0_r2)+Vcr^2./(2*m0_v))); % probability of
slamming
    % check if constraints are met; if not record how many times they
are
    % violated
    if Prob1>Limit_dw||Prob2>Limit_sl
        counter(k)=1;
    end
    % extract data from propeller curves
    Kt_coef1=interp2(RR,UU,KT,Rtotal,u(1),'cubic'); % thrust
coefficient
    Jcoef1=interp2(RR,UU,JJ,Rtotal,u(1),'cubic');   % advance ratio
    Kq_coef1=pchip(J1,KQ,Jcoef1);   % torque coefficient
    nO=Kt_coef1*Jcoef1/Kq_coef1/(2*pi);  % open water efficiency
    nD=nH*nO*nR;   % propulsive efficiency
    q1=sfc*Rtotal*u(1)/nD/nTRM;   % fuel consumption rate
    I(k)=I(k-1)+q1*dt;
end
Rcalm=pchip(speed,Rt,u(end));
Kt_coef2=interp2(RR,UU,KT,Rcalm,u(end),'cubic'); % thrust coefficient
Jcoef2=interp2(RR,UU,JJ,Rcalm,u(end),'cubic');   % advance ratio
Kq_coef2=pchip(J1,KQ,Jcoef2);   % torque coefficient
nO=Kt_coef2*Jcoef2/Kq_coef2/(2*pi);  % open water efficiency
nD=nH*nO*nR;   % propulsive efficiency
q2=sfc*Rcalm*u(end)/nD/nTRM;   % fuel consumption rate
I(N)=I(N-1)+q2*dt1 % add last part of trip when sea is assumed calm
Icalm=q2*VoyageHours*3600;  % equivalent calm water fuel consumption of
entire trip

% Calculate fraction of time in rough seas when constraints are
violated
T_unsafe=sum(counter)*dt;
fraction=T_unsafe/tt(end)
```

```
% ExtractRAO.m
% Auxilliary program used to extract the magnitude and phase of the
RAO's given by SWAN1.
% Opens subdirectories labeled by the ship's speed and loads the data
files for each heading.
% Stores the appended information in a 3-D array for each mode of
% motion. File should be located and run in the folder containing the
% speed-labeled subfolders.
% Note that SWAN1 records the response operators at the encounter wave
periods, denoted below as Twave.
% The responses in SWAN1 are given in [m/m] for mode 3, [deg/m] for
modes 4,5 and [kN/m^2] for added resistance operator.

%clear all

g=9.81; % gravitational acceleration
V=7:16; % vector of ship speeds in knots
c=[0 45 90 135 180]; % vector of relative wave direction in degrees
N=length(V);
M=length(c);

% Remote Locations:
x1=50.8; y1=0; z1=2.85; % for deck wetness
x2=40.64; y2=0; z2=-5.7; % for slamming and bow acceleration
x3=0; y3=0; z3=10.0;   % for roll
% Note: since all modes of motion are that of a rigid body, only the x-
% coordinate of the remote locations enters the calculation of RAO's

T=transpose(28:-1:1); % absolute period of waves
w=2*pi./T; % absolute frequency
k=w.^2/g;  % dispersion relation for deep water

for n=1:N
    cd(num2str(V(n))) % go to respective folder for each speed
    U=0.51444*V(n);  % convert speed to m/sec
    for m=1:M
        switch m
            case 1
                load c0
            case 2
                load c45
            case 3
                load c90
            case 4
                load c135
            case 5
                load c180
        end
        beta=c(m)*pi/180; % convert relative wave direction to radians

        % pull out the respective columns from the files created after
importing
        % SWAN1 output
        Twave=AddRes(:,3); % encounter frequency from SWAN1
        add_res_oper=AddRes(:,5);
```

```
        w_e=2*pi./Twave; % encounter frequencies at which output has
been provided from SWAN1
        we=abs(w-U*k*cos(beta)); % encounter frequencies at which
response will be stored; note that through this relation a "1-1"
correspondence is established between we and w and the spectral
integration will later be performed over w to compute the statistics

        % magnitude and phase of modes of motion
        heave_mag=LongModes(:,6);
        X3=interp1(Twave,heave_mag,T,'pchip',0);
        heave_pha=LongModes(:,7)*pi/180;
        arg3=interp1(Twave,heave_pha,T,'pchip',0);
        pitch_mag=LongModes(:,8)*pi/180;
        X5=interp1(Twave,pitch_mag,T,'pchip',0);
        pitch_pha=LongModes(:,9)*pi/180;
        arg5=interp1(Twave,pitch_pha,T,'pchip',0);
        if m==2 | m==3 | m==4
            roll_mag=TransModes(:,6)*pi/180;
            X4=interp1(Twave,roll_mag,T,'pchip',0);
            roll_pha=TransModes(:,7)*pi/180;
            arg4=interp1(Twave,roll_pha,T,'pchip',0);
        else
            X4=zeros(size(we));
            arg4=zeros(size(we));
        end
        aro=interp1(Twave,add_res_oper,T,'pchip',0);
        ARO(:,n,m)=aro; % added resistance operator

