AUTOPILOT AND GUIDANCE DESIGN FOR A MINI ROV (REMOTELY OPERATED UNDERWATER VEHICLE)

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

BY

FIRAT YILMAZ CEVHER

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

SEPTEMBER 2012

Approval of the thesis:

AUTOPILOT AND GUIDANCE DESIGN FOR A MINI ROV

(REMOTELY OPERATED UNDERWATER VEHICLE)

submitted by **FIRAT YILMAZ CEVHER** 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. M. Kemal Leblebicioğlu _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

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

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

Asst. Prof. Dr. Emre Tuna _____
Electrical and Electronics Engineering Department, METU

Asst. Prof. Dr. Gökhan O. Özgen _____
Mechanical Engineering Department, METU

Asst. Prof. Dr. Umut Orguner _____
Electrical and Electronics Engineering Department, METU

Date: **13.09.2012**

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: FIRAT YILMAZ CEVHER

Signature :

ABSTRACT

AUTOPILOT AND GUIDANCE DESIGN FOR A MINI ROV
(REMOTELY OPERATED UNDERWATER VEHICLE)

Cevher, Fırat Yılmaz
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. M. Kemal Leblebicioğlu

September 2012, 108 pages

This thesis consists of a mathematical model, autopilot and guidance design of a mini ROV (Remotely Operated Underwater Vehicle) and investigates the effects of environmental forces (ocean currents etc.) on the guidance algorithms. First of all, a non-linear 6 degrees-of-freedom (DOF) mathematical model is obtained. This model includes hydrodynamics forces and moments. There is no exact calculation method for hydrodynamic coefficients; however strip theory and results of computational fluid dynamics (CFD) analysis are used to calculate their approximate values. Linear mathematical model is obtained by linearization at trim points and it is used when designing surge speed, heading and depth controller. Guidance is examined by two methods such as way point guidance by line-of-sight (LOS) and way point guidance based on optimal control. Moreover, an online obstacle avoidance algorithm is developed. This thesis ends with the subject of navigation of the vehicle under GPS-like measurements and magnetic sensors measurements.

Keywords: Mathematical Modelling of ROV, Autopilot, Guidance, Obstacle Avoidance, Navigation

ÖZ

BİR MİNİ İNSANSIZ SUALTI ARACININ OTOPİLOT VE GÜDÜM TASARIMI

Cevher, Fırat Yılmaz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. M. Kemal Leblebicioğlu

Eylül 2012, 108 sayfa

Bu tez, bir mini insansız sualtı aracının modellenmesini, otopilot ve güdüm tasarımını ve çevresel etmenlerin (akıntı vb.) güdüm üzerindeki etkilerini içermektedir. İlk olarak, aracın 6 serbestlik dereceli doğrusal olmayan modeli elde edilmiştir. Bu model hidrodinamik kuvvet ve momentleri içermektedir. Hidrodinamik katsayıları net olarak hesaplama tekniği yoktur; ancak bu katsayıların yaklaşık değerleri şerit teoremi ve hesaplamalı akışkanlar analizi sonuçlarına göre elde edilmiştir. Denge noktaları etrafında doğrusallaştırılan model, sürat hızı, yönelim ve derinlik kontrolcüleri tasarlanırken kullanılmıştır. Güdüm, görüş hattı yöntemi ve optimal kontrol problemi olarak ele alınmıştır. Ayrıca çevrimiçi olarak kullanılabilecek bir engelden sakınma algoritması geliştirilmiştir. Tez, aracın GPS benzeri bir ölçüm ile manyetik algılayıcı ölçümleri altında navigasyonu ile tamamlanmaktadır.

Anahtar Kelimeler: İnsansız Sualtı Aracının Matematiksel Modeli, Otopilot, Güdüm, Engelden kaçma, Navigasyon

to my family
&
to my fiancée

ACKNOWLEDGMENTS

I would like to express my deep appreciation and sincere gratitude to my supervisor Prof. Dr. M. Kemal Leblebicioğlu for his leading guidance, advice, encouragement, criticism and continuous support from beginning to the end of my M.Sc. thesis.

This thesis has been supported by Desistek Robotics. I would also convey my thanks to my colleagues M.Sc. Emre Ege and Gökhan Onur for their valuable support, advices, tolerance and understanding.

Additionally, I also thank to Seda Karadeniz Kartal for her support and for helpful discussions we have made throughout the study.

I would like to express my special thanks to my fiancé for her valuable support and motivation.

Finally, but definitely not the least, nothing is adequate to express my heartfelt feelings to my beloved family forever. None of this would have been even possible without their love, perpetual support, encouragement, patience and trust.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In recent years, the studies of the unmanned underwater vehicles (UUVs) are important for safety of human life. UUVs are operated in military operations such as mine dragging, coast-guard etc., and also, they are used for civilian purposes. These vehicles are divided into two types such as autonomous underwater vehicles (AUVs) and remotely operated underwater vehicles (ROVs). Both of them can carry instruments, take samples and conduct surveys. However, the main difference between the two is that ROV is physically connected to the ship by tether, AUVs are not connected. AUVs are fully autonomous vehicles; it means that they are programmed before the operation and no human operator is needed while they are operating. On the other hand, there is at least one human needed to operate ROVs. Each has different advantages and disadvantages. For instance, as mentioned above, at least one person is required for ROVs. On the other hand, real-time video capturing is done easily by using ROVs.

All work in this thesis is done based on DST-R-100-4 ROV as shown in Figure 1.1, which is produced and designed by Desistek Robotics Inc. It is in miniROV class: its weight is 15.5 kgs; so it is transportable. It has four thrusters two of which are used in the forward direction, one of them is horizontal and the last one is vertical direction. It has a magnetic compass, a pressure sensor, a temperature sensor and an inertial measurement unit (IMU). Hence, heading angle and depth of the vehicle are obtained easily. Consequently, the vehicle can hold itself at fixed depth and heading angle by autopilots.

Briefly, an autopilot is a system that is used to guide a vehicle without assistance from a hu-

Figure 1.1: DST-R-100-4 ROV

man being. A guidance system produces reference signals for the autopilots independent of human intervention. Autopilot - guidance and navigation systems work together. For example, suppose the ROV pilot wants the ROV to go to a fixed position. Firstly, the navigation system determines where the vehicle is. Then, it sends desired position and actual position of the vehicle to the guidance system. The guidance system generates set points for controllers such as heading angle controller , surge speed controller, etc. Next, the autopilot system tries to keep the vehicle at the desired surge speed and the heading angle. In other words, the navigation system determines where the vehicle is, the guidance system determines where the vehicle should go and autopilot system enables the vehicle to perform the desired actions. Figure 1.2 summarizes relation between navigation, autopilot and guidance systems.

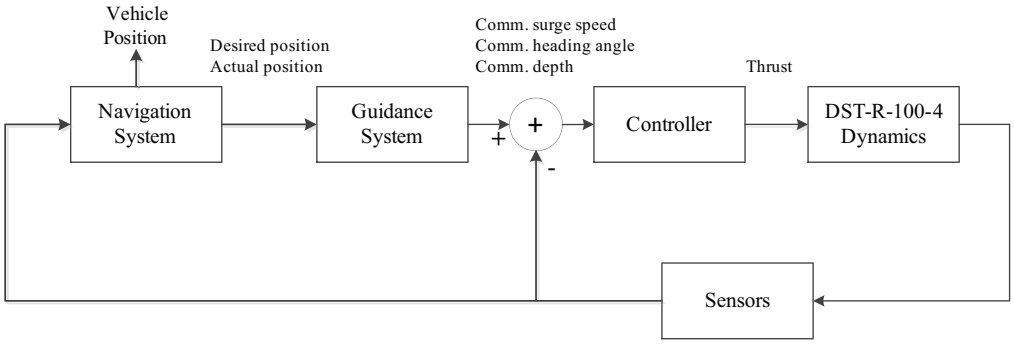


Figure 1.2: General view of an autonomous underwater vehicle structure

1.2 Literature Survey

Since the underwater environment presents some difficulties and usually an underwater vehicle has non-linear behaviour, the studies of control and guidance of the UUVs (unmanned underwater vehicles) have required special attention. Usually the mathematical model of underwater vehicles are linearized around equilibrium point [16], [17]. The resultant linearized mathematical model can either be considered coupled or decoupled with respect to its states. Different control techniques are used to design the autopilots after linearization. For example, speed autopilots may either be P or PI control techniques [10]. The same controllers can be used for position control as well. Moreover the depth control may be performed by a PID controller [46]. More complicated autopilots can be designed for more complicated behaviour. For example, depth and steering autopilot can be designed by using PID techniques as well as sliding mode control techniques [26].

Guidance system has an important role for producing the desired behaviour. Several guidance methods have been developed in the last twenty years. The most common method in guidance is the line-of-sight (LOS) guidance. Usually, the autopilot for controlling yaw responses is integrated with LOS guidance [39]. The required position and orientation can either be done in 2D or 3D environments [23]. The guidance part can be combined by obstacle avoidance strategies [44]. While the vehicle travels within the sea, it may meet obstacles. Position and geometry of these obstacles may be known and they may have different shapes [41], [35]. On the other hand, if obstacles shapes and positions are unknown, obstacle avoidance is performed with more difficulty. A real time obstacle avoidance is done by Antonelli and Chiaverini [3].

As regards the navigation of an unmanned underwater vehicles not many studies have been performed. The study of the navigation of an underwater vehicle is quite demanding. The GPS (global positioning system) for many robotic missions on land and in the air is practical; but the GPS is not available for underwater vehicles. The devices helping the navigation of underwater vehicles have too much error. The inertial measurement unit measures and reports the vehicle's velocity, orientation and gravitational forces using a combination of accelerometers and gyroscopes. The IMU (inertial measurement unit) works by detecting the current rate of acceleration using one or more accelerometers and detects changes in rotational attributes like pitch, roll and yaw using one or more gyroscopes. The major disadvantage of

using IMUs for navigation is that they typically suffer from accumulated error. As a result the guidance system is continually adding detected changes to its previously calculated positions with errors in measurements.

The navigation and guidance system (NGS) with real time path planning are discussed in [3]. Map module, supervisor module, target module and trajectory module are performed and analyzed. Also, the time history of the distance from the pipeline was simulated. The map module has some drift error because of the vehicle position which is obtained by integration of velocity measurements. There is a method that is developed for minimizing this error. This method resets the error by discovering known landmarks. The target module has some error because of the prediction of the pipeline's direction whose information is taken from a vision system.

The methodology was presented for mosaic-based visual navigation for underwater vehicles for missions where the vehicle was asked to approach a distant point [19]. The adaptive identification on the group of rigid-body rotations and its application to underwater vehicle navigation were performed in [7].

1.3 Thesis Organization

This thesis consists of a mathematical modelling of an underwater vehicle, mathematical model parameters derivation, autopilot and guidance design, obstacle avoidance, effects of ocean currents on the path and navigation of ROV. As a summary;

- Chapter 1 consists of basic definitions about unmanned underwater vehicles and literature survey about scope of this thesis.

- Chapter 2 includes non-linear and linear 6 DOF equations of motion, kinematics, rigid-body dynamics and hydrodynamic forces and moments. Firstly, a non-linear 6 degrees-of-freedom mathematical model of the underwater vehicles are obtained in general, next the model is specialized for the vehicle DST-R-100-4 related with the mechanical design properties. The model includes hydrodynamic forces and moments which consist of added mass and inertia, hydrodynamic damping forces and restoring forces. Finally, a linear time-invariant model is obtained by using linearization of the non-linear model

around special equilibrium point.

- Chapter 3 explains how model parameters are derived. The model parameters are added mass coefficients and drag coefficients. There is no exact calculation method for added mass coefficients; however their approximate values are calculated by using strip theory. Drag coefficients are obtained by results of computational fluid dynamics (CFD) analysis and with the help of Seamor ROV [45] parameters.

- Chapter 4 includes the thruster model which is divided into three parts such as; motor model, fluid model and propeller mapping. The parts are explained in detail first. Secondly, maximum values for surge, sway, heave speed and yaw rate are obtained by using LTI model and the parameters derived in Chapter 3. Thirdly, speed controllers, depth controller and heading controller are designed by using PID controller. Response of the controllers are examined by using LTI model and non-linear model.

- Chapter 5 includes the guidance and a real-time obstacle avoidance algorithm. Guidance is divided into two parts such as; way point guidance by line-of-sight (WPG - LOS) and way point guidance based on optimal control (WPG - OC). The main idea behind WPG - OC is creating trajectory that requires minimum energy. For this reason, an optimal control problem (OCP) is developed. The OCP is solved by two algorithms: genetic algorithm (GA) and interior-point algorithm (IPA), respectively. Optimal solution found by GA is used as initial solution for IPA. Consequently, global optimum solution is obtained. In addition, the effects of ocean currents on the path produced by WPG - LOS and WPG - OC are discussed. However, energy consumption is not concerned. Finally, real-time obstacle avoidance algorithm is developed for unknown obstacles.

- Chapter 6 consists of navigation of the vehicle under GPS-like measurements and magnetic sensors measurements. Since system model is non-linear, extended kalman filter (EKF) is used to estimate state variables. The EKF performance for different cases are discussed.

CHAPTER 2

MATHEMATICAL MODELLING

2.1 Introduction

Mathematical model of an underwater vehicle incorporates several discipline of statics and dynamics of vehicle motion. Statics refers to the forces and moments acting on the physical system around equilibrium points, dynamics refers to the effect of forces on the motion of objects.

Motion of the underwater vehicle is represented by a *6 degree of freedom (6 DOF)* model in order to determine the position and orientation of the body. As shown in Table 2.1, the first three coordinates and their time derivatives are used to represent positions and translational motions along the axes *x*-axis, *y*-axis and *z*-axis. The orientations and rotational motions are described by using the last three coordinates and their time derivatives.

There are twelve non-linear first order differential equations that are used to establish the

Table 2.1: Notation used for underwater vehicle

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

motion of an underwater vehicle. Six of them are called kinematic equations. Transition between the reference axis system and the body-fixed axis system is obtained by using a transformation matrix which is based on Euler angles. Final form of kinematic equations are obtained by using transformation matrix. Remaining six of equations are called dynamic equations that are obtained by using the Newton's second law.

In this chapter, first, coordinate frames used in the description of the mathematical model of an underwater vehicle are presented. Then, a non-linear mathematical model is derived by using equations of motion of the vehicle. This model includes important hydrodynamic effects. In order to obtain a linear model for the autopilot design, linearization is done around equilibrium points under some assumptions related with the mechanical design.

2.2 Kinematics

2.2.1 Coordinate Frames

Two coordinate frames are needed in order to analyse motion of underwater vehicles, such as body-fixed coordinate frame and earth-fixed coordinate frame as shown in Figure 2.1. Body-fixed coordinate frame is assumed to be fixed to the center of gravity *CG* of the vehicle and moving with it and *O* shows the CG of the vehicle in Figure 2.1.

The body axes X_O, Y_O, and Z_O coincide with the *principle axes of inertia*, and usually presented as [17]:

X_O - longitudinal axis

Y_O - transverse axis

Z_O - normal axis

When the body-fixed frame is rotating about the earth-fixed coordinate frame with a certain angular velocity, the vehicle shows a resistance to changing of its rotational position which is called the *moment of inertia*. In other words, the moment of inertia shows how hard to turn the vehicle around the coordinate axes. I_x in Equation 2.1 denotes moment of inertia around $x-$axis when the vehicle is rotating around $x-$axis. I_{xy} denotes the moment of inertia around

$y-$axis when the vehicle is rotating around $x-$axis, which is also called the *product of inertia*. Consequently, the inertia tensor, shown in Equation 2.1 is obtained with these components.

$$I_0 = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.1}$$

where $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$ and $I_{yz} = I_{zy}$ Some components of Equation 2.1 equal zero depending on the mechanical design of the vehicle. For instance, if the vehicle has three plane symmetry, inertia tensor has only diagonal components (moment of inertia components).

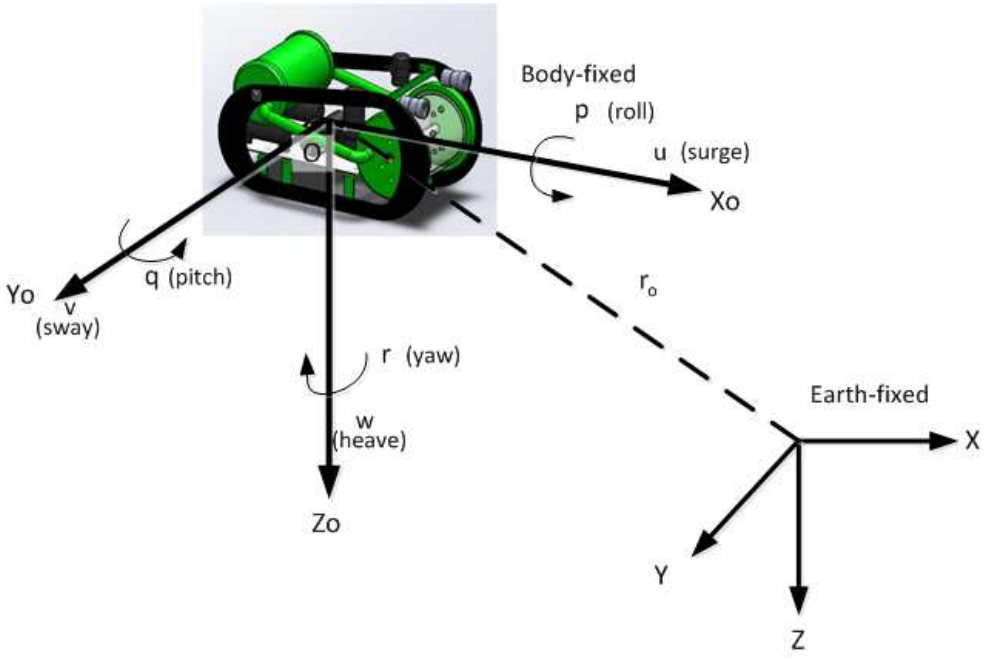


Figure 2.1: Body-fixed and Earth-fixed reference frames

The earth-fixed coordinate frame is used to describe the motion of the vehicle relative to the body-fixed frame. It is assumed that, acceleration of a point on the earth surface is negligible for underwater vehicles, since the effect of motion of the Earth is very small for underwater vehicles. X and Y axes point to the North and East directions, respectively and they are located on a plane tangent to the Earth's surface. As a result, Z axis points down to the center of the Earth. Note that, right hand rule is used for all coordinate frames. This coordinate system is also called as the local-level or the North-East-Down (NED), in literature [47].

The general motion of an underwater vehicle in 6DOF can be described by using SNAME notation, as follows [17]

$$\eta = [\eta_1^T, \eta_2^T]^T \qquad \eta_1 = [x, y, z]^T \qquad \eta_2 = [\phi, \theta, \psi]^T \tag{2.2}$$

$$\upsilon = [\upsilon_1^T, \upsilon_2^T]^T \qquad \upsilon_1 = [u, v, w]^T \qquad \upsilon_2 = [p, q, r]^T \tag{2.3}$$

$$\tau = [\tau_1^T, \tau_2^T]^T \qquad \tau_1 = [X, Y, Z]^T \qquad \tau_2 = [K, M, N]^T \tag{2.4}$$

where

η : the position and orientation vector with coordinates in the earth-fixed frame

υ : the linear and angular velocity vector with coordinates in the body-fixed frame

τ : the forces and moments acting on the vehicle in the body-fixed frame.

2.2.2 Euler Angles

As mentioned before, a coordinate transformation matrix is used for transformation from the body-fixed frame to the earth-fixed frame and vice versa, by using the Euler Angles.

The transformation matrix $T_{BE,1}(\eta_2)$ is used for linear velocity transformation, which transforms velocities from the body-fixed coordinate frame to the earth-fixed coordinate frame as;

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = T_{BE,1}(\eta_2) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.5}$$

where

$$T_{BE,1}(\eta_2) =$$

$$\begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\cos(\phi)\sin(\theta) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\psi)\sin(\phi) + \sin(\theta)\sin(\psi)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{2.6}$$

For angular velocity transformation, the transformation matrix $T_{BE,2}(\eta_2)$ relates the body-fixed angular velocity vector υ_2 and the Euler rate vector $\dot{\eta}_2$ according to:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_{BE,2}(\eta_2) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.7}$$

where

$$T_{BE,2}(\eta_2) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \tag{2.8}$$

Note that, if the pitch angle of θ equals $\pm\pi/2$, $T_{BE,2}(\eta_2)$ is undefined. Mechanical design of the vehicle guarantees that the pitch angle will never be $\pm\pi/2$ during operation, because of the fact that the center of gravity is close to the bottom side of the vehicle as well as the center of buoyancy is close to the top side of the vehicle. Consequently, the vehicle has passive roll stability.

As a result, the kinematic equations can be written in vector form as:

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} T_{BE,1}(\eta_2) & 0_{3X3} \\ 0_{3X3} & T_{BE,2}(\eta_2) \end{bmatrix} \begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} \tag{2.9}$$

or more compact form as:

$$\dot{\eta} = T_{BE}(\eta)\upsilon \tag{2.10}$$

Further information about how to obtain the coordinate transformation matrix can be found in Appendix A.

2.3 Rigid-Body Dynamics

Non-linear dynamical equations of motion of an underwater vehicle in 6DOF can be written as [17]

$$M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{2.11}$$

where

M : inertia matrix including added mass

$C(\upsilon)$: matrix of Coriolis & centripal terms including added mass

$D(\upsilon)$: hydrodynamic damping matrix

$g(\eta)$: vector of gravitational forces and moments

τ : vector of control inputs

In order to derive the rigid-body equations of motion, first and second axioms of Newton's second rule are applied. Therefore, 6DOF rigid-body equations of motion are obtained in component form as follows:

$$
\begin{aligned}
\sum X &= [\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] \\
\sum Y &= [\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] \\
\sum Z &= [\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] \\
\sum K &= I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \cdots \\
&\quad + m[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)] \\
\sum M &= I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \cdots \\
&\quad + m[z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)] \\
\sum N &= I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \cdots \\
&\quad + m[x_G(\dot{v} - wp + ur) - y_G(\dot{u} - vr + wq)]
\end{aligned}
\tag{2.12}
$$

The first three equations of Equation 2.12 express the translational motion and the last three one express the rotational motion. Equation 2.12 can be represented in more compact form such as:

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

where

υ : body-fixed linear & angular velocity vector

τ_{RB} : generalized vector of external forces & moments

M_{RB} : the rigid-body inertia matrix

C_{RB} : the rigid-body Coriolis & centripetal matrix

M_{RB} can be defined as

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mz_G \\ 0 & 0 & m & my_G & -mz_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mz_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.14}$$

C_{RB} also can be defined as

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -m(y_G q + z_G r) & m(y_G p + w) & m(z_G p - v) \\ m(x_G q - w) & -m(z_G r + x_G p) & m(z_G q + u) \\ m(x_G r + v) & m(y_G r - u) & -m(x_G p + y_G q) \end{bmatrix}$$

$$\begin{bmatrix} m(y_G q + z_G r) & -m(x_G q - w) & -m(x_G r + v) \\ -m(y_G p + w) & m(z_G r + x_G p) & -m(y_G r - u) \\ -m(z_G p - v) & -m(z_G q + u) & m(x_G p + y_G q) \\ 0 & -I_{yz} q - I_{xz} p + I_z r & I_{yz} r + I_{xy} p - I_y q \\ I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q + I_x p \\ -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_x p & 0 \end{bmatrix} \tag{2.15}$$

2.4 Hydrodynamic Forces and Moments

In literature, the hydrodynamic forces and moments on a rigid body can be considered in two parts such as radiation-induced forces and Froude-Kirilof and Diffraction forces [17]. In this thesis, it is assumed that forces on the body are forced to oscillate with wave excitation frequency and there are no incident frequency.

The radiation-induced forces and moments can be examined in three parts such as;

- *Added mass and inertia* is the inertia of surrounding fluid,
- *Hydrodynamic damping* is potential damping, skin friction, wave drift damping and damping due to vortex shedding,
- *Restoring forces* is weight and buoyancy due to Archimedes.

In Equation 2.13, τ_{RB} include hydrodynamic forces and moments (τ_H), environmental forces and moments (τ_E) and propulsion forces and moments (τ) acting on the vehicle in Equation 2.16

$$\tau_{RB} = \tau_H + \tau_E + \tau \tag{2.16}$$

Since there is no environmental forces and moments, τ_E can be omitted from Equation 2.16. Then, the following equation is obtained.

$$\tau_{RB} = \tau_H + \tau \tag{2.17}$$

where τ_H is given by

$$\tau_H = -M_A\dot{\upsilon} - C_A(\upsilon)\upsilon - D_\upsilon\upsilon - g(\eta) \tag{2.18}$$

Substitution of Equation 2.18 into Equation 2.16 together with Equation 2.13 yields the following 6 DOF dynamic equation of motion.

$$M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{2.19}$$

where

$$M = M_{RB} + M_A \quad , \quad C(\upsilon) = C_{RB}(\upsilon) + C_A(\upsilon) \tag{2.20}$$

2.4.1 Added Mass and Inertia

For a completely submerged vehicle, there is a volume of fluid particles surrounding the vehicle, which moves depending on acceleration or deceleration of the body. Added mass is the weighted integration of this entire mass of fluid particles. In other words, added mass is pressure-induced forces and moments due to a forced harmonic motion of the vehicle. It is proportional to the acceleration of the vehicle [17]. It is logical to divide added mass

forces and moments into two matrices such as the added mass matrix (M_A) and the matrix of hydrodynamic Coriolis and centripetal terms ($C_A(\upsilon)$)

Added mass coefficients can be assumed to be constant for underwater vehicles. This is because of the fact that they are independent of the wave circular frequency. Therefore, M_A and $C_A(\upsilon)$ can be defined in component form in general:

$$M_A = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{2.21}$$

$$C_A(\upsilon) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\alpha_3 & \alpha_2 \\ 0 & 0 & 0 & \alpha_3 & 0 & -\alpha_1 \\ 0 & 0 & 0 & -\alpha_2 & \alpha_1 & 0 \\ 0 & -\alpha_3 & \alpha_2 & 0 & -\beta_3 & \beta_2 \\ \alpha_3 & 0 & -\alpha_1 & \beta_3 & 0 & -\beta_1 \\ -\alpha_2 & \alpha_1 & 0 & -\beta_2 & \beta_1 & 0 \end{bmatrix} \tag{2.22}$$

where

$$\begin{aligned} \alpha_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\ \alpha_2 &= X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\ \alpha_3 &= X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\ \beta_1 &= X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\ \beta_2 &= X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r \\ \beta_3 &= X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w + K_{\dot{r}}p + M_{\dot{r}}q + N_{\dot{r}}r \end{aligned} \tag{2.23}$$

$X_{\dot{u}}$ represent added mass coefficient related with forces along X-axis (X_A), depending on the acceleration on X-axis, $\dot{u}$. X_A can be written as [2]

$$X_A = X_{\dot{u}}\dot{u} \tag{2.24}$$

$X_{\dot{u}}$ can be found by taking the derivative of the force along X-axis with respect to the acceler-

ation as

$$X_{\dot{u}} \triangleq \frac{\partial X}{\partial \dot{u}} \tag{2.25}$$

2.4.2 Hydrodynamic Damping Forces

There are several causes for hydrodynamic damping for underwater vehicles. *Potential Damping* occurs by the surface waves that have small wave length and high frequency. Since the underwater vehicle usually operates in deep water, effects of potential damping on the vehicle is very small. Therefore, this effect can be negligible for underwater vehicles. *Wave Damping* is close to potential damping that it occurs when the vehicle is moving on the surface of the water. For the same reason with the potential damping, it also can be negligible. *Skin Friction* is effective when the vehicle motion is in low-frequency, and can be represented as linear skin friction due to laminar boundary layers and quadratic skin friction due to turbulent layers. *Damping of vortex shedding* is wave drift damping. It occurs due to pressure differences on the flow path of the liquid. This damping is proportional to the projected cross-sectional area and changes proportional of square of the vehicle velocity.

