UNIVERSITEIT•STELLENBOSCH•UNIVERSITY

jou kennisvennoot • your knowledge partner

Modelling and Simulation of an Autonomous Underwater Vehicle

by

Regardt Busch

Thesis presented at the University of Stellenbosch in partial fulfilment of the requirements for the degree of

Masters of Science in Engineering

Department of Electrical Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Study leader: Dr. I.K. Peddle

April 2009

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: .....
R Busch

Date: .....

Abstract

In this thesis the mathematical modelling and simulation of an autonomous underwater vehicle is presented.

A generic six degree of freedom model suitable for AUV control applications is presented. This model is then tailored to the AUV testbed developed by IMT. The model parameters are determined from vehicle geometry alone. In addition to this, a linear model is presented and analysed in order to determine the modes of motion for AUV.

The development of a generic visualisation system suitable for underwater vehicle simulations is also presented. A generic MATLAB based AUV simulation system is developed, and used to supply the visualisation system with the necessary simulation data. Lastly, two example simulations are shown.

Uittreksel

In hierdie tesis word the modelering en simulasie van ’n outonome onderwater voertuig aangebied.

’n Generiese ses grade van vryheid model geskik vir die ontwerp van beheer stelsels word aangebied. Hierdie model word dan aangepas tot die IMT voertuig. Die model parameters word aleenlik van geometriese inligting bepaal. ’n Linieêre model word ook aangebied, en geanaliseer om die natuurlike dynamica van die voetuig te bepaal.

Die onwikkeling van ’n generiese visualiserings stelsel word ook aangebied. Hierdie stelsel is geskik vir die visualisering van underwater voertuig simulasies. ’n MATLAB gebaseerde simulator is ontwikkel om die visualiserings sagteware met die nodige simulasie data te verskaf. Om af te sluit word twee voorbeeld simulasies gewys.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations who have contributed to making this work possible:

- IMT for funding the project
- Dr I.K. Peddle of the University of Stellenbosch as my study leader.
- Mr S.J. Pauck for his assistance in testing and debugging the simulation system.
- All the past and present members of the ESL, for making this lab what it is.
- The love of my life Charmaine, for all her understanding, encouragement and patience with me, during this project.
- My parents for their continued love and support. It is your unwavering support that has made this all possible.

Contents

List of Figures

List of Tables

Nomenclature

Greek Letters

α	Angle of attack
β	Angle of sideslip
δ_{dl}	Left diving plane perturbation from trim condition
δ_{Dl}	Left diving plane total deflection
δ_{dr}	Right diving plane perturbation from trim condition
δ_{Dr}	Right diving plane total deflection
δ_e	Elevator perturbation from trim condition
δ_E	Elevator total deflection
δ_r	Rudder perturbation from trim condition
δ_R	Rudder total deflection
δ_{tl}	Left throttle perturbation from trim condition
δ_{Tl}	Left throttle total deflection
δ_{tr}	Right throttle perturbation from trim condition
δ_{Tr}	Right throttle total deflection
ϵ_T	Thrust vector angle
ϕ	Roll angle perturbation
Φ	Roll angle
θ	Pitch angle perturbation
Θ	Pitch angle
ψ	Yaw angle perturbation
Ψ	Yaw angle

$\boldsymbol{\omega}$	Angular velocity vector
ρ	Fluid density
λ	Eigenvalue

Small Letters

$\boldsymbol{e}$	Euler angle set
g	Gravitational acceleration
m	Mass
p	Roll rate perturbation
q	Pitch rate perturbation
r	Yaw rate perturbation
$\bar{v}$	Speed perturbation
u	Axial velocity perturbation
v	Lateral velocity perturbation
w	Normal velocity perturbation

Capital Letters

$\boldsymbol{F}$	Force vector
$\boldsymbol{I}$	Vehicle inertial tenser
I_x	Moment of inertia about the roll axis
I_y	Moment of inertia about the pitch axis
I_z	Moment of inertia about the yaw axis
I_{xy}	Roll and pith product of inertia
I_{xz}	Roll and yaw product of inertia
I_{yz}	Pitch and yaw product of inertia
$\boldsymbol{M}$	Moment vector
K	Roll moment
M	Pitch moment
N	Yaw moment

P	Roll rate
Q	Pitch rate
R	Yaw rate
$\boldsymbol{V}$	Velocity vector
$\bar{V}$	Speed
U	Axial velocity
V	Lateral velocity
W	Normal velocity or weight
X	Axial force
Y	Lateral force
Z	Normal force

Subscripts

0	Trim and zero angle of attack
B	Buoyancy forces and moments
G	Gravitational forces and moments
H	Hydrodynamic forces and moments

Superscripts

B	Vector coordinated in body axis
E	Vector coordinated in earth axis

Acronyms

2D	Two dimensional
3D	Three dimensional
AUV	Autonomous Underwater Vehicle
UAV	Unmanned Air Vehicle
IMT	Institute for Marine Technology

Chapter 1

Background and Introduction

1.1 Background

Autonomous Underwater Vehicles, or AUVs for short, have become very popular around the world in recent years. AUVs are mainly used in situations where it would be unfeasible to use manned or tethered vessels. This can be due a variety of reasons, including: monetary cost; crew safety; or simply due to a lack of adequate handling equipment.

In 2004 the Institute for Maritime Technology based in Simons Town started a program in order to develop South African AUV technology. This program has led to an AUV testbed, which is the vehicle under study in this project. It is primarily used by IMT in order to test a variety of concepts and payloads. Most of these tests are related to sonar technologies.

The AUV developed by IMT is shown in figure 1.1. In this photo the vehicle is on its stand at IMT's facilities, with its top cover removed for maintenance.

1.2 Project Overview

Due to the diverse requirements for this project , it is natural to divide the project into two subprojects. They are:

Figure 1.1: Vehicle at IMT Facility in Simons Town

- The development of a mathematical model to describe the dynamics of the IMT AUV testbed.
- The development of a generic six degree of freedom graphical simulation system.

Each of these two sub-projects are now briefly discussed.

1.2.1 Mathematical Modelling of IMT AUV Testbed

IMT followed a relatively informal development process while developing the AUV testbed. This has lead to a functional, but poorly understood and documented vehicle. Therefore one of the primary goals of this project is to develop a mathematical model for this vehicle. This model must be suitable for control engineering applications.

A number of factors significantly complicate the development of a mathematical model for the IMT AUV. These include:

- The lack of available information on the vehicle.

- Access to the vehicle is extremely limited, therefore trial based parameter calculation is not an option.
- Limited literature exists on the topic of AUV modelling.

The modelling is conducted in the following four step process: nonlinear AUV modelling; vehicle parameter estimation; linearisation of the AUV model; analysis of linear model.

Each of these steps are now briefly discussed.

1.2.1.1 Nonlinear AUV Modelling

The first step in modelling any vehicle is to derive a generic nonlinear model for the type of vehicle under study. This generic nonlinear model consists out of three main sets of equations. They are:

- A set of equations to describe the motion of the vehicle relative to the earth, given that the accelerations of the vehicle is known. This set of equations are known as the kinematic equations.
- Secondly, a set of equations are required in order to relate the forces and moments on the vehicle, to the vehicle's accelerations. These equations are commonly based on Newton's second law, and are called the kinetic equations.
- Lastly, equations which describe the forces and moments acting on the vessel must be derived. These equations are typically functions of the various vehicle parameters, as well as the current vehicle state.

Given that the various vehicle parameters are known, these three sets of equations fully describe the motions of the vehicle in six degrees of freedom.

1.2.1.2 Vehicle Parameter Estimation

In order to use the nonlinear equations discussed earlier, the vehicle parameters must be known. The accuracy to which these parameters can

be determined, will for the most part ,determine the accuracy of the final model. Due to the lack of access to the vehicle, the parameters calculated during this project should, at least to some degree, be considered as rough estimates.

1.2.1.3 Linearisation of the AUV Model

Linear models are very useful from a control standpoint. Therefore, a linear small perturbation AUV model is derived by linearising the non-linear model discussed earlier. This linear model will typically form the basis for future control system design work.

1.2.1.4 Analysis of the Linear Model

Finally, the linear model is analysed in order expose the various vehicle modes of motion. The modes of motion provide much insight into the vehicle dynamics. In addition to this, it provides the control engineer with valuable information about which modes need to be actively controlled.

1.2.2 Simulation System Overview

Whenever a large amount of time and/or money is invested in a project, as much as possible of the project should result in generic reusable components. With this goal in mind the modelling and simulation system is subdivided into four main logical units. They are:

- A generic six degree of freedom simulator,
- A generic vehicle visualisation package,
- A pilot interface, and
- A vehicle specific model.

The first three of these blocks form a generic simulation and visualisation system, while the final block contains all the vehicle specific data.

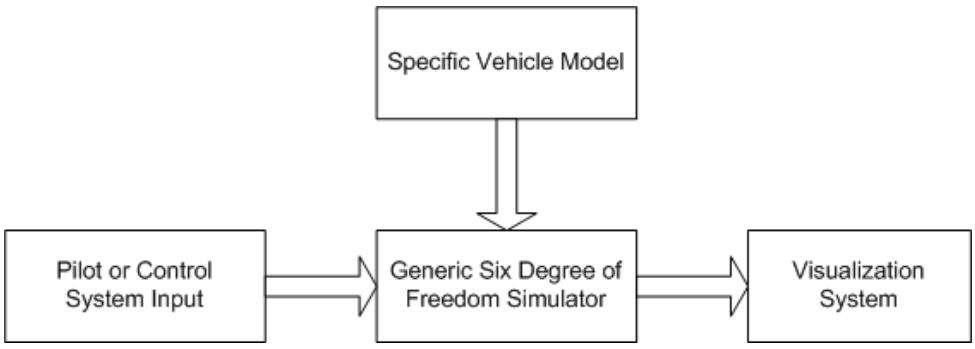

Figure 1.2: System Overview

Figure 1.2 gives a graphical representation of the system. The requirements for each of these logical components are briefly discussed in the following sections.

1.2.2.1 Generic Simulation System

The generic simulation system forms the core of the modelling and simulation environment. The mathematical groundwork required to implement this generic simulator is the topic of chapters 3 and 4. The simulation system consists out of two main components: a forces and moments block and a six degree of freedom block.

Six Degree of Freedom Block

This block implements the generic six degree of freedom equations of motion. As of the start of this project a six degree of freedom block, developed for use in UAV projects, is available in the ESL. Any modifications made to this block should be implemented in such a way as to retain backwards compatibility.

Forces and Moments Block

This block is responsible for generating the driving inputs to the six degree of freedom block. This block is less generic than the six degree of freedom block, but it should still be applicable to a range of vehicles.

To ensure that the forces and moments block is as generic as possible, it is designed to comply with the following requirements:

- Implement the block for a vehicle with the widest possible range of control device inputs. For this project the vehicle is assumed to have elevon style diving planes, an elevator and a rudder.
- Ensure that it is simple to place any number of thrusters at arbitrary locations on the vehicle.
- As far as possible vehicle specific data should be loaded from an external file, rather than hard coded.

1.2.2.2 Generic Vehicle Visualisation Package

In order to better observe the vehicle motions, a graphical visualisation is very useful. Because developing such a system entails a considerable amount of work, a generic visualisation package is developed, instead of developing a visualisation system for submarine simulations. The visualisation software developed for this project is intended to eventually replace the current visualisation tool used in the ESL. For more information about the current system see [4].

The primary reason for replacing the old system is a lack of flexibility, for example the old system is not able to load scene and vehicle models from an external file. The system is therefore not suitable for underwater simulations.

In order to ensure that the visualisation package is as generic as possible, it is developed to comply with the following requirements:

- The package must be usable for aerial vehicle as well as underwater vehicle visualisations.
- This package contains no simulation or modelling code. It simply renders what the simulation system supplies.

- The scene and model to render must be loaded from an external file, using a common 3D file format to simplify model and scene design.
- The visualisation package must be as configurable as possible as to cater to the needs of a wide range of users.
- It should be easily expandable to allow for the of rendering multiple vehicle simulations.

Although much emphasis is placed on developing a flexible generic visualisation package, ease of use should not be compromised too greatly. To help in this regard, a rich user friendly GUI must be implemented. In addition to this a variety of 2D and 3D objets must be provided in order to assist the engineer with vehicle performance analysis.

1.2.2.3 Pilot Interface

In order to use the system without a control system, a pilot input is provided. This pilot interface enables a human pilot to manually control the vehicle. Controlling the vehicle should be as intuitive as possible to a human pilot. In order to achieve this a joystick is used as input device.

As with the rest of this project, components are designed with flexibility in mind. To help achieve this the vehicle is assumed to have the following control inputs: Left and right diving plane or elevon deflection angles; left and right thruster input; an elevator deflection angle; a rudder deflection angle.

1.2.2.4 Specific Vehicle Model

As previously stated, the vehicle specific data is separated from the generic simulation and model. In order to maintain this separation, all vehicle data must be loaded into the simulation system prior to starting the simulation.

1.3 Thesis Outline

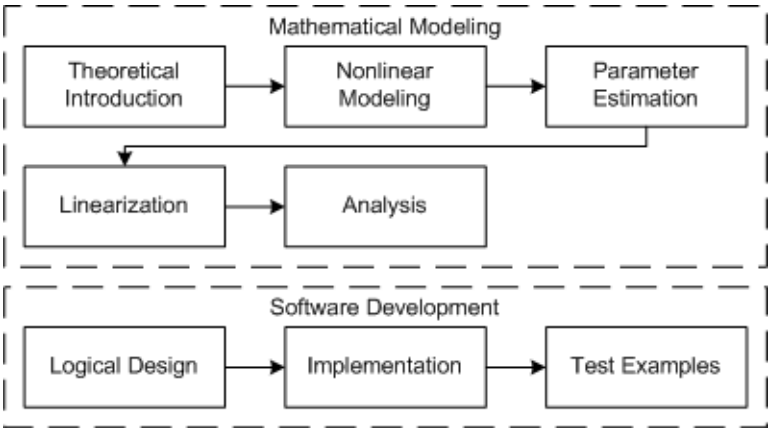

Figure 1.3: Thesis Outline

With reference to figure 1.3 this thesis is subdivided into two parts - one for each main aspect of the project.

The first part of this thesis deals with the mathematical modelling of the IMT AUV. This part of the thesis starts in chapter 2, with the introduction of basic hydrodynamic theory.

The nonlinear modelling of the vehicle commences in chapter 3, with the development of nonlinear equations of motion. In chapter 4, nonlinear models to describe the forces and moments acting on the vessel are developed. Chapter 5 can be considered as the final nonlinear modelling chapter. During this chapter the parameters required by the nonlinear model derived in chapters 3 and 4 are calculated for the vehicle under study.

In chapter 6 this nonlinear model is then linearised about a suitable trim condition. Finally, this linear model is briefly analysed, using model analysis techniques

For the second part of the thesis, the focus is shifted to the development of a generic six degree of freedom graphical simulation system. Chapter 7 deals with the development of the graphical visualisation tool.

Chapter 8 deals with the MATLAB based nonlinear simulator, as well as two example simulations.

Finally, chapter 9 gives a summary of the work conducted during this project, as well as providing a few recommendations for future work.

Chapter 2

Introduction to Marine Hydrodynamics

Hydrodynamics is the field of fluid dynamics primarily concerned with the study of incompressible fluids and the forces acting on solid bodies emersed in these incompressible fluids. Since this is a large and complex field this chapter is not intended as a complete introduction to hydrodynamics, but rather to introduce only the aspects directly related to this project.

Of primary concern in this chapter are those aspects directly related to assumptions made during this project and reasons as to why these assumptions and simplifications are valid are given.

Finally, a comparison between the operating conditions for small sub sonic aircraft, such as UAVs, and small fully submersed submarines, such as AUVs, is given and comparisons drawn.

2.1 Governing Equations for Viscous Fluids

The motions of Newtonian viscous fluids, and there interaction with bodies submerged within them, is relatively complex. Not withstanding this, their motions can primarily be described by only two equations. These two equations are now briefly discussed.

2.1.1 Navier-Stokes Equation

The theoretical approach to solving fluid dynamic problems involves describing the motion of each fluid particle by means kinematic and Newtonian equations. This approach leads to wat is known as the Navier-Stokes equation. This equation expresses the conservation of momentum in a Newtonian fluid, and is given in vector format as[5]

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V}.\nabla)\boldsymbol{V} = -\frac{1}{\rho}\nabla\rho + \nu\nabla^2\boldsymbol{V} + \frac{1}{\rho}\boldsymbol{F} \tag{2.1.1}$$

Together with the continuity equation, this vector equation completely describes the motions of Newtonian fluids. It is self evident that this equation is verry complex, therefore analytical solutions only exists for extremely simple geometries.

2.1.2 Continuity Equation

The continuity equation is discussed in detail by [5]. Wat is of primary concern here is, that, for a incompressible fluid of constant density the following is valid:

$$\nabla.\boldsymbol{V} = 0 \tag{2.1.2}$$

2.2 Hydrodynamic Similarity Parameters

A large number of nondimensional values are used in the field of hydrodynamics in order to describe the various flow properties independently of scale, these are commonly known as similarity parameters. A few of the most important similarity parameters are now introduced.

2.2.1 The Reynolds Number

The Reynolds number is the ratio of inertial forces to the viscous forces, and is given by[6]

$$R = \frac{\bar{V}_0 l}{\nu} \tag{2.2.1}$$

where, $\bar{V}_0$ is the vehicle trim velocity, l its length and ν the kinematic viscosity of the operating medium. The viscosity of water is approximately one order of magnitude smaller than that of air , while the operating velocities of AUVs are typically one or two orders of magnitudes smaller than that of subsonic aircraft, thus the Reynolds number for slower subsonic aircraft, such as UAVs, are comparable to that of AUVs and are in the region of 1×10^6.

2.2.2 The Froude Number

The Froude Number gives a measure of the importance of gravity effects on the fluid mution, and is given by[6],

$$F_r = \frac{V_0}{\sqrt{gh}} \tag{2.2.2}$$

where h is the distance to the free surface[1].

Gravity affects the hydrostatic pressure and only alters the flow if a dynamic boundary condition exists (e.g. wave movement is present). Gravity effects are negligibly small when $\sqrt{gh} \gg V_0$, which is the case when at least one of the following conditions hold: the free surface is undisturbed; it is far away; or it is entirely absent.

For this project it is assumed the vessel is operated in calm waters far from any free surface and thus these effects can be ignored.

[1]The air/water interface is a free surface.

2.2.3 The Cavitation Number

Cavitation is the process whereby the moving object causes cavities of gas to form in the liquid. The Cavitation Number is used to give an indication of the likelihood for cavitation to occur, and is given by[7],

$$\sigma = \frac{P_\infty - P_v}{\frac{1}{2}\rho V_0^2} \tag{2.2.3}$$

where P_∞ is the absolute ambient pressure and P_v the vapour pressure for the fluid of concern.

Cavitation typically occurs at high velocity or when P_∞ and P_v are close in value. The vibrations caused by the collapsing cavities of gas can cause structural fatigue as well as pitting of the surface.

Non-cavitating flow is assured when $\sigma^{-1} \ll 1$, thus to ensure that cavitation does not occur $V_0 \ll \sqrt{\frac{P_\infty - P_v}{\frac{1}{2}\rho}}$ must hold. Since this project deals with the modelling of a low speed AUV, cavitation over the AUV body and lifting surfaces is unlikely to occur and can safely be ignored. This argument does not necessarily hold for the flow over the propellers and the cavitation number for the propellers should be calculated. In this case V_0 must be substituted for V_{TI}, the propeller inlet velocity.

2.2.4 The Weber Number

The Weber Number is used to indicate the significance of surface tension while moving close to a free surface, and is given by[7]

$$W_e = \frac{\rho V_0^2 l}{\Sigma} \tag{2.2.4}$$

where Σ is the surface tension coefficient.

Surface tension can be neglected if $W_e^{-1} \ll 1$, thus to ensure that surface tension can be neglected $l \gg \frac{\Sigma}{\rho V_0^2}$ must hold.

Since it is assumed throughout this project that the vehicle is operated far from any free surfaces, surface tension effects are ignored.

2.2.5 The Mach Number

The mach number of a object relates the object's velocity through a fluid medium to the speed of sound in the medium. The Mach number gives an indication of the compressibility of the flow, and is given by[6];

$$M = \frac{\bar{V}_0}{c} \tag{2.2.5}$$

where $\bar{V}_0$ is the vehicle trim velocity and c is the speed of sound in the operating medium.

In order for the flow to be considered as noncompressible $M \ll 1$ must hold, thus the object must be moving much slower than the speed of sound in the fluid medium.

2.3 Vehicle Operating Environment

So far in this chapter various hydrodynamic concepts and parameters were introduced. In this section these parameters are now calculated for the vehicle under consideration in order to determine whether any of these parameters are insignificant and thus can safely be ignored. Finally a comparison is drawn between the operating conditions for small subsonic aircraft and that of small fully submerged submarines.

2.3.1 Operating Conditions for the IMT AUV

Due to various limitations the IMT AUV is operated exclusively in calm water with minimal ocean currents. Currently the vehicle is operated mainly in and around Simons Town harbour.

Temperature

Ocean temperature varies greatly across the globe. Based on measurements taken once a month over a two-year period at O-birth, the entrance

to the harbour, the temperature at the harbour varies between 14^oC and 18^oC, with an average temperature of 15^oC[3].

Salinity

Seawater in the worlds oceans are not uniformly saline, with the vast majority in the rage of 31 to 38 parts per thousand or, ppt for short. Based on measurements taken once a month over two years at O-birth, the salinity of seawater at Simons Town harbour varies between 34.5 ppt and 35.5 ppt, with an average salinity of 34.9 ppt[3].

Operating Depth and Pressure

The vehicle is used to operate at a depth of up to thirty meters depending on the nature of the run. Most runs are carried out between six and eight meters, however when video taping is required the runs are conducted at a depth of one and a half metres.

According to Leroy and Parthiot [8] pressure above atmospheric in the worlds oceans can be calculated to an accuracy of $\pm 8000\ Pa$ using the following equation,

$$P_i(Z,\phi) = h(Z,\phi) - \delta h_i(Z) \tag{2.3.1}$$

where P is the pressure increment in MPa, Z is the depth in meters, ϕ is the latitude, $h(Z,\phi)$ is given by

$$h(Z,\phi) = h(Z,45) \times k(Z,\phi) \tag{2.3.2}$$

$$h(Z,45) = 1.00818 \times 10^{-2}Z + 2.465 \times 10^{-8}Z^2 - 1.25 \times 10^{-13}Z^3 + 2.8 \times 10^{-19}Z^4 \tag{2.3.3}$$

$$k(Z,\phi) = \frac{g(\phi) - 2 \times 10^{-5}Z}{9.80612 - 2 \times 10^{-5}Z} \tag{2.3.4}$$

$$g(\phi) = 9.7803(1 + 5.3 \times 10^{-3} \sin^2 \phi) \tag{2.3.5}$$

and $\delta h_i(Z)$ is a corrective term, which for general oceans between 60^o N

and 40^0 S is given by.

$$\delta h_i(Z) = \frac{1.0 \times 10^{-2} Z}{Z + 100} + 6.2 \times 10^{-6} Z \tag{2.3.6}$$

Now taking the operating depth as seven meters and the latitude as 34^0 South[2], the operating pressure above atmospheric is given by equation 2.3.1 as

$$P_i = 70.505 \text{ kPa} \tag{2.3.7}$$

And finally taking the atmospheric pressure to be 101.3 kPa then total ambient pressure is

$$P_\infty \approx 171.8 \text{ kPa} \tag{2.3.8}$$

Density

The density of seawater is rarely measured directly, but is rather calculated from salinity, temperature and pressure using an equation known as the Equation of State of Seawater. This equation consists out of three polynomials and 41 constants and thus is not given here.

A number of calculators witch implement the equation exists, using such a calculator implemented at Dalhousie University[9] and taking the salinity as 34.9‰, the temperature as 15 ^oC and the pressure above atmospheric as 70.5 kPa the density is given as:

$$\rho \approx 1026 kg.m^{-3} \tag{2.3.9}$$

Mach Number

In order to calculate the Mach number at trim velocity, the speed of sound in seawater is needed. According to Mackenzie[10] this can be approxi-

[2]The approximate latitude of Simon's Town

mated by the following empirical equation[3],

$$\begin{aligned} c = {} & 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 \\ & + 1.340(S - 35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 \\ & - 1.025 \times 10^{-2}T(S - 35) - 7.139 \times 10^{-13}TD^3 \end{aligned} \tag{2.3.10}$$

where T is temperature in degree Celsius, S is salinity in parts per thousand and D is depth in meters. Now, taking the temperature to be 15^oC, the salinity as $34.9\,ppt$ and at a operating depth of seven meters, the speed of sound in seawater at Simons Town Harbour is given by,

$$\begin{aligned} c = {} & 1448.96 + 4.591 \times 15 - 5.304 \times 10^{-2}15^2 + 2.374 \times 10^{-4}15^3 \\ & + 1.340(34.9 - 35) + 1.630 \times 10^{-2}7 + 1.675 \times 10^{-7}7^2 \\ & - 1.025 \times 10^{-2}15(34.9 - 35) - 7.139 \times 10^{-13}15 \times 7^3 \\ = {} & 1506.6877\,m.s^{-1} \end{aligned} \tag{2.3.11}$$

Now, from 2.3.11 and 2.2.5,

$$M = \frac{\bar{V}_0}{c} = \frac{1.2}{1506.6877} = 0.7964 \times 10^{-3} \tag{2.3.12}$$

It is evident that $M \ll 1$, therefore it is entirely reasonable to assume that the flow is noncompressible.

Reynolds Number

From equation 2.2.1, and taking the kinematic viscosity of seawater as $1.05 \times 10^{-6}\,m^2.s^{-1}$,[5] the Reynolds Number at trim velocity is,

$$R = \frac{\bar{V}_0 l}{\nu} = \frac{1.2}{1.05 \times 10^{-6}} = 1.143 \times 10^6 \tag{2.3.13}$$

[3]This equation has a standard error of estimate of $0.070\,m.s^{-1}$

Cavitation Number

As is evident from equation 2.2.3, both the ambient pressure as well as the vapour pressure at trim condition is required in order to calculate the trim cavitation number.

The vapour pressure of seawater can be calculated from an empirical equation given by Robinson[11] as

$$\frac{P_v^0 - P_v}{P_v^0} = 9.206 \times 10^{-4}(Cl) + 2.36 \times 10^{-6} \tag{2.3.14}$$

where P_v^0 is the vapour pressure of pure water, P_v is the vapour pressure of the seawater and Cl is the chlorinity[4] of the seawater

$$P_v = P_v^0 - P_v^0 \left(9.206 \times 10^{-4}(‰Cl) + 2.36 \times 10^{-6}‰Cl^2\right) \tag{2.3.15}$$

Now taking the vapour pressure of pure water at 14 ^0C as 1.6 kPa and the chlorinity as 19.32‰, the vapour pressure is

$$P_v = 1.57 \text{ kPa} \tag{2.3.16}$$

The cavitation number can now be calculated from equation 2.2.3 using 2.3.8, 2.3.16 and 2.3.9 as,

$$\sigma = \frac{P_\infty - P_v}{\frac{1}{2}\rho V_0^2} = 230.44 \tag{2.3.17}$$

or alternatively,

$$V_0 \ll 18.22 \; m.s^{-1} \tag{2.3.18}$$

must hold to ensure cavitation does not occur.

[4]Chlorinity and salinity are related through the equation: $S = 1.80655\, Cl$

2.3.2 Comparison with UAV Operating Conditions

From a fluid dynamics standpoint there are few differences between the operation environment of small sub-sonic aircraft and that of a fully submerged submarine.

One major difference is the density of the operating medium. This, however, only comes into play in the motion equations where the dimensional values are used. The high density of water ensures that the operating medium can be considered as incompressible, which dramatically simplifies calculations. Due to this increased density the added mass terms[5] cannot be ignored as in the case with UAVs.

Another major difference is the difference in viscosity between air and water. The viscosity affects the Reynolds number, introduced earlier in this chapter, at which the vehicle operates.

[5]The concept of added mass is discussed in chapter 4

Chapter 3

Equations of Motion

This chapter constitutes the first of three chapters regarding the development of a nonlinear AUV model. In this chapter a generic set of equations to describe the motions of a rigid body with six degrees of freedom are developed.

3.1 Kinematics of Moving Reference Frames

Kinematics is the branch of dynamics primarily concerned with describing the motions of objects without any concern as to the forces and moments which generate the said motion.

3.1.1 Axis System Definitions

In order to derive a concise set of motion equations two axis systems are defined, one fixed in inertial space and the other fixed to the body of the vehicle being modelled.

Inertial Axis System

Since Newton's Laws will be used to describe the dynamics of the vehicle, and these laws are only valid with respect to an inertial reference frame, such a reference frame must be defined.

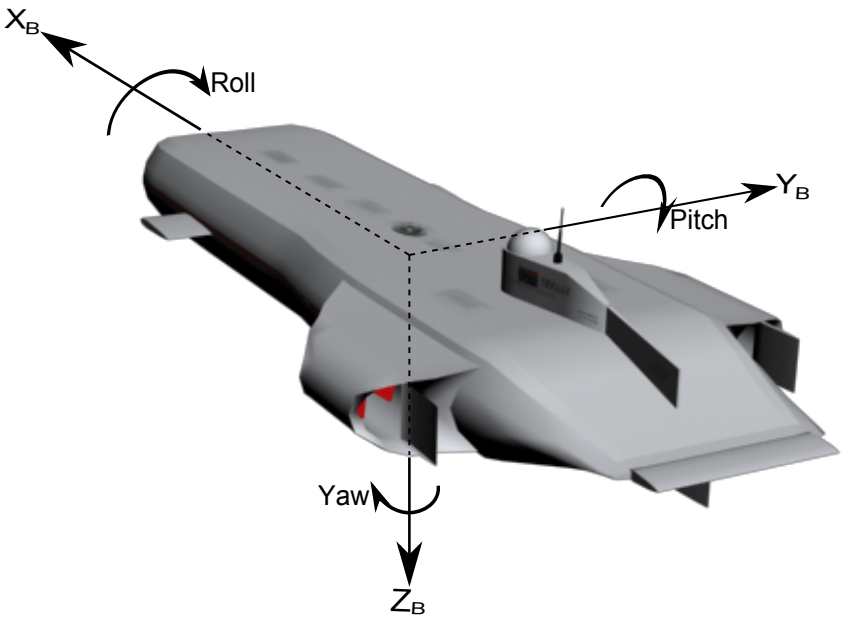

Figure 3.1: Body Fixed Axis System

It is common in these types of applications to use an earth fixed axis system as an approximation to the inertial reference frame. For this axis system to be inertial the earth is assumed to be flat and non-rotating. These assumptions are quite reasonable since, for the distances involved, the earth is nearly flat and the angular rates of the vehicle is much greater than the rotation of the earth.

The earth fixed axis system is defined as $F_E(O_E X_E Y_E Z_E)$. The origin of the earth fixed axis system is placed at some convenient point on the earth's surface O_E . In the case of underwater vehicle modelling, this will typically be on the surface of the water above the vehicle's starting position. The axis system is rectangular with the X_E axis pointing north, the Y_E pointing east and the Z_E axis pointing towards the centre of the earth.

The inertial axis system is said to span inertial space.

Body Axis System

The body axis system is a right handed reference frame with its origin affixed to the vehicle, therefore this axis system rotates and translates with the vehicle, and is defined as $F_B(O_B X_B Y_B Z_B)$. The X_B and Z_B axis lie in the vehicle's plane of symmetry, with the X_B axis positive pointing forward and the Z_B axis perpendicular to X_B and positive pointing downward. The Y_B axis lies perpendicular to the $X_B Z_B$ plane and is positive to starboard.

Three commonly used special cases follow, each simplifying a certain aspect of the modelling process considerably.

Reference Line Body Axis System One special case of the body axis system is where the X_B axis lies parallel to some longitudinal reference line. One very useful reference line is to choose the body axes system so that it coincides with the vehicle's water line. This has the effect of significantly simplifying the vehicles geometric description. In aircraft modelling the wing cord line is commonly used.

Wind Axis System When the forward direction is chosen to coincide with the velocity vector during a trim condition, the body fixed axis system is referred to as a wind fixed axes system. Compared to the reference line system it is rotated through an angle α_0 about the Y_B axis and a angle β_0 about the Z_B axis. This representation has the advantage of being the most closely related to fundamental hydrodynamic phenomena such as lift and drag, as these are both defined relative to the instantaneous free stream flow.

Stability Axis System When the forward direction is chosen to coincide with the projection of the velocity vector onto the vertical plane of symmetry during a trim condition, the body fixed axis system is referred to as a stability axes system. Compared to the reference line system, it is rotated through an angle α_0 about the Y_B axis. This representation has many similarities to the wind fixed axis system. In cases where initial symmetric flight is assumed, the trim stability axis system and the trim wind axis system are identical. This representation has the advantage

that hydrodynamic parameters are often supplied as stability axis referenced quantities.

3.1.2 Important Definitions

Before any further modelling can be conducted, a number of important definitions must be made. The following definitions are very standard, therefore little explanation is required.

Position The position of the vehicle relative to inertial space is defined as,

$$\boldsymbol{P} = P_N \boldsymbol{i}^{\boldsymbol{E}} + P_E \boldsymbol{j}^{\boldsymbol{E}} + P_D \boldsymbol{k}^{\boldsymbol{E}} \tag{3.1.1}$$

Linear Velocity The velocity of the vehicle relative to inertial space is given by,

$$\boldsymbol{V}^{\boldsymbol{B}} = U\boldsymbol{i}^{\boldsymbol{B}} + V\boldsymbol{j}^{\boldsymbol{B}} + W\boldsymbol{k}^{\boldsymbol{B}} \tag{3.1.2}$$

Angular Velocity The angular velocity of the vehicle with respect to inertial space is given by,

$$\boldsymbol{\omega}^{\boldsymbol{B}} = P\boldsymbol{i}^{\boldsymbol{B}} + Q\boldsymbol{j}^{\boldsymbol{B}} + R\boldsymbol{k}^{\boldsymbol{B}} \tag{3.1.3}$$

Force The force acting on the vehicle is given by,

$$\boldsymbol{F}^{\boldsymbol{B}} = X\boldsymbol{i}^{\boldsymbol{B}} + Y\boldsymbol{j}^{\boldsymbol{B}} + Z\boldsymbol{k}^{\boldsymbol{B}} \tag{3.1.4}$$

Moment The moment acting on the vehicle is given by,

$$\boldsymbol{M}^{\boldsymbol{B}} = K\boldsymbol{i}^{\boldsymbol{B}} + M\boldsymbol{j}^{\boldsymbol{B}} + N\boldsymbol{k}^{\boldsymbol{B}} \tag{3.1.5}$$

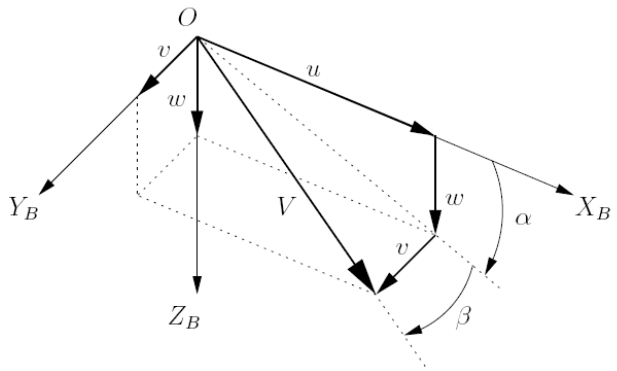

Figure 3.2: Linear Velocity in Polar Form[1]

Alternatively the velocity is often expressed in polar coordinates as,

$$\bar{V} = \sqrt{U^2 + V^2 + W^2} \tag{3.1.6}$$

$$\alpha = \tan^{-1}\left(\frac{W}{U}\right) \tag{3.1.7}$$

$$\beta = \sin^{-1}\left(\frac{V}{\bar{V}}\right) \tag{3.1.8}$$

where, $\bar{V}$ is the vehicle's speed, α its angle of attack and β its angle of sideslip.

