NAVIGATION AND CONTROL OF AN UNMANNED SEA SURFACE VEHICLE

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

BY

MURAT KUMRU

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

SEPTEMBER 2015

Approval of the thesis:

NAVIGATION AND CONTROL OF AN UNMANNED SEA SURFACE VEHICLE

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

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

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

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

Examining Committee Members:

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

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

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

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

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

Date: **08/09/2015**

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

Name, Last Name: MURAT KUMRU

Signature :

ABSTRACT

NAVIGATION AND CONTROL OF AN UNMANNED SEA SURFACE VEHICLE

Kumru, Murat
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Kemal Leblebicioğlu

September 2015, 147 pages

In this study, navigation and control algorithms for unmanned sea surface vehicles are investigated. For this purpose, firstly the mathematical model of a sea surface vehicle with two propellers providing stable maneuvering capabilities is constructed considering Newton-Euler equations. The next phase is to design a suitable navigation algorithm which integrates the solutions of "Inertial Navigation System (INS)" and external aids such as "Global Navigation Satellite System (GNSS)" and magnetometer. At this step, different loosely coupled integration algorithms are developed, and their performances are compared. After that, a model boat is assembled with necessary electrical equipment and driving system as a test platform for the navigation implementations. The results of the navigation algorithm obtained by processing real data collected from the model boat are presented. Meanwhile, the corrected navigation solution is also utilized within a parallel independent study aiming to identify the real parameters of the mathematical model of the boat. Finally, the design of various autopilot algorithms is studied taking the improved mathematical model into

account. LQR and “Feedback Linearization” based controllers are realized for this job. The results of the controllers are provided and compared.

Keywords: Sea surface vehicles, mathematical modeling, navigation, LQR, Feedback Linearization controller

ÖZ

BİR İNSANSIZ SU ÜSTÜ ARACININ NAVİGASYONU VE DENETİMİ

Kumru, Murat
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Eylül 2015 , 147 sayfa

Bu çalışmada, insansız su üstü araçları için navigasyon ve denetim algoritmaları incelenmiştir. Bu amaçla, ilk olarak çift pervaneli oluşu sebebiyle yüksek manevra kabiliyetine sahip bir su üstü aracı için bir matematiksel model oluşturulmuştur. Bu model Newton-Euler denklemleri göz önünde bulundurularak inşa edilmiştir. Sonraki aşamada "Ataletsel Navigasyon Sistemi", "Küresel Navigasyon Sistemi" ve manyetometre çözümlerini birleştiren bir navigasyon sistemi tasarlanmıştır. Bu süreçte, farklı tip gevşek bağlı mimariler geliştirilerek performansları kıyaslanmıştır. Daha sonra, bir model tekneye gerekli elektriksel donanım ve sürüş sistemi monte edilerek test altyapısı hazırlanmıştır. Model tekne üzerinden toplanan gerçek verilerin navigasyon algoritması tarafından işlenmesiyle elde edilen çözüm detaylı bir şekilde sunulmuştur. Düzeltmiş bu navigasyon çözümü, paralelde yürütülen bağımsız bir çalışma tarafından model teknenin gerçek matematiksel parametrelerinin belirlenmesi amacıyla kullanılmıştır. Son olarak, bu iyileştirilmiş modeli temel alan çeşitli otopilot algoritmaları gerçeklenmiştir. Bu iş için LQR ve geri beslemeli doğrusallaştıran tipi iki

farklı denetleç kullanılmıştır. Elde edilen sonuçlar karşılaştırmalı olarak verilmiştir.

Anahtar Kelimeler: Su üstü araçlar, matematiksel modelleme, navigasyon, LQR, geribeslemeli doğrusallaştıran denetleç

To Roza Hakmen

and the memory of Marcel Proust

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Professor Kemal Leblebicioğlu, who has been a great guide for me. His advice on both research and my career have been invaluable.

My sincere thanks also goes to my colleague Kağan Erünsal, who helped make this research experience more productive and enjoyable with his inspirational ideas.

I would like to thank Kenan Ahıska for his insightful comments at each stage of this study. He has always been trying to help and give best suggestions.

I am really grateful to Cumhur Çakmak for his vital help throughout exhausting and long experiment days in Yalıncak Lake. I am so lucky to have him as my close friend.

I want to thank Elif Sarıtaş for the unconditional and endless support. She has never lost faith in me even if I strongly disagree with her. I sincerely appreciate her sacrifice, creativity and cheerfulness.

I also wish to express my warm thanks to Selen Sarıtaş who have generously contributed to the process of review of this dissertation.

I would like to show my gratitude to Mustafa Burak Gürcan for his close interest and great effort to make the realization of this work possible.

The technical aspects of the experimental work presented in this study would not have been possible without the assistance of Onur Cem Erdoğan, Kadir Emlik, Serdar Görür and Mehmet Cihan. I am truly grateful to them.

I would like to express my thanks to Rasim Aşkın Dilan, Mehram E. Noudeh and Sonnur Öztürk for their kind assistance in the preparation of the introductory film of the experimentation process.

I would like to acknowledge ASELSAN Inc. for their financial and technical support.

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

The last but not the least, none of this would have been possible without the love and patience of my family. I can hardly express how grateful I am to my mother, my father, and brother for all of the sacrifices they have made.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

USSV	Unmanned Sea Surface Vehicles
DOF	Degrees of Freedom
PID	Proportional-Integral-Derivative
ECEF	Earth-Centered, Earth-Fixed
ECI	Earth-Centered Inertial
GLONASS	Global Orbiting Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
NAVSTAR	Navigation System with Time and Ranging
NED	North-East-Down
ESC	Electronic Speed Controller
PWM	Pulse-Width Modulation
LiPo	Lithium-Polymer
RTOS	Real Time Operating System
FMU	Flight Management Unit
DSP	Digital Signal Processor
FPU	Floating Point Unit
MAVLink	Micro Air Vehicle Link
API	Application Programming Interface
LQR	Linear Quadratic Regulator
FL	Feedback Linearization
SISO	Single-Input Single-Output
MIMO	Multi-Input Multi-Output

CHAPTER 1

INTRODUCTION

1.1 Motivation of the Study

The last few decades have seen a growing trend towards the utilization of unmanned vehicles in various military and civil applications. Additionally, the tasks, which cannot be performed with required precision and efficiency by human capabilities or may possibly endanger human life, are expected to be mostly achieved by these vehicles in the near future.

This study essentially concentrates upon unmanned sea surface vehicles (USSV) which are designed to move in uninterrupted contact with water surface.

Unmanned sea surface vehicles have been assigned with missions exhibiting great diversity such as, environmental monitoring, detection and identification of the objects on sea surface, reconnaissance, mapping, supporting autonomous underwater vehicles [10]. Consequently, there is a growing body of literature that recognizes the significance of the complete autonomy of these vehicles which can be ensured by reliable, efficient and accurate design and implementation of navigation and autopilot systems [11].

A navigation algorithm offering a robust localization method is needed as the position and velocity information of the vehicle is to be determined with high precision by using erroneous measurements.

The autopilot algorithm basically renders the movement of a surface vehicle without continuous human intervention possible. Moreover, it is required to present complex and agile maneuvers guarantying the stability of the vessel. However, the design procedure of the autopilot is quite challenging regarding to the nonlinear and coupled dynamics of the system.

In this context, this dissertation is set out to investigate the navigation and autopilot algorithms for an USSV owing to the remarkable potential and challenge in this research area. Furthermore, the implementation of the algorithms is taken seriously since it will provide the basis on which a completely autonomous USSV can be constructed.

1.2 Literature Review

A considerable amount of literature has been published on the mathematical modeling of sea surface vehicles. Almost all components of the ship systems have been investigated within the literature [12]. These studies are summarized in brief by [13].

A marine craft experiences motion in 6 degrees of freedom (DOF) while maneuvering. Thus, the model illustrating the motion in 6 DOF is primarily constructed considering the Newton-Euler or Lagrange equations as given in (1.1). The following equations make use of SNAME notation [1].

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} = \sum_{k=1}^{n} \boldsymbol{F}_k \tag{1.1}$$

where $\boldsymbol{M}$ symbolizes the system inertia matrix, and $\dot{\boldsymbol{\nu}}$ is denoted for the generalized acceleration vector. The generalized velocity vector, $\boldsymbol{\nu}$, comprises of linear and angular velocities as given in (1.2) where $(u,\ v,\ w)$ are surge, sway and heave velocities, and $(p,\ q,\ r)$ are roll, pitch and yaw rates, respectively.

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{1.2}$$

In Equation (1.1), $\boldsymbol{F}_k$ depicts the force/moment vector.

$$\boldsymbol{F}_k = \begin{bmatrix} X_k & Y_k & Z_k & K_k & M_k & N_k \end{bmatrix}^T \tag{1.3}$$

The number of parameters utilized in the model represented by Equation (1.1), which is in the order of hundreds to be able to include the effects of nonlinear hydrodynamic factors, makes a model based controller design quite complicated [2]. Besides, identification process of these parameters with sufficient accuracy poses difficulties.

Accordingly, Fossen [14] proposed a model demonstrated in (1.4) that depends on the vectorial setting using the work of Craig [15] as motivation.

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

In the above equation, $\boldsymbol{M}$ represents the system inertia matrix, $\boldsymbol{C}$ is Coriolis matrix, τ and $\boldsymbol{q}$ symbolize torques and vector of joint angles, respectively. This model was improved by a number of researchers [16], [17], [18] and [19], and ended up with Equation (1.5).

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

where vectors of Euler angles, η, and velocities, $\boldsymbol{v}$, are defined as in (1.6).

$$\begin{aligned} \eta &= \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \\ \boldsymbol{v} &= \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \end{aligned} \tag{1.6}$$

The model given in Equation (1.5) is equivalent to the one introduced by Equation (1.1) in component form. On the other hand, the former representation simplifies designing control systems by providing symmetric, skew-symmetric and positive matrices. Additionally, construction of MIMO controllers is accomplished easily due to the model and matrix properties [13].

Throughout this dissertation, the term navigation will be used to refer to the process of detection and determination of position, orientation and velocity of a marine vessel with respect to a reference coordinate frame. As the diversity and complexity of the tasks assigned to sea surface vehicles increase, autonomous navigation systems with greater accuracy are required [20]. To that end, numerous types of navigation systems are exploited, such as inertial navigation, satellite navigation, terrestrial radio navigation and feature matching [3] while the majority of the unmanned sea surface vehicles depend on radio or spread-spectrum communications, global positioning systems and inertial navigation systems [21]. To be able to obtain precise and reliable navigation solutions, more than one of these systems are generally preferred to be integrated in a structure. By doing so, it is aimed to take advantage of multiple solutions at the same time. For instance, a Global Positioning System (GPS), an inertial measurement unit (IMU), an automatic identification system and a marine radar are utilized in [22] all at once. Meanwhile, using a multi sensor fusion algorithm also reduces the tendency of the complete system to be affected by possible sensor failures [23]. In literature, there are various examples of integration algorithms based on different state estimators, such as Kalman Filter, Extended Kalman Filter or particle filter [3].

Autopilots for sea surface vehicles have been constructed since early 20th century [13]. Minorsky was the first researcher to analyze the most commonly used technique, namely Proportional-Integral-Derivative (PID) controller in detail [24]. In this study, a single-input single output (SISO) system was realized which controls the yaw position by utilizing a gyrocompass. After Kalman Filter was introduced in [25], the theory for linear quadratic optimal controllers became available. Thereafter, linear quadratic and H_∞ controllers were extensively exploited for the autopilot of surface vehicles [18], [26] [27], [28].

There have been a number of several studies for investigating nonlinear controllers for ship autopilots in 21st century. Do (2004), et al. introduced a robust adaptive controller by using Lyapunov's direct method and backstepping method for underactuated ships [29]. In [30], a sliding mode fuzzy controller was proposed for the guidance and control of autonomous underwater vehicles suggesting the viability of the technique for surface vehicles. In another study, an output feedback linearization controller with a nonlinear observer were employed to control a nonlinear vessel model while some of the states are not observable [31]. The asymptotic stability of the overall system including the internal dynamics was proven.

1.3 Organization of the Thesis

In this study, navigation and autopilot algorithms are examined for an unmanned sea surface vehicle. The motivation of this work and several significant publications detected in the wake of a comprehensive review of the literature are presented in Chapter 1.

In Chapter 2, a mathematical model for an unmanned sea surface vehicle with two independent propellers, but with no rudder is constructed. To that end, essential mathematical tools are introduced at first. Throughout the derivation of the model, rigid body dynamics, added mass dynamics, hydrodynamic damping factors, restoring and air drag forces and thruster dynamics are taken into consideration. Only sources of actuation are taken to be two distinct propellers in the model. The motion of the vessel will basically be controlled by producing necessary thrusts via these propellers.

In Chapter 3, various navigation algorithms for sea surface vehicles are described to assure the accomplishment of system identification and autopilot applications. The solutions obtained from inertial navigation system, Global Navigation Satellite System and an attitude estimation system making use of a magnetometer are integrated by different types of architectures. The one with best performance is also employed to process the real measurements collected from the experimental setup.

In Chapter 4, the details of the experimental setup are discussed. The model boat named as Pacific Islander Tugboat, the driving system comprising of Electronic Speed Controllers and brushless DC motors, and the autopilot card, called Pixhawk, are familiarized. After that, the developed softwares running on the ground station and autopilot card are explained.

In Chapter 5, two autopilots utilizing LQR and FL controllers are considered depending on the improved mathematical model of the model boat. To be able to apply LQR control theory, the model is linearized around various linearization points, and the related controllable subspaces of the representations are explored. On the other hand, FL controller is designed by taking the original nonlinear model into account. Subsequently, reference tracking and disturbance rejection performances of the controllers are comparatively analyzed.

Chapter 6 concludes the dissertation by presenting a brief summary of the study, discussing the results and listing the recommendations for possible further research work.

CHAPTER 2

MATHEMATICAL MODELING

2.1 Introduction

In this chapter, the mathematical model of an Unmanned Sea Surface Vehicle (USSV) with two independent propellers, but with no rudder is developed. In fact, the characteristics of the above mentioned drive system are only included in Section 2.7. Therefore, the rest of the model is valid for a typical USSV.

The mathematical model is created by depending on the vectorial model proposed by Fossen [13] for marine crafts. The inertial components and the forces acting on the boat are included in the model. It essentially takes centripetal and Coriolis forces, gravity and buoyancy forces, damping forces, air drag and thruster forces into account. Any rudder related force is not involved. Disturbances resulted from the environmental effects, such as wind, waves, water current are also not contained in the model. These effects are added as external forces for the assessment of disturbance rejection performances of the controllers in Chapter 5.

The coordinate frames and transformations between different frames are described in the beginning of the chapter. Next, the rigid body dynamics and the above mentioned forces are explained in details. Afterwards, the implementation of the model in MATLAB environment and the results of the simulations are briefly discussed. The acquired mathematical model is aimed to be employed in the system identification process as given in [6]. Then, the model with enhanced parameters will be used throughout the design procedures of the autopilot algorithms as expressed in Chapter 5.

2.2 Kinematics

Kinematics describing the geometrical aspects of motion is analyzed in this section. (2.1) depicts the vectorial model of Fossen for surface vehicles. The model has 6 degrees of freedom (DOF).

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

$\boldsymbol{M}$ represents system inertia matrix, $\boldsymbol{C}$ stands for Coriolis matrix, τ and $\boldsymbol{q}$ illustrate torques and vector of joint angles, respectively.

Equation (2.1) is improved by [16], [17], [18] and [19], and put into the form given by (2.2).

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\dot{\boldsymbol{v}})\boldsymbol{v} + \boldsymbol{g}(\eta) + \boldsymbol{g}_0 = \tau + \tau_{wind} + \tau_{wave} \tag{2.2}$$

where

$$\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{2.3}$$

$$\boldsymbol{v} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{2.4}$$

The model is again reshaped by Equation (2.5) combining disturbances in a vector.

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

In (2.5), the rigid body and added mass dynamics are included on the left hand side. Besides, τ_d is water damping force, τ_g stands for gravitational and buoyancy forces, τ_t is the summation of the forces produced by two propellers, τ_a is air drag forces, and τ_{dist} is used for disturbances. The details of (2.5) will be given in the forthcoming sections after the essential mathematical tools are introduced.

2.2.1 Coordinate Frames and Transformations

A coordinate system, or frame is fixed to the body with the purpose of describing the relative position and orientation of the vessel [15]. Figure 2.1 illustrates the body fixed coordinate system and the motion in 6 degrees of freedom of the vehicle.

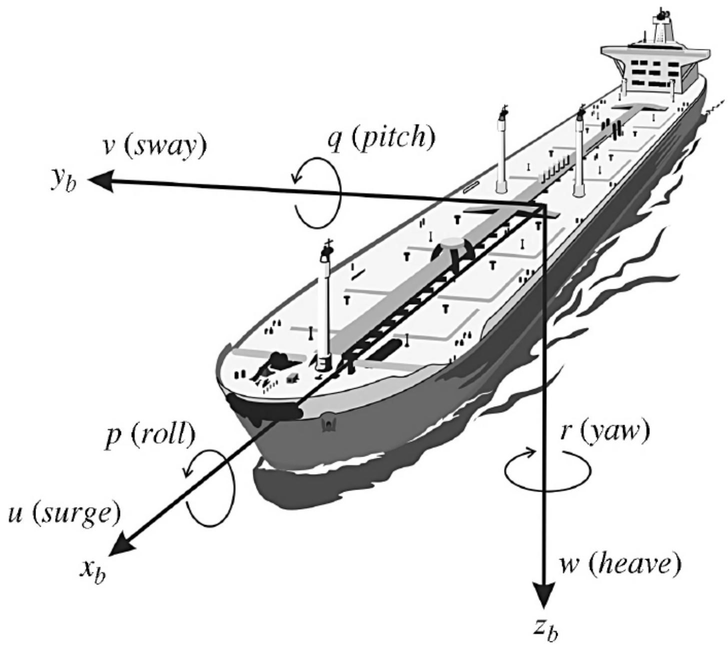


Figure 2.1: The 6 DOF velocities in the body fixed frame, [2].

The model relies on SNAME notation, the details of which given in the below table [1].

Table 2.1: The notation indicated by SNAME, [1].

DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	motion in the x direction (surge)	X	u	x
2	motion in the y direction (sway)	Y	v	y
3	motion 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

North-East-Down (NED) and the body fixed reference frames are exploited in this study. The motion of the vessel with respect to the Earth is explained in terms of NED frame symbolized by $\{n\} = \{x_n, y_n, z_n\}$ with the origin o_n. Furthermore,

$\{b\} = \{x_b, y_b, z_b\}$ with the origin o_b represents the body fixed frame. Figure 2.2 demonstrates both of the reference frames.

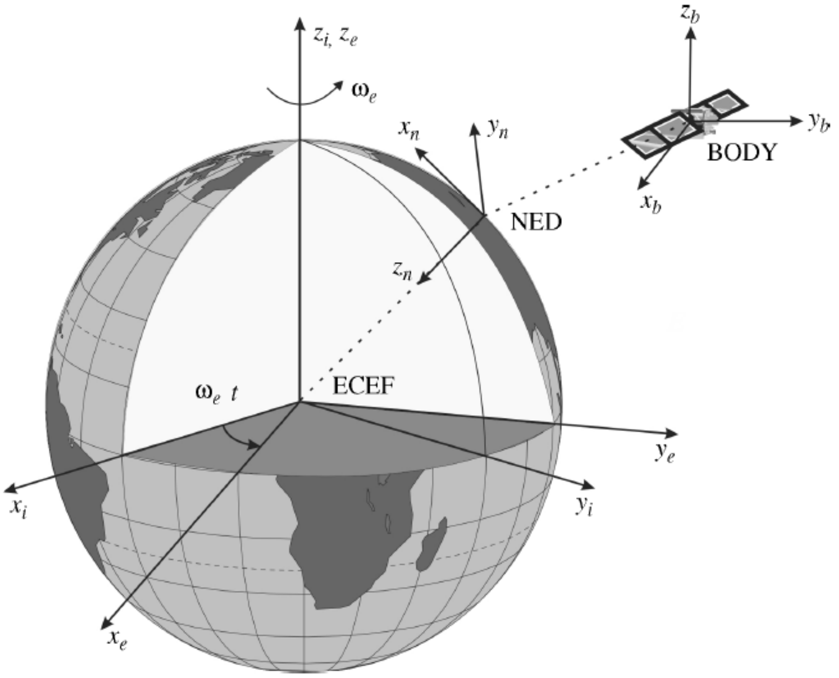


Figure 2.2: Representation of Earth and the body fixed frames, [2].

$\{n\}$ resolves the position and the orientation of the body, and the linear and angular velocities are represented in terms of the axes of $\{b\}$.

In this study, the vector, $\boldsymbol{y}^{\gamma}_{\alpha|\beta}$, corresponds to $\boldsymbol{y}$ property of α frame with respect to β frame resolved in γ frame. Consequently, the variables are indicated as follows:

$\boldsymbol{v}^b_{b|n}$: Linear velocities of o_b w.r.t. $\{n\}$ expressed in $\{b\}$,

$\boldsymbol{\omega}^b_{b|n}$: Angular velocities of $\{b\}$ w.r.t. $\{n\}$ expressed in $\{b\}$,

$\boldsymbol{f}^b_b$: Force with acting point o_b expressed in $\{b\}$,

$\boldsymbol{m}^b_b$: Moment about $\{b\}$ expressed in $\{b\}$,

Θ_{nb} : Euler angles between $\{n\}$ and $\{b\}$.

The state and input vectors are chosen as $\eta = \begin{bmatrix} \boldsymbol{p}^n_{b|n} \\ \Theta_{nb} \end{bmatrix}$, $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}^b_{b|n} \\ \boldsymbol{\omega}^b_{b|n} \end{bmatrix}$, $\tau = \begin{bmatrix} \boldsymbol{f}^b_b \\ \boldsymbol{m}^b_b \end{bmatrix}$.

2.2.1.1 Transformations

Resolving axes of the vectors are needed to be transformed throughout the derivation process. For instance, the linear velocities of the vehicle which are originally calculated in the body frame are to be transformed to $\{n\}$ frame to be able to keep track of the linear positions. For this purpose, the rotation matrix, R_b^n, is formed by concatenating the principle vectors of $\{b\}$ frame expressed in $\{n\}$ frame [15]. Equation (2.6) defines the rotation matrix in terms of the Euler angles where $s(\cdot)$ and $c(\cdot)$ are respectively used for $sin(\cdot)$ and $cosine(\cdot)$ operations to simplify the notation.

$$R_b^n(\Theta_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi c\phi + c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi c\phi + s\theta s\psi c\phi \\ -s\theta & c\theta c\phi & c\theta c\phi \end{bmatrix} \tag{2.6}$$

Linear Velocity Transformation:
The linear velocity vector, $\boldsymbol{v}_{b|n}^b$, can be transformed to $\{n\}$ as follows:

$$\boldsymbol{v}_{b|n}^n = \dot{\boldsymbol{p}}_{b|n}^n = R_b^n(\Theta_{nb})\boldsymbol{v}_{b|n}^b \tag{2.7}$$

Angular Velocity Transformation:
$T_\Theta(\Theta_{nb})$ gives the relation between angular rate vector of the body frame, $\omega_{b|n}^b$, and the rate of change of the Euler angles, $\dot{\Theta}_{nb} = \left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T$, where $t(\cdot)$ represents $tangent(\cdot)$ operation.

$$\dot{\Theta}_{nb} = \boldsymbol{T}_\Theta(\Theta_{nb})\,\omega_{b|n}^b \tag{2.8}$$

$$\boldsymbol{T}_\Theta(\Theta_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{2.9}$$

At the end, Equations (2.10) and (2.11) describe the kinematic equations of the marine craft in 6 DOF by using the transformations, and the vectors that are already defined.

$$\begin{aligned} \dot{\eta} &= J_{\Theta}(\eta)\boldsymbol{v} \qquad (2.10) \\ \begin{bmatrix} \dot{\boldsymbol{p}}^n_{b|n} \\ \dot{\Theta}_{nb} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{R}^n_b(\Theta_{nb}) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{T}_{\Theta}(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^b_{b|n} \\ \omega^b_{b|n} \end{bmatrix} \qquad (2.11) \end{aligned}$$

2.3 Rigid Body Dynamics

The equations for a marine craft relying on Newton's second law is formulated by Fossen [13] as given below.

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{v}} + \boldsymbol{C}_{RB}(\boldsymbol{v})\boldsymbol{v} = \tau \qquad (2.12)$$

where RB stands for rigid body. M_{RB} and C_{RB} are mass and Coriolis/centripetal matrices, respectively. Recall that the velocity vector, $\boldsymbol{v}$, comprises of $[u,v,w,p,q,r]^T$ represented in the body frame and the torques acting on the body, τ, is given as $[X,\ Y,\ Z,\ K,\ M,\ N]^T$.

Two different points, namely the origin of the body frame, **CO**, and the center of gravity of the body, **CG**, will be utilized in the following equations.

Newton's second law of motion is specified with regard to the conservation of linear momentum, $\vec{\boldsymbol{p}}_g$, and angular momentum, $\vec{\boldsymbol{h}}_g$, taking Euler's first and second axioms into consideration.

$$\begin{aligned} \frac{{}^i d}{dt}\vec{\boldsymbol{p}}_g &= \vec{\boldsymbol{f}}_g, \quad \vec{\boldsymbol{p}}_g = m\vec{\boldsymbol{v}}_{g|i} \qquad (2.13) \\ \frac{{}^i d}{dt}\vec{\boldsymbol{h}}_g &= \vec{\boldsymbol{m}}_g, \quad \vec{\boldsymbol{h}}_g = \boldsymbol{I}_g\vec{\omega}_{b|i} \qquad (2.14) \end{aligned}$$

Note that $\vec{\boldsymbol{f}}_g$ and $\vec{\boldsymbol{m}}_g$ are the moments acting on **CG**. Besides, the angular rate vector of $\{b\}$ with respect to $\{i\}$ is illustrated by $\vec{\omega}_{b|i}$. The inertia dyadic about **CG** is symbolized by $\boldsymbol{I}_g$.

The derivation of the model is performed under two assumptions.

- The marine craft is a rigid body. Thus, there is no need to reckon with the internal forces appearing between the individual elements of the body.
- $\{n\}$ is an inertial frame. In other words, the forces resulted from the Earth's rotational motion are omitted. Then, the following notations become equivalent.

$$\vec{\boldsymbol{v}}_{g|i} \approx \vec{\boldsymbol{v}}_{g|n} \tag{2.15}$$

$$\vec{\boldsymbol{\omega}}_{b|i} \approx \vec{\boldsymbol{\omega}}_{b|n} \tag{2.16}$$

The below formula declaring the rule for the time differentiation of a vector, $\vec{\boldsymbol{a}}$, will frequently be utilized.

$$\frac{{}^{i}d}{dt}\vec{\boldsymbol{a}} = \frac{{}^{b}d}{dt}\vec{\boldsymbol{a}} + \vec{\boldsymbol{\omega}}_{b|i} \times \vec{\boldsymbol{a}} \tag{2.17}$$

Translational Motion about CG:

$\vec{\boldsymbol{r}}_{g|i}$ represents the position vector from the origin of $\{i\}$ to **CG**. $\vec{\boldsymbol{r}}_{b|i}$ is denoted for the position vector which starts from the origin of $\{i\}$, and ends at the origin of the body frame. $\vec{\boldsymbol{r}}_g$ symbolizes the position vector from **CO** to **CG**.

Then, the following equation can be written.

$$\vec{\boldsymbol{r}}_{g|i} = \vec{\boldsymbol{r}}_{b|i} + \vec{\boldsymbol{r}}_g \tag{2.18}$$

Afterwards, the equation takes the form in below as $\{n\}$ is assumed to be inertial.

$$\vec{\boldsymbol{r}}_{g|n} = \vec{\boldsymbol{r}}_{b|n} + \vec{\boldsymbol{r}}_g \tag{2.19}$$

CG is stationary with respect to the body fixed frame for a rigid body.

$$\frac{{}^{b}d}{dt}\vec{\boldsymbol{r}}_g = \mathbf{0} \tag{2.20}$$

(2.21) is attained by combining (2.17) and (2.20).

$$\vec{\boldsymbol{v}}_{g|n} = \vec{\boldsymbol{v}}_{b|n} + \vec{\boldsymbol{\omega}}_{b|n} \times \vec{\boldsymbol{r}}_g \tag{2.21}$$

Subsequently, Euler's first axiom is put into use.

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

$$\boldsymbol{f}_g^b = m\left[\dot{\boldsymbol{v}}_{g|n}^b + S\left(\boldsymbol{\omega}_{b|n}^b\right)\boldsymbol{v}_{g|n}^b\right] \tag{2.22}$$

where $S(\cdot)$ is the operator obtaining a skew-symmetric matrix from the input vector as described in [2].

Rotational Motion about CG:

The rotational motion of the body is derived by utilizing a similar method with the previous one. Euler's second axiom is applied in the following equations.

$$\begin{aligned}
\vec{\boldsymbol{m}}_g &= \frac{{}^i d}{dt}\left(\boldsymbol{I}_g\vec{\boldsymbol{\omega}}_{b|n}\right) \\
&= \frac{{}^b d}{dt}\left(\boldsymbol{I}_g\vec{\boldsymbol{\omega}}_{b|n}\right) + \vec{\boldsymbol{\omega}}_{b|n} \times \left(\boldsymbol{I}_g\vec{\boldsymbol{\omega}}_{b|n}\right) \\
&= \boldsymbol{I}_g\dot{\vec{\boldsymbol{\omega}}}_{b|n} - \left(\boldsymbol{I}_g\vec{\boldsymbol{\omega}}_{b|n}\right) \times \vec{\boldsymbol{\omega}}_{b|n}
\end{aligned} \tag{2.23}$$

$$\boldsymbol{m}_g^b = \boldsymbol{I}_g\dot{\boldsymbol{\omega}}_{b|n}^b - S\left(\boldsymbol{I}_g\boldsymbol{\omega}_{b|n}^b\right)\boldsymbol{\omega}_{b|n}^b \tag{2.24}$$

where the inertia matrix, $\boldsymbol{I}_g$, is described in (2.25).

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

The equations describing the translational and rotational motion about **CO** are acquired by employing coordinate transformations and the parallel axis theorem.

$$\boldsymbol{f}_b^b = m\left[\dot{\boldsymbol{v}}_{b|n}^b + S\left(\dot{\omega}_{b|n}^b\right)\boldsymbol{r}_g^b + S\left(\omega_{b|n}^b\right)\boldsymbol{v}_{b|n}^b + S^2\left(\omega_{b|n}^b\right)\boldsymbol{r}_g^b\right] \tag{2.26}$$

$$\boldsymbol{m}_b^b = \boldsymbol{I}_b\dot{\omega}_{b|n}^b + S\left(\omega_{b|n}^b\right)\boldsymbol{I}_b\omega_{b|n}^b + mS\left(\boldsymbol{r}_g^b\right)\dot{\boldsymbol{v}}_{b|n}^b + mS\left(\boldsymbol{r}_g^b\right)S\left(\omega_{b|n}^b\right)\boldsymbol{v}_{b|n}^b \tag{2.27}$$

The derivation procedure is explicitly described in [2]. The details of the matrices used in Equations (2.26) and (2.27) are given below.

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

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

The rigid body mass matrix is uniquely given as in (2.30).

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

The Coriolis and centripetal matrix is illustrated in skew-symmetric form by (2.31).

$$
\boldsymbol{C}_{RB}(\boldsymbol{v}) = \begin{bmatrix}
0 & 0 & 0 & m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g r + v) \\
0 & 0 & 0 & -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\
0 & 0 & 0 & -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g) \\
-m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) & 0 & -I_{yz} q - I_{xz} p + I_z r & I_{yz} r + I_{xy} p - I_y q \\
-m(x_g q - w) & -m(z_g r + x_g p) & m(z_g p + u) & I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q - I_x p \\
m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) & -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_x p & 0
\end{bmatrix} \tag{2.31}
$$

When the origin of $\{b\}$ coincides with **CG**, and the axes of $\{b\}$ are organized in a fashion to render the inertia matrix diagonal, the matrices are simplified as shown in (2.32) and (2.33).

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

$$
\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & mu & 0 \\ 0 & mw & -mv & 0 & I_z r & -I_y q \\ -mw & 0 & mu & -I_z r & 0 & I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{2.33}
$$

2.4 Added Mass Dynamics

The motion of a body moving in liquid is characterized with the help of *added mass* terms. As the body transports some of the surrounding liquid by its motion, it is observed as if the vessel weights more than its original weight. To be able to express this incident in terms of the motion equations, added or virtual mass is to be introduced. When the vessel moves the fluid out of its headway, some of its kinetic energy is essentially transferred to the surrounding liquid. Added mass terms compensates this phenomenon in the equations. The matrices corresponding to the added mass are described in (2.34) and (2.35) by [13].

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

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

where $X_{\dot{u}}$ is defined to be the partial derivative of X with respect to $\dot{u}$.

The added mass terms are calculated under the assumption of the marine craft being in ellipsoidal shape. The work of [12] is pursued in the parameter derivation process.

An ellipsoid is represented by the following three parameters: semi-major, a; semi-minor, b; and semi- vertical, c, axes. Moreover, the equation, $\frac{x^2}{a} + \frac{y^2}{b} + \frac{z^2}{c} = 1$, holds for the parameters.

The added parameters are calculated as given below with the assumption of $b = c$.

$$
\begin{aligned}
X_{\dot{u}} &= -\frac{\alpha_0}{2-\alpha_0}\frac{4}{3}\pi\rho_w ab^2 && (2.36)\\
Y_{\dot{v}} &= Z_{\dot{w}} = -\frac{\beta_0}{2-\beta_0}\frac{4}{3}\pi\rho_w ab^2 && (2.37)\\
K_{\dot{p}} &= 0 && (2.38)\\
M_{\dot{q}} &= N_{\dot{r}} = -\frac{1}{5}\frac{\left(b^2-a^2\right)^2(\alpha_0-\beta_0)}{2\left(b^2-a^2\right)^2+\left(b^2+a^2\right)(\alpha_0-\beta_0)}\frac{4}{3}\pi\rho_w ab^2 && (2.39)
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha_0 &= \frac{2\left(1-e^2\right)}{e^3}\frac{1}{2}\log\left(\frac{1+e}{1-e}\right) - e && (2.40)\\
\beta_0 &= \frac{1}{e^2} - \frac{1-e^2}{2e^3}\log\left(\frac{1+e}{1-e}\right) && (2.41)
\end{aligned}
$$

(2.42) computes the eccentricity of the meridian elliptical section.

$$e^2 = 1 - \left(\frac{b}{a}\right)^2 \tag{2.42}$$

2.5 Hydrodynamic Damping

Hydrodynamic damping forces are basically caused by four different sources [18]. $\boldsymbol{D_P}(\boldsymbol{v})$ illustrates radiation induced potential damping caused by forced body oscillations. $\boldsymbol{D_S}(\boldsymbol{v})$ represents linear skin friction. $\boldsymbol{D_W}(\boldsymbol{v})$ corresponds to wave drift damping. $\boldsymbol{D_M}(\boldsymbol{v})$ is used for the damping due to vortex shedding.

The real, positive, non-symmetrical damping matrix, $\boldsymbol{D}(\boldsymbol{v})$, can be written in terms of these components.

$$\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D_P}(\boldsymbol{v}) + \boldsymbol{D_S}(\boldsymbol{v}) + \boldsymbol{D_W}(\boldsymbol{v}) + \boldsymbol{D_M}(\boldsymbol{v}) \tag{2.43}$$

Radiation Induced Potential Damping: While the body is made oscillating with the wave excitation frequency in default of incident waves, this type of damping emerges. On the other hand, radiation induced potential damping is usually ignored due to its being relatively smaller than the other forces.

Skin Friction:

Linear skin friction caused by laminar boundary layers and turbulent boundary layers essentially cause skin friction.

Wave Drift Damping:

Wave drift damping is associated with the interaction of the vessel with waves. The second order wave theory can be utilized to produce the equations of wave drift damping [12].

Damping due to Vortex Shedding:

The model for viscous damping force is given below.

$$f(U) = -\frac{1}{2}\rho C_D(R_n)A|U|U \tag{2.44}$$

where the speed of the boat is indicated by U, A is used for the projected cross sectional area under water, the drag coefficient is given by $C_D(R_n)$ and ρ is denoted for the water density.

The drag coefficient is determined by Reynolds number. (2.45) describes Reynolds number by using the characteristic length of the body, D, and the kinematic viscosity coefficient, v.

$$R_n = \frac{UD}{v} \tag{2.45}$$

Two separate matrices, standing for the linear and nonlinear parts, are usually used to represent the entire damping effect as shown in below.

$$\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D}_l + \boldsymbol{D}_n(\boldsymbol{v}) \tag{2.46}$$

where $\boldsymbol{D}_l$ and $\boldsymbol{D}_n(\boldsymbol{v})$ are the linear and nonlinear damping matrices, respectively.

The damping matrices are computed via "Expanded Ad Hoc damping model for high speed maneuvers" given in [18]. The linear damping matrix is described as follows.

$$\boldsymbol{D}_l = -\begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_u & 0 & 0 & 0 & Y_r \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_v & 0 & 0 & 0 & N_r \end{bmatrix} \tag{2.47}$$

The partial derivatives included in the above formula can be discovered with experiments. Furthermore, the following equation is utilized for expressing motion with low speed.

$$N_v = Y_r \tag{2.48}$$

Next, the nonlinear damping matrix is given in (2.49).

$$D_n = \begin{bmatrix} X_{|u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{|v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{|w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{|p|}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{|q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{|r|r}|r| \end{bmatrix} \tag{2.49}$$

(2.44) is exploited to compute the terms in (2.49).

Finally, the overall damping forces are evaluated with the below formula.

$$\tau_d = \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} \tag{2.50}$$

Throughout the derivation process of the damping matrices, the concerned body is assumed to be in symmetrical box shape. Additionally, R_n is taken to be greater than 10^4. Moreover, the ad hoc model is expanded with the damping component operating in the z direction of the body fixed frame.

2.6 Restoring Forces

τ_g included in (2.5) represents gravitational and buoyancy forces. As these forces fundamentally attempt to transfer the system to its equilibrium point within the range of stability, they are also called restoring forces. The parameters, **CG**, **CB**, GM_T, GM_L, are employed to be able to formulate τ_g. The center of gravity of the surface vehicle is **CG**. The center of gravity of the underwater volume of the boat is named as center of buoyancy, and illustrated by **CB**. The distance between **CG** and the transverse metacenter is known as transverse metacentric height, GM_T. The distance between **CG** and the longitudinal metacenter is known as longitudinal metacentric height, GM_L.

A cross sectional view of the vessel is illustrated in Figure 2.3. Notice that the body is rotated in the roll axis.

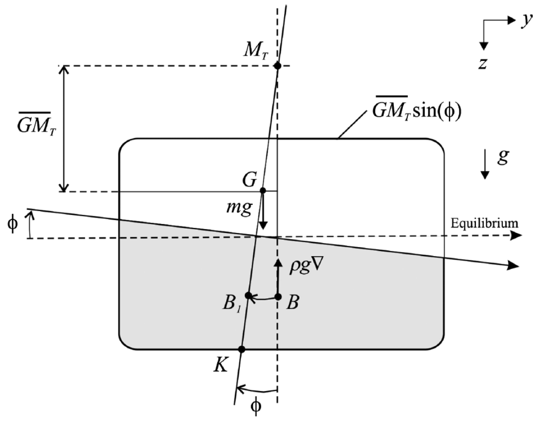


Figure 2.3: An illustration of traverse metacentric stability, [2].

The weight of the vessel is balanced with buoyancy forces when it is at rest.

$$mg = \rho_w g\nabla \tag{2.51}$$

where ρ_w is the density of the water.

The stationary state of the vessel is specified as $z = 0$. Hence, a net force is exerted on the body which is resulted from the variation of the buoyancy force when the position in the z-axis deviates from zero. The net force is formulated below.

$$Z = m\boldsymbol{g} - \rho_w \boldsymbol{g}(\nabla + \delta\nabla(z)) = -\rho_w \boldsymbol{g}\delta\nabla(z) \tag{2.52}$$

The force is rewritten with the assumption of box-shaped body, i.e., $A_{wp}(z) = A_{wp}(0)$.

$$Z \approx -\rho_w \boldsymbol{g} A_{wp} z = z_0 z \tag{2.53}$$

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

Subsequently, the force is resolved in $\{b\}$.

$$\delta \boldsymbol{f}_r^b = \boldsymbol{R}_b^n(\Theta_{nb})^{-1}\delta \boldsymbol{f}_r^n = \boldsymbol{R}_b^n(\Theta_{nb})^{-1} z_0 z = z_0 z \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix} \tag{2.54}$$

The moment arms in roll and pitch are defined as:

$$\boldsymbol{r}_r^b = \begin{bmatrix} -GM_T s\theta \\ GM_L s\phi \\ 0 \end{bmatrix} \tag{2.55}$$

The buoyancy force is transferred to the body fixed frame.

$$\boldsymbol{f}_r^b = R_b^n(\Theta_{nb})^{-1} \begin{bmatrix} 0 \\ 0 \\ -\rho_w g \nabla \end{bmatrix} = -\rho_w g \nabla \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix} \tag{2.56}$$

The restoring moment caused by $\delta \boldsymbol{f}_r^b$ can be neglected as it is quite smaller than the one resulted from $\boldsymbol{f}_r^b$ [13]. Thus, the moment is found as given in the following equation.

$$\boldsymbol{m}_r^b = \boldsymbol{r}_r^b \times \boldsymbol{f}_r^b = -\rho_w g \nabla \begin{bmatrix} GM_T s\phi c\theta c\phi \\ GM_L s\theta c\theta c\phi \\ (-GM_L c\theta + GM_L)\, s\phi s\theta \end{bmatrix} \tag{2.57}$$

In the end, the forces and moments related with the restoring effects are included in the below vector.

$$\tau_g = \begin{bmatrix} \delta \boldsymbol{f}_r^b \\ \boldsymbol{m}_r^b \end{bmatrix} = \begin{bmatrix} -z_0 z s\theta \\ z_0 z c\theta s\phi \\ z_0 z c\theta c\phi \\ -\rho_w \boldsymbol{g} \nabla GM_T s\phi c\theta c\phi \\ -\rho_w \boldsymbol{g} \nabla GM_L s\phi c\theta c\phi \\ -\rho_w \boldsymbol{g} \nabla (GM_L c\theta + GM_L)\, s\phi s\theta \end{bmatrix} \tag{2.58}$$

If small angle assumption is applied to the above equation, it is further simplified as depicted in (2.59).

$$\tau_g = \begin{bmatrix} 0 \\ 0 \\ -z_0 z \\ -\rho_w g \nabla GM_T \phi \\ -\rho_w g \nabla GM_L \theta \\ 0 \end{bmatrix} \tag{2.59}$$

Considering (2.59), the restoring force matrix is written in (2.60).

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & z_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\rho_w g \nabla GM_T & 0 & 0 \\ 0 & 0 & 0 & 0 & -\rho_w g \nabla GM_L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.60}$$

2.7 Thruster Dynamics

As previously mentioned, the model has two independent thrusters which are illustrated in Figure 2.4. In the model, the thrusters are assumed to be sources of force in x direction of the body frame, by omitting the rotational impact.

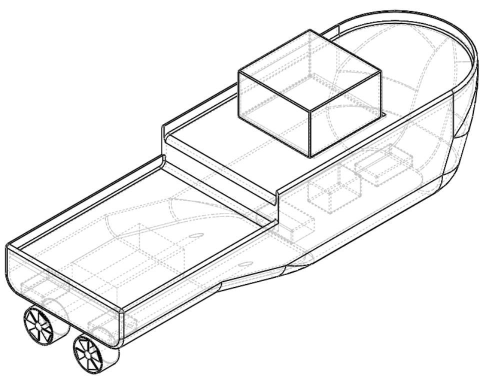


Figure 2.4: Two independent thrusters located at the stern of the boat.

$\boldsymbol{f}_R$ and $\boldsymbol{f}_L$ respectively illustrate the forces produced by right and left thrusters. The position vector from **CG** to the right thruster is denoted by $\boldsymbol{r}_{t_R}$, and $\boldsymbol{r}_{t_L}$ indicates the position vector from **CG** to the left thruster. In addition, the moments resulted by the thrusters are derived in the following equations.

$$\boldsymbol{m}_R = \boldsymbol{r}_{t_R} \times \boldsymbol{f}_R \tag{2.61}$$

$$\boldsymbol{m}_L = \boldsymbol{r}_{t_L} \times \boldsymbol{f}_L \tag{2.62}$$

As a result, the total force/moment vector of thrust is found by superposition.

$$\tau_t = \begin{bmatrix} \boldsymbol{f}_R \\ \boldsymbol{m}_R \end{bmatrix} + \begin{bmatrix} \boldsymbol{f}_L \\ \boldsymbol{m}_L \end{bmatrix} \tag{2.63}$$

2.8 Air Drag Forces

Forces correspondent to air drag are also taken into account in the model. The forces are associated with the relative velocity of the marine craft with respect to air. The relative velocity is defined in (2.64).

$$\boldsymbol{v}_r = \boldsymbol{v}_b - \boldsymbol{v}_a \tag{2.64}$$

where $\boldsymbol{v}_r$ is the relative velocity of the vehicle with regard to air, $\boldsymbol{v}_b$ and $\boldsymbol{v}_a$ are the velocities of the body and air related to inertial frame, respectively.