Without loss of generality, hydrodynamic damping forces acting on an underwater vehicle are caused by skin friction and vortex shedding. Vortex shedding damping and skin friction can be modelled as shown in Equation 2.26 and Equation 2.27, respectively.

$$D_{vor}(\upsilon) = -\frac{1}{2}\,\rho\, C_d\,(Rn)\, A_{cs}\,|\upsilon|\,\upsilon \tag{2.26}$$

where ρ denotes density of fluid, $C_d(Rn)$ denotes drag coefficient depends on Reynold number and A_{cs} denotes projected cross-sectional area.

$$D_{skin}(\upsilon) = D_{lin} \;+\; D_{quad}(\upsilon) \tag{2.27}$$

Consequently, the drag effects acting on the vehicle can be divided into quadratic and linear

drag.

$$D(\upsilon) = -$$

$$\begin{bmatrix} X_u + X_{|u|u}|u| & X_v + X_{|v|v}|v| & X_w + X_{|w|w}|w| & X_p + X_{|p|p}|p| & X_q + X_{|q|q}|q| & X_r + X_{|r|r}|r| \\ Y_u + Y_{|u|u}|u| & Y_v + Y_{|v|v}|v| & Y_w + Y_{|w|w}|w| & Y_p + Y_{|p|p}|p| & Y_q + Y_{|q|q}|q| & Y_r + Y_{|r|r}|r| \\ Z_u + Z_{|u|u}|u| & Z_v + Z_{|v|v}|v| & Z_w + Z_{|w|w}|w| & Z_p + Z_{|p|p}|p| & Z_q + Z_{|q|q}|q| & Z_r + Z_{|r|r}|r| \\ K_u + K_{|u|u}|u| & K_v + K_{|v|v}|v| & K_w + K_{|w|w}|w| & K_p + K_{|p|p}|p| & K_q + K_{|q|q}|q| & K_r + K_{|r|r}|r| \\ M_u + M_{|u|u}|u| & M_v + M_{|v|v}|v| & M_w + M_{|w|w}|w| & M_p + M_{|p|p}|p| & M_q + M_{|q|q}|q| & M_r + M_{|r|r}|r| \\ N_u + N_{|u|u}|u| & N_v + N_{|v|v}|v| & N_w + N_{|w|w}|w| & N_p + N_{|p|p}|p| & N_q + N_{|q|q}|q| & N_r + N_{|r|r}|r| \end{bmatrix} \tag{2.28}$$

2.4.3 Restoring Forces and Moments

The buoyant force and the gravitational force generate restoring forces. The buoyant force is modelled to effects the vehicle at its center of buoyancy, denoted as F_B. Similarly, the gravitational force is modelled to effect the vehicle at its center of gravity, denoted as F_G. They act in the earth-fixed frame but they are defined in the body-fixed frame. The center of buoyancy and the center of gravity are denoted as $r_B = [x_B, y_B, z_B]^T$, $r_G = [x_G, y_G, z_G]^T$, respectively.

The weight of vehicle and the buoyancy force are given in Equation 2.29 and Equation 2.30, respectively.

$$W = mg \tag{2.29}$$

$$B = \rho g V \tag{2.30}$$

where m is mass of the vehicle, g is gravity force, ρ is density of the liquid and V is volume of the vehicle. By using the Euler transformation, these forces are defined in the body-fixed frame as

$$F_G(\eta_2) = T_{BE}^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \qquad F_B(\eta_2) = -\, T_{BE}^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \tag{2.31}$$

Buoyancy and gravitational forces and moments vector in the body-fixed coordinate frame is

$$g(\eta) \;=\; \begin{bmatrix} F_G(\eta) \;+\; F_B(\eta) \\ r_G \times F_G(\eta) \;+\; r_B \times F_B(\eta) \end{bmatrix} \tag{2.32}$$

and the expanded form of Equation 2.32 is:

$$g(\eta) = \begin{bmatrix} (W - B)\sin(\theta) \\ -(W - B)\cos(\theta)\sin(\phi) \\ -(W - B)\cos(\theta)\cos(\phi) \\ -(y_G W - y_B B)\cos(\theta)\cos(\phi) + (z_G W - z_B B)\cos(\theta)\sin(\phi) \\ (z_G W - z_B B)\sin(\theta) + (x_G W - x_B B)\cos(\theta)\cos(\phi) \\ -(x_G W - x_B B)\cos(\theta)\sin(\phi) - (y_G W - y_B B)\sin(\theta) \end{bmatrix} \tag{2.33}$$

2.5 Equations of Motion

2.5.1 Non-linear Equations of Motion

After some definitions about kinematics, rigid-body dynamics and hydrodynamic forces and moments, twelve non-linear equations of motion in body-fixed reference frame vector representation can be written as follows:

$$M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{2.34}$$

If the vehicle is at rest or is moving at low speed, M takes the following symmetrical form [17]:

$$M = \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{w}} & -X_{\dot{p}} & mz_G - X_{\dot{q}} & -my_G - X_{\dot{r}} \\ -X_{\dot{v}} & m - Y_{\dot{v}} & -Y_{\dot{w}} & -mz_G - Y_{\dot{p}} & -Y_{\dot{q}} & mx_G - Y_{\dot{r}} \\ -X_{\dot{w}} & -Y_{\dot{w}} & m - Z_{\dot{w}} & my_G - Z_{\dot{p}} & -mx_G - Z_{\dot{q}} & -Z_{\dot{r}} \\ -X_{\dot{p}} & -mz_G - Y_{\dot{p}} & my_g - Z_{\dot{p}} & I_x - K_{\dot{p}} & -I_{xy} - K_{\dot{q}} & -I_{zx} - K_{\dot{r}} \\ mz_G - X_{\dot{q}} & -Y_{\dot{q}} & -mx_G - Z_{\dot{q}} & -I_{xy} - K_{\dot{q}} & I_y - M_{\dot{q}} & -I_{yz} - M_{\dot{r}} \\ -my_G - X_{\dot{r}} & mx_G - Y_{\dot{r}} & -Z_{\dot{r}} & -I_{zx} - K_{\dot{r}} & -I_{yz} - M_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.35}$$

The elements of matrices and their structure changes depending on the mechanical design of the vehicle which is created by using SOLIDWORKS. Following properties are obtained by using SOLIDWORKS.

1. The body-fixed reference frame is coincident with the center of gravity. Thus, $r_G = [0,\ 0,\ 0]^T$ m.

2. The center of buoyancy is $r_B = [0,\ 0,\ -0.015]^T$ m.

3. Moments of inertia I_O:

$$I_O = \begin{bmatrix} 0.292 & 0 & 0.04 \\ 0 & 0.724 & 0 \\ 0.04 & 0 & 0.817 \end{bmatrix} \tag{2.36}$$

According to Equation 2.36, it can be said that the vehicle has *xz*-plane symmetry. Moreover, I_{xz} is much smaller than the diagonal elements of Equation 2.36. Consequently, the vehicle can be assumed to have that it has two-plane symmetry (*xz*-plane and *yz*-plane).

4. Mass of the vehicle in the air is $m = 15.5\ kgs$. By using Equation 2.29

$$W = 152.0550\ N \tag{2.37}$$

5. Volume of the vehicle is $0.02\ m^3$. By using Equation 2.30

$$B = 192.5703\ N \tag{2.38}$$

(ρ is taken as $1000\ kg/m^3$)

Due to the vehicle properties, M and D matrices have the following forms:

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & -Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{w}} & 0 & -Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & -Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & -Y_{\dot{r}} & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.39}$$

$$D(\upsilon) = -\begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{|v|v}|v| & 0 & 0 & 0 & Y_r + Y_{|r|r}|r| \\ 0 & 0 & Z_w + Z_{|w|w}|w| & 0 & Z_q + Z_{|q|q}|q| & 0 \\ 0 & 0 & 0 & K_p + K_{|p|p}|p| & 0 & 0 \\ 0 & 0 & M_w + M_{|w|w}|w| & 0 & M_q + M_{|q|q}|q| & 0 \\ 0 & N_v + N_{|v|v}|v| & 0| & 0 & 0 & N_r + N_{|r|r}|r| \end{bmatrix} \tag{2.40}$$

2.5.2 Linear Time-Invariant Equations of Motion

Linear time-invariant (LTI) equations of motion are used for the autopilot design. Linearization is finding linear approximation of any system around equilibrium (trim) points. [31], [37].

For Equation 2.11, equilibrium points can be chosen as

$$\begin{aligned} \upsilon_0 &= [u_0,\ v_0,\ w_0,\ p_0,\ q_0,\ r_0] \\ \eta_0 &= [x_0,\ y_0,\ z_0,\ \phi_0,\ \theta_0,\ \psi_0] \end{aligned} \tag{2.41}$$

Perturbed equations for υ, η, and τ are

$$\upsilon = \upsilon_0 + \delta\upsilon \xrightarrow{d/dt} \dot{\upsilon} = \delta\dot{\upsilon} \tag{2.42a}$$

$$\eta = \eta_0 + \delta\eta \xrightarrow{d/dt} \dot{\eta} = \delta\dot{\eta} \tag{2.42b}$$

$$\tau = \tau_0 + \delta\tau \xrightarrow{d/dt} \dot{\tau} = \delta\dot{\tau} \tag{2.42c}$$

When Equation 2.42 is applied to Equation 2.11, the linearized equations of motion are obtained according to the following approximation [17]:

$$M\delta\dot{\upsilon} + \left.\frac{\partial C(\upsilon)\upsilon}{\partial\upsilon}\right|_{\upsilon_0}\delta\upsilon + \left.\frac{\partial D(\upsilon)\upsilon}{\partial\upsilon}\right|_{\upsilon_0}\delta\upsilon + \left.\frac{\partial g(\eta)}{\partial\eta}\right|_{\eta_0}\delta\eta = \delta\tau \tag{2.43}$$

Substituting initial conditions, $\dot{\eta_0} = T_{BE}(\eta_0)\upsilon_0$, in Equation 2.43 and neglecting second order terms yield

$$\delta\dot{\eta} = T_{BE}(\eta_0 + \delta\eta)\delta\upsilon + \left[T_{BE}(\eta_0 + \delta\eta) - T_{BE}(\eta_0)\right]\upsilon_0 \tag{2.44}$$

In order to construct the linear-time invariant equations of motion, it is assumed that the vehicle has only non-zero linear velocity component along x direction; so $v_0 = w_0 = 0$. The weight and buoyancy distribution of the vehicle forces the vehicle to return back to the zero pitch and zero roll position. Therefore, it is assumed that $p_0 = q_0 = \phi_0 = \theta_0 = 0$. It is also assumed that $r_0 = 0$ and $\psi_0 = constant$ in steady-state. When the vehicle moves at low speed, linear drag is dominant contrary to the fact that quadratic terms are dominant at high speeds

[40]. Consequently, the following LTI equations of motion can be obtained for the vehicle this thesis is based on.

$$\dot{x} = Ax + Bu \tag{2.45}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -M^{-1}(C+D) & -M^{-1}G \\ J & 0_{6\times 6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} M^{-1} \\ 0_{6\times 6} \end{bmatrix} u \tag{2.46}$$

where

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & -Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{w}} & 0 & -Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & -Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & -Y_{\dot{r}} & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.47}$$

$$C = \begin{bmatrix} 0 & C_{12} \\ -C_{12}^T & C_{22} \end{bmatrix} \quad \text{where} \quad C_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & (m - X_{\dot{u}})u_0 \\ 0 & -(m - X_{\dot{u}})u_0 & 0 \end{bmatrix} \quad C_{22} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.48}$$

$$D = -\begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & Y_r \\ 0 & 0 & Z_w & 0 & Z_q & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & 0 & 0 & N_r \end{bmatrix} \tag{2.49}$$

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & (W-B) & 0 \\ 0 & 0 & 0 & -(W-B) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -z_B B & 0 & 0 \\ 0 & 0 & 0 & 0 & -z_B B & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.50}$$

$$J = \begin{bmatrix} \tilde{J} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \qquad \text{where} \qquad \tilde{J} = \begin{bmatrix} \cos(\psi_0) & -\sin(\psi_0) & 0 \\ \sin(\psi_0) & \cos(\psi_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.51}$$

2.6 Conclusion

In this chapter, the mathematical model of the underwater vehicles has been obtained in general first, next the model is specialized for the vehicle DST-R-100-4 ROV. Twelve non-linear equations of motions of the vehicle in 6 DOF are obtained mechanical properties of the vehicle. Hydrodynamic forces and moments consist of added mass and inertia, hydrodynamic damping forces and restoring forces. As a summary, added mass is a virtual mass on the vehicle during its operation and it is proportional to the acceleration of the vehicle. Hydrodynamic damping forces can be divided into four parts; but skin friction and vortex shedding damping effect the vehicle mostly. Restoring forces and moments depend on the weight and buoyancy forces and their point of influence on the vehicle. This section is ended with an LTI model that is obtained by using linearization of the non-linear model around special trim point.

CHAPTER 3

MODEL PARAMETERS DERIVATION

3.1 Introduction

In order to complete the mathematical model of DST-R-100-4, hydrodynamic coefficients should be computed. Drag coefficients for motion along *x*, *y*, and *z* axis are calculated by using results of CFD analysis. Remaining drag coefficients are calculated with the help of Seamor ROV parameters[45]. Strip theory is used for calculating added mass coefficients.

3.2 Drag Coefficients Calculation

As mentioned in Chapter 2, drag force has two parts as linear and quadratic. For example, if the vehicle goes only in *x* direction, drag force acting on the vehicle can be written as shown in Equation 3.1. It looks like second order polynomial, therefore drag force can be also written as polynomial.

$$\textbf{\textit{Case 1}} \quad \begin{cases} F_{u_{drag}} &= X_{|u|u}\,|u|\,u \;+\; X_u\,u \\ y_1 &= a_1\,x^2 \;+\; b_1\,x + \;c_1 \end{cases} \tag{3.1}$$

When the vehicle operates at low speed, linear drag is dominant. For this reason, drag force can be also modelled for DST-R-100-4 as first order polynomial. Because of the fact that Equation 3.1 can be rewritten as:

$$\textbf{\textit{Case 2}} \quad \begin{cases} F_{u_{drag}} &= X_u\,u \\ y_2 &= a_2\,x \;+\; b_2 \end{cases} \tag{3.2}$$

CFD analysis is done by using Solidworks. The drag force acting on the vehicle is obtained for different velocities such as 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, and 1.5 m/s. According to CFD analysis results, *surge*, *sway* and *heave* motion drag coefficients are calculated by using MATLAB Curve Fitting Toolbox,(`cftool`).

3.2.1 Surge Motion Drag Coefficients

Data obtained from CFD analysis is shown as square boxes in Figure 3.1. The blue line indicates the curve that fits the data and the red line indicates straight line that fits the data. When obtaining polynomial equations, `cftool` uses linear least squares method. Obtained polynomial equations are as follows.

$$\begin{aligned} y_1 &= -24.7\,x^2 + (-1.576)\,x + (0.258) \\ y_2 &= -34.58\,x + 3.793 \end{aligned} \tag{3.3}$$

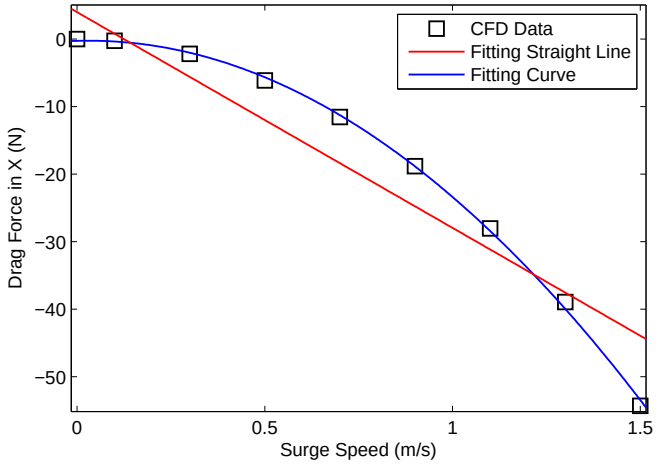


Figure 3.1: CFD data, fitted curve and fitted straight line for surge motion

According to Equation 3.3, drag coefficients for motion along x- axis direction are shown in Table 3.1.

Table 3.1: Surge motion drag coefficients for Case 1 and Case 2

Case 1	$X_{\|u\|u} = -24.7\,kg/m \qquad X_u = -1.576\,kg/s$
Case 2	$X_u = -34.58\,kg/s$

3.2.2 Sway Motion Drag Coefficients

By using the approximation explained in Section 3.2.1, drag coefficients along *y*-axis direction can be found by using Equation 3.4.

$$\begin{aligned} y_1 &= -53.38\,x^2 + (-1.046)\,x + 0.002534 \\ y_2 &= -35.93\,x + 2.839 \end{aligned} \tag{3.4}$$

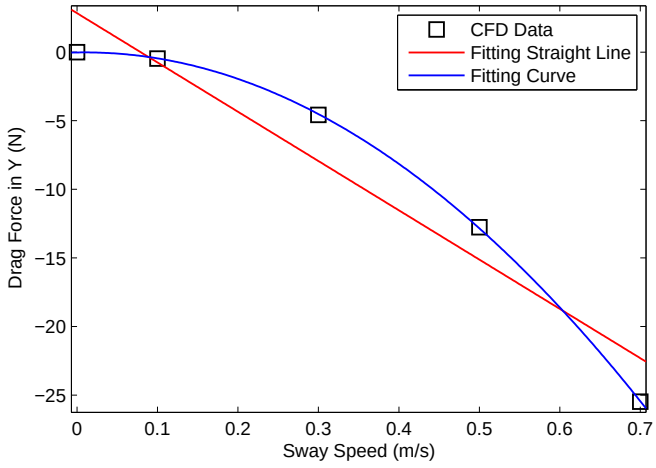


Figure 3.2: CFD data, fitted curve and fitted straight line for sway motion

According to Equation 3.4, the coefficients can be written as shown in Table 3.2.

Table 3.2: Sway motion drag coefficients for Case 1 and Case 2

Case 1	$Y_{\|v\|v} = -53.38\,kg/m \qquad Y_v = -1.046\,kg/s$
Case 2	$Y_v = -35.93\,kg/s$

3.2.3 Heave Motion Drag Coefficients

Finally, drag coefficients along *z*-axis direction are found with the same method.

$$\begin{aligned} y_1 &= -74.09\,x^2 + (-0.03852)\,x + 0.001681 \\ y_2 &= -47.59\,x + 2.071 \end{aligned} \tag{3.5}$$

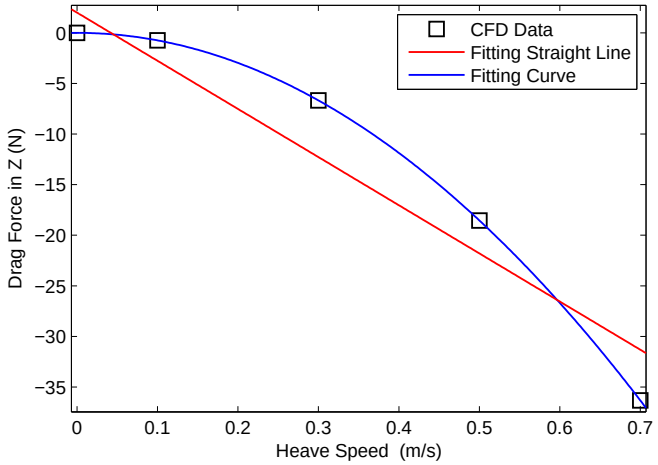


Figure 3.3: CFD data, fitted curve and fitted straight line for heave motion

According to Equation 3.5, the following Table 3.3 is obtained.

Table 3.3: Heave motion drag coefficients for Case 1 and Case 2

Case 1	$Z_{\|w\|w} = -74.09\,kg/m \qquad Z_w = -0.03852\,kg/s$
Case 2	$Z_w = -47.59\,kg/s$

3.3 Added Mass Coefficients Calculation

In literature, added mass coefficients are determined by using strip theory [33]. Basically, the vehicle is separated into number of strips, and added mass for every 2D strip is computed. In order to obtain 3D added mass coefficients, the 2D added mass is summed over the length of the body [17].

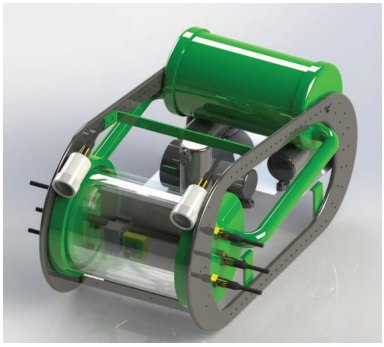


Figure 3.4: Technical drawing of DST-R-100-4 ROV

As shown in Figure 3.4, the majority of the volume of the vehicle consists of two cylinders, one of them is located at front of the vehicle, and the other one located at aft. Because of this reason, added mass coefficients are calculated based on this cylinders, separately. Final added mass coefficients are obtained by adding two of them.

For a submerged slender body, there are several equations for calculating added mass coefficients [17]. By using them, the added mass coefficients can be calculated. For added mass coefficients related with the linear acceleration of the body, the equation can be written as [8], [17], [42]:

$$X_{\dot{u}} = -\int_{-L/2}^{L/2} A_{11}^{(2D)}(y,z)dx \tag{3.6}$$

$$Y_{\dot{v}} = -\int_{-B/2}^{B/2} A_{22}^{(2D)}(x,z)dy \tag{3.7}$$

$$Z_{\dot{w}} = -\int_{-H/2}^{H/2} A_{33}^{(2D)}(x,y)dz \tag{3.8}$$

In order to obtain added mass coefficients in roll, pitch and yaw, following equations can be used:

$$K_{\dot{p}} \cong -\left(\int_{-B/2}^{B/2} y^2 A_{33}^{(2D)}(x,z)dy \; + \; \int_{-H/2}^{H/2} z^2 A_{22}^{(2D)}(x,y)dz \right) \tag{3.9}$$

$$M_{\dot{q}} \cong -\left(\int_{-L/2}^{L/2} x^2 A_{33}^{(2D)}(y,z)dx \; + \; \int_{-H/2}^{H/2} z^2 A_{11}^{(2D)}(x,y)dz \right) \tag{3.10}$$

$$N_{\dot{r}} \cong -\left(\int_{-B/2}^{B/2} y^2 A_{11}^{(2D)}(x,z)dy \; + \; \int_{-L/2}^{L/2} x^2 A_{22}^{(2D)}(y,z)dx \right) \tag{3.11}$$

Cylinder sizes are shown in Figure 3.5, and 2D added mass coefficients are shown in Table 3.4.

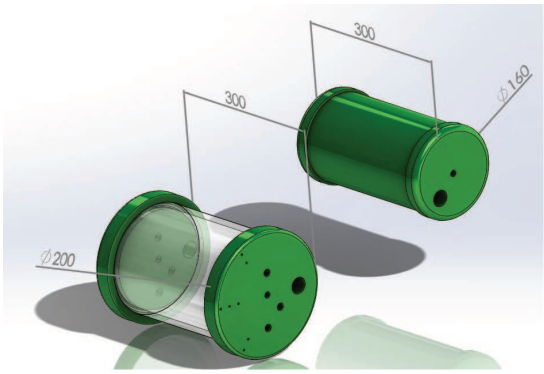
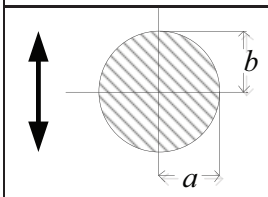
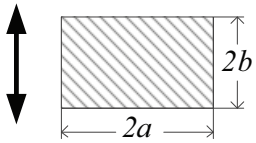


Figure 3.5: Cylinder sizes

Table 3.4: 2D added mass coefficients for some bodies

Cross-Sectional Forms Direction of Motion		2D Added Mass Coefficients
	$\frac{a}{b} = 1$ (circle)	$\pi\rho a^2$
	$\frac{a}{b} = 2$	$1.36\pi\rho a^2$
	$\frac{a}{b} = 5$	$1.21\pi\rho a^2$
	$\frac{a}{b} = 10$	$1.14\pi\rho a^2$

By using Equations 3.6 - 3.11, and 2D added mass coefficients which are defined in Table 3.4 following 3D added mass coefficients are found. Note that, $L = 0.3\ m$ and $H = B = 0.2\ m$ for front cylinder, and $L = 0.3\ m$ and $H = B = 0.16\ m$ for aft cylinder.

Table 3.5: Added mass coefficients

$X_{\dot{u}} = -43.2267\,kg$	$Y_{\dot{v}} = -15.8276\,kg$	$Z_{\dot{w}} = -43.2267\,kg$
$K_{\dot{p}} = -0.5627\,kg\,m^2/rad$	$M_{\dot{q}} = -0.2473\,kg\,m^2/rad$	$N_{\dot{r}} = -0.5627\,kg\,m^2/rad$

There is no calculation methods for off-diagonal terms of added mass matrix and drag forces. They can be obtained from experimental data. Not only for this reason, but also since the effect of these terms is very small, they can be taken as zero.

3.4 Conclusion

In this chapter, the methods are explained for calculating numerical values of added mass coefficients and drag coefficients. Since the majority volume of the vehicle consists of two cylinders, the added mass coefficients are calculated based on them by using strip theory. For drag coefficients calculation, CFD analysis results are used. Because of the fact that drag force equation can be written as polynomials, drag coefficients are obtained by using curve fitting. More details about added mass, and calculation methods of them can be found in [8], [17], [33] and [42].

CHAPTER 4

AUTOPILOT DESIGN

4.1 Introduction

An autopilot is a control system that keeps the vehicle on a steady-state course without assistance from human. There are usually more than one controllers constructing the autopilot during its operation. An autopilot generates its own commands for the vehicle in order to successfully execute the guidance commands.

In literature [17], [18], [20] and [43], autopilot of an underwater vehicles consists of heading, speed and depth controllers. For this reason, controller are designed for speed, heading angle and depth.

In this chapter, the controllers are designed by using feedback control method. Firstly, the thruster model and thruster position on the vehicle are determined. Then, linear-time invariant mathematical model is examined by applying different inputs in order to obtain maximum values of surge, sway and heave speed and yaw rate. The controllers are designed by using LTI model. However, response of the controllers are examined by using both the LTI model and the non-linear model.

4.2 Thruster Model

The vehicle has four identical thrusters in order to achieve motion and control. In literature [22], the thruster model is divided into three parts such as; motor model, fluid model and propeller mapping. General view of this structure is shown in Figure 4.1.

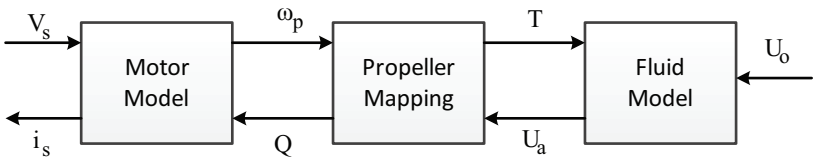


Figure 4.1: General view of the thruster model

4.2.1 Motor Model

The motor is modelled based on DC servo-motor permanent magnet type.

$$\dot{\omega}_m = -K_1\omega_m + K_2V_s - K_hQ \tag{4.1}$$

$$\omega_p = \omega_m/N \tag{4.2}$$

where ω_m is motor rotational rate, V_s is supplied voltage, Q is propeller torque. If the motor has gearbox, ω_m differs from propeller rotational rate which is denoted by ω_p and N denotes the gearbox reduction ratio. K_1 and K_2 depend on motor characteristics and K_h relates to the motor deceleration caused by propeller hydrodynamic torques loading.