        % transfer functions of composite events:

         % relative wave elevation near the fore peak (for deck
wetness)
         Hr1(:,n,m)=X3.*exp(i*arg3)-x1*X5.*exp(i*arg5)-
exp(-i*k*(x1*cos(beta)+y1*sin(beta)));

         % relative wave elevation at the keel near the bow (for
slamming)
         Hr2(:,n,m)=X3.*exp(i*arg3)-x2*X5.*exp(i*arg5)-
exp(-i*k*(x2*cos(beta)+y2*sin(beta)));

         % relative velocity of the keel near the bow (for slamming)
         Hv(:,n,m)=i*w.*Hr2(:,n,m);

         % absolute bow acceleration
         Ha(:,n,m)=-w.^2.*(X3.*exp(i*arg3)-x2*X5.*exp(i*arg5));

         % absolute roll motion
         Hroll(:,n,m)=X4.*exp(i*arg3);

         % roll acceleration
         Hroll_a(:,n,m)=-w.^2*z3.*Hroll(:,n,m);
    end
    cd ..
end
```

```
ARO(ARO<=-10)=0; % eliminate large negative values (as a result of
interpolation errors)

% store the necessary vectors and arrays; use only the magnitude of
RAO's
omega=w; V=0.51444*V; c=pi*c/180;
[W,U,C]=ndgrid(omega,V,c);
RAO_r1=abs(Hr1); RAO_v=abs(Hv); RAO_r2=abs(Hr2); RAO_a=abs(Ha);
RAO_roll=abs(Hroll); RAO_roll_acc=abs(Hroll_a);
save TransferFunctions RAO_r1 ARO omega RAO_r2 RAO_v RAO_a RAO_roll
RAO_roll_acc W U C
```

```
% wave_data.m
% Script to extract ASCII data from WAM output and convert it to
% MATLAB compatible format. The program reads the text files, scans
% out the useful information and reduces the size to contain only
% North Atlantic data. It also adjusts the content to be used in the
% optimization code.

L=137099; % number of data rows in the WAM4 ASCII output
WAVE=zeros(L,7,36);
SWELL=zeros(L,5,36);
str='WOL0803'; % part of the ASCII file name: year 08, month 03
day1=repmat('19',7,1);
day2=repmat('20',8,1);
day3=repmat('21',8,1);
day4=repmat('22',8,1);
day5=repmat('23',5,1);
day=vertcat(day1,day2,day3,day4,day5);
time=['03'; '06';'09'; '12'; '15'; '18'; '21';'00'];
t=repmat(time,5,1);
t=t(1:end-4,:);
cd('18 MAR')
for k=1:36

fid1=fopen(strcat('C:\Documents and Settings\user\My
Documents\KYRIAKOS\2.THG\wave data files\18
MAR\',str,day(k,:),t(k,:),'.W.dat'),'r');
fid2=fopen(strcat('C:\Documents and Settings\user\My
Documents\KYRIAKOS\2.THG\wave data files\18
MAR\',str,day(k,:),t(k,:),'.S.dat'),'r');
for i=1:12
    l1=fgetl(fid1);
    l2=fgetl(fid2);
end
for i=1:2
    l1=fgetl(fid1);
end
A1=textscan(fid1,'%f');
A2=textscan(fid2,'%f');
fclose(fid1);
fclose(fid2);
B1=reshape(A1{1},7,L);
B2=reshape(A2{1},5,L);
A1=B1'; A2=B2';
WAVE(:,:,k)=A1;
SWELL(:,:,k)=A2;
end


% filter longitudes and latitudes of North Atlantic
long=WAVE(:,1,1);
lat=WAVE(:,2,1);
index=find((long<=10|long>=280)&(lat<=50&lat>=30));

% store data in smaller arrays
wav=WAVE(index,:,:);
swe=SWELL(index,:,:);
```

```
% extract long-lat columns (in degrees)
long=swe(:,1,1);
lat=swe(:,2,1);
cd ..
% Use Mercator projection to extract planar coordinates (in m)
[X,Y]=MercProj(long,lat);
wave=wav(:,[1:4,7],:); % remove unused columns (max WHT and peak FREQ)
swell=swe;

