SYSTEM IDENTIFICATION AND CONTROL OF A SEA SURFACE VEHICLE

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

BY

İZZET KAĞAN ERÜNSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

SYSTEM IDENTIFICATION AND CONTROL OF A SEA SURFACE VEHICLE

submitted by **İZZET KAĞAN ERÜNSAL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _______________

Prof. Dr. M. Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Eng. Dept., METU** _______________

Examining Committee Members:

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU _______________

Prof. Dr. M. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU _______________

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Department, METU _______________

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU _______________

Assoc. Prof. Dr. Klaus Werner Schmidt
Mechatronics Engineering Department, Çankaya University _______________

Date: **08/09/2015**

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

Name, Last Name: İZZET KAĞAN ERÜNSAL

Signature :

ABSTRACT

SYSTEM IDENTIFICATION AND CONTROL OF A SEA SURFACE VEHICLE

Erünsal, İzzet Kağan
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. M. Kemal Leblebicioğlu

September 2015, 120 pages

In this study, modeling, system identification and controller design for a sea surface vehicle are performed to facilitate major goals such as autopilot design and guidance. The vehicle has been modeled by a combination of several approaches proposed in the literature. For system identification purposes, a time domain, offline, deterministic, gray-box methodology is developed. In this methodology, efficient optimization algorithms such as genetic algorithm (GA) and global search algorithm (GS) are utilized together for successful parameter identification. Finally, both a PID based piecewise controller and a Sliding Mode Controller (SMC) have been developed to control surge speed and yaw position of the vehicle. Performance comparisons of these controllers are also provided in both existence and absence of external disturbances. The model, the system identification methodology and the control structure have all been verified and validated through simulations. The developed simulation environment consists of six degrees of freedom nonlinear mathematical model of the vehicle derived using Newton-Euler equations. The experimental test bed used for the model validation and the system identification, on the other hand, is inclusive of a model vehicle and auxiliary hardware such as a fully equipped autopilot card and an external computer acting as a master on that.

Keywords: Unmanned sea surface vehicle, System identification, PID, Sliding mode controller (SMC), Genetic algorithm, Global search

ÖZ

BİR SU ÜSTÜ ARACININ SİSTEM TANILAMA VE KONTROLÜ

Erünsal, İzzet Kağan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. M. Kemal Leblebicioğlu

Eylül 2015 , 120 sayfa

Bu çalışmada bir su üstü aracının modelleme, sistem tanılama ve denetleç tasarımı; otopilot tasarımı ve güdüm amacı göz önünde bulundurularak gerçekleştirilmiştir. Araç, literatürde önerilen birkaç modelleme yönteminin birleşimi ile modellenmiştir. Sistem tanılama amacıyla, zaman alanı, çevrimdışı, gerekirci, gri kutu tanılama yöntemi seçilmiş ve geliştirilmiştir. Bu yöntemde Genetik algoritma (GA) ve Global Arama (GS) algoritması gibi verimli eniyileme algoritmaları kullanılmıştır. Son olarak aracın ileri hızı ve dönü açısını kontrol etmek için PID tabanlı parçalı denetleç ve Kayan Kipli Denetleç (SMC) tasarımları gerçekleştirilmiştir. Denetleçlerin performansı bozucu etki olmadan ve bozucu etki altında karşılaştırılmıştır. Modelleme, sistem tanılama ve kontrolcü yapısı yapılan simülasyonları ile doğrulanmıştır. Geliştirilen simülasyon ortamı altı serbestlik dereceli, doğrusal olmayan matematiksel bir modeldir ve Newton-Euler yaklaşımı ile oluşturulmuştur. Model doğrulama ve sistem tanılama için kullanılan deney düzeneği ise bir model su üstü aracı, tam fonksiyonlu otopilot kartı ve harici yönetici bilgisayar gibi yardımcı donanımlardan oluşmaktadır.

Anahtar Kelimeler: İnsansız su üstü aracı, Sistem tanılama, PID, Kayan kipli denetleç, Genetik algoritma, Global arama

To my family...

"The most beautiful people we have known are those who have known defeat, known suffering, known struggle, known loss and have found their way out of depths. These people have an appreciation, a sensitivity and an understanding of life that fills them with compassion, gentleness and a deep loving concern. Beautiful people do no just happen."

Elisabeth Kübler Ross

ACKNOWLEDGMENTS

I would like to express my sincerest thanks to my supervisor Prof. Dr. M. Kemal Leblebicioğlu for his guidance, support and valuable contributions throughout the preparation of this thesis.

I would like to express my appreciations to Murat Kumru for his support, significant contributions and close friendship throughout the thesis.

I would like to present my appreciations to Kenan Ahıska for his supportive guidance in all stages of the thesis.

I would like to express my thanks to Onur Cem Erdoğan for his valuable supports in experimental setup.

I would like to thank Mehmet Cihan, Serdar Görür and Kadir Emlik for their creative technical support.

I would like to present my thanks to Cumhur Çakmak for his unique contributions in the experiments conducted and the presentation of the thesis.

I would like to express my appreciations to Mustafa Burak Gürcan for his insightful attitudes during the study.

I would like to acknowledge the support of ASELSAN Inc. for realization of this thesis.

I would like to present my appreciations to Turkish Scientific and Technological Research Council (TUBITAK) for the support during my thesis study.

The last but not the least, I express my sincerest thanks to my family and my friends who have given me encourage and support to continue.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DOF	Degree of Freedom
GPS	Global Positioning System
NED	North - East - Down
ESC	Electronic Speed Controller
PWM	Pulse Width Modulation
LiPo	Lithium Polymer
RTOS	Real Time Operating System
DSP	Digital Signal Processor
FPU	Floating Point Unit
MAVLink	Micro Air Vehicle Link
API	Application Programming Interface
PID	Proportional - Integral - Derivative
SMC	Sliding Mode Controller
MIMO	Multi-Input, Multi Output
SISO	Single-Input, Single Output
ERA	Eigenvalue Realization Algorithm
EKF	Extended Kalman Filter
LQG	Linear Quadratic Gaussian
RMS	Root Mean Square

CHAPTER 1

INTRODUCTION

1.1 Motivation of the Thesis

Rapid progression of technology leads to various improvements for unmanned sea surface vehicles (USSV). Among these improvements, the most obvious one is the increasing level of autonomy. With this ability and suitable equipment, USSVs carry out large number of tasks. Considering military field, these vehicles are used for safeguarding operations in littoral waters in which continuous presence and surveillance is a must, as satellites for main combat ships, navigating through unknown fields that may endanger human life and expensive assets. Military focuses on developing low cost USSV with high reliability and ease of use. They aim to obtain high speed USSVs with flexible and configurable payload capacities to be able to carry out multiple tasks abroad mother ship. Future objective of military is to perform coordination and collaboration of multiple unmanned vehicles to perform specific missions. Taking into account this objective, USSVs has a vital role in becoming a bridge between above-water and under-water vehicles [12]. On the other hand, research institutes design their own USSVs in order to help other fields such as autonomous environmental monitoring, gathering and delivering samples from sea, cleaning up ocean contaminants etc. [13]. Another aim of the research institutes in designing these vehicles is improving capabilities of USSVs such as new control, navigation and guidance techniques etc. Furthermore, increasing robustness is another key task in order to increase independence of the vehicle from user [12].

Considering these applications and the future aims, the motivation of this thesis is to establish an infrastructure for USSVs in order to realize more complex future tasks. In this content, it is decided that modeling, system identification and control concepts should be focused. In the modeling work, considering different approaches, a six degree of freedom nonlinear mathematical model of a sea surface vehicle will be obtained. Then, in order to design reliable controllers, the system identification of the model will be carried out in the light of different approaches and with the help of various experiments. In final step, different controller structures will be designed to control the motion of the vehicle in aimed forward speed and heading angle. As future studies, application of different system identification techniques, validation of the controller on the experimental setup and coordinated guidance of multiple USSVs can be studied.

1.2 Literature Survey

The research of ship dynamics has been separated into two fundamental areas: Maneuvering and Seakeeping. Maneuvering studies consider ship motion in horizontal plane (surge-sway-yaw) in the absence of wave disturbance. Objective of these studies is to evaluate capabilities of the ship under the action of control devices. The models obtained in this study is utilized in ship simulators, stability analysis and tracking and course keeping autopilots. Either three or four DOF models are constructed for this study. On the other hand, Seakeeping deals with the motion when the disturbances exist and the vehicle keeps its speed and course constant. The models constructed using the seakeeping theory are used to evaluate performance indices related to wave excitation. Generally six DOF models are prepared in this concept. When two areas combined, the area is called "maneuvering in seaway" [2] [1]. There is another classification of the ship dynamics and it is based on Froude Number which is presented in Equation (1.1)

$$Fn = \frac{U}{\sqrt{Lg}} \tag{1.1}$$

where L is the length of the vessel, g is the gravitational constant and U is the speed of the vessel. Based on this number, for $Fn < 0.3 - 0.4$, hydrostatic pressure becomes dominant and the vessel operating in this regime is named as "displacement" vessel. For $0.3 - 0.4 < Fn < 1 - 1.2$, lift effects start to appear and hydrodynamic forces become dominant, then it is said that the vessel is operating in "semi-displacement" regime. For $Fn > 1 - 1.2$, flow separation is strong and aerodynamics becomes also important. Vessels operating in this regime are called "planning" vessels [2]. All of these classifications are illustrated in Figure 1.1.

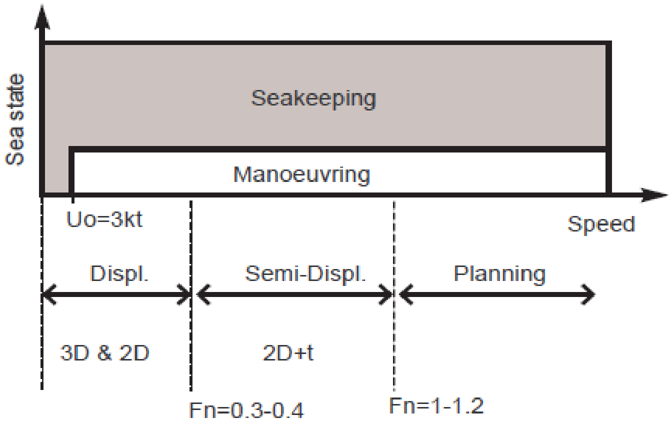


Figure 1.1: Classification of ship dynamics [2]

In this study, mostly, the maneuvering theory for either the displacement or the semi-displacement vessels are considered, however, since disturbance rejection performance is investigated, the seakeeping theory is also touched.

An extensive literature on mathematical modeling for a sea surface vehicle can be found in [3], [14] and [1]. First approach for the mathematical modeling in naval architecture is motivated by Newton's law $F = ma$, where F stands for the force, m is the mass of the body and a represents the acceleration. This classical representation is given in Equation (1.2).

$$\boldsymbol{M}\dot{\boldsymbol{v}} = \sum_{i=1}^{n} \boldsymbol{F_i} \tag{1.2}$$

where $\boldsymbol{M}$ is the system inertia matrix, $\boldsymbol{v}$ is the generalized state vector as in Equation (1.3).

$$\boldsymbol{v} = [u, v, w, p, q, r]^T \tag{1.3}$$

where u, v, w are the linear velocities in surge, sway and heave, p, q, r are the angular rates in roll, pitch and yaw. Furthermore, $\boldsymbol{F_i}$ is the vector of linearly superposed forces and moments.

If it is assumed that $n = 1$ and the linear theory is valid, there are 36 unknown terms in this equation. Considering that nonlinear terms can also be included, hundreds of terms are added to these 36 terms. It is very difficult to perform model based control with that large number of terms. Hence, it is very advantageous to reduce the number of coefficients by exploiting the physical system properties. Based on this motivation, a vectorial model is created by Fossen [14] and this model has very similar features with the robot model suggested by [15].The robot model is given in Equation (1.4).

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q} = \boldsymbol{\tau} \tag{1.4}$$

where $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}$ is the Coriolis matrix, $\boldsymbol{\tau}$ is the torques and $\boldsymbol{q}$ is the joint angles vector. Further developed models are utilized in some studies such as [3], [16] and [17]. Final form is reached as in Equation (1.5).

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{1.5}$$

where, additionally, $\boldsymbol{D}$ is the damping matrix, $\boldsymbol{g}$ is the vector of buoyancy and gravitational forces, $\boldsymbol{v}$ and $\boldsymbol{\eta}$ are the states explained in Equations (1.6) and (1.7).

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{1.6}$$

$$\boldsymbol{v} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{1.7}$$

When designing control systems, it is recommended to use the vectorial model given in Equation (1.5). The reason behind this recommendation is that by utilization of this representation, system properties such as symmetry, skew symmetry and positive definiteness can be investigated in stability analysis. Actually, when the equations are written in component form, both the classical model and the Fossen's robot-like vectorial approach is equivalent, however, for example, MIMO controllers can be easily designed with the latter.The Fossen's model has been also accepted by international community as a standard model for marine vessels for control purposes [14]. As a result, this model is taken as a basis for this study, however, necessary modifications and simplifications are also performed.

System identification is a sufficiently mature field [18]. There are some major textbooks in the literature such as [19], [20] and [21] and they include basic and more complex system identification methodologies developed. System identification practice in the literature has been mostly carried out in the form of parameter identification [18]. Although general approach is to minimize the error between true and predicted state histories with least squares regression, there are other methods such as instrumental value method [22] and Eigenvalue Realization Algorithm (ERA) [20] etc. For the first time, Nomoto [23] formulated the identification problem with application to the ship dynamics. In this study, the problem was simplified such that no computer technology was required. Moreover, the system was assumed to be linear and the model equations have been derived such that minimum number of unknown parameters were reached and they could be easily identified after a zigzag test. Although this model could be utilized in certain applications such as short term predictions, it was not convenient for realistic simulations. The controllability and the observability characteristics for the first and second order Nomoto model have been investigated by Tzeng and Chen [24] and the parameter identification for the steering dynamics of the ship based on same models were discussed by Journee [25]. Brinati and Neto [26] have utilized the Extended Kalman Filter (EKF) in order to apply the system identification methodology to the ship dynamics. Although this method required long time periods to converge and lots of data to collect, its popularity continued and let to the following studies utilizing the same approach such as [27]. In Casado's study [28],the adaptive back-stepping was applied to a highly simplified ship model, however, there was no real result related to effectiveness of the method. By eliminating inconvenient terms in the model, Abkowitz [29] has been constructed a model, however, it had limited applicability as certain regression terms may become effective in special conditions. Silman [30] suggested two efficient deterministic optimization algorithms. First one was Gauss-Newton method combined with functional regularization and multi criterial optimization technique and the other method was quasi random global search with a number of constraints. Both methods were successful in practical identification, however, it required trial-error and inputs from an experienced user. Di Mascio et al. [31] proposed a method using Genetic Algorithm (GA). The performance of this method showed superiority over the EKF based approach, however, it suffered from cancellation effects. Yoon and Rhee [32] used alternatively the EKF and the Modified Bryson-Frazier Smoother (MBSF) to restore time histories for the hydrodynamic forces. In this study, it was concluded that combination of a 20 - 20 degrees zigzag with 35 degrees turn was not rich enough to guarantee reliable estimation. Perera et al. [33] utilized the stochastic parameter approach for the nonlinear ocean vessel steering model with an Extended Kalman Filter. This study concluded that the sudden maneuvering conditions of the vehicle should be implemented in order to determine the nonlinear parameters accurately. Furthermore, frequency domain approaches were also applied to identify the parameters of the surface vessels by Selvam [34], however, since linear models were utilized in that study and in similar frequency domain studies, they are not investigated in details. Tran et al. [35] have introduced an identification procedure to estimate the hydrodynamic parameters for a range of ships. This procedure used the dynamic ship motion model with mathematical programming techniques. In this study, the proposed methodology was validated through the turning and zigzag tests and it was seen that there were improvements in the root-mean square deviations of the ship trajectory and the heading angle. Finally, Kariotoglou [36] has proposed a methodology by using a planar motion mechanism to identify the pitch, roll and yaw damping parameters and the linear drag parameter. Although promising results were obtained, there were no validation tests for this study.

Controller design for sea surface vehicles was firstly introduced with a PID control [37]. In this study, a single-input-single-output (SISO) control scheme was utilized in order to control the heading angle of the vehicle. The feedback signal in this study was the yaw position coming from a gyro-compass. In different studies, nonlinear PIDs and passive observer designs have been proposed for the control of sea surface vehicles [3], [38]. Do et al. has introduced a robust adaptive controller for under actuated ships [39]. In this study, surface vehicle tracks a path in a commanded speed in the existence of disturbances.The Lyapunov's direct method was utilized in this study. Reyhanoğlu [40] has studied on the stabilization of the autonomous surface vehicles with independent thrusters. The stabilized system had exponential convergence rates with discontinuous feedback. Bao-li [41] has studied on nonlinear controllers. The reduced order model was utilized and the robust control law with k-exponential stability was reached. Liao et al. [42] has suggested an adaptive dynamic sliding mode controller. This methodology was only for under-actuated surface vehicles. The method has showed global asymptotic stability. In their study, Naeem et al. [43] has used a Linear Quadratic Gaussian (LQG) controller with Fuzzy Logic

Control in order to carry out navigation and guidance of the ship. This controller has showed certain level of robustness. Ahıska [44], has proposed the nonlinear PID and LQR controllers for the nonlinear model of a surface vehicle. Linearization has been performed in different operating points and an auto-tuning algorithms have been implemented. Although the simulation results were satisfactory, due to environmental disturbances, the implementation has showed poor performance. Finally, Alfaro-Cid et al. [45] has proposed the decoupled sliding mode controllers with genetic algorithm optimization. In this study, without disturbances, the controller has showed very good performance; however, the controller constructed with identified parameters taking into account disturbances did not provide better performance.

1.3 Organization of the Thesis

In this thesis, modeling, system identification and controller design for a sea surface vehicle are studied. In the beginning of the thesis, an introduction part where the motivation of thesis and the selected literature are presented is given.

In Chapter 2, the mathematical model of a sea surface vehicle which is planned to be used for the purpose of the development of control and guidance algorithms will be introduced. Then, the implementation of the model in MATLAB environment and results which are obtained from the simulations will be presented.

In Chapter 3, first of all, general information about system identification will be given. Next, the methodology selected for this study will be investigated in detail. Then, the processing steps of measurement data is presented in order. Afterwards, the tools utilized for the system identification, namely, the optimization algorithms will be explained. Finally, the designed system identification experiments will be introduced and the results of the study will be provided with an evaluation.

In Chapter 4, several controllers for the motion control of the sea surface vehicle will be investigated. Throughout the design process, the mathematical model of the marine vessel which was previously derived and identified for the Pacific Islander Tug Boat will be used. Proportional-Integral-Derivative (PID) based piecewise controller and Sliding Mode Controller (SMC) techniques are basically exploited. The ultimate goal of the controllers will be to reach the desired surge velocity and yaw position which may be determined by an upper level controller. The results will be presented with comparisons for the cases with and without external disturbances.

In Chapter 5, the hardware and the software utilized in the experimental setup will be presented. In the hardware part, basically, the autopilot card and the model boat selected with their components will be investigated. In the software part, the embedded software running in the autopilot card and the ground station software prepared for remote communication will be explained.

The thesis concludes with Chapter 6. In this chapter, the summary, the evaluation of the final results and the future work for the study will be explained.

There is also one appendix part at the end of the thesis. In this part, the validation results for the simulation of the mathematical model will be given with figures.

CHAPTER 2

MATHEMATICAL MODELING OF A SEA SURFACE VEHICLE

2.1 Introduction

In this chapter, firstly, the mathematical model of a sea surface vehicle which is planned to be used for the purpose of the development of control and guidance algorithms is introduced. Then, the implementation of the model in MATLAB environment and results which are obtained from the simulations are presented.

2.2 Mathematical Modeling

Developing control algorithms with sufficient performance requires an accurate model of sea surface vehicle. For this purpose, a mathematical model based on Fossen's vectorial representation for marine vessels is constructed [1]. The model consists of inertial components and the forces acting on the vessel. The forces implemented on the current model include centripetal and Coriolis, damping, gravity, buoyancy, air drag and thruster forces. As can be seen in Chapter 5, model vehicle has left and right thrusters. Hence, both forces for left and right thrusters are included to the model. Moreover, air drag force mentioned above are only due to the motion of the vehicle in air and for this force, it is assumed that wind has zero velocity. Actual wind and sea current forces are considered as disturbances. Although mathematical representations of the disturbance forces are given in this chapter, these forces are not included in the model. In Chapter 4, these forces will be added to the model as external disturbance forces in order to measure the performances of controller structures. Furthermore, the forces associated with the behavior of rudders are not included since there is no rudder mounted on the model boat. It should also be noted that throughout the study, vectors and matrices are given either in bold or with arrows at the top.

The robot like vectorial model which is developed to define motion of the sea surface vehicles by Fossen is as follows:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{q}=\boldsymbol{\tau} \tag{2.1}$$

In Equation (2.1), $\boldsymbol{M}$ is the system inertia matrix, $\boldsymbol{C}$ is Coriolis matrix, $\boldsymbol{\tau}$ is torques applied and $\boldsymbol{q}$ is vector of joint angles. The further developed model is given by:

$$\boldsymbol{M}\dot{\boldsymbol{v}}+\boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}+\boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}+\boldsymbol{g}(\boldsymbol{\eta})+\boldsymbol{g_0}=\boldsymbol{\tau}+\boldsymbol{\tau_{wind}}+\boldsymbol{\tau_{wave}} \tag{2.2}$$

where states are

$$\begin{aligned}\boldsymbol{\eta} &= \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T && (2.3)\\ \boldsymbol{v} &= \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T && (2.4)\end{aligned}$$

The equation in (2.2) can be modified in accordance with the given explanations in the first paragraph:

$$\boldsymbol{M}\dot{\boldsymbol{v}}+\boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}=\boldsymbol{\tau_d}+\boldsymbol{\tau_g}+\boldsymbol{\tau_t}+\boldsymbol{\tau_a}+\boldsymbol{\tau_{dis}} \tag{2.5}$$

On the left hand side of (2.5), the rigid body and added mass dynamics are represented while the right hand side includes the combination of external forces. In the equation, $\boldsymbol{\tau_d}$ is the damping force of water surrounding $\boldsymbol{\tau_g}$ is the gravitational and buoyancy forces, $\boldsymbol{\tau_t}$ is the forces produced by thruster, $\boldsymbol{\tau_a}$ is the forces corresponding to air drag and $\boldsymbol{\tau_{dis}}$ represents forces related to other disturbances such as wind and current.

In the remaining part of this chapter, the notion of coordinate frames will be introduced and the model given in Equation (2.5) will be expressed in detail.

2.2.1 Coordinate Frames and Transformations

To be able to describe the position and orientation of the surface vehicle, a coordinate system, or a frame is rigidly attached to the object. By doing so, it is possible to define space characteristics (positions, orientations) of the vessel related to some reference coordinate frame [15].

Six degrees of freedom of the vehicle are illustrated in Figure 2.1.

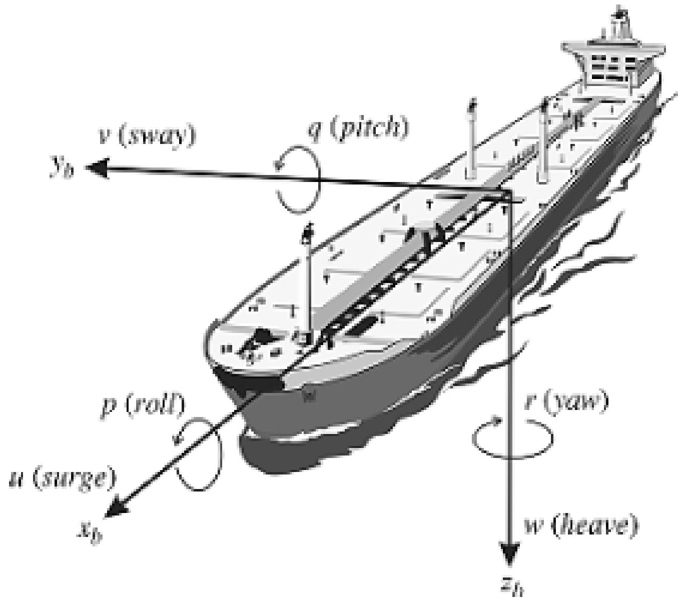


Figure 2.1: Sea surface vehicle [1]

The notation of forces and moments, linear and angular velocities and position and Euler angles are given in Table 2.1.

Table 2.1: The notation of SNAME [1]

DOF	Explanation	Forces and moments	Linear and angular velocities	Positions and Euler angles
1	motions in the x direction (surge)	X	u	x
2	motions in the y direction (sway)	Y	v	y
3	motions in the z direction (heave)	Z	w	z
4	rotation about the x axis (roll, heel)	K	p	ϕ
5	rotation about the y axis (pitch, trim)	M	q	θ
6	rotation about the z axis (yaw)	N	r	ψ

In this study, two reference frames, namely North-East-Down (NED) coordinate system and Body fixed reference frame are discussed in order to explain the motion of the vehicle.

NED frame $\{n\} = \{x_n,\ y_n,\ z_n\}$ with origin o_n is exploited as a reference to explain the motion of sea surface vehicle with respect to Earth.

Body fixed frame $\{b\} = \{x_b,\ y_b\ , z_b\}$ with origin o_b which is placed and fixed to the craft (see Figure 2.1) is cast to represent the vessel and its motion corresponding to the inertial reference frame (NED).

Reference frames can be seen from Figure 2.2.

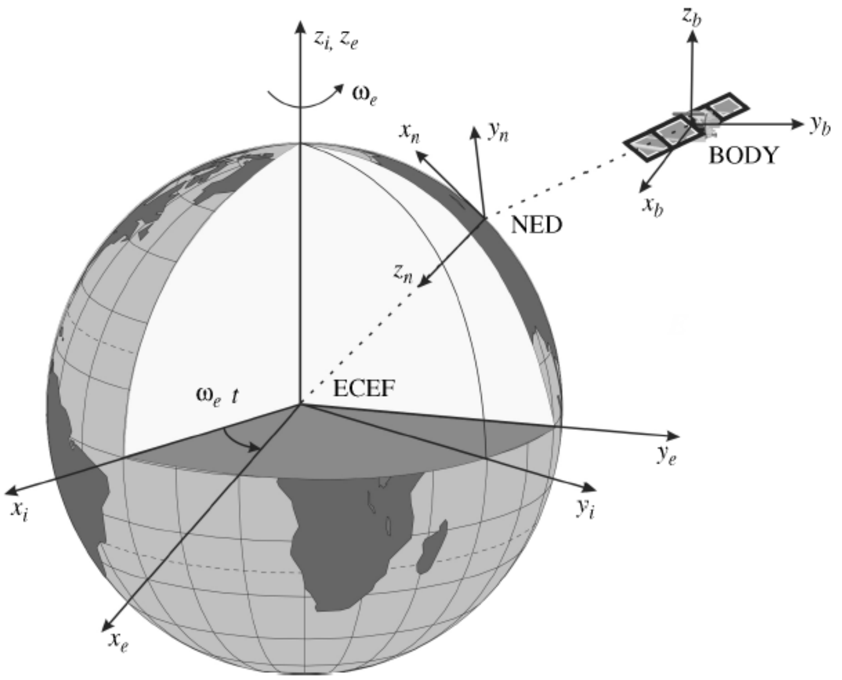


Figure 2.2: Reference frames [1]

The position and orientation of the hull are defined regarding {n} while linear and angular velocities of the vessel are expressed in {b}. Actually, states of the model are also combination of linear and angular velocities of the vehicle with respect to {b} and positions and Euler angles calculated in {n}. Finally, state notation is given in Equation 2.6

$$\boldsymbol{States} = [x\ y\ z\ \phi\ \theta\ \psi\ u\ v\ w\ p\ q\ r]^T \tag{2.6}$$

2.2.1.1 Notation

$\boldsymbol{v}_{b/n}^{b}$: Linear velocities of o_b w.r.t. {n} expressed in {b},
$\boldsymbol{w}_{b/n}^{b}$: Angular velocities of o_b w.r.t. {n} expressed in {b},
$\boldsymbol{f}_{b}^{b}$: Force with acting point o_b expressed in {b},
$\boldsymbol{m}_{b}^{b}$: Moment about o_b expressed in {b},
$\boldsymbol{\Theta}_{nb}$: Euler angles between {n} and {b}

The basics of the notation used throughout this work are given above [1]. This symbolization will be adapted and modified anywhere necessary devoted to the essentials.

The general variables arranged in vector form defining the motion of marine craft in space $\mathbb{R}^3$ are given below.

$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}_{b/n}^{n} \\ \boldsymbol{\Theta}_{nb} \end{bmatrix}$ represents the position vector from {n} to {b} and orientation of {b} expressed in Euler angles.
$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_{b/n}^{b} \\ \boldsymbol{w}_{b/n}^{b} \end{bmatrix}$ includes linear and angular velocities of body expressed in {b}.
$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{f}_{b}^{b} \\ \boldsymbol{m}_{b}^{b} \end{bmatrix}$ is the vector that indicates forces and moment acting on the body which are expressed in {b}.

2.2.1.2 Transformations

Various vector quantities (not only above indicated ones) are required to be transformed between two reference frames. To illustrate, the velocity of the rigid body should be converted into the inertial frame to be able to track the position and orientation. There are basically two kinds of transformation, namely linear and angular velocity transformations. Although derivations are not given here, representations can be seen from subsections below [1].

Linear Velocity Transformation:
Body fixed linear velocity vector, $\boldsymbol{v}_{b/n}^{b}$ can be decomposed in NED frame by using

linear velocity transformation matrix whose argument is Euler angles. Linear velocity transformation matrix is given in Equation (2.8). In equations below, sine and cosine functions are abbreviated as *s* and *c* respectively.

$$\boldsymbol{v}_{b/n}^{n} = \dot{\boldsymbol{p}}_{b/n}^{n} = \boldsymbol{R}_{b}^{n}(\boldsymbol{\Theta}_{nb})\boldsymbol{v}_{b/n}^{b} \tag{2.7}$$

$$\boldsymbol{R}_{b}^{n}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta c\phi & c\theta c\phi \end{bmatrix} \tag{2.8}$$

Angular Velocity Transformation:

Body fixed angular velocity vector, $\boldsymbol{w}_{b/n}^{b}$ and the Euler rate vector, $\dot{\boldsymbol{\Theta}}_{nb} = \left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^{T}$ can be related with the help of $\boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb})$ as in Equation (2.9). Angular velocity transformation matrix is given in Equation (2.10).

$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb})\boldsymbol{w}_{b/n}^{b} \tag{2.9}$$

$$\boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{2.10}$$

Finally, six DOF kinematic equations can be written in terms of vectors and transformation matrices defined up to this point and can be found in Equations (2.11) and (2.12).

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})\boldsymbol{v} \tag{2.11}$$

$$\begin{bmatrix} \dot{\boldsymbol{p}}_{b/n}^{n} \\ \dot{\boldsymbol{\Theta}}_{nb} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{b}^{n}(\boldsymbol{\Theta}_{nb}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{b/n}^{b} \\ \boldsymbol{w}_{b/n}^{b} \end{bmatrix} \tag{2.12}$$

2.2.2 Subparts of the Model

In this section, the components of the mathematical model given in Equation (2.5) will be investigated.

2.2.2.1 Rigid Body Dynamics

Fossen was formulated the vessel equations of motion in a vectorial form based on Newton's second law of motion as given in Equation (2.13).

$$\boldsymbol{M_{RB}}\dot{\boldsymbol{v}}+\boldsymbol{C_{RB}}(\boldsymbol{v})\boldsymbol{v}=\boldsymbol{\tau_{RB}} \tag{2.13}$$

where $\boldsymbol{M_{RB}}$ stands for the rigid- body mass matrix, $\boldsymbol{C_{RB}}$ Coriolis and centripetal matrix resulted from the rotation of {b} around {n}, $\boldsymbol{v}=[u,\ v,\ w,\ p,\ q,\ r]^T$ represents velocity vector resolved in {b} and $\boldsymbol{\tau_{RB}}=[X,\ Y,\ Z,\ K,\ M,\ N]^T$ is the vector including external forces and torques represented in {b}.

Two special points will be considered in derivation of equation of motion.
CO : Origin of {b}
CG : Center of gravity of the vehicle

Considering Euler's first and second axioms, Newton's second law of motion can be written considering conservation of both linear momentum $\vec{p}_g$ and angular momentum $\vec{h}_g$. Euler's first and second axioms can be written as:

$$\frac{{}^{i}d}{dt}\vec{p}_g = \vec{f}_g \quad \vec{p}_g=m\vec{v}_{g/i} \tag{2.14}$$

$$\frac{{}^{i}d}{dt}\vec{h}_g = \vec{m}_g \quad \vec{h}_g=\boldsymbol{I_g}\vec{w}_{b/i} \tag{2.15}$$

where $\vec{f}_g$ and $\vec{m}_g$ are the forces and moments acting on the body's CG. $\vec{w}_{b/l}$ is the angular velocity of {b} with respect to {i}, $\vec{v}_{g/i}$ is the linear velocity of the CG, m is the mass of the body, $\boldsymbol{I_g}$ is the inertia dyadic about the body's CG. There are two main assumptions when deriving the equation of motion:

- The sea surface vehicle is a rigid body. Consideration of forces between small mass elements is eliminated.
- NED frame {n} is inertial. It neglects the forces due to the Earth's motion relative to inertial reference frame. As a result:

$$\vec{v}_{g/i} \approx \vec{v}_{g/n} \tag{2.16}$$

$$\vec{w}_{b/i} \approx \vec{w}_{b/n} \tag{2.17}$$

Time differentiation of a vector $\vec{a}$ in a non-stationary reference frame $\{\mathrm{b}\}$ satisfies the following Equation [1]:

$$\frac{{}^{i}d}{dt}\vec{a} = \frac{{}^{b}d}{dt}\vec{a} + \vec{w}_{b/i} \times \vec{a} \tag{2.18}$$

This fact will be utilized in the derivation of equation of motion.

Translational Motion about CG:

Consider the following definitions:

$\vec{r}_{g/i}$: Position vector of CG relative to the inertial reference frame
$\vec{r}_{b/i}$: Position vector of CO relative to the inertial reference frame
$\vec{r}_{g}$: Position vector from the origin of the body coordinate system to the center of gravity

Hence, it follows that:

$$\vec{r}_{g/i} = \vec{r}_{b/i} + \vec{r}_{g} \tag{2.19}$$

Since inertial reference frame is NED, Equation (2.19) becomes:

$$\vec{r}_{g/n} = \vec{r}_{b/n} + \vec{r}_{g} \tag{2.20}$$

For a rigid body, CG satisfies that:

$$\frac{{}^{b}d}{dt}\vec{r}_{g} = 0 \tag{2.21}$$

By using time differentiation of a vector in a non-stationary reference frame and the above fact, the following equation can be written:

$$\vec{v}_{g/n} = \vec{v}_{b/n} + \vec{w}_{b/n} \times \vec{r}_{g} \tag{2.22}$$

Now Euler's first axiom can be used to calculate the equation of motion:

$$\begin{aligned} \vec{f}_{g} &= \frac{{}^{i}d}{dt} m\vec{v}_{g/n} \\ &= \frac{{}^{b}d}{dt} m\vec{v}_{g/n} + m\vec{w}_{b/n} \times \vec{v}_{g/n} \\ &= m\left(\dot{\vec{v}}_{g/n} + \vec{w}_{b/n} \times \vec{v}_{g/n}\right) \end{aligned}$$

$$\boldsymbol{f}_{\boldsymbol{g}}^{\boldsymbol{b}} = m\left[\dot{\boldsymbol{v}}_{\boldsymbol{g/n}}^{\boldsymbol{b}} + \boldsymbol{S}\left(\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{v}_{\boldsymbol{g/n}}^{\boldsymbol{b}}\right] \tag{2.23}$$

where $\boldsymbol{S}$ is the cross product operation defined in [1] and m is the mass of the body.

