MODELING AND SIMULATION OF A MANEUVERING SHIP

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SİNAN PAKKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

MODELING AND SIMULATION OF A MANEUVERING SHIP

submitted by **SİNAN PAKKAN** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. S. Kemal İder _____
Head of Department, **Mechanical Engineering**

Prof. Dr. M. Kemal Özgören _____
Supervisor, **Mechanical Engineering Dept., METU**

Assoc. Prof. Dr. Halit Oğuztüzün _____
Co-Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit Soylu _____
Mechanical Engineering Dept., METU

Prof. Dr. M. Kemal Özgören _____
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Halit Oğuztüzün _____
Computer Engineering Dept., METU

Asst. Prof. Dr. Yiğit Yazıcıoğlu _____
Mechanical Engineering Dept., METU

Dr. Gökmen Mahmutyazıcıoğlu _____
TÜBİTAK-SAGE

Date: 06.09.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Sinan Pakkan

Signature :

ABSTRACT

MODELING AND SIMULATION
OF A MANEUVERING SHIP

Pakkan, Sinan
M.S., Department of Mechanical Engineering
Supervisor : Prof. Dr. M. Kemal Özgören
Co-Supervisor : Assoc. Prof. Dr. Halit Oğuztüzün

September 2007, 138 pages

This thesis documents the studies conducted in deriving a mathematical model representing the dynamics of a maneuvering ship to be implemented as part of an interactive real-time simulation system, as well as the details and results of the implementation process itself. Different effects on the dynamics of ship motions are discussed separately, meaning that the effects are considered to be applied to the system one at a time and they are included in the model simply by the principle of superposition. The model is intended to include the hydrodynamic interactions between the ship hull and the ocean via added mass (added inertia), damping and restoring force concepts. In addition to these effects, which are derived considering no incident waves are present on the ocean, the environmental disturbances, such as wind, wave and ocean current are also taken into account for proposing a mathematical model governing the dynamics of the ship. Since the ultimate product of

this thesis work is a running computer code that can be integrated into an available simulation software, the algorithm development and code implementation processes are also covered. Improvements made on the implementation to achieve “better” real-time performance are evaluated comparatively in reference to original runs conducted before the application of improvement under consideration. A new method to the computation of the wave model that allows faster calculation in real-time is presented. A modular programming approach is followed in the overall algorithm development process in order to make the integration of new program components into the software, such as a new hull or propulsion model or a different integrator type possible, easily and quickly.

Keywords: Ship dynamic modeling, real-time interactive simulation, hydrodynamic model, 6DOF equations of motion, added inertia

Öz

MANEVRA YAPAN BİR GEMİNİN
MODELLENMESİ VE BENZETİMİ

Pakkan, Sinan
Yüksek lisans, Makine Mühendisliği Bölümü
Tez yöneticisi : Prof. Dr. M. Kemal Özgören
Ortak tez yöneticisi : Assoc. Prof. Dr. Halit Oğuztüzün

Eylül 2007, 138 sayfa

Bu tezde, manevra yapan bir geminin dinamiğini temsil eden ve etkileşimli, gerçek-zamanlı bir benzetim sisteminin parçası olarak uygulamaya geçirilecek bir matematiksel modelin türetilmesi çalışmaları raporlanmıştır. Gemi hareketlerinin dinamiği üzerindeki farklı etkiler ayrı ayrı değerlendirilmiş, bir başka deyişle sisteme bir seferde bu etkilerden yalnız birinin uygulandığı kabul edilmiş ve bunlar süperpozisyon prensibi ile modele dahil edilmiştir. Modelin gemi gövdesi ile okyanus arasındaki hidrodinamik etkileşimi, eklenmiş kütle (eklenmiş eylemsizlik), sönümleyici ve denge sağlayıcı kuvvetler kavramları kullanılarak, içermesi amaçlanmıştır. Okyanusun tamamen dalgasız olduğu kabulü ile türetilmiş olan bu etkilere ek olarak, gemi dinamiğini açıklayan bir matematiksel modelin ortaya konmasında rüzgar, dalga ve okyanus akıntıları gibi çevresel etkiler de göz önünde bulundurulmuştur. Bu tez çalışmasının nihai ürünü mevcut bir benzetim yazılımına eklenebile-

cek, çalışan bir bilgisayar kodu olduğundan dolayı algoritma geliştirme ve kodlama sureçleri de ele alınmıştır. "Daha iyi" bir gerçek-zaman başarımının elde edilmesi için kodda yapılan iyileştirmeler, söz konusu iyileştirmenin uygulanmasından önce yapılan testlerin sonuçları referans alınarak karşılaştırmalı biçimde değerlendirilmiştir. Dalga modelinin gerçek zamanda daha hızlı bir biçimde hesaplanmasını mümkün kılan yeni bir yöntem ortaya konmuştur. Genel algoritma geliştirme sürecinde, yeni program bileşenlerinin yazılıma kolay ve hızlı bir şekilde dahil edilmesinin mümkün kılınması için modüler bir programlama yaklaşımı takip edilmiştir.

Anahtar kelimeler: Gemi dinamik modellemesi, gerçek-zamanlı etkileşimli benzetim, hidrodinamik model, 6 serbestlik derceli hareket denklemleri

To my companion, riding with me

in the chase of ever-running horizons,

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and sincerest thanks to my supervisor Prof. Dr. M. Kemal Özgören for his tolerance and help and guidance provided by him throughout this thesis study.

I would also like to express my deepest gratitude and sincerest thanks to my co-supervisor Assoc. Prof. Dr. Halit Oğuztüzün for his guidance, innovative ideas and motivation.

I would like to state my special thanks to Dr. Gökmen Mahmutyazıcıoğlu for his time and candid motivation, and my colleagues Koray Küçük, Alaettin Arda Özyüksel, Kürşat Karail, Selçuk Ataç and İlke Aydıncak for their ideas in technical issues and "all-the-time-all-the-way support".

Technical assistance and creative ideas provided by Canku Alp Çalargün and Derviş Emre Demirocak are greatly appreciated.

I would also like to thank my family, especially my grandmother, for their support, motivation, understanding and patience.

And last, but not least, I would like to state my greatest appreciation to Aysun Çelebi, without whose support, motivation and faith in me, this thesis work would have never been completed.

Support provided by TÜBİTAK-SAGE is also acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

$\mathcal{F}_O$: Earth-fixed reference frame

$\vec{x}$, $\vec{y}$, $\vec{z}$: Axes of the earth-fixed reference frame

O : Origin of the earth-fixed reference frame

$\mathcal{F}_b$: Body-fixed reference frame

$\vec{x}_b$, $\vec{y}_b$, $\vec{z}_b$: Axes of the body-fixed reference frame

B : Origin of the body-fixed reference frame

$\hat{\boldsymbol{C}}^{(O,b)}$: Orthonormal transformation matrix from $\mathcal{F}_O$ to $\mathcal{F}_b$

u, v, w : Translational velocities defined along the $\vec{x}_b$, $\vec{y}_b$, $\vec{z}_b$ axes of the body-fixed reference frame, respectively

p, q, r : Rotational velocities defined about the $\vec{x}_b$, $\vec{y}_b$, $\vec{z}_b$ axes of the body-fixed reference frame, respectively

$\boldsymbol{v}$: 6x1 body velocity variables vector

X, Y, Z : Forces applied along the $\vec{x}_b$, $\vec{y}_b$, $\vec{z}_b$ axes of the body-fixed reference frame, respectively

K, M, N : Moments applied about the $\vec{x}_b$, $\vec{y}_b$, $\vec{z}_b$ axes of the body-fixed reference frame, respectively

$\boldsymbol{\tau}$: 6x1 force variables vector

x, y, z : Earth-fixed coordinates of point B

ϕ, θ, ψ : Euler angles defining the rotation from $\mathcal{F}_O$ to $\mathcal{F}_b$

$\boldsymbol{\eta}$: 6x1 position and orientation variables vector

$I_{xx,shaft}$: Axial moment of inertia of the shaft-rotor assembly of the propulsion system

n : Actual shaft speed

n_c : Commanded shaft speed

δ : Rudder deflection

δ_c : Commanded rudder deflection

$\boldsymbol{M}_{RB}$: Rigid-body inertia matrix

$\boldsymbol{C}_{RB}(\boldsymbol{v})$: Rigid-body Coriolis and centripetal effects matrix

$\boldsymbol{\tau}_{RB}$: 6x1 rigid-body force variables vector

CHAPTER 1

INTRODUCTION

1.1. Background Information

1.1.1. Ocean Vehicle Simulators

Real-time vehicle simulation systems appear in a vast field of usage, where the applications vary from computer games of car or flight simulations for entertainment purposes to specialized hi-tech simulators incorporating both hardware (such as a physical control cabinet and control panels of the real systems) and software used in the tactical training of aviators or naval officers.

Whether a simple commercial computer game or a million-dollar military tactical simulator is considered, the core of the system that determines the quality of operation by maintaining the conformance to physical reality is the software and the mathematical model implemented to create this software. Therefore, the main goal in the process of building such a system is generating a computer code that simulates the real world effects and the operation of the vehicle with reasonable accuracy inside the limitations imposed by computation power, test facilities to gather information, etc. available.

In nautical applications, simulators may be used for different purposes such as determining the overall performance of a ship in the design stage or after a modification is made on the hull, propulsion system, etc., or in the tactical training of a junior bridge officer [20] before going into a real mission.

There are institutions using operational examples of these simulators for educational service such as Centre for Marine Simulation (CMS) at the Marine Institute of Memorial University of Newfoundland. Full mission ship bridge simulator facility at CMS offers a navigation simulation with unlimited number of mathematical models. Figure 1 shows this simulation system of CMS.

Figure 1 – Full mission ship bridge simulator facility at CMS

The Integrated Marine Simulator (IMS) of the Australian Maritime College includes a shiphandling simulator, which is mainly used for "familiarizing ships' officers and marine pilots with new ships and

new ship types" [3]. A snapshot of the port simulation by this software can be viewed in Figure 2.

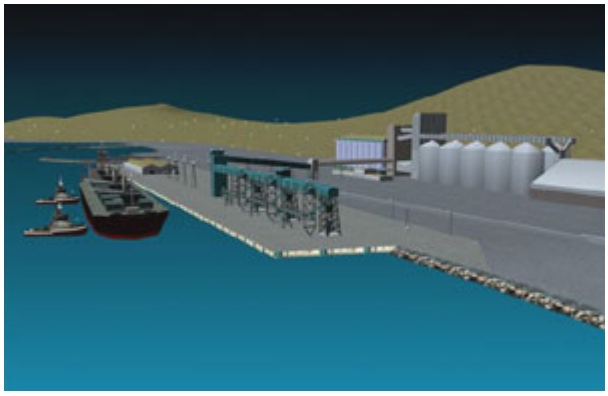

Figure 2 – Albany port simulation by IMS

Visualization issues such as ocean surface visual effects play an important role on the degree of realism of the simulation. There are specialized commercial software packages such as Vega Prime Marine of Multigen-Paradigm Inc., which are used to simulate realistic ocean surfaces in the presence of waves and interaction of ocean vehicles [9]. The application of Vega Prime Marine to a 3D simulator can be seen in Figure 3.

Figure 3 – Vega Prime Marine ocean surface simulator

An application of this software is available in the study of Sicuro [15]. The wave effects are incorporated into the simulation code by the use of Vega Prime Marine module.

1.1.2. Mathematical Models

Several studies are conducted in the field of hydrodynamic modeling of ocean vehicles in the second half of the last century. One of the most important resources published in this context is the Lectures on Ship Hydrodynamics by Martin A. Abkowitz [1], where the Taylor series expansion of the hull forces, which is one of the most basic concepts in hydrodynamic modeling of ocean vehicles, is defined comprehensively.

The technical report published by the Society of Naval Architects and Marine Engineers in 1950 constitutes a baseline in terms of nomenclature, which is used in most of the studies published in the field worldwide.

The researches accomplished on the subject of modeling surface ocean vehicles are mainly focused on 2D dynamic modeling in 3 degrees of freedom, which are surge, sway and yaw modes of the vehicle. Furthermore several simplifications on the models are employed for linearization purposes. The reason for this situation is that the major interest in deriving a dynamic model of a surface vessel arises from the need of analyzing the planar motion performance of it, and in a later stage designing a controller for steering actions. [5] presents the 3DOF mathematical model and related force coefficients of four ships (tanker, series 60, ferry, container)

showing different characteristics in a tabulated form. Since the hydrodynamic coefficients are in non-dimensional form, comparison of their effects on the characteristics of the ship is possible.

Another non-dimensional model for a Mariner class vessel is available in [7]. What is different in this model is the definition of the forward velocity. Actually, all states are defined as perturbations around the nominal values of velocities and angles. However, since these nominal values for general operation of the ship are zero except the one for the forward velocity, the states may be counted as the change in forward velocity, and the other velocities and angles themselves. It should also be noted that even if the surge velocity itself is used in the model as a state variable, the model is valid only in the proximity of a nominal velocity where the hydrodynamic coefficients are obtained due to the highly nonlinear nature of the hydrodynamic forces.

This fact gives rise to a more simplified approach in handling the ship steering equations of motion where the surge velocity is introduced to the system as a system parameter rather than a state variable, and the model becomes 2DOF with the state variables being the sway and the yaw velocities. This model proposed by Davidson and Schiff in 1946 [7], and the governing equations of motion can be given as

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{N}\left(u_0\right)\boldsymbol{v} = \boldsymbol{b}\delta_R$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{N}\left(u_0\right)$ is the matrix containing the linear damping terms and the Coriolis and centripetal effect terms,

which are functions of the forward velocity parameter u_0, and δ_R is the rudder deflection that is the input of the system. Note that great interest has been shown to the steering problem of ships and many models are derived dealing with this problem. Nomoto's first and second order models (1957), which are transfer functions relating the rudder deflection to the yaw angle of the ship, and Norrbin's nonlinear model (1963) based on the Nomoto's first order model are examples to these. More information on these models and similar other models can be found in [21].

Roll motion of the vessel is a critical mode, and it is highly coupled with the planar degrees of freedom, therefore an extension to the 2D, 3DOF models are made and 4DOF models may also be found in the literature. An example to such a model is the one presented by Son and Nomoto in 1981 [7] . The model explains the motion of the ship in surge, sway, roll and yaw degrees of freedom. The simplification that is mentioned in the previous paragraph is also applied to this model to eliminate the surge degree of freedom, which is in fact decoupled from the rest of the system in its original form, by taking it as a system parameter. Further by linearizing this model, two course-keeping models, one being nonlinear and the other linear, are derived from the initial 4DOF model of a container ship.
Hydrodynamic data of such a 4DOF model for a Multipurpose Naval Vessel is found in [12]. The hydrodynamic coefficients of the model differ from those of Son and Nomoto in being dimensional, which is preferable for the simulation of a specific vessel.

Fossen presents (1994) the implementations of the abovementioned 3DOF Mariner class vessel model and the 4DOF model of

Son and Nomoto written as a Matlab® function. Sicuro has used [15] the latter implementation in developing a real-time simulator written in C++.

It is worth mentioning that although the most general theoretical derivations in the literature are built up to cover all 6 degrees of freedom of a body translating and rotating in 3 dimensional space, the hydrodynamic data of a specific ship which contains the hydrodynamic derivatives, damping coefficients and added mass coefficients in all possible modes of motion cannot be found easily. That is to say, the available models, which are derived through scale model or full-scale ship trials or empirical methods such as strip theory, are found only up to the fourth degree of freedom excluding the heave and pitch modes of motion.

1.2. Objective and the Scope of the Thesis

The importance of and necessity for a naval tactical simulation software has been put forward in the preceding section. The objective of this thesis is developing an algorithm to solve the equations that govern the motion of a ship not only in real-time but also in the presence of continuous user inputs.

There are two important aspects to be considered in developing such an algorithm. First of these aspects is the conformance with the physical reality. The simulation as a concept, after all, is imitating the real world; therefore it should give results as close to real phenomena as possible. The second requirement to be fulfilled by the algorithm is real-time constraint. A well performing simulator should meet these two conflicting criteria. As the accuracy of the

results increase, the computation load imposed by the equation solver algorithm increase as well. If the computation time required by one cycle of the simulation loop exceeds the simulated time step in that cycle, the results obtained by the solution are useless no matter how accurate they are.

The computer code, which is the ultimate outcome of this study, is developed in Matlab® for reasons explained in section 4.1. This code will be integrated into a readily available simulation software written in a different programming language, C++, at a later stage. Therefore, in order to minimize the effort at this step, the product code of this thesis is kept as modular and easily maintainable as possible.

The mathematical models found in the literature are used as sample data for the evaluation of the code. The modularity of the solution algorithm lets the user update the program according to a different kind of ship whose data is provided to the system.

1.3. Thesis Outline

This section will shortly describe the organization of this thesis.
In this first chapter, a literature survey is presented in two subsections where the applications of naval simulators and the mathematical models are dealt with separately. Then the motivation for this thesis and the outline of the overall study are given respectively.
Chapter 2 includes brief information about the simulation system NSTMSS into which the algorithm developed as the product of this thesis will be integrated.

Following these preliminary information, the theoretical background on the subject is given. The problem formulation, which covers mainly the derivation of the equations of motion, is explained in the 3rd chapter. First of all the notation and the reference frames used for the derivation process are defined. After the equations of motion for a general rigid body are derived, the nature of the different force terms are studied and the models governing these terms, such as propulsion, ocean current or wave models are presented.

The next chapter deals with the implementation issues of this theoretical model into a computer code. The algorithm developed is investigated in two subsections as the initialization routine and the simulation loop. After the solution method of the motion equations is described, the final section of the chapter explains how the algorithm interacts with the main software.

The simulation tests and their results are handled in the 5th chapter. First the subcomponents and general issues related to the performance of the software are discussed. Then the overall performance is evaluated by running maneuvering tests on the whole of the algorithm.

In the final chapter, the results obtained are discussed, and future work to be accomplished is proposed.

CHAPTER 2

THE SIMULATION SOFTWARE NSTMSS

As mentioned in the introduction chapter, the implementation of the ship model as a computer code will be embedded into a simulation system named Naval Surface Tactical Maneuvering Simulation System, which is abbreviated as NSTMSS. NSTMSS is a "High Level Architecture (HLA) based distributed simulation system" [20].

2.1. What is HLA?

In order to understand NSTMSS, we should first give the definition of and some brief information about the HLA concept. HLA is a "general purpose architecture for distributed computer simulation systems. Using HLA, computer simulations can communicate to other computer simulations regardless of the computing platforms. Communication between simulations is managed by a runtime infrastructure (RTI)" [22].

Several simulations may run on different platforms, and they may communicate and interact with each other by RTI in run time. They may simply be included in the virtual simulation environment, which is called the *federation*, at runtime while the other simulators

are already running by themselves, communicating with, affecting and being affected by the others present in the environment.

The participant simulators in the HLA are called the federates. Some of the examples of NSTMSS federates are Meko federate (MekoFd), helicopter federate (HeliFd), environment federate (EnviFd), and tactical picture federate (TacPicFd) [20].

2.2. The Frigate Federates

Meko federate, into which the simulation code developed in this thesis will be integrated, is a platform level simulation, meaning that the main view of the simulation is given from the bridge of the ship letting the user interact with the simulation by getting visual feedback from the screen and giving user inputs such as shaft speed(s) and rudder angle or ship heading in 3 dimensional space. The name of the federate comes from the Meko class frigates, which are drawn into the federation by this federate. A second type of frigate federate is the KnoxFd, which is similar to MekoFd, but implements Knox class frigates.

The user may order rudder angle or ship heading and shaft speed, view the position of the ship in terms of latitude and longitude coordinates, view the echoes of the other ships in the federation in the radar display, view the environment thru the bridge windows and communicate with the users of the other ships, when using the simulator brought into the virtual environment by the Meko federate. The screenshot of the Meko federate user interface can be seen in Figure 4.

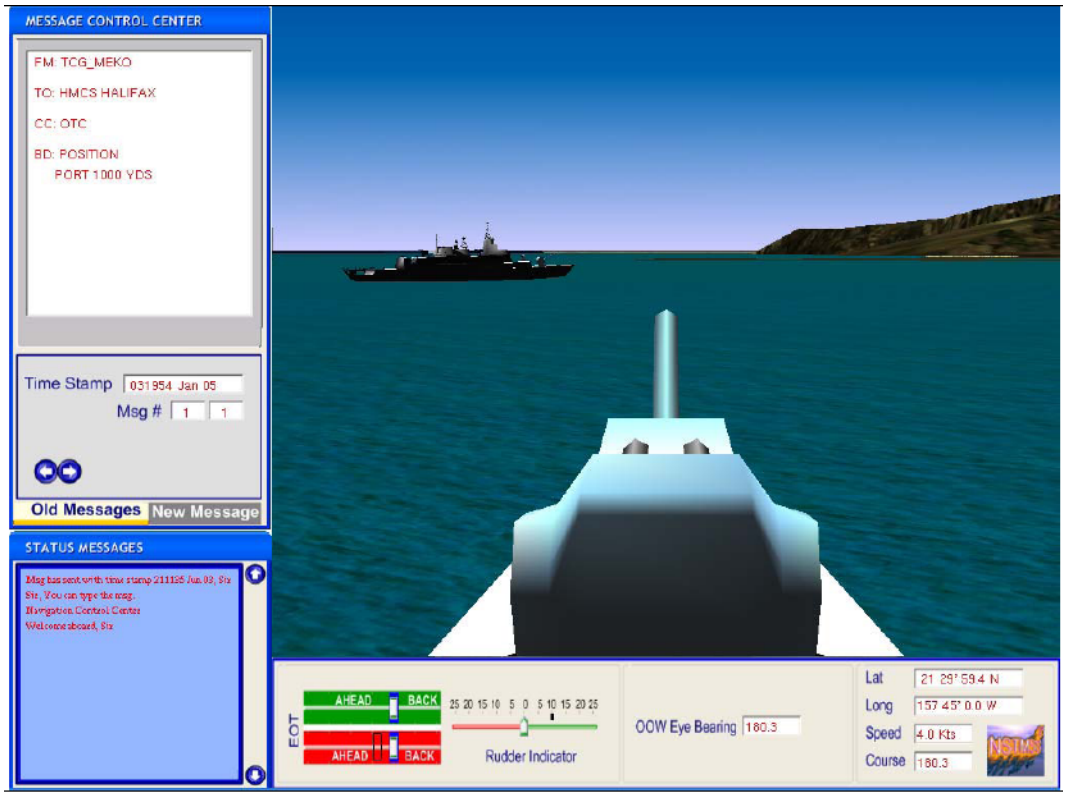

Figure 4 – User interface of the Meko federate

In the current design of NSTMSS, the interest of training is on formation maneuvering rather than ship handling. Environmental disturbances are of secondary importance in such a case, therefore an assumption is made for the environmental conditions so that the ships are sailing in still sea where no wind, wave or ocean current effects are present.

CHAPTER 3

PROBLEM FORMULATION

This chapter explains the development of the mathematical model of a generic ship that is to be used in the simulation software (NSTMSS), whose present configuration for simulating the motion of a frigate is given in the previous chapter. The notation and important definitions, which the mathematical model is based on, are given first before going into the details of the modeling process. The next step is the derivation of equations of motion. Forces acting on the ship, such as hydrodynamic hull forces, propulsion and steering inputs and environmental disturbances, are described in the sections following this general derivation step.

3.1. Reference Frame Definitions and Notation

3.1.1. Reference Frame Definitions

Figure 5, as adapted from [2], shows the orientation of a non-inertial reference frame, $\mathcal{F}_b$, that is fixed to the ship with its origin B preferably but not necessarily at its center of gravity. It also describes the names of the motions (surge, sway, heave; roll, pitch, yaw along and about the axes of $\mathcal{F}_b$), the symbols used to describe

these motions, i.e. the Euler angles (ϕ, θ, ψ) and the linear (u, v, w) and angular (p, q, r) velocity components, and the symbols denoting the force (X, Y, Z) and moment (K, M, N) components.

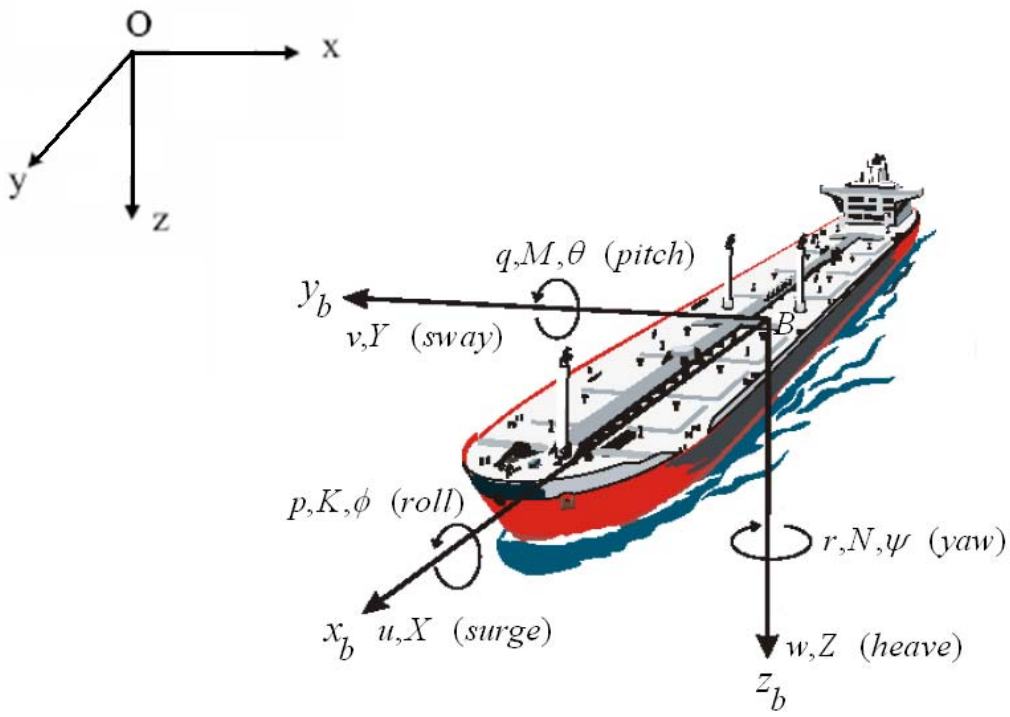

Figure 5 - Reference frames and motion variables

Figure 5 shows the most common naming convention for the forces, velocities etc. used in nautical applications [17]. The names of the motions are given in parentheses next to axes for translational motions and elliptic arrows for rotational ones. To the left of these names are the related motion variables, which are body-fixed velocities, forces, and Euler angles, if applicable. These quantities are given in a tabulated form in Table 1.

Table 1 – Standard symbols assigned for motion variables

	Direction	Name of the Motion	Force or Moment	Velocity		Definition Axis/ Frame	Position/ Orientation Variable
Translational	$\vec{x}_b$	surge	X	u		$\vec{x}$	x
	$\vec{y}_b$	sway	Y	v		$\vec{y}$	y
	$\vec{z}_b$	heave	Z	w		$\vec{z}$	z
Rotational	$\vec{x}_b$	roll	K	p		$\mathcal{F}_O \rightarrow \mathcal{F}_b$	ϕ
	$\vec{y}_b$	pitch	M	q			θ
	$\vec{z}_b$	yaw	N	r			ψ

Note that the left block of Table 1, including the forces acting on and velocities of the ship, takes the body-fixed reference frame $\mathcal{F}_b = \mathcal{F}_b\left(B, \vec{i}_b, \vec{j}_b, \vec{k}_b\right)$ as the frame of resolution. The coordinates of *B* (x, y, z) are defined in the inertial reference frame $\mathcal{F}_O = \mathcal{F}_O\left(O, \vec{i}_o, \vec{j}_o, \vec{k}_o\right)$, whereas the Euler angles, which will be explained in section 3.2.1 express the orientation of $\mathcal{F}_b$ with respect to $\mathcal{F}_O$ according to the z-y-x sequence.