(2.65) computes the air drag forces.

$$\boldsymbol{F}_a = A_a P_a C_{d,a} \tag{2.65}$$

where $\boldsymbol{F}_a$ illustrates the air drag forces, A_a is the area of the vehicle surface that interacts with air, P_a symbolizes the air pressure, and the air drag coefficient is denoted by $C_{d,a}$.

The approximation for the air pressure is given below [13].

$$P_a \approx 2.56 v_r^2 \tag{2.66}$$

The corresponding air drag torques are found by the cross product given in (2.68). The moment arms are as given below under the assumption of symmetric shape of the vehicle and **CG** being at the midpoint.

$$\lambda = \begin{bmatrix} \frac{l}{2} & \frac{wi}{2} & \frac{h}{2} \end{bmatrix}^T \tag{2.67}$$

$$\boldsymbol{T}_a = \lambda \times \boldsymbol{F}_a \tag{2.68}$$

where l, wi and h are respectively the length, width and the height of the body. Torque components are neglected in the final equation considering the small dimensions of the model boat [32]. In the end, the air drag forces are expressed as in (2.69).

$$\tau_a = \begin{bmatrix} \boldsymbol{F}_a & 0 & 0 & 0 \end{bmatrix}^T \tag{2.69}$$

2.9 Implementation

Up to now, the mathematical model of a sea surface vehicle represented by Equation (2.5) is developed in this chapter. To be able to construct a simulator running efficiently on digital environment, the continuous model is to be discretized. For this purpose, Euler's backward and forward integration methods are employed at the same time as depicted in (2.70) and (2.71) [2].

$$\begin{aligned} \boldsymbol{\nu}(k+1) &= \boldsymbol{\nu}(k) + hM^{-1}\left(\boldsymbol{\tau} - \boldsymbol{C}\left(\boldsymbol{\nu}(k)\right)\boldsymbol{\nu}(k)\right) & (2.70) \\ \eta(k+1) &= \eta(k) + \boldsymbol{J}_{\Theta}\left(\eta(k)\right)\boldsymbol{\nu}(k) & (2.71) \end{aligned}$$

where h symbolizes the sampling interval, and it is determined to be 0.01 *sec*. (2.70) makes use of the forward integration technique, while (2.71) utilizes the backward integration method.

MATLAB environment is exploited for the discrete time realization of the model. Notice that the original continuous model is structured in a fragmented fashion; hence, each subpart is realized with an independent function to render the discrete time model debug/ user friendly. These functions are recursively called by a main function, and the outputs are combined as given in (2.70). Furthermore, this method also provides the advantage of observing the contribution of each subsystem separately.

2.10 Simulation Results

After the construction of the simulator pretending the motion characteristics of a surface vehicle is accomplished, it is excessively tested to determine whether the outputs of the program are reasonable in physical sense. For example, all of the initial velocities are taken as zero, and no force is exerted on the body by thrusters, then the vehicle is verified to stay stationary at its initial Cartesian position. In this section, some of the results obtained by various tests with different initial conditions and thrust inputs will be presented.

2.10.1 $Test_1$: Zero Thrust Inputs, NonZero Initial State

In the tests, thrusters are inactive, and does not produce any force. Besides, the initial values of Euler angles, linear and angular velocities are selected to be nonzero. Figures 2.5, 2.6, 2.7 and 2.8 demonstrate this case. The velocities and Euler angles converge to zero with time. Thus, the results seem to be consistent with the expectations. Nevertheless, the body is found to be extremely over-damped in rotational degrees of freedom as given in Figures 2.6 and 2.8. A possible explanation for this might be that the damping parameters for rotational motion are calculated greater than the real values. On the other hand, this does not pose a problem since the parameters will be estimated by the virtue of the system identification algorithm.

2.10.2 $Test_2$: Nonzero, Equal Thrust Inputs, Zero Initial State

In this case, initial states for all velocities and Euler angles are determined to be zero. Furthermore, both of the thrusters exerts 2 N in the x direction of the body throughout the motion. Figures 2.9, 2.10, 2.11 and 2.12 represent the findings. These results are also in line with the predictions. The roll and yaw angles do not deviate from zero since the magnitudes of thrusts are equal. On the contrary, the pitch angle exhibits an oscillation, and converge to a nonzero value as the vessel gathers speed along the x direction of the body. Moreover, the acceleration of the body in surge axis converge to zero due to the fact that damping forces are proportional to the velocity of the vehicle.

2.10.3 $Test_3$: Nonzero, Unequal Thrust Inputs, Zero Initial State

Initial states for all velocities and Euler angles are again taken as zero in this case. However, the magnitudes of applied thrusts are not equal as given in Equation (2.72). The results are depicted by Figures 2.13, 2.14, 2.15 and 2.16.

$$\begin{aligned} \tau_t^L &= [4\ \ 0\ \ 0]\ \ N \\ \tau_t^R &= [2\ \ 0\ \ 0]\ \ N \end{aligned} \tag{2.72}$$

This test is also successful, as it is able to reflect the motion in yaw axis. The yaw angle continuously increases owing to the inequality of the applied forces.

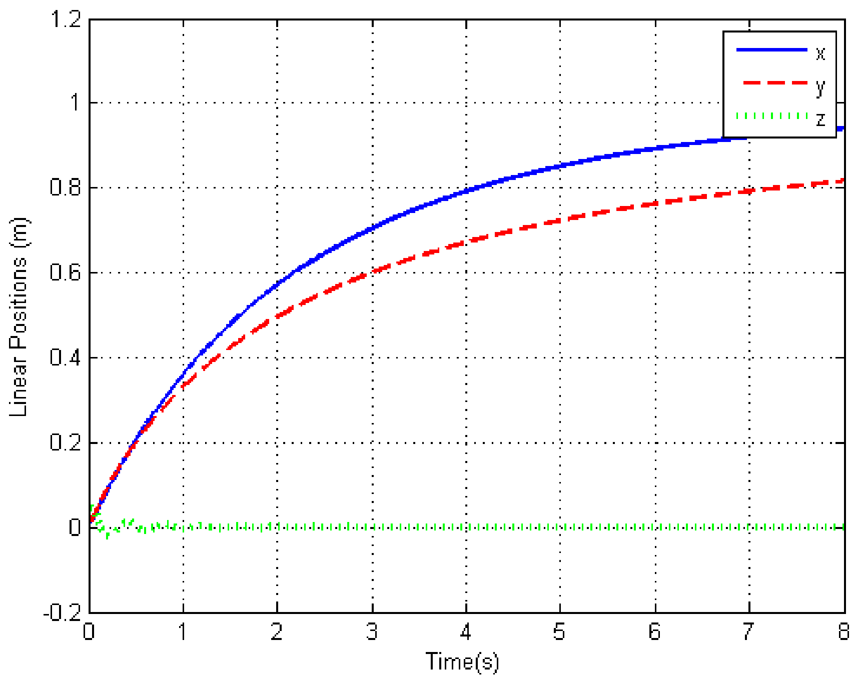


Figure 2.5: Linear positions resolved in $\{n\}$ frame during $Test_1$.

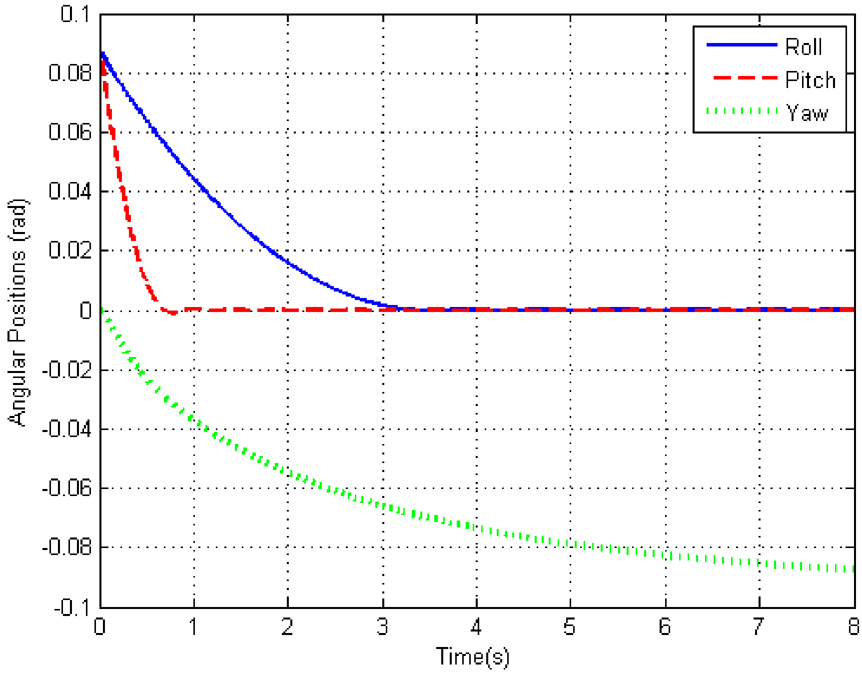


Figure 2.6: Euler angles during $Test_1$.

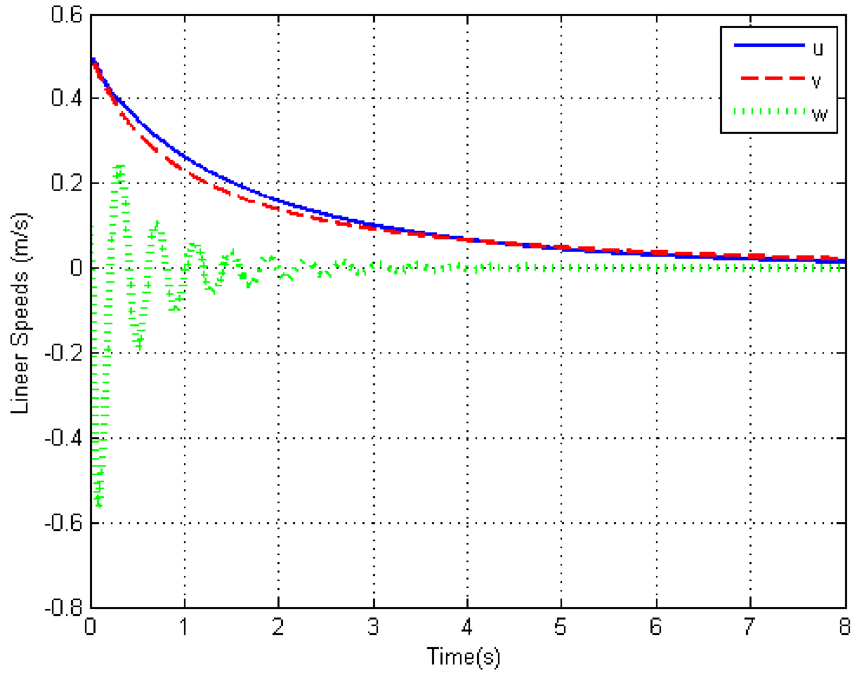


Figure 2.7: Linear velocities resolved in the body frame during $Test_1$.

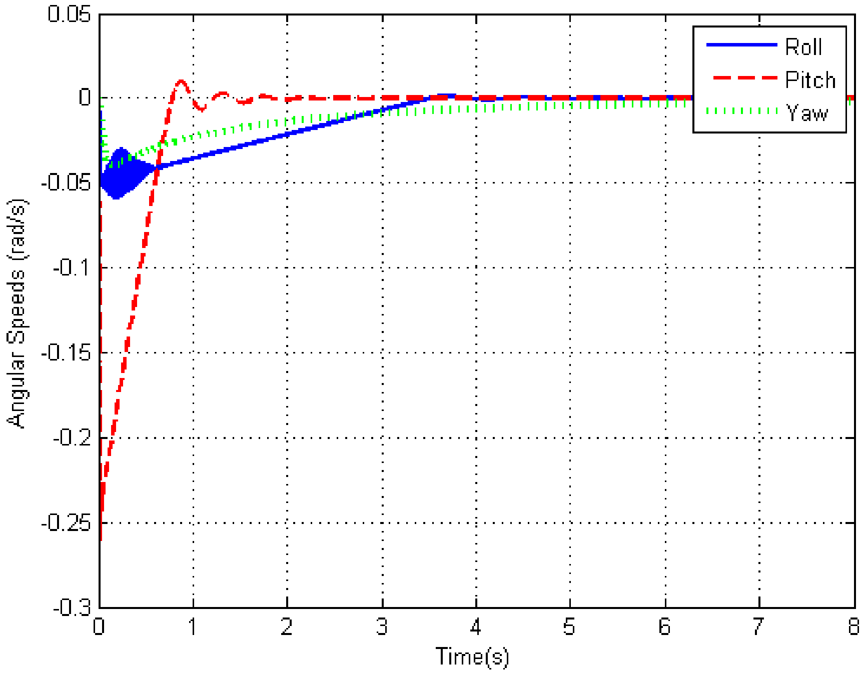


Figure 2.8: Angular rates resolved in the body frame during $Test_1$.

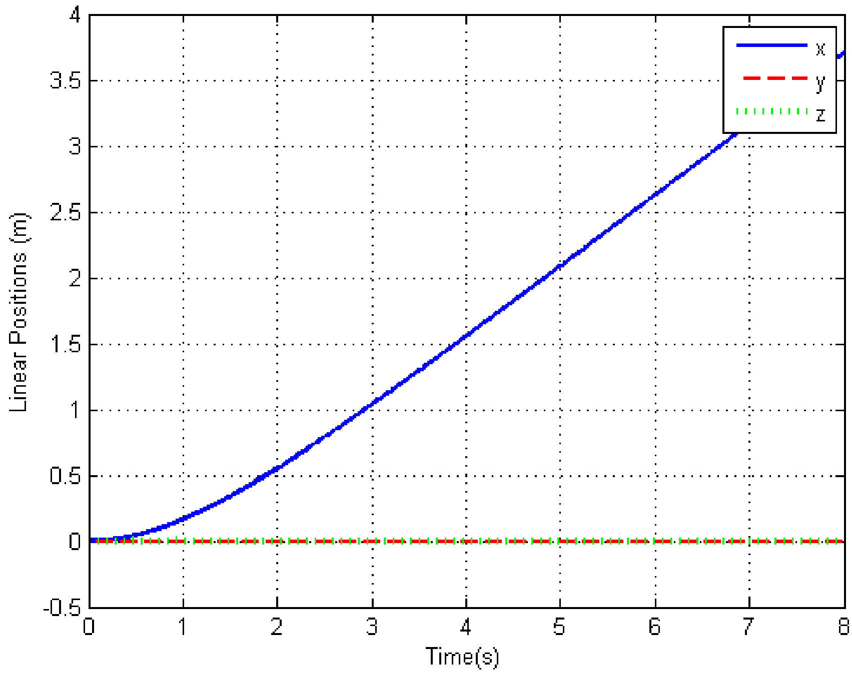


Figure 2.9: Linear positions resolved in $\{n\}$ frame during $Test_2$.

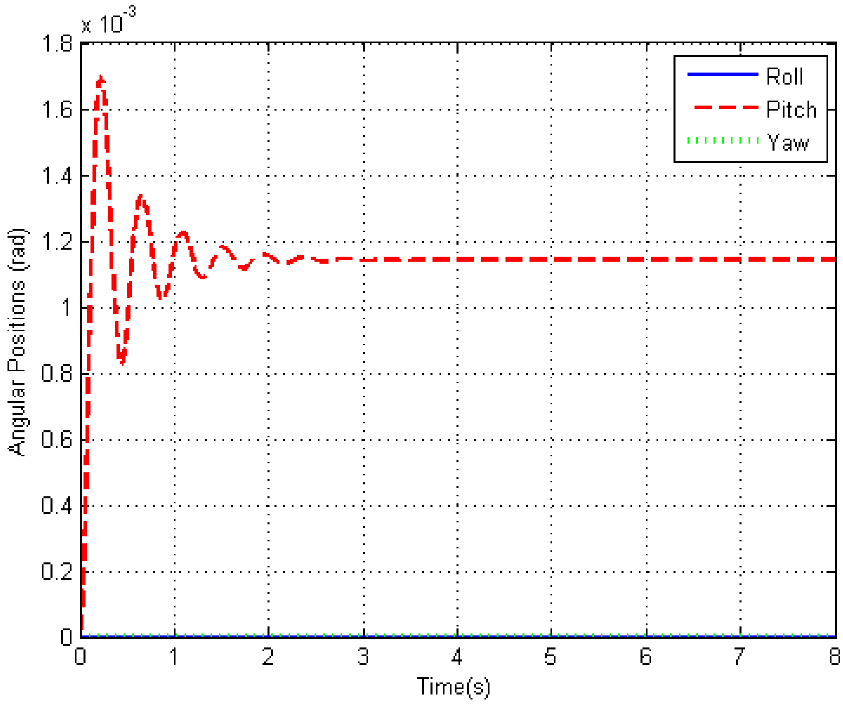


Figure 2.10: Euler angles during $Test_2$.

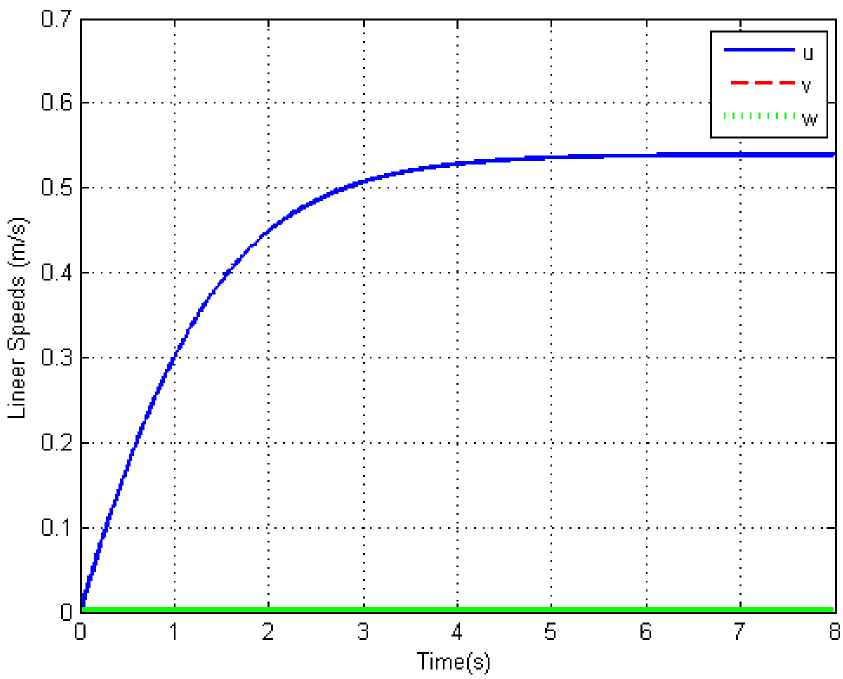


Figure 2.11: Linear velocities resolved in the body frame during $Test_2$.

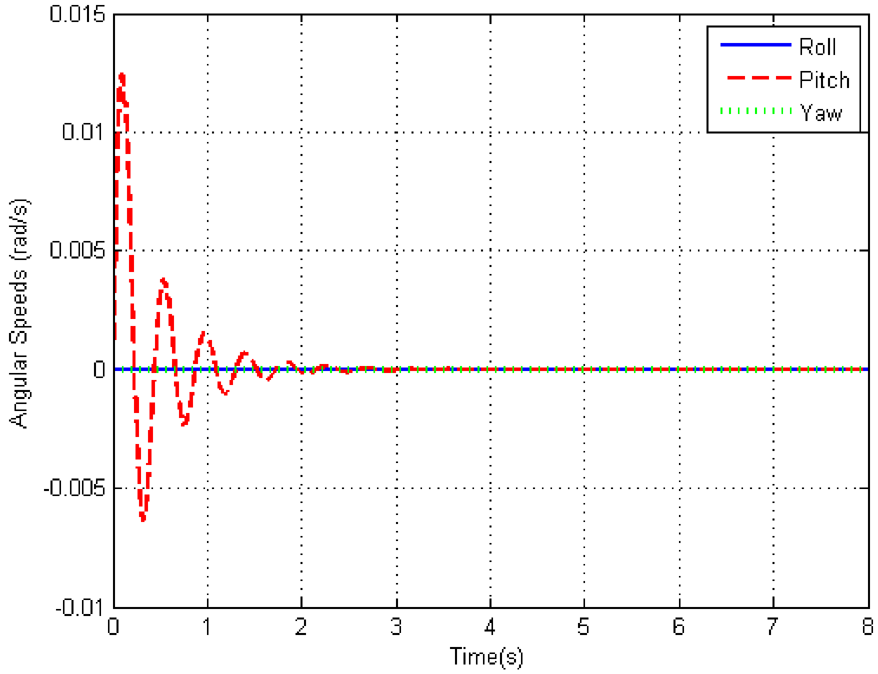


Figure 2.12: Angular rates resolved in the body frame during $Test_2$.

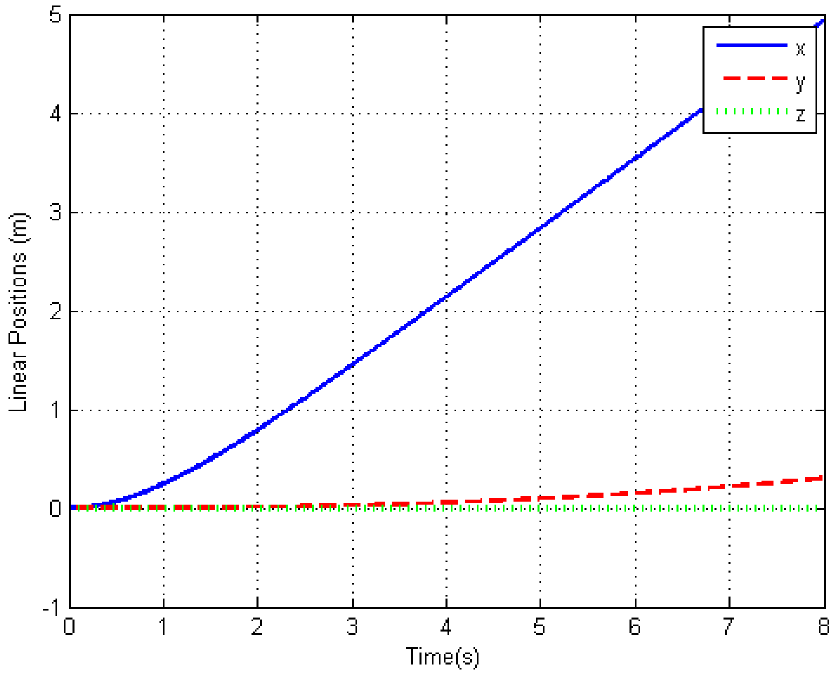


Figure 2.13: Linear positions resolved in $\{n\}$ frame during $Test_3$.

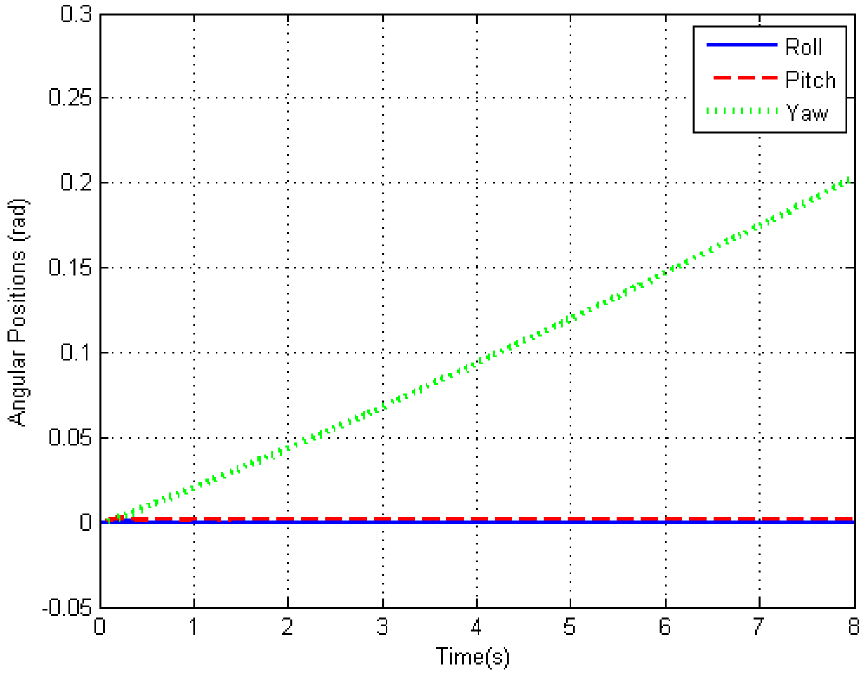


Figure 2.14: Euler angles during $Test_3$.

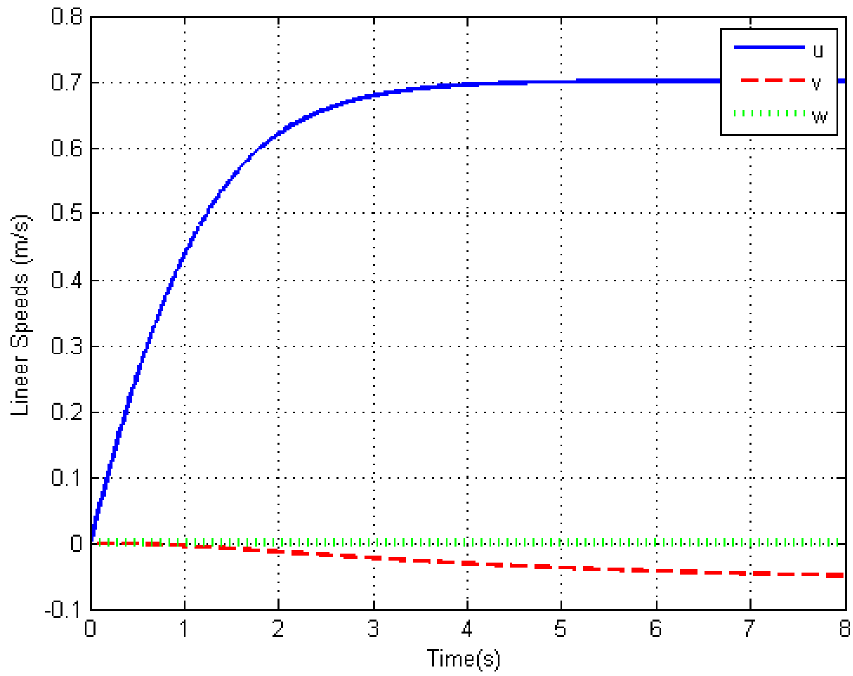


Figure 2.15: Linear velocities resolved in the body frame during $Test_3$.

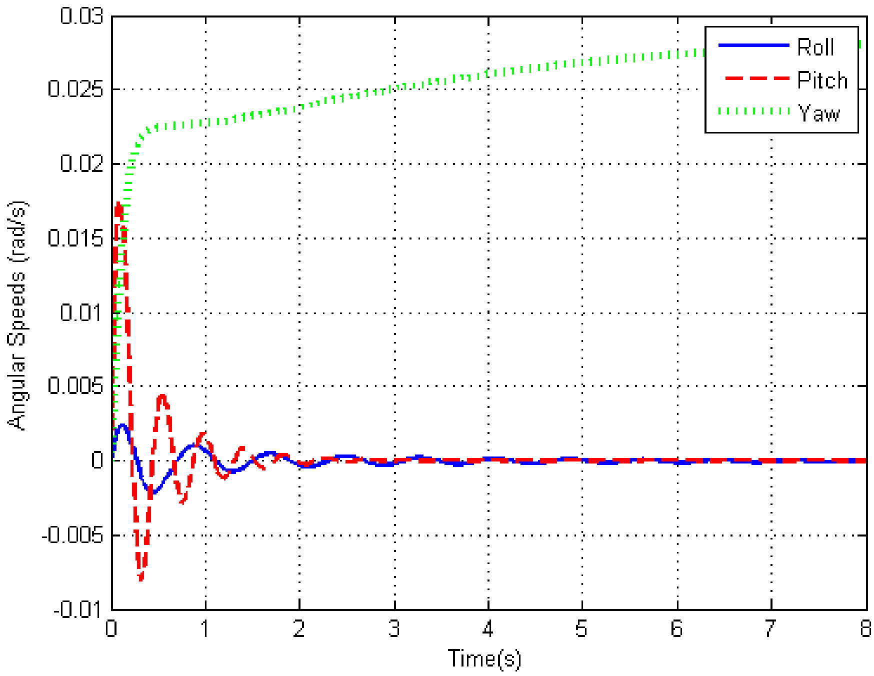


Figure 2.16: Angular rates resolved in the body frame during $Test_3$.

CHAPTER 3

NAVIGATION

3.1 Introduction

In the following chapter, various autopilot algorithms will be developed for an USSV. By doing so, it is inherently assumed that the true values of the state variables are already acquired. However, in practice it is unlikely to specify the exact value of the entire state vector due to the deficiency of measurements or the erroneous character of information sources. As a result, to be able to obtain the best possible correct value of the state vector and realize the above mentioned hypothesis, an algorithm, named *navigation*, is to be built.

It is necessary here to clarify implicitly what is meant by navigation. In this thesis, navigation is used to define the process of detection and determination of the position, orientation and velocity of an object with respect to an established reference point or frame. Position fixing and dead reckoning are two main methods that are utilized for this purpose as stated in [3].

Position fixing can be described as the technique that detects the location of a body in regard to a reference with the help of miscellaneous types of visual, radar, celestial and electronic devices. The distance and/or relative direction between the reference objects and the body may be benefited to generate position fixes. Global Positioning System (GPS) is a well-known example of a position fixing system which makes use of the measurements of distance between the GPS antenna and variable number of satellites. Dead reckoning is the process of calculating the recent position of the moving body by integrating its velocity measurement and adding it to the previous position. In this technique, a transformation with regard to the relative orientation is required since the velocities or position changes of the body are measured in the body frame. Consequently, an external source of orientation information is to be employed. The dead reckoning solution is prone to diverge from the true values of positions considering that the errors of measurements are being integrated in time. Furthermore, it entails an initialization for the locations. On the other hand, position fixing does not exhibit an accumulating fault behavior. However, the output of position fixing is not continuous and may even become unavailable due to environmental changes whereas the one of dead reckoning mostly appears to be uninterrupted. As a result, *an integrated navigation* may be applied to produce the navigation solution by combining those methods with complementary characteristics. The integrated algorithm may take advantage of both techniques at the same time.

3.2 Inertial Navigation

An inertial navigation system (INS) is a type of dead reckoning navigation system which consists of a processor and inertial sensors, namely accelerometers and gyroscopes [33]. The electronic device consisting of three mutually orthogonal and aligned accelerometer and gyroscope sets is called inertial measurement unit (IMU). The processor outputs the position, velocity and attitude of the object in three dimensional space by integrating the angular rates and specific forces measured by gyroscopes and accelerometers, respectively. The inertial processor is started from an initial point and utilizes a gravity model to convert accelerometer readings into acceleration information. Afterwards, it basically integrates the accelerations, velocities and angular rates to produce continuous navigation solution.

The errors of INS tend to increase with time as it consistently adds the errors of inertial sensors in an iterative fashion. Although INS in the market shows a great variety in accuracy and price, the drifts of error characteristics are similar. Hence, INS is usually preferred to be used together with external aids.

3.2.1 Inertial Sensors

Inertial sensors include accelerometers and gyroscopes. An accelerometer measures the physical phenomenon called specific force and a gyroscope measures angular rate. An IMU generally consists of three accelerometers and three gyroscopes which are placed in a mutually perpendicular manner. In this study, the IMU embedded to the autopilot card, Pixhawk, employs strapdown architecture since the inertial sensors are fixed with respect to the body coordinate frame. The working principles of the inertial sensors which show great diversity are not included in this thesis. However, error characteristics of these sensors are given since they are utilized in modeling phase of the integration algorithms.

3.2.1.1 Error Characteristics

The error sources are categorized into four main groups that are biases, scale factor errors, cross coupling errors and random noises for both of the sensors.

Bias is the constant error that is independent from measured quantity. Bias level exhibits quite low variation in run time although it may drastically change after the power of the sensor is cut. Biases of the accelerometer and gyroscope are respectively denoted by $\mathbf{b}_a$ and $\mathbf{b}_g$.

Scale factor errors are directly proportional to the magnitude of measurement. $\mathbf{s}_a = (s_{a,x},\ s_{a,y},\ s_{a,z})$ and $\mathbf{s}_g = (s_{g,x},\ s_{g,y},\ s_{g,z})$ correspond to the scale factor errors of accelerometers and gyroscopes in three axes.

The misalignments of the axes of inertial sensors result in cross coupling errors which are illustrated by $\mathbf{m}_{a,\alpha\beta}$ and $\mathbf{m}_{g,\alpha\beta}$ where the α and β axes are subject to misalignment.

The cross coupling and misalignment errors are combined with the following notations.

$$\mathbf{M}_a = \begin{bmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{bmatrix}, \; \mathbf{M}_g = \begin{bmatrix} s_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & s_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & s_{g,z} \end{bmatrix} \tag{3.1}$$

Lastly, the random noise is inevitably present in the measurements of inertial sensors resulted from the electrical and mechanical characteristics. $\mathbf{w}_a = (w_{a,x},\ w_{a,y},\ w_{a,z})$ and $\mathbf{w}_g = (w_{g,x},\ w_{g,y},\ w_{g,z})$ symbolize the random noises of accelerometer and gyroscope, respectively.

By taking these error sources into account accelerometer and gyro readings can be formulated as given in (3.2) and (3.3) [3].

$$\tilde{\mathbf{f}}^i_{b|i} = \mathbf{b}_a + (\mathbf{I}_3 + \mathbf{M}_a)\,\mathbf{f}^i_{b|i} + \mathbf{w}_a \tag{3.2}$$

$$\tilde{\boldsymbol{\omega}}^i_{b|i} = \mathbf{b}_g + (\mathbf{I}_3 + \mathbf{M}_g)\,\boldsymbol{\omega}^i_{b|i} + \mathbf{G}_g\mathbf{f}^i_{b|i} + \mathbf{w}_g \tag{3.3}$$

Where $\tilde{\mathbf{f}}^i_{b|i}$ and $\tilde{\boldsymbol{\omega}}^i_{b|i}$ are the outputs of the sensors and $\mathbf{f}^i_{b|i}$ and $\boldsymbol{\omega}^i_{b|i}$ are the corresponding true quantities of specific force and angular rate; $\mathbf{G}_g$ is the matrix relating the specific force and angular rate measurements; $\mathbf{I}_3$ is 3 by 3 identity matrix.

3.2.2 Mathematics of Inertial Navigation

The operations which are necessary to produce the three-dimensional inertial navigation solution from the measurements of an IMU is introduced in this subsection. As mentioned earlier, an IMU constituting of strapdown accelerometers and gyroscopes, outputs the specific forces, $\mathbf{f}^b_{b|i}$, and angular rates, $\boldsymbol{\omega}^b_{b|i}$, of the body with respect to inertial frame resolved in the body frame. Then, the outputs of the IMU are processed in the navigation processor considering the equations, details of which will be given below.

Attitude update, transformation of the specific force, velocity update and position update are four essential phases of the equations. The derivation of accelerations from specific forces with the gravitational model is included in velocity update stage. A brief overview of the basic steps is as follows:

- The process is initialized with external position, orientation, velocity information,
- Angular rates are integrated to obtain new orientation,
- Specific forces are transformed into the frame in which velocity and positions are resolved,
- Accelerations calculated by specific forces and gravitational model are integrated to obtain velocity,
- Velocities are integrated to obtain position.

The items after initialization are performed in an iterative way to generate new values from the previous solution and new measurement.

3.2.2.1 Selection of the Coordinate Frames

The inertial navigation equations, or INS mechanization equations, obviously differ in accordance with the coordinate frame in which the navigation solution is sought for. In this study, an Earth fixed tanged plane is taken into account since the sea surface vehicles operating in a local area are investigated. Therefore, the NED oriented, surface fixed frame is selected as both reference and resolving frame. This so called local navigation frame or flat Earth frame is denoted with $\{n\}$ and depicted in 3.1. It is attached to the initial position of the craft. The z-axis of the frame is directed toward the center of the Earth, and it is equivalent to the gravity vector while ignoring the local irregularities. The y-axis points east. The x-axis is attained by the projection of the line from the origin of the frame to the north pole to the y-z plane.

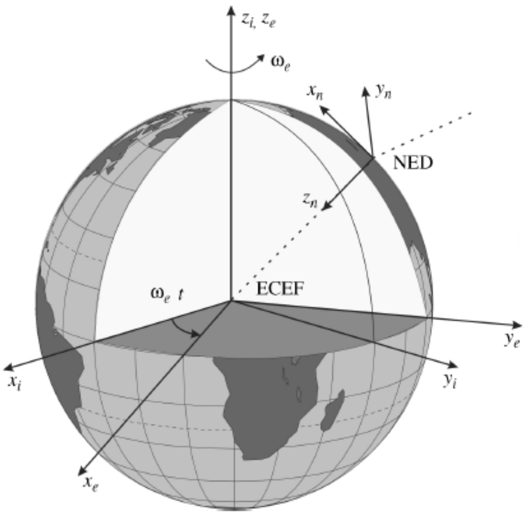


Figure 3.1: Illustration of local navigation frame $\{n\}$, [2].

Moreover, as [2] suggests, $\{n\}$ can be assumed to be inertial so that Newton's laws of motion can still be applicable. In other words, the rotation of Earth is neglected, which is a reasonable supposition owing to the fact that the marine vessels move in relatively slow speeds and the forces exerted on the body due to the Earth's rotation is rather small compared to the hydrodynamical forces. In conclusion, the equations in the rest of this section are derived to express the motion of a boat with respect to $\{n\}$ frame resolved in $\{n\}$ frame.

3.2.2.2 Kinematics

A brief summary of kinematic equations will be given in this section to provide a basis for the derivation of inertial navigation equations. To be able to illustrate the relative motion of the body with respect to some reference, coordinate frames and transformations between frames are indispensable for navigation applications. However, a detailed explanation about transformations will not be give here since it is already included in Mathematical Modeling chapter. $\mathbf{C}_{\beta}^{\gamma}$ represents the transformation matrix from β-frame to γ-frame. $\mathbf{x}_{\alpha|\beta}^{\gamma}$ refers to some property of α-frame regarding to β-frame expressed in terms of the axis of γ-frame.

The vector standing for Euler rotation between α and β frames is as follows:

$$\boldsymbol{\Theta}_{\alpha|\beta} = \begin{bmatrix} \phi_{\alpha|\beta} \\ \theta_{\alpha|\beta} \\ \psi_{\alpha|\beta} \end{bmatrix} \tag{3.4}$$

ZYX convention is utilized in (3.4) which is consistent with the previous chapter.

$\boldsymbol{\omega}^{\gamma}_{\alpha|\beta}$, denoting the rate of rotation of α-frame's axes regarding the β-frame expressed in terms of the axes of γ-frame is called *angular rate vector*. As mentioned before, angular rate vector consists of angular speeds in x, y and z axes of the resolving frame which are shown in (3.5).

$$\boldsymbol{\omega}^{\gamma}_{\alpha|\beta} = \begin{bmatrix} p_{\alpha|\beta} \\ q_{\alpha|\beta} \\ r_{\alpha|\beta} \end{bmatrix} \tag{3.5}$$

The elements of $\omega^{\gamma}_{\alpha|\beta}$ are put into skew symmetric form and symbolized as:

$$\boldsymbol{\Omega}^{\gamma}_{\alpha|\beta} = \begin{bmatrix} 0 & -r_{\alpha|\beta} & q_{\alpha|\beta} \\ r_{\alpha|\beta} & 0 & -p_{\alpha|\beta} \\ -q_{\alpha|\beta} & p_{\alpha|\beta} & 0 \end{bmatrix} \tag{3.6}$$

The time derivation of the transformation matrix is found as in (3.7) under the assumption of small angle [3].

$$\dot{\mathbf{C}}^{\alpha}_{\beta} = -\mathbf{C}^{\alpha}_{\beta}\boldsymbol{\Omega}^{\beta}_{\alpha|\beta} = -\boldsymbol{\Omega}^{\alpha}_{\alpha|\beta}\mathbf{C}^{\alpha}_{\beta} = \mathbf{C}^{\alpha}_{\beta}\boldsymbol{\Omega}^{\beta}_{\beta|\alpha} = \boldsymbol{\Omega}^{\alpha}_{\beta|\alpha}\mathbf{C}^{\alpha}_{\beta} \tag{3.7}$$

The *Cartesian position*, which is also known as Euclidean position, of the origin of α-frame with regard to the origin of the β-frame resolved in γ-frame is denoted as $\mathbf{r}^{\gamma}_{\alpha|\beta} = (x^{\gamma}_{\alpha|\beta}, y^{\gamma}_{\alpha|\beta}, z^{\gamma}_{\alpha|\beta})$. The position vector can be resolved in any frame with the help of related transformation matrix as follows:

$$\mathbf{r}^{\delta}_{\alpha|\beta} = \mathbf{C}^{\delta}_{\gamma}\mathbf{r}^{\gamma}_{\alpha|\beta} \tag{3.8}$$

The time derivative of $\mathbf{r}^{\beta}_{\alpha|\beta}$ calculated in β-frame is defined as the *velocity* of the object. It should be noted that (3.9) implies that a velocity is registered if the origin of α-frame changes location with respect to the origin of β-frame or the orientation of β-frame changes with respect to the origin of α-frame. The resolving axes of the velocity vector can also be altered as shown in (3.10).

$$\begin{aligned}\mathbf{v}^{\beta}_{\alpha|\beta} &\triangleq \dot{\mathbf{r}}^{\beta}_{\alpha|\beta} &(3.9)\\ \mathbf{v}^{\gamma}_{\alpha|\beta} &= \mathbf{C}^{\gamma}_{\beta}\mathbf{v}^{\beta}_{\alpha|\beta} &(3.10)\end{aligned}$$

The time derivative of the position vector resolved in a frame that is different from reference frame is found in (3.11). As it was mentioned previously, the vector illustrated by $\mathbf{v}^{\gamma}_{\alpha|\beta}$ is just a vector obtained by the transformation of $\mathbf{v}^{\beta}_{\alpha|\beta}$ into γ-frame. It is not corresponded to the time derivative of $\mathbf{r}^{\gamma}_{\alpha|\beta}$.

$$\begin{aligned}\dot{\mathbf{r}}^{\gamma}_{\alpha|\beta} &= \dot{\mathbf{C}}^{\gamma}_{\beta}\mathbf{r}^{\beta}_{\alpha|\beta} + \mathbf{C}^{\gamma}_{\beta}\dot{\mathbf{r}}^{\beta}_{\alpha|\beta}\\ &= \dot{\mathbf{C}}^{\gamma}_{\beta}\mathbf{r}^{\beta}_{\alpha|\beta} + \mathbf{v}^{\gamma}_{\alpha|\beta} &(3.11)\end{aligned}$$

The second time derivative of $\mathbf{r}^{\beta}_{\alpha|\beta}$ taken in β-frame is defined as the *acceleration* of the object. Furthermore, (3.13) shows the conversion of resolving axes of the acceleration.

$$\begin{aligned}\mathbf{a}^{\beta}_{\alpha|\beta} &\triangleq \ddot{\mathbf{r}}^{\beta}_{\alpha|\beta} &(3.12)\\ \mathbf{a}^{\gamma}_{\alpha|\beta} &= \mathbf{C}^{\gamma}_{\beta}\mathbf{a}^{\beta}_{\alpha|\beta} &(3.13)\end{aligned}$$

Note that $\mathbf{a}^{\gamma}_{\alpha|\beta}$ does not correspond to the time derivative $\mathbf{v}^{\gamma}_{\alpha|\beta}$ or the second time derivative of $\mathbf{r}^{\gamma}_{\alpha|\beta}$.

$$\dot{\mathbf{v}}^{\gamma}_{\alpha|\beta} = \frac{d\left(\mathbf{C}^{\gamma}_{\beta}\dot{\mathbf{r}}^{\beta}_{\alpha|\beta}\right)}{dt} = \dot{\mathbf{C}}^{\gamma}_{\beta}\dot{\mathbf{r}}^{\beta}_{\alpha|\beta} + \mathbf{a}^{\gamma}_{\alpha|\beta} \tag{3.14}$$

$$\begin{aligned}\ddot{\mathbf{r}}^{\gamma}_{\alpha|\beta} &= \ddot{\mathbf{C}}^{\gamma}_{\beta}\mathbf{r}^{\beta}_{\alpha|\beta} + \dot{\mathbf{C}}^{\gamma}_{\beta}\dot{\mathbf{r}}^{\beta}_{\alpha|\beta} + \dot{\mathbf{v}}^{\gamma}_{\alpha|\beta}\\ &= \ddot{\mathbf{C}}^{\gamma}_{\beta}\mathbf{r}^{\beta}_{\alpha|\beta} + 2\dot{\mathbf{C}}^{\gamma}_{\beta}\dot{\mathbf{r}}^{\beta}_{\alpha|\beta} + \mathbf{a}^{\gamma}_{\alpha|\beta} &(3.15)\end{aligned}$$

The first term on the right hand side of (3.15) can be reformed by (3.7) and (3.8):

$$\ddot{\mathbf{C}}^{\gamma}_{\beta}\mathbf{r}^{\beta}_{\alpha|\beta} = \left(\boldsymbol{\Omega}^{\gamma}_{\gamma|\beta}\boldsymbol{\Omega}^{\gamma}_{\gamma|\beta} - \dot{\boldsymbol{\Omega}}^{\gamma}_{\gamma|\beta}\right)\mathbf{r}^{\gamma}_{\alpha|\beta} \tag{3.16}$$

The second term on the right hand side of (3.15) can be rearranged by (3.7), (3.8) and (3.11):

$$\dot{\mathbf{C}}_{\beta}^{\gamma}\dot{\mathbf{r}}_{\alpha|\beta}^{\beta} = -\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\mathbf{r}_{\alpha|\beta}^{\gamma} - \boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\dot{\mathbf{r}}_{\alpha|\beta}^{\gamma} \tag{3.17}$$

When (3.16) and (3.17) are put into (3.15), (3.18) is finally obtained.

$$\ddot{\mathbf{r}}_{\alpha|\beta}^{\gamma} = -\left(\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma} + \dot{\boldsymbol{\Omega}}_{\gamma|\beta}^{\gamma}\right)\mathbf{r}_{\alpha|\beta}^{\gamma} - 2\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\dot{\mathbf{r}}_{\alpha|\beta}^{\gamma} + \mathbf{a}_{\alpha|\beta}^{\gamma} \tag{3.18}$$

In (3.18), $\mathbf{a}_{\alpha|\beta}^{\gamma}$ can be stated as force per unit mass that is transferred into a non-reference frame. As a result, two virtual forces are found to appear in the equation of motion if the resolving frame is rotating. The first term, $-\left(\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma} + \dot{\boldsymbol{\Omega}}_{\gamma|\beta}^{\gamma}\right)\mathbf{r}_{\alpha|\beta}^{\gamma}$, which will be witnessed by the observer located in the rotating frame is called centrifugal acceleration. The second term, $-2\boldsymbol{\Omega}_{\gamma|\beta}^{\gamma}\dot{\mathbf{r}}_{\alpha|\beta}^{\gamma}$, is known as the Coriolis acceleration.