Rotational Motion about CG:

Similar approach can be utilized to derive the rotational motion equation. Considering Euler's second axiom, the following equation can be written:

$$\begin{aligned}\vec{m}_g &= \frac{{}^i d}{dt}\left(I_g\vec{w}_{b/n}\right)\\ &= \frac{{}^b d}{dt}\left(I_g\vec{w}_{b/n}\right)+\vec{w}_{b/n}\times\left(I_g\vec{w}_{b/n}\right)\\ &= I_g\dot{\vec{w}}_{b/n}-\left(I_g\vec{w}_{b/n}\right)\times\vec{w}_{b/n}\end{aligned}$$

$$\boldsymbol{m}_{\boldsymbol{g}}^{\boldsymbol{b}}=\boldsymbol{I}_{\boldsymbol{g}}\dot{\boldsymbol{w}}_{\boldsymbol{b/n}}^{\boldsymbol{b}}-\boldsymbol{S}\left(\boldsymbol{I}_{\boldsymbol{g}}\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}} \tag{2.24}$$

where $\boldsymbol{I_g}$ is the inertia matrix about CG and is defined as:

$$\boldsymbol{I_g}\triangleq\begin{bmatrix} I_x & -I_{xy} & -I_{xz}\\ -I_{yx} & I_y & -I_{yz}\\ -I_{zx} & -I_{zy} & I_z\end{bmatrix} \tag{2.25}$$

By using matrix manipulations, coordinate system transformations and parallel axis theorem, translational and rotational motion equations about CO can be derived as:

$$\boldsymbol{f}_{\boldsymbol{b}}^{\boldsymbol{b}}=m\left[\dot{\boldsymbol{v}}_{\boldsymbol{b/n}}^{\boldsymbol{b}}+\boldsymbol{S}\left(\dot{\boldsymbol{w}}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{r}_{\boldsymbol{g}}^{\boldsymbol{b}}+\boldsymbol{S}\left(\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{v}_{\boldsymbol{b/n}}^{\boldsymbol{b}}+\boldsymbol{S}^2\left(\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{r}_{\boldsymbol{g}}^{\boldsymbol{b}}\right] \tag{2.26}$$

$$\boldsymbol{m}_{\boldsymbol{b}}^{\boldsymbol{b}}=\boldsymbol{I}_{\boldsymbol{b}}\dot{\boldsymbol{w}}_{\boldsymbol{b/n}}^{\boldsymbol{b}}+\boldsymbol{S}\left(\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{I}_{\boldsymbol{b}}\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}+m\boldsymbol{S}\left(\boldsymbol{r}_{\boldsymbol{g}}^{\boldsymbol{b}}\right)\dot{\boldsymbol{v}}_{\boldsymbol{b/n}}^{\boldsymbol{b}}+m\boldsymbol{S}\left(\boldsymbol{r}_{\boldsymbol{g}}^{\boldsymbol{b}}\right)\boldsymbol{S}\left(\boldsymbol{w}_{\boldsymbol{b/n}}^{\boldsymbol{b}}\right)\boldsymbol{v}_{\boldsymbol{b/n}}^{\boldsymbol{b}} \tag{2.27}$$

Details of the derivation can be found in [1]. Considering Equations (2.26) and (2.27), matrix components of equation of motion can be found as:

$$\boldsymbol{M_{RB}} = \begin{bmatrix} m\boldsymbol{I}_{3\times 3} & -m\boldsymbol{S}\left(\boldsymbol{r}_{\boldsymbol{c}}^{\boldsymbol{b}}\right)\\ m\boldsymbol{S}\left(\boldsymbol{r}_{\boldsymbol{c}}^{\boldsymbol{b}}\right) & \boldsymbol{I_b}\end{bmatrix} = \begin{bmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22}\end{bmatrix} \tag{2.28}$$

$$\boldsymbol{C_{RB}}(\boldsymbol{v}) = \begin{bmatrix}\boldsymbol{0}_{3\times 3} & -\boldsymbol{S}\left(\boldsymbol{M}_{11}\boldsymbol{v}_1+\boldsymbol{M}_{12}\boldsymbol{v}_2\right)\\ -\boldsymbol{S}\left(\boldsymbol{M}_{11}\boldsymbol{v}_1+\boldsymbol{M}_{12}\boldsymbol{v}_2\right) & -\boldsymbol{S}\left(\boldsymbol{M}_{21}\boldsymbol{v}_1+\boldsymbol{M}_{22}\boldsymbol{v}_2\right)\end{bmatrix} \tag{2.29}$$

$$\boldsymbol{M}_{RB}=\begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g\\ 0 & m & 0 & -mz_g & 0 & mx_g\\ 0 & 0 & m & my_g & -mx_g & 0\\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz}\\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz}\\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z\end{bmatrix} \tag{2.30}$$

$$\boldsymbol{C_{RB}}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & 0 & m(y_gq+z_gr) & -m(x_gq-w) & -m(x_gr+v) \\ 0 & 0 & 0 & -m(y_gp+w) & m(z_gr+x_gp) & -m(y_gr-u) \\ 0 & 0 & 0 & -m(z_gp-v) & -m(z_gq+u) & m(x_gp+y_gq) \\ -m(y_gq+z_gr) & m(y_gp+w) & m(z_gp-v) & 0 & -I_{yz}q-I_{xz}p+I_zr & I_{yz}r+I_{xy}p-I_yq \\ -m(x_gq-w) & -m(z_gr+x_gp) & m(z_gp+u) & I_{yz}q+I_{xz}p-I_zr & 0 & -I_{xz}r-I_{xy}q-I_xp \\ m(x_gr+v) & m(y_gr-u) & -m(x_gp+y_gq) & -I_{yz}r-I_{xy}p+I_yq & I_{xz}r+I_{xy}q-I_xp & 0 \end{bmatrix} \tag{2.31}$$

where x_g, y_g and z_g are the components of the vector from CO to CG. After assumptions below are made, the model is further simplified.

- CG coincides with the origin of {b}, i.e., $\boldsymbol{r_g^b} = \bar{0}$.
- Body axes are arranged in such a way that inertia matrix becomes diagonal, i.e., $\boldsymbol{I_b} = diag(I_x, I_y, I_x)$.

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I_b} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{C_{RB}}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_zr & -I_yq \\ -mw & 0 & mu & -I_zr & 0 & I_xp \\ mv & -mu & 0 & I_yq & -I_xp & 0 \end{bmatrix} \tag{2.33}$$

2.2.2.2 Added Mass Dynamics

Added mass is used to express the change in motion characteristics of a body moving in fluid. The body appears to have more mass than the genuine case while it moves in fluid and this physical phenomenon is overcome by the introduction of “added mass” or “virtual mass”. The term is related with the fluid surrounding the vessel. When the boat moves, it results in a motion of otherwise stationary fluid. In other words, the marine craft needs the fluid move out of its headway. For this purpose, some kinetic energy is required to be transferred from the vehicle to the surrounding fluid. This effect is reflected on the equations of motion by added mass terms given below [3] for surface vessels, assuming that heave, pitch and roll modes are neglected:

$$\boldsymbol{M_A} = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & Y_{\dot{r}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{\dot{v}} & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \tag{2.34}$$

$$\boldsymbol{C_A}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & Y_{\dot{v}}v + \frac{Y_{\dot{r}}+N_{\dot{v}}}{2}r \\ 0 & 0 & 0 & 0 & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -Y_{\dot{v}}v - \frac{Y_{\dot{r}}+N_{\dot{v}}}{2}r & X_{\dot{u}}u & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.35}$$

where $X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}}$ and other terms can be expressed similar to given one. For surface ships, it can be assumed that submerged part of the ship is a half cylinder. Then added mass terms can be estimated as:

$$X_{\dot{u}} = -0.05m \tag{2.36}$$

$$Y_{\dot{v}} = -1/2\rho_w \pi d^2 l \tag{2.37}$$

$$N_{\dot{r}} = -1/24(0.1mb^2 + \rho_w d^2 l^3) \tag{2.38}$$

$$N_{\dot{v}} = Y_{\dot{r}} \tag{2.39}$$

where m is the mass of the ship, ρ_w is the density of water, d is the draft of the chip, l and b are the length and beam of the ship, respectively.

2.2.2.3 Hydrodynamic Damping Forces

Hydrodynamic damping for ocean vehicles mainly consists of four components [14]:

$\boldsymbol{D_P}(\boldsymbol{v})$: Radiation – induced potential damping due to forced body oscillations

$\boldsymbol{D_S}(\boldsymbol{v})$: Linear skin friction due to laminar boundary layers and quadratic skin friction due to turbulent boundary layers

$\boldsymbol{D_W}(\boldsymbol{v})$: Wave drift damping

$\boldsymbol{D_M}(\boldsymbol{v})$: Damping due to vortex shedding

As a result, total damping matrix can be expressed as a sum of these components:

$$\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D_P}(\boldsymbol{v}) + \boldsymbol{D_S}(\boldsymbol{v}) + \boldsymbol{D_W}(\boldsymbol{v}) + \boldsymbol{D_M}(\boldsymbol{v}) \tag{2.40}$$

where $\boldsymbol{D}(\boldsymbol{v})$ is the damping matrix and it is real, non-symmetrical and strictly positive.

Radiation Induced Potential Damping:

This type of damping appears when a body oscillates with the wave excitation frequency without incident waves. It is generally referred to as potential damping. However, effects of potential damping terms are negligible compared to other dissipative terms.

Skin Friction:

Skin friction has two components. First component is linear skin friction due to laminar boundary layers and second component is a high frequency contributor due to turbulent boundary layers.

Wave Drift Damping:

When surface vessels advance in waves, wave drift damping forces occurs as added resistance. Values and expressions of wave drift damping can be derived from second order wave theory. Its contribution to surge motion is very significant for higher sea states.

Damping due to Vortex Shedding:

The model of the viscous damping force can be written as:

$$f(U) = -\frac{1}{2}\rho_w C_D(R_n) A|U|U \tag{2.41}$$

where U is the speed of the vessel, A is the projected cross sectional area under water, $C_D(R_n)$ is the drag coefficient and ρ_w is the water density.

As can be seen from Equation (2.41) drag coefficient is a function of Reynolds number. Reynolds number can be expressed as:

$$R_n = \frac{UD}{v} \tag{2.42}$$

where D is the characteristic length of the body and v is the kinematic viscosity coefficient. It is usually suitable to write total damping matrix as summation of linear and nonlinear damping matrices. Different damping terms effects both linear and nonlinear parts.

$$\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D_l} + \boldsymbol{D_n}(\boldsymbol{v}) \tag{2.43}$$

where $\boldsymbol{D_l}$ is the linear damping matrix and $\boldsymbol{D_n}(\boldsymbol{v})$ is the nonlinear damping matrix.

In this study, “Expanded Ad Hoc damping model for high speed maneuvers” [14] is utilized to calculate the linear and the nonlinear parts of the damping matrix. In this model, linear damping matrix can be expressed as:

$$\boldsymbol{D_l} = - \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_u & 0 & 0 & 0 & Y_r \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & N_v & 0 & 0 & 0 & N_r \end{bmatrix} \tag{2.44}$$

where terms are hydrodynamic derivatives which can be found experimentally. For low speed applications, the following assumption can be made:

$$N_v = Y_r \tag{2.45}$$

Furthermore, nonlinear damping matrix can be written as:

$$\boldsymbol{D_n} = - \begin{bmatrix} X_{|u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{|v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{|w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{|p|p}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{|q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{|r|r}|r| \end{bmatrix} \tag{2.46}$$

where terms can be calculated using Equation (2.41) with appropriate sign.

It is very important to include both linear and nonlinear damping terms since nonlinear damping terms may cause oscillations at low speeds. Linear damping terms provides exponential convergence to zero.

Considering linear and nonlinear damping matrices, damping forces can be concluded as

$$\boldsymbol{\tau_d} = -\boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} \tag{2.47}$$

In this study, it should be noted that some assumptions are made:

- Box shaped symmetrical body
- $R_n > 10^4$
- z direction damping is also added to Ad Hoc model and expanded model is obtained.

2.2.2.4 Restoring Forces

In Equation (2.5), $\boldsymbol{\tau_g}$ on the right hand side, stands for gravitational and buoyancy forces. $\boldsymbol{\tau_g}$ can also be named as restoring force since it tends to transfer the vessel to its equilibrium state for the range of stability. To be able to model $\boldsymbol{\tau_g}$, definitions below should be made:

CG: The point at which all mass of the boat is assumed to be concentrated.
CB: The center of gravity of displaced water or underwater volume of the boat.
GM_T (Transverse metacentric height): The distance between transverse metacenter and *CG*.
GM_L (Longitudinal metacentric height): The distance between longitudinal metacenter and *CG*.
As Archimedes stated, for floating boat at rest buoyancy and weight are balanced.

$$mg = \rho_w g \nabla \tag{2.48}$$

where ∇ is the volume of the water displaced by the vessel. Assume that $z = 0$ at the position of nominal water displacement. Then, any deviation of the position in z will immediately result in a net force due to change in the magnitude of buoyancy force:

$$Z = mg - \rho_w g(\nabla + \delta\nabla(z)) = -\rho_w g\delta\nabla(z) \tag{2.49}$$

By making box-shaped body assumption for the surface vessel, i.e., $A_{wp}(z) = A_{wp}(0)$, where A_{wp} is the water plane area, hence the force can be written as:

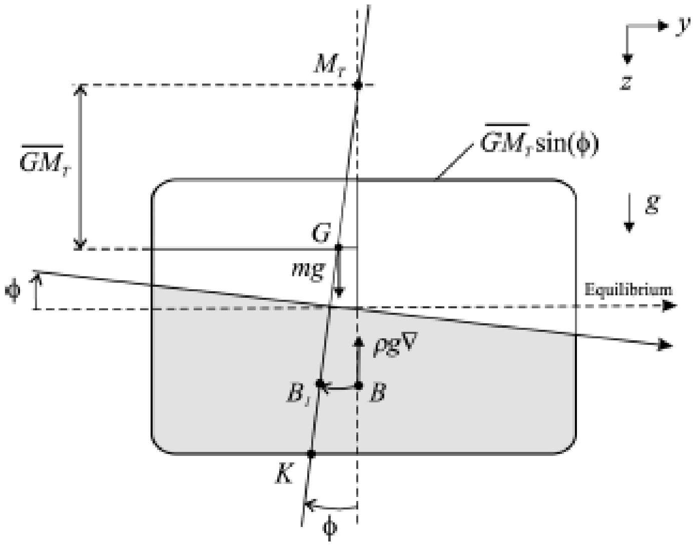


Figure 2.3: Traverse metacentric stability [1]

$$Z \approx -\rho_w g A_{wp} z = z_0 z \tag{2.50}$$

where $z_0 = -\rho_w g A_{wp}$.

Next, the force is to be decomposed in {b} frame:

$$\delta \boldsymbol{f}_r^b = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})^{-1}\boldsymbol{\delta f}_r^n = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})^{-1} z_0 z = z_0 z \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix} \tag{2.51}$$

As can be clearly seen from Figure 2.3 and considering similar figure for longitudinal axis, moment arms in roll and pitch can be written as:

$$\boldsymbol{r}_r^b = \begin{bmatrix} -GM_T s\theta \\ GM_L s\phi \\ 0 \end{bmatrix} \tag{2.52}$$

The z-direction buoyancy force is resolved in {b} as:

$$\boldsymbol{f_r^b} = \boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})^{-1} \begin{bmatrix} 0 \\ 0 \\ -\rho_w g \nabla \end{bmatrix} = -\rho_w g \nabla \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix} \tag{2.53}$$

The contribution of $\boldsymbol{\delta f_r^b}$ to the restoring moment is small enough to be neglected (compared to the effect of $\boldsymbol{f_r^b}$). The moment is calculated as:

$$\boldsymbol{m_r^b} = \boldsymbol{r_r^b} \times \boldsymbol{f_r^b} = -\rho_w g \nabla \begin{bmatrix} GM_T s\phi c\theta c\phi \\ GM_L s\theta c\theta c\phi \\ \left(-GM_L c\theta + GM_L\right) s\phi s\theta \end{bmatrix} \tag{2.54}$$

Finally, the restoring forces and moments are combined to form a single vector:

$$\boldsymbol{\tau_g} = \begin{bmatrix} \boldsymbol{\delta f_r^b} \\ \boldsymbol{m_r^b} \end{bmatrix} = \begin{bmatrix} -z_0 z s\theta \\ z_0 z c\theta s\phi \\ z_0 z c\theta c\phi \\ -\rho_w g \nabla GM_T s\phi c\theta c\phi \\ -\rho_w g \nabla GM_L s\theta c\theta c\phi \\ -\rho_w g \nabla \left(-GM_L c\theta + GM_T\right) s\phi s\theta \end{bmatrix} \tag{2.55}$$

By small angle assumption (i.e., ϕ, θ and z are small such that $s\phi \approx \phi$, $c\phi \approx 1$), final form of the force can be found:

$$\boldsymbol{\tau_g} = \begin{bmatrix} 0 \\ 0 \\ z_0 z \\ -\rho_w g \nabla GM_T \phi \\ -\rho_w g \nabla GM_L \theta \\ 0 \end{bmatrix} \tag{2.56}$$

Computation of metacenter heights for surface vessels can be carried out by using basic hydrostatics [1]. Initial estimation of them is performed using this methodology. Consider the equations below:

$$GM_T = BM_T - BG \tag{2.57}$$

$$GM_L = BM_L - BG \tag{2.58}$$

where BM_T and BM_L are the traverse and longitudinal radii of curvatures, BG is the distance between center of buoyancy and the center of gravity.

For small inclinations, radii of curvatures can be estimated as [1]:

$$BM_T = \frac{I_T}{\nabla} \tag{2.59}$$

$$BM_L = \frac{I_L}{\nabla} \tag{2.60}$$

where I_T and I_L are the traverse and longitude moment of areas respectively and ∇ is the volume displacement of water. Considering rectangular water plane for surface ships, moment of areas can be approximated as in following equations [1]:

$$I_L \approx 1/12L^3B \tag{2.61}$$

$$I_T \approx 1/12B^L \tag{2.62}$$

where L and B is the length and beam of the hull.

2.2.2.5 Thruster Forces

Two independent thrusters located at the stern of the marine craft are included in the model. They are just modeled as sources of force (rotational effects and motor characteristics are neglected).

$\boldsymbol{r_{t_R}}$: the position vector from CG to the right thruster.
$\boldsymbol{r_{t_L}}$: the position vector from CG to the left thruster.
Let $\boldsymbol{f_{t_R}}$ and $\boldsymbol{f_{t_L}}$ be the forces induced by right and left thrusters respectively.

The related moments are calculated as:

$$\boldsymbol{m_R} = \boldsymbol{r_{t_R}} \times \boldsymbol{f_{t_R}} \tag{2.63}$$

$$\boldsymbol{m_L} = \boldsymbol{r_{t_L}} \times \boldsymbol{f_{t_L}} \tag{2.64}$$

Total forces and moments produced by thrusters are computed by superposition:

$$\boldsymbol{\tau_t} = \begin{bmatrix} \boldsymbol{f_{t_R}} \\ \boldsymbol{m_R} \end{bmatrix} + \begin{bmatrix} \boldsymbol{f_{t_L}} \\ \boldsymbol{m_L} \end{bmatrix} \tag{2.65}$$

2.2.2.6 Air Drag Forces

In this section, forces related to air drag will be investigated [44]. These forces are related to the relative velocity of the vessel in air. Relative velocity of vehicle in air can be calculated as:

$$\boldsymbol{v_r} = \boldsymbol{v_b} - \boldsymbol{v_a} \tag{2.66}$$

where $\boldsymbol{v_r}$ is the relative velocity of vehicle in air, $\boldsymbol{v_b}$ is the vehicle velocity and $\boldsymbol{v_a}$ is the air or wind velocity.

The air drag forces can be calculated as:

$$F_a = A_a P_a C_{DA} \tag{2.67}$$

F_a is the air drag force, A_a is the area of the vessel in air, in other words, the area on which air drag presents, P_a is the air pressure, and C_{DA} is the air drag coefficient.

Air pressure is proportional to the square of air velocity. This is expressed as,

$$P_a \approx 2.56 v_r^2 \tag{2.68}$$

The torques can be calculated by cross product operation. Assuming vehicle is symmetric and center of gravity is at the midpoint of vehicle, moment arm and torques can be expressed as:

$$\boldsymbol{\lambda} = \begin{bmatrix} \frac{l}{2} & \frac{wi}{2} & \frac{h}{2} \end{bmatrix}^T \tag{2.69}$$

$$\boldsymbol{T_a} = \boldsymbol{\lambda} \times F_a \tag{2.70}$$

where l is length, wi is the width and h is the height of the vehicle. For small vehicles, since moment arm lengths are small, torque contribution can be neglected. As a result, air drag forces can be expressed as:

$$\boldsymbol{\tau_a} = -\begin{bmatrix} F_{ax} sgn(u) & F_{ay} sgn(v) & F_{az} sgn(w) & 0 & 0 & 0 \end{bmatrix}^T \tag{2.71}$$

In this study, it should be reminded that some assumptions are made:

- Box shaped symmetrical body
- Vehicle is small; hence torque contribution of air drag is small.

2.2.2.7 Disturbance Forces

Considering surface vessels, there are three main type of environmental disturbances. These are given in below:

- Wind-generated waves,
- Ocean currents,
- Wind forces.

For this study, only wind-generated waves are considered since it is the main disturbance effect for surface vessels. The model that has been used to represent the wave's effects on vessels constructs the forces induced by a regular sea on a block shaped ship and it is explained in [3]. It has three main components and these elements can be found in Equations (2.72), (2.74). These forces can be directly added to model as surge, sway forces and yaw torque.

$$X_{wave}(t) = \sum_{i=1}^{N} \rho_w g BLT c(\beta - \psi) s_i(t) \tag{2.72}$$

$$Y_{wave}(t) = \sum_{i=1}^{N} \rho_w g BLT s(\beta - \psi) s_i(t) \tag{2.73}$$

$$N_{wave}(t) = \sum_{i=1}^{N} 1/24 \rho_w g BL(L^2 - B^2) s2(\beta - \psi) s_i^2(t) \tag{2.74}$$

Here L, B and D are the length, beam and draft of the wetted part of the surface vessel. g_w is the density of water, $s_i(t)$ is the wave slope and $\beta - \psi$ is the angle between ship heading and wave direction.

The wave slope $s_i(t)$ can be computed as in Equation (2.75).

$$s_i(t) = A_i k_i s(w_{ei} t + \phi_i) \tag{2.75}$$

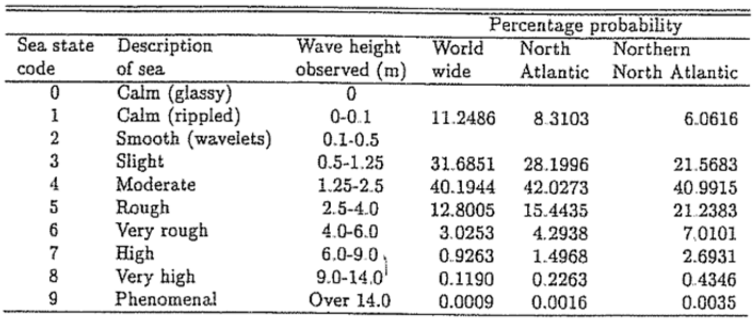
where A_i is the wave amplitude and k_i is the wave number.

Wave slope and its parameters can found by using specific wave spectral density function $S(w_i)$. In this study, Pierson-Moskowitz spectrum [3] is used. The expression of this spectrum is given in Equation (2.76).

$$S(w) = 8.1 \times 10^{-3} g^2 exp(\frac{-3.11}{H_s^2} w^{-4}) \tag{2.76}$$

where g is the gravitational constant, H_s is the significant wave height (mean of the one-third of the highest wave).

Significant wave height parameters can be calculated using Figure 2.4.



			Percentage probability		
Sea state code	Description of sea	Wave height observed (m)	World wide	North Atlantic	Northern North Atlantic
0	Calm (glassy)	0			
1	Calm (rippled)	0-0.1	11.2486	8.3103	6.0616
2	Smooth (wavelets)	0.1-0.5			
3	Slight	0.5-1.25	31.6851	28.1996	21.5683
4	Moderate	1.25-2.5	40.1944	42.0273	40.9915
5	Rough	2.5-4.0	12.8005	15.4435	21.2383
6	Very rough	4.0-6.0	3.0253	4.2938	7.0101
7	High	6.0-9.0	0.9263	1.4968	2.6931
8	Very high	9.0-14.0	0.1190	0.2263	0.4346
9	Phenomenal	Over 14.0	0.0009	0.0016	0.0035

Figure 2.4: Description of sea state [3]

Details can be found in [3].

2.3 Implementation

After the model is constructed in continuous-time domain, it is discretized in order to be tested on a computer. For this purpose, combination of Euler's backward and forward integration method is utilized as in Equations (2.77) and (2.78).

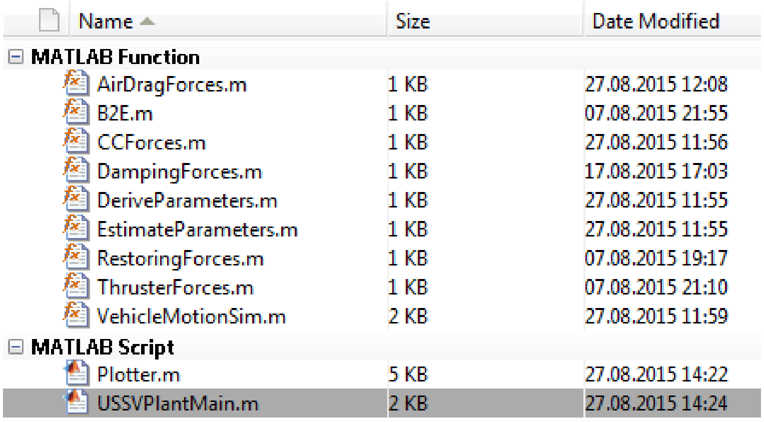
$$\begin{aligned} \boldsymbol{\nu}(k+1) &= \boldsymbol{\nu}(k)+h\boldsymbol{M}^{-1}\left(\boldsymbol{\tau}-\boldsymbol{C}(\boldsymbol{\nu}(k))\boldsymbol{\nu}(k)\right) && (2.77) \\ \boldsymbol{\eta}(k+1) &= \boldsymbol{\eta}(k)+h\boldsymbol{J}_{\Theta}(\eta(k))\boldsymbol{\nu}(k+1) && (2.78) \end{aligned}$$

where h is the sampling interval.

The mathematical model is realized in MATLAB environment. Each element is implemented with an independent function which is called by the "Vehicle Motion Simulator". This kind of approach leads to a modular design and the modules can be observed individually within the meaning of their effects on the motion.

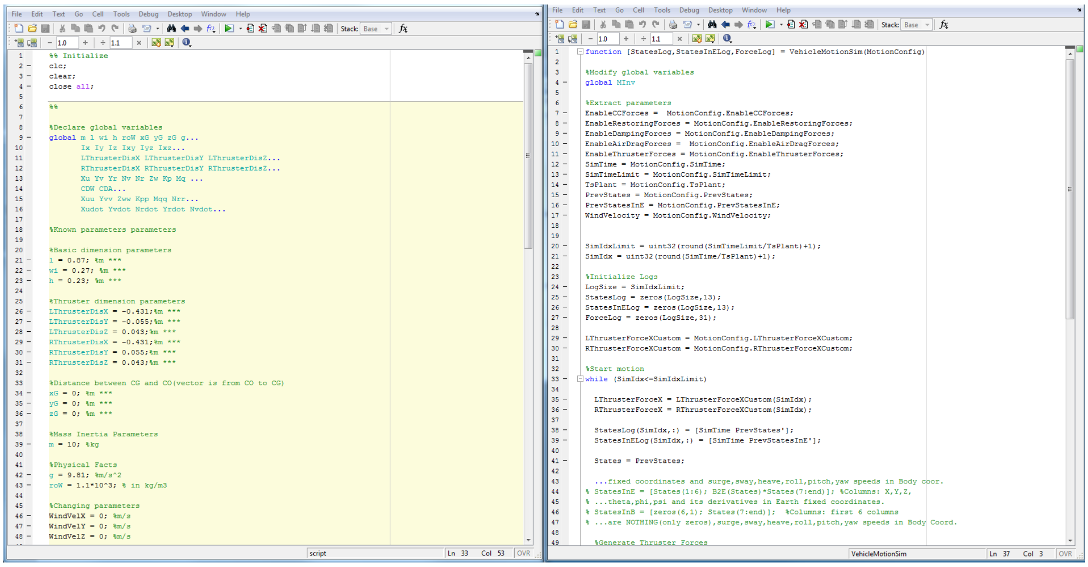
A snapshot is taken from the folder containing main file, namely 'USSVPlantMain' and other functions realizing the standalone modules are given in Figure 2.5.

Finally, Figure 2.6 represents a view from the vehicle motion simulator and main function.



Name	Size	Date Modified
MATLAB Function		
AirDragForces.m	1 KB	27.08.2015 12:08
B2E.m	1 KB	07.08.2015 21:55
CCForces.m	1 KB	27.08.2015 11:56
DampingForces.m	1 KB	17.08.2015 17:03
DeriveParameters.m	1 KB	27.08.2015 11:55
EstimateParameters.m	1 KB	27.08.2015 11:55
RestoringForces.m	1 KB	07.08.2015 19:17
ThrusterForces.m	1 KB	07.08.2015 21:10
VehicleMotionSim.m	2 KB	27.08.2015 11:59
MATLAB Script		
Plotter.m	5 KB	27.08.2015 14:22
USSVPlantMain.m	2 KB	27.08.2015 14:24

Figure 2.5: List of functions utilized in the simulator



```
%% Initialize
clc;
clear;
close all;

%%

%Declare global variables
global m l wi h roW xG yG zG g...
       Ix Iy Iz Ixy Iyz Ixz...
       LThrusterDisX LThrusterDisY LThrusterDisZ...
       RThrusterDisX RThrusterDisY RThrusterDisZ...
       Xu Yv Yr Nv Nr Zw Kp Mq ...
       CDW CDA...
       Xuu Yvv Zww Kpp Mqq Nrr...
       Xudot Yvdot Nrdot Yrdot Nvdot...

%Known parameters parameters

%Basic dimension parameters
l = 0.87; %m ***
wi = 0.27; %m ***
h = 0.23; %m ***

%Thruster dimension parameters
LThrusterDisX = -0.431;%m ***
LThrusterDisY = -0.055;%m ***
LThrusterDisZ = 0.043;%m ***
RThrusterDisX = -0.431;%m ***
RThrusterDisY = 0.055;%m ***
RThrusterDisZ = 0.043;%m ***

%Distance between CG and CO(vector is from CO to CG)
xG = 0; %m ***
yG = 0; %m ***
zG = 0; %m ***

%Mass Inertia Parameters
m = 10; %kg

%Physical Facts
g = 9.81; %m/s^2
roW = 1.1*10^3; % in kg/m3

%Changing parameters
WindVelX = 0; %m/s
WindVelY = 0; %m/s
WindVelZ = 0; %m/s
```

script Ln 33 Col 53 OVR

```
function [StatesLog,StatesInELog,ForceLog] = VehicleMotionSim(MotionConfig)

%Modify global variables
global MInv

%Extract parameters
EnableCCForces =  MotionConfig.EnableCCForces;
EnableRestoringForces = MotionConfig.EnableRestoringForces;
EnableDampingForces = MotionConfig.EnableDampingForces;
EnableAirDragForces =  MotionConfig.EnableAirDragForces;
EnableThrusterForces = MotionConfig.EnableThrusterForces;
SimTime = MotionConfig.SimTime;
SimTimeLimit = MotionConfig.SimTimeLimit;
TsPlant = MotionConfig.TsPlant;
PrevStates = MotionConfig.PrevStates;
PrevStatesInE = MotionConfig.PrevStatesInE;
WindVelocity = MotionConfig.WindVelocity;


SimIdxLimit = uint32(round(SimTimeLimit/TsPlant)+1);
SimIdx = uint32(round(SimTime/TsPlant)+1);

%Initialize Logs
LogSize = SimIdxLimit;
StatesLog = zeros(LogSize,13);
StatesInELog = zeros(LogSize,13);
ForceLog = zeros(LogSize,31);

LThrusterForceXCustom = MotionConfig.LThrusterForceXCustom;
RThrusterForceXCustom = MotionConfig.RThrusterForceXCustom;

%Start motion
while (SimIdx<=SimIdxLimit)

  LThrusterForceX = LThrusterForceXCustom(SimIdx);
  RThrusterForceX = RThrusterForceXCustom(SimIdx);

  StatesLog(SimIdx,:) = [SimTime PrevStates'];
  StatesInELog(SimIdx,:) = [SimTime PrevStatesInE'];

  States = PrevStates;

  ...fixed coordinates and surge,sway,heave,roll,pitch,yaw speeds in Body coor.
% StatesInE = [States(1:6); B2E(States)*States(7:end)]; %Columns: X,Y,Z,
% ...theta,phi,psi and its derivatives in Earth fixed coordinates.
% StatesInB = [zeros(6,1); States(7:end)];  %Columns: first 6 columns
% ...are NOTHING(only zeros),surge,sway,heave,roll,pitch,yaw speeds in Body Coord.

  %Generate Thruster Forces
```

VehicleMotionSim Ln 37 Col 3 OVR

Figure 2.6: A view from motion simulator

2.4 Simulation Tests

In this part, the outputs of the simulation will be investigated for different inputs and initial conditions. Outputs are assumed to be the state variables of the system which consist of positions, orientations in {n} frame and linear, angular velocities in {b} frame. Although tests conducted are given in this section, results can be found in Appendix A. Sampling time h is taken as 0.01 seconds. Initial estimated parameter values are utilized in the tests in order to understand the feasibility of the simulation. Exact parameters will be found in Chapter 3.

2.4.1 Zero Input, Zero State Response

For this scenario, thrusters do not exert any force on the body, and the initial condition is equal to zero vector ($\boldsymbol{\tau_t} = 0$, $\boldsymbol{\eta}(0) = 0$, $\boldsymbol{\nu}(0) = 0$). This test is designed in order to understand the stationary behavior of the model. As expected, all states remain as zero and stationary condition does not change as can be seen in Figures A.1, A.2, A.3, A.4.

2.4.2 Zero Input, Nonzero State Response

Tests in this part are designed to examine zero input response of the system. For this purpose, thrusters do not exert any force on the body, different initial conditions of state variables are applied ($\boldsymbol{\tau_t} = 0$). States with nonzero initial conditions are heave, roll, pitch positions, surge, sway, heave, roll and pitch speeds. The result of this experiment will give us information about stability characteristics of the vehicle. As can be seen from Figures A.5, A.6, A.7, A.8, all nonzero states damped to zero, as expected.

2.4.3 Equal Input, Zero State Response

In this experiment, left and right thrusters apply same amount of force on the body. Initial condition of the states are taken as zero ($\boldsymbol{\eta}(0) = 0$, $\boldsymbol{\nu}(0) = 0$). This test is designed to see the linear motion characteristics of the vessel.

$$\boldsymbol{\tau_t^L} = [2\,0\,0]^T N \tag{2.79}$$

$$\boldsymbol{\tau_t^R} = [2\,0\,0]^T N \tag{2.80}$$

As can be seen from Figures A.9, A.10, A.11, A.12, linear speed reaches a constant value, angular rates goes to zero, there is a final constant pitch angle. These are all expected results.

2.4.4 Non-equal Input, Zero State Response

In this experiment, left and right thrusters apply different amount of force on the body. Initial condition of the states are taken as zero ($\boldsymbol{\eta}(0) = 0$, $\boldsymbol{\nu}(0) = 0$). This test is carried out to see the maneuvering properties of the vehicle.

$$\boldsymbol{\tau_t^L} = [4\,0\,0] N \tag{2.81}$$

$$\boldsymbol{\tau_t^R} = [2\,0\,0] N \tag{2.82}$$

Considering Figures A.13, A.14, A.15, A.16, it can be seen that angular speeds except yaw rate damp to zero, yaw rate reaches a constant final value, both surge and sway speed goes to final nonzero values. These are also expected general trends.