3.1.2. Vectors of Motion Variables

The equations describing the rigid body motion of the ship fully in 3 dimensions are going to be given in the matrix form. The vectorial forms of the velocity, force, and position and orientation variables, as used in the equations of motion, are given as follows:

$\boldsymbol{\nu} \triangleq \left[\boldsymbol{\nu}_1^T \quad \boldsymbol{\nu}_2^T\right]^T$, where $\boldsymbol{\nu}_1 \triangleq \left[u \quad v \quad w\right]^T$ and $\boldsymbol{\nu}_2 \triangleq \left[p \quad q \quad r\right]^T$

$\boldsymbol{\tau} \triangleq \left[\boldsymbol{\tau}_1^T \quad \boldsymbol{\tau}_2^T\right]^T$, where $\boldsymbol{\tau}_1 \triangleq \left[X \quad Y \quad Z\right]^T$ and $\boldsymbol{\tau}_2 \triangleq \left[K \quad M \quad N\right]^T$

$\boldsymbol{\eta} \triangleq \left[\boldsymbol{\eta}_1^T \quad \boldsymbol{\eta}_2^T\right]^T$, where $\boldsymbol{\eta}_1 \triangleq \left[x \quad y \quad z\right]^T$ and $\boldsymbol{\eta}_2 \triangleq \left[\phi \quad \theta \quad \psi\right]^T$

Note that for the rest of the text, bold and italic lower case letters represent vectors as column matrices in appropriate reference frames and bold and upper case letters represent square matrices. It can easily be seen in the above definition that, vectors subscripted by "1" contain the variables related to translational quantities and those having the subscript "2" contain the variables related to rotational quantities.

3.2. Equations of Motion

Newtonian Mechanics is used in deriving the differential equations governing the motion of the ship in 3-dimensional space. For this purpose a 6 degree-of-freedom (DOF) mathematical model, consisting of 3 translational and 3 rotational DOFs, is put forward to include all possible rigid body motions of a single body.

First the uses of reference frames and velocity transformations used in the model are explained in the "Kinematics" section. Following this part is the "Kinetics" section in which the motion equations based on Newton's Second Law are given. After a dynamic model is introduced for a general body, the rest of the chapter discusses the factors such as the environmental effects, propulsion and control efforts, etc., particularly related to the dynamics of a

ship and explains how the equations are re-arrange so as to be used in the simulation program.

3.2.1. Kinematics

First of the two reference frames used in deriving the equations of motion, as shown in Figure 5 is the North-East-Down (NED) oriented, earth-fixed reference frame, $\mathcal{F}_O = \mathcal{F}_O\left(O, \vec{i}_o, \vec{j}_o, \vec{k}_o\right)$. This frame can be considered to be inertial for slowly moving marine vehicles.

As mentioned above a second, body-fixed reference frame is required to derive the equations of motion in 6 degrees of freedom, which is handled in the forthcoming section, "Kinetics". The main purpose of defining a second reference frame, rather than dealing with the motion variables all defined in a single reference frame is that the inertia tensor of the ship appears as a constant matrix in the body-fixed reference frame. The body-fixed frame, $\mathcal{F}_b = \mathcal{F}_b\left(B, \vec{i}_b, \vec{j}_b, \vec{k}_b\right)$, is fixed to the vessel at point B such that its x-axis ($\vec{x}_b$) points towards the bow, y-axis ($\vec{y}_b$) points towards the starboard of the ship, and z-axis ($\vec{z}_b$) is oriented downwards. The xz-plane is practically the plane of symmetry for the ship.

Since the velocities of the ship, which constitute the first half of the system states, are defined in $\mathcal{F}_b$, and $\mathcal{F}_b$ is rotating with respect to the earth-fixed reference frame, a component transformation from $\mathcal{F}_b$ to $\mathcal{F}_O$ is required. If the rotation that transforms $\mathcal{F}_O$ into $\mathcal{F}_b$ can be expressed, then the required component transformation can al-

so be achieved using the relevant rotation matrix. Two most commonly encountered methods to express the rotation of one reference frame with respect to another are angle-axis representation, and Euler angles [7], which are briefly explained below.

Angle-Axis Representation of Rotations

Letting $\vec{n}$ be the unit vector along the axis of rotation, θ be the angle of rotation, and $\begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}^T$ be the column representation of $\vec{n}$ in the fixed reference frame, the rotation defined by the couple $(\vec{n}, \theta)$ is represented using one of the two alternatives, which are Euler parameters and Euler-Rodrigues parameters.

(i) Euler Parameters as the elements of a rotation quaternion are defined as

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ \eta \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \eta \end{bmatrix} = \begin{bmatrix} n_1 \sin\left(\frac{\theta}{2}\right) \\ n_2 \sin\left(\frac{\theta}{2}\right) \\ n_3 \sin\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

(ii) Euler-Rodrigues Parameters as the elements of a 3×1 matrix are defined as

$$\boldsymbol{\rho} = \frac{1}{\eta}\boldsymbol{\varepsilon} = \begin{bmatrix} n_1 \tan\left(\frac{\theta}{2}\right) \\ n_2 \tan\left(\frac{\theta}{2}\right) \\ n_3 \tan\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Note that Euler parameter representation uses 4 parameters to express the rotation, therefore it is also called the "quaternion (or unit quaternion) representation". $\boldsymbol{e}$ is called a unit quaternion since its norm is equal to 1, i.e. $\boldsymbol{e}^T\boldsymbol{e}=1$.

Euler Angles For Expressing Rotations

Euler angle representation uses 3 basic rotations combined in a sequence to express a general 3 dimensional rotation. Practically, this sequence is composed of independent rotations about the axes of the rotating reference frame in its instantaneous rotated configurations. It should be noted that the order in which the basic rotations are performed is important, since the change of order results in a completely different final orientation.

z-y-x or yaw-pitch-roll sequence, in which the orientation of the body-fixed reference frame, $\mathcal{F}_b$, with respect to the earth-fixed reference frame, $\mathcal{F}_O$, is expressed in such a way that $\mathcal{F}_b$ is first rotated about its z-axis, then about its y-axis at this rotated state, and finally about its x-axis by appropriate angles at each stage to achieve its current orientation, is conventionally used in nautical applications, therefore is also employed in this study for expressing the orientation of the ship. The process explained above has been shown schematically in the forthcoming equation (1).

$$\mathcal{F}_o \xrightarrow[\psi]{\vec{k}_o} \mathcal{F}_m \xrightarrow[\theta]{\vec{j}_m} \mathcal{F}_n \xrightarrow[\phi]{\vec{i}_n} \mathcal{F}_b \tag{1}$$

where $\mathcal{F}_m$ and $\mathcal{F}_n$ are the intermediate frames during rotation, $\vec{k}_o$, $\vec{j}_m$ and $\vec{i}_n$ are the axes and ψ, θ and ϕ are the angles of the basic rotations.

Component Transformations

Euler angles method is employed in expressing the rotations, and the reasoning for this purpose is discussed in the "Comparison of the representation methods" section. Once the translational and rotational velocities are obtained in the body-fixed reference frame, they should be transformed into the inertial reference frame to get the time rate of change of the instantaneous coordinates of point B, and the Euler angles in this frame.

Two transformation operations, which are decoupled from each other, are performed in this context. The first transformation is to obtain the time rates of coordinates of point B from the translational velocities expressed in the body-fixed frame, and the second one is performed in order to transform rotational velocities expressed in the body-fixed frame into the Euler angle rates.

Translational velocity transformation

The translational velocity transformation matrix is obtained by the multiplication of the three basic rotation matrices given below.

$$\hat{C}^{(O,m)}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\hat{\boldsymbol{C}}^{(m,n)}(\theta)=\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}$$

$$\hat{\boldsymbol{C}}^{(n,b)}(\phi)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$

where $s\alpha=\sin(\alpha)$, $c\alpha=\cos(\alpha)$, $(\alpha=\psi,\phi,\theta)$. It should be noted here that the matrix $\hat{\boldsymbol{C}}^{(O,m)}$ is used to define the rotation of $\mathcal{F}_m$ with respect to $\mathcal{F}_O$, where $\mathcal{F}_m$ is defined in the schematic representation (1), and the other transformations $\hat{\boldsymbol{C}}^{(m,n)}$ and $\hat{\boldsymbol{C}}^{(n,b)}$ are defined accordingly. Therefore, the transformation of components from $\mathcal{F}_b$ to $\mathcal{F}_O$ is obtained by the matrix

$$\hat{\boldsymbol{C}}^{(O,b)}=\hat{\boldsymbol{C}}^{(O,m)}\hat{\boldsymbol{C}}^{(m,n)}\hat{\boldsymbol{C}}^{(n,b)}=\begin{bmatrix} c\psi c\theta & -s\psi c\phi+c\psi s\theta s\phi & s\psi s\phi+c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\phi+s\psi s\theta s\phi & -c\psi s\phi+s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

Note that $\hat{\boldsymbol{C}}^{(O,b)}$, as well as $\hat{\boldsymbol{C}}^{(O,m)}$, $\hat{\boldsymbol{C}}^{(m,n)}$, and $\hat{\boldsymbol{C}}^{(n,b)}$, is an orthonormal matrix with the important property $\left(\hat{\boldsymbol{C}}^{(O,b)}\right)^{-1}=\left(\hat{\boldsymbol{C}}^{(O,b)}\right)^{T}$.

According to the notation described in Table 1, the translational velocity transformation is defined by the following equation:

$$\dot{\boldsymbol{\eta}}_1=\boldsymbol{J}_1(\boldsymbol{\eta}_2)\boldsymbol{v}_1$$

where $\boldsymbol{J}_1(\boldsymbol{\eta}_2)=\hat{\boldsymbol{C}}^{(O,b)}$.

Angular velocity transformation

Let $\vec{\omega}_{b/O}$ vector be the angular velocity of frame $\mathcal{F}_b$ with respect to frame $\mathcal{F}_O$. Then, according to the z-y-x sequence,

$$\vec{\omega}_{b/O} = \dot{\psi}\vec{k}_o + \dot{\theta}\vec{j}_m + \dot{\phi}\vec{i}_n \tag{2}$$

since we can express the angular velocity as a vector sum of angular velocities of successive rotations in a sequence. If we represent the vector equation (2) in the matrix form, we get

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

Note that $\hat{\boldsymbol{C}}^{(b,n)} = \left(\hat{\boldsymbol{C}}^{(n,b)}\right)^{-1} = \left(\hat{\boldsymbol{C}}^{(n,b)}\right)^{T}$ and $\hat{\boldsymbol{C}}^{(n,m)} = \left(\hat{\boldsymbol{C}}^{(m,n)}\right)^{-1} = \left(\hat{\boldsymbol{C}}^{(m,n)}\right)^{T}$. After collecting the matrix terms and inverting this matrix to obtain the transformation from angular velocities in the body-fixed frame to Euler angle rates, we get

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

where $t\alpha = \tan(\alpha)$ $(\alpha = \psi, \phi, \theta)$. Or in a more compact form,

$$\dot{\boldsymbol{\eta}}_2 = \boldsymbol{J}_2(\boldsymbol{\eta}_2)\boldsymbol{v}_2$$

Finally, the overall coordinate transformation from $\mathcal{F}_b$ to $\mathcal{F}_O$ can be obtained as

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{\eta}_2) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{J}_2(\boldsymbol{\eta}_2) \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} \tag{3}$$

where

$$\boldsymbol{J}_1(\boldsymbol{\eta}_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \text{ and}$$

$$\boldsymbol{J}_2(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}$$

A simpler notation for (3) is

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta}_2)\boldsymbol{v} \tag{4}$$

Comparison of the representation methods

Since Euler angles is a three-parameter representation, it shows a singularity in a particular configuration. For the *z-y-x* sequence, this case is met when the pitch angle, $\theta = \pm 90°$, since $\cos(\pm 90°) = 0$ and this makes $\boldsymbol{J}_2(\boldsymbol{\eta}_2)$ singular.

A second disadvantage of Euler angles is that this method is costly in terms of computation, therefore slower. Euler angles method requires the evaluation of trigonometric functions at each time step,

which are harder to compute with respect to simple multiplication and addition operations required by the angle-axis representation methods.

Since the ship has a slow dynamics that can tolerate the disadvantage introduced by the use of Euler angles, the singularity case $(\theta = \pm 90°)$ is unlikely to occur even in the most extreme motions of a marine vehicle, and the simulation software NSTMSS already uses the z-y-x Euler angle sequence for transformation we can use this sequence.

3.2.2. Kinetics

As discussed earlier in the previous section, Newtonian approach (vector mechanics) is followed in the derivation of the differential equations of motion. According to the Chasle's theorem, the motion of a rigid body can be represented simply by the superposition of a translation of a point that is fixed on the body and a pure rotation of the body about that point [8]. In the problem consideration this point is the origin of the body-fixed reference frame, B.

The matrix equation of motion for the system, which will soon be introduced, is a collection of first order scalar differential equations expressed in terms of translational and angular velocities in the body-fixed reference frame. Velocities are transformed to the earth-fixed reference frame and integrated to give the position and orientation variables in a completely decoupled process as described in (3).

The governing differential equations can be obtained by applying the Euler's first and second axioms regarding the conservation of momentum, which are based on Newton's second law of motion. With the assumption of constant mass, moments of inertia and inertia products, the expressions of these axioms can be given as

$$\dot{\vec{P}}_G = m\dot{\vec{\mathbf{v}}}_G = \vec{F}_G$$
$$\dot{\vec{H}}_G = \breve{I}_G \cdot \dot{\vec{\omega}}_{b/O} = \vec{M}_G$$

where m is the mass and $\breve{I}_G$ can be resolved to get the inertia matrix defined about a reference frame that is parallel to $\mathcal{F}_b$, however with the origin fixed at the CG.

Best choice to locate the origin of the body-fixed frame is obviously the CG because of the simple form that the equations will have. However, there may be cases where the available data of a system (force coefficients, moments of inertia, etc.) is based on a reference frame whose origin is not coincident with the CG of the ship, or a simulation of the effects of a dynamic offset in the CG may be required. In such cases, these simple equations can be generalized to a case where $\mathcal{F}_b$ is not fixed on the CG.

The vectors used in the derivation of motion equations are shown together with the reference frames in Figure 6.

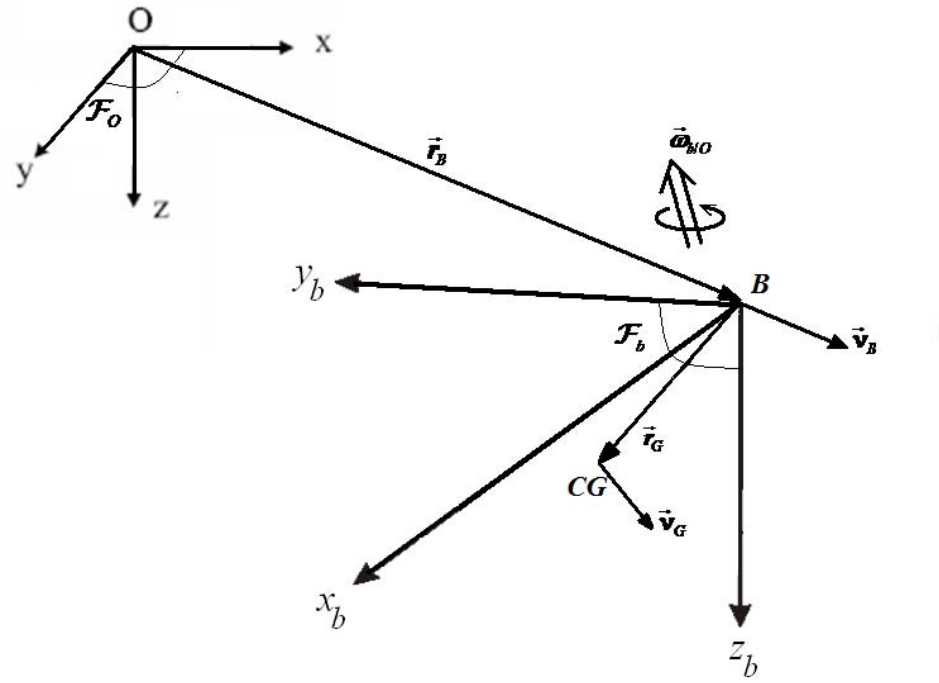

Figure 6 – Vectors used in the derivation of equation of motion

Keeping in mind that the resolution of the velocity vector $\vec{\mathbf{v}}_B$ in reference frame $\mathcal{F}_b$ is simply equal to $\boldsymbol{v}_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T$, and that of $\vec{\boldsymbol{\omega}}_{b/O}$ is defined as $\boldsymbol{v}_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T$, the resultant motion equations may be given in the matrix form in its most compact form as [7]

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{v}} + \boldsymbol{C}_{RB}\left(\boldsymbol{v}\right)\boldsymbol{v} = \boldsymbol{\tau}_{RB} \tag{5}$$

where

$$\boldsymbol{M}_{RB} \triangleq \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{I}_{3\times 3} & -m\tilde{\boldsymbol{r}}_G \\ m\tilde{\boldsymbol{r}}_G & \boldsymbol{I}_B \end{bmatrix}$$

and

$$\boldsymbol{C}_{RB}(\boldsymbol{v}) \triangleq \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -m\tilde{\boldsymbol{v}}_1 - m\left(\tilde{\tilde{\boldsymbol{v}}}_2\boldsymbol{v}_1\right) \\ -m\tilde{\boldsymbol{v}}_1 - m\left(\tilde{\tilde{\boldsymbol{v}}}_2\boldsymbol{r}_G\right) & -m\left(\tilde{\tilde{\boldsymbol{v}}}_1\boldsymbol{r}_G\right) - \left(\tilde{\boldsymbol{I}}_B\boldsymbol{v}_2\right) \end{bmatrix} \tag{6}$$

$\boldsymbol{I}_{3\times3}$ is an identity, $\boldsymbol{0}_{3\times3}$ is a null matrix of size 3x3, and $\boldsymbol{I}_B$ is the matrix of moments and products of inertia defined about $\mathcal{F}_b$ which is expressed as

$$\boldsymbol{I}_B \triangleq \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

and $\boldsymbol{r}_G$ is decomposed as $\begin{bmatrix} x_G & y_G & z_G \end{bmatrix}^T$.

As stated earlier, $\boldsymbol{C}_{RB}$, written in this form may also be obtained by utilizing the component matrices of $\boldsymbol{M}_{RB}$ ($\boldsymbol{M}_{11}$, $\boldsymbol{M}_{12}$, etc.). $\boldsymbol{C}_{RB}$, in this context is

$$\boldsymbol{C}_{RB}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -\tilde{\boldsymbol{c}}_1 \\ -\tilde{\boldsymbol{c}}_1 & -\tilde{\boldsymbol{c}}_2 \end{bmatrix}$$

where

$$\left.\begin{aligned} \boldsymbol{c}_1 &= \boldsymbol{M}_{11}\boldsymbol{v}_1 + \boldsymbol{M}_{12}\boldsymbol{v}_2 \\ \boldsymbol{c}_2 &= \boldsymbol{M}_{21}\boldsymbol{v}_1 + \boldsymbol{M}_{22}\boldsymbol{v}_2 \end{aligned}\right\} \quad \leftrightarrow \quad \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} = \boldsymbol{M}_{RB}\boldsymbol{v}$$

It should also be noted that the tilde ($\tilde{}$) operator is the cross product matrix operator, which is used to express the vector cross

product operation in the matrix form. That is, the cross product $\vec{\boldsymbol{p}} \times \vec{\boldsymbol{q}}$ can be expressed in the matrix form as

$$\tilde{\boldsymbol{p}}\boldsymbol{q} = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Equation (5) can also be written in its expanded form as follows:

$$m\left[\dot{u} - vr + wq - x_G\left(q^2 + r^2\right) + y_G\left(pq - \dot{r}\right) + z_G\left(pr + \dot{q}\right)\right] = X$$

$$m\left[\dot{v} - wp + ur - y_G\left(p^2 + r^2\right) + z_G\left(qr - \dot{p}\right) + x_G\left(qp + \dot{r}\right)\right] = Y$$

$$m\left[\dot{w} - uq + vp - z_G\left(q^2 + p^2\right) + x_G\left(rp - \dot{q}\right) + y_G\left(rq + \dot{p}\right)\right] = Z$$

$$I_{xx}\dot{p} + \left(I_{zz} - I_{yy}\right)qr - I_{xz}\left(\dot{r} + pq\right) + I_{yz}\left(r^2 - q^2\right) + I_{xy}\left(pr - \dot{q}\right)$$
$$+ m\left[y_G\left(\dot{w} - uq + vp\right) - z_G\left(\dot{v} - wp + ur\right)\right] = K$$

$$I_{yy}\dot{q} + \left(I_{xx} - I_{zz}\right)rp - I_{xy}\left(\dot{p} + qr\right) + I_{zx}\left(p^2 - r^2\right) + I_{yz}\left(qp - \dot{r}\right)$$
$$+ m\left[z_G\left(\dot{u} - vr + wq\right) - x_G\left(\dot{w} - uq + vp\right)\right] = M$$

$$I_{zz}\dot{r} + \left(I_{yy} - I_{xx}\right)pq - I_{yz}\left(\dot{q} + rp\right) + I_{xy}\left(q^2 - p^2\right) + I_{zx}\left(rq - \dot{p}\right)$$
$$+ m\left[x_G\left(\dot{v} - wp + ur\right) - y_G\left(\dot{u} - vr + wq\right)\right] = N$$

3.3. External Forces and Moments

So far, we have obtained the general form of the motion equations, which define the relation between the body-fixed velocities and the external forces (Note that the term “forces” used in this context refers to both forces and moments) on the ship. In this section we

are going to deal with different types of these external forces in detail, and decompose the simple looking rigid body force vector, $\boldsymbol{\tau}_{RB}$ term on the right hand side of equation (5), into its components in order to obtain a form for the motion equation that is used specifically for marine vehicle models.

Rigid body force vector can be partitioned to its components as [12]

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau}_{hyd} + \boldsymbol{\tau}_{prop} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}_{env}$$

where

$\boldsymbol{\tau}_{hyd}$: Vector of hydrodynamic forces

$\boldsymbol{\tau}_{prop}$: Forces arising from the thrust supplied by the propeller(s)

$\boldsymbol{\tau}_{cs}$: Forces caused by the movement of the control surfaces

$\boldsymbol{\tau}_{env}$: Environmental disturbances, which can be expanded as

$$\boldsymbol{\tau}_{env} = \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{current}$$

3.3.1. Hydrodynamic forces

Forces and moments on the ship hull caused by the dynamic interaction of the surface of the body and the water surrounding it are called the hydrodynamic forces in the most general sense. These forces may be studied as a superposition of two subclasses. First of them is the case where the ship is "forced to oscillate with the wave excitation frequency and there are no incident waves" [7]. That is, the sea surface is assumed to be still and the forces applied on the ship are due to damping, added inertia and restoring

forces, which will soon be explained. The second class includes the forces that appear when the ship is "restrained from moving and there are incident regular waves", meaning the hydrodynamic interaction of the steady ship hull with the sea where there exists waves on the sea surface.

Hydrodynamic derivatives (coefficients)

Hydrodynamic forces on the body are non-linear functions of velocities, accelerations and Euler angles [1]. This can be expressed as follows:

$$\boldsymbol{\tau}_{hyd} = \boldsymbol{\tau}_{hyd}\left(\dot{\boldsymbol{\nu}}, \boldsymbol{\nu}, \boldsymbol{\eta}\right)$$

Determining the functions governing these effects is not a straightforward task, therefore expanding them in multi-variable Taylor series, and determining the necessary coefficients instead is a frequently used method. These expansions are performed up to the third order for most practical purposes. An example of such an expansion for the roll moment may be given as follows:

$$\begin{aligned} K = & K_{\dot{v}}\dot{v} + K_{\dot{p}}\dot{p} \\ & + K_{|u|v}|u|v + K_{ur}ur + K_{|v|v}|v|v + K_{|v|r}|v|r + K_{|r|v}|r|v \\ & + K_{p}p + K_{|p|p}|p|p + K_{|u|p}|u|p + K_{|uv|\phi}|uv|\phi + K_{|ur|\phi}|ur|\phi \\ & + K_{uu\phi}u^2\phi + K_{|u|p}|u|p + K_{\phi\phi\phi}\phi^3 - \rho g \nabla G_z(\phi) \end{aligned} \tag{7}$$

The coefficients are formulated as the partial derivatives of the force under consideration (the roll moment K, for this particular

case) with respect to the multiplicand motion variables. For instance the coefficient of the $u^2\phi$ term can be expressed as

$$K_{uu\phi} = \frac{\partial^3 K}{\partial u \partial u \partial \phi} \tag{8}$$

Theoretical methods like strip theory, and system identification experiments performed in facilities such as rotating arm, free oscillator, forced oscillator, curved flow tunnel, oblique towing tests, planar motion mechanism (PMM), and roll planar motion mechanism (RPMM) [1] , [7], [12].

Strip theory is used to derive the 3-dimensional added inertia coefficients of a slender body by dividing submerged part of the hull into thin elements across the longitudinal axis, which are called the strips, and summing the 2-dimensional added inertia coefficients calculated for each strip.

In the most frequently used experimental technique, planar motion mechanism, two oscillators, one at the bow and the other at the astern, are employed to excite the ship body and the forces generated are measured to determine all hydrodynamic coefficients in 6 DOF.

In the following sections, different origins of the hydrodynamic forces are analyzed based on the definitions put forward in the first paragraph of section 3.3.1.

Non-Dimensional Variables and Parameters

It's common practice in nautical applications to non-dimensionalize the variables and parameters used in the motion equations. By non-dimensionalization, the comparison of the hydrodynamic derivatives between different types of ships, especially if they vary highly in size, becomes possible. In scale model tests to determine hydrodynamic coefficients or simply to measure performance, this concept is employed.

In order to use the non-dimensional hydrodynamic coefficients in a simulation, all other parameters and the motion variables should also be non-dimensional, and this normalization process is carried out by dividing these variables by a set of standard factors. There are 3 systems in the literature, which are referred the most, for this purpose; Prime System I, Prime System II, and Bis System. Prime System I of SNAME, proposed in 1950 is the most widely used one of these three systems. The normalization variables for this system are listed in Table 2 below:

Table 2 – Normalization factors in Prime System I of SNAME

Length	L		Linear velocity	U
Mass	$\rho/2\, L^3$		Angular velocity	U/L
Inertia Moment	$\rho/2\, L^5$		Linear acceleration	U^2/L
Time	L/U		Angular acceleration	U^2/L^2
Reference Area	L^2		Force	$\rho/2\, U^2 L^2$
Position	L		Moment	$\rho/2\, U^2 L^3$
Angle	1			

For instance the non-dimensional form of the sway velocity, v, which is denoted by v', is

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

And that of the ship's mass is

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

The set of normalization factors for the other two systems can be found in [1].

Radiation-induced hydrodynamic forces

Of the two subclasses of hydrodynamic forces, the first group as mentioned above is recognized under this category. Sources for this kind of forces can be analyzed considering the motion of the body in an ideal fluid with and without circulation, motion under viscous effects, and gravitational and buoyancy forces [12]. Most commonly, these force class is a combination of the following components:

- Added mass (or inertia in a more general sense) forces,
- Radiation induced potential damping,
- Restoring (gravitational and buoyancy) forces

Added Inertia

The interesting concept of added (virtual) inertia sounds as if it is the inertia of water that "sticks" on the hull and accelerates with the nominal mass of the body, which is actually not the case. Forced harmonic motion of the body results in the oscillation of the fluid particles increasing their kinetic energy in the close proximity of the hull surface in phase with the motion of the body. This phenomenon creates pressure induced resistance forces and moments to the motion of the vessel those are proportional to the acceleration of the body. Therefore they may be treated mathematically as they are additional inertial effects against the motion of the vessel. These effects also add up to the Coriolis and centripetal terms. Added inertia force definition is as follows:

$$\boldsymbol{\tau}_A = -\boldsymbol{M}_A\dot{\boldsymbol{\nu}} - \boldsymbol{C}_A(\boldsymbol{\nu})\boldsymbol{\nu}$$

Added inertia matrix $\boldsymbol{M}_A$ is defined as

$$\boldsymbol{M}_A \triangleq \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \triangleq -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}$$

Note that the terms contained in the added inertia matrix, $\boldsymbol{M}_A$, are single derivatives of the forces with respect to the acceleration variables. Experiments show that these terms can be decoupled from the terms containing the velocity and Euler angle variables. For ex-

ample, the first line in the expansion (7) shows components associated with the added inertia contributing to the roll moment.