It would be convenient to describe this relationship in terms of a transformation matrix and in order to do this a linear approximation for the relationship described by equations 3.1.6 to 3.1.8 is needed. For small angles of attack and sideslip the following approximation can be used,

$$\bar{V} \approx U \tag{3.1.9}$$

$$\alpha \approx \frac{W}{U} \tag{3.1.10}$$

$$\beta \approx \frac{V}{\bar{V}} \approx \frac{V}{U} \tag{3.1.11}$$

These equations can now be written in matrix form as,

$$\begin{bmatrix} \bar{V} \\ \beta \\ \alpha \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{U} & 0 \\ 0 & 0 & \frac{1}{U} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.1.12}$$

or

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & U \end{bmatrix} \begin{bmatrix} \bar{V} \\ \beta \\ \alpha \end{bmatrix} \tag{3.1.13}$$

3.1.3 Attitude Definition

In order to fully describe the vehicle's motions in six degrees of freedom, the vehicle's attitude must be known. In general, the vehicle's attitude can be considered as relative rotations of the body fixed axis system to the earth fixed axis system.

The rotation of one arbitrary axis system, B , relative to another arbitrary axis system, A, is given by the direction cosine matrix, or DCM for short. According to [12] $\mathbf{DCM^{AB}}$ is given by,

$$\begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} = \begin{bmatrix} \boldsymbol{i}^B.\boldsymbol{i}^A & \boldsymbol{j}^B.\boldsymbol{i}^A & \boldsymbol{k}^B.\boldsymbol{i}^A \\ \boldsymbol{i}^B.\boldsymbol{j}^A & \boldsymbol{j}^B.\boldsymbol{j}^A & \boldsymbol{k}^B.\boldsymbol{j}^A \\ \boldsymbol{i}^B.\boldsymbol{k}^A & \boldsymbol{j}^B.\boldsymbol{k}^A & \boldsymbol{k}^B.\boldsymbol{k}^A \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{3.1.14}$$

and, since $\mathbf{DCM^{AB}}$ is orthogonal, $\mathbf{DCM^{BA}}$ is given by,

$$\mathbf{DCM^{BA}} = [\mathbf{DCM^{AB}}]^T \tag{3.1.15}$$

Using the DCM in this manner is however inconvenient as it requires 9 parameters to be maintained at all times. This inconvenience can be overcome by parameterising the DCM using fewer parameters. Two common parametrisation methods are now briefly discussed.

Euler Angles vs Quaternions

One method used to describe this transformation matrix is by using a ordered set of three Euler angles. The Euler angle approach is commonly used since it describes the attitude of the body fixed coordinate system, relative to the earth fixed coordinate system, in an intuitive way. Using the Euler angle approach does have a drawback in that its use leads to a singularity in the kinematic equations at some attitude.

In many applications the presence of this singularity is acceptable, since by choosing the correct Euler combination it can be ensured to be outside of the vehicle's flight envelope. Good choices are at $\pm 90^o$ pitch or roll angle.

If the presence of such a singularity is unacceptable, such as with satellites, a four parameter approach is required. The most commonly used of these are Euler parameters, otherwise known as unit quaternions.

Since this project deals with marine vehicle modelling,in particular AUV modelling, the extra effort that accompanies the use of quaternions is unnecessary. The vehicle will never operate at 90^o pitch or roll angles so either one of these Euler angle combinations would suffice. In this project the Euler 321 combinations will be used.

DCM using Euler Angle Parametrisation

According to [12] the DCM can be parameterised using a set of three Euler angles and a order of rotation.

The DCM can be derived in terms of Euler angles by considering three primary rotations. As before, the relative rotation of two arbitrary rectangular axis systems, A and B, are considered. According to [12], the $\mathbf{DCM^{AB}}$ parameterised using Euler angles is given by,

$$\mathbf{DCM^{AB}} = \begin{bmatrix} c_\Psi c_\Theta & -s_\Psi c_\Phi + c_\Psi s_\Theta s_\Phi & s_\Psi s_\Phi + c_\Psi c_\Phi s_\Theta \\ s_\Psi c_\Theta & c_\Psi c_\Phi + s_\Phi s_\Theta s_\Psi & -c_\Psi s_\Phi + s_\Theta s_\Psi c_\Phi \\ -s_\Theta & c_\Theta s_\Phi & c_\Theta c_\Phi \end{bmatrix} \tag{3.1.16}$$

where, Φ is the roll angle, Θ is the pitch angle and Ψ is the yaw angle.

It is noted that as before this transformation is orthogonal, and therefore,

$$\mathbf{DCM^{BA}} = [\mathbf{DCM^{AB}}]^T \tag{3.1.17}$$

still holds.

Finally, define the Euler angle set as

$$\boldsymbol{e} = (\Phi, \Theta, \Psi) \tag{3.1.18}$$

3.1.4 Kinematic Equations for Linear Motion

A set of equations to describe the translational motion of the vessel, relative to the earth fixed axis system, in terms of the vehicle's body referenced velocity, must now be derived. Deriving these equations is a straight forward application of the transformation matrix given earlier. According to [12], the transformation of a body axis referenced vector to a earth fixed axis referenced vector is given by,

$$\mathbf{DCM^{EB}} = \mathbf{DCM^{AB}} \tag{3.1.19}$$

Using $\mathbf{DCM^{EB}}$, the following differential equation in obtained,

$$\begin{bmatrix} \dot{P_N} \\ \dot{P_E} \\ \dot{P_D} \end{bmatrix} = \begin{bmatrix} c_\Psi c_\Theta & -s_\Psi c_\Phi + c_\Psi s_\Theta s_\Phi & s_\Psi s_\Phi + c_\Psi c_\Phi s_\Theta \\ s_\Psi c_\Theta & c_\Psi c_\Phi + s_\Phi s_\Theta s_\Psi & -c_\Psi s_\Phi + s_\Theta s_\Psi c_\Phi \\ -s_\Theta & c_\Theta s_\Phi & c_\Theta c_\Phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.1.20}$$

3.1.5 Kinematic Equations for Rotational Motion

Since the time rate of change of the Euler angles do not equal the body referenced rotation rate, a transformation matrix $\boldsymbol{\omega} \Rightarrow \dot{\boldsymbol{e}}$ is needed to be able to update the Euler angles.

This is done by writing the body referenced angular rates in terms of

the three Euler angles as follows[13],

$$\boldsymbol{\omega} = \dot{\boldsymbol{\Phi}}\boldsymbol{i}_{\boldsymbol{\Phi}} + \dot{\boldsymbol{\Theta}}\boldsymbol{j}_{\boldsymbol{\Theta}} + \dot{\boldsymbol{\Psi}}\boldsymbol{k}_{\boldsymbol{\Psi}} \tag{3.1.21}$$

where

- $\boldsymbol{i}_{\boldsymbol{\Phi}}$ is the unit vector along the OX_2 axis.
- $\boldsymbol{j}_{\boldsymbol{\Theta}}$ is the unit vector along the OY_1 axis.
- $\boldsymbol{k}_{\boldsymbol{\Psi}}$ is the unit vector along the OZ_0 axis.

These unit vectors are now resolved into the body axis frame by applying the necessary transformations. The result is shown below.

$$\boldsymbol{\omega} = \boldsymbol{T}_{\boldsymbol{\Phi}}\boldsymbol{T}_{\boldsymbol{\Theta}} \begin{bmatrix} 0 \\ 0 \\ \frac{d\Psi}{dt} \end{bmatrix} + \boldsymbol{T}_{\boldsymbol{\Phi}} \begin{bmatrix} 0 \\ \frac{d\Theta}{dt} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{d\Phi}{dt} \\ 0 \\ 0 \end{bmatrix} \tag{3.1.22}$$

$$= \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \cos\Theta\sin\Psi \\ 0 & -\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{bmatrix} \frac{d\Phi}{dt} \\ \frac{d\Theta}{dt} \\ \frac{d\Psi}{dt} \end{bmatrix} \tag{3.1.23}$$

This matrix is not orthogonal, however, its inverse does exist and is given by,

$$\begin{bmatrix} \frac{d\Phi}{dt} \\ \frac{d\Theta}{dt} \\ \frac{d\Psi}{dt} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Psi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{3.1.24}$$

As stated previously this solution contains a singularity at $|\Theta| = \pi/2$, i.e. at $\pm 90^o$ pitch.

3.2 Kinetics of Rigid Bodies

In its broadest sense kinetics is the study of the effects of forces and moments on the motions of a body, thus, unlike in the case of kinematics,

the cause of the motion is of primary concern.

In this thesis Newton's second law is used to derive the six degree of freedom rigid body kinetic equations. This derivation is quite long and mathematically involved and as such is not shown here. The interested reader requiring a detailed derivation should kindly refer to appendix B, or consult one of the numerous published and unpublished works containing one.

3.2.1 Kinetic Equations for Linear Motion

From appendix B the kinetic equations for linear motion of point (x_g, y_g, z_g) is given in scalar form as,

$$\begin{aligned} X = m\big[&\dot{U} + QW - RV + \dot{Q}z_g - \dot{R}y_g + (Qy_g + Rz_g)P \\ &-(Q^2 + R^2)x_g\big] \end{aligned} \tag{3.2.1}$$

$$\begin{aligned} Y = m\big[&\dot{V} + RU - PW + \dot{R}x_g - \dot{P}z_g + (Rz_g + Px_g)Q \\ &-(R^2 + P^2)y_g\big] \end{aligned} \tag{3.2.2}$$

$$\begin{aligned} Z = m\big[&\dot{w} + PV - QU + \dot{P}y_g - \dot{Q}x_g + (Px_g + Qy_g)R \\ &-(P^2 + Q^2)z_g\big] \end{aligned} \tag{3.2.3}$$

where,

These equations can be significantly simplified by choosing the origin of the body fixed axis system to coincide with the centre of mass. After applying this simplification, the equations become,

$$X = m\left[\dot{U} + QW - RV\right] \tag{3.2.4}$$

$$Y = m\left[\dot{V} + RU - PW\right] \tag{3.2.5}$$

$$Z = m\left[\dot{W} + PV - QU\right] \tag{3.2.6}$$

3.2.2 Kinetic Equations for Rotational Motion

From appendix B the kinetic equations for rotational motion is given in scalar form as,

$$K = I_x\dot{P} + (I_z - I_y)QR - (\dot{R} + PQ)I_{xz} + (R^2 - ^2)I_{yz} + (PR - \dot{Q})I_{xy} + m\left[y_G(\dot{W} - UQ + VP) - z_G(\dot{V} - WP + UR)\right] \quad (3.2.7)$$

$$M = I_y\dot{Q} + (I_x - I_z)RP - (\dot{P} + QR)I_{xy} + (P^2 - R^2)I_{zx} + (QP - \dot{R})I_{yz} + m\left[z_G(\dot{U} - VR + WQ) - x_G(\dot{W} - WQ + VP)\right] \quad (3.2.8)$$

$$N = I_z\dot{R} + (I_y - I_x)PQ - (\dot{Q} + RP)I_{yz} + (Q^2 - P^2)I_{xy} + (RQ - \dot{P})I_{zx} + m\left[x_G(\dot{V} - WP + UR) - y_G(\dot{U} - VR + WQ)\right] \quad (3.2.9)$$

where terms such as I_x are elements of the inertial tensor. The inertial tensor can be considered as the rotational equivalent to mass and for a rigid body it is defined as,

$$I_B = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{zx} & I_y & -I_{yz} \\ -I_{zy} & -I_{zy} & I_z \end{bmatrix} \quad (3.2.10)$$

where,

$$I_x = \int_V (y^2 + z^2)\rho_A dV \qquad I_{xy} = I_{yx} = \int_V xy\rho_A dV \quad (3.2.11)$$

$$I_y = \int_V (x^2 + z^2)\rho_A dV \qquad I_{xz} = I_{zx} = \int_V xz\rho_A dV \quad (3.2.12)$$

$$I_z = \int_V (x^2 + y2)\rho_A dV \qquad I_{zy} = I_{yz} = \int_V yz\rho_A dV \quad (3.2.13)$$

Equations 3.2.7 to 3.2.9 can be significantly simplified by choosing the origin of the body fixed axis system to coincide with the centre of mass.

After applying this simplification, the equations become,

$$K = I_x\dot{P} + (I_z - I_y)QR - (\dot{R} + PQ)I_{xz} + (R^2 - Q^2)I_{yz} + (PR - \dot{Q})I_{xy} \tag{3.2.14}$$

$$M = I_y\dot{Q} + (I_x - I_z)RP - (\dot{P} + QR)I_{xy} + (P^2 - R^2)I_{zx} + (QP - \dot{R})I_{yz} \tag{3.2.15}$$

$$N = I_z\dot{R} + (I_y - I_x)PQ - (\dot{Q} + RP)I_{yz} + (Q^2 - P^2)I_{xy} + (RQ - \dot{P})I_{zx} \tag{3.2.16}$$

In most cases it is reasonable to assume that the vehicle is symmetric about the X_BZ_B plane. Assuming this results in the following simplified equations.

$$K = I_x\dot{P} + (I_z - I_y)QR - (\dot{R} + PQ)I_{xz} \tag{3.2.17}$$

$$M = I_y\dot{Q} + (I_x - I_z)RP + (P^2 - R^2)I_{zx} \tag{3.2.18}$$

$$N = I_z\dot{R} + (I_y - I_x)PQ + (RQ - \dot{P})I_{zx} \tag{3.2.19}$$

The remaining inertial cross coupling terms are now eliminated since these are generally much smaller than the diagonal terms, or zero if the body axis system coincides with the principle axis of inertia. In addition to this, these terms are generally insignificant from a control standpoint. After applying this simplification the equations become,

$$K = I_x\dot{P} + (I_z - I_y)QR \tag{3.2.20}$$

$$M = I_y\dot{Q} + (I_x - I_z)RP \tag{3.2.21}$$

$$N = I_z\dot{R} + (I_y - I_x)PQ \tag{3.2.22}$$

3.3 Summary

During this chapter a generic set of equations to describe the motion of the vehicle in six degrees of freedom were derived. These equations are summarised below.

The translational kinematic equation is given by;

$$\begin{bmatrix} \dot{P_N} \\ \dot{P_E} \\ \dot{P_D} \end{bmatrix} = \begin{bmatrix} c_\Psi c_\Theta & -s_\Psi c_\Phi + c_\Psi s_\Theta s_\Phi & s_\Psi s_\Phi + c_\Psi c_\Phi s_\Theta \\ s_\Psi c_\Theta & c_\Psi c_\Phi + s_\Phi s_\Theta s_\Psi & -c_\Psi s_\Phi + s_\Theta s_\Psi c_\Phi \\ -s_\Theta & c_\Theta s_\Phi & c_\Theta c_\Phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.3.1}$$

The rotational kinematic equation is given by;

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \cos\Theta\sin\Psi \\ 0 & -\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{3.3.2}$$

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Psi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{3.3.3}$$

The six degree of freedom rigid body kinetic equations are given by,

$$X = m(\dot{U} + QW + RV) \tag{3.3.4}$$

$$Y = m(\dot{V} + RU - PW) \tag{3.3.5}$$

$$Z = m([\dot{W} + PV - QU) \tag{3.3.6}$$

$$K = I_{x_C}\dot{P} + (I_{z_C} - I_{y_C})QR \tag{3.3.7}$$

$$M = I_{y_C}\dot{Q} + (I_{x_C} - I_{z_C})RP \tag{3.3.8}$$

$$N = I_{z_C}\dot{R} + (I_{y_C} - I_{x_C})PQ \tag{3.3.9}$$

Chapter 4

Forces and Moments

This chapter constitutes the second of three chapters concerned with the development of a nonlinear AUV model. In chapter 3 generic six degree of freedom rigid body equations of motion were derived. In this chapter, models to describe the driving inputs to these equations are now developed.

4.1 Forces and Moments Overview

There may be numerous contributors to the driving forces and moments. For this project it is assumed that the only driving inputs are due to thruster, hydrodynamic, gravity and buoyancy related forces and mommets. Thus, the driving inputs for the equations developed in this chapter are given by,

$$X = X_T + X_H + X_B + X_G \tag{4.1.1}$$

$$Y = Y_T + Y_H + Y_B + Y_G \tag{4.1.2}$$

$$Z = Z_T + Z_H + Z_B + Z_G \tag{4.1.3}$$

$$K = K_T + K_H + K_B + K_G \tag{4.1.4}$$

$$M = M_T + M_H + M_B + M_G \tag{4.1.5}$$

$$N = N_T + N_H + N_B + N_G \tag{4.1.6}$$

where subscript T indicates thruster, H hydrodynamic, B buoyancy and G gravity.

A model for each of these contributors must now be developed and is the topic of this chapter.

4.2 Hydrodynamic Forces and Moments

Deriving an accurate hydrodynamic model is one of the most difficult tasks in modelling an underwater vehicle, and in general also introduces the largest errors into the model.

It is more intuitive to describe the hydrodynamic forces and moments with reference to a wind or stability fixed coordinate system. The relationship between the hydrodynamic forces and moments referred to the wind fixed axis system and the body fixed axis system is given by[1],

$$X_H = -D^W \cos\alpha \cos\beta - Y^W \cos\alpha \sin\beta + L^W \sin\alpha \tag{4.2.1}$$

$$Y_H = -D^W \sin\beta + Y^W \cos\beta \tag{4.2.2}$$

$$Z_H = -D^W \sin\alpha \cos\beta + Y^W \sin\alpha \sin\beta - L^W \cos\alpha \tag{4.2.3}$$

$$K_H = K^W \cos\alpha \cos\beta - M^W \cos\alpha \sin\beta - N^W \sin\alpha \tag{4.2.4}$$

$$M_H = K^W \sin\beta + M^W \cos\beta \tag{4.2.5}$$

$$N_H = K^W \sin\alpha \cos\beta + M^W \sin\alpha \sin\beta + N^W \cos\alpha \tag{4.2.6}$$

where, α and β are the angle of attack and sideslip respectively.

In most cases the angle of sideslip is small and can safely be ignored,

[1] For details on wind axis transformations see Cook[14]

leading to the following simplified relationship.

$$X_H = X^S \cos\alpha - Z^S \sin\alpha \tag{4.2.7}$$
$$Y_H = Y^S \tag{4.2.8}$$
$$Z_H = Z^S \cos\alpha + X^S \sin\alpha \tag{4.2.9}$$
$$K_H = K^S \cos\alpha - N^S \sin\alpha \tag{4.2.10}$$
$$M_H = N^S \tag{4.2.11}$$
$$N_H = N^S \cos\alpha + K^S \sin\alpha \tag{4.2.12}$$

The forces of lift and drag are defined with respect to the free instantaneous velocity vector. Now, assuming that the angle of sideslip remains small, the drag and lift forces are given by.

$$D = -X^S \tag{4.2.13}$$
$$L = -Z^S \tag{4.2.14}$$

It is common to use a nondimensional representation when modelling hydrodynamic forces and moments. The primary reason for using such a representation is that the values in the model are independent of platform size and velocity. This simplifies comparing the model parameters for various vehicles. The following nondimensional representation is used throughout this project,

$$D = \bar{q} l^2 C_D \tag{4.2.15}$$
$$Y^S = \bar{q} l^2 C_Y \tag{4.2.16}$$
$$L = \bar{q} l^2 C_L \tag{4.2.17}$$
$$K^S = \bar{q} l^3 C_K \tag{4.2.18}$$
$$M^S = \bar{q} l^3 C_M \tag{4.2.19}$$
$$N^S = \bar{q} l^3 C_N \tag{4.2.20}$$

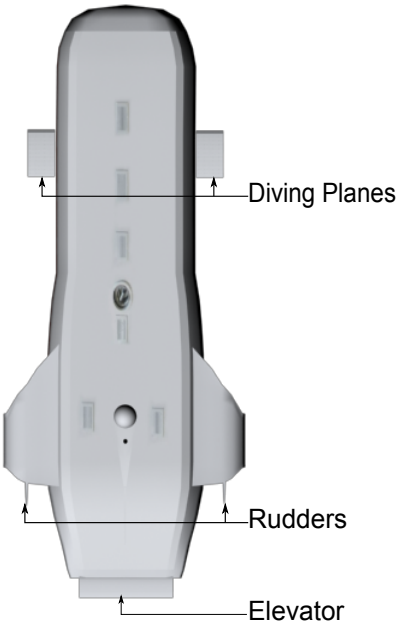

Figure 4.1: AUV Hydrodynamic Actuator Configuration

where l is the length of the vehicle and $\bar{q}$ is the kinetic pressure given by,

$$\bar{q} = \frac{1}{2}\rho \bar{V}^2 \tag{4.2.21}$$

4.2.1 Hydrodynamic Control Surfaces

In order to derive a hydrodynamic model, all the hydrodynamic control surfaces must first be defined. For the purpose of this project it is assumed that the vehicle has the following hydrodynamic actuator configuration:

- There is a diving plane on each side of the vessel. In order to improve flexibility of the model it is assumed that the diving planes can be used for common mode as well as differential mode control. Define the left and right diving plane deflection angle as δ_{dl} and δ_{dr} respectively, with a positive deflection angle defined so that it causes a negative pitching moment.

- There is a single large elevator at the rear of the vessel. The elevator acts as the primary pitch control device. Define the elevator deflection angle as δ_e, with a positive deflection angle defined so that it causes a negative pitching moment.
- The vessel has two rudders at the aft of the main body. These rudders are behind the thrusters and therefore rudder performance will be affected by the thrusters. For this project it is assumed that the rudders are only controlled in common mode and is the primary yaw control device. Define the rudder deflection angle as δ_r, with a positive deflection angle defined such that it causes a negative yawing moment.

This configuration is given graphically by figureA.4. The hydrodynamic control vector can now be defined as,

$$\delta_h = [\delta_{dl} \quad \delta_{dr} \quad \delta_e \quad \delta_r] \tag{4.2.22}$$

4.2.2 Hydrodynamic Derivatives

In this project the method of hydrodynamic derivatives is used to describe the hydrodynamic forces of moments. This in essence is simply a Taylor series expansion about a convenient reference condition. In order to significantly simplify the model, a number of assumptions are made. These are:

- The fluid forces and moments are assumed to be independent of position and orientation. For this to hold the ocean must be assumed to be homogenous and all wind and wave effects must be ignored.
- The fluid forces and moments are assumed to be independent of control derivative acceleration perturbations. This is a reasonable assumption, since the control surfaces have insignificant volume compared with the rest of the vessel and, as will later be shown, this type of derivative is typically a function of displacement.

It is common to approximate the perturbations in the various hydrodynamic coefficients caused by small deviations from the reference conditions to be linear in the perturbed motion and control variables. Showing only the linear terms, the expression for C_L is give by,

$$\begin{aligned} C_L = C_{L_0} &+ C_{L_{\bar{v}}}\bar{v} + C_{L_\alpha}\alpha + C_{L_\beta}\beta + C_{L_p}\hat{p} + C_{L_q}\hat{q} + C_{L_r}\hat{r} \\ &+ C_{L_{\dot{\bar{v}}}}\dot{\bar{v}} + C_{L_{\dot{\alpha}}}\dot{\alpha} + C_{L_{\dot{\beta}}}\dot{\beta} + C_{L_{\dot{p}}}\dot{\hat{p}} + C_{L_{\dot{q}}}\dot{\hat{q}} + C_{L_{\dot{r}}}\dot{\hat{r}} \\ &+ C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e + C_{L_{\delta_r}}\delta_r \end{aligned} \tag{4.2.23}$$

where,

$$\hat{p} = \frac{l}{2\bar{V}}p \tag{4.2.24}$$

$$\hat{q} = \frac{l}{2\bar{V}}q \tag{4.2.25}$$

$$\hat{r} = \frac{l}{2\bar{V}}r \tag{4.2.26}$$

In the same way the remaining 5 coefficients must be expanded about the reference condition.

Aircraft and marine vehicles typically exhibit various symmetries which can be used to significantly simplify the modelling of the vehicle. For the purpose of this project it is assumed that the vessel is symmetric about the $X_B Z_B$ plane.

This symmetry ensures that lateral forces and moments are independent of the longitudinal motion variables, and using this property ensures that all derivatives relating longitudinal motion to lateral forces and moments are zero. The same is not true for derivatives relating lateral motion to longitudinal forces and moments, but in order to completely decouple the longitudinal and lateral equations they are set to zero.

It is also noted that since symmetric initial flight is assumed, C_{Y_0}, C_{K_0} and C_{N_0} are all zero.

The decoupled model is now given by,

$$C_D = C_{D_0} + C_{D_{\bar{v}}}\bar{v} + C_{D_\alpha}\alpha + C_{D_{\hat{q}}}\hat{q} + C_{D_{\dot{\bar{v}}}}\dot{\bar{v}} + C_{D_{\dot{\alpha}}}\dot{\alpha} + C_{D_{\dot{\hat{q}}}}\dot{\hat{q}} + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{4.2.27}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_{\hat{p}}}\hat{p} + C_{Y_{\hat{r}}}\hat{r} + C_{Y_{\dot{\beta}}}\dot{\beta} + C_{Y_{\dot{\hat{p}}}}\dot{\hat{p}} + C_{Y_{\dot{\hat{r}}}}\dot{\hat{r}} + C_{Y_{\delta_{dl}}}\delta_{dl} + C_{Y_{\delta_{dr}}}\delta_{dr} + C_{Y_{\delta_r}}\delta_r \tag{4.2.28}$$

$$C_L = C_{L_0} + C_{L_{\bar{V}}}\bar{V} + C_{L_\alpha}\alpha + C_{L_{\hat{q}}}\hat{q} + C_{L_{\dot{\bar{V}}}}\dot{\bar{V}} + C_{L_{\dot{\alpha}}}\dot{\alpha} + C_{L_{\dot{\hat{q}}}}\dot{\hat{q}} + C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e \tag{4.2.29}$$

$$C_K = C_{K_\beta}\beta + C_{K_{\hat{p}}}\hat{p} + C_{K_{\hat{r}}}\hat{r} + C_{K_{\dot{\beta}}}\dot{\beta} + C_{K_{\dot{\hat{p}}}}\dot{\hat{p}} + C_{K_{\dot{\hat{r}}}}\dot{\hat{r}} + C_{K_{\delta_{dl}}}\delta_{dl} + C_{K_{\delta_{dr}}}\delta_{dr} + C_{K_{\delta_r}}\delta_r \tag{4.2.30}$$

$$C_M = C_{M_0} + C_{M_{\bar{V}}}\bar{V} + C_{M_\alpha}\alpha + C_{M_{\hat{q}}}\hat{q} + C_{M_{\dot{\bar{V}}}}\dot{\bar{V}} + C_{M_{\dot{\alpha}}}\dot{\alpha} + C_{M_{\dot{\hat{q}}}}\dot{\hat{q}} + C_{M_{\delta_{dl}}}\delta_{dl} + C_{M_{\delta_{dr}}}\delta_{dr} + C_{M_{\delta_e}}\delta_e \tag{4.2.31}$$

$$C_N = C_{N_\beta}\beta + C_{N_{\hat{p}}}\hat{p} + C_{N_{\hat{r}}}\hat{r} + C_{N_{\dot{\beta}}}\dot{\beta} + C_{N_{\dot{\hat{p}}}}\dot{\hat{p}} + C_{N_{\dot{\hat{r}}}}\dot{\hat{r}} + C_{N_{\delta_{dl}}}\delta_{dl} + C_{N_{\delta_{dr}}}\delta_{dr} + C_{N_{\delta_r}}\delta_r \tag{4.2.32}$$

Determining values for all the derivatives above would be a difficult and time consuming task, in particular the drag related terms are extremely difficult to obtain with reasonable accuracy. To help in this regard the following simplified expression for the drag coefficient can be used:

$$C_D = C_{D_{basic}}(\alpha) + C_{D_{\dot{\bar{v}}}}\dot{\bar{V}} + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{4.2.33}$$

where $C_{D_{basic}}(\alpha)$ gives an approximation of the vessel drag coefficient as a function of the angle of attack[2]. It is evident that this simplification makes the assumption that $C_{D_{\bar{v}}}$, $C_{D_{\hat{q}}}$, $C_{D_{\dot{\alpha}}}$ and $C_{D_{\dot{\hat{q}}}}$ are all insignificant.

Many of the other remaining derivatives are also insignificant. According to [15] the only important longitudinal derivatives are Z_w, Z_q, M_w and M_q wich are dimensional analogues to C_{L_α}, $C_{L_{\hat{q}}}$, C_{M_α} and $C_{M_{\hat{q}}}$.

[2]This function is not assumed to be linear

For the lateral derivatives [15] considers, Y_v, Y_r, N_v and N_r to be significant. In addition to this, L_p is considered significant, if banked turns are important. These lateral derivatives are dimensional analogous to C_{Y_β}, C_{Y_r}, C_{N_β}, C_{N_r} and C_{K_p} in the model used for this project. Now, showing only the significant terms, the hydrodynamic model becomes,

$$C_D = C_{D_{basic}}(\alpha) + C_{D_{\dot{\vec{v}}}}\dot{\vec{v}} + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{4.2.34}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_{\hat{\beta}}}\hat{r} + C_{Y_{\dot{\beta}}}\dot{\beta} + C_{Y_{\delta_r}}\delta_r \tag{4.2.35}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\hat{q} + C_{L_{\dot{\alpha}}}\dot{\alpha} + C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e \tag{4.2.36}$$

$$C_K = C_{K_p}\hat{p} + C_{K_r}\hat{r} + C_{K_{\dot{p}}}\dot{\hat{p}} + C_{K_{\delta_{dl}}}\delta_{dl} + C_{K_{\delta_{dr}}}\delta_{dr} \tag{4.2.37}$$

$$\begin{aligned} C_M = {} & C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\hat{q} + C_{M_{\dot{q}}}\dot{\hat{q}} + C_{M_{\delta_{dl}}}\delta_{dl} + C_{M_{\delta_{dr}}}\delta_{dr} \\ & + C_{M_{\delta_e}}\delta_e \end{aligned} \tag{4.2.38}$$

$$\begin{aligned} C_N = {} & C_{N_\beta}\beta + C_{N_p}\hat{p} + C_{N_r}\hat{r} + C_{N_{\dot{r}}}\dot{\hat{r}} + C_{N_{\delta_{dl}}}\delta_{dl} \\ & + C_{N_{\delta_{dr}}}\delta_{dr} + C_{N_{\delta_r}}\delta_r \end{aligned} \tag{4.2.39}$$

4.2.3 Added Mass

The terms such as $C_{L_{\dot{\alpha}}}\dot{\alpha}$ are commonly called added mass terms. Added (or virtual) mass forces and moments are simply pressure induced forces and moments which are proportional to acceleration. This has the same form as mass in Newton's equations and thus it is common to express this in terms of an added mass matrix which, is then added to the inertia matrix.

Added mass is often misunderstood as a finite amount of fluid which moves with the vehicle, and thus the total mass of the system is given by the summation of vehicle's mass and the mass of this finite amount of fluid. This is however incorrect since the motions of the vehicle places the entire fluid in motion with decaying amplitude.

It is entirely valid to simply leave the added mass terms as part of the hydrodynamic model, but doing so has a number of disadvantages. Amongst them are:

- Leaving the added mass terms in the hydrodynamic model dramatically increases the complexity of the model, since it is then also dependent on linear and angular accelerations.
- When the forces and moments are dependent on acceleration the solution contains algebraic loops. These algebraic loops cause difficulties during simulation.

In order to solve these problems the added mass terms are rather included in the dynamics equations derived in section 3.2, these equations now become,

$$X = (m - X_{\dot{u}})\dot{U} + m(QW - RV) \tag{4.2.40}$$
$$Y = (m - Y_{\dot{v}})\dot{V} + m(RU - PW) \tag{4.2.41}$$
$$Z = (m - Z_{\dot{w}})\dot{W} + m(PV - QU) \tag{4.2.42}$$
$$K = (I_{x_C} - K_{\dot{p}})\dot{P} + (I_{z_C} - I_{y_C})QR \tag{4.2.43}$$
$$M = (I_{y_C} - M_{\dot{q}})\dot{Q} + (I_{x_C} - I_{z_C})RP \tag{4.2.44}$$
$$N = (I_{z_C} - N_{\dot{r}})\dot{R} + (I_{y_C} - I_{x_C})PQ \tag{4.2.45}$$

And the hydrodynamic model, excluding the added mass terms, is given by.

$$C_D = C_{D_{basic}}(\alpha) + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{4.2.46}$$
$$C_Y = C_{Y_\beta}\beta + C_{Y_r}\hat{r} + +C_{Y_{\delta_r}}\delta_r \tag{4.2.47}$$
$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\hat{q} + C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e \tag{4.2.48}$$
$$C_K = C_{K_p}\hat{p} + C_{K_r}\hat{r} + C_{K_{\delta_{dl}}}\delta_{dl} + C_{K_{\delta_{dr}}}\delta_{dr} \tag{4.2.49}$$
$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\hat{q} + C_{M_{\delta_{dl}}}\delta_{dl} + C_{M_{\delta_{dr}}}\delta_{dr} + C_{M_{\delta_e}}\delta_e \tag{4.2.50}$$
$$C_N = C_{N_\beta}\beta + C_{N_p}\hat{p} + C_{N_r}\hat{r} + C_{N_{\delta_{dl}}}\delta_{dl} + C_{N_{\delta_{dr}}}\delta_{dr} + C_{N_{\delta_r}}\delta_r \tag{4.2.51}$$

As can be seen, the hydrodynamics model is now independent of acceleration and the dynamics equations are only slightly more complex than before.

4.2.4 Rudders Within the Thruster Exhaust Flow

When the rudders are located just behind the thrusters as in the case with the IMT vehicle, the model must take this into consideration. The effect of this configuration is that the flow velocity over the rudders is greater than the free stream velocity, which improves rudder performance.

In general, the effect of the thrusters on rudder performance can be complex and to determine an exact model would require significant experimentation and/or CFD analyses. Such a study is beyond the scope of this project. In order to derive a simple model the following assumptions are made:

- The flow is assumed constant and irrotational over the entire rudder span. This is true for a thruster with two counter rotating propellers, each with an infinite number of blades.
- The flow velocity over each rudder is identical and equal to the thruster exhaust velocity. This allows the rudders to modelled as a unit.
- Rudder forces and moments scale quadratically with the flow velocity over the rudder.

Keeping the above mentioned assumptions in mind, and noting that the thruster exhaust velocity is not considered as constant, the model is now given by,

$$C_D = C_{D_{basic}}(\alpha) + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{4.2.52}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_r}\hat{r} + \frac{V_T^2}{\bar{V}_0^2}C_{Y_{\delta_r}}\delta_r \tag{4.2.53}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\hat{q} + C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e \tag{4.2.54}$$

$$C_K = C_{K_p}\hat{p} + C_{K_r}\hat{r} + C_{K_{\delta_{dl}}}\delta_{dl} + C_{K_{\delta_{dr}}}\delta_{dr} \tag{4.2.55}$$

$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\hat{q} + C_{M_{\delta_{dl}}}\delta_{dl} + C_{M_{\delta_{dr}}}\delta_{dr} + C_{M_{\delta_e}}\delta_e \tag{4.2.56}$$

$$C_N = C_{N_\beta}\beta + C_{N_p}\hat{p} + C_{N_r}\hat{r} + C_{N_{\delta_{dl}}}\delta_{dl} + C_{N_{\delta_{dr}}}\delta_{dr} + \frac{V_T^2}{\bar{V}_0^2}C_{N_{\delta_r}}\delta_r \tag{4.2.57}$$

where, V_T is the thruster exhaust velocity. Equations to determine the exhaust velocity are derived later in this chapter.