4.2.2 Propeller Mapping

There is a relation between motor rotational rate and propeller speed (U_p) such as:

$$U_p = 0.7R\omega_m/N \tag{4.3}$$

where R is propeller radius. Since section average flow velocity, U_a, is perpendicular to propeller velocity, the total velocity is obtained from Equation 4.4.

$$V = \sqrt{U_p^2 + U_a^2} \tag{4.4}$$

The effective angle of attack (α_e) occurs between V and the propeller can be modelled as shown in Equation 4.5.

$$\alpha_e = (\pi/2 - p) - \arctan\left(\frac{U_a}{U_p}\right) \tag{4.5}$$

According to α_e, proposed lift and drag force coefficients (denoted by $C_{L_{max}}$ and $C_{D_{max}}$, respectively) changes. Lift and drag forces on the blades can be written as:

$$F_L = 0.5\rho V^2 A C_{L_{max}} \sin(2\alpha_e) \tag{4.6}$$

$$F_D = 0.5\rho V^2 A C_{D_{max}}(1 - \cos(2\alpha_e)) \tag{4.7}$$

By using Equation 4.6 and Equation 4.7, thrust force (T) and torque (Q) can be written as:

$$T = F_L(\cos\theta) - F_D(\sin\theta) \tag{4.8}$$

$$Q = 0.7R(F_L(\sin\theta) + F_D(\cos\theta)) \tag{4.9}$$

where

$$\theta = p - \alpha_e$$

4.2.3 Fluid Model

Fluid model represents the fluid dynamics. According to changes on angular rate of the blade, there are some lags in the axial components of the flow. Because of the fact that inlet flow effect occurs on the blade. The momentum equation is applied to a control volume surrounding the inlet flow.

$$T = (\rho AL\gamma)\dot{U}_a + (\rho A\Delta\beta)\bar{U_a}|\bar{U_a}| \tag{4.10}$$

where L is tunnel length, $\Delta\beta$ is momentum coefficient, and A is tunnel cross-sectional area. Equation 4.10 can be rearranged as:

$$\dot{U}_a = -K_4 K_3^{-1}\bar{U_a}|\bar{U_a}| + K_3^{-1}T \tag{4.11}$$

where

$$K_3 = \rho AL\gamma \qquad K_4 = \rho A\Delta\beta \qquad \bar{U_a} = (U_a - U_0)$$

More details about thruster model can be found in [22] and [4].

Thrusters on the vehicle have a thrust about 2.2 kgf, which is approximately 22 $Newton$. Therefore, maximum 22 N thrust can be applied to the vehicle.

4.3 Obtaining Maximum Values of u, v, w and r

The LTI mathematical model is obtained in Chapter 2 and model parameters are derived in Chapter 3. By using them, maximum values of surge, sway, heave speed and yaw rate are

obtained under different input. Input of the model is chosen as thrust, so maximum values of input depends on position and orientation of the thrusters.

As mentioned before, DST-ROV-100-4 has two forward (T_1 and T_2), one horizontal (T_3) and one vertical T_4 thrusters as shown in Figure 4.2. It also means that number of inputs of the system is four. According to locations of thrusters, forces and moments acting on the vehicle can change.

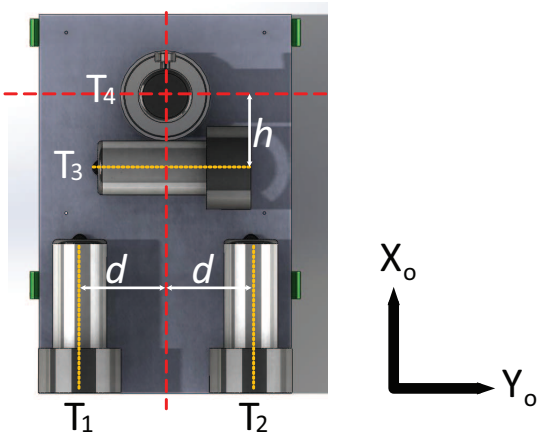


Figure 4.2: Thrusters position on the vehicle

T_1 and T_2 create forces in forward/backward direction and yaw moment. If they work at the same rpm, force along X_0 is obtained, but if there is a difference between rpm of thrusters, yaw moment occurs since there is a moment arm which is denoted by d. T_3 creates force in horizontal direction and yaw moment. Moment arm for horizontal thruster is denoted by h. Vertical thruster is located at the cg of the vehicle, so T_4 only creates force in vertical direction. The vector representation of forces and moments produced by thrusters can be written as:

$$\begin{bmatrix} \sum X \\ \sum Y \\ \sum Z \\ \sum K \\ \sum M \\ \sum N \end{bmatrix} = \begin{bmatrix} T_1 + T_2 \\ T_3 \\ T_4 \\ 0 \\ 0 \\ T_1 d - T_2 d + T_3 h \end{bmatrix} \tag{4.12}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -M^{-1}(C+D) & -M^{-1}G \\ J & 0_{6\times6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} M^{-1} \\ 0_{6\times6} \end{bmatrix} \begin{bmatrix} u \\ 0_{6\times6} \end{bmatrix} \tag{4.13}$$

In Equation 4.13, u denotes input of the system and it can be written as:

$$u = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -d & d & h & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{4.14}$$

where

$$d = 0.095\ m \qquad h = 0.067\ m$$

By using simulink model is shown in Figure 4.3, response of the system can be examined under different inputs.

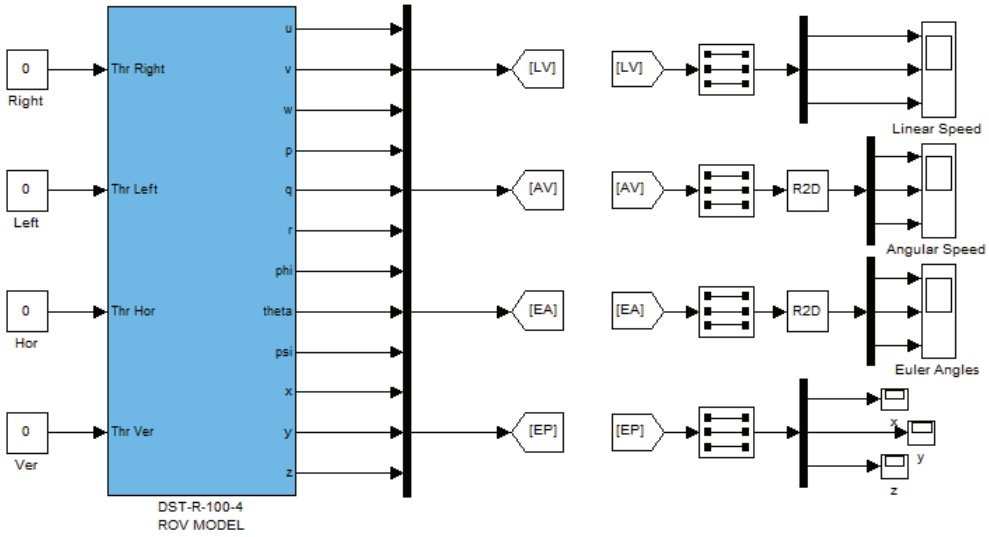


Figure 4.3: Simulink model of DST-R-100-4

There are several test cases in order to examine response of the system and determine the maximum speed of the vehicle. As mentioned before, thrusters produce maximum 22 N and it uses as inputs for determining maximum speeds.

Test Case 1: $T_1 = 22\ T_2 = 22\ T_3 = 0\ T_4 = 0$

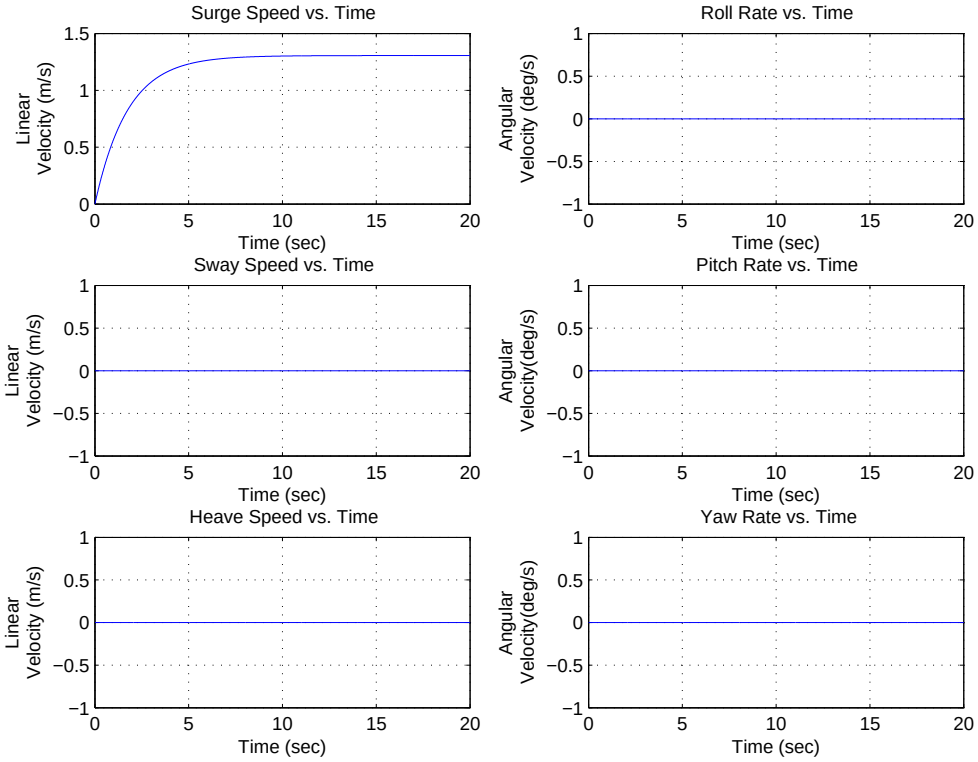


Figure 4.4: Linear and angular velocities for test case 1

As shown in Figure 4.4, maximum surge speed is 1.3 m/s. Remaining linear velocities and angular velocities equal zero. For this reason, the system is stable.

Test Case 2: $T_1 = 0\ T_2 = 0\ T_3 = 22\ T_4 = 0$

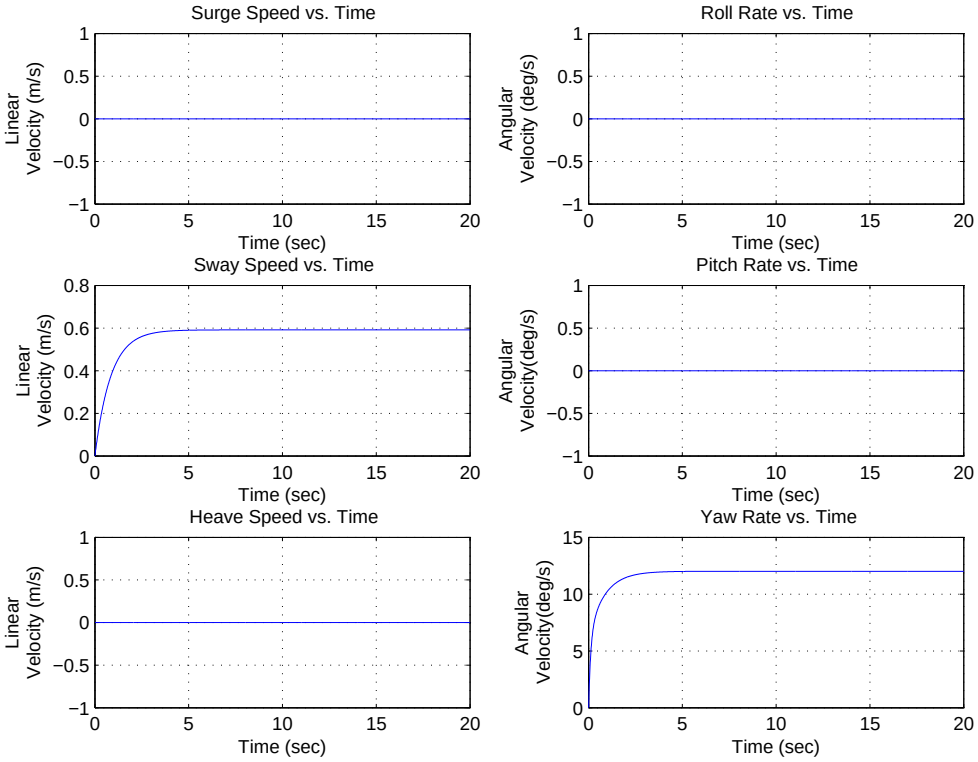


Figure 4.5: Linear and angular velocities for test case 2

As shown in Figure 4.5, maximum sway speed is 0.55 m/s. Since there is a moment arm h, the vehicle has yaw moment so that yaw angle occurs. Both linear and angular velocities converge a constant value. For this reason, it can be said that the system is stable when T_3 thruster works.

Test Case 3: $T_1 = 0$ $T_2 = 0$ $T_3 = 0$ $T_4 = 22$

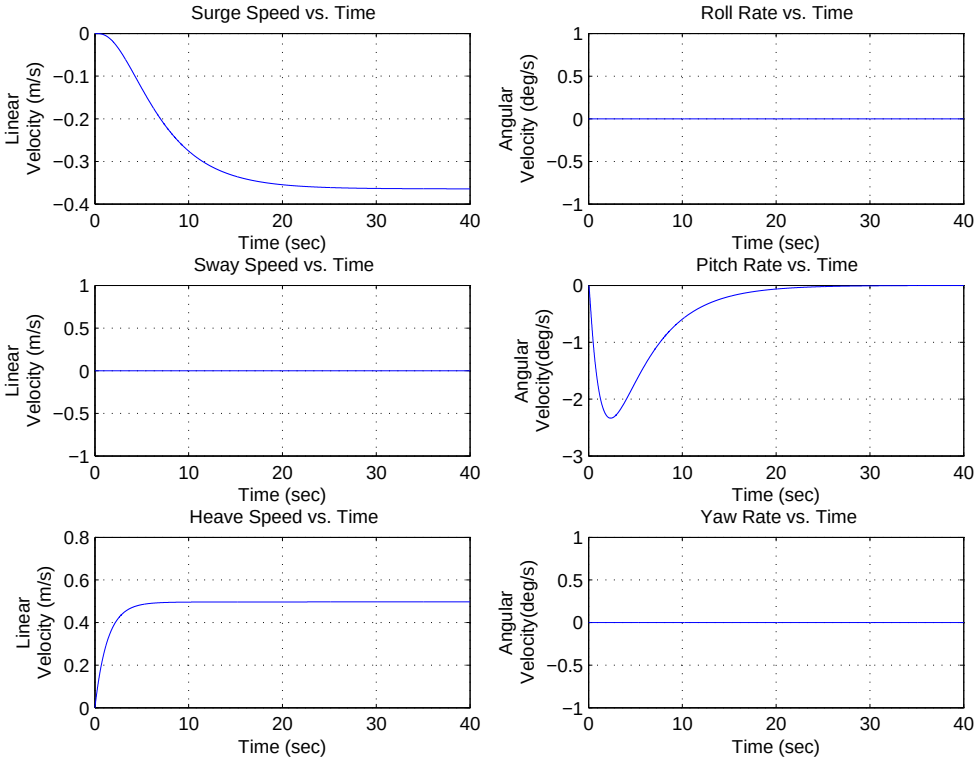


Figure 4.6: Linear and angular velocities for test case 3

As shown in Figure 4.6, maximum heave speed is 0.5 m/s. Also, surge speed occurs and converges to a constant value when vertical thruster works. This is also observed during pool test. It is assumed that the drag force is identical for cylinders, but front cylinder's cross-sectional area is bigger than the aft one. It means that drag force acting on front cylinder is greater than the aft cylinder. This may be a cause of pitch angle for this case. Moreover, the system is stable under T_4 because of the fact that all velocities converge to a constant value.

Test Case 4: $T_1 = 22\ T_2 = -22\ T_3 = 0\ T_4 = 0$

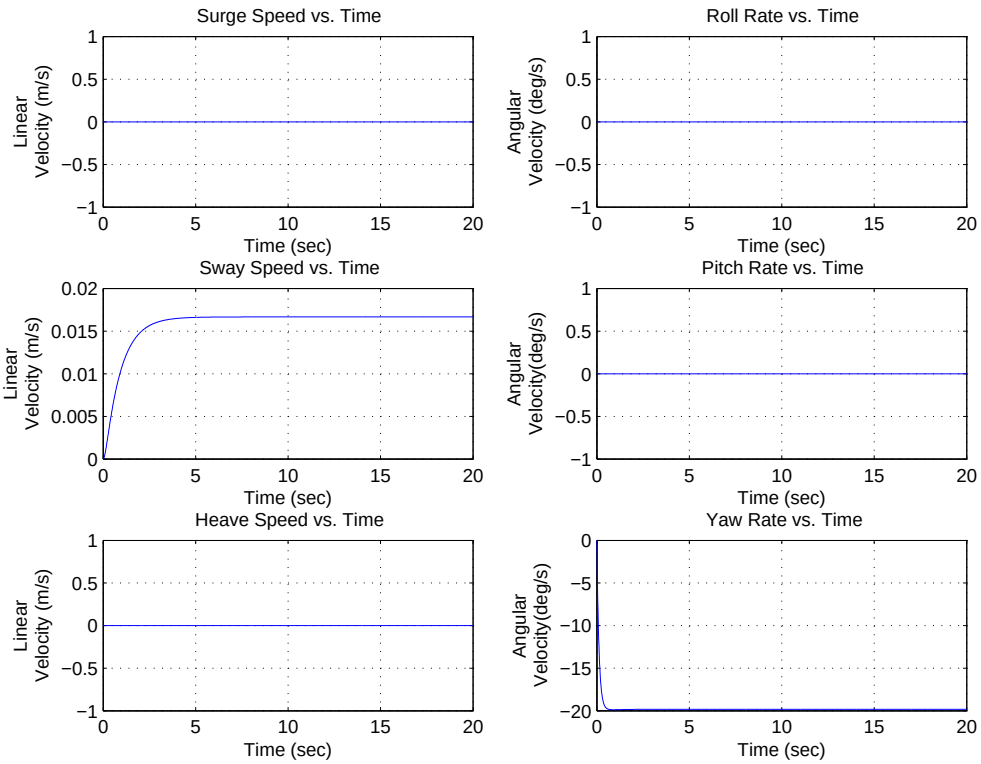


Figure 4.7: Linear and angular velocities for test case 4

In order to turn the vehicle around itself, right thruster and left thruster should work at same rpm, but their directions are opposite. According to Figure 4.7, yaw rate converges a constant value and other velocities are either zero or so small. In addition, maximum yaw rate is 20 deg/s.

Test Case 5: $T_1 = 0\ T_2 = 22\ T_3 = 0\ T_4 = 0$

For this test case, only right thruster works and it produces forward thrust and yaw moment.

As a result of tests, maximum value for u, v, w, r is obtained as shown in Equation 4.15. It is also shown that the vehicle thruster configuration is adequate for different motion types. For example, by using forward thruster and horizontal thruster, the vehicle moves along Y-axis. Moreover, test case 5 shows that the vehicle moves both X and Y direction by using only one

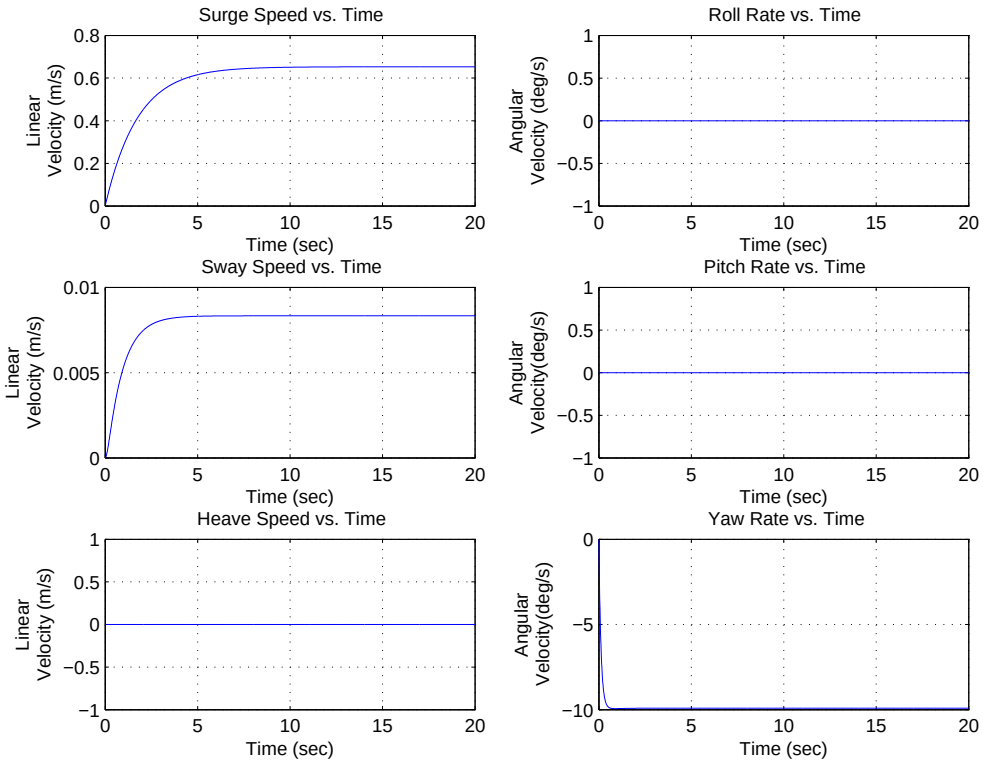


Figure 4.8: Linear and angular velocities for test case 5

forward thruster.

$$u_{max} = 1.3\,m/s, \quad v_{max} = 0.55\,m/s, \quad w_{max} = 0.5\,m/s, \quad r_{max} = 20\,deg/s \tag{4.15}$$

4.4 Controller Design

Proportional - integral - derivative controller is commonly used feedback controller. PID controller calculates error between reference and output of the system. The main objective of the PID is to minimize this error by adjusting control input. For DST-R-100-R, control inputs are T_1, T_2, T_3 and T_4. A block diagram of the PID controller is shown in Figure 4.9.

As shown in Figure 4.9, the control input of the system (u_c) can be written as:

$$u_c(t) = K_p e(t) + K_i \int_0 \tau\Big(e(\tau)d\tau\Big) + K_d \frac{d\big(e(t)\big)}{dt} \tag{4.16}$$

where

$$e(t) = u(t) - y(t)$$

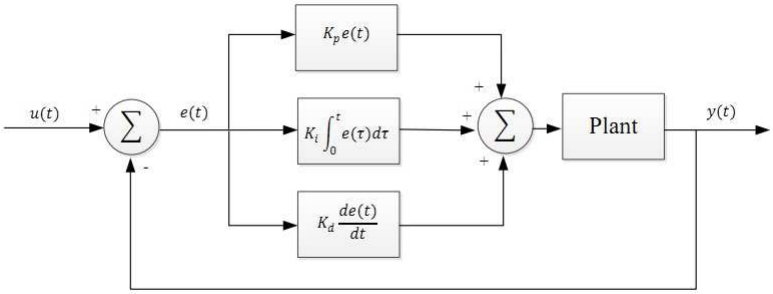


Figure 4.9: Block diagram of the PID controller

PID controller has three constant parameters such as K_p, K_i and K_d. They are also called as PID gains. K_p is proportional gain that is effective in decreasing both rise time and steady-state error, but it increases overshoot. K_i is integral gain that is mostly effective in eliminating steady-state error. However, it increases the settling time and overshoot. Lastly, K_d is derivative gain that decreases overshoot and the settling time [34].

In this thesis, PID gains are obtained from auto-tuning options of MATLAB PID block. These values are used as initial guesses for MATLAB *Response Optimization Toolbox*. Final gain values for PID controller are found by the toolbox.

4.4.1 Speed Controller

As mentioned before, DST-R-100-4 ROV has four velocity components such as u, v, w and r. For each one, PID controller is designed independently. Note that, there is a *Force Distribution Law* block as shown in Figure C.1. It is transition block between desired forces and thruster inputs. Since lateral movement of the vehicle can be obtained by using forward thrusters, and horizontal thruster, they can be managed logically. In this thesis, horizontal thruster is combined with forward thrusters only for producing sway speed. In order to make force distribution law applicable for all ROVs, only forward thrusters are used since many ROVs do not have a horizontal thruster.

Forward thrusters can be managed as follows. Note that, it is assumed that thruster rpm is proportional with generated force.

- *Only forward force is commanded:* For this situation, right and left thrusters work at the same rpms and the same direction. Their directions change related with the commanded force direction.

- *Only yaw moment is commanded:* For this situation, right and left thrusters work at the same rpms, but their directions are opposite to each other. According to commanded yaw moment direction, their directions change.

- *Both yaw moment and forward thrust are commanded:* For this situation, right and left thrusters work at different rpms. For example, for the vehicle to turn around a circle in clockwise direction, $T_{1_{rpm}}$ must be greater than $T_{2_{rpm}}$. Mathematically, this relation can be written as follows.

$$\begin{aligned} T_{1_{rpm}} &= k \\ T_{2_{rpm}} &= k - \Gamma k \end{aligned} \tag{4.17}$$

  where

$$0 < \Gamma < 2 \tag{4.18}$$

  The diameter of circle inversely proportional with thruster weight, denoted by Γ.

Note that, in order to obtain yaw moment, T_3 is not used since most ROVs do not have a horizontal thruster. Force distribution law is used for not only making force distribution simple, but also making the law applicable for more ROVs.

The controllers are designed by using the LTI model, firstly. Next, response of the controllers are examined by using both the LTI model and non-linear model.

Surge Speed Controller

As shown in Figure 4.10, error signal is obtained from the difference between the desired and actual surge speed. According to this error, PID controller generates reference force signal.

In Figure 4.11, responses of surge speed controller and right and left thrusters are shown by using both the LTI and the non-linear model. For surge speed controller, the reference signal is set to $1\,m/s$, and the controller reaches the desired value in 4 $secs$.

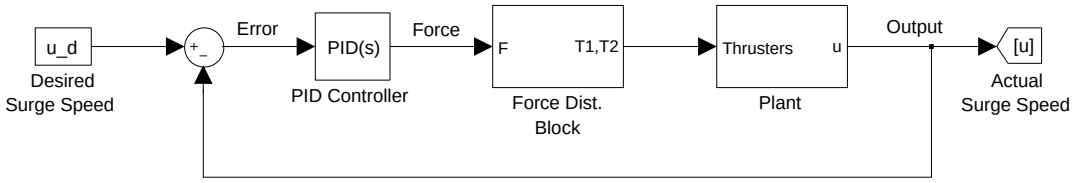


Figure 4.10: PID controller block diagram for surge speed controller

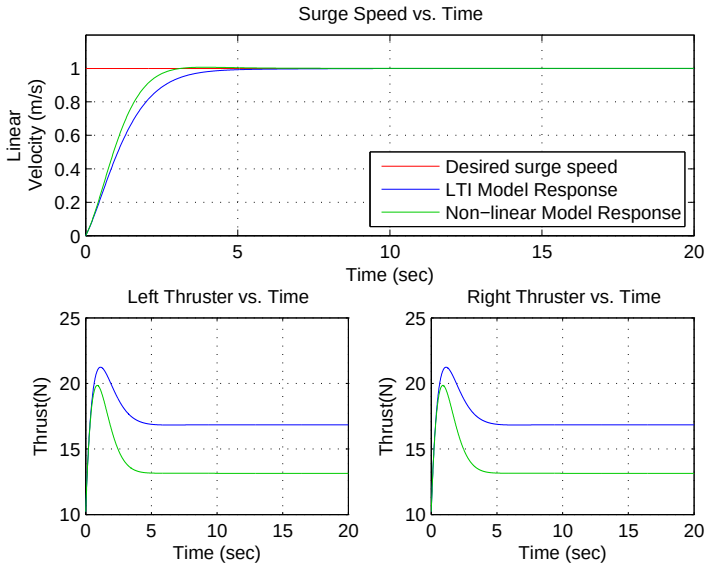


Figure 4.11: Surge speed controller and right and left thrusters response

Sway Speed Controller

The error signal for sway speed controller is obtained from the difference between the plant output and the desired signal. Reference value for sway speed controller is 0.5 m/s. Responses of when using the two model is shown in Figure 4.13.