Similar to the derivation of $\boldsymbol{C}_{RB}(\boldsymbol{v})$ from $\boldsymbol{M}_{RB}$ in (6), the Coriolis and centripetal effects matrix associated with added inertias, $\boldsymbol{C}_A(\boldsymbol{v})$, may be given as

$$\boldsymbol{C}_A(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & -\tilde{\boldsymbol{a}}_1 \\ -\tilde{\boldsymbol{a}}_1 & -\tilde{\boldsymbol{a}}_2 \end{bmatrix}$$

where

$$\left.\begin{aligned} \boldsymbol{a}_1 &= \boldsymbol{A}_{11}\boldsymbol{v}_1 + \boldsymbol{A}_{12}\boldsymbol{v}_2 \\ \boldsymbol{a}_2 &= \boldsymbol{A}_{21}\boldsymbol{v}_1 + \boldsymbol{A}_{22}\boldsymbol{v}_2 \end{aligned}\right\} \quad \leftrightarrow \quad \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix} = \boldsymbol{M}_A\boldsymbol{v}$$

Damping

Together with the radiation induced potential damping and the restoring forces we may collect the radiation induced forces in a single term, $\boldsymbol{\tau}_R$, as

$$\boldsymbol{\tau}_R = -\boldsymbol{M}_A\dot{\boldsymbol{v}} - \boldsymbol{C}_A(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{D}_P(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{\eta})$$

However for a more general definition of the hydrodynamic forces, we may define a general damping matrix to include the damping forces, which are not of radiation-induced nature, in the system such as skin friction $\left(\boldsymbol{D}_S(\boldsymbol{v})\boldsymbol{v}\right)$, wave drift damping $\left(\boldsymbol{D}_W(\boldsymbol{v})\boldsymbol{v}\right)$ and

damping due to vortex shedding $\left(\boldsymbol{D}_M(\boldsymbol{v})\boldsymbol{v}\right)$. Defining an extended damping matrix to include all these effects as

$$\boldsymbol{D}(\boldsymbol{v}) \triangleq \boldsymbol{D}_P(\boldsymbol{v}) + \boldsymbol{D}_S(\boldsymbol{v}) + \boldsymbol{D}_W(\boldsymbol{v}) + \boldsymbol{D}_M(\boldsymbol{v}) \tag{9}$$

In order to satisfy the dissipative nature of the damping function for both positive and negative velocities, one approach to achieve this result is assuming the function as an odd function of the velocities. As an example, remembering that the expansions are generally truncated past the third order terms, damping force in the surge direction becomes

$$X_D = -X_u u - X_{uuu} u^3$$

in this context. As an alternative to this assumption, a second order damping model is proposed with the name "second order modulus model" [16]. Starting with the well-known damping term in the Morrison's equation [19] in 1 DOF, which is

$$D = -\frac{1}{2}\rho C_D A |U| U$$

we may make a generalization to obtain this quadratic damping model in 6 DOF. For the sake of consistency with the notation, the quadratic damping forces may be expressed as follows [7]:

$$\boldsymbol{d}_q = \begin{bmatrix} |\boldsymbol{v}|^T \boldsymbol{D}_1 \boldsymbol{v} \\ |\boldsymbol{v}|^T \boldsymbol{D}_2 \boldsymbol{v} \\ |\boldsymbol{v}|^T \boldsymbol{D}_3 \boldsymbol{v} \\ |\boldsymbol{v}|^T \boldsymbol{D}_4 \boldsymbol{v} \\ |\boldsymbol{v}|^T \boldsymbol{D}_5 \boldsymbol{v} \\ |\boldsymbol{v}|^T \boldsymbol{D}_6 \boldsymbol{v} \end{bmatrix}$$

where $\boldsymbol{D}_i = \boldsymbol{D}_i(\rho, C_D, A)$, $(i = 1 \ldots 6)$, and $|\boldsymbol{v}|$ denotes a vector whose components are the absolute values of $\boldsymbol{v}$. Note that, since the velocities are defined in the body frame, and water is an incompressible media, we may consider ρ and A constant. However, C_D depends on the Reynold's number, therefore the velocity. $\boldsymbol{D}_i$ matrices may be selected constant for practical purposes. However, they will be valid only about a nominal velocity vector in this case.
The total damping force on the body may be modeled by a linear and a quadratic term. That is,

$$\begin{aligned} \boldsymbol{d} = \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} &= \boldsymbol{d}_l + \boldsymbol{d}_q \\ &= \left(\boldsymbol{D}_l + |\boldsymbol{v}|^T \boldsymbol{D}_q\right)\boldsymbol{v} \end{aligned}$$

The relation between these two components and the damping matrix definition given in (9) is

$$\begin{aligned} \boldsymbol{D}_l &\triangleq \boldsymbol{D}_P + \boldsymbol{D}_S \\ |\boldsymbol{v}|^T \boldsymbol{D}_q &\triangleq \boldsymbol{D}_W(\boldsymbol{v}) + \boldsymbol{D}_M(\boldsymbol{v}) \end{aligned}$$

Furthermore, we may express the damping matrix in terms of hydrodynamic derivatives using the index notation as follows:

$$\boldsymbol{D}(\boldsymbol{v}) = \left[D_{ij} \right] = \left[\boldsymbol{f}(i)_{v(j)} + \sum_{k=1}^{6} \boldsymbol{f}(i)_{|v(k)|v(j)} \right]$$

Restoring forces

Gravitational and buoyancy forces are studied under this title. These forces are functions of position and orientation variables. Before going into the details of each force term, a brief explanation about the term metacentric stability will be given [12], [7], [5].

The static stability condition of the ship under the effect of restoring forces, which are buoyancy and weight, is called the metacentric stability. A metacentrically stable ship is one that is able to restore its posture after a perturbation is applied on the roll or pitch degree of freedom of the ship. This phenomenon is valid for a band of angles. That is to say, if a critical angle is exceeded, the ship would tend to settle down about a new equilibrium position. It is not expected to encounter such a limiting case in pitch motion. However, capsizing, which occurs when the critical roll angle is exceeded, is a major danger and a very important concern in the design of ships.

Considering the roll motion of the ship within the aforementioned safety band of roll angles, there is a restoring moment in the form

$$K_{r,\phi} = -\rho g \nabla \left| GM_T \right| \sin(\phi)$$

where ρ is the density of water, ∇ is the volume of displaced water, $\left|GM_T\right|$ is the transverse metacentric height, and ϕ is the roll

angle. Here, it should be noted that $K_{r,\phi}$ is the restoring moment itself, not a force coefficient as in (8) The definition of the transverse metacenter, M_T point, can be seen on Figure 7. $|GM_T|$ is the distance between CG and M_T, and BC_1 and BC_2 are the original and disturbed positions of the center of buoyancy, respectively.

Similarly, there is a longitudinal metacenter, M_L, which acts as center of rotation in case of disturbance in the pitch degree of freedom. In general, longitudinal metacentric height of a particular ship is 10 to 30 times greater than the transverse metacentric height of it. This fact is a major reason for the roll motion being very critical and analyzed quite frequently and widely, while pitch motion having relatively less importance.

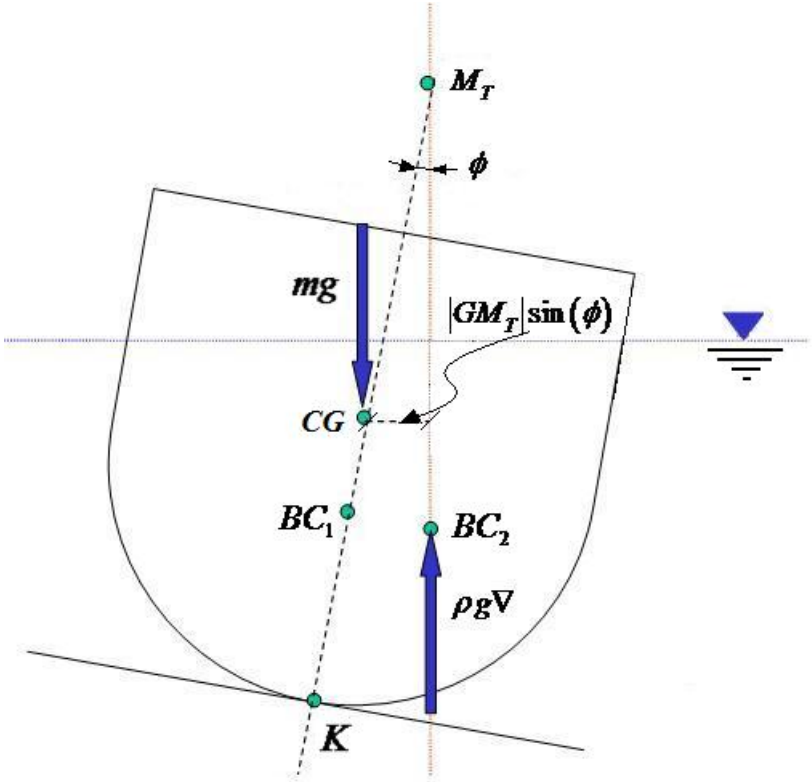

Figure 7 – Transversal restoring forces on the ship

Heave motion of the ship is dominated by the effect of restoring forces. If the hull is assumed to have a shape of a rectangular prism, and a constant waterline area (A_w), which is the area enclosed by the curve at the intersection of the body and the water surface, then the vertical force that forms as a result of the deviation in the vertical position of the ship with respect to the equilibrium position, can be given by [5]

$$F_z = \rho g A_w \Delta z$$

where $\Delta z \triangleq z - z_{eq}$. Let the representation of the restoring force vector in the earth-fixed reference frame be $\overline{\boldsymbol{f}}_r^{(O)}$. Then,

$$\overline{\boldsymbol{f}}_r^{(O)} = \begin{bmatrix} 0 & 0 & -F_z \end{bmatrix}$$

Since the resolution of this vector in the body-frame is required to be used in the equations of motion, we may obtain it by the following transformation:

$$\overline{\boldsymbol{f}}_r^{(b)} = \boldsymbol{J}_1^{-1} \overline{\boldsymbol{f}}_r^{(O)}$$

Since the inverse of the orthonormal transformation matrix $\boldsymbol{J}_1$ can be taken as $\boldsymbol{J}_1^{-1} = \boldsymbol{J}_1^T$, we can obtain $\overline{\boldsymbol{f}}_r^{(b)}$ as

$$\overline{\boldsymbol{f}}_r^{(b)} = \boldsymbol{J}_{1(3)}{}^T F_z$$

where $\boldsymbol{J}_{1(3)}$ stands for the 3rd row of $\boldsymbol{J}_1$. Furthermore, the effect of pitch angle (θ) on the heave motion (along $\vec{z}_b$), and the effect of the vertical displacement (along $\vec{z}$) on the pitch motion can be given by the influence coefficient $-\rho g A_w x_{BC}$, where x_{BC} is the location of the buoyancy center in the surge direction. That is to say,

$$Z_{r,\theta} = -\rho g A_w x_{BC} \sin(\theta)$$
$$M_{r,z} = -\rho g A_w x_{BC} z$$

Finally, the restoring forces vector $\boldsymbol{g}(\boldsymbol{\eta})$ may be obtained by the collection of these force components as follows:

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} \overline{\boldsymbol{f}}_r^{(b)} + \begin{bmatrix} 0 \\ 0 \\ \left(-\rho g A_w x_{BC}\right) z \end{bmatrix} \\ -\rho g \nabla \left| GM_T \right| \sin(\phi) \\ -\rho g \nabla \left| GM_L \right| \sin(\phi) + \left(-\rho g A_w x_{BC}\right) \sin(\theta) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \rho g A_w \left(z - z_{eq}\right) \sin(\theta) \\ \rho g A_w \left(z - z_{eq}\right) \cos(\theta) \sin(\phi) \\ \rho g A_w \left(z - z_{eq}\right) \cos(\theta) \cos(\phi) - \rho g A_w x_{BC} z \\ -\rho g \nabla \left| GM_T \right| \sin(\phi) \\ -\rho g \nabla \left| GM_L \right| \sin(\phi) - \rho g A_w x_{BC} \sin(\theta) \\ 0 \end{bmatrix} \tag{10}$$

As a result, we obtain the overall hydrodynamic forces vector denoted by $\boldsymbol{\tau}_{hyd}$ as

$$\boldsymbol{\tau}_{hyd} = -\boldsymbol{M}_A \dot{\boldsymbol{v}} - \boldsymbol{C}_A(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{\eta})$$

Note that the $\boldsymbol{\tau}_{hyd}$ includes only the first subclass of the hydrodynamic forces with a slight modification on the damping term. The second type of hydrodynamic forces, which are Froude-Kriloff and diffraction forces, are contained in the $\boldsymbol{\tau}_{env}$ term.

If the equation of motion is re-constructed, defining combined mass matrix and combined Coriolis matrix as

$$\boldsymbol{M} \triangleq \boldsymbol{M}_{RB} + \boldsymbol{M}_A \text{ and } \boldsymbol{C}(\boldsymbol{v}) \triangleq \boldsymbol{C}_{RB}(\boldsymbol{v}) + \boldsymbol{C}_A(\boldsymbol{v}),$$

it takes the following form

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{prop} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}_{env} \tag{11}$$

Froude-Kriloff and diffraction forces

As mentioned above, this is the second subclass of hydrodynamic forces that deal with the wave effects on the body that is hypothetically kept stationary. It consists of forces that arise from the wave potential "assuming no interaction between the waves and the body" (Froude-Kriloff), and those associated with the diffraction potential existent as a result of "interaction between the body and the waves" [18].

Details of the wave model will be explained later on. We may conclude this section by pointing out that these forces appear in the system equations through the term $\boldsymbol{\tau}_{wave}$, which is a component of $\boldsymbol{\tau}_{env}$.

3.3.2. Control Forces

This class of external forces includes the forces input to the system by the user, which are propulsion and the forces generated by the deflections of the control surfaces. In most of the models available in the literature, the effect of propulsion is only on the surge mode. The most common type of the control surfaces is the rudder, and the related sub-section mainly deals with the rudder forces.

Propulsion Model

A first order model is used to represent the dynamics of a single shaft. The transfer function between the commanded and the actual shaft speeds is,

$$\frac{n}{n_C} = \frac{1}{T_n s + 1}$$

where n and n_C are the actual and the commanded shaft speeds, respectively, and T_n is the time constant for the shaft dynamics. For practical purposes, assuming a linear relation between shaft speed and propeller generated thrust is reasonable. That is,

$$T_p = K_T n$$

K_T being a constant gain relating the actual shaft speed to the propeller generated thrust, T_p. This relation may be rewritten as

$$T_p = T_{\max} \frac{n}{n_{\max}}$$

so that the basic motor characteristics such as maximum shaft speed and maximum thrust can be used to obtain the propeller thrust.

A second issue to be considered in the propulsion model is the effect of the ship hull on the flow going into the propeller. A reduction occurs in the thrust supplied by the propeller in the vicinity of

the ship hull. Then the thrust available to move the ship, T_a, when this effect is considered can be expressed as,

$$T_a = (1-t)T_p$$

where t is the thrust deduction number varying between 0.05 and 0.2 [12]. Then the final expression for the available thrust as a function of the shaft speed

$$T_a = \left[\frac{(1-t)T_{\max}}{n_{\max}}\right] n$$

Two configurations for the propulsion system are considered, which are single-screw and twin-screw. Meko federate of NSTMSS uses a twin-screw propulsion model, whereas most of the models available in the literature have single-screw shafts.

Due to the offsets present in the alignment of the propeller shaft(s), secondary effects are observed on the motion modes other than surge, such as a pitch moment contribution. In both single and twin-screw cases, the line(s) of application of the thrust force(s) are considered to be parallel to the $\vec{x}_b$ axis of the body frame. In a single-screw configuration, assuming that the line of application of the thrust force, lies in the xz-plane of $\mathcal{F}_b$, one of the secondary contributions is on the pitch moment through the offset between the line of application of the thrust and the $\vec{x}_b$ axis, which is shown in Figure 8.

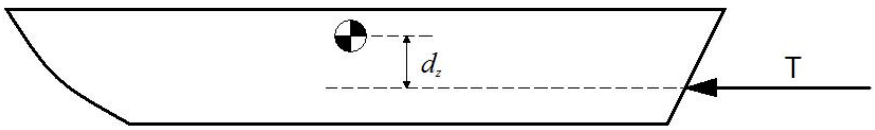

Figure 8 – Propulsion system vertical offset

The contribution of the supplied thrust to the pitch moment is given by the following expression:

$$M_{prop} = T_a d_z$$

A counter torque is caused by the axial moment of inertia of the rotor-shaft assembly on the ship in the roll degree of freedom while it is accelerating and by the drag torque on the propeller which is a function of the shaft speed. In twin-screw propellers, this effect vanishes when both shafts are rotating at the same speed and/or accelerating at the same rate, which is the case observed most of the time when following a straight ahead course, since the senses of rotation for the shafts are opposite to each other (see Figure 9, adapted from [20]). The roll moment applied on the ship body by a single shaft is

$$K_{prop} = I_{xx,shaft}\dot{n} + D_{prop}(n)$$

where $D_{prop}(n)$ is the drag moment on the propeller, which is a function of the shaft speed, assuming the positive sense of rotation counter clockwise as viewed from behind.

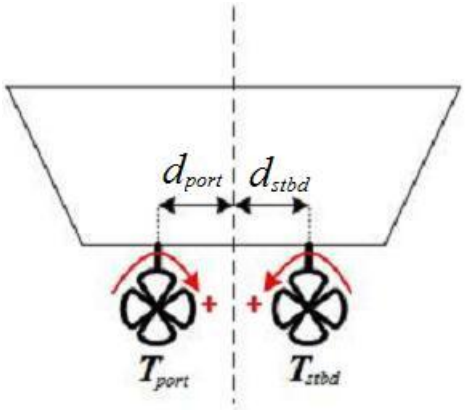

Figure 9 – Twin-screw shaft configuration

In such a configuration, an additional moment is generated on the yaw degree of freedom in case of thrust difference between the propellers. The expression for this yaw moment is,

$$N_{prop} = \left(T_{port} - T_{stbd}\right) d_{shaft} / 2$$

noting that T_{port} and T_{stbd} are the available thrusts obtained by the port and starboard propellers, respectively, and assuming that $d_{port} = d_{stbd}$ and d_{shaft} is the distance between the parallel shafts, i.e. $d_{shaft} = d_{port} + d_{stbd}$.

As a result, the forces on the body of the ship caused by the propulsion system may be given as

$$\boldsymbol{\tau}_{prop} = \begin{bmatrix} T_a \\ 0 \\ 0 \\ I_{xx,shaft}\dot{n} + D_{prop} \\ T_a d_z \\ 0 \end{bmatrix}$$

for a single-screw propulsion scheme, and

$$\boldsymbol{\tau}_{prop} = \begin{bmatrix} T_{port} + T_{stbd} \\ 0 \\ 0 \\ I_{xx,shaft}\left(\dot{n}_{stbd} - \dot{n}_{port}\right) + D_{prop,stbd}\left(n\right) - D_{prop,port}\left(n\right) \\ \left(T_{port} + T_{stbd}\right) d_z \\ \left(T_{port} - T_{stbd}\right) d_{shaft}/2 \end{bmatrix}$$

for a twin-screw one.

Rudder Model

Rudder deflection is the primary control input to the system for the steering action. Although in a twin-screw propulsion scheme, the thrust difference between the shafts supplies an additional yaw moment on the body, it only acts as an aid to maneuver the ship. The force generated on the rudder upon a deflection δ is schematically shown in Figure 10 [12].

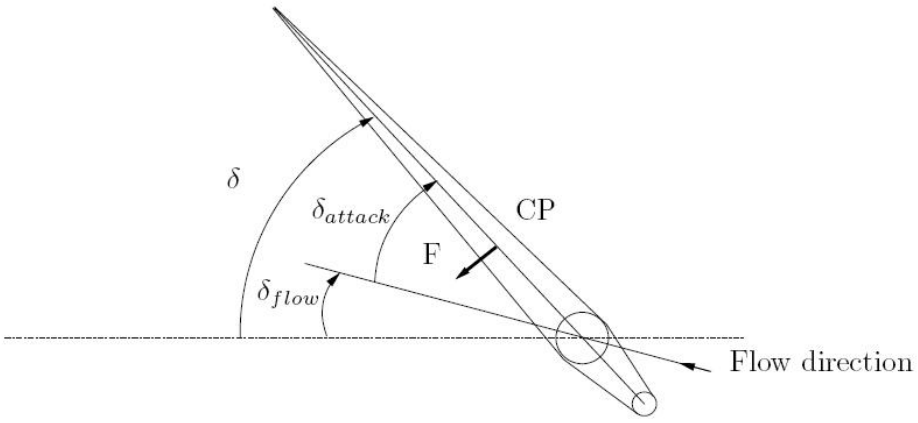

Figure 10 – Rudder deflection and force generated

The rudder is similar to a symmetric airfoil generating lift upon a given angle of attack, which is the angle between the air flow velocity and the airfoil symmetry axis. The rudder force is generated the same way in the existence of an angle of attack, δ_{attack} as shown in Figure 10. Noting also that the velocity of the flow past the rudder may have a lateral component caused by the ocean currents, wave motion or the sway motion of the ship , the flow direction and the longitudinal axis of the ship may not coincide in general, and there exists an angle, δ_{flow}, between the flow velocity and the ship's longitudinal axis. The total deflection of the rudder from its original position is the sum of these two angles, i.e.

$$\delta = \delta_{flow} + \delta_{attack}$$

A more extensive approach to the rudder force expression, which also includes the hydrodynamic drag in addition to the lift generated, is given in [14]. According to this approach, the total rudder force is the vector summation of the two orthogonal forces, lift and drag, i.e.

$$F_{rud} = \sqrt{D_{rud}^2 + L_{rud}^2}$$

Lift and drag forces on the rudder can be given as

$$L_{rud} = \frac{1}{2} C_L \rho_{water} V_{flow}^2 A_{rud}$$

$$D_{rud} = \frac{1}{2}\left(C_D + \frac{C_L^2}{0.9\pi\Lambda} \right) \rho_{water} V_{flow}^2 A_{rud}$$

where C_L and C_D are the lift and drag coefficients, and Λ and A_{rud} are the aspect ratio and surface area of the rudder, respectively. It is worth mentioning that the drag force on the rudder, D_{rud}, is the force component that is in the direction of the flow. Note that the term $\frac{C_L^2}{0.9\pi\Lambda}$ is associated with the lift-induced drag force on the rudder [23]. If the lateral component of the flow velocity is neglected, i.e. it is assumed that $\delta = \delta_{attack}$, the forces applied on the ship body when a rudder deflection is given become

$$
\begin{aligned}
X_{rud} &= -D_{rud} \cdot \delta \\
Y_{rud} &= L_{rud} \cdot \delta \\
K_{rud} &= -\left|GP\right|_v L_{rud} \cdot \delta \\
N_{rud} &= -\left|GP\right|_l L_{rud} \cdot \delta
\end{aligned}
$$

where $\left|GP\right|_v$ and $\left|GP\right|_l$ are the vertical and longitudinal offsets of center of pressure of the rudder and the center of gravity of the ship, respectively. In these definitions, δ is taken positive if the rudder is deflected to port side, which is a clockwise rotation as viewed from top.

A second method in obtaining the rudder forces, which is more appropriate for practical purposes, is the use of rudder force coefficients similar to the hydrodynamic coefficients explained in section 3.3.1. These coefficients can also be determined by test facilities, such as RPMM, and used in the model similarly to the hydrodynamic coefficients. The rudder roll moment expression obtained this way is given in equation (12) as an example [12].

$$K_{rud} = K'_{\delta}\delta' + K'_{\delta\delta}\delta'^2 + K'_{\delta\delta\delta}\delta'^3 + K'_{\delta v}\delta' v' + K'_{\delta vv}\delta' v'^2 + K'_{\delta u}\delta'\Delta u' + K'_{\delta\delta u}\delta'^2\Delta u' + K'_{\delta\delta\delta u}\delta'^3\Delta u' \quad (12)$$

where $\Delta u'$ is the perturbation in the surge velocity around the nominal surge velocity and the variables or parameters with prime symbols (δ', $K'_{\delta\delta u}$, etc.) indicate non-dimensionalization on them. Also, 3DOF parameter sets for four ships (tanker, series 60, ferry, and container), including these rudder force coefficients are given in [1].

3.3.3. Environmental Disturbances

Three sources of environmental effects on the ship are studied. These are,

- Wave effects,
- Wind disturbances,
- Ocean currents.

Wave effects

Mathematical representation of the wave elevation as a function of space and time can be given as a sum of sinusoids with random phases. This is expressed by the following formula [13], [10]:

$$\begin{aligned}\zeta(x,y,t) = &\sum_{i=1}^{N} A_i \cos\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right) \\ &+ \sum_{i=1}^{N} \frac{1}{2} k_i A_i^2 \cos 2\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right) \\ &+ f\left(A_i^3\right) + \ldots\end{aligned} \quad (13)$$

Once the frequency content of complex ocean waves is generated, they can be modeled as a finite sum of simple sinusoids based on this spectrum. Wave elevation, denoted by ζ, is a function of the coordinates of the grid which is parallel to the xy-plane of the inertial reference frame and time.

The 1st order sum with the coefficient A_i is used to represent the oscillatory motion of the vehicle, while the 2nd order sum whose coefficients are $\frac{1}{2}k_i A_i^2$ accounts for the wave drift forces in the above Stoke's expansion. An expansion of this kind up to the 2nd order is enough for most simulation purposes.

A simulation application of this formulation can be observed in the Vega Marine Module® developed by the Multigen-Paradigm Inc. [15]. It uses a rather simple form of this formula, where a 1st order approximation is used and the coefficients for each wave component are simply equal to $0.112H_s$.

The algorithm to calculate the wave forces acting on the ship can be described as follows: First the spectral density is computed according to the assumptions and parameters selected. This process will soon be explained in this section. Once the spectrum is available, it is chopped into N pieces of width $\Delta\omega$ along the frequency axis, and $S(\omega_i)$'s are calculated where ω_i's are the frequencies randomly selected from each of the intervals (Note that Vega Marine Module® takes $N=10$). Also note that randomly selecting the sample frequencies reduces the number of samples from about

1000 to about 10, for a realistic simulation. After the coefficients A_i are calculated according to the formula

$$A_i = \sqrt{2S(\omega_i)\Delta\omega} \tag{14}$$

and wave numbers are obtained by

$$k_i = \frac{\omega_i^2}{g} \tag{15}$$

wave elevation can be computed using (13). In this formula, χ is the heading of the wavefront on the grid determined by the x and y coordinates, ϕ_i's are the uniformly distributed random phase angles for each sinusoidal wave components.

Note that (15) is valid under the assumption of infinite water depth, since the actual relation is

$$\omega_i^2 = k_i g \tanh(k_i d)$$

d being the water depth. Noting that $k_i = \frac{2\pi}{\lambda_i}$, where λ_i is the wave length of the i^{th} component, it is reasonable to use (15) to determine k_i for cases of $d/\lambda_i > 1/2$.