4.2.5 Extended Range Model

The model derived up to this point is a standard small perturbation model. It is now attempted to extend this model to larger perturbations. The primary difficulty in modelling these larger perturbations is that the hydrodynamic coefficients can no longer be considered linear in the perturbed motion and control variables. This can be modelled in a number of ways, including:

- A higher order model can be used. A third order model is typically used as it offers a good compromise between accuracy and complexity.
- A model with non-constant derivatives can be used. Instead of constant derivatives, the derivatives are a function of the vehicle state.

In order to use a third order model, a large number of derivatives must be determined. Due to the lack of experimental data available on the IMT AUV this was deemed impractical. Thus the method of derivatives as a function of the vessel state is used in this project.

This turns out to be a relatively simple method of modelling perturbations outside of the linear range since the modelling software used supplies the stability derivatives in terms of trim velocity, Reynolds number and angle of incidence, or in the case of the control derivatives in terms of trim velocity, Reynolds number and deflection angle.

In order to further simplify the model it is assumed that the vessel is operated in a homogenous ocean at a fairly constant depth. Assuming this ensures that the Reynolds number is primarily a function of velocity. Through experimentation with the modelling software it was determined that at these extremely low Mach numbers the derivatives vary little with velocity. These two assumptions lead to a model where the

stability derivatives are simply an unknown function of the angle of incidence and the control derivatives an unknown function of the deflection angle.

Taking the above mentioned into account, the simplified extended model is given by:

$$\begin{aligned} C_D &= C_{D_{basic}}(\alpha) \\ &\quad + C_{D_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{D_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{D_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.2.58}$$

$$\begin{aligned} C_Y &= C_{Y_\beta}(\alpha)\beta + C_{Y_r}(\alpha)\hat{r} \\ &\quad + \frac{V_T^2}{\bar{V}_0^2} C_{Y_{\delta_r}}(\delta_r)\delta_r \end{aligned} \tag{4.2.59}$$

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}(\alpha)\alpha + C_{L_q}(\alpha)\hat{q} \\ &\quad + C_{L_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{L_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{L_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.2.60}$$

$$\begin{aligned} C_K &= C_{K_p}(\alpha)\hat{p} + C_{K_r}(\alpha)\hat{r} \\ &\quad + C_{K_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{K_{\delta_{dr}}}(\delta_{dr})\delta_{dr} \end{aligned} \tag{4.2.61}$$

$$\begin{aligned} C_M &= C_{M_0} + C_{M_\alpha}(\alpha)\alpha + C_{M_q}(\alpha)\hat{q} \\ &\quad + C_{M_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{M_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{M_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.2.62}$$

$$\begin{aligned} C_N &= C_{N_\beta}(\alpha)\beta + C_{N_p}(\alpha)\hat{p} + C_{N_r}(\alpha)\hat{r} \\ &\quad + C_{N_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{N_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + \frac{V_T^2}{\bar{V}_0^2} C_{N_{\delta_r}}(\delta_r)\delta_r \end{aligned} \tag{4.2.63}$$

4.3 Buoyancy and Gravity Related Forces and Moments

The gravity as well as the buoyancy model is now developed. After that the so called righting moment is briefly discussed. For the purpose of this project these forces and moments are considered fixed at design times and not controllable[3].

[3] In some circumstances this assumption my be incorrect, eg. for submersibles with a controllable ballast.

4.3.1 Gravity Related Forces and Moments

The gravitational force acts through the vehicles centre of gravity r_G, and its magnitude is given by,

$$W = mg \tag{4.3.1}$$

where m is the vehicle's mass, including the mass of water in flooded spaces, and g is the gravitational acceleration.

Noting that gravity is positive downwards, the gravitational force coordinated in the earth fixed reference frame is given by;

$$F_G^E = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \tag{4.3.2}$$

And the gravitational moment vector is given by,

$$M_G^E = r_G \times F_G^E \tag{4.3.3}$$

These equations can be written in terms of a body fixed coordinate system using the transformation matrix given by 3.1.19 as,

$$F_G^B = \begin{bmatrix} -W \sin\Theta \\ W \cos\Theta \sin\Phi \\ W \cos\Theta \cos\Phi \end{bmatrix} \tag{4.3.4}$$

$$M_G^B = \begin{bmatrix} y_G W \cos\Theta \cos\Phi - z_G W \cos\Theta \sin\Phi \\ -z_G W \sin\Theta - x_G W \cos\Theta \cos\Phi \\ x_G W \cos\Theta \sin\Phi + y_G W \sin\Theta \end{bmatrix} \tag{4.3.5}$$

Since the origin of the body fixed coordinate system was choosing to coincide with the vehicle's centre of mass and the vehicle is operated in an uniform gravitational field. The centre of gravity must coincides with the origin of the body fixed coordinate system, resulting in the following

simplified equations,

$$F_G^B = \begin{bmatrix} -W\sin\Theta \\ W\cos\Theta\sin\Phi \\ W\cos\Theta\cos\Phi \end{bmatrix} \tag{4.3.6}$$

$$M_G^B = 0 \tag{4.3.7}$$

4.3.2 Buoyancy Related Forces and Moments

The buoyancy force acts through the vehicle's centre of buoyancy r_B, and its magnitude is given by,

$$B = \rho g \Delta \tag{4.3.8}$$

where Δ is the volume of fluid displaced by the vehicle, ρ is the density of the fluid and g is the gravitational acceleration.

Noting that buoyancy is positive upwards, the buoyancy force coordinated in the earth fixed reference frame is give by;

$$F_B^E = \begin{bmatrix} 0 \\ 0 \\ -B \end{bmatrix} \tag{4.3.9}$$

And the buoyancy moment vector is given by,

$$M_B^E = r_B \times F_B^E \tag{4.3.10}$$

These equations can be written in terms of a body fixed coordinate

system using the transformation matrix given by 3.1.19 as,

$$F_B^B = \begin{bmatrix} B\sin\Theta \\ -B\cos\Theta\sin\Phi \\ -B\cos\Theta\cos\Phi \end{bmatrix} \tag{4.3.11}$$

$$M_B^B = \begin{bmatrix} -y_B B\cos\Theta\cos\Phi + z_B B\cos\Theta\sin\Phi \\ z_B B\sin\Theta + x_B B\cos\Theta\cos\Phi \\ -x_B B\cos\Theta\sin\Phi - y_B B\sin\Theta \end{bmatrix} \tag{4.3.12}$$

4.3.3 Restoring Moment

Most underwater vehicles use gravity and buoyancy to generate a so called restoring moment. This is achieved by placing the centre of buoyancy above, usually directly above, the centre of gravity. Assuming that the centre of gravity is at the origin and that the centre of buoyancy is directly above the centre of mass, summing the gravity and buoyancy related moments give.

$$M_G^B + M_B^B = \begin{bmatrix} z_B B\cos\Theta\sin\Phi \\ z_B B\sin\Theta \\ 0 \end{bmatrix} \tag{4.3.13}$$

Noting that for the case described above z_B is strictly negative, it can be seen that the restoring moment counteracts pitch and roll but has no effect on yaw.

4.4 Thruster Forces and Moments

A model for the AUV thrusters is now developed using a two step approach. Firstly a model for a single idealised thruster is developed, then two of these thrusters are combined into a twin thruster configuration as is used by the IMT vessel.

4.4.1 Model for a Single Idealised Thruster

In this project momentum theory is used to derive a thruster model which predicts the available thrust as a function of forward speed as well as predicting thruster exhaust velocity.

As noted by McCronmick and Barnes [16] the following assumptions are made in order to use classical momentum theory: the thrust loading is uniform over the entire propeller; the flow exiting the propeller is irrotational; a slipstream separates the flow passing through the propeller from the free-stream flow; far ahead and far behind the propeller the static pressure is equal to that of the free-stream static pressure.

Statically Thrusting Propeller

Consider a statically thrusting propeller, where the thrust produced by the thruster is linearly related to input power[4] and the input power is linearly related to the throttle position. For thruster x, this is given by;

$$\begin{aligned} T_{x_s} &= c_T P_x \\ P_x &= c_P \delta_t \end{aligned} \tag{4.4.1}$$

where, subscript s indicates static conditions, c_T is the power to thrust constant, P_x the input power, c_P the throttle to power constant and δ_t is the throttle position.

The velocity increment added at the statically thrusting propeller is given by[16]

$$w_{x_s} = \sqrt{\frac{T_s}{2\rho A}} \tag{4.4.2}$$

where A is the area of the propeller disk.

[4]See figure A.5 in order to verify this

Predicting Thrust as a Function of Forward Velocity

Now that the thrusters can be described at static conditions, momentum theory is used to derive a model for thrust and excused velocity as a function of forward velocity.

According to McCronmick and Barnes [16] the thrust produced by an idealised propeller is related to forward velocity by

$$\frac{T}{2T_{x_s}}\left\{\frac{U}{w_{x_s}}+\left[\frac{U}{w_{x_s}}^2+\frac{4T}{T_{x_s}}\right]^{\frac{1}{2}}\right\}=1 \tag{4.4.3}$$

where T_{x_s} and w_{x_s} is given by 4.4.1 and 4.4.2 respectively. Rewriting equation 4.4.1 in the cubic form gives,

$$T^3+\frac{UT_{x_s}^2}{w_{x_s}}T-T_{x_s}^3=0 \tag{4.4.4}$$

Now, applying the cubic equation to 4.4.4 yields the following three roots

$$T=\begin{cases} C_1+C_2 \\ -\frac{1}{2}(C_1+C_2)+\frac{1}{2}i\sqrt{3}(C_1-C_2) \\ -\frac{1}{2}(C_1+C_2)-\frac{1}{2}i\sqrt{3}(C_1-C_2) \end{cases} \tag{4.4.5}$$

where,

$$C_1=\sqrt[3]{C_4+\sqrt{C_3^3+C_4^2}} \qquad C_2=\sqrt[3]{C_4-\sqrt{C_3^3+C_4^2}} \tag{4.4.6}$$

$$C_3=\frac{UT_{x_s}^2}{3w_{x_s}} \qquad C_4=-\frac{T_{x_s}^3}{2} \tag{4.4.7}$$

Finally, it is noted that since thrust is strictly real, the first root must be taken.

Predicting Exhaust Velocity as a Function of Thrust and Forward Velocity

Since the vehicle hydrodynamic model is dependent on thruster exhaust velocity, an estimate of this is required. Again, a result from momentum theory is used. According to McCronmick and Barnes [16]

$$T = \rho\pi r^2 U_2(U_2 - U_0) \tag{4.4.8}$$

where U_0 and U_2 are the stream velocities far in front and behind the thruster respectively. Now, solving for U_2 from equation 4.4.8 gives the following two roots.

$$U_2 = \begin{cases} \frac{1}{2}\left(U_0 + \sqrt{U_0^2 + \frac{4T}{\rho\pi r^2}}\right) \\ \frac{1}{2}\left(U_0 - \sqrt{U_0^2 + \frac{4T}{\rho\pi r^2}}\right) \end{cases} \tag{4.4.9}$$

It is noted that since the velocity must be real, the root resulting in a real velocity must be selected.

Finally, assuming that the stream velocity at the rudder is approximately equal to the velocity through the propeller, U_1, we have,

$$V_T \approx U_1 = \frac{U_2 + U}{2} \tag{4.4.10}$$

4.4.2 Model for a Twin Thruster Configuration

The IMT vehicle uses a twin thruster configuration - one on each side at the aft of the AUV.

The left and right thrusters are located at r_l and r_r respectively. Assuming that the force vectors lie parallel to the $X_B Z_B$ plane and form an angle ϵ_T to the $X_B Y_B$ plane, the total force coordinated into the body axis

system is given by.

$$F_T^B = \begin{bmatrix} (T_l + T_r)\cos\epsilon_T \\ 0 \\ -(T_l + T_r)\sin\epsilon_T \end{bmatrix} \tag{4.4.11}$$

Unlike the idealised model derived so far, the actual thrusters also generate a significant rolling moment on the vessel. This is due the fact both turn in the same direction, and therefore friction torque does not cancel. It is not attempted to derive a model for this moment as very little data is available for the thrusters used on the IMT vehicle.

Taking only the thrust into account, the moment exerted on the vessel by the thrusters is given by

$$\begin{aligned} M_T^B &= r_{T_1} \times F_{T_1}^B + r_{T_2} \times F_{T_2}^B \\ &= \begin{bmatrix} -(y_{T_1}T_1 + y_{T_2}T_2)\sin\epsilon_T \\ (z_{T_1}T_1 + z_{T_2}T_2)\cos\epsilon_T + (x_{T_1}T_1 + x_{T_2}T_2)\sin\epsilon_T \\ -(y_{T_1}T_1 + y_{T_2}T_2)\cos\epsilon_T \end{bmatrix} \end{aligned} \tag{4.4.12}$$

4.5 Summary

During this chapter models to describe the forces and moments acting on the vehicle were derived. These are summarised below.

The kinetic equations, including added mass effects, are given by,

$$X = (m - X_{\dot{u}})\dot{U} + m(QW - RV) \tag{4.5.1}$$
$$Y = (m - Y_{\dot{v}})\dot{V} + m(RU - PW) \tag{4.5.2}$$
$$Z = (m - Z_{\dot{w}})\dot{W} + m(PV - QU) \tag{4.5.3}$$
$$K = (I_{x_C} - K_{\dot{p}})\dot{P} + (I_{z_C} - I_{y_C})QR \tag{4.5.4}$$
$$M = (I_{y_C} - M_{\dot{q}})\dot{Q} + (I_{x_C} - I_{z_C})RP \tag{4.5.5}$$
$$N = (I_{z_C} - N_{\dot{r}})\dot{R} + (I_{y_C} - I_{x_C})PQ \tag{4.5.6}$$

The hydrodynamic coefficients are given by following set of equations.

$$\begin{aligned} C_D = {} & C_{D_{basic}}(\alpha) \\ & + C_{D_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{D_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{D_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.5.7}$$

$$\begin{aligned} C_Y = {} & C_{Y_\beta}(\alpha)\beta + C_{Y_r}(\alpha)\hat{r} \\ & + \frac{V_T^2}{\bar{V}_0^2} C_{Y_{\delta_r}}(\delta_r)\delta_r \end{aligned} \tag{4.5.8}$$

$$\begin{aligned} C_L = {} & C_{L_0} + C_{L_\alpha}(\alpha)\alpha + C_{L_q}(\alpha)\hat{q} \\ & + C_{L_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{L_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{L_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.5.9}$$

$$\begin{aligned} C_K = {} & C_{K_p}(\alpha)\hat{p} + C_{K_r}(\alpha)\hat{r} \\ & + C_{K_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{K_{\delta_{dr}}}(\delta_{dr})\delta_{dr} \end{aligned} \tag{4.5.10}$$

$$\begin{aligned} C_M = {} & C_{M_0} + C_{M_\alpha}(\alpha)\alpha + C_{M_q}(\alpha)\hat{q} \\ & + C_{M_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{M_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + C_{M_{\delta_e}}(\delta_e)\delta_e \end{aligned} \tag{4.5.11}$$

$$\begin{aligned} C_N = {} & C_{N_\beta}(\alpha)\beta + C_{N_p}(\alpha)\hat{p} + C_{N_r}(\alpha)\hat{r} \\ & + C_{N_{\delta_{dl}}}(\delta_{dl})\delta_{dl} + C_{N_{\delta_{dr}}}(\delta_{dr})\delta_{dr} + \frac{V_T^2}{\bar{V}_0^2} C_{N_{\delta_r}}(\delta_r)\delta_r \end{aligned} \tag{4.5.12}$$

The gravity model is given by,

$$F_G^B = \begin{bmatrix} -W\sin\Theta \\ W\cos\Theta\sin\Phi \\ W\cos\Theta\cos\Phi \end{bmatrix} \tag{4.5.13}$$

$$M_G^B = 0 \tag{4.5.14}$$

The buoyancy model is given by,

$$F_B^B = \begin{bmatrix} B\sin\Theta \\ -B\cos\Theta\sin\Phi \\ -B\cos\Theta\cos\Phi \end{bmatrix} \tag{4.5.15}$$

$$M_B^B = \begin{bmatrix} -y_B B\cos\Theta\cos\Phi + z_B B\cos\Theta\sin\Phi \\ z_B B\sin\Theta + x_B B\cos\Theta\cos\Phi \\ -x_B B\cos\Theta\sin\Phi - y_B B\sin\Theta \end{bmatrix} \tag{4.5.16}$$

And finally, the twin thruster model is given by,

$$F_T^B = \begin{bmatrix} (T_l + T_r)\cos\epsilon_T \\ 0 \\ -(T_l + T_r)\sin\epsilon_T \end{bmatrix} \tag{4.5.17}$$

$$\begin{aligned} M_T^B &= r_{T_1} \times F_{T_1}^B + r_{T_2} \times F_{T_2}^B \\ &= \begin{bmatrix} -(y_{T_1}T_1 + y_{T_2}T_2)\sin\epsilon_T \\ (z_{T_1}T_1 + z_{T_2}T_2)\cos\epsilon_T + (x_{T_1}T_1 + x_{T_2}T_2)\sin\epsilon_T \\ -(y_{T_1}T_1 + y_{T_2}T_2)\cos\epsilon_T \end{bmatrix} \end{aligned} \tag{4.5.18}$$

Chapter 5

Calculating Model Parameters

In this chapter techniques used to derive the various model parameters for the model developed in chapters 3 and 4 are discussed. These techniques are then used to calculate the parameters for the IMT AUV.

5.1 Inertial Parameters

Measuring these parameters for the IMT vessel proved problematic since the vessel employs a flooded body design. This means that the vehicle can not simply be weighed to determine the mass, and the principle moments of inertia can not be measured using the double pendulum test[1].

The mass has to be calculated from the vessel displacement, the density of the seawater and the total net buoyancy[2], using the following equation,

$$m = \rho\Delta - \frac{B}{g} \tag{5.1.1}$$

Now, taking the vessel displacement as 0.832 m^3, the density of seawater

[1] For information on the double pendulum test see [13]

[2] Net positive buoyancy can easily be measured at the surface by weighing the vessel down.

as 1026 $kg.m^{-3}$ and the net buoyancy as 98.1 N, the mass becomes,

$$m = 843.63 \text{ kg} \tag{5.1.2}$$

In order to determine the vessel's principle moments of inertia, the vessel was drawn in Autodesk Inventor from measurements, photos and sketches. In calculating the vessel moments of inertia it is assumed that the vehicle is of uniform density. This assumption, although not entirely correct, is reasonable, since the vast majority of the vehicle is flooded with water. In addition to this, making this assumption should have little effect from a control engineering standpoint. Using this method, and taking the mass as 843.63 kg, the following values are obtained.

$$I_x = 65.05 \text{ kg}.m^2 \qquad I_y = 497.24 \text{ kg}.m^2 \qquad I_z = 547.00 \text{ kg}.m^2 \tag{5.1.3}$$

5.2 Hydrodynamic Model Parameters

In this section the parameters required by the hydrodynamic model developed in chapter 4 are calculated. In order to determine which properties can be neglected the hydrodynamic similarity parameters and flight conditions were discussed and calculated in chapter 2.

5.2.1 Linear Hydrodynamic Derivatives

' The linear hydrodynamic derivatives required by the model derived in chapter 4 must now be determined. A variety of different tools and techniques have been developed over the years for the purpose of calculating hydrodynamic derivatives. These tools and/or techniques can typically be arranged into one of the following groups:

- **Experimentation based estimation**: As previously stated, access to the vehicle was extremely limited. Therefore these type of methods were not considered.

- **Empirical methods**: A variety of empirical estimation methods were developed over the years. Most of these system date back to the time before powerful computes existed. Some of them were widely used, and tested, at that time, and therefore should prove reasonably accurate.

- **Computational fluid dynamics**: Modern CFD based systems employ a variety of numerical methods in order to solve complex fluid dynamical problems. These numerical methods are in general computationally expensive, and therefore powerful computer systems are required. In addition to this most CFD packages are prohibitively expensive, and as such were not considered for this project.

From the arguments given above it is self evident that some form of empirical estimation must be used. Although this decision is taken more out of necessity than out of choice, these methods should be reasonable accurate. For more information on some of these methods please refer to [15].

One the most widely used of theses empirical systems, the USAF stability and control Datcom, is used for this project.

5.2.1.1 USAF Stability and Control Datcom

The USAF stability and control data compendium was developed starting in the early 60's by McDonnell Douglas under contract from the United States Air Force. It constitutes a extensive collection of analytical and empirical methods which have been verified against a large amount of experimental data.

There are numerous reasons for using the Datcom for parameter estimation. Some of them are,

- Viscosity effects can be accounted for by specifying the Reynolds Number.

- The Datcom has previously been used for estimating hydrodynamic parameters with reasonable accuracy.

- A large part of the effort involved with using the Datcom is due to compressibility of the operating medium - since water is incompressible this leads to a large reduction in effort involved.
- A computerised version of the Datcom, called the Digital Datcom, can be used to avoid the lengthy calculations involved with using the Datcom.

5.2.1.2 Characteristics from the Datcom

The following hydrodynamic stability derivatives are output by the Digital Datcom for a Wing-Body, Horizontal Tail, Vertical Tail configuration with straight tapered surfaces,

- C_D: Total drag coefficient at angle of attack in degrees.
- C_L: Total lift coefficient at angle of attack in degrees.
- C_{L_α}: Lift coefficient due to angle of attack.
- C_{L_q}: Lift coefficient due to pitch rate.
- C_{Y_β}: Side force due to angle of sideslip.
- C_{Y_p}: Side force due to roll rate.
- C_{K_β}: Rolling moment due to angle of sideslip.
- C_{K_p}: Rolling moment due to roll rate.
- C_M: Total pitching moment at alpha in degrees.
- C_{M_α}: Pitching moment due to angle of attack.
- C_{M_q}: Pitching moment due to pitch rate.
- C_{N_β}: Yawing moment due to angle of sideslip.
- C_{N_r}: Yawing moment due to yaw rate.

In addition to these stability derivatives, the following coefficient increments due to control surface deflection are also supplied by the Datcom,

- δC_{L_E}: Lift coefficient increment due to elevator deflection.
- δC_{K_A}: Rolling moment coefficient increment due to aileron deflection.
- δC_{M_E}: Pitching moment coefficient increment due to elevator deflection.
- δC_{N_A}: Yawing moment coefficient increment due to aileron deflection.

As can be seen from the list above, there are derivatives which the Datcom does not provide. The most important amongst these are the effect of the rudder on the yawing moment as well as the rudders effect on side force.

Although the Datcom also provides an estimate for some Added Mass derivatives such as $C_{L_{\dot{\alpha}}}$, the literature advises against using these Datcom methods and suggest using other methods to derive an added mass model.

5.2.1.3 Basic Characteristics

Using the Digital Datcom to calculate the significant stability derivatives proved relatively straightforward for the most part, with the exception of modelling the vertical tail. This difficulty arises in that the Digital Datcom can supply derivatives for a configuration where the vertical tail is above the reference plane or where the vertical tail is below the reference line, but not a configuration where the vertical tail is above and below the reference line as in the case with the vehicle under study.

The vertical tail configuration used primarily affects the lateral motions of the vehicle. With a normal high tail as used on most aircraft the asymmetric vertical tail induces, for example, a rolling moment on the vehicle during a sideslip condition. Using a more symmetric tail such

as on the IMT AUV this effect is greatly reduced. The tail configuration also has a small effect on the longitudinal model in the form of slightly increased drag.

This problem can be solved in a fairly simple manner by combining three different configurations. They are:

- A configuration without a vertical tail, from here on referred to as the base configuration.
- A configuration with a vertical tail above the reference plane, from here on this configuration is referred to as the high-tail configuration.
- And a configuration with a vertical tail below the reference plane, from here on this configuration is referred to as the low-tail configuration.

Now, making the assumption that the two tail panels have no effect on each other, the complete model is given by[3]

$$Combined\ Tail = Base + (High\ Tail - Base) + (Low\ Tail - Base) \quad (5.2.1)$$

Figure 5.1 and 5.2 shows the basic vehicle hydrodynamic characteristics calculated by the Digital Datcom, with all derivatives given in per radian. With reference to these figures, the following important observations are made:

- The lift curve slope, C_{L_α} is nearly invariant for angles of attack between approximately -8^o and 8^o. This is known as the linear lift region.
- Since C_{M_α} is positive, the vehicle is hydrodynamically, statically unstable for longitudinal motion.

[3]This equation is correct in general, since *High Tail* and *Low Tail* contains all geometry form *Base*. Therefore, derivatives not effected by the tail configuration remain unchanged.

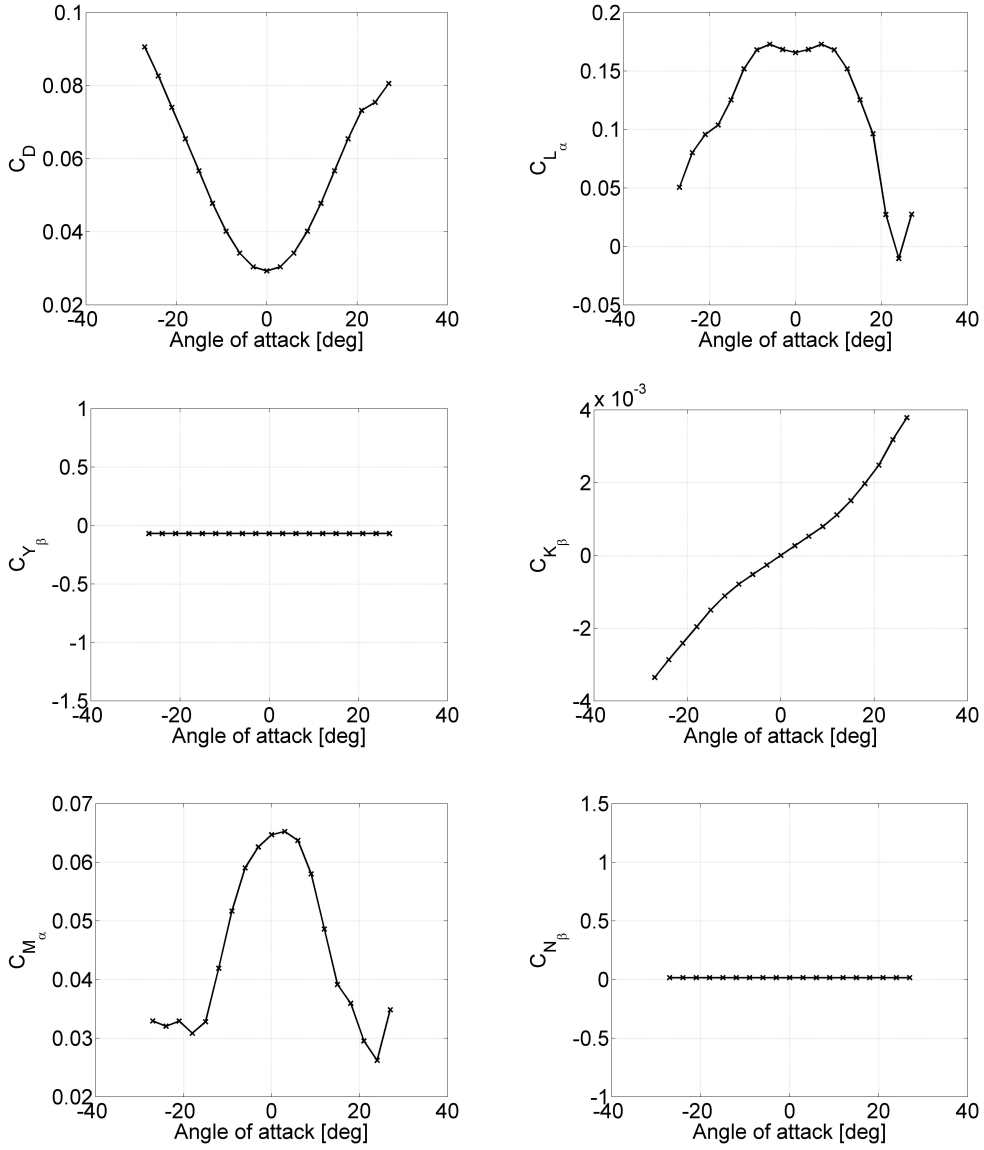

Figure 5.1: Static Stability Derivatives

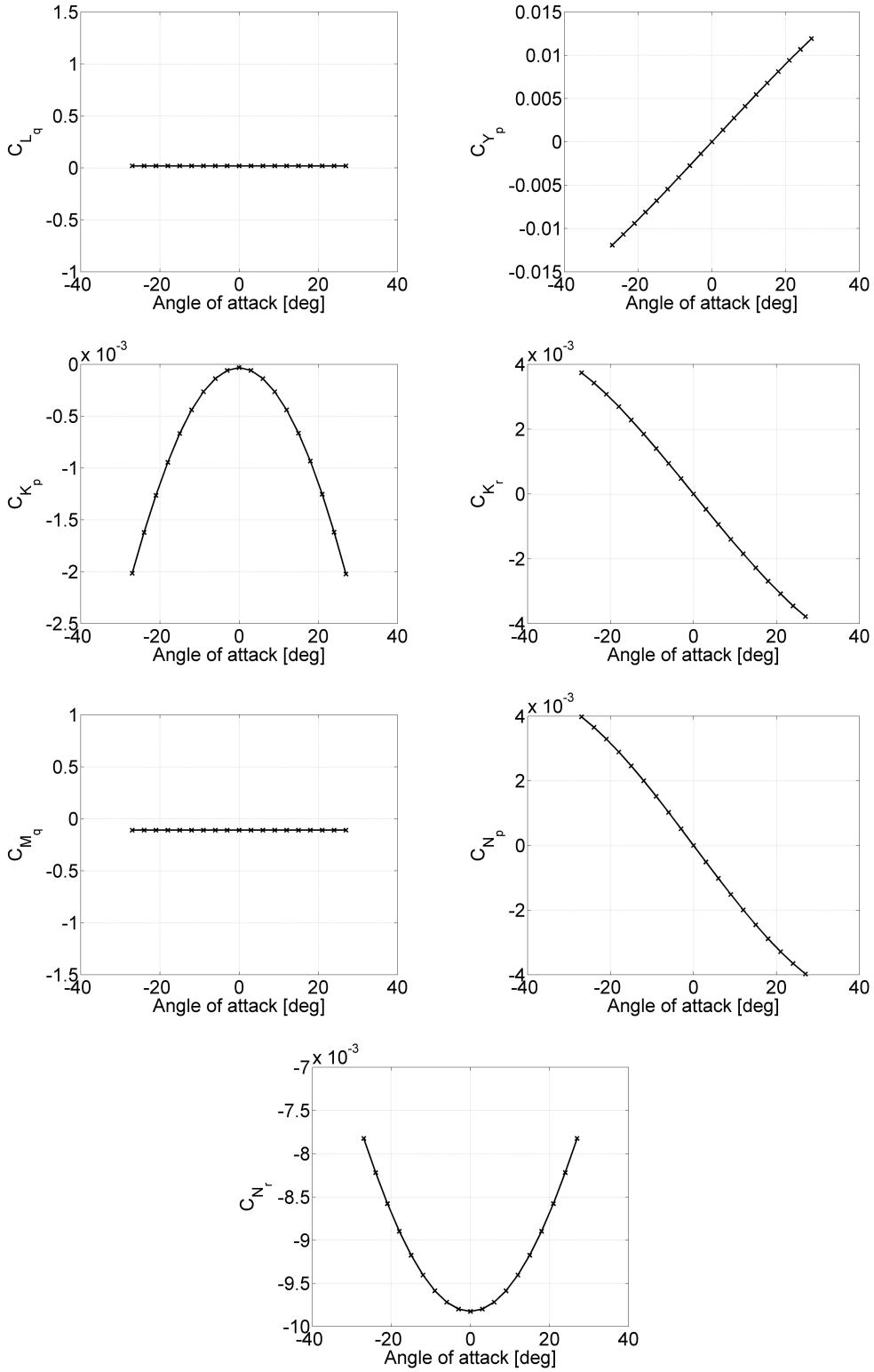

Figure 5.2: Dynamic Stability Derivatives

Parameter	IMT Datcom	ARCS Datcom	ARCS Tests
C_{M_α}	0.065	0.0009	0.0045
C_{Y_β}	-0.07	-0.045	-0.049
C_{N_β}	0.016	-0.0013	-0.0011

Table 5.1: Comparison with ARCS Vehicle

- It is also noted that since C_{N_β} is positive, the vessel is statically directionally stable.

In order to demonstrate that the data obtained is reasonable, it is compared with data for the ARCS vehicle[17]. This comparison is presented in table 5.1. While comparing the data the following should be noted:

- Unlike the IMT vehicle, the ARCS vehicle is equipped with rear planes. These provide a significant stabilising effect.
- The data given in [17] is in terms of normal and lateral velocities and not in terms of angle of attack and sideslip. It is assumed that these representations are equivalent.

Diving-Plane Characteristics

In order to increase the flexibility of the model, it is assumed that the diving planes can be used to provide both symmetric and asymmetric control. The major difficulty with this assumption arises in that control surfaces in Digital Datcom can be defined as either symmetrical or asymmetrical, but not both. This limitation can easily be overcome using the following method:

- Model the symmetrical control characteristics by defining the diving planes as a plain symmetrical control device, i.e. elevator. The digital datcom assumes that symmetrical control surfaces are placed on the most aft lifting surface, thus in order to force the correct placement no horizontal tail may be defined.

- Model the asymmetrical control characteristics by defining the diving planes as plain asymmetrical control surfaces, i.e. ailerons. All asymmetrical control devices, with the exception of the all-moveable horizontal tail, are defined for the main wing even when a horizontal tail is defined.

- Combine the symmetrical and asymmetrical characteristics in an appropriate way to obtain the complete model.

The first two steps stated above are relatively straightforward and require no further explanation[4], but the final step though is non-trivial and warrants further explanation. When combining the two models the following should be noted:

- Digital Datcom assumes that all control surfaces are symmetric about the $X_B Z_B$ plane, thus the derivatives supplied by the Digital Datcom are for left and right diving planes combined.

- Since the derivatives supplied by the datcom are for both diving planes, the Datcom values must be halved to obtain the individual diving plane derivatives.

- Finally, it must be noted that the rolling moment due to diving plane defection supplied by the asymmetric diving plane Datcom case, $C_{K_{\delta_d}}$, is defined such that a positive deflection causes a negative rolling moment, while the individual diving plane deflection are defined in terms of pitching moment, as was stated in chapter 3.

[4]For further information please consult the Digital Dactcom User Manual [18]

Noting the above, the diving plane derivatives are calculated from the symmetric and asymmetric data as follows.

$$
\begin{aligned}
C_{D_{\delta_{dl}}}(\delta_{dl}) &= \frac{1}{2}C_{D_{\delta_d}}(\delta_d) & C_{D_{\delta_{dr}}}(\delta_{dr}) &= \frac{1}{2}C_{D_{\delta_d}}(\delta_d) \\
C_{L_{\delta_{dl}}}(\delta_{dl}) &= -\frac{1}{2}C_{L_{\delta_d}}(\delta_d) & C_{L_{\delta_{dr}}}(\delta_{dr}) &= -\frac{1}{2}C_{L_{\delta_d}}(\delta_d) \\
C_{M_{\delta_{dl}}}(\delta_{dl}) &= -\frac{1}{2}C_{M_{\delta_d}}(\delta_d) & C_{M_{\delta_{dr}}}(\delta_{dr}) &= -\frac{1}{2}C_{M_{\delta_d}}(\delta_d) \\
C_{K_{\delta_{dl}}}(\delta_{dl}) &= -\frac{1}{2}C_{K_{\delta_d}}(\delta_d) & C_{K_{\delta_{dr}}}(\delta_{dr}) &= \frac{1}{2}C_{K_{\delta_d}}(\delta_d)
\end{aligned} \tag{5.2.2}
$$

Figure 5.3 shows the diving plane characteristics calculated from the symmetric and asymmetric diving plane cases, using 5.2.2. [5] With reference to this figure, the following observations are made:

- Between approximately -15^o and 15^o, the diving planes provide nearly linear control actuation.
- The left and right diving plane roll coefficient increments are mirror images of each other. This is due to the definition of a positive diving plane deflection.
- Stall effects are visible at extreme deflections.