3.2.2.3 Earth Surface and Gravity Models

In the application part of this study, the position solution is acquired in the form of geodetic latitude, longitude and altitude from an unit consisting GPS module. For this reason, the models for the Earth's surface are introduced in this section.

To be able obtain the position with respect to the Earth's surface, defining a reference surface regarding the center and the axes of the Earth is the general convention. After that, the set of longitude, latitude and altitude is utilized to describe the location of the body considering this reference surface.

The Ellipsoid Model:

The surface of the Earth is approximated with an ellipsoid placed at the mean sea level. The equatorial radius and the eccentricity of the ellipsoid, e, is commonly indicated to define the ellipsoid [34]. (3.19) gives the definition of eccentricity.

$$e = \sqrt{1 - \frac{R_p^2}{R_0^2}} \tag{3.19}$$

R_0 stands for the length of semi-major axis that is distance from center to any point fixed on the equator. R_p denotes the length of semi-minor axis that is the distance from center to either pole (see 3.2). The polar radius and eccentricity attained from the statements of the equatorial radius and flattening made by World Geodetic System 1984 [35] as follows:

- $R_0 = 6,378,137.0\,m,\ R_p = 6,356,752.3142\,m,\ e = 0.0818191908425.$

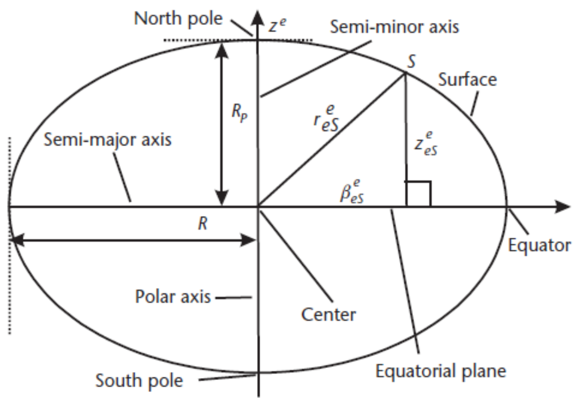


Figure 3.2: Ellipsoid model of the Earth's surface, [3].

Curvilinear Position:

Three mutually orthogonal coordinates which are aligned with the local navigation frame resolve position information. The *height* (h) or *altitude* is the distance between the body and the surface along the normal line to the Earth's surface. The point on surface where the normal line intersects with the surface is symbolized by *S(b)*. Thus, the *latitude* (L) is the north-south axis coordinate of *S(b)*. The *longitude* (λ) is the east-west axis coordinate of *S(b)*.

The transverse radius of curvature for east-west motion, R_E, shows at which rate longitude varies over the surface perpendicular to a meridian. It is expressed in (3.20).

$$R_E(L) = \frac{R_0}{\sqrt{1 - e^2\ \sin^2 L}} \tag{3.20}$$

The Cartesian positions in ECEF frame is computed with the following equation set [3].

$$\begin{aligned}
x^e_{b|e} &= \left(R_E(L_{S(b)}) + h_{S(b)}\right)\cos L_{S(b)}\cos\lambda_{S(b)} \\
y^e_{b|e} &= \left(R_E(L_{S(b)}) + h_{S(b)}\right)\cos L_{S(b)}\sin\lambda_{S(b)} \\
z^e_{b|e} &= \left[(1 - e^2)\ R_E(L_{S(b)}) + h_{S(b)}\right]\sin L_{S(b)}
\end{aligned} \tag{3.21}$$

Specific Force and Gravitation:

Specific force, **f**, is defined as forces exerted on a body with respect to an inertial frame, excluding the gravitational ones per unit mass. *Gravitation* is the force resulted from the attraction between the masses the body and the Earth. The acceleration caused by gravitation is denoted by γ. The accelerometers measure the specific force on the body with respect to an inertial frame as shown in (3.22).

$$\mathbf{f}^{i}_{b|i} = \mathbf{a}^{i}_{b|i} - \boldsymbol{\gamma}^{i}_{b|i} \tag{3.22}$$

The acceleration resulted from gravity is signified by $\mathbf{g}^{i}_{b}$. It should be noted that a stationary object with respect to the Earth frame only senses the acceleration which is the reaction to the gravity.

Throughout this work, the calculation of the gravitational force is based on World Geodetic System 1984 [35].

3.2.2.4 Phases of Inertial Navigation

As previously explained with the justifications, the navigation solution is decided to be sought for the flat local-navigation-frame, $\{n\}$, which is assumed to be inertial. Thus, the specific forces, $\mathbf{f}^{b}_{b|i}$, and angular rates, $\boldsymbol{\omega}^{b}_{b|i}$ are accepted to be measured with respect to local frame. Consequently, the mechanization equations are derived for this inertial frame. Throughout the derivation process, several first order approximations are made and the iterations are supposed to be performed at the rate of IMU output. Figure 3.3 illustrates the basic diagram of the algorithm. The integration time, τ_i is the time interval between two successive process cycles. The (–) and (+) signs represent the value of the related property at the beginning and end of a process cycle, respectively.

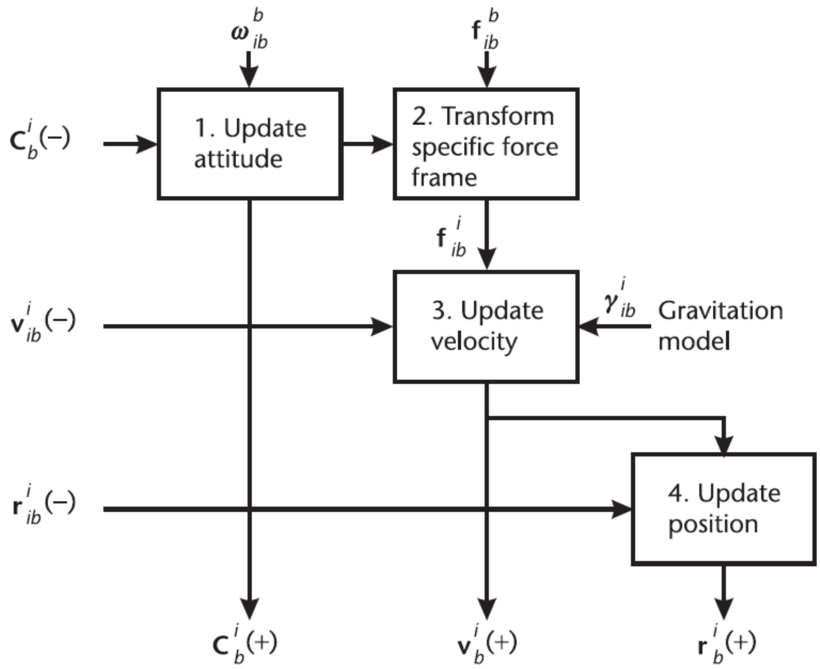


Figure 3.3: Diagram of inertial navigation equations, [3].

Attitude Update:

At the beginning of every cycle, the angular rate measurement of the IMU is used to update the transformation matrix, $\boldsymbol{C}_b^i$ which transforms the vectors from body to local frame. Meanwhile, $\{i\}$ is preferred on purpose in the following equations to emphasize inertial behavior of the local frame, $\{n\}$. There is no discrepancy to replace $\{i\}$ with $\{n\}$ in any step.

(3.23) is adapted from (3.7).

$$\dot{\mathbf{C}}_b^i = \mathbf{C}_b^i \boldsymbol{\Omega}_{b|i}^b \tag{3.23}$$

$\boldsymbol{\Omega}_{b|i}^b$ formed by (3.6) can be rewritten as $\boldsymbol{\Omega}_{b|i}^b = S(\boldsymbol{\omega}_{b|i}^b)$ where $S(\cdot)$ represents the operator forming skew-symmetric matrix of the input vector.

By integrating (3.7), the transformation matrix is found as:

$$\mathbf{C}_b^i(t+\tau_i) = \mathbf{C}_b^i(t) exp\left(\int_t^{t+\tau} \boldsymbol{\Omega}_{b|i}^b \; dt\right) \tag{3.24}$$

Under the assumption of constant angular rates over the integration interval and applying first order approximation to power series expansion, (3.24) takes the following form [3]:

$$\mathbf{C}_b^i(+) \approx \mathbf{C}_b^i(-)(\mathbf{I}_3 + \boldsymbol{\Omega}_{b|i}^b \tau_i) \tag{3.25}$$

and

$$\mathbf{I}_3 + \boldsymbol{\Omega}_{b|i}^b \tau_i = \begin{pmatrix} 1 & -r_{b|i}^b\ \tau_i & q_{b|i}^b\ \tau_i \\ r_{b|i}^b\ \tau_i & 1 & -p_{b|i}^b\ \tau_i \\ -q_{b|i}^b\ \tau_i & p_{b|i}^b\ \tau_i & 1 \end{pmatrix} \tag{3.26}$$

where $\boldsymbol{\omega}_{b|i}^b = \left[p_{b|i}^b\ \ q_{b|i}^b\ \ r_{b|i}^b\right]^T$.

Specific-Force Transformation:

The output of the IMU is resolved in the body frame. Hence, it is required to be transformed into local frame to acquire the velocities by integration.

$$\mathbf{f}_{b|i}^i(t) = \mathbf{C}_b^i(t)\ \mathbf{f}_{b|i}^b(t) \tag{3.27}$$

The coordinate transformation matrix is approximated by the average of the values appear at (+) and (–).

$$\mathbf{f}_{b|i}^i \approx \frac{1}{2}\left(\mathbf{C}_b^i(-) + \mathbf{C}_b^i(+)\right)\mathbf{f}_{b|i}^b \tag{3.28}$$

Velocity Update:

The inertial acceleration is attained by the summation of the specific force measurement and the gravitational acceleration:

$$\mathbf{a}_{b|i}^i = \mathbf{f}_{b|i}^i + \boldsymbol{\gamma}_{b|i}^i\left(\mathbf{r}_{b|i}^i\right) \tag{3.29}$$

$\boldsymbol{\gamma}_{b|i}^i$ is the function of Cartesian-position of the object with respect to the Earth's center and yields the magnitude of the gravitational acceleration. By taking the slowly changing profile of the gravitational field into account, $\mathbf{r}_{b|i}^i(-)$ is delivered as the input of the function.

The velocity in local frame is as follows since the derivative is taken in the reference frame:

$$\dot{\mathbf{v}}_{b|i}^i = \mathbf{a}_{b|i}^i \tag{3.30}$$

The velocity update is found by integrating (3.30) in the following way as the exact value of acceleration during the integration interval is not available:

$$\mathbf{v}^i_{b|i}(+) = \mathbf{v}^i_{b|i}(-) + \mathbf{a}^i_{b|i}\tau_i \tag{3.31}$$

Position Update:
The time derivative of the Cartesian position is exactly equal to the velocity since the reference and resolving frames are identical.

$$\dot{\mathbf{r}}^i_{b|i} = \mathbf{v}^i_{b|i} \tag{3.32}$$

The approximated velocity through the process interval is the average of the start and end values. Thus, the position is formulated as:

$$\begin{aligned} \mathbf{r}^i_{b|i}(+) &= \mathbf{r}^i_{b|i}(-) + \left(\mathbf{v}^i_{b|i}(-) + \mathbf{v}^i_{b|i}(+)\right)\frac{\tau_i}{2} \\ &= \mathbf{r}^i_{b|i}(-) + \mathbf{v}^i_{b|i}(-)\tau_i + \mathbf{a}^i_{b|i}\frac{\tau_i^2}{2} \\ &= \mathbf{r}^i_{b|i}(+) + \mathbf{v}^i_{b|i}(+)\tau_i - \mathbf{a}^i_{b|i}\frac{\tau_i^2}{2} \end{aligned} \tag{3.33}$$

3.3 Radio and Satellite Navigation

Radio signals have being exploited for navigation purposes for more than eight decades now [3]. There are several different methods in use which process radio signals to obtain position and velocity information. Within the scope of this work, *passive ranging* is the only one to be introduced. In such a system, a receiver collects timing signals that are transmitted from numerous reference stations. When the synchronization of the clocks of both receiver and transmitters is achieved, first the distance to transmitters, then the position of the receiver can be calculated.

3.3.1 Global Navigation Satellite Systems

Global navigation satellite systems (GNSS) refers to the passive ranging systems which furnish a positioning solution in three dimensional space by processing radio signals broadcasted by orbiting satellites. There are currently several operating GNSS and regional extensions, the number of which is predicted to radically increase in the near future. Navigation by Satellite Ranging and Timing (NAVSTAR) Global Positioning System (GPS) known shortly as GPS is the most famous one among these. In six different orbits, up to 32 satellites are included in GPS which are built and owned by U.S. government. The other globally functional satellite navigation system is Global Navigation Satellite System (GLONASS) which is operated by Russian Aerospace Defence Forces. It includes 24 satellites in three orbital planes. These two distinct systems can now be benefited at the same time to increase the position tracking accuracy of the navigation system [4].

The space, control/ground and the user segments are three fundamental modules of the design of a satellite navigation system [36]. The orbiting satellites constitute the space segment. They essentially transmit radio signals to the control segment and users. It takes about 12-hour for the satellite vehicles to complete their full revolution around Earth. The orbits represented in Figure 3.4 are almost circular and established on six separate planes which are inclined 60° with respect to the equatorial plane. Those orbit planes are designed to divide the equator into identical six pieces to provide an uniform global coverage as illustrated in the below figure. This leads to such a constellation that any user on the Earth's surface is able to find at least four of satellites in its direct line-of-sight at any time.

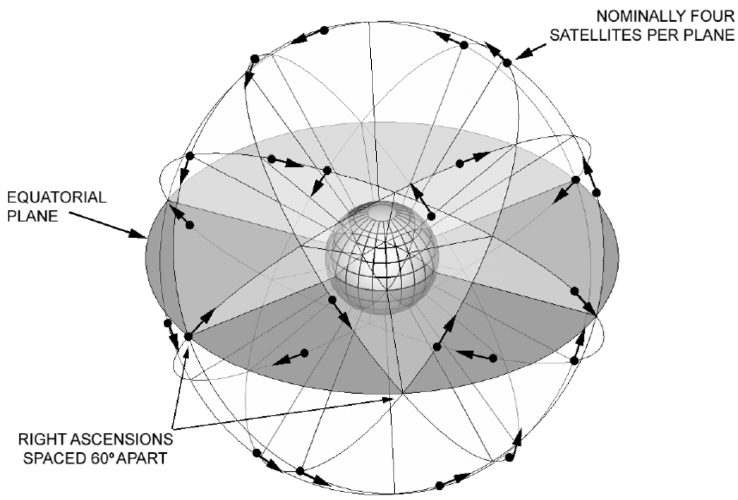


Figure 3.4: GPS satellites orbiting in 6 different planes, [4].

The signals produced by satellite vehicles comprise of ranging measurements and navigation data messages. Those signals are decoded by the receivers to obtain system parameters.

The ground segment keeps track of the health and status of the space segment. It constitutes of several monitor stations, one or more control stations and uplink stations [3]. The ranging measurements are gathered from satellite vehicles and transmitted to the control stations by the monitor stations which utilize the measurements to identify the orbits and calibrate the clocks of satellite vehicles since they are already located at precisely specified positions and have synchronized clocks. The control stations informed by the ranging measurements decide the maneuvers to be performed by satellites to track the correct orbits.

The user equipment consists of an antenna, a receiver and a set of processors. The broadcast signals are captured and transformed into electrical ones. Then, the receiver takes the electrical signals and produces a time reference by employing the synchronized clock. Consequently, the processors calculate the distances between the antenna and each of satellites and compute a position and velocity solution in turn. The details of the algorithms run in the processors of the user equipment is out of the scope of this work and will not be presented here.

In Figure 3.5 a GNSS which is receiving messages from four satellite system is illustrated. $\mathbf{r}_{ia}^{i}$ is the antenna position, $\mathbf{r}_{isj}^{i}$ and ρ_{Cj} are the position and the range measurement of j^{th} satellite, respectively. All position vectors are established with respect to inertial frame and resolved also in inertial frame. By using pseudo-range and the change of pseudo-range measurements, GNSS navigation solution containing the three dimensional position and velocity of the user antenna is furnished.

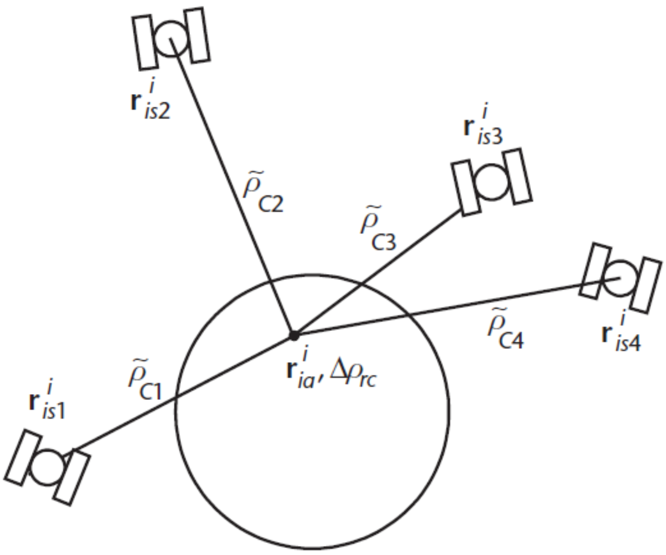


Figure 3.5: Producing position solution by signals from four satellites, [3].

The output of GNSS which will further be employed in integrated navigation techniques is denoted as follows.

$$\hat{\mathbf{m}}_G = \begin{pmatrix} \hat{\mathbf{r}}^e_{a|e,G} \\ \hat{\mathbf{v}}^e_{a|e,G} \end{pmatrix} \tag{3.34}$$

The solution, $\hat{m}_G$ is assumed to be generated in ECEF frame which is in conformance with the hardware used in experiments.

3.4 Attitude Estimation

Attitude information of the body is to be obtained from an external source due to growing error characteristics of INS. However, a sensor that directly outputs the orientation of the vessel is not available. Consequently, an algorithm estimates the attitude from magnetometer and accelerometer measurements is constructed.

As mentioned earlier, accelerometer measures only the acceleration resulting from the reaction of gravity when the body is stationary with respect to the Earth's surface. Additionally, the translational accelerations can be assumed to be smaller by approximately a factor of 10 than the gravitational acceleration. Thus, it is possible to detect the direction of gravitational force by only using this measurement. By doing so, the z, axis of the local navigation frame is assumed to be aligned with the direction of gravitational force since the vehicle operates on sea surface and local irregularities can be neglected.

The other reference measurement is taken from the 3-axis magnetometer mounted on the hull. It measures the total magnetic flux density resolved in the body frame and can be interpreted to obtain the vector directed from the body to the magnetic north pole. Moreover, this vector is guarantied to lie on the (x, z) plane of the local navigation frame as it a NED coordinate frame [37]. Afterward, y axis is simply acquired by the cross product of two measurements using the fact that it should be perpendicular to previously mentioned plane. At this point, the y and z axes of local navigation frame which are resolved in the body frame are attained. Finally, the x axis is derived by using mutual orthogonality property of the axes.

Before the calculation of attitude from measurements, an extended version of Kalman Filter is employed to eliminate sensor noises. The state and measurement vectors are selected as depicted in (3.35).

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\omega}^b_{b|n} \\ \dot{\boldsymbol{\omega}}^b_{b|n} \\ \boldsymbol{r}^b_g \\ \boldsymbol{r}^b_m \end{bmatrix}, \quad \boldsymbol{z} = \begin{bmatrix} \tilde{\boldsymbol{\omega}}^b_{b|n} \\ \tilde{\boldsymbol{r}}^b_g \\ \tilde{\boldsymbol{r}}^b_m \end{bmatrix} \tag{3.35}$$

where $\omega^b_{b|n}$ is the angular rate vector of the body measured regarding $\{n\}$, $\dot{\omega}^b_{b|n}$ is the time derivative of angular rate vector with respect to $\{n\}$, $\boldsymbol{r}^b_g$ is the gravitational field vector represented in body frame and $\boldsymbol{r}^b_m$ is the vector of magnetic field resolved in body frame. $\boldsymbol{\omega}^b_{b|n}$ and $\boldsymbol{r}^b_m$ are directly measured by gyroscope and magnetometer, respectively. $\boldsymbol{r}^b_g$ is obtained by using accelerometer measurement. Recall that $\{n\}$ illustrates the local frame which is supposed to be inertial.

3.4.1 Extended Kalman Filter Implementation

3.4.1.1 Prediction

The rate of change of angular rate vector is assumed to be constant over a period of cycle, τ_i, which results in (3.36). The subscripts and superscripts are omitted in below equations to simplify the notation although all variables are still resolved in the body frame.

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \tau_i\ \dot{\boldsymbol{\omega}}_k \tag{3.36}$$

To be able to predict the field vectors, the time derivatives are formed in accordance with the following vector derivation equation:

$$\begin{aligned} \boldsymbol{v}^b &= \dot{\boldsymbol{r}}^b + \ \boldsymbol{\omega}^b \times \boldsymbol{r}^b \\ &= \dot{\boldsymbol{r}}^b + S\left(\boldsymbol{\omega}^b\right)\boldsymbol{r}^b \end{aligned} \tag{3.37}$$

where $\boldsymbol{v}^b$ stands for the time derivative of field in body frame, $\dot{\boldsymbol{r}}^b$ denotes the derivate taken in inertial frame and ω^b is the angular rate vector between the inertial and body frames. The first term on the right hand side of (3.37), $\dot{\boldsymbol{r}}^b$ is taken as zero considering the uniformity of the gravity and magnetic vector fields during a translational motion [37]. Consequently, the nonlinear system model is written in the following way:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \mathbf{w}) = \begin{pmatrix} \boldsymbol{\omega}_k + \tau_i\ \dot{\boldsymbol{\omega}}_k + \mathbf{w}_1 \\ \dot{\boldsymbol{\omega}}_k + \mathbf{w}_2 \\ \boldsymbol{r}_{g,k} + S\left(\boldsymbol{\omega}_k\right)\boldsymbol{r}_{g,k}\tau_i + \mathbf{w}_3 \\ \boldsymbol{r}_{m,k} + S\left(\boldsymbol{\omega}_k\right)\boldsymbol{r}_{m,k}\tau_i + \mathbf{w}_4 \end{pmatrix} \tag{3.38}$$

where $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ and $\mathbf{w}_4$ represent process noises.

The nonlinear model is linearized around the predicted state vector in (3.39) [37].

$$A_k = \left.\frac{\partial\left(f(\boldsymbol{x}_k, \mathbf{w})\right)}{\partial \boldsymbol{x}_k}\right|_{\hat{x}_k^-} = \begin{bmatrix} \mathbf{I}_3 & \tau_i\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\tau_i S\left(\boldsymbol{r}_{g,k}\right) & \mathbf{0}_3 & \mathbf{I}_3 + \tau_i S\left(\boldsymbol{\omega}_k\right) & \mathbf{0}_3 \\ -\tau_i S\left(\boldsymbol{r}_{m,k}\right) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \tau_i S\left(\boldsymbol{\omega}_k\right) \end{bmatrix} \tag{3.39}$$

The predicted states are calculated with the original nonlinear model in (3.40).

$$\hat{\boldsymbol{x}}_{k+1}^{-} = f(\hat{\boldsymbol{x}}_k, \mathbf{0}) \tag{3.40}$$

The covariance matrix belonging to predicted states is iterated as:

$$P_{k+1}^{-} = A_k P_k A_k^T + Q_k \tag{3.41}$$

where the process covariance matrix, Q_k, is a diagonal matrix consisting of variances of the states. The value of Q_k is tuned through implementation since it is not directly measured.

3.4.1.2 Correction

The measurement vector, z_k, consisting of three measurements is presented as:

$$\begin{aligned} z_k &= H_k \boldsymbol{x}_k + R_k \\ H_k &= \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{pmatrix}, \quad R_k = \begin{pmatrix} \sigma_\omega^2 \\ \sigma_{r_g}^2 \\ \sigma_{r_m}^2 \end{pmatrix} \end{aligned} \tag{3.42}$$

where R_k stands for the measurement covariance matrix. The measurement model is assumed to be fixed over time as the new readings of gyroscope, accelerometer and magnetometer arrive at the same time step within a synchronized data package in the utilized autopilot card, Pixhawk.

The Kalman gain, posteriori estimation and propagation of error covariance are given in the following equations, respectively.

$$\mathbf{K}_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \tag{3.43}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k (z_k - H_k \hat{\boldsymbol{x}}_k^-) \tag{3.44}$$

$$P_k = (\mathbf{I} - \mathbf{K}_k H_k)) P_k^- \tag{3.45}$$

3.4.2 Equations to Obtain Attitude

To obtain orientation, the outputs of the Kalman filter, $\hat{\boldsymbol{r}}_g$ and $\hat{\boldsymbol{r}}_m$ are employed. At the end of the procedure, basis vectors of the local frame, $\boldsymbol{e}_{n_x}^b$, $\boldsymbol{e}_{n_y}^b$, $\boldsymbol{e}_{n_z}^b$, which are resolved in the body frame will be determined. Realizing the bases in column form, the coordinate transformation matrix from local to body frame can be obtained by the concatenation of three vectors in the following manner.

$$\mathbf{C}_n^b = \begin{bmatrix} \boldsymbol{e}_{n_x}^b & \boldsymbol{e}_{n_y}^b & \boldsymbol{e}_{n_z}^b \end{bmatrix} \tag{3.46}$$

As previously explained, the direction of gravitational force is accepted to be aligned with z.

$$\boldsymbol{e}_{n_z}^b = \frac{\hat{\boldsymbol{r}}_g}{||\hat{\boldsymbol{r}}_g||} \tag{3.47}$$

When the translational acceleration of the boat is different than zero, (3.47) introduces an error to attitude estimation.

Moreover, $\boldsymbol{e}_{n_y}^b$ is perpendicular to north direction due to the fact that the local frame is a NED frame of reference. Thus, $\hat{\boldsymbol{r}}_m$ and $\boldsymbol{e}_{n_z}^b$ is employed to determine $\boldsymbol{e}_{n_y}^b$.

$$\boldsymbol{e}_{n_y}^b = \frac{\boldsymbol{e}_{n_z}^b \times \hat{\boldsymbol{r}}_m}{||\boldsymbol{e}_{n_z}^b \times \hat{\boldsymbol{r}}_m||} \tag{3.48}$$

Finally, $\boldsymbol{e}_{n_x}^b$ is simply acquired by the succeeding cross product.

$$\boldsymbol{e}_{n_x}^b = \frac{\boldsymbol{e}_{n_y}^b \times \boldsymbol{e}_{n_z}^b}{||\boldsymbol{e}_{n_y}^b \times \boldsymbol{e}_{n_z}^b||} \tag{3.49}$$

Having obtained the transformation matrix, Euler angles can be recovered by basic trigonometric manipulations.

3.5 Integrated Navigation

INS offers a continuous navigation solution with high bandwidth, smooth short term errors and low random noise. Nevertheless, it is not suitable for stand alone use as the accuracy of the output reduces by the tendency of the errors to increase with time.

The RMS errors are not bounded due to the continuous integration of errors involved in the sensor measurements and gravity model. On the other hand, the errors in GNSS solution depend on the availability and geometric distribution of the satellite vehicles that it can track. Therefore, its performance does not degrade with time. RMS errors of position and velocity solutions are bounded if there are enough number of available satellites. However, the output of GNSS is updated at a relatively low rate and it can even be unavailable as a result of the shortage of GNSS signals. Additionally, a typical GNSS does not provide attitude solution. Consequently, an integrated navigation system is proposed taking the complementary characteristics of two methods into account. Finally, the attitude estimates are also benefited to prevent Euler angles of INS from divergence. The entire system is designed to improve the overall performance by reducing RMS errors in position, velocity and attitude solutions. To be able to evaluate the expected integrated performance, statistical models presenting the contribution of individual error sources to the performance of the overall system are utilized.

3.5.1 Integration Architectures

3.5.1.1 INS and GNSS Integration

In this study, a loosely coupled INS/GNSS integration is enforced due to its simplicity and redundancy [4]. In this structure, the position and velocity outputs of GNSS are directly provided to the integration filter as measurements. It is a cascaded design since the outputs of GNSS are already processed by a navigation filter. An error state Kalman filter employed as the integration algorithm. The integrated navigation solution is achieved by the correction of INS solution regarding the states of Kalman filter. There are two feasible techniques to apply those corrections: open loop and closed loop. In this study, both are investigated, realized and compared.

Open Loop Configuration:

In the open loop architecture, which is shown in the below figure, the Kalman estimates of state errors are utilized to correct the INS solution yielding the corrected navigation solution. However, those errors are not fed back to INS. As a result, the raw INS and integrated solutions are separately maintained.

The raw inertial navigation solution is represented by $\tilde{\mathbf{C}}_b^n$, $\tilde{\mathbf{v}}_{b|n}^n$ and $\tilde{\mathbf{r}}_{b|n}^n$. The corrected or integrated inertial navigation solution is denoted by $\hat{\mathbf{C}}_b^n$, $\hat{\mathbf{v}}_{b|n}^n$ and $\hat{\mathbf{r}}_{b|n}^n$. The attitude, velocity and position errors are defined by the following equations.

$$\hat{\mathbf{C}}_b^n = \delta\hat{\mathbf{C}}_b^{n\mathrm{T}}\tilde{\mathbf{C}}_b^n \tag{3.50}$$

$$\hat{\mathbf{v}}_{b|n}^n = \tilde{\mathbf{v}}_{b|n}^n - \delta\hat{\mathbf{v}}_{b|n}^n \tag{3.51}$$

$$\hat{\mathbf{r}}_{b|n}^n = \tilde{\mathbf{r}}_{b|n}^n - \delta\hat{\mathbf{r}}_{b|n}^n \tag{3.52}$$

The attitude errors in (3.50) can be rearranged under the assumption of small angle [3].

$$\hat{\mathbf{C}}_b^n = \left(\mathbf{I}_3 - \left[S\left(\delta\hat{\Theta}_{b|n}\right)\right]\right)\tilde{\mathbf{C}}_b^n \tag{3.53}$$

where $\delta\hat{\Theta}_{b|n}^n$ is the estimation of Euler angle error by the Kalman filter.

State Selection and System Model:

The state vector of the integrating Kalman filter is selected as below:

$$\boldsymbol{x} = \begin{pmatrix} \delta\Theta_{b|n} \\ \delta\mathbf{v}_{b|n}^n \\ \delta\mathbf{r}_{b|n}^n \\ \mathbf{b}_a \\ \mathbf{b}_g \end{pmatrix} \tag{3.54}$$

$\delta\Theta_{b|n}$, $\delta\mathbf{v}_{b|n}^n$ and $\delta\mathbf{r}_{b|n}^n$ stand for attitude, velocity and position errors, respectively. The error states are resolved in local frame given that the navigation solution is sought for the local navigation frame. Besides, the biases of accelerometer, $\mathbf{b}_a$, and gyroscope, $\mathbf{b}_g$, are also involved in the state vector considering the fact that those are the most dominant sources of error in inertial navigation solution. Before proceeding with the derivation of state propagation model of the filter, it is found essential to recall that the local navigation frame is assumed to be inertial.

Taking the derivative of (3.53), (3.55) is obtained

$$S\left(\delta\dot{\Theta}_{b|n}\right) \approx \delta\dot{\mathbf{C}}_b^n \tag{3.55}$$

By using (3.7) and (3.50), (3.55) takes the form:

$$\tilde{\mathbf{C}}_b^n \left[S\left(\delta\dot{\Theta}_{b|n}\right)\right] \approx \tilde{\boldsymbol{\Omega}}_{b|i}^b - \boldsymbol{\Omega}_{b|i}^b \tag{3.56}$$

(3.56) is further manipulated by the approximation, $\delta\Theta_{b|n}\delta\dot{\Theta}_{b|n} \approx 0$ referring to linearity assumption of the states and finalized by (3.57). Details of the derivation can be found in [3].

$$\dot{\Theta}_{b|n} \approx \hat{\mathbf{C}}_b^n \left(\tilde{\boldsymbol{\omega}}_{b|n}^b - \boldsymbol{\omega}_{b|n}^b\right) = \hat{\mathbf{C}}_b^n \delta\boldsymbol{\omega}_{b|n}^b \tag{3.57}$$

The only error that is modelled as the state of the filter is the bias component. Thus, (3.57) becomes:

$$\dot{\Theta}_{b|n} \approx \hat{\mathbf{C}}_b^n \mathbf{b}_g \tag{3.58}$$

The time derivative of the velocity in local frame can be defined as:

$$\dot{\mathbf{v}}_{b|n}^n = \mathbf{f}_{b|n}^n + \boldsymbol{\gamma}_{b|n}^n \tag{3.59}$$

Thus, the time derivative of the velocity error can be written as in (3.60).

$$\delta\dot{\mathbf{v}}_{b|n}^n = \tilde{\mathbf{f}}_{b|n}^n - \mathbf{f}_{b|n}^n + \tilde{\boldsymbol{\gamma}}_{b|n}^n - \boldsymbol{\gamma}_{b|n}^n \tag{3.60}$$

As can be seen from (3.60), there are two distinct sources of velocity error. The first one, represented by $\delta\mathbf{f}_{b|n}^n = \tilde{\mathbf{f}}_{b|n}^n - \mathbf{f}_{b|n}^n$ corresponds to the error in specific force while the second component, symbolized by, $\delta\boldsymbol{\gamma}_{b|n}^n = \tilde{\boldsymbol{\gamma}}_{b|n}^n - \boldsymbol{\gamma}_{b|n}^n$, stands for the error in gravitational acceleration. The gravitational term will be neglected regarding to the slow variation of the position of a marine craft operating in a local area.

The specific forces are originally measured in body frame; hence, transforming those into local frame by an erroneous transformation matrix also introduces error:

$$\begin{aligned} \delta\mathbf{f}_{b|n}^n &= \tilde{\mathbf{C}}_b^n \tilde{\mathbf{f}}_{b|n}^b - \mathbf{C}_b^n \mathbf{f}_{b|n}^b && (3.61)\\ &\approx \hat{\mathbf{C}}_b^n \left(\tilde{\mathbf{f}}_{b|n}^b - \mathbf{f}_{b|n}^b\right) + \left(\tilde{\mathbf{C}}_b^n - \mathbf{C}_b^n\right)\hat{\mathbf{f}}_{b|n}^b && (3.62) \end{aligned}$$

The only error source in accelerometer measurement which is modelled as a state of the filter is the bias component.

$$\delta\mathbf{f}_{b|n}^b = \tilde{\mathbf{f}}_{b|n}^b - \mathbf{f}_{b|n}^b \approx \mathbf{b}_a \tag{3.63}$$

Finally, the model for velocity error state is completed by putting (3.53) and (3.63) into (3.61).

$$\delta\dot{\mathbf{v}}^n_{b|n} = -S\left(\hat{\mathbf{C}}^n_b\hat{\mathbf{f}}^b_{b|n}\right)\delta\Theta_{b|n} + \hat{\mathbf{C}}^n_b\mathbf{b}_a \tag{3.64}$$

The time derivative of position in inertial frame is simply velocity. Consequently, the time derivative of position error in local frame is equal to velocity error.

$$\delta\dot{\mathbf{r}}^n_{b|n} = \delta\mathbf{v}^n_{b|n} \tag{3.65}$$

The time derivatives of the gyro and accelerometer biases are taken as zero.

$$\dot{\mathbf{b}}_a = \mathbf{0}, \quad \dot{\mathbf{b}}_g = \mathbf{0} \tag{3.66}$$

In the end, the system model can be put into standard form which is represented in (3.67).

$$\begin{pmatrix} \delta\dot{\Theta}_{b|n} \\ \delta\dot{\mathbf{v}}^n_{b|n} \\ \delta\dot{\mathbf{r}}^n_{b|n} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \end{pmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}^n_b \\ -S\left(\hat{\mathbf{C}}^n_b\hat{\mathbf{f}}^b_{b|n}\right) & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}^n_b & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{pmatrix} \delta\Theta_{b|n} \\ \delta\mathbf{v}^n_{b|n} \\ \delta\mathbf{r}^n_{b|n} \\ \mathbf{b}_a \\ \mathbf{b}_g \end{pmatrix} \tag{3.67}$$

The model given in (3.67) is discretized with propagation time step, τ_s, by first order approximation of the power series. In application part of this study, τ_s is selected to be equal to INS integration time, τ_i, which is about 10 ms. The resulting state transition matrix, Φ, is indicated in (3.68).

$$\Phi = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}^n_b\tau_s \\ -S\left(\hat{\mathbf{C}}^n_b\hat{\mathbf{f}}^b_{b|n}\right)\tau_s & \mathbf{I}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}^n_b\tau_s & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3\tau_s & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.68}$$

The time variation of velocity and attitude errors resulted from the integration of accelerometer and gyro errors are modeled as white noises. As discussed in Inertial

Sensors section, the biases of IMU can also be described as white noises. Therefore, the INS system noise covariance matrix, Q_{INS} is defined as in (3.69) [3].

$$Q_{INS} = \begin{pmatrix} n_{rg}^2\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & n_{ra}^2\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & n_{bad}^2\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & n_{bgd}^2\mathbf{I}_3 \end{pmatrix} \tag{3.69}$$

where n_{rg}^2, n_{ra}^2, n_{bad}^2 and n_{bgd}^2 stand for the power spectral densities of the gyro random noise, accelerometer random noise, gyro bias variation and accelerometer bias variation, respectively.

During the correction phase of the integration filter, the differences between outputs of GNSS user equipment and inertial navigation solution are provided as measurement input. $\mathbf{z}_G$ and $\hat{\mathbf{z}}_G$ denote the Kalman filter measurement based on GNSS outputs and its estimate, respectively. They are defined as:

$$\mathbf{z}_G = \tilde{\mathbf{m}}_G - \tilde{\mathbf{m}}_I \tag{3.70}$$

$$\hat{\mathbf{z}}_G = \delta\hat{\mathbf{m}}_G - \delta\hat{\mathbf{m}}_I \tag{3.71}$$

where $\tilde{\mathbf{m}}_G$ and $\tilde{\mathbf{m}}_I$ respectively indicate the outputs of GNSS and INS.

Both velocity and position outputs of GNSS are utilized in this work. Thus, the innovation can be described in the following way [3].

$$\delta\mathbf{z}_{G,k}^{n-} = \begin{pmatrix} \delta\mathbf{r}_{b|n,k}^n \\ \delta\mathbf{v}_{b|n,k}^n \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{r}}_{a|n,G}^n - \hat{\mathbf{r}}_{b|n}^n - \hat{\mathbf{C}}_b^n I_{a|b}^b \\ \hat{\mathbf{v}}_{a|n,G}^n - \hat{\mathbf{v}}_{b|n}^n - \hat{\mathbf{C}}_b^n \left[S\left(\hat{\boldsymbol{\omega}}_{b|n}^b\right) I_{a|b}^b\right] \end{pmatrix}_k \tag{3.72}$$

where $I_{a|b}^b$ illustrates the lever arm from the origin of body frame to the GNSS antenna resolved in body frame.

By computing the Jacobian of measurement vector given by (3.72) regarding the state vector illustrated by (3.54), the measurement matrix is found in (3.73).

$$H_{G,k} = \begin{bmatrix} \left[S\left(\hat{\mathbf{C}}_b^n I_{a|b}^b\right)\right] & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \left[S\{\hat{\mathbf{C}}_b^n\left(S\left(\hat{\boldsymbol{\omega}}_{b|n}^b\right) I_{a|b}^b\right)\}\right] & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}_b^n\left[S\left(I_{a|b}^b\right)\right] \end{bmatrix}_k \tag{3.73}$$

Considering to the fact that the length of the above level arm is about a few centimeters, the attitude errors and gyro biases are hardly related with the measurements. Therefore, the measurement matrix is simplified by neglecting those terms.

$$H_{G,k} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}_k \tag{3.74}$$

The GNSS measurement covariance matrix, R_G is assumed to be a time invariant diagonal matrix as the hardware in use does not provide any related information in real time.

Closed Loop Configuration:

This method is realized by periodically feeding the estimated attitude, velocity and position errors back to the inertial navigation solution. Kalman filter is immediately reinitialized whenever the corrections are fed back due to the fact that the difference between INS and integrated navigation solution is zeroed by doing so. The states, system model, measurement model and covariance matrices involved in the integrating Kalman filter are identical to those of the open loop structure. In this configuration, the corrections based on the outputs of error state filter are applied using below three formulas.

$$\hat{\mathbf{C}}_b^n(+) = \delta\hat{\mathbf{C}}_b^{n\mathrm{T}}\hat{\mathbf{C}}_b^n(-) \approx \left(I_3 - \left[S\left(\delta\hat{\Theta}_{b|n}\right)\right]\right)\hat{\mathbf{C}}_b^n(-) \tag{3.75}$$

$$\hat{\mathbf{v}}_{b|n}^n(+) = \hat{\mathbf{v}}_{b|n}^n(-) - \delta\hat{\mathbf{v}}_{b|n}^n \tag{3.76}$$

$$\hat{\mathbf{r}}_{b|n}^n(+) = \hat{\mathbf{r}}_{b|n}^n(-) - \delta\hat{\mathbf{r}}_{b|n}^n \tag{3.77}$$

where the variables with (+) form the corrected navigation solution while the variables with (-) are the results by the prediction phase of the filter.

The above given corrections are fed back to inertial navigation solution at the cycles in which new GNSS measurement becomes available that occurs at about 5 Hz for the used GNSS user equipment. Therefore, between two measurement updates, the integrated solution is iterated by the mechanization equations of INS. For the same interval of time, the Kalman filter continues to operate as well by making predictions of the errors.

3.5.1.2 INS, GNSS and Attitude Estimation Integration

Being independent of its configuration type, integrated INS/GNSS architecture does not have a separate attitude measurement. It practically keeps track of the Euler angles by processing and integrating the angular rate measurements. Thus, the attitude solution has a tendency to diverge in a finite interval of time, which renders the external aid for attitude compulsory.