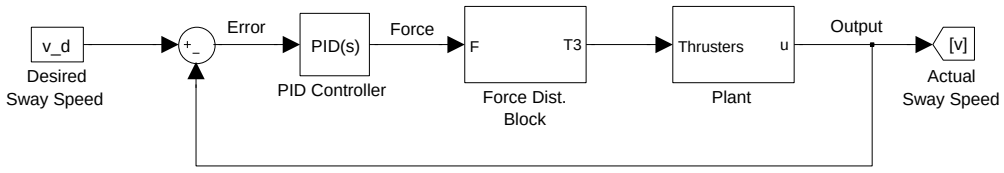


Figure 4.12: PID controller block diagram for sway speed controller

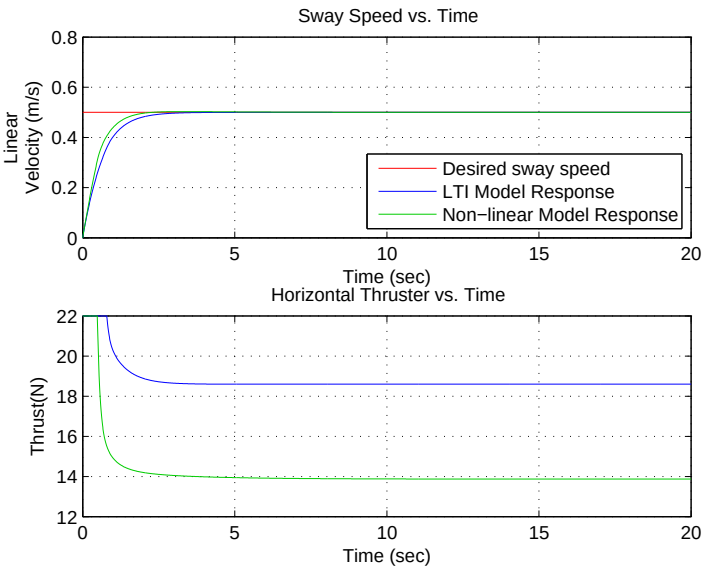


Figure 4.13: Sway speed controller and horizontal thruster response

As shown in Figure 4.13, controller reaches the desired sway speed in about 3 *secs*.

Heave Speed Controller Design

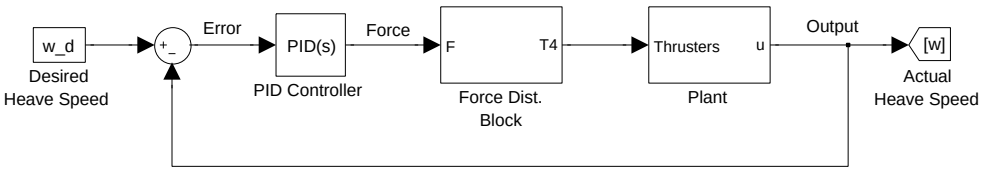


Figure 4.14: PID controller block diagram for heave speed controller

For heave speed controller, the reference value is set to 0.3 m/s. Responses of the vertical thruster and the controller are shown in Figure 4.15. The settling time for this controller is about 3.5 *secs* for both models.

Yaw Rate Controller Design

Error signal is generated from the difference between the actual and the desired yaw rate. Due to this error, PID controller generates plant input.

Response of yaw rate controller and thrusters are shown in Figure 4.17 when the LTI model

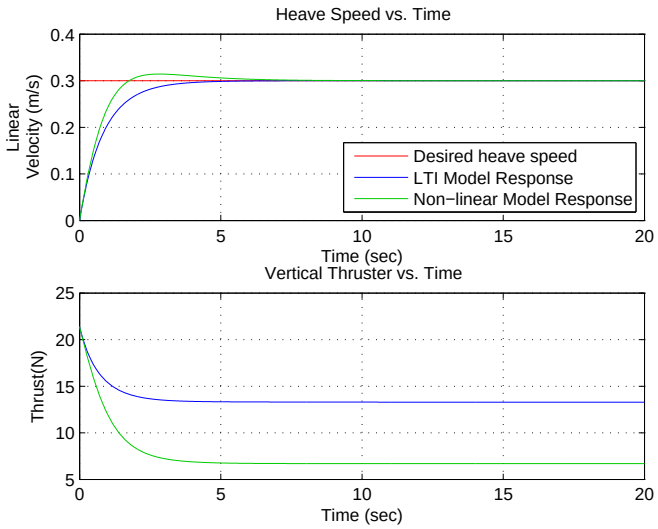


Figure 4.15: Heave speed controller and vertical thruster response

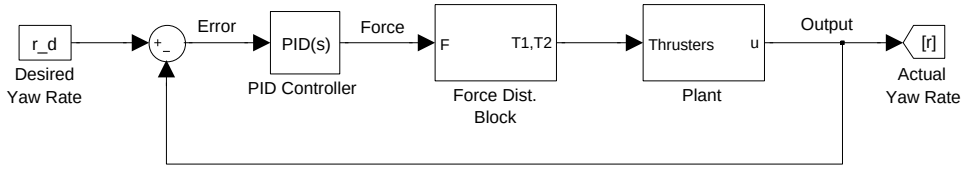


Figure 4.16: PID controller block diagram for yaw rate controller

is used. The reference value is set to 5 deg/s and the settling time is 0.5 *secs*. When the non-linear model is used, the settling time is 3 *secs*.

When designing speed controllers as well as other controllers, it is considered that there will be no overshoot at controller responses. Since overshoot forces to the thruster to change their rotation direction, it causes vibration on thrusters. Changing motor direction also draws higher current than changing its rpm. Thus, it is not good for both energy consumption and thruster endurance.

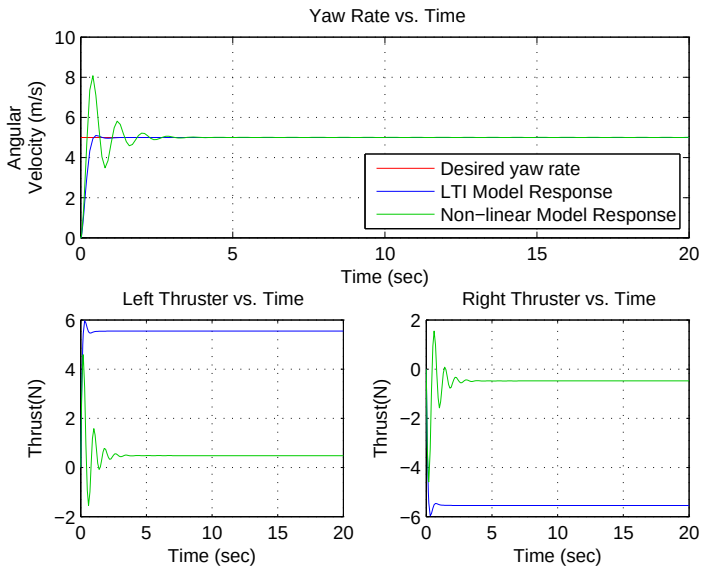


Figure 4.17: Yaw rate controller and right and left thrusters response

4.4.2 Heading Controller

In order to design heading controller, there are two methods such as cascaded PID controller design and single loop PID controller. Cascaded PID controller contains two PID controllers called as outer PID controller and inner PID controller. Output of the outer PID controller provides reference signal for the inner one as shown in Figure 4.18. The inner loop, whose response is faster than the outer one, consists of yaw rate controller which is described in Section 4.4.1.

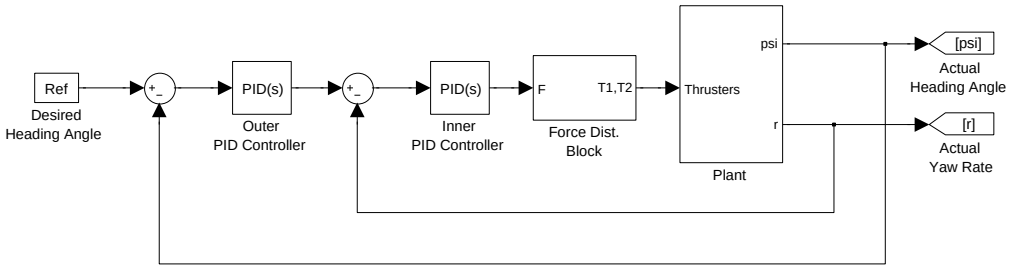


Figure 4.18: Cascade PID controller block diagram for heading angle

Additionally, heading controller can be designed as shown in Figure 4.19 in such a way that

there is only one PID controller. It generates input signal for the plant depending on the difference between desired and the actual heading angle. This model is used for heading angle controller because steady-state error can not be eliminated from the cascaded one. In addition, for high input value, the cascaded one displays high overshoot.

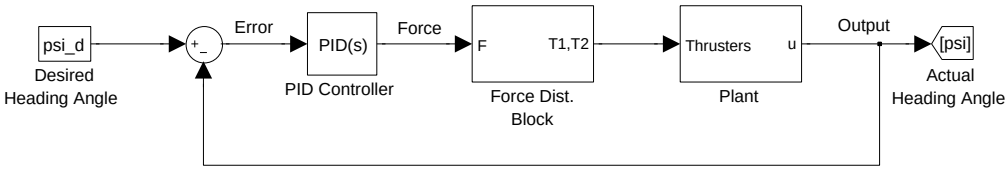


Figure 4.19: Single loop PID controller block diagram for heading angle

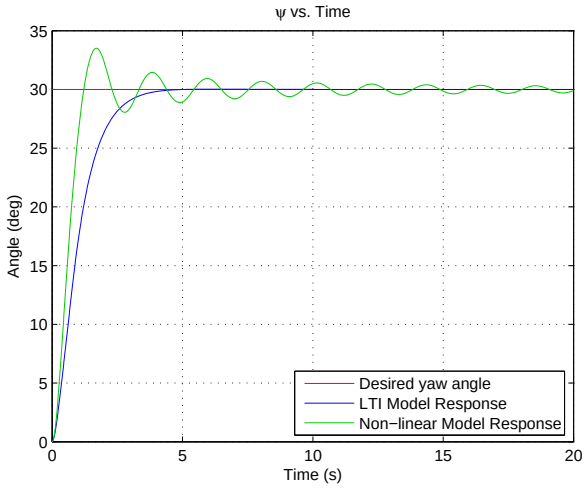


Figure 4.20: Heading angle controller response

As shown in Figure 4.20, desired angle is set to 30 *deg*. The heading controller reaches desired point about 4 *secs* when LTI model is used. There is also no steady-state error.

4.4.3 Depth Controller

For depth controller, cascaded PID controller is used as shown in Figure 4.21. The error between actual and the desired depth is a reference signal for the heave speed controller.

In Figure 4.22, the vehicle reaches the desired depth in 40 *secs* since its maximum heave

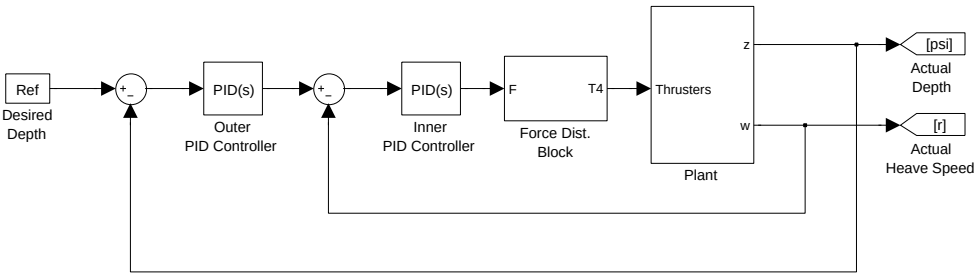


Figure 4.21: Cascaded PID controller block diagram for depth

speed is 0.5 m/s. Response of the LTI model and the non-linear model is close to each other.

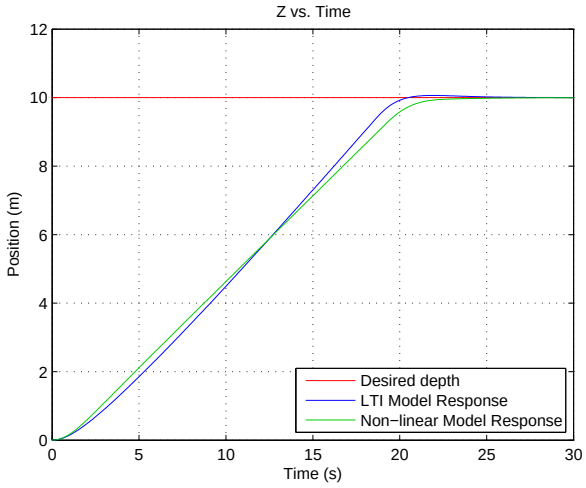


Figure 4.22: Depth controller response

4.5 Conclusion

In this chapter, firstly, the thruster model is presented. Basically, it consists of a permanent magnet DC motor model, propeller mapping and fluid model. Secondly, DST-R-100-4 model responses are examined under several cases. According to results, vehicle behaviour under different inputs are examined such as getting the maximum speed of the vehicle along X, Y and Z-axis as well as yaw rate. Moreover, stability check is done due to the results and it shows that both angular and linear velocities of the vehicle converge a constant value for all test cases. It implies that the vehicle is stable. Next, speed and heading controllers are

designed by using the PID controller which is the most common feedback control method. A cascaded PID controller is used for depth controller. The controllers are designed by using the LTI model and response of them is examined by using the LTI model and the non-linear model.

CHAPTER 5

GUIDANCE AND OBSTACLE AVOIDANCE

5.1 Introduction

Suitable guidance law is crucial when making any vehicle autonomous. For autonomous operations, the vehicle must have authentic navigation, guidance and control systems. Fossen [17] points out that guidance is the action of determining course, attitude and speed of the vehicle, relative to some reference frame, to be followed by the vehicle. Therefore, guidance is the key element for reliable autonomous operations. Guidance system generates set points for controllers.

There are several guidance methods for underwater vehicles [32] such as:

- Way Point Guidance by Line-of-Sight (LOS)
- Vision Based Guidance
- Guidance Using Chemical Signals
- Lyapunov Based Guidance
- Proportional Navigation Guidance (PNG) for AUVs
- Guidance Using Magnetometers for Cable Tracking
- Electromagnetic Guidance

In this thesis, way point guidance by LOS in 3D is used since it is common for ROVs [5]. In addition, way point guidance based on optimal control for the yaw plane is examined. Way

point guidance based on optimal control means that while the vehicle goes towards a way point, it uses optimal input values. These values are obtained by solution of optimal control problem. There is a *Database* that stores optimal solutions for various heading angles and distances. Section 5.3 includes definition and solution methods of optimal control problem, several test cases in order to compare solution methods performance, generating database and way point guidance by using database.

5.2 Way Point Guidance by LOS

Way point guidance by LOS is very common for underwater vehicles since it is very simple and requires low computational effort. Basically, the path which is desired to be tracked, is divided into number of way points. These way points are selected by ROV pilot. In other words, the vehicle knows positions of way points before operation, and it plans its motion to reach way points sequentially. At the beginning of the operation, first way point is called as *present way point*. When the distance between desired way point and the vehicle is sufficiently small distance, next way point is a new present way point for the vehicle. This procedure runs until the vehicle reaches the final way point.

5.2.1 Way Point Guidance by LOS in the Yaw Plane Motion

In Figure 5.1, actual position of the vehicle is denoted by $(x(t), y(t))$, current k-th way point location is denoted by $P_k = (x_d(k), y_d(k))$. Line-of-sight is the line connects the vehicle and the way point. The length of this line is denoted by d_k and it is also called as distance between the present way point and the vehicle. ψ denotes heading angle, α denotes LOS angle and ρ denotes radius of circle of acceptance.

The reference heading angle is followed by LOS angle because if difference between heading angle and LOS angle is zero then the way point is located in front of the vehicle. α is computed by using Equation 5.1 by the four quadrant inverse tangent function.

$$\alpha = \tan^{-1} \frac{\Delta y}{\Delta x} \tag{5.1}$$

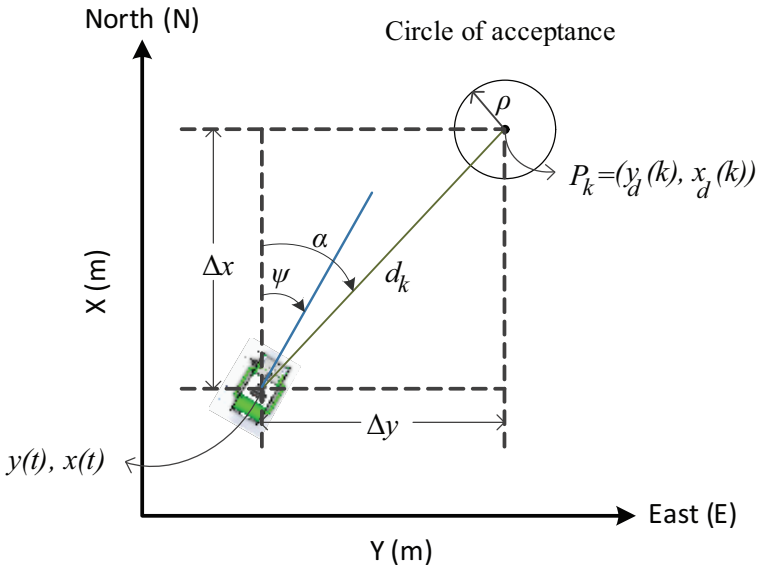


Figure 5.1: Line-of-sight guidance definitions for the yaw plane

where

$$\begin{aligned} \Delta x &= x_d(k) - x(t) \\ \Delta y &= y_d(k) - y(t) \end{aligned} \tag{5.2}$$

The present way point P_k is achieved, when the vehicle enters circle of acceptance. In other words, if the distance between way point and the vehicle is smaller than ρ, the next way point $P_{k+1} = (y_d(k+1), x_d(k+1))$ is selected as the present way point. Mathematically, the vehicle selects next way point when the following condition is satisfied.

$$d_k = \sqrt{(\Delta x)^2 + (\Delta y)^2} \leq \rho \tag{5.3}$$

ρ can be set to two times length of the vehicle [21].

As shown in Figure 5.2, the path which is produced by the vehicle is getting smoother while Γ decreases as explained in Section 4.4.1. However, lower values for Γ causes some way points to be unreachable. As a solution for this problem, the following remedy has been developed [5]. If the condition is satisfied, then present way point is changed with the next way point. This method is also called as *Missed Way Point Detection*.

$$\begin{aligned} d_k &< \rho_{max} \\ d_k &> d_{min} \end{aligned} \tag{5.4}$$

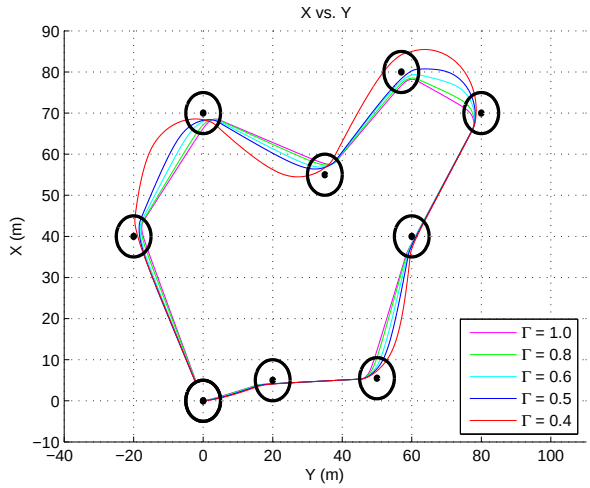


Figure 5.2: Way point guidance by LOS with different thruster weights

where $\rho_{max} > \rho$ and it denotes maximum allowed miss distance, d_k is actual distance and d_{min} is achieved minimum distance. In Figure 5.3, the path is generated by adding missed

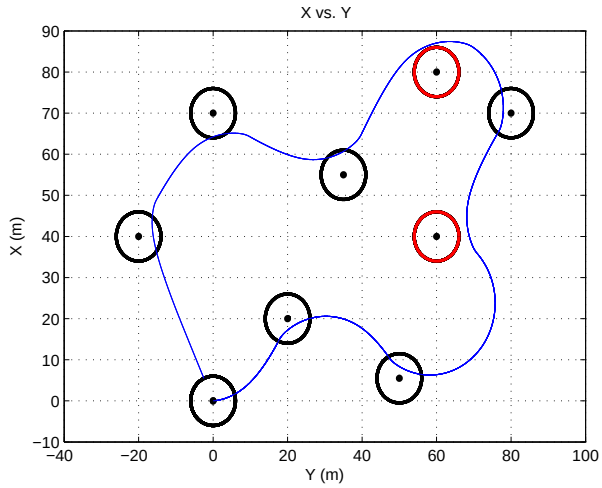


Figure 5.3: Missed way point detection

way point detection algorithm to way point guidance by LOS. Red circles shows that they are missed by the vehicle.

In addition, final surge speed (u_f) and final heading (ψ_f) can be defined for the last way point. Guidance sets these values as references for the surge speed controller and the heading

controller when distance between the vehicle and the last way point is sufficiently small. This distance may be taken the same or greater than ρ_{max}.

As a case study, $u_f = 0.2\ m/s$ and $\psi_f = -160\ deg$ are selected as final values. According shown in Figure 5.4, the vehicle travels all way points. When it reaches to the final way point, its surge speed and heading angle is 0.21 m/s and -165.43 deg.

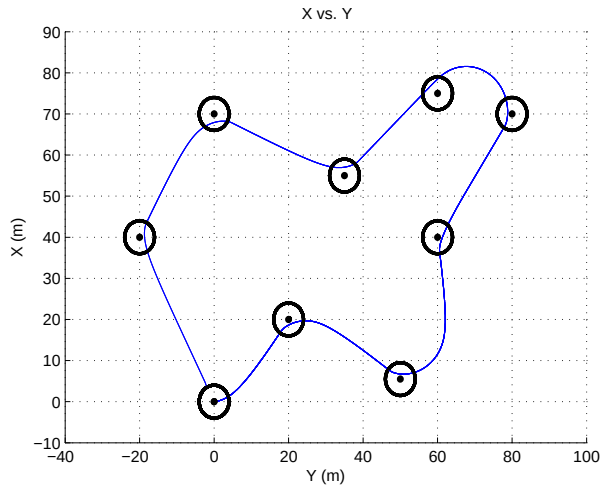


Figure 5.4: Way point guidance by LOS with u_f and ψ_f

5.2.2 Way Point Guidance by LOS in the Pitch Plane Motion

DST-R-100-4 does not have any actuator to control its pitch angle, θ. For this reason, surge and heave speeds are effective for reaching final point and by this way it differs from way point guidance in the yaw plane.

In Figure 5.5, β denotes LOS angle in the pitch plane and can be found by Equation 5.5.

$$\beta = \tan^{-1} \frac{\Delta z}{\Delta x} \tag{5.5}$$

where

$$\begin{aligned} \Delta x &= x_d(k) - x(t) \\ \Delta z &= z_d(k) - z(t) \end{aligned} \tag{5.6}$$

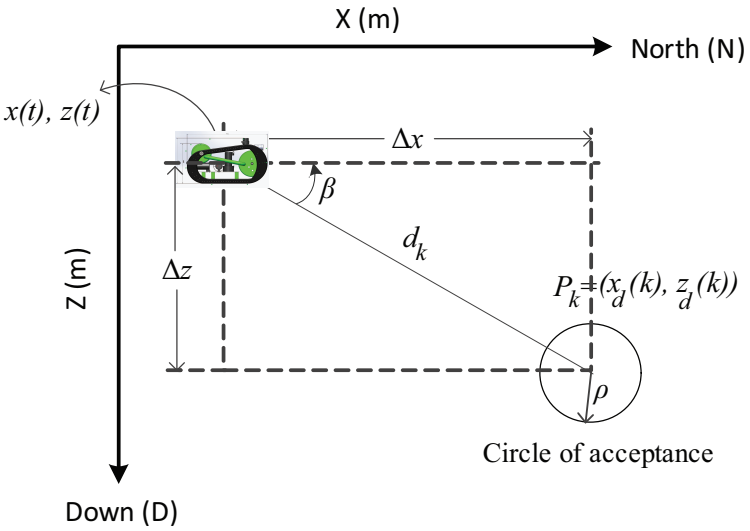


Figure 5.5: Line-of-sight guidance definitions for pitch plane

When the following condition is satisfied, next way point is selected.

$$d_k = \sqrt{(\Delta x)^2 + (\Delta z)^2} \leq \rho \tag{5.7}$$

Algorithm 1 explains how to change surge and heave speeds of the vehicle in order to reach the way point. u_f and w_f can be selected for last way point. When selecting w_f, its sign must be same with velocity sign between last two way points.

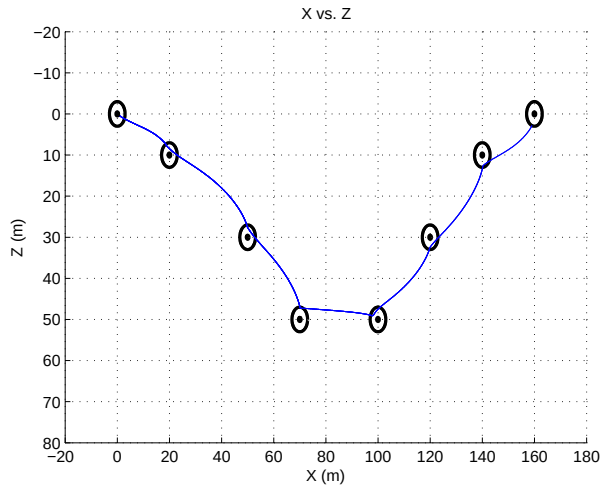


Figure 5.6: Way point guidance by LOS in pitch plane

As a case study, initial position of the vehicle is (0,0) and the final way point is located at (160,0). Other way points are shown in Figure 5.6. Final values for surge and heave speeds are set to 0.6 and 0.2 m/s, respectively. According to Figure 5.6, the vehicle reaches every way point successfully and its final surge and heave speeds are 0.53 and 0.29 m/s, respectively.

Algorithm 1 Determining u and w

$t_x = |\Delta X|/u_{max}$
$t_z = |\Delta Z|/w_{max}$
if $t_x \geq t_z$ **then**
  $t = t_x$
else
  $t = t_z$
end if
$u = |\Delta x|/t$
$w = |\Delta z|/t$
if $x(t) > x_d(k)$ **then**
  $u = -u$
end if
if $z(t) > z_d(k)$ **then**
  $w = -w$
end if

5.2.3 Way Point Guidance by LOS in 3D Space

Way point guidance by LOS in 3D space is done by getting together above algorithms. Sphere of acceptance is used instead of circle of acceptance. Radius of sphere of acceptance provides which distance is enough for selecting next way point. If the following condition is satisfied, next way point is selected.

$$d_k = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} \leq \rho \tag{5.8}$$

u_f and ψ_f can be defined for last way point. If the height of last two way points are different, w_f also can be defined.

