

CONTROL AND GUIDANCE OF AN UNMANNED SEA SURFACE VEHICLE

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

BY

KENAN AHISKA

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

SEPTEMBER 2012

Approval of the thesis:

**CONTROL AND GUIDANCE OF AN UNMANNED SEA SURFACE
VEHICLE**

submitted by **KENAN AHISKA** 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. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

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

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____

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

Assist. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU _____

M.Sc. Bülent Bilgin
ASELSAN Inc. _____

Date: 11/09/2012

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 : Kenan AHISKA

Signature :

ABSTRACT

CONTROL AND GUIDANCE OF AN UNMANNED SEA SURFACE VEHICLE

Ahıska, Kenan

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kemal Leblebiciođlu

September 2012, 123 pages

In this thesis, control and guidance algorithms for unmanned sea surface vehicles are studied. To design control algorithms of different complexity, first a mathematical model for an unmanned sea surface vehicle is derived. The dynamical and kinematical equations for a sea surface vehicle are obtained, and they are adapted to real life conditions with necessary additions and simplifications. The forces and torques effecting on the vehicle are investigated in detail. Control algorithms for under-actuated six degrees-of-freedom model are designed. PID and LQR controllers are implemented to attain desired surge speed and yaw position. The autopilots are designed and their performances are compared. Based on the autopilots, a guidance algorithm is implemented to achieve desired motions of the vehicle. An obstacle avoidance algorithm is proposed for safe motion among the obstacles. A next-point generation algorithm is designed to direct the vehicle to the most appropriate next way-point if the one ahead is missed. The effects of disturbances on the motion of the vehicle are studied thoroughly on simulation results. PID controller for an unmanned sea surface vehicle is implemented on ArduPilot Mega v1.4 cart controlling a Traxxas Spartan model boat. The performance of the controller is validated. Simulations and experimental results are provided.

Keywords: Sea surface vehicles, mathematical modeling, PID controller, LQR controller, autopilot, guidance, obstacle avoidance.

ÖZ

BİR İNSANSIZ SU ÜSTÜ ARACININ GÜDÜM VE DENETİMİ

Ahıska, Kenan

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Eylül 2012, 123 sayfa

Bu tezde, insansız su üstü araçları için güdüm ve denetim algoritmaları çalıştırılmıştır. Değişik karmaşıklık seviyelerinde denetim algoritmalarını tasarlamak için, öncelikle bir insansız su üstü aracı için bir matematiksel model elde edilmiştir. Su üstü araçları için dinamik ve kinematik denklemler elde edilerek ve gerekli eklemeler ve sadeleştirmelerle gerçek hayat şartlarına uyarlanmıştır. Araçlara etkiyen kuvvet ve momentler ayrıntılı bir şekilde incelenmiştir. Eksik çalıştırıcılı altı serbestlik dereceli model için denetim algoritmaları tasarlanmıştır. İleri hız ve sapma açısı isteklerini elde etmek için PID ve LQR denetimcileri uygulanmıştır. Otopilotlar tasarlanmış ve performansları kıyaslanmıştır. Otopilotları temel alarak, aracın istenilen yönelimi yapması için bir güdüm algoritması uygulanmıştır. Engeller arasında güvenli hareket için bir engelden kaçma algoritması önerilmiştir. Eğer önündeki nokta ıskalanmışsa, aracı sıradaki en uygun hedef noktaya yönlendirecek bir sonraki nokta üretici algoritma tasarlanmıştır. Simülasyon sonuçları üzerinde, bozucuların aracın hareketi üzerindeki etkileri ayrıntılı olarak incelenmiştir. Bir Traxxas Spartan model teknesini kontrol eden ArduPilot Mega v1.4 kartı üzerinde PID denetimcisi uygulanmıştır. Denetimcinin performansı donanımsal olarak doğrulanmıştır. Benzetimler ve deneysel sonuçlar verilmiştir.

Anahtar Kelimeler: Su üstü araçlar, matematiksel modelleme, PID denetimcisi, LQR denetimcisi, otopilot, güdüm, engelden kaçma.

To lands and people of AHISKA,

ACKNOWLEDGEMENTS

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

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

I would also like to thank Turkish Scientific and Technological Research Council (TÜBİTAK) for their financial support during my M.Sc. studies.

I would like to thank my colleagues Ümit Büyükkeleş, Uđur Dođan Gül, and Aycan Uçar for their contributions throughout the preparation of this thesis.

I would like to present my appreciations to Halil Onur Şirin, Serkan Nas and İzzet Kađan Erünsal for their supports in realization of communication part of the thesis.

I would also like to thank Nejat Ulusal for his support in procurement of experimental setup.

I would also like to express my appreciations to İhsan Çetin for his technical support.

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

TABLE OF CONTENTS

ABSTRACT.....	IV
ÖZ.....	V
ACKNOWLEDGEMENTS	VII
TABLE OF CONTENTS	VIII
LIST OF TABLES	X
LIST OF FIGURES	XI
CHAPTERS	
1. INTRODUCTION.....	1
1.1 Motivation of the Thesis	1
1.2 Academic Review	2
1.3 Organization of the Thesis.....	7
2. MATHEMATICAL MODELING OF SEA SURFACE VEHICLES.....	9
2.1 Introduction	9
2.2 Coordinate Frames and Transformations.....	10
2.3 Rigid Body Dynamics	18
2.4 Added Mass Dynamics	25
2.5 Damping Factors.....	27
2.6 Restoring Forces and Moments	30
2.7 Forces Induced by Thruster	33
2.8 Forces Induced by Rudder	35
2.9 Air Drag and Wind Forces.....	38
2.10 Forces due to Water Current.....	39
2.11 Conclusion	40

3. AUTOPILOT DESIGN FOR SEA SURFACE VEHICLES.....	42
3.1 Introduction	42
3.2 Fundamentals for the Autopilot Design.....	43
3.3 Proportional-Integral-Derivative (PID) Controller.....	47
3.4 Linear Quadratic Regulator (LQR) Controller	55
3.5 Design of Yaw Autopilot.....	60
3.6 Comparison between PID and LQR Autopilots	62
3.7 Conclusion.....	68
4. GUIDANCE DESIGN FOR SEA SURFACE VEHICLES.....	70
4.1 Introduction	70
4.2 Way-Point Guidance Algorithm.....	72
4.3 Obstacle Avoidance Algorithm	74
4.4 Next-Point Generation Algorithm	79
4.5 Effects of Disturbances.....	81
4.6 Conclusion.....	89
5. EXPERIMENTAL WORK.....	91
5.1 Introduction	91
5.2 Experimental Setup.....	92
5.3 Experimental Results.....	101
5.4 Conclusion.....	108
6. CONCLUSION AND FUTURE WORKS.....	112
6.1 Conclusion.....	112
6.2 Future Works	114
REFERENCES.....	116
APPENDICES	
A. DERIVATION OF ADDED MASS PARAMETERS.....	120
B. DERIVATION OF DAMPING PARAMETERS.....	122

LIST OF TABLES

TABLES

Table 2.1. Notation for water vehicle motion	11
Table 3.1. Yaw position-surge velocity limits for the yaw autopilot.....	61
Table 4.1. Yaw position-surge velocity limits for the yaw autopilot.....	81

LIST OF FIGURES

FIGURES

Figure 2.1. Sea surface vehicle [1].....	10
Figure 2.2. Reference frames [1]	13
Figure 2.3. SIMULINK model for 6 DOF kinematic equations	17
Figure 2.4 Rudder angles and descriptions [38].....	36
Figure 2.5 Mathematical model of a sea surface vehicle in SIMULINK	41
Figure 3.1 PID controller for surge speed for $u = 1, \psi = 0$	48
Figure 3.2. PID auto-tuning algorithm flowchart	50
Figure 3.3. Model developed to determine weights of as single point PID controller.....	51
Figure 3.4. Weight determining algorithm flow chart	53
Figure 3.5. PID controller developed for USSV	54
Figure 3.6. PID controller step responses for $u = 1, \psi = \pi/4$	55
Figure 3.7. LQR controller design	59
Figure 3.8. LQR controller step responses for $u = 1, \psi = \pi/4$	60
Figure 3.9. Yaw autopilot algorithm.....	62
Figure 3.10. Closed-loop eigenvalues of autopilots for $u = 1, \psi = \pi/4$	64
Figure 3.11. Comparison of surge velocity outputs of autopilots under speed hold.....	65
Figure 3.12. Comparison of yaw position outputs of autopilots under speed hold.....	66
Figure 3.13. Closed-loop system eigenvalues for small perturbation across $u = 1, \psi = \pi/4$	68
Figure 4.1. Guidance algorithm flowchart.....	73
Figure 4.2. Simulation results for guidance algorithm.....	74
Figure 4.3. Obstacle avoidance algorithm flowchart	77
Figure 4.4. Obstacle avoidance algorithm simulation-1	78
Figure 4.5. Obstacle avoidance algorithm simulation-2	79
Figure 4.6. Next point generation algorithm.....	80
Figure 4.7. Single direction wind disturbed guidance system	82

Figure 4.8. Zoom in of single direction wind disturbances in Figure 4.7 -1.....	83
Figure 4.9. Zoom in of single direction wind disturbances in Figure 4.7 -2.....	84
Figure 4.10. Cross wind disturbed guidance system.....	85
Figure 4.11. Single direction current disturbed guidance system.....	86
Figure 4.12. Zoom in of single direction current disturbances in Figure 4.11.....	87
Figure 4.13. Cross current disturbed guidance system.....	88
Figure 5.1. Outer view of Traxxas Spartan model 5707 boat [44].....	93
Figure 5.2. A view from inside of the boat [44].....	93
Figure 5.3. ArduPilot Mega v1.4 view [45].....	95
Figure 5.4. Top view of Oilpan IMUShield [45].....	96
Figure 5.5. Autopilot card together with IMU Shield [45].....	97
Figure 5.6. MATLAB GUI prepared for experiments.....	98
Figure 5.7. SIMULINK External Mode model for tests.....	99
Figure 5.8. Inside view of the vehicle under test.....	100
Figure 5.9. Experiment results for $u = 1$ and $\psi = 0$	102
Figure 5.10. Experiment results for $u = 1$ and $\psi = 30$	102
Figure 5.11. Experiment results for $u = 1$ and $\psi = -30$	103
Figure 5.12. Errors between feedbacks and command for $u = 1$ with different yaw position commands.....	103
Figure 5.13. Experiment results for $u = 2$ and $\psi = 0$	104
Figure 5.14. Experiment results for $u = 2$ and $\psi = 30$	105
Figure 5.15. Experiment results for $u = 2$ and $\psi = -30$	105
Figure 5.16. Errors between feedbacks and command for $u = 2$ with different yaw position commands.....	106
Figure 5.17. Initial position of the vehicle.....	109
Figure 5.18. Yaw position of 0 degrees, vehicle hits the target.....	110
Figure 5.19. Yaw position of 30 degrees, vehicle is headed to 30 degrees.....	110
Figure 5.20. Yaw position of -30 degrees, vehicle is headed to -30 degrees.....	111

CHAPTER 1

INTRODUCTION

1.1 Motivation of the Thesis

Development of science and technology with increased notion in human rights provided unmanned vehicles as an address to the missions endangering human life. As knowledge on motion of objects in water accumulates, design and implementation of unmanned sea vehicles are realized.

Unmanned sea vehicles can be grouped into two categories: unmanned underwater vehicles and unmanned sea surface vehicles. These vehicles are capable to operate with various degrees of autonomy. The unmanned sea surface vehicles displace water at rest, and moves in continuous contact to the sea surface.

Unmanned sea surface vehicles have been utilized in many military and civil missions. With proper equipment, these vehicles can be used for reconnaissance, detection, classification, localization or identification of objects on the sea surface, lands near to sea or underwater. The missions for attacking targets as a weapon platform or torpedo can be achieved as well. Similarly, unmanned sea surface vehicles (USSV) can be used to sense or jam signals from other sources.

In this thesis, control and guidance of unmanned sea surface vehicles are to be studied. To develop an accurate control algorithm, a detailed mathematical modeling for a USSV is derived. This model includes dynamical and kinematical perspectives of motion of the vehicle as well as the effecting forces and torques. The actuators in the modeled vehicle are rudder and thruster.

The controllers allow the autopilots to rule the motion of the vehicle in a desired forward speed and yaw position. The guidance algorithms create the commands for autopilots according to the mission of the vehicle with a way-point trajectory tracking approach.

In their motions under control of the guidance algorithm, the vehicles may encounter obstacles they should be avoiding or they may miss a certain way point. For these cases separate algorithms are to be developed.

In this thesis, these algorithms are designed and simulated. One of the controllers developed for the model are implemented on a model boat. The experimental results are used to support and validate the algorithms in use.

In the lights of these results, upper level controllers for USSV to fulfill various mission requirements can be developed. As future studies, guidance of USSV can be experimentally validated, a detailed navigation system can be developed and a mission assignment algorithm for USSV groups can be studied.

1.2 Literature Review

From Archimedes, motion of objects on surface of water is one of the most important problems in mathematics and physics. Literature on mathematical modeling of a sea surface vehicle and techniques in control and guidance of these vehicles can be summarized in [1].

Mathematical modeling for sea surface vehicles is first approached by the classical Newtonian method:

$$\mathbf{M}\dot{\mathbf{v}} = \sum_{i=1}^n \mathbf{F}_i \quad (1.1)$$

In (1.1), \mathbf{M} is system inertia matrix, and $\dot{\mathbf{v}}$ is the generalized acceleration vector:

$$\mathbf{v} = [u, v, w, p, q, r]^T \quad (1.2)$$

In (1.2), (u, v, w) are linear velocities in surge, sway and heave, and (p, q, r) are angular velocities in roll, pitch and yaw. The linearly super-positioned forces and moments in (1.1) are as follows.

$$\mathbf{F}_i = [X_i, Y_i, Z_i, K_i, M_i, N_i]^T \quad i = 1, \dots, n \quad (1.3)$$

The equation set (1.1) - (1.3) uses SNAME notation [2]. The classical model consists of 72 entries for $n = 1$ in (1.1), only linear damping and added mass terms are covered. The nonlinear components should be included in the representation and this will add hundreds of elements to the model [3]. Therefore, the classical model with large number of coefficients becomes complicated; furthermore, the accuracy in determination of the coefficients can be inadequate [1].

Another mathematical model for marine vessels is derived by Fossen [4]. This model is based on the general robot model [5], [6]:

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

In (1.4), \mathbf{M} is system inertia matrix, \mathbf{C} is Coriolis matrix, $\boldsymbol{\tau}$ torques and \mathbf{q} is a vector of joint angles. This equation uses 6 degrees of freedom (DOF) vectorial representation. The model is further developed in various studies [7], [8], and [9]. The vectorial model is finalized as [10]:

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

where

$$\begin{aligned} \mathbf{v} &= [u, v, w, p, r, q]^T \\ \boldsymbol{\eta} &= [x, y, z, \phi, \theta, \psi]^T \end{aligned} \quad (1.6)$$

In (1.5), \mathbf{D} represents damping matrix, \mathbf{v} is vector of linear and angular velocities, and $\boldsymbol{\eta}$ is vector of position and Euler angles.

Model in [10] is not only a less complicated representation compared to the classical model, but it also possesses advantages in controller design due to its properties like symmetry, skew-symmetry, and positiveness in matrices. These properties make it easier to construct multi-input-multi-output (MIMO) controllers with reductions in model with basic matrix properties. The fundamentals of mathematical modeling are based on these equations as regards to portions of literature, as well as in this thesis.

Studies on control design for sea surface vehicles have started in early 20th century. The most common closed loop control problem solution is a three-term control law named as Proportional-Integral-Derivative (PID) control. A detailed analysis of PID controller for sea surface vehicles is formulated in 1922 [11]. A single-input-single-output (SISO) control with heading (yaw position) feedback from a gyrocompass was proposed. In this study, the autopilot compares the heading with set-point reference signal and computes the rudder command for corrective rudder servo action. Following 1995, nonlinear PID controllers, passive observer design and observer backstepping designs have been applied in the control of sea vessels [8], [12], and [13].

Early in 1960s, Kalman filtering and linear quadratic (LQ) optimal controllers were postulated [14], [15]. H-infinity control design techniques became another alternative in autopilot design starting from 1980s [16]. The advantage of these techniques over PID controllers is their frequency dependent notch filtering of 1st order wave induced disturbances [1].

Controller design for sea surface vehicles is one of the most studied topics in 21st century as well. In Do, *et.al*, a robust adaptive controller for under-actuated ships is proposed [17]. The nonlinear controller postulated follows a path in a desired speed in presence of environmental disturbances such as wave, wind and ocean current. The method applied uses Lyapunov's direct method and backstepping techniques.

In his study, Reyhanoğlu mentioned asymptotical stabilization of autonomous surface vehicles with two independent thrusters, propeller and rudder on a locally controllable equilibrium point [18]. The stabilization of the system has exponential convergence rates with discontinuous feedback.

Similar studies are conducted by Lefeber, *et.al*, and they stated that simple state feedback law renders tracking error dynamics globally k-exponentially stable [19]. Similar to many other studies, they neglected the dynamics associated with heave, roll and pitch since the motion is kept in the horizontal plane. Furthermore, the study was validated with experiments in a disturbance-free environment.

Lefeber's studies are further investigated by Jiang. The method proposed is an application of Lyapunov's direct method, and gives two solutions with inherent cascade structures that are

exponentially stable; however, the transient response becomes less qualifying [20]. In this study, unmeasured thruster dynamics are also taken into consideration.

Similar studies on stable nonlinear controllers are conducted by Bao-li. Different from the others, he offers a method that achieves global asymptotically stable rather than one with a global exponential convergence rate [21]. In this study, again for reduced order vehicle model, a robust control law with k-exponentially stability is postulated.

In their study, Liao, *et.al*, suggest an adaptive dynamic sliding mode control for path following control of underactuated surface vehicle [22]. The method is also globally asymptotically stable. The implementation benefits from backstepping method in sliding mode control of the nonlinear system with model errors and unknown external disturbances.

In their study, Naeem, *et.al*, proposed a linear quadratic Gaussian (LQG) autopilot for unmanned sea surface vehicle. They used fuzzy logic control over LQG to achieve control, navigation and guidance for the Springer USSV. The fuzzy training is accommodated with a system identification algorithm with Kalman filtering. The controller proposed has achieved a certain level of robustness [23].

In another study, Caccia, *et.al*, compared effectiveness for precision and power consumption of extended Kalman filtering and a simple PID guidance technique. The control tasks are chosen as auto-heading, auto-speed and straight line following [24]. The only available sensors are GPS and compass in this work.

Other control techniques are used for control and guidance of unmanned sea surface vehicles as well. Guo, *et.al*, designed a sliding mode fuzzy controller for guidance and control of autonomous underwater vehicle, which can be applied to sea surface vehicles up to a certain extent. They demonstrated the feasibility of sliding mode fuzzy controller in motion control and line of sight guidance of sea vehicles. The experimental results validated the effectiveness of the controller over the model uncertainties, non-linearities in vehicle dynamics, and disturbances such as ocean currents and waves.

In another study, Johansen, *et.al*, suggested a multi-parametric quadratic programming method and computed linear representation of the least-squares optimal control allocation for motions of surface vehicles in horizontal plane with low speed maneuvering and dynamic positioning [26]. They said that the attainable thrust region can be decomposed into

finite union of convex polyhedral sets and convex quadratic programming can be applied with any number of rudders and thrusters. The method proposed is claimed to have low computational complexity and high software reliability.

In addition to control and autopilot design for unmanned sea surface vehicles, the scientists are focused on guidance algorithms for these vehicles as well. Qiaomei, *et.al*, offers line of sight guidance scheme for feasible trajectory tracking of surface vessels [27]. The way-point guidance method offered is based on a neural adaptive control and is validated with experimental results.

Bibuli, *et.al*, suggested a line following guidance control for sea surface vehicles and experimented their method. The proposal is rooted in Lyapunov's control technique and resulted in high precision reference tracking with low rudder activity. The approach differs from the other ones known in the literature in the definition of the error variables to be stabilized to zero [28].

Another interesting study on guidance of sea surface vehicles is conducted by Svec, *et.al*. They presented a look-ahead based trajectory planning algorithm which combines heuristic search and locally bounded optimal planning using variation of the minimax game-tree search [29]. The algorithm is designed to keep the vehicle in its trajectory with necessary actuations.

Another objective in guidance of the sea surface vehicles is to avoid obstacles or collisions. Related to this topic, Krishnamurthy, *et.al*, offered a robust inner-loop controller and a hierarchical combination of wide-area, intermediate-area and local-area planning and obstacle avoidance algorithm [30]. They simulated and emulated their algorithm with hardware-in-the-loop tests and observed that multiple sea surface vehicle system avoids passive obstacles in simulations and exact environment.

Naeem, *et.al*, conducted a study on reactive path planning algorithm to avoid collisions according to the international rules of COLREG. Their strategy consists of way-point guidance by line of sight together with a manual biasing scheme. Their method obeys the rules of COLREG related to amount of maneuver, and a deliberative grid based path planning algorithm is modified accordingly [31]. The autopilot used in this study is PID based.

Another study on collision avoidance based on COLREG rules is conducted by Zhuang, *et.al.* Their algorithm is based on relative coordinate approach. In their study, they simulated the results and demonstrated an effective implementation of the rules on motion of the vehicle [32]. The limitations on maneuvers are also accomplished.

1.3 Organization of Thesis

In this thesis, control and guidance algorithms are developed for an unmanned sea surface vehicle. The thesis starts with an introduction part where the motivation of the study and some selected publications on this subject are presented.

In Chapter 2, a mathematical model for unmanned sea surface vehicles will be derived. The mathematical model both includes the kinematics of the vehicle as well as a detailed explanation of each of the external forces and torques acting on the system. The vehicle will be affected from gravitational and buoyancy forces, centripetal and Coriolis forces, damping, air drag due to its motion and winds, ocean currents, and actuating forces. The actuations will be observed in two forms: thruster and rudder commands. Thruster will mainly provide motion in surge direction, and rudder will yield changes in yaw position.

In Chapter 3, autopilots will be designed based on the mathematical model developed in Chapter 2. The controllers to be employed are PID and LQR based. Before proceeding with controller design, the nonlinear model will be linearized. For the underactuated system, a controllable subspace will be found. The controllers will be developed for the linearized model valid on different linearization points. There will be a real-time interpolation of the model behaviors on different linearization points. Sample time analysis for the controllers will also be conducted. Yaw autopilots will be designed. The autopilots will be compared on their performance on speed hold tests. Closed-loop eigenvalues of the autopilots will be compared. At the end of this chapter, passive stability of the vehicle will also be investigated.

In Chapter 4, a way-point guidance algorithm developed for the motion of the sea surface vehicle will be studied. The guidance algorithm is to provide reference signals to the autopilots designed in Chapter 3. Between the autopilot and the guidance blocks, there will be an online obstacle avoidance system. Whenever an obstacle is encountered, this algorithm will provide reference signals to autopilot taking higher priority with respect to

the guidance block. Due to motions to avoid obstacles or in effect of environmental disturbances, the vehicle may miss a way-point. In such a case, a way-point selection algorithm will provide reference to the guidance algorithm. The proposed methods are simulated. The effects of disturbances on the simulations are investigated, and the limits of the developed strategies are determined.

In Chapter 5, experimental setup and experimental results will be presented. The autopilot designed for the unmanned sea surface vehicle is realized on Traxxas Spartan model boat as the sea surface vehicle and ArduPilot Mega cart as the platform where the algorithm runs. The vehicle is moved autonomously according to the commands provided to the onboard autopilot card via wireless communication. In this chapter, experimental results will be evaluated. The possible improvements in the experimental setup and the effects of environmental conditions will be discussed.

The thesis concludes with Chapter 6, where summary and future works are presented and discussed. In this chapter, all the work that has been done in this study will be summarized. The simulation and experimental results will be stated and evaluated. The possible future works for the control and guidance of sea surface vehicles and other possible academic areas related to the studies on these vehicles will be mentioned.

There are appendices at the end of the thesis. There, the parameters used in the mathematical modeling of the unmanned sea surface vehicle will be derived.

CHAPTER 2

MATHEMATICAL MODELING OF SEA SURFACE VEHICLES

2.1 Introduction

In order to develop successful control algorithms for unmanned sea surface vehicles, we need to have an accurate model of the vehicle under consideration. In this section, mathematical modeling of an unmanned sea surface vehicle will be investigated.

The mathematical model of sea surface vehicles is based on Fossen's vectorial model for marine vehicles [1]. The model consists of inertial components and the forces affecting the vehicle. The main forces of interest are centripetal and Coriolis forces, damping forces, gravity and buoyancy forces, air drag, sea current forces, forces due to thruster motion and forces due to motion of rudder. Other effects on motion of a sea surface vehicle, such as waves are taken as disturbances to the system and they are not modeled. In mathematical modeling, certain assumptions are made and their validities are explained.

In the beginning of this chapter, reference frames on which the forces and moments are defined are introduced. While positions and orientations of the vehicle are designated with respect to the vehicle, velocities are described in reference frame fixed to Earth.

The chapter starts with definitions of and transformations between the coordinate frames, and thoroughly explains each factor in the mathematical model of the vehicle. The section ends with the overall mathematical model simulated in MATLAB SIMULINK.

2.2 Coordinate Frames and Transformations

In this section, coordinate frames to be used in modeling of unmanned sea surface vehicles will be explained. An example of a sea surface vehicle is depicted in Figure 2.1 below.

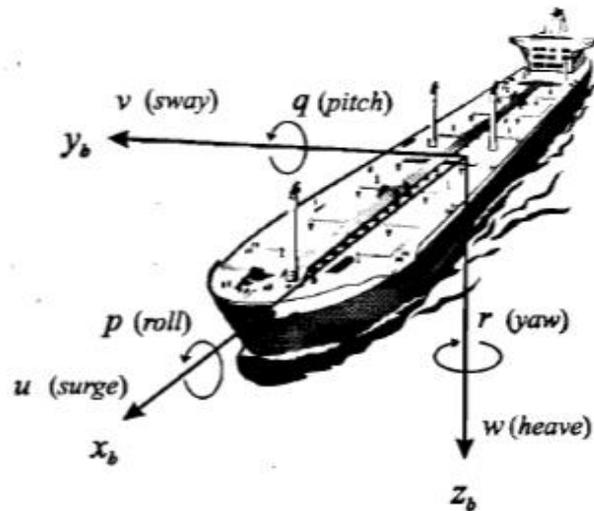


Figure 2.1. Sea surface vehicle [1]

As shown in Figure 2.1, there are 6 independent coordinates, namely the position and orientation of the sea surface vehicle. The forces and moments, and resulting motions with 6 degrees of freedom (DOF) are shown in Table 2.1. The first 3 of the coordinates represents positions, linear velocities and forces in translation motions; whereas, the last 3 lines are the orientations, angular velocities and moments in rotational motions.

Table 2.1. Notation for water vehicle motion

DOF		Forces and Moments	Linear and Angular Velocities	Positions and Orientations
1	motion in x-direction (surge)	X	u	x
2	motion in y-direction (sway)	Y	v	y
3	motion in z-direction (heave)	Z	w	z
4	rotation about x-axis (roll)	K	p	ϕ
5	rotation about y-axis (pitch)	M	q	θ
6	rotation about z-axis (yaw)	N	r	ψ

Surge is the direction from aft to fore on the ship. It is the longitudinal axis. The movement in surge is not only due to propulsion but also the tendency for a boat to move forward on a wave crest, i.e., surfing and backward in a trough. It is clear that for a vehicle under way, a swell moving backwards and forwards means alternatively accelerated and retarded motion with respect to the direction of the waves.

Sway is the direction towards the starboard. It is the transversal axis. On this axis, sideslip due to centripetal forces and effects of drifts due to successive wave pushes are observed. If not under way, sea surface vehicles tend to be broadside on to the swell.

Heave is the direction from top to bottom. It is the normal axis. The motion on this axis may be due to the change in buoyancy as well as each passing of wave. In the latter case, motion in heave will be periodic with the motion in pitch and roll.

Roll is the rotation about the longitudinal x-axis. The rotation on this axis is important for the comfort of those onboard. The axis of rotation for a boat changes with the center of gravity of the board changes in space as the ship rotates.

Pitch is the rotation about the transverse y-axis. Pitching is the difficult motion when heading into sea and it increases when the conditions of synchronism occur.

Yaw is the rotation about the vertical z-axis. Motion in yaw axis can be explained as the tendency to distract from the course. There is no restorative moment in yaw axis, unlike the roll and pitch axes. The restorative moment shall be applied by an external control surface such as rudders [33].

To define position and orientation of the vehicle, motions and rotations of it, and the forces and torques applied on the abovementioned directions, it is needed to state reference frames. In literature, there are mainly four reference frames: Earth-centered inertial reference frame (ECIF), Earth-centered Earth-fixed reference frame (ECEF), North-East-Down reference frame (NED) and body-fixed reference frame.

Earth-centered inertial frame takes the center of Earth as its origin. z-axis is towards the North Pole. The inertial reference frame is a nonaccelerating frame in which Newton's laws of motion apply. The reference frame is used mostly for terrestrial navigation.

Earth-centered Earth-fixed reference frame takes the center of Earth as its origin, and z-axis is towards the North Pole. However the other two axes rotates with Earth. Hence in speed of rotation of Earth 7.2921×10^{-5} rad/s, this frame rotates with respect to Earth-fixed inertial reference frame. The reference frame is used for vessels with low speed relative to Earth rotation, and guidance, navigation and control of trans-continental motions of ships.

North-East-Down (NED) reference frame is defined on surface of Earth. The x-axis is towards North Pole, the y-axis is towards East and z-axis points downwards normal to surface of Earth. In our everyday life, we use this reference frame. For local ship motions for which the tangent plane on the surface of Earth can be assumed to be same, this reference frame is generally utilized. In this study, as Earth-fixed reference frame, NED will be used. It is also assumed that in NED, Newton's laws are still applicable.

Body-fixed reference frame is fixed to the vehicle. In this reference frame, origin is usually chosen as the center of gravity (CG) of the vehicle, and x-axis is directed from aft to fore of the ship, y-axis is directed to starboard and z-axis is directed from top to bottom. The reference frames are shown in Figure 2.2, below.

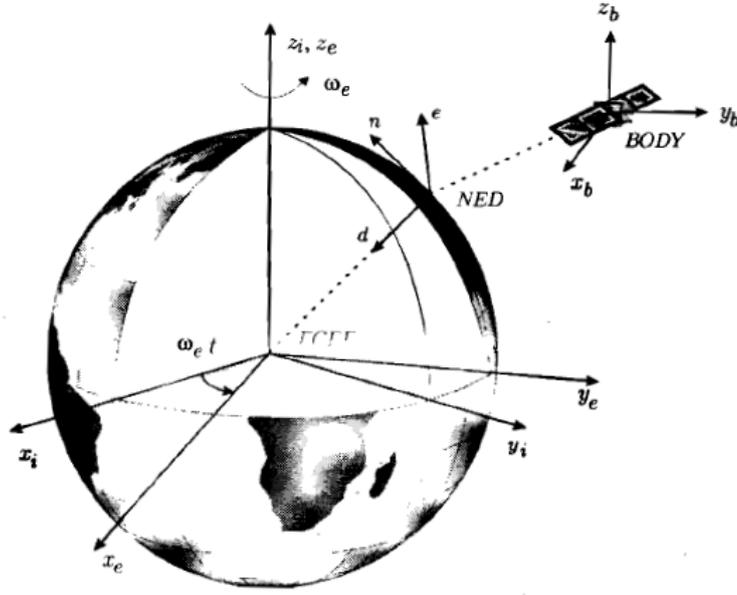


Figure 2.2. Reference frames [1]

While position and orientation of the vehicle are defined in an inertial frame, for our case the NED frame, the linear and angular velocities, and forces and torques, can be measured in body-fixed reference frame. Hence, we require a reference frame transformation between these reference frames used in this study.

From now on, as Earth-fixed reference frame, NED frame, will be abbreviated as e-frame, and body fixed reference frame will be abbreviated as b-frame. In order to obtain the transformations, we need some background information. Firstly, note that the coordinates are independent from each other and the transformation matrix is a 3x3 orthogonal matrix.