Wave slope, s_i, can be computed by taking the directional derivative of $\zeta(x,y,t)$ in the wavefront direction. Let $\vec{u}$ be a unit vector in the direction of the wavefront. That is,

$$\vec{u} = \vec{i}\cos(\chi) + \vec{j}\sin(\chi)$$

s is defined as

$$s(x,y,t) = D_{\vec{u}}\zeta(x,y,t) = \vec{u}\cdot\mathrm{grad}\left(\zeta(x,y,t)\right) \tag{16}$$

where $\mathrm{grad}\left(\zeta(x,y,t)\right)$, the gradient of $\zeta(x,y,t)$, is defined as

$$\mathrm{grad}\left(\zeta(x,y,t)\right) \triangleq \vec{i}\,\frac{\partial\zeta(x,y,t)}{\partial x} + \vec{j}\,\frac{\partial\zeta(x,y,t)}{\partial y}$$

Since $\zeta(x,y,t)$ can be represented in a more compact form as

$$\zeta(x,y,t) = \sum_{j=1}^{n}\sum_{i=1}^{N}\frac{1}{j}k_i^{j-1}A_i^j\cos j\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right)$$

where n represents the order of approximation,

$$\begin{aligned}\mathrm{grad}(\zeta) = {} & \vec{i}\sum_{j=1}^{n}\sum_{i=1}^{N}k_i\sin(\chi)\frac{1}{j}k_i^{j-1}A_i^j\sin j\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right) \\ & + \vec{j}\sum_{j=1}^{n}\sum_{i=1}^{N}k_i\cos(\chi)\frac{1}{j}k_i^{j-1}A_i^j\sin j\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right) \\ = {} & \left(\vec{i}\sin(\chi) + \vec{j}\cos(\chi)\right) \\ & \sum_{j=1}^{n}\sum_{i=1}^{N}\frac{1}{j}k_i^{j}A_i^j\sin j\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right)\end{aligned} \tag{17}$$

Inserting the gradient in (17) into equation (16), s can be given in a more explicit form as

$$
\begin{aligned}
s(x,y,t) &= \left(\cos(\chi)\sin(\chi)+\cos(\chi)\sin(\chi)\right) \\
&\quad \sum_{j=1}^{n}\sum_{i=1}^{N}\frac{1}{j}k_i^j A_i^j \sin j\left(\omega_i t - k_i\left(x\sin\chi + y\cos\chi\right)+\phi_i\right) \\
&= \sin(2\chi)\sum_{j=1}^{n}\sum_{i=1}^{N}\frac{1}{j}k_i^j A_i^j \sin j\left(\omega_i t - k_i\left(x\sin\chi + y\cos\chi\right)+\phi_i\right)
\end{aligned}
\tag{18}
$$

Note that (13) and (18) include frequency components as viewed from a stationary point on the sea surface, which the moving ship body is not. Therefore, we should introduce the Doppler Effect experienced by the moving body adding a correction factor that depends upon the angle between the heading of the ship and that of the wavefront (angle of encounter), and total planar velocity of the ship. These modified frequency components that the ship encounters are called the "encounter frequencies" and calculated according to

$$
\omega_{e,i} = \omega_i - k_i U\cos\left(\gamma_{wave}\right) \tag{19}
$$

where $\omega_{e,i}$ is the encounter frequency associated with the i^{th} wave component, U is the total planar velocity and γ_{wave} is the angle of encounter defined as $\gamma_{wave} \triangleq \psi - \chi$. The terminology of the sea condition according to the angle of encounter is given in Figure 11.

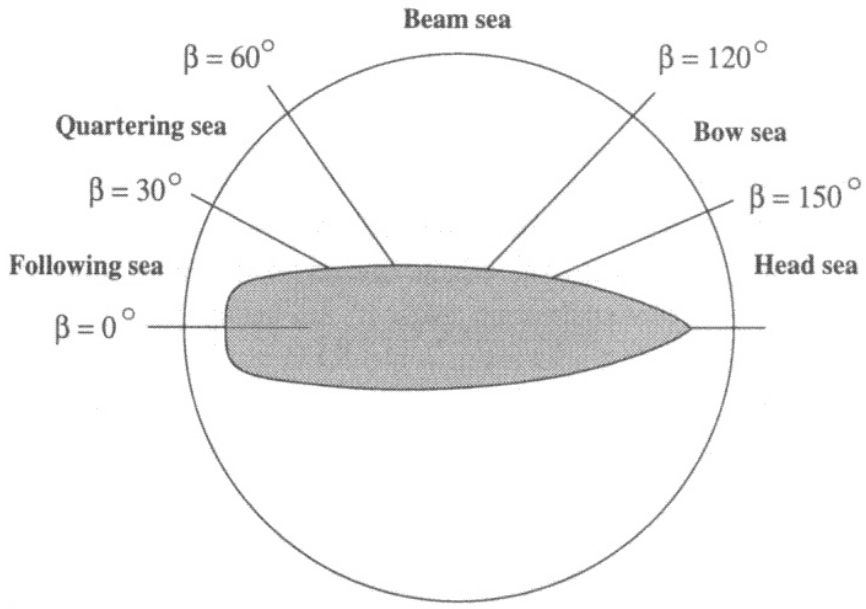

Figure 11 – Sea condition according to the wave direction, [7]

Since we have the wave elevation and the wave slope, we may calculate the wave forces acting on the hull. However, in computing these forces, we should use encounter frequencies instead of arbitrary frequency components, which means replacing the ω_i term in (13) and (18) by $\omega_{e,i}$. We may denote the new elevation of slope, which are functions of $\omega_{e,i}$ rather than ω_i, by ζ_e and Following formulas [7] give the wave forces in surge, sway and yaw motions:

$$X_{wave} = \rho g B L T \cos(\gamma_{wave}) s \tag{20.a}$$

$$Y_{wave} = -\rho g B L T \sin(\gamma_{wave}) s \tag{20.b}$$

$$N_{wave} = \sum_{i=1}^{N} \frac{1}{24} \rho g B L \left(L^2 - B^2\right) T k_i^2 \sin(2\gamma_{wave}) \zeta_{e,i} \tag{20.c}$$

where B, L, and T are breadth, length and draft of the ship respectively, ζ_i is the sum of all orders of approximation for the i^{th} frequency component, defined as

$$\zeta_{e,i} = \sum_{j=1}^{n} k_i^{j-1} A_i^j \cos j\left(\omega_i t - k_i\left(x\sin(\chi) + y\cos(\chi)\right) + \phi_i\right)$$

The wave forces vector $\boldsymbol{\tau}_{wave}$ can finally be defined as

$$\boldsymbol{\tau}_{wave} \triangleq \begin{bmatrix} X_{wave} & Y_{wave} & 0 & 0 & 0 & N_{wave} \end{bmatrix}^T$$

Wave spectral densities

There are different sources of waves such as winds, earthquakes, and tide. Figure 12 as taken from [19] shows the frequency content of ocean waves. Dashed lines show the range of frequencies for the wave generation mechanisms, and solid lines indicate that of the damping and restoring forces. Note that, in this study, we are only dealing with wind generated waves.

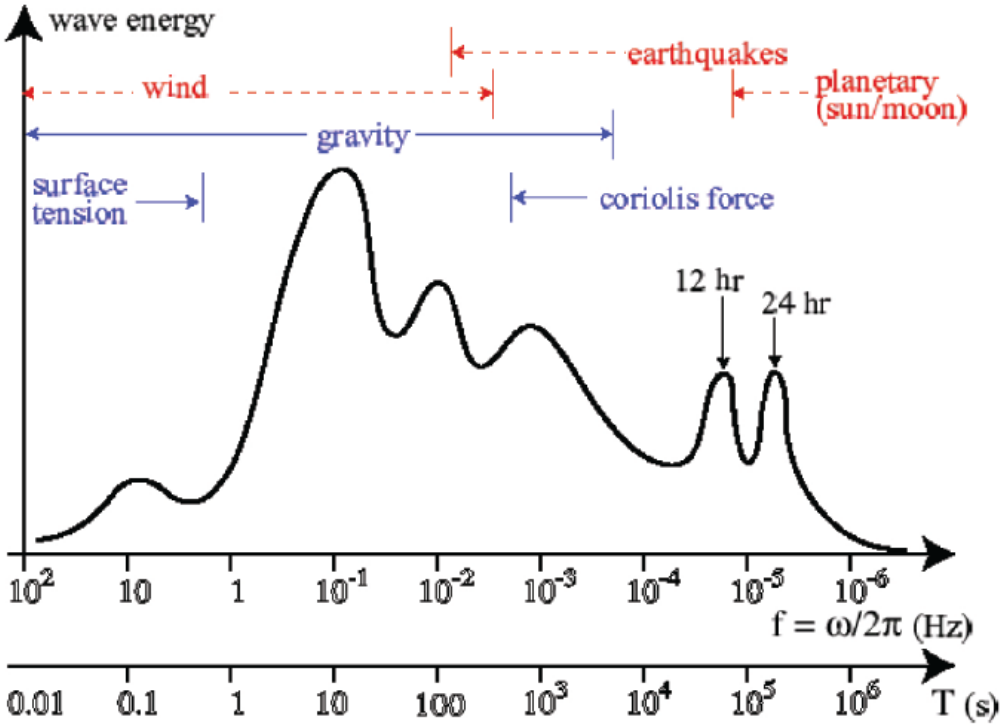

Figure 12 – Wave energy spectra for ocean waves

Power spectral densities of ocean waves are generated by use of a number of different formulations mostly based upon the statistics of data collected for a long period of time. The method of calculating the wave amplitudes using these frequency spectra is given in (14), and in this section we will analyze the spectral formulations developed and used so far [7], [19]. Before explaining the spectra one by one, we may first give some information on the factors that affect the shape of the wave spectrum, and also make the related terminology clearer.

Wind velocity and the effectiveness of the wind over the sea is an important factor sea development. Fetch, which is the distance that the wind is able to blow along over the sea surface, is the main

source of this effectiveness. Fetch limitations occur when there are physical obstacles that avoid the sea to fully develop.

Decaying waves that are the remnants of a distant storm are called swells. Their presence violates the assumption of uni-directional waves, which most of the spectrum types have.

Seafloor topographic forms determine the depth of the sea, which plays an important role in the shape of the spectrum. Note that in shallow waters, higher frequency waves are expected.

Neumann Spectrum: This one-parameter spectrum is not used in recent wave studies. However, it is of historical importance, since it is the earliest formulation derived in the literature. The spectrum is represented as

$$S(\omega) = C\omega^{-6} \exp\left(-2g^2\omega^{-2}V^{-2}\right)$$

where

$S(\omega)$: Spectral density of the wave component with frequency ω

C : Empirical constant

V : Wind velocity

Unit for the wave spectrums is $\left[m^2 s\right]$.

A typical spectrum of the form

$$S(\omega) = \frac{A}{\omega^5} \cdot e^{-\frac{B}{\omega^4}}$$

is agreed upon a few years later, and this form is used by the more recent studies.

***Pierson – Moskowitz Spectrum*:** Developed by the off-shore industry. It is a one-parameter spectrum, significant wave height, denoted by either H_s or $H^{1/3}$, being the parameter.

The spectrum is based upon the following assumptions:

- Infinite depth
- Fully developed seas
- North Atlantic data
- Unlimited fetch
- Uni-directional seas
- No swell

We have the PM spectrum defined as

$$S(\omega) = A\omega^{-5} \cdot e^{-B\omega^{-4}}$$

where

$$A = 8.1 \cdot 10^{-3} g^2$$

$$B = 0.74\left(\frac{g}{V_{19.4}}\right)^4$$

$V_{19.4}$: Wind velocity at 19.4 m above the sea level

Under the assumption that waves can be represented by a Gaussian distribution and $S(\omega)$ is narrow banded, the second coefficient may be related with the significant wave height.

$$B = 0.0323\left(\frac{g}{H_s}\right)^2 = \frac{3.11}{H_s{}^2}$$

Bretschneider Spectrum: Since the "fully developed sea" condition makes the PM Spectrum inapplicable for many cases, the two-parameter Bretschneider Spectrum that is also valid for non-fully developed seas replaced the PM Spectrum. These two parameters defining the spectrum are

ω_0 : Modal frequency

H_s : Significant wave height

Note that the definition of the modal frequency is

$$S_{\max}(\omega) = S(\omega_0)$$

The formulation for the wave spectrum is given as

$$S(\omega) = \frac{1.25}{4}\ \frac{\omega_0{}^4}{\omega^5}\ H_s{}^2\ \exp\left(-1.25(\omega_0/\omega)^4\right) \quad \left[m^2 s\right]$$

However, there are limitations on the use of this spectrum. It is valid under the following assumptions:

- Infinite water depth
- North Atlantic data
- Unlimited fetch
- Uni-directional seas
- No swell

If the following selections for the parameters of the Bretschneider's approach are made, representation reduces to that of Pierson – Moskowitz.

$$H_s = 0.21\frac{V^2}{g} \tag{21}$$

$$\omega_0 = 0.88\frac{g}{V} = 0.40\sqrt{\frac{g}{H_s}}$$

Modified Pierson – Moskowitz: The use of this spectrum is recommended by International Towing Tank Conference (ITTC) and International Ship and Offshore Structures Congress (ISSC). The spectrum is valid under the following conditions:

- Infinite depth
- Fully developed seas
- North Atlantic data
- Unlimited fetch
- No swell

$$S(\omega) = \frac{4\pi^3 {H_s}^2}{{T_z}^4 \omega^5} \exp\left(-\frac{16\pi^3}{{T_z}^4 \omega^4}\right)$$

Also note that $T_z = 0.710T_0 = 0.921T_1$. The periods $T_0 \left(= 2\pi/\omega_0\right)$, T_z and T_1 are named modal (or fundamental) period, average zero-crossings period and average wave period, respectively. (For the definitions of T_z and T_1, see [7])

JONSWAP Spectrum: The name of this spectrum is an acronym for Joint North Sea Wave Project, its use is recommended by ITTC, and it is widely used in the literature.
Assumptions made to use this spectrum are

- Non-fully developed seas
- Finite water depth
- Limited fetch

$$S(\omega) = 155\frac{H_s{}^2}{T_1{}^4\omega^5}\exp\left(-\frac{944}{T_1{}^4\omega^4}\right)(\gamma)^Y$$

where

$$Y = \exp\left[-\left(\frac{0.191\cdot\omega\cdot T_1 - 1}{\sqrt{2}\sigma}\right)^2\right]$$

$$\sigma = \begin{cases} 0.07 & \text{for } \omega \le 5.24/\mathrm{T_1} \\ 0.09 & \text{for } \omega > 5.24/\mathrm{T_1} \end{cases}$$

The relations between the other periods for this spectrum are

$$T_z = 0.710T_0 = 0.921T_1$$

An alternative representation of this spectrum is given below:

$$S(\omega)=\frac{ag^2}{\omega^5}e^{-\frac{5}{4}\left(\frac{\omega_0}{\omega}\right)^4}(\gamma)^Y$$

where

$$Y=-\frac{(\omega-\omega_0)^2}{2\sigma^2\omega_0{}^2} \qquad a=0.076\cdot\overline{x}^{(-0.22)}$$

$$\sigma=\begin{cases}0.07 & \text{for } \omega\le\omega_0\\ 0.09 & \text{for } \omega>\omega_0\end{cases} \qquad \overline{x}=\frac{gx}{U^2}$$

The γ parameter takes a value between 1 and 7. For most applications it is selected to be 3.

Ochi Spectrum: This is a three-parameter spectrum developed to account for developing seas or swell by introduction of the λ parameter. By adjusting this parameter the width of the spectrum can be altered. As λ gets smaller, a narrower spectrum for swell is achieved, and as it gets larger, spectrum gets wider to represent developing seas. The representation is as follows:

$$S(\omega)=\frac{1}{4}\frac{\left(\frac{4\lambda+1}{4}\omega_0{}^4\right)^\lambda}{\Gamma(\lambda)}\frac{H_s{}^2}{\omega^{4\lambda+1}}e^{-\frac{4\lambda+1}{4}\left(\frac{\omega_0}{\omega}\right)^4}$$

where $\Gamma(\lambda)$ is the gamma function of λ.

All spectra mentioned above are valid for single storm. That is, they are caused by one particular storm and therefore have only one peak. However, according to the Linear Wave Theory (Airy Theory), two spectra can be superimposed to represent the situation when waves from two different storms, say a storm and a swell, meet.

$$S(\omega) = S_{storm}(\omega) + S_{swell}(\omega)$$

Wind forces

For surface ocean vehicles, also the wind has to be taken into consideration, since it is an important source of influence that affects the motion of the ship. The interaction of the wind with a body is composed of two major effects, first of which is the effect of the mean velocity of the wind whose variation is slow, and the second one is the turbulent component. It should be noted that as the mass of the body increases, and the ratio of projected areas being affected by the wind to the mass decreases, the turbulent effects become less important. For a frigate as in our case, we may neglect these effects and consider only the mean velocity of the wind.
Wind forces affecting the planar motion of the ship can be given by the following expressions:

$$X_{wind} = \frac{1}{2} C_{D,x} \rho_{air} U_r^2 A_x \tag{22.a}$$

$$Y_{wind} = \frac{1}{2} C_{D,y} \rho_{air} U_r^2 A_y \tag{22.b}$$

$$N_{wind} = \frac{1}{2} C_N \rho_{air} U_r^2 A_y L \tag{22.c}$$

where $C_{D,x}$, $C_{D,y}$, and C_N are the wind coefficients related to the surge, sway and yaw motions, respectively, A_x and A_y are the areas projected along the $\vec{x}_b$ and $\vec{y}_b$ axes of the body frame, L is the length of the ship, ρ_{air} is the density of air, and finally U_r is the total velocity of the wind relative to the ship defined as

$$U_r = \sqrt{u_{r,wind}^2 + v_{r,wind}^2}$$

where

$$u_{r,wind} = V_{wind}\cos(\gamma_{wind}) - u$$
$$v_{r,wind} = V_{wind}\sin(\gamma_{wind}) - v$$

in the lack of ocean currents. To include the current effects in the above equations, the velocity terms u and v may simply be replaced by $u - u_C$ and $v - v_C$, respectively, where the subscript C denotes the current velocity. γ_{wind} is the angle between the wind direction and the direction of the ship motion defined as $\gamma_{wind} \triangleq \chi - \psi$. In case of unidirectional seas $\gamma_{wind} = -\gamma_{wave}$ by definition.

Note that the coefficients $C_{D,x}$, $C_{D,y}$, and C_N are also functions of γ_w. These data may be given in a tabulated form based on experiments, or may be approximated as

$$C_{D,x} = C'_{D,x}\cos(\gamma_{wind})$$
$$C_{D,y} = C'_{D,y}\sin(\gamma_{wind})$$
$$C_N = C'_N\sin(2\gamma_{wind})$$

where $C'_{D,x}$, $C'_{D,y}$, and C'_N are empirical constants.

The wind forces vector $\boldsymbol{\tau}_{wind}$ can be defined similar to the wave forces vector as

$$\boldsymbol{\tau}_{wind} \triangleq \begin{bmatrix} X_{wind} & Y_{wind} & 0 & 0 & 0 & N_{wind} \end{bmatrix}^T$$

In this chapter, the mathematical model that describes the motion of the ship under the influence of hydrodynamic, environmental and control forces is studied. The implementation issues of this theoretical model to software will be discussed in the next chapter.

Current forces

The effects of the ocean currents are introduced to the system by defining a relative velocity vector of the ship with respect to the ocean current. That is,

$$\vec{\mathbf{v}}_{B,rel} = \vec{\mathbf{v}}_B - \vec{\mathbf{v}}_C$$

where $\vec{\mathbf{v}}_C$ is the current velocity, whose expression in $\mathcal{F}_O$ is simply

$$\overline{\mathbf{v}}_C^{(O)} \triangleq \begin{bmatrix} u_C & v_C & w_C \end{bmatrix}$$

which can also be expressed in the body-frame as

$$\begin{aligned} \overline{\mathbf{v}}_C^{(b)} &\triangleq \begin{bmatrix} u'_C & v'_C & w'_C \end{bmatrix} \\ &= \boldsymbol{J}_1^{-1} \overline{\mathbf{v}}_C^{(O)} \end{aligned}$$

Assuming also that the current is irrotational, i.e. $\boldsymbol{v}_C = \begin{bmatrix} \overline{\mathbf{v}}_C^{(b)T} & 0 & 0 & 0 \end{bmatrix}^T$, Fossen has shown [7] that the equations of motion (11) and (4) can be modified to reflect the ocean currents' effect as

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{D}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{prop} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}'_{env} \tag{23}$$

where $\boldsymbol{\tau}'_{env} = \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind}$, $\boldsymbol{v}_r = \boldsymbol{v} - \begin{bmatrix} \overline{\mathbf{v}}_C^{(b)T} & 0 & 0 & 0 \end{bmatrix}^T$, and $\boldsymbol{\tau}_{current} = \boldsymbol{\tau}_{current}(\boldsymbol{v}_C)$ is integrated into the left hand side of the equation by replacing $\boldsymbol{v}$ by $\boldsymbol{v}_r$. Since the current forces are assumed to be constant or slowly varying, i.e. $\dot{\boldsymbol{v}}_C \cong 0$, we can also replace $\dot{\boldsymbol{v}}$ by $\dot{\boldsymbol{v}}_r$, and the equation of relative motion finally becomes

$$\boldsymbol{M}\dot{\boldsymbol{v}}_r + \boldsymbol{C}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{D}(\boldsymbol{v}_r)\boldsymbol{v}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{prop} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}'_{env} \tag{24}$$

Since the velocity transformation (4) does not change, the relative velocity term should be used in this equation as follows:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta}_2)(\boldsymbol{v}_r + \boldsymbol{v}_C)$$

Since, $\boldsymbol{J}(\boldsymbol{\eta}_2)\boldsymbol{v}_C = \begin{bmatrix} \overline{\mathbf{v}}_C^{(O)T} & 0 & 0 & 0 \end{bmatrix}^T$, the equation that transforms the body-fixed velocities into the rates of earth-fixed coordinates and Euler angles become

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta}_2)\boldsymbol{v}_r + \begin{bmatrix} \overline{\mathbf{v}}_C^{(O)T} & 0 & 0 & 0 \end{bmatrix}^T \tag{25}$$

CHAPTER 4

IMPLEMENTATION OF THE SIMULATION CODE

4.1. Development Environment

Matlab® is selected as the development environment in the implementation process of the mathematical model derived. The final code that is to be integrated to the simulation loop of NSTMSS is required to be in C++ for compatibility purposes, since rest of the original code is written in this language. However, the tools provided by such as the graphing facilities, and the ease of coding and result evaluating makes it a useful platform to develop the simulation routine. Furthermore, the control design for the steering autopilot makes Matlab® an almost no-alternative choice.

This chapter will mainly focus on the content of the algorithm and how the individual program components interact with each other. Since the algorithm development process is independent of the tool/language used in the process, the issues regarding the integration with the available software will be discussed separately.

4.2. The Algorithm

The algorithm is designed in a fashion that is as modular as possible, therefore making the investigation of the tasks performed by different program components easier, since they can be treated separately. This way, their interactions with each other and with the main program (input/output relationships) can also be handled without considering the internal dynamics of each component.
The program flow can be studied in two steps:

- Initialization routine
- Simulation loop

4.2.1. Initialization Routine

When the main program is started, this routine is invoked first to retrieve data such as ship parameters, environmental conditions and information that is related to the solution method of the problem. Furthermore, some time-taking operations that would reduce the volume of work done per cycle when the loop is started are performed at this stage. In this section, these steps are explained in detail.

Parameter acquisition

Environment Parameters: A scenario of environmental conditions is introduced to be used by the motion equations. These parameters include the wind velocity and direction, water density, magnitude and direction of the ocean current velocity and gravitational acceleration. The user may or may not indicate the wind ve-

locity directly. A frequently used way to describe the wind speed is the Beaufort Scale which is characterized by the formula

$$V_{wind} = 0.836B^{3/2} \tag{26}$$

Note that the wind velocity is given in meters per second.
In Table 3 the velocity intervals in different units are given with their corresponding Beaufort numbers. Note that the mean velocities, also given in the table, are calculated via (26).

Table 3 – Beaufort Scale

Beaufort number	Wind velocity				Mean wind velocity (kt / kph / mph)	Description	Wave height	
	kt	kph	mph	m/s			m	ft
0	0	0	0	0-0.2	0 / 0 / 0	Calm	0	0
1	1-3	1-6	1-3	0.3-1.5	2 / 4 / 2	Light air	0.1	0.33
2	4-6	7-11	4-7	1.6-3.3	5 / 9 / 6	Light breeze	0.2	0.66
3	7-10	12-19	8-12	3.4-5.4	9 / 17 / 11	Gentle breeze	0.6	2
4	11-15	20-29	13-18	5.5-7.9	13 / 24 / 15	Moderate breeze	1	3.3
5	16-21	30-39	19-24	8.0-10.7	19 / 35 / 22	Fresh breeze	2	6.6
6	22-27	40-50	25-31	10.8-13.8	24 / 44 / 27	Strong breeze	3	9.9
7	28-33	51-62	32-38	13.9-17.1	30 / 56 / 35	Near gale	4	13.1
8	34-40	63-75	39-46	17.2-20.7	37 / 68 / 42	Gale	5.5	18
9	41-47	76-87	47-54	20.8-24.4	44 / 81 / 50	Strong gale	7	23
10	48-55	88-102	55-63	24.5-28.4	52 / 96 / 60	Storm	9	29.5
11	56-63	103-119	64-73	28.5-32.6	60 / 112 / 70	Violent storm	11.5	37.7
12	64-80	120+	74-95	32.7-40.8	73 / 148 / 90	Hurricane	14+	46+

The last two columns in Table 3 are the significant wave heights related to the mean velocities through the equation (21).

A second indirect method to determine the wind velocity and the significant wave height is by using the sea state codes proposed by World Meteorological Organization (*see* Table 4).

Table 4 – Sea state codes and corresponding wave heights

Code Figure	Descriptive terms	Height in meters
0	Calm (glassy)	0
1	Calm (rippled)	0-0.1
2	Smooth (wavelets)	0.1-0.5
3	Slight	0.5-1.25
4	Moderate	1.25-2.5
5	Rough	2.5-4
6	Very rough	4-6
7	High	6-9
8	Very high	9-14
9	Phenomenal	14+

The wind velocity to achieve the wave height given in Table 4 can be found by equation (21).

To summarize, the user may indicate the Beaufort number, the sea state code, or the wind velocity itself to introduce the wind velocity and the significant wave height to the system. If indirect indication is used, mean values of the ranges given for a code is taken as the velocity or height.

This step also includes reading of the initial states, if any, from the file. The user may desire to start the simulation with nonzero initial conditions. If not stated explicitly, the ship is assumed to be at rest with all coordinates and Euler angles equal to 0, except the vertical

position, z, which is equal to z_{eq}. That is, $\mathcal{F}_b$ is in the same orientation with $\mathcal{F}_O$, and offset in $\vec{z}$ axis z_{eq} much.

Vessel Model Parameters: Physical properties of the ship such as mass, inertia matrix and length, and hydrodynamic coefficients are input to the system via a vessel parameters file. Physical parameters of the vessel are listed below in Table 5.

Table 5 – Physical parameters of the vessel

Parameter Name	Symbol	Variable Name	Unit
Mass	m	M	kg
Inertia matrix	$\boldsymbol{I}_B$	I_B	kgm^2
Displaced water volume	∇	V_disp	m^3
Length	L	L	m
Breadth	B	B	m
Draft	T	T	m
Transverse metacentric height	$\lvert GM_T \rvert$	GM_T	m
Longitudinal metacentric height	$\lvert GM_L \rvert$	GM_L	m
CG location	$\boldsymbol{r}_G$	r_G	m

Hydrodynamic coefficients are also in this file and they are grouped according to their multiplicands. First group consists of the linear added inertia terms. Their variable names are of the form <Force><velocity>d. For instance, the force coefficient contributing to the force in the surge direction proportional to the acceleration again in the surge direction, which is $X_{\dot{u}}$, can be named as "Xud". "d" at the end of the variable name stands for the over-dot operator of the subscript.

Second group covers the linear, quadratic, and in some cases third order damping terms. Naming is similar to that of the added inertia coefficients. That is, for linear terms, <Force><velocity>, and for quadratic terms <Force>a<absolute velocity><2nd velocity>. In this context, the variable names for the damping terms X_u and $X_{|u|u}$ are defined as "Xu" and "Xauu".

Restoring force coefficients, which are functions of η_2 are given in group 3. The multiplicands of these coefficients may not be the Euler angles themselves as given in the subscript, but a trigonometric function of these angles as explained in the "Restoring forces" section. In such a case, the coefficient will most probably be calculated, rather than given directly. If the coefficients are multiplied by the Euler angles themselves, variable names are in the <Force><Euler angle><Euler angle>… form. However, when trigonometric functions are involved, variable name format for the coefficients will be <Force>t<Euler angle>, where t stands for trigonometric.

According to the model and the data set available, the control force coefficients may also be given in a separate group. These forces will be functions of the rudder deflection and some other variables. For example, the coefficient $X_{\delta\delta\delta u}$ is represented by "Xdddu" in the code.

Finally, the rest of the available experimental parameters that do not belong to any of these four groups are given in a fifth group. Note that these coefficients are obtained as a result of the system

identification procedures and may require additional models to be explained.

Internal parameters: Information regarding the numerical solution method of equations of motion, degrees of freedom that the mathematical model is able to use (due to the data set available), controller gains etc. are input to the simulation via an "internal parameters file".