The attitude estimation, denoted by $\hat{\Theta}_{b|n,M}$ is already obtained by using the accelerometer and magnetometer measurements. It is illustrated with a hat symbol, since it is generated by the preceding Kalman filter. The system model and process covariance matrix used in the integrating Kalman filter are kept unchanged in this structure. The measurement vector of the filter is augmented with the attitude estimation. Then, the innovation takes the form indicated in (3.78).

$$\delta \mathbf{z}_k^{n-} = \begin{pmatrix} \delta \mathbf{r}^n_{b|n,k} \\ \delta \mathbf{v}^n_{b|n,k} \\ \delta \Theta_{b|n,k} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{r}}^n_{a|n,G} - \hat{\mathbf{r}}^n_{b|n} - \hat{\mathbf{C}}^n_b I^b_{a|b} \\ \hat{\mathbf{v}}^n_{a|n,G} - \hat{\mathbf{v}}^n_{b|n} - \hat{\mathbf{C}}^n_b \left[S\left(\hat{\boldsymbol{\omega}}^b_{b|n}\right) I^b_{a|b} \right] \\ \hat{\Theta}_{b|n,M} - \hat{\Theta}_{b|n} \end{pmatrix}_k \tag{3.78}$$

The measurement matrix takes the following form.

$$H_k = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}_k \tag{3.79}$$

Moreover, the measurement covariance matrix is established by combining the covariance matrices associated with the attitude estimation, $R_{M,k}$ and GNSS.

$$R_k = \begin{pmatrix} R_G & \mathbf{0} \\ \mathbf{0} & R_{M,k} \end{pmatrix} \tag{3.80}$$

Incomplete Measurement:

The complete measurement vector will not be available for every correction phase since the attitude and GNSS measurements are delivered at different rates. To avoid waiting a new measurement supplied by one information source for the other one to complete entire vector or using the old measurement for multiple times, the measurement model is adapted according to the type of available update.

If new GNSS and attitude measurements are both available at the same time the integrating filter uses (3.79) and (3.80).

If only GNSS data is received, the filter makes use of (3.72) (3.73).

In case of the appearance of an unaccompanied attitude measurement, the model is modified as below.

$$\delta \mathbf{z}_k^{n-} = \left(\delta \Theta_{b|n,k}\right) = \left(\hat{\Theta}_{b|n,M} - \hat{\Theta}_{b|n}\right)_k \tag{3.81}$$

$$H_k = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}_k \tag{3.82}$$

$$R_k = R_{M,k} \tag{3.83}$$

3.6 Estimation Algorithm

In this study, two distinct estimation algorithms, one to produce attitude estimation and the other to integrate information coming from multiple sources, are implemented. Both of the algorithms are based on the Kalman Filter. Although they slightly vary in application, their main structures, which will simply be introduced in this section, are substantially identical.

The Kalman Filter estimates the states of a linear dynamic system from a series of noisy measurement in an iterative and efficient manner. It propagates the states and error covariances according to the model of the system. Subsequently, exploiting the measurements, measurement accuracy information and predicted states, it revises the states and covariances. The filter steps will briefly be provided below while the details of the derivation can be found in [38].

w_k and v_k are system and measurement noises; Q_k and R_k are corresponding covariance matrices, respectively.

$$x_{k+1} = \Phi_k x_k + w_k$$

$$y_k = H_k x_k + v_k$$

Initialization of the filter.

$$\begin{aligned}
\hat{x}_0 &= E[x_0] \\
P_0 &= E\left[(x_0 - E[x_0])(x_0 - E[x_0])^{\mathrm{T}}\right]
\end{aligned}$$

K_k is the Kalman gain and z_k is denoted for the measurement.

$$\begin{aligned}
\hat{x}_{k+1}^- &= \Phi_k \hat{x}_k^+ \\
P_{k+1}^- &= \Phi_k P_k^+ \Phi_k^{\mathrm{T}} + Q_k \\
K_{k+1} &= P_{k+1}^- H_{k+1}^{\mathrm{T}} \left(H_{k+1} P_{k+1}^- H_{k+1}^{\mathrm{T}} + R_{k+1}\right)^{-1} \\
\hat{x}_{k+1}^+ &= \hat{x}_{k+1}^- + K_{k+1}\left(z_{k+1} - H_{k+1}\hat{x}_{k+1}^-\right) \\
P_{k+1}^+ &= (I - K_{k+1}H_{k+1}) P_{k+1}^-
\end{aligned}$$

3.7 Simulation Results

In this section, navigation solutions corresponding to the same example of motion of a body are obtained by using three different integration techniques which are previously introduced. The first architecture integrates GNSS and distinct INS solutions in an open loop fashion, and it is symbolized by Arc_1. An external estimate of attitude is not supplied. The second architecture, Arc_2, differs from the first one as a result of being in closed loop configuration. No additional source of information is available for this method. The third architecture, Arc_3, is again implemented in closed loop structure whereas it also receives attitude estimations as measurement.

To be able to investigate and compare the performances of navigation algorithms, sensor measurements that are consistent with the given models of inertial sensors are produced first. Moreover, the measurements are aimed to be rendered compatible with those collected via the components of autopilot card employed in experimental studies. Consequently, the following regulations are considered by taking the data sheets of the sensors into account:

- The output rate of 100 Hz is chosen for IMU and magnetometer while the velocity and position solutions are provided at 5 Hz by GNSS,

- Scale factor errors of accelerometer, gyroscope and magnetometer are ± 3 %, ± 3 % and ± 1 %, respectively.

- Cross coupling errors of accelerometer, gyroscope and magnetometer are ± 2 %, ± 2 % and ± 1 %, respectively.

- Bias values are initialized by the uniform distributions of $\boldsymbol{U}(-0.491, 0.491)$ m/sec^2 and $\boldsymbol{U}(-0.589, 0.589)$ $^\circ/sec$ for all three axes of accelerometer and gyroscope, respectively.

- The random noises of accelerometer, gyroscope and magnetometer are represented by rate noise spectral densities of 0.0034 $m/sec^2/\sqrt{Hz}$, $0.0014^\circ/sec/\sqrt{Hz}$, 5 $mgauss/\sqrt{Hz}$, respectively.

- The horizontal position accuracy of GNSS is about 2.5 m when it is connected to GPS satellite vehicles.

The accelerometer and gyroscope measurements are exemplified in 3.6 and 3.7.

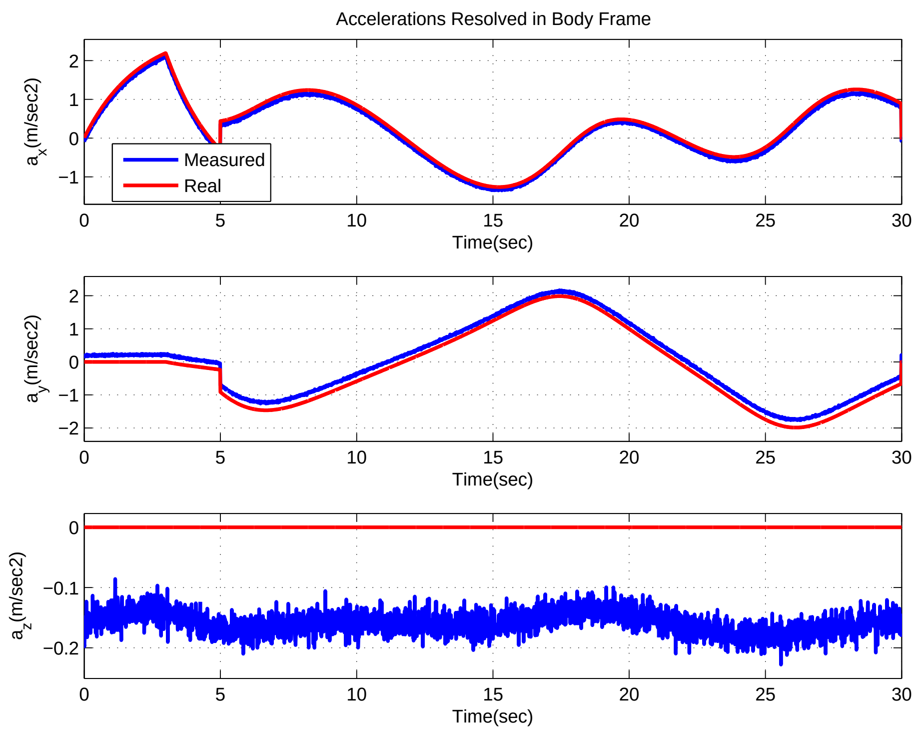


Figure 3.6: The true and measured accelerations in body frame.

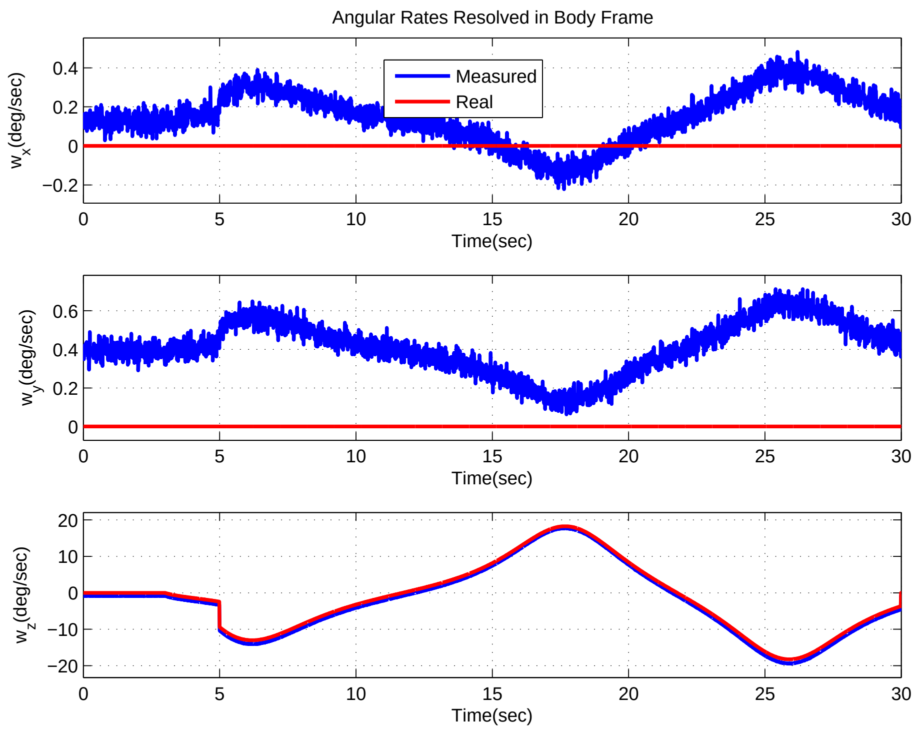


Figure 3.7: The true and measured angular rates in body frame.

The navigation solution is obtained in local navigation frame which is a NED frame. The simulated motion starts from Cartesian position of (0, 0, 0) with respect to local frame, and draws a line for a while, then exhibits a S curve as demonstrated in Figure 3.8. Throughout the motion, the altitude and accordingly z position is fixed at zero as represented by Figure 3.9 since the vessel is expected to not drift apart from the surface of the water. Furthermore, the only angular motion appears in yaw axis where roll and pitch axes are constant at 0° as in Figure 3.11. Once the details of the motion are revealed, the sensor measurements are produced considering the above mentioned error models. The corresponding GNSS position and velocity solutions are given in Figures 3.12 and 3.13.

Due to the fact that there is no established feedback loop in Arc_1, an independent navigation solution of INS given by Figures 3.14, 3.15 and 3.16 emerges. The estimates and real values of the errors in this solution are illustrated by Figure 3.17, 3.18 and 3.19. Taking those errors into consideration, Arc_1 forms the corrected navigation solution as in Figures 3.20, 3.21 and 3.22. The errors in the corrected navigation solution are showed in Figures 3.23, 3.24 and 3.25. The estimated accelerometer and gyroscope biases by the error-state Kalman filter are represented by Figures 3.26 and 3.27, respectively.

The navigation solution of Arc_2 and the errors in this solution that cannot be compensated by the algorithm are depicted by Figures 3.28, 3.29, 3.30, 3.31, 3.32 and 3.33.

The navigation solution of Arc_3 and the errors in this solution that cannot be compensated by the algorithm are presented by Figures 3.34, 3.35, 3.36, 3.37, 3.38 and 3.39.

It is observed that in Arc_1, the solution of INS drifts with time due to the biases and random noises of inertial sensors. However, it is able to track the true values by estimating this error of INS. The estimated accelerometer bias values in three axes converge to $(-0.10, 0.20, -0.17)\, m/sec^2$ in order. Expected values of the true biases are calculated to be equal to $(-0.086, 0.19, -0.17)\, m/sec^2$. Consequently, these are estimated quite effectively. On the other hand, the estimation values of gyroscope biases do not converge to a significant point because of the absence of an external attitude or angular rate measurement in this structure. Furthermore, the same deficiency results in the divergence of Euler angles as there is no source of information to correct gyroscope errors.

Arc_2 exhibits quite improved error performance compared to Arc_1 without using any additional sensor. Nevertheless, it is also not capable of tracking the true value of Euler angles.

The main difference between the performances of Arc_3 and Arc_2 is that the errors in the Euler angle estimates of the former one are about zero because it makes use of angle estimates obtained from magnetometer measurements.

3.40 illustrates the trajectories obtained by three different integrated navigation architectures. Moreover, the RMS errors in position, velocity and attitude of the methods are described in Table 3.1, 3.2 and 3.3.

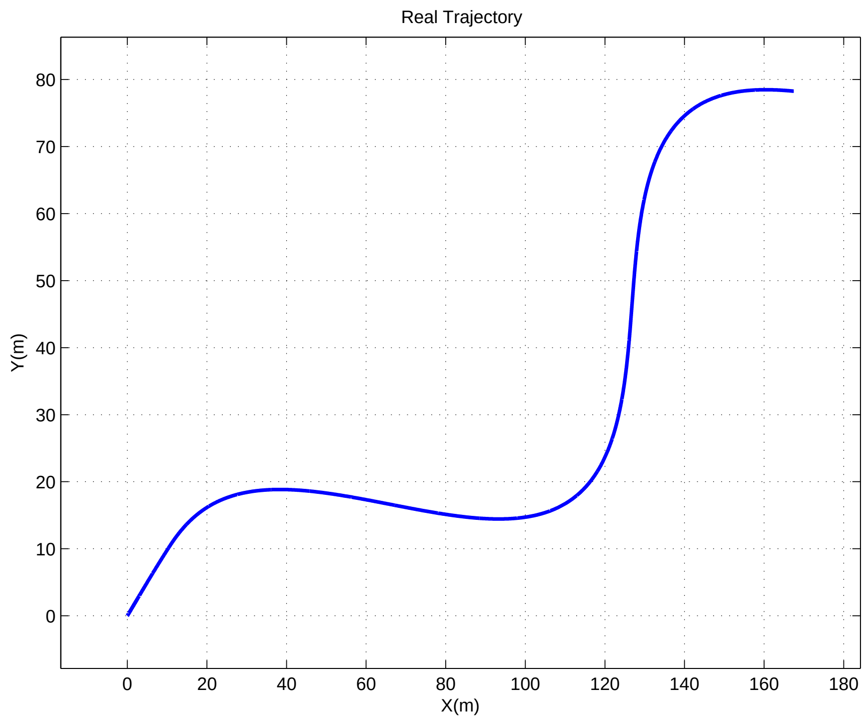


Figure 3.8: Simulated motion in X-Y coordinates of local navigation frame.

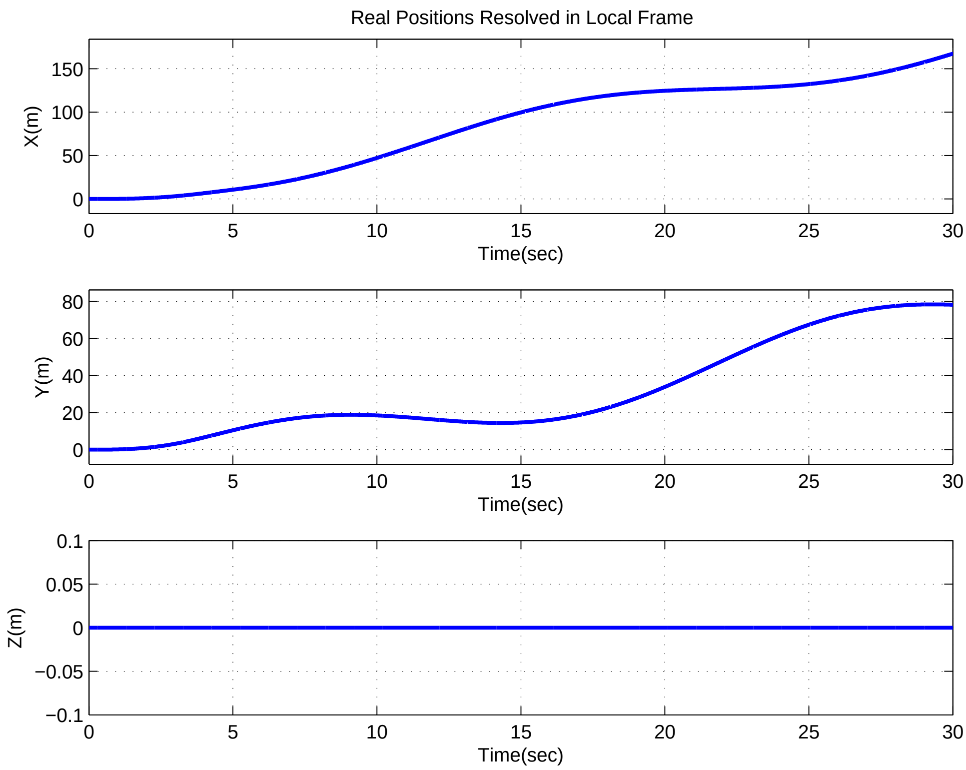


Figure 3.9: The true Cartesian positions in local navigation frame.

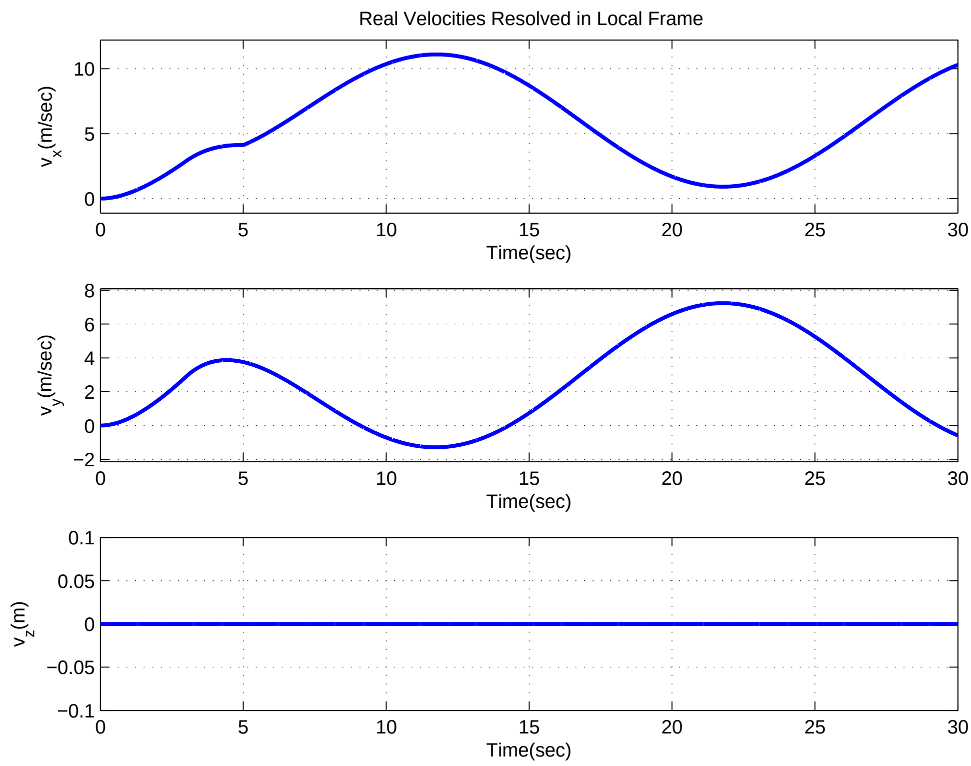


Figure 3.10: The true linear velocities in local navigation frame.

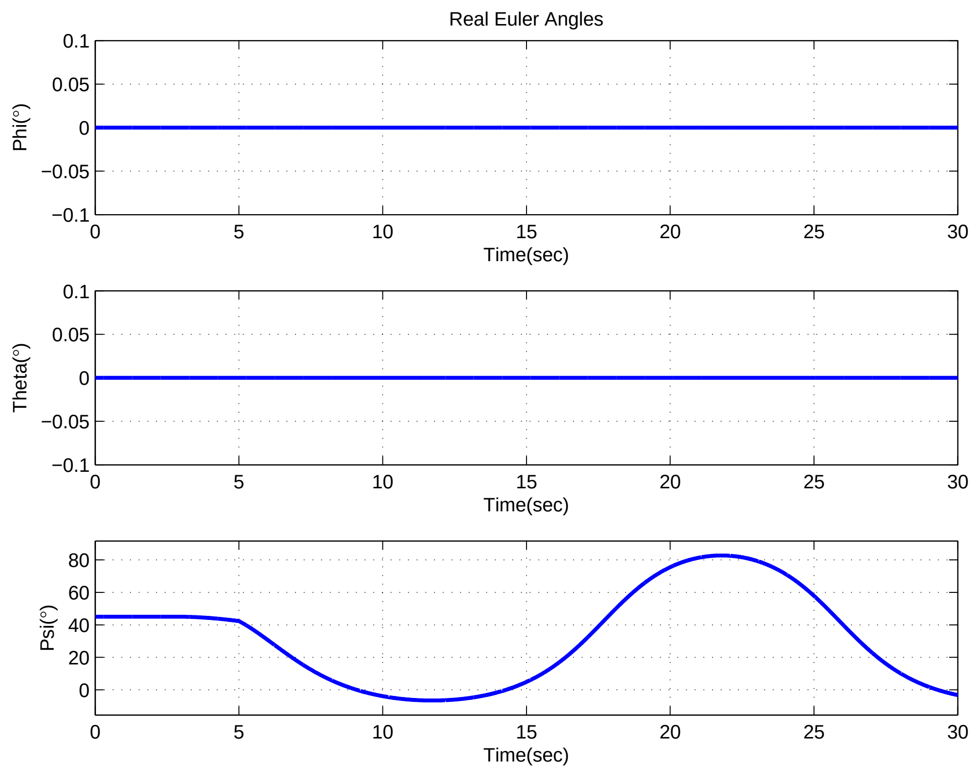


Figure 3.11: The true attitude of the body with respect to local navigation frame.

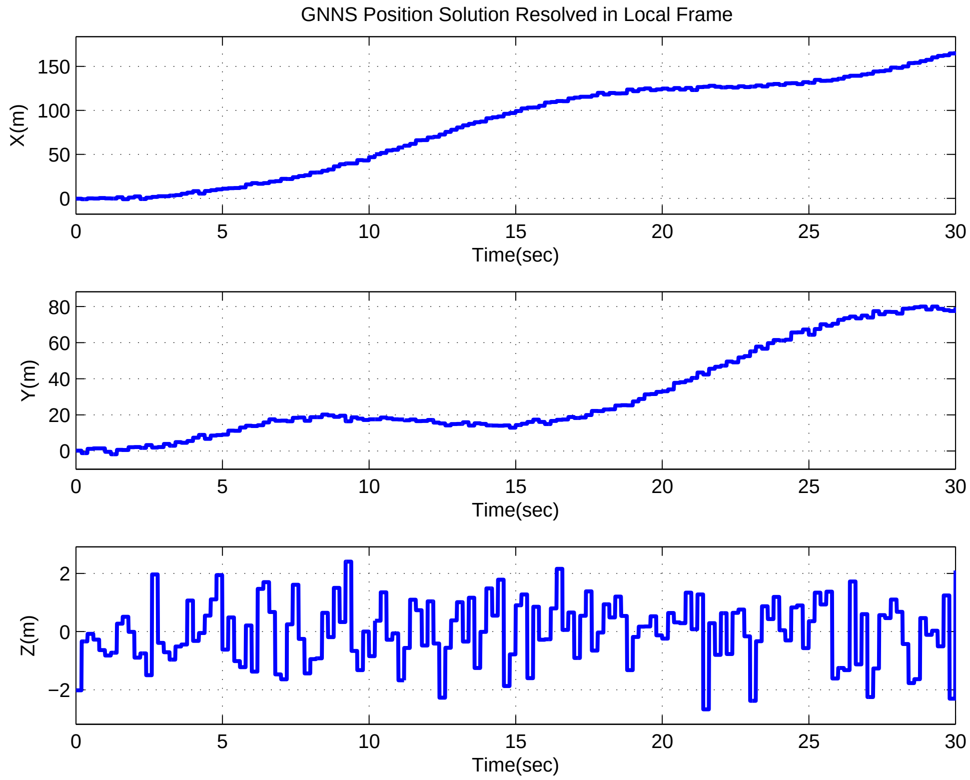


Figure 3.12: GNSS indicated position of the body local navigation frame.

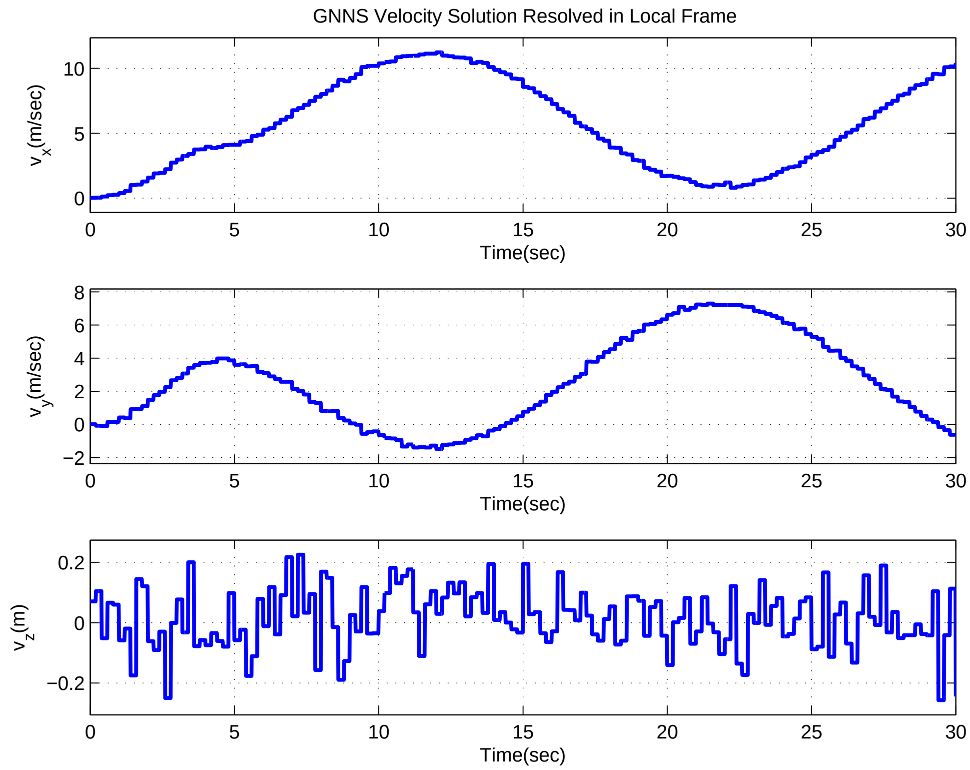


Figure 3.13: GNSS indicated linear velocity of the body local navigation frame.

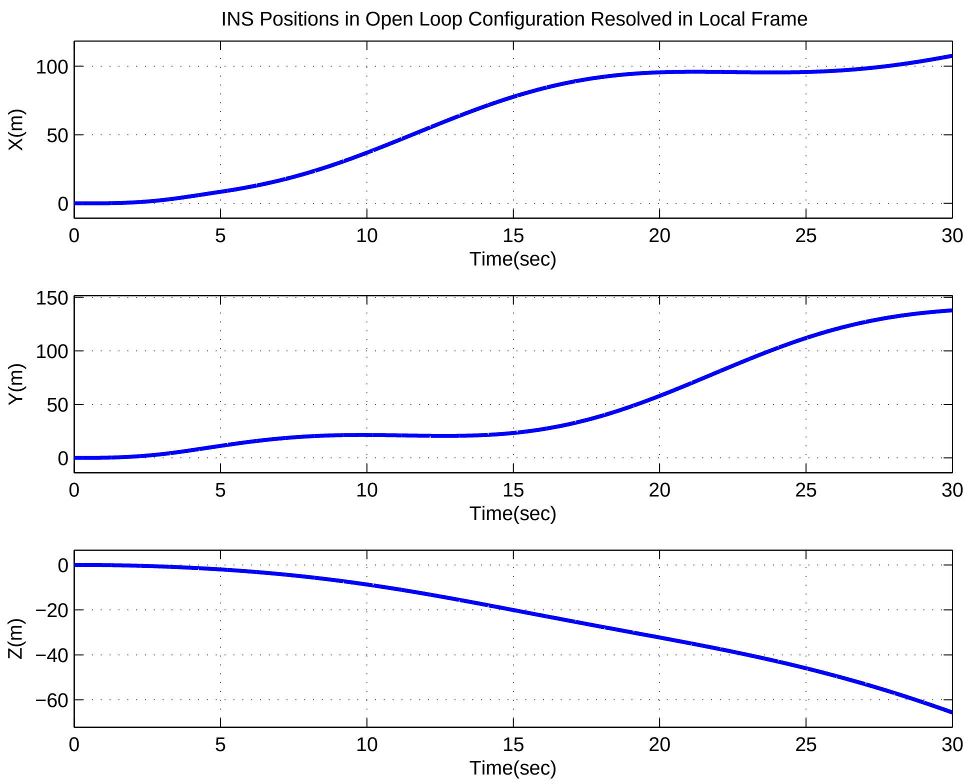


Figure 3.14: The open loop INS indicated position in local navigation frame.

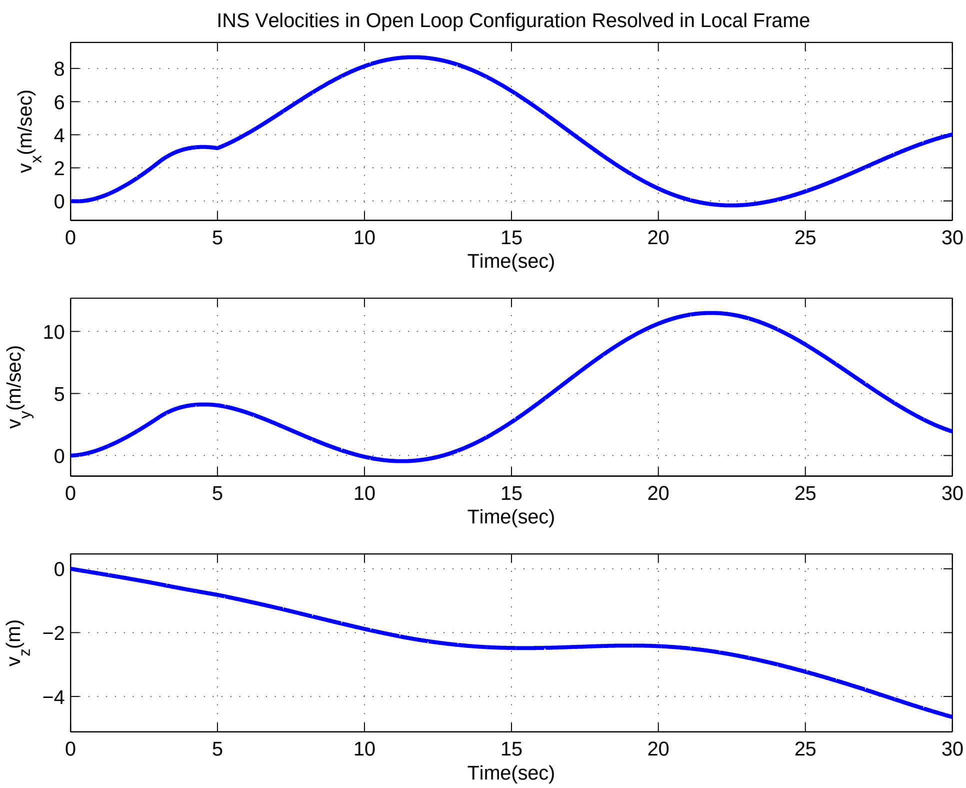


Figure 3.15: The open loop INS indicated velocity in local navigation frame.

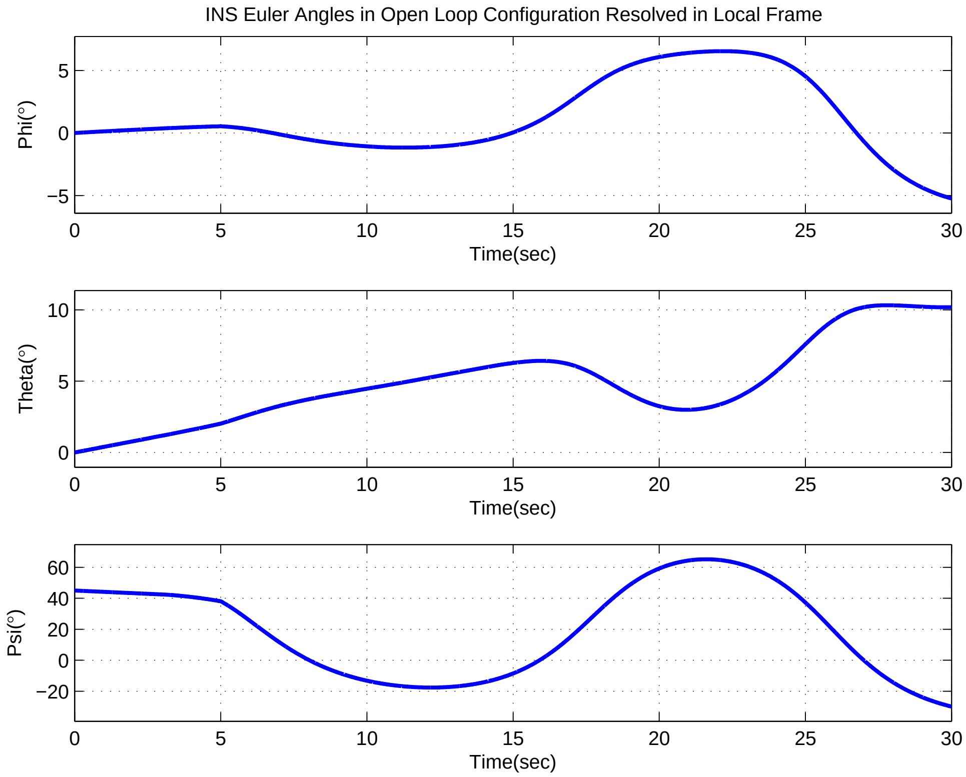


Figure 3.16: The open loop INS indicated attitude w.r.t. local navigation frame.

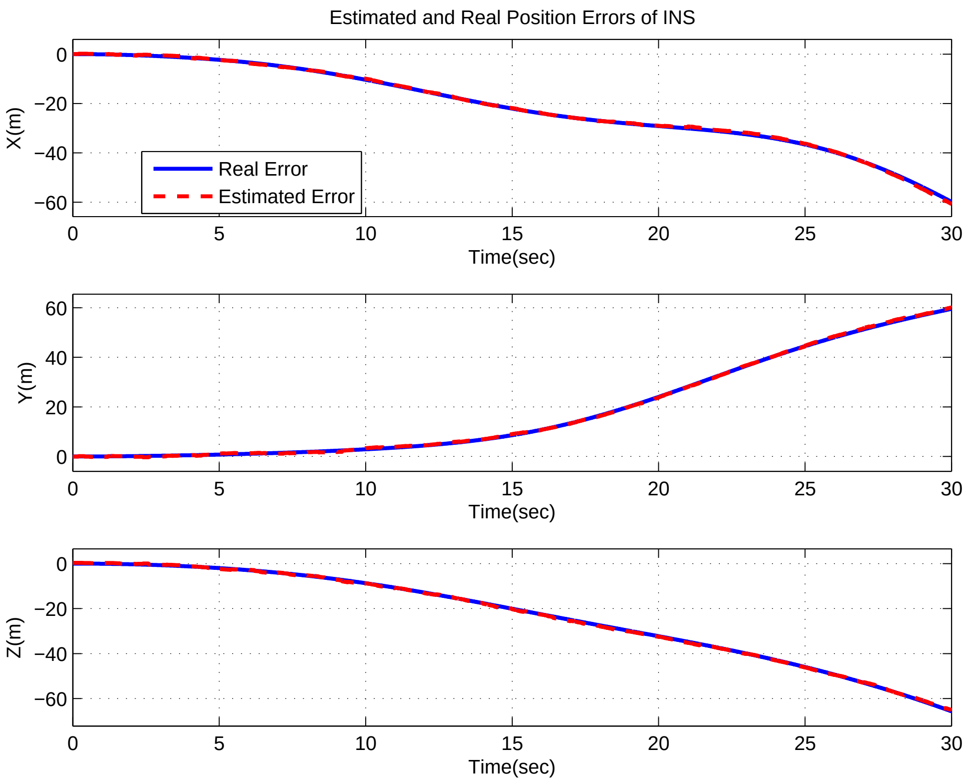


Figure 3.17: The real and estimated position errors of INS in Arc_1.

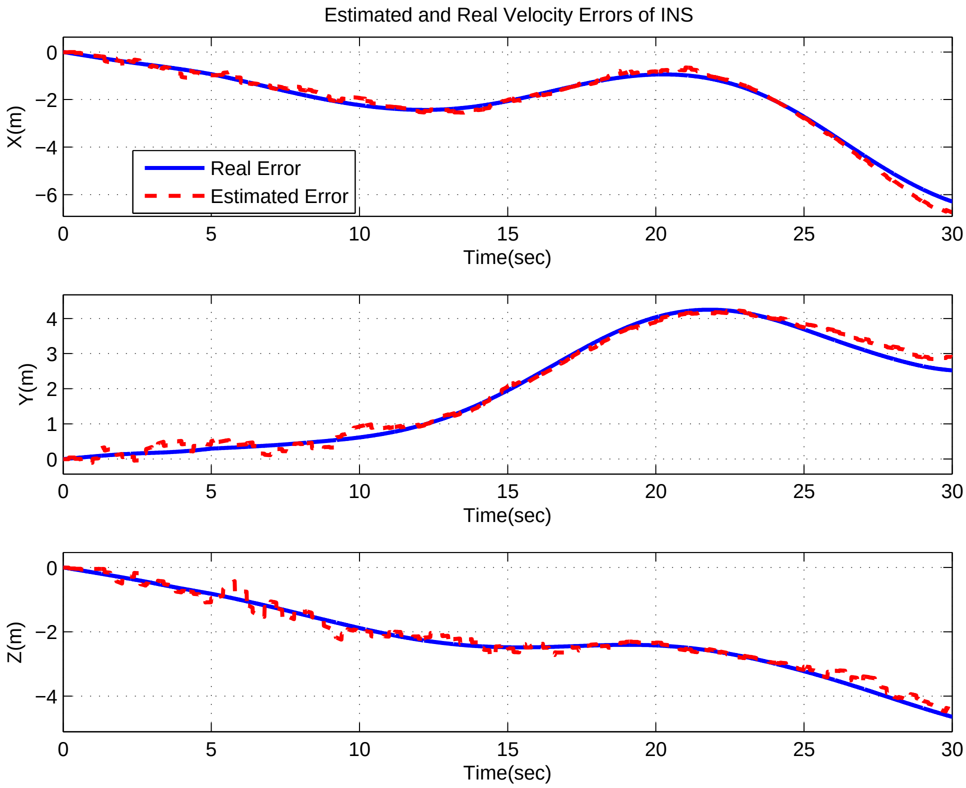


Figure 3.18: The real and estimated velocity errors of INS in Arc_1.

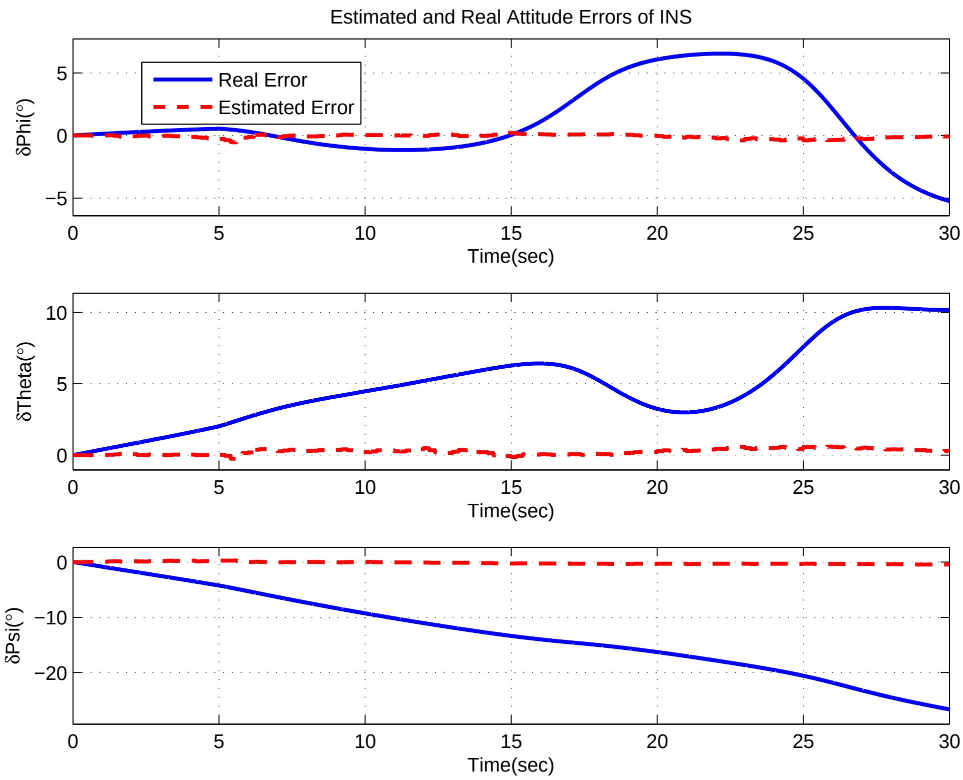


Figure 3.19: The real and estimated attitude errors of INS in Arc_1.

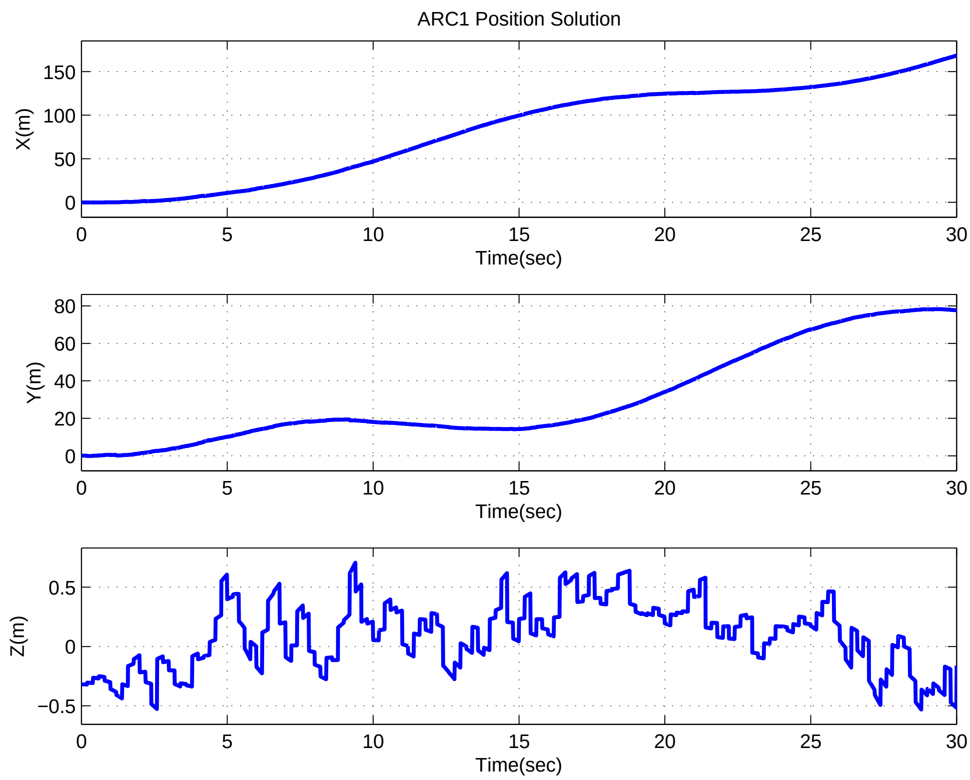


Figure 3.20: Position solution in local navigation frame by Arc_1.