Definition: Rotation Matrix

A rotation matrix is an element of set of orthogonal matrices of order n, i.e., satisfies the following properties:

$$\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I} \tag{2.1}$$

$$\det(\mathbf{R}) = 1 \tag{2.2}$$

Definition: Skew-Symmetric Matrix

A matrix is said to be skew-symmetric matrix of order n if:

$$\mathbf{S} = -\mathbf{S}^T \quad (2.3)$$

For a skew-symmetric matrix, the diagonal elements are zero, and the off-diagonal elements satisfy:

$$s_{ij} = -s_{ji} \quad \text{for } i \neq j \quad (2.4)$$

Definition: Cross Product Operator

The vector cross product \boldsymbol{x} is defined as:

$$\boldsymbol{x} \times \boldsymbol{a} := \mathbf{S}(\boldsymbol{\lambda})\boldsymbol{a} \quad (2.5)$$

\mathbf{S} is an element of skew-symmetric matrix set of order 3. \mathbf{S} is defined as:

$$\mathbf{S}(\boldsymbol{\lambda}) = -\mathbf{S}^T(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad \text{where } \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (2.6)$$

Lemma: Euler's Rotation Theorem

Every change in the relative orientation of two rigid bodies or reference frames A and B can be produced by simple rotation of B in A according to Euler's theorem (1976).

Let \boldsymbol{v}^b be a vector in body-fixed reference frame and \boldsymbol{v}^e be a vector in earth-fixed reference frame. Let \boldsymbol{v}^b be expressed as $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T$ and $|\boldsymbol{\lambda}| = 1$. The rotation about β axis can be expressed as a transformation matrix as follows [34], [35]:

$$\boldsymbol{v}^e = \mathbf{R}_b^e \boldsymbol{v}^b \quad \text{where } \mathbf{R}_b^e = \mathbf{R}_{\boldsymbol{\lambda}, \beta} \quad (2.7)$$

$$\mathbf{R}_{\boldsymbol{\lambda}, \beta} = \mathbf{I}_{3 \times 3} + \sin(\beta) \mathbf{S}(\boldsymbol{\lambda}) + (1 - \cos(\beta)) \mathbf{S}^2(\boldsymbol{\lambda}) \quad (2.8)$$

In (2.8), $\mathbf{I}_{3 \times 3}$ is the identity matrix and according to definition of skew-symmetric matrices $\mathbf{S}^2(\boldsymbol{\lambda}) = \boldsymbol{\lambda}\boldsymbol{\lambda}^T - \mathbf{I}_{3 \times 3}$. Hence (2.8) becomes:

$$\begin{aligned}
R_{11} &= (1 - \cos(\beta))\lambda_1^2 + \cos(\beta) \\
R_{12} &= (1 - \cos(\beta))\lambda_1\lambda_2 - \lambda_3 \sin(\beta) \\
R_{13} &= (1 - \cos(\beta))\lambda_1\lambda_3 + \lambda_2 \sin(\beta) \\
R_{21} &= (1 - \cos(\beta))\lambda_2\lambda_1 + \lambda_3 \sin(\beta) \\
R_{22} &= (1 - \cos(\beta))\lambda_2^2 + \cos(\beta) \\
R_{23} &= (1 - \cos(\beta))\lambda_2\lambda_3 - \lambda_1 \sin(\beta) \\
R_{31} &= (1 - \cos(\beta))\lambda_3 - \lambda_2 \sin(\beta) \\
R_{32} &= (1 - \cos(\beta))\lambda_3\lambda_2 + \lambda_1 \sin(\beta) \\
R_{33} &= (1 - \cos(\beta))\lambda_3^2 + \cos(\beta)
\end{aligned} \tag{2.9}$$

Now, in order to complete transformation between body and earth-fixed reference frames, rotation about three angles are necessary.

Rotation about the yaw angle, ψ with z -axis fixed:

$$\mathbf{R}_{z,\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

Rotation about the pitch angle, θ with y -axis fixed:

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.11}$$

Rotation about the roll angle, ϕ with x -axis fixed:

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.12}$$

Linear Velocity Transformation:

The linear velocities of the water vehicle are attained in body-fixed reference frame. The motion in body-fixed reference frame should be transformed to earth-fixed reference frame for complete mathematical modeling. The body-to-earth transformation can be achieved as follows:

$$\mathbf{R}_b^e(\boldsymbol{\theta}) = \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi} \quad (2.13)$$

The earth-to-body transformation is as follows:

$$\mathbf{R}_e^b(\boldsymbol{\theta}) = \mathbf{R}_b^e(\boldsymbol{\theta})^{-1} = \mathbf{R}_{x,\phi}^T \mathbf{R}_{y,\theta}^T \mathbf{R}_{z,\psi}^T \quad (2.14)$$

(2.14) can be expressed explicitly as:

$$\mathbf{R}_b^e(\boldsymbol{\theta}) = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) \cos(\phi) + \cos(\psi) \sin(\theta) \sin(\phi) \\ \sin(\psi) \cos(\theta) & \cos(\psi) \cos(\phi) + \sin(\phi) \sin(\theta) \sin(\psi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) \\ & \sin(\psi) \sin(\phi) + \cos(\psi) \cos(\phi) \cos(\theta) \\ & -\cos(\psi) \sin(\phi) + \sin(\theta) \sin(\psi) \cos(\phi) \\ & \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.15)$$

The order in (2.13) can be changed. It is important to preserve the order throughout the modeling for consistency.

Note that (2.15) is definite unless the angles θ and ϕ are $\pi/2$. If the rotations around x -axis or y -axis reach $\pi/2$, the boat will sink down. The model is invalid while the vehicle is sinking. The controller and user commands should guarantee the vehicle to be out of neighborhood of these thresholds.

Angular Velocity Transformation:

The transformation between body-fixed and earth-fixed referenced angular velocities is as follows [4]:

$$\mathbf{w}_e = \mathbf{T}_\theta(\boldsymbol{\theta}) \mathbf{w}_b \quad (2.16)$$

$$\mathbf{T}_\theta^{-1}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta) \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (2.17)$$

The earth-to-body angular velocity transformation is:

$$T_{\theta}^{-1}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta) \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (2.18)$$

6 DOF Kinematic Equations:

The 6 DOF kinematic equations is the relation between body-fixed linear and angular velocities and earth-fixed position.

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{v} \quad (2.19)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} = J(\boldsymbol{\eta}) \begin{bmatrix} \boldsymbol{v}_b \\ \boldsymbol{w}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_b^e(\boldsymbol{\theta}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & T_{\theta}(\boldsymbol{\theta}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_b \\ \boldsymbol{w}_b \end{bmatrix} \quad (2.20)$$

SIMULINK Model:

The 6 DOF kinematic equations are modeled in SIMULINK. The model of the body-to-earth embedded MATLAB function can be seen as a block in Figure 2.3.

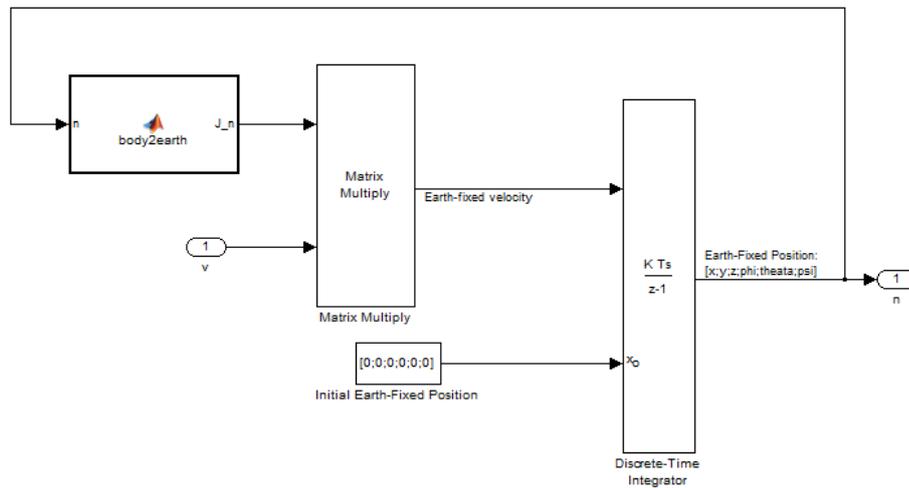


Figure 2.3. SIMULINK model for 6 DOF kinematic equations

2.3 Rigid Body Dynamics

The mathematical model of motion of a sea surface vehicle obeys Newton's laws of motion, the sum of forces and torques is equal to inertia times the acceleration. To define motion of the sea surface vehicles, a vectorial model is developed by Fossen [1]:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} \quad (2.21)$$

The right hand side of (2.21) is the rigid body dynamics and added mass dynamics terms. In this section, rigid body dynamics for a water vehicle will be mathematically modeled.

Translational Motion:

The mass of the rigid body is defined as the integral over the volume of the vehicle:

$$m = \int_V \rho^b dV \quad (2.22)$$

In (2.22), ρ^b stands for the mass density of the body. The first assumption in modeling is the constant body mass assumption, i.e., it is assumed that the mass of the vehicle does not change with time.

For an inertial reference, the center of gravity is represented with a vector \mathbf{r}_c . Any point on the vehicle, \mathbf{r}' is the vector sum of \mathbf{r}_c and \mathbf{r} , where \mathbf{r} is the vector from that point to \mathbf{r}_c . Since \mathbf{r}_c is constant over the entire volume of the vehicle, as in its definition, we attain:

$$\int_V \mathbf{r}\rho^b dV = \int_V \mathbf{r}'\rho^b dV - \int_V \mathbf{r}_c\rho^b dV = m\mathbf{r}_c - \mathbf{r}_c \int_V \rho^b dV = 0 \quad (2.23)$$

Now, the motion of the vessel is measured in body-fixed reference frame. Taking earth-fixed reference frame as the inertial frame, this motion can be expressed as:

$$\mathbf{v}_c^b = \mathbf{v}_o^b + \boldsymbol{\omega}_e^b \times \mathbf{r}_c^b \quad (2.24)$$

The above expression means that, the motion of the center of gravity of the vessel in b-frame is the sum of motion of origin in b-frame and angular velocity of vessel in e-frame expressed in b-frame cross-product with the vector between the center of gravity and origin

in b-frame. Now, we know the translational motion in b-frame, we can transform it to e-frame.

$$\mathbf{v}_c^e = \mathbf{R}_b^e \mathbf{v}_c^b = \mathbf{R}_b^e (\mathbf{v}_o^b + \mathbf{w}_e^b x \mathbf{r}_c^b) \quad (2.25)$$

If we take time differentiation of (2.25), we attain the acceleration:

$$\begin{aligned} \dot{\mathbf{v}}_c^e &= \mathbf{R}_b^e \left(\dot{\mathbf{v}}_o^b + \dot{\mathbf{w}}_e^b x \mathbf{r}_c^b + \mathbf{w}_e^b x \dot{\mathbf{r}}_c^b \right) + \dot{\mathbf{R}}_b^e (\mathbf{v}_o^b + \mathbf{w}_e^b x \mathbf{r}_c^b) \\ &= \mathbf{R}_b^e \left(\dot{\mathbf{v}}_o^b + \mathcal{S}(\dot{\mathbf{w}}_e^b) \mathbf{r}_c^b + \mathcal{S}(\mathbf{w}_e^b) \mathbf{v}_c^e + \mathcal{S}^2(\mathbf{w}_e^b) \mathbf{r}_c^b \right) \end{aligned} \quad (2.26)$$

Note that $\dot{\mathbf{R}}_b^e = \mathbf{R}_b^e \mathcal{S}(\mathbf{w}_e^b)$ and $\dot{\mathbf{r}}_c^b = \mathbf{0}$ since the vector between the center of gravity and origin is constant.

Lemma: Euler's First and Second Axioms

Newton's second law can be written as conservation of both linear momentum and angular momentum of center of gravity:

$$m \dot{\mathbf{v}}_c = \dot{\mathbf{p}}_c = \mathbf{f}_c \quad (2.27)$$

$$I \dot{\mathbf{w}}_b^e = \dot{\mathbf{h}}_c = \mathbf{m}_c \quad (2.28)$$

Using Euler's first axiom:

$$m \dot{\mathbf{v}}_c^e = \dot{\mathbf{f}}_c^e = \mathbf{R}_b^e \dot{\mathbf{f}}_c^b \quad (2.29)$$

Similarly for origin point we attain:

$$m \left(\dot{\mathbf{v}}_o^b + \mathcal{S}(\dot{\mathbf{w}}_e^b) \mathbf{r}_c^b + \mathcal{S}(\mathbf{w}_e^b) \mathbf{v}_c^e + \mathcal{S}^2(\mathbf{w}_e^b) \mathbf{r}_c^b \right) = \dot{\mathbf{f}}_c^b \quad (2.30)$$

If we take the center of gravity as the origin of the vessel, i.e., if \mathbf{r}_c set to $\mathbf{0}$, we have:

$$m \left(\dot{\mathbf{v}}_o^b + \mathcal{S}(\mathbf{w}_e^b) \mathbf{v}_c^e \right) = \dot{\mathbf{f}}_c^b \quad (2.31)$$

Here note that the translational motion is independent of the attack point, $\dot{\mathbf{f}}_c^b = \dot{\mathbf{f}}_o^b$.

Rotational Motion:

Definition: Angular Momenta

Angular momenta about the origin are defined as:

$$\mathbf{h}_o = \int_V (\mathbf{r}' \times \mathbf{v}) \rho^b dV \quad (2.32)$$

In (2.32), \mathbf{v} stands for the velocity of the volume element dV :

$$\mathbf{v} = \mathbf{v}_o^b + \mathbf{w}_e^b \times \mathbf{r}' \quad (2.33)$$

(2.32) becomes:

$$\begin{aligned} \mathbf{h}_o &= \int_V (\mathbf{r}' \times \mathbf{v}_o^b) \rho^b dV + \int_V (\mathbf{r}' \times \mathbf{w}_e^b \times \mathbf{r}') \rho^b dV \\ &= \int_V (\mathbf{r} + \mathbf{r}_c^b) \rho^b dV \times \mathbf{v}_o^b + \int_V (\mathbf{r}' \times \mathbf{w}_e^b \times \mathbf{r}') \rho^b dV \\ &= m \mathbf{r}_c^b \times \mathbf{v}_o^b + \int_V (\mathbf{r}' \times \mathbf{w}_e^b \times \mathbf{r}') \rho^b dV \end{aligned} \quad (2.34)$$

Note that in (2.34), we used the facts \mathbf{r}_c^b is constant over the volume and $\int_V \mathbf{r} \rho^b dV = 0$

Now the second term in (2.34) can be defined as inertia dyadic:

$$\begin{aligned} \int_V (\mathbf{r}' \times \mathbf{w}_e^b \times \mathbf{r}') \rho^b dV &= \int_V -\mathbf{r}' \times (\mathbf{r}' \times \mathbf{w}_e^b) \rho^b dV = \\ &= \int_V -\mathcal{S}(\mathbf{r}') \mathcal{S}(\mathbf{r}') \mathbf{w}_e^b \rho^b dV = \mathbf{I}_o \mathbf{w}_e^b \end{aligned} \quad (2.35)$$

Hence, (2.34) becomes:

$$\mathbf{h}_o = m \mathbf{r}_c^b \times \mathbf{v}_o^b + \mathbf{I}_o \mathbf{w}_e^b \quad (2.36)$$

The angular momentum of the center of gravity (CG) can be directly found from that of the origin. The vector between these points is shown as \mathbf{r}_c^b .

$$\mathbf{h}_c = \mathbf{h}_o - m \mathbf{r}_c^b \times \mathbf{v}_c^b = \mathbf{I}_o \mathbf{w}_e^b + m \mathbf{r}_c^b \times \mathbf{v}_o^b - m \mathbf{r}_c^b \times \mathbf{v}_c^b \quad (2.37)$$

Up to now we have dealt with angular momentums in body-fixed reference frame. According to Euler's second axiom, to find the moment about CG, we shall find the time derivative of angular momentum about CG. To find out the moment, let us first look at moment in earth-fixed reference frame:

$$\begin{aligned} \mathbf{m}_c^e &= \dot{\mathbf{h}}_c^e = (\mathbf{R}_b^e \dot{\mathbf{h}}_c^b) = \mathbf{R}_b^e \dot{\mathbf{h}}_c^b + \dot{\mathbf{R}}_b^e \mathbf{h}_c^b = \mathbf{R}_b^e (\dot{\mathbf{h}}_c^b + \mathbf{S}(\mathbf{w}_e^b) \mathbf{h}_c^b) \\ &= \mathbf{R}_b^e (\mathbf{I}_o \dot{\mathbf{w}}_e^b + m \mathbf{r}_c^b \times (\dot{\mathbf{v}}_o^b - \dot{\mathbf{v}}_c^b) + \mathbf{S}(\mathbf{w}_e^b) (\mathbf{I}_o \mathbf{w}_e^b + m \mathbf{r}_c^b \times \mathbf{v}_o^b - \\ &\quad m \mathbf{r}_c^b \times \mathbf{v}_c^b)) \end{aligned} \quad (2.38)$$

$$\mathbf{m}_c^e = \mathbf{R}_b^e (m \dot{\mathbf{r}}_c^b - \mathbf{r}_c^b \times \mathbf{f}_c^b) = \mathbf{R}_b^e (m \dot{\mathbf{r}}_c^b - \mathbf{r}_c^b \times m (\dot{\mathbf{v}}_c^b + \mathbf{w}_e^b \times \mathbf{v}_c^b)) \quad (2.39)$$

If we equate (2.38) to (2.39), we can attain \mathbf{m}_o^b as:

$$\mathbf{m}_o^b = \mathbf{I}_o \dot{\mathbf{w}}_e^b + \mathbf{S}(\mathbf{w}_e^b) \mathbf{I}_o \mathbf{w}_e^b + m \mathbf{S}(\mathbf{r}_c^b) \dot{\mathbf{v}}_o^b + m \mathbf{S}(\mathbf{r}_c^b) \mathbf{S}(\mathbf{w}_e^b) \mathbf{v}_o^b \quad (2.40)$$

If we take $\mathbf{r}_c^b = \mathbf{0}$, we have:

$$\mathbf{m}_c^b = \mathbf{I}_c \dot{\mathbf{w}}_e^b + \mathbf{S}(\mathbf{w}_e^b) \mathbf{I}_c \mathbf{w}_e^b \quad (2.41)$$

Definition: Inertia Matrix

The inertia matrix about o is a symmetric positive definite matrix such that:

$$\mathbf{I}_o = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (2.42)$$

In (2.42),

$$\begin{aligned} I_x &= \int_V (y^2 + z^2) \rho^b dV \\ I_y &= \int_V (x^2 + z^2) \rho^b dV \\ I_z &= \int_V (y^2 + x^2) \rho^b dV \end{aligned} \quad (2.43)$$

$$I_{xy} = I_{yx} = \int_V xy \rho^b dV$$

$$I_{zy} = I_{yz} = \int_V zy \rho^b dV$$

$$I_{xz} = I_{zx} = \int_V xz \rho^b dV$$

According to parallel axes theorem [1], we can find \mathbf{I}_c from \mathbf{I}_o as:

$$\mathbf{I}_c = \mathbf{I}_o + m\mathbf{S}^2(\mathbf{r}_c^b) \quad (2.44)$$

6 DOF Rigid Body Equations of Motion:

If we combine (2.31) and (2.40), we attain the equations of motion as:

$$\begin{aligned} m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] &= X \\ m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] &= Y \\ m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] &= Z \\ I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ &\quad + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K \\ I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\ &\quad + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \\ I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\ &\quad + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wp)] = N \end{aligned} \quad (2.45)$$

(2.45) can be written in vector form as follows:

$$\mathbf{M}_{RB} \dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} \quad (2.46)$$

According to SNAME notation [2], in (2.46):

$$\begin{aligned} \mathbf{v} &= [u, v, w, p, r, q]^T \\ \boldsymbol{\tau} &= [X, Y, Z, K, M, N]^T \end{aligned} \quad (2.47)$$

Definition: Rigid Body System Inertia Matrix

In (2.46), \mathbf{M}_{RB} is defined as rigid body inertia matrix. The matrix is unique for the system and it is symmetric, positive definite and has derivative of $\mathbf{0}$.

$$\begin{aligned}\mathbf{M}_{RB} &= \mathbf{M}_{RB}^T > 0 \\ \dot{\mathbf{M}}_{RB} &= \mathbf{0}\end{aligned}\tag{2.48}$$

From (2.45), we can have \mathbf{M}_{RB} as:

$$\begin{aligned}\mathbf{M}_{RB} &= \begin{bmatrix} m\mathbf{I}_{3 \times 3} & -m\mathcal{S}(\mathbf{r}_c^b) \\ m\mathcal{S}(\mathbf{r}_c^b) & \mathbf{I}_o \end{bmatrix} = \\ &= \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix}\end{aligned}\tag{2.49}$$

Definition: Coriolis-Centripetal Matrix

In (2.46), $\mathbf{C}_{RB}(\mathbf{v})$ is a symmetric matrix and defined as:

$$\begin{aligned}\mathbf{M}_{RB} &= \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{22} & \mathbf{M}_{22} \end{bmatrix} \\ \mathbf{C}_{RB}(\mathbf{v}) &= \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathcal{S}(\mathbf{M}_{11}\mathbf{v}_1 + \mathbf{M}_{12}\mathbf{v}_2) \\ -\mathcal{S}(\mathbf{M}_{11}\mathbf{v}_1 + \mathbf{M}_{12}\mathbf{v}_2) & -\mathcal{S}(\mathbf{M}_{21}\mathbf{v}_1 + \mathbf{M}_{22}\mathbf{v}_2) \end{bmatrix}\end{aligned}\tag{2.50}$$

(2.50) can be written as:

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) \\ m(x_g q - w) & -m(z_g r + x_g p) & m(z_g q + u) \\ m(x_g q + v) & m(y_g r - u) & -m(x_g p + y_g q) \end{bmatrix}\tag{2.51}$$

$$\left[\begin{array}{ccc} m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g q + v) \\ -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ 0 & -I_{yz}q - I_{xz}p + I_z r & I_{yz}r + I_{xy}p - I_y q \\ I_{yz}q + I_{xz}p - I_z r & 0 & -I_{xz}r - I_{xy}q + I_x p \\ -I_{yz}r - I_{xy}p + I_y q & I_{xz}r + I_{xy}q - I_x p & I_z \end{array} \right]$$

With some further assumptions (2.45) and its representation in matrix form (2.49) and (2.51) can be simplified.

Take the origin coinciding with the center of gravity:

$$\mathbf{r}_c^b = [0 \ 0 \ 0]^T \quad (2.52)$$

Rotate the body axes and translate the origin such that \mathbf{I}_o becomes diagonal:

$$\mathbf{I}_o = \text{diag}\{I_x, I_y, I_z\} \quad (2.53)$$

With these assumptions (2.45) becomes:

$$\begin{aligned} m[\dot{u} - vr + wq] &= X \\ m[\dot{v} - wp + ur] &= Y \\ m[\dot{w} - uq + vp] &= Z \\ I_x \dot{p} + (I_z - I_y)qr &= K \\ I_y \dot{q} + (I_x - I_z)rp &= M \\ I_z \dot{r} + (I_y - I_x)pq &= N \end{aligned} \quad (2.54)$$

(2.49) becomes:

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_o \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \quad (2.55)$$

(2.51) becomes:

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mv & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_z r & -I_y q \\ -mv & 0 & mu & -I_z r & 0 & +I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \quad (2.56)$$

2.4 Added Mass Dynamics

In previous chapter, we have investigated the mathematical modeling of rigid body dynamics. Before proceeding with modeling the external forces, we need to discuss the dynamics of the surrounding fluid, i.e., added mass dynamics as well. Similar to rigid body dynamics, added mass dynamics also consists of inertia, and centripetal and Coriolis matrices. The added mass dynamics will be considered together with rigid body on the left hand side of (2.21). In this section, added mass will be modeled.

Definition: Euler-Lagrange Equation

Let T and P denote the vessels kinetic and potential energies, respectively. Then the Lagrangian, L is:

$$L = T - P \quad (2.57)$$

The Euler-Lagrange equation is:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\eta}} \right) - \left(\frac{\partial L}{\partial \eta} \right) = J^{-T}(\eta) \boldsymbol{\tau} \quad (2.58)$$

Euler-Lagrange equation is valid in any reference frame. If selected the position and orientation vector in Earth-fixed and the translational and angular velocities in body-fixed reference frames, we still have the integral of velocities with no physical interpretation in terms of position and orientation. Hence Lagrangian method cannot be used directly to formulate the equations of motion.

Definition: Kirchhoff's Equations of Motion (Quasi-Lagrangian Approach)

Let body-fixed linear velocity be $\mathbf{v}_1 = [u \ v \ w]^T$ angular velocity be $\mathbf{v}_2 = [p \ q \ r]^T$ the force be $\boldsymbol{\tau}_1 = [X \ Y \ Z]^T$, and the moment be $\boldsymbol{\tau}_2 = [K \ M \ N]^T$. The kinetic energy is:

$$\mathbf{T} = \frac{1}{2} \mathbf{v}^T \mathbf{M} \mathbf{v} \quad (2.59)$$

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \mathbf{v}_1} \right) + \mathbf{S}(\mathbf{v}_2) \left(\frac{\partial \mathbf{T}}{\partial \mathbf{v}_1} \right) = \boldsymbol{\tau}_1 \quad (2.60)$$

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \mathbf{v}_2} \right) + \mathbf{S}(\mathbf{v}_2) \left(\frac{\partial \mathbf{T}}{\partial \mathbf{v}_2} \right) + \mathbf{S}(\mathbf{v}_1) \left(\frac{\partial \mathbf{T}}{\partial \mathbf{v}_1} \right) = \boldsymbol{\tau}_2 \quad (2.61)$$

In (2.60) and (2.61), \mathbf{S} is skew-symmetric cross-product operator.

As done in rigid-body modeling, added mass can be modeled in terms of the inertia matrix \mathbf{M}_A and the Coriolis and centripetal terms $\mathbf{C}_A(\mathbf{v})$.

Definition: Added Mass Inertia Matrix

$$\mathbf{M}_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (2.62)$$

where $X_{\dot{u}}$ is the hydrodynamic derivative defined as:

$$X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}} \quad (2.63)$$

Definition: Added Mass Coriolis-Centripetal Matrix

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (2.64)$$

In (2.64):

$$a_1 = X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \quad (2.65)$$

$$a_2 = Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r$$

$$a_3 = Z_{\dot{u}}u + Z_{\dot{v}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r$$

$$b_1 = K_{\dot{u}}u + K_{\dot{v}}v + K_{\dot{w}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r$$

$$b_2 = M_{\dot{u}}u + M_{\dot{v}}v + M_{\dot{w}}w + M_{\dot{p}}p + M_{\dot{q}}q + M_{\dot{r}}r$$

$$b_3 = N_{\dot{u}}u + N_{\dot{v}}v + N_{\dot{w}}w + N_{\dot{p}}p + N_{\dot{q}}q + N_{\dot{r}}r$$

For sea surface vehicles, surge mode is decoupled from the steering dynamics due to xz -plane symmetry. Furthermore, the heave, pitch and roll modes are neglected under the assumption that these motion variables are small compared to surge speed. For forward speed, u , being greater than zero, $u \gg 0$:

$$\mathbf{M}_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & Y_{\dot{r}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{\dot{v}} & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (2.66)$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -Y_{\dot{v}}v - \frac{Y_{\dot{r}} + N_{\dot{r}}}{2}r \\ 0 & 0 & 0 & 0 & 0 & X_{\dot{u}}u \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ +Y_{\dot{v}}v + \frac{Y_{\dot{r}} + N_{\dot{r}}}{2}r & X_{\dot{u}}u & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.67)$$

When $u = 0$, then we can replace $N_{\dot{v}}$ with $Y_{\dot{r}}$ in (2.66) and (2.67), and \mathbf{M}_A becomes symmetric.

The derivations of added mass parameters are presented in Appendix A.

2.5 Damping Factors

From this subsection on, we will deal with the right hand side of (2.21). The equation can be expanded as follows:

$$M\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_g + \boldsymbol{\tau}_t + \boldsymbol{\tau}_r + \boldsymbol{\tau}_a + \boldsymbol{\tau}_c \quad (2.68)$$

In (2.68), left hand side stands for the dynamics of rigid body and the added mass, while the right hand side is the sum of external forces. In the equation above, $\boldsymbol{\tau}_d$ is the forces and torques due to damping of water surrounding, $\boldsymbol{\tau}_g$ is gravitational and buoyancy forces and torques, $\boldsymbol{\tau}_t$ is forces and torques provided by the thruster, $\boldsymbol{\tau}_r$ represents the forces and torques due to rudder, $\boldsymbol{\tau}_a$ stands for the forces and torques due to air drag and wind, and $\boldsymbol{\tau}_c$ is for forces and torques of water (ocean) currents. In the upcoming subsections the mathematical models of each of these components will be investigated.

In this part, damping factors on motion of sea surface vehicles will be investigated. This term is represented with $\boldsymbol{\tau}_d$ in (2.68).

Definition: Potential Damping

The radiation-induced damping is called potential damping. This term is neglected because it is small compared to other dissipative terms such as viscous damping.

Definition: Friction

For low frequency motion of sea surface vehicles linear skin friction due to laminar boundary theory is essential [36]. For high frequency motion in addition to linear skin friction, a nonlinear or quadratic skin friction due to turbulent boundary layer will exist.

Definition: Wave Drift Damping

The wave drift forces are proportional to square of the significant wave height. That is why wave drift damping is of crucial importance in surge motion for higher sea states. This type of damping can be modeled as added resistance for sea surface vessels moving in waves.

Definition: Damping due to Vortex Shedding

In viscous fluid, frictional forces are present and the system is not conservative with respect to energy. The viscous damping due to vortex shedding can be modeled as:

$$\boldsymbol{f}(\boldsymbol{U}) = \frac{1}{2}\rho^w C^w(R_n)A^w|\boldsymbol{U}|\boldsymbol{U} \quad (2.69)$$

In (2.69), \mathbf{U} is the speed of vessel, A is the cross-sectional area under water $C^w(R_n)$ is the drag coefficient and ρ^w is the density of water.

As explained above, there are many damping factors both linear and quadratic. It is convenient to write damping as:

$$\mathbf{D}(\mathbf{v}) = (\mathbf{D}^l + \mathbf{D}^n(\mathbf{v}))\mathbf{v} \quad (2.70)$$

In (2.70), \mathbf{D}^l is the linear damping matrix, and $\mathbf{D}^n(\mathbf{v})$ is nonlinear damping matrix.

Theorem: Hydrodynamic Damping Matrix

For a rigid body moving through water, $\mathbf{D}(\mathbf{v})$ should be real, non-symmetric and strictly positive.

$$\mathbf{D}(\mathbf{v}) > 0 \quad \forall \mathbf{v} \in \mathbf{R}^6 \quad (2.71)$$

Linear damping matrix for sea surface vehicles can be modeled considering the xz -symmetry which decouples surge mode from the steering modes (sway and yaw). The linearized damping forces and moments (neglecting heave, roll and pitch) are as follows:

$$\mathbf{D}^l = - \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & Y_r \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_v & 0 & 0 & 0 & N_r \end{bmatrix} \quad (2.72)$$

Similar to linear damping matrix, noting that $|r|r$ and $|r|v$ components are negligibly small, nonlinear damping matrix can be modeled as:

$$\mathbf{D}^n(\mathbf{v}) = - \begin{bmatrix} X_{|u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{|v|v}|v| & 0 & 0 & 0 & Y_{|v|r}|v| \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{|v|v}|v| & 0 & 0 & 0 & N_{|v|r}|v| \end{bmatrix} \quad (2.73)$$

The axial quadratic terms can be calculated as:

$$X_{|u|u} = \frac{\partial X}{\partial (u|u)} = \frac{1}{2} \rho^w C_D (R_n) A \quad (2.74)$$

It is of high importance to model both nonlinear and linear damping factors together. Because nonlinear damping matrix has an oscillatory behavior and linear damping matrix bounds these oscillations exponentially.

The total damping forces and torques can be written as:

$$\boldsymbol{\tau}_d = (\mathbf{D}^l + \mathbf{D}^n(\mathbf{v}))\mathbf{v} \quad (2.75)$$

Linear and quadratic damping parameters are derived as shown in Appendix B.

2.6 Restoring Forces and Moments

The second term in right hand side of (2.68) is $\boldsymbol{\tau}_g$ representing the gravitational and buoyancy forces. These forces are named as restoring forces as well. In this subsection, these restoring forces will be investigated.