Elevator Characteristics

Modelling of the vehicle elevator is straightforward and warrants no in-detail explanation. It simply entails the placement of a horizontal tail and plain symmetrical flaps of the proper geometry at the rear of the vessel. Figure 5.4 shows the elevator characteristics obtained from the Digital Datcom. With reference to this figure the following observations are made:

- The elevator provides nearly linear actuation between approximately -15^o and 15^o.

[5] The digital Datcom does not supply control device characteristics in terms of control derivatives, but rather as a coefficient increment due to deflection.

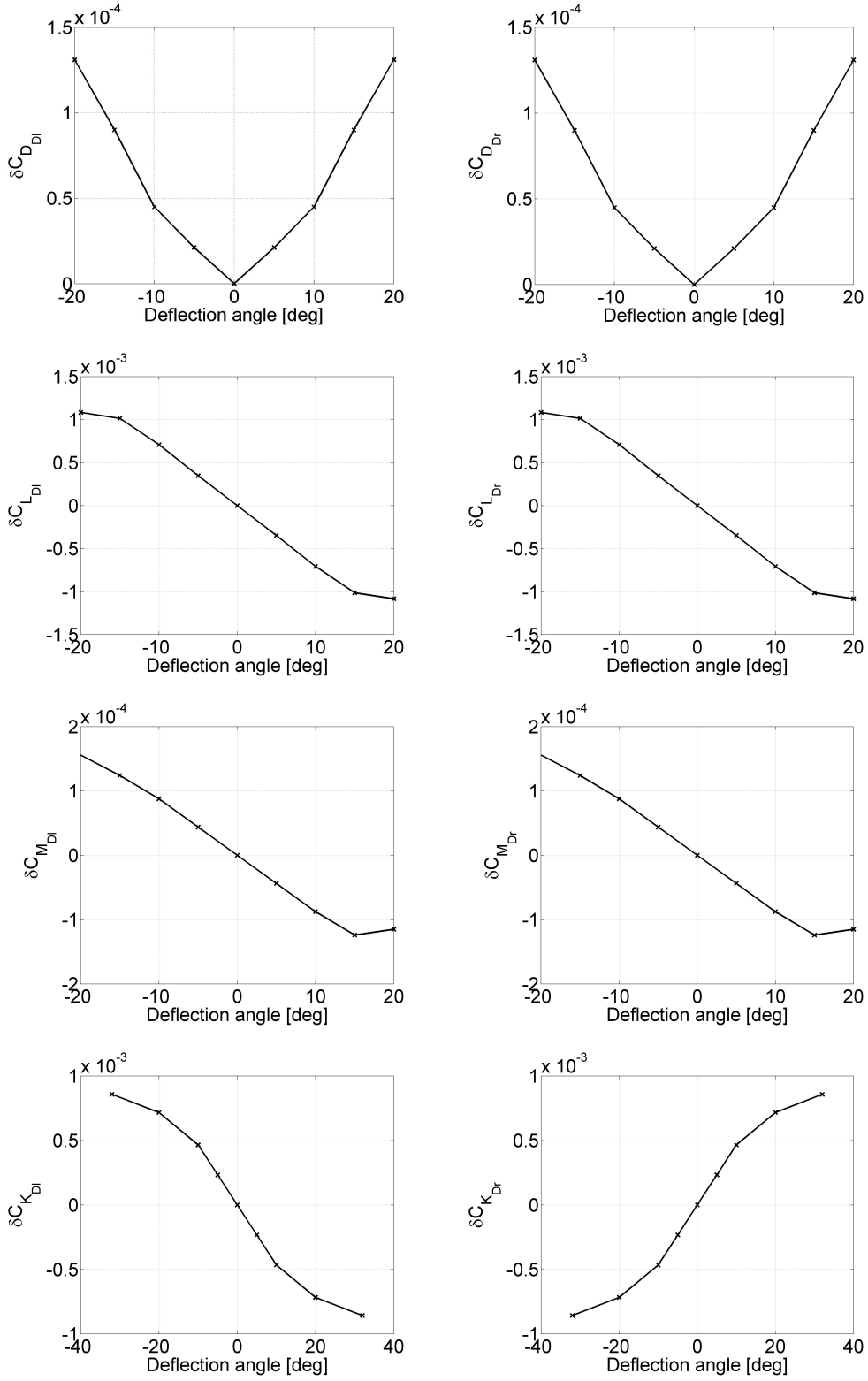

Figure 5.3: Diving-Plane Derivatives

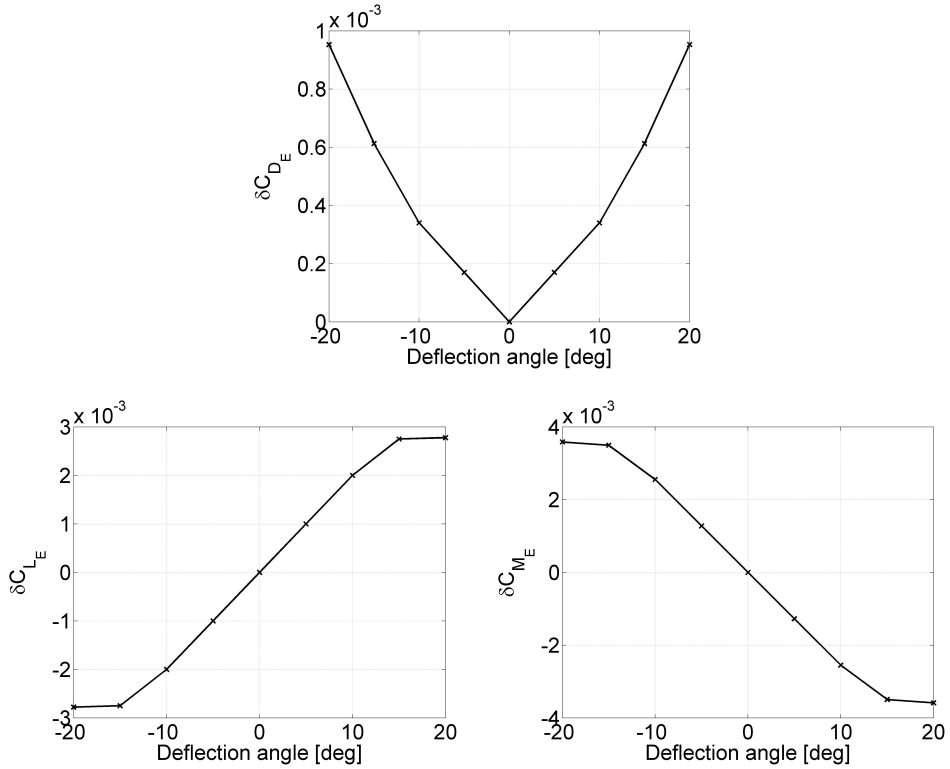

Figure 5.4: Elevator Derivatives

- At lager deflections, stall effects become visible.

Rudder Characteristics

Unlike the elevator, modelling of the vehicle rudders is non-trivial as the Digital Datcom does not supply any vertical control surfaces. Since the rudders are extremely important to the operation of the vehicle and thus could not simply be ignored, a solution to this problem had to be found.

A number of different solutions were considered, amongst them are

- Using standard aerofoil equations to calculate the side force and yawing moment due to deflection.
- Using some other modelling software to compute the rudder characteristics.

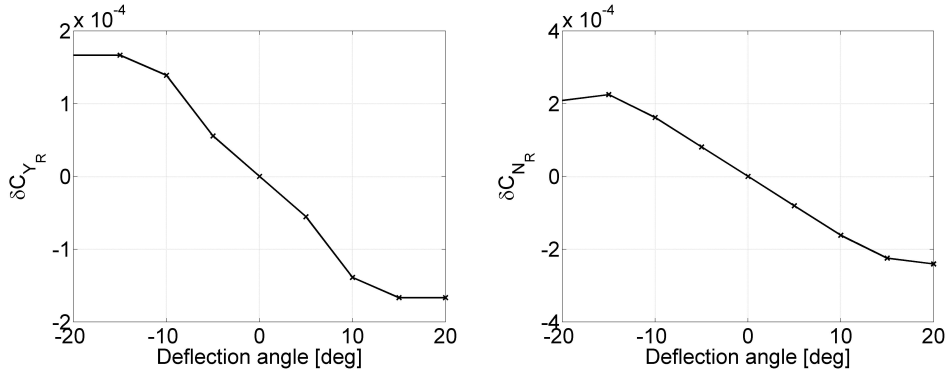

Figure 5.5: Rudder Derivatives

- Model the rudders with the Digital Datcom using the available control devices is a non-standard method.

In this project the third option was chosen, as this has the advantage of using the same well tested modelling software throughout. In order to use the available Digital Datcom control devices, and noting that rudders are the lateral equivalent to elevators ,it is assumed that rudders and elevators are identical, except for the orientation of the control surface.

Keeping this assumption in mind the rudders are modelled using Digital Datcom as follows,

- Place elevators of correct geometry at the same location as the rudders.
- Rotate the coefficient increments by 90^o, i.e lift becomes side force and pitching moment becomes yawing moment.
- Scale the yawing moment coefficient increment by the appropriate amount in order to compensate for the different pitching and yawing moment arms.

Figure 5.5 shows the rudder characteristics obtained by using the method described above. With reference to this figure the following observations are made:

- As with the other control devices, the rudders provide nearly linear actuation between approximately -15^o and 15^o.[6]
- Beyond this linear range,stall effects become visible.

5.2.2 Added Mass Derivatives

In this section equations to calculate the added mass terms are developed. Two common methods used in the calculation of added mass coefficients for submersibles are discussed: first using a strip theory approximation is considered and secondly an ellipsoidal approximation is investigated.

Added Mass Using Strip Theory Approximations

One common method[2] used to calculate the added mass terms is to use strip theory. This method is only accurate for streamlined bodies, and thus it cannot be expected to be highly accurate for the IMT vehicle. According to Fossen[2] the following formulae can be used for a fully

[6]The steps in the data is due to low numeric resolution

submerged slender body,

$$X_{\dot{u}} = -\int_{-L/2}^{L/2} A_{11}^{(2D)}(y,z)dx \approx 0.10m \tag{5.2.3}$$

$$Y_{\dot{v}} = -\int_{-L/2}^{L/2} A_{22}^{(2D)}(y,z)dx \tag{5.2.4}$$

$$Z_{\dot{w}} = -\int_{-L/2}^{L/2} A_{33}^{(2D)}(y,z)dx \tag{5.2.5}$$

$$\begin{aligned} K_{\dot{p}} &= -\int_{-L/2}^{L/2} A_{44}^{(2D)}(y,z)dx \\ &= -\left(\int_{-B/2}^{B/2} y^2 A_{33}^{(2D)}(x,z)dy + \int_{-H/2}^{H/2} z^2 A_{22}^{(2D)}(x,y)dz\right) \end{aligned} \tag{5.2.6}$$

$$\begin{aligned} M_{\dot{q}} &= -\int_{-L/2}^{L/2} A_{55}^{(2D)}(x,y)dx \\ &= -\left(\int_{-L/2}^{L/2} x^2 A_{33}^{(2D)}(y,z)dx + \int_{-H/2}^{H/2} z^2 A_{11}^{(2D)}(x,y)dz\right) \end{aligned} \tag{5.2.7}$$

$$\begin{aligned} N_{\dot{r}} &= -\int_{-L/2}^{L/2} A_{66}^{(2D)}(x,y)dx \\ &= -\left(\int_{-B/2}^{B/2} y^2 A_{11}^{(2D)}(x,z)dy + \int_{-L/2}^{L/2} x^2 A_{22}^{(2D)}(y,z)dx\right) \end{aligned} \tag{5.2.8}$$

where $A_{xx}^{(2D)}$ are the two dimensional added mass coefficients, which, according to Fossen[2], can be approximated using the values for the simple geometries shown in figure 5.6.

Added Mass of an Arbitrary Ellipsoid

An alternative approach to using strip theory is to approximate the vehicle as a geometry for which the added mass is known. In this project the vessel is approximated as an ellipsoid.

Formulae for the simplified case of a prolate[7] ellipsoid exists, and are given by Fossen[2], Lamb[19] as well as by Imlay[20]. These simplified equations are adequate in many cases and is for example used extensively in the work done by Humphreys and Watkinson [21].

[7] An ellipsoid where 2 of the semi-axis are equal, i.e an ellipsoid with a circular cross section.

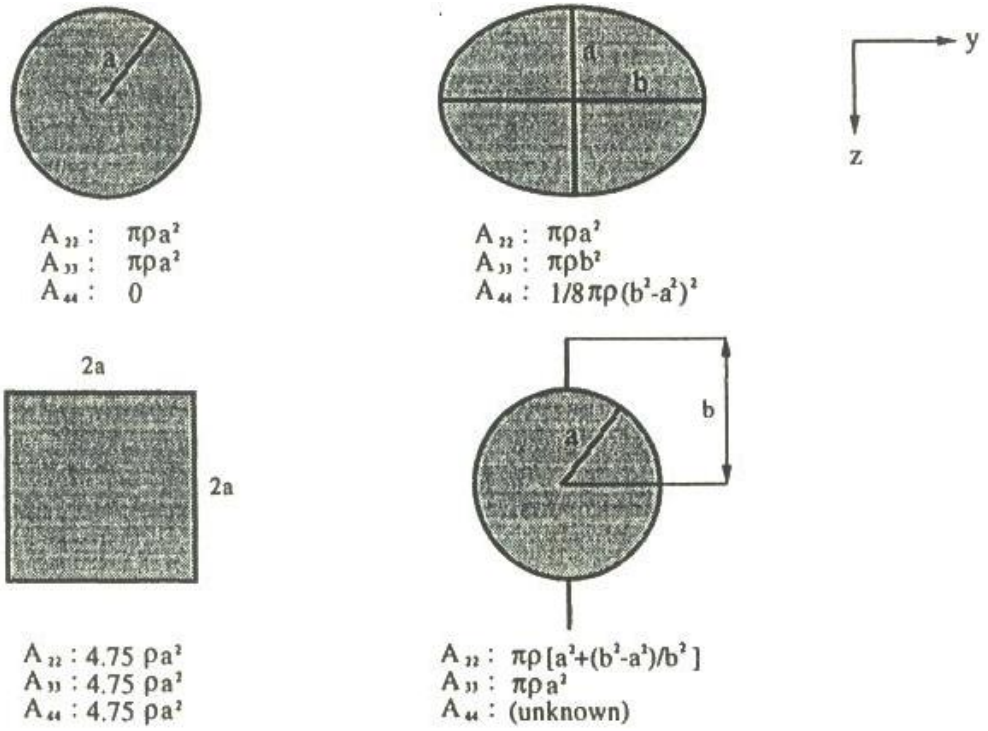

Figure 5.6: Two dimensional added mass coefficients for some basic geometries[2]

In the case of the AUV under consideration, the vessel cross-section is simply too oblique to use a prolate ellipsoidal approximation. In this case equations for the full unconstrained ellipsoid must be used. Now define an arbitrary ellipsoid as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{5.2.9}$$

where a, b and c are the semi-axes of the ellipsoid.

Lamb[19] published equations for the added mass of an arbitrary el-

lipsoid. These equations using the notation of Imlay[20] are given by

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.10}$$

$$Y_{\dot{v}} = -\frac{\beta_0}{2-\beta_0}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.11}$$

$$Z_{\dot{w}} = -\frac{\gamma_0}{2-\gamma_0}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.12}$$

$$K_{\dot{p}} = -\frac{1}{5}\,\frac{(b^2-c^2)^2(\gamma_0-\beta_0)}{2(b^2-c^2)+(b^2+c^2)(\beta_0-\gamma_0)}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.13}$$

$$M_{\dot{q}} = -\frac{1}{5}\,\frac{(c^2-a^2)^2(\alpha_0-\gamma_0)}{2(c^2-a^2)+(c^2+a^2)(\gamma_0-\alpha_0)}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.14}$$

$$N_{\dot{r}} = -\frac{1}{5}\,\frac{(a^2-b^2)^2(\beta_0-\alpha_0)}{2(a^2-b^2)+(a^2+b^2)(\alpha_0-\beta_0)}\,\frac{4}{3}\,\pi\rho abc \tag{5.2.15}$$

where α_0,β_0 and γ_0 are constants which describe the geometry of the ellipsoid. According to Lamb[19] these constants are given by

$$\alpha_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)^{\frac{3}{2}}(b^2+\lambda)^{\frac{1}{2}}(c^2+\lambda)^{\frac{1}{2}}} \tag{5.2.16}$$

$$\beta_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)^{\frac{1}{2}}(b^2+\lambda)^{\frac{3}{2}}(c^2+\lambda)^{\frac{1}{2}}} \tag{5.2.17}$$

$$\gamma_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)^{\frac{1}{2}}(b^2+\lambda)^{\frac{1}{2}}(c^2+\lambda)^{\frac{3}{2}}} \tag{5.2.18}$$

The primary effort to finding the added mass of an ellipsoid is now in solving the integrals given by equation 5.2.16 to 5.2.18. The solving of these integrals result in elliptic integrals of the first and second kind, which must be solved numerically.

Now taking a, b and c as 1.49 m, 0.339 m and 0.12 m respectively, then solving 5.2.16, 5.2.17 and 5.2.18 numerically gives,

$$\alpha_0 = 0.0599 \qquad \beta_0 = 0.4969 \qquad \gamma_0 = 1.4432 \tag{5.2.19}$$

Substituting 5.2.19 back into equations 5.2.10 to 5.2.15 gives the added

mass for the ellipsoid.

$$\begin{aligned} X_{\dot{u}} &= -7.8392 & Y_{\dot{v}} &= -83.933 \\ Z_{\dot{w}} &= -658.09 & K_{\dot{p}} &= -6.1723 \\ M_{\dot{q}} &= -258.81 & N_{\dot{r}} &= -30.829 \end{aligned} \tag{5.2.20}$$

Considering that the vehicle has a flooded mass of approximately 850 kg the only two really significant added mass terms are $Z_{\dot{w}}$ and $M_{\dot{q}}$.

5.3 Gravity and Buoyancy Model Parameters

In chapter 4 models for the forces and moments due to gravity and buoyancy were developed. In order to use these models a number of parameters are required. Calculating these parameters is the topic of this section.

Since it was assumed that the centre of gravity is located at the origin, the gravity model only requires a single parameter: the weight of the vessel. The weight, W in equation 4.3.6, can be calculated from equation 4.3.1 using 5.1.2 as,

$$W = 8276.01 \text{ N} \tag{5.3.1}$$

In order to use the buoyancy model the centre of buoyancy must be known. In a homogeneous fluid, the centre of buoyancy is located at the centroid of the displaced volume. The centroid of a volume can be calculated from,

$$c_x = \frac{\iiint x \, \mathrm{d}V}{V} \qquad c_y = \frac{\iiint y \, \mathrm{d}V}{V} \qquad c_z = \frac{\iiint z \, \mathrm{d}V}{V} \tag{5.3.2}$$

Using this is however impractical. Further, it is noted that in order to provide static stability, marine vehicles are designed so that their centre of buoyancy is directly above the centre of gravity. Therefore, only the vertical location must be estimated.

This can be done by rolling or pitching the vessel to a known angle

and then measuring the restoring moment. This restoring moment can then be related to the vertical location of the centre of buoyancy through equation 4.3.13.

Unfortunately the equipment required for this test was not available, necessitating the use of a less accurate estimate. For this project, the vertical location of the centre of buoyancy was estimated by varying the location and trying to find the best possible match to the dynamics of the IMT vessel.

Noting the above, the centre of buoyancy is estimated as,

$$x_B = 0 \qquad y_B = 0 \qquad z_B = -0.1 \text{ m} \tag{5.3.3}$$

Finally, the force due to buoyancy, B, can be calculated from 4.3.8 as.

$$B = 8374.13 \text{ N} \tag{5.3.4}$$

5.4 Thruster Model Parameters

A model to describe the AUV thrusters was derived in chapter 4 using momentum theory. In order to use this model, a number of parameters are required. They are: the area of the theoretical propeller disk; the input power to thrust constant under static conditions; the throttle to power constant; the density of the operating medium; the position of each thruster; and the thrust setting angle.

In this project it was assumed that the theoretical disk area is identical to the area covered by the turning propeller. More advanced versions of momentum theory compensate for blade geometry by using a so called effective radius. Due to the lack of adequate data this technique is not used here. Now, given that the propeller radius is 88.5 mm[3], the area of the of theoretical disk is

$$A = 0.0246 \ m^2 \tag{5.4.1}$$

As stated in chapter 4, the input power and produced thrust are assumed to be linearly related under static conditions. This linear relationship is described by the power to thrust constant, and is given by[8]

$$c_T = \frac{T_{max}}{P_{max}} = 0.4 \tag{5.4.2}$$

In addition to this it is also assumed that the throttle position and input power are linearly related through the throttle to power constant. This constant is given by,

$$c_P = \frac{P_{max}}{\delta_{T_{max}}} = 320 \tag{5.4.3}$$

where, $\delta_{T_{max}}$ is assumed to be unity.

As before, the density of the seawater is assumed to be constant and is given as

$$\rho = 1026\ kg.m^{-3} \tag{5.4.4}$$

The positon of each of the two thrusters are required to use the model. Both thrusters are positioned at the rear of the vessel, one on the starboard side and one on the port side. It is assumed that the thrusters are placed symmetrically, with respect to the X_BZ_B plane, and on the X_BY_B plane. The two thrusters are positioned at[9],

$$x_{T_l} = 0.94\ m \qquad x_{T_r} = 0.94\ m \tag{5.4.5}$$

$$y_{T_l} = -0.61\ m \qquad y_{T_r} = 0.61\ m \tag{5.4.6}$$

$$z_{T_l} = 0\ m \qquad z_{T_r} = 0\ m \tag{5.4.7}$$

where l and r denotes left and right respectively.

Finally, the thrust setting angle is needed. Due to the lack of informa-

[8] See table A.2.

[9] Although not required by the thruster model x_{T_l} and x_{T_r} are also given

tion to the contrary it is assumed to be zero.

$$\epsilon_T = 0 \tag{5.4.8}$$

Chapter 6

Linearisation and Analysis

In this chapter the nonlinear AUV model developed in the previous chapter is linearised.

A linear model is particularly useful from a control design standpoint, as standard linear control theory can be used. In addition to significantly simplifying control system design, a linear model can also provide much insight into the system dynamics.

6.1 Small Disturbance Theory

Small disturbance theory is a linearisation method in which it is assumed that nonlinear functions can be rewritten in terms of a trim or steady state value plus a linear deviation from this steady state value[22]. Small disturbance theory has been widely used to derive linear models for aircraft as well as underwater vehicles.

In order to use small disturbance theory the following assumptions are made[13][22]:

- The product of two small perturbations is assumed to be zero.
- The cosine of a small angle is assumed to be unity.
- The sine of a small angle is assumed to be equal to the size of the small angle in radians.

Using this linearisation method is relatively simple. The following summarises the main steps involved in linearising a model using small disturbances theory.

- Recast each variable into its trim plus perturbed variable, for example $A = A_0 + a$.
- Multiply out and apply the small disturbance assumptions mentioned above.
- Finally, remove the steady state equation from the combined equation. What remains is the linearised equation.

By limiting the type of trim conditions allowed, the linearisation process can be significantly simplified. In this project a number of assumptions are made about the initial trim flight condition. These are,

- Initial symmetric flight is assumed, thus all lateral trim values are zero.
- In addition it is assumed that the only hydrodynamic actuator without a zero trim value is the elevator, thus the trim pitching moment must be cancelled by the trim elevator setting.

Based on the assumptions mentioned above, the system variables can now be written in terms of trim values and perturbation variables, as [1]

$$\boldsymbol{V_B} = (U_0 + u)\boldsymbol{i_b} + v\boldsymbol{j_b} + (W_0 + w)\mathbf{k_b} \tag{6.1.1}$$

$$\boldsymbol{\omega_B} = p\mathbf{i_b} + q\boldsymbol{j_b} + r\boldsymbol{k_b} \tag{6.1.2}$$

$$\boldsymbol{e} = (\phi, \Theta_0 + \theta, \psi) \tag{6.1.3}$$

$$\boldsymbol{c} = (\delta_{dl}, \delta_{dr}, \delta_{E_0} + \delta_e, \delta_r, \delta_{Tl_0} + \delta_{tl}, \delta_{Tr_0} + \delta_{tr}) \tag{6.1.4}$$

Further it is assumed that the angle of attack and sideslip remains small throughout the manoeuvre considered, and as such the angle of

[1] Note that as before the reference line body axis system is used.

attack can be approximated as,

$$\alpha \approx \frac{W}{U_0} \tag{6.1.5}$$

the angle of sideslip as,

$$\beta \approx \frac{v}{U_0} \tag{6.1.6}$$

and the total velocity can be approximated as,

$$\bar{V} \approx U_0 \tag{6.1.7}$$

6.2 Linear Equations of Motion

The small disturbance theory is now used to linearise the AUV equations of motion. To recap, the nonlinear AUV equations of motions are given by

$$X = (m - X_{\dot{u}})\dot{U} + m(QW - RV) \tag{6.2.1}$$
$$Y = (m - Y_{\dot{v}})\dot{V} + m(RU - PW) \tag{6.2.2}$$
$$Z = (m - Z_{\dot{w}})\dot{W} + m(PV - QU) \tag{6.2.3}$$
$$K = (I_x - K_{\dot{p}})\dot{P} + (I_z - I_y)QR \tag{6.2.4}$$
$$M = (I_y - M_{\dot{q}})\dot{Q} + (I_x - I_z)RP \tag{6.2.5}$$
$$N = (I_z - N_{\dot{r}})\dot{R} + (I_y - I_x)PQ \tag{6.2.6}$$

In order to significantly simplify the EOMs all lateral-longitudinal cross-coupling terms are now eliminated, resulting in the following simplified

nonlinear equations.

$$X = (m - X_{\dot{u}})\dot{U} + mQW \tag{6.2.7}$$
$$Y = (m - Y_{\dot{v}})\dot{V} \tag{6.2.8}$$
$$Z = (m - Z_{\dot{w}})\dot{W} - mQU \tag{6.2.9}$$
$$K = (I_x - K_{\dot{p}})\dot{P} \tag{6.2.10}$$
$$M = (I_y - M_{\dot{q}})\dot{Q} \tag{6.2.11}$$
$$N = (I_z - N_{\dot{r}})\dot{R} \tag{6.2.12}$$

Equation 6.2.7 to 6.2.12 are now linearised using small disturbance theory, resulting in the following linearised EOMs.

$$X = (m - X_{\dot{u}})\dot{u} + mW_0 q \tag{6.2.13}$$
$$Y = (m - Y_{\dot{v}})\dot{v} \tag{6.2.14}$$
$$Z = (m - Z_{\dot{w}})\dot{w} - mU_0 q \tag{6.2.15}$$
$$K = (I_x - K_{\dot{p}})\dot{p} \tag{6.2.16}$$
$$M = (I_y - M_{\dot{q}})\dot{q} \tag{6.2.17}$$
$$N = (I_z - N_{\dot{r}})\dot{r} \tag{6.2.18}$$

The kinematic equations are also linearised using small disturbance theory. As before, all longitudinal-lateral cross-coupling terms are eliminated resaluting in the following linearised translational kinematic equations,

$$\dot{x} = \dot{p_N} = \cos\Theta_0 u - \sin\Theta_0 U_0\theta + \sin\Theta_0 w - \cos\Theta_0 W_0\theta \tag{6.2.19}$$
$$\dot{y} = \dot{p_E} = \cos\Theta_0 U_0\psi + v - W_0\phi + \sin\Theta_0 W_0\psi \tag{6.2.20}$$
$$\dot{d} = \dot{p_D} = -\sin\Theta_0 u - \cos\Theta_0 U_0\theta + \cos\Theta_0 w - \sin\Theta_0 W_0\theta \tag{6.2.21}$$

and the following linearised rotational kinematic equations.

$$\dot{\phi} = p + \tan\Theta_0 r \tag{6.2.22}$$

$$\dot{\theta} = q \tag{6.2.23}$$

$$\dot{\psi} = \frac{r}{\cos\Theta_0} \tag{6.2.24}$$

where x gives the longitudinal position error, h the depth perturbation and y the lateral cross track error.

6.3 Linear Hydrodynamic Model

Linear equations to describe the hydrodynamic forces and moments must now be developed. Assuming that the angle of attack remains small, the nonlinear hydrodynamic force and moment equations are given by,

$$X_H = -\bar{q}l^2 C_D + \bar{q}l^2 C_L \alpha \tag{6.3.1}$$

$$Y_H = \bar{q}l^2 C_Y \tag{6.3.2}$$

$$Z_H = -\bar{q}l^2 C_L - \bar{q}l^2 C_D \alpha \tag{6.3.3}$$

$$K_H = \bar{q}l^3 C_K - \bar{q}l^3 C_N \alpha \tag{6.3.4}$$

$$M_H = \bar{q}l^3 C_M \tag{6.3.5}$$

$$N_H = \bar{q}l^3 C_N + \bar{q}l^3 C_K \alpha \tag{6.3.6}$$

where,

$$C_D = C_{D_{basic}} + C_{D_{\delta_{dl}}}\delta_{dl} + C_{D_{\delta_{dr}}}\delta_{dr} + C_{D_{\delta_e}}\delta_e \tag{6.3.7}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_r}\hat{r} + \frac{V_{T_0}^2}{\bar{V}_0^2}C_{Y_{\delta_r}}\delta_r \tag{6.3.8}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\hat{q} + C_{L_{\delta_{dl}}}\delta_{dl} + C_{L_{\delta_{dr}}}\delta_{dr} + C_{L_{\delta_e}}\delta_e \tag{6.3.9}$$

$$C_K = C_{K_p}\hat{p} + C_{K_r}\hat{r} + C_{K_{\delta_{dl}}}\delta_{dl} + C_{K_{\delta_{dr}}}\delta_{dr} \tag{6.3.10}$$

$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\hat{q} + C_{M_{\delta_{dl}}}\delta_{dl} + C_{M_{\delta_{dr}}}\delta_{dr} + C_{M_{\delta_e}}\delta_e \tag{6.3.11}$$

$$C_N = C_{N_\beta}\beta + C_{N_p}\hat{p} + C_{N_r}\hat{r} + C_{N_{\delta_{dl}}}\delta_{dl} + C_{N_{\delta_{dr}}}\delta_{dr} + \frac{V_{T_0}^2}{\bar{V}_0^2}C_{N_{\delta_r}}\delta_r \tag{6.3.12}$$

Noting that unlike in the nonlinear case the rudder scaling factor, $\frac{V_{T_0}^2}{\bar{V}_0^2}$, is considered constant.

The hydrodynamic force and moment perturbation equations can be approximated as a first order Tayler series expansion. Now, ignoring all longitudinal-lateral cross-coupling terms, the perturbation equations are given by,

$$X_H = X_{H_u}u + X_{H_w}w + X_{H_q}q + X_{H_{\delta_{dl}}}\delta_{dl} + X_{H_{\delta_{dr}}}\delta_{dr} + X_{H_{\delta_e}}\delta_e \tag{6.3.13}$$

$$Y_H = Y_{H_v}v + Y_{H_p}p + Y_{H_r}r + Y_{H_{\delta_{dl}}}\delta_{dl} + Y_{H_{\delta_{dr}}}\delta_{dr} + Y_{H_{\delta_r}}\delta_r \tag{6.3.14}$$

$$Z = Z_{H_u}u + Z_{H_w}w + Z_{H_q}q + Z_{H_{\delta_{dl}}}\delta_{dl} + Z_{H_{\delta_{dr}}}\delta_{dr} + Z_{H_{\delta_e}}\delta_e \tag{6.3.15}$$

$$K = K_{H_v}v + K_{H_p}p + K_{H_r}r + K_{H_{\delta_{dl}}}\delta_{dl} + K_{H_{\delta_{dr}}}\delta_{dr} + K_{H_{\delta_r}}\delta_r \tag{6.3.16}$$

$$M = M_{H_u}u + M_{H_w}w + M_{H_q}q + M_{H_{\delta_{dl}}}\delta_{dl} + M_{H_{\delta_{dr}}}\delta_{dr} + M_{H_{\delta_e}}\delta_e \tag{6.3.17}$$

$$N = N_{H_v}v + N_{H_p}p + N_{H_r}r + N_{H_{\delta_{dl}}}\delta_{dl} + N_{H_{\delta_{dr}}}\delta_{dr} + N_{H_{\delta_r}}\delta_r \tag{6.3.18}$$

where, for example, X_{H_u} is given by $\frac{\partial X_H}{\partial U}$ at the trim condition. Each of the partial derivatives in equation 6.3.13 to 6.3.18 must now calculated.

It should be noted that the expressions for these derivatives are fairly complex, and dependent on the trim angle of attack, since they are coordinated into the reference line body axis system.

6.3.1 Longitudinal Stability and Control Derivatives

The longitudinal stability and control derivatives describe the hydrodynamic forces and moments experienced by the vessel due to a perturbation in the longitudinal motion and control variables. As before, it is assumed that lateral motion and control perturbations have no effect on longitudinal dynamics.