Wave model base data generation

Second task to be accomplished during the initialization phase following the parameter input is generating arrays that would help evaluate the wave elevation and the wave forces more efficiently at each cycle of the simulation loop, once it is invoked.
A flow chart explaining the wave force calculation is given in Figure 13:

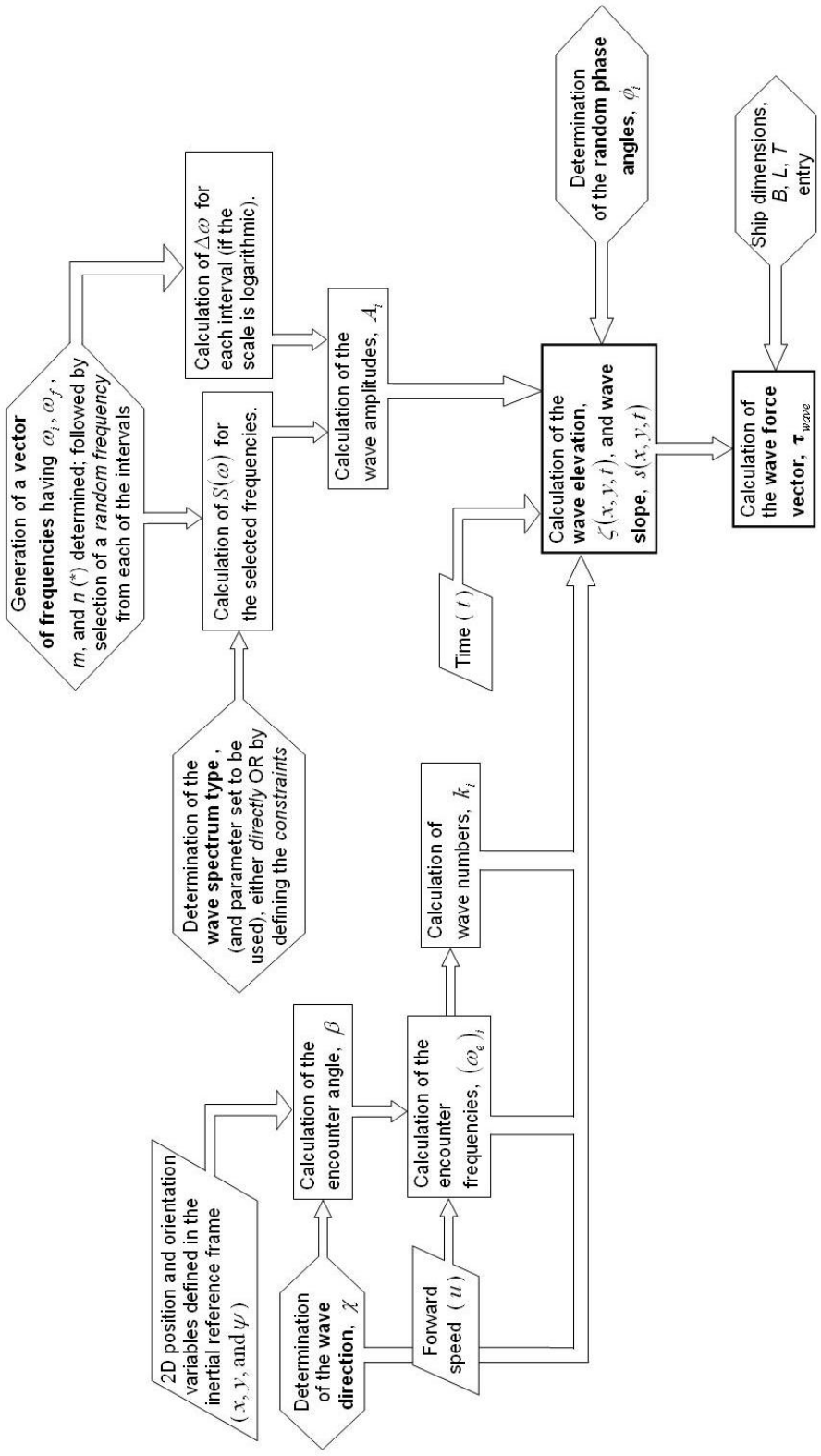

Figure 13 – Wave force calculation flow chart

The above wave model algorithm is quite "heavy" in terms of processing effort, therefore is not suitable for a real time application to evaluate the wave effects online without any modification. Therefore, the internal parameters of the wave model those remain unchanged throughout the simulation, provided that physical conditions determining the wave characteristics such as wind speed and/or significant wave height are constant, may be calculated before the simulation loop and stored for use at every cycle of the real-time integration loop. These internal parameters are amplitude, frequency, wave number and phase angle for each wave component. Once these parameters are stored, they can be used in the simulation repeatedly without any recalculation.

Furthermore, it should be noted that there are two reasons why a wave model is required for the simulation. First of these is to include the effects of the ocean waves on the dynamics of the ship in the simulation, as mentioned so far. The other use of the wave model is for visual purposes. Rendering an ocean surface image at every frame is realized by the elevation data, which is the output of this model. Remember that the wave elevation for a specific point $P(x, y)$ at time t is given in (13).

Given a specific spectral formulation, a grid of planar coordinates and a time interval, the above formula is sufficient to generate input for the visualization of the ocean surface. The wave elevation calculated over 2 dimensional space and time as a result of this function is the *actual* wave elevation, and none of the inputs to this function are dependent on the current state of the ship.

However, it is explained that the wave forces acting on the ship depend on the velocity and the heading of the ship implicitly through the modification of wave components used in computing the wave elevation. This modification of frequency components into encounter frequencies was given in (19). In order to reduce the computation time taken by each cycle in the simulation loop, elevation data may be calculated as a 3 dimensional (x, y, and t) array once before the loop, stored and accessed as required. The problem that may arise in such a procedure is that the amount of space occupied on the memory or on the disk may become too large to handle. Since the wave motion is a repeated one, this problem can be solved by truncating the time dimension at a period which is an integer multiple of the fundamental period, T_0, and using this data periodically. Note that fundamental period is the period of the sinusoidal wave component with the highest amplitude in the spectrum, and selection of the repetition period this way is to avoid apparent discontinuities in wave motion. All wave components except the one with the highest amplitude will certainly have discontinuities along the time scale.

In order to use this method effectively, a fixed time step integrator should be selected, since the elevation data stored and to be accessed later in the loop are uniformly spaced in terms of time intervals; otherwise interpolation should be performed to obtain the elevation data at non-uniform time steps resulted from variable time-step integration, which means extra computational load for the cycle.

However, we know that when the wave elevation function is used to obtain the wave forces on the ship hull, the component frequencies, ω_i, are replaced by the encounter frequencies, $\omega_{e,i}$, whose definition is given in (19).

This situation necessitates the computation of the wave elevation modified according to the frequencies of encounter for each cycle. Instead of calculating wave elevation (and also the wave slope) to obtain the wave forces on the ship starting from scratch, elevation data computed before the simulation loop based on the random component frequencies may be used with some correction performed by feeding back the states of the ship at each time interval. This process is explained below:

Let,

$$\alpha_i \triangleq \omega_i t - k_i\left(x\sin\chi + y\cos\chi\right) + \phi_i \text{ and } \beta_i \triangleq k_i U\cos\left(\gamma_{wave}\right)t$$

It is known that the wave elevation used for the calculation of wave forces is

$$\begin{aligned}\zeta_e\left(x,y,t\right) = &\sum_{i=1}^{N} A_i\cos\left(\omega_{ei}t - k_i\left(x\sin\chi + y\cos\chi\right) + \phi_i\right)\\ &+\sum_{i=1}^{N}\frac{1}{2}k_i A_i^{\,2}\cos 2\left(\omega_{ei}t - k_i\left(x\sin\chi + y\cos\chi\right) + \phi_i\right) + f\left(A_i^{\,3}\right) + \ldots\end{aligned} \tag{27}$$

Then (27) can be expressed in terms of α_i and γ_i for simplicity as

$$\zeta_e(x,y,t) = \sum_{i=1}^{N} A_i \cos(\alpha_i - \beta_i) + \sum_{i=1}^{N} \frac{1}{2} A_i^2 \cos 2(\alpha_i - \beta_i) + f\left(A_i^3\right) + \ldots$$

Expanding each cosine expression,

$$\begin{aligned}
\zeta_e(x,y,t) &= \sum_{i=1}^{N} A_i \left[\cos(\alpha_i)\cos(\beta_i) + \sin(\alpha_i)\sin(\beta_i)\right] \\
&\quad + \sum_{i=1}^{N} \frac{1}{2} k_i A_i^2 \left[\cos(2\alpha_i)\cos(2\beta_i) + \sin(2\alpha_i)\sin(2\beta_i)\right] \\
&\quad + f\left(A_i^3\right) + \ldots \\
&= \sum_{i=1}^{N} \cos(\beta_i) A_i \cos(\alpha_i) + \sum_{i=1}^{N} \sin(\beta_i) A_i \sin(\alpha_i) \\
&\quad + \sum_{i=1}^{N} \cos(2\beta_i) \frac{1}{2} k_i A_i^2 \cos(2\alpha_i) + \sum_{i=1}^{N} \sin(2\beta_i) \frac{1}{2} k_i^2 A_i^2 \sin(2\alpha_i) \\
&\quad + f\left(A_i^3\right) + \ldots
\end{aligned}$$

we get

$$\zeta_e(x,y,t) = \sum_{j=1}^{n} \sum_{i=1}^{N} \left[\cos(j\beta_i) \frac{1}{j} k_i^{j-1} A_i^{\,j} \cos(j\alpha_i) + \sin(j\beta_i) \frac{1}{j} k_i^{j-1} A_i^{\,j} \sin(j\alpha_i)\right] \quad (28)$$

Once we calculate

$$\frac{1}{j} k_i^{j-1} A_i^{\,j} \cos(j\alpha_i) \text{ and } \frac{1}{j} k_i^{j-1} A_i^{\,j} \sin(j\alpha_i) \quad (i = 1..N \text{ and } j = 1..n),$$

where N is the number of selected frequencies, and n is the order of approximation for the wave model, before the simulation loop is started, all that remains to be evaluated at each cycle are the correction terms for the frequency shift due to speed of the ship and

angular difference between the ship's heading and the wavefront. It is important to note that each of these terms are functions of time and space. Together with the frequency (i) and order (j) indices, these terms become 5 dimensional arrays. It should be kept in mind that storing the wave elevation without correction, i.e. $\zeta(x,y,t)$, for visualization also saves time for the real-time cycles. The only remaining terms to be evaluated online are

$$\cos(j\beta_i) \text{ and } \sin(j\beta_i)$$

which are 2 dimensional arrays of frequency (due to the wave number term in β_i), and order of approximation, since time (t) and ship position (x,y) information are unique for a given time step during simulation.

To summarize the procedure, if

- **i.** The wave model uses N frequency intervals, and
- **ii.** Assumes an expansion up to the n^{th} order for the wave motion,
- **iii.** Fixed step integrator divides the repetition period into tn time steps, and
- **iv.** The simulation is run on a grid of $xn \times yn$ points,

an array defined as

$$C_\zeta \triangleq \left[C_{\zeta,ji(xi)(yi)(ti)}\right] = \left[\frac{1}{j}k_i^j A_i^j \cos(j\alpha_i)\right] \tag{29}$$

with the variable name `Czeta`, and its sine counterpart as

$$S_\zeta \triangleq \left[S_{\zeta,ji(xi)(yi)(ti)}\right] = \left[\frac{1}{j} k_i^j A_i^j \sin\left(j\alpha_i\right)\right] \tag{30}$$

with the variable name `Szeta` should be stored. Index notation is used in (29) and (30), and the indices are defined as $i = 1 \ldots N$, $j = 1 \ldots n$, $xi = 1 \ldots xn$, $yi = 1 \ldots yn$, and $ti = 1 \ldots tn$. Therefore, C_ζ and S_ζ are of size $n \times N \times xn \times yn \times tn$; By utilizing these arrays only $2 \times n \times N$ trigonometric functions are evaluated and that many multiplications are performed at each time step during the simulation loop. Note that the order of the wave number terms in (29) and (30) are 1 more than that in the components required to calculate $\zeta_e(x, y, t)$ as given in (28). This is because the wave elevation function is never used by itself in computation of the wave forces (see (20.a), (20.b), and (20.c)). The use of these arrays is going to be made clearer in section 4.2.2, where the loop tasks are explained.
Furthermore, an array of the wave elevation itself, $\zeta(x, y, t)$ (variable name: `Zeta`), which is of size $xn \times yn \times tn$ is stored as well for the rendering of the sea surface for visual purposes.

Note that the wave slope may also be calculated and stored as bulk data just like the wave elevation as explained above. However, since its formulation is very similar to that of the wave elevation, a second approach is to generate the wave slope online, by using the elevation data and wave numbers vector only.

Starting from (18), the wave slope function to be used for the force calculations can be given as

$$s_e(x,y,t) = \sin(2\chi)\sum_{j=1}^{n}\sum_{i=1}^{N}\left[k_i\left(\cos(j\beta_i)\frac{1}{j}k_i^{j-1}A_i^{\,j}\cos(j\alpha_i) + \sin(j\beta_i)\frac{1}{j}k_i^{j-1}A_i^{\,j}\sin(j\alpha_i)\right)\right] \tag{31}$$

Vessel and environmental model initialization

This part of the initialization routine deals with the preprocessing of the vessel parameters, and getting them ready for the main loop. First step is the formation and inversion of the overall inertia matrix. With the assumption of constant mass and constant CG location, we have a constant mass matrix, i.e. $\dot{\boldsymbol{M}} = \boldsymbol{0}$. Following the summation of the rigid-body and added inertia matrices to form the overall inertia matrix, $\boldsymbol{M}$, it can be inverted once before the loop, and used throughout the simulation (see section 4.2.2 for the use of $\boldsymbol{M}^{-1}$).

If the available hydrodynamic parameter set is a non-dimensional one, they should be dimensionalized before use in the simulation. Non-dimensionalization system is indicated at the beginning of the vessel parameters file with the variable ndSys. If the hydrodynamic coefficients are already dimensional, then this task is aborted and they are used directly.

As a final task under this step, different force terms using constant coefficients made up of several constants are initialized for faster

calculation inside the loop. Wave forces given in (20.a), (20.b), and (20.c) can be rewritten collecting the constant terms

$$X_{wave} = C_{wave,X} \cos(\beta) s \tag{32.a}$$

$$Y_{wave} = C_{wave,Y} \sin(\beta) s \tag{32.b}$$

$$N_{wave} = C_{wave,N} \sin(2\beta) \sum_{i=1}^{N} k_i^2 \zeta_{e,i} \tag{32.c}$$

where

$$C_{wave,X} \triangleq \rho g B L T$$

$$C_{wave,Y} \triangleq -\rho g B L T$$

$$C_{wave,N} \triangleq \frac{1}{24} \rho g B L \left(L^2 - B^2\right) T$$

Similarly for the wind and restoring forces, we may multiply all the constants in the equations (22.a), (22.b), (22.c), and (10) before the loop, and calculate the forces by updating the angle and the velocity during the loop. Note here that since the wind and the wave direction are accepted to be the same with the assumption of unidirectional seas, the relative angles β and γ_w become identical.

Ocean current, which is assumed to be 2 dimensional, is defined by a velocity magnitude (V_C) and an Azimuth angle (β_C). These parameters are converted into velocity components in $\mathcal{F}_O$ as

$$u_C = V_C \cos(\beta_C)$$

$$v_C = V_C \sin(\beta_C)$$

4.2.2. Simulation Loop

Once the initialization phase of the program is completed, the real-time simulation loop is started to compute the states of the ship and send them to the requesting simulation system for visualization of the current state of the ship.

As a matter of fact, the frame refresh rate that is controlled by the simulation system is not the same with the integration rate (that is number of time-steps per unit time) dictated by the numerical integrator. Therefore, embedding the computation loop into the main loop which also controls other visual actions of the program, so as to allow one cycle of the integration process to be run and the obtained results to be immediately displayed on the screen, is not a feasible method. One solution to this problem is selecting the frequency of the integrator an integer multiple of that of the frame refreshing loop, so that the integrator runs a predefined number of times to obtain a solution that will be displayed. However, a more flexible method that would allow freely selecting each of the two time steps is completely decoupling the processes of state computation and frame refreshing from each other. The details of this method are given in Section 4.4 where the interaction of the equation of motion solution algorithm with the simulation system is explained.

The equations of motion in the body frame, (24), requires some modifications to be ready for the numeric solution. In fact, the set of first order differential equations that make up the initial value problem to be solved are this modified form of (24), and (25) itself:

$$\dot{\boldsymbol{v}}_r = \boldsymbol{M}^{-1}\left(\boldsymbol{\tau}_{prop} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}_{env} - \boldsymbol{C}\left(\boldsymbol{v}_r\right)\boldsymbol{v}_r - \boldsymbol{D}\left(\boldsymbol{v}_r\right)\boldsymbol{v}_r - \boldsymbol{g}\left(\boldsymbol{\eta}\right)\right) \triangleq \boldsymbol{f}_1\left(\boldsymbol{v}_r, \boldsymbol{\eta}, t\right) \quad (33.a)$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}\left(\boldsymbol{\eta}_2\right)\boldsymbol{v}_r + \begin{bmatrix} \overline{\mathbf{v}}_C^{(O)T} & 0 & 0 & 0 \end{bmatrix}^T \triangleq \boldsymbol{f}_2\left(\boldsymbol{v}_r, \boldsymbol{\eta}, t\right) \quad (33.b)$$

Note that $\boldsymbol{v}_0 = \boldsymbol{v}\left(t = t_0\right)$ and $\boldsymbol{\eta}_0 = \boldsymbol{\eta}\left(t = t_0\right)$ are given.

Numerical integration

In order to obtain the states at time $t = t_{i+1}$, provided that those at time $t = t_i$ are available, the following procedure is employed if Euler's method is used for integration:
Letting the overall state vector be

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\eta} \end{bmatrix}$$

The value of the state vector at $t = t_{i+1}$, i.e. $\boldsymbol{x}_{i+1}$ can be found as

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{f}\left(\boldsymbol{x}, t\right) \cdot \Delta t$$

where

$$\boldsymbol{x}_{i+1} \triangleq \boldsymbol{x}\left(t = t_{i+1}\right)$$
$$\boldsymbol{x}_i \triangleq \boldsymbol{x}\left(t = t_i\right)$$
$$\Delta t \triangleq t_{i+1} - t_i$$
$$\boldsymbol{f}\left(\boldsymbol{x}, t\right) = \begin{bmatrix} \boldsymbol{f}_1\left(\boldsymbol{v}, \boldsymbol{\eta}, t\right) \\ \boldsymbol{f}_2\left(\boldsymbol{v}, \boldsymbol{\eta}, t\right) \end{bmatrix}$$

Another integrator that is widely used for numerical solution of differential equations is the Runge-Kutta fourth order, shortly recognized as "RK4".

Formulation to obtain $\boldsymbol{x}_{i+1}$ using this method is as follows [6]:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

where

$$k_1 = \boldsymbol{f}\left(\boldsymbol{x}_i, t_i\right)$$
$$k_2 = \boldsymbol{f}\left(\boldsymbol{x}_i + \frac{\Delta t}{2}k_1, t_i + \frac{\Delta t}{2}\right)$$
$$k_3 = \boldsymbol{f}\left(\boldsymbol{x}_i + \frac{\Delta t}{2}k_2, t_i + \frac{\Delta t}{2}\right)$$
$$k_4 = \boldsymbol{f}\left(\boldsymbol{x}_i + \Delta t k_3, t_i + \Delta t\right)$$

According to the numerical integrator type selected in the internal parameters file, one of these methods is employed in the solution of the equations of motion by invoking the related function.

Program flow

This section will cover the actions taken by the program during one cycle of the real-time simulation loop. It is worth mentioning before going into the details of these actions that the main software performs tasks other than the solution of the ship dynamics, such as rendering the results of the solution to the screen as an output. Therefore, only a limited computation capacity is left for the solu-

tion of the motion equations, which means this solution process should be much faster than real-time.

1. Input acquisition: As the first action in the loop, the commands input by the user are read in. These are the commanded values of the shaft speed(s) (n_c) and the rudder deflection(s) (δ_c). A second alternative as a steering input to the system is the desired heading (ψ_c) rather than the commanded rudder deflection. However, this input can be used in the presence of a closed loop control system that accounts for the error between the actual heading (ψ) and the desired one.

2. Trigonometric function evaluation: In the transformation matrix and the restoring forces, the trigonometric functions of the Euler angles are used. For the sake of computational efficiency, each of the trigonometric functions of the Euler angles in the transformation matrices should be computed only once in each cycle of the simulation loop. Details of this process and performance gain is investigated in the next chapter where the tests on the program components are discussed.

3. Coriolis and hydrodynamic force calculations: The terms on the right hand side of (33.a) inside the parentheses except the environmental disturbances are evaluated in this step. Coriolis and centripetal, damping, restoring, control and propelling forces are evaluated according to the models proposed in Chapter 3.
However, in the presence of a well-defined model based on experimental data, the format of the coefficients may vary. Therefore, in such a case the formal representations of the force terms men-

tioned above are by-passed, and experimental coefficients and force definitions are used instead. For instance, the right hand side terms may include environmental disturbances, rigid body Coriolis and centripetal effects, and propelling effort separately, while treating all other external forces on the ship together as functions of velocities, Euler angles and rudder deflection, in each direction of motion. As the forces under consideration are highly coupled functions of these variables, they are given as Taylor series expansions, an example of which can be seen in (7). The RPMM model of the Multipurpose Naval Vessel in [12] is an example of such a model.

The multiplicands of these coefficients, which are combinations of motion variables and control inputs, are calculated and stored within the current cycle of the loop before getting multiplied by the corresponding coefficients so as to minimize the number of operations per cycle. The effect of this procedure on the performance of the code is discussed in the following chapter.

4. Wave force vector calculation: Wave forces are calculated by the help of the base data generated in the initialization phase. Since the current position of the ship and the time are known, the cosine and sine arrays given in (29) and (30) can be reduced to a dimension of 2 as a first step. This is done by accessing the 2-dimensional arrays at $C_{\zeta}(xi, yi, ti)$ and $S_{\zeta}(xi, yi, ti)$. Let these arrays be defined as

$$\begin{aligned} C_{\zeta(n\times N)} &\triangleq C_{\zeta}(xi, yi, ti) \\ S_{\zeta(n\times N)} &\triangleq S_{\zeta}(xi, yi, ti) \end{aligned} \tag{34}$$

And the correction terms $\left[\cos\left(j\gamma_i\right)\right]$ and $\left[\sin\left(j\gamma_i\right)\right]$, where the square brackets denote the index notation for arrays, are also evaluated and ready to use as $n\times N$ arrays.

Using (31) and the array definitions in (34), the online computation of s_e inside the loop is performed as

$$s_e\left(x,y,t\right)=\sin\left(2\chi\right)\sum_{j=1}^{n}\left[C_{\zeta(n\times N)}\circ\left[\cos\left(j\beta_i\right)\right]+S_{\zeta(n\times N)}\circ\left[\sin\left(j\beta_i\right)\right]\right]$$

where "$\circ$" operator denotes term-by-term multiplication of equal sized matrices, rather than ordinary matrix multiplication. Then the wave forces X_{wave} and Y_{wave} can be calculated by (32.a) and (45.b). And for the calculation of the yaw moment caused by the waves, N_{wave}, we may use (45.c), where the summation term $\sum_{i=1}^{N}k_i^2\zeta_{e,i}$ is computed according to

$$\sum_{i=1}^{N}k_i^2\zeta_{e,i}=\sum_{j=1}^{n}\left\{\left[C_{\zeta(n\times N)}\circ\left[\cos\left(j\beta_i\right)\right]+S_{\zeta(n\times N)}\circ\left[\sin\left(j\beta_i\right)\right]\right]_{(n\times N)}\cdot\boldsymbol{k}_{(N\times 1)}\right\}$$

Note that the vector of wave numbers, $\boldsymbol{k}$, is defined as

$$\boldsymbol{k}\triangleq\begin{bmatrix}k_1 & k_2 & \dots & k_N\end{bmatrix}^T$$

5. Wind force calculation: Wind forces are calculated according to (22.a), (22.b), and (22.c), also having the constant coefficients computed in the initialization step before going into the loop.

6. Multiplication by $\boldsymbol{M}^{-1}$***:*** Once all of the force terms are calculated and added up to form a column vector of size equal to the degree of freedom of the model, this vector is pre-multiplied by the $\boldsymbol{M}^{-1}$ to get the velocity derivatives, i.e.

$$\dot{\boldsymbol{v}}_r = \boldsymbol{M}^{-1}\boldsymbol{f}_1\left(\boldsymbol{v},\boldsymbol{\eta},t\right)$$

operation is performed.

7. Integration of the body fixed velocities: According to the numerical integrator type and the time step indicated in the internal parameters file, the velocities are updated from their previous values to current values. State derivatives obtained in step 6 are employed in this process.

8. Velocity transformation: Current values of the velocities as expressed in the body-fixed reference frame are transformed into the rates of earth-fixed coordinates and Euler angles using the equation (33.b). In the implementation process of this relation, a planar ocean current, i.e. $w_C = 0$, assumption is a reasonable one. By also utilizing this simplification, velocity transformation process simply becomes

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}\left(\boldsymbol{\eta}_2\right)\boldsymbol{v}_r + \begin{bmatrix} u_C \\ v_C \\ \boldsymbol{0}_{4\times 1} \end{bmatrix}$$

9. Integration in the earth-fixed frame: The transformed velocities and Euler angle rates are integrated to get the current values of the earth-fixed coordinates and the Euler angles as the last step of the simulation loop.

After the cycle is completed and the variables required by the main program are calculated, the loop starts from the beginning. Note that, as mentioned earlier, the variables are passed to the main program only when the visual interface is required to be refreshed.

4.3. Adaptation to Lower DOF Models

Since full hydrodynamic data sets are not available all the time, simpler, lower degree-of-freedom simulations may be required for different purposes, the simulation algorithm explained so far should be adaptable to such cases.

The most significant problem that arises in this adaptation process is the matrix inversion operation applied for the constant mass matrix, $\boldsymbol{M}$. Let $\boldsymbol{M}'$ be the lower DOF mass matrix obtained by deletion of the rows and columns corresponding to the missing degrees of freedom. The algorithm is modified to handle these cases so that the same framework developed for a full DOF model can be used for models described by lower DOF data sets without the loss of generality. The following steps are followed for this purpose:

1. Missing degrees of freedom are entered via the Vessel Parameters input file.

2. The function that forms and inverts the inertia matrix takes this information as an input argument and deletes the rows and columns corresponding to missing degrees of freedom. Then inverts this matrix to obtain $\boldsymbol{M}'^{-1}$. Both $\boldsymbol{M}'$ and $\boldsymbol{M}'^{-1}$ are augmented by adding rows and columns of zeros correspond-

ing to the missing DOFs. For instance, let the missing degrees of freedom be heave and pitch modes of motion, and the mass matrix and its inverse for this 4DOF model be

$$\boldsymbol{M}' = \begin{bmatrix} m'_{11} & m'_{12} & m'_{13} & m'_{14} \\ m'_{21} & m'_{22} & m'_{23} & m'_{24} \\ m'_{31} & m'_{32} & m'_{33} & m'_{34} \\ m'_{41} & m'_{42} & m'_{43} & m'_{44} \end{bmatrix} \text{ and } \boldsymbol{M}'^{-1} = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ \mu_{41} & \mu_{42} & \mu_{43} & \mu_{44} \end{bmatrix}$$

Then the augmented forms of these matrices that will be used in the simulation will be

$$\boldsymbol{M}_{6\times6} = \begin{bmatrix} m'_{11} & m'_{12} & 0 & m'_{13} & 0 & m'_{14} \\ m'_{21} & m'_{22} & 0 & m'_{23} & 0 & m'_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ m'_{31} & m'_{32} & 0 & m'_{33} & 0 & m'_{34} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ m'_{41} & m'_{42} & 0 & m'_{43} & 0 & m'_{44} \end{bmatrix} \text{ and } \boldsymbol{M}^{inv}_{6\times6} = \begin{bmatrix} \mu_{11} & \mu_{12} & 0 & \mu_{13} & 0 & \mu_{14} \\ \mu_{21} & \mu_{22} & 0 & \mu_{23} & 0 & \mu_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_{31} & \mu_{32} & 0 & \mu_{33} & 0 & \mu_{34} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_{41} & \mu_{42} & 0 & \mu_{43} & 0 & \mu_{44} \end{bmatrix}$$

Time rates of the body velocities are equal to zero at each time step, if $\boldsymbol{M}^{inv}_{6\times6}$ is used as the mass matrix inverse in equation (33.a).