The static stability is referred as metacentric stability in hydrostatic literature. A metacentric stable vessel will resist changes from the equilibrium points in heave, roll and pitch.

For sea surface vehicles, the hydrodynamic restoring forces, gravity and buoyancy forces depend on the vessels metacentric height, location of center of gravity (CG), location of center of buoyancy (CB), shape and size of the water plane. In equations below A_{wp} denotes the cross sectional area in the water plane, \mathbf{GM}_T is the transverse metacentric height in m, i.e., distance between transverse metacenter and CG, and \mathbf{GM}_L is the longitudinal metacentric height in m, i.e., distance between longitudinal metacenter and CG.

Definition: Metacenter

Metacenter is the theoretical point when a vertical line through CB intersects another vertical line through new CB when the vessel is displaced, or tilted in the water.

Gravity and Buoyancy Forces

For a floating vessel at rest, gravitational and buoyancy forces are in balance:

$$m\mathbf{g} = \rho^w \mathbf{g}\mathbf{V} \quad (2.76)$$

Now, let z denote the displacement in heave and assume that $z = 0$ denote the position corresponding to nominal water displacement \mathbf{V} . Hence we have the hydrostatic force:

$$\mathbf{Z} = m\mathbf{g} - \rho^w \mathbf{g}(\mathbf{V} + \delta\mathbf{V}(z)) = -\rho^w \mathbf{g}\delta\mathbf{V}(z) \quad (2.77)$$

In (2.77), $\delta\mathbf{V}(z)$ denotes the displaced water due to variations in heave position z . It can be obtained as:

$$\delta\mathbf{V}(z) = \int_0^z A_{wp}(\xi)d\xi \quad (2.78)$$

Now assume that $A_{wp}(z) = A_{wp}(0)$ for all z . Then (2.77) becomes:

$$\mathbf{Z} \approx -\rho^w \mathbf{g}A_{wp}(0)z = -\mathbf{z}_0z \quad (2.79)$$

The restoring force for any force downwards decomposed in b-frame is:

$$\begin{aligned} \delta\mathbf{f}_r^b &= \mathbf{R}_b^n(\boldsymbol{\theta})^{-1}\delta\mathbf{f}_r^n \\ &= \mathbf{R}_b^n(\boldsymbol{\theta})^{-1} \begin{bmatrix} 0 \\ 0 \\ -\mathbf{z}_0z \end{bmatrix} = -\mathbf{z}_0z \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{bmatrix} \end{aligned} \quad (2.80)$$

Now note that moment arms in roll and pitch can be related to moment arms $\mathbf{GM}_T \sin(\phi)$ and $\mathbf{GM}_L \sin(\theta)$ in roll and pitch, and a force pair $\mathbf{W} = \mathbf{B} = \rho^w \mathbf{g}\mathbf{V}$ in z -direction.

$$\mathbf{r}_r^b = \begin{bmatrix} -\mathbf{GM}_L \sin(\theta) \\ \mathbf{GM}_T \sin(\phi) \\ 0 \end{bmatrix} \quad (2.81)$$

$$\mathbf{f}_r^b = \mathbf{R}_b^n(\boldsymbol{\theta})^{-1} \begin{bmatrix} 0 \\ 0 \\ -\rho^w \mathbf{g}\mathbf{V} \end{bmatrix} = -\rho^w \mathbf{g}\mathbf{V} \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.82)$$

Neglecting $\delta\mathbf{f}_r^b$, the restoring moment becomes:

$$\mathbf{m}_r^b = \mathbf{r}_r^b \chi \mathbf{f}_r^b = -\rho^w \mathbf{g} \nabla \begin{bmatrix} \mathbf{GM}_T \sin(\phi) \cos(\theta) \cos(\phi) \\ \mathbf{GM}_L \sin(\theta) \cos(\theta) \cos(\phi) \\ (-\mathbf{GM}_L \cos(\theta) + \mathbf{GM}_T) \sin(\phi) \sin(\theta) \end{bmatrix} \quad (2.83)$$

Now assume that

$$\mathbf{r}_r^b \chi \delta \mathbf{f}_r^b = \mathbf{0} \quad (2.84)$$

This assumption is valid, since this term is small compared to (2.83). Finally, we reach the gravity and buoyancy restoring forces:

$$\boldsymbol{\tau}_g = - \begin{bmatrix} \delta \mathbf{f}_r^b \\ \mathbf{m}_r^b \end{bmatrix} = \begin{bmatrix} -z_0 z \sin(\theta) \\ z_0 z \cos(\theta) \sin(\phi) \\ z_0 z \cos(\theta) \cos(\phi) \\ \rho^w \mathbf{g} \nabla \mathbf{GM}_T \sin(\phi) \cos(\theta) \cos(\phi) \\ \rho^w \mathbf{g} \nabla \mathbf{GM}_L \sin(\theta) \cos(\theta) \cos(\phi) \\ \rho^w \mathbf{g} \nabla (-\mathbf{GM}_L \cos(\theta) + \mathbf{GM}_T) \sin(\phi) \sin(\theta) \end{bmatrix} \quad (2.85)$$

Assuming yz-symmetry and small angle (linear) theory holds, restoring matrix can be written as follows [1]:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -z_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\rho^w \mathbf{g} \nabla \mathbf{GM}_T & 0 & 0 \\ 0 & 0 & 0 & 0 & -\rho^w \mathbf{g} \nabla \mathbf{GM}_L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.86)$$

Now note that the buoyancy force is equal to gravitational force for a semisubmersible vehicle in balance. Hence (2.86) becomes:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m & 0 & 0 & 0 \\ 0 & 0 & 0 & -m \mathbf{GM}_T & 0 & 0 \\ 0 & 0 & 0 & 0 & -m \mathbf{GM}_L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.87)$$

Equation in (2.87) will yield oscillatory behavior since it ignores the drag of water. From equality of gravitational and buoyancy forces at stable for a surface vehicle, the vehicle

should sink for a certain amount, which can be found by equating the buoyancy of the sinking area to the weight. Furthermore, the damping of water proportional to the speed of vessel on axes of buoyancy should be included in equations. Hence (2.87) becomes:

$$\mathbf{g} = [0 \ 0 \ g_1 \ g_2 \ g_3 \ 0]^T \quad (2.88)$$

where

$$g_1 = -9.81mz - \mu l w_i w \quad (2.89)$$

$$g_2 = -9.81\mathbf{GM}_T m \phi - \mu l w_i p \mathbf{GM}_T \quad (2.90)$$

$$g_3 = -9.81\mathbf{GM}_L m \theta - \mu l w_i q \mathbf{GM}_L \quad (2.91)$$

In equations above, l and w_i are length and width of the vehicle, respectively, and μ is the drag coefficient of the corresponding surface against the motion of the vehicle. In summary, the overall restoring force can be defined as:

$$\boldsymbol{\tau}_g = \mathbf{g} \quad (2.92)$$

2.7 Forces Induced by Thruster

The third term in right hand side of (2.68) is $\boldsymbol{\tau}_t$ representing the force induced by the thruster of the sea surface vehicle. In this section, the effects of propulsion system, the thruster and its resistance will be investigated. The propulsion device is considered as a propeller.

The propeller induced forces and torques are evaluated for the sea surface vehicle moving in forward velocity with propeller advancing in literature [33]. The effects of propeller are observed as force in surge and as torque in pitch.

$$\tau_t = \begin{bmatrix} \rho^w \kappa^2 D_p^4 K_t \\ 0 \\ 0 \\ 0 \\ \rho^w \kappa^2 D_p^5 K_t - 2\pi J_{pp} \dot{\kappa} \\ 0 \end{bmatrix} \quad (2.93)$$

In (2.93), ρ^w is the water density; κ is the number of propeller revolutions, D_p is the propeller diameter, J_{pp} is the added moment of rotary inertia of the propeller, K_t is the thrust coefficient. K_t is function of advance constant J_p which can be expressed as:

$$J_p = \frac{X(u)}{\kappa D_p} \quad (2.94)$$

In (2.94), $X(u)$ is the forward speed. There are various sources of resistance in thrust. The simplest way to express the resistance is as follows [38]:

$$X(u) = X_{u|u} u|u \quad (2.95)$$

where right hand side of (2.95) can be expanded as (2.74). Another important effect of thruster on the moving vessel is named wake fraction. The water around the stern attains a forward motion in the direction of the motion of the hull. The moving water is called the wake and it produces a difference between the speeds of the ship. The wake speed is expressed as:

$$J_p = (1 - w) \frac{X(u)}{\kappa D_p} \quad (2.96)$$

In (2.96), w is called the wake fraction. The wake fraction is generally a positive number; however, for vehicles moving in high speeds it can be non-positive. In those cases the wake fraction can be taken as zero [39], in our design, wake is taken as zero too.

Another effect of the propulsion system is thrust deduction. The effect is due to the interaction between the propeller and the vehicle. When a ship is towed, an area of high pressure over the stern occurs and the total resistance of propulsion reduces. The decrease in resistance is modeled as given in [8].

The thrust coefficient, K_t in (2.93) can be obtained from curves of propeller. The model can be chosen as cubic fit [33] of a function of advance constant as follows:

$$K_t = 0.154J_p^3 - 0.361J_p^2 - 0.222J_p + 0.430 \quad (2.97)$$

The force due to thrust is observed in surge axis as expected. Unless there is a flow anomaly, there is no contribution of thrust on sway axis. Heave is also assumed to be zero; as well as the roll.

Back to (2.93), if we assume constant rotation of propeller, we can drop the second term. Note that this assumption is valid, since right after the thrust command, the propeller rotation gets its new value and remain constant. However, as the propeller revolutions increase, for a small boat, the bow will rise. This is due to the torque component of thrust in pitch. We need to take the thrust reduction effect into account as well.

The rotation of the propeller will generate a motion in yaw direction as well. If not under control, this effect will result in a large arched trajectory. This is to be balanced by the other torques.

In general including the thrust reduction and assumptions, we can obtain forces and torques due to thrust as (2.98) below. In the equation C_{pu} is named as thrust reduction coefficient and C_{pq} is called pitch propeller coefficient.

$$\boldsymbol{\tau}_t = \begin{bmatrix} C_{pu}\rho^w\kappa^2D_p^4K_t \\ 0 \\ 0 \\ 0 \\ C_{pq}\rho^w\kappa^2D_p^5K_t \\ 0 \end{bmatrix} \quad (2.98)$$

2.8 Forces Induced by Rudder

In this section, the rudder model and the forces induced by rudder of the sea surface vehicle will be investigated, which is represented by $\boldsymbol{\tau}_r$ in (2.68). Rudder is the second actuator of our system other than the thruster. In Figure 2.4, rudder angles and their descriptions are given.

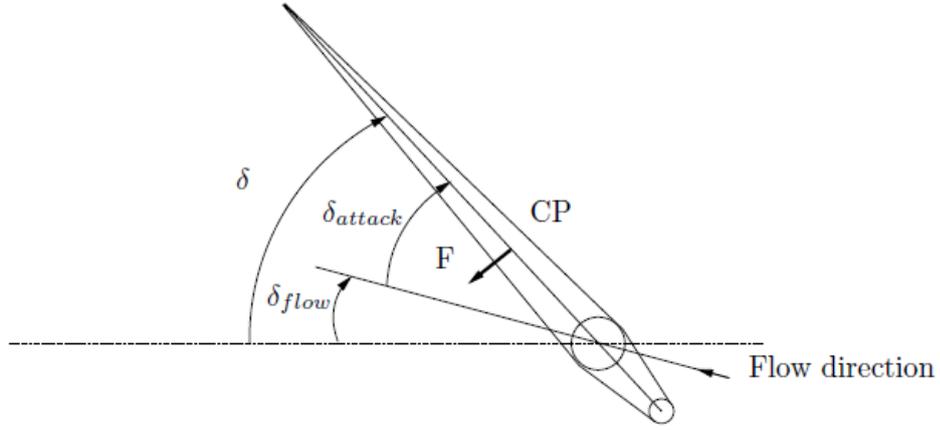


Figure 2.4 Rudder angles and descriptions [38]

The rudder angle command is δ . The effective force and torques are created by the rudder is related to the δ_{attack} , which is the angle command after the angle due to water flow is subtracted.

The total hydrodynamic forces and moments on the rudder can be modeled as a forces and torques acting on a single point which is called center of pressure (CP). $CP = [x_{cp} \ y_{cp} \ z_{cp}]^T$ and it expressed in body-fixed frame. The forces on the rudder are normal to the center plane of the rudder [39]. The force normal to rudder surface can be expressed as:

$$\mathbf{F} = \left\{ \begin{array}{ll} 1/2\rho^w C_F A_r V_{av}^2 \sin\left(\frac{\pi}{2} \frac{\delta_{attack}}{\delta_{stall}}\right) & \text{if } |\delta_{attack}| < \delta_{stall} \\ 1/2\rho^w C_F A_r V_{av}^2 \text{sgn}(\delta_{attack}) & \text{otherwise} \end{array} \right\} \quad (2.99)$$

In (2.99), C_F is the lift coefficient, A_r is the rudder area, and V_{av} is the flow passing through the rudder. δ_{stall} is the rudder stall angle, which is in a range of 35 to 45 degrees. The angle of attack δ_{attack} is the relative angle between the rudder and the flow as shown in Figure 2.4. The angle of attack can be calculated from the sway velocity, v , surge velocity, u and the turn rate of the ship $(x_{cp} - x_G)r$ as follows:

$$\delta_{attack} = \delta - \delta_{flow} = \delta - \arctan\left(\frac{v + (x_{cp} - x_G)r}{u}\right) \quad (2.100)$$

The forces and torques due to rudder induced on the sea surface vessel are given in (2.101) and (2.102), respectively. Note that after the forces are found the moments are calculated with the cross product of the forces created by the moment arm. The cross product is as defined in (2.5) and (2.6).

$$\begin{aligned} \mathbf{X}_{rudder} &= -\mathbf{F}(u, V_{av}, v, r, \delta)\sin(\delta) \\ \mathbf{Y}_{rudder} &= \mathbf{F}(u, V_{av}, v, r, \delta)\cos(\delta) \end{aligned} \quad (2.101)$$

$$\mathbf{Z}_{rudder} = \mathbf{0}$$

$$\begin{aligned} [\mathbf{K}_{rudder}\mathbf{M}_{rudder} \mathbf{N}_{rudder}]^T \\ = (\mathbf{CP} - \mathbf{CG})\mathbf{x}[\mathbf{X}_{rudder}\mathbf{Y}_{rudder} \mathbf{Z}_{rudder}]^T \end{aligned} \quad (2.102)$$

Now back to (2.99), we need to find the flow passing through the rudder. Note that in general, rudder is located behind the propeller; hence, the flow passing the rudder V_{av} is affected from the propeller. The average flow can be expressed as [40]:

$$V_{av}^2 = V_a^2 + C_T T \quad (2.103)$$

where

$$C_T \approx \frac{6.4}{\pi\rho w\zeta D_p} \quad (2.104)$$

In (2.104), ζ is the rudder span, D_p is the propeller diameter. T in (2.103) is the thrust generated. The average flow can be expressed as follows [38]:

$$V_{av}^2 = [(1 - w^2) - C_T^2 \frac{X_{uu}}{(1-t)}]U^2 \quad (2.105)$$

t in (2.105) is thrust deduction number of order 0.05-0.2 [8]. By substituting (2.105) in (2.99) from (2.101) and (2.102), the forces and moments due to rudder can be found as

follows (Note that rudder effect should be inserted into the system either by terms of forces or by terms of torques):

$$\boldsymbol{\tau}_r = [X_{rudder} Y_{rudder} Z_{rudder} K_{rudder} M_{rudder} N_{rudder}]^T \quad (2.106)$$

2.9 Air Drag and Wind Forces

In this subsection, forces on the sea surface vehicle due to air drag and wind will be investigated. These forces and torques are represented in (2.68) with $\boldsymbol{\tau}_a$. Since, the vehicle is not completely sinking (in fact, most of it is above the sea surface), it is affected from the wind and the air drag due to its motion. Note that, the real forces exist in three linear directions, the torques are created due to rotational effects of these forces which are proportional with the distance between the surface of the vehicle on which the force is acting and the center of gravity, which may be taken as midpoint of the vehicle for the sake of computational ease, assuming symmetry. Hence, the main idea in computation of forces and torques due to air drag and wind is first to find the forces and then obtain the torques.

Whether it has origin of wind or air drag due to velocity of the sea surface vehicle, the forces acting on the vehicle due to air is related to the relative velocity of the vessel in air. This can be attained by subtracting velocity of the vehicle from the velocity of the wind. The force due air drag can be stated as:

$$\mathbf{F}_a = A_a \mathbf{P}_a C_{d,a} \quad (2.107)$$

In (2.107), \mathbf{F}_a is the force due to air drag, A_a is the area of the vehicle in touch with air, in other words, the area on which air drag force affect, \mathbf{P}_a is the air pressure, and $C_{d,a}$ is the drag coefficient of air. For plate surfaces air has drag coefficient of 2 [8].

$$P_a \approx 2.56 V_a^2 \quad (2.108)$$

Air pressure is proportional with the squared of air velocity as shown in (2.108). Note the V_a in (2.108) is the relative speed of air with respect to the vehicle.

Substituting relative velocities and the areas of vehicle above water in (2.107), we can find forces due to air drag and wind. Note that since the vehicle has small dimensions, it can be assumed that wind and air drag affects each point on the planar surface of vehicle equally. The torques can be found by taking cross-product of the moment arm with the forces. For computational ease, assuming symmetry, the center of gravity is taken as midpoint of the vehicle. Then the moment arm and the torques become:

$$\boldsymbol{\lambda} = \left[\frac{l}{2} \quad \frac{wi}{2} \quad \frac{h}{2} \right]^T \quad (2.109)$$

$$\boldsymbol{T}_a = \boldsymbol{\lambda} \times \boldsymbol{F}_a \quad (2.110)$$

In (2.109), l is length, wi is the width and h is the height of the vehicle. The cross product in (2.110) is as defined in (2.5) and (2.6). However note that for a wind to make the vehicle rotate, it should act in opposite direction to the two ends of the vehicle. For a small vehicle, the rotating contribution of wind and air drag on the vehicle can be neglected. From (2.107) and (2.110), forces and torques due to air drag and wind can be stated as:

$$\boldsymbol{\tau}_a = [\boldsymbol{F}_a \ 0 \ 0 \ 0]^T \quad (2.111)$$

2.10 Forces due to Water Current

In this subsection effect of water current (ocean currents) on our mathematical model will be investigated. These forces and torques are represented in (2.68) with $\boldsymbol{\tau}_c$. The forces and moments due to ocean currents can be expressed as:

$$\boldsymbol{\tau}_c = (\boldsymbol{M}_{FK} \dot{\boldsymbol{V}}_c) + (\boldsymbol{M}_{DF} \dot{\boldsymbol{V}}_c + \boldsymbol{D}_p(\boldsymbol{V}_c) \boldsymbol{V}_c) + ((\boldsymbol{D}_s(\boldsymbol{V}_c) + \boldsymbol{D}_w(\boldsymbol{V}_c) + \boldsymbol{D}_m(\boldsymbol{V}_c)) \boldsymbol{V}_c) \quad (2.112)$$

In (2.112), the first term is Froude-Kriloff term, the second term is diffraction forces and the third one is the viscous forces. In the equation, \boldsymbol{V}_c is the fluid velocity in body-fixed reference frame. The Froude-Kriloff inertia matrix \boldsymbol{M}_{FK} represents the inertia of the displaced water. The calculation of \boldsymbol{M}_{FK} is similar to \boldsymbol{M}_{RB} ; however, mass of displaced water can be taken as the volume of the vehicle under water. The center of gravity of the

water can be taken as the $(0, 0, h/2+l)$ where l is the length of vehicle under water, and h is the height of the vehicle. Note that l changes with time.

The dominant term in (2.112) is Froude-Kriloff term. It can be assumed that there is no damping due to water current. Now since the velocity of the water current is with respect to body-fixed reference frame, as the vehicle moves, the V_c changes. Assuming constant flow, the derivative of the current flow will be equal to the negative acceleration of the vehicle in movement. Hence the forces and moments due to water flow can be attained as:

$$\boldsymbol{\tau}_c = \mathbf{M}_{FK}\dot{\mathbf{V}} \quad (2.113)$$

Now note that, though we have included water flow as a force into system, it is needed to modify the equations found above since some of the velocities shall include the water flow as well. Hence, in equations above, as the velocity we need to use the total velocity of flow and the vehicle. By doing so, we can attain the correct Earth-fixed position and orientation; since we include the water flow in our derivations.

2.11 Conclusion

In this chapter, a nonlinear mathematical model of a sea surface vehicle has been developed, as presented in the equation below.

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_g + \boldsymbol{\tau}_t + \boldsymbol{\tau}_r + \boldsymbol{\tau}_a + \boldsymbol{\tau}_c \quad (2.114)$$

In (2.114), left hand side stands for the dynamics of rigid body and the added mass, while the right hand side is the sum of external forces. In the equation above, $\boldsymbol{\tau}_d$ is the forces and torques due to damping of water surrounding, $\boldsymbol{\tau}_g$ is gravitational and buoyancy forces and torques, $\boldsymbol{\tau}_t$ is forces and torques provided by the thruster, $\boldsymbol{\tau}_r$ represents the forces and torques due to rudder, $\boldsymbol{\tau}_a$ stands for the forces and torques due to air drag and wind, and $\boldsymbol{\tau}_c$ is for forces and torques of water (ocean) currents. In the previous subsections the mathematical derivations of each of these components were investigated.

Note that waves are not included in this mathematical model; their effects are to be considered as disturbances.

Based on the mathematical model developed, a SIMULINK model is created. The model is demonstrated in Figure 2.5, below.

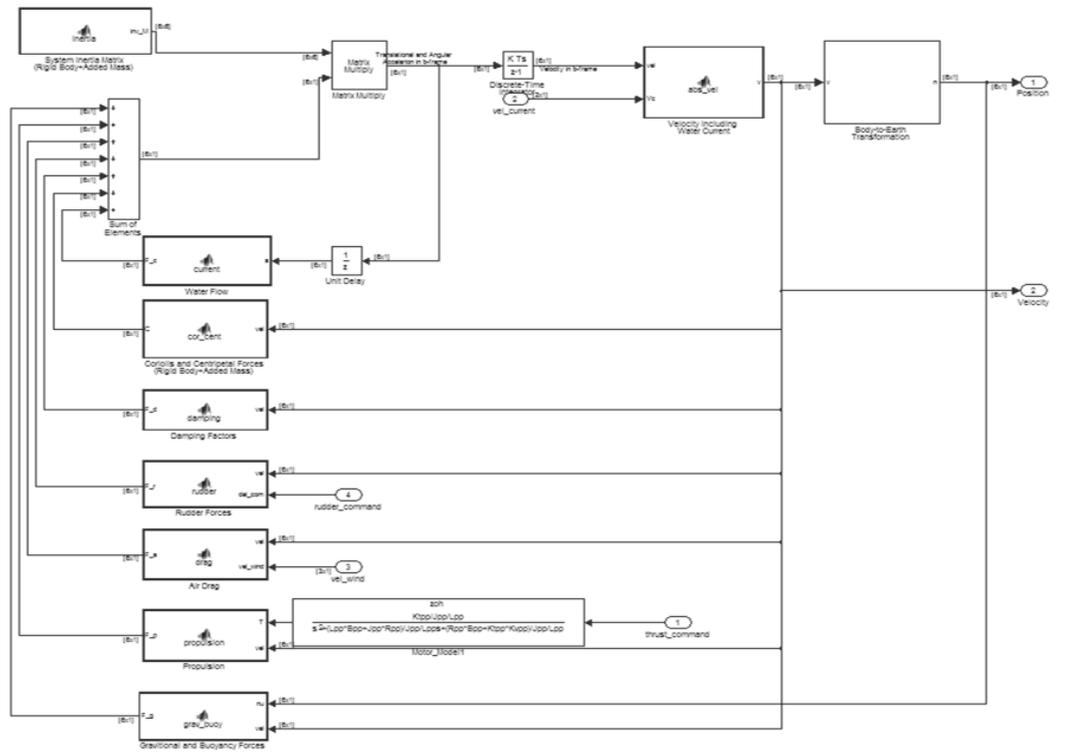


Figure 2.5 Mathematical model of a sea surface vehicle in SIMULINK

In Figure 2.5, body to earth transformation block is the subsystem form of Figure 2.3. The mathematical model takes 1-by-1 thrust command, 1-by-1 rudder command, 3-by-1 wind velocity and 2-by-1 current velocity as inputs and outputs the 6-by-1 Earth-fixed position vector. The thrust command enters the system as an output of a DC motor.

CHAPTER 3

AUTOPILOT DESIGN FOR SEA SURFACE VEHICLES

3.1 Introduction

In this section, various autopilots will be designed for the unmanned sea surface vehicle (USSV), whose mathematical model has been derived in the previous section. There will be two major autopilot design structures, i.e., controllers for USSV: Proportional-Integral-Derivative (PID) controller, Linear Quadratic Regulator (LQR) controller. The controllers will be designed according to the linearized model but they are applied directly on the nonlinear model.

In the autopilot design, wind and current velocity and un-modeled disturbances (such as wave) are considered as zero. There are two actuators in the model; the thruster and the rudder, which are taking their commands from a user or an upper level controller. The aim in the control of USSV or the goal of autopilots is to reach a desired surge velocity and yaw position.

In this section, first, the linearization of the model, the controllable subspace of the system, and sample time limitations for the controllers are discussed. In the following part, autopilots based on PID controllers are designed and applied on the system. Next, autopilots based on LQR controllers are designed and applied on the system. In the final subsection, these PID and LQR based autopilots are compared with respect to their performance and stability.

3.2 Fundamentals for the Autopilot Design

In this subsection, we construct various linearized models of the USSV, find its controllable subspace, and discuss the interpolation of the controllers associated with different linearization points and the sample time requirements of the discrete time controllers.

Linearization

First, remember our model in (2.19) and (2.114) is given by:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v} \quad (3.1)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_g + \boldsymbol{\tau}_t + \boldsymbol{\tau}_r + \boldsymbol{\tau}_a + \boldsymbol{\tau}_c \quad (3.2)$$

Now, we will not consider the forces and torques due to current, as it depends on an external force, and the air drag term will only include drag due to self-velocity of the vehicle. The state-space model desired to be achieved will include 6-DOF position and velocity as state, and thrust and rudder commands as control inputs.

$$\begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\mathbf{v}} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \boldsymbol{\eta} \\ \mathbf{v} \end{bmatrix} + \mathbf{B}\mathbf{u} \quad (3.3)$$

We define a state consisting of 6 positions/orientations and 6 linear/angular velocities. In (3.3), our state is of dimension 12, \mathbf{A} is a 12-by-12 matrix, \mathbf{B} is 12-by-2 matrix and \mathbf{u} is 2-by-1 column vector. In order to obtain (3.3), Equation (3.1) and (3.2) are interpreted as (3.4):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3.4)$$

Now, (3.4) is linearized around a point $(\mathbf{x}_0, \mathbf{u}_0)$:

$$\begin{aligned} \dot{\mathbf{x}}|_{x_0} + \Delta\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}|_{x_0, u_0} \Delta\mathbf{x} \\ &+ \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}|_{x_0, u_0} \Delta\mathbf{u} \end{aligned} \quad (3.5)$$

(3.5) can be written as:

$$\Delta \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{A}'\Delta \mathbf{x} + \mathbf{B}\Delta \mathbf{u} \quad (3.6)$$

Now, in order to obtain the standard form of state space representation, we augment the state vector and attain the following final form:

$$\begin{bmatrix} \Delta \dot{\mathbf{x}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}' & \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \Delta \mathbf{u} \quad (3.7)$$

(3.7) is the final form, and we have linearized our nonlinear formulas. The components of matrices are presented in the previous chapter.

The linearization is done for each pair in surge speed in the range of 0.5 to 5m/s with 0.5m/s steps and yaw position from -180 to 180 degrees with 20 degrees steps.

Controllable Subspace

Our aim in autopilot is to design controllers based on PID and LQR techniques. The controllability of (\mathbf{A}, \mathbf{B}) in different quiescent points is of curial importance since LQR controller works only on a controllable pair, (\mathbf{A}, \mathbf{B}) [41]. Hence at this step, one needs to find a controllable subspace of our state transition matrices (\mathbf{A}, \mathbf{B}) .

Let us consider the state space representation:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \quad (3.8)$$

In (3.8), \mathbf{x} is n-by-1 column vector, \mathbf{A} is n-by-n evolution matrix, \mathbf{B} is n-by-m matrix, \mathbf{u} is m-by-1 column vector, \mathbf{y} is r-by-1 column vector, \mathbf{C} is r-by-n matrix and \mathbf{D} is r-by-m matrix. Let the controllability matrix has rank $k < n$:

$$\text{rank}(\mathbf{C}\mathbf{O}) = \text{rank}([\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}]) = k < n \quad (3.9)$$

We form an n-by-n matrix \mathbf{P}^{-1} as:

$$\mathbf{P}^{-1} = [\mathbf{q}_1 \ \mathbf{q}_2 \ \mathbf{q}_3 \ \dots \ \mathbf{q}_n] \quad (3.10)$$

where first k columns of \mathbf{P}^{-1} are the linearly independent k columns of \mathbf{CO} . The other $(n-k)$ are chosen arbitrarily to make \mathbf{P} nonsingular. The transformation $\mathbf{x} = \mathbf{P}^{-1}\mathbf{x}_c$ or equivalently $\mathbf{x}_c = \mathbf{P}\mathbf{x}$ will transform (3.8) into:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}_c \\ \dot{\mathbf{x}}_{uc} \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_c & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{uc} \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_{uc} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_c \\ \mathbf{0} \end{bmatrix} \mathbf{u} \\ \mathbf{y} &= [\mathbf{C}_c \quad \mathbf{C}_{uc}] \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_{uc} \end{bmatrix} + \mathbf{D}\mathbf{u} \end{aligned} \quad (3.11)$$

where \mathbf{A}_c is k -by- k matrix, and \mathbf{A}_{uc} is $(n-k)$ -by- $(n-k)$ matrix. The k -dimensional sub-equation of (3.11) is:

$$\begin{aligned} \dot{\mathbf{x}}_c &= \mathbf{A}_c\mathbf{x}_c + \mathbf{B}_c\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_c\mathbf{x}_c + \mathbf{D}\mathbf{u} \end{aligned} \quad (3.12)$$

The state space representation in (3.37) is completely controllable and has the same transfer matrix as (3.8).

The proof of this derivation is based on the fact that the transformation $\mathbf{x} = \mathbf{P}^{-1}\mathbf{x}_c$ changes the basis of the state space from the orthonormal bases to the columns of $\mathbf{Q} = \mathbf{P}^{-1}$. The i th column of the evolution matrix in (3.11) is the representation of \mathbf{A}_{qi} with respect to the orthonormal basis of columns of \mathbf{Q} . The column vectors \mathbf{A}_{qi} can be linearly dependent on the set of \mathbf{q}_i within $i=1,2,\dots,k$ range and are linearly independent of \mathbf{q}_i s $i=k, k+1, \dots, n$ set. Hence the evolution matrix is formed as in (3.11). Similar arguments are valid for \mathbf{B} and \mathbf{C} matrices, as well.

Now, $\mathbf{x}_c = \mathbf{P}\mathbf{x}$ transformation divides the n -dimensional state space into two subspaces. The first subspace is k -dimensional and all the vectors are of the form $[\mathbf{x}_c \ 0]$ and the second subspace is $(n-k)$ dimensional and its vectors are of the form $[0 \ \mathbf{x}_{uc}]$. Since the representation in (3.37) is controllable, the control input \mathbf{u} can transfer \mathbf{x}_c from any state to any other state; however, it has no effect on \mathbf{x}_{uc} , neither directly nor through \mathbf{x}_c since they are independent.

For different linearization points, controllable subspaces are found. The dimension of controllable subspaces are found to be 6, while, the sum of the dimensions of controllable and uncontrollable subspaces are 12.

For a linearization point, four transfer functions can be obtained. For instance, constant surge speed, $u = 1$, and zero yaw angle, $\psi = 0$, the linearized (\mathbf{A}, \mathbf{B}) matrices are taken. For \mathbf{C} matrix, a 2-by-12 zero matrix, with (1,7)th and (2,6)th entries being chosen as 1. With this choice of \mathbf{C} matrix, only surge speed and yaw position is left to be observed.