CHAPTER 3

SYSTEM IDENTIFICATION

Efficient design of autopilot systems for unmanned surface vessels requires accurate and computationally simple plant models. In Chapter 2, six DOF mathematical model of a surface vessel is introduced and developed. However, as mentioned in that chapter, only estimated values of parameters are known. From controller design point of view, accurate values of these parameters should be found. This leads to better understanding of the surface vessel model and provides valuable insight for controller structure. In this chapter, first of all, general information about system identification is given. Next, the system identification methodology utilized in this study is investigated. Then, off-line processing of collected data for the purpose of system identification is mentioned. In the next section, experiments prepared for system identification are explained and given with results. In the final section, an evaluation based on this methodology is carried out.

3.1 Introduction

In literature, there are several classifications of system identification methods. First one can be divided into three categories, namely white-box, gray-box and black-box analyses [21]. White-box models are totally derived from first principles of nature. All equations and parameters either are determined by first principles or have direct relation with them. While white-box models do not have dependencies on experimental data, black-box models completely depends on empirical data. No prior knowledge is required about the model and first principles are not utilized in black-box models. Gray-box models represents a compromise between black and white-box models. Both first principles and measurement data are evaluated. Typically, model structure is created by using prior knowledge on first principles and model parameters are determined by experimental data. In second classification, there are on-line and off-line identification methods. According to [46], "The off-line system identification is the determination of model of a system using a batch of measured data where the whole batch is available all stages of the procedure". It is related to processing of data. On the other hand, in on-line methods, identification and measuring are running in parallel with some amount of lag of latter. This method is generally utilized in optimized and adaptive controllers. The off-line methods are always superior to on-line ones due to increased complexity of algorithms that can be implemented. The third classification is about whether parameters are considered as stochastic or not. Although most of the studies about system identification is carried out with deterministic parameters [46], some recent studies also consider the stochastic ones as in [33]. In last classification, there are time and frequency domain analyses. Frequency domain analysis is mostly studied on the linear models, however, time domain analysis can be accomplished in both linear and nonlinear systems. In frequency domain analysis, parameters to be identified are coefficients of the numerator and denominator polynomials in transfer functions. Some advantages of the frequency domain analysis are that ability to truncate the frequency measured and explicit capturing of the system modes [18].

3.2 Methodology of the Study

In this study, gray-box, off-line, deterministic, time domain analysis is performed. It is gray-box analysis because a mathematical model is developed for this study. Moreover, off-line methods are utilized due to high performance characteristics of this type of algorithms. Furthermore, deterministic approach is implemented because, as will be mentioned in the next section, collected data is processed with stochastic methods in order to obtain more accurate measurements. It is decided that there is no need for future estimation. Finally, time domain method is considered since the system model implemented is nonlinear.

System identification procedure is started with determination of parameters to be identified. By careful investigation of six DOF model, it can be concluded that there are twenty three parameters to be studied. These parameters can be grouped as four inertia parameters I_x, I_y, I_z, I_{xz}, eight linear damping parameters X_u, Y_v, Z_w, K_p, M_q, N_r, Y_r, N_v, six nonlinear damping parameters X_{uu}, Y_{vv}, Z_{ww}, K_{pp}, M_{qq}, N_{rr}, four added mass parameters $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{r}}$ and one air drag parameter C_{ad}. The number of remaining parameters are four. These are I_{yz}, I_{xy}, Y_r, N_v, namely two inertia and two coupled damping parameters. Due to xz plane symmetry of the ship, inertia parameters are taken to be zero and coupled damping parameters mentioned are considered as zero in order to reduce the complexity. With this assumption, there are no parameters left to be identified. The next step is about deciding the procedure to find these parameters. Literature survey for system identification of unmanned sea surface vessels is given in Chapter 1. From this literature survey and by careful consideration of available resources, experiments below are to be performed.

- Free motion tests for roll, pitch, yaw and heave movements,
- Thrust measurement tests,
- Linear motion tests,
- Spiral maneuver tests,
- Zigzag maneuver tests.

The parameters that can be determined from above experiments are observed from the governing differential equations of the model. In order to obtain these differential equations, six DOF mathematical model is utilized. This model is rewritten by using symbolic tools in MATLAB. Known constant terms are substituted into the model and unknown parameters and states are kept as symbolic. Then, by using combined (backward-forward) Euler method, derivatives of the states mentioned in mathematical modeling section are obtained. Differential equations are in the form of (3.1).

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{P}) \tag{3.1}$$

where $\boldsymbol{x}$ is state vector and $\mathbf{P}$ are parameters to be identified. Details of (3.1) will be given in Section 3.5.

After obtaining specific parameters for experiments, measurement data should be processed in order to obtain more accurate data. This processing includes filtering, INS-GPS integration, sample point adjustment and point to point transformation of data. Details of the study will be given in Section 3.3. Next step is to decide the type of the optimization algorithms and the structure of cost functions for system identification purposes. Considering that global solution is required to be found and there are smooth cost functions and constraints, genetic and global search algorithms are decided to be used. More detailed information about optimization algorithms will be discussed in Section 3.4. Furthermore, cost functions are specified based on the experiment type, however, general structure of the cost function is as in Equation (3.2).

$$CostFunc = \sum_{i=1}^{N_{var}} \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(x_{i,sim} - x_{i,test}\right)^2}{N_{point}}} \right) \tag{3.2}$$

This kind of cost uses Euclidean metric and it is a good measure in the sense that different lengths of test and simulation results can be easily compared for the same cost function. It is a least squares type of cost function for the error between simulation and test results. Exact cost function representation for the specific test will be given in Section 3.5, however, for this section, it can be said that states that are in the cost function are linear and angular positions. Positions are selected for minimization because it is the end product of the simulations.

Another point that should be considered is the determination of initial values or ranges of the parameters. For inertial parameters, 3D technical drawing of the model ship is utilized to predict the rough values of them. By specifying the corresponding densities and volumes of the materials placed in the model ship, CAD program can easily find the values. Furthermore, the calculation of the estimated values given in Chapter 2 are utilized for added mass parameters. Damping parameters are very difficult to estimate, as a result, a wide initial range is specified considering that optimization algorithm can found the global optimal solution within this range.

Validation is an important sub-part of the system identification procedure. Experiment numbers are adjusted such that after finding optimal solutions, these parameters are tested on the fresh data. However, if there is no fresh data due to loss issues, the optimal parameters are verified in the random old data. Final optimal parameter values and final cost function values are calculated based on normalized cost function values of each test. Equations (3.3) and (3.4) show mentioned calculations.

$$AvgOptParams = \sum_{i=1}^{N_{exp}} \left(\frac{\mathbf{P_i}}{NCFV_i} \right) \tag{3.3}$$

$$AvgCFV = \sum_{i=1}^{N_{exp}} \left(\frac{CFV_i}{NCFV_i} \right) \tag{3.4}$$

where $\mathbf{P_i}$ is the vector containing parameters, $NCFV_i$ is the normalized cost function value and CFV_i is the cost function value for each set.

3.3 Processing of Measurement Data

Raw sensor measurement values are collected from the autopilot card. These values should be processed in order to obtain required accurate data. The processing stages are given in order as in below:

- Navigation work
- Sample time adjustment
- Operation point transformation

Different types of sensors are included in the autopilot card. Sensors utilized in this study are a gyroscope, an accelerometer, a GPS and a magnetometer. Raw measurement data is obtained from these sensors and this information needs to be processed to get more accurate and noiseless data set. For this purpose, three different navigation architectures are developed. These architectures utilize the INS, GNSS and external attitude estimation systems. First architecture integrates GNSS and distinct INS solution in an open loop manner. In this architecture, external attitude estimation is not included. The basic difference between the first and second architecture is that second one implements the algorithms in a closed loop fashion. The last architecture is the improved model of the second one, namely it uses attitude estimation measurement.

Considering position, attitude and velocity RMS errors, loosely coupled, closed loop architecture shows the best performance and ,as a result, measurement data is decided to be processed by this algorithm. The details of this study is given in [47] and it is a parallel independent study utilizing same experimental setup.

Next step is the adjustment of the sample time for the outputs of the navigation algorithm. This output has an average of 200 Hz sampling frequency but it is not constant. Since the sampling frequency of discrete mathematical model is 100 Hz, output should be adjusted to this frequency. If this adjustment is not made, cost functions can not be used. This process carried out by weighted averaging of output considering relative distances of sample points. Details of the study is not given here.

Final stage is the operating point transformation. It is known that entire mathematical modeling is performed in CG, on the other hand, whole sensor information is collected at a point different from CG, lets call this point CO. Hence, sensor information should be transformed to point CG in order to carry out system identification work. This process is carried out by system transformation matrix $H(r_p^b)$ given in [1]. Observation of this matrix reveals that angular states does not change with this transformation and linear states changes only proportional to the distance between CO and CG. Considering this information and the fact that the norm of the vector from CO to CG is about 170 mm, this transformation is not included in this study, however, this investigation is required for proper progression. Before continue, it should also be noted that, for experiments, the useful measurement data is picked from the long logging sessions.

3.4 Optimization Algorithms

As mentioned previously two global search optimization algorithms are selected for this study. These are genetic algorithm (GA) and global search algorithm (GS). Brief descriptions of them are given in the following subsections.

3.4.1 Genetic Algorithm

Genetic algorithm is a generic optimization algorithm that can be utilized to solve both constraint and unconstrained optimization problems. Basic principle of it is the natural selection learned from biological evolution. At each step known as generations, GA creates a population based on selection, crossover and mutation rules. Selection rules choose individuals named as parents that contribute to the population at next step. Crossover rules determine the procedure for combining two parents to create a child. Mutation rules performs random changes to individuals in order to increase the population diversity. By applying these rules, each generation evolves towards an optimal solution [48].

Genetic algorithm can be applied to variety of optimization problems that may not be suitable for many standard optimization algorithm. For example, problems with non-smooth, highly non-linear and stochastic nature can be solved very easily with genetic algorithm [48].

For this study, MATLAB's standard "ga" algorithm in the "Global Optimization Toolbox" is selected due to flexibility reasons. Careful investigation of algorithm combined with the trial and error method results in the following parameters selected for GA. Other parameters are set to default values. Details of the algorithm and parameters can be found in [48].

Table 3.1: GA parameters

Population Size	30
Elite Count	10% of population
Crossover Fraction	0.7
Lower Bound Constraint	depends on experiment
Upper Bound Constraints	depends on experiment
Generation Limit	100
Time Limit	1200
Mutation	Adaptive
Creation	Uniform

One example for convergence of GA is provided in Figure 3.1,the figure shows best and mean values of generations, population diversity, population content and stopping criteria.

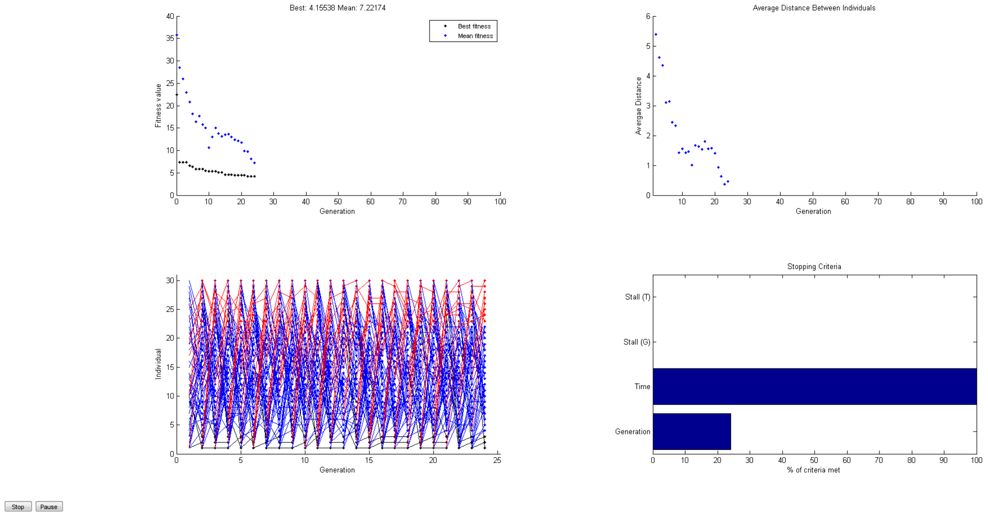


Figure 3.1: GA convergence example

3.4.2 Global Search Algorithm

Global search algorithm is the primary recommendation of MATLAB, if one wants to find global minimum on a single processor for smooth cost functions. Global search algorithm uses a local solver of MATLAB known as *fmincon*. It creates trial points after running from a initial start point by using "Scatter Search Algorithm". Then it runs from best start point among generated trial points. Next step is to loop through remaining trail points and to run a local search algorithm for them. If the remaining trail points satisfy basin, score and constraint requirements, it creates a global solution vector [49]. Careful consideration of the algorithm and some trials give the parameter set below for GS. Other parameters are set to default values.

3.5 System Identification Experiments

Experiments are conducted in two different test areas. Roll, pitch, yaw, heave free motion and thrust measurement tests are carried out in an experimental pool in Aselsan Inc. Linear, spiral and zigzag motion tests(maneuvering tests) are performed in the

Table 3.2: GS parameters

Local Solver	fmincon
Local Solver Algorithm	interior-point
Lower Bound Constraint	depends on experiment
Upper Bound Constraints	depends on experiment
Time Limit	1200

scientific research facility of Aselsan Inc. in Yalıncak Lake, METU. Experiments are completely designed and conducted in an open loop fashion. In other words, PWM inputs are provided to the left and right thrusters. The reasons behind this are that, at this stage, controller implementation is not carried out and open-loop experiments are more easy to design with an experienced user involved in the process.

3.5.1 Free Motion Tests

For free motion tests, the following procedure is applied. It has similar characteristics as the procedure given in [36].

- Model sea surface vehicle is placed at the middle of the pool.
- Using external inclinometer and digital calliper, variety of initial conditions are provided.
- From external computer, data acquisition is started and measurement data logged into an SD card placed in the autopilot module.
- When the vehicle reaches stability, data acquisition is stopped from external computer.

For all optimal parameter calculations, GA is utilized due to sufficient performance characteristics. Results given in the following sections should be evaluated from this point of view.

3.5.1.1 Roll Motion Test

Initial representation of the experiment is given in Figure 3.2.

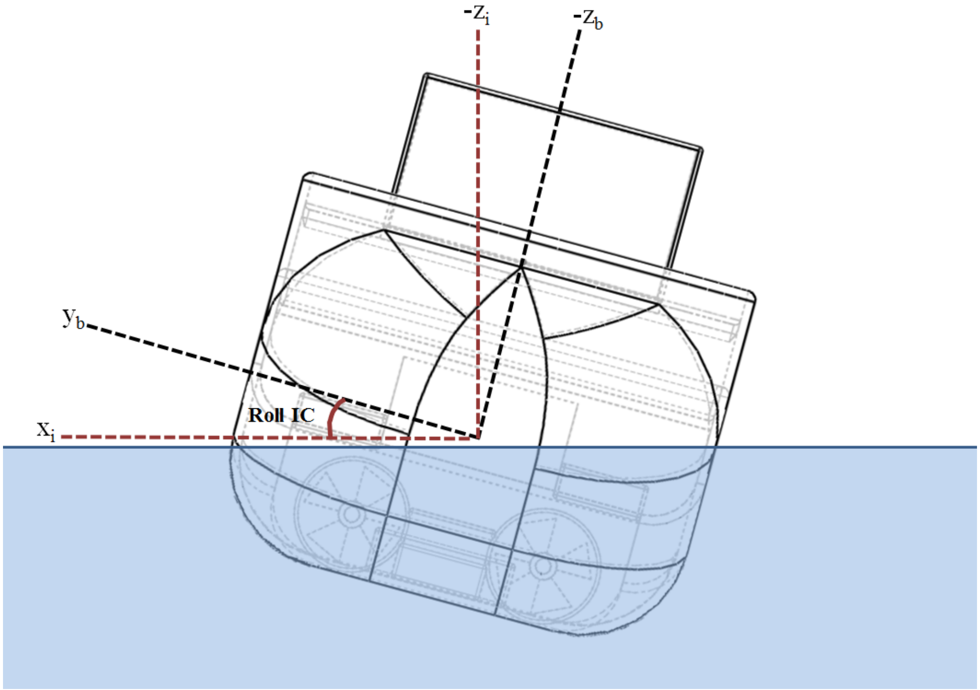


Figure 3.2: Roll motion initial representation

Considering free roll motion of the vehicle from predetermined initial condition and neglecting surge, sway, heave, pitch and yaw motions, decoupled equations below are obtained.

$$\begin{aligned}\dot{p} =& f_a(I_x, I_z, I_{xz}, Y_{\dot{v}}, N_{\dot{r}}, Y_{\dot{r}})\phi + \\ & f_b(I_x, I_z, I_{xz}, Y_{\dot{v}}, N_{\dot{r}}, Y_{\dot{r}}, K_p)p + \\ & f_c(I_x, I_z, I_{xz}, Y_{\dot{v}}, N_{\dot{r}}, Y_{\dot{r}}, K_{pp})|p|p\end{aligned} \tag{3.5}$$

$$\dot{\phi} = p \tag{3.6}$$

As can be seen from Equations (3.5) and (3.6), effective parameters in decoupled roll motion are I_x, I_z, I_{xz}, $Y_{\dot{v}}$, $N_{\dot{r}}$, $Y_{\dot{r}}$, K_p, K_{pp}.

Selected cost function can be seen from Equation (3.7).

$$RollCostFunc = \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} (\phi_{i,sim} - \phi_{i,test})^2}{N_{point}}} \right) \tag{3.7}$$

There are four roll free motion tests conducted. Three of them are utilized for identification and the remaining one is used for validation. Example for the convergence of simulation data with the optimized parameters are given in Figure 3.3.

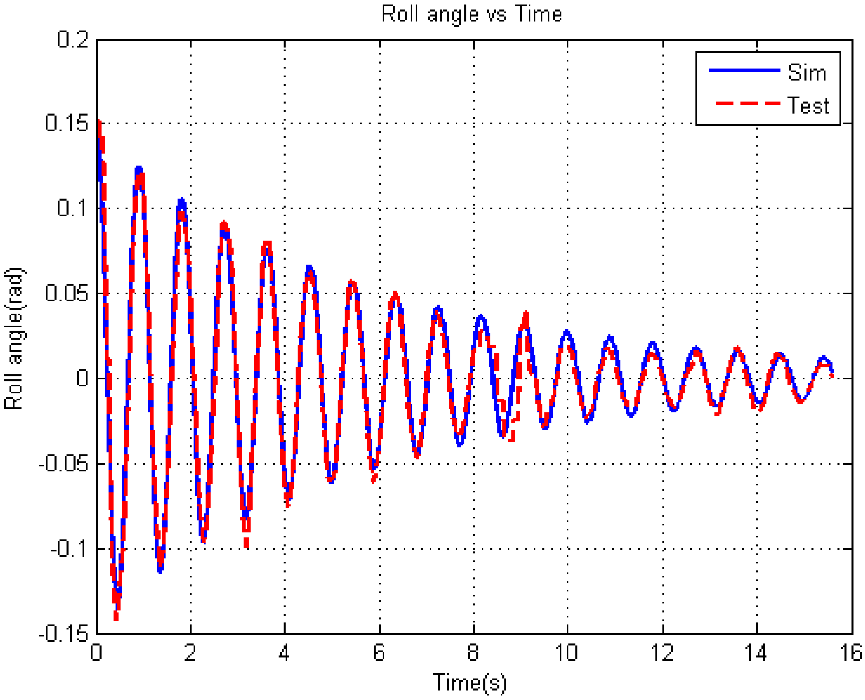


Figure 3.3: Example of roll convergence

Average optimized parameters, cost function and validated cost function values are given in Table 3.3. It should be noted that it is logical to have negative damping and added mass parameters. These results are consistent with the model structure explained in Chapter 2.

For validation test, simulation and test results are given in Figure 3.4.

Table 3.3: Roll free motion results

I_x	0.12164
I_z	0.59988
I_{xz}	0.03107
$Y_{\dot{v}}$	-2.29440
$N_{\dot{r}}$	-0.06237
$Y_{\dot{r}}$	-2.54660
K_p	-0.02807
K_{pp}	-0.04382
Final Cost Function Value	0.01310
Validated Cost Function Value	0.03411

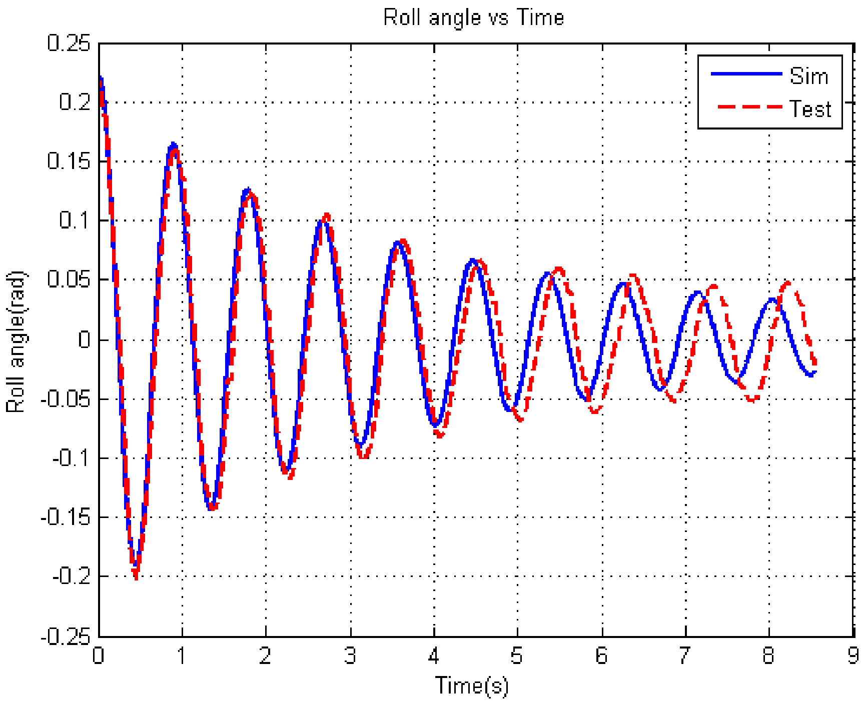


Figure 3.4: Validation result for roll motion

3.5.1.2 Pitch Motion Test

Initial representation of the experiment can be seen from Figure 3.5.

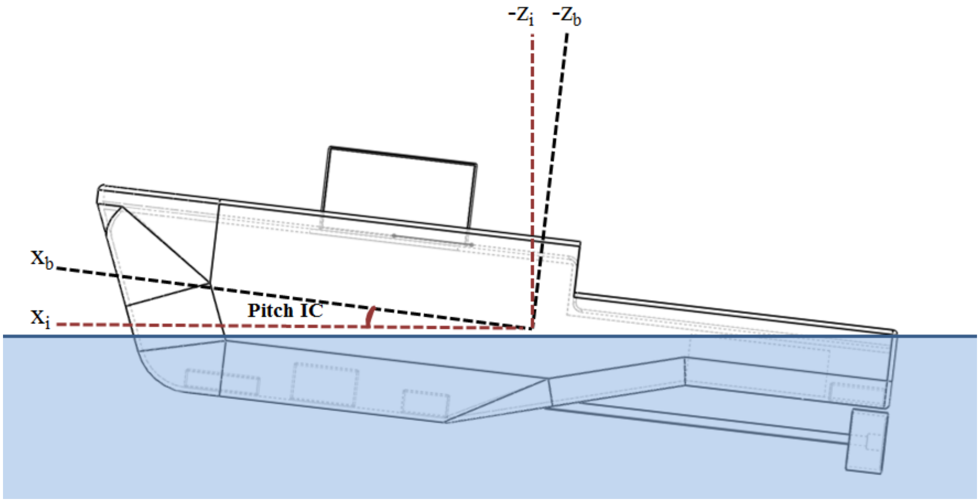


Figure 3.5: Pitch motion initial representation

Taking into account the free pitch motion of the vehicle from predetermined initial condition and neglecting surge, sway, heave, roll and yaw motions, decoupled equations below are obtained.

$$\dot{q} = f_a(M_q, I_y)q + f_b(I_y)\theta + f_c(M_{qq}, I_y)|q|q \tag{3.8}$$

$$\dot{\theta} = q \tag{3.9}$$

As can be seen from Equations (3.8) and (3.9), effective parameters in the decoupled pitch motion are I_y, M_q, M_{qq}.

Utilized cost function for the experiment can be seen from Equation (3.10).

$$PitchCostFunc = \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(\theta_{i,sim} - \theta_{i,test}\right)^2}{N_{point}}} \right) \tag{3.10}$$

There are three pitch free motion tests. All of them are used for identification and randomly selected one is analyzed for validation. Example for the convergence of simulation data with the optimized parameters are given in Figure 3.6.

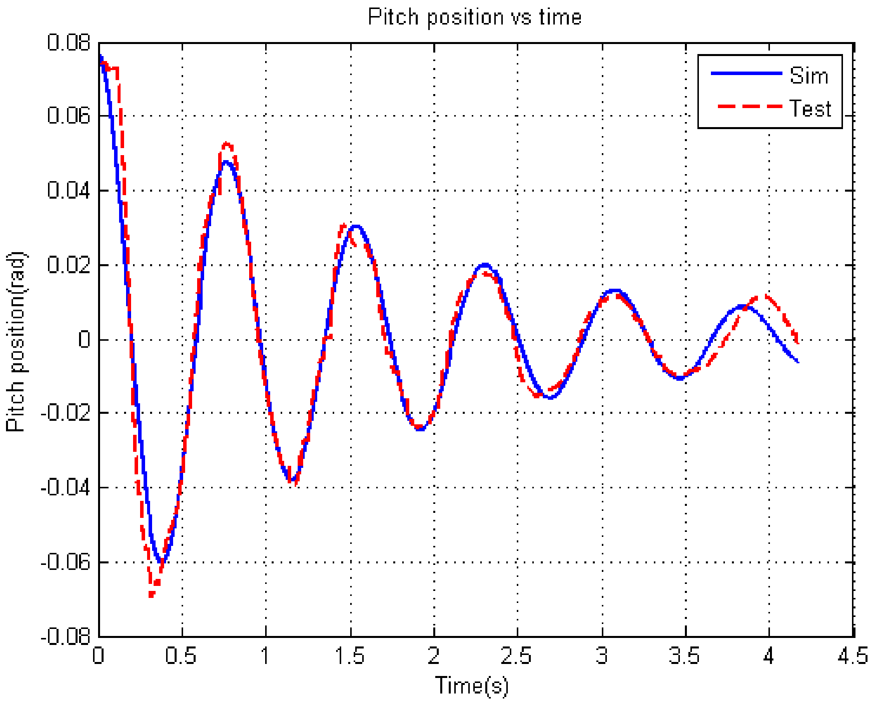


Figure 3.6: Example of pitch convergence

Average optimized parameters, cost function and validated cost function values are given in Table 3.4.

Table 3.4: Pitch free motion results

I_y	2.14421
M_q	-2.43171
M_{qq}	-0.76863
Final Cost Function Value	0.00906
Validated Cost Function Value	0.01230

Simulation and test results for validation test are given in Figure 3.7.

3.5.1.3 Yaw Motion Test

Initial representation of the experiment is given in Figure 3.8.

Unlike pitch and roll motion tests, in this experiment, small torques are applied to the vehicle in order to provide initial speeds. After torque is released, vehicle starts

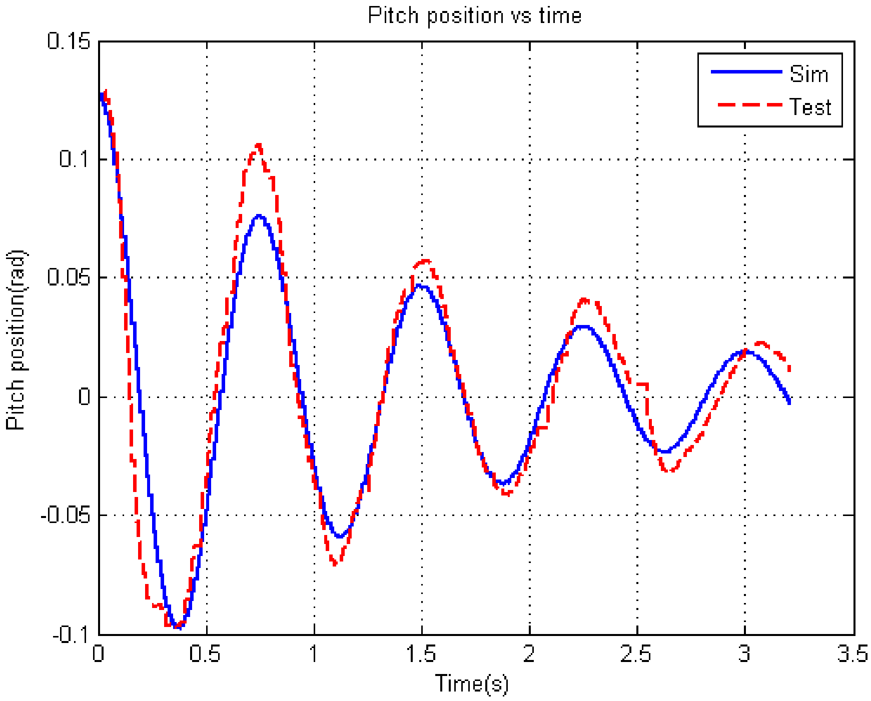


Figure 3.7: Validation result for pitch motion

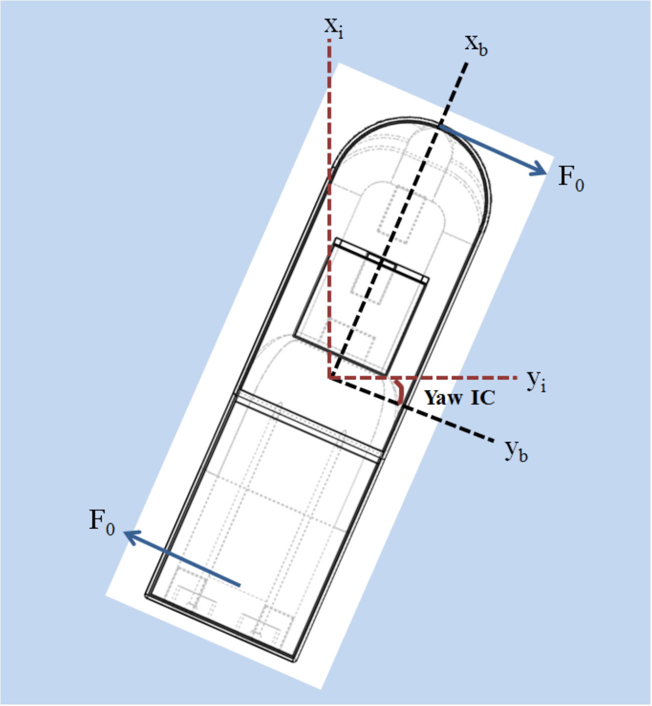


Figure 3.8: Yaw motion initial representation

to slow down in free motion and, at some point, it stops. This free motion window is considered in the calculations. Taking into account the yaw motion of the vehicle from predetermined initial condition with mentioned procedure and neglecting surge, sway, heave, roll and pitch motions, decoupled equations below are obtained.

$$\begin{aligned}\dot{r} =& f_a(I_x, I_z, I_{xz}, Y_{\dot{v}}, N_{\dot{r}}, Y_{\dot{r}}, N_r, Y_r)r+ \\ & f_b(I_x, I_z, I_{xz}, Y_{\dot{v}}, N_{\dot{r}}, Y_{\dot{r}}, N_{rr})|r|r\end{aligned} \tag{3.11}$$

$$\dot{\psi} = r \tag{3.12}$$

As can be seen from Equations (3.11) and (3.12), effective parameters in decoupled yaw motion are I_x, I_z, I_{xz}, $Y_{\dot{v}}$, $N_{\dot{r}}$, $Y_{\dot{r}}$, N_r, Y_r, N_{rr}.

Selected cost function for the experiment can be found in Equation (3.13).

$$YawCostFunc = \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(\psi_{i,sim} - \psi_{i,test}\right)^2}{N_{point}}} \right) \tag{3.13}$$

Four yaw free motion tests are conducted, three of them is for identification and one of them is for validation. Example for the convergence of simulation data with the optimized parameters is given in Figure 3.9.

Average optimized parameters, cost function and validated cost function values can be seen in Table 3.5.

Table 3.5: Yaw free motion results

I_x	0.058413
I_z	0.84536
I_{xz}	0.01881
$Y_{\dot{v}}$	-2.13782
$N_{\dot{r}}$	-0.03478
$Y_{\dot{r}}$	-1.95838
N_r	-0.18514
N_{rr}	-0.42855
Final Cost Function Value	0.01596
Validated Cost Function Value	0.17830

Simulation and test results for validation test can be found in Figure 3.10.

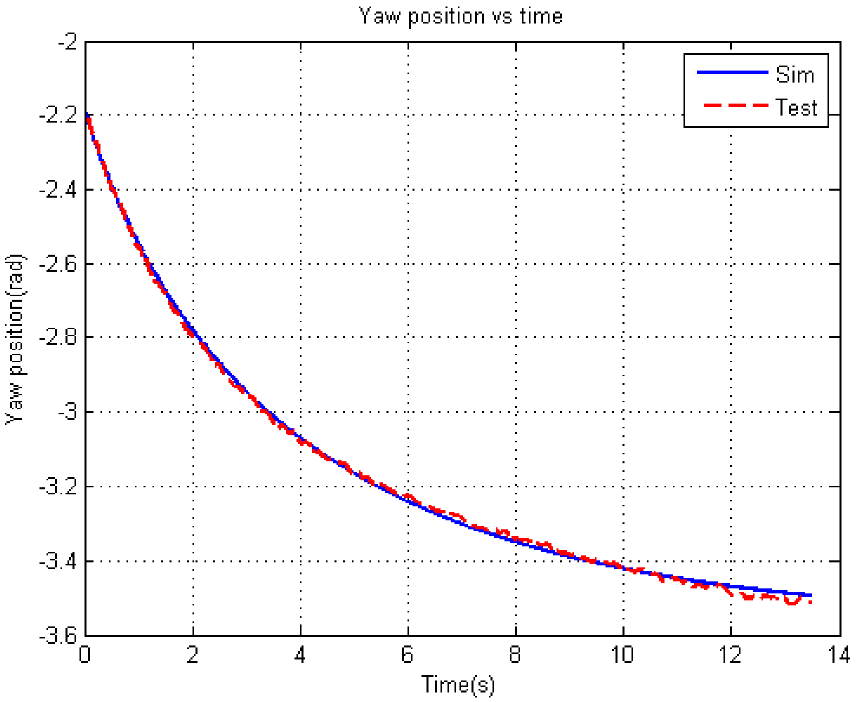


Figure 3.9: Example of yaw convergence

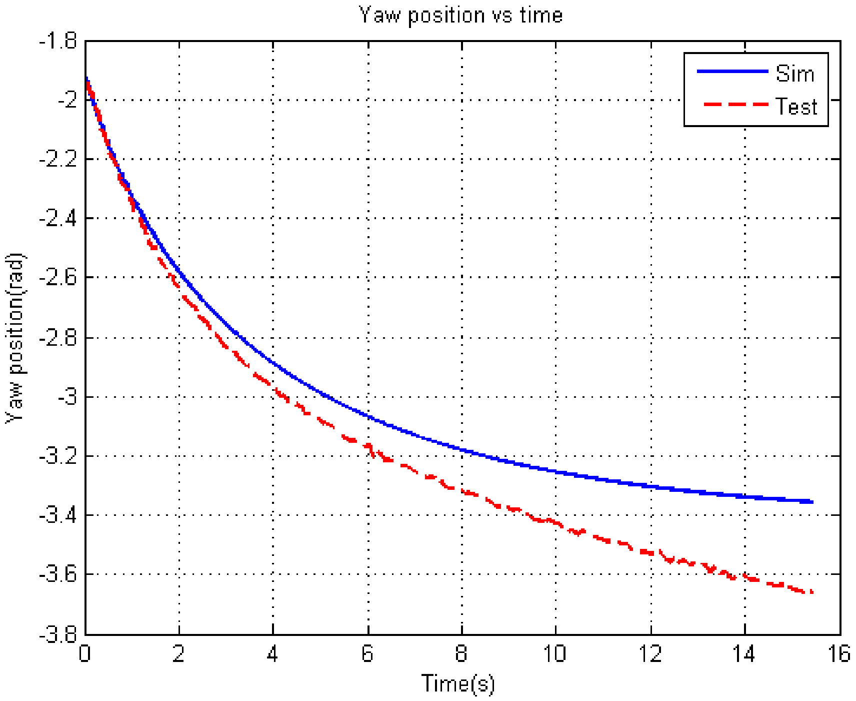


Figure 3.10: Validation result for yaw motion

3.5.1.4 Heave Motion Test

Figure 3.11 demonstrates initial representation of the test.