% rearrange wave data
for i=1:36
    wave(:,1,i)=Y; % column of longitudes
    wave(:,2,i)=X; % column of latitudes
    wave(:,4,i)=wave(:,4,i)*pi/180; % convert mean direction to rad
    wave(:,5,i)=wave(:,5,i)*2*pi; % convert mean frequency to rad/sec
    swell(:,1,i)=Y;
    swell(:,2,i)=X;
    swell(:,4,i)=swell(:,4,i)*pi/180;
    swell(:,5,i)=swell(:,5,i)*2*pi;
end

save NorthAtlantic18Mar08 wave swell
```

```
% ResistanceCalc.m
% Script to calculate total calm water resistance.
% ref: PNA vol.II
% Calculation of calm water ship resistance.
% Friction drag: from ITTC lines
% Form drag: from Holtrop regression model;
% Wave drag: from SWAN2
% Final result in kN. All units in SI (water density in [tonne/m^3])
% Output is stored in CalmRes.mat

% properties of salt water at 19 deg C
niu=1.07854e-6;
rho=1.025;

load WaveRes % loads Cw,WetSurf,Vol and V from SWAN2 converged runs
S=WetSurf;
load ship_particulars % loads L,T,B,Cp,Sbk
Re=V*L/niu;
Fn=V/sqrt(g*L);
% Holtrop calculation of form factor k and allowance coefficient Ca
Ca=0.006*(L+100)^(-0.16)-0.00205;
K2_bk=1.4; % bilge keel form factor K2=1+k2=1.4
Lr_over_L=1-Cp+0.06*Cp*LCB/(4*Cp-1);  %LCB enters as % of LWL: (+) fwd,
(-) aft
K1=0.93+0.4871*((B/L)^1.0681)*((T/L)^0.4611)*((1/Lr_over_L)^0.1216).*(L
^3./Vol).^0.3649*(1-Cp)^(-0.6042);
K2=K2_bk;
K=K1+(K2-K1)*Sbk./(S+Sbk); % correction for bilge keels
% Friction drag coefficient from ITTC lines;
Cf=0.075./((log(Re)-2).^2);

% Total drag coefficient
Ct=K.*Cf+Cwave+Ca;
% Viscous drag coefficient (including appendage and model-ship
correlation)
Cvisc=K.*Cf+Ca;
% Total drag
Rt=.5*rho*Ct.*(S+Sbk).*V.^2;
% Viscous drag
Rv=.5*rho*Cvisc.*(S+Sbk).*V.^2;
% Wave drag
Rw=.5*rho*Cwave.*S.*V.^2;
hold on
plot(Fn,Rv,'-o')
plot(Fn,Rw,'-g^','MarkerFaceColor','g','MarkerEdgeColor','k')
plot(Fn,Rt,'--md','MarkerFaceColor','r','MarkerEdgeColor','k')
legend('Viscous (Holtrop)','Wave (SWAN)','Total')
xlabel('Froude Number   U/\surdgL')
ylabel('Calm Water Resistance  (kN)')
title('Steady Resistance Components')
grid on

save CalmRes Rt Rv Rw
```

```
% Holtrop.m
% Estimation of total resistance based on Holtrop & Mennen (1982) and
Holtrop (1984).
% Wave resistance and the correlation allowance Ca are calculated using
% the regression coefficients of Holtrop-Mennen (1982). Form factors
% for the ship and bilge keels are estimated based on Holtrop (1984).

load ship_particulars
load WaveRes
load CalmRes
% properties of salt water at 19 deg C
niu=1.07854e-6;
rho=1.025;
Fn=V/sqrt(g*L); % Froude number range
W=rho*g*Vol; % ship weight in kN

%*** Wave making resistance Rw
lr=L^3./Vol;
if (B/L)<=0.11
    C4=-.2296*(B/L)^0.3333;
elseif (B/L)>0.11&(B/L)<0.25
    C4=B/L;
else
    C4=0.5-0.0625*(L/B);
end
d=-0.9;
if Cp<0.8
    C5=8.0798*Cp-13.8673*Cp^2+6.9844*Cp^3;
else
    C5=1.7301-0.7067*Cp;
end
m1=0.01404*L/T-1.7525*(Vol.^(1/3))/L-4.7932*B/L-C5;
if lr<=512
    C6=-1.69385;
elseif lr>512&lr<1727
    C6=1.69385+(L./Vol.^(1/3)-8)/2.36;
else
    C6=0;
end
m2=C6*0.4*exp(-0.034*Fn.^(-3.29));
if (L/B)<12
    lamda=1.446*Cp-0.03*(L/B);
else
    lamda=1.446*Cp-0.36;
end
C2=1;  %no bulb
C3=1;  % no transom
iE=125.67*(B/L)-162.25*Cp^2+234.32*Cp^3+0.1551*LCB^3;
C1=2223105*C4^3.7861*(T/B)^1.0796*(90-iE)^(-1.3757);

Rwave=W.*(C1*C2*C3.*exp(m1.*Fn.^d+m2.*cos(lamda*Fn.^(-2))));