The obtained system will be a two-input (thruster and rudder command control inputs), and a two-output (yaw position and surge speed), i.e., a MIMO system. There are four transfer functions in the s domain representation.

Interpolation

For any value of surge speed and yaw position, a weighted combination of the model linearized at different linearization points are used in an interpolation, and when needed, in an extrapolation process.

The interpolation is the weighted sum of the control inputs coming from controllers obtained for various linearization points.

$$\mathbf{c} = \sum w e_i \mathbf{c}_i, \quad \sum w e_i = 1 \quad (3.13)$$

In (3.37), \mathbf{c} is the command vector representing thruster command and rudder command. The command is obtained as the weighted sum of the commands acquired for the i th linearization point. Sum of weights is equal to 1.

The weights are determined by the following procedure:

$$w e_i = \frac{\xi_i}{\sum \xi_i}$$

$$\xi_i = \frac{1}{\|x - x_i\| + \epsilon_1} \quad \epsilon_1 > 0 \quad (3.14)$$

$$\xi_i = 0 \text{ if } \xi_i < \epsilon_2$$

ξ_i s are the reciprocal of distance from the current state to the linearization point i . ϵ_1 is a small number close, but not equal to, zero. This number is inserted to have finite ξ_i when a linearization point is hit. ϵ_2 is chosen to filter out the effects of linearization points not close to the operating point.

Sample Time of Autopilots

The sample time of the autopilots is determined according to phase margins of four linearized models of the system in each linearization points. Delays are inserted to the linearized systems and maximum delay for zero phase margins are recorded. Minimum of the four phase margins for each linearization points are determined. Among all minimum delays for linearization points, the minimum of the minimums are selected.

The minimum of the maximum delay that can be inserted to the linearized systems are observed to occur for rudder command to yaw position transfer functions, as expected; since, this transfer function has the slowest response among others. The maximum tolerable delay is found to be 0.69 sec. This delay value corresponds to approximately 1.44 Hz of sampling frequency.

Now, note that only open loop plant transfer function is considered in determining the sampling time of the autopilots. However, controllers themselves will introduce phase delays to the open loop system. Hence the sampling frequency above will be unstable with additional however small delays introduced into the system. Thus, by rule of thumb, at least ten times the frequency found above is chosen as the sampling frequency in the simulations. The sampling frequencies of the autopilots are selected as 16Hz, in other words, the sample time is chosen as 0.0625 sec.

3.3 Proportional-Integral-Derivative (PID) Controller

In a PID controller, the proportional-integral-derivative of the error function, the difference between the reference input and output feedback is fed as the control input to the system. The derivative term is first order filtered; the controller is in fact, of type PIDF.

We have four PID controllers to be implemented in our system; the first two take feedback from surge speed, and the desired surge speed or yaw position as reference inputs, and feed the PID output as the control input to the thruster command and rudder command. The corresponding transfer functions are the ones interacting among thruster command and rudder command and the surge speed of the four abovementioned transfer functions. The next pair makes the same job from yaw position feedback.

The transfer function of the SISO system, taking thruster command as input and surge speed as output for surge speed of 1 and yaw position of 0 degrees is as follows:

$$u(s) = \frac{0.4597}{s+0.118} \tau_{t,x}(s) \quad (3.15)$$

In (3.37), $u(s)$ is the surge speed and $\tau_{t,x}(s)$ is force in surge direction of thruster torque. This equation is obtained as a result of matrix transformation of the (A, B, C) matrices as mentioned above.

A simpler system is designed in SIMULINK and a PID controller is implemented on this design as shown in the figure below.

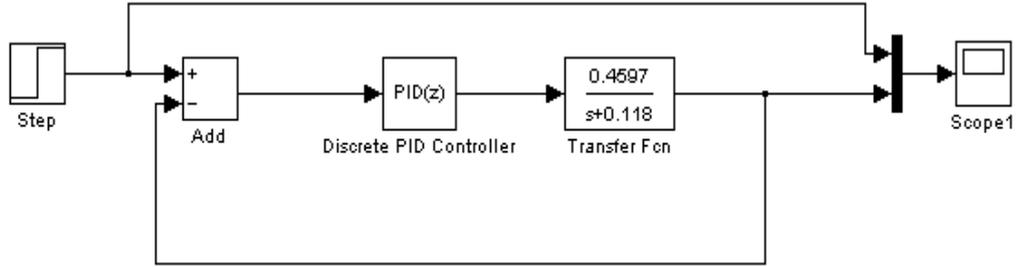


Figure 3.1 PID controller for surge speed for $u = 1, \psi = 0$

In Figure 3.1, a PID controller is shown for surge speed-thruster command for $u = 1, \psi = 0$. There are three more transfer functions for surge speed-rudder command, yaw position-thruster command and yaw position-rudder command. Furthermore, there are as many of these four PID controllers as the number of linearization points for different surge speeds and yaw positions. In order to be used in our design, all these PID controllers are to be designed and interpolation of them should be used to generate thruster commands and rudder commands. The controllers are of the following form:

$$c = \left(K_p + K_i \frac{T_s}{z-1} + K_d \frac{1}{T_f + \frac{T_s}{z-1}} \right) \quad (3.16)$$

The controllers are designed according to the following criteria. For controllers excluding yaw position-rudder command, should have a phase margin of 60 degrees, the closed loop system should be stable, its step response settling time should be less than 1 sec, and overshoot should be less than 10%. Since yaw position-rudder command transfer function itself has a dominant pole at lower frequencies, the settling time criterion is looser approximately 10 to 50 times.

For each linearization point, four linearized model and corresponding four discrete time PID controllers are generated. PID parameters of the controllers are auto-tuned. In auto-tuning algorithm, first the continuous time single-input-single-output transfer function is discretized with a sample time of 1/16. The algorithm tries to achieve a controller satisfying the abovementioned conditions. This job is accomplished by using MATLAB's 'pidtune' function. This function requires cross-over frequency and phase margins for closed loop system as argument and tunes the controller parameters according to the method in [42]. The phase margin is kept as 60 degrees; however an upper algorithm is developed to choose cross-over frequency accurately enough.

In determination of PID parameters, the main objectives are to keep step response settling time close to 1, overshoot less than 10%, and PID parameters less than 20, as much as possible. The algorithm uses trial and error method to achieve all of these conditions [42]. In each step, a new cross-over frequency is entered, a PID controller generated, closed-loop response is evaluated, and the direction to increase or decrease the cross-over frequency is determined, similar to a solution of an optimization problem, until all the criteria are met. The algorithm flowchart is presented below. In Figure 3.2, CF stands for the cross-over frequency determined by the algorithm, ST is the settling time criterion entered by the user, st is the settling time of the response, OS is overshoot criteria, os is the overshoot of the response, iter is the iteration count, end is the condition to end the algorithm if reached to 2. "Generate PIDF" means developing a PIDF controller using 'pidtune' function, "generate SR" means acquiring a step response with unit feedback law. The conditions to be met are $st < 1.2ST$, $os < 1.1OS$, $P < 20$, $I < 20$ and $D < 20$.

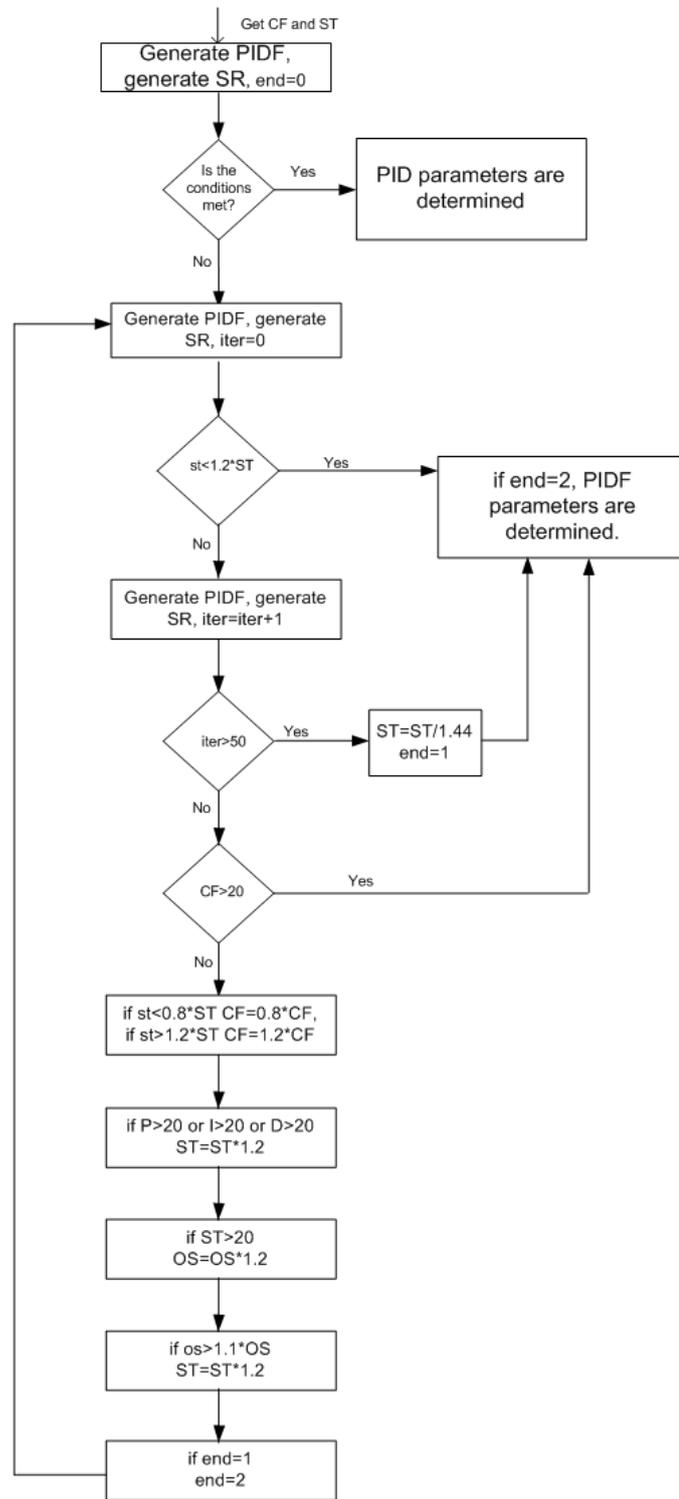


Figure 3.2. PID auto-tuning algorithm flowchart

The controllers imported to a model to determine weights of each controller in the system. The weights are found according to the result of error minimizing optimization problem. Sum of surge velocity-thruster command PID controller weight and yaw position-thruster command PID controller weight is fixed to 1. Similarly sum of yaw position-rudder command PID controller weight and surge velocity-rudder command PID controller weight are fixed to 1. The weights are determined for the linear model. Below in Figure 3.3, the model used to determine the weights is illustrated.

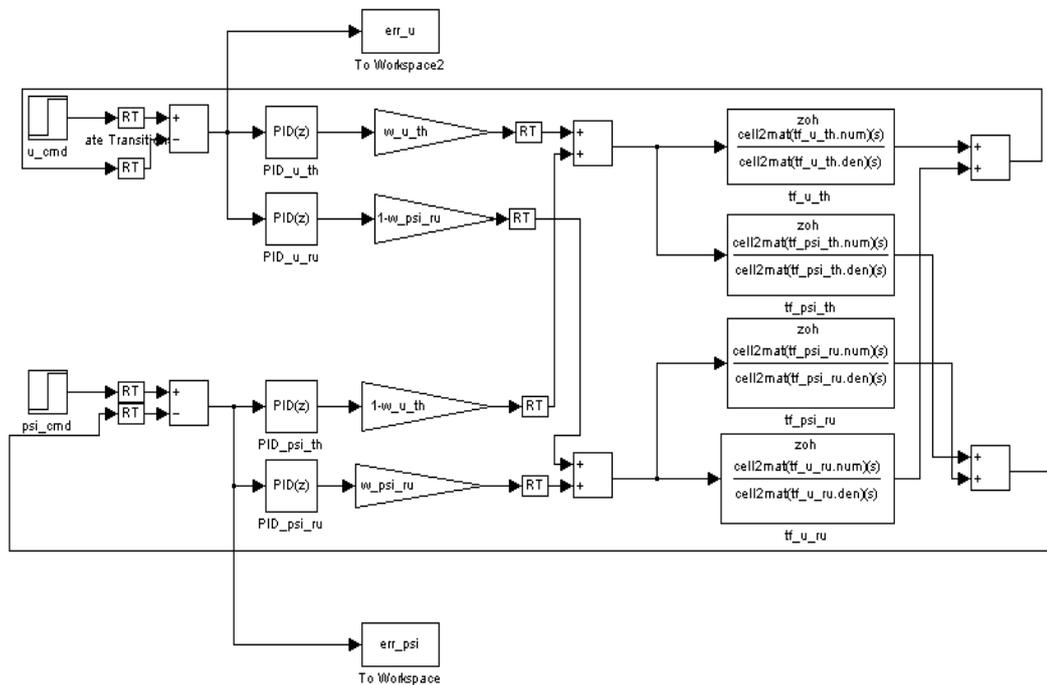


Figure 3.3. Model developed to determine weights of as single point PID controller

The weights of the controllers are determined offline. The flowchart of the optimization algorithm used to minimize the errors is shown in the figure below. The terms in the flowchart can be found from the figure above. With this algorithm we determine the weights which are expected to be around the following range: w_{u_th} : 0.8-1, w_{psi_th} : 0-0.2, w_{u_ru} : 0-0.4, w_{psi_ru} : 0.6-1. The expected values are achieved, i.e., more of the thrust

command is obtained from surge speed feedback, and more of the rudder command information from yaw position.

The algorithm solves the minimization problem in a classical way. First of all, note that we have two parameters to be determined, w_{u_th} and w_{psi_ru} , since w_{psi_th} and w_{u_ru} are their complements, respectively. The algorithm evaluates errors obtained from the SIMULINK model and for a certain number of iterations; it tunes the two parameters one by one. The method first determines an initial interval of uncertainty (iiu). This iiu is 10% of minimum of w_{u_th} (or w_{psi_ru} depending of which parameter is tuning) or $1-w_{u_th}$. According to the flowchart, either the parameter is increased or decreased. If a direction change occurs, then the iiu is halved. The iteration number is chosen as N for each parameter, and M for overall.

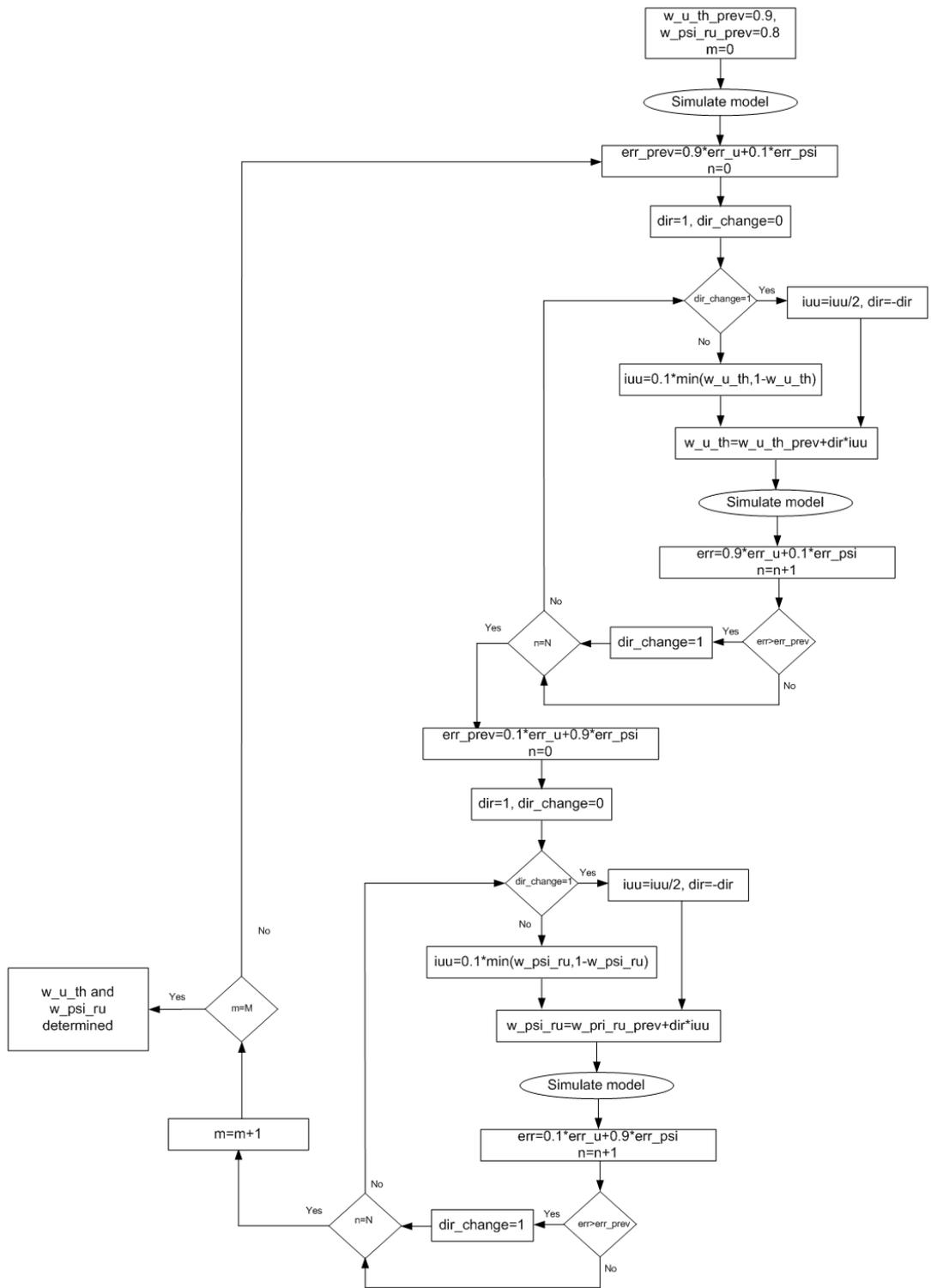


Figure 3.4. Weight determining algorithm flow chart

Now, for each linearization point, for each direct and cross transfer functions PID controller P, I, D, F parameters are determined and for each linearization point, weights of each controller are assigned. According to the current values of the states, namely yaw position and surge velocity, the interpolation weights are determined and the overall PID controller that is implemented as shown in Figure 3.5.

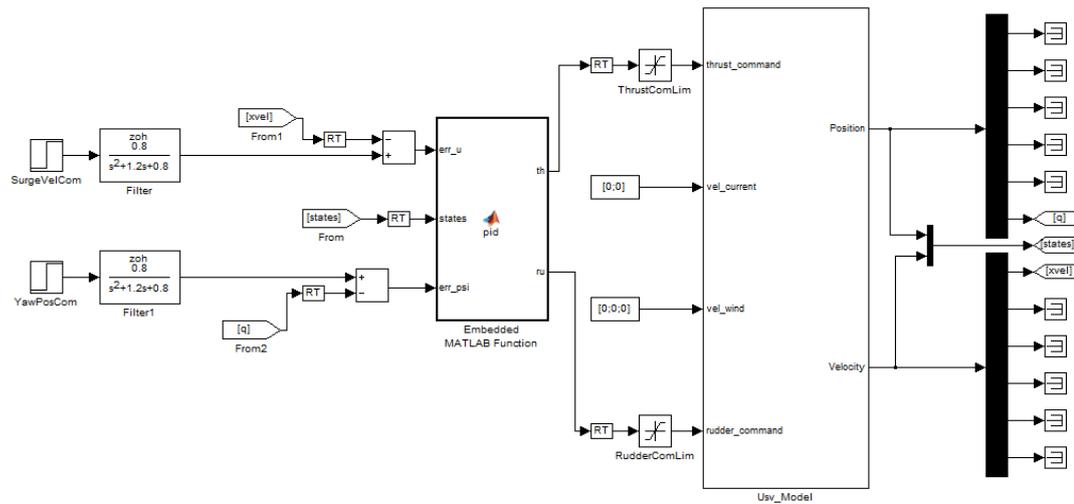


Figure 3.5. PID controller developed for USSV

After interpolating the control inputs at linearization points, the control inputs to the system for the present states are acquired. The control inputs are fed to the system. The model developed in SIMULINK is as shown below. The reference inputs are input to the PID controller again with a low pass filter of rise time 4.5sec, overshoot 6% and no steady-state error. The input and output commands of PID has limits that are coming from the nature of actuators. The sample of time of the controller is taken as 1/16sec.

Step response of PID controller in Figure 3.5, is shown in Figure 3.6 for surge speed command of 1m/s and yaw position command of $\pi/4$. In the plots, green lines are the commands and the blue lines are the responses of the PID controller together with plant. It can be seen that there are steady-state errors in yaw position step response which is degrading slowly.

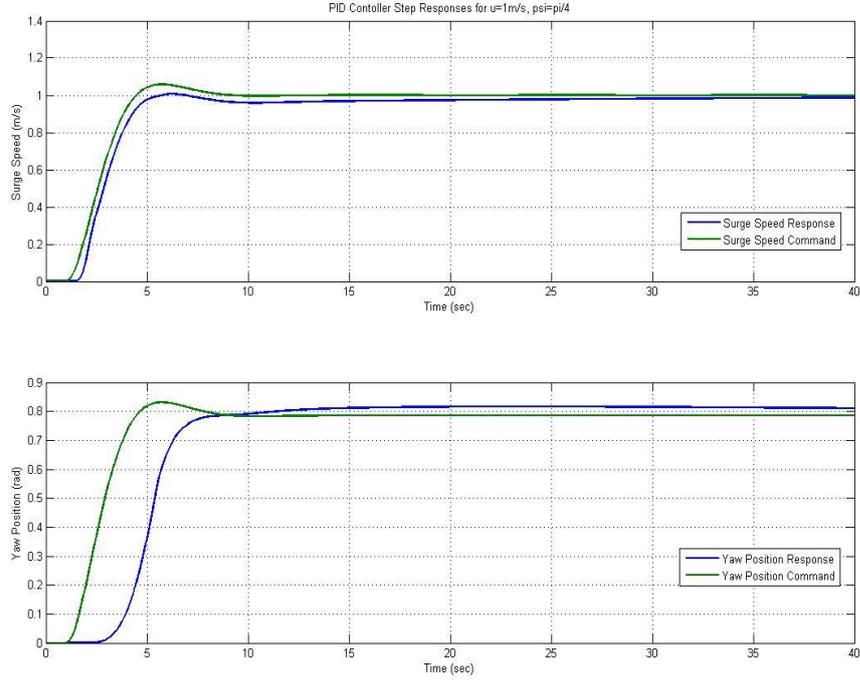


Figure 3.6. PID controller step responses for $u = 1$, $\psi = \pi/4$

3.4 Linear Quadratic Regulator (LQR) Controller

Another method of control, the second autopilot design in this study is Linear Quadratic Regulator (LQR) controller. Before proceeding with the autopilot design and simulation results, below fundamentals of LQR controller are discussed.

Let us define a linear system as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.17)$$

Let us define the quadratic cost functional as:

$$\begin{aligned} J(\mathbf{u}) = & \frac{1}{2} [(\mathbf{x}(t_f) - \mathbf{r}(t_f))]^T \mathbf{H} [(\mathbf{x}(t_f) - \mathbf{r}(t_f))] + \\ & \int_{t_0}^{t_f} \frac{1}{2} [(\mathbf{x}(t) - \mathbf{r}(t))]^T \mathbf{Q} [(\mathbf{x}(t) - \mathbf{r}(t))] + \end{aligned} \quad (3.18)$$

$$\frac{1}{2} \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) dt$$

(3.18) is the general cost function in linear tracking problems. t_f is the fixed final time, \mathbf{H} and \mathbf{Q} are real symmetric semi-definite matrices and \mathbf{R} is real symmetric positive definite matrix. Here, we assume that $\mathbf{x}(t_f)$ is left free, i.e., states and controls are unbounded. The aim is to get the state close to the reference without much expenditure of control effort. From (3.18) the Hamiltonian can be written as:

$$\begin{aligned} \mathbf{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = & \frac{1}{2} [(\mathbf{x}(t) - \mathbf{r}(t))^T(t) \mathbf{Q} [(\mathbf{x}(t) - \\ & \mathbf{r}(t))] + \frac{1}{2} \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) + \boldsymbol{\lambda}^T(t) [\mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t)] \end{aligned} \quad (3.19)$$

In (3.19), $\boldsymbol{\lambda}(t)$ is co-state vector. Optimal control $\mathbf{u}(t)$ can be obtained as:

$$\begin{aligned} \frac{\partial \mathbf{H}}{\partial \mathbf{u}} = & \mathbf{0} \\ \mathbf{R} \mathbf{u}(t) + \mathbf{B}^T \boldsymbol{\lambda}(t) = & \mathbf{0} \end{aligned} \quad (3.20)$$

From (3.20), we can acquire

$$\mathbf{u}(t) = -\mathbf{R}^{-1} \mathbf{B}^T \boldsymbol{\lambda}(t) \quad (3.21)$$

Above we have assumed \mathbf{R} is invertible. The co-state equation is as follows:

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial \mathbf{H}}{\partial \mathbf{x}} = -\mathbf{Q} \mathbf{x}(t) - \mathbf{A}^T \boldsymbol{\lambda}(t) \quad (3.22)$$

(3.21) and (3.17) gives:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) - \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \boldsymbol{\lambda}(t) \quad (3.23)$$

The final boundary condition of $\boldsymbol{\lambda}(t)$ is as follows. Now assume a transformation and define $\mathbf{s}(t)$ such that

$$\boldsymbol{\lambda}(t_f) = \mathbf{H} \mathbf{x}(t_f) - \mathbf{H} \mathbf{r}(t_f) \quad (3.24)$$

$$\lambda(t) = \mathbf{K}x(t) + \mathbf{s}(t) \quad (3.25)$$

(3.21) becomes

$$\mathbf{u}(t) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}x(t) - \mathbf{R}^{-1}\mathbf{B}^T\mathbf{s}(t) \quad (3.26)$$

If we differentiate (3.25):

$$\dot{\lambda}(t) = \dot{\mathbf{K}}x(t) + \mathbf{K}\dot{x}(t) + \dot{\mathbf{s}}(t) \quad (3.27)$$

Substituting the previous 4 equations into (3.27) yields:

$$-\mathbf{Q}x(t) - \mathbf{A}^T\mathbf{K}x(t) = \dot{\mathbf{K}}x(t) + \mathbf{K}\mathbf{A}x(t) - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}x(t) \quad (3.28)$$

$$\mathbf{Q}\mathbf{r}(t) - \mathbf{A}^T\mathbf{s}(t) = \dot{\mathbf{s}}(t) + \mathbf{K}\mathbf{A}x(t) - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{s}(t) \quad (3.29)$$

The Riccati equation is solved for \mathbf{K} with the final condition given below in (3.28):

$$\mathbf{K}(t_f) = \mathbf{H} \quad (3.30)$$

Equation (3.29) will be solved for \mathbf{s} with the final condition of:

$$\mathbf{s}(t_f) = \mathbf{H} \quad (3.31)$$

Equations (3.28) and (3.30) solve the final value problem. \mathbf{K} is an n-by-n matrix; hence, creating a system of n^2 first-order differential equations. However since \mathbf{K} is symmetric [41], the number of first order differential equations to be solved is $n(n+1)/2$.

It is shown by Kalman that [43] if the plant is completely controllable, if $\mathbf{H} = \mathbf{0}$, and if \mathbf{A} , \mathbf{B} , \mathbf{R} and \mathbf{Q} matrices are constant then \mathbf{K} becomes a constant matrix as time goes to infinity. Then for a time-independent \mathbf{K} matrix, we obtain the Riccati equation:

$$\mathbf{0} = \mathbf{K}\mathbf{A} + \mathbf{A}^T\mathbf{K} + \mathbf{Q} - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} \quad (3.32)$$

Similarly for time-independent \mathbf{s} matrix, we obtain the following equation:

$$\mathbf{0} = -[\mathbf{A}^T - \mathbf{KBR}^{-1}\mathbf{B}^T]\mathbf{s} + \mathbf{Qr}(t) \quad (3.33)$$

Now, λ and hence the optimal control \mathbf{u} can be found from (3.25). For small intervals of time, equations (3.32) and (3.37) can be merged; thus, (3.25) becomes:

$$\lambda(t) = \mathbf{K}(\mathbf{x}(t) - \mathbf{r}(t)) \quad (3.34)$$

The linear model of the system subject to LQR controller is attained by linearization on different points as explained in previous sections and outputs in different linearization points are interpolated.

LQR controller requires, as mentioned above, controllable (\mathbf{A}, \mathbf{B}) pair; hence, first on these linearization points, controllable subspaces are found. The unitary transformation matrix which presents separated controllable and uncontrollable modes of states is also acquired. The projections of cost matrices, \mathbf{Q} and \mathbf{R} , on the controllable subspace are also obtained with this transformation matrix.

The controllable portions of \mathbf{A} , \mathbf{B} , \mathbf{Q} and \mathbf{R} matrices are the ones to be used in time-invariant Riccati equation. The readily-built ‘care’ function in MATLAB is used to solve the continuous time algebraic Riccati equation. On different linearization points, corresponding Kalman gain matrix, \mathbf{K} , and state transformation matrix, \mathbf{T} are evaluated offline. This part is the most time-consuming part of the algorithm; however, the matrices of interest are obtained offline and will be embedded to the system during implementation.

After interpolating the control inputs at linearization points, the control inputs to the system for the present states are acquired. The control inputs are fed to the system. The model developed in SIMULINK is as shown below. The reference inputs are input to the LQR controller again with a low pass filter of rise time 4.5sec, overshoot 6% and no steady-state error. The input and output commands of LQR has limits as in the previous controller. The sample of time of the controller is again taken as 1/16sec.

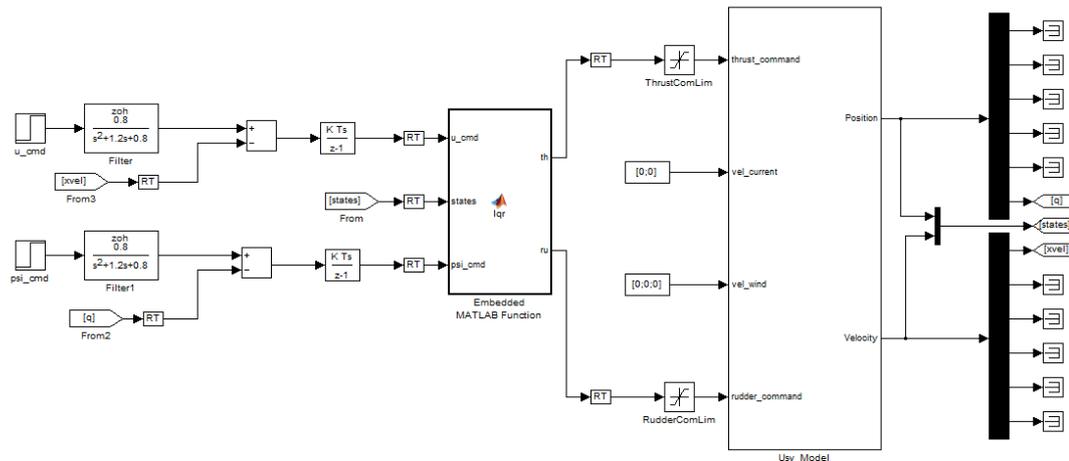


Figure 3.7. LQR controller design

In Figure 3.7 an LQR controller designed for the USSV is provided. The states to be controlled, namely surge velocity and yaw position have entries with greater diagonal coefficients in Q matrix, while the other entries will be smaller. R matrix is chosen according to results to make a compromise between two actuators of the model.

The responses of LQR controlled USSV is supposed to have a steady-state error. This is referred as a control of system with no integrator in literature. In order to overcome this problem, an artificial integrator is inserted into the system. The commands to the controller are not taken directly from the upper controller but the integral of the error between the command and the feedback is used.

The step response of the LQR controller for surge speed of 1m/s and yaw position of $\pi/4$ are provided below. The commands are shown in blue lines, and the step responses are plotted in green. The results are obtained to be better compared to those of the PID controller.