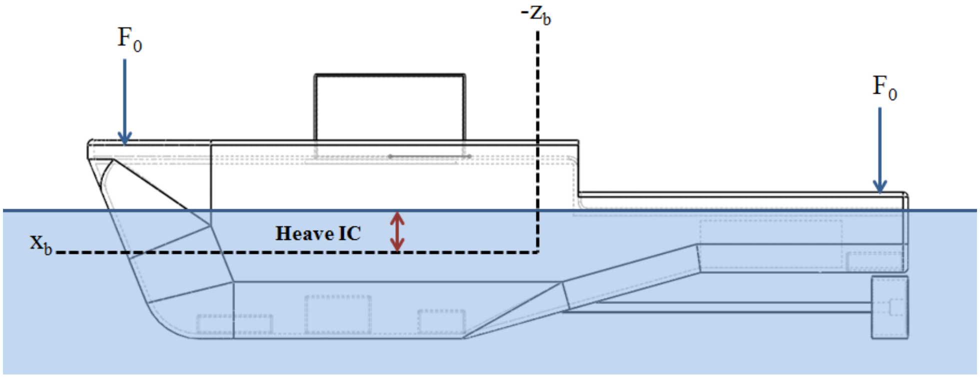


Figure 3.11: Heave motion initial representation

Similar to yaw motion test, there are also initial forces in heave test. These forces are applied to the vehicle such that no torque is created. After measuring the initial displacement from the stable equilibrium point with a digital calliper, the force acting on the vehicle is zeroed. Then, damped motion is logged. Taking into account the heave motion of the vehicle from predetermined initial condition with mentioned procedure and neglecting surge, sway, roll, pitch and yaw motions, decoupled equations below are obtained.

$$\dot{w} = f_a(Z_w)w + f_b(Z_{ww})|w|w \tag{3.14}$$

$$\dot{z} = w \tag{3.15}$$

As can be deduced from Equations (3.14) and (3.15), effective parameters in decoupled heave motion are Z_w, Z_{ww}.

When measured data is examined, it is seen that the accuracy of GPS data for heave is very low. Hence, these parameters are manually tuned considering the damping time using the dynamical model. Optimized values are given in Table 3.6.

Table 3.6: Heave free motion results

Z_w	-1
Z_{ww}	-15

Before ending this subsection, one point should be made explicit. There are some shared parameters determined from free motion tests. These parameters are I_x, I_z, ,I_{xz}, $Y_{\dot{r}}$, $Y_{\dot{v}}$, $N_{\dot{r}}$ and they are common in yaw and roll free motion tests. Final optimized values are taken as the average of them.

3.5.2 Thrust Measurement Test

In Chapter 2, thruster forces are assumed to be known. However, in order to analyze the experiments with thrusters, there is a need for obtaining the relation between applied motor voltage and thrust. Instead of dealing with the complex modeling characteristics of thruster, it is decided to obtain this relationship as a lookup table and to use this table in the tests with thrusters. The experimental setup is constructed as in the Figure 3.12. During experiment the vehicle is kept stationary. This test is inspired from [50].

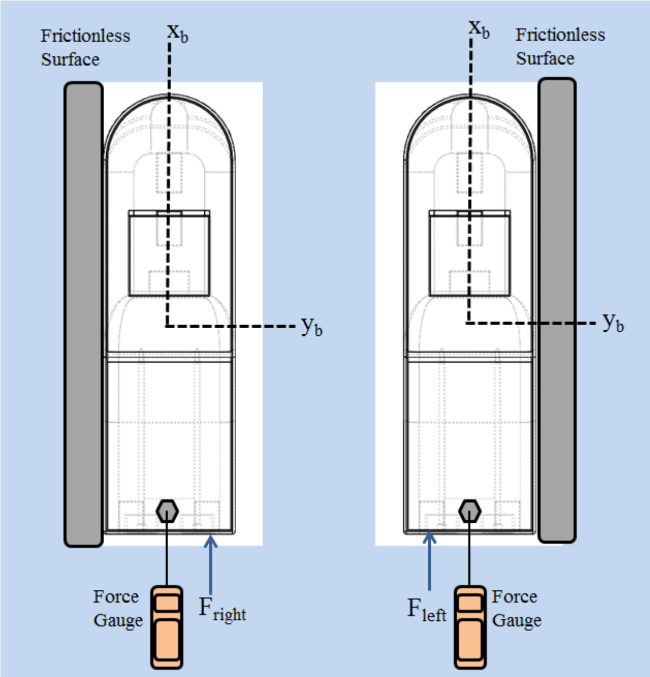


Figure 3.12: Thruster test representation

A digital force gauge is utilized to measure the linear force created by the single thruster. Torque generated by the single thruster is assumed to be eliminated by friction-less surface. Another assumption is that friction-less surface does not create any linear force in the reverse thrust direction. With these assumptions are in mind, different PWM values for the specific and measured battery voltage are applied to the thrusters separately and thrust values are recorded. The recorded PWM values are evaluated with battery voltage and for both thrusters, voltage versus thrust figures are obtained. Considering that, in the test that thrusters work, both battery voltage values and PWM values are recorded, corresponding voltage values applied to motors can be found and input to the model. For left and right thrusters, linearly interpolated thrust-voltage relation is given in the Figure 3.13.

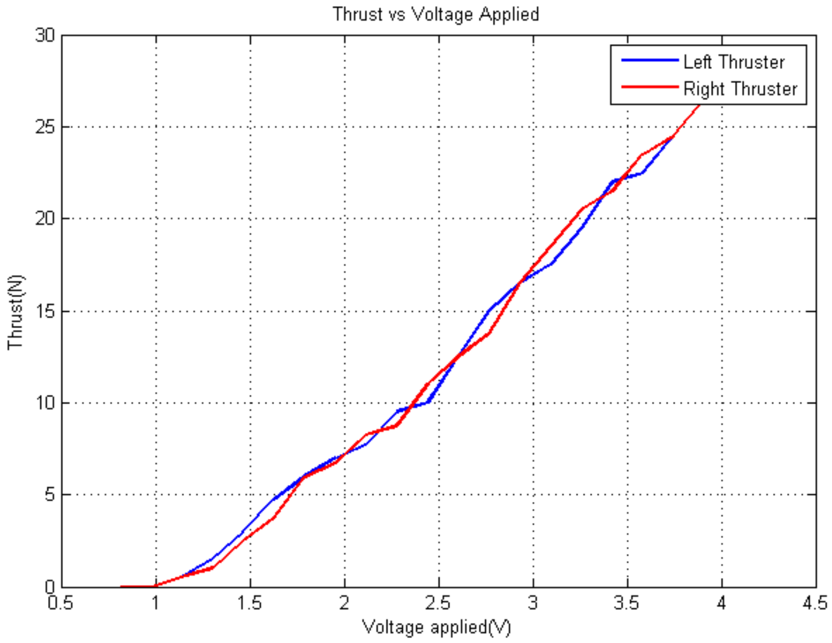


Figure 3.13: Thrust-voltage relation

The experiments show that, battery voltage variation during the experiment cannot be recorded due to data loss during experiment. The expected situation is that battery voltage slightly falls during tests. Furthermore, it is seen that accuracy of linear force gauge is "1 N". As a result, it is decided that for both left and right thrusters, two parameters are included for the voltage thrust lookup table. It is assumed that there can be maximum amount of 5% offset and 10% line scale up in the look-up tables. These four parameters are also included in the list of parameters to be optimized. These are named as L_{TO}, L_{TS}, ,R_{TO}, R_{TS} where L means left, TO and TS are thruster offset and thruster scale.

3.5.3 Tests with Thrusters

These tests can be grouped into three categories. Using thrusters in an open-loop fashion, vehicle is forced to follow linear, spiral and zigzag paths. It should be noted that, the best results among GA and GS are selected for future investigation. This note should be kept in mind for the tests with thruster. Considering general characteristics, GS is shown to be superior to GA in the same time limit. In Section 3.5.3.2, one result comparing GA and GS is also given. In Section 3.6, results are discussed.

3.5.3.1 Linear Motion Test

Linear motion test is also considered as a decoupled motion test. Namely, sway, heave, roll, pitch and yaw motions assumed to be neglected. Keeping this assumption in mind, governing differential equations can be obtained as in Equations (3.16) and (3.17)

$$\begin{aligned}\dot{u} =& f_a(X_u, X_{\dot{u}})u + f_b(X_{uu}, X_{\dot{u}}, C_{DA})|u|u + \\ & f_c(X_{\dot{u}})F_{left} + f_d(X_{\dot{u}})F_{right}\end{aligned} \tag{3.16}$$

$$\dot{x} = u\cos(\psi) \tag{3.17}$$

As can be seen from equations, effective parameters for linear motion are $X_{\dot{u}}$, X_u, X_{uu}, C_{DA}.

The cost function for the experiment can be found in Equation (3.18).

$$SurgeLinCostFunc = \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} (x_{i,sim} - x_{i,test})^2}{N_{point}}}\right) + \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} (y_{i,sim} - y_{i,test})^2}{N_{point}}}\right) \tag{3.18}$$

The number of linear motion experiments is three. Validation is carried out in a randomly selected from these three. Example for the convergence of simulation data with the optimized parameters are given in Figures 3.14 and 3.15.

Average optimized parameters, cost function and validated cost function values can be found in Table 3.7.

Simulation and test results for validation test can be found in Figures 3.16 and 3.17 .

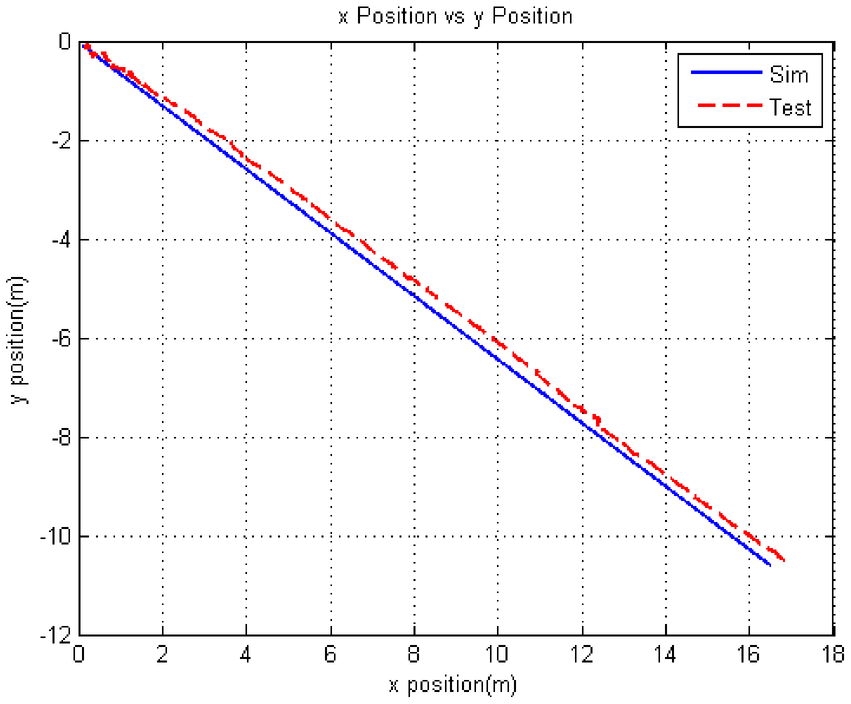


Figure 3.14: Example of surge convergence

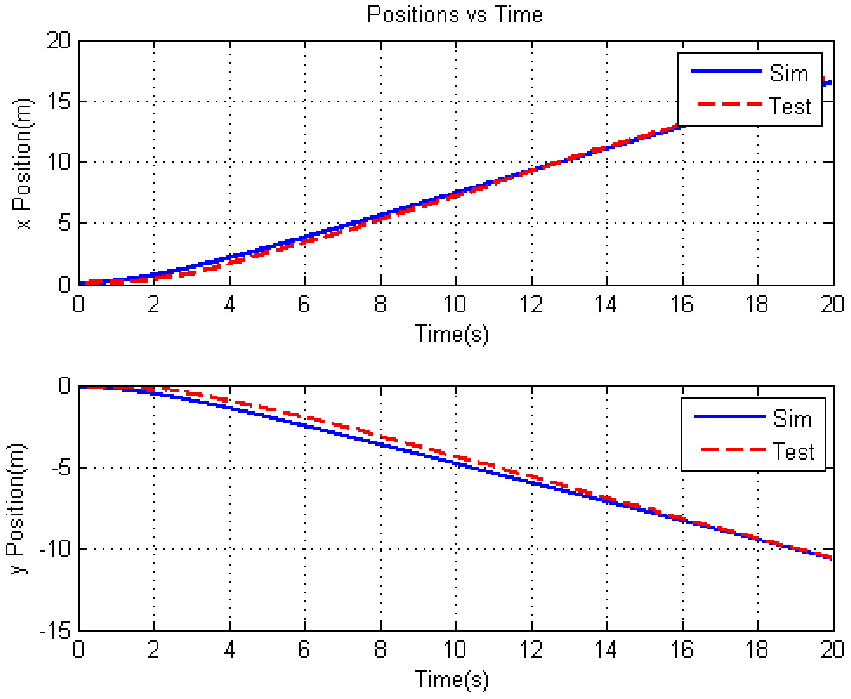


Figure 3.15: Example of surge convergence

Table 3.7: Linear motion results

X_u	-3.67165
X_{uu}	-0.98647
$X_{\dot{u}}$	-1.16633
C_{DA}	1.17930
Final Cost Function Value	1.27294
Validated Cost Function Value	0.74758

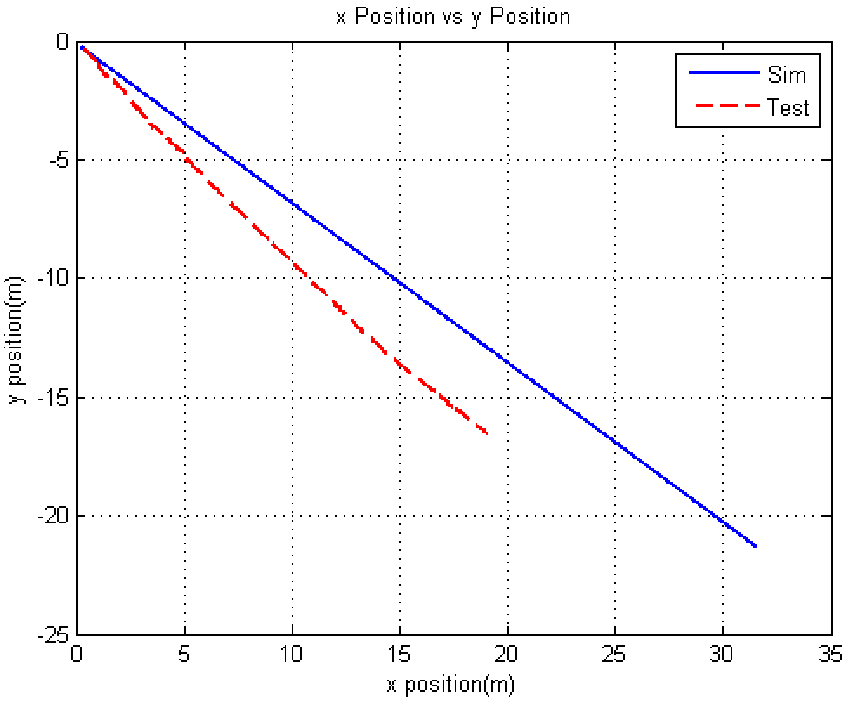


Figure 3.16: Validation result for linear surge motion

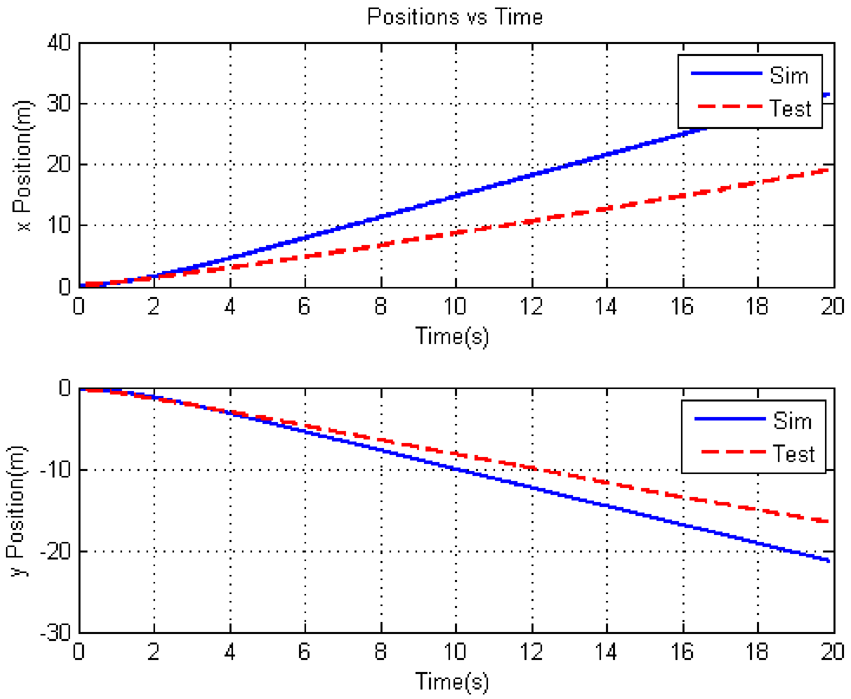


Figure 3.17: Validation result for linear surge motion

3.5.3.2 Spiral Maneuvering Test

This and the following subsection contain the most complicated motion tests. In these tests, it is assumed that surge, sway and yaw motions are decoupled from other motions which are neglected in the governing differential equations. Similar to previous studies, governing differential equations are obtained for coupled three DOF motion. Surge coupled motion equations can be found in Equations (3.19) and (3.20).Furthermore, sway coupled motion equations can be seen from Equations (3.21) and (3.22). Finally, Equations (3.23) and (3.24) shows the yaw coupled equations.

$$\begin{aligned}\dot{u} =&f_a(X_u,X_{\dot{u}})u+f_b(X_{uu},X_{\dot{u}},C_{DA})|u|u+\\ &f_c(N_{\dot{v}},Y_{\dot{r}},X_{\dot{u}})r^2+f_d(Y_{\dot{v}},X_{\dot{u}})rv+\\ &f_e(X_{\dot{u}})F_{left}+f_f(X_{\dot{u}})F_{right}\end{aligned} \tag{3.19}$$

$$\dot{x} = u\cos(\psi) - v\sin(\psi) \tag{3.20}$$

$$
\begin{aligned}
\dot{v} =&f_a(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_r,N_r)r+\\
&f_b(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_v,N_v)v+\\
&f_c(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},X_{\dot{u}})rv+\\
&f_d(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_{vv},C_{DA})|v|v+\\
&f_e(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},X_{\dot{u}})uv+\\
&f_f(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},N_{rr})|r|r
\end{aligned}
\tag{3.21}
$$

$$
\dot{y} = v\cos(\psi) + u\sin(\psi) \tag{3.22}
$$

$$
\begin{aligned}
\dot{\psi} =&f_a(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_r,N_r)r+\\
&f_b(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_v,N_v)v+\\
&f_c(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},X_{\dot{u}})ru+\\
&f_d(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},X_{\dot{u}})uv+\\
&f_f(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},N_{rr})|r|r+\\
&f_g(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}},Y_{vv},C_{DA})|v|v+\\
&f_h(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}})F_{left}+\\
&f_i(I_x,I_z,I_{xz},N_{\dot{r}},Y_{\dot{r}},Y_{\dot{v}})F_{right}+
\end{aligned}
\tag{3.23}
$$

$$
\dot{\psi} = r \tag{3.24}
$$

Combining all equations, parameters to be optimized for coupled maneuvers are I_x, I_z, ,I_{xz}, X_u, X_{uu}, $X_{\dot{u}}$, Y_v, Y_r, Y_{vv}, $Y_{\dot{r}}$, $Y_{\dot{v}}$, N_r, N_v, N_{rr}, $N_{\dot{r}}$, C_{DA}, L_{TO}, L_{TS}, ,R_{TO}, R_{TS}.

Equation (3.25) gives the cost function for the experiment.

$$
SpiralMotCostFunc = \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(x_{i,sim} - x_{i,test}\right)^2}{N_{point}}}\right) + \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(y_{i,sim} - y_{i,test}\right)^2}{N_{point}}}\right) + \left(\sqrt{\frac{\sum_{j=1}^{N_{point}} \left(\psi_{i,sim} - \psi_{i,test}\right)^2}{N_{point}}}\right) \tag{3.25}
$$

Before running the optimization algorithms, initial conditions and upper-lower constraints should be determined from previous tests, such as free motion tests, if the optimization of that parameter exist. If a parameter does not lie in the list of parameters to be optimized in previous tests, a wide initial estimate is made.

There are three spiral maneuvering tests. All of them are utilized for optimization and randomly selected one is used for validation. Example for the convergence of the simulation data with optimized parameters are given in Figures 3.18 and 3.19 for GS and 3.20 and 3.21 for GA. Since GS is superior to GA, GS results are taken.

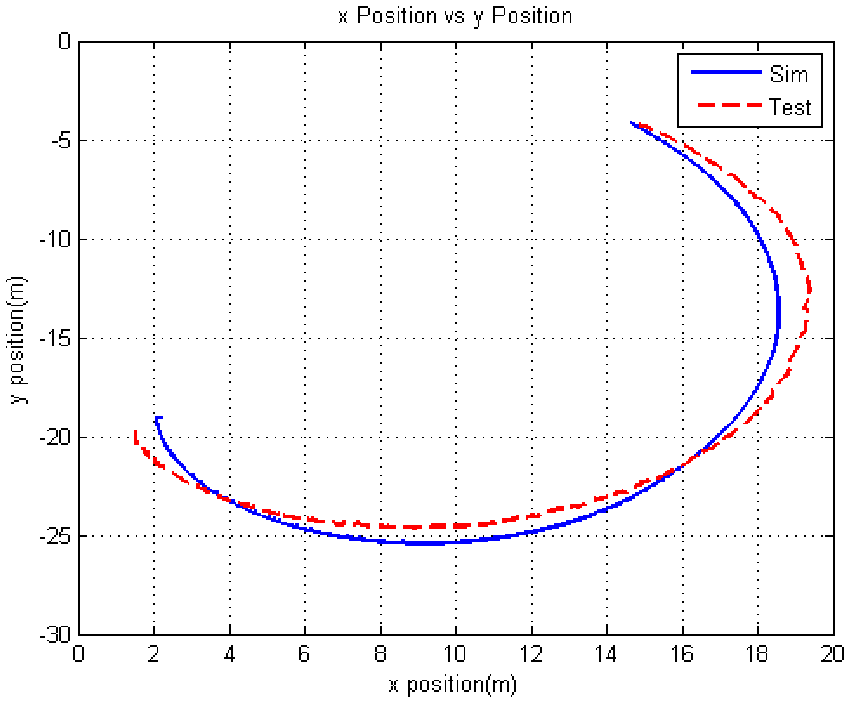


Figure 3.18: Example of spiral convergence for GS

Optimized parameter values and corresponding cost function are given in Table 3.8.

3.5.3.3 Zigzag Maneuvering Test

Theory for zigzag motion test is similar to one for spiral motion. Governing differential equations, cost function values and procedure for determining initial conditions are exactly same.

There are three zigzag maneuvering tests. All of them are utilized for optimization and randomly selected one is used for validation. Example for convergence of simu-

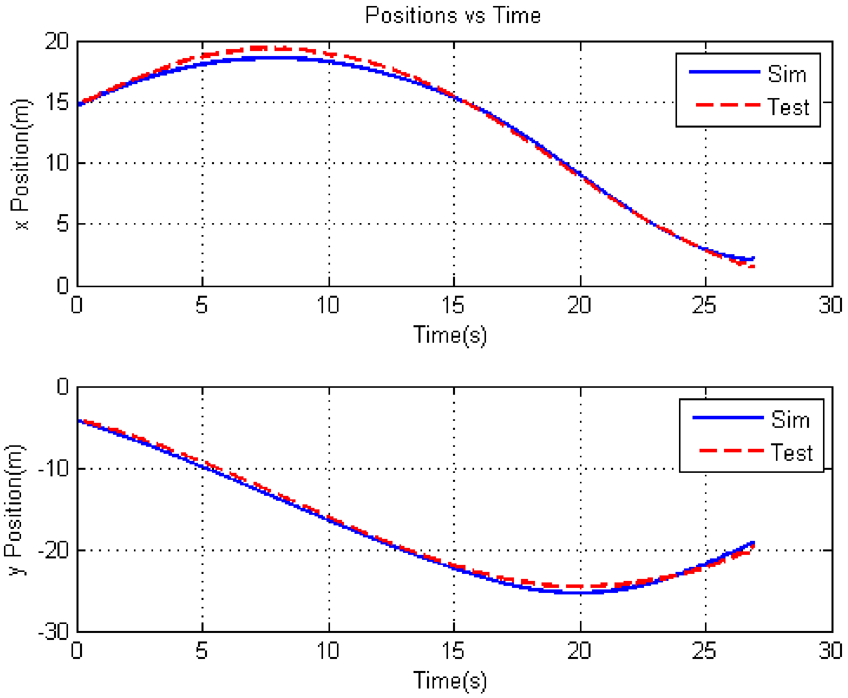


Figure 3.19: Example of spiral convergence for GS

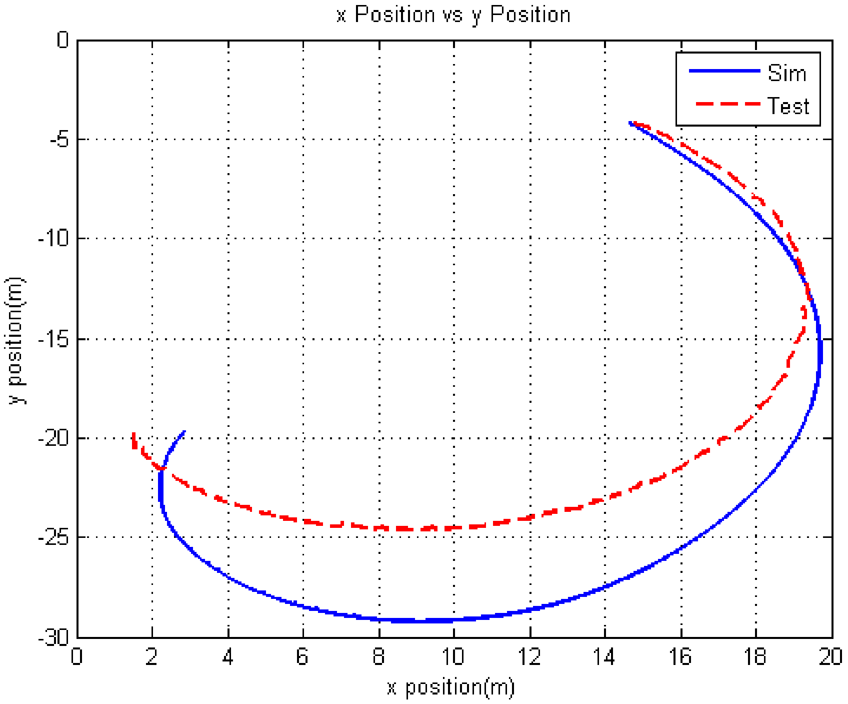


Figure 3.20: Example of spiral convergence for GA

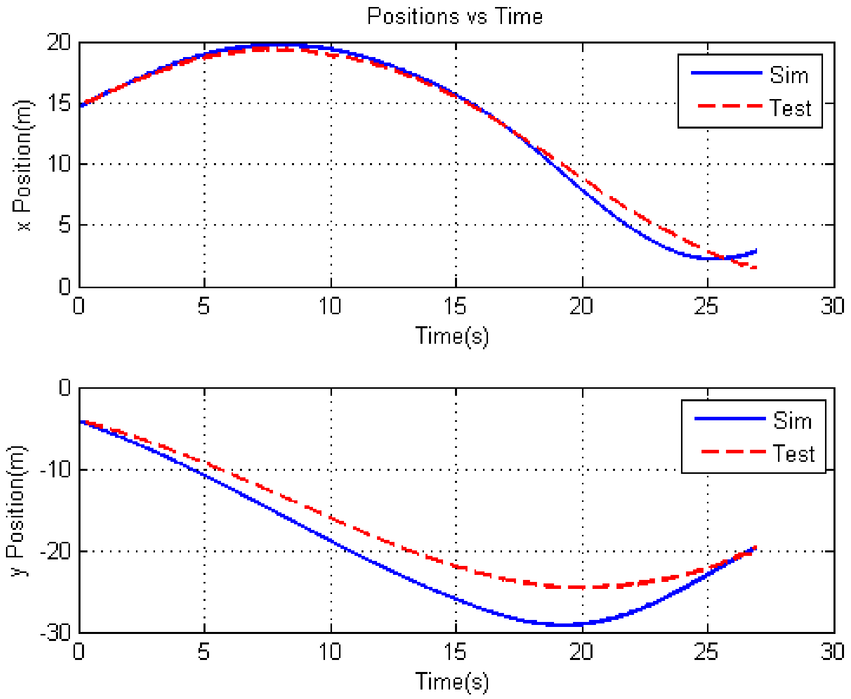


Figure 3.21: Example of spiral convergence for GA

Table 3.8: Spiral motion results

X_u	-5.76909	N_{rr}	-0.33077
X_{uu}	-2.17161	$N_{\dot{r}}$	-0.04531
$X_{\dot{u}}$	-0.87818	I_x	0.08979
Y_v	-3.98659	I_z	0.63994
Y_r	-0.0001	I_{xz}	0.02279
Y_{vv}	-3.95131	C_{DA}	1.2417
$Y_{\dot{r}}$	-1.92760	L_{TO}	-0.33782
$Y_{\dot{v}}$	-1.05279	L_{TS}	0.99086
N_r	-0.12392	R_{TO}	0.33309
N_v	-0.0001	R_{TS}	0.99792
Final Cost Function Value	0.76694	-	-

lation data with optimized parameters are given in Figures 3.22 and 3.23.

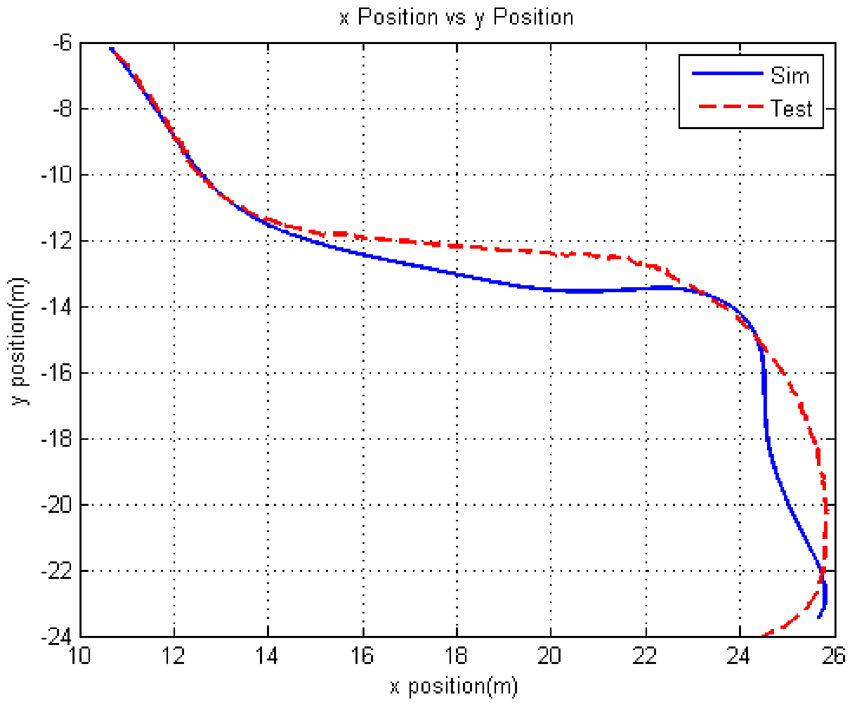


Figure 3.22: Example of zigzag convergence

Optimized parameter values and corresponding cost function are given in Table 3.9.

Table 3.9: Zigzag motion results

X_u	-5.17399	N_{rr}	-0.06316
X_{uu}	-2.71233	$N_{\dot{r}}$	-0.05332
$X_{\dot{u}}$	-1.70245	I_x	0.08812
Y_v	-2.15620	I_z	0.67312
Y_r	-0.0001	I_{xz}	0.02280
Y_{vv}	-3.87410	C_{DA}	1.26030
$Y_{\dot{r}}$	-0.06634	L_{TO}	-0.92010
$Y_{\dot{v}}$	-1.05590	L_{TS}	0.92270
N_r	-0.04762	R_{TO}	0.50610
N_v	-0.0001	R_{TS}	1.06632
Final Cost Function Value	2.15360	-	-

Final values of effective parameters are calculated based on normalized weights of corresponding cost function in spiral and zigzag motion tests. After calculation of final values, these are tested on the two random motion pattern. These results are

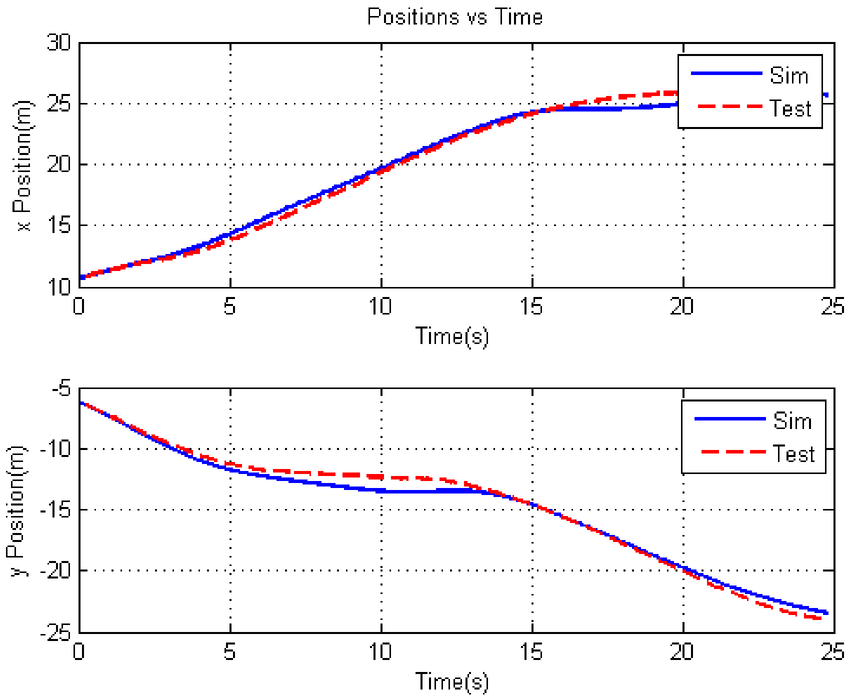


Figure 3.23: Example of zigzag convergence

given in Figures 3.24, 3.25, 3.26 and 3.27.

3.6 Evaluation of Results

As can be seen from the figures of free motion tests, convergences are very successful. Validation tests also reflects this achievement. The main reason behind this situation is that these tests are carried out in a controlled environment. Other reasons maybe that there are no thrust inputs and the mathematical model is flexible enough to fit the free motion characteristics of the vehicle.