Since the rates and initial conditions of the body velocities are equal to zero, these degrees of freedom will not alter throughout the simulation.

3. Since the pitch angle and the motion along the z-axis of the earth-fixed frame are ignored, the time rates of these variables obtained by the velocity transformation equations are set to zero after the transformation at each time step.

Note that such a solution is the same as solving an $n' \times n'$ system where n' is the degree of freedom of the model. However, the same algorithm may be obtained with such an approach.

4.4. Interaction with the Main Program

Currently, the system dynamics simulation loop of the original software runs together with the rest of the program in a dependent manner. The motion equations are solved and the η vector, which is composed of the data required to update the posture of the ship at the end of the time step is calculated inside the cycle. Here, it should be noted that, at the end of each integration step the results are reflected to the user. This means that the integration step of the solution algorithm is dictated by the frame refresh rate of the main loop, which not only calls the equations of motion module for state computation but also realizes the graphical display operations, taking user inputs, etc.

Since it is desirable to have control on the time step of the integration procedure, such a method is not very convenient. Especially for lower order integrators such as Euler, the frame refresh rate, which is currently 16 Hz, is not enough to obtain accurate solutions. On the other hand, rendering the results of the numerical solution at every time step is not possible considering the limited computation capability of the computers. Therefore, a solution to this problem may be to decouple the two processes from each other and use different time steps for each separate operation. This could be done by creating an independent thread (an executable

that runs independently from the main program) for the solution of the equations of motion. As the solution runs by itself on a separate thread, the main program gets the instantaneous values of the states whenever they are required to be used for the frame updating process, which most probably occurs at a lower rate than the numerical integration. Furthermore, the use of different integration algorithms running at quite different rates becomes possible with this approach.

CHAPTER 5

SIMULATIONS AND RESULTS

5.1. Test Procedure

In order to validate the simulation code, Matlab® software, which is also the development environment, is utilized. Test platform is a 1.83 MHz Core2Duo® processor mobile PC with 2GB RAM running Windows Vista® operating system. 7.2.0.232 version of Matlab® is used in obtaining the results.

Since it is not possible to completely isolate a specific operation from the others (such as background processes of the operating system itself) in runtime because of the operating system, the performance of the code depends heavily on the current state on the machine. Also, when working with functions in Matlab®, the software parses functions into the memory when they are first called, which requires more time to complete than further runs. Therefore, to eliminate these misleading results, the process (a portion or whole of the program, a function, data set, etc.) is run once for the parsing phase to complete, and then the same process is run 10 times repeatedly and average of the times to complete each run is taken into consideration, or for short operations, such as evaluat-

ing functions of a few lines, a high number of (on the order of 10^5 or so) cycles is run. It should be noted that the word "performance" often refers to the computation time in this context. Furthermore, a comparative approach is followed in these tests, meaning that the results are not considered as an indication of the absolute performance of the code, rather they give idea about how a program component or an operation contributes to the overall performance in terms of percentages. The percentage performance gain is defined as

$$G_P \triangleq \frac{\left(t_{org} - t_{imp}\right)}{t_{org}} \cdot 100$$

where t_{org} is the original computation time and t_{imp} stands for the duration after the improvement is implemented.

5.2. Component Tests

Before going into the overall program runs, individual aspects such as algorithm inside a function, the presence of a function, evaluation orders or phases of certain variables (which variable is computed first, or at what stage of the program), etc. are considered and tested in terms of computation time. These tests are grouped according to the nature of the improvement made on the program.

5.2.1. Variable Reuse

Some functions of the variables are used for more than once throughout the program. Evaluating these functions each time they

are required during the simulation loop is an ineffective method, therefore such repetitive function calls of the same variable should be avoided by storing the values returned by these functions once, and then reusing these values as required inside the cycle.

Trigonometric Functions of the Euler Angles

In forming the transformation matrices, $\boldsymbol{J}_1(\eta_2)$ and $\boldsymbol{J}_2(\eta_2)$, trigonometric functions of the Euler angles are used. For the sake of computational efficiency, each of the trigonometric functions of the Euler angles in the transformation matrices should be computed only once in each cycle of the simulation loop. This way, only 8 trigonometric function evaluations (3 sines, 3 cosines, and 2 tangents) are performed per time step. Performance gain of this approach is 14%.

A further improvement in this context is utilizing the relations between the trigonometric functions of the same angle. The trials have shown that computation time required for obtaining the tangent of an angle as the ratio of sine and cosine functions rather than calling the tangent function itself is 42% less than that for obtaining them in the regular way. Also obtaining sine and cosine of the same variable consecutively as

$$c\theta = \cos(\theta)$$
$$s\theta = \sqrt{1-(c\theta)^2}$$

is compared to regularly evaluating both functions, however it is seen that it takes 17% more time, degrading the performance.

For the overall process, in which the evaluation of all 6 trigonometric functions (sines and cosines) is considered only 3 cosines are evaluated, and the tangents are determined by algebraic operations, performance gain is 36%.

State Products

Since the hydrodynamic forces are obtained by multivariable Taylor series expansions as explained in the related section above, several multiplications of the velocity components, Euler angles and rudder deflection have to be performed for this purpose. Number of operations regarding the hydrodynamic force calculation is dramatically reduced if the multiplicands of same combinations are computed only once in the same loop. For instance, multiplicands of the force coefficients $K_{\phi uu}$ and $Y_{\phi uu}$ are the same and differ from that of $N_{\phi u|u|}$ only in sign. Therefore, calculating the product ϕuu once in the cycle is possible instead of three times. The performance gain obtained this way depends strongly on the set of hydrodynamic coefficients available.

Coefficient Initialization

As explained in Section 4.2.1, multiplication of several terms in calculating the wave and wind forces throughout the simulation loop is avoided by obtaining these coefficients once in the initialization routine and using those repeatedly at each time step without recalculation.

5.2.2. Function Calls

It is known that using separate functions for individual operations increases the maintainability as it degrades the performance of the code. Excessive number of function calls slows down the program, which is especially a great disadvantage for a real-time one, while accomplishment of many operations in one single and long routine makes the code unreadable and therefore very hard to maintain. For this reason a compromise has to be made between these two conflicting requirements, speed and maintainability.

Decreasing the number of function calls is very critical, since the program may perform even worse if the operations that are aimed to speed up the program are executed in separate functions. For instance, the state products and the trigonometric functions of the Euler angles may be evaluated in a function, the transformation matrices may be obtained in some other, which accept the values of the trigonometric functions as an argument, and a third one may calculate the hydrodynamic forces and moments by passing the state products into this function. However, such a sequence runs even slower than the original one where no improvements are applied. Instead of using 3 functions, the trigonometric functions may be evaluated inside the function that forms the transformation matrices, and the state products may be obtained in the hydrodynamic force computation function, resulting in 2 functions for the same operation. This way, a positive performance gain is observed.

5.2.3. Base Data Generation in the Wave Model

Details of the computation of wave forces are given in Section 4.2.2. It is explained in this section that the wave elevation to be used in the calculation of the wave forces is based on the encounter frequencies, which are functions of the total velocity of the ship and the heading angle. On the other hand, the ocean surface should be simulated for a realistic view of the environment. The wave elevation used for this purpose obviously does not depend on the current state (total velocity and heading angle) of the ship. Therefore, instead of calculating two wave elevations separately, the one for the sea surface rendering is calculated and the data obtained is stored for use in the calculation of the elevation and slope, which are required for the wave forces.

If the wave elevation and slope are to be calculated by picking up random frequencies, calculating the wave component amplitudes, modifying these component frequencies to encounter frequencies, picking up random phase angles and performing the necessary calculations at each time step, the computation of only the wave forces take longer than the real time. Since this is not an option, component frequencies, therefore the wave amplitudes, and the phase angles may be determined at the beginning of the simulation and used repeatedly in the loop. This method gives a reasonable computation time; however using the method summarized in the above paragraph further reduces the computation effort imposed by the wave model.

5.3. Maneuvering Trials

5.3.1. Model Used For the Maneuvering Tests

As also explained before, data sets including all 6 degrees of freedom for a surface vessel are hard to find in the literature. The lower DOF spatial models generally exclude pitch and heave degrees of freedom. Such a model is found in [12] for a Multipurpose Naval Vessel of 48 m length (length between perpendiculars (lbp or p/p)) and 360 ton mass. Figure 14 shows the definition of p/p and other important geometric features of a ship.

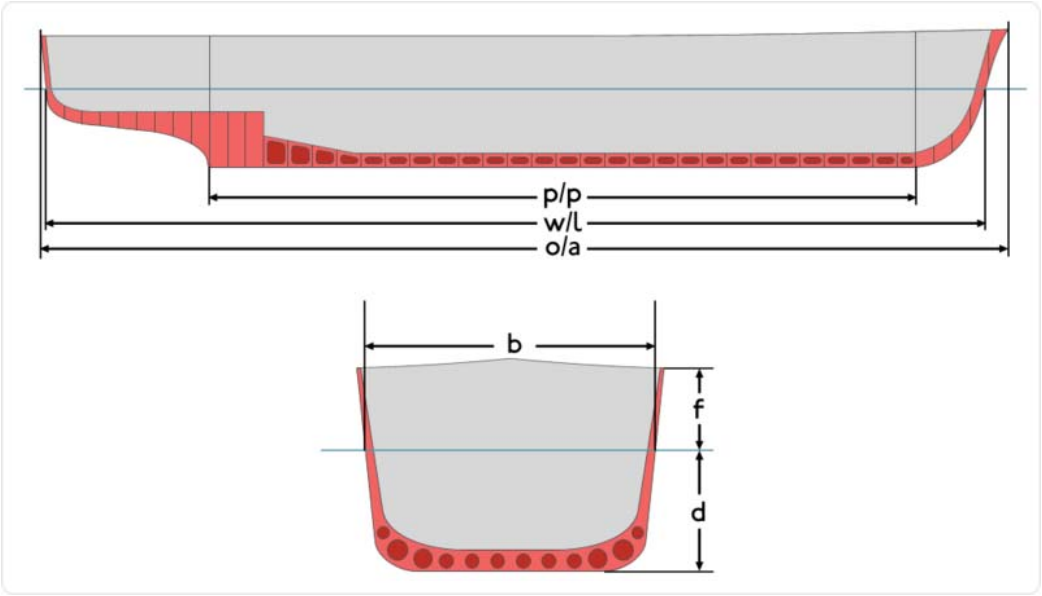

Figure 14 – Main geometric features of a ship, [24]

The abbreviations in the above figure are listed as follows with their explanations:

p/p : Length between perpendiculars

w/l : Waterline length

o/a : Overall length

b : Breadth

d : Draft

Another important feature of the model is that the added Coriolis forces are not explicitly expressed in the equations of motion; rather they are given by means of hydrodynamic coefficients together with the other forms of hydrodynamic forces such as damping. The Vessel Parameters file (`vesselParam.m`) is given in Appendix A.

Three changes are made on the available added inertia coefficients. In making these corrections, other models in the literature, such as the one proposed for a container ship by Son and Nomoto [7] are analyzed. Firstly, $Y_{\dot{u}} = -393000$ is replaced by $Y_{\dot{v}} = -393000$, since a negative $Y_{\dot{u}}$ means that a lateral force is generated on the ship as it accelerates in the forward direction, which is physically meaningless. However, $Y_{\dot{v}}$ is a significant derivative which is available in all hydrodynamic models and the order of magnitude of $\frac{X_{\dot{u}}}{Y_{\dot{v}}}$ is in good agreement with the other models in the literature after this change is applied.

The second modification is made by replacing $K_{\dot{u}}$ by $K_{\dot{v}}$, which is the originally available coefficient in the resource. Having a non-zero $K_{\dot{u}}$ term is not reasonable for a normally operating ship, since a roll motion as the ship accelerates in the forward direction while going in a straight course is an unexpected behavior.

A third and final modification is applied on the signs of $K_{\dot{v}}$ and $Y_{\dot{p}}$. In the available models, these two coefficients are very close or

equal to each other, and positive. Also, they are given equal in magnitude in the Multipurpose Naval Vessel model, however the sign of $Y_{\dot{p}}$ is negative. The sign of this coefficient is changed to positive; however this change does not affect the dynamics of the ship considerably. Figure 15 shows the simulation of the yaw velocity response to a rudder command of 20° applied at $t = 20s$. Note that there is only a minor difference in the response which can only be distinguished with a close-up as given in Figure 16.

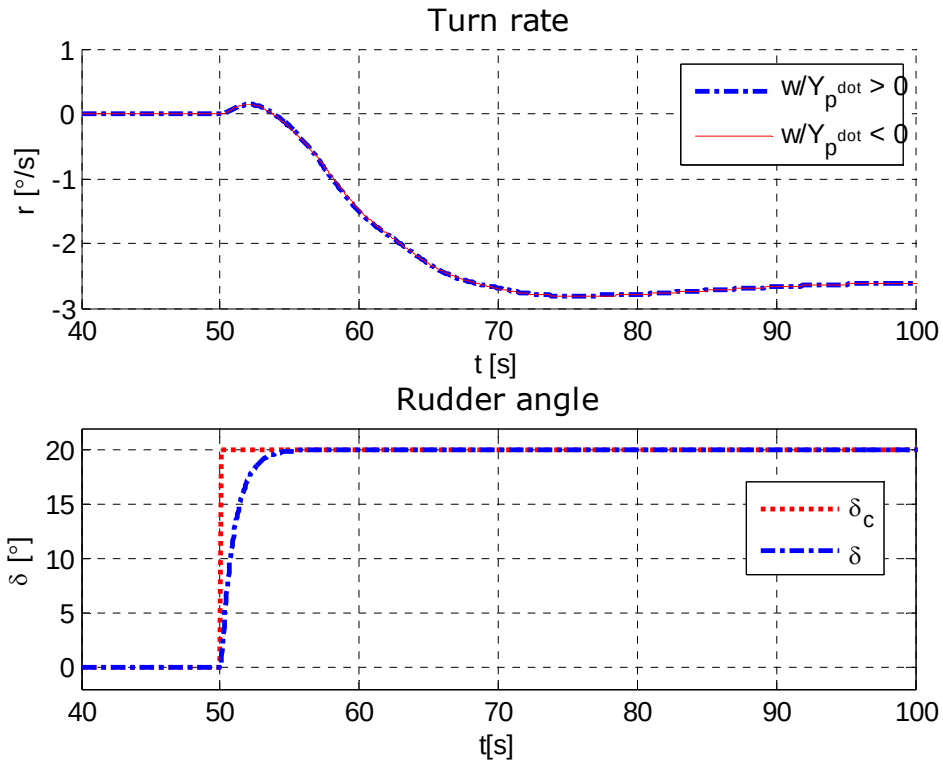

Figure 15 – Turn rate response to a rudder step input

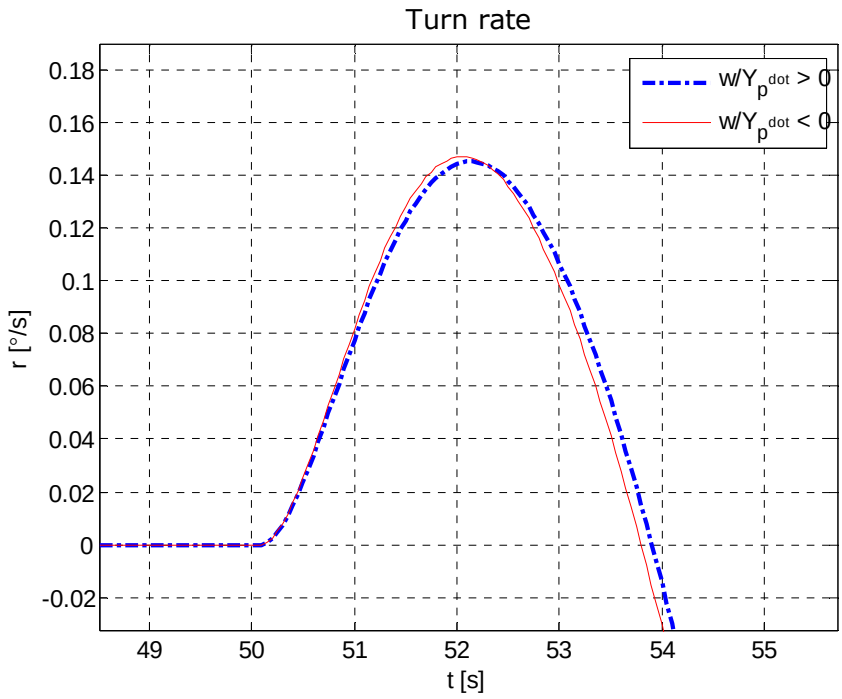

Figure 16 – Turn rate response to a rudder step input (close-up)

Rudder and shaft models are adopted from the container ship model of Son and Nomoto. The shaft model is defined as follows:

$$\frac{n}{n_c} = \frac{1}{T_m s + 1}, \; T_m = \begin{cases} 18.83, & n < 20 \text{ rpm} \\ 5.65/n, & n > 20 \text{ rpm} \end{cases}$$

and the rudder model is

$$\frac{\delta}{\delta_c} = \frac{1}{s+1}$$

Their implementation can be found in Appendix B. The maximum thrust that is generated at the maximum shaft speed is determined so as to provide the nominal velocity at "half ahead" surge com-

mand while the ship is following a straight course. Maximum thrust determined this way is found out to be 32 kN and the maximum velocity at "full ahead" surge command is then obtained as 18 m/s.

5.3.2. Types of Maneuvering Trials

Full scale trials of the ships for testing the maneuvering performance are described in the Maneuvering Trial Code of ITTC (International Towing Tank Conference) [1]. In the evaluation of the code performance, these tests may be conducted in the virtual environment. Most frequently conducted maneuvering tests are briefly explained below:

Turning Circle Test: As the ship is following a straight course at maximum speed, the rudder angle is commanded to a value not less than 15 degrees, preferably to its maximum until the turn rate reaches steady state. In practice, this is achieved upon the completion of a 540-degree turning circle. Terminology related to this test is given in Figure 17.

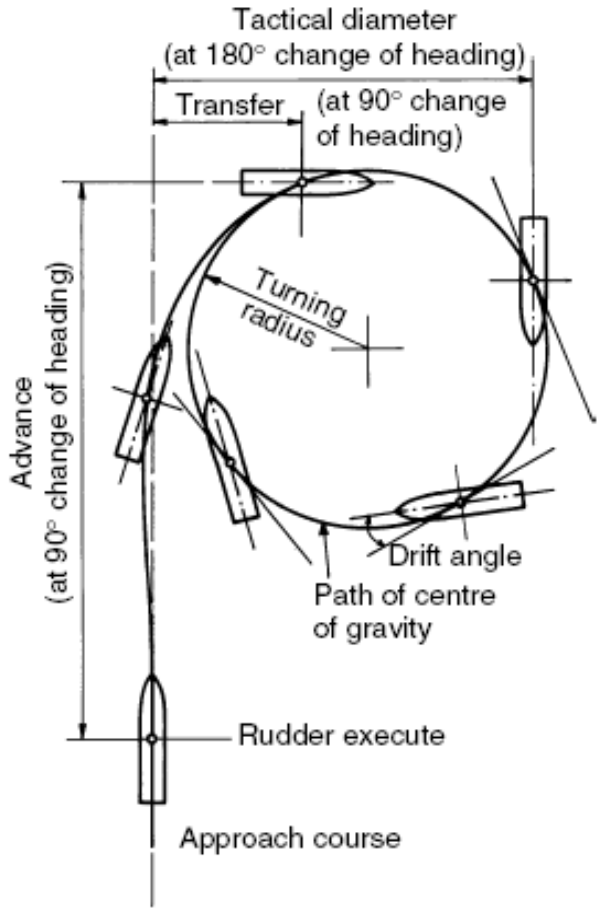

Figure 17 – Turning circle maneuver terminology

Spiral Maneuvers: There may be instabilities on the yaw response of a ship to the rudder input due to mechanical reasons such as misalignment of the rudder or the propeller shaft. There are some tests that are used to determine the yaw stability of ships. Spiral maneuvers are examples of such tests. There are two kinds of spiral maneuvers: Dieudonné's (direct) spiral maneuver and Bech's (indirect or reverse) spiral maneuver.

Dieudonné's spiral maneuver: The maneuver is initiated by a turning circle maneuver. After a constant turning rate is achieved, the

rudder is turned to its original position incrementally. A steady turn rate should be reached at each step. Same procedure is applied for rudder commands on the other side of the zero position.

Bech's spiral maneuver: Ship is turned at a constant turn rate and mean rudder angle is recorded. This maneuver is simpler, and more practical than Dieudonné's spiral maneuver, and provides the relation between the yaw rate and rudder angle of an unstable ship completely, which cannot be obtained by a direct spiral maneuver.

Pull-out Maneuver: This maneuver also is preceded by a turning circle maneuver. After a steady turn rate is achieved, rudder command is immediately set to zero and the rate of turn is observed. For stable ships, the value should decay to zero while a residual turn rate for steady state indicates instability.

Zig-Zag Maneuver: At a straight course, the rudder is given a deflection δ_z and held until the ship gains a heading of ψ_z. At this point the rudder is reversed to an angle of $-\delta_z$ to obtain a heading of $-\psi_z$. This alternation of the rudder angle is repeated at least 5 times to complete the maneuver.

Stopping Trial: Following that the ship has reached maximum velocity upon a full ahead surge command, engine is stopped and full astern is applied. Head reached (distance travelled until a full stop), lateral deviation and stopping time are recorded.

5.3.3. Integrator Performances

Two integrators (Euler and RK4) are evaluated in terms of accuracy and real-time performance. Test maneuver selected for this purpose is the Zig-Zag maneuver since it is the most complicated of the standard maneuvers listed above. A 20-20 Zig-Zag maneuver, which is composed of rudder alternations between -20° and +20° at heading angle changes between -20° and +20°.

First RK4 is used for integration with different time steps and the time step value at which the simulation results converge to a unique solution is searched. The maneuver is simulated for a time span of 0 to 300 seconds with time increments 0.25s, 0.2s, 0.15s, and 0.1s. The results are given below:

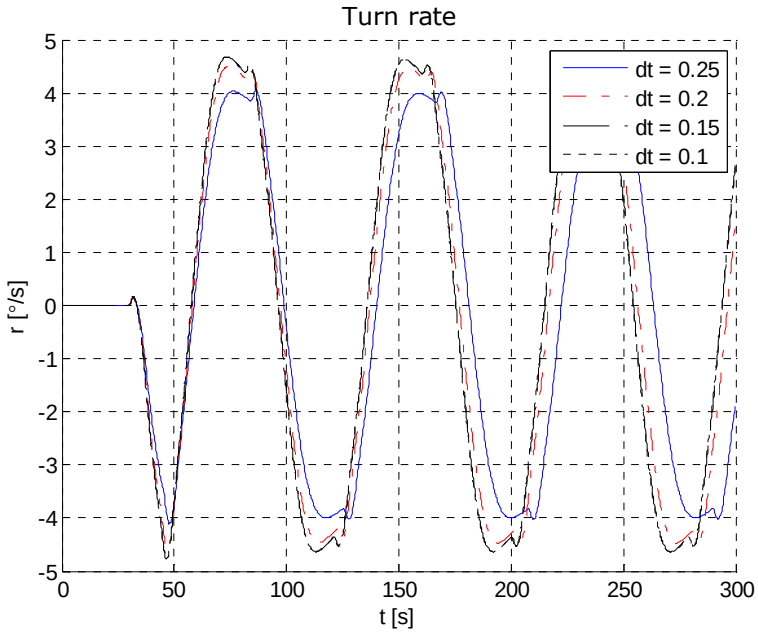

Figure 18 – Turn rates for RK4 solutions with different time steps

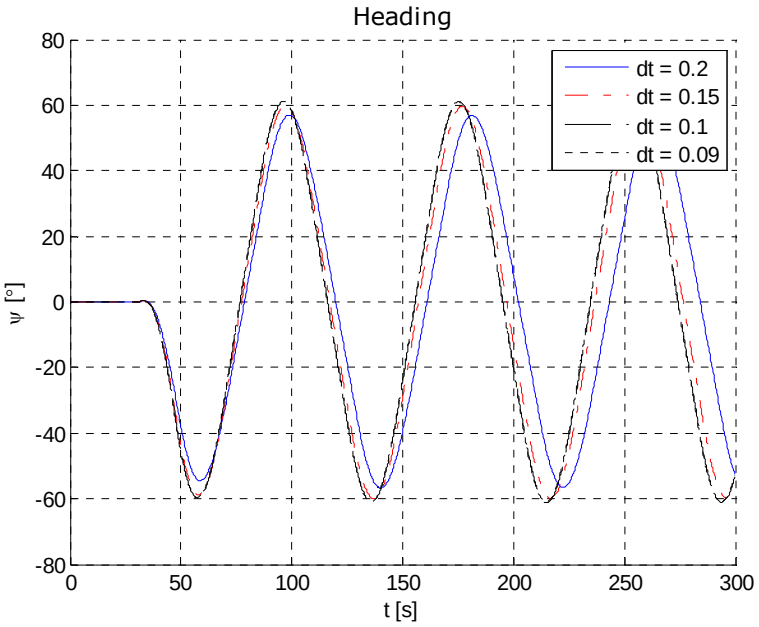

Figure 19 – Heading angles for RK4 solutions with different time steps

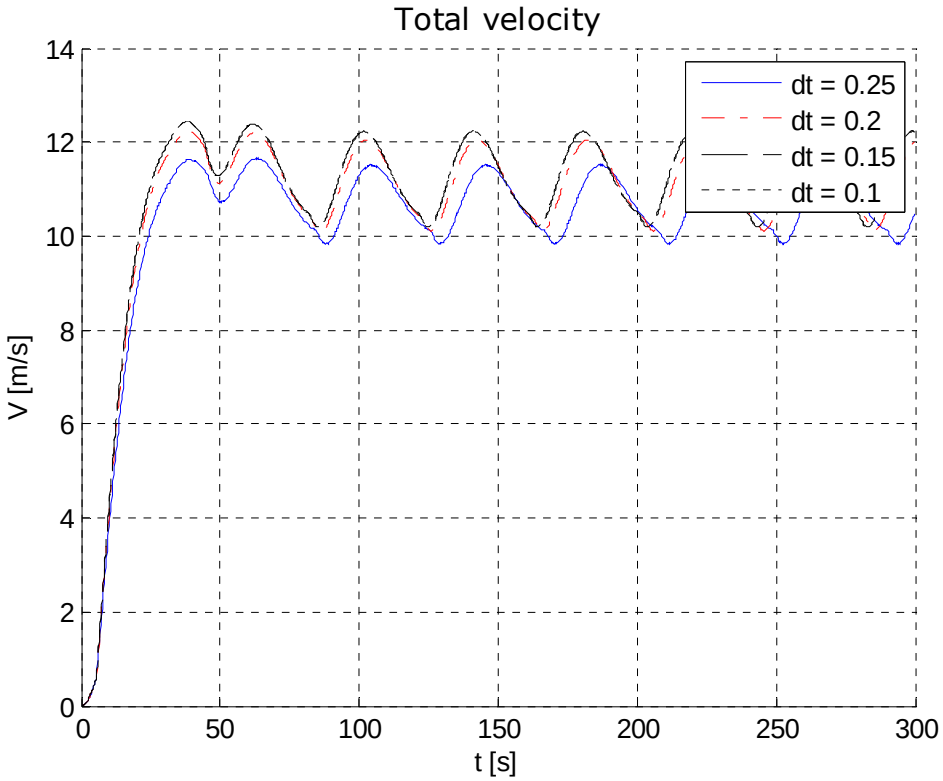

Figure 20 – Total velocity for RK4 solutions with different time steps

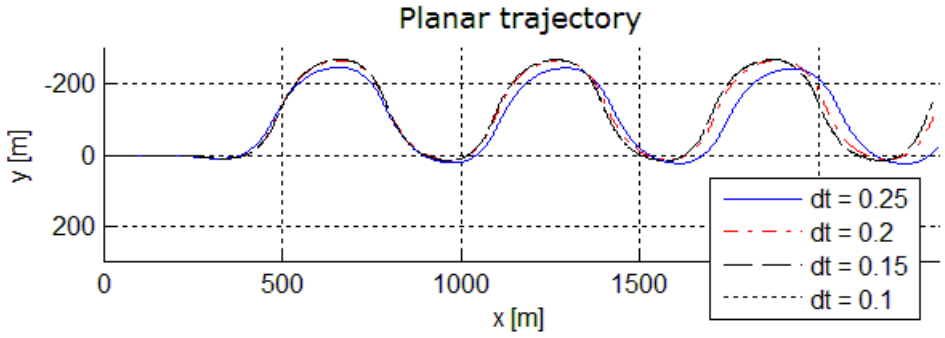

Figure 21 – Planar trajectory for RK4 solutions with different time steps

Note that the planar trajectory is plotted as viewed from above, so that z-axis of the earth-fixed frame is into the paper plane. The solutions obtained by 0.1s and 0.15s time increments exactly. Therefore, 0.15s seems like an appropriate choice of time step for use with the RK4 integrator, based on this standard maneuver. However, when the ship is forced to the limits of rudder angle and shaft speed, the results are not acceptable with these integration intervals. If the time steps are lowered to 0.15s 0.1s, 0.08s and 0.7s, the results come out as follows:

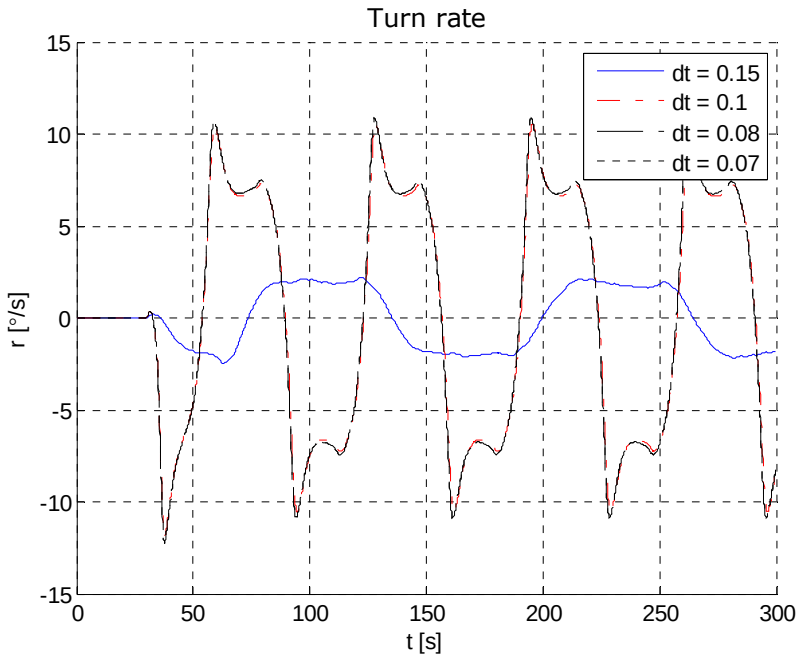

Figure 22 – Turn rates for RK4 solutions with finer time steps

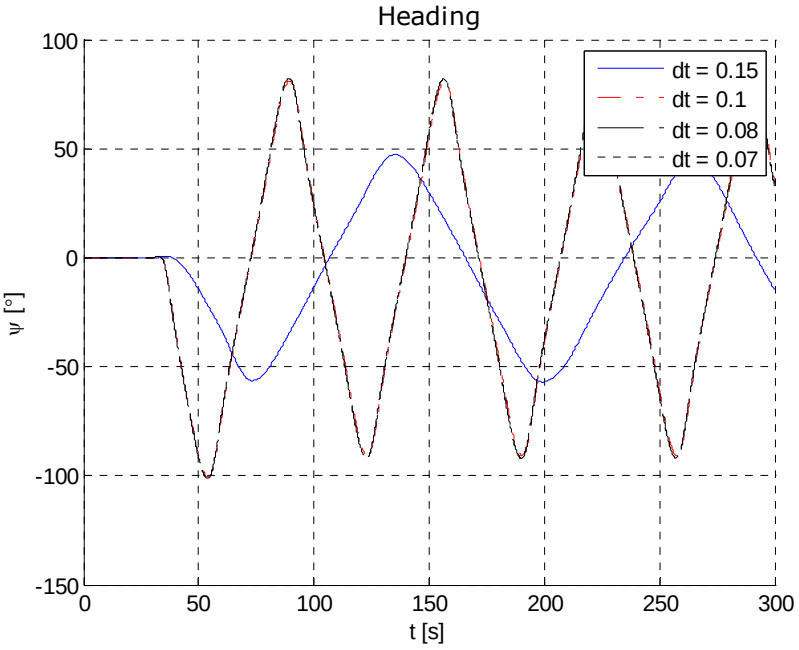

Figure 23 – Heading angles for RK4 solutions with finer time steps

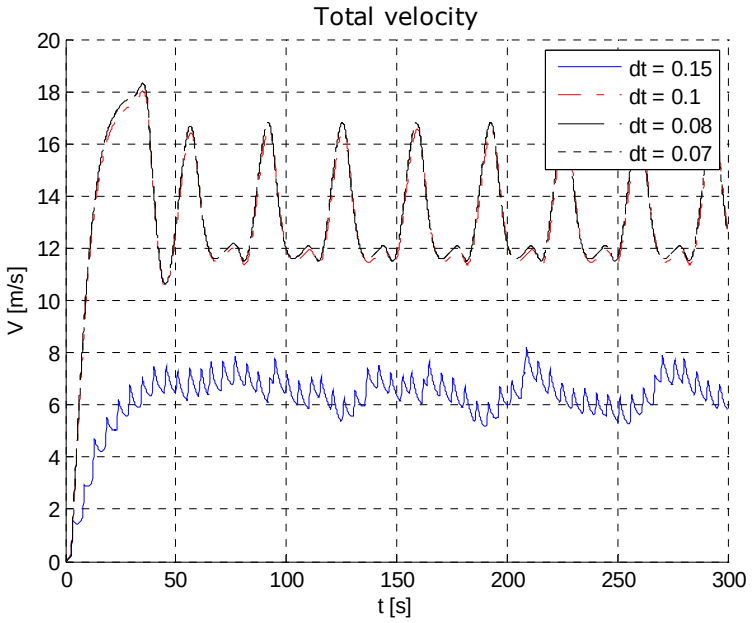

Figure 24 – Total velocity for RK4 solutions with finer time steps

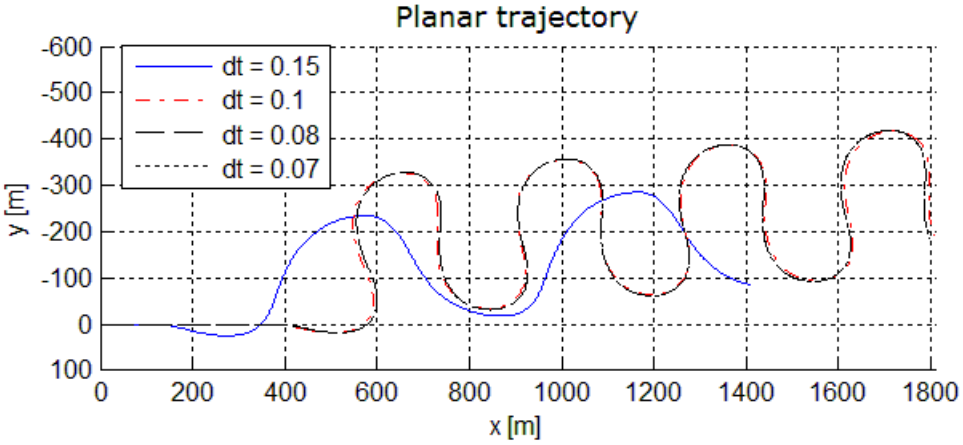

Figure 25 – Planar trajectory for RK4 solutions with finer time steps

Since the limiting cases are simulated and the fastest changes in the state variables are achieved in the simulation whose results are given from Figure 22 to Figure 25 above, it may be concluded that the time step for the RK4 integrator can be selected as 0.08s safely for all possible maneuvers with this model.

After the RK4 integration time interval is determined, the time step for the Euler integrator is investigated taking the solution obtained by RK4 with 0.08s intervals as reference. 0.08s, 0.6s and 0.5s are selected as the time intervals for the simulation. The deviations can be best observed in the total velocity graphics given in Figure 26.

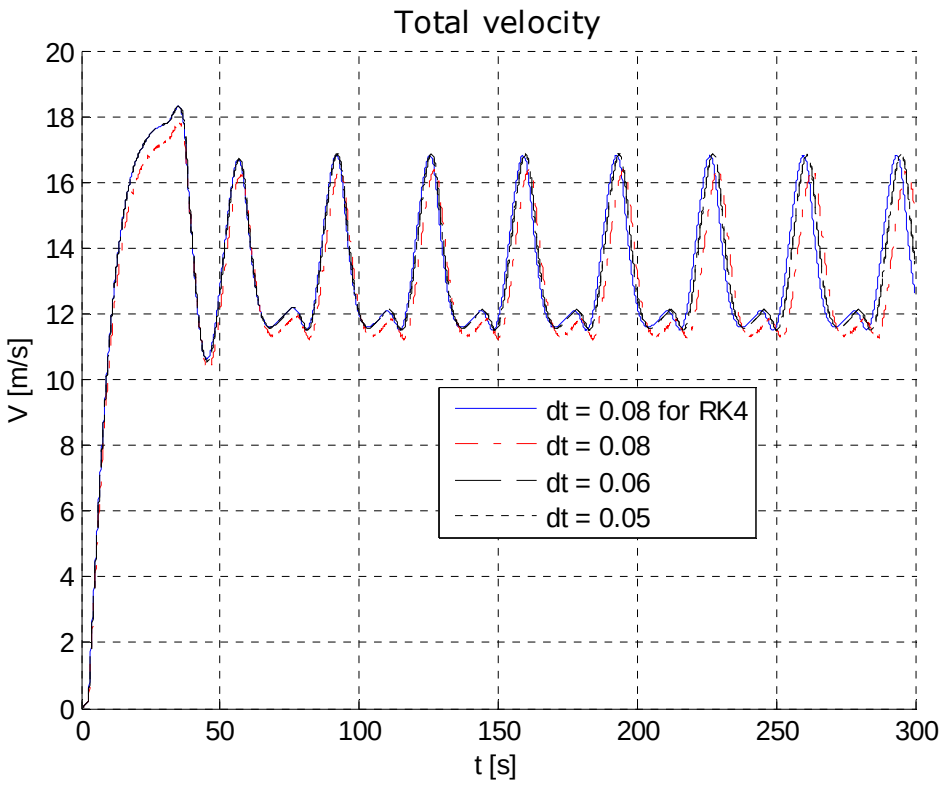

Figure 26 – Euler integrator results on the total velocity graph

The average computation times for different time steps is given in Table 6.

Table 6 – Average computation times for different time steps

	dt	t_{comp}
RK4	0.15	1.076
	0.1	1.618
	0.09	1.798
	0.08	2.028
Euler	0.08	0.721
	0.06	1.065
	0.05	1.377

Considering Table 6 and Figure 26, Euler integrator with a 0.06s time step is selected as the optimum solution in terms of both accuracy and real-time performance. However, it should be noted that this test needs to be performed after every modification (such as addition of a new module) to the program, which increases the complexity, therefore the computational load of the algorithm.

5.3.4. Standard Maneuvering Tests Applied to the Ship

In this section, the results of some of the standard maneuvers, whose definitions are given in Section 5.3.2 are analyzed. In these simulations, the integrator parameters determined in the preceding section are used. The zig-zag maneuver, which is already performed in order to determine the integrator performances, is excluded from this section.

Turning Circle

The ship is given a 45° (maximum rudder angle possible) rudder command to the port side after following an initial straight course

for 30 seconds. When the roll angle is analyzed, excessive values that cannot be physically achieved in normal operation are observed (see Figure 27).

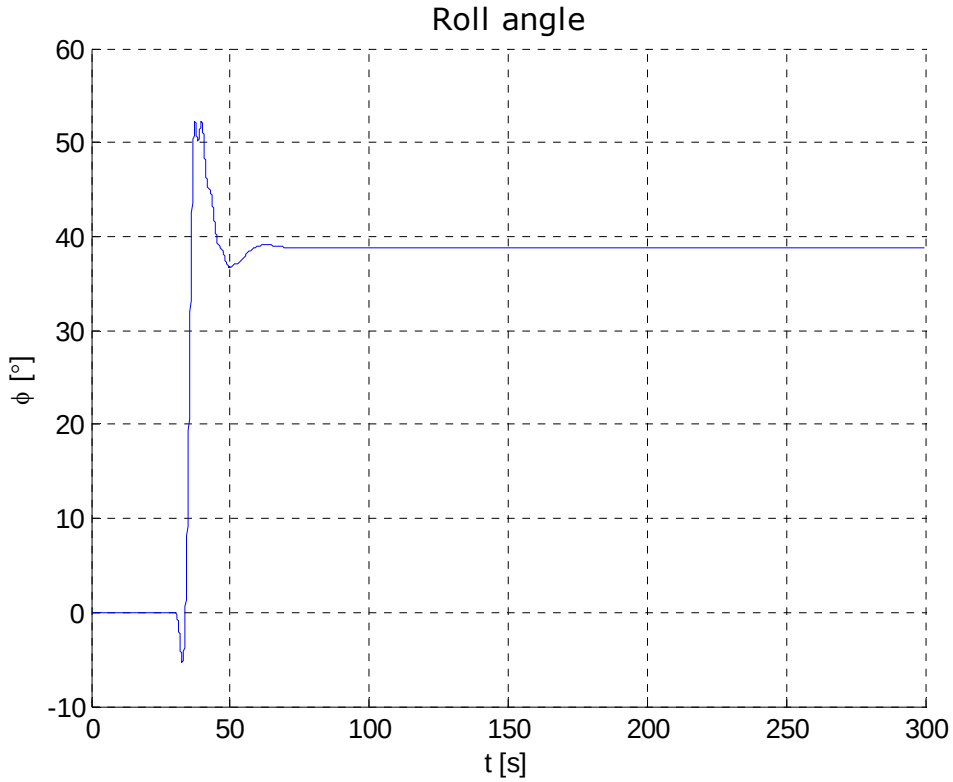

Figure 27 – Excessive roll angles observed in the turning circle

In order to simulate a physically realizable case, the maneuver is repeated at half ahead and 18° rudder command instead of maximum values for the control inputs. These values are selected to obtain 20° roll angle in steady state, which is accepted as the limiting case for a corvette. The simulation results are as follows:

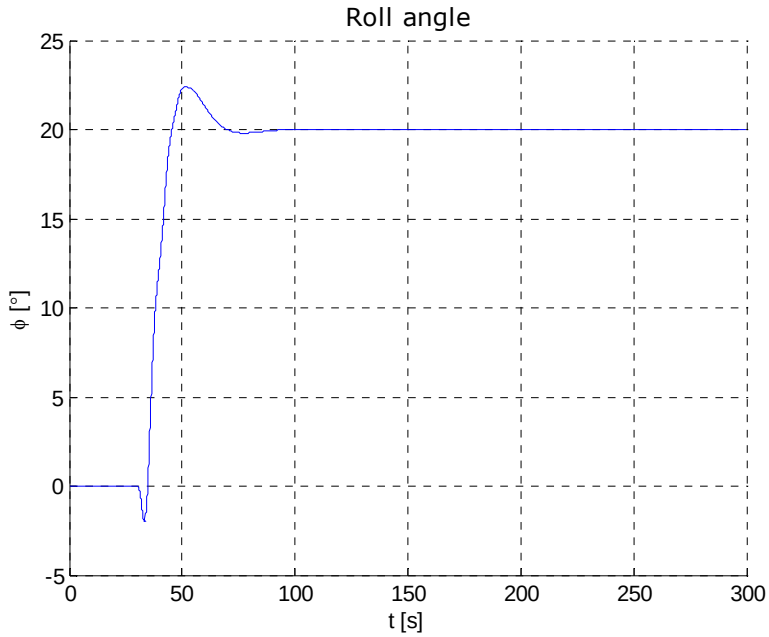

Figure 28 – Roll angle at 18° rudder and half ahead propulsion command

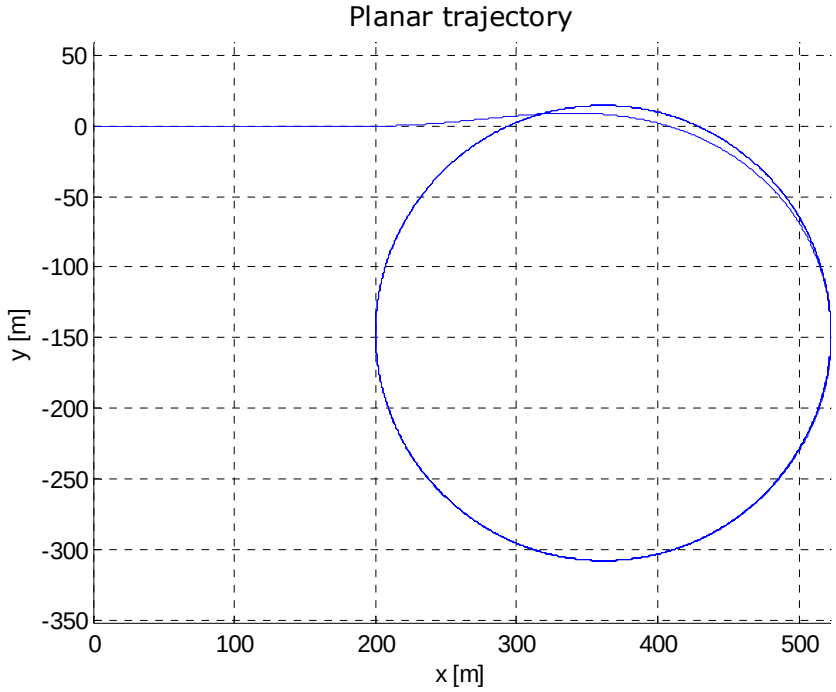

Figure 29 – Planar trajectory at 18° rudder and half ahead propulsion command

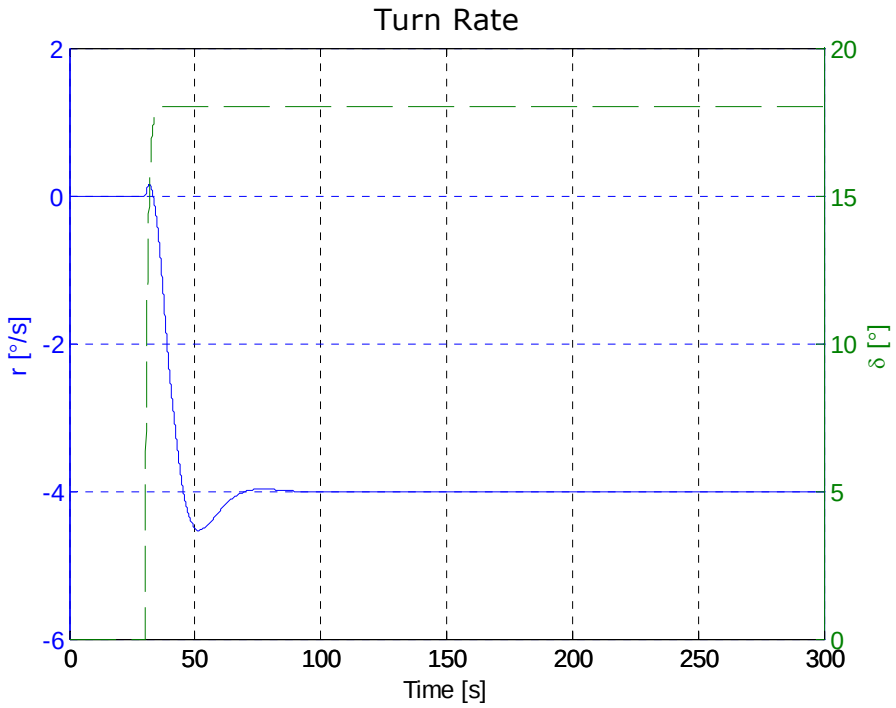

Figure 30 – Turn rate at 18° rudder and half ahead propulsion command

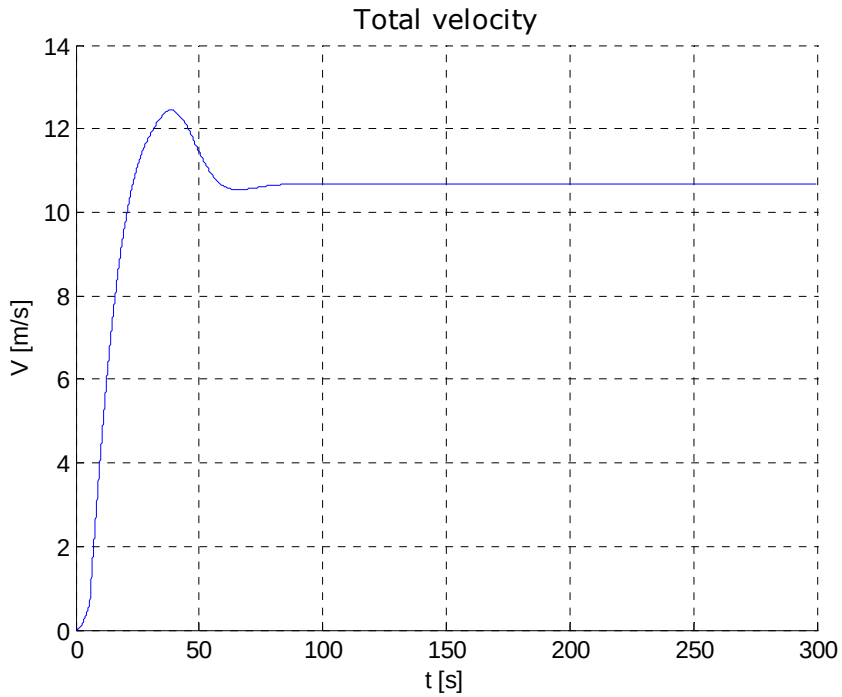

Figure 31 – Total velocity at 18° rudder and half ahead propulsion command

From Figure 29, the tactical diameter can be read as 320 m, which is less than 7 lbp, therefore inside the acceptable region [1].

Stopping Trial

The ship is given a full ahead surge command and after 100 seconds, the command is reversed until the ship stops completely. The surge velocity graph below is the most important result of this test.

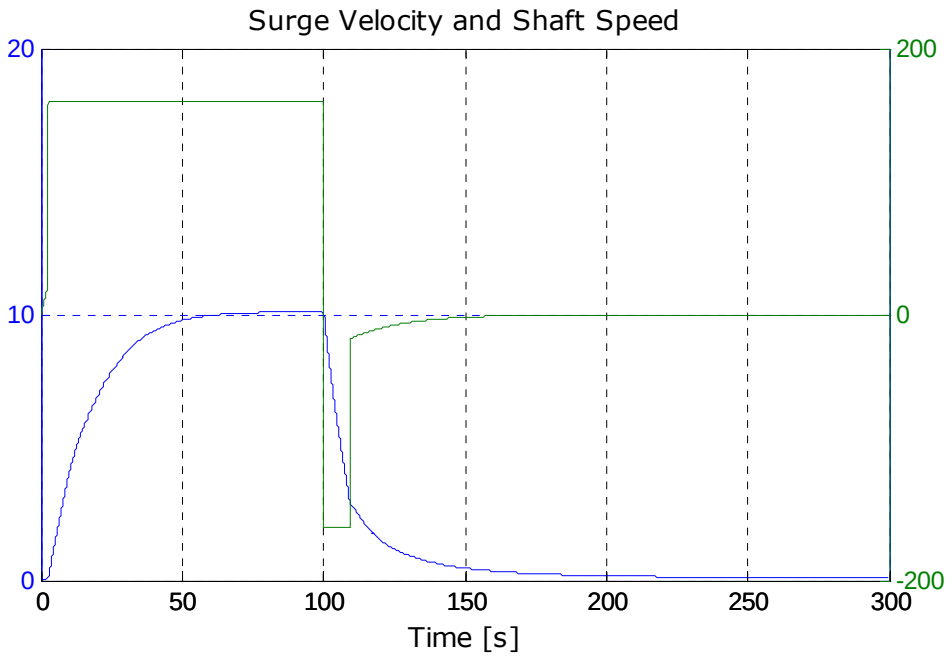

Figure 32 – Surge velocity and shaft speed at stopping trial

It can be observed from Figure 32 that the time required for the ship to come to a zero surge velocity is about 60 seconds. Another important result regarding the simulation code is that the time step selected according to the zig-zag maneuver is too large to handle

the faster dynamics of the shaft. With a few trials, it is observed that the time step should be 0.3 seconds, maximum in order to achieve the numerical stability for the solution. A 100-second simulation takes 0.93 seconds to run with this time step.

Pull-out Maneuver

The rudder is held at its zero position until the ship gains a steady surge velocity. After 70 seconds of a straight line cruise, which is necessary for this purpose, the rudder is given a 20 degree starboard command, and this is held until the ship attains a steady state turn rate. At 200^{th} second of the simulation, at which the turn rate is stabilized, the rudder is brought back to its original position and the behavior of the ship is observed. The turn rate (yaw velocity) graph is given in Figure 33.

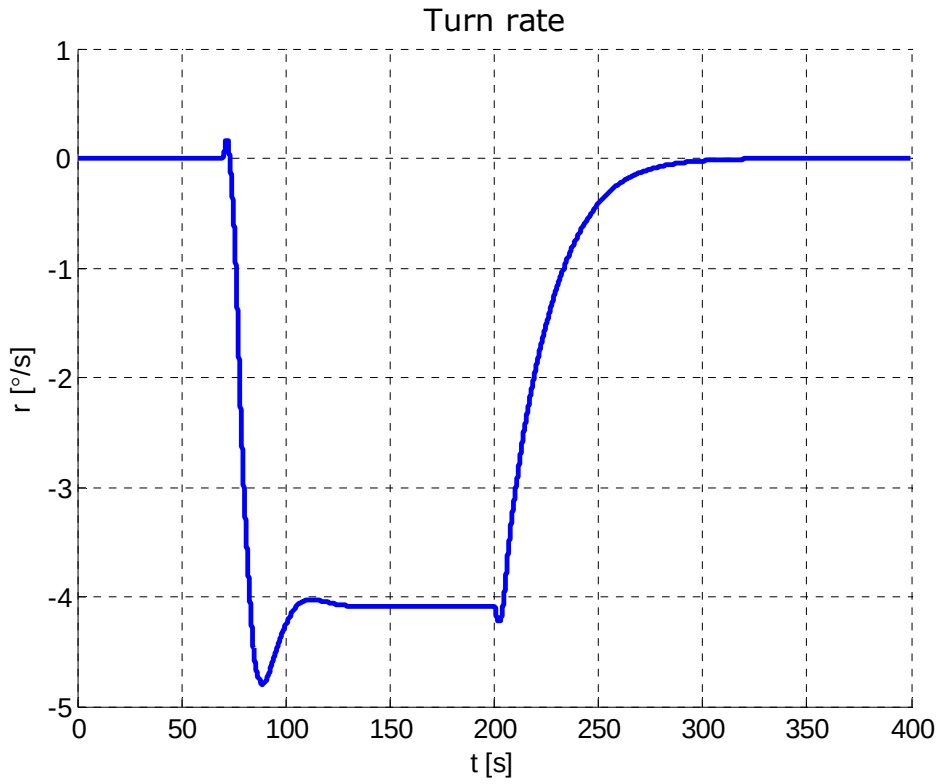

Figure 33 – Turn rate at pull-out maneuver

By analyzing the above graph, it is straightforward to conclude that the ship is yaw-stable, since no residual value is observed for the turn rate at steady state after the rudder is taken back to its zero position.