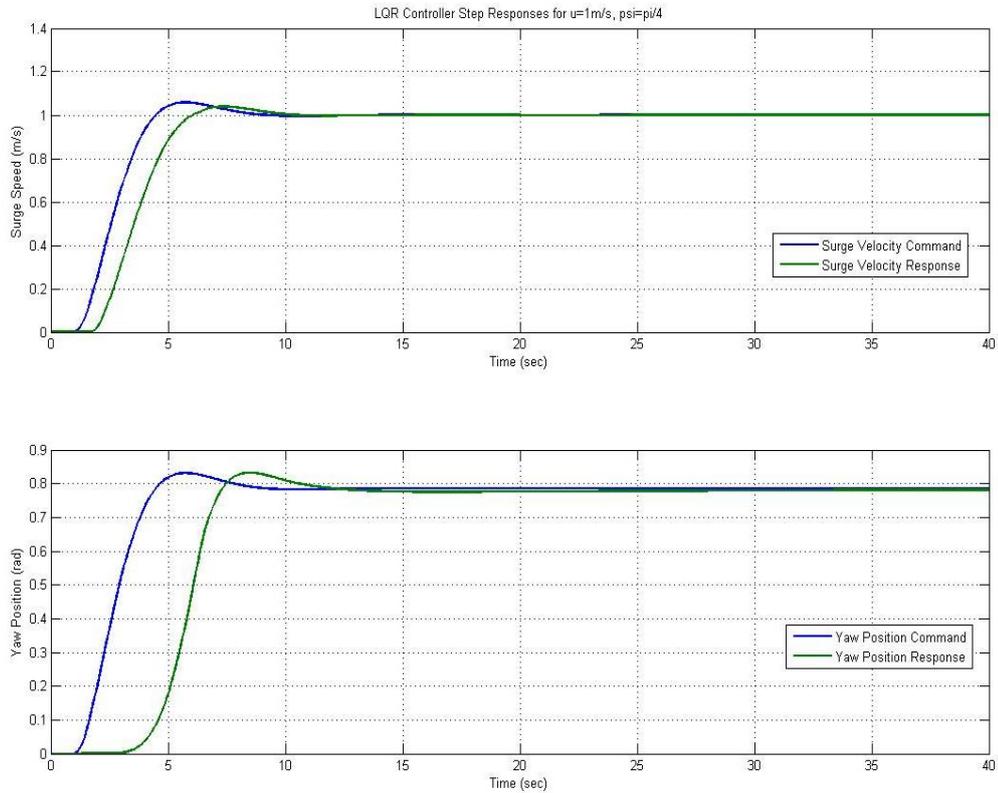


Figure 3.8. LQR controller step responses for $u = 1$, $\psi = \pi/4$

3.5 Design of Yaw Autopilot

In the previous subsections, methods to determine parameters of controller to control step responses of closed loop model of USSV are introduced. Now note that, although an upper level controller may request any yaw position and surge velocity command, the system cannot turn any angle at any speed. In this subsection, yaw autopilots will be designed to solve this problem.

When a yaw position command is applied to the closed loop system, if the angle request can be responded with the current speed of the vehicle, the system dynamics will act according to the controllers designed. However, if the angle of turn is greater than a certain yaw command, the vehicle may not achieve the required angle at that speed and may sink.

To acquire a safe control, a yaw autopilot should be designed. This upper controller has a simple task to do; the controller should take any surge velocity and yaw position request, for speeds greater than the limit speed to turn the yaw angle, the speed is decreased to that limit, after yaw position is achieved, the speed will be increased back to its original value. There is also an absolute upper limit for speed requests.

With empirical data obtained from controllers designed above, the yaw position commands and corresponding surge velocity commands that provides safe journey are tabulated below.

Table 3.1. Yaw position-surge velocity limits for yaw autopilot

Yaw Angle	Surge Velocity
10	3.40
20	2.60
30	2.00
40	1.60
50	1.20
60	1.00
70	0.80
80	0.40

Below the algorithm of the yaw autopilot is described.

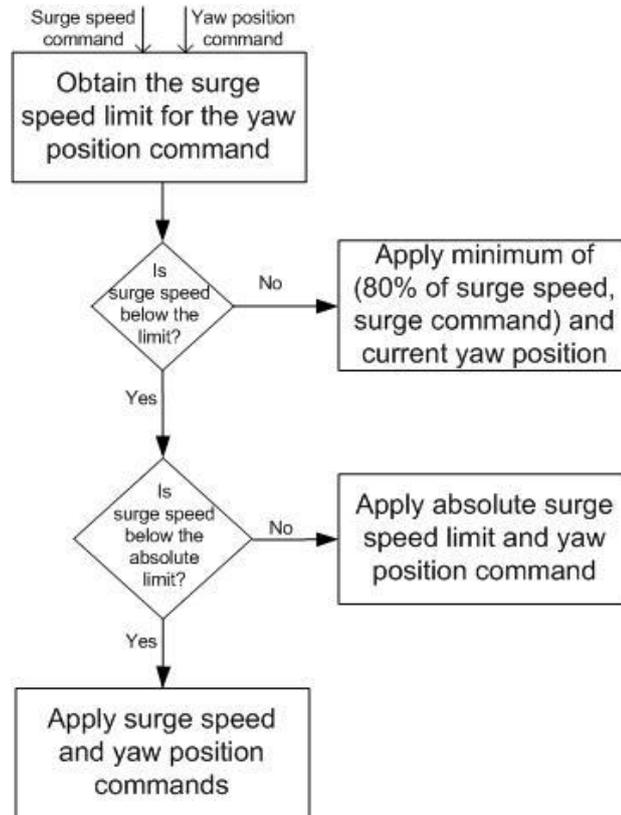


Figure 3.9. Yaw autopilot algorithm

3.6 Comparison between PID and LQR Autopilots

In the previous subsections two autopilots are designed. In this subsection, the designed autopilots will be compared. The comparison will be done on closed loop eigenvalue locations and the speed hold yaw autopilot behaviors.

Before comparing the autopilots, let us remember the model of the unmanned sea surface vehicle. We have two actuating signals to the systems, the thruster command and the rudder command. These commands make us move in surge position, yaw position and hence in sway position by generating velocities in these directions. Now the stability of the vehicle

on the other axes is implicit in the model of the vehicle itself. It has gravitational and buoyancy forces to stabilize itself in the other degrees of freedom.

The closed loop eigenvalues of the system are obtained according to linearized model where many of the positions and velocities are taken as zero. It is evident that the vehicle moves in the chosen directions and cannot be linearized on the other points. However, taking the entries as zero, the mathematical model has \mathbf{A} matrix with many elements of zero. Without controllers, the eigenvalues of \mathbf{A} have positive, negative or zero real parts.

Full stabilization is achieved when eigenvalues with positive and zero real parts are transferred to the negative side by a controller. However the controller itself has dimension of 6 out of 12. That is why the stability of the system is provided by injecting small attenuators on the linearized models. It is found that the eigenvalues now have entries with positive and negative real parts, but not with zero real parts. By the autopilots, our aim is to transfer these positive eigenvalues to negative side.

For surge velocity 1m/s and yaw position $\pi/4$, controllers are obtained as described in the previous subsections. The feedback loop is closed and transfer functions of closed loop systems are obtained. The closed-loop eigenvalues are derived from these transfer functions, or state-space representations. The eigenvalues far from zero in the negative plane means stability, while eigenvalues close to zero or on imaginary axis or with positive real part will yield unstable responses.

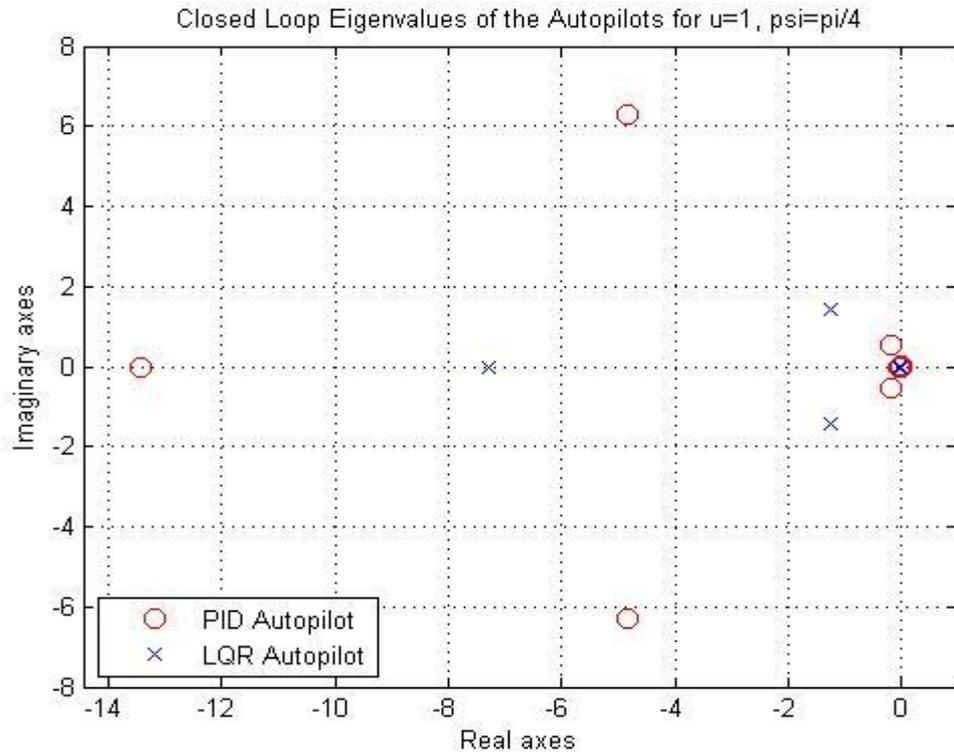


Figure 3.10. Closed-loop eigenvalues of autopilots for $u = 1$, $\psi = \pi/4$

The distribution of closed loop eigenvalues of autopilots is presented in Figure 3.10. The eigenvalues of the LQR controller are $\{-1.2368+1.24174i, -1.2368-1.4174i, -0.0446, -0.0050, -7.2701, -0.0071\}$ which are all negative; and the closed-loop eigenvalues of the PID controller are $\{-13.4377, -4.8419+6.2718i, -4.8419-6.2718i, -0.2035+0.5543i, -0.2035-0.5543i, -0.0361, -0.0017+0.0030i, -0.0017-0.0030i\}$, which are also all negative. The eigenvalues of the PID controller are closer to the imaginary axis; low performance of the PID controller can be explained by this distribution.

Although possessing negative real parts, the closed-loop eigenvalues of both PID and LQR autopilots are very close to the imaginary axis. Note that the reduced controlled model of the overall system has a transformation matrix between the real state vector of 13 entries and controllable subspace of it with dimension of 6, and this matrix may be very complicated. Hence it becomes difficult to distinguish which closed-loop eigenvalue corresponds to which of the actual states. Yet, the system has actuations on surge speed and yaw position;

hence a direct control over two of the dimensions. Surge position has a controlling behavior with a pole at imaginary axis, since it is the integral of the measures being controlled. Sway speed and positions are controlled indirectly with yaw position control; hence, the controlling actions on these states are weak. Positions and speeds in heave, roll, and pitch directions are rooted in passivity of the vehicle that are not directly controllable, but may provide contribution with off-diagonal entries in state transformation matrix.

The second method of comparison of the two autopilots is their speed-hold behaviors. In this test, speed is kept constant and with yaw position references are applied in a sequence.

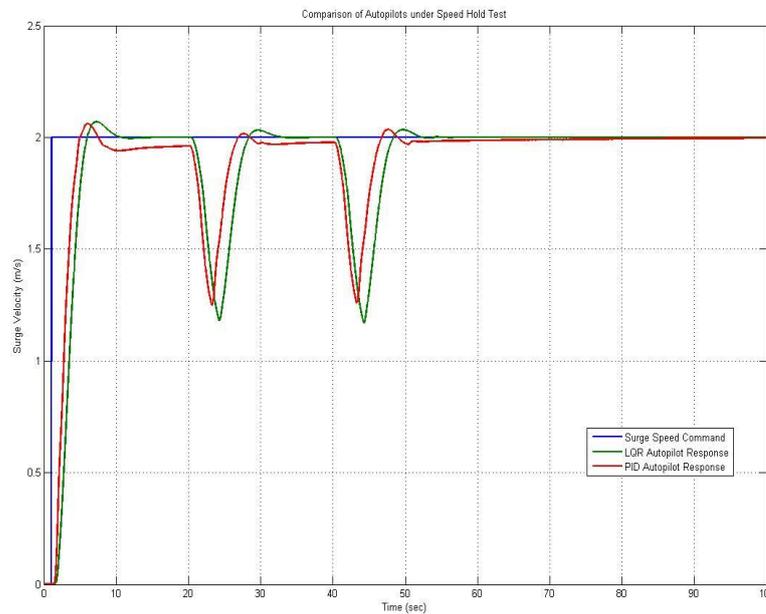


Figure 3.11. Comparison of surge velocity outputs of autopilots under speed hold

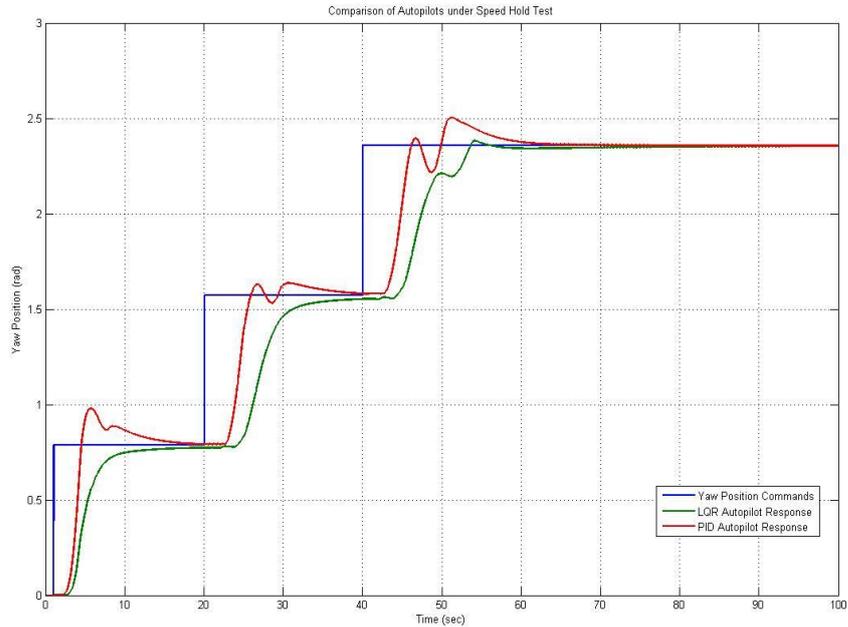


Figure 3.12. Comparison of yaw position outputs of autopilots under speed hold

From the two figures above, blue lines are commands, green lines are LQR autopilot responses and the red lines are PID autopilot responses. We can conclude that the LQR controller behaves more successfully compared to the PID controller. The velocity degradations in order to turn the yaw command are greater in the PID controller. Furthermore, yaw position overshoots are bigger in the PID controller response. Note that the LQR controller has slower response but it is less oscillatory. The oscillations in the LQR controller degrade with time; while those in the PID controller keep being constant. Furthermore, steady-state error of the PID autopilot is significant compared to that of the LQR autopilot.

Before concluding the autopilot design chapter, it is meaningful to make a stability analysis of the overall controlled closed loop system. Note that we have an (\mathbf{A}, \mathbf{B}) pair for each of the linearized systems and generated PID and LQR controllers for these linearized systems.

A stability analysis with a PID controller will be misleading since it is difficult to distinguish poles of the controller and the system when the loop is closed. We may not be able to observe those poles making the system unstable with the closed loop PID controller.

Stability analysis with an LQR controller needs further effort. Note that an LQR controller is only defined on the controllable subspace of an (\mathbf{A}, \mathbf{B}) pair. Remember the following equation:

$$\begin{bmatrix} \dot{\mathbf{x}}_c \\ \dot{\mathbf{x}}_{uc} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_c & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{uc} \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_{uc} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_c \\ \mathbf{0} \end{bmatrix} \mathbf{u} \quad (3.35)$$

We know that for zero reference signal:

$$\mathbf{u} = -\mathbf{K}\mathbf{x}_c \quad (3.36)$$

Now we can merge (3.35) and (3.37) as follows:

$$\begin{bmatrix} \dot{\mathbf{x}}_c \\ \dot{\mathbf{x}}_{uc} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_c - \mathbf{B}_c\mathbf{K} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{uc} \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_{uc} \end{bmatrix} \quad (3.37)$$

(3.37) is a valid representation; since the effects of control signal that is obtained from the controllable states with Kalman feedback gain are only observed on controllable states. The uncontrolled states are not related to Kalman feedback or the input; otherwise, they should be in the controllable subspace as well.

Now (3.37) is a first order differential equation set, stability of which can be directly obtained from the eigenvalues of the evolution matrix. The LQR controller and the Kalman feedback gain is found over a linearization point with surge speed and yaw position variables. Other states are kept at zero. Such a linearization may lead to zero eigenvalues in the evolution matrix. Hence the stability can be provided by inserting attenuations into the linearized model. The attenuations on position and orientations are bigger compared to those on linear and angular velocities.

The stability of the system is examined with various attenuations across one of the linearization point: $u = 1$ m/s, and $\psi = \pi/4$. An example for the slight attenuations is as follows: $x = -1e-4$, $y = 1e-4$, $z = -1e-4$, $\phi = \pi/180$, $\theta = -\pi/180$, $v = 1e-3$, $w = -1e-5$, $p = 1e-5$, $q = 1e-5$, $r = 1e-5$. The pole-zero plot for the eigenvalues are provided in the figure below:

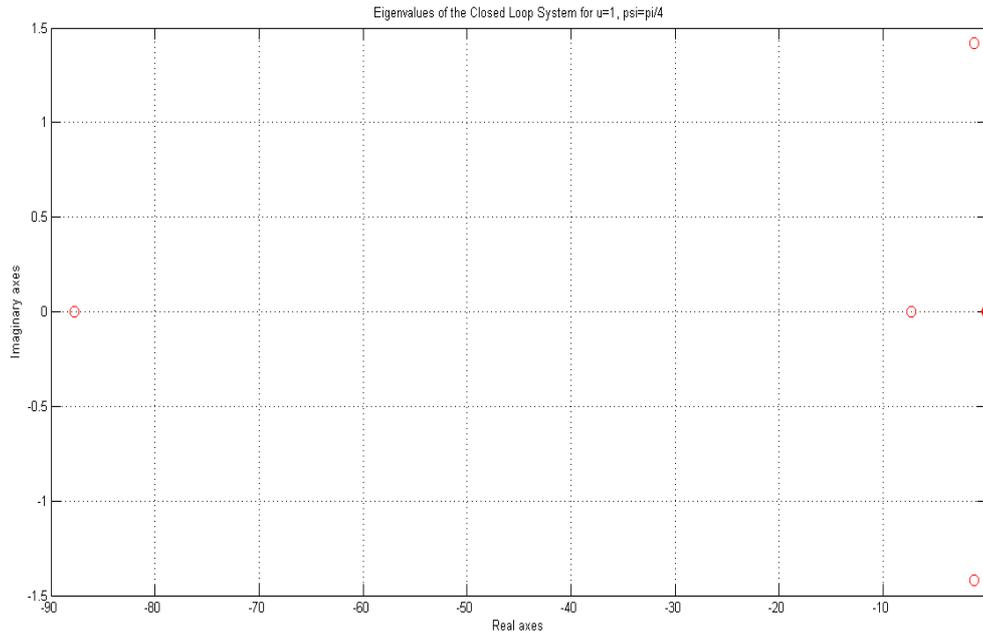


Figure 3.13. Closed-loop system eigenvalues for small perturbation across $u = 1$, $\psi = \pi/4$

From Figure 3.13, we observe many of the poles are near the imaginary axis; this is because the attenuations are very small compared to the linearization point entries. The eigenvalues are $\{-87.7580, -7.2701, -1.2368 + 1.4174i, -1.2368 - 1.4174i, -0.0446, -0.0072, -0.0049, -0.0806, -0.0001, -0.0001 + 0.0100i, -0.0001 - 0.0100i, -0.0001 + 0.0058i, -0.0001 - 0.0058i\}$. The overall system seems to be stable. This is an expected result, since the vehicle should be stable when it is under control. This result gives us clue about the validity of the mathematical model developed.

3.7 Conclusion

In this chapter, for an unmanned sea surface vehicle, mathematical model of which has been derived in Chapter 2, autopilots are designed. The yaw autopilots consist of controllers for surge speed and yaw position. There are two types of controllers studied, PID and LQR controllers.

The mathematical model obtained in Chapter 2 is nonlinear. In order to use linear controllers in the design, first the nonlinear model is linearized at various linearization points. Controllers designed for these linearized models are interpolated during real time operation of the system.

An LQR controller is defined for a completely controllable system. Hence, the controllable subspaces of the linearized models are found and LQR controllers are designed for these models.

The sample time of the controllers are determined by simulation as 16Hz. The controllers are operated in 1/16s sample time while the model operates at 1/800s sample time in simulations.

PID parameter and weights of cross-coupling terms on actuating signals are determined by an algorithm based on optimization. For the LQR design, an appropriate cost function is defined.

PID and LQR controllers are implemented and simulated. The simulation results are compared on speed hold tests. The closed-loop eigenvalues of the systems are also determined. It is observed that, even though being close to each other, the LQR controller is more stable compared to the PID controller.

System stability has also been checked and the mathematical model is validated.

CHAPTER 4

GUIDANCE DESIGN FOR SEA SURFACE VEHICLES

4.1 Introduction

In the previous chapter, autopilot algorithms are designed and simulated. The autopilot system requires surge velocity and yaw position commands from an upper controller, namely the pilot. These commands are fed to the system in accordance with the path the unmanned sea surface vehicle is supposed to follow.

The path to be followed can be provided by an upper controller or from the trajectory of another vehicle to be tracked. According to the aim of the mission, this guidance algorithm logic may change.

In this section, a path following guidance algorithm is designed. The algorithm is a single-step guidance algorithm to follow way-points provided by an upper controller. This algorithm takes two of the upcoming targeting position coordinates, and as it reaches the first point, the next point in the sequence becomes the primary step-ahead position target, and a new point from the upcoming the trajectory is requested. In the upcoming first subsection, the guidance algorithm is described in detail.

The aim in the guidance algorithm is to reach to the present way-point given within a limit of error, and attain the necessary yaw position to sail to the upcoming point. The algorithm is to supply necessary surge velocity and yaw position commands to the autopilots to move in surge and turn to the necessary orientation in yaw to move to the next point.

In the following subsection, an obstacle avoidance algorithm is designed and simulated. The obstacle avoidance block has its place between the autopilots and the guidance algorithm. In its way to the next point, surge velocity and yaw positions being commanded by the

guidance algorithm, the vehicle should avoid any obstacle on its way. Note that the points chosen from path and fed one-by-one as the guidance point, should be far enough from the obstacles for proper motion.

Under the guidance algorithm, the obstacle avoidance block works in each autopilot cycle, by the feedback from obstacle sensors, and changes the direction and speed of the motion. When there is no obstacle, only the guidance algorithm rules the motion.

Following the path provided by an upper controller, the vehicle may miss some of the way-points on its guided mission due to obstacles. An algorithm operates to overcome this problem and change the next way-point to the most appropriate one. This algorithm works on top of way-point guidance algorithm, and is investigated in detail.

The next-point generation algorithm continuously evaluates the possibility for the vehicle to reach the upcoming point, and other points ahead its way. With the distractions from the route towards the upcoming point, with proper weights, the next point is changed to the most suitable one. With this algorithm, the final position is reached as soon as possible while visiting all possible guidance points. Furthermore, this algorithm releases the need of placement of guidance points away from the obstacles, by enabling the vehicle not to visit them.

In the last subsection of this chapter, effects of disturbances on the guidance algorithm are investigated. So far, the autopilot and guidance algorithms are designed for hypothetical vehicles moving in a medium surrounded with no current flow, or wind; i.e., with no disturbance. Within the scope of this thesis, measurements are assumed to be noise-free. However, the effects of the wind and water currents are needed to be taken into consideration. The simulations are conducted and by trial and error method, the limits on amount of disturbances tolerable by the model are determined.

The performance of the guidance algorithm under disturbed environment is studied in the final subsection. The output with effects of disturbances is fed to the guidance algorithm as state information while guidance points are kept constant. This study is important for validation of algorithms in real life.

In each subsection, simulation results are provided.

4.2 Way-Point Guidance Algorithm

In this subsection, guidance algorithm will be explained in detail. The limits of the guidance algorithm, the points USSV can be guided and, the algorithm logic are investigated. The subsection finalizes with simulation results.

The guidance algorithm is implemented to guide the vehicle to a desired targeting point. The targeting point can be the next point in a path determined by an upper path planning algorithm according to the mission requirements or it can be the location of a moving target vehicle being followed. Depending on whether a target is being followed or next way-points are being provided to the guidance block, the logic of the running algorithm changes.

In this thesis, the way-points are assumed to be provided in order of presence to the algorithm. By this assumption, the system may require an upper controller to determine the way-points. The way-points are either determined by an algorithm or input from the user interface, yet they should be in consistency with the behavior of the vehicle.

In the guidance algorithm, it is important to choose correct surge speed and yaw position commands. The surge speed command should be related to the distance to be travelled. The commands are generated accordingly:

$$u = \frac{|P_{2,x}-P_{1,x}|^2 + |P_{2,y}-P_{1,y}|^2}{(|P_{2,x}-P_{1,x}| + |P_{2,y}-P_{1,y}|)^2} \quad (4.1)$$

$$\psi = \tan^{-1}\left(\frac{P_{2,y}-y}{P_{2,x}-x}\right) \quad (4.2)$$

In equations (4.1) and (4.2) P is a way-point. P_1 is the way-point that is just passed, and P_2 is the point to be reached, i.e., the next way-point. The surge speed command is kept constant during the travel from one guidance point to the next point. Now note that (4.1) yields a result slower than 1m/s. While the vehicle tries to achieve its orientation towards the next way-point the speed is kept in its initial value in (4.1), and this speed is chosen to be slow.

The yaw position command is calculated at each step. The orientation is the angle between the targeting point and the current position of the vehicle. The vehicle under control of autopilot tries to follow the guidance commands.

After the orientation of the vehicle coincides with the direction of the path from the current position to the next way-point, in order minimize the amount of time to reach the next way-point, the speed of the vehicle is increased by 20 percent. Note that this state of the algorithm will be enabled only if the yaw command of the guidance mission is achieved. In the first instants of the movement towards the next way-point, or under the command of obstacle avoidance, the orientation will be lost.

On its way to the targeting point, after the vehicle gets into a radius around the targeting point, the way-point is switched to the next one. This is the third and final state of the guidance algorithm. The radius around the targeting point can be chosen according to the speed of the vehicle and also the distance between the way-points.

The guidance algorithm explained above is presented in Figure 4.1 as a flowchart. The algorithm takes the next way-point as input and the surge speed and yaw position commands as outputs to the autopilot.

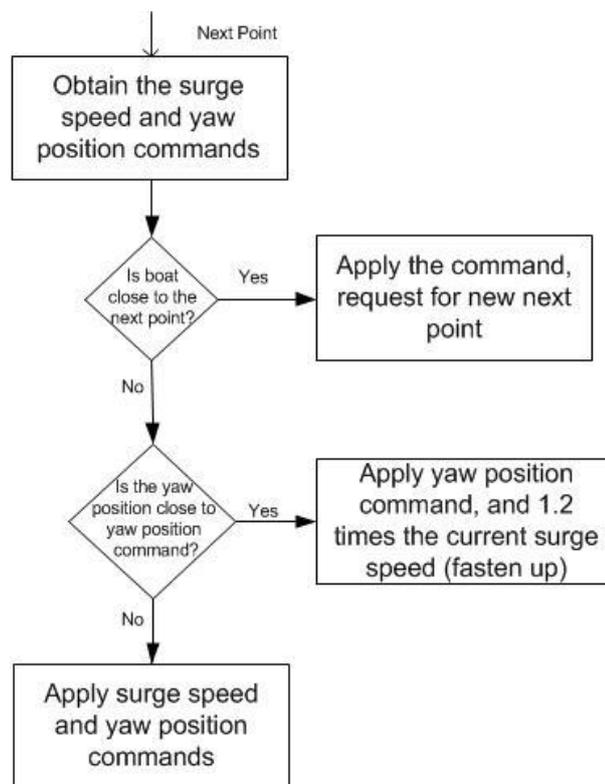


Figure 4.1. Guidance algorithm flowchart

Below, in Figure 4.2, an implementation of the guidance algorithm is illustrated. As being observed, in summary, the algorithm consists of three steps: get the orientation and do it at a lower speed, increase speed to reach to the next way-point, and turn to the next way-point if the vehicle is within a certain radius around it.

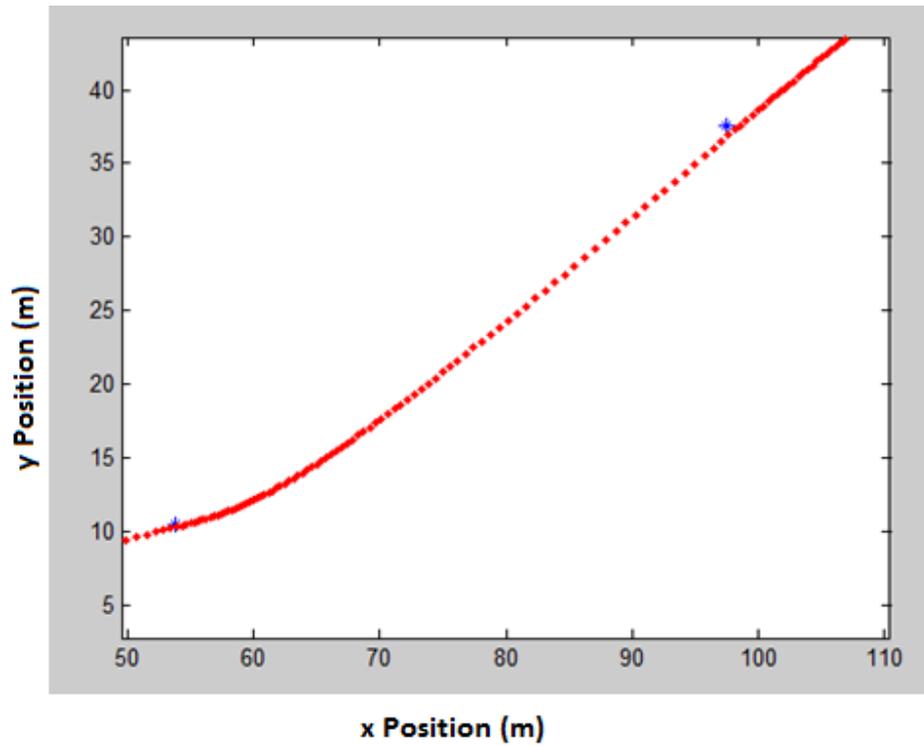


Figure 4.2. Simulation results for guidance algorithm

4.3 Obstacle Avoidance Algorithm

In this subsection, obstacle avoidance algorithm will be investigated. The obstacle avoidance algorithm aims to avoid the obstacles with as less deflection from the guided path as possible. The algorithm operates between the guidance algorithm and autopilot, in the sampling frequency of the autopilot.

The obstacles, in simulations, are selected in circular shape with different radii on the path of the way-points. The vehicle is assumed to have 4 On/Off obstacle sensors looking

towards $-90/45$, $-45/0$, $0/45$, and $45/90$ degrees of current orientation of the vehicle over a certain range.

The way-points to be followed by the vehicle should be determined in such a way that the obstacles are not placed too close to a way-point. The radius of the circle around a way-point, and the sensor range should be taken into consideration while selecting the way-points in order not to miss them. Otherwise, the way-points can be missed, and the trajectory of the vehicle may result in an undesirable route.

When an obstacle is encountered, the vehicle is supposed to avoid it by rotating around the obstacle. Note that the algorithm cannot just change its direction in order to avoid the obstacle; since it has to follow the path towards to the way-point.

If an obstacle is sensed within 45 degrees of direction of motion; i.e., if the second or third sensors alarms of an obstacle, the vehicle changes its direction to avoid the obstacle. This motion is followed until the signals of the middle sensors drop to zero. Of course, the sensors at the edges, i.e., the first and the forth ones, will start to sense the obstacle. This is because, although not in the direction of motion, the obstacle is still there in the range of the sensors.