Considering tests with thruster, although optimization algorithms converges successfully, it can be seen that validation tests are not very successful. There are several reasons for this divergence. First of all, these test are not conducted in a controlled medium. Disturbance effects of each test are not same. Since the sea surface vehicle is relatively small, it is easily influenced by wind and wind generated currents, etc. Our mathematical model does not consider these disturbance effects since it is difficult

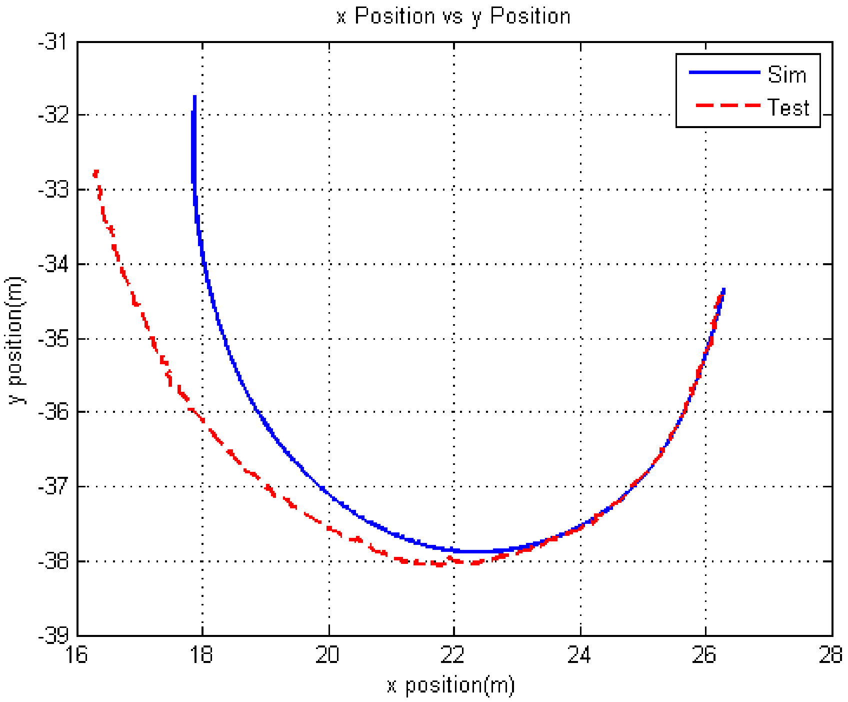


Figure 3.24: Random motion path 1

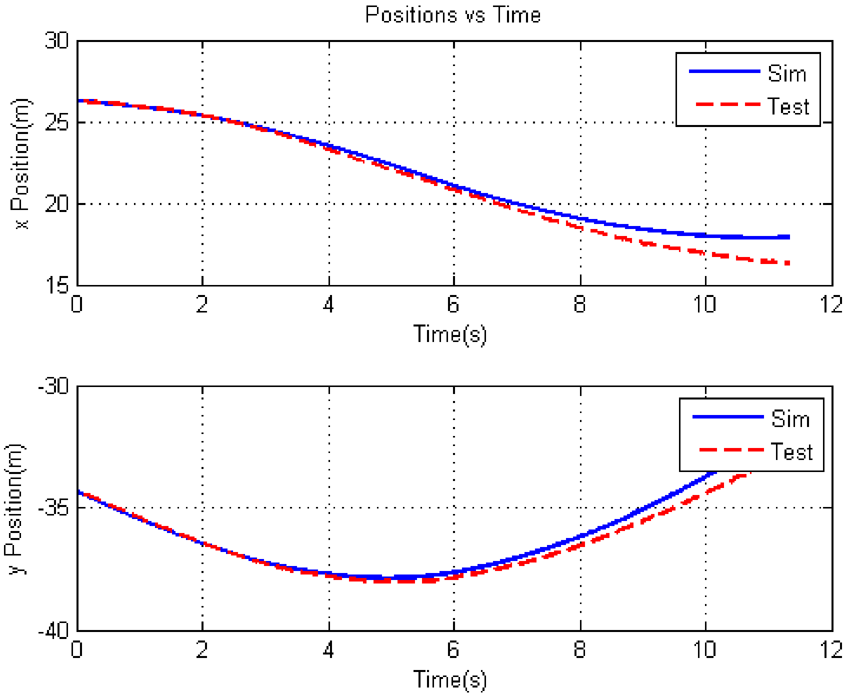


Figure 3.25: Random motion path 1

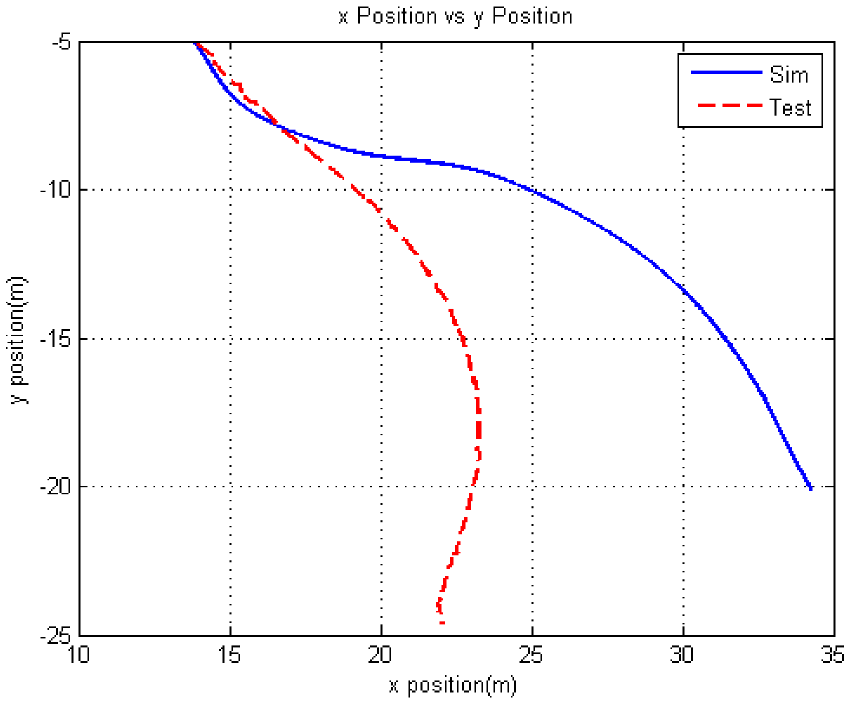


Figure 3.26: Random motion path 2

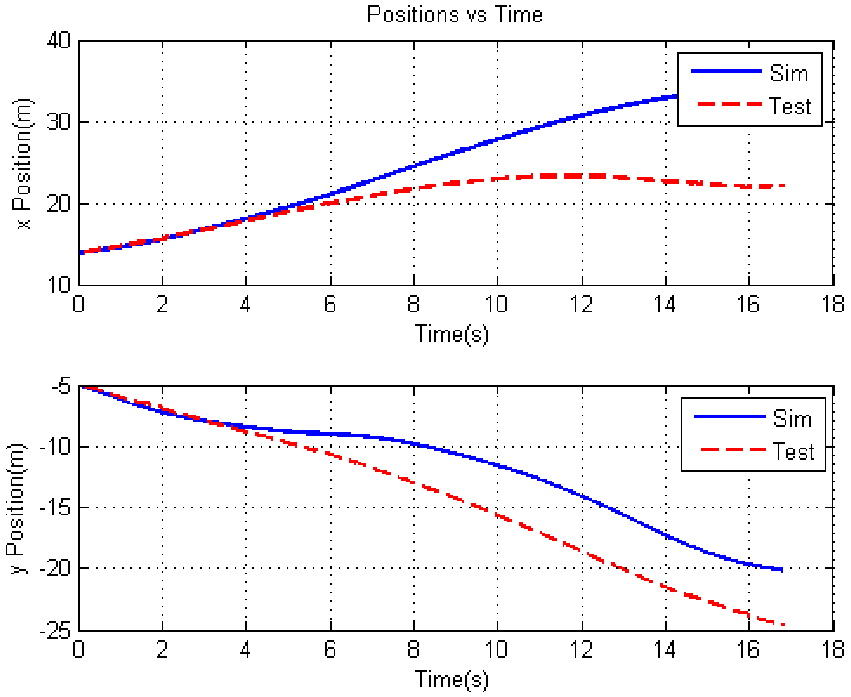


Figure 3.27: Random motion path 2

to measure and implement. Next, loss of battery voltage records collected in thrust measurement tests and insufficient accuracy of force gauge are two significant drawbacks. If this situation does not happen, there would be no need for offset and scaling parameters. Moreover, look-up table approach does not contain valuable information about transient characteristics of thrust. However, in the tests with thruster, it can be seen that transient regions can occur. Furthermore, there maybe some drawbacks of the model such that it can not reflect the full characteristics of the vehicle. Finally, the system identification experiments may not excite all the modes of the boat, hence some parameters may not be effective in that motion.

When the point comes to optimization algorithms, it is seen that GA works sufficiently well on free motion tests. Hence this algorithm is utilized in these tests. However, GS gives better results in the tests with thruster. The main difference between the free motion test and tests with thruster from optimization algorithm point of view is that the number of optimized parameters in tests with thruster is about three-four times of number in free motion tests. It can be said that GA does not converge well with the given diversity in a predetermined time compared to GS. However, if enough time and diversity are provided to GA, results are expected to be similar to that of GS.

CHAPTER 4

CONTROLLER DESIGN

4.1 Introduction

In this chapter, several controllers for the motion control of the sea surface vehicle are investigated. Throughout the design process, the mathematical model of marine vessel which was previously derived and identified for Pacific Islander Tug Boat is used. Proportional-Integral-Derivative (PID) based controller and Sliding Mode Controller (SMC) techniques are basically exploited. As stated in Chapter 2, there are two actuators in the model: left thruster and right thruster. The ultimate goal of the controllers is to reach the desired surge velocity and yaw position which maybe determined by an upper level controller. For this purpose, the ‘correct’ values of torques to be applied by left and right thrusters are calculated.

4.2 PID based Piecewise Controller Design

In this section, first of all, nonlinear mathematical model of the vehicle is linearized at different operating points. Next, controllable and observable subspace of linearized models, namely minimal realization of the models, are found using Kalman Decomposition. In the following part, transfer functions of these models are obtained. Then, controllers based on PID for each linearization point are designed and applied on the linear models. It is the reason why this controller has a piecewise nature. Finally, PID controllers are tested on the nonlinear model and results are obtained, however, results are presented after Sliding Mode Controller section for the existence and absence of environmental disturbances. Results are also compared with the ones of the SMC.

4.2.1 Work on Nonlinear Model

In this section, the nonlinear mathematical model of the vehicle is linearized at different operating points. Then, for each linearization point, minimal realization of the linearized model is found by using Kalman Decomposition. As a next step, corresponding transfer functions for each linearization point are calculated.

4.2.1.1 Linearization

Let's remember the nonlinear model derived on chapter 2, which is given in Equations (4.1) and (4.2).

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu} && (4.1) \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} &= \boldsymbol{\tau_d} + \boldsymbol{\tau_g} + \boldsymbol{\tau_t} + \boldsymbol{\tau_a} + \boldsymbol{\tau_{dis}} && (4.2) \end{aligned}$$

This nonlinear model should be linearized at different operating points in order to be able to design PID type of linear controller. In linearization process, state vector includes the positions and orientations in Earth Fixed Reference Frame and linear and angular velocities in Body Fixed Reference Frame. Input includes the left and right thruster forces. Outputs are the surge speed and yaw position of the vehicle. The desired state space equation is given in Equations (4.3) and (4.4).

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} &= \boldsymbol{A} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\nu} \end{bmatrix} + \boldsymbol{B}\boldsymbol{u} && (4.3) \\ \boldsymbol{y} &= \boldsymbol{C} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\nu} \end{bmatrix} + \boldsymbol{D}\boldsymbol{u} && (4.4) \end{aligned}$$

In these equations, state vector is 12-by-1 vector, $\boldsymbol{A}$ is 12-by-12 matrix, $\boldsymbol{B}$ is 12-by-2 matrix and $\boldsymbol{u}$ is 2-by-1 column vector, whose first term is left thruster force and the second term is right thruster force. $\boldsymbol{C}$ is 2-by-12 matrix and $\boldsymbol{D}$ is 2-by-2 matrix. In order to obtain Equations (4.3), consider Equations (4.1) and (4.2) as in (4.5).

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{4.5}$$

Now, apply linearization by Taylor series expansion to (4.5) and obtain (4.6) [51].

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_0},\boldsymbol{u_0}) + \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}|_{\boldsymbol{x_0},\boldsymbol{u_0}}\Delta\boldsymbol{x} + \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}|_{\boldsymbol{x_0},\boldsymbol{u_0}}\Delta\boldsymbol{u} \tag{4.6}$$

Equation (4.6) can be written in the form of Equation (4.7).

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_0},\boldsymbol{u_0}) + \boldsymbol{A'}\Delta\boldsymbol{x} + \boldsymbol{B}\Delta\boldsymbol{u} \tag{4.7}$$

Since linearization points are not equilibrium points, the term $\boldsymbol{f}(\boldsymbol{x_0},\boldsymbol{u_0})$ is different than zero. As a result, in order to obtain the standard form of state space representation, augmentation of the state vector is required. After this augmentation, Equation (4.7) can be converted to final form as in Equation (4.8)

$$\begin{bmatrix} \Delta\dot{\boldsymbol{x}} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A'} & \boldsymbol{f}(\boldsymbol{x_0},\boldsymbol{u_0}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ 1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \Delta\boldsymbol{u} \tag{4.8}$$

Equation (4.8) represents the linearized form of the nonlinear model described in Equations (4.1) and (4.2). Details of terms in Equation (4.8) are not given here due to complexity of the terms, however, considering previous chapter and a computational tool like MATLAB, terms can be found easily.

It is noted that in order to deal with small order transfer functions and to speed up the computation, linearization is performed on reduced state model. Namely, new states vector becomes as in Equation (4.9).

$$\bar{\boldsymbol{x}} = [x, y, \psi, u, v, r]^T \tag{4.9}$$

Corresponding A, B, C and D matrices are also modified.

This type of linearization is carried out for different values of surge speed and yaw position. Surge speed range for linearization is from 0 m/s to 5 m/s. Yaw position range for linearization is from 0 rad to pi/2 rad. For each variable, 10 different linearization points are taken, as a result, totally 100 number of linearizations are performed.

4.2.1.2 Kalman Decomposition

In order to manage with minimal order transfer functions, Kalman decomposition is utilized to extract the controllable/observable states. After obtaining minimal realization of the state space representation, transfer functions can be obtained. With these transfer functions, tuning of PID's becomes a straightforward process.

Consider the linear system given in Equations (4.10) and (4.11) [52].

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4.10}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \tag{4.11}$$

where $\boldsymbol{A}$ is $n \times n$, $\boldsymbol{B}$ is $n \times p$, $\boldsymbol{C}$ is $q \times n$, $\boldsymbol{D}$ is $q \times p$ and n is the number of states, p is the number of inputs and q is the number of outputs.

Let $\bar{\boldsymbol{x}} = \boldsymbol{P}\boldsymbol{x}$ where $\boldsymbol{P}$ is nonsingular, then it is known that the state equations become as in (4.12) and (4.13).

$$\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{B}}\boldsymbol{u} \tag{4.12}$$

$$\bar{\boldsymbol{y}} = \bar{\boldsymbol{C}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{D}}\boldsymbol{u} \tag{4.13}$$

where $\bar{\boldsymbol{A}} = \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^{-1}$, $\bar{\boldsymbol{B}} = \boldsymbol{P}\boldsymbol{B}$, $\bar{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{P}^{-1}$ and $\bar{\boldsymbol{D}} = \boldsymbol{D}$.

Equations (4.12) and (4.13) are algebraically equivalent to Equations (4.3) and (4.4).

With Kalman decomposition and proper selection of $\boldsymbol{P}$ matrix(details of selection of $\boldsymbol{P}$ matrix are not given here), system can be converted to the form in the Equations (4.14) and(4.15) [52].

$$\begin{bmatrix} \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{CO}} \\ \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{C\tilde{O}}} \\ \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{\tilde{C}O}} \\ \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{\tilde{C}\tilde{O}}} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{A}}_{\boldsymbol{CO}} & 0 & \bar{\boldsymbol{A}}_{13} & 0 \\ \bar{\boldsymbol{A}}_{21} & \bar{\boldsymbol{A}}_{\boldsymbol{C\tilde{O}}} & \bar{\boldsymbol{A}}_{23} & \bar{\boldsymbol{A}}_{24} \\ 0 & 0 & \bar{\boldsymbol{A}}_{\boldsymbol{\tilde{C}O}} & 0 \\ 0 & 0 & \bar{\boldsymbol{A}}_{24} & \bar{\boldsymbol{A}}_{\boldsymbol{\tilde{C}\tilde{O}}} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}}_{\boldsymbol{CO}} \\ \bar{\boldsymbol{x}}_{\boldsymbol{C\tilde{O}}} \\ \bar{\boldsymbol{x}}_{\boldsymbol{\tilde{C}O}} \\ \bar{\boldsymbol{x}}_{\boldsymbol{\tilde{C}\tilde{O}}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{B}}_{\boldsymbol{CO}} \\ \bar{\boldsymbol{B}}_{\boldsymbol{C\tilde{O}}} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u} \tag{4.14}$$

$$\boldsymbol{y} = \begin{bmatrix} \bar{\boldsymbol{C}}_{\boldsymbol{CO}} & 0 & \bar{\boldsymbol{C}}_{\boldsymbol{\tilde{C}O}} & 0 \end{bmatrix} + \boldsymbol{D}\boldsymbol{u} \tag{4.15}$$

In the Equations (4.14) and(4.15), controllable, uncontrollable, observable and unobservable states are decomposed.

Minimal realization of the system is obtained by using only the controllable and observable states resulting from the Kalman decomposition.

$$\dot{\bar{\boldsymbol{x}}}_{\boldsymbol{CO}} = \bar{\boldsymbol{A}}_{\boldsymbol{CO}}\bar{\boldsymbol{x}}_{\boldsymbol{CO}} + \bar{\boldsymbol{B}}_{\boldsymbol{CO}}\boldsymbol{u} \tag{4.16}$$

$$\boldsymbol{y} = \bar{\boldsymbol{C}}_{\boldsymbol{CO}}\bar{\boldsymbol{x}}_{\boldsymbol{CO}} + \boldsymbol{D}\boldsymbol{u} \tag{4.17}$$

Considering Equations (4.16) and(4.17), one can calculate the transfer functions of the system by using expression given in Equation (4.18)

$$G(s) = \bar{\boldsymbol{C}}_{\boldsymbol{CO}}(s\boldsymbol{I} - \bar{\boldsymbol{A}}_{\boldsymbol{CO}})^{-1}\bar{\boldsymbol{B}}_{\boldsymbol{CO}} \tag{4.18}$$

Since this system is two inputs - two outputs MIMO system, for each linearization point, four transfer functions exist. It is noted that transfer functions can be directly calculated by using $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ matrices. However due to numerical errors of computational tools such as MATLAB, minimum order transfer functions with all pole-zero cancellations may not be obtained. Obtaining minimal realization and then finding transfer functions is the most straightforward way.

At this point, it is noted that for PID tuning purposes, directly observed states are taken as surge speed and yaw position and this is also the actual case. Furthermore, this selection of observed states simplifies the transfer functions.

4.2.2 Controller Design

In this section, for each transfer function, continuous time PID controllers are tuned by considering desired transient response, stability and robustness conditions. Next, for each closed loop system, time constants are found in order to determine the sample time of the controller. Then, appropriate unique sample time for all linearization points is chosen. In the following step, for each linearization point, four PID controllers are combined and by using linear models, weights of each of them are found by using a simple optimization process. As a final step, performances of PID's with optimized weights are evaluated on the nonlinear model.

4.2.2.1 PID Tuning Algorithm

For tuning purposes, PIDF type of continuous type controller in parallel form is chosen. PIDF type of controller is similar to PID controller but it also has first order filter on derivative term. General form is given in Equation (4.19).

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \tag{4.19}$$

For each linearization point, there are four PID controllers to be implemented. Two of them take errors from surge speed and creates commands for left and right thrusters, other two take error from yaw position and generates force commands for the left and the right thrusters. For tuning purposes, “pidtune” function of MATLAB [53] is utilized with an additional upper tuning algorithm. “pidtune” function accepts plant transfer function, type of “PID” utilized, target crossover frequency and target phase margin as input arguments and outputs the transfer function of controller and some information related to controller such as stability, reached crossover frequency and phase margin. Upper tuning algorithm gives input to this function considering desired transient response and robustness characteristics. This is achieved by using the following piece of information: Target crossover frequency is roughly equal to the closed loop bandwidth. To get a faster response this value can be increased but stability may degrade. Although selected crossover frequency restricts the achievable phase margin, a suitable value of phase margin makes system more robust and stable; however increasing phase margin too much may slow the response.

Considering the information in previous paragraph, below target conditions are determined:

- Settling time is less than 1 sec for 5% settling limit,
- Maximum overshoot for step response should be below 10%,
- Maximum gain crossover frequency is 40 rad/sec,
- Phase margin is 60 degrees.

However, if gain crossover frequency reaches the limit, settling time and maximum overshoot conditions can be loosen. With this algorithm, PID parameters are also kept in limits due to gain crossover limit. The flowchart of the algorithm is given in Figure 4.1.

4.2.2.2 Sample Time Selection for Controller

Since PID’s are implemented on an embedded environment, suitable sample time for discretization should be found. Assume that this discretization is made by using zero order hold (ZOH). As can be seen in Figure 4.2 , zero order hold with h time-step implies $h/2$ time step of delay of input signal [4].

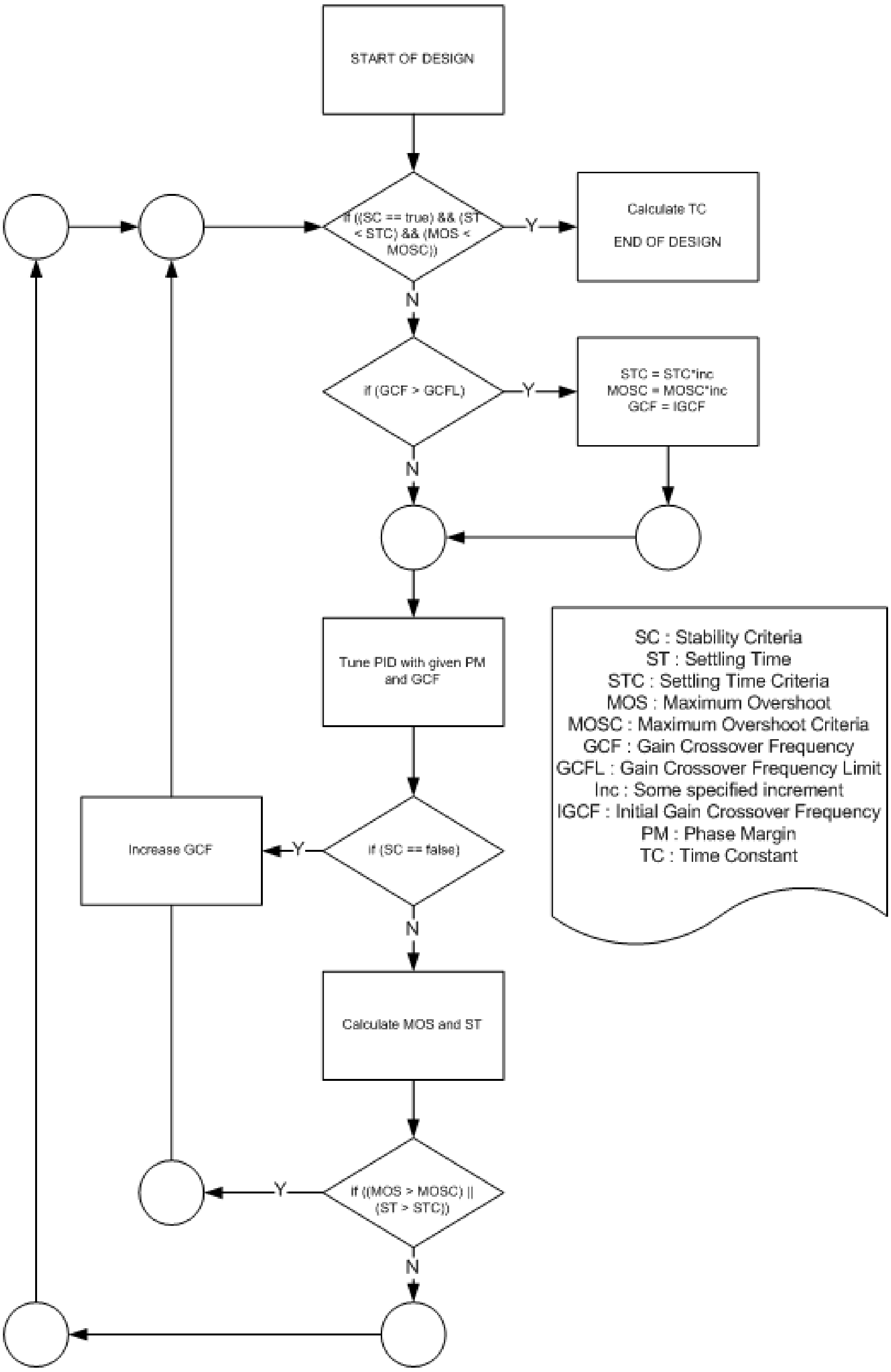

START OF DESIGN
if ((SC == true) && (ST < STC) && (MOS < MOSC))
Y
Calculate TC
END OF DESIGN
N
if (GCF > GCFL)
Y
STC = STC*inc
MOSC = MOSC*inc
GCF = IGCF
N
Tune PID with given PM and GCF
SC : Stability Criteria
ST : Settling Time
STC : Settling Time Criteria
MOS : Maximum Overshoot
MOSC : Maximum Overshoot Criteria
GCF : Gain Crossover Frequency
GCFL : Gain Crossover Frequency Limit
Inc : Some specified increment
IGCF : Initial Gain Crossover Frequency
PM : Phase Margin
TC : Time Constant
Increase GCF
Y
if (SC == false)
N
Calculate MOS and ST
Y
if ((MOS > MOSC) || (ST > STC))
N

Figure 4.1: PID tuning algorithm flowchart

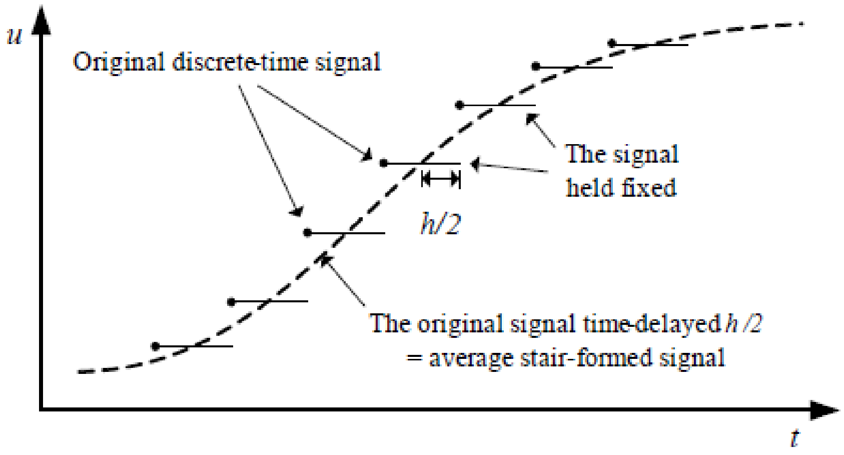

u
Original discrete-time signal
The signal held fixed
h/2
The original signal time-delayed h/2 = average stair-formed signal
t

Figure 4.2: ZOH and delay relationship [4]

Suppose that a continuous time PID controller is tuned for a system as in this case. If one applies these parameters directly to the system, the control loop will get reduced stability because of this approximate $h/2$ delay. "As a rule of thumb stability reduction is tolerable if the time delay is less than on tenth of the response time of the control system as it would have been with a continuous time controller" [4]. This can be seen in Equations (4.20) and(4.21).

$$\frac{h}{2} \leq \frac{T_r}{10} \tag{4.20}$$

which gives,

$$h \leq \frac{T_r}{5} \tag{4.21}$$

In previous section, for each PID, time constants of closed loop systems were found. Reciprocal of time constant is the response time of the system. (Response time is the 63% rise time.) For all compensated systems, response times are found. Minimum response time is 0.25 seconds. Hence sample time should be smaller than 0.05 seconds. Considering safety factors sample time for controller is taken as 0.02 sec.

4.2.2.3 Weight Determining Algorithm for PIDs

After tuning four PID's for each linearization point, these PIDs are imported to a simplified linear model to determine the weights of each PID. The model is given in Figure 4.3.

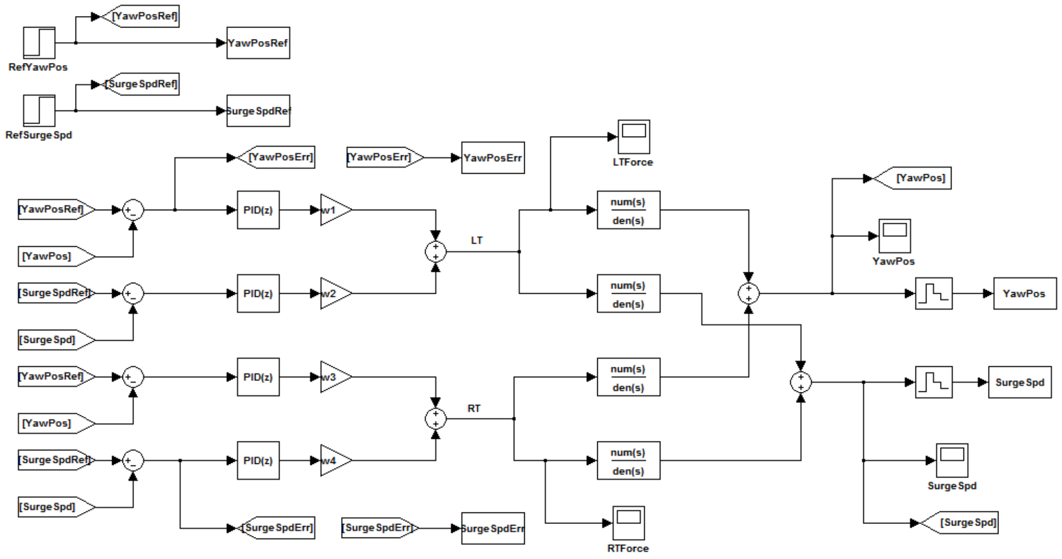


Figure 4.3: Simplified model for PID optimization

By using this model and an error minimizing optimization algorithm, weights of each PID for each linearization point are determined. In this simple one dimensional optimization problem, weight and cost function calculations are given in Equations from 4.22 to 4.25

$$w_1 = w_3 \tag{4.22}$$

$$w_2 = 1 - w_1 \tag{4.23}$$

$$w_4 = 1 - w_3 \tag{4.24}$$

$$PIDCostFunc = YawSpeedError^2 + SurgeSpeedError^2 \tag{4.25}$$

4.2.2.4 Interpolation of Parameters

PID parameters are determined only for 100 operation points. An interpolation algorithm should be utilized in order to handle other points between these operation points. This interpolation algorithm takes any operating point as an input argument and based on the distances of this point to the linearization points, determines the PID parameters to be utilized for that point.

Suppose that x is any operating point, x_i is any linearization point, weights for each linearization point are determined by the following procedure given in Equations (4.26), (4.27) and (4.28) [44].

$$we_i = \frac{\xi_i}{\sum \xi_i} \tag{4.26}$$

$$\xi_i = \frac{1}{\|x - x_i\| + \varepsilon_1} \quad if \;\; \varepsilon_1 > 0 \tag{4.27}$$

$$\xi_i = 0 \;\; if \;\; \xi_1 < \varepsilon_2 \tag{4.28}$$

where we_i is the weights of each linearization point and ε_1 and ε_2 are pre-determined small numbers. ε_1 is inserted to obtain finite ξ_i when a linearization point is hit. ε_2 is selected to filter out the effects of linearization points that have long distance to operating point. It is known that,

$$\sum we_i = 1 \tag{4.29}$$

Let's assume that any PID parameter of a linearization point is p_i, then, PID parameter p for any operating point is calculated as in Equation 4.30.

$$p = \sum we_i p_i \tag{4.30}$$

p can be any controller parameter such as K_p, K_i, K_d and T_f.

4.3 Sliding Mode Controller Design

In this section, first of all, modifications and decoupling procedure of the states on previously derived model are introduced. Next, basic theory of sliding mode controller is given. Then, stability analysis for the system is explained. In the following section, chattering issues related the sliding mode controller are discussed. After that part, general form of SMC is given. In the next part, parameter selection for the controller implementation is explained. Results are presented after this section for the cases with and without disturbances, furthermore resulting data is evaluated considering the ones of PID.

4.3.1 Decoupling of States

In order to implement SMC easily, system should be decoupled into propulsion and heading subsystems and actuation inputs should be arranged to adapt to this change.

If one considers the nonlinear model derived as in Equation (4.31),

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{x}) \tag{4.31}$$

careful selection of the states to be included in the decoupled subsystem is needed. Here x is the state vector, $\boldsymbol{A}$ is the system matrix, $\boldsymbol{B}$ is the input matrix, $\boldsymbol{u}$ is the input vector and $\boldsymbol{f}(\boldsymbol{x})$ describes the "any difference that would cause the system to deviate from its equilibrium point such as nonlinearities, unmodeled dynamics and external disturbances" [5].

The subsystem's states are chosen such a way that dominant dynamics of the propulsion and heading are decoupled from dynamics that have very little influence on them [5]. This decoupling creates two fundamental subsystems which are heading and propulsion subsystem, both in the form of Equation (4.32).

$$\dot{\boldsymbol{x}}_{\boldsymbol{s}} = \boldsymbol{A}_{\boldsymbol{s}}\boldsymbol{x}_{\boldsymbol{s}} + \boldsymbol{B}_{\boldsymbol{s}}\boldsymbol{u}_{\boldsymbol{s}} + \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{x}_{\boldsymbol{s}}) \tag{4.32}$$

Propulsion subsystem only includes the state of surge speed in body coordinate system, heading subsystem includes sway speed, yaw speed in body coordinate system and yaw position in earth fixed coordinate system.

As mentioned previously, the actuation inputs are the left thruster force and right thruster force. However, these actuation inputs cannot be directly utilized in mentioned subsystems because both have some effect on heading and propulsion subsystem. As a result, these forces are transformed to the surge force and yaw torque actuation signals as can be seen in Figure 4.4. Yaw torque only acts in heading subsystem and surge force has only effect on propulsion subsystem.