Axial Velocity Perturbation Derivatives

An axial velocity perturbation primarily affects lift and drag. Although the effect of an axial velocity perturbation on the pitching moment is usually negligeable for subsonic flight, an expression for this is given for the sake of completeness. Coordinating this into the reference line body axis system gives the following.

$$X_{H_u} = -\frac{1}{2}\rho l^2(2U_0C_{D_{basic}} + 2U_0C_{D_{\delta_e}}\delta_{e_0} - W_0C_{L_0} - W_0C_{L_{\delta_e}}\delta_{e_0}) \quad (6.3.19)$$

$$Z_{H_u} = -\frac{1}{2}\rho l^2(2C_{L_0}U_0 + C_{L_\alpha}W_0 + 2C_{L_{\delta_e}}\delta_{e_0}U_0 + W_0C_{D_{basic}} + W_0C_{D_{\delta_e}}\delta_{e_0}) \quad (6.3.20)$$

$$M_{H_u} = \frac{1}{2}\rho l^3(2C_{M_0}U_0 + C_{M_\alpha}W_0 + 2C_{M_{\delta_e}}\delta_{e_0}U_0) \quad (6.3.21)$$

Normal Velocity Perturbation Derivatives

A small normal velocity perturbation is in essence a small perturbation in angle of attack, see equation 6.1.5. Angle of attack effects lift, drag as well as the pitching moment coefficients. Coordinated into the reference line body axis, this is given by

$$X_{H_w} = \frac{1}{2}\rho l^2(2C_{L_\alpha}W_0 + C_{L_0}U_0 + C_{L_{\delta_e}}\delta_{e_0}U_0) \quad (6.3.22)$$

$$Z_{H_w} = -\frac{1}{2}\rho U_0 l^2(C_{L_\alpha} + C_{D_{basic}} + C_{D_{\delta_e}}\delta_{e_0}) \quad (6.3.23)$$

$$M_{H_w} = \frac{1}{2}\rho U_0 l^3 C_{M_\alpha} \quad (6.3.24)$$

Pitch Rate Perturbation Derivatives

A pitch rate perturbation causes a change in the vessel lift coefficient and as such generates a normal as well as axial force perturbation. A pitch rate perturbation also generates a pitching moment perturbation, providing the vehicle with natural pitch rate dampening.

$$X_{H_q} = \frac{1}{4}\rho l^3 C_{L_q} W_0 \tag{6.3.25}$$

$$Z_{H_q} = -\frac{1}{4}\rho U_0 l^3 C_{L_q} \tag{6.3.26}$$

$$M_{H_q} = \frac{1}{4}\rho U_0 l^4 C_{M_q} \tag{6.3.27}$$

Elevator Deflection Perturbation Derivatives

The elevator is used as the primary pitch control device and as such it can be excepted that an elevator deflection perturbation generates a significant pitching moment perturbation. In addition to this, an elevator deflection perturbation has an effect on both the drag as well as the lift coefficient, resulting in axial and normal force perturbations.

$$X_{H_{\delta_e}} = -\frac{1}{2}\rho U_0 l^2 (U_0 C_{D_{\delta_e}} - C_{L_{\delta_e}} W_0) \tag{6.3.28}$$

$$Z_{H_{\delta_e}} = -\frac{1}{2}\rho U_0 l^2 (U_0 C_{L_{\delta_e}} + C_{D_{\delta_e}} W_0) \tag{6.3.29}$$

$$M_{H_{\delta_e}} = \frac{1}{2}\rho U_0^2 l^3 C_{M_{\delta_e}} \tag{6.3.30}$$

Diving Planes Deflection Perturbation Derivatives

The diving planes, when operated symmetrically, provide additional longitudinal control capability. As is the case for the elevator, a symmetric diving plane deflection perturbation results in an axial force, normal

force as well as pitching moment perturbations.

$$X_{H_{\delta_{dl}}} = \frac{1}{2}\rho U_0 l^2(-U_0 C_{D_{\delta_{dl}}} + C_{L_{\delta_{dl}}} W_0) \tag{6.3.31}$$

$$X_{H_{\delta_{dr}}} = \frac{1}{2}\rho U_0 l^2(-U_0 C_{D_{\delta_{dr}}} + C_{L_{\delta_{dr}}} W_0) \tag{6.3.32}$$

$$Z_{H_{\delta_{dl}}} = -\frac{1}{2}\rho U_0 l^2(U_0 C_{L_{\delta_{dl}}} + C_{D_{\delta_{dl}}} W_0) \tag{6.3.33}$$

$$Z_{H_{\delta_{dr}}} = -\frac{1}{2}\rho U_0 l^2(U_0 C_{L_{\delta_{dr}}} + C_{D_{\delta_{dr}}} W_0) \tag{6.3.34}$$

$$M_{H_{\delta_{dl}}} = \frac{1}{2}\rho U_0^2 l^3 C_{M_{\delta_{dl}}} \tag{6.3.35}$$

$$M_{H_{\delta_{dr}}} = \frac{1}{2}\rho U_0^2 l^3 C_{M_{\delta_{dr}}} \tag{6.3.36}$$

6.3.2 Lateral Stability and Control Derivatives

The lateral stability and control derivatives describe the hydrodynamic forces and moments experienced by the vessel due to a perturbation in the lateral motion and control variables. As before, it is assumed that longitudinal motion and control perturbations have no effect on lateral dynamics.

Lateral Velocity Perturbation Derivatives

A small perturbation in lateral velocity is in essence a small perturbation in the angle of sideslip, see equation 6.1.6. This perturbation primarily induces a perturbation in the yawing moment coefficient and, to a lesser extent, in the lateral force coefficient. Coordinated into the reference line body axis, this is given by

$$Y_{H_v} = \frac{1}{2}\rho U_0 l^2 C_{Y_\beta} \tag{6.3.37}$$

$$K_{H_v} = -\frac{1}{2}\rho l^3 C_{N_\beta} W_0 \tag{6.3.38}$$

$$N_{H_v} = \frac{1}{2}\rho U_0 l^3 C_{N_\beta} \tag{6.3.39}$$

Roll Rate Perturbation Derivatives

A roll rate perturbation induces a perturbation in the roll moment coefficient, and this provides the vessel with natural roll rate damping. In addition to this a small perturbation in the yaw moment coefficient may also be generated. Coordinating this into the reference line body axis system gives the following.

$$Y_{H_p} = 0 \tag{6.3.40}$$

$$K_{H_p} = \frac{1}{4}\rho l^4 (U_0 C_{K_p} - C_{N_p} W_0) \tag{6.3.41}$$

$$N_{H_p} = \frac{1}{4}\rho l^4 (U_0 C_{N_p} + C_{K_p} W_0) \tag{6.3.42}$$

Yaw Rate Perturbation Derivatives

A yaw rate perturbation induces a perturbation in the yaw moment coefficient, and this provides the vessel with natural yaw rate damping. In addition to this a small perturbation in the roll moment, as well as lateral force coefficients, are also generated. Coordinating this into the reference line body axis system gives the following.

$$Y_{H_r} = \frac{1}{4}\rho U_0 l^3 C_{Y_r} \tag{6.3.43}$$

$$K_{H_r} = \frac{1}{4}\rho l^4 (U_0 C_{K_r} - C_{N_r} W_0) \tag{6.3.44}$$

$$N_{H_r} = \frac{1}{4}\rho l^4 (U_0 C_{N_r} + C_{K_r} W_0) \tag{6.3.45}$$

Rudder Deflection Perturbation Derivatives

The rudder is the primary yaw control device, and as such it is expected that a rudder deflection perturbation will generate a significant yawing moment coefficient perturbation. In addition to this a rudder deflection perturbation also generates a lateral force coefficient perturbation. Coor-

dinating this into the reference line body axis system gives the following.

$$Y_{H_{\delta_r}} = \frac{1}{2}\rho l^2 V_{T_0}^2 C_{Y_{\delta_r}} \tag{6.3.46}$$

$$K_{H_{\delta_r}} = -\frac{1}{2}\frac{\rho}{U_0} l^3 V_{T_0}^2 C_{N_{\delta_r}} W_0 \tag{6.3.47}$$

$$N_{H_{\delta_r}} = \frac{1}{2}\rho l^3 V_{T_0}^2 C_{N_{\delta_r}} \tag{6.3.48}$$

Diving Planes Deflection Perturbation Derivatives

The diving planes, when operated asymmetrically, are used as the primary roll control device, and as such it is expected that a diving plane deflection perturbation will generate a significant rolling moment coefficient perturbation. In addition to this a diving plane deflection also generates a lateral force as well as a yawing moment coefficient perturbation. Coordinating this into the reference line body axis system gives the following.

$$Y_{H_{\delta_{dl}}} = Y_{\delta_{dr}} = 0 \tag{6.3.49}$$

$$K_{H_{\delta_{dl}}} = -\frac{1}{2}\rho U_0 l^3 (-U_0 C_{K_{\delta_{dl}}} + C_{N_{\delta_{dl}}} W_0) \tag{6.3.50}$$

$$K_{H_{\delta_{dr}}} = -\frac{1}{2}\rho U_0 l^3 (-U_0 C_{K_{\delta_{dr}}} + C_{N_{\delta_{dr}}} W_0) \tag{6.3.51}$$

$$N_{H_{\delta_{dl}}} = \frac{1}{2}\rho U_0 l^3 (U_0 C_{N_{\delta_{dl}}} + C_{K_{\delta_{dl}}} W_0) \tag{6.3.52}$$

$$N_{H_{\delta_{dr}}} = \frac{1}{2}\rho U_0 l^3 (U_0 C_{N_{\delta_{dr}}} + C_{K_{\delta_{dr}}} W_0) \tag{6.3.53}$$

6.4 Linear Gravitation Model

Deriving a linear gravitation model is a fairly straightforward application of small disturbance theory. To recap, the nonlinear gravity model is

given by,

$$F_G^B = \begin{bmatrix} -W\sin\Theta \\ W\cos\Theta\sin\Phi \\ W\cos\Theta\cos\Phi \end{bmatrix} \tag{6.4.1}$$

$$M_G^B = 0 \tag{6.4.2}$$

Now, applying small disturbance theory to this model yields the following linear small perturbation gravity model.

$$X_G = -W\cos\Theta_0\theta \tag{6.4.3}$$

$$Y_G = W\cos\Theta_0\phi \tag{6.4.4}$$

$$Z_G = -W\sin\Theta_0\theta \tag{6.4.5}$$

$$K_G = M_G = N_G = 0 \tag{6.4.6}$$

6.5 Linear Buoyancy Model

A linear buoyancy model is now derived. This is done in a similar manner as the linear gravity model. Assuming that the centre of buoyancy is located directly above the centre of gravity, the nonlinear buoyancy model is given by the following equations.

$$F_B^B = \begin{bmatrix} B\sin\Theta \\ -B\cos\Theta\sin\Phi \\ -B\cos\Theta\cos\Phi \end{bmatrix} \tag{6.5.1}$$

$$M_B^B = \begin{bmatrix} z_B B\cos\Theta\sin\Phi \\ z_B B\sin\Theta \\ 0 \end{bmatrix} \tag{6.5.2}$$

Now, applying small disturbance theory to this model yields the following linear small perturbation buoyancy model.

$$X_B = B\cos\Theta_0\theta \tag{6.5.3}$$
$$Y_B = -B\cos\Theta_0\phi \tag{6.5.4}$$
$$Z_B = B\sin\Theta_0\theta \tag{6.5.5}$$
$$K_B = z_B B\cos\Theta_0\phi \tag{6.5.6}$$
$$M_B = z_B B\cos\Theta_0\theta \tag{6.5.7}$$
$$N_B = 0 \tag{6.5.8}$$

6.6 Linear Thruster Model

Linear equations to describe the thruster forces and moments must now be developed. To recap, the nonlinear thruster forces and moments are given by,

$$F_T^B = \begin{bmatrix} (T_l + T_r)\cos\epsilon_T \\ 0 \\ -(T_l + T_r)\sin\epsilon_T \end{bmatrix} \tag{6.6.1}$$

$$M_T^B = \begin{bmatrix} -(y_{T_1}T_1 + y_{T_2}T_2)\sin\epsilon_T \\ (z_{T_1}T_1 + z_{T_2}T_2)\cos\epsilon_T + (x_{T_1}T_1 + x_{T_2}T_2)\sin\epsilon_T \\ -(y_{T_1}T_1 + y_{T_2}T_2)\cos\epsilon_T \end{bmatrix} \tag{6.6.2}$$

where T_l and T_r is the thrust generated by the left and right thruster respectively. Also note that the thrust setting angle ϵ_T is assumed constant throughout the manoeuvre under consideration. The thrust for an arbitrary thruster is implicitly given by the following nonlinear equation,

$$T^2 + \frac{Vc_T^2c_P^2\delta t^2}{\sqrt{\frac{c_Tc_P\delta t}{2\rho A}}} - \frac{c_T^3c_P^3\delta t^3}{T} = 0 \tag{6.6.3}$$

where c_t, c_p and A are defined in section 4.4.

In order to linearise the thruster model, the thrust perturbation can

be approximated by a first order Taylor series expansion. From equation 6.6.3 it is evident that the thrust perturbation is a function of the velocity as well as the throttle perturbation, and is given by

$$T = T_u u + T_{\delta_t} \delta_t \tag{6.6.4}$$

Equation 6.6.4 is now substituted back into the thruster force and moment equations, giving

$$X_T = \left(Tl_u u + Tl_{\delta_{tl}} \delta_{tl} + Tr_u u + Tr_{\delta_{tr}} \delta_{tr}\right) \cos \epsilon_T \tag{6.6.5}$$

$$Y_T = 0 \tag{6.6.6}$$

$$Z_T = -\left(Tl_u u + Tl_{\delta_{tl}} \delta_{tl} + Tr_u u + Tr_{\delta_{tr}} \delta_{tr}\right) \sin \epsilon_T \tag{6.6.7}$$

$$K_T = -\left(y_{T_l}(Tl_u u + Tl_{\delta_{tl}} \delta_{tl}) + y_{T_r}(Tr_u u + Tr_{\delta_{tr}} \delta_{tr})\right) \sin \epsilon_T \tag{6.6.8}$$

$$\begin{aligned} M_T = &\left(z_{T_l}(Tl_u u + Tl_{\delta_{tl}} \delta_{tl}) + z_{T_r}(Tr_u u + Tr_{\delta_{tr}} \delta_{tr})\right) \cos \epsilon_T \\ &+ \left(x_{T_l}(Tl_u u + Tl_{\delta_{tl}} \delta_{tl}) + x_{T_r}(Tr_u u + Tr_{\delta_{tr}} \delta_{tr})\right) \sin \epsilon_T \end{aligned} \tag{6.6.9}$$

$$N_T = -\left(y_{T_1}(Tl_u u + Tl_{\delta_{tl}} \delta_{tl}) + y_{T_r}(Tr_u u + Tr_{\delta_{tr}} \delta_{tr})\right) \cos \epsilon_T \tag{6.6.10}$$

The thruster forces and moments are now described in terms of first order Taylor series expansions. Ignoring lateral-longitudinal cross coupling, the expansions are given by

$$X_T = X_{Tl_u} u + X_{Tl_{\delta_{tl}}} \delta_{tl} + X_{Tr_u} u + X_{Tr_{\delta_{tr}}} \delta_{tr} \tag{6.6.11}$$

$$Y_T = 0 \tag{6.6.12}$$

$$Z_T = Z_{Tl_u} u + Z_{Tl_{\delta_{tl}}} \delta_{tl} + Z_{Tr_u} u + Z_{Tr_{\delta_{tr}}} \delta_{tr} \tag{6.6.13}$$

$$K_T = K_{Tl_{\delta_{tl}}} \delta_{tl} + K_{Tr_{\delta_{tr}}} \delta_{tr} \tag{6.6.14}$$

$$M_T = M_{Tl_u} u + M_{Tl_{\delta_{tl}}} \delta_{tl} + M_{Tr_u} u + M_{Tr_{\delta_{tr}}} \delta_{tr} \tag{6.6.15}$$

$$N_T = N_{Tl_{\delta_{tl}}} \delta_{tl} + N_{Tr_{\delta_{tr}}} \delta_{tr} \tag{6.6.16}$$

where, for example, X_{Tl_u} is given by $Tl_u \cos \epsilon_T$

The partial derivatives T_u and T_{δ_t} must now be calculated by differentiating 6.6.3 with respect to u and δ_t respectively at the trim condition,

giving

$$T_u = \frac{\partial T}{\partial V} = -\frac{T_0^2\sqrt{2\rho A}\left(c_T c_P \delta t_0\right)^{\frac{3}{2}}}{2T_0^3 + \left(c_T c_P\right)^3} \tag{6.6.17}$$

$$T_{\delta t} = \frac{\partial T}{\partial \delta t_0} = \frac{3\left(c_T c_P\right)^3 \delta t^2 T_0 - \frac{3}{2}V_0 T_0^2\sqrt{2\rho A\delta t_0}\left(c_T c_P\right)^{\frac{3}{2}}}{2T_0^3 + \left(c_T c_P\right)^3} \tag{6.6.18}$$

Finally, under certain condition it might be necessary to model the throttle delay. This can easily be achieved using the following ordinary first order differential equation[2].

$$\dot{\delta t} = \frac{1}{\tau_T}\delta t_c - \frac{1}{\tau_T}\delta t \tag{6.6.19}$$

where, δt_c is the commanded throttle perturbation and τ_T is the thruster time constant.

6.7 Complete Linear Model

The linear forces and moments model developed in section 6.3 through 6.6 is now combined with the linear EOMs from section 6.2 and grouped into a longitudinal and lateral set.

6.7.1 Longitudinal Linear Model

The linear longitudinal small perturbation model describes the longitudinal vessel dynamics when disturbed from trimmed equilibrium by a perturbation in one or more of the longitudinal motion or control vari-

[2]If any other actuator delays prove to be significant they can be modelled in an analogous manner.

ables, and is given by,

$$\begin{aligned}
&X_{H_u}u + X_{H_w}w + X_{H_q}q + X_{H_{\delta_{dl}}}\delta_{dl} + X_{H_{\delta_{dr}}}\delta_{dr} + X_{H_{\delta_e}}\delta_e \\
&\quad - W\cos\Theta_0\theta + B\cos\Theta_0\theta + X_{T_{l_u}}u + X_{T_{l_{\delta_{tl}}}}\delta_{tl} \\
&\quad + X_{T_{r_u}}u + X_{Tr_{\delta_{tr}}}\delta_{tr} = (m - X_{\dot{u}})\dot{u} + mW_0q \qquad (6.7.1)\\
&Z_{H_u}u + Z_{H_w}w + Z_{H_q}q + Z_{H_{\delta_{dl}}}\delta_{dl} + Z_{H_{\delta_{dr}}}\delta_{dr} + Z_{H_{\delta_e}}\delta_e \\
&\quad - W\sin\Theta_0\theta + B\sin\Theta_0\theta + Z_{Tl_u}u + Z_{Tl_{\delta_{tl}}}\delta_{tl} \\
&\quad + Z_{Tr_u}u + Z_{Tr_{\delta_{tr}}}\delta_{tr} = (m - Z_{\dot{w}})\dot{w} - mU_0q \qquad (6.7.2)\\
&M_{H_u}u + M_{H_w}w + M_{H_q}q + M_{H_{\delta_{dl}}}\delta_{dl} + M_{H_{\delta_{dr}}}\delta_{dr} + M_{H_{\delta_e}}\delta_e \\
&\quad + z_B B\cos\Theta_0\theta + M_{Tl_u}u + M_{Tl_{\delta_{tl}}}\delta_{tl} \\
&\quad + M_{Tr_u}u + M_{Tr_{\delta_{tr}}}\delta_{tr} = (I_y - M_{\dot{q}})\dot{q} \qquad (6.7.3)\\
&\dot{\theta} = q \qquad (6.7.4)
\end{aligned}$$

Note that the inclusion of the pitch angle update equation is required for the set of longitudinal equations to be solvable. These equations are now rewritten in terms of an inertia matrix, a natural dynamics matrix and a control matrix. This form displays the primary system dynamics in a far more elegant and compact manner, and is given by,

$$\mathbf{M_l}\dot{\mathbf{x}}_\mathbf{l} = \mathbf{A_l}\mathbf{x_l} + \mathbf{B_l}\mathbf{u_l} \qquad (6.7.5)$$

where,

$$\mathbf{x_l} = \begin{bmatrix} u & w & q & \theta \end{bmatrix}^T \tag{6.7.6}$$

$$\mathbf{u_l} = \begin{bmatrix} \delta_e & \delta_{dl} & \delta_{dr} & \delta_{tl} & \delta_{tr} \end{bmatrix} \tag{6.7.7}$$

$$\mathbf{M_l} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Z_{\dot{w}} & 0 & 0 \\ 0 & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.7.8}$$

$$\mathbf{A_l} = \begin{bmatrix} X_{H_u} + X_{Tl_u} + X_{Tr_u} & X_{H_w} & X_{H_q} - mW_0 & (B - W)\cos\Theta_0 \\ Z_{H_u} + Z_{Tl_u} + Z_{Tr_u} & Z_{H_w} & Z_{H_q} + mU_0 & (B - W)\sin\Theta_0 \\ M_{H_u} + M_{Tl_u} + M_{Tr_u} & M_{H_w} & M_{H_q} & z_B B\cos\Theta_0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6.7.9}$$

$$\mathbf{B_l} = \begin{bmatrix} X_{H_{\delta_e}} & X_{H_{\delta_{dl}}} & X_{H_{\delta_{dr}}} & X_{Tl_{\delta_{tl}}} & X_{Tr_{\delta_{tr}}} \\ Z_{H_{\delta_e}} & X_{H_{\delta_{dl}}} & Z_{H_{\delta_{dr}}} & Z_{Tl_{\delta_{tl}}} & Z_{Tr_{\delta_{tr}}} \\ M_{H_{\delta_e}} & M_{H_{\delta_{dl}}} & M_{H_{\delta_{dr}}} & M_{Tl_{\delta_{tl}}} & M_{Tr_{\delta_{tr}}} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.7.10}$$

Finally, in order not to cloud the primary longitudinal dynamics, equations not directly required by the other equations were omitted. These are now restated here for the sake of convenience.

$$\dot{p_N} = \cos\Theta_0 u - \sin\Theta_0 U_0\theta + \sin\Theta_0 w - \cos\Theta_0 W_0\theta \tag{6.7.11}$$

$$\dot{d} = -\sin\Theta_0 u - \cos\Theta_0 U_0\theta + \cos\Theta_0 w - \sin\Theta_0 W_0\theta \tag{6.7.12}$$

6.7.2 Lateral Linear Model

The linear lateral small perturbation model describes the lateral vessel dynamics when disturbed from trimmed equilibrium by a perturbation

in one or more of the lateral motion or control variables, and is given by,

$$Y_{H_v}v + Y_{H_p}p + Y_{H_r}r + Y_{H_{\delta_r}}\delta_r + W\cos\Theta_0\phi - B\cos\Theta_0\phi = (m - Y_{\dot{v}})\dot{v} \tag{6.7.13}$$

$$K_{H_v}v + K_{H_p}p + K_{H_r}r + K_{H_{\delta_{dl}}}\delta_{dl} + K_{H_{\delta_{dr}}}\delta_{dr} + K_{H_{\delta_r}}\delta_r + z_B B\cos\Theta_0\phi + K_{Tl_{\delta_{tl}}}\delta_{tl} + K_{Tr_{\delta_{tr}}}\delta_{tr} = (I_x - K_{\dot{p}})\dot{p} \tag{6.7.14}$$

$$N_{H_v}v + N_{H_p}p + N_{H_r}r + N_{H_{\delta_{dl}}}\delta_{dl} + N_{H_{\delta_{dr}}}\delta_{dr} + N_{H_{\delta_r}}\delta_r + K_{Tl_{\delta_{tl}}}\delta_{tl} + K_{Tr_{\delta_{tr}}}\delta_{tr} = (I_z - N_{\dot{r}})\dot{r} \tag{6.7.15}$$

$$\dot{\phi} = p + \tan\Theta_0 r \tag{6.7.16}$$

Note that the inclusion of the roll angle update equation is required for the set of lateral equations to be solvable. As was done with the longitudinal model, the lateral equations are now rewritten in terms of an inertia matrix, a natural dynamics matrix and a control matrix. The lateral equations are now given by,

$$\mathbf{M_t}\dot{\mathbf{x}}_\mathbf{l} = \mathbf{A_t}\mathbf{x_t} + \mathbf{B_t}\mathbf{u_t} \tag{6.7.17}$$

where,

$$\mathbf{x_t} = \begin{bmatrix} v & p & r & \phi \end{bmatrix}^T \tag{6.7.18}$$

$$\mathbf{u_l} = \begin{bmatrix} \delta_r & \delta_{dl} & \delta_{dr} & \delta_{tl} & \delta_{tr} \end{bmatrix} \tag{6.7.19}$$

$$\mathbf{M_r} = \begin{bmatrix} m - Y_{\dot{v}} & 0 & 0 & 0 \\ 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & I_z - N_{\dot{r}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.7.20}$$

$$\mathbf{A_l} = \begin{bmatrix} Y_{H_v} & Y_{H_p} & Y_{H_r} & \cos\Theta_0(W-B) \\ K_{H_v} & K_{H_p} & K_{H_r} & z_B B\cos\Theta_0 \\ N_{H_v} & N_{H_p} & N_{H_r} & 0 \\ 0 & 1 & \tan\Theta_0 & 0 \end{bmatrix} \tag{6.7.21}$$

$$\mathbf{B_l} = \begin{bmatrix} Y_{H_{\delta_r}} & 0 & 0 & 0 & 0 \\ K_{H_{\delta_r}} & K_{H_{\delta_{dl}}} & K_{H_{\delta_{dr}}} & K_{Tl_{\delta_{tl}}} & K_{Tr_{\delta_{tr}}} \\ N_{H_{\delta_r}} & N_{H_{\delta_{dl}}} & N_{H_{\delta_{dr}}} & N_{Tl_{\delta_{tl}}} & N_{Tr_{\delta_{tr}}} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.7.22}$$

Finally, in order not to cloud the primary lateral dynamics, equations not directly required by the other equations were omitted. These are now restated here for the sake of convenience.

$$y = \cos\Theta_0 U_0 \psi + v - W_0\phi + \sin\Theta_0 W_0 \psi \tag{6.7.23}$$

$$\dot{\psi} = \frac{r}{\cos\Theta_0} \tag{6.7.24}$$

6.8 Specific Linear Model

Now that the development of the small perturbation linear model is complete, the flight and vehicle specific numeric values must be calculated.

6.8.1 Trim Calculations

Before the vehicle specific model can be determined, all required trim variables must be calculated. During trimmed equilibrium all forces and moments must cancel, thus

$$X_{H_0} + X_{G_0} + X_{B_0} + X_{T_0} = 0 \tag{6.8.1}$$
$$Y_{H_0} + Y_{G_0} + Y_{B_0} + Y_{T_0} = 0 \tag{6.8.2}$$
$$Z_{H_0} + Z_{G_0} + Z_{B_0} + Z_{T_0} = 0 \tag{6.8.3}$$
$$K_{H_0} + K_{G_0} + K_{B_0} + K_{T_0} = 0 \tag{6.8.4}$$
$$M_{H_0} + M_{G_0} + M_{B_0} + M_{T_0} = 0 \tag{6.8.5}$$
$$N_{H_0} + N_{G_0} + N_{B_0} + N_{T_0} = 0 \tag{6.8.6}$$

Since initial symmetric flight is assumed, the only non zero trim variables are: U_0, W_0, Θ_0, δ_{e_0}, δ_{tl_0} and δ_{tr_0}. This results in the following simplified trim equations.

$$X_{H_0} + X_{G_0} + X_{B_0} + X_{T_0} = 0 \tag{6.8.7}$$
$$Z_{H_0} + Z_{G_0} + Z_{B_0} = 0 \tag{6.8.8}$$
$$M_{H_0} + M_{B_0} = 0 \tag{6.8.9}$$

Now, combining these trim equations with the force and moment models from chapter 4 and using the data from chapter 5, the trim values are calculated as[3],

$$\begin{aligned} U_0 &= 1.2\ m.s^{-1} & W_0 &= -0.08399\ m.s^{-1} \\ \delta_{e_0} &= -0.10862 \text{ rad} & \delta_{tl_0} &= \delta_{tr_0} = 0.757 \\ V_{T_0} &= 1.9216\ m.s^{-1} & \Theta_0 &= -0.069992 \text{ rad} \end{aligned} \tag{6.8.10}$$

[3]In order to simplify calculations, small angle approximations are used.

6.8.2 Longitudinal Model

Now that all the required trim values have been calculated, the longitudinal flight specific model can be determined.

With the exception of the trim variables, all parameter data is obtained from chapter 5 and appendix A. Since the hydrodynamic parameters were given in the form of graphs, the exact values used while deriving the flight specific model are given below.

$$
\begin{aligned}
&C_{D_{basic}} = 0.0297 && C_{D_{\delta_e}} = 0 \\
&C_{D_{\delta_{dl}}} = 0 && C_{D_{\delta_{dr}}} = 0 \\
&C_{L_0} = 0 && C_{L_\alpha} = 0.168 \\
&C_{L_q} = 0.0185 && C_{L_{\delta_e}} = 0.0105 \\
&C_{L_{\delta_{dl}}} = 0.00387 && C_{L_{\delta_{dr}}} = 0.00387 \\
&C_{M_0} = 0 && C_{M_\alpha} = 0.0652 \\
&C_{M_q} = -0.10957 && C_{M_{\delta_e}} = -0.0146 \\
&C_{M_{\delta_{dl}}} = 0.000474 && C_{M_{\delta_{dr}}} = 0.000474
\end{aligned} \tag{6.8.11}
$$

It should be noted that unlike in the nonlinear model $C_{D_{\delta_e}}$, $C_{D_{\delta_{dl}}}$ and $C_{D_{\delta_{dr}}}$ were taken to be zero. This is necessary, since these values are all discontinuous at, or close to, the trim value. In addition to this, these contribute very little to the total vessel drag.

The flight specific longitudinal model is now calculated as,

$$\mathbf{M_l}' = \begin{bmatrix} 851.47 & 0 & 0 & 0 \\ 0 & 1501.7 & 0 & 0 \\ 0 & 0 & 756.05 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.8.12}$$

$$\mathbf{A_l}' = \begin{bmatrix} -356.68 & -134.82 & 60.069 & 97.88 \\ 88.258 & -1080.4 & 858.23 & -6.862 \\ -27.051 & 1075 & -2731.8 & -835.36 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6.8.13}$$

$$\mathbf{B_l}' = \begin{bmatrix} -4.8879 & -1.8022 & -1.8022 & 89.517 & 89.517 \\ -69.836 & -1.8022 & -25.748 & 0 & 0 \\ -266.19 & -9.4523 & -9.4523 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.8.14}$$

Finally, the longitudinal position as well as the depth errors are given by,

$$\dot{p_N} = 0.99616u + 0.20982\theta + -0.087536w \tag{6.8.15}$$

$$\dot{d} = 0.087536u - 1.2046\theta + 0.99616w \tag{6.8.16}$$

6.8.3 Lateral Model

The flight specific lateral model is determined in a similar manner as the flight specific longitudinal model. As before, the required data is supplied by chapter 5 and appendix A, with the exact hydrodynamic

parameters used during this section given below.

$$
\begin{aligned}
&C_{Y_\beta} = -0.069778 && C_{Y_r} = 0 \\
&C_{Y_{\delta_r}} = -0.0006366 && C_{K_p} = -5.9367 \times 10^{-5} \\
&C_{K_r} = 0 && C_{K_{\delta_{dl}}} = 0.002667 \\
&C_{K_{\delta_{dr}}} = -0.002667 && C_{N_\beta} = -0.057593 \\
&C_{N_p} = 0 && C_{N_r} = -0.00992 \\
&C_{N_{\delta_r}} = -0.000307 && C_{N_{\delta_{dl}}} = 0 \\
&C_{N_{\delta_{dr}}} = 0
\end{aligned} \tag{6.8.17}
$$

The flight specific lateral model is now calculated as,

$$
\mathbf{M_t}' = \begin{bmatrix} 927.56 & 0 & 0 & 0 \\ 0 & 71.222 & 0 & 0 \\ 0 & 0 & 577.83 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.8.18}
$$

$$
\mathbf{A_t}' = \begin{bmatrix} -384.13 & 0 & 0 & -97.88 \\ 18.009 & -0.8049 & -17.364 & -835.36 \\ 257.3 & 0.056336 & -248.09 & 0 \\ 0 & 1 & -0.070107 & 0 \end{bmatrix} \tag{6.8.19}
$$

$$
\mathbf{B_t}' = \begin{bmatrix} -10.854 & 0 & 0 & 0 & 0 \\ -2.0573 & -53.198 & 53.198 & 0 & 0 \\ -29.394 & 3.7235 & -3.7235 & -55.501 & 55.501 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.8.20}
$$

Finally, the lateral cross track error is given as

$$
y = 1.2046\psi + v + 0.10518\phi \tag{6.8.21}
$$

$$
\dot{\psi} = \frac{r}{0.99616} \tag{6.8.22}
$$

6.9 Natural Modes of Motion

So far in this chapter the nonlinear AUV model has been linearised, and the linear model specific to the IMT testbed determined. This vehicle specific model is now analysed using modal analysis in order to determine the vehicle's natural modes of motion. In this section only the primary natural vehicle dynamics are analysed. Since the conversion to modal form is only briefly discussed, the interested reader should kindly refer to [13] for more information.

6.9.1 Longitudinal Modes of Motion

The specific longitudinal vehicle model developed in the previous section must now be converted to state space form. Since only the natural dynamics are analysed the input matrix is set to zero.

Noting the above, the vehicle's natural longitudinal dynamics, in state space form, are given by,

$$\dot{\boldsymbol{x}}_l = \boldsymbol{A}_l \boldsymbol{x}_l \tag{6.9.1}$$

with,

$$\boldsymbol{x}_l = \begin{bmatrix} u & w & q & \theta \end{bmatrix}^T \tag{6.9.2}$$

$$\begin{aligned} \boldsymbol{A}_l &= \boldsymbol{M}_l^{-1} \boldsymbol{A}_l^{'} \\ &= \begin{bmatrix} -0.4189 & -0.1583 & 0.0705 & 0.11495 \\ 0.05877 & -0.7194 & 0.5715 & -0.00457 \\ -0.03516 & 1.3983 & -3.5532 & -1.0866 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{6.9.3}$$

According to [13], the system matrices can be converted to the model form by using a transformation matrix containing the eigenvectors of the

system. This matrix is given by,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.0213 & -0.144 & -0.295 & 0.962 \\ 0.192 & -0.122 & 0.441 & -0.00496 \\ -0.944 & -0.285 & 0.225 & -0.0998 \\ 0.268 & 0.742 & 0 & 0.2524 \end{bmatrix} \begin{bmatrix} z_{l_1} \\ z_{l_2} \\ z_{l_3} \\ z_{l_4} \end{bmatrix} \tag{6.9.4}$$

The longitudinal system dynamics can now be written in terms of the eigenvalues of the system. This is given in state space form as,

$$\dot{\boldsymbol{z}}_{\boldsymbol{l}} = \boldsymbol{F}_{\boldsymbol{l}} \boldsymbol{z}_{\boldsymbol{l}} \tag{6.9.5}$$

with,

$$\boldsymbol{z}_{\boldsymbol{l}} = \begin{bmatrix} z_{l_1} & z_{l_2} & z_{l_3} & z_{l_4} \end{bmatrix}^T \tag{6.9.6}$$

$$\boldsymbol{F}_{\boldsymbol{l}} = \begin{bmatrix} -3.529 & 0 & 0 & 0 \\ 0 & -0.38368 & 0.30341 & 0 \\ 0 & -0.30341 & -0.38368 & 0 \\ 0 & 0 & 0 & -0.39525 \end{bmatrix} \tag{6.9.7}$$

Each of the eigenvalues in equation 6.9.7 represent a natural longitudinal mode of motion for the IMT AUV testbed. These are now briefly discussed.

Eigenvalue: -3.529

$$\omega_n = 3.529 \tag{6.9.8}$$

$$\tau = 0.283 \tag{6.9.9}$$

$$\zeta = 1 \tag{6.9.10}$$

This eigenvalue represents a non-oscillatory exponentially decaying

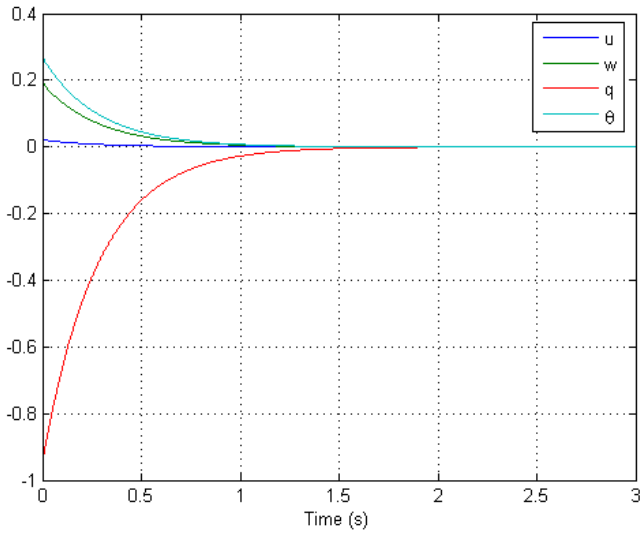

Figure 6.1: Response for longitudinal mode at $\lambda = -3.529$

mode of motion, and is associated with the following eigenvector,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.0213 \angle 0^o \\ 0.192 \angle 0^o \\ 0.944 \angle 180^o \\ 0.268 \angle 0^o \end{bmatrix} \tag{6.9.11}$$

As is evident from the eigenvector given above, this mode is primarily visible in pitch rate, and to a lesser extent , in the pitch angle as well as in the angle of attack. Physically, this mode represents the vessel's tendency to counteract a disturbance in pitch rate.

In order to show a time domain response to this mode, the longitudinal model is initialised with the values of the eigenvector associated with the mode of motion. The response obtained is shown in figure 6.1.