Under this condition, the obstacle avoidance algorithm turns the vehicle towards to obstacle. Hence the algorithm, in total, gets away of the obstacle if it is in the range and in the certain neighborhood of the direction of motion, and gets closer to the obstacle if it is in the range but out of the certain neighborhood of the direction of motion. With this behavior of the algorithm, the vehicle rotates around the obstacle.

The rotation about the obstacle is needed to be ended, and the motion towards the way-point should be reactivated, eventually. The motion towards the way-point will be initiated if one of the two conditions is satisfied. First, if the guidance algorithm and the obstacle avoidance algorithm command are in opposite directions, and there is no obstacle in the neighborhood of motion of the vehicle. This condition means that, if the edge sensors are sensing the obstacle, the vehicle will not be commanded towards the obstacle under the control of the obstacle avoidance algorithm, but instead, it chooses to turn around towards the way-point, the orientation opposite the obstacle and ahead to the way-point.

The second condition is as follows: if the current direction of the vehicle is towards a certain neighborhood of the direction ahead to the way-point, and the middle sensors are off, the vehicle is oriented towards the way-point. The logic of this second criterion is the same as the first one. The only difference here is that, the motion towards the way-point will not be followed only if the command of the obstacle avoidance algorithm is in the opposite way, but also if the direction of the vehicle is in the close neighborhood of the direction to the way-point. Again, note that the motion will not be oriented towards the way-point if the vehicle is in a close neighborhood of the guided direction but there exists an obstacle ahead of the vehicle, i.e., the middle sensors are alarming.

The obstacle avoidance algorithm not only adjusts the direction of motion but it also, changes the speed of the vehicle. Facing with an obstacle, the current speed of the vehicle may be too fast to avoid it. Hence, the speed of vehicle is fixed at a certain value. This value is the 40% of the speed of the vehicle before it encounters an obstacle. Note that the degradation in the speed of the vehicle is not related to making a turn as in the yaw autopilots. The speed is not decreased in order to turn to a given angle, but so as not to collide with the obstacle.

The algorithm explained in detail above is shown in the flowchart given below. The turning angles are 9 degrees if the obstacle is in front and 10 degrees if the sensors 1 or 4 are alarming. The values are determined empirically.

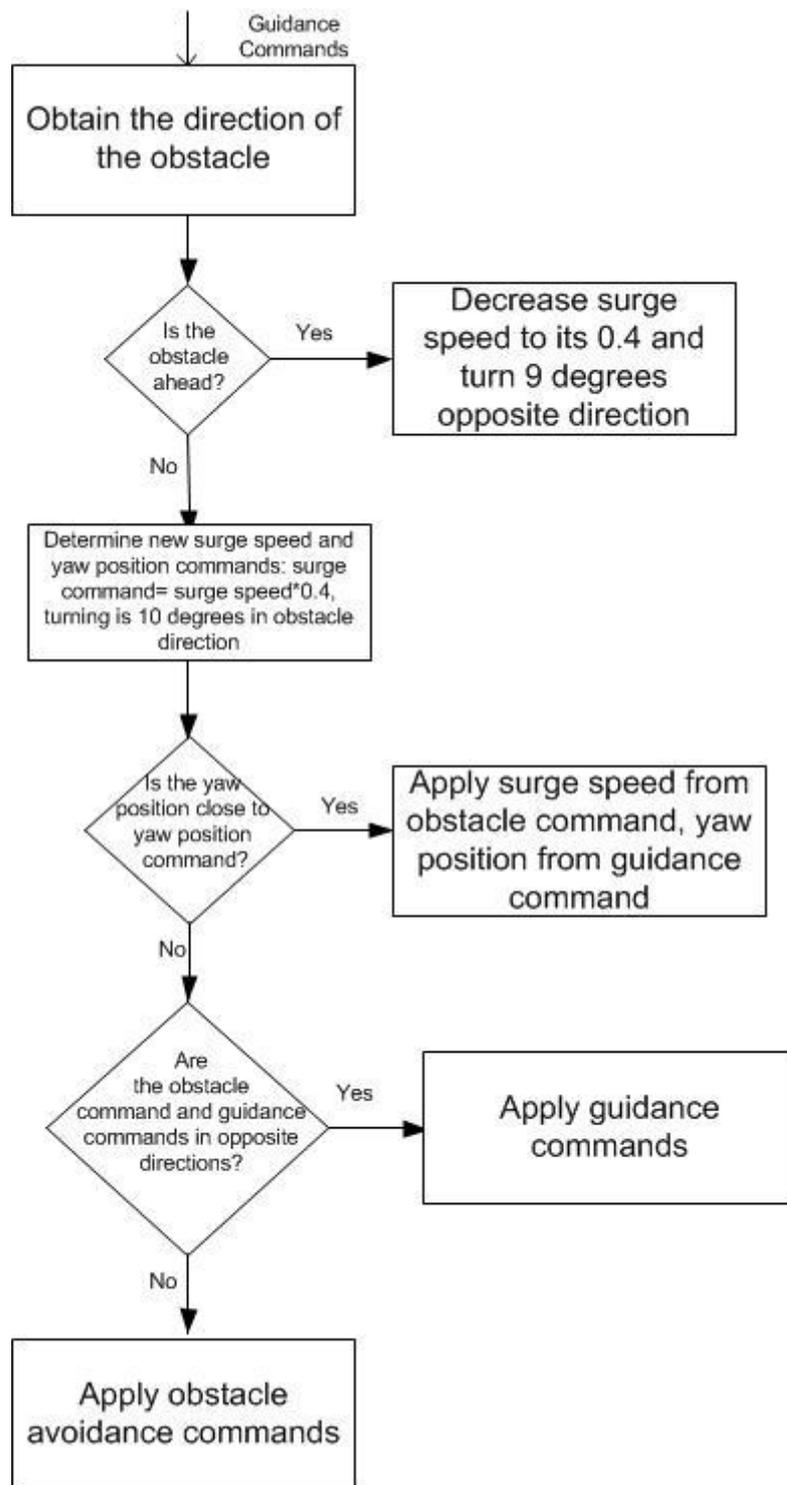


Figure 4.3. Obstacle avoidance algorithm flowchart

In Figure 4.4, a simulation is presented about the behavior of the vehicle encountered with an obstacle. The vehicle gets closer to the first way-point from the previous one with a faster speed. Then the speed is decreased to fix the orientation towards the second way-point. These commands are applied by the guidance algorithm. When the obstacle is sensed, under control of obstacle avoidance algorithm, the vehicle starts to rotate around the obstacle with a slower speed. This motion is followed until the next way-point is oriented in the opposite direction of the obstacle avoidance command. At this point, the guidance algorithm takes the control, and the motion becomes headed to the second way-point, the speed increases.

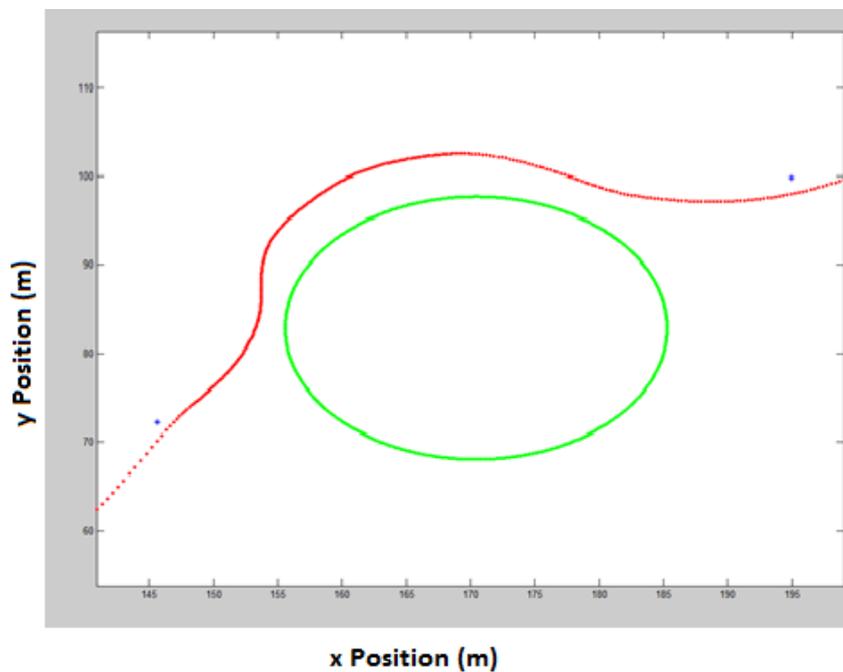


Figure 4.4. Obstacle avoidance algorithm simulation-1

In Figure 4.5, a second simulation output can be observed. Different from the first one, the motion of the vehicle under the control of the obstacle avoidance algorithm is more depictive. The vehicle passes through the first way-point with a faster speed, and orients itself towards the next way-point by degrading its speed and turning, under commands from the guidance algorithm. Then the obstacle is sensed, the vehicle slows down, but the existing behavior continues for a certain amount of time, and the vehicle gets closer to the

obstacle. Then, under control of the obstacle avoidance algorithm, the vehicle starts its motion around the obstacle. Now, note that due to its original speed, the radius of the rotation is different this time. If the vehicle is left to rotate around the obstacle only, this radius will be the sensor range. Different from the first simulation, in this time, the vehicle heads towards to the next way-point, when it satisfies the second criterion. The orientation of the vehicle becomes close the direction of the motion towards the second way-point.

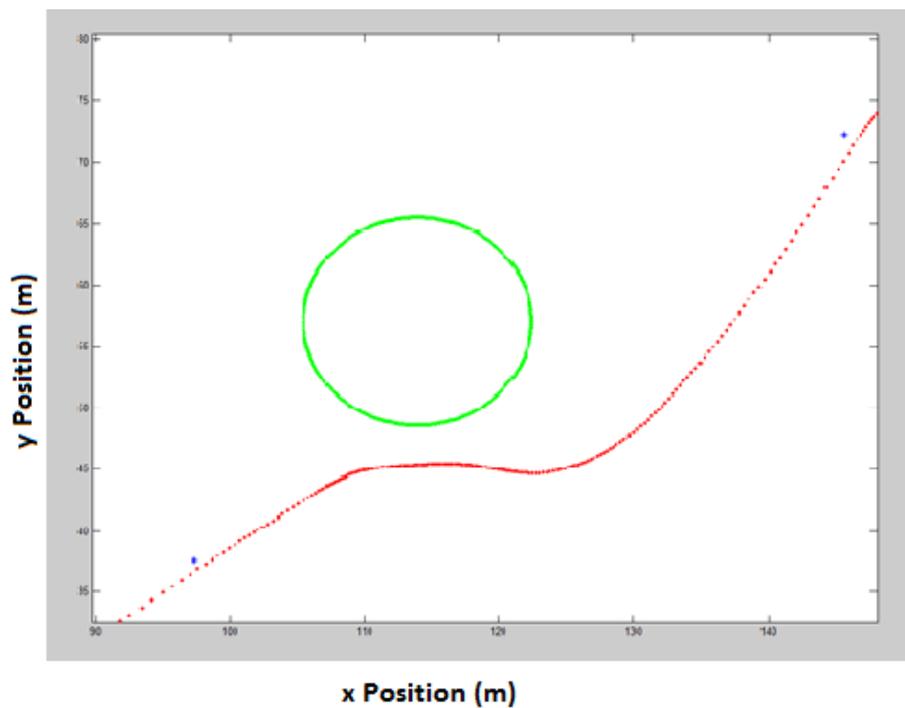


Figure 4.5. Obstacle avoidance algorithm simulation-2

4.4 Next-Point Generation Algorithm

In its trajectory following mission, unmanned sea surface vehicles are guided to way-points among which obstacles exist. While avoiding the obstacle, the vehicle may get too far from the next way-point such that it will be more logical to move towards another upcoming way-point, rather than turning back to the next one. In this section, an algorithm developed to resolve this problem is investigated.

When the current orientation of the vehicle deflects from the direct path towards the next point by a certain angle, the algorithm is activated. This certain angle is chosen as 45 degrees. In other words, unless the vehicle is positioned 45 degrees scattered from the next way-point, it will be guided to that point.

When the algorithm is activated, the angles between orientation of the vehicle and the orientation of the directions towards to each other way-points are calculated. The distances of the way-points to the vehicle are also measured. The angles and the distances are normalized with π and 10 m, respectively. The normalized values are used in the cost functional.

As the next point, the way-point minimizing the cost is selected. The flowchart of the algorithm mentioned above is shown in Figure 4.6.

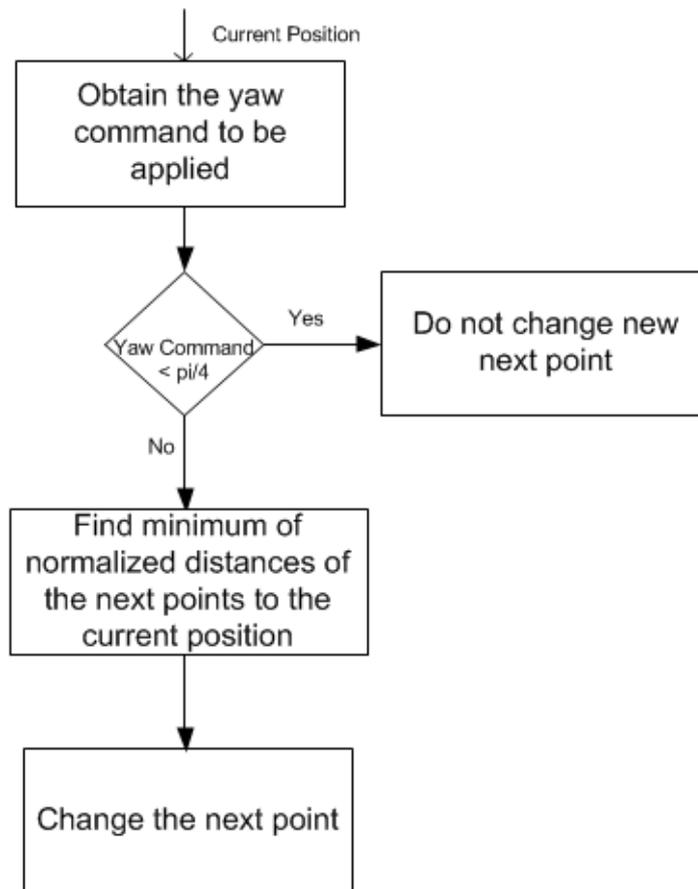


Figure 4.6. Next point generation algorithm

4.5 Effects of Disturbances

Up to this point, we have developed a mathematical model for the sea surface vehicle, PID and LQR controllers for the mathematical model, yaw autopilot over these controllers, guidance algorithms and obstacle avoidance logic. In all the design steps, we have taken the vehicle in an environment with no disturbance.

There are several disturbances that may affect the control of the vehicle. These are winds, currents, waves and other disturbances. The waves and the other disturbance are not modeled in the system. Their effect will be taken into consideration in the implementation part eventually.

Below, effects of disturbances in various directions are discussed. By the term single disturbance, it is meant that the disturbance is in either one of $+x$, $-x$, $+y$, $-y$ directions. By term cross disturbance, it is meant that, the disturbance is in either one of $+x$ and $+y$, $+x$ and $-y$, $-x$ and $+y$, $-x$ and $-y$ directions. The results in figures below have the colors presented in the table, as presented.

Table 4.1. Yaw position-surge velocity limits for the yaw autopilot

Color	Single Disturbance Direction	Cross Disturbance Direction
Red	No disturbance	No disturbance
Cyan	$+x$	$+x$ and $+y$
Magenta	$-x$	$-x$ and $+y$
Yellow	$+y$	$+x$ and $+y$
Black	$-y$	$-x$ and $-y$

The effects of wind contribute to the system with the square of the wind velocity multiplied with a constant related to the shape of the cross-sectional area, and the cross-sectional area itself. In the simulations, the effects of wind on the guidance mission are observed.

For the simulations wind power of 3, which corresponds to 12 Knots, i.e., 20 km/h, i.e., 5.56 m/s wind is applied to the system. This value is found by trial and error. In other words, wind disturbances greater than 12 Knots are found to deviate the vehicle from its path and make it collapse with an obstacle.

The wind is applied from +x, -x, +y, and -y directions. Together with no wind results, the outcomes are shown in the figure below.

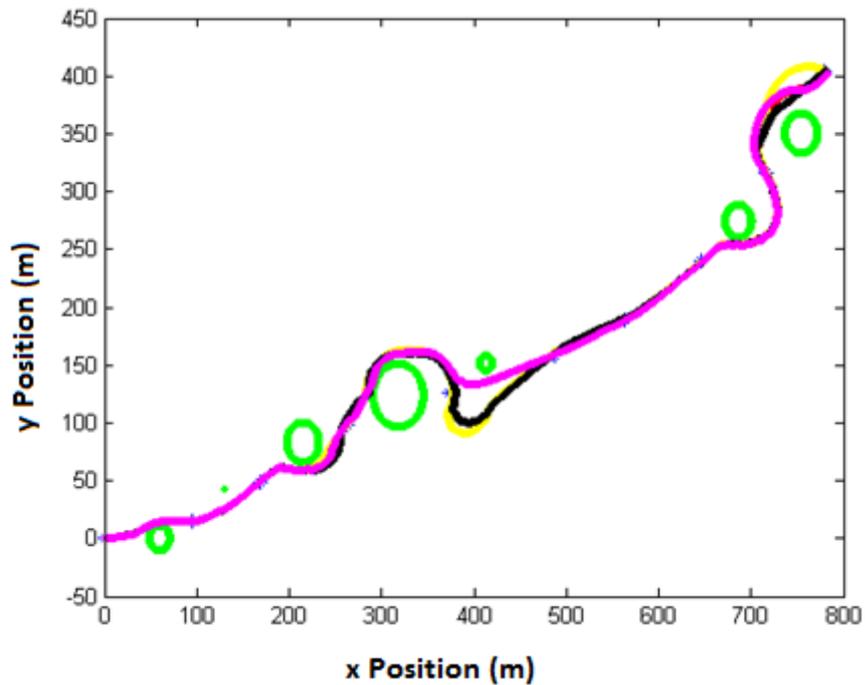


Figure 4.7. Single direction wind disturbed guidance system

In Figure 4.7, single direction wind disturbances applied guidance system outcomes are presented. In the figure, though it is not perfectly clear, blue points are guidance points, green circles are the obstacles to be avoided, red line is the trajectory of the vehicle under no disturbance, cyan line is the trajectory of the vehicle with wind disturbance in direction of +x in order of 5.56 m/s, magenta line is the trajectory of the vehicle with wind disturbance in direction of -x in order of 5.56 m/s, yellow line is the trajectory of the vehicle with wind

disturbance in direction of +y in order of 5.56 m/s, and black line is the trajectory of the vehicle with wind disturbance in direction of -y in order of 5.56 m/s.

From the figure, we observe that no disturbance and +/- x direction disturbances coincide on each other almost completely, since the cross-sectional area is limited compared to other side and the LQR controller has a main objective of controlling the vehicle speed in x direction. The disturbances in y directions have considerable effects on trajectory. This result is rooted in the wider cross-sectional area compared to the x direction, and the lighter control constraints on y direction. Note that position and velocity of the vehicle in y direction, i.e., sway direction is controlled indirectly through yaw commands on rudder.

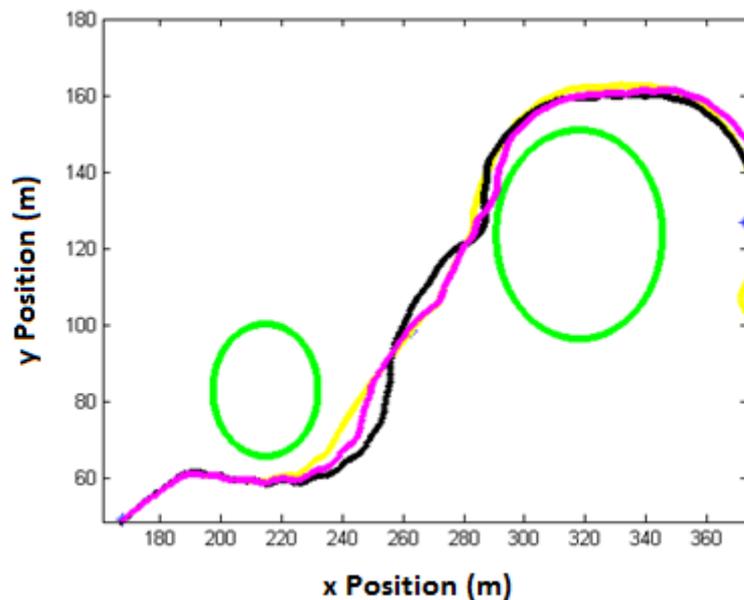


Figure 4.8. Zoom in of single direction wind disturbances in Figure 4.7 -1

In Figure 4.8, we can see a zoom in the trajectories obtained in Figure 4.7, between the third and the fourth obstacles. Remember that yellow line has disturbance on +y direction and black one has disturbance on -y direction. Accordingly the directions of the disturbances, the trajectories deflects from the original line, magenta in the figure, and after crossing one of the targeting point, their deflections continues their effect until they come in a horizontal line again.

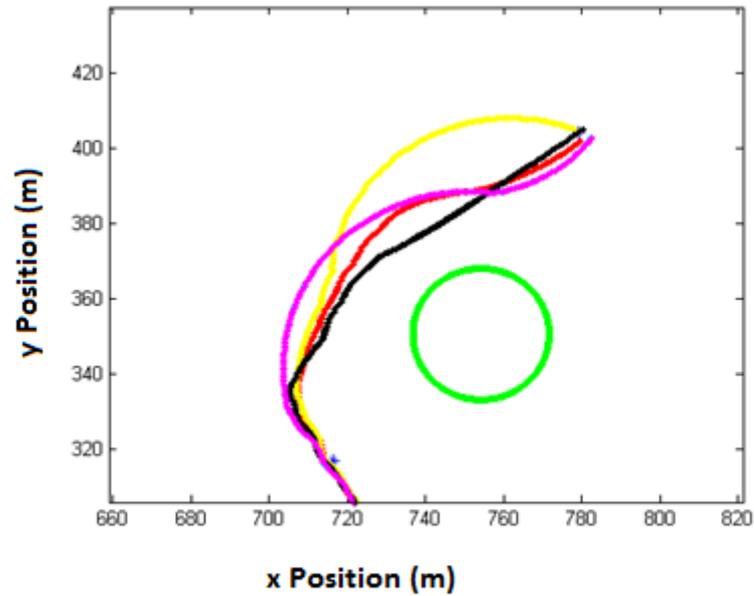


Figure 4.9. Zoom in of single direction wind disturbances in Figure 4.7 -2

In Figure 4.9, we can see a zoom in the trajectories obtained in Figure 4.7, between the last two targeting points and the last obstacle. Remember that yellow line has disturbance on +y direction and black one has disturbance on $-y$ direction, magenta line one has disturbance on $-x$ direction, which completely coincides with $+x$ direction disturbance and the red line is the original trajectory. Again we can see the deflections in y directions according to the disturbances, and the smoothing effect of wind in x direction generates a slow sweep turning around the obstacle.

It is observed that disturbances with cross terms have the same behavior with the disturbance on y-directions. The wind factor on x-direction has negligible effects on the motion of the vehicle. The simulation results are shown in figure below and they are too close to those in Figure 4.7.

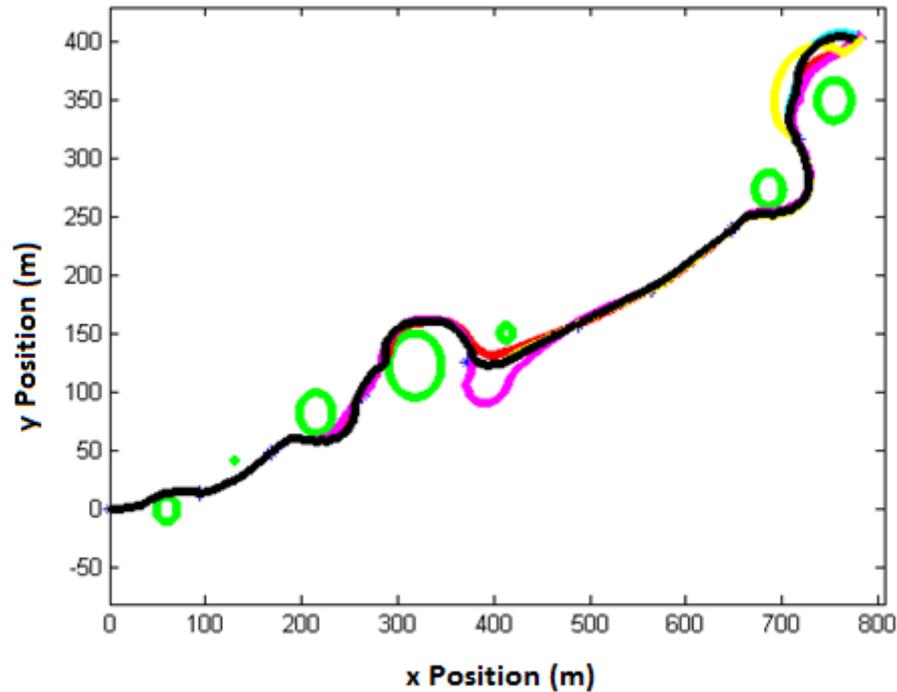


Figure 4.10. Cross wind disturbed guidance system

In Figure 4.10, blue points are guidance points, green circles are the obstacles to be avoided, red line is the trajectory of the vehicle under no disturbance, cyan line is the trajectory of the vehicle with wind disturbance in directions of +x and +y in order of 5.56 m/s each, magenta line is the trajectory of the vehicle with wind disturbance in directions of -x and +y in order of 5.56 m/s each, yellow line is the trajectory of the vehicle with wind disturbance in directions of +x and -y in order of 5.56 m/s each, and black line is the trajectory of the vehicle with wind disturbance in directions of -x and -y in order of 5.56 m/s each.

Another modeled disturbance for the system is the sea current flows. The current flow enters to the system as direct contribution to the velocity of the vehicle. Furthermore, in its motion, the vehicle does job against the motion of current proportional to the acceleration of the vehicle.

In the simulation, water current of 1 Knot; i.e., 0.51 m/s is taken as disturbance. In Bosphorus typical current flow is 3 Knots. Similar to what is done in wind disturbance; water current disturbance effects are investigated both in separate single directions and in coupled terms.

It is observed that in disturbances on +/- y directions, these water current flow results instability that the vehicle collides with one of the obstacles. Hence the water current applied as disturbance on +/- y directions is taken as 0.5 Knots. Again this value is determined by trial and error.

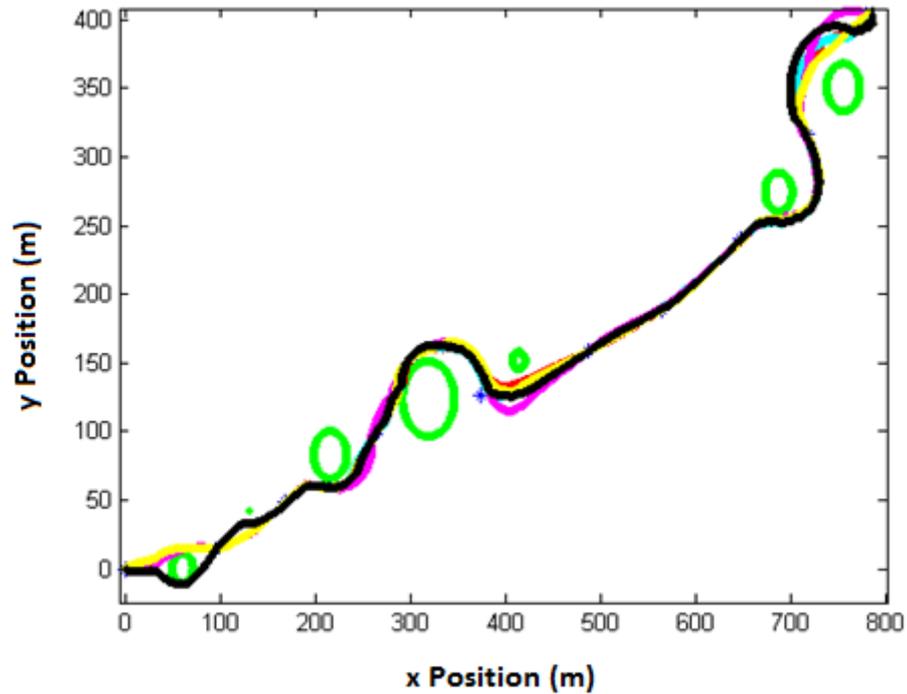


Figure 4.11. Single direction current disturbed guidance system

In Figure 4.11, single direction current disturbances applied guidance system outcomes are presented. In the figure, blue points are guidance points, green circles are the obstacles to be avoided, red line is the trajectory of the vehicle under no disturbance, cyan line is the trajectory of the vehicle with current disturbance in direction of +x in order of 0.51 m/s, magenta line is the trajectory of the vehicle with current disturbance in direction of -x in order of 0.51 m/s, yellow line is the trajectory of the vehicle with current disturbance in direction of +y in order of 0.25 m/s, and black line is the trajectory of the vehicle with current disturbance in direction of -y in order of 0.25 m/s.

It is observed in the figure that $-y$ direction disturbance resulted crossing the first obstacle from bottom side of it, while the original trajectory and the other disturbed ones prefers the top side.

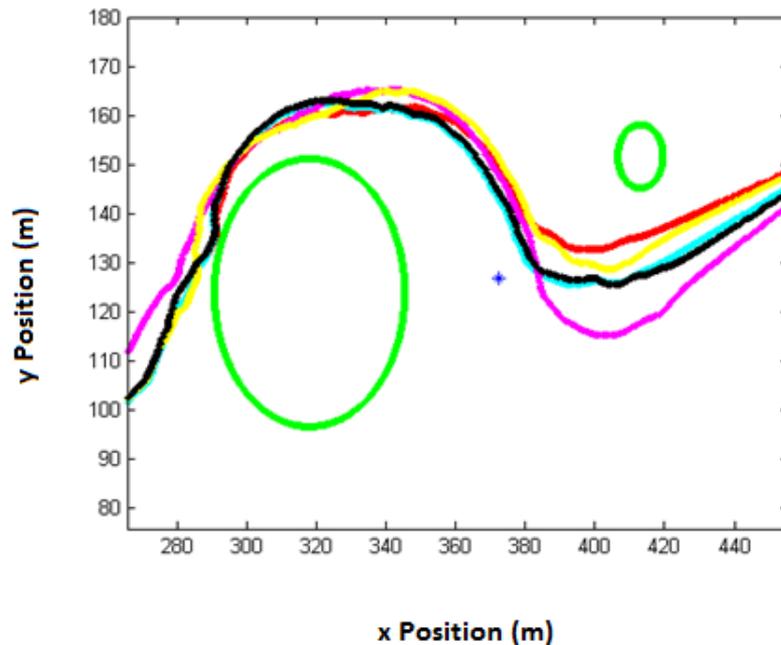


Figure 4.12. Zoom in of single direction current disturbances in Figure 4.11

In Figure 4.12, a zoom in view of Figure 4.11 around fourth and fifth obstacles is depicted. It is observed that since $-x$ direction disturbance starts from a point backwards from the other lines, it turns around to the targeting point with a greater radius of trajectory. All the disturbances caused greater radiuses compared to the original trajectory. It worth noting that disturbances in x directions made no effect on getting closer to the forth obstacle, i.e., the original line together with the disturbed trajectories in y directions follow a closer path to the obstacle.

For cross current effects, the current speed is degraded in all directions to 0.25 m/s. The simulation results are provided in the figure below. However, it is observed that, in the figure, the trajectory shown with the black line, the motion with disturbance in $-x$ and $-y$ direction collides with the fourth obstacle. Hence, we can conclude that, system may not

overcome such a disturbance. In the figure, red line is the trajectory of the vehicle under no disturbance, cyan line is the trajectory of the vehicle with current disturbance in direction of +x and +y in order of 0.25 m/s each, magenta line is the trajectory of the vehicle with current disturbance in direction of -x and +y in order of 0.25 m/s each and yellow line is the trajectory of the vehicle with current disturbance in direction of +x and -y in order of 0.25 m/s each.