In above figure, τ_1 corresponds to surge force and τ_3 corresponds to yaw torque. τ_2 is given here for completeness and will be utilized in disturbance analysis. The relation between the thruster forces, the surge force and the yaw torque can be found in Equations (4.33) and (4.34).

$$F_{LT} = \frac{\tau_1 d_{RT} + \tau_3}{d_{RT} + d_{LT}} \tag{4.33}$$

$$F_{RT} = \frac{\tau_1 d_{LT} - \tau_3}{d_{RT} + d_{LT}} \tag{4.34}$$

where d_{RT} and d_{LT} are the right and left thruster distances to CG in y direction respec-

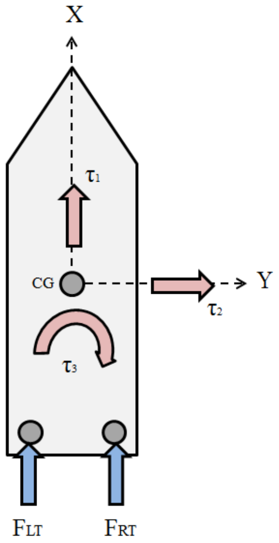


Figure 4.4: Representation of the actuation signals

tively. The decoupled subsystem can be written as in Equations (4.35) and (4.36).

$$\dot{u} = A_p u + B_p \tau_1 + f(u) \tag{4.35}$$

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{A_h} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} + B_h \tau_3 + f(v, r, \psi) \tag{4.36}$$

4.3.2 Basics of Sliding Mode Controller Theory

Sliding mode control(SMC) is considered as a robust control methodology. As a result it compensates the changes in the plant and external disturbances without significant performance measure. Considering structure of the controller, it consists of a nominal part that provides main control action and an additional term for dealing with the disturbances and unmodeled dynamics [45]. Nominal part is called equivalent term and additional control action is named as switching term as in Equation (4.37).

$$u_{SM} = u_{eq} + u_{sw} \tag{4.37}$$

As mentioned in previous sections, the control problem is to make the system response track a desired command or trajectory. SMC achieves this task by comparing the actual states $\boldsymbol{x}$ with the desired ones $\boldsymbol{x_d}$. In its structure, it creates a surface which is a function of tracking error $\hat{\boldsymbol{x}} = (\boldsymbol{x} - \boldsymbol{x_d})$. This surface is called as sliding surface $\boldsymbol{\sigma}(\hat{\boldsymbol{x}})$. SMC provides a control action that pulls the system to the sliding surface. When sliding surface is zero, it said that system is in "the Sliding Mode" [45]. This sliding surface is selected such that when the system is in the sliding mode, all states goes to desired ones eventually. Equivalently, all errors go to zero eventually. This can be seen from Figure 4.5.

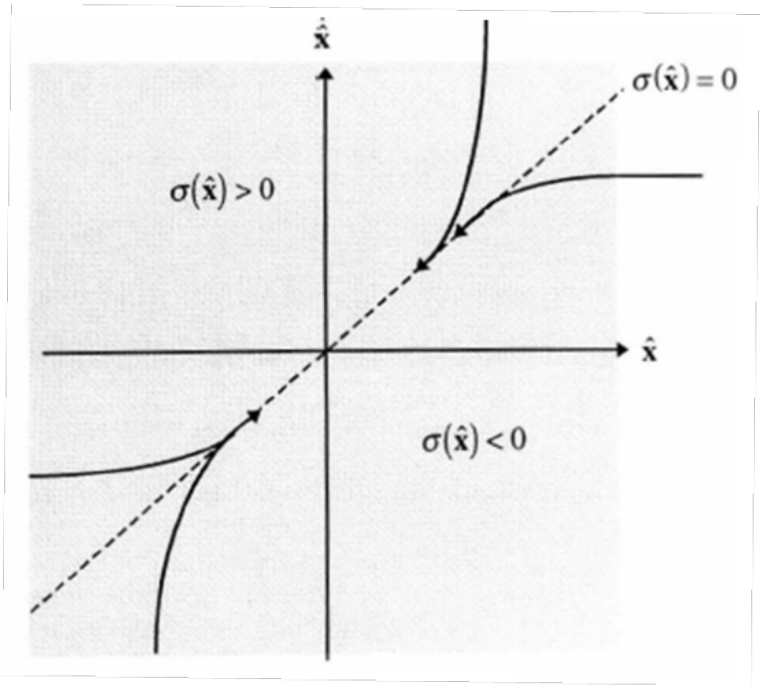


Figure 4.5: Sliding surface and state trajectory [5]

In order for the error of the state to go to zero, global asymptotic convergence of the surface to a stable equilibrium point is needed [5]. This can be shown by choosing a suitable Lyapunov function $V(\boldsymbol{\sigma})$. This function should satisfy the Lyapunov Stability Theorem which is given below:

Lyapunov's Global Stability Theorem: If a scalar function $V(\boldsymbol{\sigma})$ of a variable $\boldsymbol{\sigma}$ has a continuous first order derivatives and satisfies the following conditions,

- $V(\boldsymbol{\sigma})$ is positive definite for all $\boldsymbol{\sigma}$,
- $\dot{V}(\boldsymbol{\sigma})$ is negative definite for all $\boldsymbol{\sigma}$,

- When $V(\boldsymbol{\sigma})$ goes to infinity $||\boldsymbol{\sigma}||$ should go to infinity.

then the equilibrium at the origin of this function is globally asymptotically stable. Assume that as Lyapunov function, below function is chosen as in Equation (4.38).

$$V(\boldsymbol{\sigma}) = \frac{1}{2}\boldsymbol{\sigma}^2 \tag{4.38}$$

This function satisfies both first and third conditions. The remaining condition can be arranged such as given in Equation (4.39).

$$\dot{V}(\boldsymbol{\sigma}) = \dot{\boldsymbol{\sigma}}\boldsymbol{\sigma} < 0 \tag{4.39}$$

If above condition is met for selected sliding surface, the surface will go to zero. If the sliding surface is also chosen such that when the surface goes to zero, the state error goes to zero, then the states will follow the desired state trajectory.

The sliding surface utilized in this work is in the form of Equation (4.40).

$$\boldsymbol{\sigma}(\hat{\boldsymbol{x}}) = \boldsymbol{h}^T\hat{\boldsymbol{x}} = \boldsymbol{h}^T(\boldsymbol{x} - \boldsymbol{x_d}) \tag{4.40}$$

where $\boldsymbol{h}$ is the right eigen-vector (corresponds to zero eigenvalue) of the desired closed loop system matrix $\boldsymbol{A_c}$, $\boldsymbol{x}$ is the states and $\boldsymbol{x_d}$ is the desired states. The reason behind the selection of this sliding surface can be found in [5].

The equivalent component of the control action generally selected as a linear controller. For this work, a state-space feedback gain controller for reference tracking is chosen as in Equation 4.41.

$$u_{eq} = \boldsymbol{K_f}\boldsymbol{x_d} - \boldsymbol{k}^T\boldsymbol{x} \tag{4.41}$$

where $\boldsymbol{k}$ is the feedback gain coming from robust Pole Placement Theory proposed by [54]. The reason behind the selection of this method is that "it minimizes the sensitivity of the closed loop poles to perturbations in the coefficients of the matrices of the system" [45].

The switching term is a non-linear term and provides additional control action. The general form of switching term is given in Equation (4.42) and is derived by taking the derivative of Equation (4.40). Details of the derivation can be found in [5].

$$u_{sw} = (\boldsymbol{h}^T\boldsymbol{b})^{-1}(\boldsymbol{h}^T\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \eta sgn(\boldsymbol{\sigma}(\hat{\boldsymbol{x}}))) \tag{4.42}$$

where $\boldsymbol{b}$ is the input matrix, $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ is the derivative of the desired system matrix, η is the switching gain which determines the amplitude of the further switching action. As a result, the general form of SMC can be expressed as in Equation (4.43).

$$u_{SM} = \boldsymbol{K_f x_d} - \boldsymbol{k}^T \boldsymbol{x} + (\boldsymbol{h}^T \boldsymbol{b})^{-1}(\boldsymbol{h}^T \dot{\boldsymbol{x}}_{\boldsymbol{d}} - \eta sgn(\boldsymbol{\sigma}(\hat{\boldsymbol{x}}))) \tag{4.43}$$

4.3.3 Stability Analysis

Considering previous chapter, the only condition for global asymptotic stability and convergence to equilibrium point that has not been discussed in detail is the Equation (4.39). Consider this equation again.

$$\begin{aligned} \dot{V}(\boldsymbol{\sigma}) = \dot{\boldsymbol{\sigma}}\boldsymbol{\sigma} &= \boldsymbol{\sigma}(\boldsymbol{h}^T \Delta \boldsymbol{f}(\boldsymbol{x}) - \eta sgn(\boldsymbol{\sigma})) \\ &= \boldsymbol{\sigma}\boldsymbol{h}^T \Delta \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{\sigma}\eta sgn(\boldsymbol{\sigma}) \\ &= \boldsymbol{\sigma}\boldsymbol{h}^T \Delta \boldsymbol{f}(\boldsymbol{x}) - \eta |\boldsymbol{\sigma}| \end{aligned} \tag{4.44}$$

This equation should satisfy the stability condition. This is given in Equation (4.45)

$$\dot{V}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}\boldsymbol{h}^T \Delta \boldsymbol{f}(\boldsymbol{x}) - \eta |\boldsymbol{\sigma}| < 0 \tag{4.45}$$

Arranging Equation (4.45) and knowing that the switching gain should be positive, one can obtain Equation (4.46).

$$\eta > \|\boldsymbol{h}^T\| \|\Delta \boldsymbol{f}(\boldsymbol{x})\| \tag{4.46}$$

This condition ensures that stability occurs. In other words, the switching gain must be large enough to compensate any deviations from the nominal operating point of the plant [5].

4.3.4 Chattering Issues

Large switching gains may be beneficial to improve robustness and stability; however, it can lead to phenomenon called as “chattering”. Chattering occurs due to inclusion of sign function in the switching term. When this phenomenon comes through, control input starts to oscillating around the zero sliding surface. Since the oscillation is in the control input, namely in actuation signal, actuator may face unwanted and harmful wear, tear, vibration, sound, etc. [55]. Hence, the performance may degrade in time.

The most utilized way to solve this problem is the method known as "soft switching". In soft switching, instead of signum function, other smooth functions around the switching zone are used. In this study, hyperbolic tangent function is selected for that purpose. This function has the same asymptotes as the sign function, however, around the zero value of the sliding surface, there is a continuous transition area. This area is known as "boundary layer". By inclusion of boundary layer thickness parameter Φ, the thickness of the boundary layer can also be adjusted. The value of switching function with respect to sliding surface for different switching functions can be found in Figure 4.6.

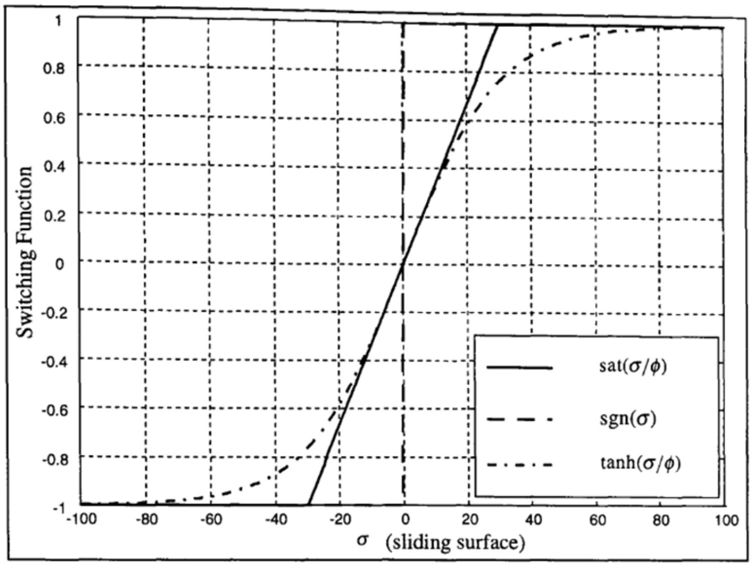


Figure 4.6: Switching function vs sliding surface [5]

The most important aspect in selection of the boundary layer thickness is that it should be large enough to cause soft switching. However, very large values of boundary layer thickness cause only proportional control action. The last arrangement of controller can be found in Equation (4.47)

$$u_{SM} = \boldsymbol{K_f x_d} - \boldsymbol{k}^T \boldsymbol{x} + (\boldsymbol{h}^T \boldsymbol{b})^{-1} \left(\boldsymbol{h}^T \dot{\boldsymbol{x}}_{\boldsymbol{d}} - \eta tanh \left(\frac{\boldsymbol{\sigma}(\hat{\boldsymbol{x}})}{\Phi} \right) \right) \tag{4.47}$$

Furthermore, if stability analysis is repeated, it can be shown that if the stability condition found outside the boundary layer is met sufficiently, the condition inside is automatically satisfied. The details can be found in [5].

4.3.5 General Form of Controller

Considering all information presented, the SMCs for heading and propulsion subsystems can be written as in Equations (4.48) and (4.49).

$$\begin{aligned}
\tau_1 &= K_{fp}u_d - \boldsymbol{k_p}^T u + (\boldsymbol{h_p}^T\boldsymbol{B_p})^{-1}\left(\boldsymbol{h_p}^T\dot{\boldsymbol{u}}_{\boldsymbol{d}} - \eta_p tanh\left(\frac{\boldsymbol{\sigma_p}(u-u_d)}{\Phi_p}\right)\right) \quad (4.48)\\
\tau_3 &= \boldsymbol{K_{fh}}\boldsymbol{x_d} - \boldsymbol{k_h}^T\boldsymbol{x_h} + (\boldsymbol{h_h}^T\boldsymbol{B_h})^{-1}\left(\boldsymbol{h_h}^T\dot{\boldsymbol{x}}_{\boldsymbol{hd}} - \eta_h tanh\left(\frac{\boldsymbol{\sigma_h}(\boldsymbol{x_h}-\boldsymbol{x_{hd}})}{\Phi_h}\right)\right) \quad (4.49)
\end{aligned}$$

Considering the propulsion subsystem, there are some parameters that should be determined. They are given below:

- Switching gain,
- Boundary layer thickness.

Considering the heading subsystem, parameters are,

- Two poles in equivalent term,
- Switching gain,
- Boundary layer thickness.

One pole in the equivalent term of the propulsion subsystem and one pole in the equivalent term of the heading subsystem are chosen as zero since the theory is based on this assumption. At this point, either these parameters can be estimated manually or can be utilized in an optimization algorithm. For this study, the first way is chosen.

4.3.6 Parameter Selection

After computation of the model parameters from nonlinear model derived for the propulsion and the heading subsystems, the study is classified into two groups. First one is the simulation without disturbance and the second one is the simulation with

Table 4.1: Selected parameters for SMC

Parameters	**Without Disturbances**	**With Disturbances**
K_{fp}	1	1
η_p	4	4
Φ_p	0.1	0.1
K_{fh}	1	1
p_{h1}	-1.5	-1.5
p_{h2}	1.4	-1.4
η_h	0.2	0.2
Φ_h	0.03	0.03

disturbances. Simulation parameters that are determined for this study are given in Table 4.1

The details of reasons for parameter selection are not given here. However, as a summary, the poles are placed based on the time response characteristics,the switching gains are selected considering unmodeled dynamics and the boundary layer thickness are determined by using chattering characteristics.

4.4 Evaluation of Results

Performance of the both PID and SMC are tested on the nonlinear model by giving step commands in different amplitudes. Each step command is initially filtered with a low-pass filter with the characteristics of 4.5 seconds rise time and maximum of 6% overshoot. This filter is utilized in order to eliminate high frequency components in the commands. Three different scenarios are designed for this study. While the surge speed command is 1 m/s and the yaw position command is $\pi/4$ rad in the first scenario, the surge speed of 0.5 m/s and the yaw position of $\pi/2$ rad request is generated in the second one. On the other hand, in the final test, the same input as in the first scenario is applied with the existence of disturbances. For each performance test, reference, feedback, error and actuation inputs are demonstrated for both PID and SMC in a comparative manner. These outputs for scenarios are illustrated from Figures 4.7 to 4.15. Furthermore, the performances of above mentioned controllers are also compared based on root-mean-square(RMS) values of the error for both surge speed and yaw position. These results are presented in Table 4.2.

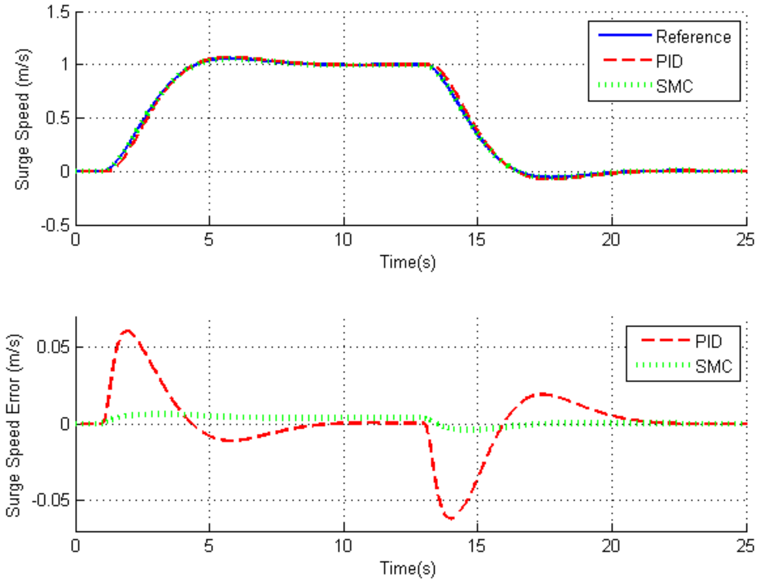


Figure 4.7: Surge speed results for Scenario 1

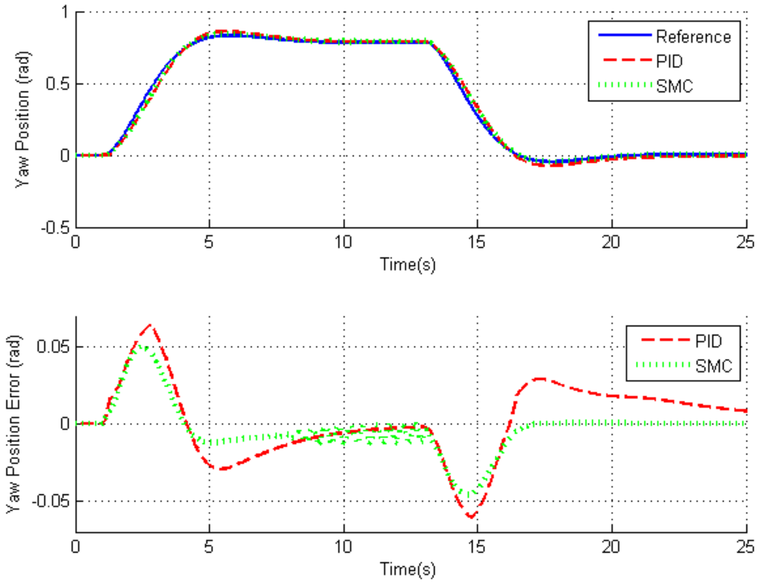


Figure 4.8: Yaw position results for Scenario 1

Considering general performance of the controllers, it can be concluded that SMC is superior to PID for all scenario cases. Another general result maybe that if ampli-

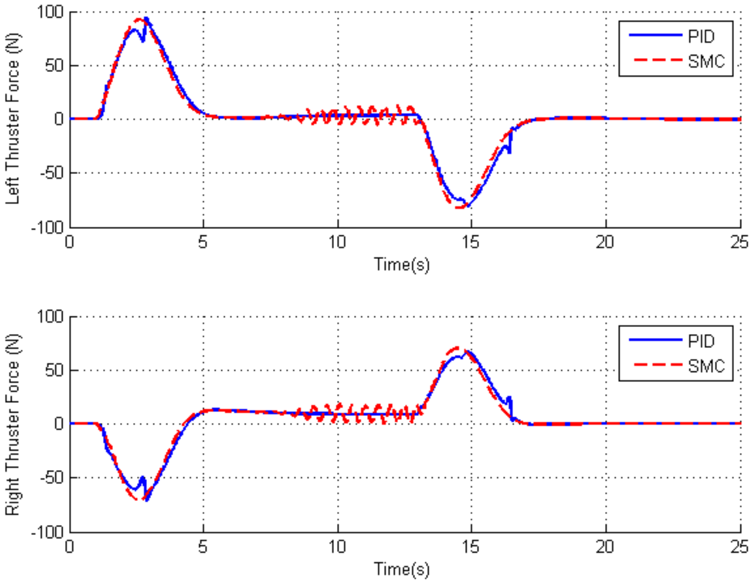


Figure 4.9: Actuation inputs for Scenario 1

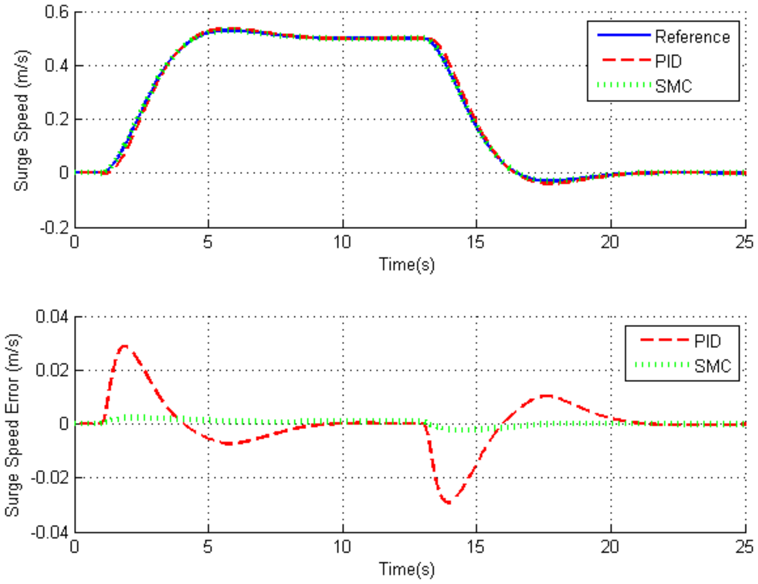


Figure 4.10: Surge speed results for Scenario 2

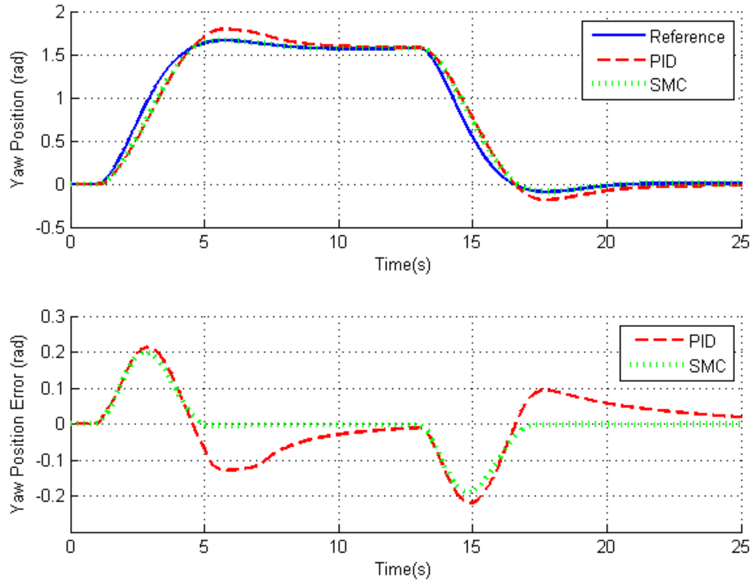


Figure 4.11: Yaw position results for Scenario 2

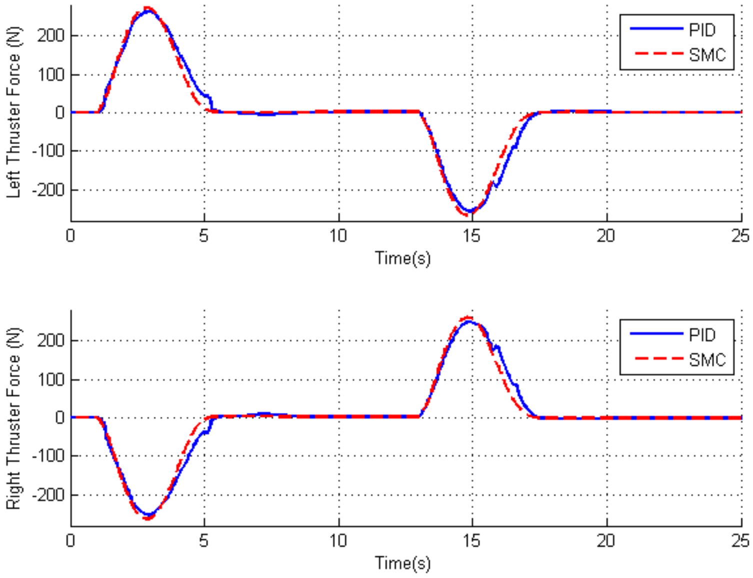


Figure 4.12: Actuation inputs for Scenario 2

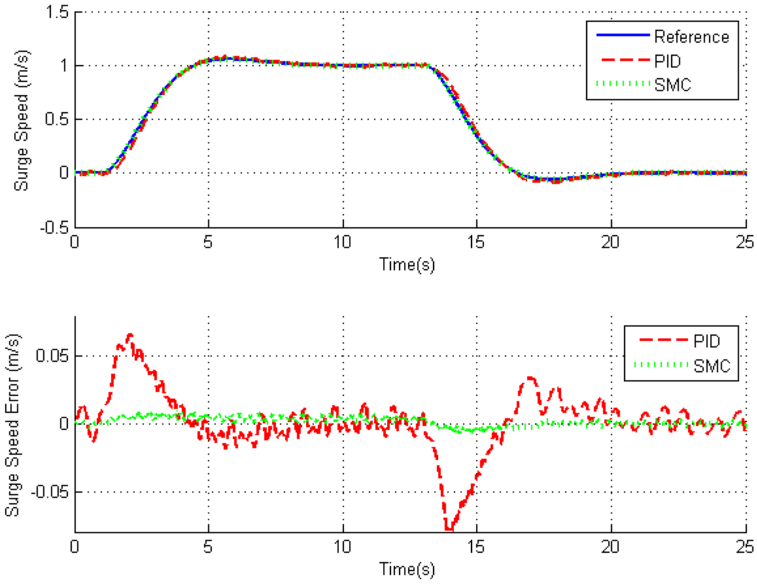


Figure 4.13: Surge speed results for Scenario 3

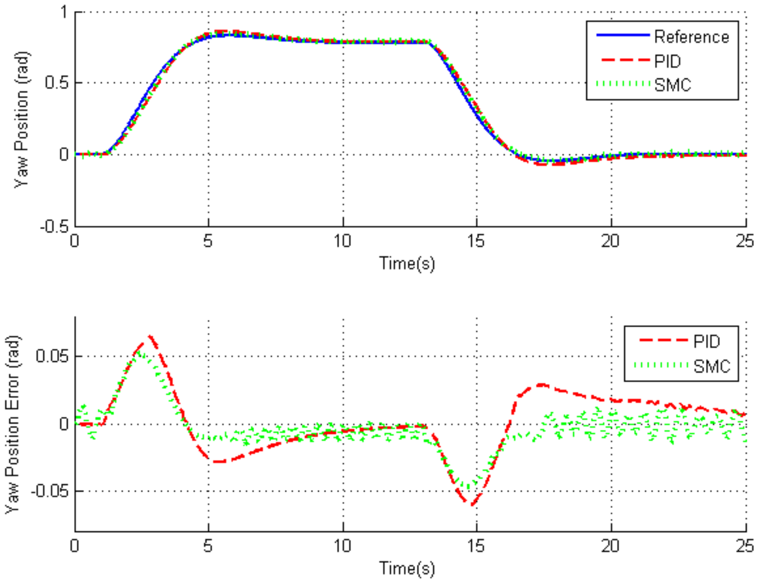


Figure 4.14: Yaw position results for Scenario 3

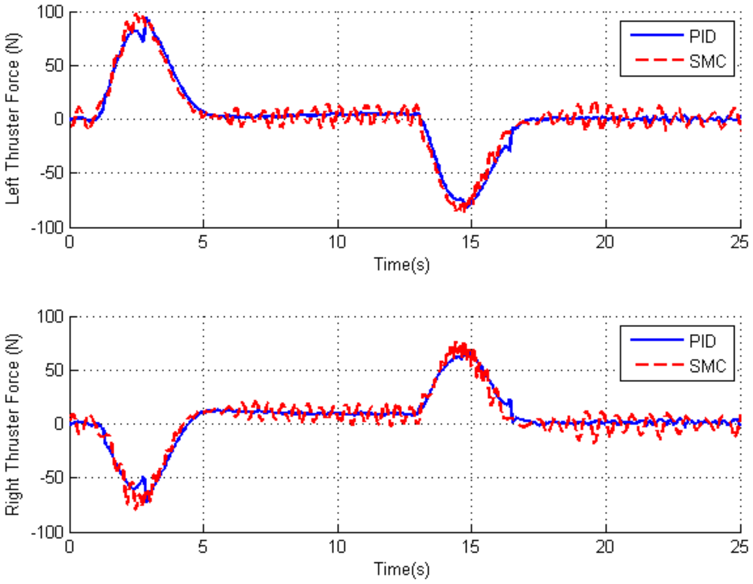


Figure 4.15: Actuation inputs for Scenario 3

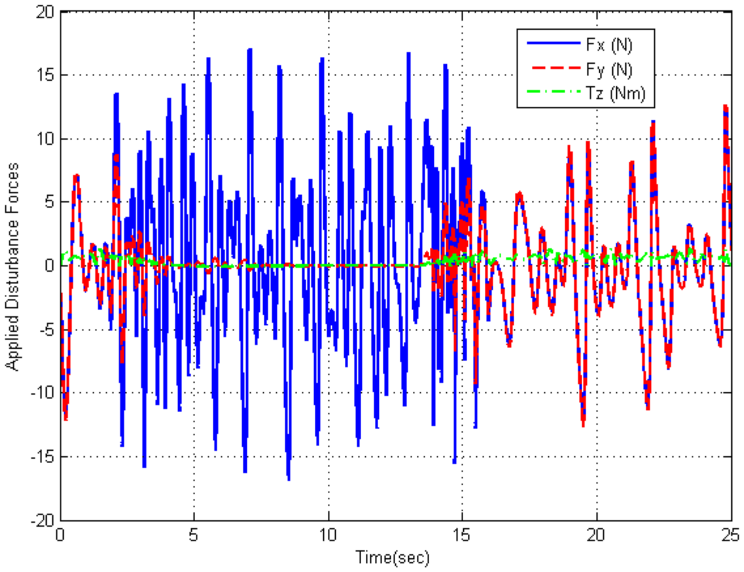


Figure 4.16: Disturbance forces applied

Table 4.2: RMS error values for scenarios

Scenario Number		1	2	3
Surge Speed	**PID**	0.0207	0.0096	0.0227
	SMC	0.0033	0.0011	0.0036
Yaw Position	**PID**	0.0246	0.0935	0.0243
	SMC	0.0170	0.0699	0.0178

tudes of the references increases, performance degrades. Observation of Figure 4.16, reveals that effects of the disturbance force on surge speed are more than that of yaw position. As a result, disturbances cause more performance reduction in surge control performance. When transient responses are considered, it is clear that response time of SMC is shorter and tracks the reference better for all cases. Furthermore, from error figures, it can be deduced that steady state errors go to zero for both controllers. For cases without disturbance, the trend of the actuation inputs seem very similar, however, there are small amounts of ripple in some instants for SMC case. It is observed that chattering is not fully eliminated, however, the amplitude and the frequency of it is suitable for this application. As expected, disturbance rejection capability of SMC is higher than that of PID. The main reason behind this situation is that SMC is a nonlinear controller and it is designed to eliminate the disturbance effects with switching term, on the other hand, PID based controller is designed for only some linearization points and other operation points are handled with interpolation. Another reason maybe that controllers are tested on nonlinear model and entire PID design carried out in linear models. It is not very surprising to have some deviation between linear and nonlinear models. RMS error values also show that surge speed performance of the controllers are better than yaw position performance of them. The essential reason behind this observation is that yaw position dynamics are more complex. As a result, design process gives more degraded results.

CHAPTER 5

EXPERIMENTAL SETUP

5.1 Introduction

It will be very beneficial to summarize the work that carried out until this point. First of all, the mathematical model of the sea surface vehicle is constructed and verified by using simulations. Next, an offline navigation algorithm is prepared in order to obtain more accurate data as mentioned in [47]. Then, a model sea surface vehicle is obtained and equipped with the electrical instruments, the driving system and some mechanical parts such as ballast weights and protective foams in order to perform reliable and safe experiments. Following step is to design and conduct the system identification experiments. The measurement data collected from the vehicle in these experiments are first processed by the navigation algorithm, and then utilized to identify the parameters of the mathematical model constructed. There are various types of software and hardware utilized in system identification and verification experiments. These tools will be presented in this section.

Above mentioned tools can be grouped into four parts.

- A Ready to Run (RTR) model boat, named Pacific Islander TugBoat with all its equipment such as drive system, regulators, cooling system, batteries, etc. This sea surface vehicle is shown in Figure 5.1.
- Fully functional advance autopilot card, called as Pixhawk, placed in the middle-front section of the boat.
- An embedded software inside the Pixhawk. Its purposes are collecting sensor

information, driving electrical motors, providing state-mode transitions, communicating with ground station and recording and storing information related to code.

- Ground station computer and its software are utilized to command the Pixhawk and collect necessary information online. Software of this computer is coded in MATLAB/Simulink environment. This software communicates with the Pixhawk over a wireless module using a serial communication protocol named Micro Air Vehicle Link (MAVLink).

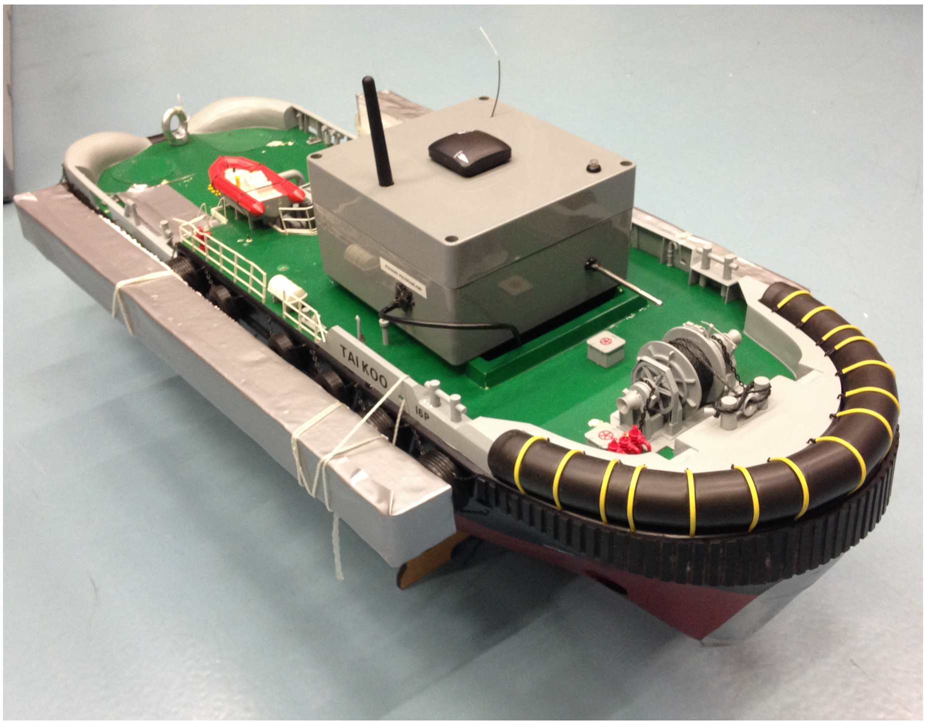


Figure 5.1: Pacific Islander Tug Boat

In the following sections, the hardware and the software employed are explained separately. In hardware part, model boat, its components and the autopilot card are investigated. On the other hand, in software part, the embedded code prepared for Pixhawk and the commanding software in ground station are mentioned.

5.2 Hardware Architecture

In this part, hardware employed is explained in a nutshell.

5.2.1 Autopilot Card and Peripherals

In this study, a Pixhawk autopilot card is decided to be used. There are some important reasons for this selection. First of all, it is the next generation autopilot card and its hardware is more elegant compared to similar autopilot cards. The other reason is that it is affordable and easy to obtain. Moreover, it is a very flexible hardware, in other words, it is customizable, open-source and open-hardware. Finally, it is very efficient since it includes real time operation system (RTOS). Programming can be easily carried out by using C/C++ and with the help of an integrated development environment (IDE). The IDE used in this study is Eclipse. Pixhawk without peripherals can be viewed from Figure 5.2. It is considered as a complete autopilot system with its software and hardware.