Eigenvalue: $-0.38368 \pm 0.30341j$

$$\omega_n = 0.48915 \tag{6.9.12}$$
$$\tau = 2.04 \tag{6.9.13}$$
$$\zeta = 0.784 \tag{6.9.14}$$

This eigenvalue represents a oscillatory, decaying mode of motion with a period of approximately 12.8 s, and is associated with the following eigenvector,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.32827 \angle -116^o \\ 0.45756 \angle 105.46^o \\ 0.36311 \angle 141.71^o \\ 0.742 \angle 0^o \end{bmatrix} \tag{6.9.15}$$

As is evident from the eigenvector given above, this mode is visible in all the longitudinal variables. Physically, this mode shows the interaction between the restoring moment and the vessel's longitudinal hydrodynamic properties. As the restoring moment is reduced this mode eventually becomes unstable.

In order to show a time domain response to this mode, the longitudinal model is initialised with the real values of the eigenvector corresponding to the mode of motion. The response obtained is shown in figure 6.2.

Eigenvalue: -0.39525

$$\omega_n = 0.39525 \tag{6.9.16}$$
$$\tau = 2.53 \tag{6.9.17}$$
$$\zeta = 1 \tag{6.9.18}$$

This eigenvalue represents a non-oscillatory exponentially decaying

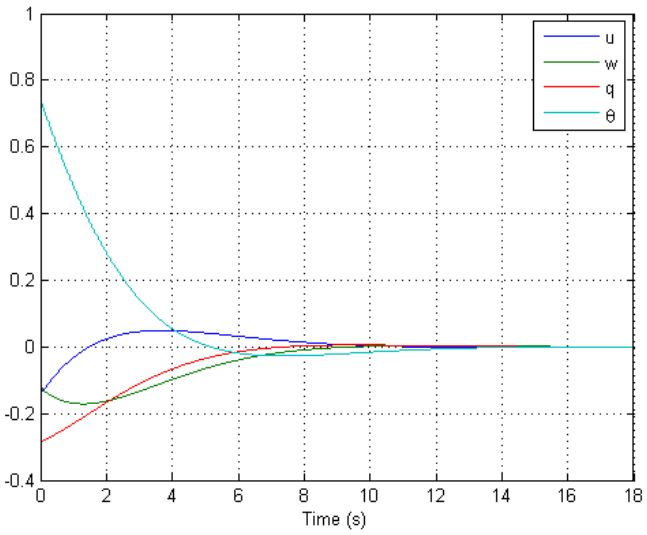

Figure 6.2: Response for longitudinal mode at $\lambda = -0.38368 \pm 0.30341j$

mode of motion, and is associated with the following eigenvector,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.962 \angle 0^o \\ 0.00496 \angle 180^o \\ 0.0998 \angle 180^o \\ 0.2524 \angle 0^o \end{bmatrix} \tag{6.9.19}$$

As is evident from the eigenvector given above, this mode is primarily observed in the axial velocity[4]. Physically, this mode represents the vehicle's tendency to return to the trim velocity after an axial velocity perturbation.

In order to show a time domain response to this mode, the longitudinal model is initialised with the values of the eigenvector associated with the mode of motion. The response obtained is shown in figure 6.3.

[4]Where the first mode was primarily visible in pitch rate.

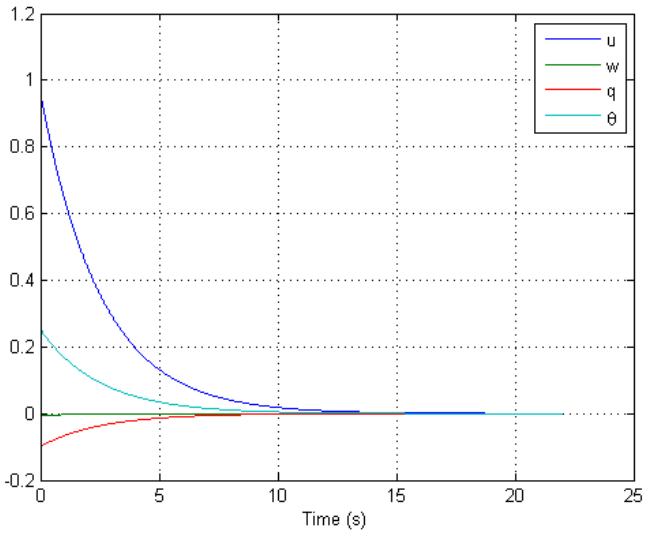

Figure 6.3: Response for longitudinal mode at $\lambda = -0.39525$

6.9.2 Lateral Modes of Motion

The specific lateral vehicle model developed in the previous section must now be converted to state space form. As before, only the natural dynamics are analysed, and therefor, the input matrix is set to zero.

Noting the above, the vehicle's natural lateral dynamics, in state space form, are given by,

$$\dot{\boldsymbol{x}}_t = \boldsymbol{A}_t \boldsymbol{x}_t \tag{6.9.20}$$

with,

$$\boldsymbol{x}_t = \begin{bmatrix} v & p & r & \phi \end{bmatrix}^T \tag{6.9.21}$$

$$\begin{aligned} \boldsymbol{A}_t &= \boldsymbol{M}_t^{-1} \boldsymbol{A}_t' \\ &= \begin{bmatrix} -0.414 & 0 & 0 & -0.1055 \\ 0.253 & -0.0113 & -0.244 & -11.737 \\ 0.436 & 9.54 \times 10^{-5} & -0.4199 & 0 \\ 0 & 1 & -0.0701 & 0 \end{bmatrix} \end{aligned} \tag{6.9.22}$$

According to [13], the system matrixes can be converted to the model form by using a transformation matrix containing the eigenvectors of the system. This matrix is given by,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0616 & 0.0704 & -0.00851 & -0.001005 \\ -0.0783 & 0.0762 & -0.96 & 0 \\ -0.9948 & 0.9945 & -0.000258 & 0.0010769 \\ 0.0191 & -0.0167 & 0.000362 & 0.28016 \end{bmatrix} \begin{bmatrix} z_{t_1} \\ z_{t_2} \\ z_{t_3} \\ z_{t_4} \end{bmatrix} \tag{6.9.23}$$

The lateral system dynamics can now be written in terms of the eigenvalues of the system. This is given in state space form as,

$$\dot{\boldsymbol{z}}_{\boldsymbol{l}} = \boldsymbol{F}_{\boldsymbol{l}} \boldsymbol{z}_{\boldsymbol{l}} + \boldsymbol{G}_{\boldsymbol{l}} \boldsymbol{u}_{\boldsymbol{l}} \tag{6.9.24}$$

with,

$$\boldsymbol{z}_{\boldsymbol{t}} = \begin{bmatrix} z_{t_1} & z_{t_2} & z_{t_3} & z_{t_4} \end{bmatrix}^T \tag{6.9.25}$$

$$\boldsymbol{F}_{\boldsymbol{t}} = \begin{bmatrix} -0.4469 & 0 & 0 & 0 \\ 0 & -0.389 & 0 & 0 \\ 0 & 0 & -0.0047 & 3.4262 \\ 0 & 0 & -3.4262 & -0.0047 \end{bmatrix} \tag{6.9.26}$$

Each of the eigenvalues in equation 6.9.26 represent a natural lateral mode of motion for the IMT AUV testbed. These are now briefly discussed.

Eigenvalue: -0.4469

$$\omega_n = 0.4469 \tag{6.9.27}$$

$$\tau = 2.24 \tag{6.9.28}$$

$$\zeta = 1 \tag{6.9.29}$$

This eigenvalue represents a non-oscillatory exponentially decaying

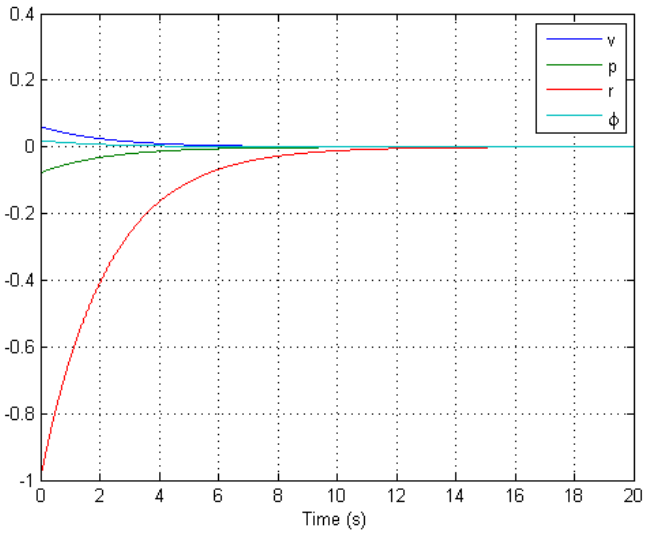

Figure 6.4: Response for lateral mode at $\lambda = -0.4469$

mode of motion, and is associated with the following eigenvector,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0616 \angle 0^o \\ 0.0783 \angle 180^o \\ 0.9948 \angle 180^o \\ 0.0191 \angle 0^o \end{bmatrix} \tag{6.9.30}$$

As is evident from the eigenvector given above, this mode is primarily visible in the yaw rate.

In order to show a time domain response to this mode, the lateral model is initialised with the values of the eigenvector corresponding to the mode of motion. The response obtained is shown in figure 6.4.

Eigenvalue: -0.389

$$\omega_n = 0.389 \tag{6.9.31}$$

$$\tau = 2.57 \tag{6.9.32}$$

$$\zeta = 1 \tag{6.9.33}$$

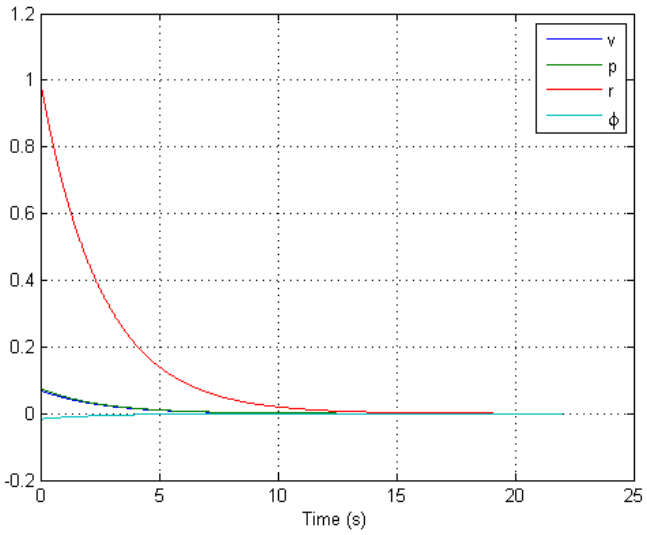

Figure 6.5: Response for lateral mode at $\lambda = -0.389$

This eigenvalue represents a non-oscillatory exponentially decaying mode of motion, and is associated with the following eigenvector,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0704 \angle 0^o \\ 0.0762 \angle 0^o \\ 0.9945 \angle 0^o \\ 0.0167 \angle 180^o \end{bmatrix} \tag{6.9.34}$$

As is evident from the eigenvector given above, this mode is primarily visible in the yaw rate.

In order to show a time domain response to this mode, the lateral model is initialised with the values of the eigenvector associated with the mode of motion. The response obtained is shown in figure 6.5.

Eigenvalue: $-0.0047 \pm 3.4262j$

$$\omega_n = 3.4262 \tag{6.9.35}$$

$$\tau = 0.292 \tag{6.9.36}$$

$$\zeta = 0.0014 \tag{6.9.37}$$

This eigenvalue represents a lightly damped oscillatory, decaying mode of motion with a period of approximately 1.83 s, and is associated with the following eigenvector,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.00857 \angle -173.26^o \\ 0.96 \angle 180^o \\ 0.00111 \angle 103.47^o \\ 0.28016 \angle 89.926^o \end{bmatrix} \tag{6.9.38}$$

As is evident from the eigenvector given above, this mode is primarily visible in the roll rate. Physically, this mode represents the interaction between the resorting moment and the vessel's lateral hydrodynamic properties. It must be noted that the restoring moment has no direct effect on yaw rate and yaw angle.

In order to show a time domain response to this mode, the lateral model is initialised with the real values of the eigenvector corresponding to the mode of motion. The response obtained is shown in figure 6.6.

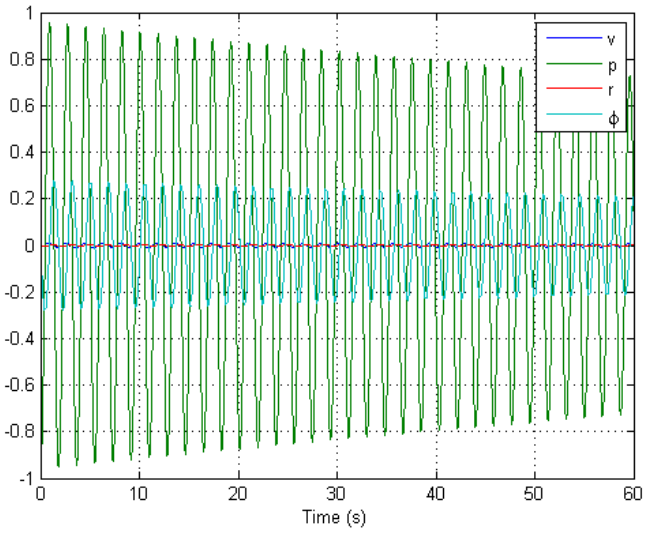

Figure 6.6: Response for lateral mode at $\lambda = -0.0047 \pm 3.4262j$

Chapter 7

Visualisation Software

In section 1.2.2 it was determined that a generic visualisation tool should be developed. The visualisation tool developed during this project is the topic of this chapter.

The intent of this chapter is not to provide a detailed discussion of the implementation, but rather to provide an overview of the system developed as well as providing details on some of the more interesting aspects of the implementation.

7.1 Development Libraries Used

Before any software development can be conducted the primary development toolkits and libraries must be chosen. In the case of the visualisation software the following types of libraries are required:

- an application development framework or GUI library;
- a library to provide hardware accelerated 3D
- and preferably a library to handle the external 3D file format in order to simplify scene and vehicle loading

A number of criteria were considered while choosing the libraries used during the development of the software. The most important of these criteria are:

- **Code Portability**: It is always advantageous to produce platform independent code as far as possible. This ensures that possible future platform changes are as simple as possible. To help in this regard, cross platform toolkits are preferred.

- **Open Source**: As far as possible only open source tool kits and libraries are considered. These tools are typically well maintained and supported by a large development community. There is also a large monetary saving in using open source tools, as many commercial development tool kits are prohibitively expensive.

- **Detailed Documentation**: Few aspects in software development lead to as many delays and difficulties as incomplete or, even worse, incorrect documentation. As far as possible only well documented libraries will be used.

- **Widely Used**: As far as possible only widely used tools and libraries are used, since a large user base typically reduces the number of undocumented bugs.

- **Performance**: When developing 3D application only high performance toolkits can typically be used. This has the effect of eliminating many cross platform systems, specifically those utilising virtual machines or other managed code schemes. Thus, for the development of the visualisation system, only toolkits generating native code are considered.

Taking the criteria mentioned above into consideration, the following toolkits and/or libraries are used: QT as the main application development framework and widget library; OpenGL is used to provide high performance hardware accelerated 3D; and lib3DS provides file handling for the 3DS file format.

Each of these tools, as well as to what extent they meet the criteria mentioned above, are now briefly discussed.

7.1.1 QT

Development of what would eventually become the QT toolkit started in 1991, with the first public release occurring in 1995. Today QT is a widely used toolkit available on all major platforms in open source and commercial versions.

The visualisation software developed for this project makes extensive use of QT 4, the latest version of QT released in 2005. QT 4 is the first version to be released under both open source and commercial licenses for all supported platforms. Previously open source versions were only available for the Linux/X11 and Linux emended platforms.

QT is exceptionally well documented by means of detailed reference documentation and numerous example and demo programs.

Good performance is expected, since native platform GUI rendering is used and native binaries are generated. In addition the toolkit provides numerous useful features including integration with OpenGL by means of a standard OpenGL widget, as well as many advanced object orientated features.

7.1.2 OpenGL

In order to provide high performance hardware accelerated 3D graphics, OpenGL is used as the rendering system for the visualisation software. The OpenGL API was first released in 1992 as the result of an initiative lead by SGI to develop a platform and vendor independent 2D and 3D graphics API[23].

OpenGL is currently the most widely used cross-platform rendering system and implemented on all major platforms. The main competitor to OpenGL is Microsoft's Direct3D system which is only available on the Windows platform and as such is not considered for this project.

Another major advantage with using OpenGL is the exceptional documentation available in the form of online guides as well as various published books. The best known of these books, the *OpenGL Programming Guide* (commonly know as the *The Red Book*, due to the colour of its

cover), was extensively used throughout the development of the rendering code for this project.

7.1.3 lib3DS

In order to manage the proprietary .3DS file format, lib3ds is used. This library provides the only good, although definitely not perfect, alternative to Autodesk's commercial 3DS tools.

According to [24], development started in 1996, with the first release occurring in 1999. The library is under fairly constant development, and is widely used by the open source community. For this project version 1.2.0 is used.

The lib3DS library provides good cross platform portability as it is written in ANSI-C, and according to [24] is compatible with both the GNU build system as well with Microsoft Visual C++ 8.

One major disadvantage of using this library is the lack of proper documentation. As expected this caused many difficulties including numerous bugs. Typically libraries as poorly documented as lib3ds should not even be considered, but in this case no better alternative could be found.

7.2 Object Orientated Concepts and Techniques

A number of object orientated techniques where used during the development of the visualisation software, the less-common of these techniques are briefly introduced here.

7.2.1 Multiple Inheritance

Multiple inheritance is en extension of the normal inheritance concept, in which a class is allowed to derive from more than one parent class.

Using multiple inheritance should for the most part be avoided, since its use may result in unexpected ambiguities. One of the most common complications which may result from the use of multiple inheritance, is the so called diamond inheritance. A detailed discussion on this as well as multiple inheritance in general can be found in[25].

The type of multiple inheritance, multiple interface inheritance, used in this project is however very simple, and for the most part its use is not controversial. Interface inheritance, although not directly supported by c++, is simple to implement using virtual classes. This type of multiple inheritance is very useful for enforcing requirements on the derived classes, as well as enabling derived classes to be placed in a single non-void array, list or container.

7.2.2 Model/View Architecture

The basic concept behind a model/view architecture is the separation between data storage and data access, or views. QT provides numerous model/view classes, which significantly simplifies the implementation of such a system. For more information about the model/view architecture or the model/view classes provided by QT, please consult the QT reference documentation[26].

7.3 Implementation Overview

The visualisation software, named QGLEngine, is developed in C++ and consists out of approximately 40 classes. These classes can roughly be subdivided into a few main functional groups. They are: OpenGL renderer; OpenGL camera ; 3DS file loader; 3D Objects system; 2D Objects system; Save/Load system; a network sever; and the GUI.

Each of these functional groups are now briefly discussed.

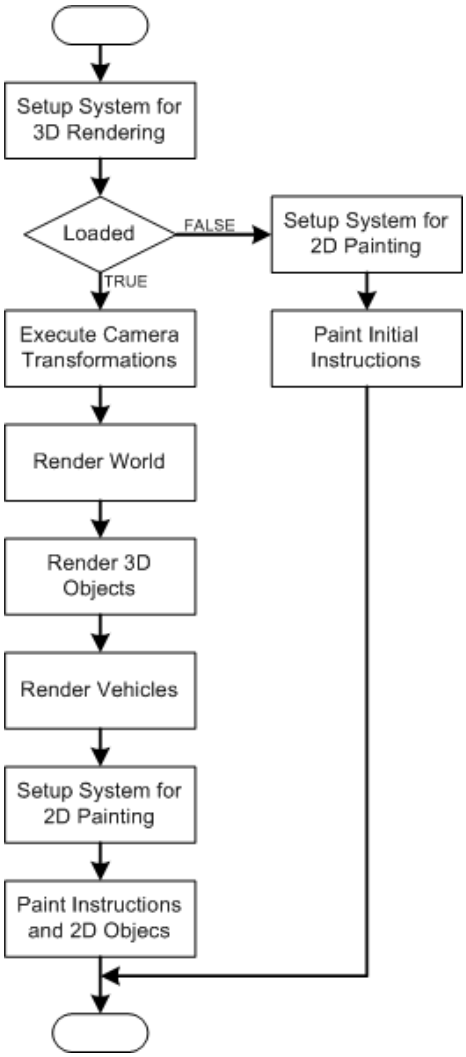

Figure 7.1: Main Rendering Cycle

7.3.1 OpenGL Renderer

The OpenGL renderer is primarily responsible for rendering the 3D world and the 2D graphics. The renderer is essentially implemented in a single class, QGLWidgit.

The renderer developed for QGLEngine is, for the most part, very simple and employs no advanced rendering techniques, such as the culling of invisible polygons. One interesting aspect of the renderer though is the use of over painting in order to render instructions and 2D objects.

The term over painting refers to using the widget toolkits 2D painting system to paint over the 3D scene. This has a number of advantages over using OpenGL directly in order to paint the 2D graphics. Amongst these are:

- The 2D painting functions provided by the QT toolkit are higher level functions and thus their use significantly simplifies the design of the 2D objects.
- Using the QT painting system improves font handling, as the toolkit's font system can be used.
- Data access is significantly simplified since the same toolkit is used for data management and for object painting.

Figure 7.1 shows the main steps used by the renderer to paint the scene as well as to over paint the 2D objects and instructions.

7.3.2 OpenGL Camera

In order to better visualise the motions of the vehicle being simulated, a multi-mode OpenGL camera is implemented. The camera is primarily implemented in a single c++ class, namely glCamera.

The camera implemented for this project supports four different camera modes. They are:

- **Earth fixed mode**: In this mode the camera is placed at a stationary user defined position. From this fixed location the camera tracks

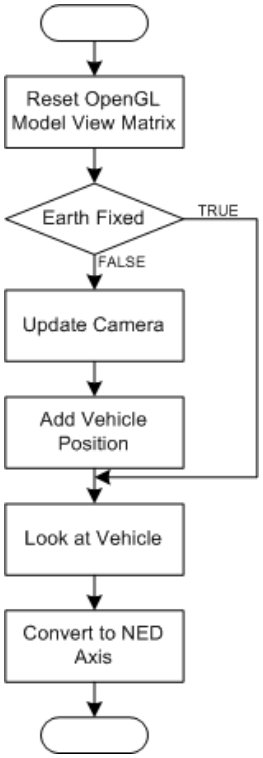

Figure 7.2: Camera Execute

the vehicle being simulated by pitching and yawing to the correct attitude, while remaining in the upright position.

- **Body fixed mode**: When this mode is selected the camera is fixed to the vehicle's body fixed axis system[1], therefore the camera translates and rotates with the vehicle being simulated.

- **Wind fixed mode**:In this mode the camera is fixed to the vehicle's instantaneous wind fixed axis system at some user defined point. This camera mode is identical to the body fixed mode, with the exception that the camera is rotated through the angles of attack and sideslip.

- **Pilot view mode**: This mode was developed at the request of some students in the ESL. It is a relatively simple body fixed camera

[1]The body fixed point at wich the camera is attached, is user defined.

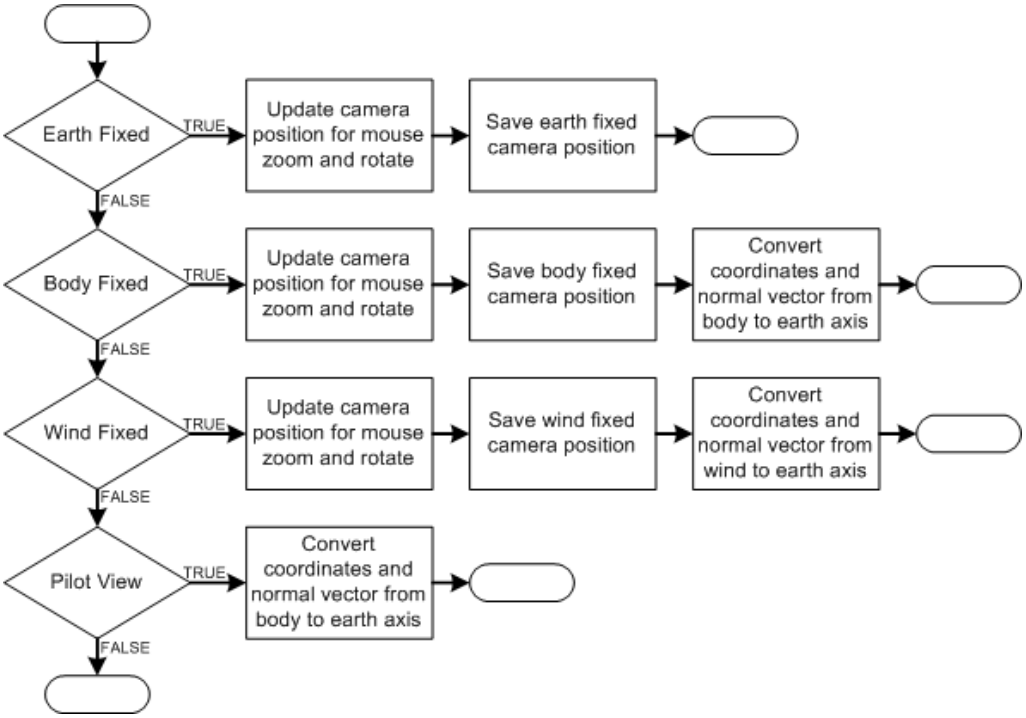

Figure 7.3: Camera Update

mode in which the camera is placed on the vehicle. In this mode the vehicle is not rendered.

The OpenGL camera developed for this project is divided into two main functional units, these are:

- **Camera Execute**: This function is called by the renderer on each rendering cycle and is responsible for applying all camera related transformations. Figure 7.2 shows this graphically.
- **Camera Update**: This function is called each time the camera's position and/or orientation relative to inertial space must be updated. Figure 7.3 shows this graphically.

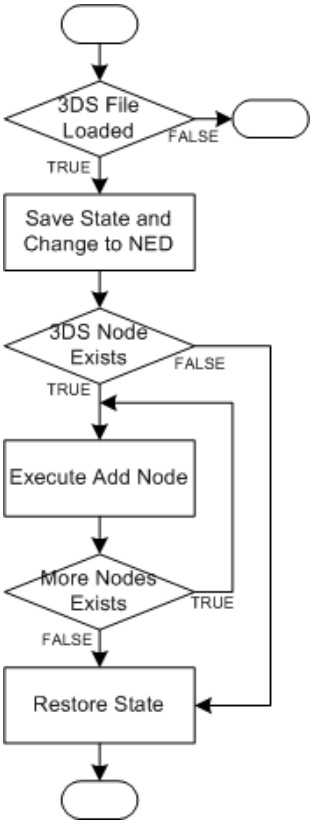

Figure 7.4: File Loader Draw

7.3.3 3DS File Loader

The 3DS file loader is responsible for loading the world and vehicle models from the external 3DS files. This module is primarily implemented in a single class, namely file_loader_3ds.

The 3DS file loader developed for this project is based on the lib3DS player example as well as the libQGLViewer 3dsViewer example. These examples where combined, modified and expanded in order to suit the needs of this project. The 3DS file loader developed for this project is able to reliably import all basic scene and model data. This includes the following.

- All polygon information can be imported from the 3DS file. This includes information such as vertex locations as well as normal vector

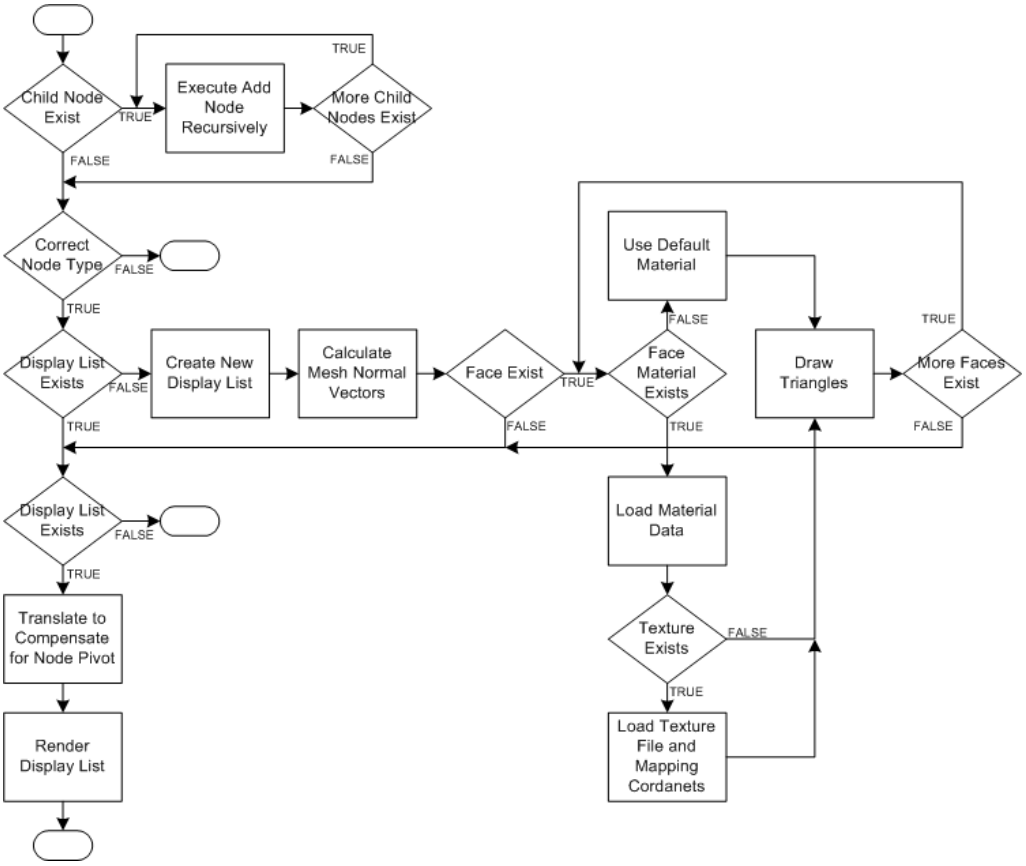

Figure 7.5: File Loader Add Node

data.

- Material information is imported in the form of ambient, defuse, specular and shininess values.
- All texture mapping information can be imported from a 3DS file. This includes mapping coordinates as well as the texture file name.

Since only basic data is imported some, useful information is unfortunately lost, including the following:

- No lighting information is imported and as such the scene lighting is hard coded into the rendering engine.

- Environmental effects are not imported. The primary disadvantage with this is that in order to include fog effects, which are necessary for realistic underwater scenes, support must be provided by other means.

- No data to describe advanced rendering effects are imported. This includes effects such as bump mapping, which can be very useful in generating realistic low polygon surfaces.

The file loader developed during this project is subdivided into two main functional units, they are:

- **File Loader Draw**: This function uses the add node function in order to paint the vehicle or scene contained in the 3DS file. It is called by the renderer each time the vehicle or scene must be rendered. Figure 7.4 depicts this graphically.

- **File Loader Add Node**: When executed for the first time, this recursive function loads the model data from the external 3DS file and saves this data in display lists. After this the display lists are rendered. During subsequent rendering cycles only the second step is required, therefore performance is significantly improved. Figure 7.5 shows this process graphically.

7.3.4 3D Objects System

As part of the visualisation software a 3D objects system was implemented. The purpose of this system is to assist the engineer in annalising various performance criteria. These criteria may include: waypoint following; longitudinal oscillations, and turning performance.

With reference to figure 7.6, the 3D Objects system developed for QGLEngine can roughly be divided into two main components. They are: a 3D objects rendering system and a waypoints data management system.

Each of these components are now briefly discussed.

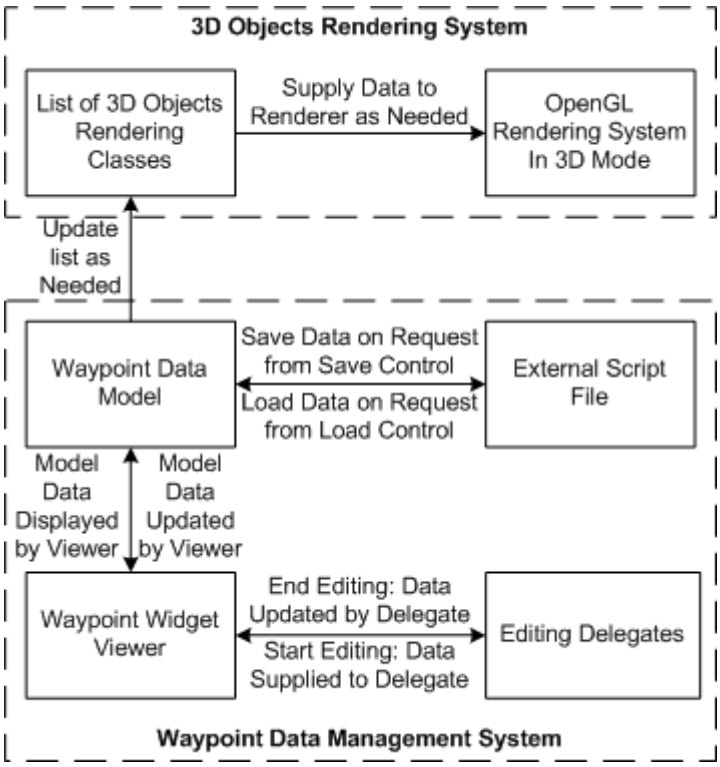

Figure 7.6: 3D Objects System

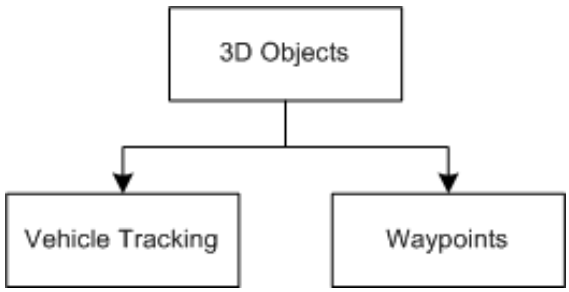

Figure 7.7: 3D Objects System

7.3.4.1 3D Objects Rendering System

The 3D objects rendering system is implemented as a two level hierarchy of classes. Its implementation employs numerous object orientated concepts. These include: inheritance; virtual functions, data encapsulation; polymorphism; and operator overloading.

At the time of writing this thesis, only two 3D objects are available, as shown in figure 7.7. Therefore the 3D objects rendering system primarily consists out of three classes, they are:

- **Object3D**: This is the parent class for the 3D objects system. It is responsible for all operations which are common between all the different types of 3D objects. This class also enforces the requirements for all the child 3D objects classes.
- **Waypoint**: This class implements the waypoint 3D object. The waypoints generated by this class are spherical and placed at a user defined location. The radius as well as the colour of each waypoint is user defined.
- **traceLine**: This class implements the 3D vehicle tracking lines. Vehicle tracking lines are very useful for analysing vehicle performance. This class implements a fairly simple tracking system. It allows the user to clear the tracing lines, pause the tracing process, and show/hide the tracking lines.

7.3.4.2 Waypoint Data Management

In order to allow for the creation, editing, deletion, loading and saving of waypoints, some form of waypoint data management must be provided. As with the rest of this development effort, it is attempted to make use of modern object orientated techniques. Therefore one such technique, the model/view architecture, is selected.

With reference to figure 7.6, the waypoint data management system consists of three main components. They are:

- **Waypoint Data Model**: This component provides a viewer independent data store for all waypoint related data. It is implemented in a single class namely, WaypointTableModel. An access function is provided in order to enable access to the data model from anywhere in QGLEngine.