The planar trajectory followed during this maneuver is given in Figure 34, while

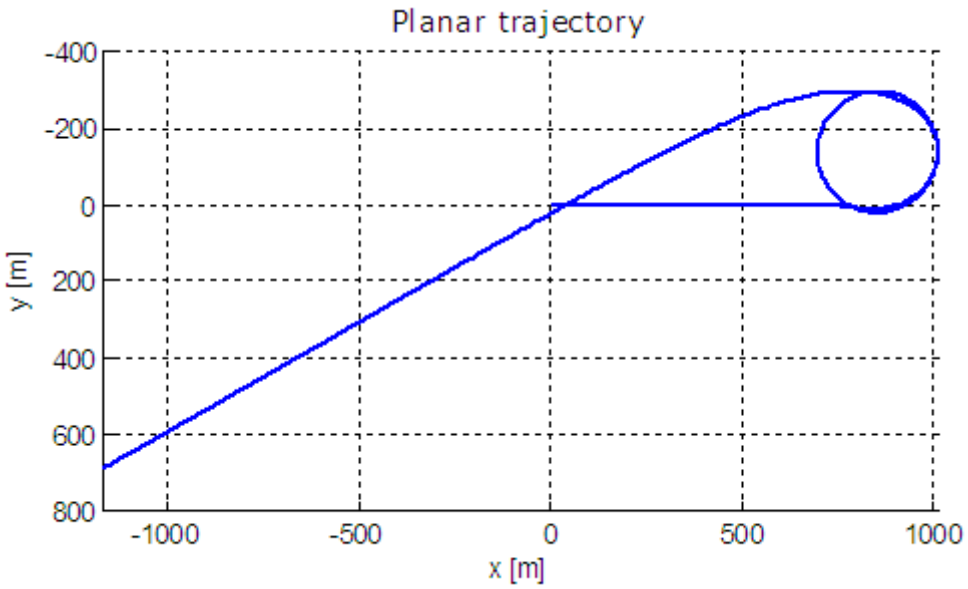

Figure 34 – Planar trajectory followed at pull-out maneuver

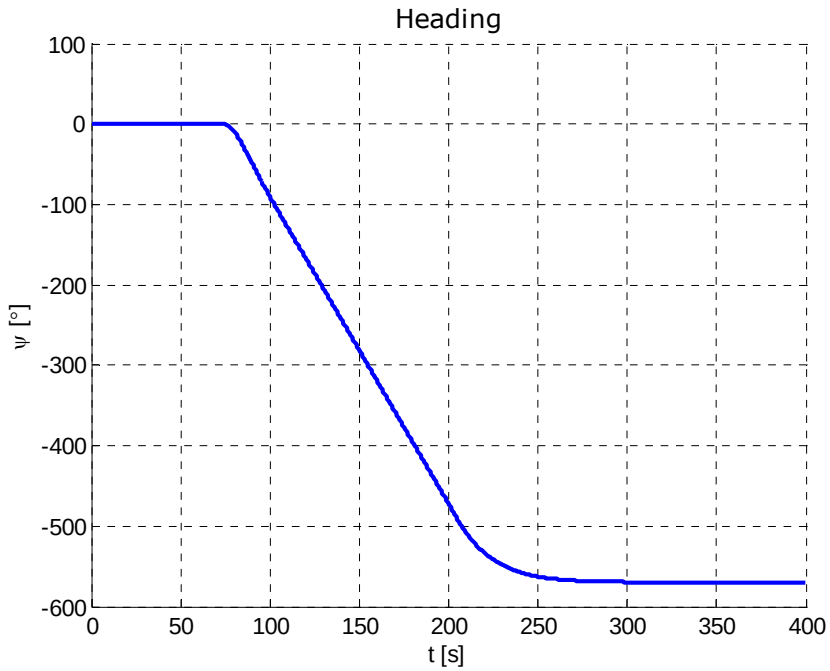

Figure 35 – Heading angle at pull-out maneuver

A 400-second simulation of the pull-out maneuver takes 1.67 seconds to run with 0.6 seconds time step.

Direct Spiral Maneuver

The maneuver is similar to the previous one, however the rudder is taken back to mid-ship position incrementally. 20 degrees of rudder angle is recovered to zero with pauses at 20-10-5-3-2-1-0 degrees. At each increment 30 seconds is allowed to pass to reach incremental turn rates. The planar trajectory, turn rate, rudder deflection, heading angle and roll angle graphs are given from Figure 36 through Figure 40, respectively.

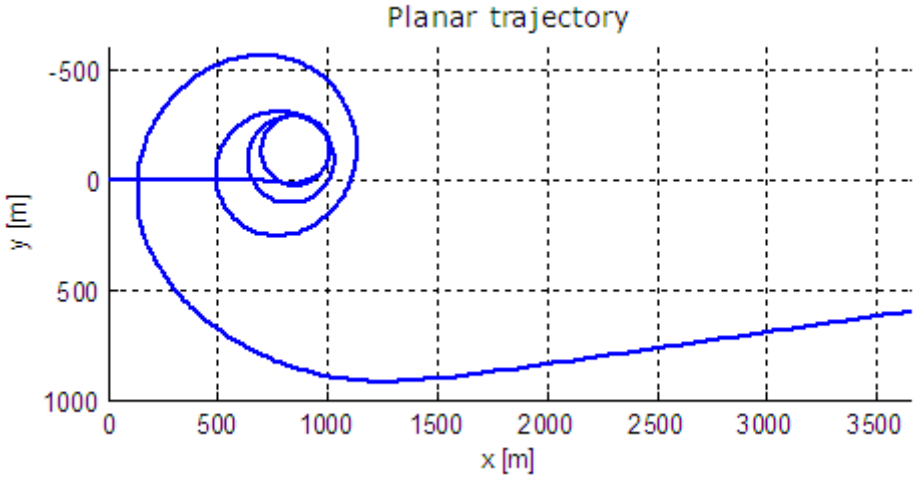

Figure 36 – Planar trajectory at spiral maneuver

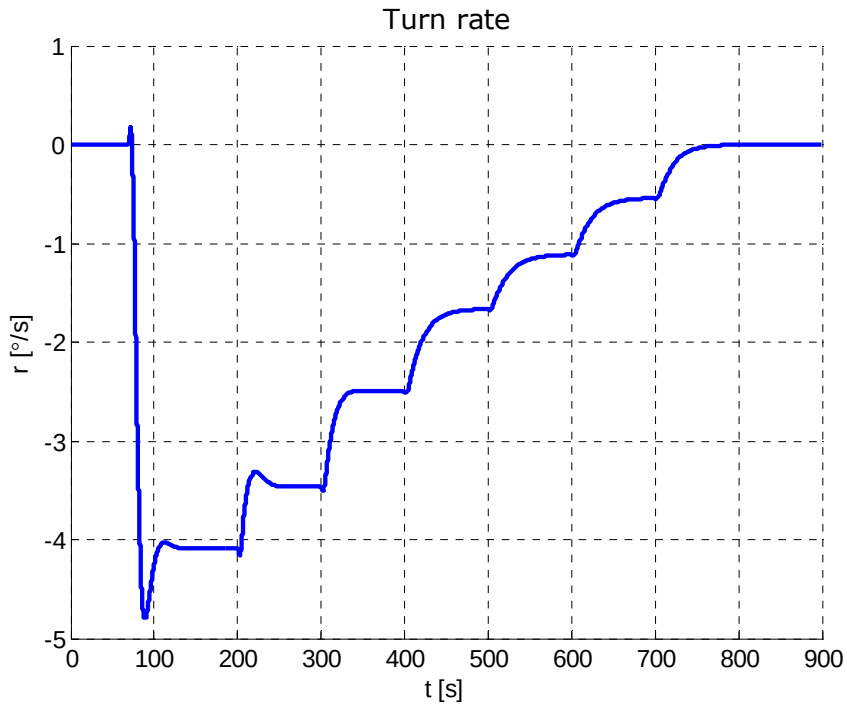

Figure 37 – Turn rate at spiral maneuver

The spiral shape achieved can be clearly observed in the planar trajectory graph given in Figure 36. The yaw stability of the ship is also indicated by Figure 37, where the turn rate of the ship is observed to come to a steady value at each increment of the rudder deflection.

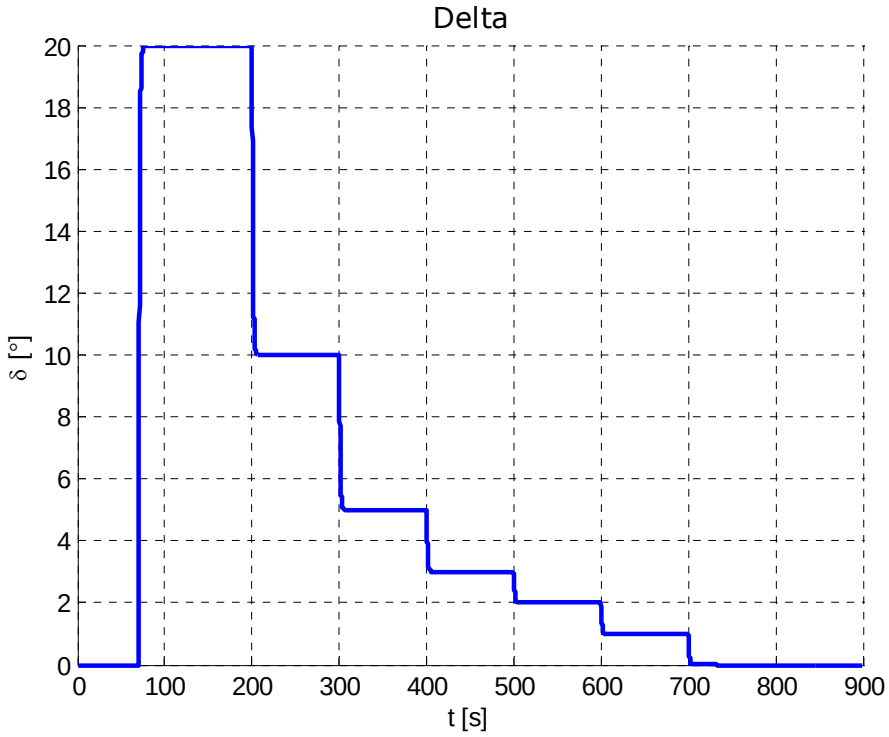

Figure 38 – Rudder deflection at spiral maneuver

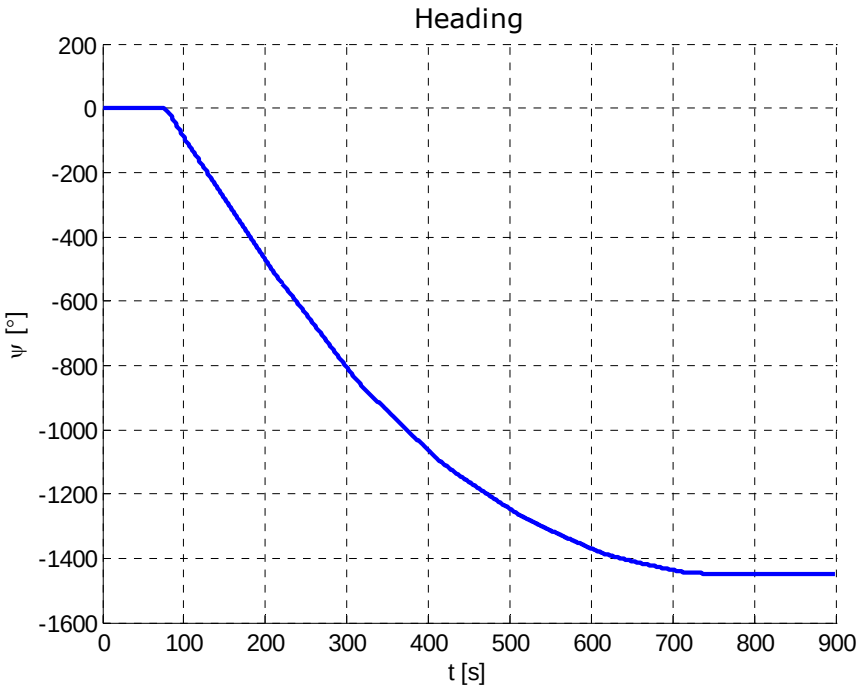

Figure 39 – Heading angle at spiral maneuver

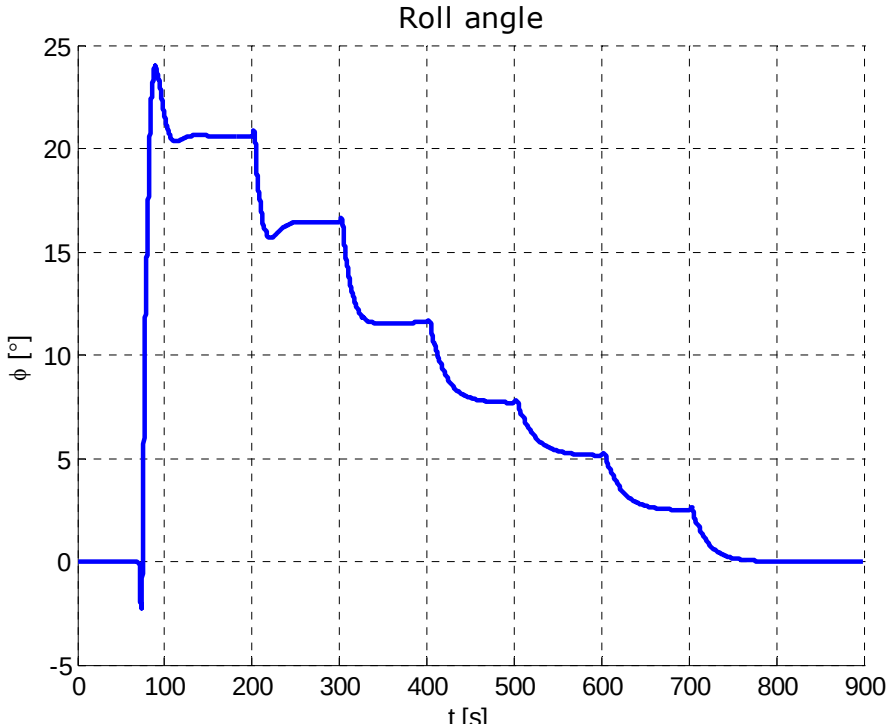

Figure 40 – Roll angle at spiral maneuver

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1. Summary of the Thesis

Tactical simulators are of great importance in the training of naval officers in ship handling and formation practices. Soft simulators, which are composed of only software that can usually be run on standard computers, are inexpensive, practical and flexible solutions that can be offered to this tactical training problem. Naval Surface Tactical Maneuvering Simulation System (NSTMSS), being an HLA-based simulation system, is an example of such soft simulators.

The development of a mathematical model governing the rigid body motion of a ship in the presence of propulsive, hydrodynamic and environmental disturbance forces, and the implementation of this model into a readily available soft simulator, NSTMSS, is studied in this thesis. First of all, the model is put forward by introducing the equations of motions, and explaining the forces that act on the ship in detail. Important concepts such as added mass, hydrodynamic derivatives and buoyancy effects that are characteristic to ship dynamics are discussed. This part also includes the derivation

of propulsion and steering models, and environmental disturbance models.

The implementation of the derived mathematical model is handled in a separate chapter. The algorithm is explained in two main parts: Initialization routine and simulation loop. The former part of the code deals primarily with the operations to be performed in order to make the calculations in the simulation loop in a faster manner. The repetitive operations such as the formation of the constant force coefficients are performed once in the initialization routine and then saved for later use in the loop. Second stage of the algorithm is the simulation loop, at each cycle of which the body accelerations are calculated by means of the motion equations, these accelerations are integrated to obtain the body velocities, component transformations are applied to obtain the time rates of position and orientation variables, and finally they are integrated to give the major output of this code, the position of the origin of the body-fixed frame and the Euler angles.

Following the explanation of the simulation code, the tests that are run on this code and the results of these tests are documented in the fifth chapter. Firstly, the tests on the program components are explained. Then standard maneuvering trials are performed on the full program to simulate the total behavior of the ship to a given set of commands.

In the final chapter, a brief overview of the thesis is given; the results obtained by the component tests and maneuvering trials are

evaluated; and further improvements to be accomplished are discussed.

6.2. Discussions and Conclusions

Some computational improvements are made on different parts of the program. These improvements include the transformation matrix construction, hydrodynamic force calculation, wave force computation, etc. The operations performed in this context are mainly based on reusing the calculated variables. The calculation may have been performed either during the initialization routine, or at each cycle of the real-time simulation loop. For instance, recalculation of the trigonometric functions of the Euler angles, which are used in the transformation matrices, is avoided by calculating them once in each cycle and assigning them to variables for reuse in the cycle. These improvements, except the wave force calculation, have positive, however minor influence on the real-time performance of the code.

An important factor that affects the real-time performance dominantly is the type of numerical integrator selected. For slowly varying dynamical systems, Euler integrator is an appropriate choice for numerical solution of differential equations. In addition to this method, Runge-Kutta 4^{th} order is also employed in the tests as an alternative. The results show that although the time step is greater in RK4 as compared to Euler integrator, the computation time of the Euler integrator is less than that of RK4. Therefore, using Euler's integration algorithm for the time being seems to be more appropriate. However, when the features such as collision, gun firing

or helicopter landing are added to the simulation, RK4 will most probably be necessary, since the performance of the Euler integrator is degraded dramatically as the speed of the dynamics of the system increases. For this purpose, both integrators will be kept in the simulation code and the performance of the code will be evaluated in terms of integration method every time when a major change such as the ones mentioned above is applied to the system. The current real-time performance of the algorithm can be accepted as satisfactory, since although the code is run on a rather slower platform such as Matlab®, the computation times for a 300-second simulation in real-time vary between 1 to 2 seconds.

Discussion of the results obtained as an output of the maneuvering trials strongly depends on the validity of the model. It should be noted that the obtained Multipurpose Naval Vessel model does not contain a propulsion and a rudder model. Therefore, they are adapted from other resources and calibrated according to the performance data of the ship such as nominal speed, and limiting list angle (roll angle) for a similar ship. Keeping these facts in mind, an unexpected behavior is encountered in the turning circle maneuver, so that the ship experiences a high steady-state list angle at full ahead surge command and maximum rudder angle. The limiting case in terms of steering and propulsion commands for this maneuver is determined as half ahead surge at 18° rudder angle. Even in such a case the total velocity of the ship is close to 11 m/s, which is quite high for a marine vehicle. To overcome these uncertainties, other models with propulsion and rudder parameters available should be provided and tested.

6.3. Future Work

The ultimate goal of this whole study is creating an algorithm that will simulate the dynamics of a ship with reasonable accuracy and good real-time performance. The tool used for this purpose is an interpreter based programming environment, on which the algorithm runs slower in comparison to a language that is compiled to run directly on the machine, such as C++. Since the code will finally be integrated to the main software using this language, its performance, whose results are known in Matlab®, should also be tested after implementation in C++ and integration into the main software.

Due to some limitations introduced by Matlab®, some minor modifications inside the functions are required to apply configuration changes on the model, such as replacing the single screw propulsion profile to a twin screw configuration. In order to eliminate user intervention to the internal structure of the code, all changes on the model should be made via input files only. Therefore, while the integration process into the main software is carried out, this aspect should be considered, as well.

One of the most important problems faced during the study has been to find complete data sets of ships of interest, i.e. including all necessary parameters. The problem aroused due to the fact that this valuable information is either required to be purchased or classified, therefore not published. For a realistic simulation, a complete data set should be obtained.

The rudder model can be used to derive and implement fin stabilizer models. Since the same principles are used for roll stabilization of the ships by fins, an appropriate transformation (in terms of location and orientation of the fin stabilizer on the ship) of the rudder model with appropriate parameter values can be used to simulate this effect.

Since the wave model is based upon bulk data, generated in advance, the change of wind velocity and direction is not possible during the simulation. Dynamic updating of the wind velocity and direction may be integrated into the system by utilizing the environment federate of NSTMSS. In such a case each ship object in the simulation takes the base data from the environment federate, which may be run on a separate computer. This way, also a finer mesh of the ocean surface elevation data may be stored in both space and time, which provides a smoother and more realistic simulation of the ocean surface.

REFERENCES

[1] Abkowitz, M. A. *Lectures on Ship Hydrodynamics – Steering and Maneuvrability – Technical Report Hy-5*, Hydro- and Aerodynamics Laboratory, Lyngby, Denmark, 1964.

[2] Arrichiello, F., Chiaverini, S. *A Simulation Package for Coordinated Motion Control of a Fleet of Under-Actuated Surface Vessels*, Proseedings of 5th MATHMOD Conference, Vienna, 2006.

[3] Australian Maritime College – Ship Handling Simulator, http://www.amc.edu.au/facilities/integrated.marine.simulator/ship.handling/, August 2007.

[4] Berge, S. P., Fossen, T. I. *On the Properties of the Nonlinear Ship Equations of Motion*, Mathematical and Computer Modeling of Dynamical Systems, Vol. 6 pp. 365-381, 2000.

[5] Bertram, V. *Practical Ship Hydrodynamics*, Butterworth-Heinemann, 2000.

[6] Chapra, S. C., Canale, R. P. *Numerical Methods For Engineers*, McGraw Hill, 1998.

[7] Fossen, T. I. *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, 1994.

[8] Ginsberg, J. H. *Advanced Engineering Dynamics*, Harper & Row Publishers, New York, 1988.

[9] Multigen-Paradigm Inc. website, http://www.multigen-paradigm.com/products/runtime/vega_prime/modules/marine.shtml, September 2007.

[10] Newman, J. *Marine Hydrodynamics*, MIT Press, Cambridge, MA. , 1977.

[11] Özgören, M. K. *Topological analysis of 6-joint serial manipulators and their inverse kinematic*, Journal of Mechanism and Machine Theory, Vol. 37, 2002, pp. 511-547

[12] Pérez, T. and Blanke, M. *Mathematical Ship Modeling For Control Applications*, Technical Report, Technical University of Denmark, 2000.

[13] Pérez, T. and Blanke, M. *Simulation of Ship Motion in Seaway*, Technical Report, Technical University of Denmark, 2000.

[14] Sarch, M.G. *Fin Stabilizers As Marine Control Surfaces*, California, 2003.

[15] Sicuro, D.L.L. *Physically Based Modeling and Simulation Of A Ship In Open Water 3-D Virtual Environment*, California, 2003.

[16] Skjetne, R., Smogeli, Ø. N., Fossen, T. I. *A Nonlinear Ship Maneuvering Model: Identification and adaptive control with experiments for a model ship*, Modeling, Identification and Control, Vol. 25 pp. 3-27, 2004.

[17] Society of Naval Architects and Marine Engineers, *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*, Technical Research Bulletin, 1950.

[18] Suleiman, B. M. *Identification of Finite-Degree-Of-Freedom Models for Ship Motions*, Virginia, 2000.

[19] Techet, A. H. *Lecture Notes for 13.42 – Design Principles for Ocean Vehicles, Lecture 6 – Ocean Waves* (http://ocw.mit.edu/NR/rdonlyres/Mechanical-Engineering/2-22Spring-2005/83BD1F93-87BD-4FC5-B1A9-8714F2F8CB80/0/lec6_wavespectra.pdf), MIT OpenCourseWare, 2005.

[20] Topçu, O. *Naval Surface Tactical Maneuvering Simulation System – Technical Report*, Ankara, 2006.

[21] Van Amerongen, J. *Adaptive Steering of Ships*, Netherlands, 1982.

[22] Wikipedia, *High Level Architecture*, http://en.wikipedia.org/wiki/High_Level_Architecture

[23] Raymer, D. P. *Aircraft Design: A Conceptual Approach Fourth Edition*, AIAA Education Series, 2006.

[24] Wikipedia, *P/p,* http://en.wikipedia.org/wiki/P/p

APPENDIX A

```
% VESSEL PARAMETERS FILE
%=========================================================================
% Environmental parameters and initial states are given in this file
%=========================================================================
%=========================================================================
% DO NOT EDIT!!!
Xud = 0; Xvd = 0; Xwd = 0; Xpd = 0; Xqd = 0; Xrd = 0;
Yud = 0; Yvd = 0; Ywd = 0; Ypd = 0; Yqd = 0; Yrd = 0;
Zud = 0; Zvd = 0; Zwd = 0; Zpd = 0; Zqd = 0; Zrd = 0;
Kud = 0; Kvd = 0; Kwd = 0; Kpd = 0; Kqd = 0; Krd = 0;
Mud = 0; Mvd = 0; Mwd = 0; Mpd = 0; Mqd = 0; Mrd = 0;
Nud = 0; Nvd = 0; Nwd = 0; Npd = 0; Nqd = 0; Nrd = 0;
%=========================================================================

%=========================================================================
% Vessel parameters
A_w = 105.6;         %' Waterline area [m^2] (approx. L*T)
m = 360000;          % Mass [kg]
Ixx  = 3.4e6;        % Roll moment of inertia [kgm^2]
Iyy = 60e6;          %' Pitch moment of inertia [kgm^2]
Izz = 60e6;          % Yaw moment of inertia [kgm^2]

xG = -3.38;          % x-coordinate of the CG [m]
yG = 0;              %' y-coordinate of the CG [m]
zG = -1.75;          % z-coordinate of the CG [m]
L = 48;              % Length [m]
B = 8.6;             % Beam [m]
T = 2.2;             % Draft [m]
GM_T = 0.776;        % Transverse metacentric height
BM_T = 0.97;         % Distance from center of buoyancy to transverse
                     % metacenter
GM_L = 7.76;         %' Longitudinal metacentric height
z_eq = 0;            %' Equilibrium position in the vertical
                     % direction
x_BC = 0;            %' Buoyancy center in the surge direction

Cw_X = 0.4;          %' Wind drag coefficient (X)
Cw_Y = 0.4;          %' Wind drag coefficient (Y)
Cw_N = 0.02;         %' Wind drag coefficient (N)
A_x = 25;            %' Frontal projection area of the ship [m^2]
A_y = 120;           %' Lateral projection area of the ship [m^2]

nmax = 160;         % Propeller speed at full ahead [rpm]
TA = 200000;        % Thrust available at full ahead [N]
deltamax = 45;      % Rudder angle limit [degree]
deltadotmax = 20;   % Rudder rate limit [degree/s]
```

```
shaft_offset = 0;
Ixx_shaft = 0;

I_B = diag([Ixx,Iyy,Izz]); clear Ixx Iyy Izz
r_G = [xG,yG,zG]; clear xG yG zG


V_disp = m/rho_w;    % Displaced water volume [m^3]
%=======================================================================

%=======================================================================
% Hydrodynamic derivatives
% Added inertia terms
Xud = -17400;  Yvd = -393000;   Kud = 0;         Nvd = 538000;
               Yrd = -1.4e6;    Krd = 0.0;       Nrd = -38.7e6;
               Ypd = -0.296e6;  Kpd = -0.774e6;  Npd = 0.0;

% Force coefficients
Xauu = -1960; Yauv = -11800;  Kauv = 9260;         Nauv = -92000;
Xvr = 0.33*m; Yur = 131000;   Kur = -102000;       Naur = -4.71e6;
              Yavv = -3700    Kavv = 29300;        Narr = -202e6;
              Yavr = -0.794e6;  Kavr = 0.621e6;    Narv = -15.6e6;
              Yarv = -0.182e6;  Karv = 0.142e6;    Nauavf = -0.214e6;
              Yauavf = 10800;   Kauavf = -8400;    Naruf = -4.98e6;
             Yauarf = 0.251e6; Kauarf = -0.196e6; Nauuf = -8000;
             Yuuf = -74;       Kuuf = -1180;
            Yduu = 2*3.5044e3; Kaup = -15500;
                               Kapp = -0.416e6;
                               Kp = -0.5e6;
                               Kfff = -0.325*rho_w*g*V_disp;
%=======================================================================

%=======================================================================
% Mathematical model deviations from the general framework
%
% Missing DOF list
% (1) -> Surge
% (2) -> Sway
% (3) -> Heave
% (4) -> Roll
% (5) -> Pitch
% (6) -> Yaw
missDOF = [3 5];

% Ignore added Coriolis matrix
iaCm = 1;
%=======================================================================
```

APPENDIX B

```
% IMPLEMENTATION OF THE SHAFT AND RUDDER MODEL
%==============================================

% Shaft speed saturation (commanded shaft speed is saturated)
if abs(nc) > nmax
    nc = sign(nc)*nmax;
end

% Shaft dynamics
if abs(n) < 20
    Tm = 18.83; % Time constant for idle speed
else
    Tm = 5.65/abs(n);
end
n_dot = (nc - n)*(1/Tm);

% Rudder angle saturation (commanded rudder angle is saturated)

if abs(deltac) > deltamax
    deltac = sign(deltac)*deltamax;
end

% Rudder rate saturation
deltadot = deltac - delta;

if abs(deltadot) > deltadotmax
    deltadot = sign(deltadot)*deltadotmax;
end
```