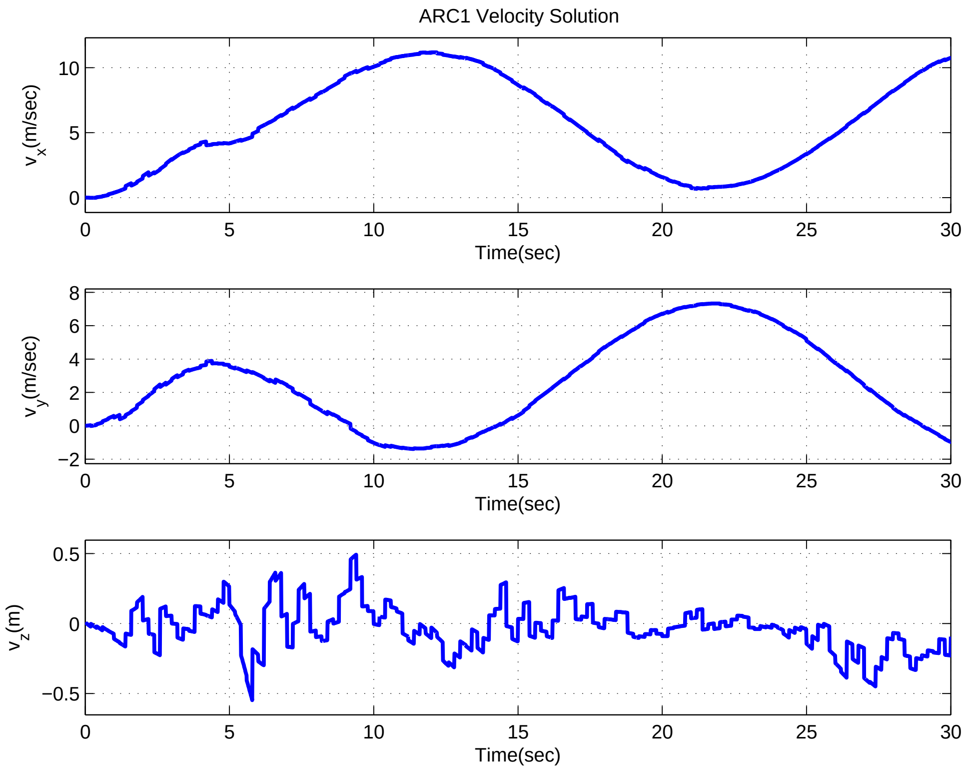


Figure 3.21: Velocity solution in local navigation frame by Arc_1.

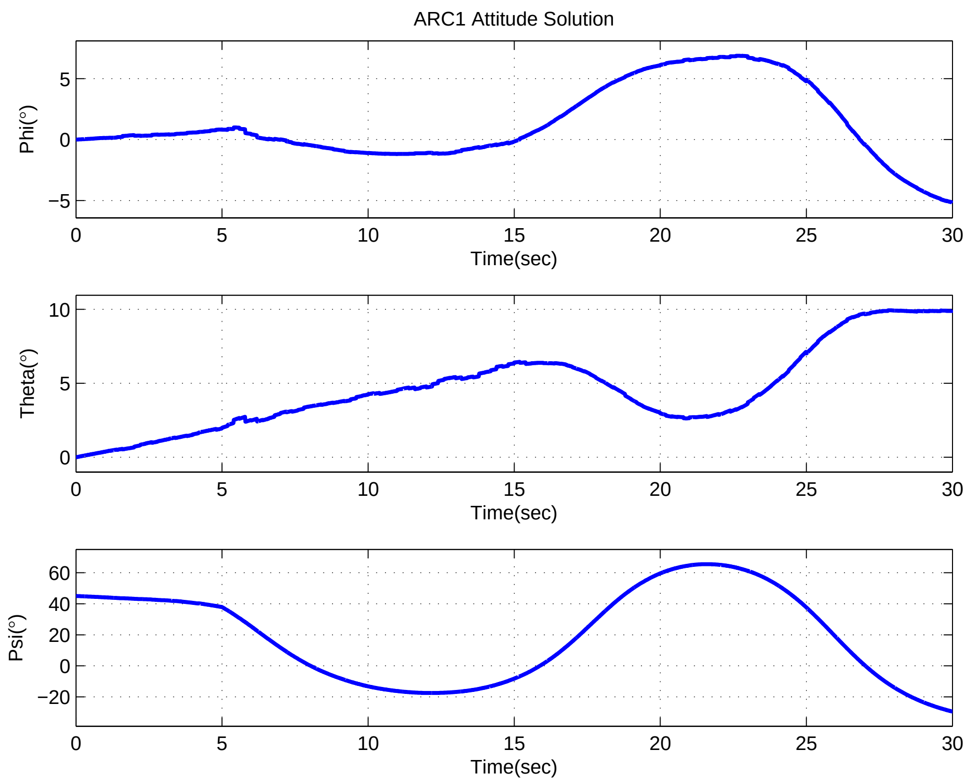


Figure 3.22: Attitude solution w.r.t. local navigation frame by Arc_1.

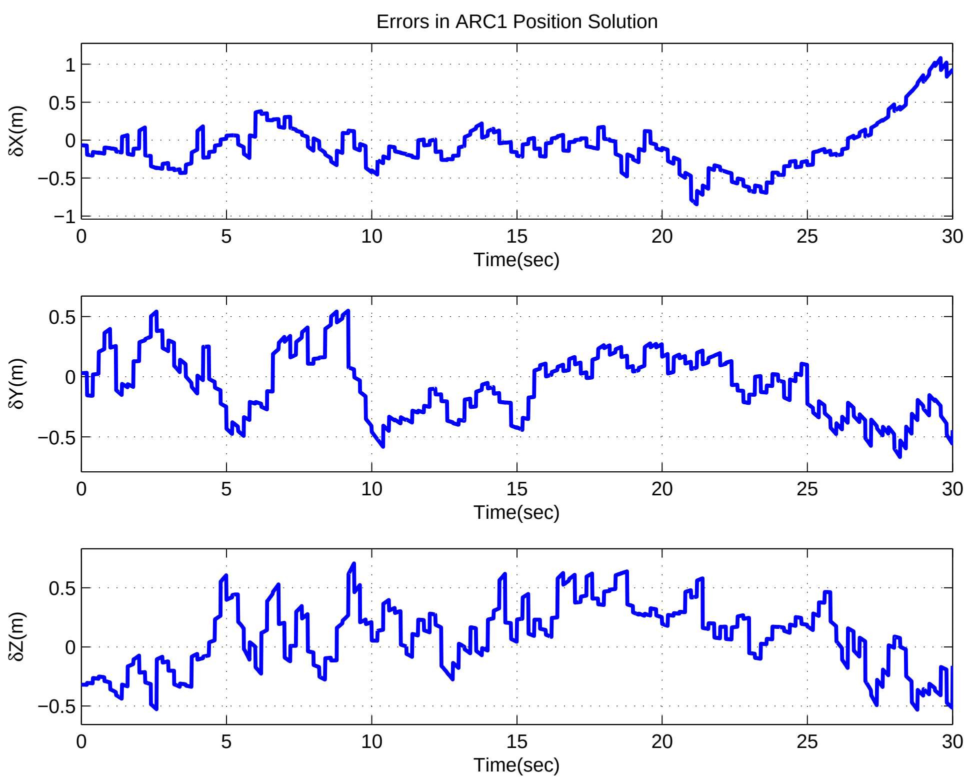


Figure 3.23: Errors in position solution of Arc_1 in local navigation frame.

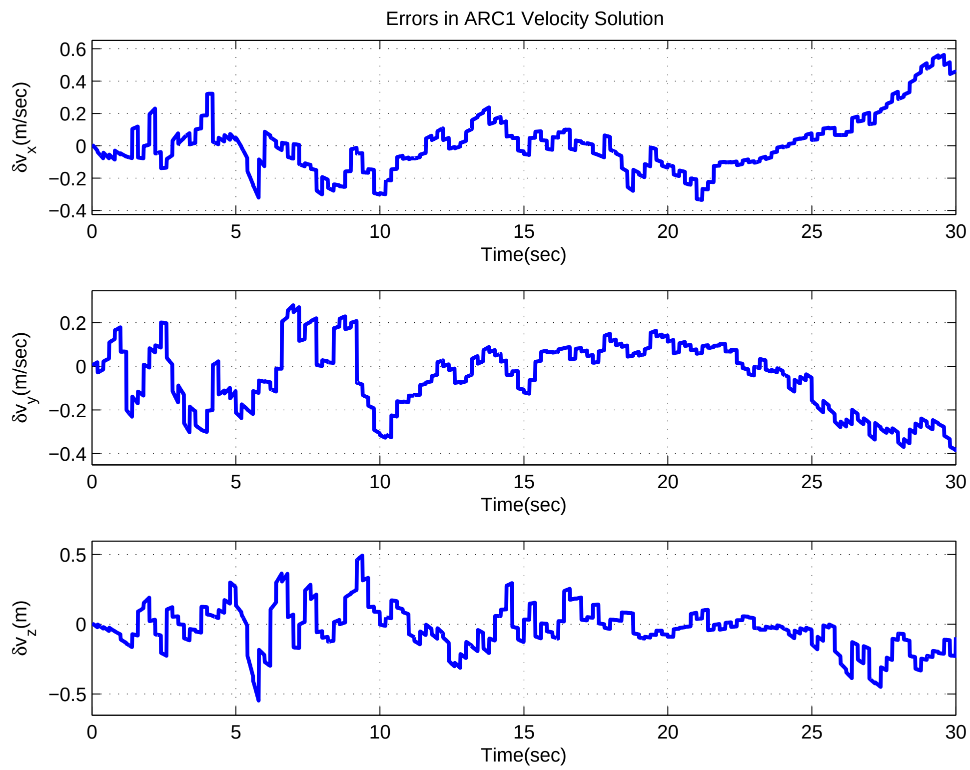


Figure 3.24: Errors in velocity solution of Arc_1 in local navigation frame.

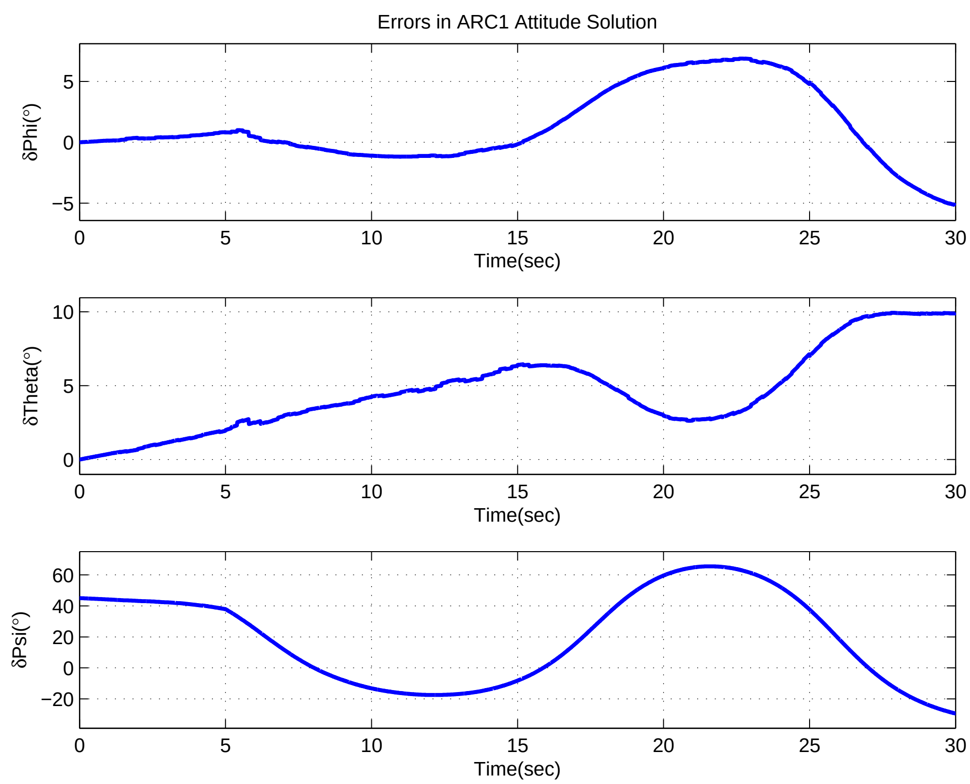


Figure 3.25: Errors in attitude solution of Arc_1 w.r.t. local navigation frame.

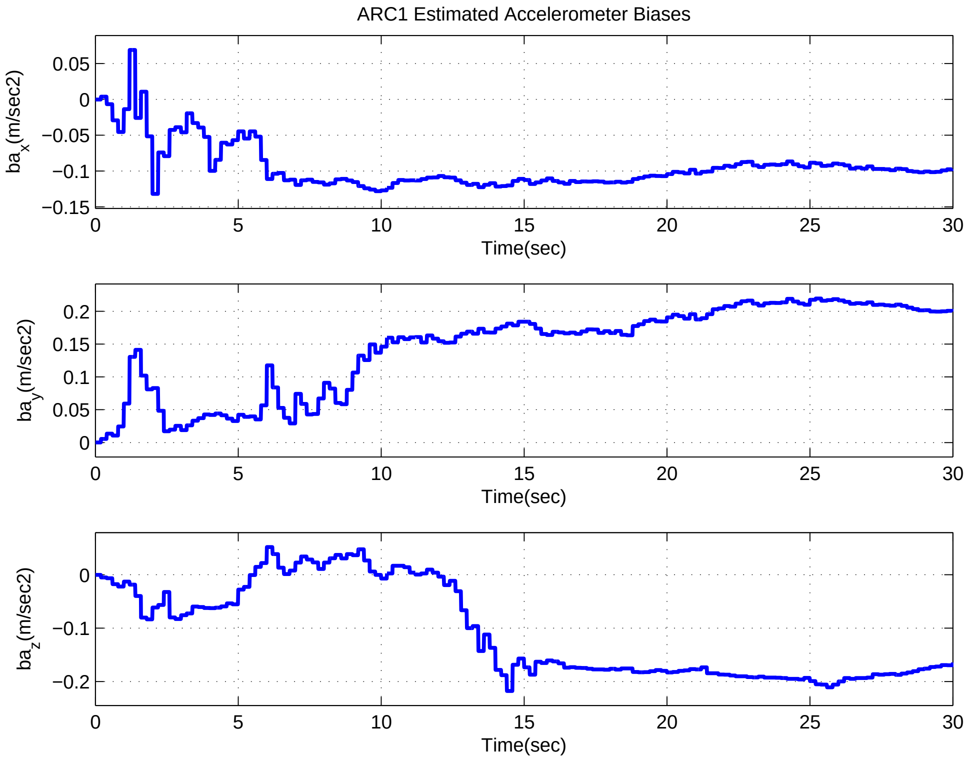


Figure 3.26: Estimated accelerometer biases by Arc_1.

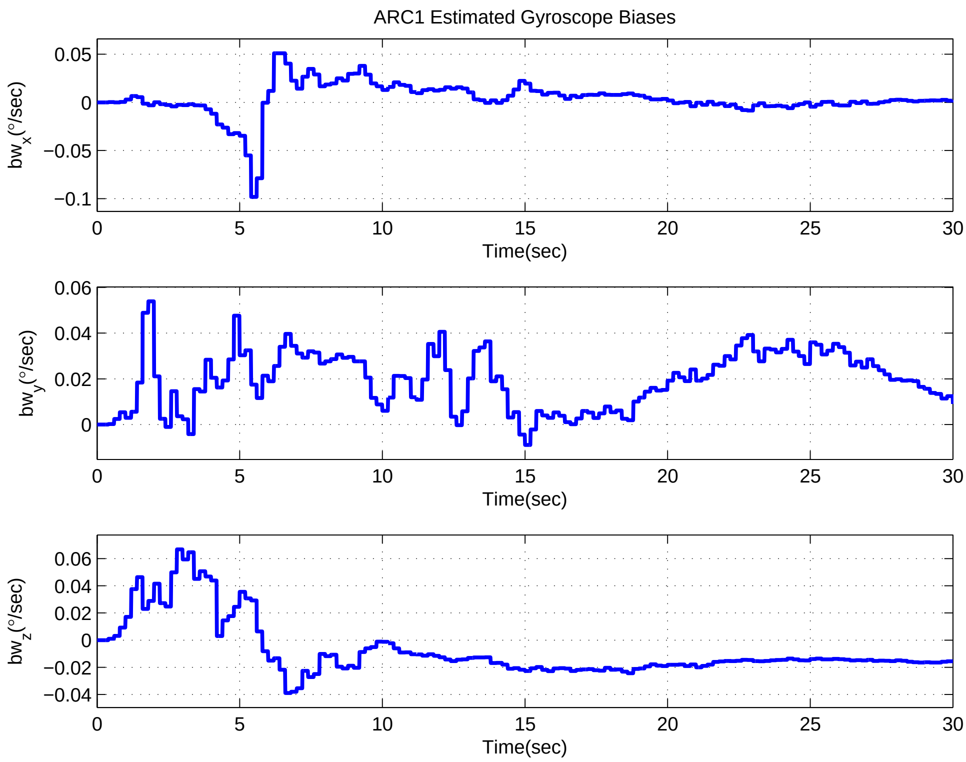


Figure 3.27: Estimated gyroscope biases by Arc_1.

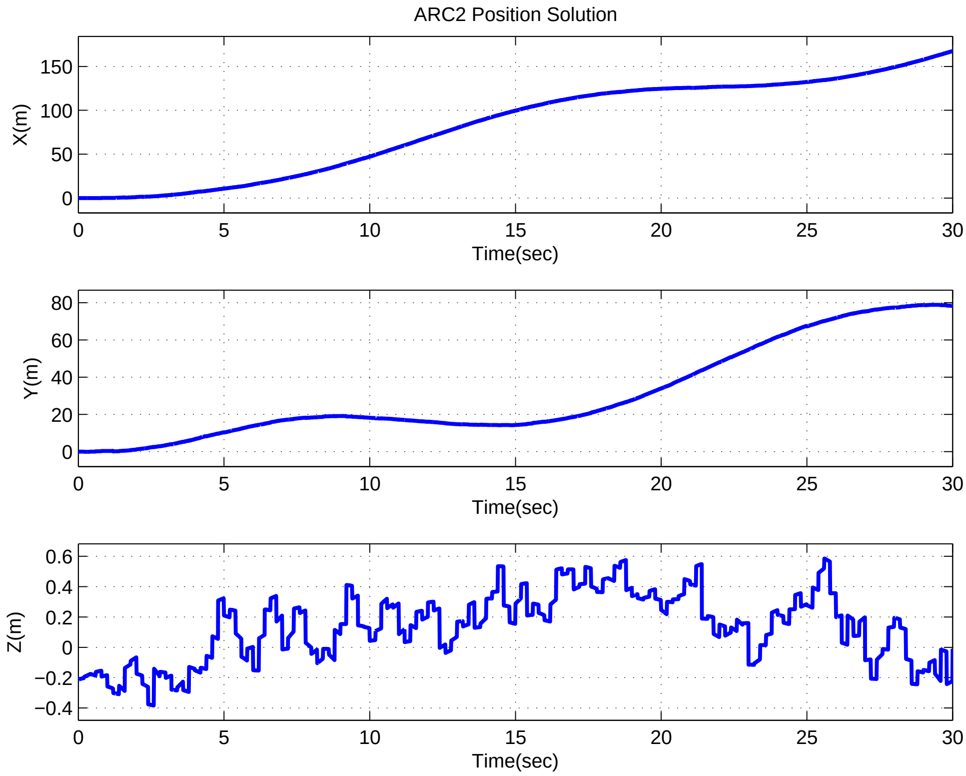


Figure 3.28: Position solution in local navigation frame by Arc_2.

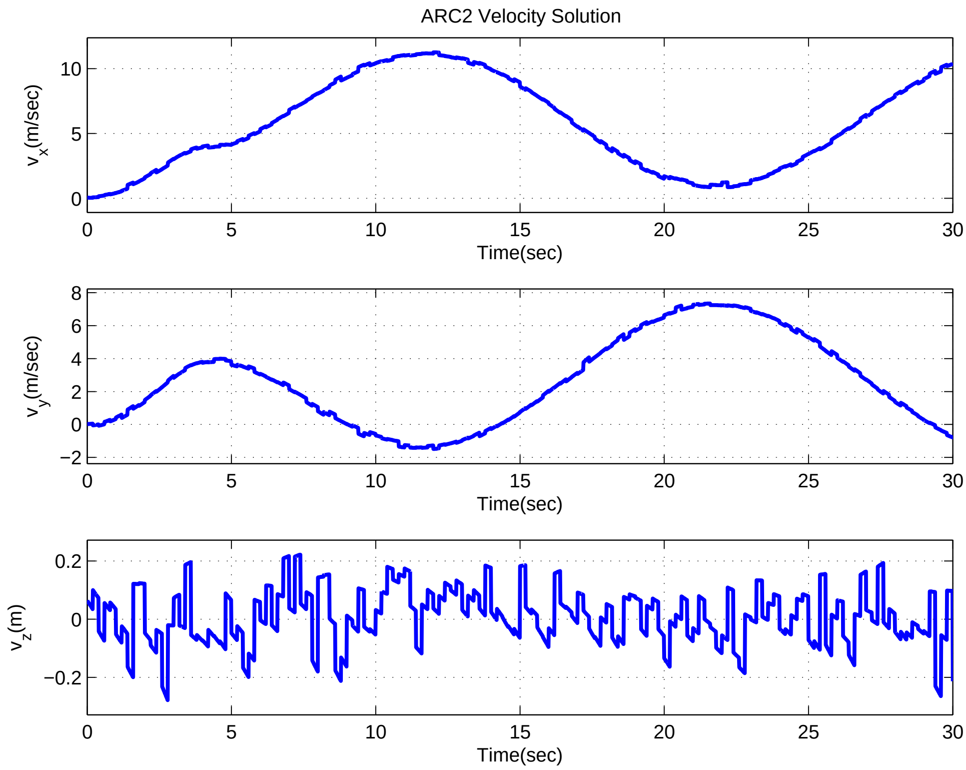


Figure 3.29: Velocity solution in local navigation frame by Arc_2.

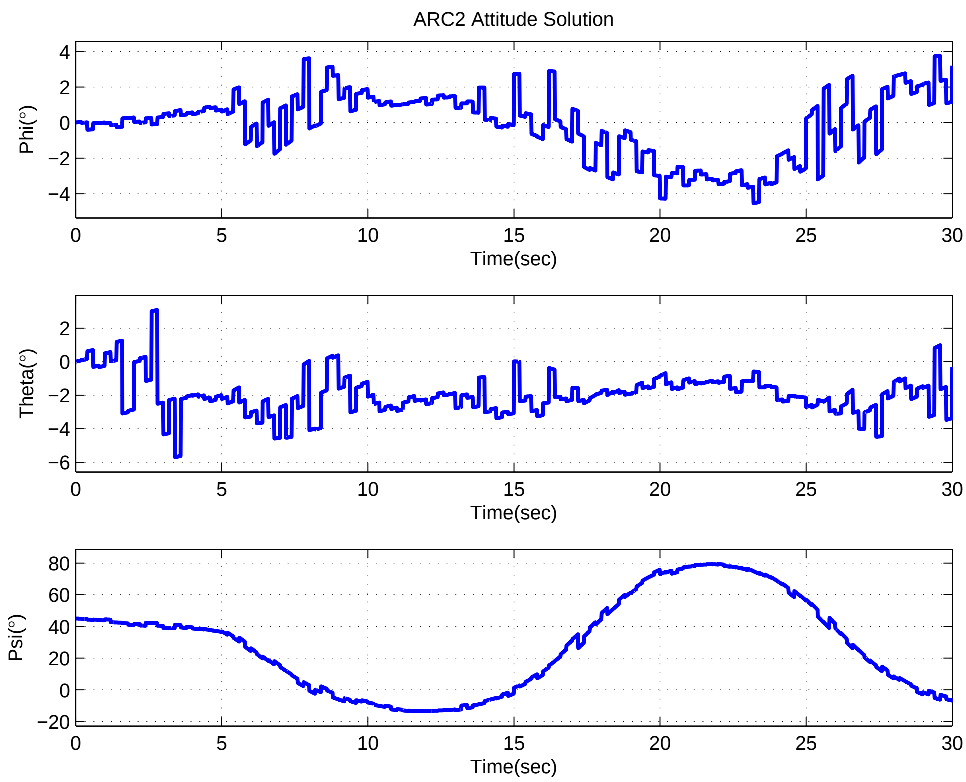


Figure 3.30: Attitude solution w.r.t. local navigation frame by Arc_2.

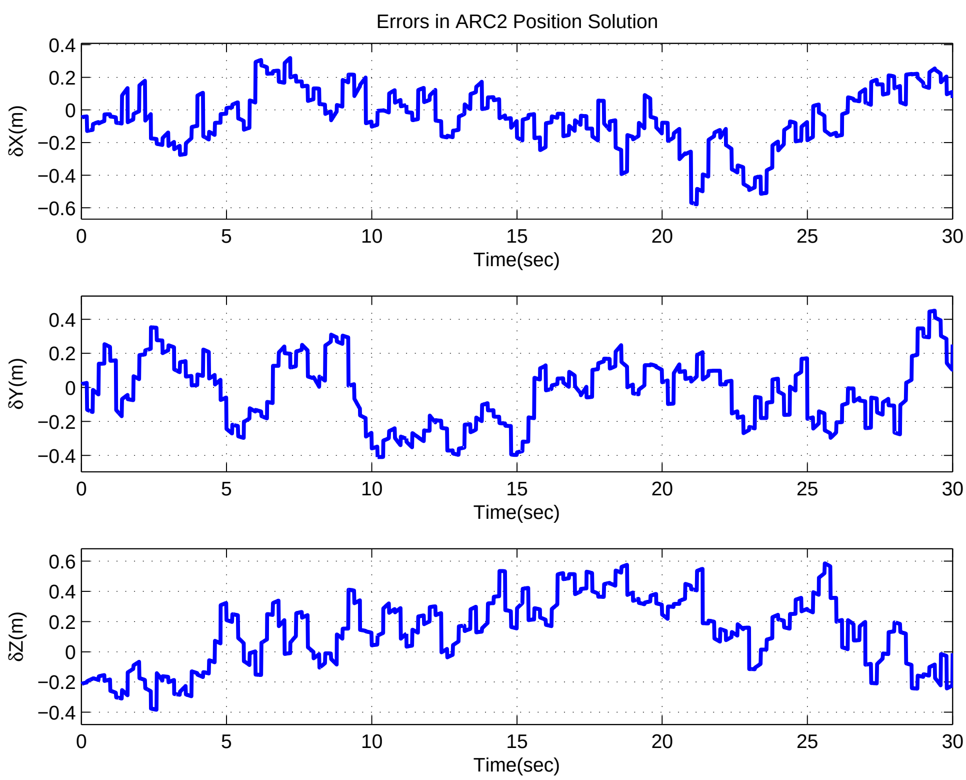


Figure 3.31: Errors in position solution of Arc_2 in local navigation frame.

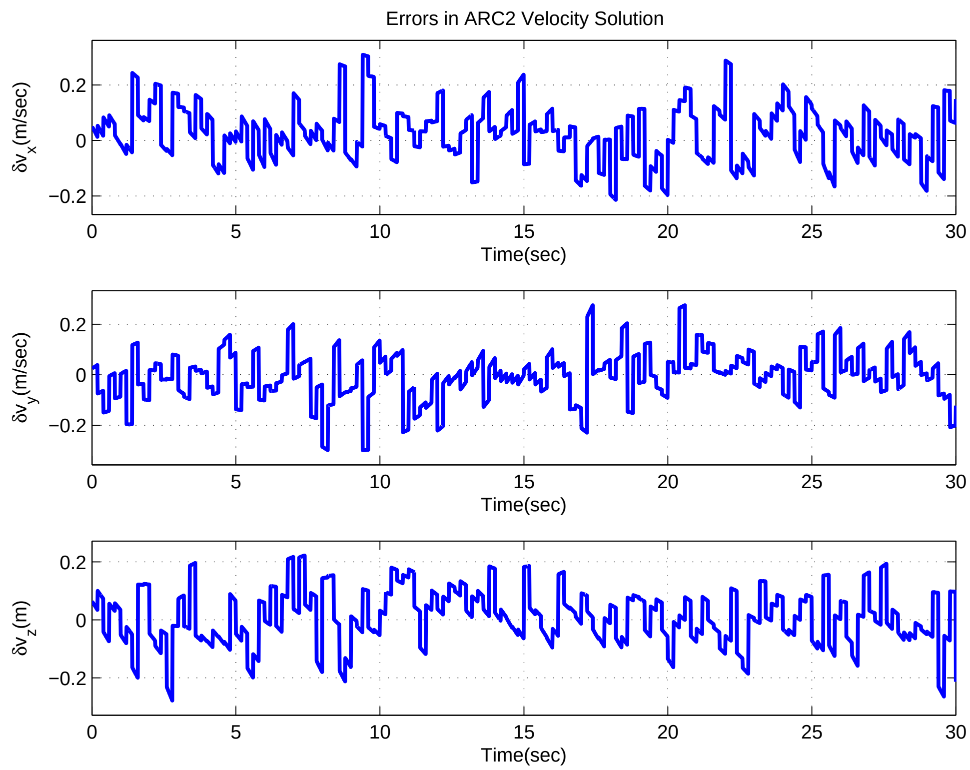


Figure 3.32: Errors in velocity solution of Arc_2 in local navigation frame.

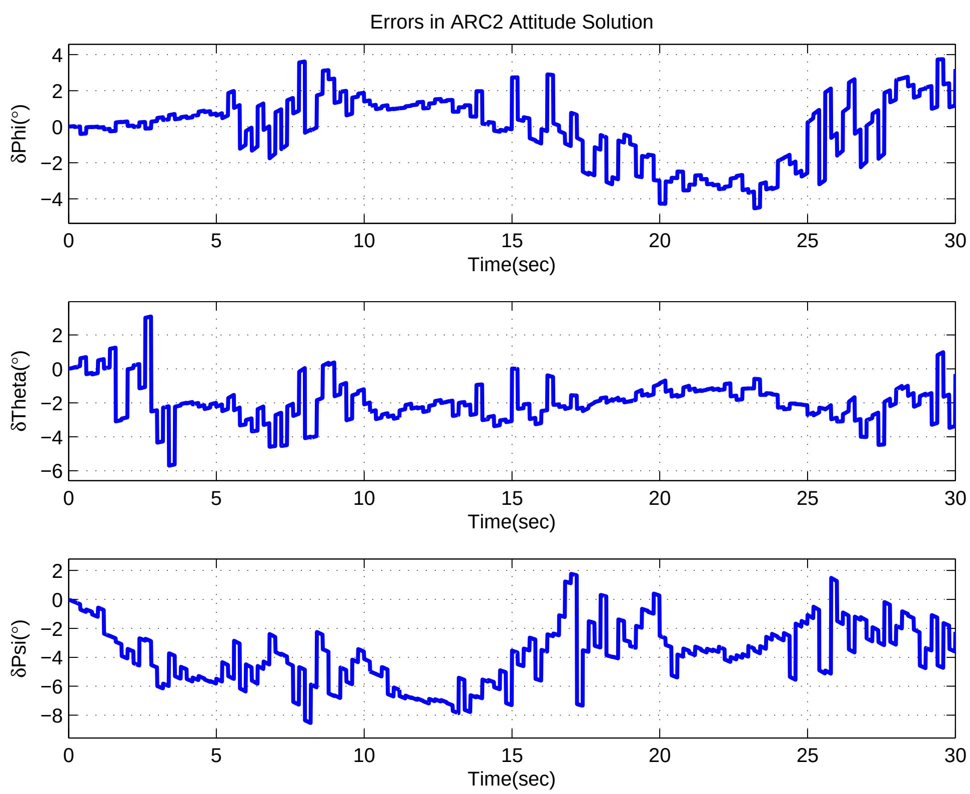


Figure 3.33: Errors in attitude solution of Arc_2 w.r.t. local navigation frame.

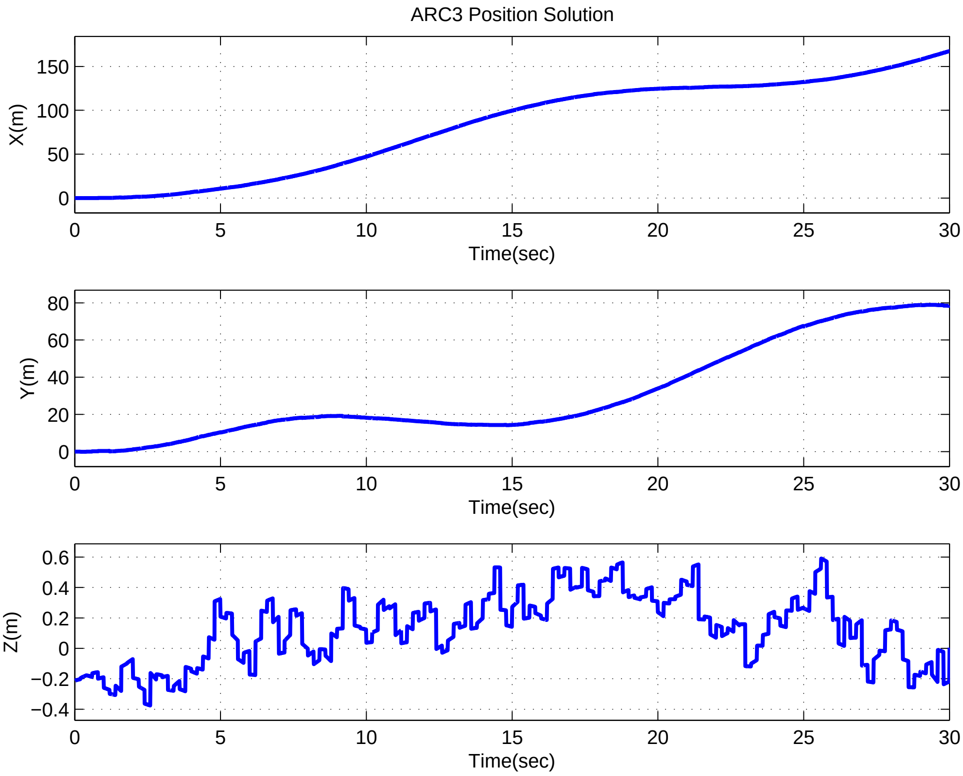


Figure 3.34: Position solution in local navigation frame by Arc_3.

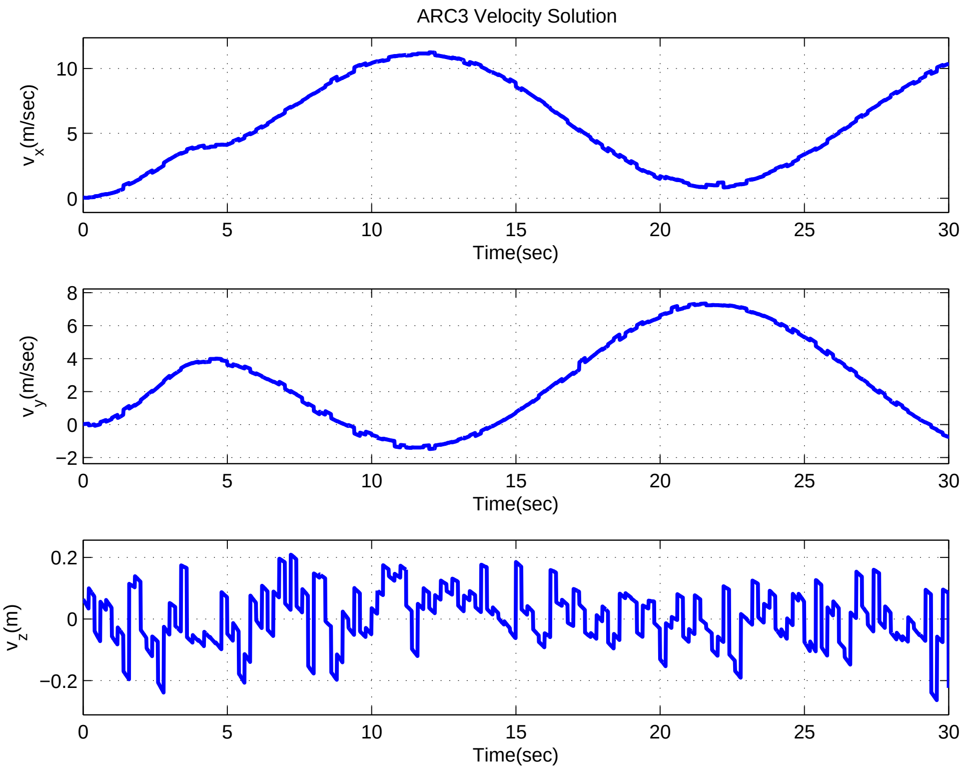


Figure 3.35: Velocity solution in local navigation frame by Arc_3.

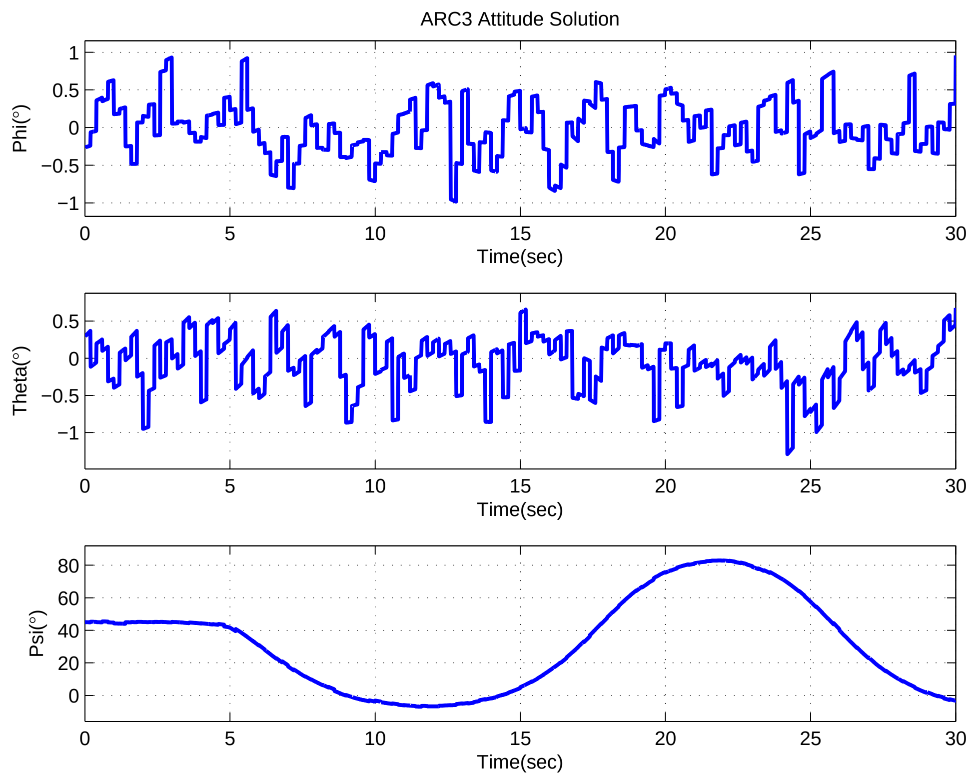


Figure 3.36: Attitude solution w.r.t. local navigation frame by Arc_3.

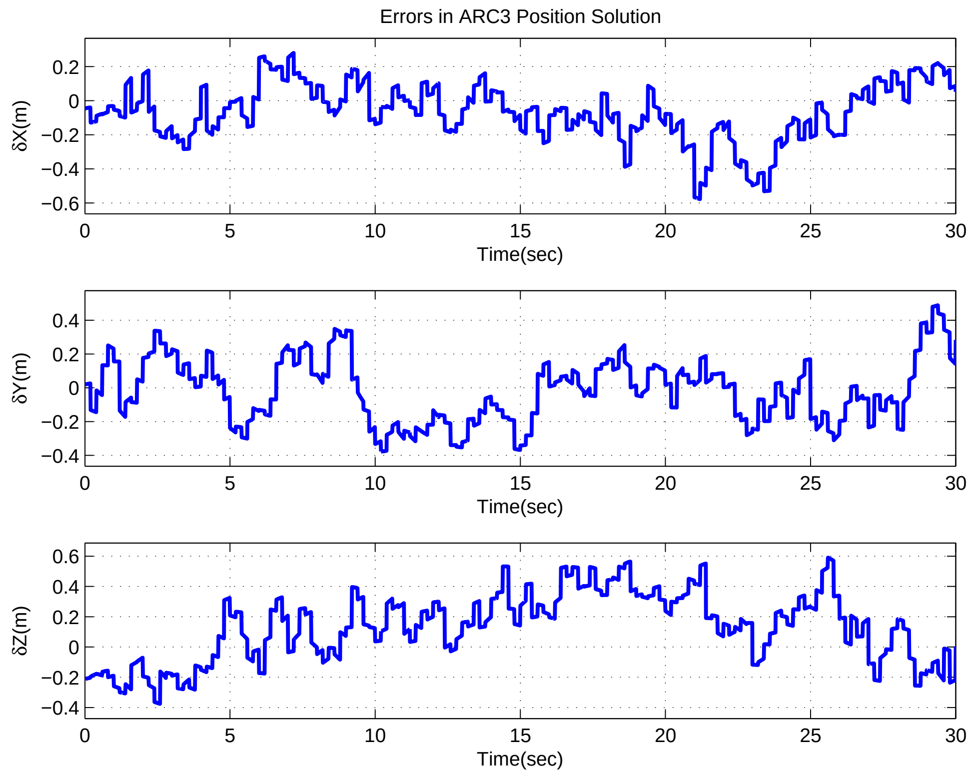


Figure 3.37: Errors in position solution of Arc_3 in local navigation frame.

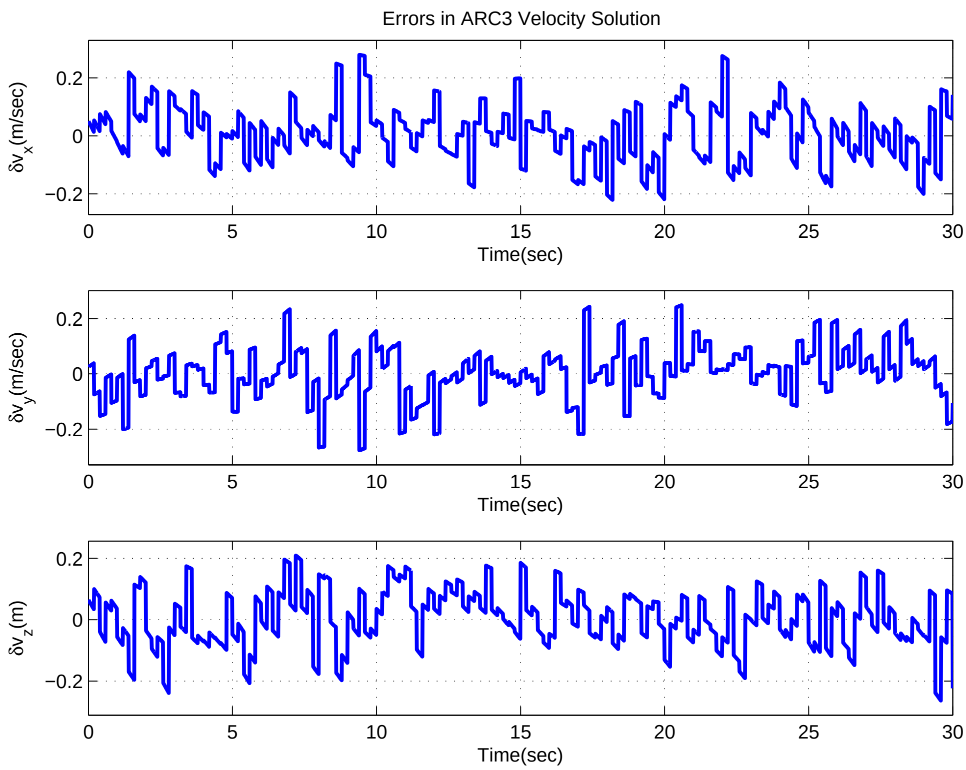


Figure 3.38: Errors in velocity solution of Arc_3 in local navigation frame.

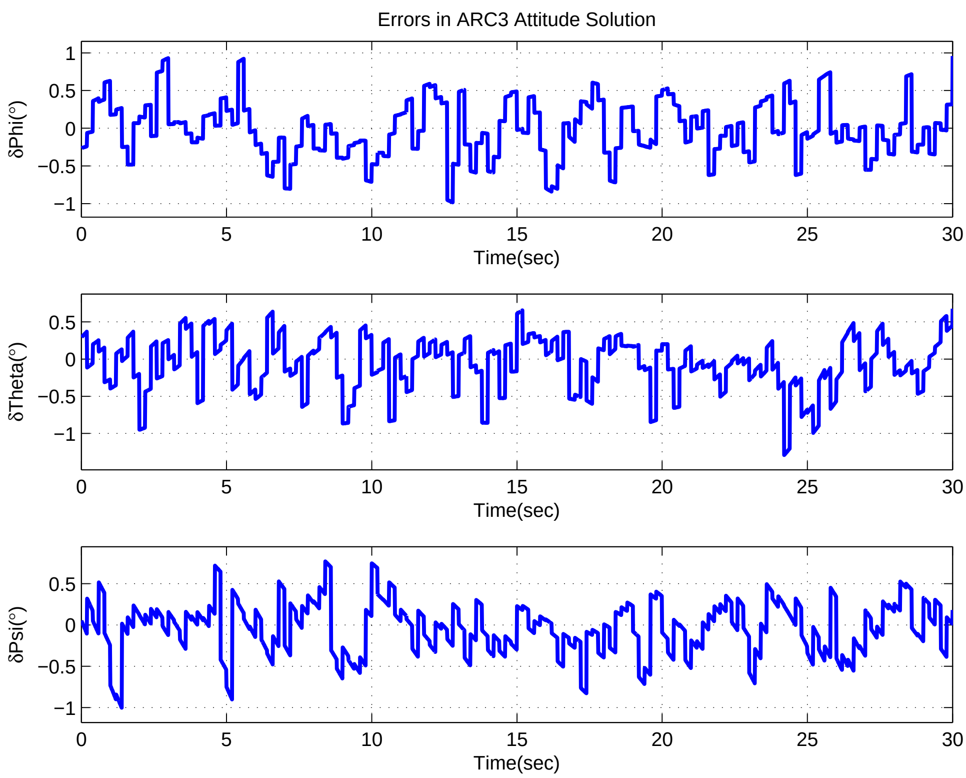


Figure 3.39: Errors in attitude solution of Arc_3 w.r.t. local navigation frame.