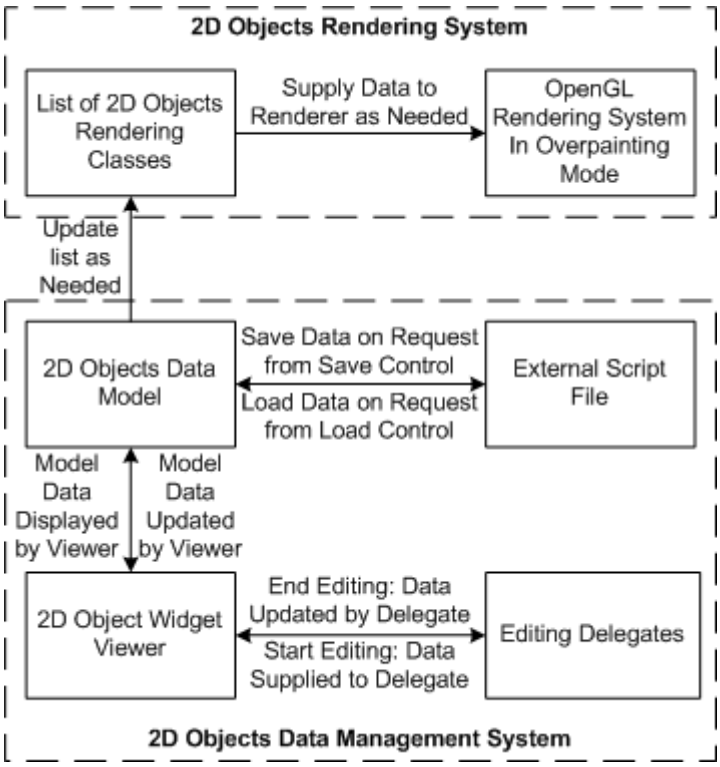

Figure 7.8: 2D Objects System

- **Waypoint Widget Viewer**: The waypoint widget acts as the primary view to the model, and is described later in this section.
- **Editing Delegates**: A number of editing delegates are provided in order to constrain the user input to acceptable values. Without these delegates it would be easy for the user to accidently input incorrectly formatted data. The actual delegates implemented are discussed later.

In addition to these three components, the shared external script file is also shown as part of the system, as it provides a means of editing, saving and loading the model.

7.3.5 2D Objects System

As part of the visualisation software a 2D objects system was implemented. As with the 3D objects system, the purpose of this is to assist the engineer in analysing various performance criteria. These criteria may include: angle of incidence; actuator deflection angles, cross track error, and many more.

With reference to figure 7.8, the 2D Objects system developed for QGLEngine can roughly be divided into two main components. They are: a 2D objects rendering system; and a 2D objects data management system.

Each of these components are now briefly discussed.

7.3.6 2D Objects Rendering System

The 2D objects rendering system is implemented as a two level hierarchy of classes. As with the 3D system, the implementation employs numerous object orientated concepts. These include: inheritance; virtual functions, data encapsulation; polymorphism; and operator overloading.

As previously mentioned the 2D objects are not implemented directly in OpenGL, but rather make use of QT's 2D painter system. Using the 2D painter system significantly simplified the development of the 2D objects.

For the purpose of positioning the 2D objects on the rendering canvas, the canvas is subdivided into a fixed number of equally sized blocks. The size of these blocks are automatically rescaled as a function of the canvas resolution. Each of the 2D objects are designed to occupy a fixed number of these blocks, and therefore automatically rescale as the canvas size changes.

At the time of writing this thesis, three different 2D objects have been implemented, as shown in figure 7.9. Therefore the 2D objects rendering system consists primarily out of 4 classes. They are:

- **Object2D**: This is the parent class for the 2D objects system. It is

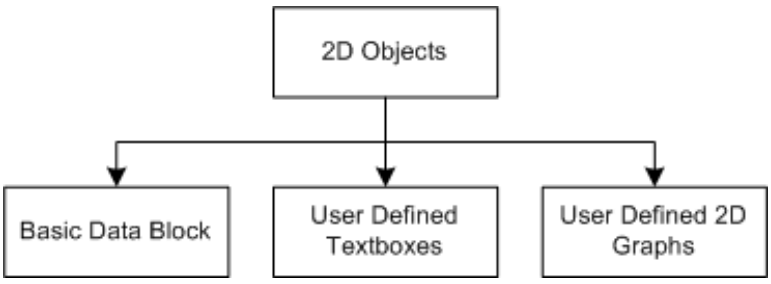

Figure 7.9: 2D Objects Rendering System

responsible for all operations which are common between all the different types of 2D objects. This class also enforces the requirements for all the child 2D objects classes.

- **BasicData**: This 2D object supplies the user with basic vehicle data. This data includes the vehicle position, attitude, as well as the velocity vector. The basic data object is always visible, and not user hidable.

- **Textbox**: This object allows for the creation of user defined text boxes. The user must specify the following: the position of the text box; the data channel to connect to; the valid data range; the text box update period; and finally the colour of the text box.

- **Graph**: This object allows for the creation of user defined two dimensional graphs. The user must specify the following: the position of the graph; the data channel to connect to; the valid data range; the graph box update period; and finally the colour of the graph.

7.3.6.1 2D Objects Data Management

As was the case with the 3D objects system, it is necessary to provide some form of data management for the 2D objets system. As before, a model/view architecture is selected.

With reference to figure 7.8, the 2D objects data management system consists out of three main components. They are:

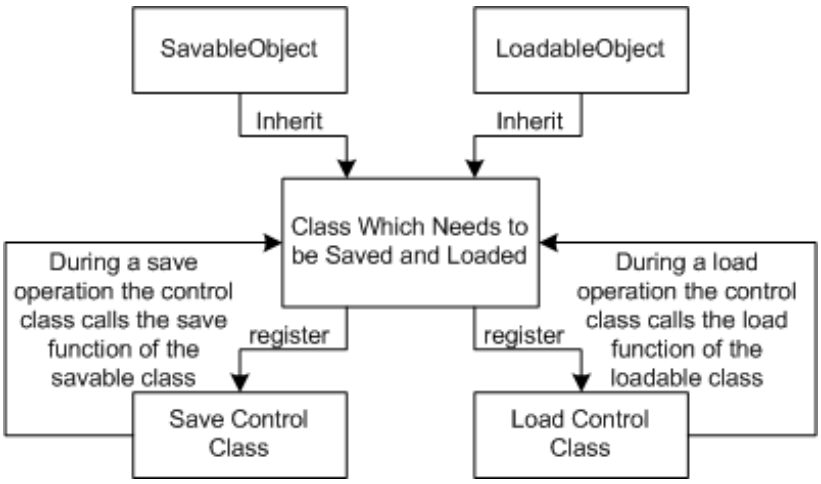

Figure 7.10: Save/Load System

- **2D Objects Data Model**: This component provides a viewer independent data store for all 2D object related data. It is implemented in a single class, namely Object2DTableModel. An access function is provided in order to enable access to the data model from anywhere in QGLEngine.

- **2D Objects Widget Viewer**: The 2D objects widget acts as the primary view to the model, and is described later in this section.

- **Editing Delegates**: A number of editing delegates are provided in order to constrain the user input to acceptable values. Without these delegates it would be easy for the user to accidently input incorrectly formatted data. The actual delegates implemented are discussed later.

In addition to these three components, the shared external script file is also shown as part of the system, since it provides a means of editing, saving and loading the model.

7.3.7 Save Load System

During the development of QGLEngine the constantly changing save and load requirements proved frustrating, since the related function needed to be constantly rewritten. In order to overcome this difficulty a save and load system was developed.

This system is based on a delegating concept which operates as follows: All classes whom require save and/or load facilities must inherit the corresponding pure virtual class or classes[2]; the saveable or loadable class must implement all functions required by the inherited pure virtual class(es); finally, the saveable and/or loadable class must register its need with the corresponding control class(es).

During a save or load operation control is passed to the corresponding control class. This class then performs all necessary initialisation procedures after which it passes control to all registered saveable or loadable classes.

Figure 7.10 gives a simplified graphical representation of this system.

7.3.8 Network Server

The network server provides all the required network functionality for QGLEngine and is primarily implemented in the server class.

The network server developed for this project is fairly simple. It is responsible for receiving data packets from the MATLAB simulator and updating all necessary variables with the newly received data.

In order to make the network server as simple and light weight as possible, UDP was chosen as the sole supported network protocol. UDP is a simple lightweight datagram based network protocol, and is not suitable for environments were reliable data transfer is important. Reliable data transfer was deemed unimportant in this project, since the loss of data will at the worst result in a few missed rendering frames.

[2]The concept of multiple inheritance is used. The class can inherit saveableObject and/or loadableObject.

DATA PACKET	
Variable	***Size***
Frame Number	Integer (4 Bytes)
North	Double (8 Bytes)
East	Double (8 Bytes)
Down	Double (8 Bytes)
Psi (Roll)	Double (8 Bytes)
Theta (Pitch)	Double (8 Bytes)
Phi (Yaw)	Double (8 Bytes)
Vbar (Velocity)	Double (8 Bytes)
Alpha (Angle of Attack)	Double (8 Bytes)
Beta (Angle of Sideslip)	Double (8 Bytes)
Upload Count	Unsigned Integer (4 Bytes)
Data Channel (User Data)	32 Doubles (256 Bytes)

Figure 7.11: Data Packet Definition

The data is transferred in well defined packets, of which the definition must be known to both the sending and receiving parties. Figure 7.11 shows the data packet definition used in this project.

7.3.9 Graphical User Interface

A major development goal for QGLEngine is ease of use, and to help in this regard a rich user friendly GUI was developed. The GUI allows the user to manipulate the various system settings without the need to edit scripts, or use the command line interface.

Figure 7.12 shows the main window shortly after startup. As there is no running visualisation the camera setting dock widget on the left is greyed out.

The GUI primarily consist out of five components, they are: the way-points settings widget; the 2D Objects settings widget; the rendering settings widget; the camera settings widget; and the new simulation dialog.

Each of the main GUI components are now briefly discussed.

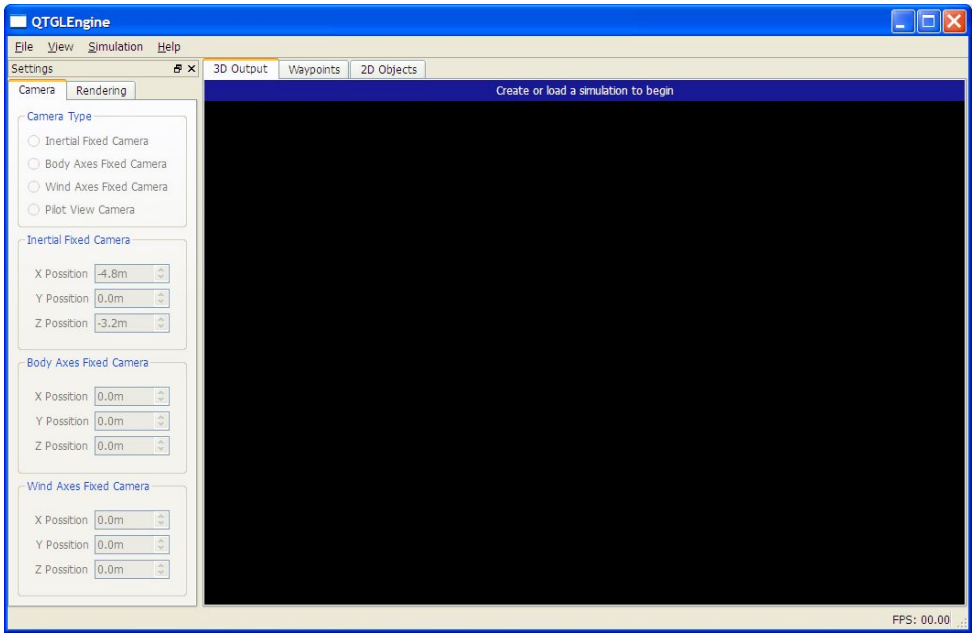

Figure 7.12: Main Window shortly after system startup.

7.3.9.1 Waypoints Settings Widget

When the waypoint 3D object was introduced earlier in this section it was mentioned that in order to place a waypoint a number of parameters must be specified by the user. This GUI component makes specifying these parameters easy.

In an attempt to minimise errors, the user input is constrained by means of editing delegates. A number of different delegates were implemented. They are:

- **ColorListDelegate**: This delegate is used to provide the user with a simple method of selecting the waypoint colour. All available colours are displayed in a drop down list as colour previews as well as the colour name.

- **DoubleSpinBoxDelegate**: This delegate is used where ever the user must be constrained to floating point numbers. In the waypoint settings widget it is used in all columns with the exception of the colour setting column.

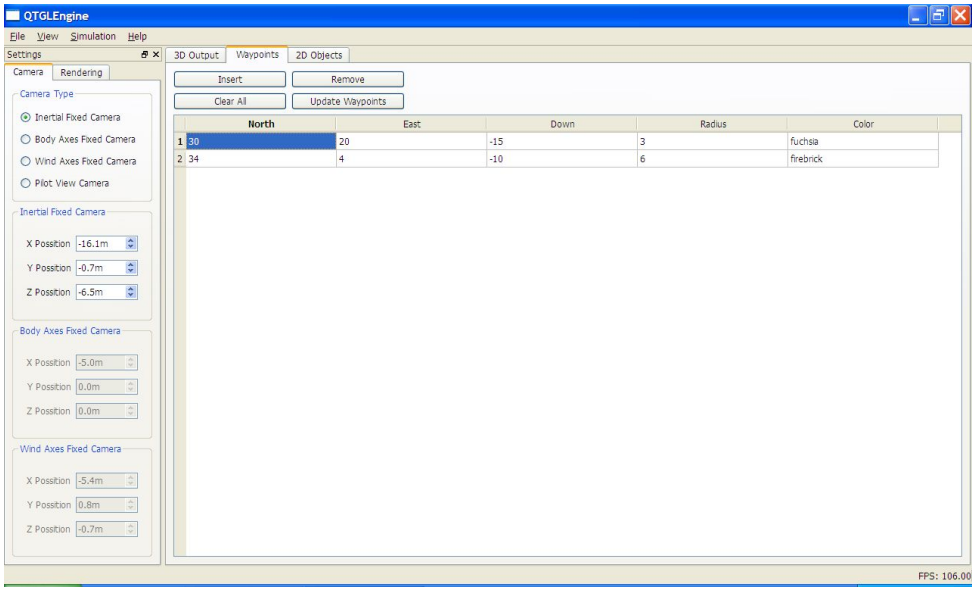

Figure 7.13: Waypoint Settings Tab and Camera Settings Widget

Figure 7.13 shows the waypoint settings widget with two waypoints defined.

7.3.9.2 2D Objects Settings Widget

When the 2D objects were introduced earlier in this section it was mentioned that in order to use these objects a number of parameters must be specified by the user. This GUI component attempts to make specifying these parameters easy.

As was done for the waypoints settings widget, the user input is constrained by means of editing delegates. A number of different delegates where implemented, they are:

- **SpinBoxDelegate**: This delegate is used where ever the user input must be constrained to integer values. In the 2D objects settings widget this delegate is used in the data channel, x coordinate, and y coordinate columns.
- **DoubleSpinBoxDelegate**: This delegate is used where ever the user input must be constrained to floating point numbers. In the

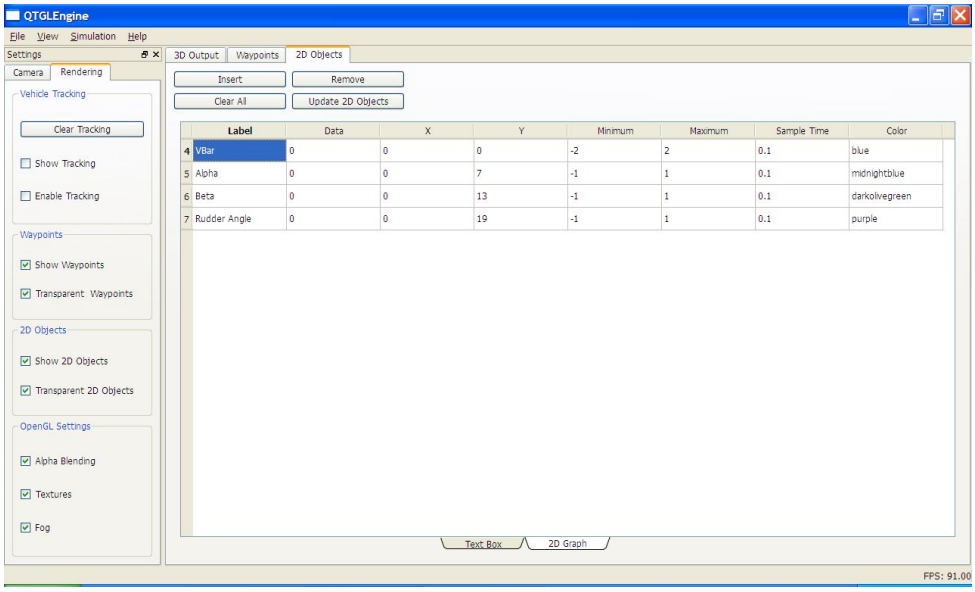

Figure 7.14: 2D Objects Settings Tab and Rendering Settings Widget

2D objects settings widget this delegate is used in the minimum, maximin, and sample time columns.

- **ColorListDelegate**: This delegate is used to provide the user with a simple method of selecting the 2D object background colour. All available colours are displayed in a drop down list as colour previews as well as the colour name.

7.3.9.3 Rendering Settings Tab

The rendering settings tab, shown in figure 7.14, allows the user to edite various renderer specific functions. The functions include the following:

- Showing or hiding of waypoints.
- Enable or disable alpha blending for waypoints.
- Showing or hiding of 2D objects.
- Enable or disable alpha blending for 2D objects.

- Enable or disable the OpenGL fog system

Note that in the current version of QGLEngine the OpenGL alpha blending and texture systems can not be disabled on a global level.

In addition to these renderer specific settings, the vehicle tracking system is also controlled from this widget, allowing the user to enable or disable the tracking systems, show or hide the trace lines, and clear all the lines.

7.3.9.4 Camera Settings Tab

As was previously stated a multi-mode OpenGL based camera was implemented for use in this projects. This widget, shown in figure 7.13, is designed to provide easy access to all camera related settings. These settings include:

- The camera mode to use.
- The position of the earth fixed camera relative to the earth fixed axis system.
- The position of the body fixed camera relative to the body fixed axis system.
- The position of the wind fixed camera relative to the instantaneous wind fixed axis system.

Note that only the selected mode's position is editable, while all other settings are greyed out.

7.3.9.5 New Visualisation Dialog

The new visualisation dialog box, shown in figure 7.15, makes it easy for the user to create a new visualisation. In order to use this function the user must specify the following: the path to the scene .3DS file; the size of the scene in meters; the path to the vehicle .3DS file; the size of the vehicle in meters; and finally the x-, y- and z-offsets for the vehicle.

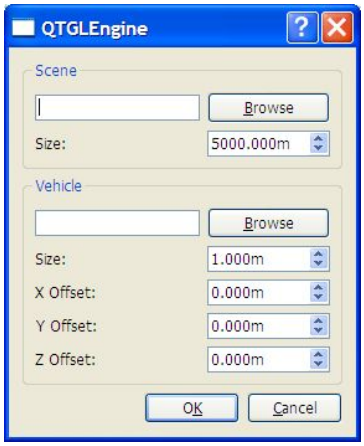

Figure 7.15: New Visualisation Dialog Box

7.4 Known Issues

As can be excepted with a complex software system developed under severe time constraints, some issues remain unresolved. These are summarised below.

7.4.1 Performance

The QGLEngine can be considered somewhat of a performance hog, with most of the blame lying with the very simple renderer used. It is however not deemed feasible to implement a complex and efficient renderer for this project. Some of the blame might also lie with the GUI, where considerable resources are wasted due to the excessively fast update cycle.

7.4.2 Unimplemented Functionality

At the time of writing this thesis, some functionality still remains unimplemented. This unimplemented functionality is primarily related to the OpenGL settings which, with the exception of the fog setting, has not yet been implemented, and therefore has no effect on the rendering.

7.4.3 File Import

The file import process does not always occur as smoothly as was expected. The blame for this seems to lie with the lib3DS library, since the same files also fail to import properly with other software based on this library.

Chapter 8

Simulation System and Examples

In section 1.2.2 it was determined that a six degree of freedom nonlinear simulator should be developed. The simulator developed during this project is the primary topic of this chapter.

The intent of this chapter is not to provide a detailed discussion of the implementation, but rather to provide an overview of the system developed. In addition to this two simulation examples are shown.

8.1 Nonlinear Six Degree of Freedom Simulator

In order to simulate the motions of the vehicle in six degrees of freedom, a nonlinear MATLAB simulator was developed. The high level design for this simulator was conducted in section 1.2.2.

The intent of this section is not to provide a detailed discussion about the implementation of the simulator, but rather to briefly discuss the simulator's main components.

8.1.1 Six Degree of Freedom Block

At the start of this project a six degree of freedom block was already available in the ESL, therefore no detailed discussion of this block is provided

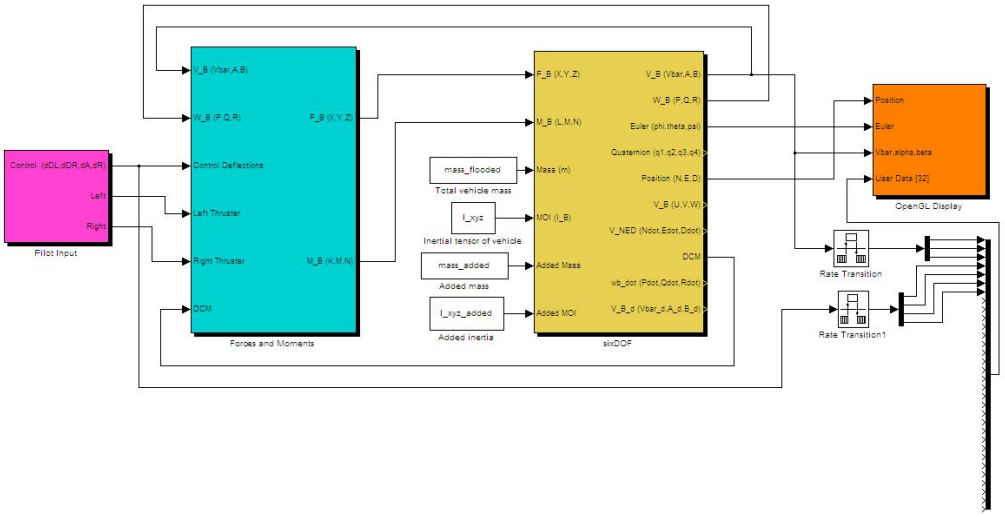

Figure 8.1: Main Simulation System Diagram

here.

A number of modifications where however made to the block[1]. They include: adding support for added mass as well as added inertia input; and adding outputs for accelerations of motion variables.

8.1.2 Forces and Moments Block

Most of the effort involved in developing the MATLAB simulator was in developing the Forces and Moments block. As shown in figure 8.2, this simulink block consists out of a number of sub-blocks. They are: a hydrodynamic model block; a gravity model block; a buoyancy model block; and finally, a thruster model block.

Each of these blocks are now briefly discussed.

Hydrodynamic Model Block

This block is the most complex of all the forces and moment blocks, and is predominantly implemented as a MATLAB s-function. This block im-

[1]Mr. S.J. Pauck is thanked for his assistance in modifying this block.

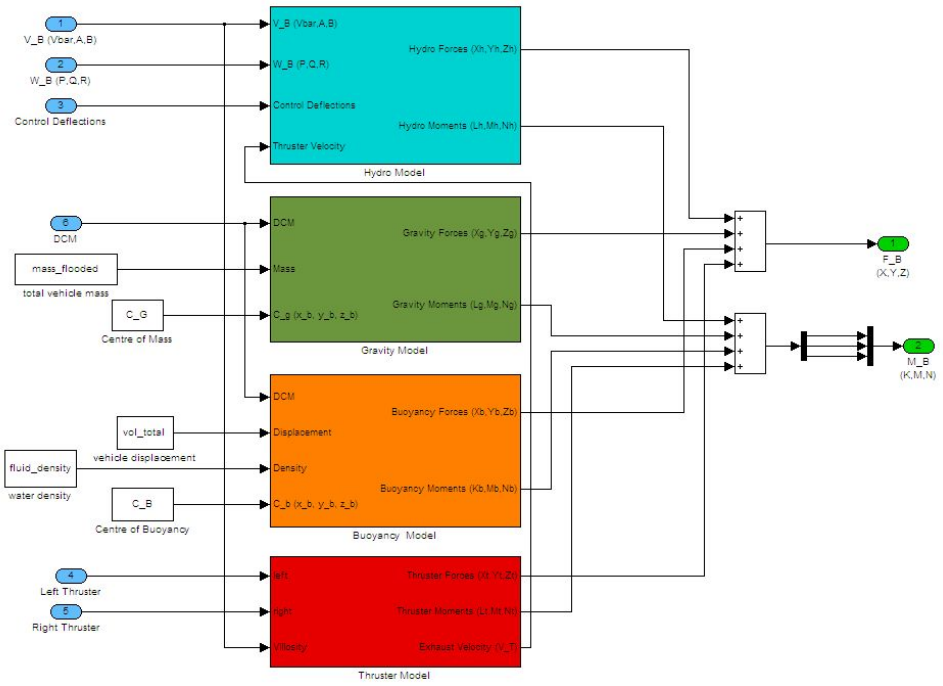

Figure 8.2: Forces and Moments Simulation System Diagram

plements the extended hydrodynamic model derived in chapter 4, with the following exceptions:

- Since it is attempted to create a generic system, additional data supplied by the Datcom was included.
- The Datcom supplies control actuator characteristics in the form of increments due to deflection. It was decided to use this data directly, instead of converting it to control derivatives.

With reference to figure 8.3, the hydrodynamic forces and moments block consists out of the following sub-blocks:

- **Stability Derivatives Block**: This block is responsible for generating all the necessary stability derivatives. It uses the imported DATCOM data and the current angle of attack in order to calculate the stability derivatives for the current flight condition. These derivatives are calculated using piecewise linear interpolation.

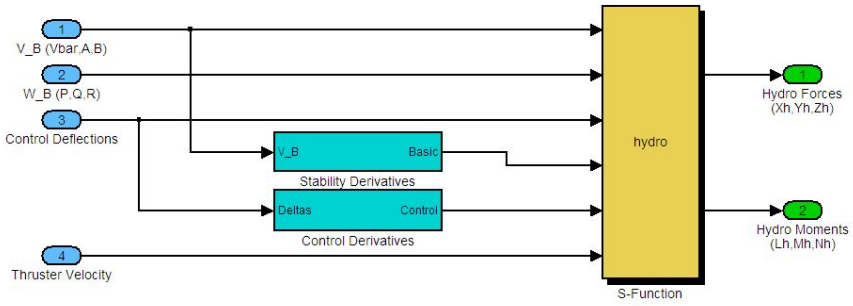

Figure 8.3: Hydrodynamic Forces and Moments Simulink Diagram

- **Control Derivatives Block**: This block is responsible for generating all the necessary coefficient increments due to control surface deflection. It uses the imported DATCOM data and the current control surface deflection angle in order to calculate the coefficient increments for the current flight condition. These coefficient increments are calculated using piecewise linear interpolation.
- **Hydrodynamic S-function Block**: This s-function block implements the hydrodynamic forces and moments equations derived in chapter 4. It uses the derivatives and coefficient increments supplied by the other two blocks in order to calculate the hydrodynamic forces and moments.

Gravity Model Block

This block is very simple. It merely implements the gravity forces and moments model derived in chapter 4, therefore no detailed explanation in warranted. In order to use this block, the following must be provided: the centre of gravity; the mass of the vehicle; and the gravitational acceleration.

Buoyancy Model Block

As with the gravity block, this block too is very simple. It merely implements the buoyancy forces and moments model derived in chapter

4. As before, no detailed explanation is warranted. In order to use this block the following must be provided: the centre of buoyancy; the displacement of the vehicle; the density of the operating medium; and the gravitational acceleration constant.

Thruster Model Block

This block implements the nonlinear thruster model derived in chapter 4. For the most part this is fairly simple. However, care should be taken to ensure that correct root is selected in all practical situations. In order to use this block, the following parameters must be supplied: the area of the propeller disk; the throttle position to power constant; and the power to static thrust constant.

8.1.3 Simulink Network Client

This network interface supplies the visualisation software with all necessary simulation data. Its implementation is based on the Aerosim toolbox's Flightgear interface. As was the case for the visualisation software, the UDP network protocol is used. The data packets sent by the client are by necessity the same as those received by the server. The definition for these packets are given by figure 7.11. In order to use this block, the following must be known: the IP address for the host running the visualisation display; the host port to connect to; and, finally, the sample time to use.

8.1.4 Pilot Interface

As was stated in section 1.2.2 a pilot input for manual flight must be provided. The pilot interface is developed around the joystick input supplied as part of the Aerosim block set.

With reference to figure 8.4, the pilot interface consists out of the following subblocks:

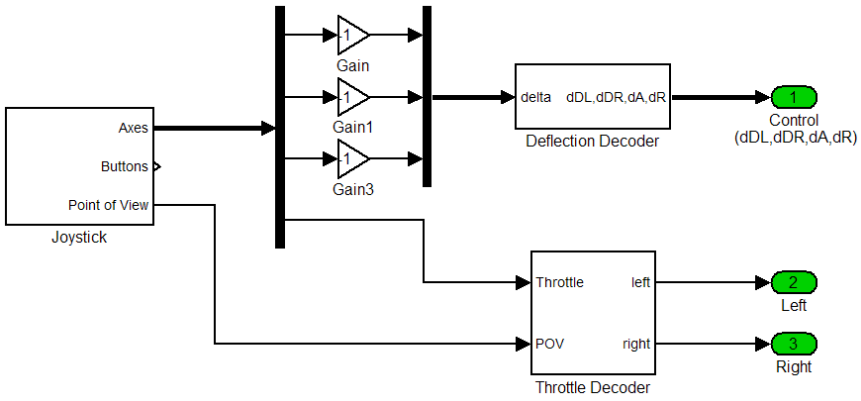

Figure 8.4: Pilot Interface Simulink Diagram

- **Joystick**: This block decodes the outputs supplied by the Aerosim joystick. It supplies the output in the same format as the MATLAB Aeroscope Toolbox Joystick.
- **Deflection Decoder**: This block is responsible for converting the pilot's roll,pitch and yaw commands into actual control surface deflection commands.
- **Throttle Decoder**: This block is used to combine the inputs from the joystick's single throttle and point of view switch into a left and right throttle opening. This block allows the pilot to use differential thrust control without the joystick having two throttles.

8.2 Examples

In order to better demonstrate the flexibility of the graphical simulation system two examples are shown, one aircraft example and one underwater vehicle example.

8.2.1 Underwater Vehicle Simulation Example

This example shows the visualisation software while rendering a fairly complex AUV scene. In this case the visualisation software is using the

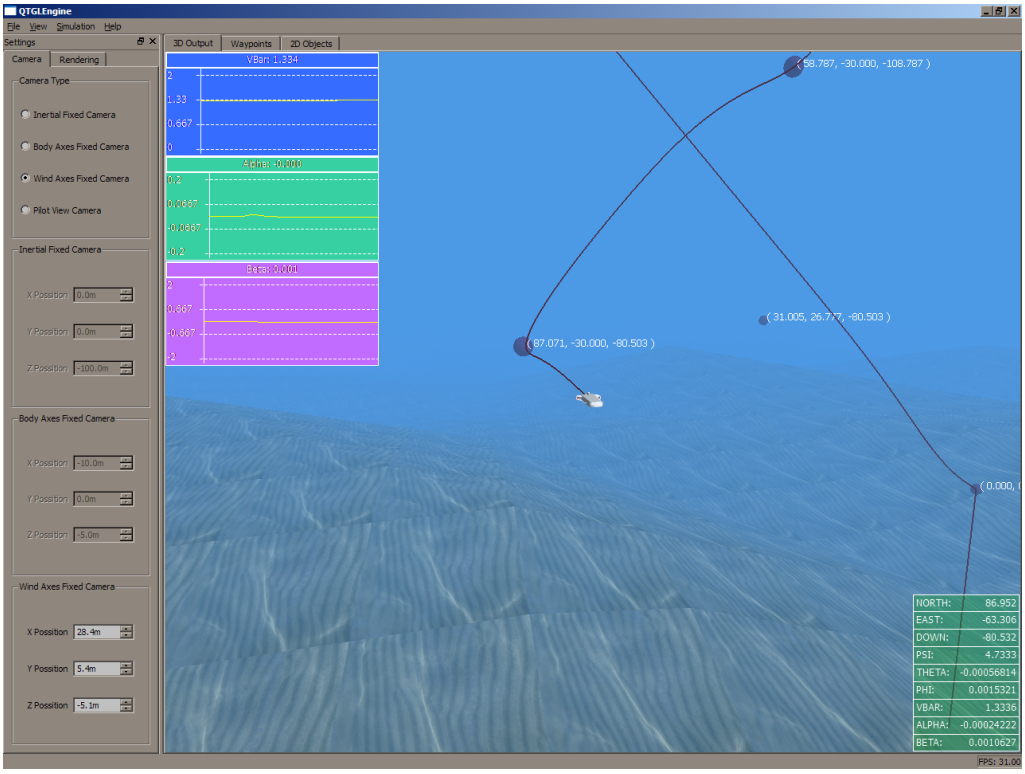

Figure 8.5: AUV Simulation Example

MATLAB back-end developed for this project with the addition of the autopilot being developed as a follow on project. For this example all available renderer features, including the fog system, are enabled.

With reference to figure 8.6, the following is visible: three different 2D graphs; the basic data object; a number of waypoints; vehicle tracking lines; the IMT AUV model; and a simple underwater seascape.

8.2.2 Aircraft Simulation Example

This example shows the visualisation software while rendering a simple UAV scene. In this case the visualisation software is not receiving data from the back-end developed for this project, but rather from an auto pilot system developed by other students in the ESL.

With reference to figure 8.6, the following is visible: three different 2D graphs; the basic data object; vehicle tracking lines; the UAV model; and

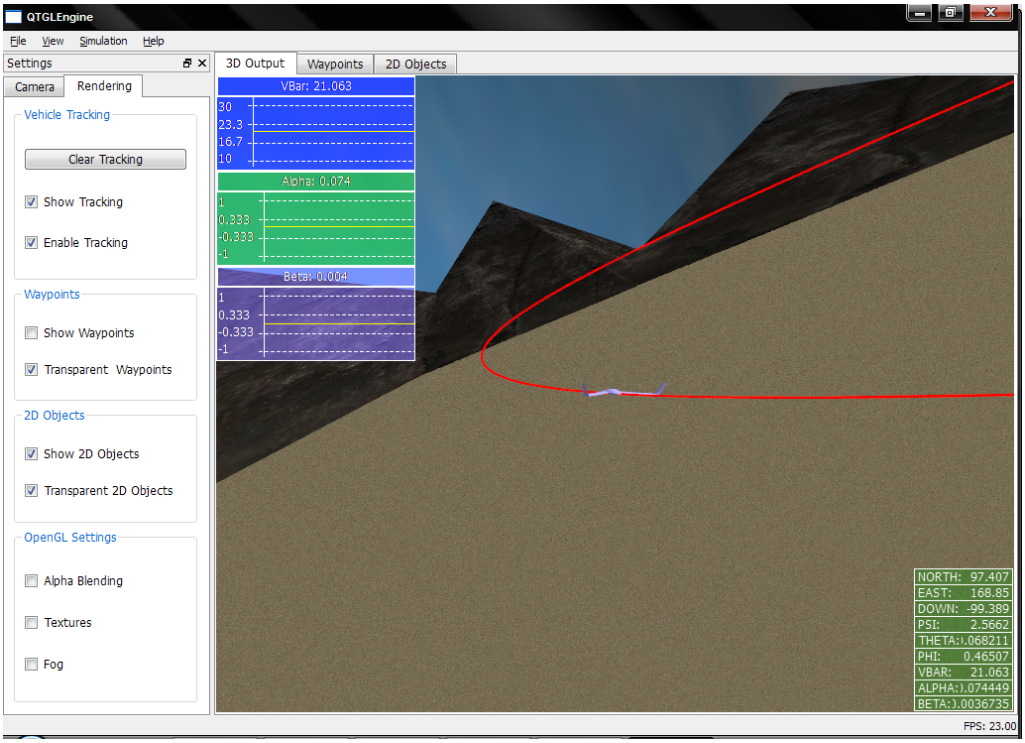

Figure 8.6: UAV Simulation Example

a simple desert landscape. Also note that the renderer setup used for this visualisation is shown in the figure.