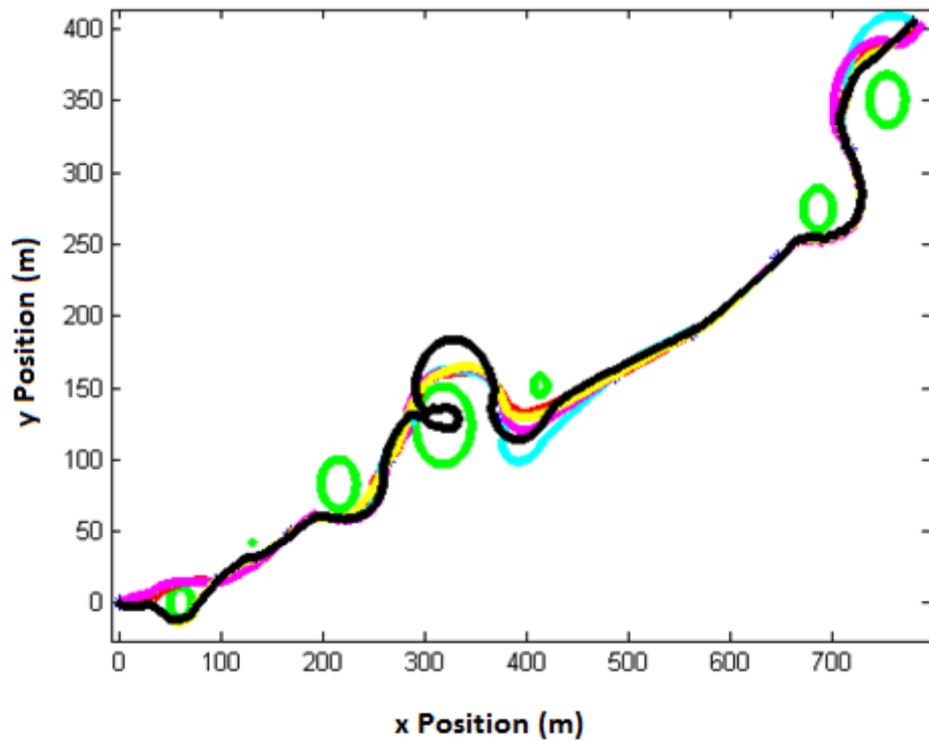


Figure 4.13. Cross current disturbed guidance system

From the results shown in Figure 4.13, analysis of further combinations of the disturbances is skipped. There may be unstable guidance results for both wind and water current disturbed systems. It is observed that, as expected, the disturbance as water current has more significant effect on the motion of the vehicle compared to disturbances due to wind. Yet the system can tolerate wind and current disturbances up to 12 Knots and 0.5 Knots, respectively, which can be sufficient for many cases.

The overall results shows us, up to a certain amount of disturbances the autopilot algorithms are able to maintain the course. However, there are cases where this could not have been achieved. For these cases, new autopilots designed for systems with disturbances should be implemented.

4.6 Conclusion

In this chapter, guidance algorithm developed for unmanned sea surface vehicle has been explained. The guidance algorithm has an aim to follow the given way-points, i.e., it is a way-point guidance method.

Guidance algorithm creates surge speed and yaw position references to the autopilots designed in Chapter 3. Given upcoming two way-points, the algorithm calculates the heading and the speed of the vehicle. After getting closer to the heading angle, the algorithm increases its speed to reach the target faster. When getting closer to the way-point, the algorithm requests for the next way-point.

Between guidance and autopilot blocks, there exists an obstacle avoidance structure. Avoiding obstacles are very important for safe motion on a naval mission. When an obstacle is sensed ahead, the obstacle avoidance algorithm takes control of the autopilot from the guidance block. If the vehicle is just moving towards the obstacle, the speed is decreased and the heading will be changed. If the obstacle is on its way, but it is between the way-point and the current position of the vessel, the algorithm directs the vehicle towards the obstacle until it becomes enfaced with the obstacle. By doing this, the vehicle can rotate around the obstacle. When the obstacle is not between the way-point and the current position of the vessel, it is ignored.

In its motion to avoid the obstacle or under the effects of disturbances, the vehicle may miss the way-point it is heading to. For these cases, a next-point generation algorithm has been implemented above the guidance algorithm. When the angle between heading and the direct line from the position of the vessel to the way-point becomes greater than $\pi/4$, the algorithm solves an optimization problem to minimize the cost assigned for the distance and the heading angle requirements for each of the other way-points.

At the end of this chapter, effects of wind and current disturbances on the guidance algorithm have been investigated. First disturbances on each of single directions, and then their mixture were applied to them vehicle. It is observed that the guidance algorithm may tolerate disturbances up to 12 Knots wind and 0.5 Knots current. These values are the upper limits of the endurance of the model on disturbances, and they are found by trial and error. Multidirectional disturbances are more effective on the vehicle.

CHAPTER 5

EXPERIMENTAL WORK

5.1 Introduction

In this thesis, so far, mathematical modeling of unmanned sea surface vehicles is obtained, PID and LQR controllers are designed for this mathematical model, and yaw autopilots are implemented. Based on the implemented autopilots, guidance and obstacle avoidance algorithms are applied. The simulations are validated under environment with wind and water current disturbances as well.

The simulations are verified, up to a certain extent, in this study on a model boat as the unmanned sea surface vehicle. With this experimental work, controller algorithms are emulated on a system in an environment containing both modeled and un-modeled disturbances.

In this study, on Traxxas Spartan Model 5707 boat, PID controllers for surge speed and yaw position reference tracking are implemented. The controllers are inserted into ArduPilot Mega v1.4 card with Olipan IMU Shield, and autonomously controlled the motion of the vehicle.

The controller cards are mounted inside the boat, communicated with a PC via wireless communication through XBee channels. The position and orientation as well as linear and angular speeds are attained from the system and observed on PC. The surge speed and yaw position commands, PID parameters for surge speed and yaw position are transferred to the system via the same port.

The inputs to the system and the outputs from the system are evaluated in MATLAB/SIMULINK 'External Mode' and can be accessed with a MATLAB GUI.

The parameters of the controllers should be determined according to the parameters of the system, in fact, in an environment of cumulative disturbance. The parameters of the system are to be determined with system identification algorithms. However, for the sake of simplicity this step is skipped, and the parameters are auto-tuned again, with the help of GUI.

Only implemented algorithms to the boat are the controller algorithms. Further steps in guidance and obstacle avoidance may be inserted on another controller unit getting commands from the user and sending them to the autopilot card, and getting feedback from the autopilot and sending them to the user, as a further study; that is also kept out of the scope of this thesis.

In this chapter, experimental setup and the experimental results are demonstrated in the following subsections, in turn. In experimental setup subsection, the boat, the autopilot card with IMU shield, and the MATLAB GUI are explained. In the experimental results subsection, the outputs obtained from the autonomous sea surface vehicle under PID controller are presented.

5.2 Experimental Setup

In this subsection, experimental setup used in this thesis will be explained. First the unmanned sea surface vehicle, Traxxas Spartan Model 5707 model boat will be introduced. The properties of the boat, the components inside, and the capabilities will be explained in detail. In the second part, ArduPilot Mega v1.4 autopilot card will be demonstrated. Available extensions and the features of the card will be shown.

Following the autopilot controller card, an IMU shield used, Oilpan IMUShield vH will be examined. The wireless communication is achieved with XBee wireless modules and these modules are to be covered.

The communication with the unmanned sea surface vehicle is interfaced to the user with a MATLAB GUI. This GUI will be explained as well. The external mode SIMULINK model for the communication will be shown.

In the thesis, as the experimental environment, Traxxas Spartan Model 5707 model boat is employed. An outer view of the boat is shown in figure below.



Figure 5.1. Outer view of Traxxas Spartan model 5707 boat [44]

This boat has weight of 2.041 kg, and dimensions of 1.037 m in length, 0.24 m in width and 0.147 m in height. The boat has hull length of 0.927 m [44].

The boat can reach up to 13.4 m/s forward speed. The shape of the boat gets sharper towards to frontier end in order to decrease friction due to forward speed. It also includes stabilizing metal planes for roll and pitch stabilization [44].

In Figure 5.2, a view from inside of the boat is provided.



Figure 5.2. A view from inside of the boat [44]

In the figure above, the components are as follows. On the far left, the rudder (steering) RC servo, RC 2506, is placed. Next to it, the receiver, through which the boat communicates to its remote controller, takes places. Next to the receiver, the electronic speed controller (ESC) VXL-6S can be seen.

The power to the rudder servo is provided by the receiver. The receiver gets its power from ESC. ESC pulls power the batteries placed on sides of the boat. The batteries are connected in series. Each provides 8.4V with 3Ah. The power input to the ESC is total of 16.8V and it works under voltages greater than 14V.

ESC regulates the power to 5V to the receiver. The receiver getting commands from the remote controller in user, sends signals to control rudder servo and ESC, and these signals have amplitudes of 5V also.

Both ESC and RC servo has commands in terms of duty cycle of length 1 ms to 2 ms, minimum and maximum in period of 10ms. In autonomous controlling of the boat, this period is kept as 20 ms.

We need the ESC to drive the motor. The power from the receiver alone is not enough to drive the motor for propeller. ESC takes commands through a RC Servo interface and generates necessary phase currents to the propeller motor. The motor in use is 540XL brushless motor.

We can also observe the water cooling mechanism in Figure 5.2. With the motion of the vessel, water enters to the system through a small hole placed on rudder and it exits the system as shown on the far right of the figure. The system cools both ESC and the motor.

The connections inside the system are water-proof.

In the study, the receiver will be discarded from the system and instead of it, the autopilot cards will function. The autopilot cards will get their energy in 5V from the ESC, similar to the receiver, and will provide signals to the receiver and the RC servo necessary to control surge speed and yaw position of the vehicle.

The controllers run on a controller card mounted into the vehicle. This card consists of an autopilot card, a sensor card, some sensors which are not on the sensor card, and finally wireless telecommunication hardware to achieve communication between the user and the vehicle.

The autopilot card used in this study is ArduMega Autopilot v1.4. In Figure 5.3, top view of the autopilot card is depicted.

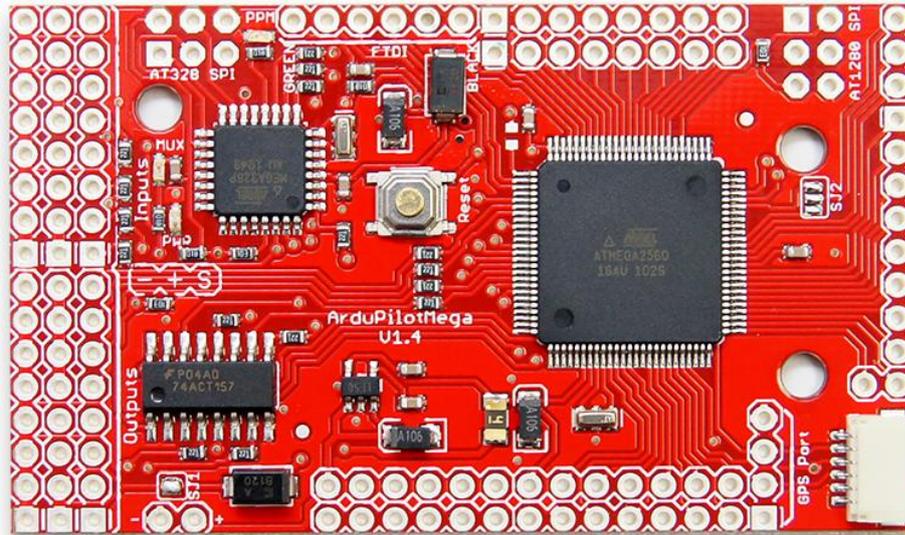


Figure 5.3. ArduPilot Mega v1.4 view [45]

The card contains an ATMEL ATMEGA2560 8bit microcontroller with 256KB in-system programmable flash memory. The microcontroller can perform 135 instructions per clock cycle and has 16 MIPS at 16MHz operation. In addition to its 256KB flash memory it also has 4KB EEPROM and 8KB SRAM. EEPROM and FLASH inside the controller has 100000 and 10000 write/erase cycle lifetimes, respectively [45].

ATMEGA2560 has, 4 16-bit timer/counters, 4 8-bit PWM channels, output compare modulator, 16 channels of 10-bit ADC, 2 SPI interface, on-chip oscillator, and on-chip analog comparator. It also supports power-on reset and programmable brown-out detection, sleep modes, standby modes, and 86 programmable I/O lines. The power consumption of the microcontroller is 500uA at 1.8V when operating at 1MHz [45].

On ArduPilot Mega v1.4, other than ATMEGA2560, there are a reset push-button which resets the operation of the microcontroller, and makes it boot from the code in its internal flash memory, 8 output ports for RC motor controls, 8 input ports, SPI interface, interfaces

to IMU Shield card on which the sensors are mounted, a port for GPS, voltage regulators and buffer integrated circuit.

The autopilot card is the place where we embed our code. The code runs on ATMEGA2560, obtaining feedbacks from the sensors probably connected to the pins of the controller, and produces actuating signals, as PWM for RC Servo motors, through its pins which can reach to its ports.

There is also an expansion port on the autopilot card for mounting GPS. GPS provides us the position of the vehicle with respect to the Earth, when it is in open air.

The major sensors for autopilot feedbacks are placed on a sensor card, namely IMU Shield of Oilpan. In Figure 5.4, below, a top view of the sensor card is depicted.

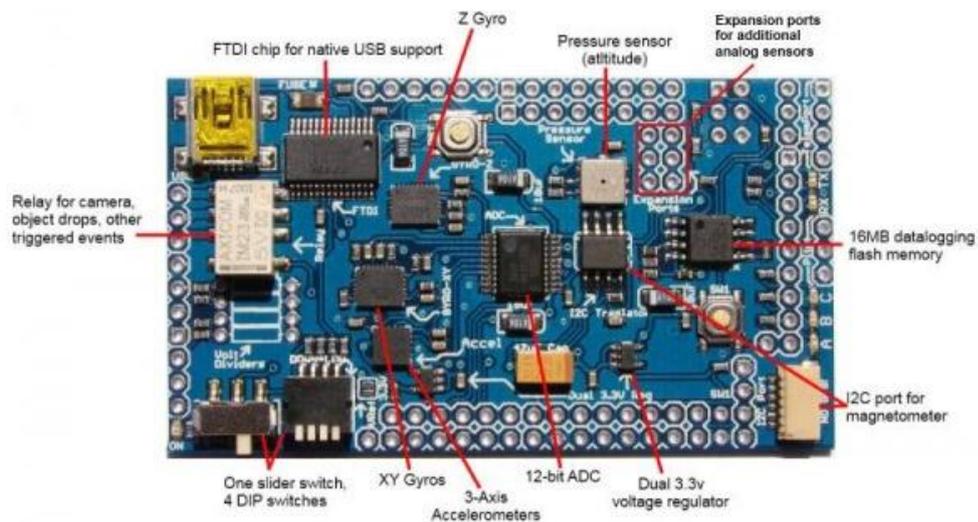


Figure 5.4. Top view of Oilpan IMUShield [45]

On the IMUShield, there exists a port for mini-usb through which the autopilot card is programmed, and FTDI chip for USB support. The card contains 12 bit ADC and pressure sensor which are not in use in this study. The pressure sensor can be used to gather altitude information for unmanned air vehicles. There is 16MB flash memory placed on this IMU shield. The switches enable the controller to power from ESC.

The most important components on this IMU shield are on-board gyroscopes and on-board accelerometers. The gyroscopes provide angular speed information. There are two separate integrated circuits on the board for gyroscopes, one for x and y axes and one for z axis gyroscope information. The accelerometers measure linear acceleration effecting to the board. Note that the effect of gravity is included in the accelerometer data.

There is also an IC2 port and its driver mounted on the board. The I2C port is connected to the magnetometer. Magnetometer gives us orientation of the vehicle in space.

A further expansion on IMU shield is for telemetry connections. Places on top right of the card, we can connect XBee modules to the IMU shield and communicate with outside world and user.

XBee modules are two card with specific IDs that communicate with each other. One of the modules is connected to the extension pins of IMU shield, and the other card is connected to USB port of the PC in use. The communication protocol obeys the standard serial communication rules.

The IMU shield is mounted on top of the autopilot card. The functional block diagram is shown in figure below.

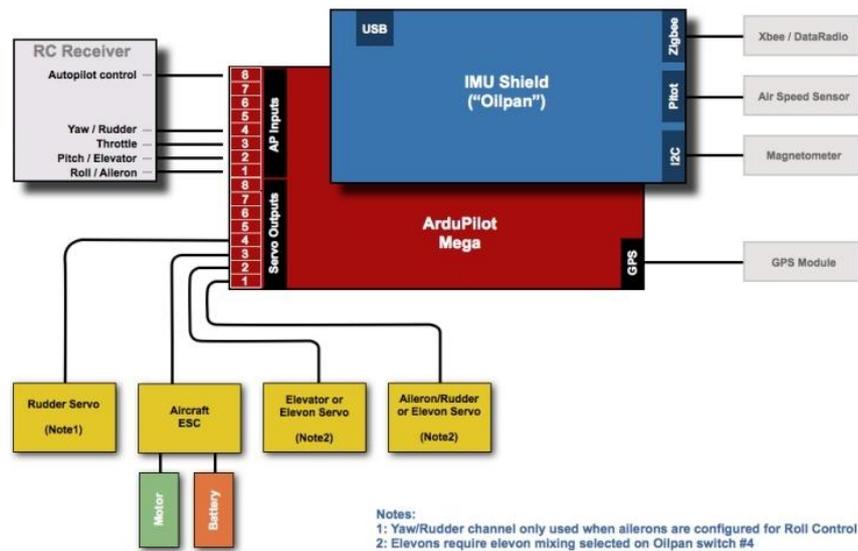


Figure 5.5. Autopilot card together with IMU Shield [45]

In summary, the autopilot card runs the controller algorithm. It gets 3 axes gyroscope and accelerometer information from IMU shield, and 3 axes magnetometer information through IMU Shield, and GPS information from its expansion pins. The feedbacks are received, the code is executed and the outcomes are extracted through the pins on the autopilot card. The third and the fourth servo channels are used. The third channel provides power from ESC to the autopilot card, as well as the IMU shield, the sensors connected to these cards and XBee module. On the same channel, PWM signals for RC servo motor for propeller is provided to the ESC. The rudder servo is controlled with the connections on the fourth channel.

During the experiments, the communication with the vehicle through XBee module is achieved in MATLAB/SIMULINK environment. A MATLAB GUI is prepared for user to enter the controller parameters and commands, and to observe the outputs.

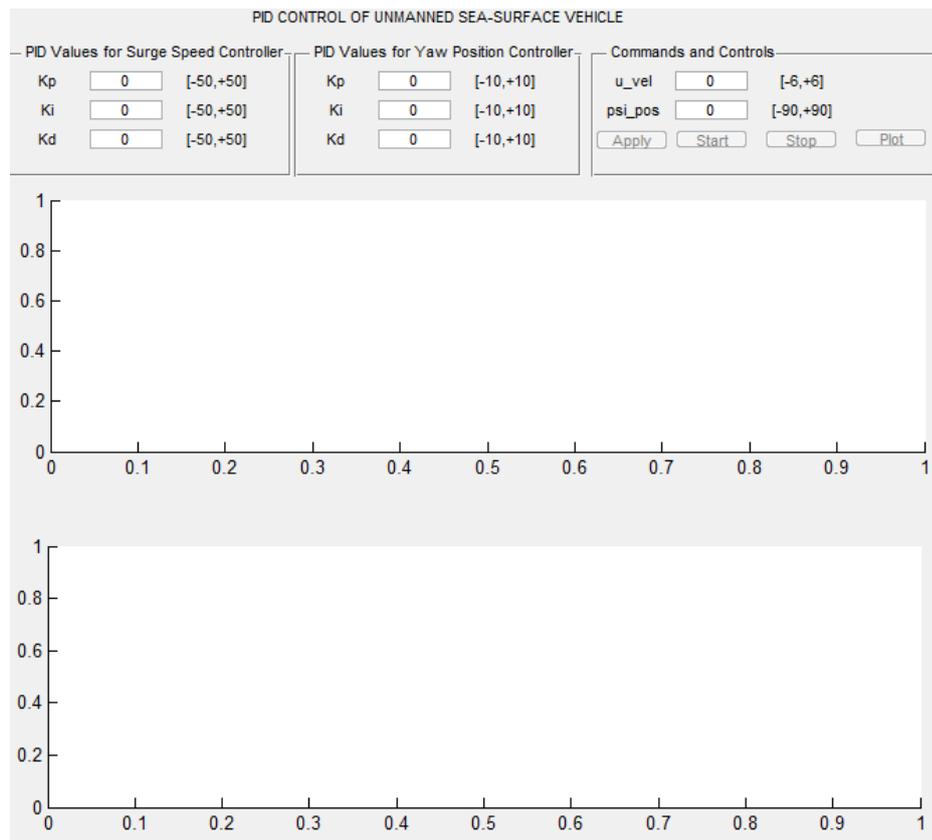


Figure 5.6. MATLAB GUI prepared for experiments

In this GUI, user enters PID parameters for surge speed controller and yaw position controller. Then he enters yaw position and surge velocity commands. If the surge velocity command is greater than 0, then ‘Apply’ button becomes enabled.

If ‘Apply’ button is clicked on, then the communication with the vehicle starts. First of all, a calibration process is initiated on the vehicle. The sensors onto and connected to the ArduPilot Mega and IMU Shield will be calibrated. In this process, drifts of the sensors will be measured. If the drift values are above the limits, then the process fails.

After successful calibration, PID parameters and commands will be sent to the vehicle. The parameters will be stored in EEPROM of ArduMega Pilot autopilot card. Then the parameters will be read back and checked. Until what is read matches the parameters transmitted, this process will go on.

As the calibration and parameter storing finishes, ‘Start’ button will be enabled. If the user clicks on ‘Start’ button, tests will start. During the test, the following SIMULINK model will run on ‘External Mode’ through a COM terminal devoted to XBee.

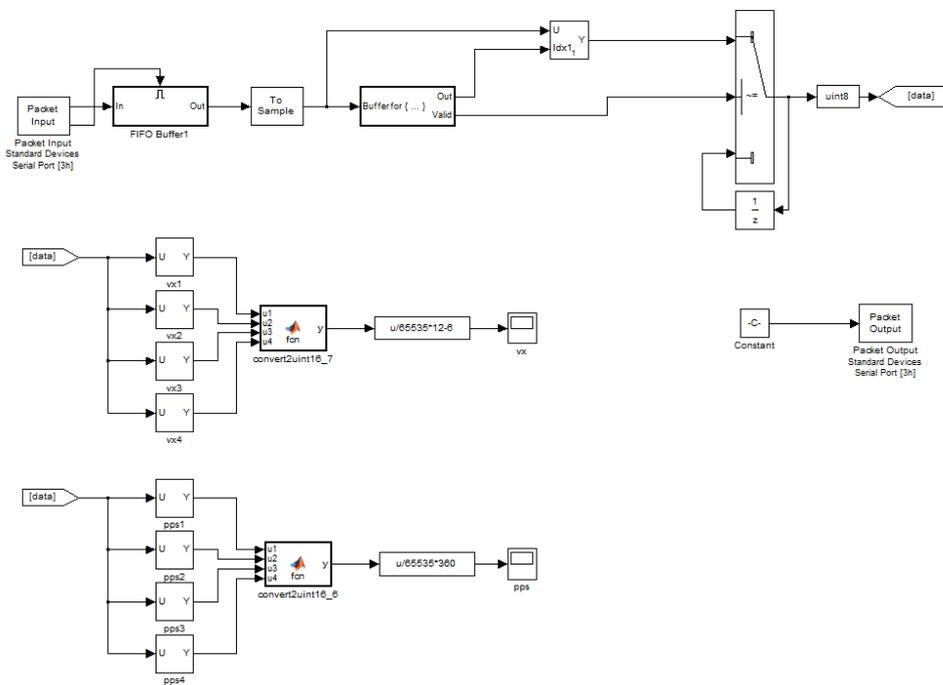


Figure 5.7. SIMULINK External Mode model for tests

During the tests, PC will send a header to the vehicle. With this data transformation the vehicle will run the test and notice that the communication is established and on. The vehicle will return the feedback data for surge speed and yaw position with a header to PC. The SIMULINK model will search for header bytes in the data stream, and the data following the header will be taken and stored. For feedbacks, with limited baud-rate, only surge speed and yaw position data are transferred.

When the user wants to stop the experiment, following the enabling of ‘Start’ button, ‘Stop’ button gets activated and he needs to click on this button. After ‘Stop’ is clicked on, the experiment terminated, and the data obtained are stored. The data can be plotted by clicking on ‘Plot’ button in GUI.

If for any reason, the connection gets lost, the vehicle stops working. The user should click on ‘Stop’ button to initiate the process again.

For the tests, the vehicle is modified. Below, an inside view of the vehicle is shown.



Figure 5.8. Inside view of the vehicle under test

The batteries provided by the company of Traxxas Spartan are 3Ah. The current can be pulled by the RC servo drivers are approximately 3A maximum. Hence, a full and perfect battery of those provided may only work for an hour. In fact, in field tests this value falls to 20 to 40 minutes.

For a better experimental environment, the batteries are changed with more powerful batteries. Inspired Energy NL2024ED 14.4V Lithium Ion Battery is used instead. This battery has a weight of 200gr and it is 7.5Ah. The battery is placed on center of the vehicle in order to prevent distractions from the heading due to unbalanced vehicle dynamics.

On top of the battery a water-proof box is placed. Inside this box, ArduPilot Mega autopilot card, IMU Shield sensor card, GPS module, magnetometer, and XBee module are located. The antenna of the XBee module has its way out of the box through a small water proven hole.

Both the controller box and the battery stabilized inside the boat. The outer cover the boat is closed. Between its cover and the boat, water proof silicone ring is implemented.

5.3 Experimental Results

In this subsection experimental results will be depicted and explained. During the experiments, the PID controller is tuned for an acceptable performance for $u = 1$, $\psi = 0$. Then the parameters are applied for different yaw positions and different speed values.

In the experimental part of this study, over one hundred experiments are conducted. The experiments are done with the model boat modified according to the need of the test as explained above. The experiments are held in Yalincak Gölet, METU with the permission of ASELSAN Inc.

The experiments are realized in open air. Hence, the environment during the experiments has both modeled and un-modeled disturbances, wind, current and waves. Furthermore, during the test, the non-ideal behaviors of the vehicle are also observed, as expected.

In the figures below, experimental results for $u = 1$ with $\psi = 0, 30, -30$ are shown. After tuning, the PID parameters for surge speed controller are $P = 28$, $I = 24$ and $D = 0$, and the PID parameters for yaw position controller are $P = 1$, $I = 0.4$ and $D = 0$.

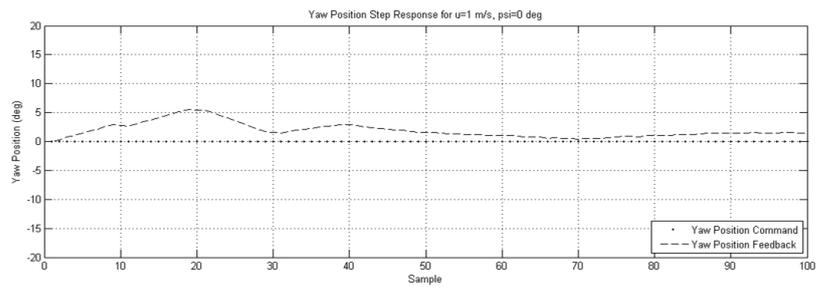
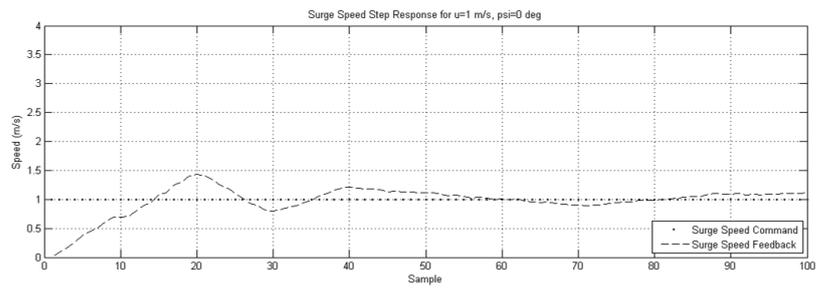


Figure 5.9. Experiment results for $u = 1$ and $\psi = 0$

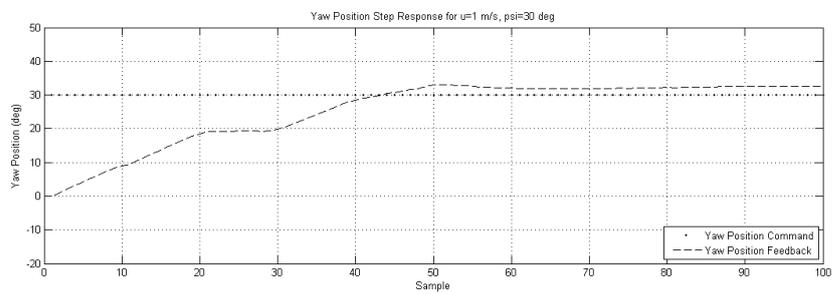
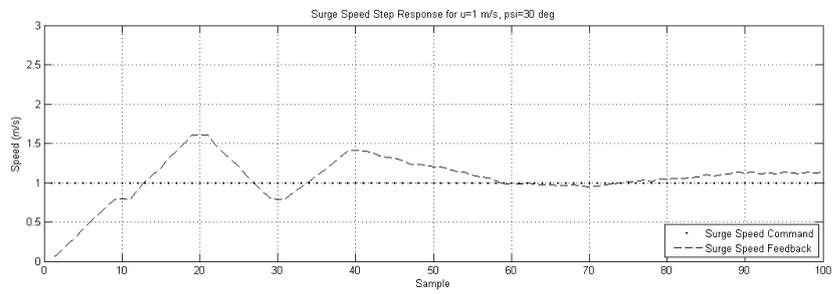


Figure 5.10. Experiment results for $u = 1$ and $\psi = 30$

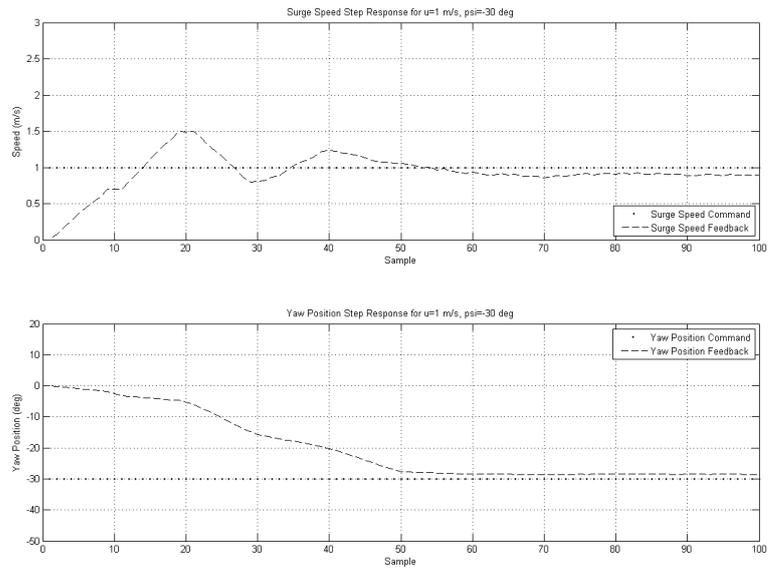


Figure 5.11. Experiment results for $u = 1$ and $\psi = -30$

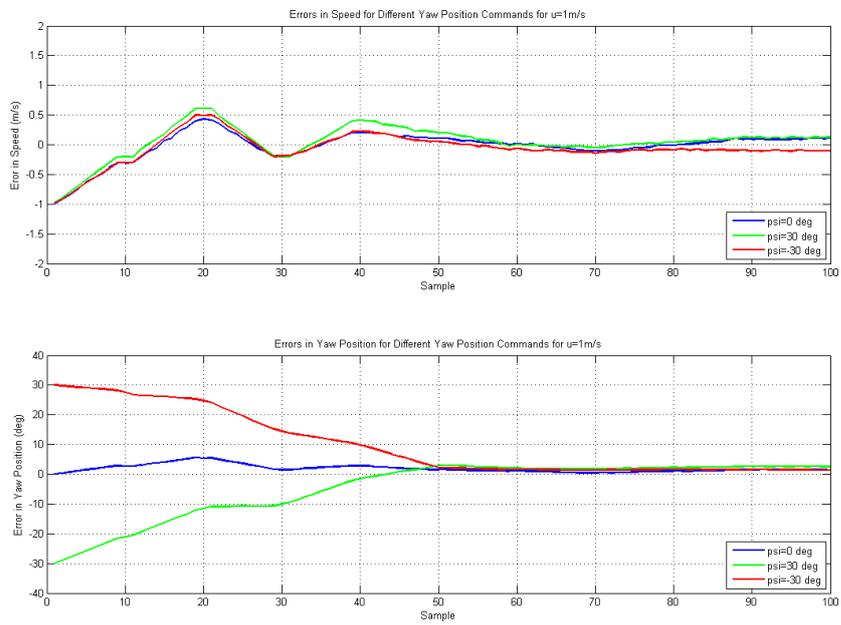


Figure 5.12. Errors between feedbacks and command for $u = 1$ with different yaw position commands

In Figure 5.12, errors in surge speed and yaw position feedbacks and commands for $u = 1$ under different yaw position commands are shown. From the figure, we can see that the most unstable behavior of the vehicle is observed when it is headed to +30 degrees yaw command. This can be seen from the maximum overshoot results for this angle in surge speed and yaw position.