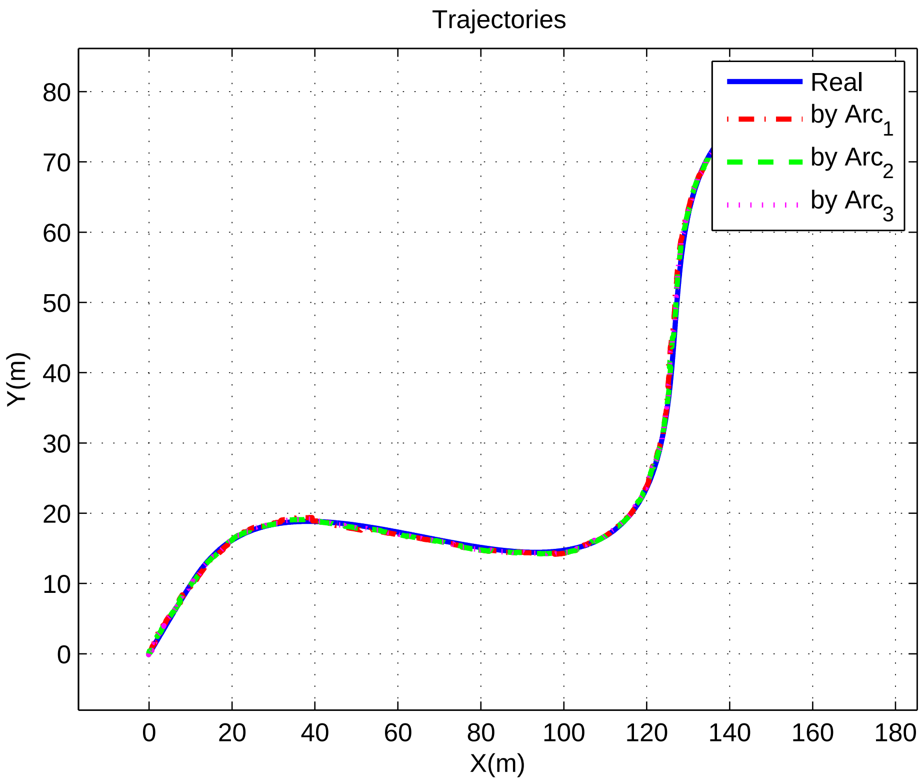


Figure 3.40: Comparison of the trajectories obtained by three integration architectures.

Table 3.1: RMS position errors in local navigation frame.

RMS Position Error	*X*	*Y*	*Z*
Arc_1	0.334	0.274	0.297
Arc_2	0.181	0.195	0.262
Arc_3	0.179	0.190	0.260

Table 3.2: RMS velocity errors in local navigation frame.

RMS Velocity Error	v_x	v_y	v_z
Arc_1	0.176	0.162	0.160
Arc_2	0.100	0.096	0.095
Arc_3	0.096	0.095	0.090

Table 3.3: RMS attitude errors w.r.t. local navigation frame.

RMS Attitude Error	ϕ	θ	ψ
Arc_1	3.36	5.36	14.54
Arc_2	1.84	2.28	4.34
Arc_3	0.37	0.36	0.32

3.8 Experimental Results

The navigation algorithm, previously represented by Arc_3, implementing loosely coupled, closed loop integration of INS, GNSS solutions and attitude estimates is also tested with real measurements collected via Pixhawk autopilot card. It is aimed to exploit the integrated navigation solutions that are obtained by experiments in system identification process of the model boat. For this purpose, numerous tests are performed in an indoor test pool and Yalıncak Lake located in the campus of METU. Three examples will be introduced in this section.

3.8.1 Indoor Tests

To be able to identify the angular damping parameters of the model boat, the experiments are made in an indoor pool which provides controlled test conditions by eliminating environmental disturbances. Throughout experimentation process, the GNSS is not able to connect any satellite vehicle as depicted in Figure 3.41. Thus, the integrated position and velocity solutions are not corrected with measurements and drifted away as given in Figures 3.42 and 3.43. As only angular states are investigated in this case, it does not pose a problem. The attitude solution of the navigation algorithm and the estimated attitude are given in Figure 3.44. In this experiment, the vessel is disturbed on its roll axis on purpose, and the damping characteristic is examined. The noise content of estimates is seemed to be compensated by the navigation algorithm.

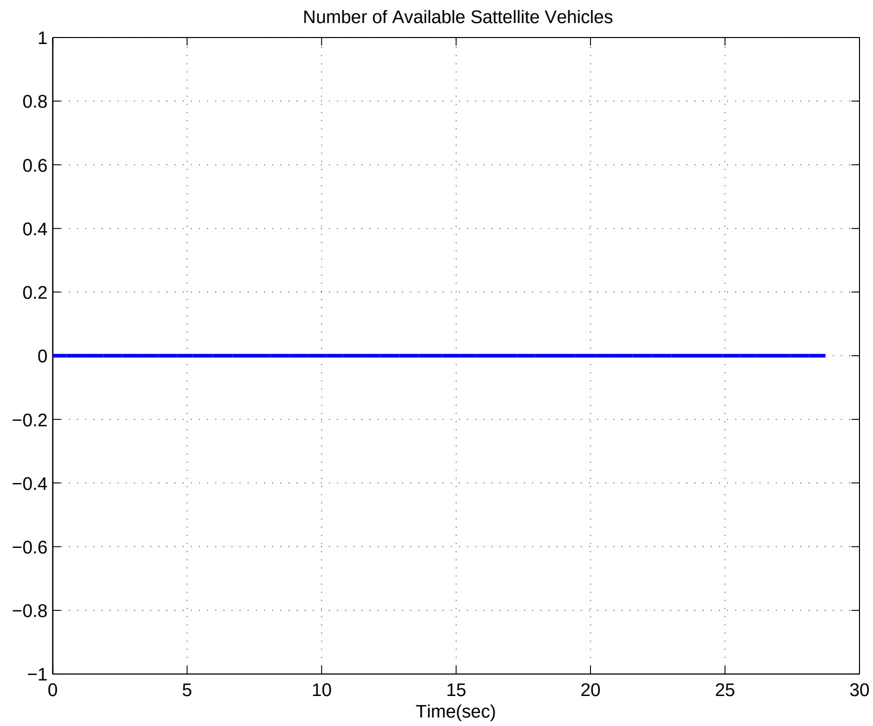


Figure 3.41: The number of available GNSS satellites during the indoor test.

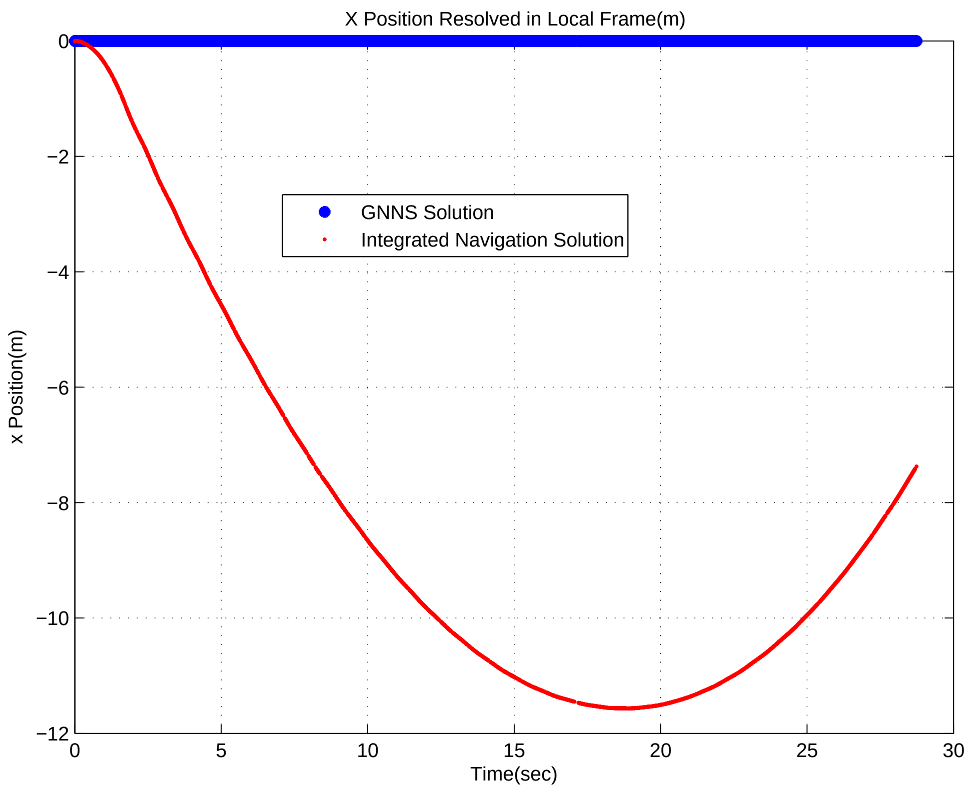


Figure 3.42: X-Position solutions in local navigation frame during indoor test.

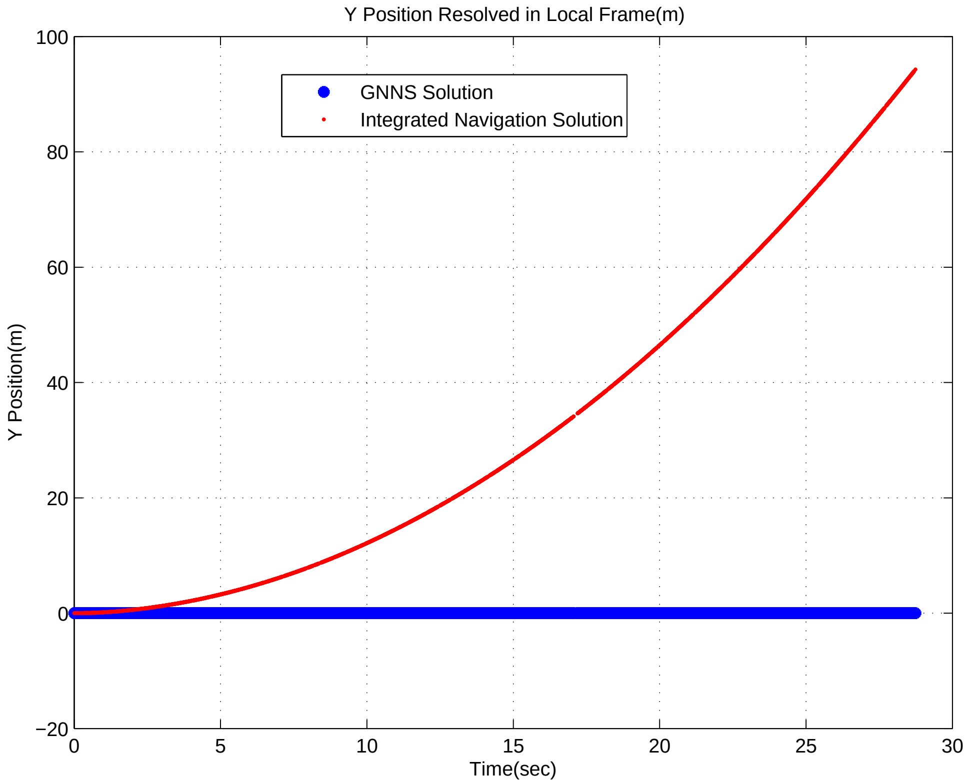


Figure 3.43: Y-Position solutions in local navigation frame during indoor test.

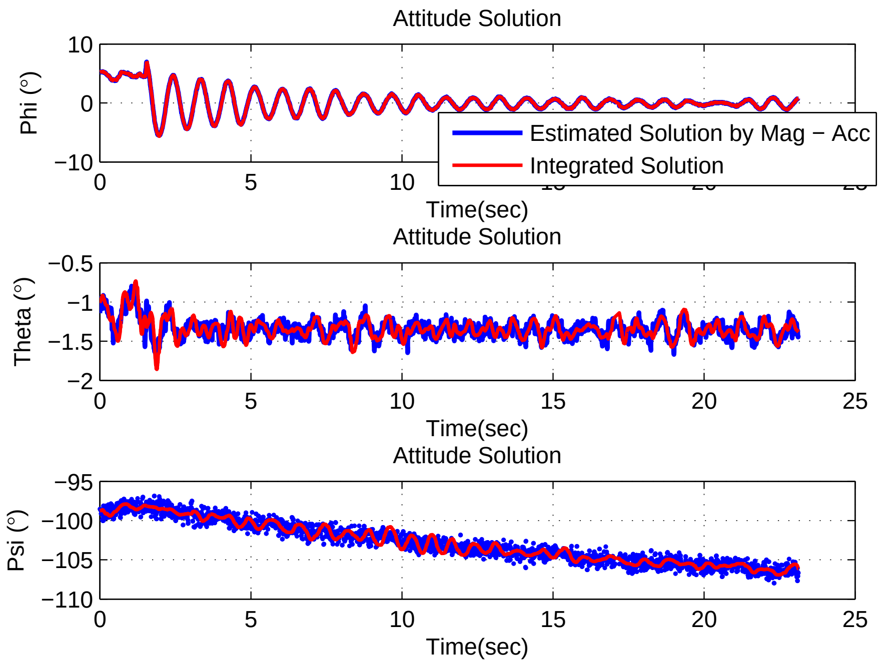


Figure 3.44: Comparison of attitude solution with estimates.

3.8.2 Outdoor Tests

The outdoor experiments are performed in the Yalıncak Lake facilities of METU. To be able to identify all parameters of the vessel as accurate as possible, tests are carried out along various types of trajectories aiming to separate the subparts of the overall mathematical model. All of the trajectories are started from the end of the pier. In this section, one will be informed about two categories of the experiments.

The details of the linear motion test are given in Figures 3.45 to 3.55. The circular motion test is depicted by Figures 3.56 to 3.66. During the tests, there was no available reference for the true values of states. Precisely calibrated optical or acoustic systems could have been employed as models to be compared with the Kalman estimates; however, they were not preferred due to their high costs.

Although the motion in both cases starts from the end of the pier, an offset is detected as seen in Figures 3.45 and 3.56. This is caused by the GNSS position solution which is the only source of position information in integration algorithm. Nonetheless, this does not result in any discrepancy in the results as long as the origin of the local frame is fixed to that shifted starting point.

The GNSS position solutions are observed to be nonuniform in time as illustrated in Figures 3.50, 3.52, 3.61 and 3.63. This is resulted from the inefficiency of the data logger application employed in the autopilot card although all of the functions are already assured to be executed periodically in real time.

The number of the satellites seen by the GNSS antenna is determined to be 9 for all the time as given in Figures 3.48 and 3.59. Thus, the system does not enter an ambiguity state during experimentation. This also clarifies the low noise levels on position solution of GNSS as detailed in 3.46 and 3.57.

The integration of inertial sensors improves the update rates of navigation solution and fills the gaps between successive GNSS data as shown in Figures 3.47, 3.50, 3.52, 3.58, 3.61 and 3.63. Additionally, combination of inertial sensors, GNSS solution and attitude estimates provides a navigation solution with sufficient bandwidth for possible control applications [39].

Linear Motion Test:

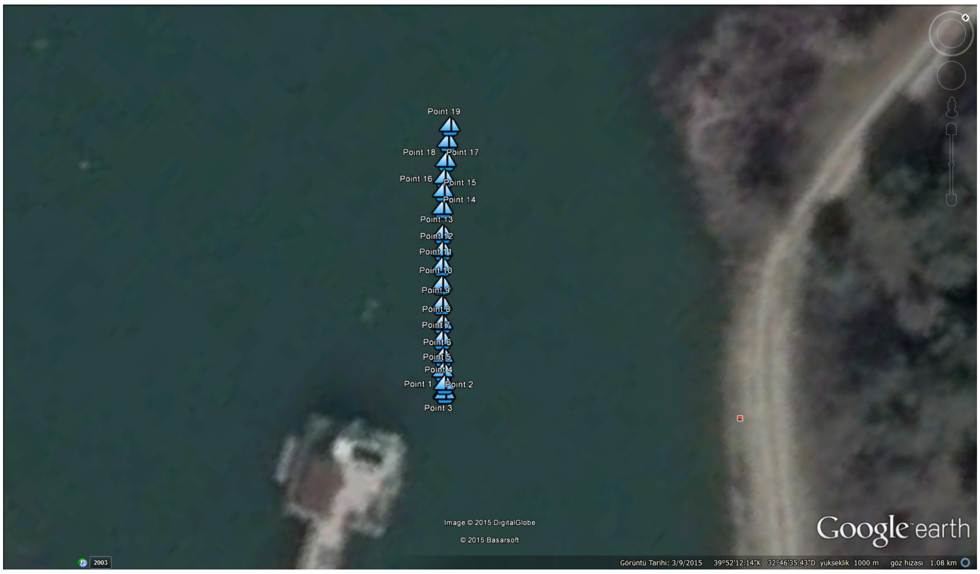


Figure 3.45: Top view of the linear motion captured by Google Earth software.

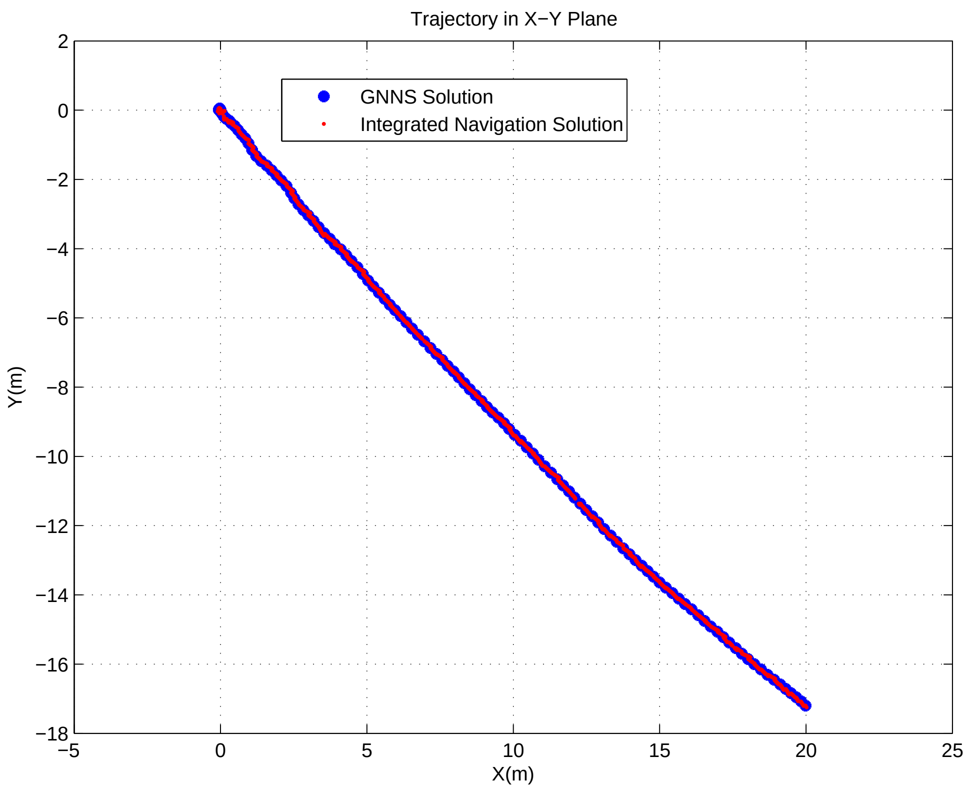


Figure 3.46: Linear trajectories in X-Y plane by integrated navigation and GNSS.

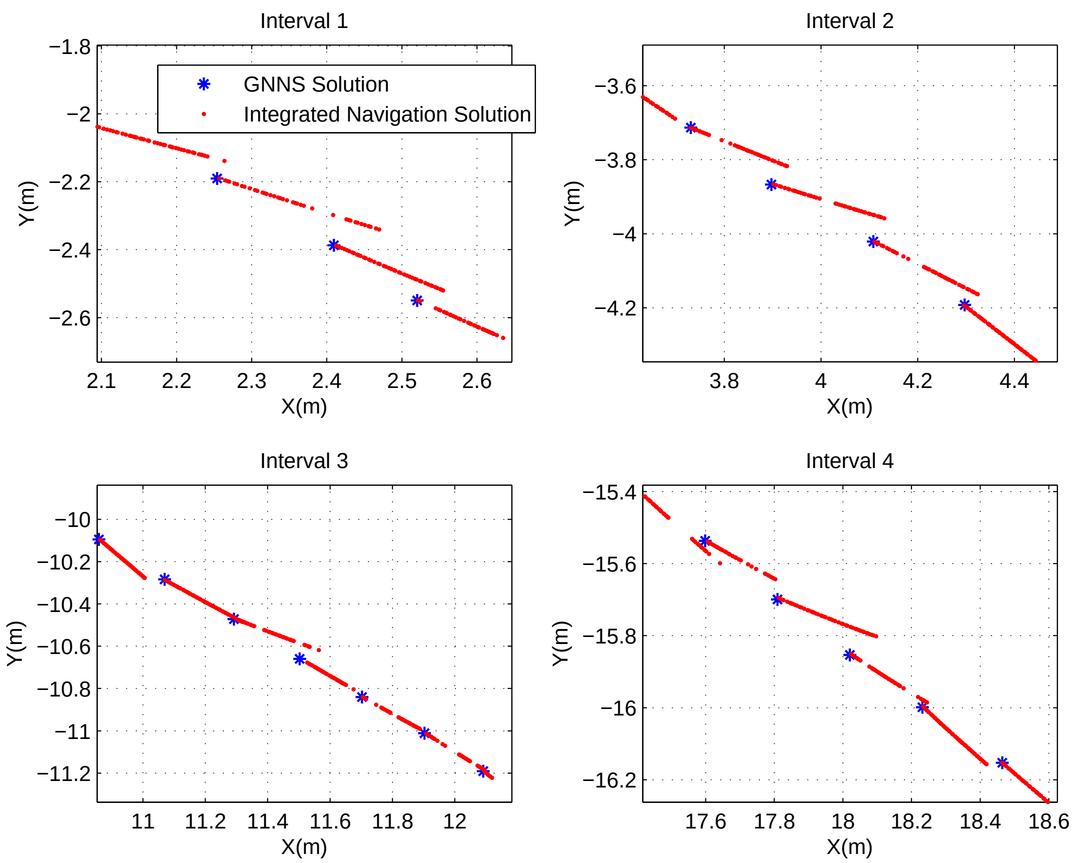


Figure 3.47: Linear trajectories in X-Y plane in four subintervals.

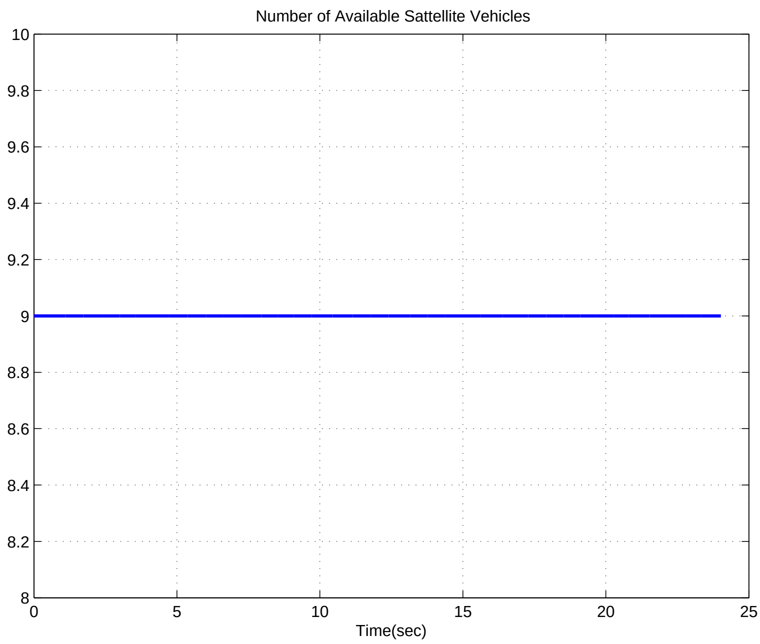


Figure 3.48: The number of available GNSS satellites during the linear motion test.

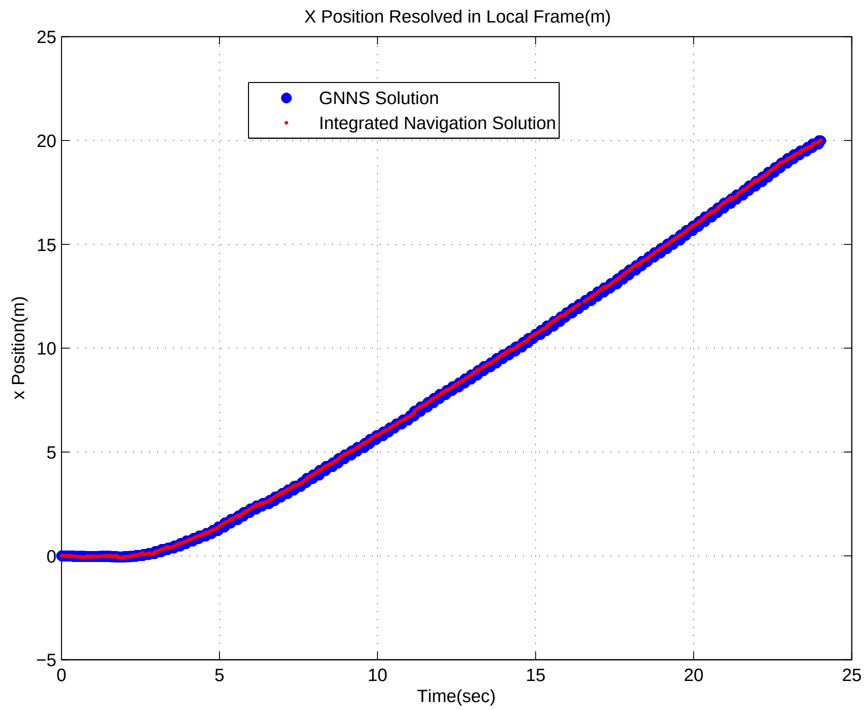


Figure 3.49: X position in local navigation frame by integrated navigation and GNSS through linear motion test.

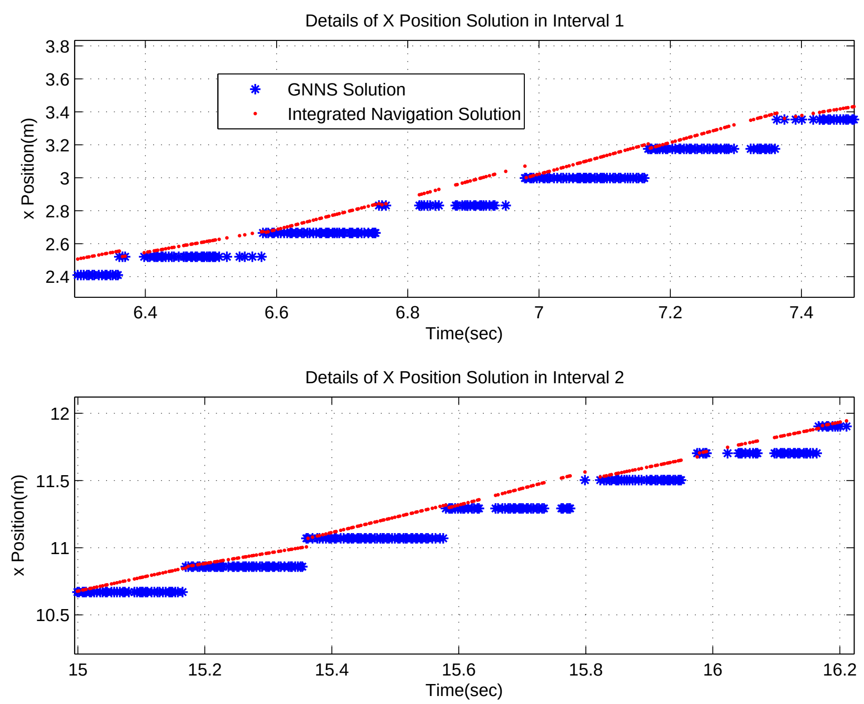


Figure 3.50: A zoomed sight of X position by integrated navigation and GNSS through linear motion test.

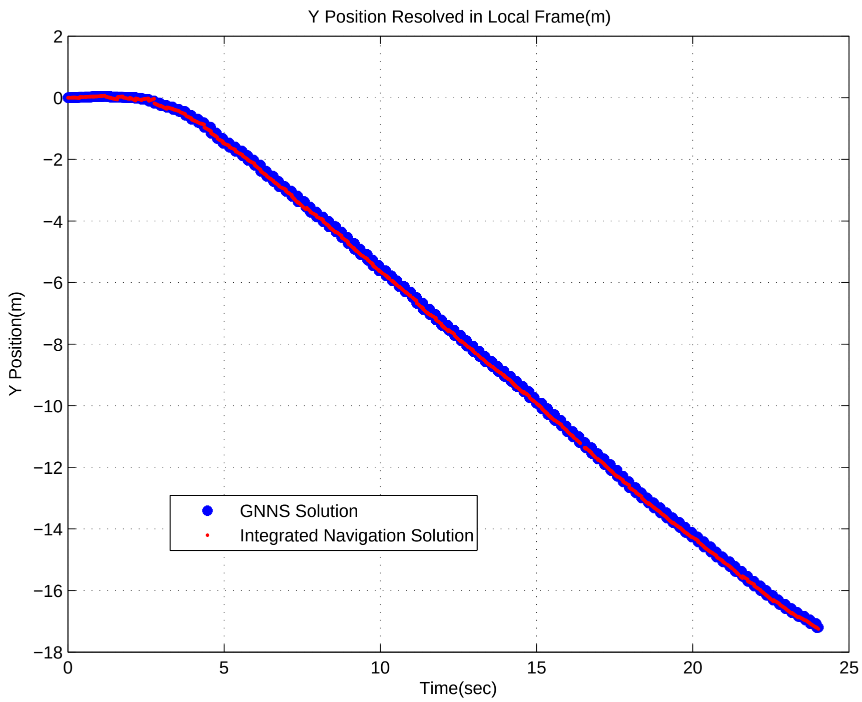


Figure 3.51: Y position in local navigation frame by integrated navigation and GNSS through linear motion test.

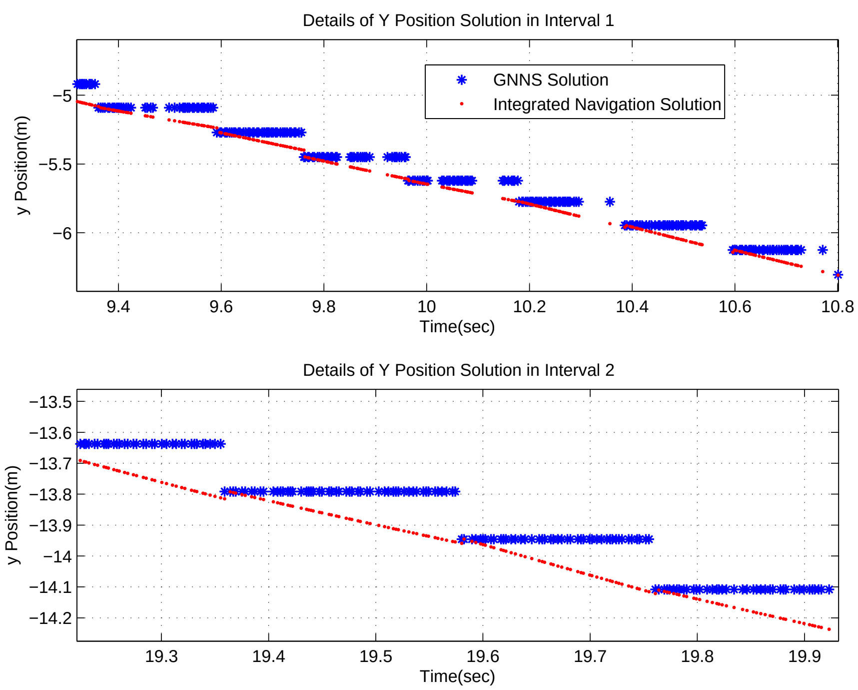


Figure 3.52: A zoomed sight of Y position by integrated navigation and GNSS through linear motion test.

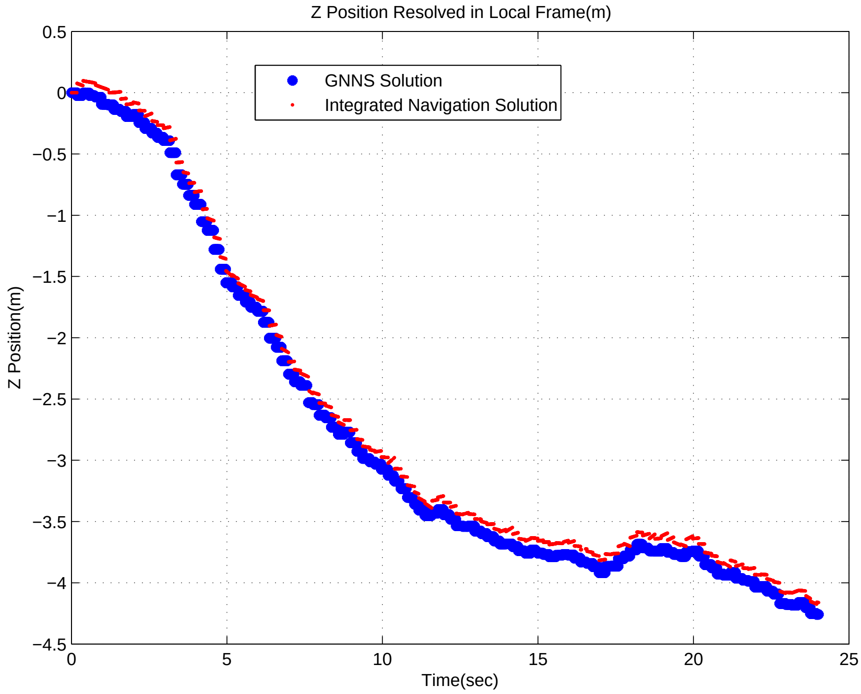


Figure 3.53: Z position in local navigation frame by integrated navigation and GNSS through linear motion test.

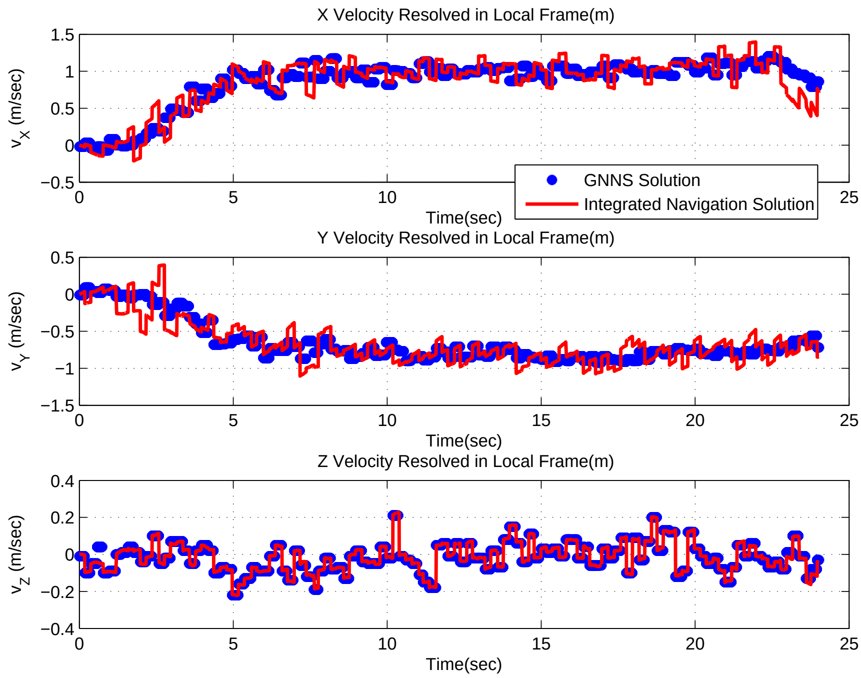


Figure 3.54: Velocity solution in local navigation frame by integrated navigation and GNSS through linear motion test.

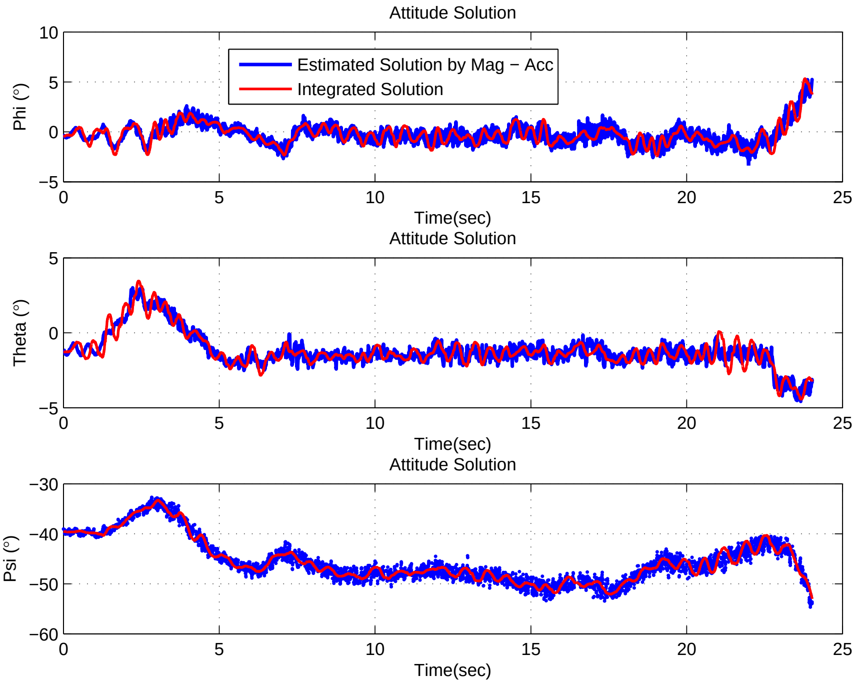


Figure 3.55: Attitude solution w.r.t. local navigation frame indicated by integrated navigation through linear motion test.

Circular Motion Test:

Figure 3.56: Top view of the circular motion captured by Google Earth software.

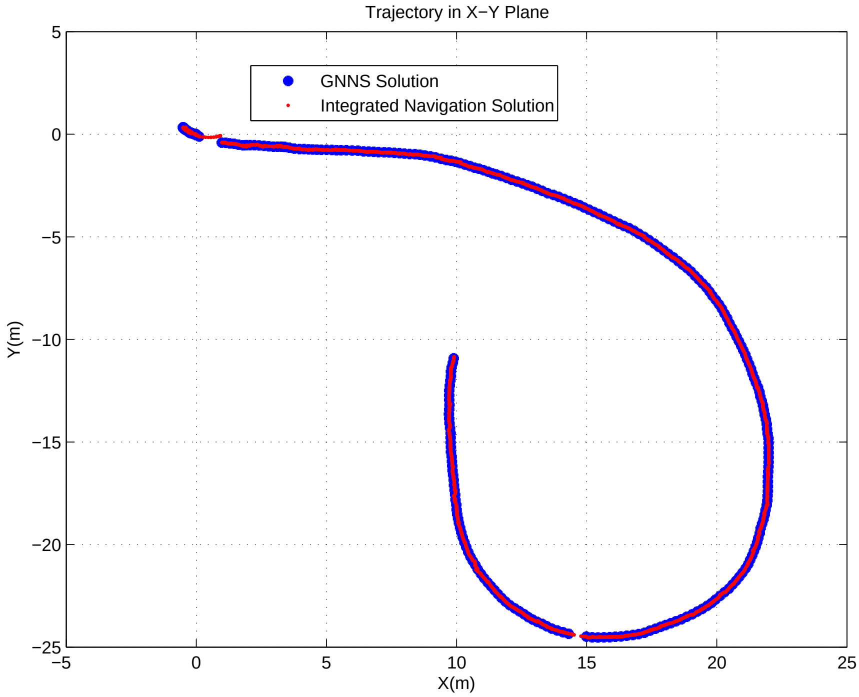


Figure 3.57: Circular trajectories in X-Y plane by integrated navigation and GNSS.

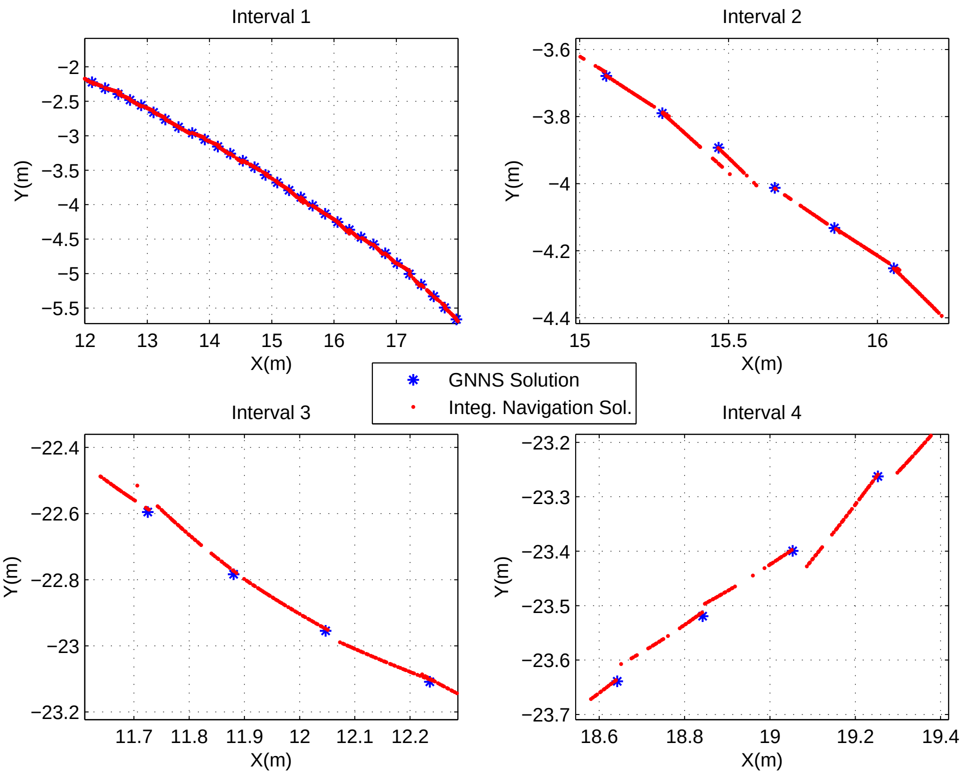


Figure 3.58: Circular trajectories in X-Y plane in four subintervals.

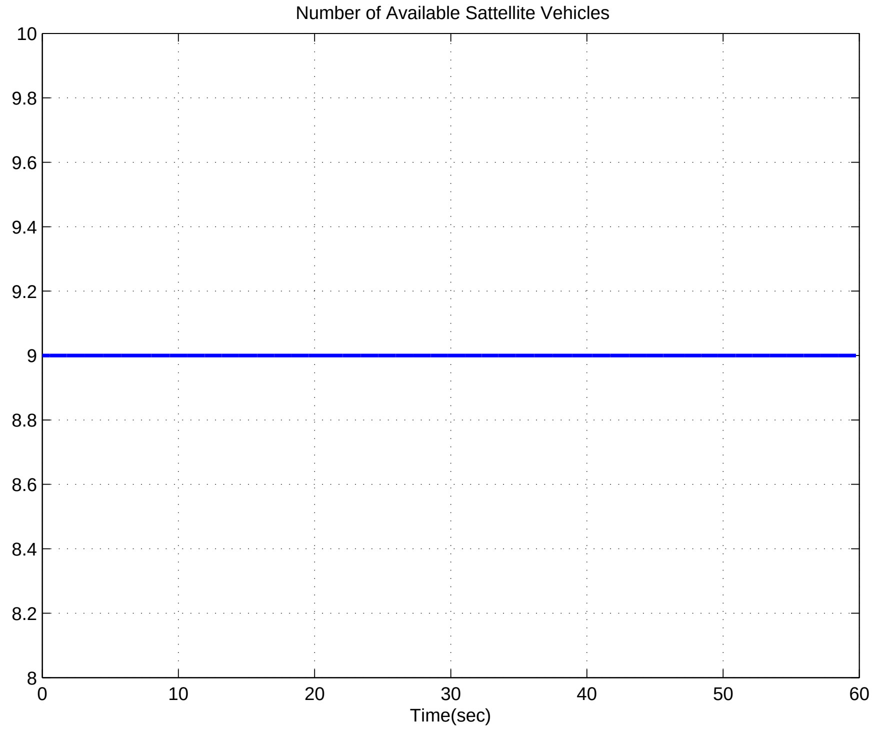


Figure 3.59: The number of available GNSS satellites during the circular motion test.