%*** Viscous resistance (1+k1)*Rf
Cstern=0; % normal afterbody sections
c14=1+0.011*Cstern;
LR=L*(1-Cp+0.06*Cp*LCB/(4*Cp-1));
```

```
K1=0.93+0.487118*c14*((B/L)^1.06806)*((T/L)^0.46106)*((LR/L)^0.121563).
*(L^3./Vol).^0.36486*(1-Cp)^(-0.604247);
Re=V*L/niu;
Cf=0.075./(log(Re)-2).^2;
Rviscous=K1.*Cf.*0.5*rho.*S.*V.^2;

%*** Appendage Resistance (bilge keels only)
K2_bk=1.4;
Rapp=0.5*rho*V.^2*Sbk*K2_bk.*Cf;

%*** Model - Ship correlation
if (T/L)<=0.04
    Ca=0.006*(L+100)^(-0.16)-0.00205+0.003*sqrt(L/7.5)*Cb^4*C2*(0.04-
T/L);
else
    Ca=0.006*(L+100)^(-0.16)-0.00205;
end
Ra=0.5*rho*V.^2.*(S+Sbk)*Ca;

%*** Total Resistance
Rtotal=Rviscous+Rapp+Ra+Rwave;

% Comparison of Holtrop and SWAN wave resistance prediction
figure
hold on
plot(Fn,Rwave,'-o')
plot(Fn,Rw,'-g*')
legend('Holtrop-Mennen','SWAN')
xlabel('Froude Number  V/\surdgL')
ylabel('Wave Resistance  (kN)')
title('Prediction of Steady Wave Resistance')
grid on
```

```
% PropSolver.m
% Script to solve for the operating point of the propeller
% for any loading condition, using B series curves.
% Output is stored in a data file called PropData.mat.
clear all
rho=1.025;  % water density (in tonne/m^3)
D=3.2;      % propeller diameter in m
w=0.249;    % wake fraction
t=0.176;    % thrust deduction
EAR=0.75;   % effective area ratio
P_D=1.2;    % pith to diameter ratio
z=5;        % number of blades
nTRM=0.94;  % transmission efficiency (shaft & gearbox, i.e. nS*nG)
nH=1.0972;  % hull efficiency (1-t/1-w)
nR=1.035;   % relative rotating efficiency
U=0.51444*(7:16);  %speed range
R=1:5:700;  % range of resistance
options=optimset('Display','off');
for n=1:length(U)
    for m=1:length(R)
        x=fsolve(@(x) Ktsolve(x,U(n),EAR,P_D,z,rho,D,w,t,R(m)),[0
0],options);
        KT(n,m)=x(1);
        JJ(n,m)=x(2);
    end
end
J1=0:.01:1.333;
KQ=Kq(J1,EAR,P_D,z);
[RR,UU]=meshgrid(R,U);
save PropData KT KQ J1 JJ RR UU EAR w t D P_D z nH nR nTRM
```

```
function F=Ktsolve(x,U,EAR,P_D,z,rho,D,w,t,R)
% Function to define the nonlinear system:
%       Kt=f(J)     (from B-series curves)
%       Kt=c*J^2    (from loading curve)
% The vector x contains the two unknown scalars Kt, J.

F=[x(1)-Kt(x(2),EAR,P_D,z);x(1)-R/(1-t)/(rho*D^2*U^2*(1-w)^2)*x(2)^2];
```

```
function kq=Kq(J,EAR,P_D,z)
% Propeller torque coefficient from B-series curves (Troost regression
% coefficients in data file Kqcoef.dat)
load Kqcoef
for n=1:length(J)
    S(:,n)=b.*J(n).^sKq.*P_D.^tKq.*EAR.^uKq.*z.^vKq;
end
kq=sum(S);
kq(kq<0)=0;
```

```
function kt=Kt(J,EAR,P_D,z)
% Propeller thrust coefficient from B-series curves (Troost regression
% coefficients in data file Ktcoef.dat)
load Ktcoef
S=a.*J.^sKt.*P_D.^tKt.*EAR.^uKt.*z.^vKt;
kt=sum(S);
kt(kt<0)=0;
```

```
function [X,Y]=MercProj(lambda,phi)
% Function to convert spherical coordinates on the surface of the earth
% to plane coordinates using Mercator projection.
% The latitude scale is distorted as it approaches the poles.
% The origin is set at the intersection of Equator and Greenwich
meridian.
% Reference: Richardus and Adler, 1972
% INPUT: longitude and latitude in degrees
% OUTPUT: X,Y coordinates in meters

R=6372797.6; % mean radius of earth in meters
% longitude: zero at Greenwich meridian, increasing eastbound
ind=lambda>180;
lambda(ind)=lambda(ind)-360; % if lambda west, make it negative
Y=R*lambda*pi/180;

% latitude: zero at Equator, increasing northbound
X=R*log(tand(45+phi/2));
```