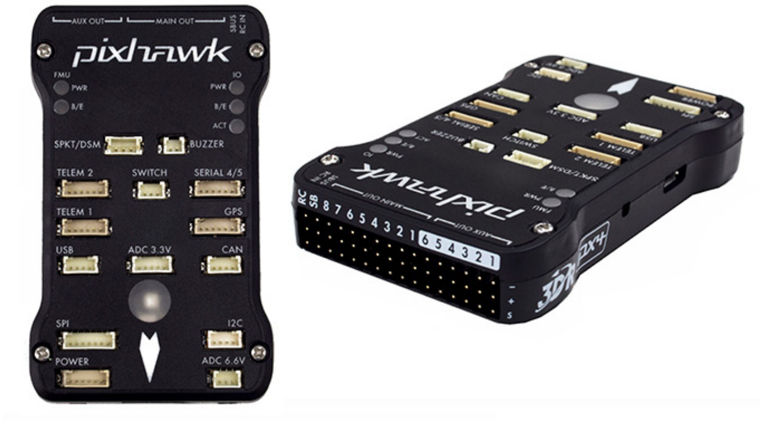


Figure 5.2: The Pixhawk autopilot card [6]

The Pixhawk autopilot system hardware can be grouped into three main categories. First group is a Flight Management Unit (FMU) and is depicted in Figure 5.3.

FMU consists of below units,

- 168 Mhz Cortex-M4F combining a 32 bit microcontroller,

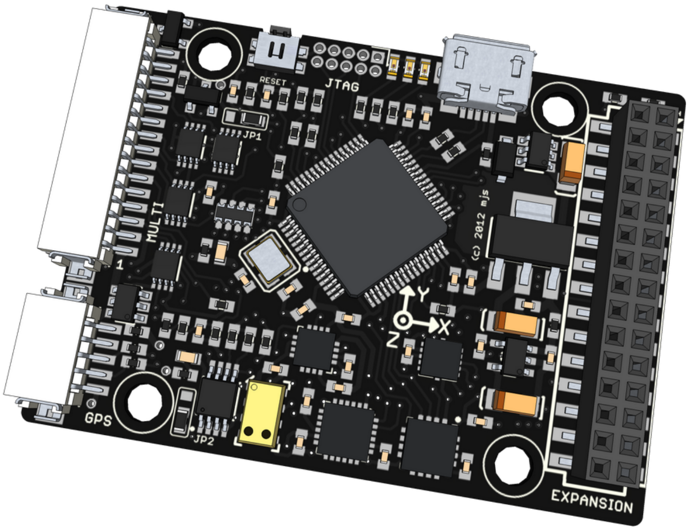


Figure 5.3: The Pixhawk FMU [7]

- A Digital Signal Processor (DSP),
- 192 KB SRAM,
- 1024 KB flash memory,
- Floating Point Unit (FPU),
- USB Bootloader,
- MicroSD slot for further memory requirements.

In addtion, FMU contains following sensors,

- MPU-6000 by Invensense including a 3 axis accelerometer and a 3 axis gyro,
- L3GD20, 3 axis gyro which is a product of ST Microelectronics,
- LSM303D which combines a 3 axis accelerometer and a magnetometer by ST Microelectronics,
- MS5611, pressure sensor by Measurement Specialties Inc.

The reason behind the support of two sets of sensors (gyros and accelerometers) is the consideration of emergency cases. The producer does not want to have an emergency case due to critical measurements. FMU also offers UART, I2C, SPI, CAN interfaces, among which only UART utilized in this study. Autopilot algorithms also run on FMU.

The second group is the IO Module and it is a kind of carrier board for the FMU. The main duty of the module is to provide the interfaces and regulated 5V for the FMU. It includes the following components,

- 24 Mhz Cortez-M3 failsafe microcontroller,
- Multiple power outputs for the peripherals,
- 2 solid state relays,
- 8 high speed PWM outputs.

IO Module is illustrated in Figure 5.4.

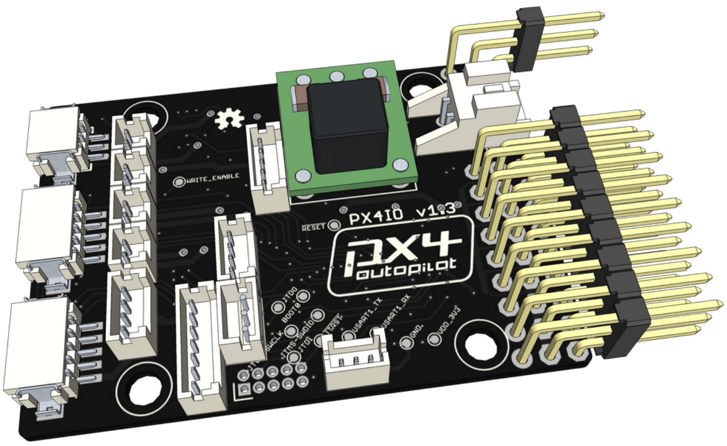


Figure 5.4: IO Module of the Pixhawk [8]

The third group is the peripherals of the Pixhawk and these are demonstrated in Figure 5.5. These peripherals are explained below:

- U-blox NEO-7 GPS module with a compass with and update rate about 5 Hz,
- Safety switch with and internal LED: Switch is utilized to activate the PWM outputs and internal LED provides visual information about states, modes and emergency cases of the software,

- External buzzer: Gives audial information related to software,
- 3DR telemetry radios: Provides wireless connection with ground station up to several kilometers. Designed to be work with MAVLink protocol.

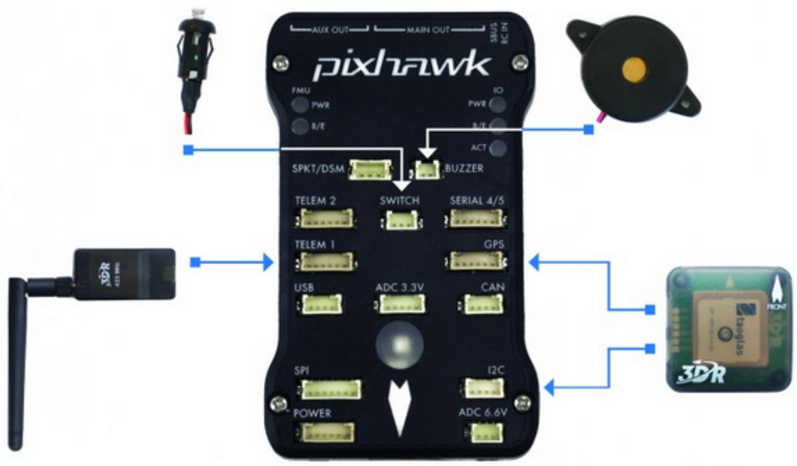


Figure 5.5: Peripherals of the Pixhawk [7]

5.2.2 Model Boat with Components

The model Pacific Islander TugBoat is a 1/40 detailed scale model of the real ship. A kind of composite material is utilized in its structure, hence it is very resistant to crashes. Dimensions are 900 mm in length, 290 mm in beam and about 580 mm in height. Excluding batteries, the weight of it is about 10.2 kg. Due to its size, weight and inertia characteristics, it is quite robust model ship compared to high speed fiberglass model boats. This boat is equipped with the two large, distinct, 4 bladed propellers which may provide agile maneuvering capabilities. Furthermore, kort nozzles are included around the propellers. Their function is to change the direction of the thrust. It is controlled by a small size servo motor, however, since thrusters are utilized separately, kort nozzles are not functional in this study. The propellers with kort nozzles of the sea surface vehicle can be seen from Figure 5.6.

Power is transmitted to these propellers with the help of drive system components. This system consists of a power source, electronic speed controllers (ESC), electric

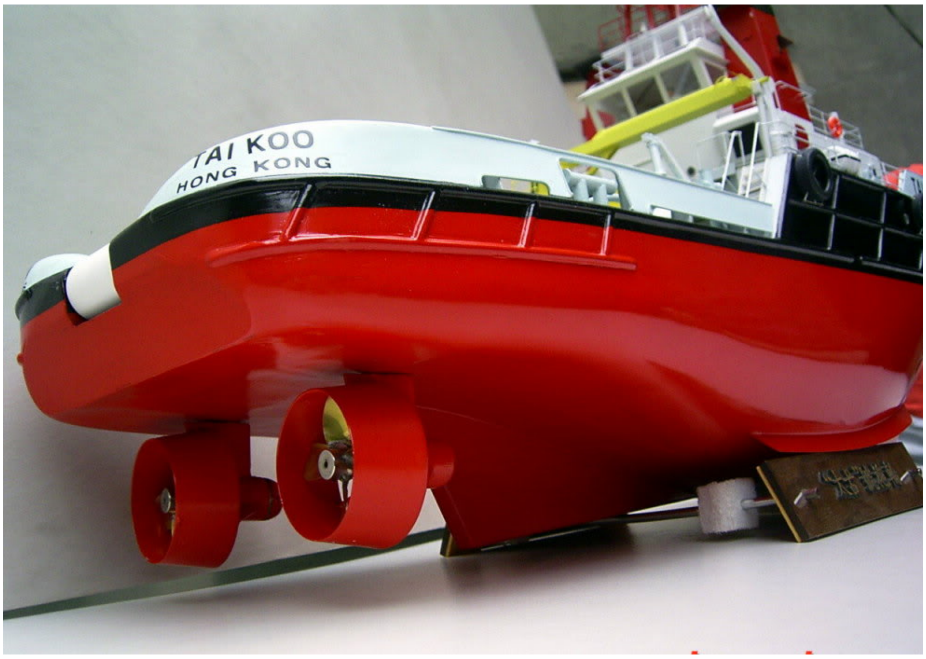


Figure 5.6: Twin propellers with kort nozzles [9]

motors and mechanical transmission parts. As a power source, a 4000 mAh, 8.4 V Lithium-polymer (Li-Po) two cell battery is chosen and it powers entire system. Li-Po battery is selected due to its high continuous power characteristic. Battery is directly connected to the ESCs and pump of the water cooling system and is connected to the autopilot card and its components via 5V output regulator. Two 500 size, brushed DC electric motors and two brushed electronic speed controllers(ESC) are included in the original delivery of the boat. These motors and ESCs are replaced by high power, waterproof, brushless counterparts due to safety and maintainability reasons. The ESC utilized in the final system is a product of Hobbywing Technology. The model name of it is Seaking-120 A-V3 and a picture of it can be found in Figure 5.7, moreover, some features of it are presented in Table 5.1

Table 5.1: Features of ESC

Continuous current (A)	120
Max Current (A)	720
Input	2-6 cell Li-Po
BEC Output (W)	6V/5A
Weight (g)	280

Figure 5.7: ESC utilized in the model boat [10]

This ESC can provide 120 A continuous and 720 A instantaneous current. It has a programming feature in that one can adjust turn direction, battery cell number, fine tuning parameters etc. PWM signals are provided by the Pixhawk to ESCs. The electric motors are also produced by Hobbywing Technology and it has model number of 3660SL 3180KV. This brushless electric motor is illustrated in Figure 5.8 and some features can be found in Table 5.2

Figure 5.8: Brushless electric motor in the setup [10]

The Pixhawk provides PWM signals to ESCs and controls the speed of the propellers.

Table 5.2: Features of brushless electric motor

Kv (rpm/V)	3180
Max Current (A)	94
Max Voltage (V)	18
Power (W)	1690
Weight (g)	280

The PWM signal basically corresponds to a constant voltage signal, the level of which varies based on the duty cycle of PWM signal and battery voltage supplied. As a result, the motion of the vessel is controlled by changing the pulse widths of two PWM signals. The last elements in the power transmission system are mechanical transmission parts. The electric motor is connected to the transmission shaft of the propeller by a flexible coupling. Flexible coupling is utilized in this structure because it eliminates alignment problems.

Considering all these features and by making necessary measurements such as dimension, weight, density, CAD model of the vehicle is constructed. A view from this model is provided in Figure 5.9. This model provides valuable information about dimensions and inertia.

5.3 Software Architecture

In this section, software architecture and the implemented codes both for the Pixhawk and the ground station are explained briefly.

5.3.1 The Pixhawk Software

Four major layers can be constructed when mentioning the high level software architecture of the Pixhawk. These are Application Programming Interface (API), Application Framework, Libraries and Operating System. This structure with corresponding examples are demonstrated in Figure 5.10.

Application developers use the API to create useful programs. Nodes, that are actually

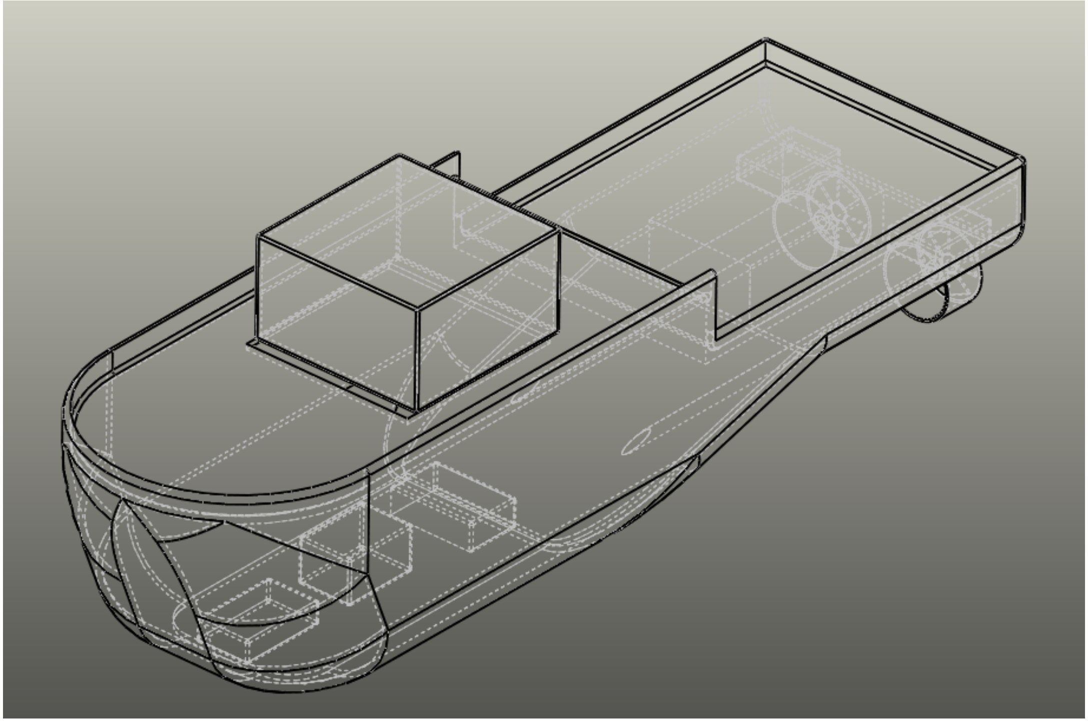


Figure 5.9: CAD model of the model boat

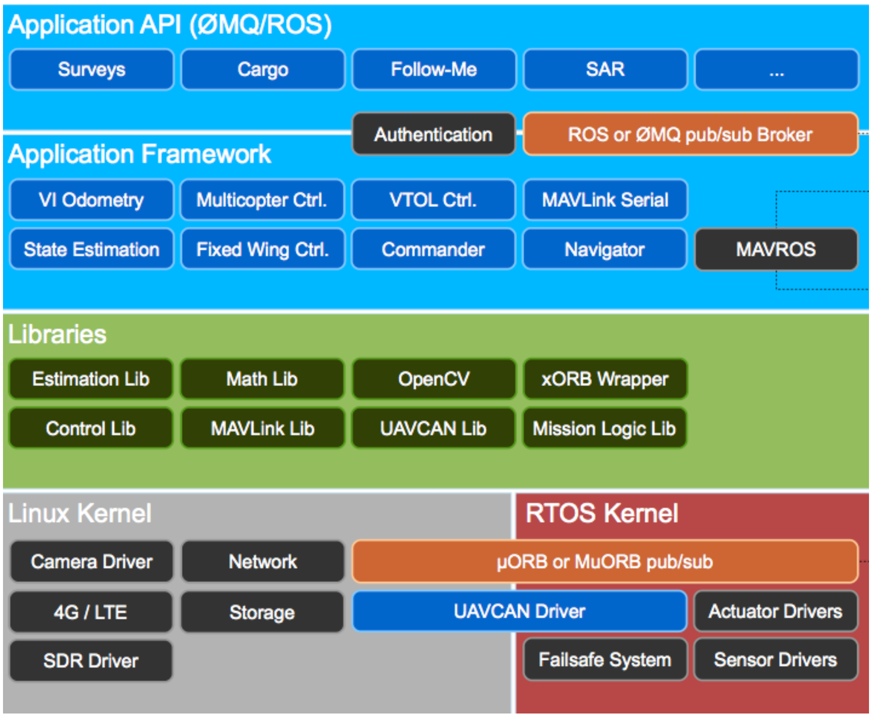
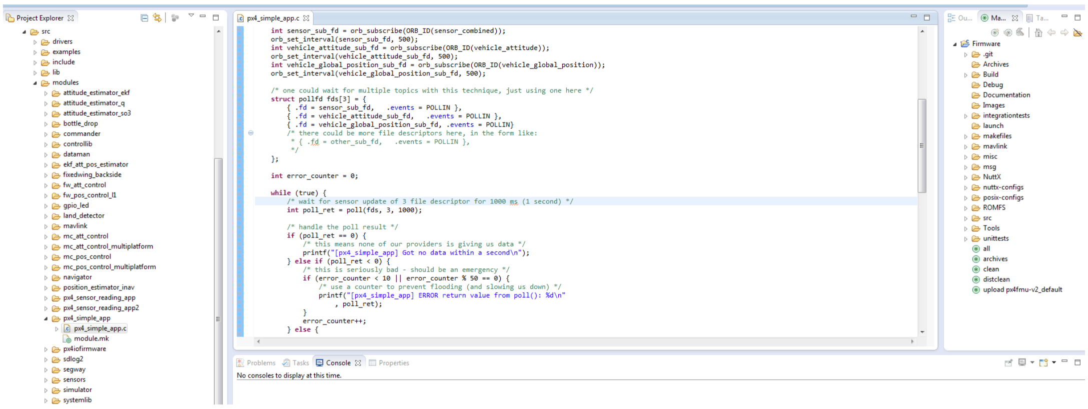


Figure 5.10: High-level software architecture of the Pixhawk [8]

main applications establishing the core functionality, are realized in the Application Framework. Libraries utilized by the applications are contained in the Libraries layer. These are the code pieces that can be called by different applications. The most distinct example of it is the math libraries. All hardware drivers, failsafe systems and network functions are included in the Operation System layer [8].

An important feature of the Pixhawk software architecture is the publish-subscribe pattern object request broker. This structure provides continuous communication between the applications. For example, assume that an application needs sensor information in order to perform a specific task, then this application subscribes to a bus, named topic and obtains these sensor values. Also suppose that it processes the sensor information and obtains a valuable output. If this output will be utilized by another application, the application generates this output publishes this output to another topic. This method is known as subscribe-publish pattern and prevents locking issues in communication between multiple applications.

For this study, a specific application is written for the purposes of driving DC motors, collecting and recording sensor information, communicating with ground station and performing safe mod and state transitions. The application basically has two modes which are Pre-operational and Operational modes. In pre-operational mode, the software can collect entire sensor information, communicate with the ground station, however, it is not able to drive the motors and record any information collected or generated. Pre-operational mode is implemented considering safety reasons such as emergency cases and carrying out sensor calibration works. On the other hand, in operational mode, motors are able to powered and all necessary information can be collected and logged into an SD card. Validation and system identification experiments are conducted in this mode. Underneath this mode, there are also some states such as unlimited PWM, limited PWM, however, details of them are not given here. A screenshot taken from an application written in Eclipse to be embedded to the Pixhawk is given in Figure 5.11.



```c
    int sensor_sub_fd = orb_subscribe(ORB_ID(sensor_combined));
    orb_set_interval(sensor_sub_fd, 500);
    int vehicle_attitude_sub_fd = orb_subscribe(ORB_ID(vehicle_attitude));
    orb_set_interval(vehicle_attitude_sub_fd, 500);
    int vehicle_global_position_sub_fd = orb_subscribe(ORB_ID(vehicle_global_position));
    orb_set_interval(vehicle_global_position_sub_fd, 500);

    /* one could wait for multiple topics with this technique, just using one here */
    struct pollfd fds[3] = {
        { .fd = sensor_sub_fd,   .events = POLLIN },
        { .fd = vehicle_attitude_sub_fd,   .events = POLLIN },
        { .fd = vehicle_global_position_sub_fd, .events = POLLIN}
        /* there could be more file descriptors here, in the form like:
         * { .fd = other_sub_fd,   .events = POLLIN },
         */
    };

    int error_counter = 0;

    while (true) {
        /* wait for sensor update of 3 file descriptor for 1000 ms (1 second) */
        int poll_ret = poll(fds, 3, 1000);

        /* handle the poll result */
        if (poll_ret == 0) {
            /* this means none of our providers is giving us data */
            printf("[px4_simple_app] Got no data within a second\n");
        } else if (poll_ret < 0) {
            /* this is seriously bad - should be an emergency */
            if (error_counter < 10 || error_counter % 50 == 0) {
                /* use a counter to prevent flooding (and slowing us down) */
                printf("[px4_simple_app] ERROR return value from poll(): %d\n"
                    , poll_ret);
            }
            error_counter++;
        } else {
```

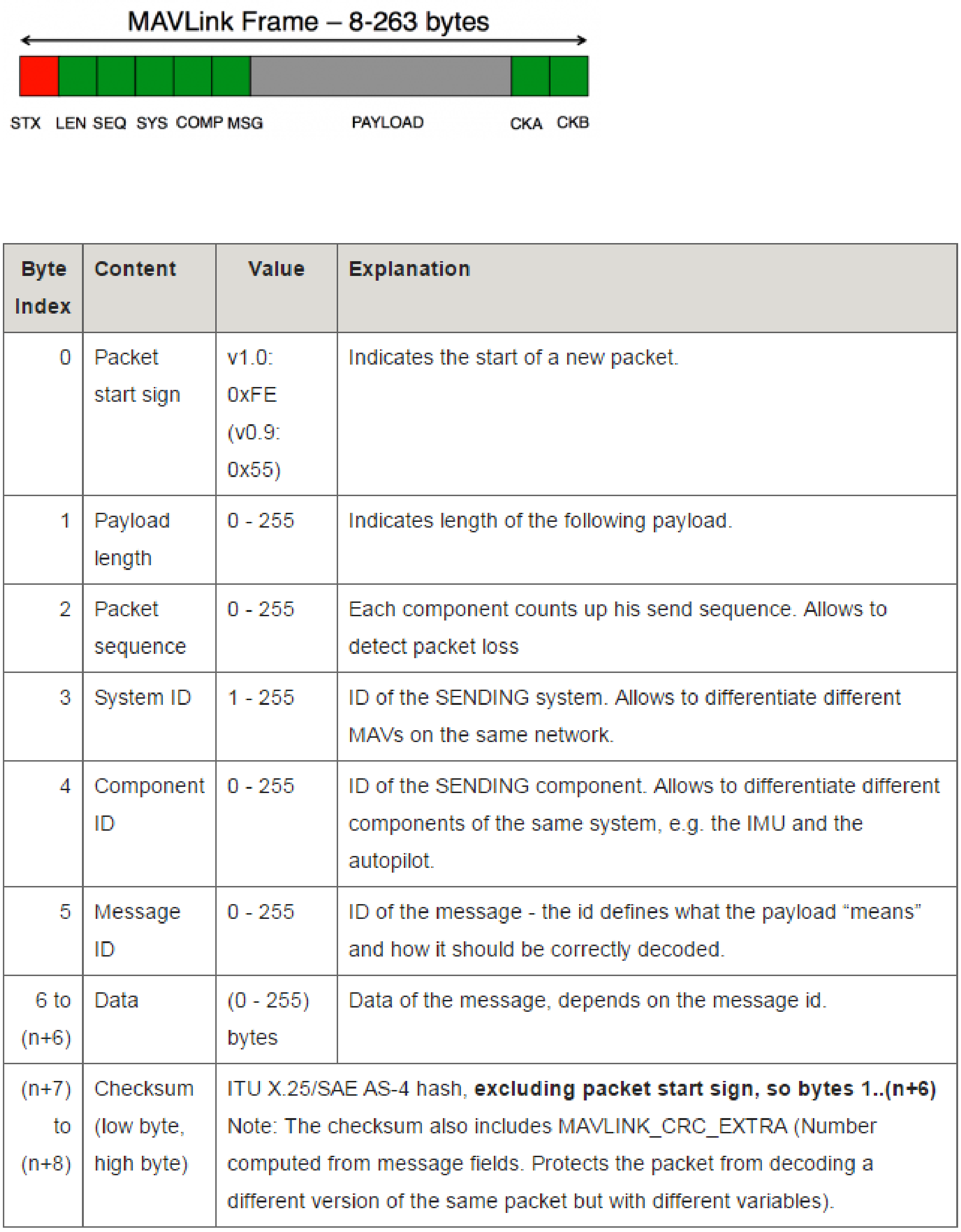
No consoles to display at this time.

Figure 5.11: A Pixhawk application written in Eclipse IDE

5.3.2 Ground Station Software

The ground station software is a MATLAB/Simulink model that implements MAVLink protocol in order to communicate with the Pixhawk over the telemetry device. "MAVLink is a very lightweight, header only message marshaling library for micro air vehicles.". This protocol was first released early 2009 by Lorenz Meier and extensively tested on PX4, Pixhawk and APM environments [11]. The frame anatomy is illustrated in the Figure 5.12.

A screenshot taken from implemented code in MATLAB/Simulink environment is given in Figure 5.13. Basically this program, receives all information coming from the Pixhawk and transmits mode-state commands, parameters values and continuous instructions such as PWM values to the Pixhawk. It can also work in different baud rates and sampling times. All useful standard messages of MAVLink are implemented in this program.



Byte Index	Content	Value	Explanation
0	Packet start sign	v1.0: 0xFE (v0.9: 0x55)	Indicates the start of a new packet.
1	Payload length	0 - 255	Indicates length of the following payload.
2	Packet sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload “means” and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, **excluding packet start sign, so bytes 1..(n+6)** Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

Figure 5.12: A MAVLink frame [11]

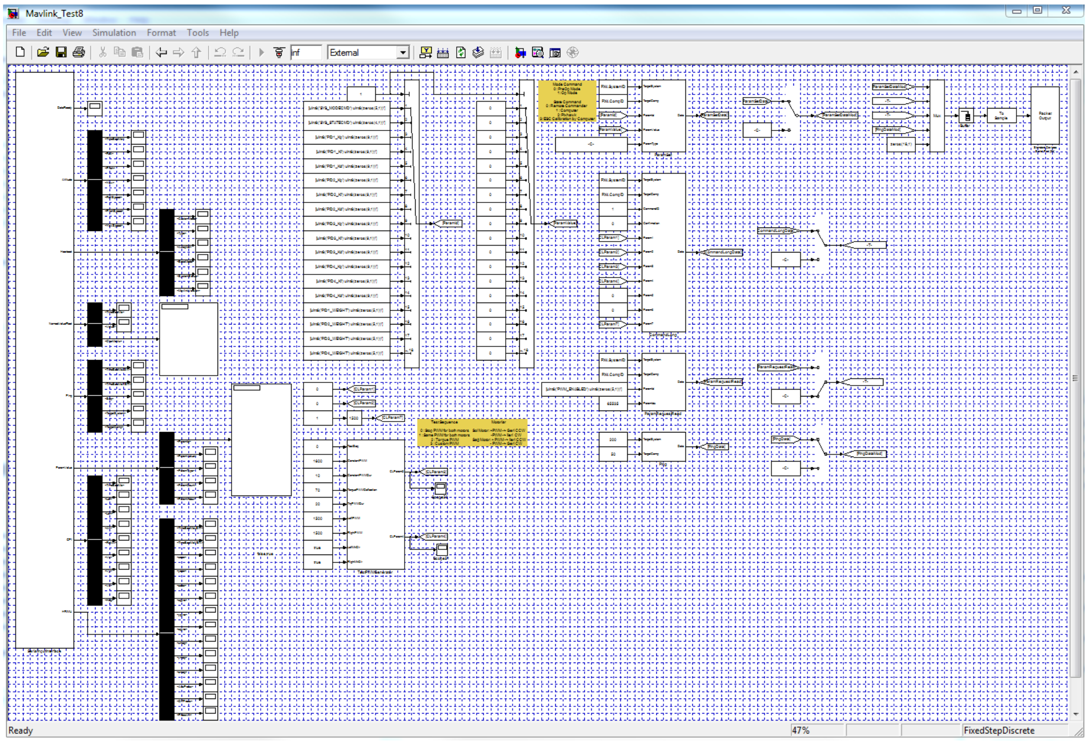


Figure 5.13: Ground station software

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

The main focus areas of this thesis are the modeling, the system identification and the controller design for sea surface vehicles. In order to establish an infrastructure for these studies, six DOF nonlinear mathematical model of a sea surface vehicle has been constructed based on the Fossen's vectorial model. An extensive literature review revealed that the Fossen's approach is suitable for this study. In the modeling study, the coordinate frame transformations and the kinematics,the rigid body and the added mass dynamics, the hydrodynamics and the hydrostatics concepts were utilized to create a complete and sufficient model structure. However, some essential assumptions were also made in order to simplify the model. Based on this model, a simulation of the vehicle has been prepared in MATLAB environment and the general attitude of it was verified and validated by designing simple test cases.

The following step was to prepare the experimental setup for validation, system identification and controller design work. The experimental setup consists of a ready to run (RTR) model boat, a fully functional advanced autopilot card and auxiliary hardware such as an experiment pool, measurement devices, ground station computer etc. In order to utilize the mentioned experimental setup efficiently, two fundamental softwares have been prepared. These are the embedded software running in the autopilot card and the ground station software. By using combination of these softwares, actuation of the boat, collecting and recording sensor measurement and state-mode-parameter switching can be easily and safely carried out.

After completion of the experimental setup work and considering the literature survey, a time domain, offline, deterministic, gray-box system identification methodology was decided to be used for the parameter identification of the sea surface vehicle. In this methodology, the efficient optimization algorithms such as Genetic Algorithm (GA) and Global Search Algorithm (GS) have been utilized together in order to obtain successful results. Considering this methodology, experiments are designed for full parameter identification. The number of experiments was adjusted so that validation of the identified parameters were also fulfilled. The validation tests show that the free motion tests are very successful for the identification of the corresponding parameters. Another observation coming from the validation tests is that although the methodology gives convergent results in some degree, the tests with actuation should be improved in order to identify especially the parameters that affect the maneuvering performance. These improvements will be given in the future work section. It should be noted that prior to utilization of the collected measurement data by the identification algorithms, this data has been processed by some algorithms in order to obtain more accurate and noiseless data. One of these is the navigation algorithm prepared and investigated by Kumru [47]. This work is the parallel independent study utilizing the same experimental setup as this thesis.

The next stage is the investigation and the design of controllers. A PID based piecewise controller and a Sliding Mode Controller (SMC) have been developed to control the surge speed and the yaw position of the vehicle. In design process of PID based controller, linearization of the nonlinear mathematical model is carried out at different operation points. Then, the controllable and the observable subspaces have been found for each linearization point in order to obtain the transfer functions. Afterwards, an optimization based design process was executed and the transfer functions of PIDs were obtained. Moreover, by using interpolation algorithms, other operating points were also handled. In the design stage of SMC, first of all, some simplifications and decoupling of the states have been performed on the nonlinear model. Based on this simplified model, the controller was manually designed taking into account the basic principles and the chattering issues. Performance of both controllers are tested on the nonlinear model for special scenarios that include cases with and without environmental disturbances. Results show that SMC is superior to PID based controller for all test cases according to RMS errors and transient response characteristics.

6.2 Future Work

The most obvious future study is the verification of the controllers on the experimental setup. Although both controllers display sufficient performance in the simulation, there may be significant deviations for the real case, especially under different disturbance conditions. Hence, further research is needed in order to fully validate the controllers. This study may reveal that more complex controller architectures are needed for sufficient disturbance rejection characteristics.

The future works for the system identification are very obvious. The experiments should be repeated considering items below for the best performance in maneuvering. However, it should be noted that the findings of this study have provided lots of insight about the future tests.

- The linear, spiral and zigzag motion tests should be conducted in a more controlled environment such as an olympic pool with very little disturbance effects such as wind and wind generated waves.
- Battery voltage should be recorded carefully for the thrust measurement tests.
- Force gauge with higher accuracy should be utilized for the thrust measurement tests.
- The linear free motion tests should be added to the list.
- In order to include the transient characteristics of the thrusters, a simple thruster model should be constructed and its parameters should be determined based on the look-up table values.
- The optimized results should be validated using more complex maneuvers such as full turns in order to be sure that all modes of the motion are excited.
- Effects of coupled six DOF motion should be carefully investigated.

Considering the experimental setup, the system identification study and the controller design work as infrastructures, fully autonomous guidance of this sea surface vehicle would be a good research area. Actually, two more model boats that have the same characteristics as the utilized one have been ordered. The construction of them is continuing nowadays. When they are ready, an improved infrastructure for the coordinated and the collaborated guidance of these vehicles can be studied. If this study is performed, the autonomy level of the ship will significantly increase. This guidance work may make contributions to the research fields such as mapping, patrolling, evasion, pursuit and reconnaissance, etc.

REFERENCES

[1] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[2] T.I. Fossen and T. Perez. Time-domain models of marine surface vessels for simulation and control design based on seakeeping computations. *Proc. of the IFAC MCMC'06*, 2006.

[3] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, 1994.

[4] Discretizing a pid controller. `https://www.researchgate.net`. Accessed: August 2015.

[5] E.W. McGookin. *Optimization of sliding mode controllers for marine applications: a study of methods and implementation issues*. PhD thesis, University of Glasgow, UK, September 1997.

[6] 3dr platform update. `http://3drobotics.com/`. Accessed: August 2015.

[7] Pixhawk autopilot card peripherals. `http://copter.ardupilot.com/wiki/common-pixhawk-wiring-and-quick-start/`. Accessed: August 2015.

[8] Pixhawk autopilot card main web page. `https://pixhawk.org/start`. Accessed: April 2015.

[9] Pacific islander tugboat. `https://scalespeed.com`. Accessed: August 2015.

[10] Hobby wing products. `http://www.hobbywing.com/product.asp/`. Accessed: August 2015.

[11] Mavlink communication protocol. `http://qgroundcontrol.org/`. Accessed: June 2015.

[12] A. Motwani. A survey of uninhabited surface vehicles. *MIDAS Technical Report*, 2012.

[13] D. Rodrigues et al. A study of the feasibility of autonomous surface vehicles. *Interactive Qualifying Project*, 2012.

[14] T.I. Fossen. *Marine Control Systems, Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 2002.

[15] J.J. Craig. *Introduction to Robotics*. Prentice Hall, 2005.

[16] T.I. Fossen and O.E. Fjellstad. Nonlinear modeling of marine vehicles in 6 degrees of freedom. *International Journal of Mathematical Modeling of Systems*, 1(1):17–27, 1995.

[17] T.I. Fossen and S.P. Berge. On the properties of the nonlinear ship equations of motion. *Journal of Mathematical and Computer Modeling of Dynamical Systems*, 6(4):365–381, 2000.

[18] C.R. Sonnenburg. *Modeling, identification and control of an unmanned surface vehicle*. PhD thesis, Virgina Polytechnic Institute and State University, December 2012.

[19] L. Ljung. *System Identification: Theory for the User*. Prentice hall, 1987.

[20] J. Juang. *Applied System Identification*. Pearson, 1994.

[21] N. Oliver. *Nonlinear System Identification, From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, 2001.

[22] K.Y. Wong and E. Polak. Identification of linear discrete time systems using the instrumental variable method. *IEEE Transaction on Automatic Control*, 12(6):707–718, 1967.

[23] K. Nomoto et al. On the steering qualities of ships. *Int. Shipbuilding Prog.*, 4(35), 1957.

[24] C. Tzeng and J. Chen. Fundamental properties of linear ship steering dynamic models. *J. Mar. Sci. Tech.*, 7(2):79–88, 1999.

[25] J.M.J. Journee. A simple method for determining the maneuvering indices k and t from zigzag trial data. *Report No: 0267*, 1970.

[26] H.L. Brinati and A.R. Neto. Application of the extended kalman filtering to the identification of the ship hydrodynamic coefficients. *Third Brazilian Congress of Mechanical Engineering*, 1975.