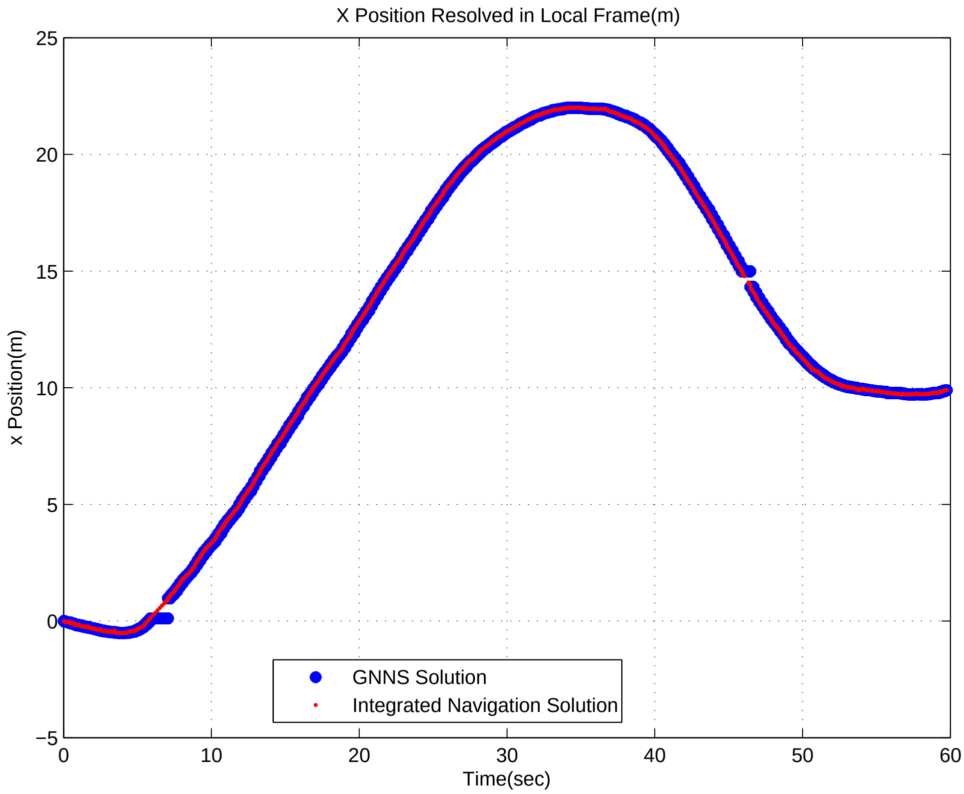


Figure 3.60: X position in local navigation frame by integrated navigation and GNSS through circular motion test.

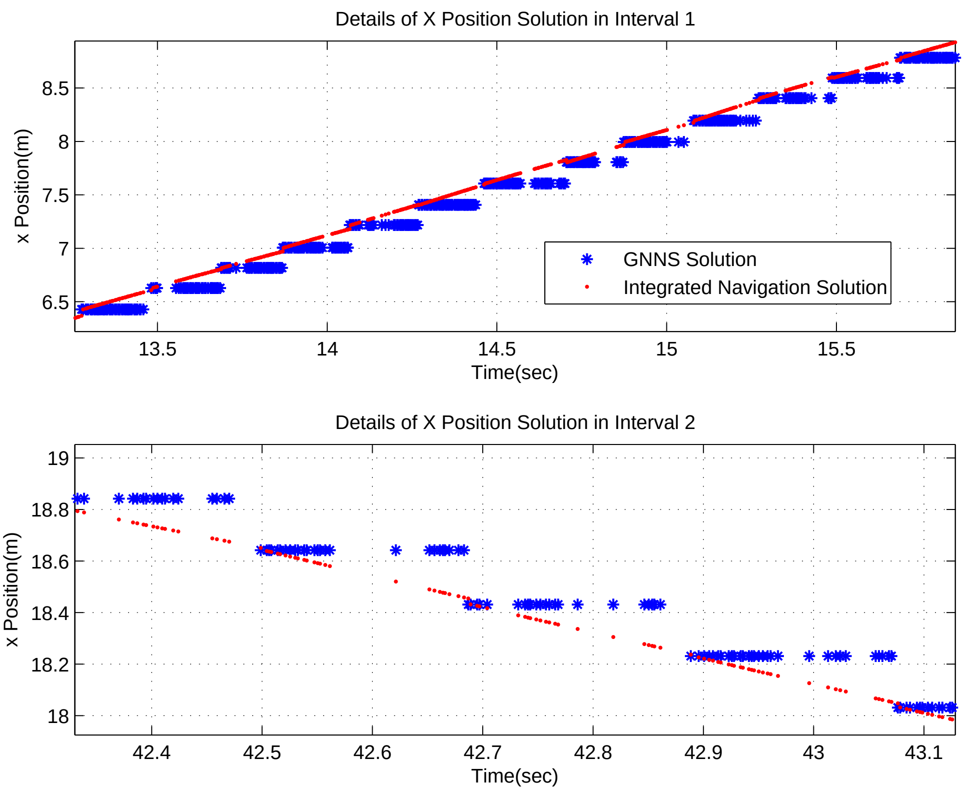


Figure 3.61: A zoomed sight of X position by integrated navigation and GNSS through circular motion test.

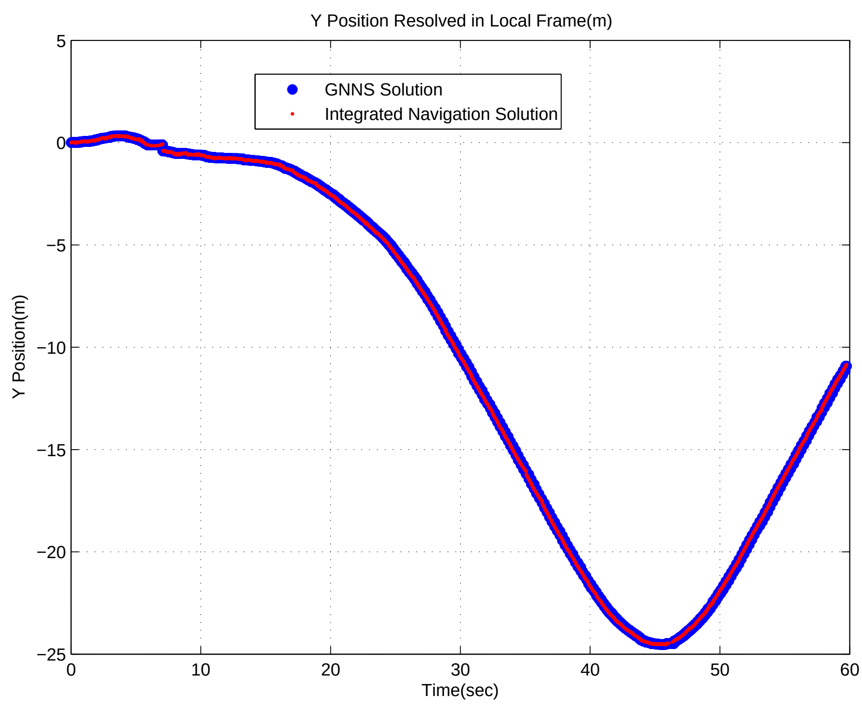


Figure 3.62: Y position in local navigation frame by integrated navigation and GNSS through circular motion test.

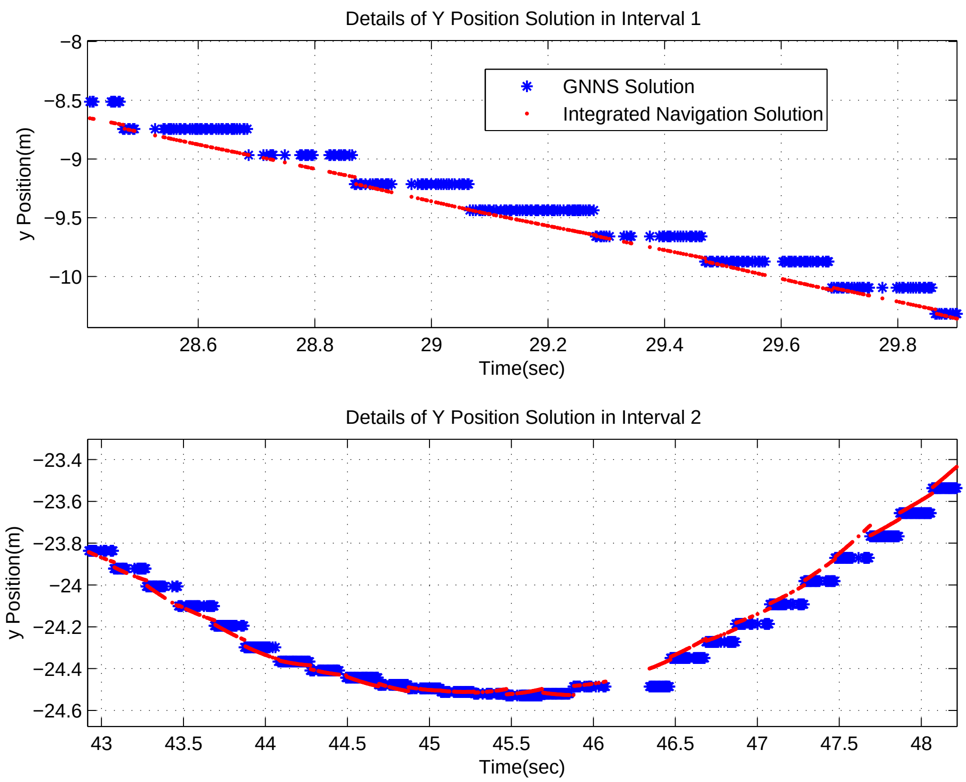


Figure 3.63: A zoomed sight of Y position by integrated navigation and GNSS through circular motion test.

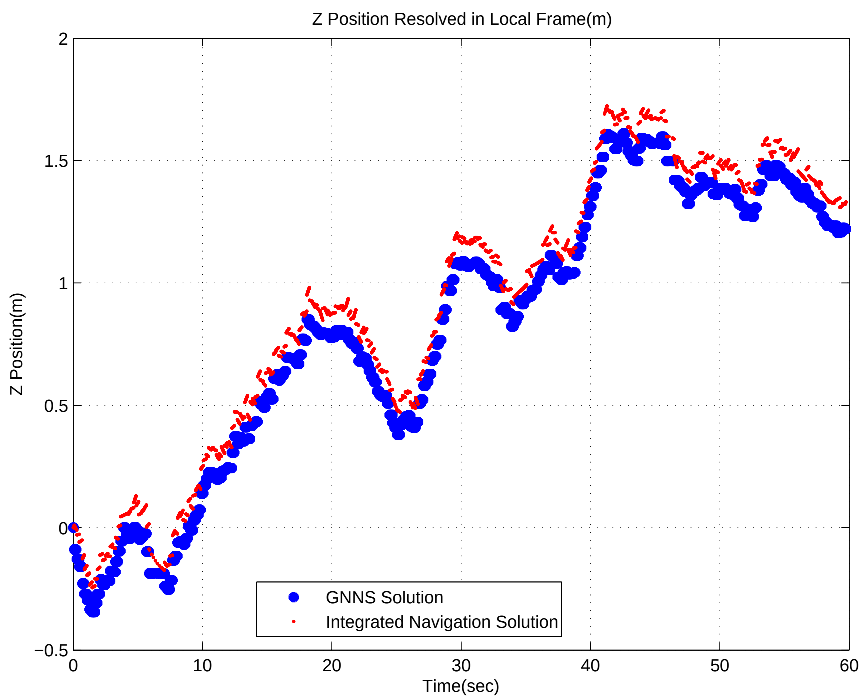


Figure 3.64: Z position in local navigation frame by integrated navigation and GNSS through circular motion test.

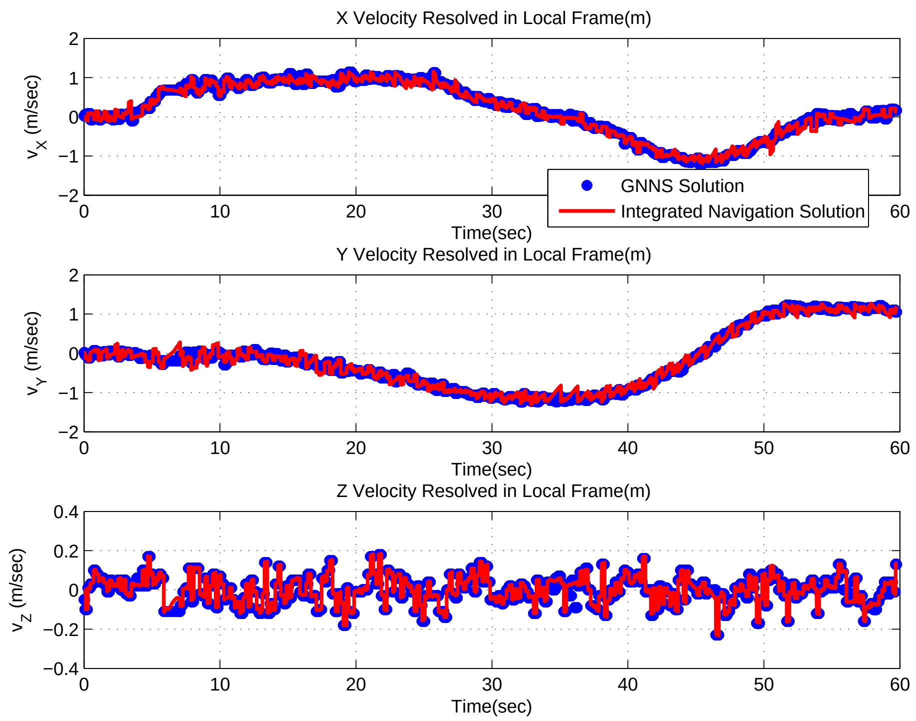


Figure 3.65: Velocity solution in local navigation frame by integrated navigation and GNSS through circular motion test.

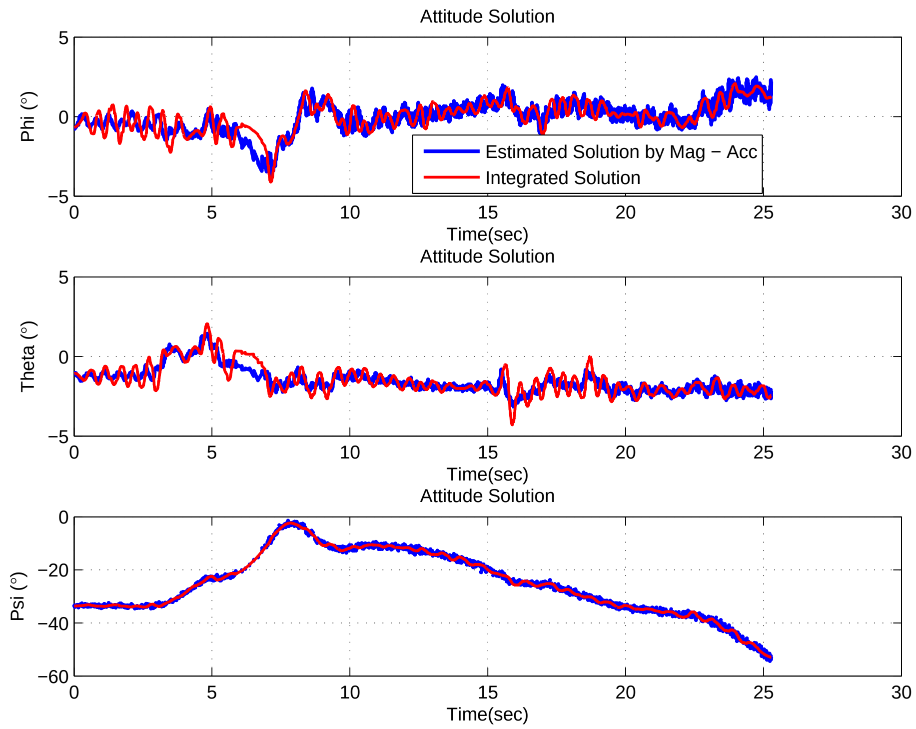


Figure 3.66: Attitude solution w.r.t. local navigation frame indicated by integrated navigation through circular motion test.

CHAPTER 4

EXPERIMENTAL SETUP

4.1 Introduction

Up to now, the progress of this study can be summarized as follows: The mathematical model of an unmanned sea surface vehicle is built at first. Later, the navigation algorithm is constructed to prepare promising data sets. Afterwards, a model boat is integrated with the electrical equipments and the driving system to be able to realize the controlled motion of the vessel. The measurements collected from the boat are first processed by the navigation algorithm, and then used to identify the mathematical parameters of the boat as described in [6]. The details of the software and hardware employed through the experimentation will be presented in this chapter.

In this study, a model boat namely Pacific Islander Tugboat demonstrated in Figure 4.1 is employed. A ready-to-use, advanced autopilot card, Pixhawk, which is mounted in the place of the wheel house of the boat is utilized. A software embedded on the autopilot card is developed for communication with the ground station, driving the motors in accordance with the commands taken from the ground, collecting and storing the sensor readings during motion. Moreover, another software running on a PC, called ground station, is constructed to send thruster commands to the vehicle by using a specific communication protocol named Micro Air Vehicle Link (MAVLink). This program is implemented on MATLAB/Simulink environment. It is able to transmit commands for state transitions and the amount of thrusts to be applied by each propeller by making use of a serial communication channel over a wireless module.

Figure 4.1: Pacific Islander Tugboat.

In this chapter, the setup is introduced by dividing it into two main sections: hardware and software. For instance, the hardware components and peripherals of the autopilot card will be expressed in the former section while the architecture of the software will be described in the latter one.

4.2 The Hardware Utilized in the Experiments

4.2.1 The Model Boat, Driving System and Batteries

The model Pacific Islander Tugboat represented is a highly detailed model of the real tugboat that is scaled by 1:40. It is made of composite rendering it quite resistant to possible crushes. The length is about 900 mm, the beam is 29 mm and the height is around 58 mm. Excluding the electronics, ballast weights and batteries, the weight is about 9.5 kg. It is delivered with two brushed DC motors and electronic speed controllers (ESC) which are later replaced by waterproof and high power equivalents. The vessel has two distinct, large, 4 bladed propellers made of brass whereby the agile maneuvers become feasible. In addition, dual Kort Nozzles changing the direction of thrust exerted on the water are included although they are fixed and not functional associated with the rationale of this study. The propellers of the marine craft are depicted in Figure 4.2.

Figure 4.2: Two independent propellers of the model boat, [5].

The drive system comprises of two separated subparts each containing a brushless DC motor and an ESC to power the motor. Although the boat is already dispensed with a ready to use drive system with brushed DC motors and ESCs, these are replaced with water-proof, high-power, and brushless counterparts due to reliability, maintenance and performance concerns. The contemporary Seaking-120 A-V3 ESC and 3660SL 3180KV brushless DC motor are illustrated by Figures 4.3 and 4.4, respectively. Both are produced by Hobbywing Technology. The ESC is able to provide 120 A continuous and up to 720 A peak current. As can be seen in the figures, the motors feature pre-installed water cooling jackets, and both motors and ESCs are operated with water cooling systems.

Figure 4.3: The current ESC utilized in the boat.

Figure 4.4: The current DC Motor utilized in the boat.

The ESCs essentially take pulse-width modulation (PWM) signals from the Pixhawk and control the speed of the propellers. The PWM signal basically corresponds to a constant voltage signal, the level of which varies related to the duty cycle. In other words, the motion of the vessel is controlled by arranging the pulse widths of two PWM signals. Besides, the relation between the applied voltage and the magnitude of output thrust is obtained as described in Figure 4.5. By doing so, the commands can be arranged to represent the magnitude of thrust vector.

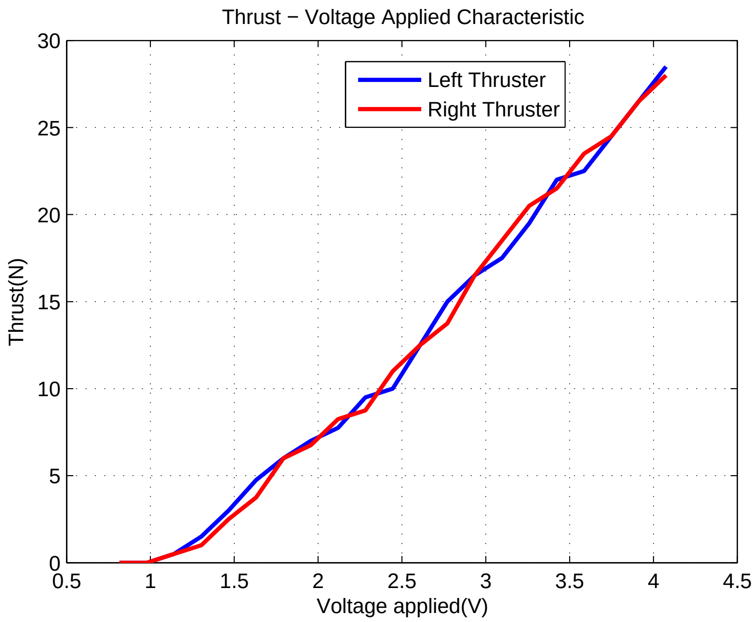


Figure 4.5: Thrust-voltage characteristic of the drive system, [6].

A 4000 mAh, 8.4 V lithium-polymer (LiPo) battery consisting of two cells is selected as the power source of the entire system. It is directly connected to the ESCs, motors and the pump of the water cooling system. Furthermore, a voltage regulator is employed to service constant 5 V to the Pixhawk. All of the sensors are attached to the Pixhawk and energized by its support.

4.2.2 The Autopilot Card and Sensors

In this study, a Pixhawk autopilot card represented by Figure 4.6 is acquired owing to its being ready to use at reasonable price, customizable, open-source, open-hardware, and efficient with a real-time operating system (RTOS). It is provided as a complete system with its software and hardware and can easily be programmed by using C/C++ via an integrated development environment such as Eclipse.

Figure 4.6: Pixhawk autopilot card, [6].

Pixhawk basically consists of a Flight Management Unit (FMU) and an Input/Output (IO) module. The FMU depicted in Figure 4.7 contains 192 KB SRAM, 1024 KB flash memory, a USB Bootloader and a 168 Mhz Cortex-M4F combining a 32 bit microcontroller, a Digital Signal Processor (DSP) and a Floating Point Unit (FPU). It has a microSD slot to satisfy needs for extra memory. It offers the following interfaces: UART, I2C, SPI and CAN. There are four different sensor units on FMU: a MPU-6000 by Invensense including a 3 axis accelerometer and a 3 axis gyro; a L3GD20 3 axis gyro by ST Microelectronics; a LSM303D, combining a 3 axis accelerometer and a magnetometer by ST Microelectronics and a MS5611, pressure sensor by Measurement Specialities INC. As previously mentioned, gyroscopes measure angular rate information and the accelerometers output specific force measurements. Two pairs of the sensors are included in FMU considering emergency cases as these are of vital importance during an operation based on the measurements. The autopilot algorithms are basically realized on the FMU. The IO module is designed as a carrier board to the FMU such that it provides interfaces and a stable 5V power. It includes a 24 Mhz Cortez-M3 failsafe microcontroller, 8 high speed PWM outputs, 2 solid state relays and multiple power outputs for the peripherals.

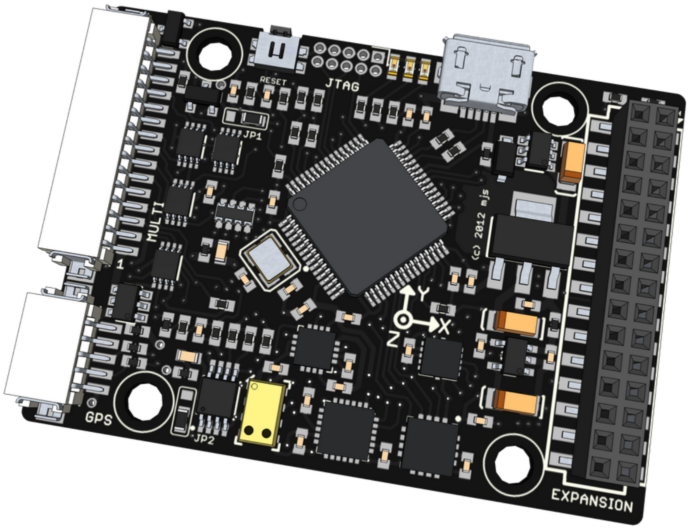


Figure 4.7: An illustration of Pixhawk-FMU, [7].

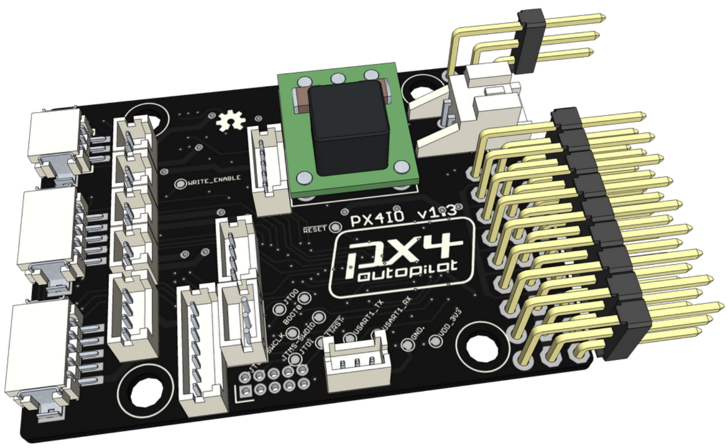


Figure 4.8: An illustration of Pixhawk-IO module, [7].

Figure 4.9 demonstrates the peripherals connected to Pixhawk. In this figure, the top left and right units are receptively the safety switch with an internal LED and the buzzer. These serve the purpose of informing the outside world about the instantaneous state of the software, whether there exists an emergency case or not. Besides, the safety switch activates the PWM outputs. The below right component contains a u-blox NEO-7 GPS module with a compass. The update rate of the GPS is about 5 Hz. Finally, the 3DR telemetry radio is located at the bottom left corner of the figure. It creates a connection with the ground station. It is specialized for working with MAVLink packets and allows ranges up to several kilometers.

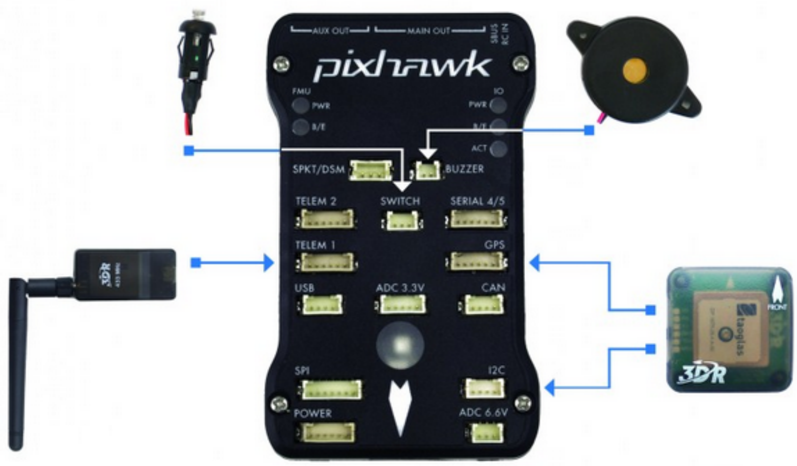


Figure 4.9: A diagram of the peripherals, [8].

4.3 The Developed Softwares

In this section, the high level software architecture of Pixhawk is briefly introduced at first, then the developed programs running on the autopilot and the ground station are presented.

4.3.1 Pixhawk Software

Pixhawk utilizes a high level architecture consisting of nodes using semantic channels such as 'attitude' or 'position' to broadcast the system state in a messaging network as described in [7]. The organization is divided into the following four major layers as illustrated by Figure 4.10/:

- Application Programming Interface (API) of applications,
- Application framework,
- Libraries
- Operating system.

The API of applications is designed for application developers. The application framework realizes main applications called nodes, establishing the core for operation controls. All of the system libraries are included in libraries layer. Finally, hardware drivers, networking and failsafe systems are embedded into operating system layer [7].

Pixhawk uses a publish-subscribe pattern object request broker to maintain communication between processes/applications. To illustrate, a special application collects measurements from various sensors into a packet and is required to transmit the packet to the receivers with minimum latency. For this purpose, the nodes exchange messages over buses, namely topics. That is to say, an application takes the necessary message for its process from related bus/topic, and it also transmits a message to the outside by publishing it on a topic. In short, the nodes does not know with whom they are communicating. The locking issues are prevented by this method.

In this study, the Pixhawk is basically intended for driving the DC motors in accordance with the thrust commands received from the ground station and recording the measurements of all sensors into a log file. To that end, an application is developed with two main states, namely, pre-operational mode and operational mode. When the software is in pre-operational mode, it collects all sensor data, communicates with the base station over the telemetry device; however, it is not able to power the engines. This mode is essentially aimed for safety issues while the vessel is still located on land for debug purposes. Besides, the sensor calibration procedures for accelerometers and gyroscopes may also be performed in this state since the boat is guarantied to be stationary in the absence of disturbances. The operational mode enables the PWM signals to be applied to related channels. The application handles the communication over topics as formerly described.

4.3.2 The Ground Station Software

The software running on the ground station is designed to communicate with the Pixhawk over the telemetry device. MAVLink which is a header only message library first released early 2009 by Lorenz Meier is benefited as the communication protocol [40]. The structure of a MAVLink message can be illustrated as follows: first 6 bytes are used for the header, and these are followed by variable number of bytes representing transferred information, called payload; the last 2 bytes are reserved for the checksum. Figure 4.11 demonstrates the details of the message. The ground station software is implemented by using MATLAB/Simulink environment due to its being convenient, stable and easy-to-use.

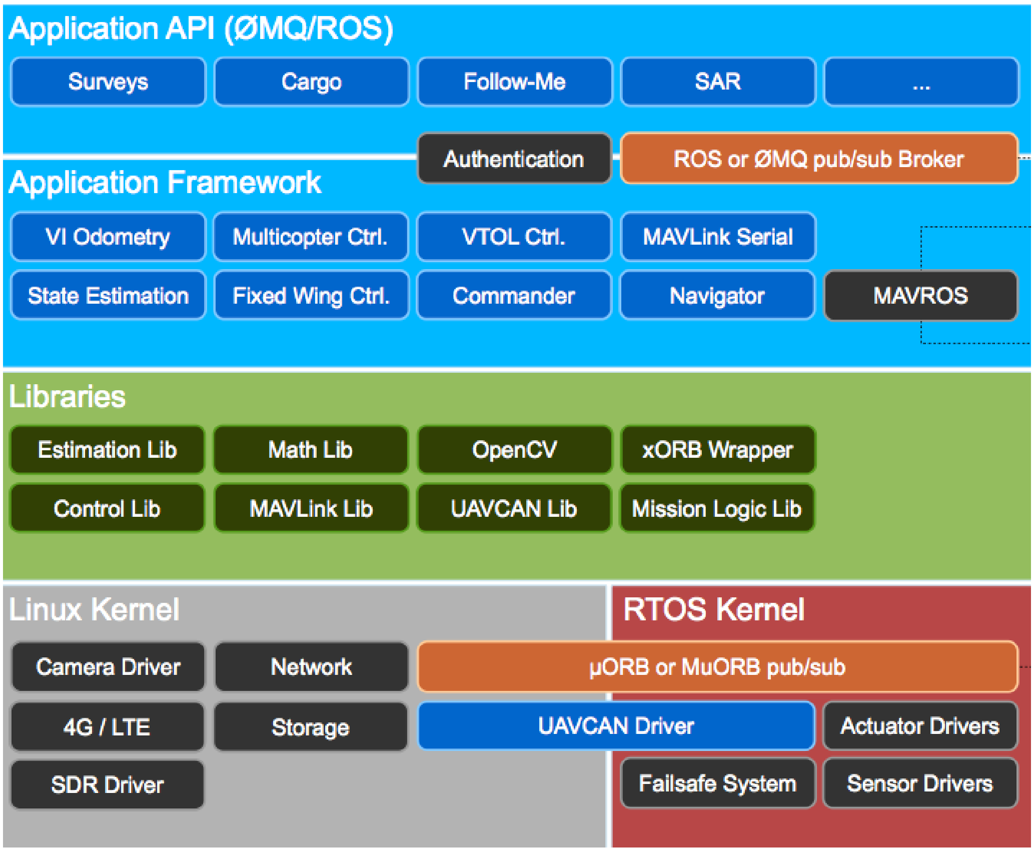
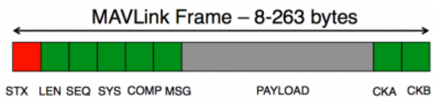


Figure 4.10: Pixhawk high-level software architecture, [7].



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

Figure 4.11: Structure of a MAVLink Message, [9].

CHAPTER 5

AUTOPILOT DESIGN

5.1 Introduction

In this chapter, several autopilots to control the motion of a sea surface vehicle are examined. The design procedure of the autopilots is based on the introduced mathematical model of a marine vessel. Furthermore, the parameter set of the model Pacific Islander Tug Boat which is identified by the study of Erünsal, [6], is also taken into account in the design process. To that end, Linear Quadratic Regulator (LQR) and Feedback Linearization controllers are utilized in the autopilot algorithm. Therefore, both a linear and an advanced nonlinear types of controllers are studied.

Disturbances rooted in the surroundings of the vehicle are set to be zero throughout the autopilot design, whereas the disturbance rejection characteristics of the algorithms are also presented in the section reserved for the acquired results. As stated in the Experimental Setup chapter, there are two actuators in the model, namely, left and right thrusters. The ultimate goal of the controllers is to force the vehicle to reach the desired surge velocity and yaw positions which are set by an upper level controller. For this purpose, the magnitudes of thrusts to be applied by left and right thruster are calculated.

In this chapter, the fundamentals for the design of LQR controller are familiarized at first. Then, the theory and implementation details of LQR are expressed. Afterwards, the basis of FL controller design is discussed, and the implementation is enlarged upon. Finally, simulation results obtained by both of the controllers are clarified and compared.

5.2 Autopilot Design with Linear Quadratic Regulator (LQR) Controller

In this section, the mathematical tools, which are essential for understanding the theory of LQR are presented at first. Afterwards, the problem of linear quadratic regulation is introduced, and the optimal control law for this problem is attained under certain assumptions. Next, the command tracking problem of the marine vessel system is transformed into a regulation problem via augmentation of the state vector. Finally, the controller rooted in the previous examinations is designed for the platform.

5.2.1 Fundamentals for LQR Design

5.2.1.1 Linearization

Recall the derived mathematical model of a marine craft given by (5.1) and (5.2). The environmental force and torque components will be neglected in (5.2) as described in [32].

$$\dot{\eta} = \boldsymbol{J}_{\Theta}\boldsymbol{\nu} \tag{5.1}$$

$$\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu}) = \tau_d + \tau_g + \tau_t + \tau_a + \tau_c \tag{5.2}$$

where

$$\begin{aligned} \eta &= \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T \\ \boldsymbol{\nu} &= \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \end{aligned}$$

The above nonlinear model is required to be linearized around determined potential operating points to be able to make use of LQR controllers. Subsequently, the plant is to be presented in the classical state space representation. During this procedure, environmental effects (i.e., wind, water current) are set to zero, and the effect of air drag is supposed to occur only from velocity of the vehicle. Equation (5.3) is the continuous time linear representation of the system model.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad where\ \boldsymbol{x} = \begin{bmatrix} \eta \\ \nu \end{bmatrix} \tag{5.3}$$

The state vector, $\boldsymbol{x}$, including positions, orientations resolved in Earth fixed frame and linear, angular velocities resolved in the body fixed frame is in $\mathbb{R}^{12}$. Therefore, the state transition matrix, $\boldsymbol{A}$, appears to be a 12-by-12 matrix. The input vector, $\boldsymbol{u}$ comprising of the magnitudes of thrusts applied by left and right propellers in order, is in $\mathbb{R}^2$. Note that, only the magnitudes are included in $\boldsymbol{u}$ as two scalars. As a result, $\boldsymbol{B}$ is a 12-by-2 matrix and formed in such a way that the first and second columns correspond to the left and right thrusters, respectively.

The nonlinear model referring to the above mentioned state vector can be written as given in (5.4).

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{5.4}$$

After that, a linear approximation of the model around an operation point $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ can be found as illustrated in (5.5) and (5.6). Meanwhile, the linearization points are not necessarily the equilibrium points, i.e., $f(\boldsymbol{x}_0, \boldsymbol{u}_0) \neq \boldsymbol{0}$.

$$\boldsymbol{A} = \left. \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0)} \tag{5.5}$$

$$\boldsymbol{B} = \left. \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}} \right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0)} \tag{5.6}$$

The linearization procedure is carried out for the possible pairs of surge speeds and yaw positions which are respectively selected from the following sets: $u = \{0.1, 1.1, 2.1, 3.1\}$, $\psi = \left\{0, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi\right\}$.

5.2.1.2 Finding Controllable Subspaces of the Representation

The representation of the plant is required to be completely controllable to be able to apply LQR controller [41]. For that reason, the controllability of $(\boldsymbol{A},\boldsymbol{B})$ pairs is to be inspected.

Let the system be defined as in (5.7),

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{Ax}+\boldsymbol{Bu} \\ \boldsymbol{y} &= \boldsymbol{Cx}+\boldsymbol{Du} \end{aligned} \tag{5.7}$$

where the state vector, $\boldsymbol{x}$, is n-by-1; $\boldsymbol{A}$ is a n-by-n matrix; input vector, $\boldsymbol{u}$, is m-by-1; $\boldsymbol{B}$ is a n-by-m matrix. The output vector, $\boldsymbol{y}$, is r-by-1. Accordingly, $\boldsymbol{C}$ and $\boldsymbol{D}$ are matrices whose dimensions are respectively r-by-n and r-by-m.

The controllability matrix $\boldsymbol{Co}$ is defined in (5.8).

$$\boldsymbol{Co} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \dots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{5.8}$$

If the rank of the controllability matrix is not equal to n, the system is said to be not completely controllable. Thereafter, the subspace with the property of complete controllability is to be found at each of the linearization points. To be able to place the poles of the system into desired locations, the controllable subspaces are found by using the Staircase Algorithm [42]:

Let $\bar{\boldsymbol{x}} = \boldsymbol{Tx}$ be the new state vector. Then, the representation immediately becomes:

$$\begin{aligned} \dot{\bar{\boldsymbol{x}}} &= \bar{\boldsymbol{A}}\bar{\boldsymbol{x}}+\bar{\boldsymbol{B}}\boldsymbol{u} \\ \boldsymbol{y} &= \bar{\boldsymbol{C}}\bar{\boldsymbol{x}} \end{aligned} \tag{5.9}$$

where $\bar{\boldsymbol{A}} = \boldsymbol{TAT}^T$, $\bar{\boldsymbol{B}} = \boldsymbol{TB}$ and $\bar{\boldsymbol{C}} = \boldsymbol{CT}$. Besides, (5.10) reveals the internal structures of recently obtained matrices.

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_{uc} & \boldsymbol{0} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{c} \end{bmatrix} \quad \bar{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_{c} \end{bmatrix} \quad \bar{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C}_{uc} & \boldsymbol{C}_{c} \end{bmatrix} \tag{5.10}$$

The pair, $(\boldsymbol{A}_c, \boldsymbol{B}_c)$ is assured to be controllable, and it is noted that the transfer functions of the systems represented by (5.7) and (5.9) are identical as depicted in (5.11).

$$\bar{\boldsymbol{C}}\left(s\boldsymbol{I} - \bar{\boldsymbol{A}}\right)^{-1}\bar{\boldsymbol{B}} = \boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} \tag{5.11}$$

The controllable subspaces are acquired by utilizing this method for all of the linearization points.

5.2.1.3 The Choice of the Sample Time for Discrete Autopilot

The autopilots designed in this section are desired to be realized on a digital autopilot card. Therefore, a proper sample time is to be determined. The selection of the sample time is performed by taking the phase margins of the linearized models of the system into consideration. In other words, the response times are calculated by applying reference signals for all points. The maximum of the response times is found to be equal to 0.25 sec [6]. Hence, the sampling time should be smaller than one fifth of this value as a rule of thumb. Eventually, the sampling frequency of the controller is determined to be 50 Hz by adding some safety margin.

5.2.2 LQR Theory

A short summary of the optimal control theory is given in this section.

Consider the continuous time system in (5.7). The linear quadratic cost function defined in (5.12) is to be minimized with the purpose of regulation of this system.

$$\boldsymbol{J} = \frac{1}{2}\boldsymbol{x}^T(t_f)\boldsymbol{H}\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f}\left[\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right]dt \tag{5.12}$$

$\boldsymbol{J}$ is the general performance index for the linear regulation problems. t_f is final time, $\boldsymbol{H}$ and $\boldsymbol{Q}$ are real, positive semi definite and symmetric matrices. $\boldsymbol{R}$ is a real, positive definite and symmetric matrix. It is assumed that there is no constraint on the input and state vectors. By minimizing (5.12), the state vector is inherently desired to be close to the origin in moderation of the control signal.

Subsequently, the Hamiltonian can be written as in (5.13),

$$\boldsymbol{H}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{P}(t)\right) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}(t) + \frac{1}{2}\boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t) + \boldsymbol{P}(t)\left[\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)\right] \tag{5.13}$$

where $\boldsymbol{P}(t)$ is called co-state vector.

The feature of optimality implies (5.14) and (5.15).

$$\dot{\boldsymbol{P}}(t) = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}} = -\boldsymbol{Q}\boldsymbol{x}(t) - \boldsymbol{A}^T\boldsymbol{P}(t) \tag{5.14}$$

$$\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{u}} = 0 = \boldsymbol{R}\boldsymbol{u}^T(t) + \boldsymbol{B}^T\boldsymbol{P}^T(t) \tag{5.15}$$

By using (5.15), the input vector is formalized as in (5.16).

$$\boldsymbol{u}(t) = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(t) \tag{5.16}$$

Note that since $\boldsymbol{R}$ is chosen to be positive definite, it is also invertible.

The substitution of (5.16) into (5.7) ends up with (5.17).

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(t) \tag{5.17}$$

The boundary condition of $\boldsymbol{P}(t)$ is given in (2.22):

$$\boldsymbol{P}(t_f) = \boldsymbol{H}\boldsymbol{x}(t_f) \tag{5.18}$$

The relation between co-state and state vectors is found as in (5.19) by taking the boundary condition into consideration [41].

$$\boldsymbol{P}(t) = \boldsymbol{K}(t)\boldsymbol{x}(t) \tag{5.19}$$

By substituting (5.19) into (5.16), the input is rewritten.

$$\boldsymbol{u}(t) = -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{K}(t)\boldsymbol{x}(t) \tag{5.20}$$

Note that $\boldsymbol{K}(t)$ depends on t_f, and $\boldsymbol{P}(t)$ is a linear function of the state vector.

Take the derivative of (5.19):

$$\dot{\boldsymbol{P}}(t) = \dot{\boldsymbol{K}}(t)\boldsymbol{x}(t) + \boldsymbol{K}(t)\dot{\boldsymbol{x}}(t) \tag{5.21}$$

When (5.7), (5.14), (5.19) and (5.20) are put into (5.21), the following equations are attained.

$$-\boldsymbol{Q}\boldsymbol{x}(t) - \boldsymbol{A}^T\boldsymbol{K}(t)\boldsymbol{x}(t) = \dot{\boldsymbol{K}}(t)\boldsymbol{x}(t) + \boldsymbol{K}(t)\boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{K}(t)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K}(t)\boldsymbol{x}(t) \tag{5.22}$$

$$-\boldsymbol{Q} - \boldsymbol{A}^T\boldsymbol{K}(t) = \dot{\boldsymbol{K}}(t) + \boldsymbol{K}(t)\boldsymbol{A} - \boldsymbol{K}(t)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T K(t) \tag{5.23}$$

$\boldsymbol{K}$ can be solved with the boundary condition, $\boldsymbol{K}(t_f) = \boldsymbol{H}$. It is a Riccati type differential equation. Since $\boldsymbol{K}$ is an n-by-n, symmetric matrix, n(n+1)/2 differential equations are to be solved [41]. Kalman [43] has shown that if

(i) The representation is completely controllable,

(ii) $\boldsymbol{H} = 0$

(iii) $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{R}$ and $\boldsymbol{Q}$ are constant matrices,

then, $\boldsymbol{K}$ appears to be constant as $t \to t_f$ implying that the optimal control law for the mentioned infinite-duration regulation problem is stationary.

In that case, for fixed $\boldsymbol{K}$, i.e., $\dot{\boldsymbol{K}}(t) = 0$, the Riccati equation takes the following form:

$$0 = \boldsymbol{Q} + \boldsymbol{A}^T\boldsymbol{K} + \boldsymbol{K}\boldsymbol{A} - \boldsymbol{K}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K} \tag{5.24}$$

$\boldsymbol{K}$ matrices are calculated at every linearization point by solving the above equation.