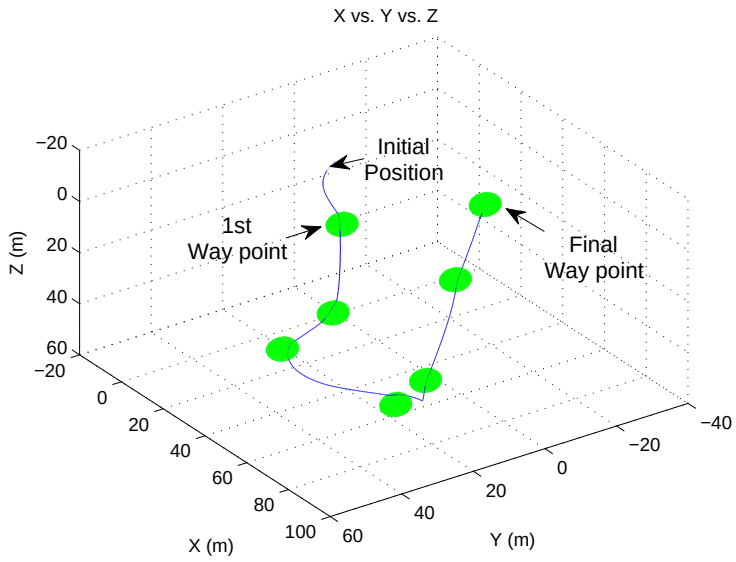
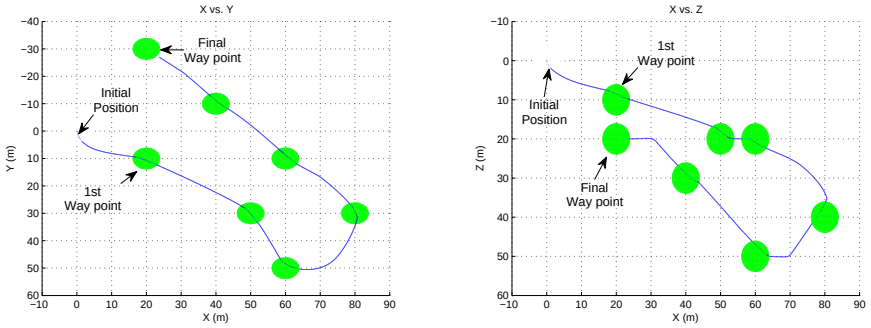
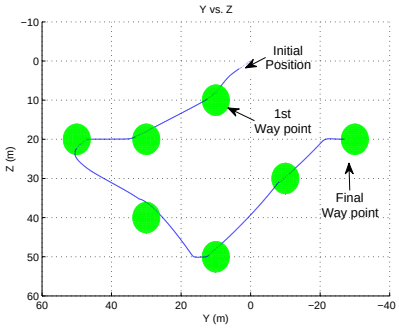


Figure 5.7: Way point guidance by LOS in 3D



(a) X-Y view

(b) X-Z view



(c) Y-Z view

Figure 5.8: Way point guidance by LOS in 3D as 2D views

As a case study, initial position of the vehicle is (0,0,0) and final way point is located at (20,-30,20). Other way points are shown in Figure 5.7. u_f=0.4 m/s, w_f=0.2 m/s and ψ_f=-145 *deg* are selected as a final values. The vehicle reaches all way points successfully. In Figure 5.8(a), the trajectory is shown in the $X-Y$ plane. In Figure 5.8(b), the trajectory is shown in the $X-Z$ plane. Note that, the vehicle reaches the desired height before reaching final point. This is also shown in Figure 5.8(c) which is trajectory of the $Y-Z$ plane. Final values for $u_f(t_f)$=0.37 m/s, $w_f(t_f)$=0.16 m/s and $\psi_f(t_f)$=-135 *deg* at last way point.

5.3 Way Point Guidance Based on Optimal Control

DST-R-100-4 vehicle can travel from starting point to final point in different trajectories as shown in Figure 5.9. Except green trajectory, all trajectories are generated changing heading angle. For green trajectory, no need to change the angle. In Figure 5.10(a), the vehicle goes to final point with fixed heading angle. In order to do that, the vehicle goes along X-axis and along Y-axis, respectively. The vehicle also does this movement step-by-step (red lines in Figure 5.10(a)) or just one time (green lines in Figure 5.10(a)). Another topic about manoeuvring with fixed heading angle is that whether the vehicle can go to the final point by moving along X-axis and Y-axis at the same time shown in orange line as shown in Figure 5.10(a).

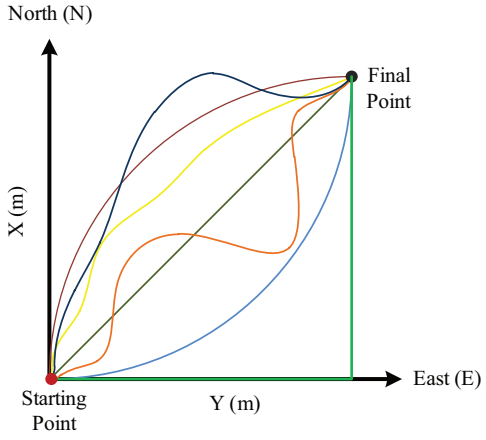
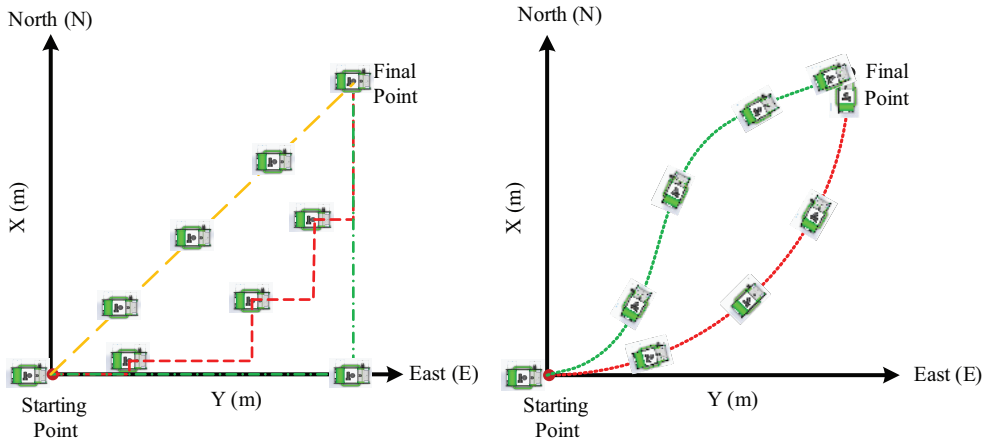


Figure 5.9: Path possibilities

The vehicle also goes to the final point by changing its heading angle as shown in Figure 5.10(b). In the figure, there are two different trajectories that are generated by different head-



(a) Fixed heading angle (b) Changing heading angle

Figure 5.10: Manoeuvring heading angle with changing and fixed

ing angles. It is clear that there are infinitely many trajectories that can be generated by using this approach. Up to here, the yaw plane movement is considered. The vehicle also makes a movement in the pitch plane by using its vertical and forward thrusters. Since there is no actuator for creating pitch angle, trajectory possibilities are shown in Figure 5.10(b) is not valid for the pitch plane motion. On the other hand, all possibilities (green lines, red lines and orange lines) are shown in Figure 5.10(a) are valid. Each trajectory for both the yaw plane and the pitch plane motion has different amount of energy consumption, survey time etc. In this thesis, the trajectory between starting point and final point are generated subject to minimum energy consumption by finding optimal input values for controllers.

Energy-optimal trajectory can be computed by using numerical solution of a special optimal control problem (OCP) [25]. Spangelo [38] points out that dissipated electrical power has a relation with thruster force of $i - th$ thruster, denoted by T_i. Proof of the relation is explained in Appendix B. The relation can be modelled as:

$$P_i = \frac{2\pi K_Q(J_0)}{\sqrt{\rho} D_i |K_T(J_0)|^{1.5}} |T_i|^{1.5} \tag{5.9}$$

where J_0 is advanced ratio, D is diameter of propeller, K_Q and K_T are torque and thruster coefficients, respectively.

Equation 5.9 can be rewritten as

$$P_i = \xi_i |T_i| \tag{5.10}$$

where ξ_i is weight factor and constant, but it changes depending on characteristics of thrusters.

A cost function of OCP is a weighted combination of energy consumption and the state histories. By using Equation 5.10, the cost function, J, can be written as follows.

$$J = \frac{1}{2} \left(\Delta \bar{x}(t_f)\right)^T S \, \Delta \bar{x}(t_f) + \lambda \int_{t_0}^{t_f} \sum_{i=1}^{4} |T_i|^{1.5} d\tau \tag{5.11}$$

where

t_0: Initial time

t_f: Final time

$\bar{x}_f$: Desired final state

$\bar{x}(t_f)$: Final state

S: Diagonal weighting matrix and $S > 0$

λ: Scalar weighting constant and $\lambda > 0$

$\Delta \bar{x}(t_f)$: Difference between desired final state and actual final state.
As mathematical notation, $\Delta \bar{x}(t_f) = \bar{x}(t_f) - \bar{x}_f$

Basically, $\bar{x} = [x \; y]^T$ for the yaw plane and $\bar{x} = [x \; z]^T$ for the pitch plane. Also, final speeds and final orientation can be added to the state vector.

For the yaw plane motion, after adding u_f and ψ_f to the state vector, it equals $\bar{x} = [x \; y \; u \; \psi]^T$. There are constraints on surge speed and heading angle such as $u \leq 1.3 \; m/s$ and $\psi \leq \pi \; rad$, respectively. After putting together all these constraints, the following OCP can be defined mathematically by using Equation 5.11.

$$min \; J = \frac{1}{2} \left(\Delta \bar{x}(t_f)\right)^T S \, \Delta \bar{x}(t_f) + \lambda \int_{t_0}^{t_f} \sum_{i=1}^{4} |T_i|^{1.5} d\tau \tag{5.12}$$

subject to

$$\begin{aligned} 0 \leq u \leq 1.3 \\ -\pi \leq \psi \leq \pi \end{aligned} \tag{5.13}$$

Boundary conditions are

$$\bar{x}(t_0) = \begin{bmatrix} x_i \\ y_i \\ u_i \\ \psi_i \end{bmatrix} \quad , \quad \bar{x}_f = \begin{bmatrix} x_d \\ y_d \\ u_f \\ \psi_f \end{bmatrix} \tag{5.14}$$

Since time consumption is not critical, final time is fixed. Therefore, this leads to a *Two Point Boundary Value Problem, (TPBVP).*

In order to generate trajectory like Figure 5.10(b), u and ψ must be adjusted at each time interval. Solution methods try to find an optimal solution for OCP in Equation 5.12, by changing states at each time interval. Consequently, travelling time is divided into n equal intervals and solution methods generate reference values (u and ψ) for each time interval. As a result, number of variables of OCP in Equation 5.12 equals $2n$.

Because of the fact that Equation 5.12 is called as *constrained non-linear multi-variable function*, OCP is solved by using constrained non-linear optimization.

5.3.1 Constrained Non-Linear Optimization

For constrained non-linear optimization, numerical algorithms are divided into two methods such as gradient-based and direct search method. Gradient-based methods use *gradient* (first derivatives) or *Hessian* (second derivatives) information. The non-linear interior point method, sequential quadratic programming (SQP) method, the augmented Lagrangian method can be given as examples of gradient search methods. Direct search methods do not use gradients or Hessians information. Genetic algorithm and differential evolution are the two most popular one.

Non-linear constrained optimization problems can be formulated as:

$$\begin{aligned} &\textit{minimize} && f(\bar{x}) \\ &\textit{subject to} && h_i(\bar{x}) = 0, \quad i = 1, ..., m \\ & && g_j(\bar{x}) \leq 0, \quad j = 1, ..., p \end{aligned} \tag{5.15}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^m \to \mathbb{R}$ and $m \leq n$. Feasible point is any point satisfying the constraints and set of all feasible points is called as the feasible set.

5.3.1.1 Non-linear Optimization with Equality and Inequality Constraints

For constrained optimization problem consisting of inequality constraints and equality constraints, the following problem can be considered.

$$\begin{aligned} &\textit{minimize} && f(\bar{x}) \\ &\textit{subject to} && \mathbf{h}(\bar{x}) = \mathbf{0} \\ & && \mathbf{g}(\bar{x}) \leq \mathbf{0} \end{aligned} \tag{5.16}$$

First of all, equality constraints are multiplied by Lagrange multipliers and inequality constraints are multiplied by KKT multipliers (which is also called as barrier term) in order to include them into the objective function, $f(\bar{x})$. For this kind of problem, Karush-Kuhn-Tucker (KKT) condition is introduced. This conditions is also called as first-order necessary condition for a point to be a local minimizer. There are also second-order necessary and sufficient conditions. If $\bar{x}^*$ is a strict local minimizer of problem defined in Equation 5.16, $\bar{x}^*$ should satisfy all conditions.

Deep analysis about constrained non-linear optimization algorithms and solving methods are out of scope of this thesis. Further information about these subjects can be found in [6], [13], [27] and [28]. For solving constrained non-linear optimization problem which is defined in Equation 5.12, *Optimization Toolbox* by MATLAB is used. It provides `fmincon` as a solver for constrained non-linear optimization. Optimization Toolbox uses four algorithm to solve these kind of problems such as the interior point algorithm, SQP algorithm, the active set algorithm and trust-region reflective algorithm [30]. *The interior point algorithm (IPA)* is selected as an algorithm because it is based on a barrier function and all iteration strictly feasible with respect to bounds during optimization. This algorithm also serves several options for calculating the Hessians [29].

5.3.1.2 Genetic Algorithm

Genetic Algorithm (GA) [13] is another solution method for solving OCP. GA is a stochastic method for global search and optimization. There are three main principles of natural evo-

lution, which are used by GA, such as reproduction, natural selection and diversity of the species, maintained by the differences of each generation with the previous.

GA works with a set of individuals (chromosomes) which represents possible solutions of the task. The selection principle is applied by using a criterion, giving an evaluation (crossover and mutation) for the individual with respect to the desired solution. The best-suited individuals create the next generation. Basically, GA can be explained as follows.

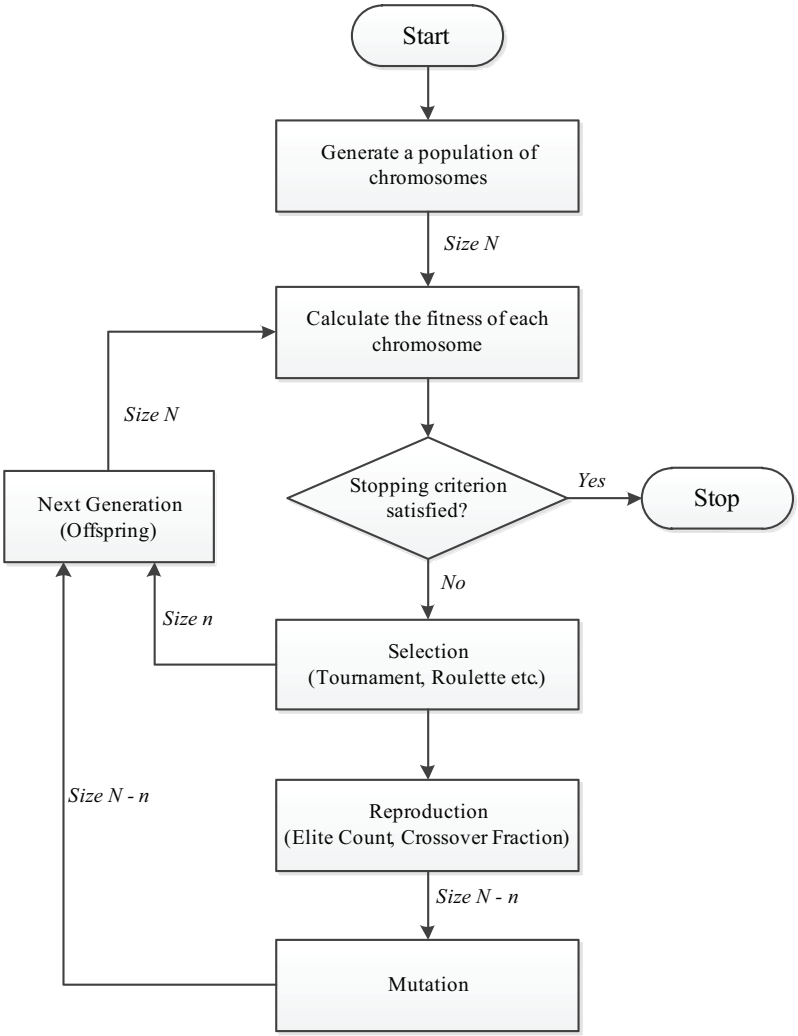


Figure 5.11: Genetic algorithm flow chart

1. Generate initial population: Initial population is generated randomly. Chromosomes can satisfy constraints, which is called feasible populations, as well as can be unfeasible. All populations have the same number (N) of individuals because of easier com-

putational procedure. [13]

2. Calculate the function values by using generated chromosomes. This is also called fitness value of each chromosomes.

3. Check stopping criteria: There are several criteria to stop the algorithm such as:

   - Value of the function: The algorithm stops when the best individual reach to around of the value of the function which is set at the beginning of the algorithm.
   - Maximum number of generations: The algorithm stops when generations (iterations) reach to the maximum number. This is commonly used as a stopping criteria because it guarantees that the algorithm gives a result, nevertheless there is no guarantee that the result is extremum or not.
   - Stall generation: The algorithm stops when there is no improvement of the fitness value of the best individual within generations.

4. Selection: The selection function chooses individuals for the next generation based on their fitness value. The best n number of individuals are directly transferred to next generation. There are several method in order to select these individuals as tournament, roulette etc.

5. Reproduction: At this stage, it is determined how to create new individuals for next generations. Elite count specifies the number of individuals that are guaranteed to transfer to next generations and crossover fraction specifies the fraction of next generations that crossover produces. It is important to get offspring individuals because of inheriting the best possible combination of the genes of their parents.

6. Mutation: It means that small random changing on individuals in the population. It provides genetic diversity.

7. New generation: It includes elite individuals and individuals who are passed mutation and crossover. Population size of new individuals are also N.

The number of variables of OCP is reflected in the dimension of the chromosome vectors of GA. As mentioned before, if travelling time is divided into n equal intervals, number of

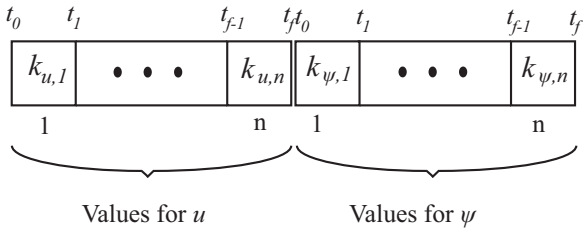


Figure 5.12: Chromosome structure

variables equals 2n. In other words, number of variables for each chromosome is 2n. Chromosome structure is shown in Figure 5.12 and $k_{u,n}$ and $k_{\psi,n}$ denote values of u and ψ at n-th time interval.

In this thesis, the OCP is solved by using GA first. The crossover rate and population size are set to 0.65 and 150, respectively. High value of mutation rate may help to generate more random offsprings that provide high diversity. For this reason, it is set to 0.7. As a selection operator, tournament is selected and the number of elite individuals is set to 10. Secondly, IPA is used for solving the OCP. The solution found by GA is used as initial values for IPA. MATLAB Optimization Toolbox is used for implementing GA and IPA.

Case Studies

The following OCP is solved for three cases by using LTI model.

$$min\ J = \frac{1}{2}\left(\Delta\bar{x}(t_f)\right)^T S\ \Delta\bar{x}(t_f) + \lambda \int_{t_0}^{t_f} \sum_{i=1}^{4} |T_i|^{1.5} d\tau \tag{5.17}$$

subject to

$$\begin{aligned} 0 \le u \le 1.3 \\ -\pi \le \psi \le \pi \end{aligned} \tag{5.18}$$

Boundary conditions are

$$\bar{x}(t_0) = \begin{bmatrix} x_i \\ y_i \\ u_i \\ \psi_i \end{bmatrix} \quad , \quad \bar{x}_f = \begin{bmatrix} x_d \\ y_d \\ u_f \\ \psi_f \end{bmatrix} \tag{5.19}$$

Firstly, boundary conditions are selected as;

$$\bar{x}(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \bar{x}_f = \begin{bmatrix} 20 \\ 40 \\ 0.8 \\ 60 \end{bmatrix} \tag{5.20}$$

In Figure 5.13, the trajectory created by IPA is smoother than the trajectory created by GA. Also, cost of IPA is smaller than GA. According to this result, it is said that the trajectory created by IPA requires minimum energy. Note that, IPA uses GA solution as initial value.

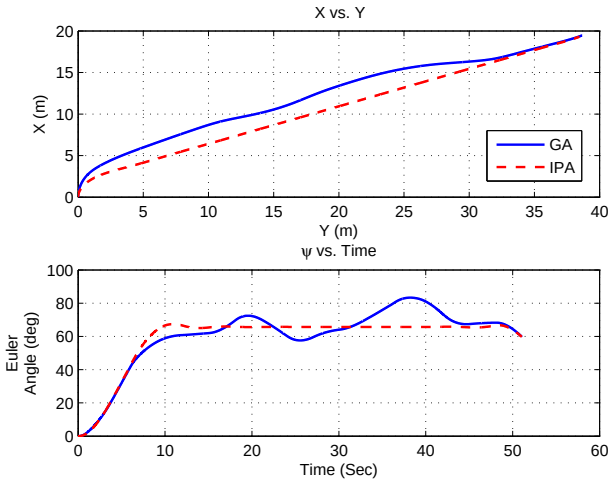


Figure 5.13: OCP solution results for $\bar{x}_f = [20\ 40\ 0.8\ 60]^T$

Numerical values for final state are shown in Table 5.1.

Table 5.1: OCP solution results for $\bar{x}_f = [20\ 40\ 0.8\ 60]^T$

	X (m)	Y (m)	ψ (deg)	u (m/s)	Cost
Desired	20	40	60	0.8	
GA	19.4640	38.6670	60.1343	0.8416	158.0677
IPA	19.3063	38.4625	60.0013	0.7705	154.7997

Secondly, boundary conditions are selected as;

$$\bar{x}(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0.4 \\ 10 \end{bmatrix} \quad , \quad \bar{x}_f = \begin{bmatrix} 30 \\ 10 \\ 0.6 \\ 40 \end{bmatrix} \tag{5.21}$$

Optimal trajectories and heading angle changes are shown in Figure 5.14.

Depending on cost shown in Table 5.2, the trajectory requires minimum energy is created by IPA.

Table 5.2: OCP solution results for $\bar{x}_f = [30\ 10\ 0.6\ 40]^T$

	X (m)	Y (m)	ψ (deg)	u (m/s)	Cost
Desired	30	10	40	0.6	
GA	28.3468	9.9700	40.6013	0.5759	106.9107
IPA	28.3984	9.4772	39.9919	0.5705	104.8662

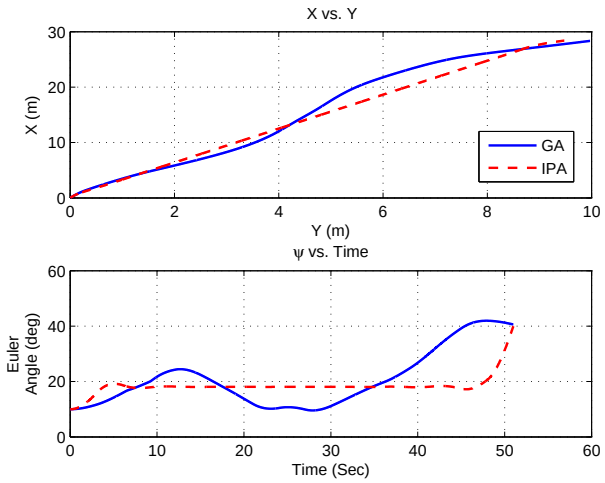


Figure 5.14: OCP solution results for $\bar{x}_f = [30\ 10\ 0.6\ 40]^T$

The state vector of OCP can be expanded by adding position along the Z-axis and w_f. Then, state components and boundary conditions are changed as follows:

$$\bar{x} = [x\ y\ z\ u\ \psi\ w]^T \tag{5.22}$$

$$\bar{x}(t_0) = \begin{bmatrix} x_i \\ y_i \\ z_i \\ u_i \\ \psi_i \\ w_i \end{bmatrix} \quad , \quad \bar{x}_f = \begin{bmatrix} x_d \\ y_d \\ z_d \\ u_d \\ \psi_d \\ w_d \end{bmatrix} \tag{5.23}$$

According to solution of OCP by using boundary values in Equation 5.23, optimal trajectory in 3D space is obtained. Equation 5.12 and considering all constraints yield following OCP with boundary conditions are defined in Equation 5.23. Moreover, it is expected that algorithms create reference values for u, ψ and w. For this reason, number of variables for each chromosome must be $3n$, where the travelling time divided into n interval. The constraints on u,ψ, and w are still valid.

$$min\ J = \frac{1}{2}\left(\Delta\bar{x}(t_f)\right)^T S\ \Delta\bar{x}(t_f) + \lambda \int_{t_0}^{t_f} \sum_{i=1}^{4} |T_i|^{1.5} d\tau \tag{5.24}$$

subject to

$$\begin{aligned} 0 \le u \le 1.3 \\ -\pi \le \psi \le \pi \\ 0 \le w \le 0.5 \end{aligned} \tag{5.25}$$

Boundary conditions are selected as follows for case study.

$$\bar{x}(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.6 \\ 10 \\ 0 \end{bmatrix}, \quad \bar{x}_f = \begin{bmatrix} 30 \\ 20 \\ 10 \\ 0 \\ 40 \\ 0.2 \end{bmatrix} \tag{5.26}$$

Optimal trajectories are generated from IPA and GA are shown in Figure 5.15.

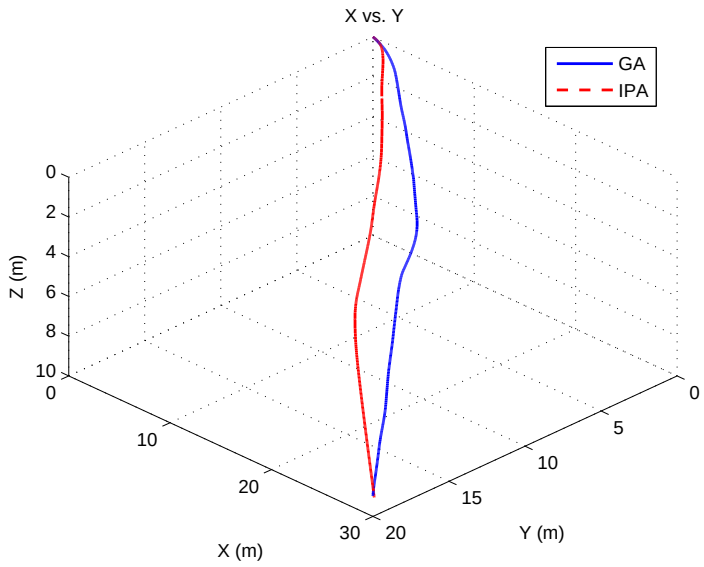


Figure 5.15: OCP solution results for $\bar{x}(t_0) = [0\,0\,0\,0.6\,10\,0]^T$ and $\bar{x}_f$ =[30 20 10 0 40 0.2]^T

According to results shown in Table 5.3, there are small differences between the desired final state and final state at final time when IPA is used.