It can also be observed that for yaw position command of -30 degrees, speed value is always smaller compared to 0 and 30 degrees of yaw commands.

For the same PID values, experiments are also conducted for surge speed of 2 m/s. In figures below, the results are demonstrated.

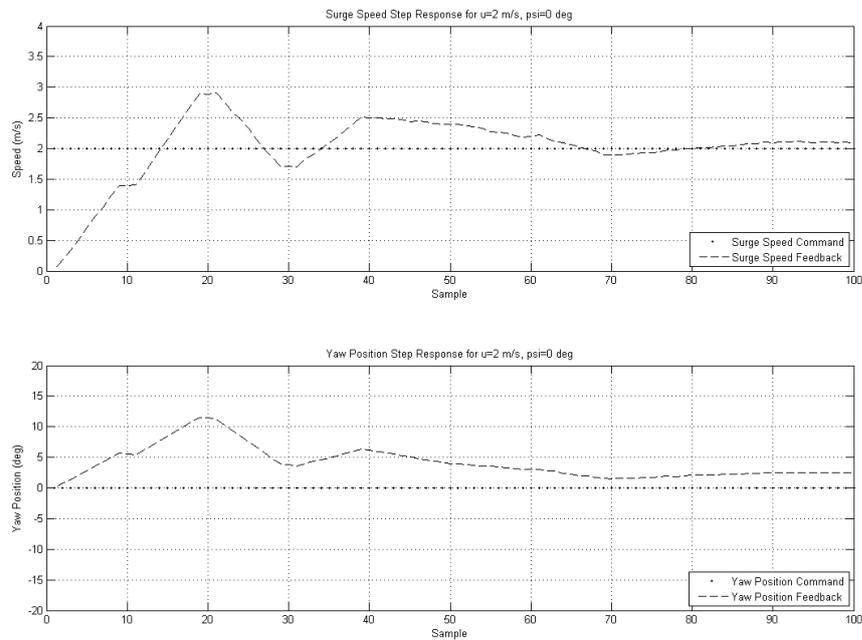


Figure 5.13. Experiment results for $u = 2$ and $\psi = 0$

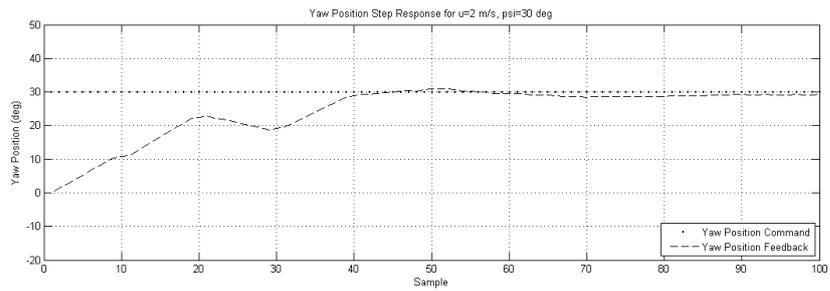
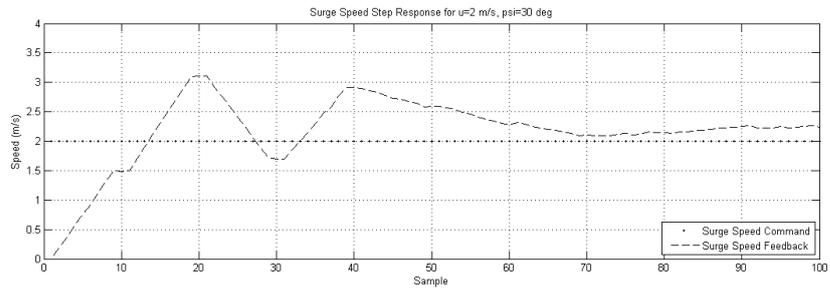


Figure 5.14. Experiment results for $u = 2$ and $\psi = 30$

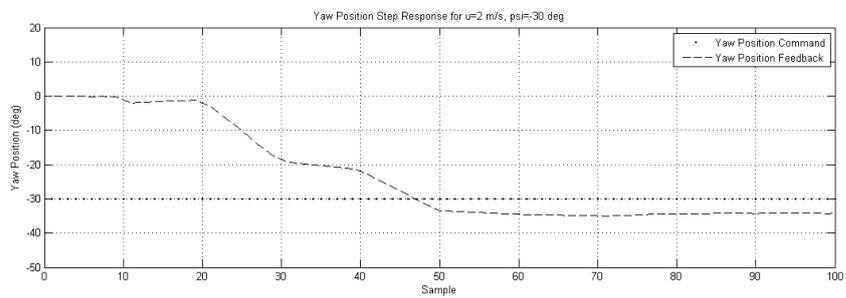
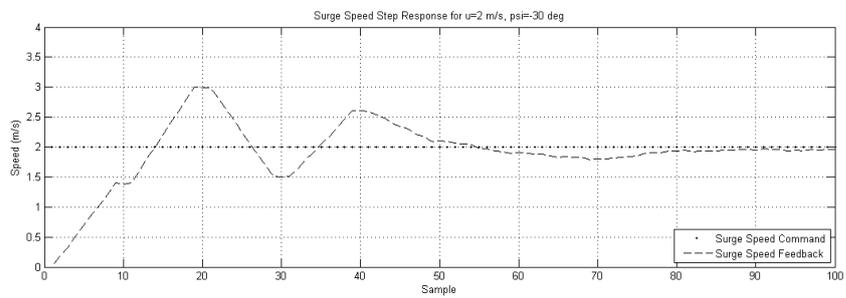


Figure 5.15. Experiment results for $u = 2$ and $\psi = -30$

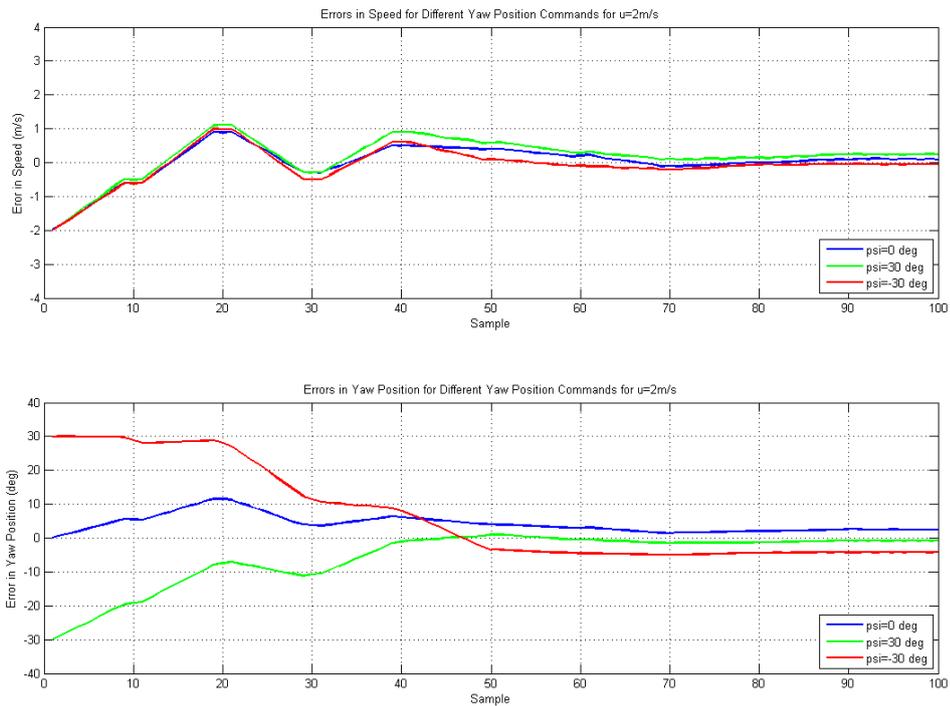


Figure 5.16. Errors between feedbacks and command for $u = 2$ with different yaw position commands

In Figure 5.16, errors in surge speed and yaw position feedbacks and commands for $u = 2$ under different yaw position commands are shown. From the figure, we can see that the most unstable behavior of the vehicle is observed when it is headed to +30 degrees yaw command, again. This can be seen from the maximum overshoot results for this angle in surge speed and yaw position.

It can also be observed that for yaw position command of -30 degrees, speed value is always smaller compared to 0 and 30 degrees of yaw commands. Furthermore, the errors in yaw positions are greater compared to those with surge speed of 1m/s.

In the figures, it is observed that overshoots in surge speed reach to 50%. Note that for different PID parameters for surge speed and yaw position controllers, the results would be different. However, with the tests being conducted over and over, a working and relatively consistent parameters are determined to be the ones mentioned above. With different

parameters though overshoot gets smaller, the response is so slow to be observed in a motion in range of XBee communication.

For better performance, the experimental setup can be improved as well. During its calibration process, noise and disturbance acting on the vehicle changes its behavior during the tests drastically. In order to prevent this, the calibration can be done in a disturbance-free environment inside the shore, and without moving the vehicle a lot, i.e., keeping the initial heading constant, the vehicle should be able to start moving.

The sensors used in this study also lack high level of accuracy. The sensors can be changed with more accurate ones. However, it worth noting that, inside a sea surface vehicle we have a limited space and the weight of the vehicle is of curial importance. For this rather than improving the sensors, maybe other types of sensors can be employed in the experimental setup. For instance, rather than derivation of speed through other sensors, mainly by integrating filtered accelerometer output, a pitotube may be utilized.

The position of rudder on this model boat, is not in the middle or right back of the thruster. This gives a significant error in controlling the yaw position. A better rudder placement may be considered.

Note that, rudder gives the yaw position to the vehicle in a passive way. Rather than a rudder, maybe two thrusters placed on right back and left back of the vehicle should be utilized. By doing this, we may achieve orientation of the vehicle while it is not moving forward.

The effects of disturbances may also be reduced. The aerodynamic structure of the vehicle can be improved. Though it is important to keep the vehicle light for faster speeds, without enough inertia, it is more prone to distract from its course under the effects of disturbances. Moreover, the vehicle should be built, and the components inside it should be placed in such a way that, the center of gravity of the vehicle does not cause instability.

We command the motion of the vehicle with the actuators mounted to it. The powers of the actuators are important to keep the course in spite of disturbances. The limits of actuation may not be enough to overcome every disturbance. Not only is the power of the actuators, but also the resolution in actuation is critical for better performance.

For longer tests, the range of XBee communication should be improved as well. More powerful antenna modules can be used to keep the signal range bigger. With longer tests, we may observe the behavior of the vehicle for a greater time and distance and we will be able to change PID parameters to those presenting slower response with smaller overshoots.

5.4 Conclusion

In this chapter, experimental work conducted in this thesis are explained and demonstrated. As the experimental setup, a Traxxas Spartan model boat is used and a controller in a wireless manner. The communication is achieved with XBee modules.

The controller algorithm is inserted into an ArduPilot Mega autopilot card, and the vehicle has its motion autonomously according to the parameters and commands entered by the user.

The sensors to close the feedback loop are gyroscopes, accelerometers, GPS and magnetometer. The sensors are mainly connected to a sensor card IMU Shield which is mounted on top the autopilot card.

For controlling the parameter and commands to be entered to the vehicle, a MATLAB GUI is designed. The GUI calls a SIMULINK model running in 'External Mode' and obtains the data, saves and plots the results.

The tests are conducted in an open environment; hence, not only the non-ideal characteristics of the vehicle but also the disturbances such as wind, current and waves are to be overcome during the experiments.

In the experiments, mainly 6 tests are done. The tests are to keep surge speed in 1 m/s, and 2m/s and yaw position at 0, -30, +30 degrees, and combinations of them. The tests are done for the PID controller and PID parameters of the controller are selected for surge speed of 1 m/s and yaw position of 0 degree.

The results are presented in figures above. It has been observed that the yaw commands of 30 degrees have less qualifying outcomes. As the speed increases, the errors in yaw position increases as well. It is also observed that -30 degrees yaw position command has more negative offsets during steady state compared to 0 and 30 degrees.

In the figures below, photographs of the vehicle with respect to a stable point in the lake are shown. The vehicle has orientation of 0 degrees, 30 degrees and -30 degrees in the figures, respectively.



Figure 5.17. Initial position of the vehicle



Figure 5.18. Yaw position of 0 degrees, vehicle hits the target



Figure 5.19. Yaw position of 30 degrees, vehicle is headed to 30 degrees



Figure 5.20. Yaw position of -30 degrees, vehicle is headed to -30 degrees

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this study, it is aimed to investigate mathematical behavior of unmanned sea surface vehicles, design controller and guidance algorithms to rule the motion of these vehicles in simulations and experimentally.

In the beginning of the study, a detailed literature survey is conducted. It has been observed that mathematical model of these vehicles are mainly governed by equations proposed by Fossen [1]. In various studies, this model is taken as the basis; and some minor modifications are done when necessary.

Based on this vectorial model, equations describing the dynamics and kinematics of an unmanned sea surface vehicle have been proposed. The model has external forces and torques acting on the inertia of the vehicle. These external forces are Coriolis and centripetal forces, damping forces, gravitational and buoyancy forces, air drag due to motion of the vehicle and the winds, forces induced by ocean currents, forces acting on the body due to its motion in the surrounding water and the actuating forces, i.e., forces induced by rudder and thruster. Each of these components is explained in detail.

Following the description of the mathematical model, autopilots are designed for the vehicle. In order to design autopilots, linearized models of the nonlinear model have been extracted at various reasonable linearization points. The results on these linearization points are interpolated in later use. PID and LQR controllers for surge speed control and yaw position control are simulated. Since the LQR controller needs to be run on a controllable system, controllable subspace of the mathematical model has been extracted and LQR

theory has been applied on this subspace. The autopilots have logic to adjust the speed of the vehicles according to heading angle commands also.

The performances of the autopilots are compared in simulations. Their closed-loop eigenvalues are compared, and it has been observed that the LQR controller is slightly more successful compared to the PID controller. Passive stability of the vehicle has also been checked and the mathematical model has been verified.

Based on the LQR autopilot a way-point guidance algorithm is implemented. The algorithm provides surge speed and yaw position commands to the autopilots. When the necessary heading is reached, the guidance algorithm fastens the vehicle until it reaches the next point. An obstacle avoidance algorithm has also been implemented. As encountered with an obstacle, this algorithm rules the commands to the autopilots. The strategy is to avoid the obstacles, if it is needed to pass through it, and then rotate around the obstacle, until the other side is reached. Furthermore, in case the course of the vehicle distracted, and it is impossible to reach the next way-point, another algorithm generates the next way-point command to the guidance algorithm.

The guidance algorithm together with obstacle avoidance logic is simulated under disturbances of wind and current. It has been observed that wind disturbances up to 12 Knots do not affect the vehicle modeled in simulation. The cross wind disturbances make the vehicle distract from its course; nevertheless, the vehicle still can keep the general course and avoid the obstacles. However, disturbances from ocean current affect the vehicle significantly. Disturbance of 0.5 Knots in cross directions make the vehicle collapse by hitting an obstacle. Further variations in disturbances are not simulated, since a baseline has thought to be found.

Yet, the vehicle modeled can tolerate the disturbances up to certain extent. For a more successful behavior in an environment with disturbance, the controllers should have been designed for the mathematical model taking the disturbances into account.

PID controller for surge speed and yaw position are implemented on hardware with Traxxas Spartan model boat with ArduMega Pilot autopilot card together with Olipan IMU Shield on which the sensors are mounted. The controller has been tested, and autonomous motion in real life has been observed. The tests are done in Yalıncağ Göllet, where the vehicle is subjected to the disturbances such as wind, current and waves.

The performance of the PID controller in hardware is not completely successful. The overshoots reach to 50%. This insufficient behavior may be of various reasons. The model boat is not a perfect design, the sensors from which the feedback is taken are noisy, the disturbances in the environment may overcome the ability of the actuators to move the vehicle, and the drifts in sensors may vary if small percent of disturbance is available during calibration.

The experimental results, at least, validated the idea of controlling unmanned sea surface vehicles with the methods have been proposed in this thesis.

6.2 Future Works

In this thesis, control and guidance of an unmanned sea surface vehicle is studied, simulated and partially experimentally validated.

On a more professional sea surface vehicle, the PID controller being tuned for this vehicle can be verified and better results can be obtained. With some system identification studies, the parameters of the vehicle under experiment can be obtained, and the controller designed in this thesis for simulations can be emulated directly.

Only PID controllers have been verified in this thesis. If the mathematical model parameters of the vehicle are known, LQR controllers can be implemented as well. It has been observed in the simulations that the LQR controller may work more successfully compared to the PID controller.

The guidance algorithm together with obstacle avoidance strategy can be emulated as well. With necessary sensors and availability inside the vehicle, a fully autonomous vehicle which can guide itself, avoids from vehicle and does the mission it has been assigned to, may be obtained.

As a further study, a number of such kinds of sea surface vehicles can be used in a mission of reconnaissance, attack or tracking. The assignments of missions to each of the vehicle, a common guidance and navigation logics prepared may have a contribution to literature.

Unmanned sea surface vehicles can be used in a mission in which it operates with not only other unmanned surface vehicle but also with other autonomous vehicles underwater, aerial

or land. The common logic in such a mission, or the communication between the parts of the system can be studied as well.

REFERENCES

- [1] Fossen, T. I., *Marine Control Systems Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*, 2002.
- [2] Sname, The Society of Naval Architects and Marine Engineers, *Nomenclature of Treating the Motion of a Submerged Body through a Fluid, Technical and Research Bulletin*, vol. 1-5, 1950.
- [3] Abkowitz, M. A., *Lectures on Ship Hydrodynamics, Steering and Manoeuvrability, Technical Report Hy-5*, 1964.
- [4] Fossen, T. I., *Nonlinear Modeling and Control of Underwater Vehicles*, PhD thesis, Norwegian University of Science and Technology, 1991.
- [5] Craig, J. J., *Introduction to Robotics*, 1989.
- [6] Sciavicco, L. and Siciliano, B., *Modeling and Control of Robot Manipulators*, 1996.
- [7] Sagatun, S. I. and Fossen, T. I., *Lagrangian Formulation of Underwater Vehicles' Dynamics*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 1029-1034, 1991.
- [8] Fossen, T. I., *Guidance and Control of Ocean Vehicles*, 1994.
- [9] Fossen, T. I. and Fjellstad, O. E., *Nonlinear Modeling of Marine Vehicles in 6 Degrees of Freedom*, International Journal of Mathematical Modeling of Systems, vol. 1(1), pp. 17-28, 1995.
- [10] Berge, S. P. and Fossen, T. I., *On the Properties of the Nonlinear Ship Equations of Motion*, Journal of Mathematical and Computer Modeling of Dynamical Systems, vol. 6(4), pp. 365-381, 2000.
- [11] Minorsky, N., *Directional Stability of Automatic Steered Bodies*, Journal of American Society of Naval Engineers, vol. 34 (2), pp. 280-309, 1922.

- [12] Grovlen, A. and Fossen, T. I., *Nonlinear Control of Dynamic Positioned Ships Using Only Position Feedback: An Observer Backstepping Approach*, Proceedings of the 35th IEEE Conference on Decision and Control, pp. 3388-3393, 1996.
- [13] Strand, J. P., *Nonlinear Passive Observer for Ships with Adaptive Wave Filtering*, PhD thesis, Norwegian University of Science and Technology, 1999.
- [14] Kalman, R. E., *A New Approach to Linear Filtering and Prediction Problems*, ASME Transactions, Series D: Journal of Basic Engineering, vol. 82, pp. 35-42, 1960.
- [15] Kalman, R. E. and Bucy, R. S., *New Results in Linear Filtering and Prediction Theory*, ASME Transactions, Series D: Journal of Basic Engineering, vol. 83, pp. 95-108, 1961.
- [16] Donha, D. C., Desanj, M. R., Katebi, M. R. and Grimble, M. J., *H-infinity Adaptive Controller for Autopilot Applications*, Special Issue on Marine Systems of International Journal of Adaptive Control and Signal Processing, vol. 12(8), pp. 623-648, 1998.
- [17] Do, K. D., Jiang, Z. P. and Pan, J., *Robust Adaptive Path Following for Underactuated Ships*, Automatica: A Journal of International Federation of Automatic Control, vol. 40, pp. 929-944, 2004.
- [18] Reyhanoğlu, M., *Exponential Stabilization of an Underactuated Autonomous Surface Vessel*, Automatica: A Journal of International Federation of Automatic Control, vol.33, pp. 2249-2254, 1997.
- [19] Lefeber, E., Pettersen, K. Y. and Nijmeiker, H., *Tracking Control of an Underactuated Ship*, IEEE Transactions on Control Systems Technology, vol. 11, 52-61, 2003.
- [20] Jiang, Z. P., *Global Tracking Control of Underactuated Ships by Lyapunov's Direct Method*, Automatica: A Journal of International Federation of Automatic Control, vol.38, pp. 301-309, 2002.
- [21] Bao-li, M., *Global k-exponential Asymptotic Stabilization of Underactuated Surface Vessels*, Systems and Control Letters, vol. 58, pp. 194-201, 2009.
- [22] Liao, Y. L., Wan, L. and Zhuang, J. Y., *Backstepping Dynamical Sliding Mode Control Method for the Path Following of the Underactuated Surface Vessel*, Procedia Engineering, vol. 15, pp. 256-263, 2011.

- [23] Naeem, W., Sutton, R. and Chudley, J., *Soft Computing Design of a Linear Quadratic Gaussian Controller for an Unmanned Surface Vehicle*, 14th Mediterranean Conference on Control and Automation, pp. 1-6, 2006.
- [24] Caccia, M., Bibuli, M., Bono, R. and Bruzzone, G., *Basic Navigation, Guidance and Control of an Unmanned Surface Vehicle*, *Autonomous Robots*, vol. 25(4), pp. 349-365, 2008.
- [25] Guo, J., Chiu, F. C., Huang and C. C., *Design of a Sliding Mode Fuzzy Controller for the Guidance and Control of an Autonomous Underwater Vehicle*, *Ocean Engineering*, vol. 30(16), pp. 2137-2155, 2003.
- [26] Johansen, T. A., Fuglseth, T. P., Tondel and P. Fossen, T. I., *Optimal Constrained Control Allocation in Marine Surface Vessels with Rudders*, *Control Engineering Practice*, vol. 16(4), pp. 457-464, 2008.
- [27] Qiaomei, S., Guang R., Jin Y. and Xiaowei, Q., *Autopilot Design for Unmanned Surface Vehicle Tracking Control*, 3rd International Conference on Measuring Technology and Mechatronics Automation, 2011.
- [28] Bibuli, M., Burzzone, G., Caccia, M., Indiveri and G., Zizzari, A. A., *Line Following Guidance Control: Application to the Charlie Unmanned Surface Vehicle*, IEEE International Conference on Intelligent Robots and Systems, 2008.
- [29] Svec, P., Schwartz, M., Thakur, A. and Gupta, S. K., *Trajectory Planning with Look-Ahead for Unmanned Sea Surface Vehicles to Handle Environmental Disturbances*, IEEE International Conference on Intelligent Robots and Systems, 2011.
- [30] Krishnamurthy, P. and Khorrami, F., *A Hierarchical Control and Obstacle Avoidance System for Unmanned Sea Surface Vehicles*, 50th IEEE Conference on Decision and Control, 2011.
- [31] Naeem, W., Irwin, G. W. and Yang A., *COLREGs-based Collision Avoidance Strategies for Unmanned Surface Vehicles*, *Journal of Mechatronics*, in press, 2011.
- [32] Zhuang, J. Y., Su, Y. M., Liao, Y. L. and Sun, H. B., *Motion Planning of USV Based on Marine Rules*, *Procedia Engineering*, vol. 15, pp. 269-276, 2011.

- [33] Browning, A. W., *A Mathematical Model to Simulate Small Boat Behavior*, Bournemouth Polytechnic, 1990.
- [34] Hughes, P. C., *Spacecraft Attitude Dynamics*, 1986.
- [35] Kane, T. R., Likins, P. W. and Levinson, D. A., *Spacecraft Dynamics*, 1983.
- [36] Faltinsen, O. M. and Sortland, B., *Slow Drift Eddy Making Damping on a Ship*, Applied Ocean Research, vol. 9(1), pp. 37-46, 1987.
- [37] Lamb, H., *Hydrodynamics*, 1945.
- [38] Perez, T., Blanke, M., *Mathematical Ship Modeling for Control Applications*, Technical Report EE02036.
- [39] Lewis, E. V., *Principles of Naval Architecture vol II:Resistance, Propulsion and Vibration*, Society of Naval Architecture and Marine Engineers, 1988.
- [40] Berlekom, W. B., *Effects of Propeller Loading on Rudder Efficiency*, Proceedings 4th Ship Control Symposium, pp. 5.83-5.98, 1975.
- [41] Kirk, D. E., *Optimal Control Theory: An Introduction*, 1970.
- [42] Aström, K. J. and Hägglund, T., *Advanced PID Control*, NC: Instrumentation, Systems, and Automation Society, 2006.
- [43] Kalman, R. E., *Contributions to the Theory of Optimal Control*, Bol.Soc.Mat., pp. 102-119, 1960.
- [44] Traxxas Inc., *Traxxas Spartan Model 5707 Owner's Manual*, 2011.
- [45] DIY Drones Inc., *ArduPilot Mega, APM1 Quick Start Guide*, 2012.

APPENDIX A

DERIVATION OF ADDED MASS PARAMETERS

In this appendix, derivation of added mass parameters will be explained. The body of the vehicle is assumed to be an ellipsoid. The details for derivation of added mass parameters for sea surface vehicle with the assumption of an ellipsoid rigid body can be found in literature as well [33].

The equation of an ellipsoid is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (\text{A.1})$$

where a , b and c are the semi-major, semi-minor and semi-vertical axes of the ellipsoid respectively.

According to Lamb's theory [37], due to symmetry within an ellipsoid, only the added mass coefficients on the leading diagonals will have nonzero values. Therefore:

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0} \frac{4}{3} \pi \rho^w abc \quad (\text{A.2})$$

$$Y_{\dot{v}} = -\frac{\beta_0}{2-\beta_0} \frac{4}{3} \pi \rho^w abc \quad (\text{A.3})$$

$$Z_{\dot{w}} = -\frac{\gamma_0}{2-\gamma_0} \frac{4}{3} \pi \rho^w abc \quad (\text{A.4})$$

$$K_{\dot{p}} = -\frac{1}{5} \frac{(b^2-c^2)^2 (\gamma_0-\beta_0)}{2(b^2-c^2)+(b^2+c^2)(\beta_0-\gamma_0)} \frac{4}{3} \pi \rho^w abc \quad (\text{A.5})$$

$$M_{\dot{q}} = -\frac{1}{5} \frac{(c^2-a^2)^2 (\alpha_0-\gamma_0)}{2(c^2-a^2)+(c^2+a^2)(\gamma_0-\alpha_0)} \frac{4}{3} \pi \rho^w abc \quad (\text{A.6})$$

$$N_{\dot{r}} = -\frac{1}{5} \frac{(a^2-b^2)^2(\beta_0-\alpha_0)}{2(a^2-b^2)+(a^2+b^2)(\alpha_0-\beta_0)} \frac{4}{3} \pi \rho^w abc \quad (\text{A.7})$$

In the equations above, α_0 , β_0 and γ_0 are numerical quantities that describe the relative proportions of the ellipsoid.

If $b = c$, i.e., the ellipsoid is prolate, then we have:

$$\beta_0 = \gamma_0 \quad (\text{A.8})$$

The off-diagonal added mass coefficients will be still zero, but the diagonal entries will be reduced to:

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0} \frac{4}{3} \pi \rho^w ab^2 \quad (\text{A.9})$$

$$Y_{\dot{v}} = Z_{\dot{w}} = -\frac{\beta_0}{2-\beta_0} \frac{4}{3} \pi \rho^w ab^2 \quad (\text{A.10})$$

$$K_{\dot{p}} = 0 \quad (\text{A.11})$$

$$M_{\dot{q}} = N_{\dot{r}} = -\frac{1}{5} \frac{(b^2-a^2)^2(\alpha_0-\beta_0)}{2(b^2-a^2)+(b^2+a^2)(\beta_0-\alpha_0)} \frac{4}{3} \pi \rho^w ab^2 \quad (\text{A.12})$$

In the equations above:

$$\alpha_0 = \frac{(1-e^2)}{e^3} \left(\log \left(\frac{1+e}{1-e} \right) - e \right) \quad (\text{A.13})$$

$$\beta_0 = \frac{1}{e^2} - \frac{(1-e^2)}{2e^3} \log \left(\frac{1+e}{1-e} \right) \quad (\text{A.14})$$

where e is the eccentricity of the meridian elliptical section:

$$e = 1 - \left(\frac{b}{a} \right)^2 \quad (\text{A.15})$$

APPENDIX B

DERIVATION OF DAMPING PARAMETERS

In this appendix, linear and quadratic damping parameters used in mathematical model will be derived. The derivation is done according to the explanations in [33].

$$X_{|u|u} = \frac{1}{2} \rho^w C_D A_{UT} \quad (\text{B.1})$$

In (B.1), quadratic damping in direction of surge speed is presented. ρ^w stands for density of water, C_D is the drag coefficient of water on the cross-sectional area of A_{UT} , and A_{UT} is the traverse cross-sectional area underwater.

$$A_{UT} = \int_0^{wi} \int_0^{h_0} dy dz \quad (\text{B.2})$$

In (B.2), A_{UT} is presented, h_0 is the length of the vehicle underwater, and wi is the width of the vehicle. Note that vehicle is assumed to be in touch with water in its entire traverse surface.

With similar perspective longitudinal and water-plane cross-sectional areas underwater can be defined as:

$$A_{UL} = \int_0^{h_0} \int_0^l dx dz \quad (\text{B.3})$$

$$A_{WP} = \int_0^l \int_0^{wi} dx dy \quad (\text{B.4})$$

Similar to (B.1), quadratic damping parameters on other translational axes can be defined as:

$$Y_{|v|v} = \frac{1}{2} \rho^w C_D A_{UL} \quad (\text{B.5})$$

$$Z_{|w|w} = \frac{1}{2} \rho^w C_D A_{WP} \quad (\text{B.6})$$

The cross quadratic damping coefficients to be used can be calculated as:

$$Y_{|v|r} = Y_{|r|v} = \frac{1}{2} \rho^w C_D \int_0^l \int_0^{wi} x^2 y^2 dx dy \quad (\text{B.7})$$

$$N_{|r|r} = \frac{1}{2} \rho^w C_D \int_0^l \int_0^{wi} x^2 y^3 dx dy \quad (\text{B.8})$$

Linear damping coefficients can be found as follows:

$$X_u = \frac{1}{2} \rho^w C_D \int_0^l x dx \quad (\text{B.9})$$

$$Y_v = \frac{1}{2} \rho^w C_D \int_0^{wi} y dy \quad (\text{B.10})$$

$$Y_r = N_v = \frac{1}{2} \rho^w C_D \int_0^{h_0} \int_0^{wi} z y dy dz \quad (\text{B.11})$$

$$N_r = \frac{1}{2} \rho^w C_D \int_0^{h_0} \int_0^{h_0} z^2 dz \quad (\text{B.12})$$