Chapter 9

Conclusion and Recommendations

9.1 Summary

This thesis documents the development of a detailed mathematical model to describe the dynamics of the IMT AUV testbed. In addition to this the development of a user friendly graphical simulation system is also discussed. These two subprojects are now briefly summarised below.

9.1.1 Summary on Mathematical Modelling

The mathematical modelling is started by introducing a number of basic hydrodynamic concepts. After this, the actual modelling process commenced. This process was started by first deriving a set of nonlinear motion equation, followed by the development of nonlinear forces and moments models, and finally, the parameters specific to the IMT vehicle were estimated.

After the nonlinear modelling was complete, the model was linearised using small perturbation theory. This model was then analysed in order to provide insight into the vehicle's natural modes of motion.

9.1.2 Summary on Graphical Simulation System

The development of the graphical simulation system was discussed in two parts. Firstly, the development of a generic visualisation package was discussed in detail. Secondly a MATLAB based simulator was developed in order to supply the visualisation system with all the necessary simulation data. In order to demonstrate the flexibility of the system, the discussion was ended by showing two different simulation examples.

9.2 Conclusions on Mathematical Modelling

In the introduction to this thesis, it was pointed out that very little information was available on the IMT vehicle. Therefor, the parameters calculated during this project should be considered as estimates. This being said, it is however expected that these parameters should prove to be more than accurate enough from a control system design standpoint.

In addition to this, the model analysis shows that for the most part the vehicle dynamics are quite benign. This was expected since IMT implemented a very rudimentary control system on the vehicle.

The only real point of concern is the very lightly damped roll dynamics. This is unlikely to be correct, since IMT implemented no roll damping controller on the vehicle. The most likely source of this error lies with approximated geometry used for parameter estimation. This should however not substantially affect the performance of a well designed control system.

9.3 Conclusions on Graphical Simulation System

In general the goals set forth for the visualisation and simulation system was achieved, and in some instances surpassed. The system was designed with expandability and flexibility in mind. The focus does have a drawback however, in that some performance was sacrificed. Due

to the low cost of commodity computer hardware this compromise was deemed to be more than reasonable.

The system developed is so flexible that with no modification of the visualisation system it can even be used to replay real world AUV mission data.

Finally, due to the use of open source development tools throughout the development of the visualisation software, no expensive commercial development tools are required in order to modify the system.

9.4 Recommendations

Although the model and system developed during this project is entirely usable, the following should be investigated for future work:

- Calculate the hydrodynamic parameters using other empirical methods or using modern CFD software, if this becomes available.
- Investigate the use of a controllable ballets in order to improve vehicle performance.
- In order to increase the frame rate a more advanced renderer should be implemented.
- The use of other 3D file formats should be investigated, and if deemed useful, import filters for these should be implemented.
- More 3D objects should implemented. These can include planned flight path lines.

Appendices

Appendix A

Vehicle Data

Very little is known about the IMT AUV. This appendix summarises the little information which is available. In order to determine even the most basic data, measurements and photos were taken of the vehicle at IMT. Due to the method this data was collected, it should be considered as approximate.

A.1 Basic Vehicle Data

Parameter	Value
Dry Mass	360.5 kg
Overall Length	3 m
Overall With	1.5 m
Overall Hight	0.6 m
Volume	0.832 m^3
Flooded Mass	843.63 kg
Net Positive Buoyancy	981 N
Maximum Velocity	1.2 $m.s^{-1}$

Table A.1: Basic Vehicle Parameters

To calculate the flooded mass it is assumed that seawater has a density of 1026 $kg.m^{-3}$ and that the vehicle is 10 kg positively buoyant.

A.2 Geometric Data

Figures A.1, A.2, A.3, and A.4 shows the estimated geometry for the IMT AUV testbed.

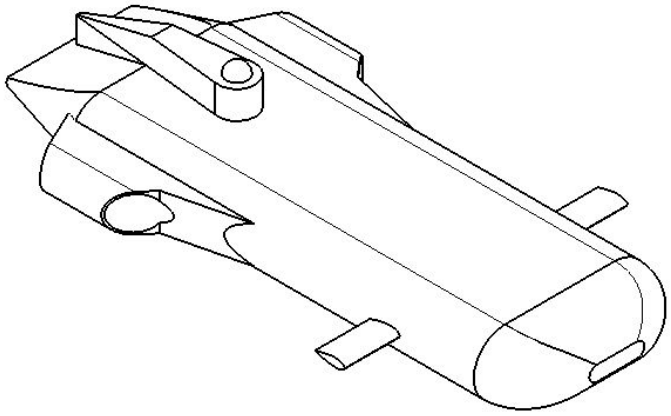

Figure A.1: IMT AUV (Isometric View)

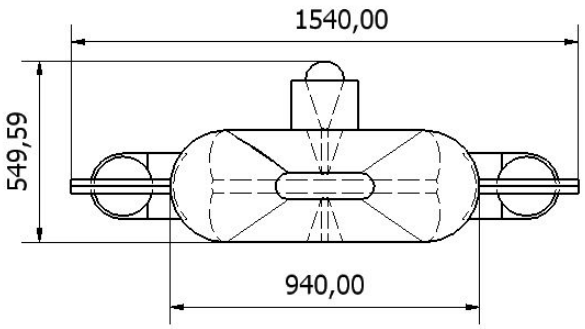

Figure A.2: IMT AUV (Front View)

A.3 Truster Data

The IMT vehicle is powered by two Magnetically Coupled DC Thruster units designed by Seaeye for use on ROVs. Each is capable of producing

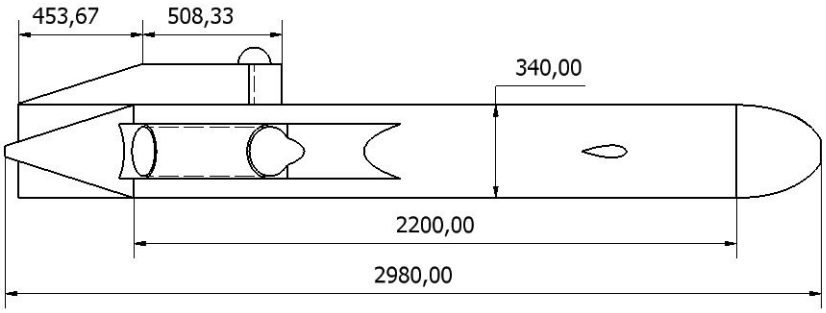

Figure A.3: IMT AUV (Side View)

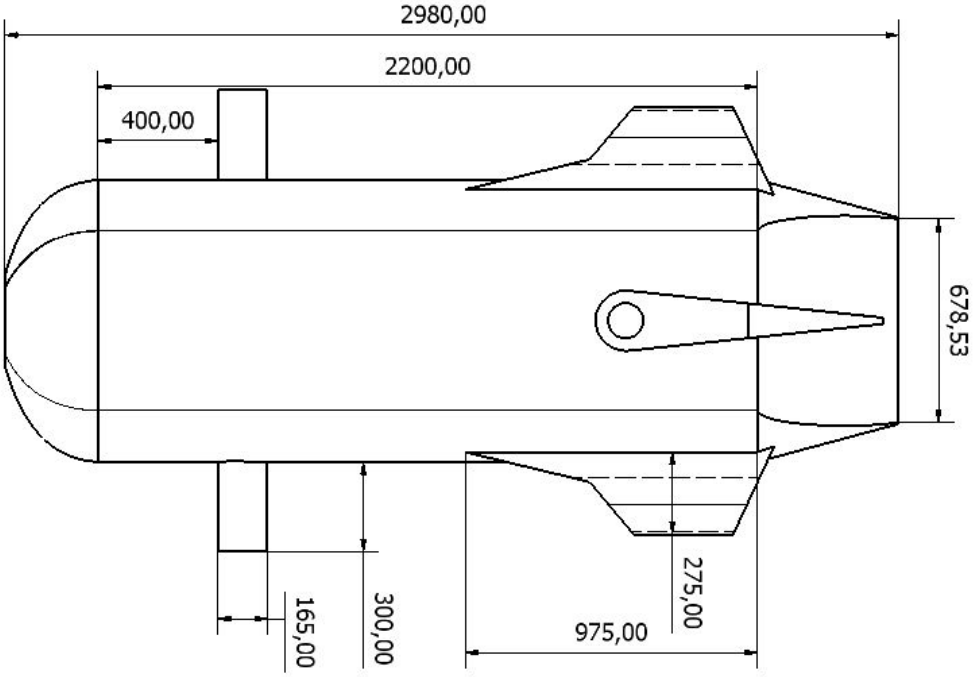

Figure A.4: IMT AUV (Top View)

13 kgf of thrust at 320W input. This results in a combined bollard pull of 26 kgf. Figure A.5 shows the power vs thrust curve for a single thruster.

Key thruster characteristics are summarised in table A.2.

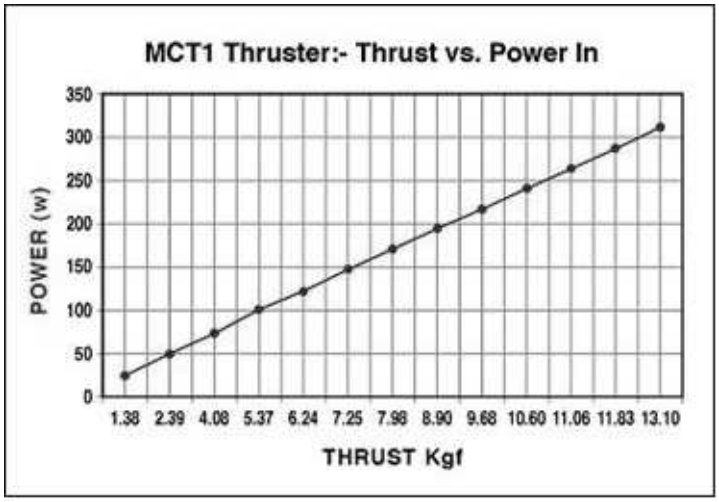

Figure A.5: Thrust vs Power[3]

Parameter	Value
Number of Thrusters	2
Total Bollard Pull	255 N
Total Input Power	640 W

Table A.2: Key Thruster Parameters

A.4 Hydrodynamic Actuator Data

The hydrodynamic actuators on the IMT AUV testbed all have a maximum deflection of $\pm 25^o$ and move at a rate of 14.3^o per second.

A.5 Onboard Electronics

The vehicle carries an impressive array of onboard electronics. These provide nanisation as well as extensive data logging.

A.5.1 Depth Measurement

An ENTRAN EXPO pressure sensor is used for depth measurement with an accuracy of 0.75%

A.5.2 Velocity Measurement

A Sontek ARGONAUT Doppler Velocity Log, DVL, supplies velocity measurements in all three axis at an accuracy of $0.077\,m.s^{-1}$. The DVL also supplies the distance to the seabed.

A.5.3 Inertial Navigation System

The IXSEA PHINS inertial navigation system is used. It supplies roll, pitch and yaw measurements. When used with GPS initialization and DVL assistance the INS has an positioning accuracy of $3m.h^{-1}$

A.5.4 Global Positioning System

CSI Wireless DGPS. The GPS only operates when the vehicle is on the surface and is used to initialize other sensors.

Appendix B

Detailed Mathematical Derivations

B.1 Kinetic Equations

B.1.1 Newton-Euler Formulation

Euler expresses Newton's second law in terms of conservation of both linear and angular momentum[2],

$$\boldsymbol{f}_c = \frac{d}{dt}(\boldsymbol{p}_c)\Big]_E = \frac{d}{dt}(m\boldsymbol{v}_c)\Big]_E \tag{B.1.1}$$

$$\boldsymbol{m}_c = \frac{d}{dt}(\boldsymbol{h}_c)\Big]_E = \frac{d}{dt}(I_c\boldsymbol{\omega})\Big]_E \tag{B.1.2}$$

where I_c is the body's inertial tensor about its centre of gravity, $\boldsymbol{f}_c$ and $\boldsymbol{m}_c$ are the forces and moments referred to the body's centre of gravity and ω is its angular velocity vector.

B.1.2 Momentum of a Particle

The momentum of a single particle is given by[7];

$$\boldsymbol{F} = \frac{d}{dt}(\boldsymbol{p})\Big]_E \tag{B.1.3}$$

$$= \frac{d}{dt}(m\boldsymbol{v})\Big]_E \tag{B.1.4}$$

A rigid body consists out of many of these particles. Each of these particles can be indexed. The total force on one of these particles is given by;

$$\boldsymbol{F}_{Bi} + \boldsymbol{R}_i = \frac{d}{dt}(m_i\boldsymbol{v}_i)\Big]_E \tag{B.1.5}$$

For a rigid body all internal forces must sum to zero, thus;

$$\boldsymbol{F}_B = \int_V \frac{d}{dt}\boldsymbol{v}\rho_A dV \tag{B.1.6}$$

B.1.3 Linear Momentum in a moving frame

The total velocity of a single particle in a rigid body is given by[7],

$$\boldsymbol{v}_i = \boldsymbol{v}_0 + \boldsymbol{\omega} \times \boldsymbol{r}_i \tag{B.1.7}$$

From equation B.1.6 and B.1.7,

$$\boldsymbol{F}_B = \int_V \frac{d}{dt}(\boldsymbol{v}_0 + \boldsymbol{\omega} \times \boldsymbol{r})\rho_A dV \tag{B.1.8}$$

Now using the following definitions;

$$m = \int_V \rho_A dV \tag{B.1.9}$$

$$m\boldsymbol{r}_G = \int_V \rho_A \boldsymbol{r} dV \tag{B.1.10}$$

The total force on the rigid body is given by;

$$\boldsymbol{F}_B = m\,\frac{d}{dt}(\boldsymbol{v}_0)\Big]_E + m\,\frac{d}{dt}(\boldsymbol{\omega}\times\boldsymbol{r}_g)\Big]_E \tag{B.1.11}$$

$$= m\left(\frac{d}{dt}(\boldsymbol{v}_0)\Big]_E + \frac{d}{dt}(\boldsymbol{\omega})\Big]_E \times \boldsymbol{r}_g + \frac{d}{dt}(\boldsymbol{r}_g)\Big]_E \times \boldsymbol{\omega}\right) \tag{B.1.12}$$

Since it would be convenient to express B.1.12 in terms of body axes, the equation of Coriolis[13] is used which relates the time derivative of an arbitrary vector to an inertial reference frame to the same vector's time derivative with respect to a body fixed reference frame. This is given by,

$$\frac{d}{dt}(\boldsymbol{R})\Big]_E = \frac{d}{dt}(\boldsymbol{R})\Big]_B + \omega\times\boldsymbol{R} \tag{B.1.13}$$

Using equation B.1.13, and noting that for a rigid body the following is valid,

$$\frac{d}{dt}(\boldsymbol{\omega})\Big]_E = \frac{d}{dt}(\boldsymbol{\omega})\Big]_B + \boldsymbol{\omega}\times\boldsymbol{\omega} = \frac{d}{dt}(\boldsymbol{\omega})\Big]_B \tag{B.1.14}$$

$$\frac{d}{dt}(\boldsymbol{r})_g\Big]_E = \frac{d}{dt}(\boldsymbol{r}_g)\Big]_B + \boldsymbol{\omega}\times\boldsymbol{r}_g = \boldsymbol{\omega}\times\boldsymbol{r}_g \tag{B.1.15}$$

The total force on a rigid body is given by,

$$\boldsymbol{F}_B = m\left(\frac{d}{dt}(\boldsymbol{v}_0)\Big]_B + \boldsymbol{\omega}\times\boldsymbol{v}_0 + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}_g) + \frac{d}{dt}(\boldsymbol{\omega})\Big]_B \times \boldsymbol{r}_g\right) \tag{B.1.16}$$

The equation above can be expressed in scalar form as,

$$X = m\left[\dot{U} + QW - RV + \dot{Q}z_g - \dot{R}y_g + (Qy_g + Rz_g)P - (Q^2 + R^2)x_g\right] \tag{B.1.17}$$

$$Y = m\left[\dot{V} + RU - PW + \dot{R}x_g - \dot{P}z_g + (Rz_g + Px_g)Q - (R^2 + P^2)y_g\right] \tag{B.1.18}$$

$$Z = m\left[\dot{w} + PV - QU + \dot{P}y_g - \dot{Q}x_g + (Px_g + Qy_g)R - (P^2 + Q^2)z_g\right] \tag{B.1.19}$$

B.1.4 Angular Momentum

According to Euler's second axiom, the external moment on a body is equal to the time rate of change of its angular momentum[2],

$$M_B = \frac{d}{dt}(h)\Big]_E \tag{B.1.20}$$

where h is the body's angular momentum.

The angular momentum for a particle is given by;

$$h_i = r_i \times p_i \tag{B.1.21}$$
$$= r_i \times m_i v_i \tag{B.1.22}$$

Define the inertial tensor I_B for a rigid body, as[2]

$$I_B = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{zx} & I_y & -I_{yz} \\ -I_{zy} & -I_{zy} & I_z \end{bmatrix} \tag{B.1.23}$$

where,

$$I_x = \int_V (y^2 + z^2)\rho_A dV \qquad I_{xy} = I_{yx} = \int_V xy\rho_A dV \tag{B.1.24}$$
$$I_y = \int_V (x^2 + z^2)\rho_A dV \qquad I_{xz} = I_{zx} = \int_V xz\rho_A dV \tag{B.1.25}$$
$$I_z = \int_V (x^2 + y2)\rho_A dV \qquad I_{zy} = I_{yz} = \int_V yz\rho_A dV \tag{B.1.26}$$

Alternatively I_B can be expressed in vectorial form as,

$$I_B\omega_B = \int_V r \times (\omega_B \times r)\rho_A dV \tag{B.1.27}$$

In a rigid body the sum of all internal forces must be zero, thus for a rigid body the total angular momentum is given by,

$$h = \int_V r \times v\rho_A dV \tag{B.1.28}$$

Differentiating equation B.1.28 with respect to time, yields,

$$\frac{d}{dt}(h)\Big]_E = \int_V r \times \frac{d}{dt}(v)\Big]_E \rho_A dV + \int_V \frac{d}{dt}(r)\Big]_E \times v\rho_A dV \tag{B.1.29}$$

$$= M_B + \int_V \frac{d}{dt}(r)\Big]_E \times v\rho_A dV \tag{B.1.30}$$

where the moment vector M_B is given by,

$$M_B = \int_V r \times \frac{d}{dt}(v)\Big]_E \rho_A dV \tag{B.1.31}$$

Now, noting that for a rigid body the following is valid,

$$\frac{d}{dt}(r)\Big]_E = v - v_0 \tag{B.1.32}$$

$$m\,\frac{d}{dt}(r_G)\Big]_E = \int_V \frac{d}{dt}(r)\Big]_E \rho_A dV = m(\omega_B \times r_G) \tag{B.1.33}$$

From equation B.1.30 , B.1.32 and B.1.33,

$$\frac{d}{dt}(h)\Big]_E = M_B - v_0 \times \int_V v\rho_A dV \tag{B.1.34}$$

$$= M_B - v_0 \times \int_V \frac{d}{dt}(r)\Big]_E \rho_A dV \tag{B.1.35}$$

$$= M_B - mv_o \times (\omega_B \times r_g) \tag{B.1.36}$$

The next step is to write another equation for $\dot{h}$ so that $\dot{h}$ can be eliminated from B.1.36,

$$h = \int_V r \times v\rho_A dV \tag{B.1.37}$$

$$= \int_V r \times v_0\rho_A dV + \int_V r \times (\omega \times r)\rho_A dV \tag{B.1.38}$$

$$= \left(\int_V r\rho_A dV\right) \times v_0 + I_B\omega \tag{B.1.39}$$

$$= mr_G \times v_0 + I_B\omega \tag{B.1.40}$$

Taking I_B to be constant and differentiating B.1.40 with respect to time, yields

$$\frac{d}{dt}(h)\Big]_E = I_B \frac{d}{dt}(\omega)\Big]_E + m\left(\frac{d}{dt}(r_G)\Big]_E \times v_0 + r_G \times \frac{d}{dt}(v_0)\Big]_E\right) \quad \text{(B.1.41)}$$

Now using B.1.15 and the equation of Coriolis,

$$\begin{aligned}\frac{d}{dt}(h)\Big]_E = I_B \frac{d}{dt}(\omega)\Big]_B + \omega \times (I_B\omega) + m(\omega \times r_G) \times v_0 \\ + mr_G \times \left(\frac{d}{dt}(v_0)\Big]_B + \omega \times v_0\right)\end{aligned} \quad \text{(B.1.42)}$$

Using the fact that

$$(\omega \times r_G) \times v_0 = -v_0 \times (\omega \times r_G) \quad \text{(B.1.43)}$$

And eliminating $\dot{h}$ from B.1.36, B.1.42,

$$M = I_B \frac{d}{dt}(\omega)\Big]_B + \omega \times (I_B\omega) + mr_G \times \left(\frac{d}{dt}(v_0)\Big]_B + \omega \times v_0\right) \quad \text{(B.1.44)}$$

Alternatively this equation can be written in scalar form as,

$$\begin{aligned}K = I_x\dot{P} + (I_z - I_y)QR - (\dot{R} + PQ)I_{xz} + (R^2 - Q^2)I_{yz} + (PR - \dot{Q})I_{xy} \\ + m\left[y_G(\dot{W} - UQ + VP) - z_G(\dot{V} - WP + UR)\right] \quad \text{(B.1.45)}\\ M = I_y\dot{Q} + (I_x - I_z)RP - (\dot{P} + QR)I_{xy} + (P^2 - R^2)I_{zx} + (QP - \dot{R})I_{yz} \\ + m\left[z_G(\dot{U} - VR + WQ) - x_G(\dot{W} - WQ + VP)\right] \quad \text{(B.1.46)}\\ N = I_z\dot{R} + (I_y - I_x)PQ - (\dot{Q} + RP)I_{yz} + (Q^2 - P^2)I_{xy} + (RQ - \dot{P})I_{zx} \\ + m\left[x_G(\dot{V} - WP + UR) - y_G(\dot{U} - VR + WQ)\right] \quad \text{(B.1.47)}\end{aligned}$$

B.2 Wind Axis Transformation

Deriving a transformation matrix to convert a body axis referenced vector to a wind axis referenced vector is a fairly straightforward process.

Unlike the three parameter Euler angle based transformation matrices derived earlier in this chapter, this transformation is only two parameter and thus the order of the transformations are arbitrary.

First consider a rotation about the Y_0 axis through an angle α. The transformation $V_0 \Rightarrow V_1$ is given by,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{B.2.1}$$

Now consider a rotation about the Z_1 axis through an angle β. The transformation $V_1 \Rightarrow V_2$ is given by,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{B.2.2}$$

Thus, the transformation $V_0 \Rightarrow V_2$ is given by,

$$\boldsymbol{V_2} = \left[\boldsymbol{T_\beta T_\alpha}\right] \boldsymbol{V_0} \tag{B.2.3}$$

or, alternatively the transformation $V_B \Rightarrow V_W$ is given by,

$$\boldsymbol{V_W} = \left[\boldsymbol{T_\beta T_\alpha}\right] \boldsymbol{V_B} \tag{B.2.4}$$

Finally the complete transformation is given by,

$$\begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{B.2.5}$$

Since B.2.5 is orthogonal the transformation from wind to body is given

by,

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} \tag{B.2.6}$$

B.3 Converting Between Body Axis Reference and Stability Axis Reference Derivatives

B.3.1 Converting Added Mass Derivatives

Since the model derived in chapter 4 are in terms of Wind axis referenced derivatives and the derivatives supplied by strip theory are in terms of body axis referenced derivative some transformation between the two representations are required.

The forces and moments transformations are considered separately. First consider the force transformation. The force due to acceleration referred to the Wind axis system can be expressed as,

$$\begin{bmatrix} D \\ Y \end{bmatrix} = \begin{bmatrix} D_{\dot{V}} & D_{\dot{\beta}} & D_{\dot{\alpha}} \\ Y_{\dot{V}} & Y_{\dot{\beta}} & Y_{\dot{\alpha}} \\ L_{\dot{V}} & L_{\dot{\beta}} & L_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} \tag{B.3.1}$$

or,

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = \begin{bmatrix} -D_{\dot{V}} & -D_{\dot{\beta}} & -D_{\dot{\alpha}} \\ Y_{\dot{V}} & Y_{\dot{\beta}} & Y_{\dot{\alpha}} \\ -L_{\dot{V}} & -L_{\dot{\beta}} & -L_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} \tag{B.3.2}$$

And, the force due to acceleration referred to the Body axis system is

given by,

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & Z_{\dot{w}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{B.3.3}$$

From equations B.2.5, B.3.2 and B.3.3.

$$\begin{bmatrix} -D_{\dot{V}} & -D_{\dot{\beta}} & -D_{\dot{\alpha}} \\ Y_{\dot{V}} & Y_{\dot{\beta}} & Y_{\dot{\alpha}} \\ -L_{\dot{V}} & -L_{\dot{\beta}} & -L_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & Z_{\dot{w}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{B.3.4}$$

Assuming that the rate of change in U is far smaller than the rate of change in α and β, the following approximated relationship between the acceleration in rectangular and polar coordinates can be obtained from equation 3.1.12.

$$\begin{bmatrix} \dot{V} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{U} & 0 \\ 0 & 0 & \frac{1}{U} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{B.3.5}$$

Substituting B.3.5 into B.3.4 gives,

$$\begin{bmatrix} -D_{\dot{V}} & -D_{\dot{\beta}} & -D_{\dot{\alpha}} \\ Y_{\dot{V}} & Y_{\dot{\beta}} & Y_{\dot{\alpha}} \\ -L_{\dot{V}} & -L_{\dot{\beta}} & -L_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{U} & 0 \\ 0 & 0 & \frac{1}{U} \end{bmatrix} \begin{bmatrix} \dot{u}_W \\ \dot{v}_W \\ \dot{w}_W \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & Z_{\dot{w}} \end{bmatrix} \begin{bmatrix} \dot{u}_B \\ \dot{v}_B \\ \dot{w}_B \end{bmatrix} \tag{B.3.6}$$

Now the transformation of a vector from body to wind reference is

given by B.2.6. Substituting this into B.3.6 gives.

$$\begin{bmatrix} -D_{\dot{V}} & -D_{\dot{\beta}} & -D_{\dot{\alpha}} \\ Y_{\dot{V}} & Y_{\dot{\beta}} & Y_{\dot{\alpha}} \\ -L_{\dot{V}} & -L_{\dot{\beta}} & -L_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{U} & 0 \\ 0 & 0 & \frac{1}{U} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$
$$\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & Z_{\dot{w}} \end{bmatrix} \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{B.3.7}$$

Now, multiplying out this matrix equation and ignoring the off diagonal components give,

$$D_{\dot{V}_W} = (\cos\alpha)^2(\cos\beta)^2 X_{\dot{u}} + (\sin\beta)^2 Y_{\dot{v}} + (\sin\alpha)^2(\cos\beta)^2 Z_{\dot{w}} \tag{B.3.8}$$
$$Y_{\dot{\beta}_W} = U\left[(\cos\alpha)^2(\sin\beta)^2 X_{\dot{u}} + (\cos\beta)^2 Y_{\dot{v}} + (\sin\alpha)^2(\sin\beta)^2 Z_{\dot{w}}\right] \tag{B.3.9}$$
$$L_{\dot{\alpha}_W} = U\left[(\sin\alpha)^2 X_{\dot{u}} + (\cos\alpha)^2 Z_{\dot{w}}\right] \tag{B.3.10}$$

The moment due to angular acceleration referred to the Wind axis system can be expressed as,

$$\begin{bmatrix} K_W \\ M_W \\ N_W \end{bmatrix} = \begin{bmatrix} K_{\dot{p}_W} & K_{\dot{q}_W} & K_{\dot{r}_W} \\ M_{\dot{p}_W} & M_{\dot{q}_W} & M_{\dot{r}_W} \\ N_{\dot{p}_W} & N_{\dot{q}_W} & N_{\dot{r}_W} \end{bmatrix} \begin{bmatrix} \dot{p}_W \\ \dot{q}_W \\ \dot{r}_W \end{bmatrix} \tag{B.3.11}$$

And, the moment due to angular acceleration referred to the Body axis system is given by,

$$\begin{bmatrix} K_B \\ M_B \\ N_B \end{bmatrix} = \begin{bmatrix} K_{\dot{p}_B} & 0 & 0 \\ 0 & M_{\dot{q}_B} & 0 \\ 0 & 0 & N_{\dot{r}_B} \end{bmatrix} \begin{bmatrix} \dot{p}_B \\ \dot{q}_B \\ \dot{r}_B \end{bmatrix} \tag{B.3.12}$$

From equations B.2.5, B.3.11 and B.3.12.

$$\begin{bmatrix} K_{\dot{p}_W} & K_{\dot{q}_W} & K_{\dot{r}_W} \\ M_{\dot{p}_W} & M_{\dot{q}_W} & M_{\dot{r}_W} \\ N_{\dot{p}_W} & N_{\dot{q}_W} & N_{\dot{r}_W} \end{bmatrix} \begin{bmatrix} \dot{p}_W \\ \dot{q}_W \\ \dot{r}_W \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$
$$\begin{bmatrix} K_{\dot{p}_B} & 0 & 0 \\ 0 & M_{\dot{q}_B} & 0 \\ 0 & 0 & N_{\dot{r}_B} \end{bmatrix} \begin{bmatrix} \dot{p}_B \\ \dot{q}_B \\ \dot{r}_B \end{bmatrix} \tag{B.3.13}$$

Now since, according to Cook[14], the angular velocity can be treated as a vector and the angular acceleration is merely its time derivative it too can be treated as a vector. Now considering this statement equation B.2.6 can be used to convert the body referenced angular acceleration vector to the equivalent wind axis referenced angular acceleration vector.

$$\begin{bmatrix} K_{\dot{p}_W} & K_{\dot{q}_W} & K_{\dot{r}_W} \\ M_{\dot{p}_W} & M_{\dot{q}_W} & M_{\dot{r}_W} \\ N_{\dot{p}_W} & N_{\dot{q}_W} & N_{\dot{r}_W} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$
$$\begin{bmatrix} K_{\dot{p}_B} & 0 & 0 \\ 0 & M_{\dot{q}_B} & 0 \\ 0 & 0 & N_{\dot{r}_B} \end{bmatrix} \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{B.3.14}$$

Finally, multiplying out this matrix equation and ignoring the off diagonal components give,

$$K_{\dot{p}_W} = (\cos\alpha)^2(\cos\beta)^2 K_{\dot{p}_B} + (\sin\beta)^2 M_{\dot{q}_B} + (\sin\alpha)^2(\cos\beta)^2 N_{\dot{r}_B} \tag{B.3.15}$$

$$M_{\dot{q}_W} = (\cos\alpha)^2(\sin\beta)^2 K_{\dot{p}_B} + (\cos\beta)^2 M_{\dot{q}_B} + (\sin\alpha)^2(\sin\beta)^2 N_{\dot{r}_B} \tag{B.3.16}$$

$$N_{\dot{r}_W} = (\sin\alpha)^2 K_{\dot{p}_B} + (\cos\alpha)^2 N_{\dot{r}_B} \tag{B.3.17}$$

Bibliography

[1] Willem J. Hough, "Autonomous aerobatic flight of a fixed wing unmanned aerial vehicle", Master's thesis, University of Stellenbosch, March 2007.

[2] Thor I. Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley and Sons, 1994.

[3] Jacob Venter, "Email correspondence", Institute for Marine Technology, 2008.

[4] M.C. Silberbauer, "Simulation visualisation systems", Tech. Rep., University of Stellenbosch, 2005.

[5] J. N. Newman, *Marine Hydrodynamics*, The MIT Press, 1980.

[6] "Marine hydrodynamics 2.20 lecture notes", Department of Ocean Engineering, Massachusetts Institute of Technology, 2005.

[7] Michael S. Triantafyllou and Franz S. Hover, *Manoeuvring and Control of Marine Vehicles*, Department of Ocean Engineering, Massachusetts Institute of Technology, Nov. 2003.

[8] Claude C. Leroy and Francois Parthiot, "Depth-pressure relationships in the oceans and seas", *J. Acoust. Soc Am.*, vol. 103, no. 3, pp. 1346–1352, March 1998.

[9] "Department of oceanography, dalhousie university", http://www.phys.ocean.dal.ca/ kelley/seawater/density.html.

[10] Kenneth V. Mackenzie, "Nine-term equation for sound speed in the oceans", *J. Acoust. Soc Am.*, vol. 70, no. 3, pp. 807–812, Sept 1981.

[11] R. A. Robinson, "The vapour pressure and osmotic equivalence of sea water", *J. Mar. biol. Ass. U.K.*, vol. 33, pp. 449–455, 1954.

[12] Iain K. Peddle, *Acceleration Based Manoeuvre Flight Control System for Unmanned Aerial Vehicles*, PhD thesis, University of Stellenbosch, December 2008.

[13] Iain K. Peddle, "Autonomous flight of a model aircraft", Master's thesis, University of Stellenbosch, April 2005.

[14] M. V. Cook, *Flight Dynamics Principles*, Butterworth Heinemann, 1997.

[15] D.A. Jones, D.B. Clarke, I.B. Brayshaw, J.L. Barillon, and B. Anderson, *The Calculation of Hydrodynamic Coefficients for Underwater Vehicles*, DSTO Platforms Sciences Laboratory, Jul. 2002.

[16] Barnes W. McCronmick, Jr, *Aerodynamics of V/STOL Flight*, Academic Press, 1967.

[17] M. Nahon, "Determination of undersea vehicle hydrodynamic derivatives using the usaf datcom", *OCEANS '93. 'Engineering in Harmony with Ocean'. Proceedings*, vol. 2, pp. 283–288, Oct. 1993.

[18] McDonnell Douglas Astronautics Company, *The USAF Stability and Control DATCOM*, 1979, Vol 1 Users Manual.

[19] Horace Lamb, Sir, *Hydrodynamics*, Cambridge University Press, 6 edition, 1932.

[20] Frederick H. Imlay, "The complete expression for added mass of a rigid body moving in an ideal fluid", Tech. Rep., David Taylor Model Basin, Jul. 1961.

[21] D. E. Humphreys and K. W. Watkinson, "Prediction of acceleration hydrodynamic coefficients for underwater vehicles from geometric parameters", Tech. Rep., Naval Coastal Systems Laboratory, Feb. 1978.

[22] Bruce Eckel and Chuck Allison, *Thinking In C++*, vol. 2: Practical Programming, Pearson Prentice Hall, 2004.

[23] "Sgi website", http://www.sgi.com/, 2008.

[24] "lib3ds website", http://www.lib3ds.org/, 2008.

[25] Thomas R. Yechout, *Introduction to Aircraft Flight Mechanics*, AIAA Education Series, 2003.

[26] Trolltech, *Qt Reference Documentation*, 2008.

[27] James Diebel, "Representing attitude; euler angles, unit quaternions, and rotation vectors", Oct. 2006, Stanford University.

[28] Murray R. Spiegel and John Liu, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 1999.

[29] Bernard Etkin and Lloyd Duff Reid, *Dynamics of Flight: Stability and Control*, John Wiley and Sons, third edition, 1996.

[30] M. C. Koen, "Modelling and simulation of an rpv for flight control system purposes", University of Pretoria.

[31] Mason Woo, Jackie Neider, and Tom Davis, *OpenGL Programming Guide (Redbook)*, Addison Wesley, second edition, Apr. 1997.

[32] Jasmin Blanchette and Mark Summerfield, *C++ GUI Programming with Qt 4*, Prentice Hall, Jun. 2006.