Table 5.3: OCP solution results for $\bar{x}(t_0) = [0\,0\,0\,0.6\,10\,0]^T$ and $\bar{x}_f = [30\ 20\ 10\ 0\ 40\ 0.2]^T$

	X (m)	Y (m)	Z (m)	u (m/s)	ψ (deg)	w (m/s)	Cost
Desired	30	20	10	0	40	0.2	
GA	29.4493	19.6265	9.2204	0.2902	39.9206	0.1708	214.7413
IPA	29.4262	19.5323	9.3373	0.0234	39.8823	0.2323	178.8950

As a result, OCP is solved by the most popular global search method, GA, firstly. Solution obtained from GA is used as initial state for IPA. According to the results, cost values are always smaller when this sequence is used. For this reason, it can be said that the most optimum trajectory is created by using the sequence.

5.3.2 Database Generation

Database includes optimal trajectories which are generated for various initial and final speeds, final heading angle and distance by obtaining solution of OCP in Equation 5.12.

First of all, distance between desired point and the vehicle is divided into three, called as *small*, *medium*, and *large* and the distance set is denoted by d_{db}. Next, final heading angle is split into five different values and the set including them is denoted by ψ_{db}. Each distance and heading angle refers points in coordinate system as shown in Figure 5.16. Since the distance is a scalar value, different heading angle with constant distance covers a circle with its radius equal to the distance. In order to generate database, these points are used at final state boundary condition.

$$x_{d(i,j)} = \sin(\psi_{db,j})\, d_{db,i} \tag{5.27}$$

$$y_{d(i,j)} = \cos(\psi_{db,j})\, d_{db,i} \tag{5.28}$$

where

$$\begin{aligned} i &= 1,2,3 \\ j &= 1,2,3,4,5 \\ d_{db} &= \{d_{db,1},\ d_{db,2},\ d_{db,3}\} \\ \psi_{db} &= \{\psi_{db,1},\ \psi_{db,2},\ \psi_{db,3},\ \psi_{db,4},\ \psi_{db,5}\} \end{aligned} \tag{5.29}$$

and $x_{d(i,j)}$ ($y_{d(i,j)}$) refers point along X (Y)-axis depending on i-th distance and j-th heading angle. u_i and u_f are quantized for each $x_{d(i,j)}$ and $y_{d(i,j)}$.

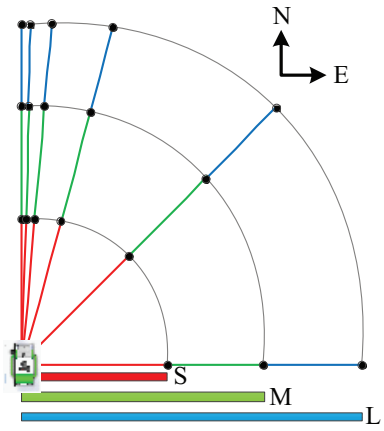


Figure 5.16: Distance and heading angle scheme for database generation

Equation 5.29, $d_{db,1}$ denotes small distance as shown in Figure 5.16 as red bar. $d_{db,2}$ and $d_{db,3}$ denote medium and large, respectively.

For numerical solution of OCP, u_i, u_f, d_{db} and ψ_{db} are selected as follows.

$$\begin{aligned} d_{db} &= \{40, 20, 3\}\, m \\ \psi_{db} &= \{90, 45, 10, 5, 1\}\, deg \\ u_i &= \{0, 0.65, 1.3\}\, m/s \\ u_f &= \{0, 0.65, 1.3\}\, m/s \end{aligned} \tag{5.30}$$

During database generation, firstly, optimal solution is obtained by using GA, then this solution is used as an initial state for IPA. By using these algorithms, there are 135 different optimal solutions at database. Note that 0 heading angle is not obtained from algorithms, since there is no manoeuvre for this case.

5.3.3 Algorithm for Guidance Based on Optimal Control

There is an algorithm that uses the database in logical way to reach desired point. As mentioned above, relative heading angle is divided into five and relative distance is divided into three. By using them, desired point can be reachable. For instance, suppose the vehicle is located at the starting point and wants to go final point as shown in Figure 5.17.

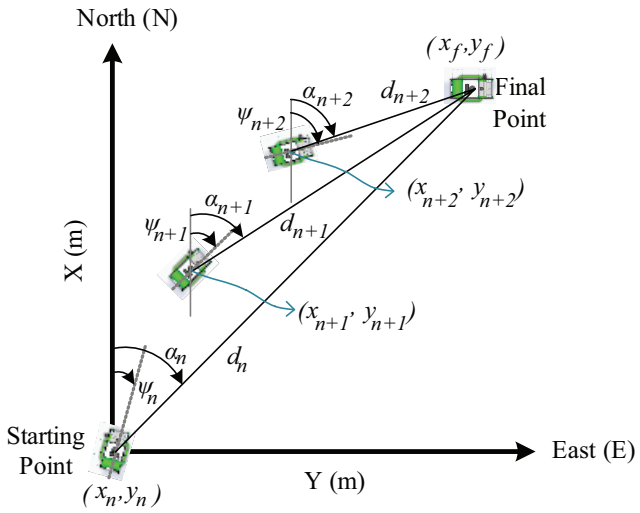


Figure 5.17: The vehicle position and heading angle changing during algorithm runs

Firstly algorithm calculates parameters, heading error ψ_{err}, which is difference between LOS angle and current heading angle, and distance d between its current location and final point. Next, it selects optimal solution from the database depending on parameters by the following Algorithm 2. Note that, κ is a distance constant and $\kappa > 1$. Next, the solution is applied to the system. Then, new distance is calculated. If it is bigger than circle of acceptance (ρ), then the same procedure runs again until $d < \rho$.

In order to illustrate the algorithm, the vehicle starts its motion from (0,0) with $\psi_0 = 0$ to (50,40). The optimal trajectory is shown in Figure 5.18.

This algorithm is developed for only one point. In order to make it usable for way point guidance, the algorithm selects next way point when the vehicle enters circle of acceptance.

Algorithm 2 Selecting solution from database depending on ψ_{err} and d_n

d_{db} = [40 20 3]

ψ_{db} = [90 45 10 5 1]

while *true* **do**

$\Delta x_n = x_f - x(t)$

$\Delta y_n = y_f - y(t)$

$\alpha_n = tan^{-1}\frac{\Delta y}{\Delta x}$

$\psi_{err} = \psi_n - \alpha_n$

$d_n = \sqrt{(\Delta x)^2 + (\Delta y)^2}$

if $d_n \geq \rho$ **then**

for $i = 1 \rightarrow 5$ **do**

if $\psi_{err} \geq \psi_{db}(i)$ **then**

$\psi_{com} \leftarrow \psi_{db}(i)$

break

else

$\psi_{com} \leftarrow 0$

end if

end for

for $j = 1 \rightarrow 3$ **do**

if $d_n \geq \kappa d_{db}(j)$ **then**

$d_{com} \leftarrow d_{db}(j)$

break

end if

end for

else

break

end if

end while

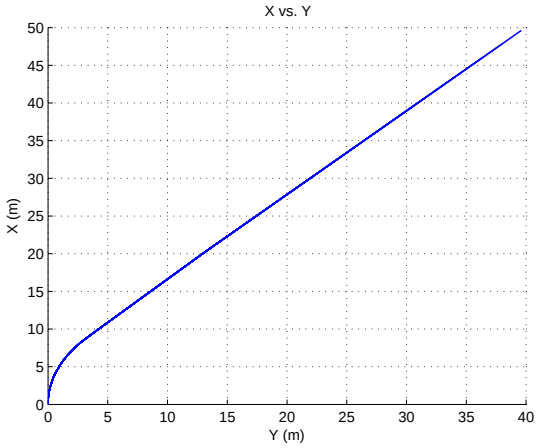


Figure 5.18: Algorithm 2 performance for a single point

5.3.4 Way Point Guidance Based on Optimal Control in Yaw Plane Motion

As mentioned above, the Algorithm 2 is used in order to obtain way point guidance based on optimal control. When the vehicle enters the circle of acceptance, next way point is its new desired point. This goes until the vehicle reaches the final way point. In Figure 5.19, the vehicle travels along all way points by using optimal trajectories for different κ. By this way, effects of varying κ are shown. Note that, ρ selects as 3 for along simulation. u_f and ψ_f can be also inserted this simulation.

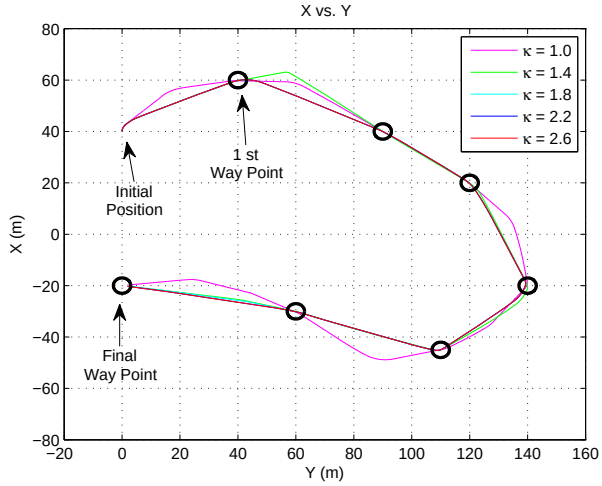


Figure 5.19: Way point guidance based on optimal control in the yaw plane

Way point guidance based on optimal control (WPG - OC) and way point guidance by LOS (WPG - LOS) are compared as shown in Figure 5.20. Both of them travels every way points successfully. Note that, $\kappa = 2.2$, $\Gamma = 0.4$, $\rho = 3$, $u_f = 0.7\,m/s$ and $\psi_f = -40$ degree.

Energy consumption of WPG - OC is 11298 $W\,s$(Watt Second - Joule) where energy consumption of WPG - LOS is 11349 $W\,s$. As a result, WPG - OC requires more minimum energy than WPG - LOS. In addition, for WPG - OC $u_f(t_f) = 0.6982$ and $\psi_f(t_f) = -55.4327$, for WPG - LOS $u_f(t_f) = 0.6650$ and $\psi_f(t_f) = -58.2057$

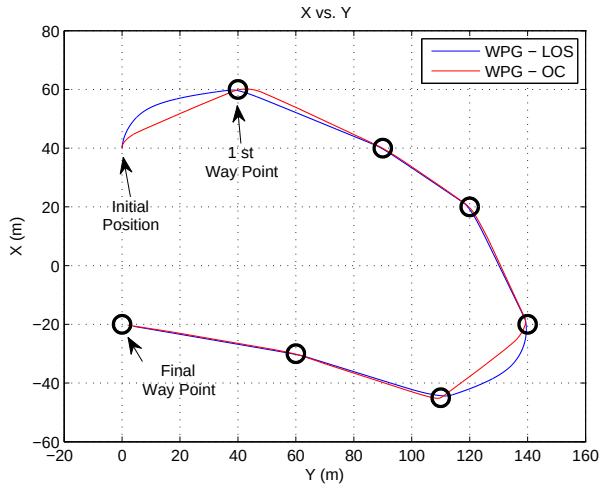


Figure 5.20: WPG - LOS vs. WPG - OC for map 1

In Figure 5.21, the two guidance methods are compared by using different map. Moreover, $\kappa = 2.2$, $\Gamma = 0.4$, $\rho = 3$, $u_f = 0.4\,m/s$ and $\psi_f = 25$ degree. Energy consumption when using WPG - LOS is 18583 $W\,s$, and $u_f(t_f) = 0.4554$ and $\psi_f(t_f) = 29.6025$. For WPG - OC, energy consumption is 18496 $W\,s$, and $u_f(t_f) = 0.4792$ and $\psi_f(t_f) = 33.8754$.

As a result, when WPG - OC is used, not only the vehicle requires minimum energy, but also it satisfies desired the final speed and the heading angle.

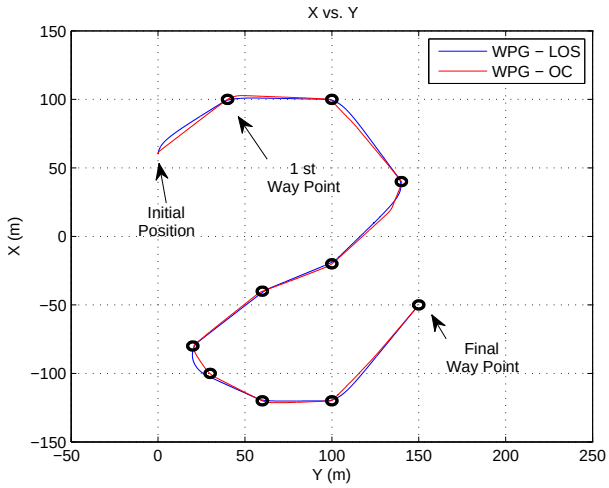


Figure 5.21: WPG - LOS vs. WPG - OC for map 2

5.3.5 Way Point Guidance Based on Optimal Control in 3D Space

In order to guide the vehicle in 3D space, there are two ways such as creating database that includes additionally depth information, or using the database for the yaw plane motion and using Algorithm 1 explained in Section 5.2.2.

In this thesis, the second way is used. Sphere of acceptance definition is the same with the definition in Section 5.2.3.

As a case study, (0,-20,0) is selected as an initial position. In addition, $u_i = 0$, $w_i = 0$ and $\psi_i = 0$ are selected. Final values at final way point are selected $u_f = 0.5$, $w_f = 0.3$ and $\psi_f = -60$.

In Figure 5.22, the vehicle reaches every way points successfully and $u_f(t_f) = 0.4754$, $w_f(t_f) = 0.0023$ and $\psi_f(t_f) = -75.1862$. Final heave speed is far from desired one, since the vehicle reach final height before motion is not completed yet. Hence, height of the vehicle does not change for last several meters.

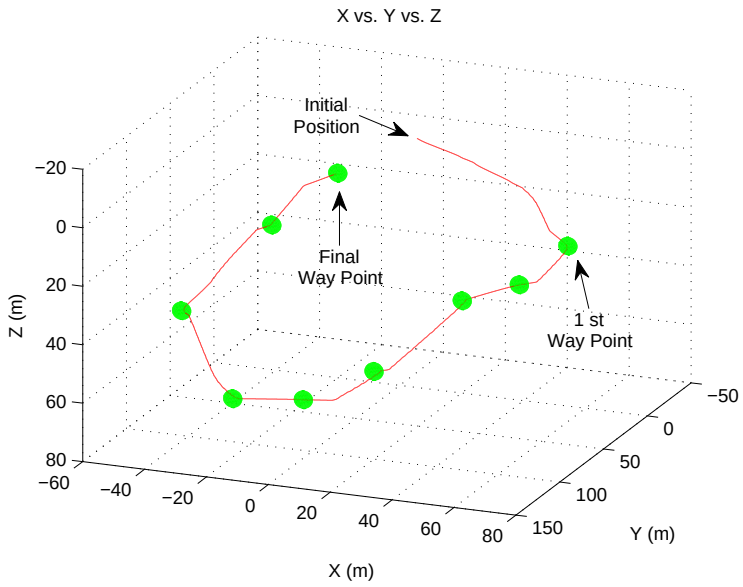
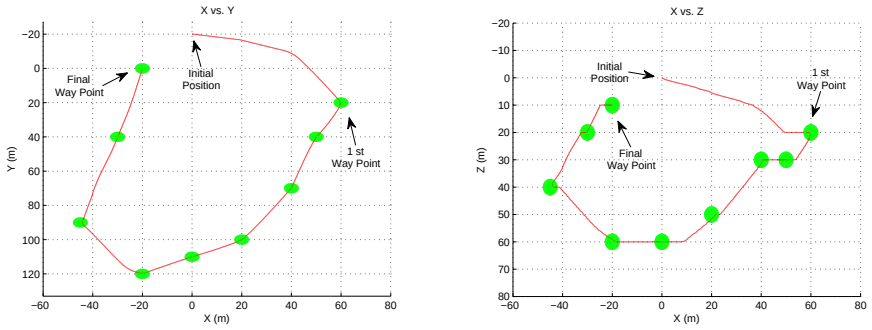
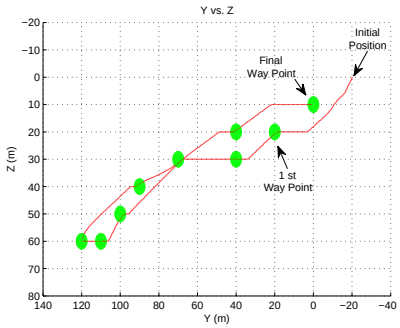


Figure 5.22: Way point guidance by optimal control in 3D



(a) X-Y view

(b) X-Z view



(c) Y-Z view

Figure 5.23: Way point guidance by optimal control in 3D as 2D views

5.4 The Effects of Ocean Currents on the Path

6 DOF dynamic equation of motion is derived assumed that there are no current waves. In order to examine ocean currents effects on the path produced by the vehicle, ocean current terms are added to the Equation 2.19. Ocean currents are circulating system that they are caused by gravity, varying water density in different parts of ocean and wind friction [36]. Ocean currents can be represented relative to body-fixed frame [17] such as

$$\upsilon_r = \upsilon - \upsilon_c \tag{5.31}$$

where $\upsilon_c = [u_c \; v_c \; w_c \; 0 \; 0 \; 0]^T$ is body-fixed current velocities.

It can be assumed that velocities of ocean current are slowly varying or constant, hence $\dot{\upsilon_c} = 0$. Consequently, it can be written that

$$\dot{\upsilon_r} = \dot{\upsilon} \tag{5.32}$$

Consequently, Equation 2.19 has following forms after adding ocean current terms.

$$M\dot{\upsilon} + C(\upsilon_r)\upsilon_r + D(\upsilon_r)\upsilon_r + g(\eta) = \tau \tag{5.33}$$

If $[{u_c}^E \; {v_c}^E \; {w_c}^E]$ denotes earth-fixed current velocity vector, body-fixed current velocities are obtained from following equation.

$$\begin{bmatrix} u_c \\ v_c \\ w_c \end{bmatrix} = T^T_{BE,1}(\eta_2) \begin{bmatrix} {u_c}^E \\ {v_c}^E \\ {w_c}^E \end{bmatrix} \tag{5.34}$$

By using Equation 5.34, ocean current effects on the path produced by WPG - LOS and WPG - OC are examined. Effects of ocean currents on energy consumption is not taken into account. Only, behaviour of the vehicle when using WPG - OC and WPG - LOS is shown.

5.4.1 Ocean Current Effects on WPG - LOS

Firstly, the effects are examined varying u_c^E and v_c^E separately, where remaining current components are fixed zero. In Figure 5.24, ocean current occurs along X axis and blue arrows represent the direction of the current. Initial position for the vehicle is (-20,-20), destination position is (20,20) and $\psi_i = 0$ degree.

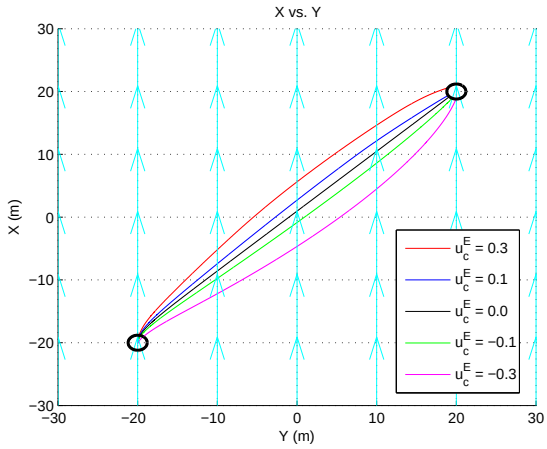


Figure 5.24: Ocean current effects on WPG - LOS for $\psi_i = 0$, $v_c^E = 0$, $w_c^E = 0$ and varying u_c^E

If ψ_i selects as 60 degree and remaining variables stay as a constant, the effects are looking like in Figure 5.25.

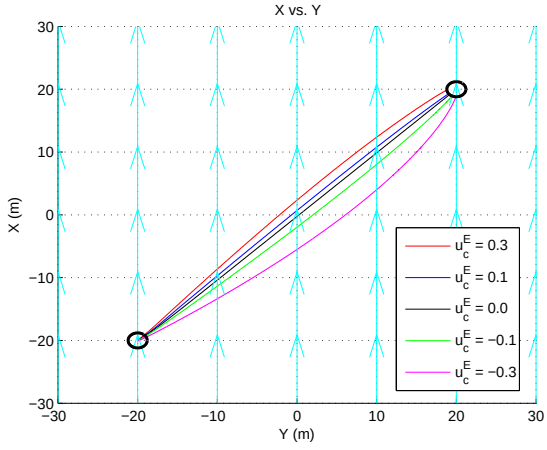


Figure 5.25: Ocean current effects on WPG - LOS for $\psi_i = 60$, $v_c^E = 0$, $w_c^E = 0$ and varying u_c^E

In Figure 5.26, ocean current occurs only along Y axis. Again, initial position for the vehicle is (-20,-20,20), destination position is (20,20,20) and $\psi_i = 0$ degree.

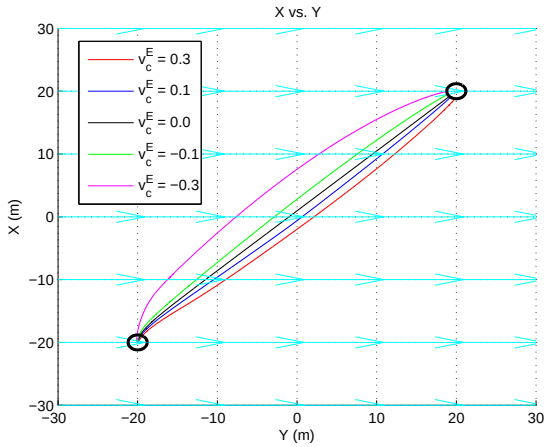


Figure 5.26: Ocean current effects on WPG - LOS for $\psi_i = 0$, $u_c^E = 0$, $w_c^E = 0$ and varying v_c^E

Finally, u_c^E and v_c^E affect on the vehicle at the same time. In Figure 5.27 and Figure 5.28, way point guidance by LOS is simulated under different ocean currents.

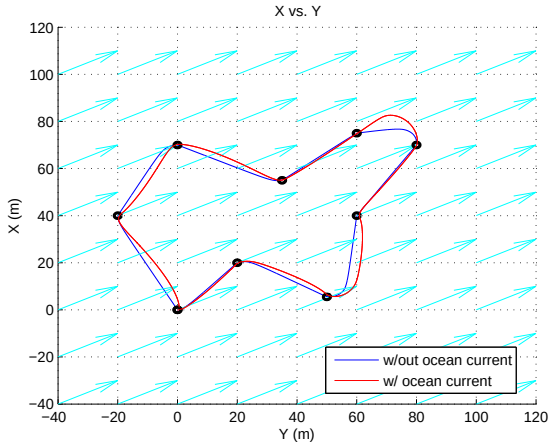


Figure 5.27: Ocean current effects on WPG - LOS for $\psi_i = 90$, $u_c^E = 0.1$, $v_c^E = 0.2$ and $w_c^E = 0$

Blue arrows show current direction. For both figures, initial point for the vehicle is (0,0), the first way point is located at (20,20) and the final way point is the same with initial position. In Figure 5.27, $u_c^E = 0.1$, $v_c^E = 0.2$ and $w_c^E = 0$ are ocean current components. $u_c^E = -0.2$, $v_c^E = 0.2$ and $w_c^E = 0$ are selected as a ocean current magnitude for simulation in Figure 5.28.

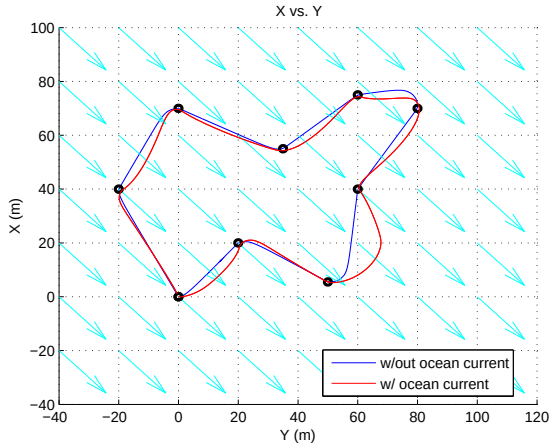


Figure 5.28: Ocean current effects on WPG - LOS for $\psi_i = 90$, $u_c^E = -0.2$, $v_c^E = 0.2$ and $w_c^E = 0$

5.4.2 Ocean Current Effects on WPG - OC

In Figure 5.29, Figure 5.30, and Figure 5.31, WPG - OC under different current directions and varying magnitude are shown. For all simulation, the vehicle starts its motion from (-10,-50) and goes through all way points to final way point is located at (-70,-50). Blue arrows show current. Also, ψ_i is set to 0. Note that, energy consumption is not concerned when using WPG - OC. Just only the vehicle behaviour is examined when WPG - OC is used.

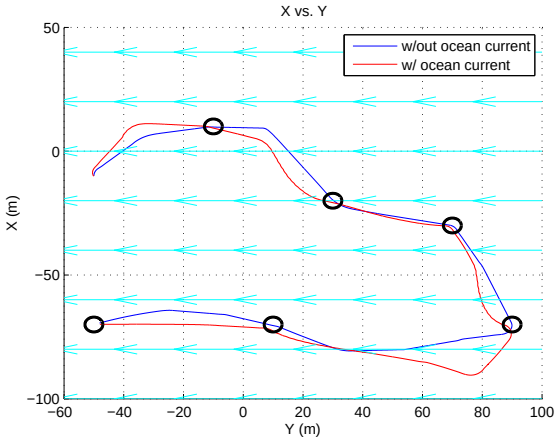


Figure 5.29: Ocean current effects on WPG - OC for $\psi_i = 0$, $u_c^E = 0$, $v_c^E = -0.2$ and $w_c^E = 0$

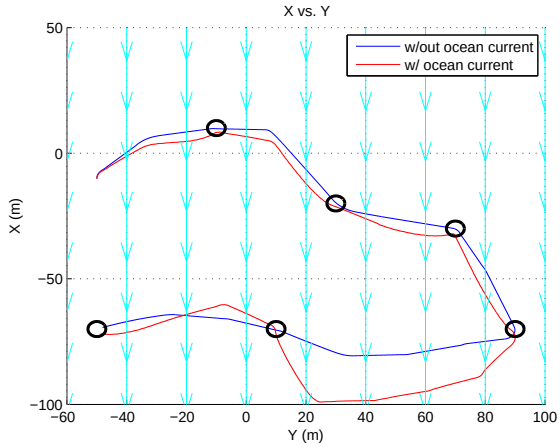


Figure 5.30: Ocean current effects on WPG - OC for $\psi_i = 0$, $u_c^E = -0.1$, $v_c^E = 0$ and $w_c^E = 0$

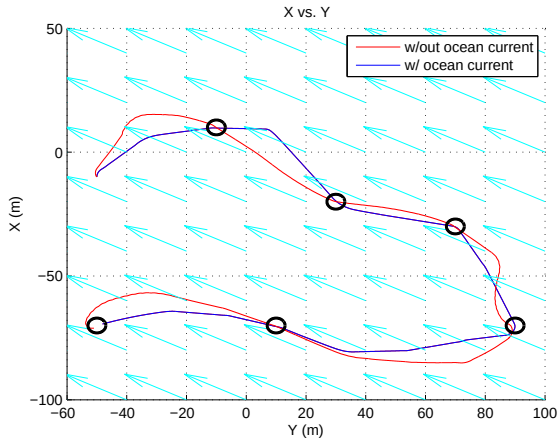


Figure 5.31: Ocean current effects on WPG - OC for $\psi_i = 0$, $u_c^E = 0.1$, $v_c^E = -0.2$ and $w_c^E = 0$

5.5 Obstacle Avoidance

The vehicle may meet obstacles while it travels within the sea. For some environment, position and shape of obstacles can be known. If shape and position of obstacles are unknown, obstacle avoidance is performed with more difficulty in real - time operation.