[27] W.W. Zhou and M. Blanke. Nonlinear recursive prediction error method applied to identification of ship steering dynamics. *Eight Ship Control Systems Symposium*, 1987.

[28] M.H. Casado et al. Identification of nonlinear ship model parameters based on the turning circle tests. *J. Ship Res. 51*, 2007.

[29] M.A. Abkowitz. Measurement of ship resistance, powering and maneuvering coefficients from simple trails during a regular voyage. *Trans. SNAME 96*, 1988.

[30] G.I. Silman. Creation of mathematical models for ship maneuvering motion on the basis of full scale trails data. *Propulsion and Steering Arrays and Ship Maneuvering*, pages 82–97, 1988.

[31] A. DiMascio et al. Investigation of twin-screw naval maneuverability behavior. *J. Ship Res*, 55(4):221–248, 2011.

[32] H.K. Yoon and K.P. Rhee. Identification of hydrodynamics coefficients in ship maneuvering equations of motion by estimation-before-modeling technique. *Ocean Engineering*, 30(18):2379–2404, 2003.

[33] L.P. Perera et al. System identification of nonlinear vessel steering. *Journal of Offshore Mechanics and Arctic Engineering*, 137(3):031302, 2015.

[34] R.P. Selvam et al. A frequency domain system identification method for linear ship maneuvering. *International Ship Building Progress*, 52(1):5–27, 2005.

[35] K.T. Tran et al. Identification of hydrodynamic coefficients from sea trails for ship maneuvering simulations. *Transport Research Arena*, 2014.

[36] N. Kariotoglou. Steer-by-wire of electric boats with two engines, modeling and control. Master's thesis, Swiss Federal Institute of Technology Zurich, September 2010.

[37] N. Minorsky. Directional stability of automatic steered bodies. *Journal of American Society of Naval Engineers*, 42(2):280–309, 1922.

[38] A. Grovlen and T.I. Fossen. Nonlinear control of dynamic positioned ships using only position feedback: An observer backstepping aproach. *IEEE Conference on Decision and Control*, 3:3388–3393, 1996.

[39] K.D. Do et al. Robust adaptive path following for underactuated ships. *Automatica: A Journal of International Federation of Automatic Control*, 40(6):929–944, 2004.

[40] M. Reyhanoglu. Exponential stabilization of an underactuated autonomous surface vessel. *Automatica: A Journal of International Federation of Automatic Control*, 33(12):2249–2254, 1997.

[41] M. Bao-li. Global k-exponential asymptotic stabilization of underactuated surface vessels. *Systems and Control Letters*, 58(3):194–201, 2009.

[42] Y.L. Liao et al. Backstepping dynamical sliding mode control method for the path following of the underactuated surface vessel. *Procedia Engineering*, 15:256–263, 2011.

[43] W. Naeem et al. Soft computing design of a linear quadratic gaussian controller for an unmanned surface vehicle. *14th Mediterranean Conference on Control and Automation*, pages 1–6, 2006.

[44] K. Ahıska. Control and guidance of an unmanned sea surface vehicle. Master's thesis, Middle East Technical University, September 2012.

[45] E. Alfaro-Cid et al. Genetic algorithm optimization of decoupled sliding mode controller: simulated and real results. *Control Engineering Practice 13*, 13(6):739–748, 2005.

[46] S. Sutulo and C.G. Soares. An algorithm for offline identification of ship maneuvering mathematical models from free-running tests. *Ocean Engineering*, 79:10–25, 2014.

[47] M. Kumru. Navigation and control of a sea surface vehicle. Master's thesis, Middle East Technical University, September 2015.

[48] Matlab 2011b documentation, genetic algorithm, 2015. Accessed: 2015-07-24.

[49] Matlab 2011b documentation, global search algorithm, 2015. Accessed: 2015-08-05.

[50] K.R. Muske and H. Ashrafiuon. Identification of a control oriented nonlinear dynamic usv model. *American Control Conference*, pages 562–567, 2008.

[51] A.Packard et al. *Dynamics Systems and Feedback*. Lecture Notes, 2002.

[52] H. J. Braslavsky. *Control System Design Lecture 16, Controllability and Observability, Canonical Decomposition*. 2015.

[53] Matlab 2011b documentation, pidtune, 2015. Accessed: August 2015.

[54] J. Kaustky et al. Robust pole assignment in linear state feedback. *International Journal of Control*, 41(5):1129–1155, 1985.

[55] E. Alfaro-Cid et al. A comparative study of genetic operators for controller parameter optimization. *Control Engineering Practice 17*, 17(1):185–197, 2009.

APPENDIX A

MODEL VALIDATION TESTS

For each test, below figures are provided.

- x, y, z positions,
- Roll, pitch, yaw angles,
- x, y, z rates,
- Roll, pitch, yaw angles rates.

A.1 Zero Input, Zero State Response

Results are given in Figures A.1, A.2, A.3, A.4.

A.2 Zero Input, Nonzero State Response

Results can be found in Figures A.5, A.6, A.7, A.8.

A.3 Equal Input, Zero State Response

Figures A.9, A.10, A.11, A.12 show the results.

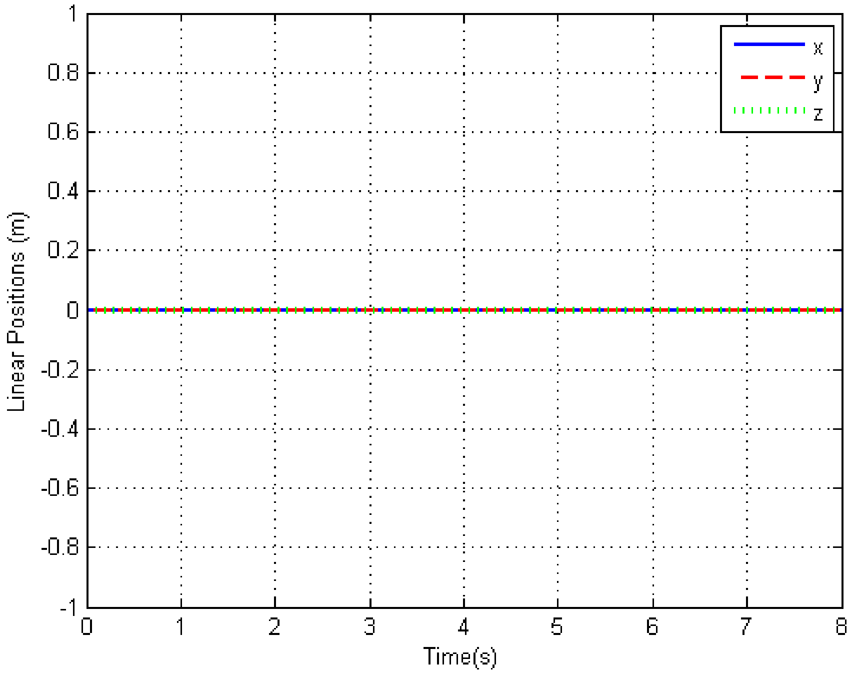


Figure A.1: x, y, z positions

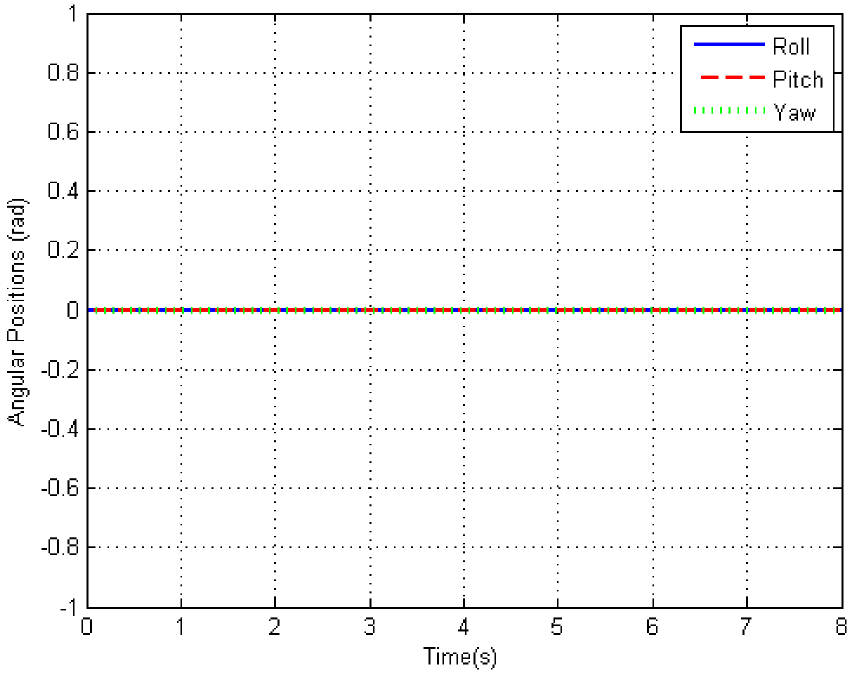


Figure A.2: Roll, pitch, yaw positions

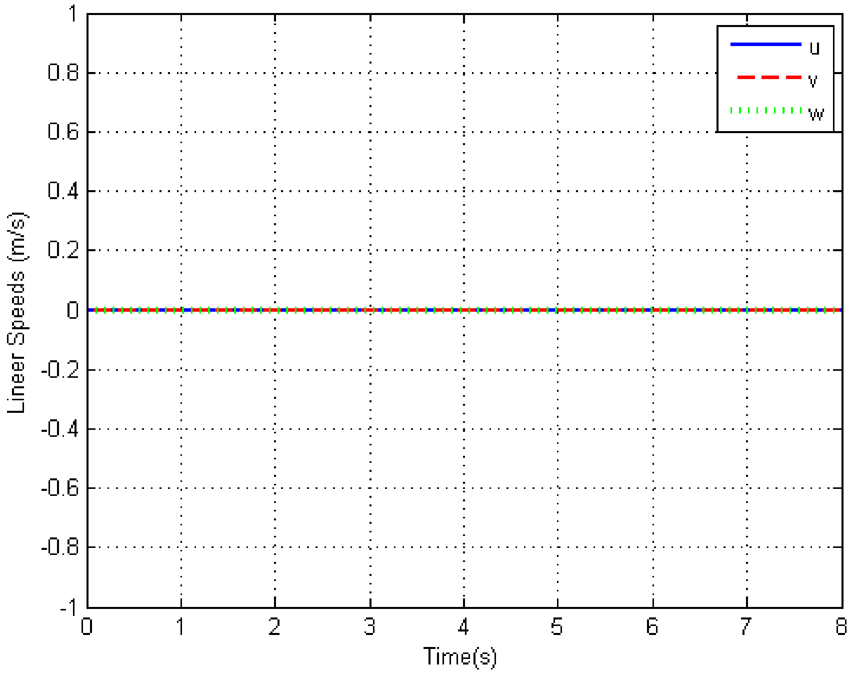


Figure A.3: x, y, z speeds

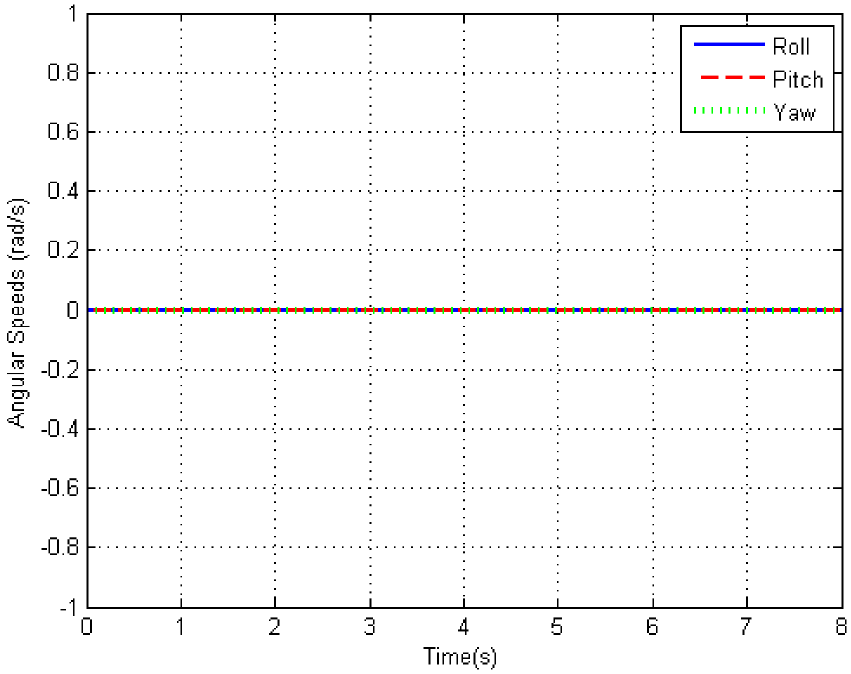


Figure A.4: Roll, pitch, yaw rates

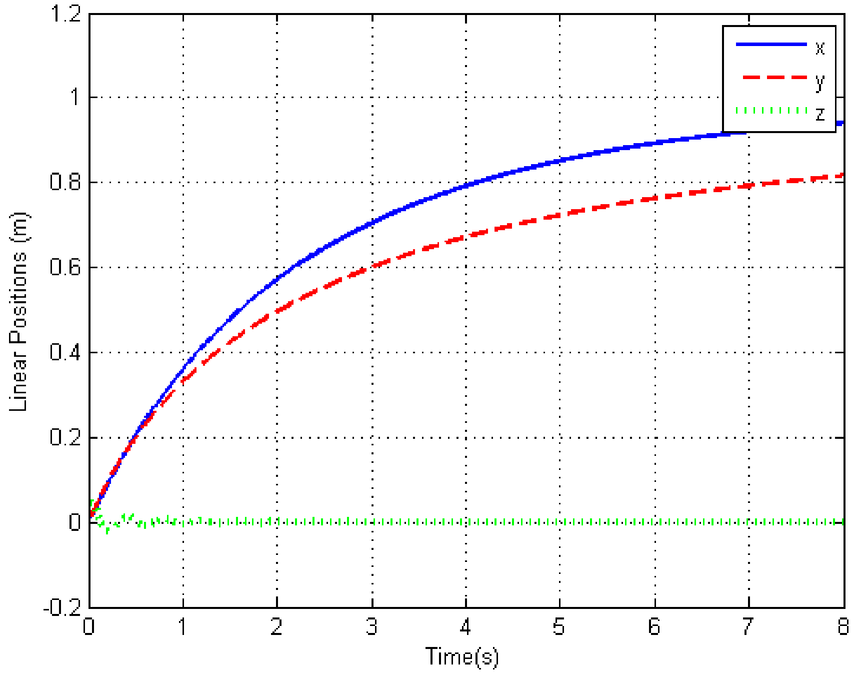


Figure A.5: x, y, z positions

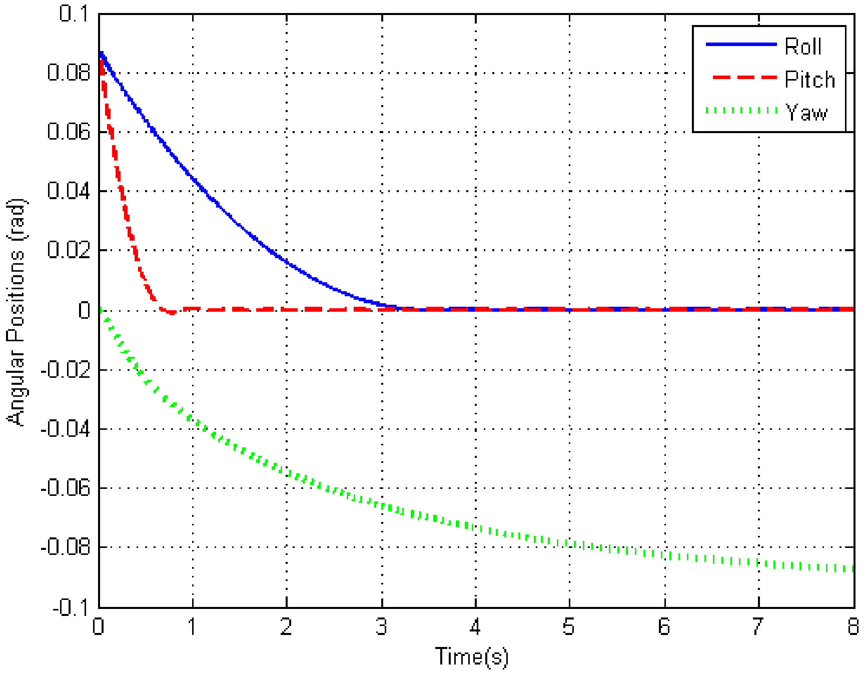


Figure A.6: Roll, pitch, yaw positions

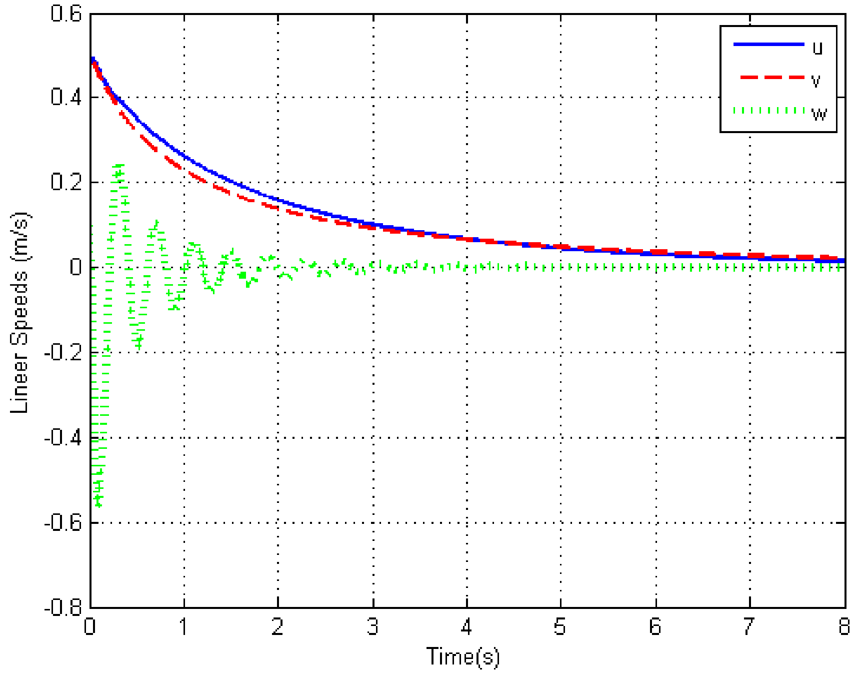


Figure A.7: x, y, z speeds

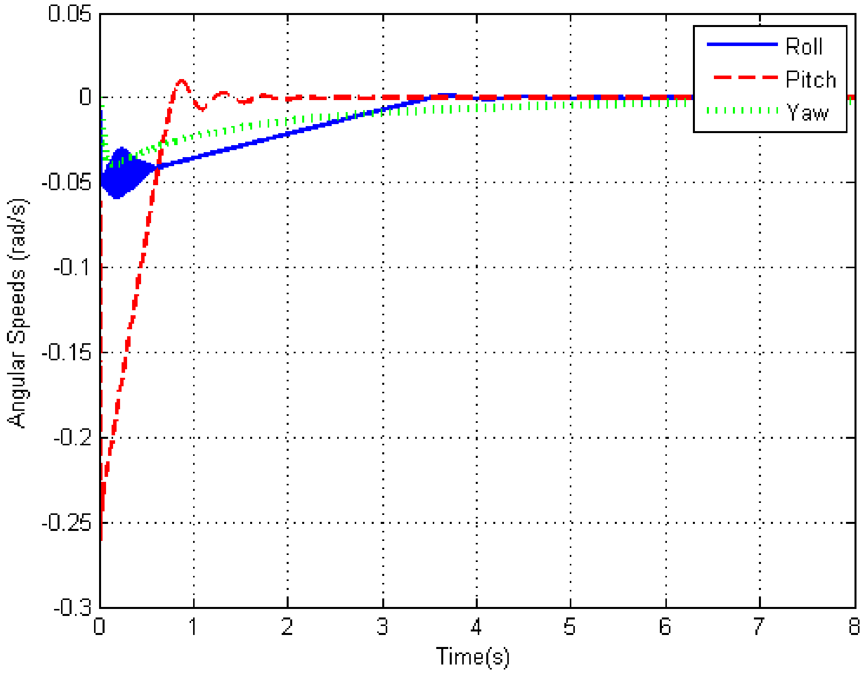


Figure A.8: Roll, pitch, yaw rates

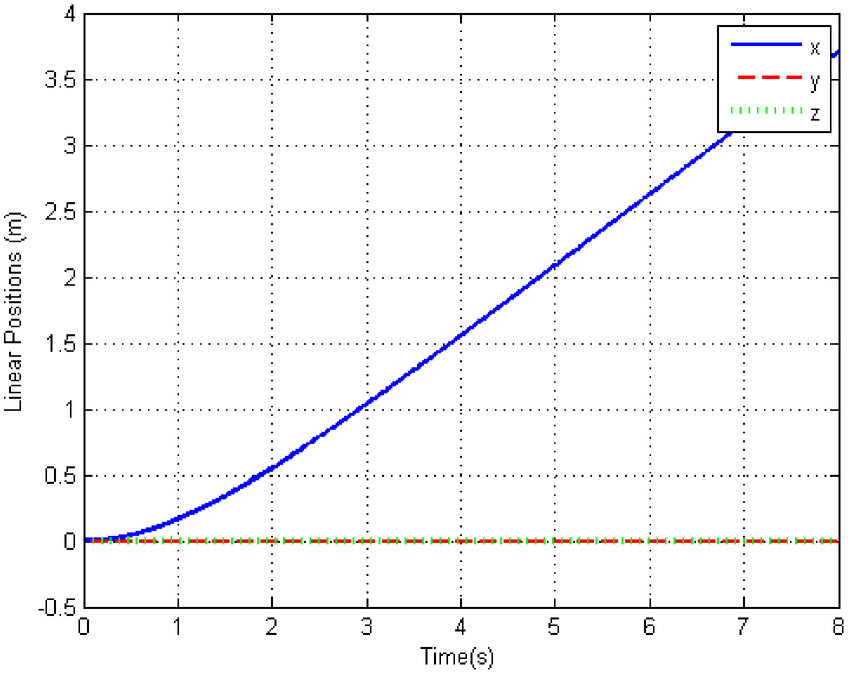


Figure A.9: x, y, z positions

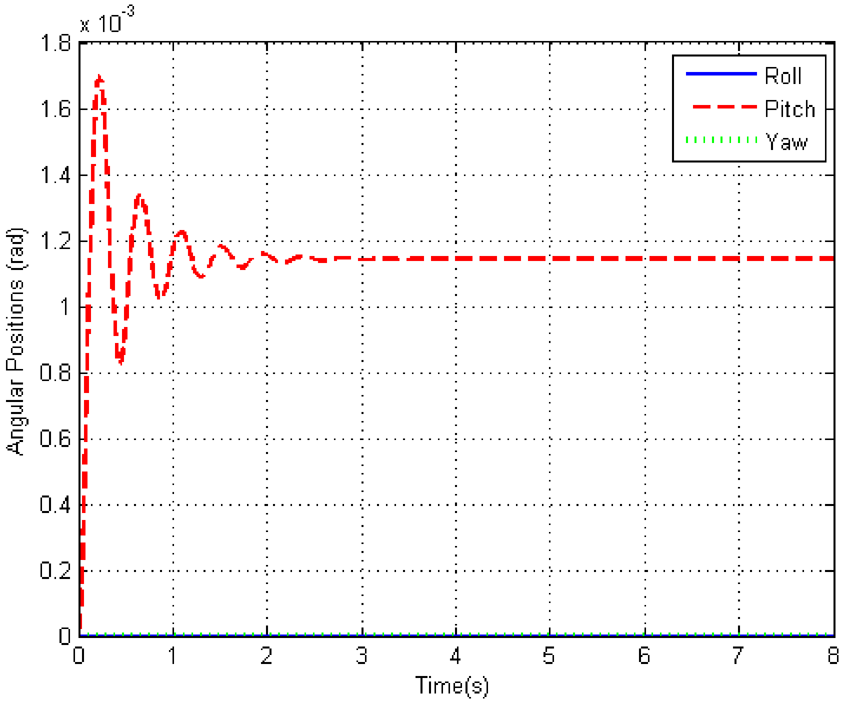


Figure A.10: Roll, pitch, yaw positions

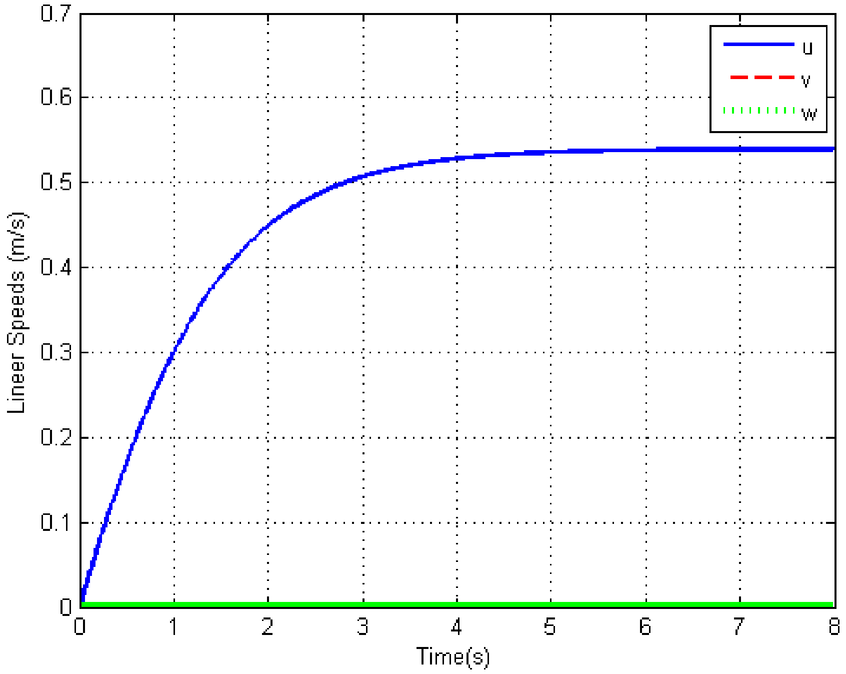


Figure A.11: x, y, z speeds

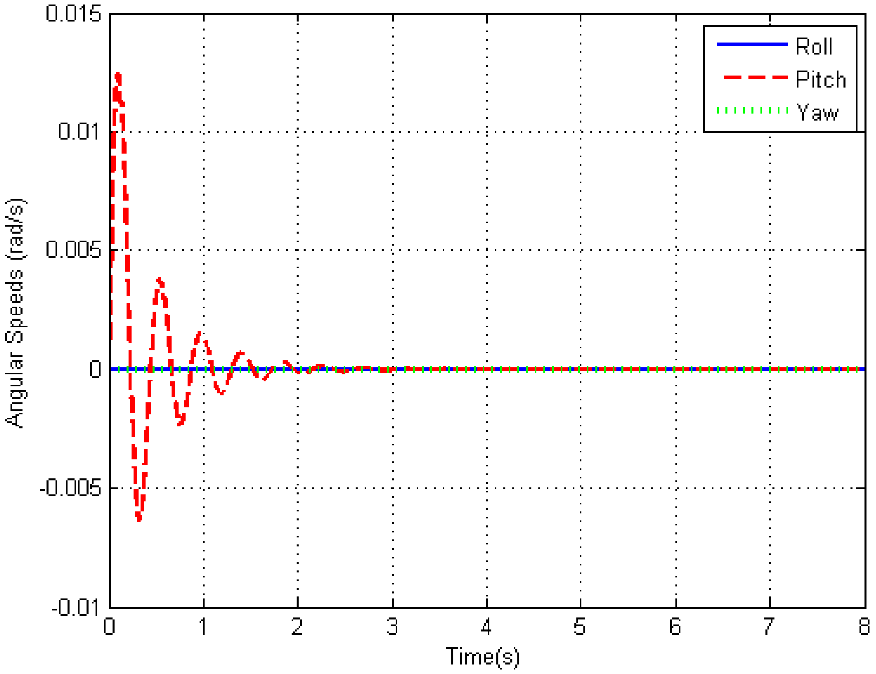


Figure A.12: Roll, pitch, yaw rates

A.4 Non-equal Input, Zero State Response

Figures A.13, A.14, A.15, A.16 contain the results.

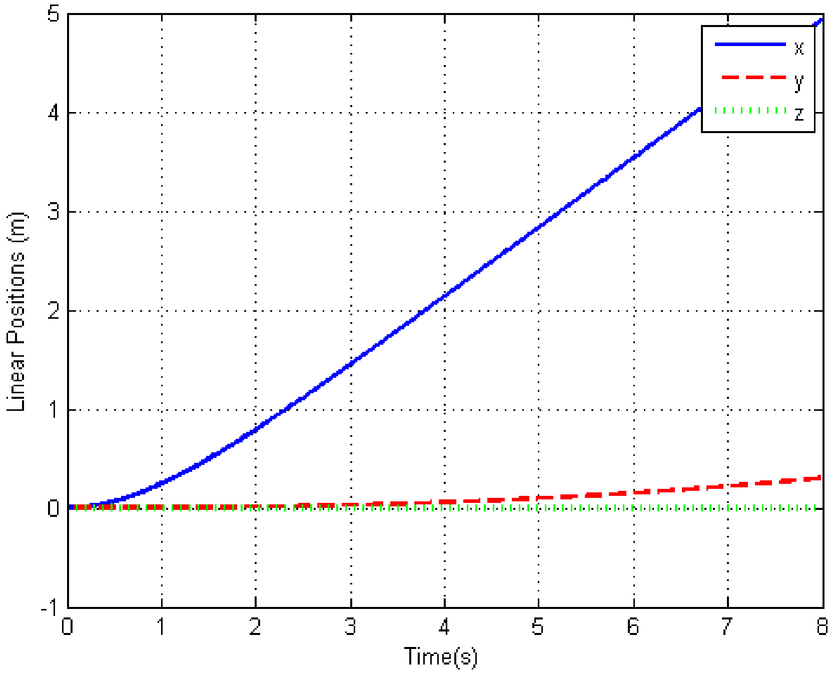


Figure A.13: x, y, z positions

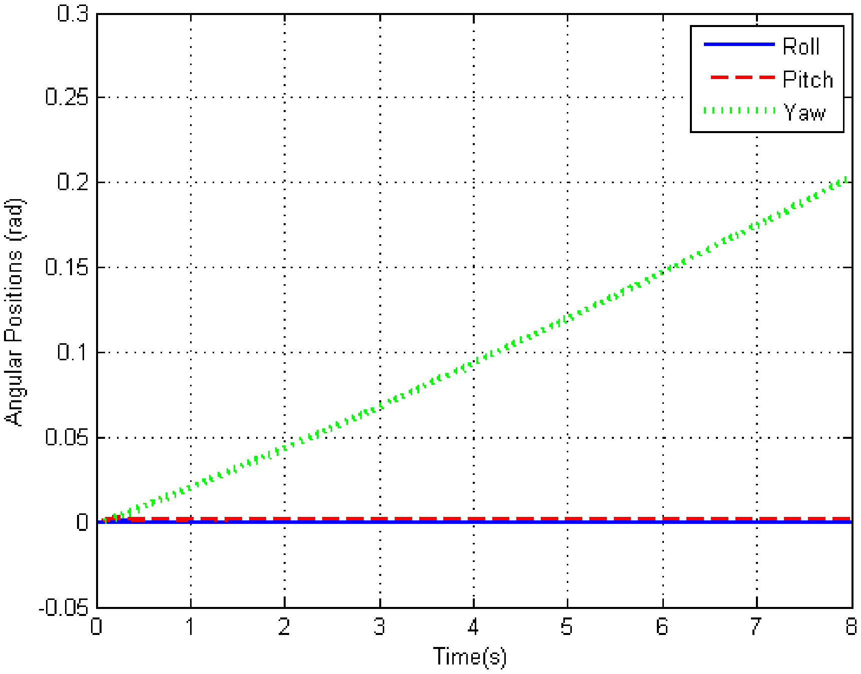


Figure A.14: Roll, pitch, yaw positions

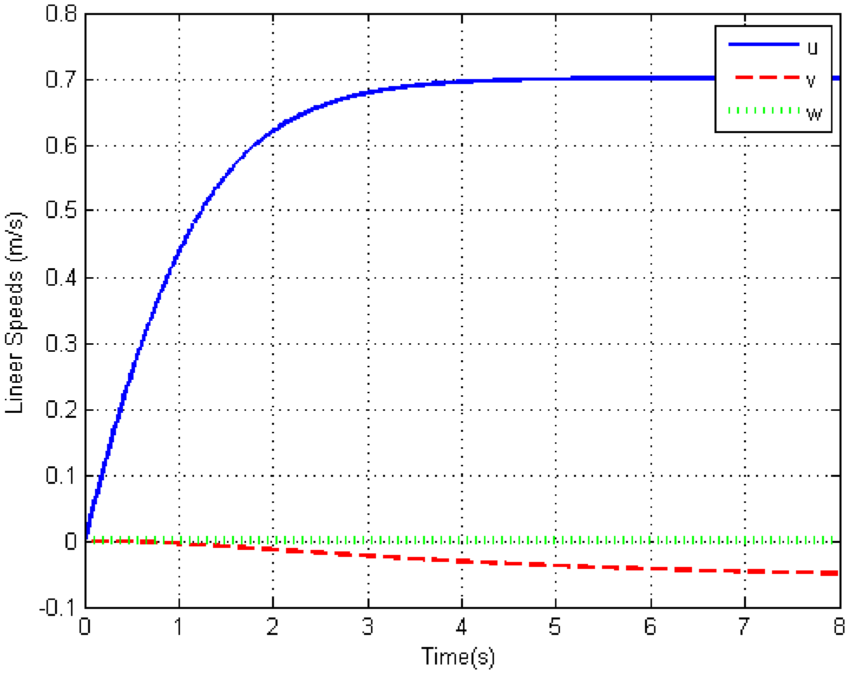


Figure A.15: x, y, z speeds

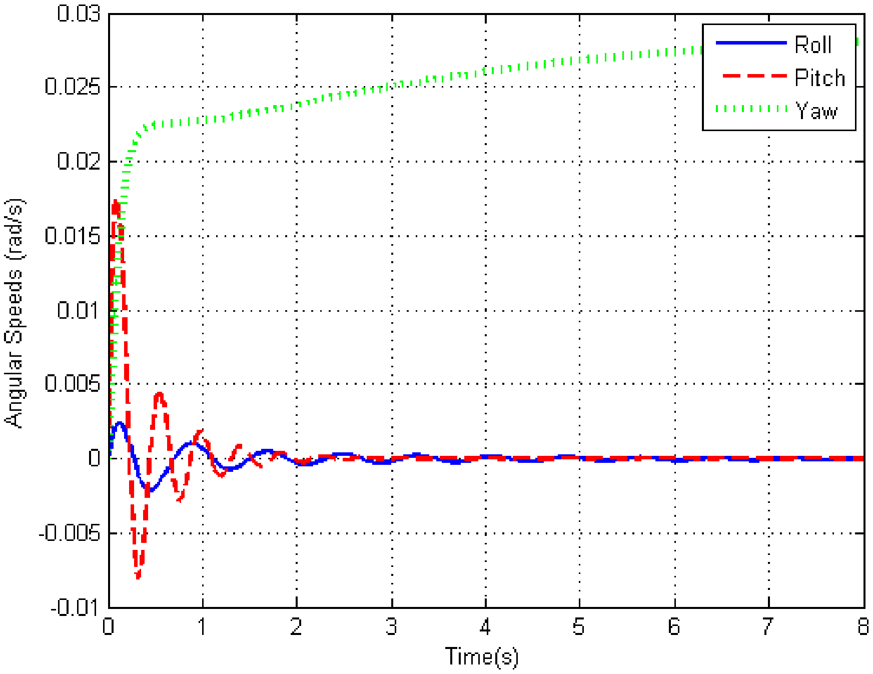


Figure A.16: Roll, pitch, yaw rates