In this thesis, Algorithm 3 is developed for escaping from obstacles that there is no information about their shapes and positions. If the vehicle has forward looking sonar, it provides acoustic images of the environment. Any object inside sonar beam area is detected by the vehicle.

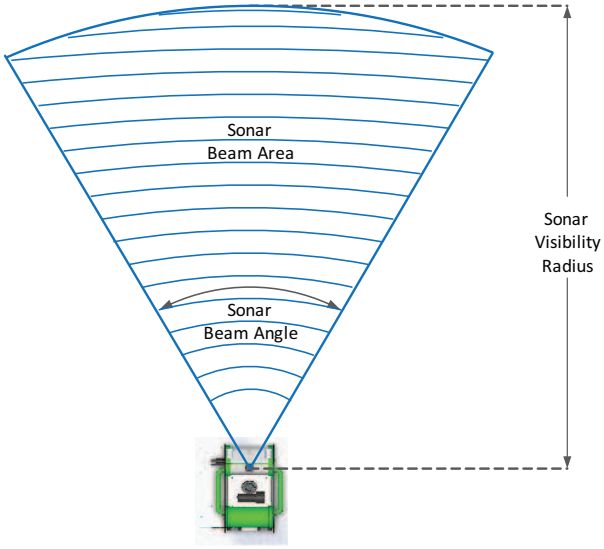


Figure 5.32: Sonar beam angle, sonar visibility distance and sonar beam area

Sonar beam area depends on sonar beam angle and sonar visibility radius as shown in Figure 5.32. For example, *BlueView* P900 - 90 D series sonar has 90 degrees beam angle and maximum range is 100 meters where P900 - 130 D series one has 130 degrees beam angle [1] with the same range. In simulation, P900 - 90 D specifications are used such as beam angle is 90 degrees and range is selected as a 30 meters which is within its optimal range.

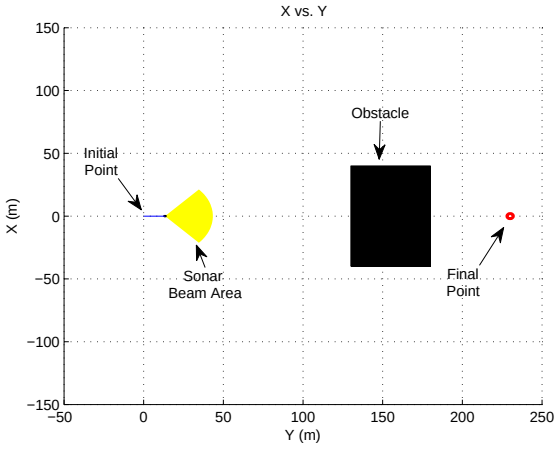


Figure 5.33: Example for obstacle in path

Assume that the vehicle has a guidance law based on line-of-sight guidance which is defined in Section 5.2. Briefly, the vehicle try to minimize error angle, φ, between LOS angle and

heading angle. While the vehicle follows LOS angle, there is an obstacle on its trajectory as shown in Figure 5.33.

The vehicle goes to final point until sonar detects the obstacle. When it meets the obstacle, the vehicle change its heading angle step-by-step as shown in Figure 5.34. The vehicle minimize φ step-by-step if there is no obstacle in sonar beam area and $-\epsilon \leq \varphi \leq \epsilon$.

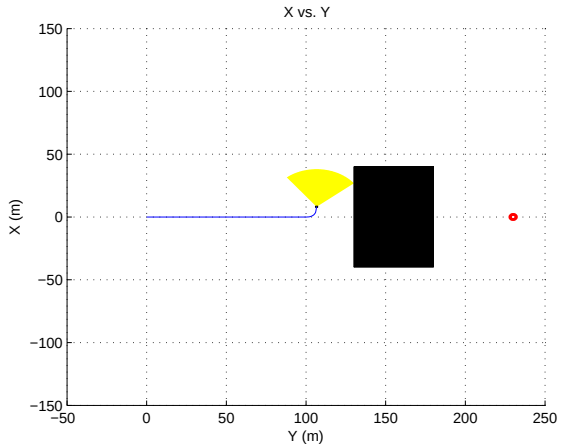


Figure 5.34: Changing heading angle because of detecting the obstacle

In Figure 5.35, the vehicle escapes from obstacle and it minimizes φ. When $\varphi = 0$, it keeps its heading angle to reach final point.

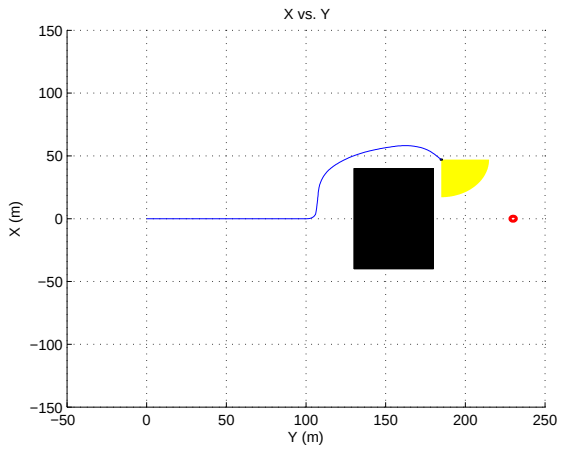


Figure 5.35: The vehicle escaped from obstacle

Finally, the vehicle follows trajectory in order to escape obstacle and reach final point as

shown in Figure 5.36.

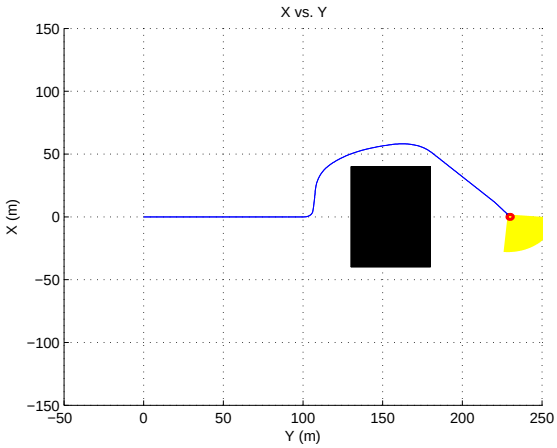


Figure 5.36: Total trajectory of the vehicle when escaping from obstacle

The algorithm works successfully on complex environment as shown in Figure 5.37. Firstly, the vehicle changes its heading angle in order to minimize φ. Then, it meets first obstacle and changes its heading angle. After escaping first one, it meets second one. Finally, it reach final point after escaping second one.

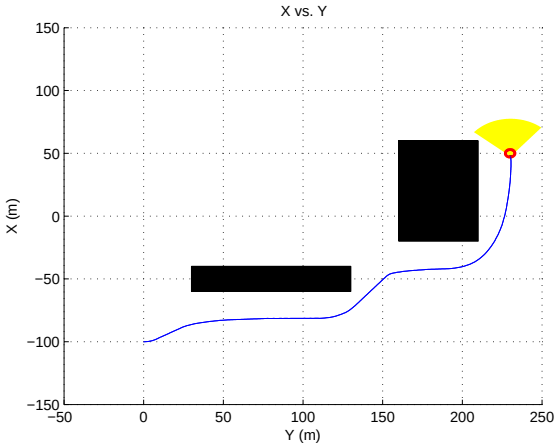


Figure 5.37: Total trajectory of the vehicle when escaping from multiple obstacles

If position and shape of the obstacles are known, it may be more logical (point of operation time and energy consumption) that the vehicle passes between obstacles.

Algorithm 3 Real-time obstacle avoidance algorithm

$\varphi = |\psi - \alpha|$ (ψ:heading angle α: los angle)

if obstacle is detected **then**

change heading angle

if Obstacle stays at right relative to the vehicle **then**

$\psi = \psi - 5$

else if Obstacle stays at left relative to the vehicle **then**

$\psi = \psi + 5$

else

$\psi = \psi - 5$

end if

else if there is a heading error **then**

change heading angle

if $\varphi > 5$ **then**

$\psi = \psi - 5$

else if $\varphi < -5$ **then**

$\psi = \psi + 5$

else

$\psi = \varphi$

end if

end if

5.6 Conclusion

In this chapter, guidance methods and real-time obstacle avoidance algorithm (Algorithm 3) are discussed. Way point guidance by LOS is very common for underwater vehicles. Basic idea behind this method is that heading angle of the vehicle follows LOS angle in the yaw plane. For the pitch plane, depth and surge speed controller is used because of the fact that there is no actuator to create pitch angle. When the vehicle enters circle of acceptance, it is assumed that reaching of present way point is achieved and next way point is selected as a present way point. By combining the pitch plane and the yaw plane way point guidance algorithm, way point guidance by LOS in 3D is achieved. In addition, heading angle, surge speed and heave speed are selectable at the final way point.

Way point guidance is also obtained based on optimal control. The vehicle travels from initial position to final position on infinitely many trajectories that may requires different energy. Because of energy consumption is crucial for miniROV, an optimal control problem (OCP) is derived in order to find a trajectory provides reaching final point with minimum energy. The OCP is constrained non-linear multi variable type function that its solution is obtained constrained non-linear optimization. As a solution method, interior point algorithm (gradient-based) and genetic algorithm (global search) are used. Firstly, the OCP is solved by using GA. Next, optimal solution of GA is used as initial state for IPA. By this way, it is guaranteed that the solution is found by IPA is probably a global optimum one. Finally, the trajectory requires minimum energy is generated. By using this approach, the database is created. The database consists of different optimal trajectories that requires minimum energy. These trajectories are created for different distance, surge speed and heading angle. Hence, the vehicle can go from initial position to final position with minimum energy by using optimal trajectories. The database is created for the yaw plane. In order to obtain way point guidance based on optimal control in 3D space, the method which is used in way point guidance by LOS in the pitch plane is used.

Next, ocean current effects on the path produced by guidance methods are examined. Note that, energy consumption is not concerned when the effects are examined. Behaviour of the vehicle is shown when the vehicle use the two guidance method. Ocean currents can be defined relative to body-fixed frame. If ocean currents are defined relative to earth-fixed frame, coordinate transformation matrix is used in order to find body-fixed frame conjugates.

Finally, real-time obstacle avoidance algorithm (Algorithm 3) is developed by using looking-forward sonar. If the vehicle operated unknown environment, there is no information about shape and position of the obstacles. The sonar is used In order to detect obstacles, looking forward sonar is used. The vehicle escapes obstacles by using this information. The algorithm works properly not only one obstacle but also multiple obstacles.

CHAPTER 6

NAVIGATION

6.1 Introduction

It is quite hard to determine the location of a submerged object. The main reason is that the RF wave cannot penetrate more than a few millimetres underwater, in general. For this reason, GPS (Global Positioning System) is not available for underwater vehicle.

Assume that there are two vehicles; one is submerged and the other one is on the surface and they are connected to each other by fiber optic cable that is also used for data transmission. Submerged vehicle consists of a magnetic sensor and a pressure sensor and surface vehicle has a GPS module and sonar which is located as seeing the submerged vehicle. The surface vehicle knows its position by using GPS, and it also knows the submerged vehicle's position relative itself by using sonar. Magnetic sensor provides heading angle and depth of the submerged vehicle is calculated by using pressure sensor. If all of these information come together, submerged vehicle position in 3D space and orientation can be found.

It is assumed that system explained above is created. The system provides GPS-like measurements and heading measurement with high covariances. Non-linear filter such as the Extended Kalman Filter, the Unscented Kalman Filter, the Particle Filter etc. is used in order to filter noisy data, since the system model is non-linear. In this thesis, the extended kalman filter (EKF) is used for estimation.

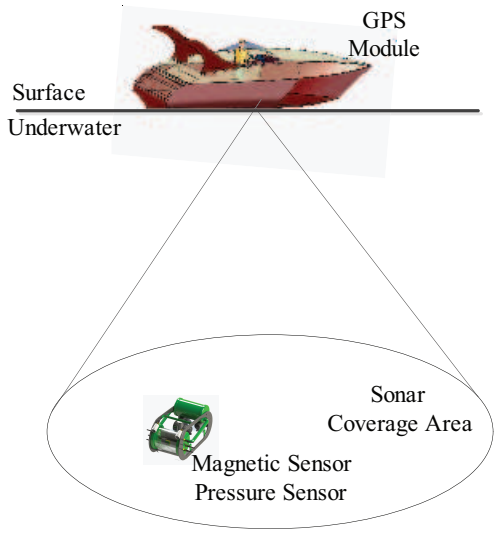


Figure 6.1: Submerged and surface vehicles for navigation

6.2 System Model Representation

In Chapter 4, the dynamical model of DST-R-100-4 guarantees that there is no roll rate during vehicle's operation. Also, sway speed, v, is close to zero and ϕ and v is neglected. Hence, the system model has the following forms.

$$\begin{aligned} \dot{x} &= u\cos(\psi)\cos(\theta) + w\cos(\psi)\sin(\theta) \\ \dot{y} &= u\sin(\psi)\cos(\theta) + w\sin(\psi)\sin(\theta) \\ \dot{z} &= -u\sin(\theta) + w\cos(\theta) \\ \dot{\psi} &= r/\cos(\theta) \end{aligned} \tag{6.1}$$

6.3 Extended Kalman Filter (EKF)

Extended kalman filter is a non-linear version of Kalman Filter. The fundamental concept of this filter involves the notation that the true state is sufficiently close to be estimated state. Therefore, the error dynamics can be represented fairly accurately by a linearized first order Taylor series expansion. More details about EKF can be found in [9], [14], and [15].

Truth model with discrete-time measurements can be written as follows.

$$\begin{aligned} \bar{x}_k &= f(\bar{x}_{k-1}, \bar{u}_{k-1}) + \bar{w}_{k-1} \\ \bar{z}_k &= h(\bar{x}_k) + \bar{v}_k \end{aligned} \tag{6.2}$$

where $\bar{w}_k$ and $\bar{v}_k$ are the process and measurement noises, respectively. They are assumed to be zero mean Gaussian noises with covariance Q_k and R_k, respectively.

In order to compute predicted state from previous estimate, function f is used. Similarly, function h is used for computing predicted measurement from the predicted state. f and h cannot be applied directly to covariances, thus Jacobian of them is computed. By this way, non-linear system model is linearized around current estimate. At each iteration, Jacobian is found by using current predicted states. Generally, EKF divided into two phases such as predict and update. In order to produce an estimate of the state at current time, the predicted phase uses the state estimate from previous time. It is also called as priori state estimate since it does not consist observation information from current time. The update phase current observation information is combined with priori state estimate in order to refine the state estimate, which is also called as posteriori state estimate. First two equations in Equation 6.3 represent predict phase, others represent update phase.

Discrete-time EKF can be summarized as follows.

$$\begin{aligned} &\textit{A priori state estimate} && \hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, \bar{u}_{k-1}) \\ &\textit{A priori estimate covariance} && P_{k|k-1} = F_{k-1} P_{k-1|k-1} {F_{k-1}}^T + Q_{k-1} \\ &\textit{Innovation} && \tilde{y}_k = \bar{z}_k - h(\hat{x}_{k|k-1}) \\ &\textit{Innovation covariance} && S_k = H_k P_{k|k-1} {H_k}^T + R_k \\ &\textit{Kalman gain} && K_k = P_{k|k-1} {H_k}^T {S_k}^{-1} \\ &\textit{A posteriori state estimate} && \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \\ &\textit{A posteriori covariance} && P_{k|k} = (I - K_k H_k) P_{k|k-1} \end{aligned} \tag{6.3}$$

where

$$F_{k-1} = \left.\frac{\partial f}{\partial \bar{x}}\right|_{\hat{x}_{k-1|k-1}, \bar{u}_{k-1}} \qquad H_k = \left.\frac{\partial h}{\partial \bar{x}}\right|_{\hat{x}_{k|k-1}} \tag{6.4}$$

$\hat{x}_{k|k}$ denotes a posteriori state estimate at time k given observation and including at time k. $P_{k|k}$ denotes a posteriori error covariance matrix.

EKF can be started with $\hat{x}_{0|0} = \hat{x}_0$ and $P_{0|0} = E[\bar{x_0} - \hat{x}_{0|0}]$. Equation 6.1 can be written in discrete-time form as follows.

$$\begin{aligned} x_k &= x_{k-1} + u_{k-1}\cos(\psi_{k-1})\cos(\theta_{k-1}) + w_{k-1}\cos(\psi_{k-1})\sin(\theta_{k-1}) \\ y_k &= y_{k-1} + u_{k-1}\sin(\psi_{k-1})\cos(\theta_{k-1}) + w_{k-1}\sin(\psi_{k-1})\sin(\theta_{k-1}) \\ z_k &= z_{k-1} - u_{k-1}\sin(\theta_{k-1}) + w_{k-1}\cos(\theta_{k-1}) \\ \psi_k &= \psi_{k-1} + r_{k-1}/\cos(\theta_{k-1}) \end{aligned} \tag{6.5}$$

In Equation 6.5, θ and ψ are measured by using 3-axis magnetic compass that is located at the submerged vehicle. Moreover, u and w are measured by using doppler velocity logger (DVL).

6.4 Simulation Results

In Figure 6.2, real path is shown. In this simulation, the vehicle just only change its position in the yaw plane. There is no change along Z direction.

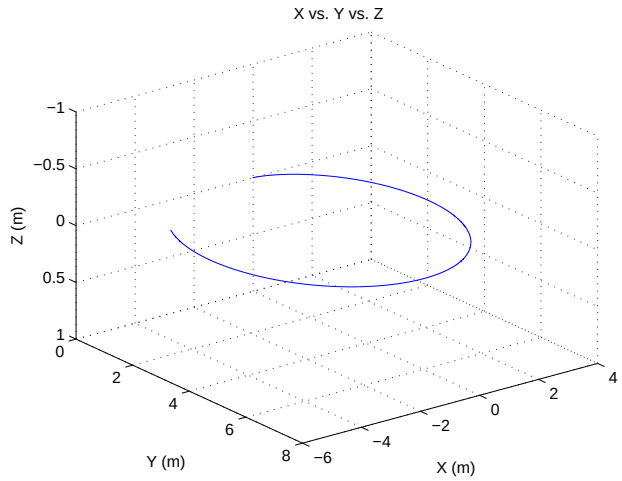


Figure 6.2: Real path for simulation 1

In Figure 6.3, $\hat{x}_0$ is selected as [0 0 0 0]. According to results, it can be said that EKF converges real values successfully.

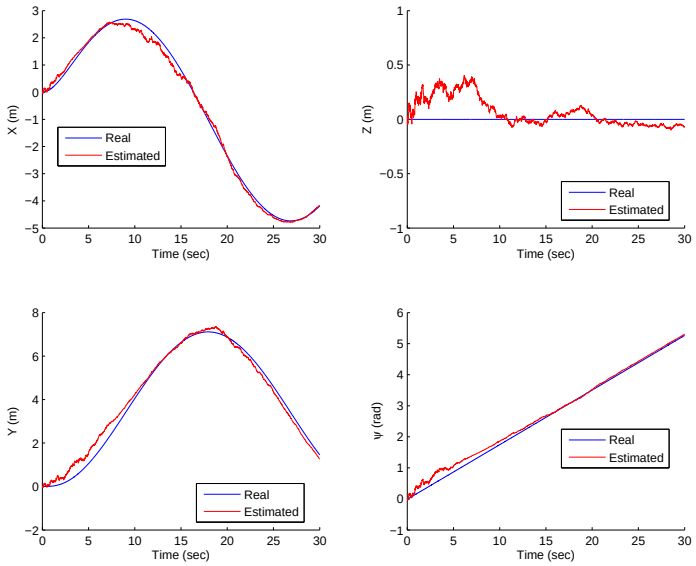


Figure 6.3: EKF results for simulation 1, $\hat{x}_0$ = [0 0 0 0]

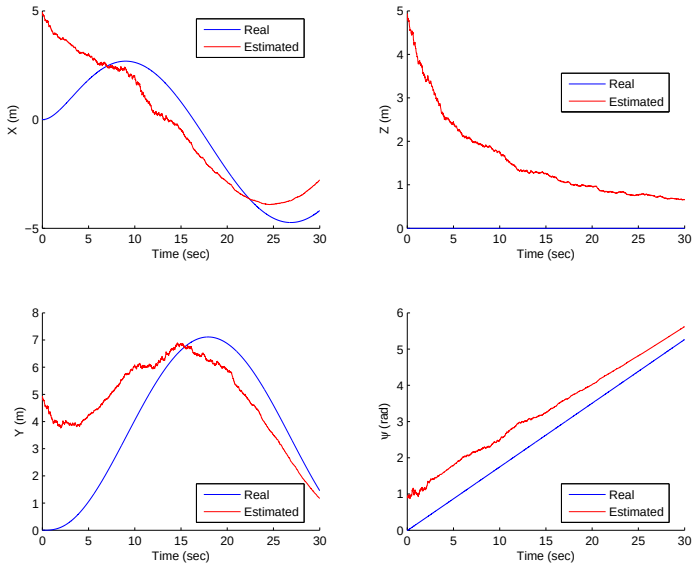


Figure 6.4: EKF results for simulation 1, $\hat{x}_0$ = [5 5 5 1]

In Figure 6.4, $\hat{x}_0$ is changed to [5 5 5 1] and EKF converges to real values.

In Figure 6.5, it is assumed that noise magnitude suddenly increases ten times of after 15

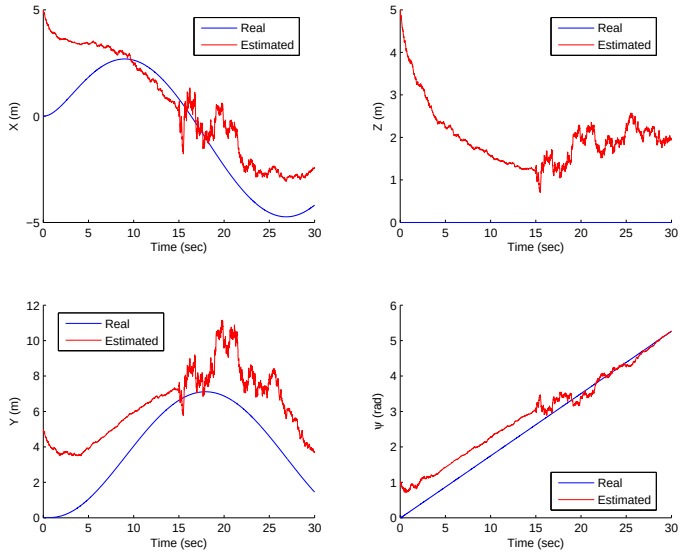


Figure 6.5: EKF results under different noise for simulation 1, $\hat{x}_0$ = [5 5 5 1]

seconds. Under this condition, EKF results are not satisfying.

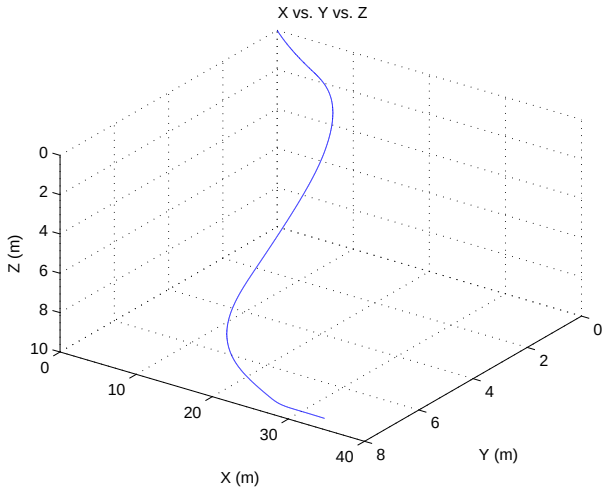


Figure 6.6: Real path for simulation 2

The vehicle starts its motion from (0,0,0) and real path is shown in Figure 6.2. In this simulation, the vehicle makes movement in 3D space.

In Figure 6.7, $\hat{x}_0$ is selected as [0 0 0 0]. According to results, it can be said that EKF

converges real values successfully.

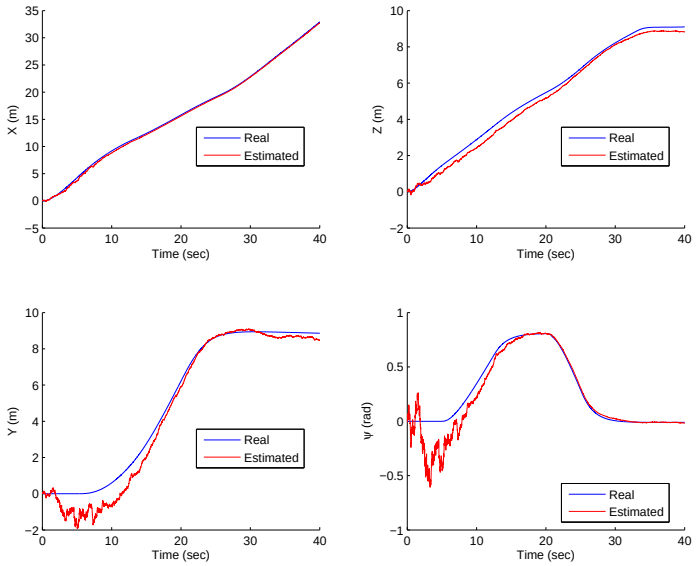


Figure 6.7: EKF results for simulation 2, $\hat{x}_0$ = [0 0 0 0]

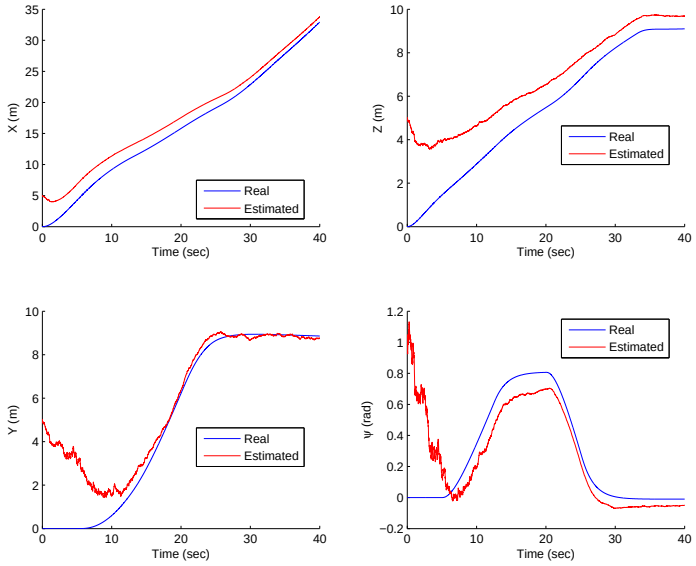


Figure 6.8: EKF results for simulation 2, $\hat{x}_0$ = [5 5 5 1]

In Figure 6.8, $\hat{x}_0$ is changed to [5 5 5 1] and EKF converges to real values.

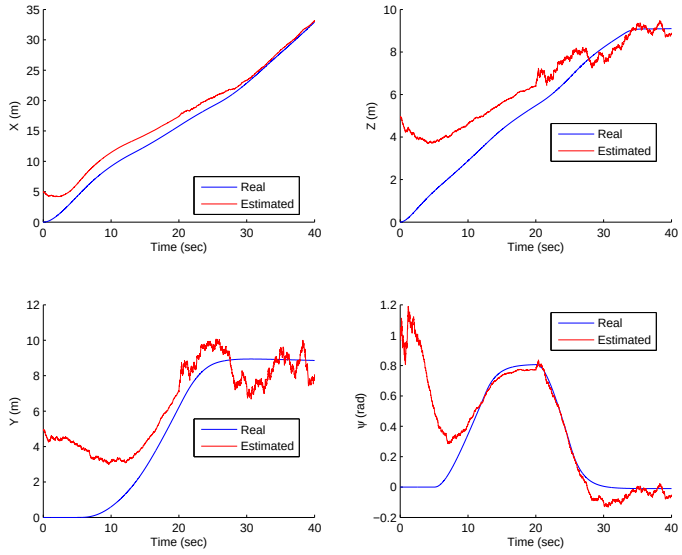


Figure 6.9: EKF results under different noise for simulation 2, $\hat{x}_0$ = [5 5 5 1]

In Figure 6.9, it is assumed that noise magnitude suddenly increases ten times of after 20 seconds. Under this condition, EKF results are not satisfying.

6.5 Conclusion

In this chapter, it is assumed that there is a navigation system that provides GPS-like measurements with high covariances. Basically, the system works as follows: Underwater vehicle has several sensors such as a pressure sensor and a magnetic compass that provide depth and heading angle of the vehicle. Next, there is a surface vehicle that has GPS module and a sonar whose beam covers the underwater vehicle that provides submerged vehicle position relative to the surface vehicle. By putting all these informations together, submerged vehicle position and heading are obtained. Under this assumption, extended kalman filter is designed in order to filter noisy data. EKF works successfully except at very high noise. If noise increases, EKF performance decreases. By designing EKF, it is shown that if considered system would be realized, there will be new point of view about underwater vehicle navigation problem.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summary

Main motivation behind this study is autopilot and guidance designs for the DST-R-100-4 mini ROV, and examining the effects of ocean currents on the path produced by the vehicle, developing real-time obstacle avoidance strategy and navigation.

Mathematical model of DST-R-100-4 ROV is obtained by considering the vehicle as a rigid-body. The model has non-linearity caused by hydrodynamic forces and moments. In order to design the autopilot(s), the model is linearized around some trim points. Most effective way to determine hydrodynamic coefficients is pool tests, and there is no exact way to calculate them. Approximate values of them are calculated by using the strip theory and CFD results that are acquired from SolidWorks. By using these parameters, behaviours of the vehicle under different inputs are examined. According to results, limiting values for surge, heave, sway and yaw speed are found. Moreover, stability check is performed based on these studies. PID controller technique is used for surge speed, heading and depth controllers (autopilots).

As for guidance, two methods are presented: way point guidance by line-of-sight and way point guidance based on optimal control theory. Guidance in 3D is obtained by using both of them, separately. Way point guidance by LOS is a very common approach for the guidance of underwater vehicles. Briefly, in this method, there are several way points that the vehicle is required to reach sequentially. When the vehicle enters the circle of acceptance of a way point, it is assumed that the vehicle has actually reached that particular way point and the next way point is selected as the way point to be attended. In addition, heading angle, surge speed and heave speed are chosen as variables to be satisfied for the final way point.

Main idea behind the way point guidance based on optimal control theory is finding the trajectory that requires minimum energy. The vehicle may travel between two points along infinitely many trajectories each of which requires spending different energy. Hence, an optimal control problem (OCP) is presented which aims to find the optimum trajectory. The OCP is solved by using two different methods, interior point method (gradient-based) and genetic algorithm (global search). Optimal solution which is obtained by using the genetic algorithm is used as the initial point for the interior point algorithm. By this way, it is guaranteed that the solution found is probably a global optimum one. In order to make this idea applicable for arbitrary initial states, a database is created. Next, both guidance methods are examined under the effects of ocean currents. Studies about this topic were presented at SAVTEK and TOK at first, [11], [12], and [24].

Obstacles in the simulations make the environment more realistic because any sea or lake can not be expected to be clear of obstacles. Therefore, dealing with obstacles in the simulation environment is very important. There are many obstacle avoidance algorithm and strategies, but only one real-time obstacle avoidance algorithm is examined in this study. The real-time obstacle avoidance algorithm is developed based on the measurements obtained from a looking-forward sonar. If the vehicle is operated in an unknown environment, there is no information about shape and position of obstacles. The sonar is used to detect the obstacles and the vehicle is supposed to escape from them by using this information.

Finally, this study is ended with a simple and straight-forward introduction to the navigation of the vehicle. Since, RF signals are absorbed by water, RF communication is not possible underwater. Therefore, it is suggested that there are one submerged and one surface vehicle operating together. Position and heading of the underwater vehicle are obtained by using sensors and sonar. Briefly, the system works as follows: Underwater vehicle has several sensors such as a pressure and a magnetic compass providing depth and heading angle of the vehicle. Next, there is the surface vehicle having a GPS module and a sonar whose beam covers the underwater vehicle. By this way, it provides the position of the submerged vehicle relative to itself. By putting all these informations together, submerged vehicle's position and heading with high covariances are obtained. Under this assumption, an extended Kalman filter (EKF) is designed in order to filter the noisy data and fusing all the available information for navigation.

7.2 Results and Future Work

First of all, DST-R-100-4 was not ready for pool tests when this study was begun. Therefore, parameter derivation is done under some assumptions and approximate calculation techniques. First pool test was done in May 2012. The simulation results and vehicle's real physical behaviour was observed to be very close to each other, especially as regards to maximum speeds, passive stability, manoeuvre capabilities. Depending on these test results, it can be said that the mathematical model of the vehicle is valid. However, in order to make the model more realistic, thruster model can be added as part of the Simulink model. Moreover, system identification can be used for verifying accuracy of the dynamical model of the vehicle.

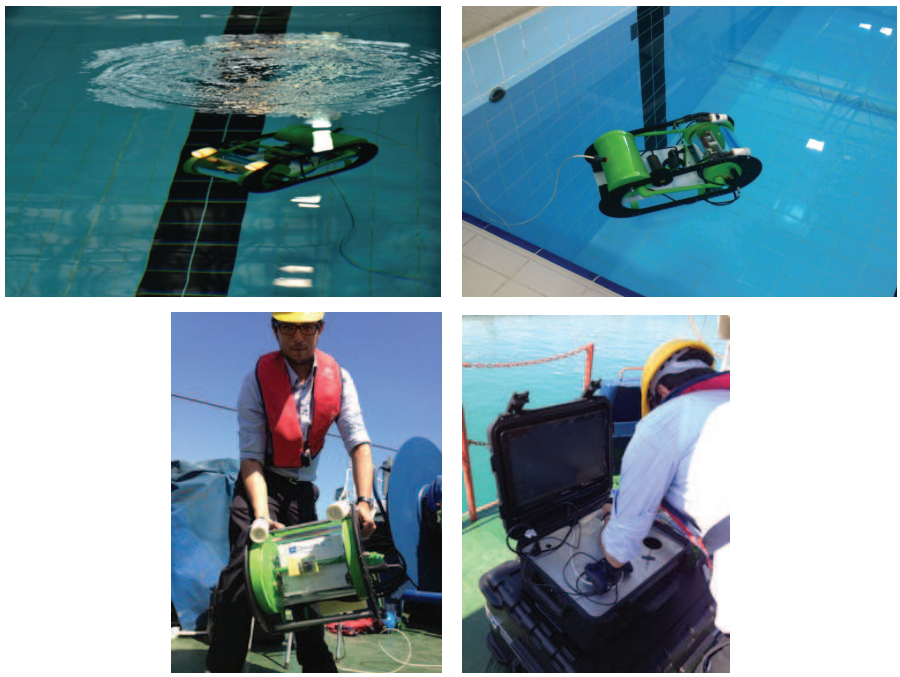


Figure 7.1: Pictures from pool and sea tests

Since PID control is much simpler than other control techniques such as linear-quadratic regulator (LQR), sliding mode control and requires less computational need, it is preferred. But, other control techniques such as sliding mode control, linear quadratic regulator (LQR) may provide more effective results. Performance of the WPG - OC is proportional with size of the database. If the size of the database is sufficiently large, energy consumption may be decrease. For this reason, when creating database, final heading angle (ψ_{db}) and distance (d_{db}) may be divided into more than 5 and 3, respectively.

Real-time obstacle avoidance is done successfully by using a looking-forward sonar under the assumption that the sonar detects any obstacle in a suitable range and the vehicle knows it properly. This is the outcome of perfect sonar image processing. Hence, this is another topic to be searched before using an obstacle avoidance algorithm.

The EKF is used in order to filter the noisy data. According to simulation results, performance of the EKF decreases while noise increases. Other non-linear filters such as the unscented kalman filter, the particle filter may provide more effective results.

REFERENCES

[1] *BlueView P Series 2D Deepwater Imaging Sonar*.

[2] G. Antonelli and G. Antonelli. *Underwater Robots*. Springer-Verlag Berlin/Heidelberg, 2006.

[3] G. Antonelli, S. Chiaverini, R. Finotello, and R. Schiavon. Real-time path planning and obstacle avoidance for rais: an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 26(2):216–227, 2001.

[4] R. Bachmayer, L. Whitcomb, and M. Grosenbaugh. An accurate four-quadrant nonlinear dynamical model for marine thrusters: Theory and experimental validation. *IEEE Journal of Oceanic Engineering*, 25(1):146–159, 2000.

[5] V. Bakaric, Z. Vukic, and R. Antonic. Improved basic planar algorithm of vehicle guidance through waypoints by the line of sight. In *First International Symposium on Control, Communications and Signal Processing*, pages 541–544. IEEE, 2004.

[6] M. S. Bazaraa. *Nonlinear Programming*. Wiley Online Library, 1993.

[7] B. Bingham and W. Seering. Hypothesis grids: Improving long baseline navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 31(1):209–218, 2006.

[8] C. Brenned. *A Review of Added Mass and Fluid Inertial*. Department of The Navy, Naval Civil Engineering Laboratory, California, 1982.

[9] E. Brookner and J. Wiley. *Tracking and Kalman filtering made easy*. Wiley Online Library, 1998.

[10] M. Caccia and G. Veruggio. Guidance and control of a reconfigurable unmanned underwater vehicle. *Control Engineering Practice*, 8(1):21–37, 2000.

[11] F. Y. Cevher, S. Karadeniz Kartal, and M. K. Leblebicioğlu. Bir İnsansız sualtı aracı: DST-R-100-4 için optimal güdüm - otopilot tasarımı. 2012.

[12] F. Y. Cevher, S. Karadeniz Kartal, and M. K. Leblebicioğlu. İnsansız sualtı aracı için optimal sapma otopilot tasarımı. 2012.

[13] E. Chong and S. Żak. *An Introduction to Optimization*. Wiley-interscience, 2004.

[14] C. Chui and G. Chen. *Kalman filtering: with real-time applications*. Springer Verlag, 2009.

[15] J. Crassidis and J. Junkins. *Optimal estimation of dynamic systems*, volume 24. Chapman & Hall, 2011.

[16] E. De Souza and N. Maruyama. Intelligent uuvs: Some issues on rov dynamic positioning. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):214–226, 2007.

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

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

[19] N. Gracias, S. Van Der Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic-based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 28(4):609–624, 2003.

[20] A. Healey. Guidance laws, obstacle avoidance and artificial potential functions. *IEEE Control Engineering Series*, 69:43, 2006.

[21] A. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339, 1993.

[22] A. Healey, S. Rock, S. Cody, D. Miles, and J. Brown. Toward an improved understanding of thruster dynamics for underwater vehicles. *IEEE Journal of Oceanic Engineering*, 20(4):354–361, 1995.

[23] P. Jantapremjit and P. Wilson. Guidance-control based path following for homing and docking using an autonomous underwater vehicle. pages 1–6, 2008.

[24] S. Karadeniz Kartal, F. Y. Cevher, and M. K. Leblebicioğlu. Bir İnsansız sualtı aracı: DST-R-100-4 için 3D optimal otopilot tasarımı. 2012.

[25] D. Kirk. *Optimal Control Theory: An Introduction*. Dover Pubns, 2004.

[26] S. Lee, K. Sohn, S. Byun, and J. Kim. Modeling and controller design of manta-type unmanned underwater test vehicle. *Journal of mechanical science and technology*, 23(4):987–990, 2009.

[27] S. Leyffer and A. Mahajan. Nonlinear constrained optimization: Methods and software. *Argonee National Laboratory, Argonne, Illinois*, 60439, 2010.

[28] D. Luenberger and Y. Ye. *Linear and Nonlinear Programming*, volume 116. Springer Verlag, 2008.

[29] MathWorks. Constrained nonlinear optimization algorithms, 2012.

[30] MathWorks. Nonlinear programming, 2012.

[31] D. McLean. *Automatic Flight Conrol Systems*. Design, Analysis and Research Corporation, 1998.

[32] W. Naeem, R. Sutton, S. Ahmad, and R. Burns. A review of guidance laws applicable to unmanned underwater vehicles. *Journal of Navigation*, 56(1):15–29, 2003.

[33] J. Newman. *Marine Hydrodynamics*. MIT Press, 8th edition, 1977.

[34] K. Ogata. *Modern Control Engineering*. Prentice Hall PTR, 2001.

[35] Y. Petillot, I. Tena Ruiz, and D. Lane. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE Journal of Oceanic Engineering*, 26(2):240–251, 2001.

[36] G. Roberts and R. Sutton. *Advances in unmanned marine vehicles*, volume 69. Iet, 2006.

[37] J. Roskam. *Airplane Flight Dynamics and Automatic Flight Controls*. Design, Analysis and Research Corporation, 1998.

[38] I. Spangelo and O. Egeland. Trajectory planning and collision avoidance for underwater vehicles using optimal control. *IEEE Journal of Oceanic Engineering*, 19(4):502–511, 1994.

[39] R. Sutton, R. Burns, and P. Craven. Intelligent steering control of an autonomous underwater vehicle. *Journal of Navigation*, 53(03):511–525, 2000.

[40] W. Wang and C. Clark. Modeling and simulation of the videoray pro iii underwater vehicle. *OCEANS 2006-Asia Pacific*, pages 1–7, 2007.

[41] Y. Wang, L. Han, M. Li, Q. Wang, J. Zhou, and M. Cartmel. A real-time path planning approach without the computation of cspace obstacles. *Robotica*, 22(02):173–187, 2004.

[42] K. Wendel. *Hydrodynamic Masses and Hydrodynamic Moments of Inertia*. Department of The Navy,The David W. Taylor Model Basin, Washington , D.C., 1956.

[43] P. Wilson. Autonomous homing and docking tasks for an underwater vehicle. 2009.

[44] X. Wu, Z. Feng, J. Zhu, and R. Allen. Line of sight guidance with intelligent obstacle avoidance for autonomous underwater vehicles. pages 1–6, 2006.

[45] Z. Xuhui. Parameter identification of rov by decoupled dynamical models with projective mapping method, 1995.

[46] S. Zanoli and G. Conte. Remotely operated vehicle depth control. *Control engineering practice*, 11(4):453–459, 2003.

[47] P. Zipfel. *Modelling and Simulation of Aerospace Vehicle Dynamics*. Prentice Hall, NY, 1990.

APPENDIX A

Coordinate Transformation Matrix

A.1 Linear Velocity Transformation Matrix

$$T_{BE,1}(\eta_2) = R_z(\psi)R_y(\theta)R_x(\phi) \tag{A.1}$$

$$T_{BE,1}(\eta_2) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi)0 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta)\cos(\psi) & -\sin(\psi) & \sin(\theta)\cos(\theta) \\ \cos(\theta)\sin(\theta) & \cos(\psi) & \sin(\theta)\sin(\psi) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\cos(\phi)\sin(\theta) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\psi)\sin(\phi) + \sin(\theta)\sin(\psi)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}$$

A.2 Angular Velocity Transformation Matrix

$$\upsilon_2 = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x(\phi)^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x(\phi)^T R_y(\theta)^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = T_{BE,2}^{-1}(\eta_2)\,\dot{\eta}_2 \tag{A.2}$$

$$\begin{aligned}
\upsilon_2 &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}^T \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}\cos(\phi) \\ -\dot{\theta}\sin(\phi) \end{bmatrix} + \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\phi)\sin(\theta) & \cos(\phi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta) & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}\cos(\phi) \\ -\dot{\theta}\sin(\phi) \end{bmatrix} + \begin{bmatrix} -\dot{\psi}\sin(\theta) \\ \dot{\psi}\sin(\phi)\cos(\theta) \\ \dot{\psi}\cos(\phi)\cos(\theta) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_{BE,2}^{-1}(\eta_2)\,\dot{\eta}_2
\end{aligned}$$

Thus,

$$T_{BE,2}(\eta_2) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \tag{A.3}$$

APPENDIX B

Relation Between Dissipated Power and Thrust

Spangelo [38] points out that dissipated electrical power has a relation with thruster force, denoted by T. Proof of the relation is as follows.

Equation B.1 shows load torque from propeller.

$$Q = \rho D^5 K_Q(J_0) n|n| \tag{B.1}$$

Equation B.1 shows thrust force.

$$T = \rho D^4 K_T(J_0) n|n| \tag{B.2}$$

Dynamical model of DC motor can be formulated as:

$$L_a \dot{i_a} = -R_a i_a - 2\pi K_M n + u_a \tag{B.3}$$

$$2\pi J_m \dot{n} = K_M i_a - Q \tag{B.4}$$

Equation B.4 can also be written as the following form.

$$i_a = \frac{2\pi J_m}{K_M}\dot{n} + \frac{Q}{K_M} \tag{B.5}$$

Assuming dynamic of the armature circuit is fast in Equation B.3 yields Equation B.6

$$u_a = R_a i_a + 2\pi K_M n \tag{B.6}$$

Dissipated electrical power can be written as:

$$P = u_a i_a \tag{B.7}$$

Substituting Equation B.6 into Equation B.7, the following equation is obtained.

$$P = R_a {i_a}^2 + 2\pi K_M n i_a \tag{B.8}$$

Right hand-side of Equation B.8 has two terms which refer dissipation in armature resistance and conversion from electrical to mechanical power. The dissipation in armature resistance is so small, then $R_a {i_a}^2$ is negligible. If Equation B.5 is substituted into remaining part of Equation B.8, the following equation is obtained.

$$P = (2\pi)^2 J_m n\dot{n} + 2\pi Q n \tag{B.9}$$

First term of Equation B.9 refers power needed to accelerate the propeller, second terms refers power dissipation due to interaction between propeller and water. Power dissipation for motor is written as:

$$P = 2\pi Q n \tag{B.10}$$

Substituting Equation B.1 into Equation B.10 yields the following equation.

$$P = 2\pi\rho D^5 K_Q(J_0) n^2 |n| \tag{B.11}$$

Equation B.2 can also be written as:

$$n^2 = \frac{T}{\rho D^4 K_T(J_0)} \tag{B.12}$$

Result of square root of Equation B.12 yields the following equation.

$$n = \frac{|T|^{0.5}}{\sqrt{\rho} D^2 |K_T(J_0)}|^{0.5} \tag{B.13}$$

Substituting Equation B.12 and Equation B.13 into Equation B.11 yields the following equation.

$$P = \frac{2\pi K_Q(J_0)}{\sqrt{\rho} D |K_T(J_0)|^{1.5}} |T|^{1.5} \tag{B.14}$$

APPENDIX C

Simulink Models

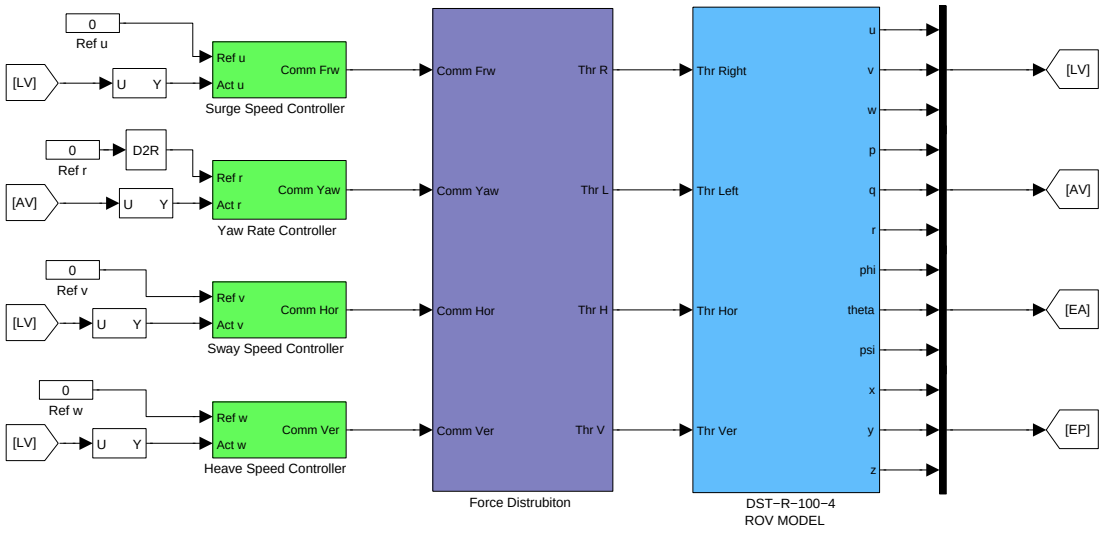


Figure C.1: Simulink model for speed controllers

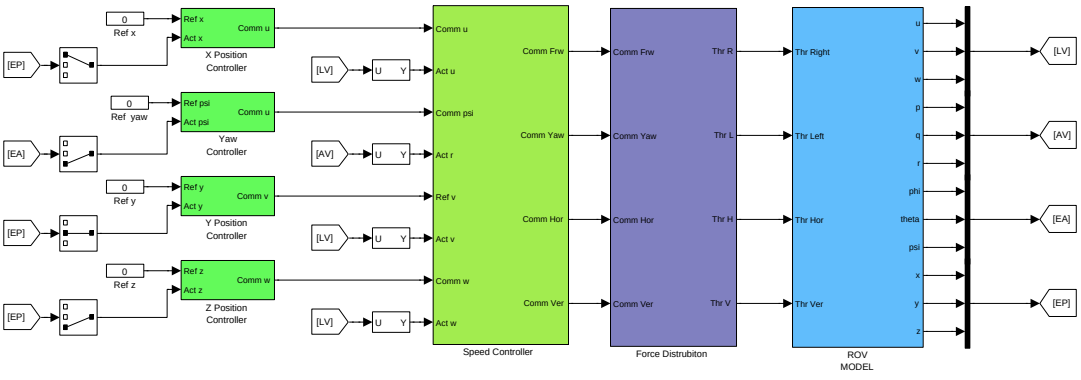


Figure C.2: Simulink model for position controllers

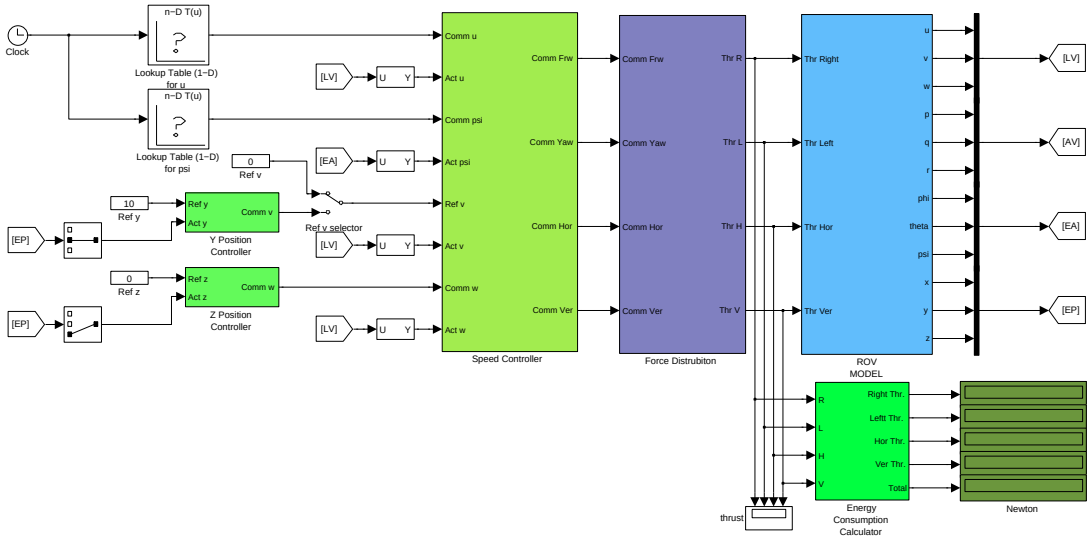


Figure C.3: Simulink model for optimal yaw plane controller

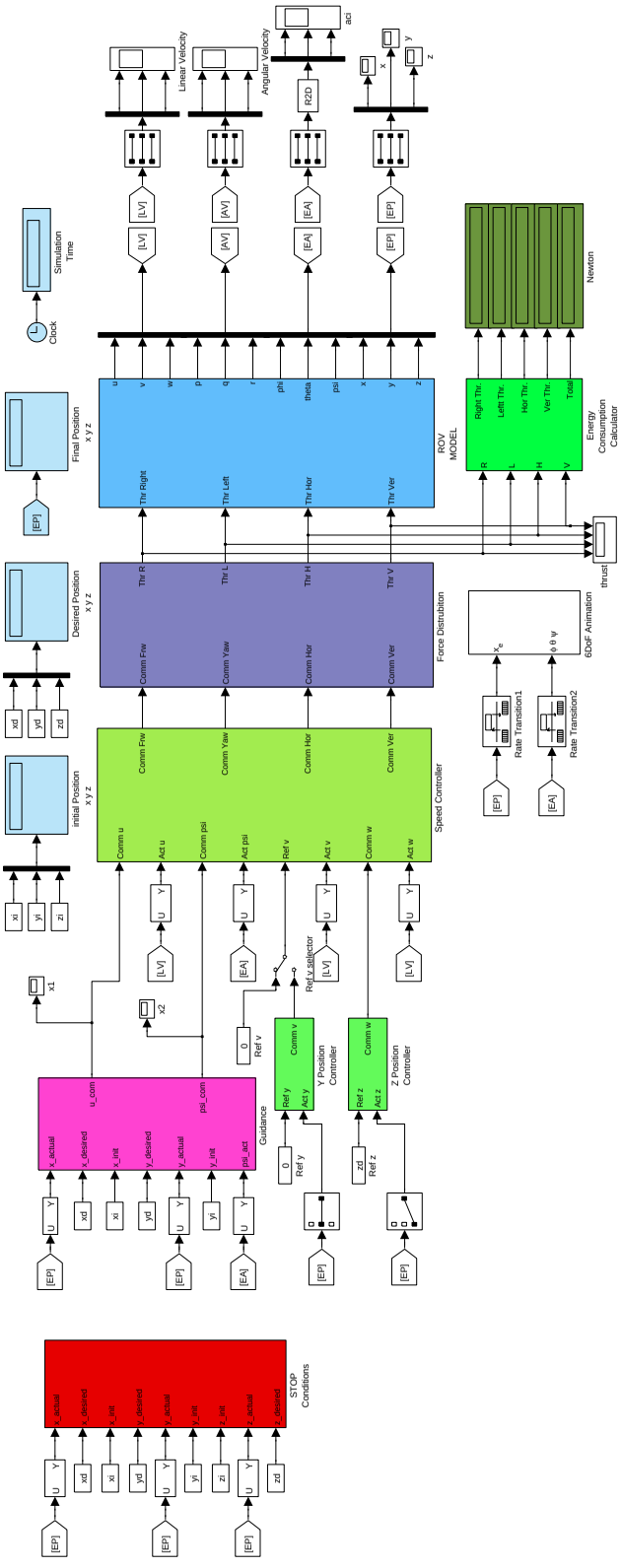


Figure C.4: Simulink model for way point guidance by LOS