5.2.3 LQR Design for Tracking Problems

Up to now, optimal control theory of linear regulation problem is introduced. However, the desired value of the state vector is not necessarily zero for the marine craft system. The vessel is commanded to track reference signals consisting of specified surge speed and yaw position commands.

To apply LQR controller on a tracking problem, the state vector is augmented by the integral of errors between the reference signal, $\boldsymbol{r}(t)$ and the states desired to be controlled.

In other words, a tracking problem is transformed into a regulation one by involving the reference in augmented integral states [2].

The system developed up to now is indicated by 5.25.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{5.25}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{5.26}$$

The vector, $\boldsymbol{z}$, is defined as:

$$\dot{\boldsymbol{z}} = \boldsymbol{r}(t) - \boldsymbol{y} = \boldsymbol{r}(t) - \boldsymbol{C}\boldsymbol{x} \tag{5.27}$$

where $\boldsymbol{r}(t)$ is the reference signal, comprising of the desired yaw position and surge velocity to be followed by the system.

The state vector is rearranged to include $\boldsymbol{z}$.

$$\boldsymbol{x}_{aug} = \begin{bmatrix} \boldsymbol{x} \\ z \end{bmatrix} \tag{5.28}$$

$$\dot{\boldsymbol{x}}_{aug} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ z \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{1} \end{bmatrix} \boldsymbol{r} \tag{5.29}$$

Let $\boldsymbol{A}_{aug} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{0} \end{bmatrix}$, $\boldsymbol{B}_{aug} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix}$, and denote the desired values of states and inputs respectively by $\boldsymbol{x}_{aug,d}$, and $\boldsymbol{u}_d$. Thereby, (5.29) is rewritten in the following formula.

$$\dot{\boldsymbol{x}}_{aug,d} = \begin{bmatrix} \dot{\boldsymbol{x}}_d \\ \dot{z}_d \end{bmatrix} = \boldsymbol{A}_{aug}\boldsymbol{x}_{aug,d} + \boldsymbol{B}_{aug}\boldsymbol{u}_d + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{1} \end{bmatrix} \boldsymbol{r} \tag{5.30}$$

The desired values are subtracted from the instantaneous state values while $\boldsymbol{r}$ is assumed to be constant, $\boldsymbol{r}(t) = \boldsymbol{r}$.

$$\dot{\boldsymbol{x}}_{aug} - \dot{\boldsymbol{x}}_{aug,d} = \begin{bmatrix} \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d \\ \dot{z} - \dot{z}_d \end{bmatrix} = \boldsymbol{A}_{aug} \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_d \\ z - z_d \end{bmatrix} + \boldsymbol{B}_{aug}(\boldsymbol{u} - \boldsymbol{u}_d) \tag{5.31}$$

The state error, $\boldsymbol{e}$, and the input error, $\boldsymbol{v}$, are described as:

$$\boldsymbol{e} = \boldsymbol{x}_{aug} - \boldsymbol{x}_{aug,d}, \quad \boldsymbol{v} = \boldsymbol{u} - \boldsymbol{u}_d \tag{5.32}$$

Thereafter, the notation of (5.31) is simplified.

$$\dot{\boldsymbol{e}} = \boldsymbol{A}_{aug}\boldsymbol{e} + \boldsymbol{B}_{aug}\boldsymbol{v} \tag{5.33}$$

Finally, the problem is transformed into regulation by (5.33).

Let the optimal control law, $\boldsymbol{v}^{\star}$, for the above system exist.

$$\boldsymbol{v}^{\star} = -\boldsymbol{K}_{aug}\boldsymbol{e} \tag{5.34}$$

The following equation is acquired by putting (5.34) into (5.33).

$$\dot{\boldsymbol{e}} = (\boldsymbol{A}_{aug} - \boldsymbol{B}_{aug}\boldsymbol{K}_{aug})\,\boldsymbol{e} \tag{5.35}$$

$\boldsymbol{K}_{aug}$ can be calculated with the method expressed in the previous section.

The LQR gain, $\boldsymbol{K}_{aug}$, is partitioned.

$$\boldsymbol{K}_{aug} = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \tag{5.36}$$

Thus, $\boldsymbol{v}$ takes the below form:

$$\boldsymbol{v} = -\begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_d \\ \boldsymbol{z} - \boldsymbol{z}_d \end{bmatrix} = -\boldsymbol{K}_1(\boldsymbol{x} - \boldsymbol{x}_d) - \boldsymbol{K}_2(\boldsymbol{z} - \boldsymbol{z}_d) \tag{5.37}$$

(5.38) is attained by combining (5.32) and (5.37) while $\boldsymbol{u}_d$ and $\boldsymbol{z}_d$ are taken as zero.

$$\boldsymbol{u} = -\boldsymbol{K}_1(\boldsymbol{x} - \boldsymbol{x}_d) - \boldsymbol{K}_2\boldsymbol{z} \tag{5.38}$$

$\boldsymbol{K}_2$ can be symbolized by $-\boldsymbol{K}_I$ in regard to $\boldsymbol{z}$ standing for the integral of the error.

$$\boldsymbol{u} = -\boldsymbol{K}_1\,(\boldsymbol{x} - \boldsymbol{x}_d) + \boldsymbol{K}_I\boldsymbol{z} \tag{5.39}$$

where $\boldsymbol{K}_I = -\boldsymbol{K}_2$.

5.2.4 Implementation of LQR Controller

In the preceding section, theoretical basis for obtaining the proportional and integral coefficients of the controller is already provided. In this section, the implementation of the LQR controller in MATLAB environment is briefly discussed.

At first, the nonlinear equations of the system are linearized by a ready to use function, '*jacobian*', in MATLAB. Thus, $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices are attained. Next, the state vector is augmented by appending the error information to convert the tracking problem into regulation. Thereafter, the corresponding state and input values are substituted into the matrices at each linearization point, and the procedure is ended up with $(\boldsymbol{A},\boldsymbol{B})$ pairs where $\boldsymbol{A}$ is 12-by-12 and $\boldsymbol{B}$ is 12-by-2. Consequently, the augmented pairs, $(\boldsymbol{A}_{aug},\boldsymbol{B}_{aug})$, are also determined where $\boldsymbol{A}_{aug}$ is 14-by-14 and $\boldsymbol{B}_{aug}$ is 14-by-2. Likewise, $\boldsymbol{C}$ causing the surge speed and yaw position appear at the output, is a constant 2-by-14 matrix for all of the linearization points.

After that, $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are established. During the selection of these matrices, the natural limits originated from the physical characteristics of the actuators are taken into account. It is an iterative process in which the matrices are recursively revised after the assessment of time response of the closed loop system related to the certain reference signals.

At that time, the controllable subspaces of the plant are to be found out. The transformation matrices, $\boldsymbol{T}$, are required to project the cost matrices onto the controllable subspaces. Subsequently, controller gains are obtained by solving the continuous-time differential Riccati equations with '*care*' function in MATLAB. The last step is to transform the Kalman gains back into the original spaces. The controller coefficients are acquired after repeating the above steps for all of the linearization points.

5.2.4.1 Combining the Outputs of the Controllers

So far, the system is linearized around various points, and the controller design is accomplished for these points. Consequently, the control input to be fed to the original nonlinear system will be produced by combination of the controller outputs which is calculated in a weighted manner.

In equation (5.40), $\boldsymbol{u}$ refers to the control input which consists of the forces that will be applied by left and right thrusters. $\boldsymbol{u}_i$ is the output of the i^{th} controller designed considering the i^{th} linearization point. k_i indicates the rate at which i^{th} controller will affect the control input [32].

$$\boldsymbol{u} = \sum_i k_i \boldsymbol{u}_i, \quad where \quad \sum_i k_i = 1 \tag{5.40}$$

The weights are calculated at every iteration of the controllers by the procedure given below [32].

$$\begin{aligned}
k_i &= \frac{\xi_i}{\sum \xi_i} \\
\xi_i &= \frac{1}{\|x - x_i\| + \varepsilon}, \quad \varepsilon = 10^{-6} \\
\xi_i &= 0 \ \text{ if } \ \xi_i < \frac{\beta}{100} \max(\xi)
\end{aligned} \tag{5.41}$$

ξ_i essentially demonstrates how close the current state vector is to i^{th} linearization point. ε is added as a security factor during the computation of the reciprocal of the related distance. ξ_i that is smaller than β percent of the maximum of ξ_i's is set to zero to prevent this linearization point from appearing in the control input.

5.3 Autopilot Design with Feedback Linearization (FL) Controller

Feedback linearization technique is basically employed to convert the nonlinear system into a linear one. Consequently, the well established linear system theory can be employed to manipulate and control the system dynamics. It completely differs from Jacobian linearization that is mentioned in the previous section. Jacobian linearization is a linear approximation of the system dynamics, whereas feedback linearization is realized by selection of a control law which is a nonlinear mapping of the state feedback [44].

5.3.1 Fundamentals for Feedback Linearization

To start with, a single-input single-output (SISO) nonlinear system is depicted in 5.42,

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \\
\boldsymbol{y} &= h(\boldsymbol{x})
\end{aligned} \tag{5.42}
$$

where $\boldsymbol{x}$ is in $\mathbb{R}^{12}$; f, and g are smooth vector fields over $\mathbb{R}^{12}$ and h is the nonlinear output function. $\boldsymbol{U}$ is an open set enclosing an equilibrium point, $\boldsymbol{x}_0$, of the system with zero input. $\boldsymbol{x}$ is assumed to be in $\boldsymbol{U}$ for the succeeding calculations [44].

$\boldsymbol{y}$ is differentiated with respect to time.

$$
\dot{\boldsymbol{y}} = \frac{\partial h}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial h}{\partial \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\partial h}{\partial \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{u} \triangleq L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} \tag{5.43}
$$

where $L_f h(\boldsymbol{x})$ and $L_g h(\boldsymbol{x})$ are called Lie derivatives of $h(\boldsymbol{x})$ regarding $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$, respectively. If $L_g h(\boldsymbol{x})$ is not close to zero for all $\boldsymbol{x} \in \boldsymbol{U}$, then the control input can be as given in (5.44).

$$
\boldsymbol{u} = \frac{1}{L_g h(\boldsymbol{x})}\left(-L_f h(\boldsymbol{x}) + \boldsymbol{v}\right) \tag{5.44}
$$

By doing so, the relation between the input and the output is linearized.

$$
\dot{\boldsymbol{y}} = \boldsymbol{v} \tag{5.45}
$$

When the above assumption $(L_g h(\boldsymbol{x}) \neq \boldsymbol{0})$ is not satisfied, and $L_g h(\boldsymbol{x}) = \boldsymbol{0}$ for all $\boldsymbol{x} \in \boldsymbol{U}$, the output is differentiated once more.

$$
\ddot{\boldsymbol{y}} = \frac{\partial \dot{\boldsymbol{y}}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial L_f h}{\partial \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\partial L_f h}{\partial \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{u} \triangleq L_f^2 h(\boldsymbol{x}) + L_f L_g h(\boldsymbol{x})\boldsymbol{u} \tag{5.46}
$$

Subsequently, if $L_g L_f h(\boldsymbol{x})$ is not close to zero for all $\boldsymbol{x} \in \boldsymbol{U}$, the input is put into the form illustrated in (5.47),

$$
\boldsymbol{u} = \frac{1}{L_g L_f h(\boldsymbol{x})}\left(-L_f^2 h(\boldsymbol{x}) + \boldsymbol{v}\right) \tag{5.47}
$$

which renders the input-output relation linear.

$$\ddot{\boldsymbol{y}} = \boldsymbol{v} \tag{5.48}$$

The system is said to have *strict relative degree*, γ, at $\boldsymbol{x}_0$ when (5.49) is satisfied [44].

$$\begin{aligned} L_g L_f^k h(\boldsymbol{x}) &= 0 \\ L_g L_f^{\gamma-1} h(\boldsymbol{x}_0) &\neq 0 \;\; \forall \boldsymbol{x} \in \boldsymbol{U}, \;\; k = 0, \ldots, \gamma - 2. \end{aligned} \tag{5.49}$$

Then, the control law is established by the following equation:

$$\boldsymbol{u} = \frac{1}{L_g L_f^{\gamma-1} h(\boldsymbol{x})} \left(-L_f^{\gamma} h(\boldsymbol{x}) + \boldsymbol{v} \right) \tag{5.50}$$

which results in the input-output relation expressed in (5.51).

$$\boldsymbol{y}^{\gamma} = \boldsymbol{v} \tag{5.51}$$

Recall the mathematical model of the vessel given in the following equations,

$$\begin{aligned} \dot{\eta} &= \boldsymbol{J}_{\Theta}(\eta)\, \boldsymbol{v} \\ M\dot{\boldsymbol{v}} + C(\boldsymbol{v})\boldsymbol{v} &= \tau_d + \tau_g + \tau_t + \tau_a + \tau_c \end{aligned}$$

where

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

The above dynamics can also be put into the form described in (5.42). The state vector including positions, orientations in Earth fixed frame and linear, angular velocities in the body fixed frame is in $\mathbb{R}^{12}$. The input vector, $\boldsymbol{u}$, containing left and right thruster forces is in $\mathbb{R}^2$. Besides, the output vector, $\boldsymbol{y}$, which is selected to comprise of yaw position and surge velocity is also in $\mathbb{R}^2$. The system is multi-input, multi-output (MIMO) where the outputs are driven to the desired states with the help of two inputs. Fortunately, the theory of feedback linearization for square systems including as many outputs as inputs is well established as revealed by [44].

Square systems can be represented in the following form,

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= f(\boldsymbol{x}) + g_1(\boldsymbol{x})u_1 + \cdots + g_l(\boldsymbol{x})u_l \\
y_1 &= h_1(\boldsymbol{x}) \\
&\vdots \\
y_l &= h_l(\boldsymbol{x})
\end{aligned}
\tag{5.52}
$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^l$, $\boldsymbol{y} \in \mathbb{R}^l$, and f, g_i are smooth vector fields, and h functions are also smooth.

Take the derivative of i^{th} output with respect to time:

$$
\dot{y}_i = L_f h_i + \sum_{k=1}^{l} (L_{g_k} h_i)\, u_k \tag{5.53}
$$

If $L_{g_k} h_i(\boldsymbol{x}) = 0$ for all k, the inputs of the system do not affect the derivative of i^{th} output.

Thereafter, γ_i is to be introduced. It is a variable similar to the strict relative degree in SISO case such that at least one of the inputs influences $y_i^{\gamma_i}$ as shown in (5.54).

$$
y_i^{\gamma_i} = L_f^{\gamma_i} h_i + \sum_{k=1}^{l} L_{g_k} \left(L_f^{\gamma_i - 1} h_i \right) u_k \tag{5.54}
$$

where the summation on the right hand side of (5.54) is not identically equal to zero.

At that point, let us define the matrix $\boldsymbol{A}(\boldsymbol{x})$ as:

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} L_{g_1} L_f^{\gamma_1 - 1} h_1 & \dots & L_{g_l} L_f^{\gamma_1 - 1} h_1 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{\gamma_l - 1} h_l & \dots & L_{g_l} L_f^{\gamma_l - 1} h_l \end{bmatrix} \tag{5.55}$$

Hereby, the definition of *vector relative degree* is introduced.

The system has relative degree $(\gamma_1, \gamma_2, \dots, \gamma_l)$ at $\boldsymbol{x}_0$ if

$$L_{g_k} L_f^m h_i(\boldsymbol{x}) \equiv 0, \;\; 0 \le m \le \gamma_i - 2, \tag{5.56}$$

for $k = 1, \dots, l$, and $\boldsymbol{A}(\boldsymbol{x}_0)$ is nonsingular.

Therefore, we can write a system with a definite relative degree vector as given in (5.57).

$$\begin{bmatrix} y_1^{\gamma_1} \\ \vdots \\ y_l^{\gamma_l} \end{bmatrix} = \begin{bmatrix} L_f^{\gamma_1} h_1 \\ \vdots \\ L_f^{\gamma_l} h_l \end{bmatrix} + \boldsymbol{A}(\boldsymbol{x}) \begin{bmatrix} u_1 \\ \vdots \\ u_l \end{bmatrix} \tag{5.57}$$

The nonsingular property of $\boldsymbol{A}(x)$ is already guarantied for the neighborhood of $\boldsymbol{x}_0$. As a result, the control input for (5.57) can now be designated in a similar fashion with Equation (5.50).

$$\boldsymbol{u} = -\boldsymbol{A}^{-1}(\boldsymbol{x}) \begin{bmatrix} L_f^{\gamma_1} h_1 \\ \vdots \\ L_f^{\gamma_l} h_l \end{bmatrix} + \boldsymbol{A}^{-1}(\boldsymbol{x}) \boldsymbol{v} \tag{5.58}$$

The control law given in (5.58) achieves the following closed loop dynamics:

$$\begin{bmatrix} y_1^{\gamma_1} \\ \vdots \\ y_l^{\gamma_l} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_l \end{bmatrix} \tag{5.59}$$

At this point, linearization of the system by a static-state feedback control law is accomplished. Furthermore, it should be noted that the outputs of the system are all decoupled. After that, well known linear controller techniques, such as pole placement, LQR or PID can be designed by using (5.59).

If the relative degree is less than the order of the system (i.e., $(\gamma_1, \gamma_2, \ldots, \gamma_l) < n$), then the control law is not able to take some of the states into account which are called *internal dynamics*. In other words, some part of the system is caused to be unobservable [45]. In this case, there are several methods in the literature to investigate the behavior of the system. For instance, one can put the system into a special structure, namely *normal form*, which decouples the internal and external dynamics. It is a way to take a formal look at the internal dynamics and guarantee the stability of the system during the operation of the controller. Additionally, once the input control law is decided, the nonlinear equations can be directly tested for stability purposes. The latter method is much simpler than the former one since normal form requires solving (n-r) partial differential equations to figure the corresponding state transformation out [45].

Finally, if the stability of the internal dynamics is verified, it means that an applicable solution to the control problem is accomplished.

5.3.2 Implementation of the FL Controller

As asserted in earlier sections, the ultimate goal of the autopilot is to drive the yaw position and surge velocity to the desired values. Henceforth, the output of the system is chosen as in (5.60).

$$\boldsymbol{y} = \begin{bmatrix} \psi \\ u \end{bmatrix} \tag{5.60}$$

Afterwards, referring to Equation (5.42), $h(x)$ can be written in the following form:

$$\boldsymbol{y} = h(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x} \tag{5.61}$$

where

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.62}$$

$$\boldsymbol{x} = \begin{bmatrix} \eta \\ \boldsymbol{v} \end{bmatrix} \tag{5.63}$$

The system dynamics can also be put into form of (5.42) by the necessary manipulations.

$$\begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\Theta}(\eta)\boldsymbol{v} \\ \boldsymbol{M}^{-1}\left[\tau_d + \tau_g + \tau_a + \tau_c - \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}\right] \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{M}^{-1}\tau_t \end{bmatrix} \tag{5.64}$$

It is seen that $\boldsymbol{M}^{-1}\tau_t$ is a linear, constant function of left and right forces applied by the thrusters. Therefore, it is rewritten as:

$$\boldsymbol{M}^{-1}\tau_t = \boldsymbol{G}_t\boldsymbol{u} \tag{5.65}$$

where $\boldsymbol{G}_t$ is a 6-by-2 constant matrix, and $\boldsymbol{u}$ is the input vector whose elements are the forces exerted onto body by left and right thrusters in order.

Finally, (5.64) takes the following form:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\Theta}(\eta)\boldsymbol{v} \\ \boldsymbol{M}^{-1}\left[\tau_d + \tau_g + \tau_a + \tau_c - \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}\right] \end{bmatrix} + \boldsymbol{G}\boldsymbol{u} \tag{5.66}$$

where $\boldsymbol{G} = \begin{bmatrix} 0 \\ \boldsymbol{G}_t \end{bmatrix}$ is a 12-by-2 fixed matrix, and let $\boldsymbol{G} = [\ \boldsymbol{g}_1 \quad \boldsymbol{g}_2\ ]$. $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are first and second columns of $\boldsymbol{G}$, respectively.

The system is achieved to be represented in the form of (5.42).

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$$

$$\text{where} \quad \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\Theta}(\eta)\boldsymbol{v} \\ \boldsymbol{M}^{-1}\left[\tau_d + \tau_g + \tau_a + \tau_c - \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}\right] \end{bmatrix} \ \textit{and} \ \ g(\boldsymbol{x}) = \boldsymbol{G} \tag{5.67}$$

From now on, the implementation of the feedback linearization controller will be based on the representation given by (5.67).

By taking Lie derivatives of the output function, the following observations are made:

(i) $L_{g_1}\psi \equiv 0$, $L_{g_2}\psi \equiv 0$ meaning that the input vector does not appear in the first derivative of ψ,

(ii) $L_{g_1}L_f\psi \neq 0$, $L_{g_2}L_f\psi \neq 0$,

(iii) $L_{g_1}L_f u \neq 0$, $L_{g_2}L_f u \neq 0$,

In other words, the relative degrees of ψ and u are found to be 2 and 1, respectively ($\gamma_1 = 2$, $\gamma_2 = 1$).

Eventually, Equations (5.68) and (5.69) declare the output dynamics.

$$\begin{bmatrix} \ddot{\psi} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} L_f^2\psi \\ L_f u \end{bmatrix} + \boldsymbol{A}(\boldsymbol{x})\boldsymbol{u} \tag{5.68}$$

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} L_{g_1}L_f\psi & L_{g_2}L_f\psi \\ L_{g_1}u & L_{g_2}u \end{bmatrix} \tag{5.69}$$

Thus, the control law can be selected as:

$$\boldsymbol{u} = -\boldsymbol{A}^{-1}(\boldsymbol{x}) \begin{bmatrix} L_f^2\psi \\ L_f u \end{bmatrix} + \boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{v} \tag{5.70}$$

which brings about the following relation.

$$\begin{bmatrix} \ddot{\psi} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{5.71}$$

Note that

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} 0.073\frac{\cos(\phi)}{\cos(\theta)} + 0.046\frac{\sin(\phi)}{\cos(\theta)} & 0.046\frac{\sin(\phi)}{\cos(\theta)} - 0.073\frac{\cos(\phi)}{\cos(\theta)} \\ 0.024 & 0.024 \end{bmatrix} \tag{5.72}$$

which is invertible for the region $\boldsymbol{R} = \left(-\frac{\pi}{2} < \phi < \frac{\pi}{2}, -\frac{\pi}{2} < \theta < \frac{\pi}{2}\right)$. Meanwhile, the given system equations are also not valid out of $\boldsymbol{R}$ since the vessel is not able to swim, and tends to sink down.

To conclude, the problem is reduced to control a plant, consisting of a double integrator and an integrator, with two inputs.

It should be noted that since the relative degrees of the plant are less than the order of the system, there will inevitably emerge internal dynamics that are unobservable. These states can be examined by representing the system in normal form or the nonlinear system with the determined input control law can initially be linearized by Jacobian linearization and tested by the mathematical tools utilized for the analysis of stability of linear systems.

5.4 Simulation Results

In this section, the responses of the close loop systems with LQR and FL controllers to various step reference signals are examined. The reference signals are applied after being filtered by a low pass filter with 4.5 second rise time and 6% overshoot. It should be noted that the outputs of the controllers, which correspond to the forces that will be applied by the thrusters, are limited in accordance with the characteristics of the motors.

Throughout the simulations, the vessel model constructed in Mathematical Modeling

Chapter is employed. The outputs of the controller are fed to the model, then the resulting positions, velocities and attitudes are conveyed back to the controller as the feedback signal.

The results of three different tests are shared in this section. For the first case, denoted by Ref_1, the magnitudes of the step commands are $\frac{\pi}{4}$ rad and 1 m/sec for yaw angle and surge speed, respectively, as illustrated in Figures 5.1, 5.2 and 5.3. In the second test, symbolized by Ref_2, the magnitudes of the commands are determined to be $\frac{\pi}{2}$ rad for yaw angle and 0.5 m/sec for surge speed. Figures 5.4, 5.5 and 5.6 present details of this instance. In the last case, Ref_1 is combined with the disturbance forces depicted in Figure 5.10. As can be seen from the figure, the disturbances are essentially applied in surge and sway directions although a relatively small moment around z-axis emerges. Figures 5.7, 5.8 and 5.9 reveal the details of this condition. In addition, the RMS yaw position and surge speed errors for the three cases with different controllers are represented in Tables 5.1 and 5.2, respectively.

At first, the time responses of the controllers can be compared. The results with FL controller are observed to be enhanced in terms of overshoot and rise time for all of the simulations. Moreover, the maximum difference between the reference and output signals is much smaller for FL controller. The steady-state errors converge to zero for all of the cases even though the system with LQR cannot exactly reach zero steady-state error in yaw position for one side of the motion as depicted in Figure 5.1. This is interpreted to be resulted from the lack of LQR to notice the nonlinear characteristics of the plant since it is founded on the linearized model. On the other hand, the settling times are calculated to be close. Meanwhile, the improved performance of FL is not originated from the maximum amount of obtained thrusts. As Figure 5.3 and 5.6 indicate, the highest levels of forces produced in both cases are quite similar. FL also exhibits better disturbance rejection performance as shown in Figures 5.7 and 5.8. The disturbance forces basically degrade the surge speed tracking performances as the moment component perturbing yaw position is determined to be relatively weaker.

FL is again preferred to LQR by taking the RMS errors into consideration. For three of the cases, RMS errors for FL are much smaller. The increasing magnitude of the step command is confirmed to result in the degradation in terms of the relative criteria.

The results arise as expected since FL controller is based on the plant model which is exactly same with the one used for the tests. In other words, it is assumed that the motion of the marine vessel is precisely modeled for the feedback linearized system, unlike the LQR controlled case. Given that the model of the vessel is already identified by the experimentations, the assumption appears to be reasonable. However, one should be aware of the fact that any discrepancy between the model and the real hardware may associate with robustness issues since FL directly exploits the presumed system model.

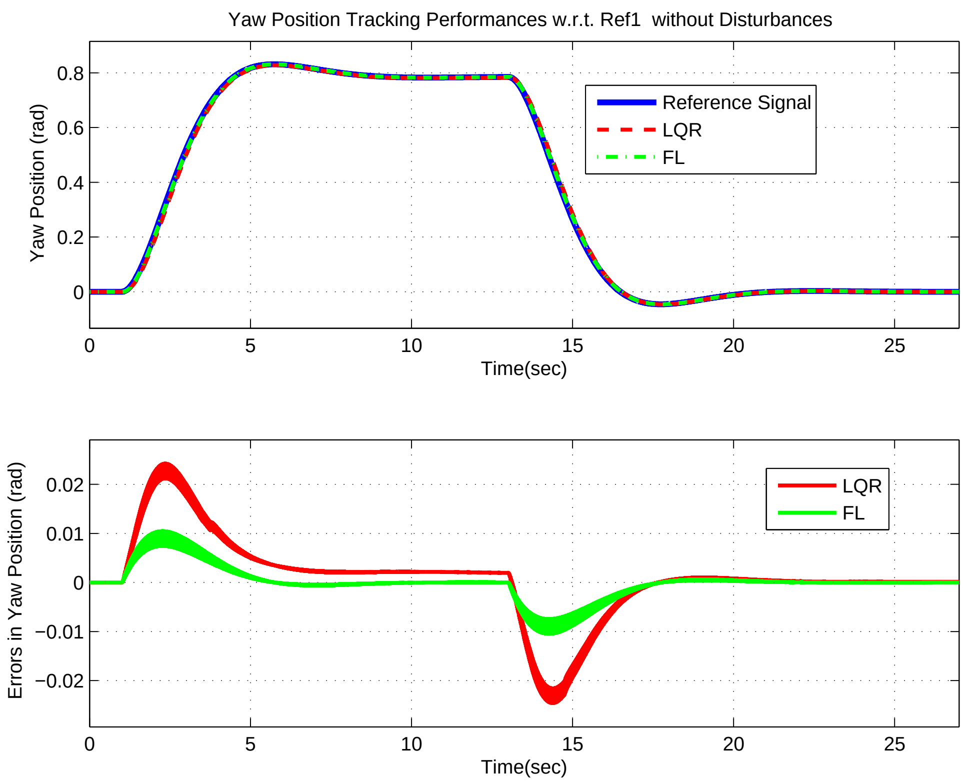


Figure 5.1: Yaw position performances of the controllers in Ref_1 without disturbances.

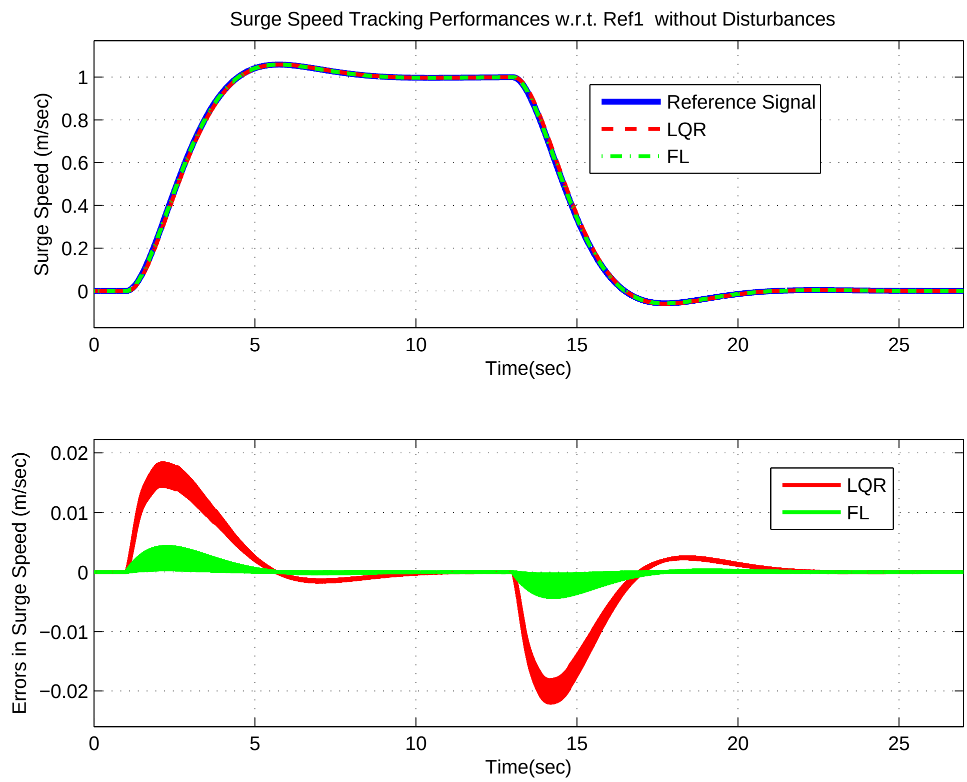


Figure 5.2: Surge speed performances of the controllers in Ref_1 without disturbances.

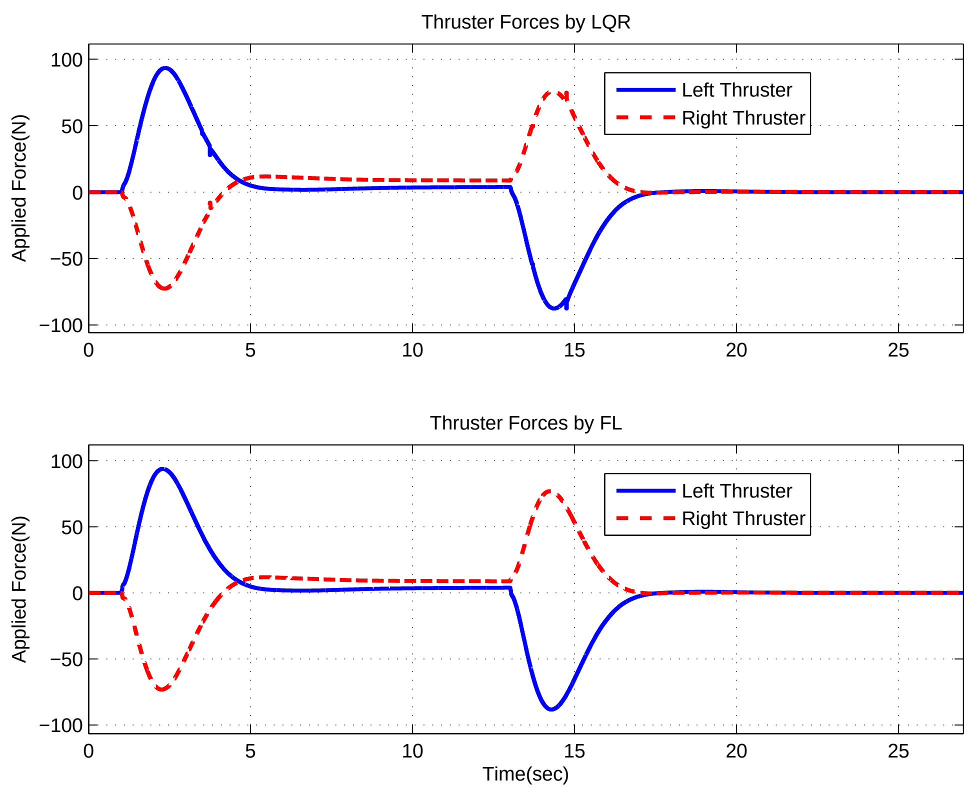


Figure 5.3: Thruster commands of the controllers in Ref_1 without disturbances.

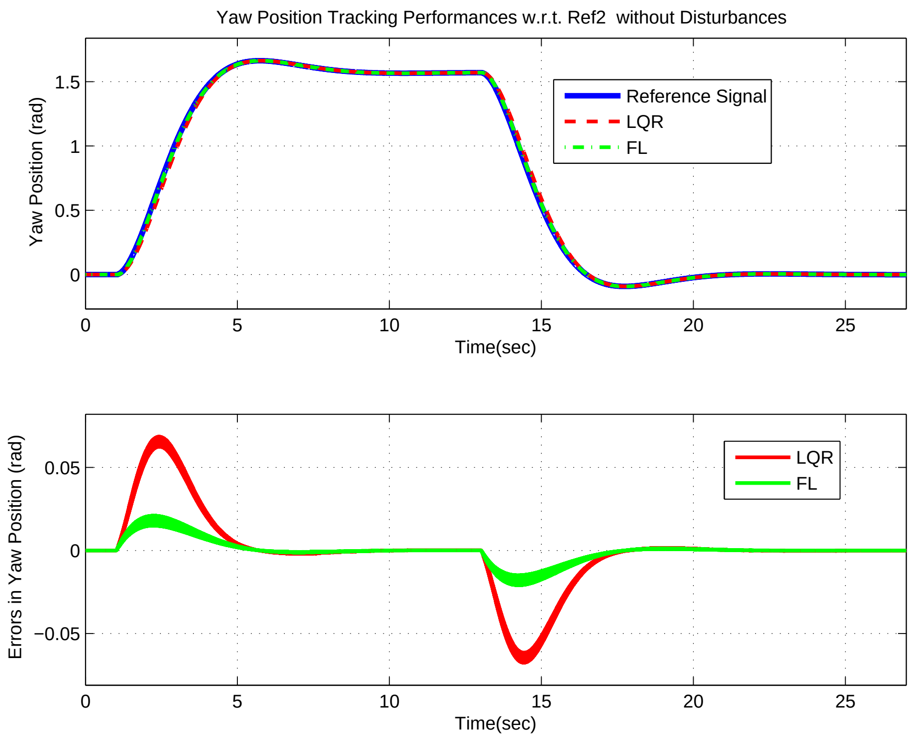


Figure 5.4: Yaw position performances of the controllers in Ref_2 without disturbances.

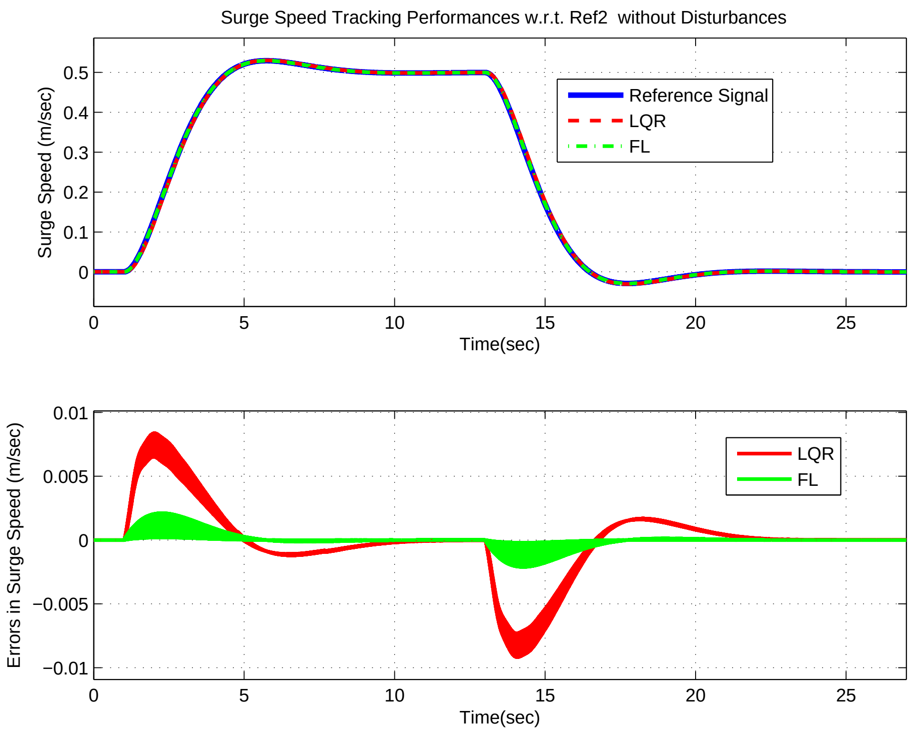


Figure 5.5: Surge speed performances of the controllers in Ref_2 without disturbances.

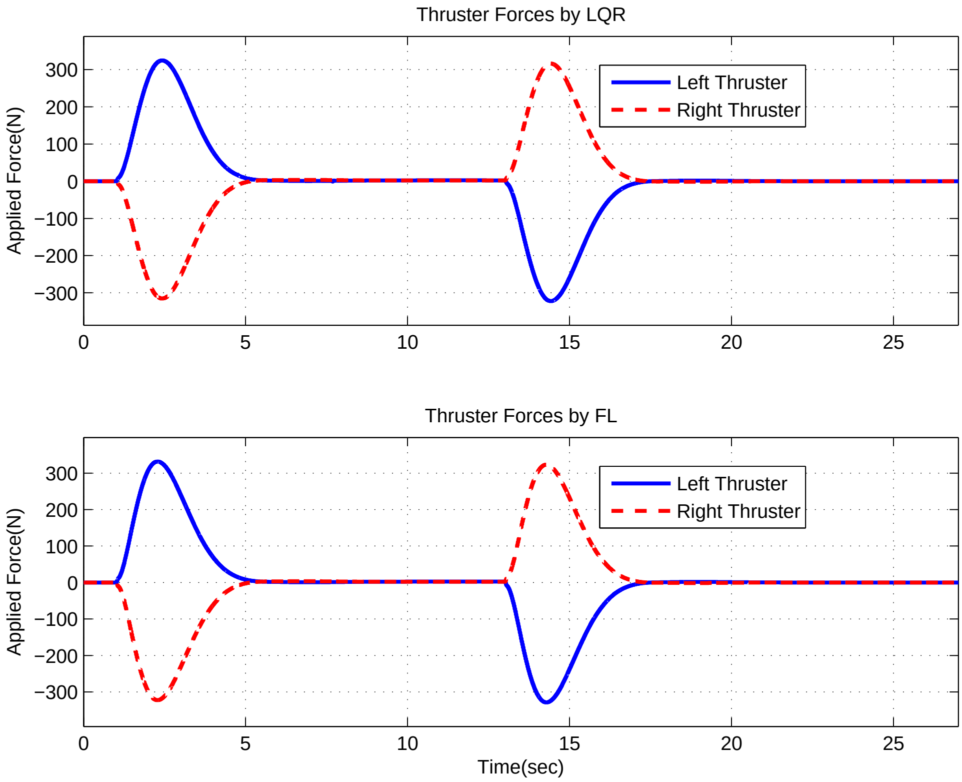


Figure 5.6: Thruster commands of the controllers in Ref_2 without disturbances.

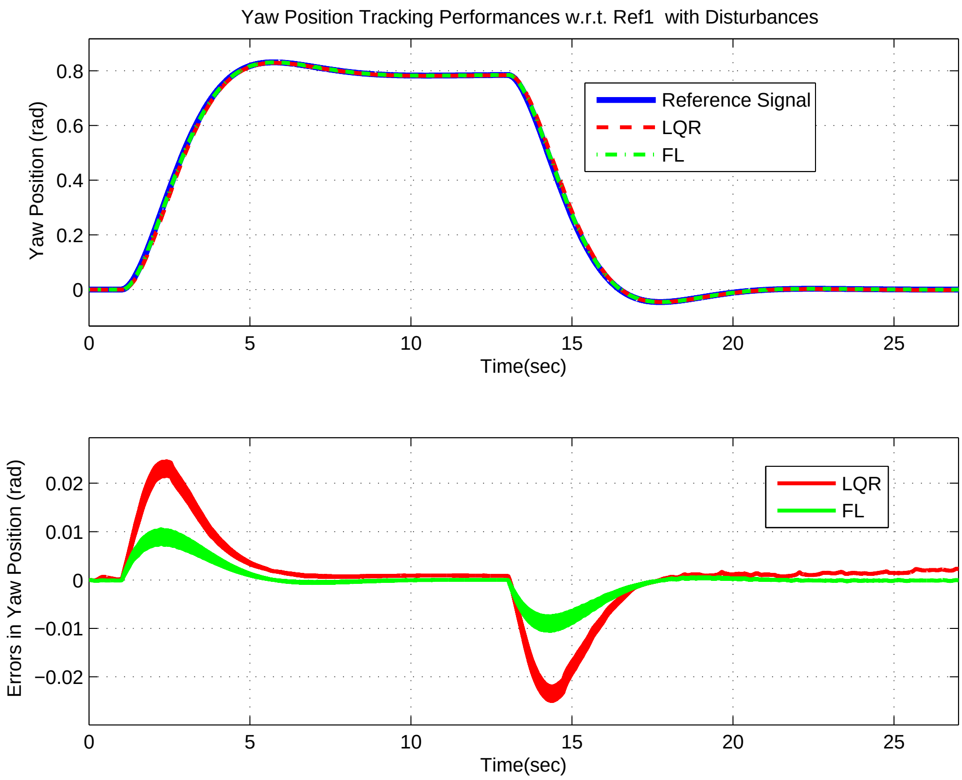


Figure 5.7: Yaw position performances of the controllers in Ref_1 with disturbances.

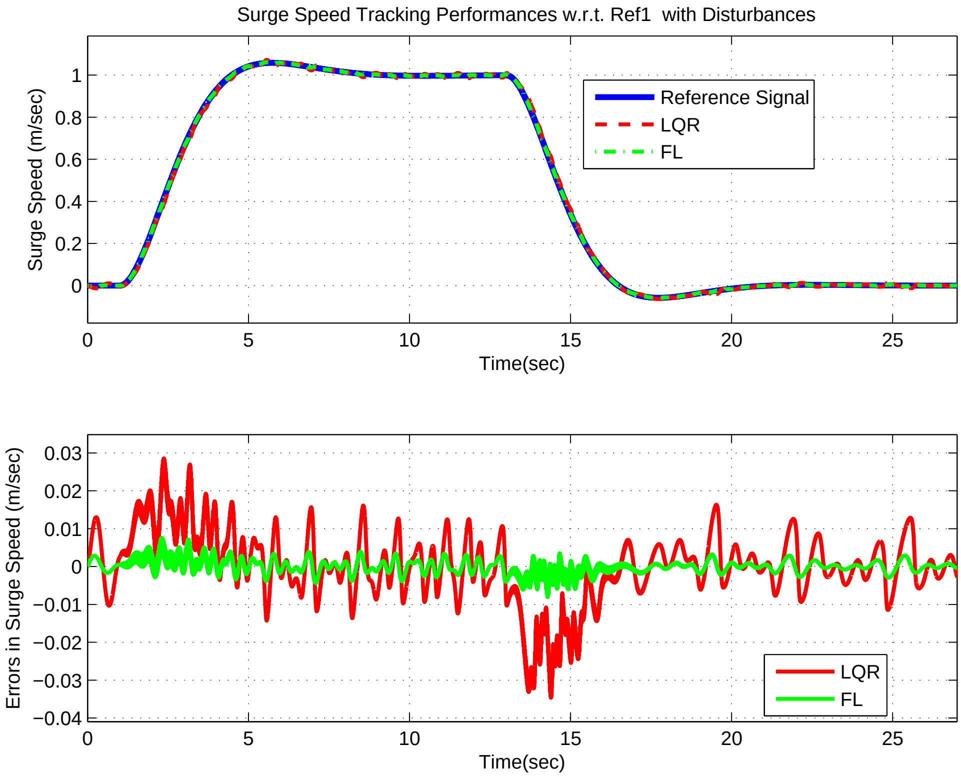


Figure 5.8: Surge speed performances of the controllers in Ref_1 with disturbances.

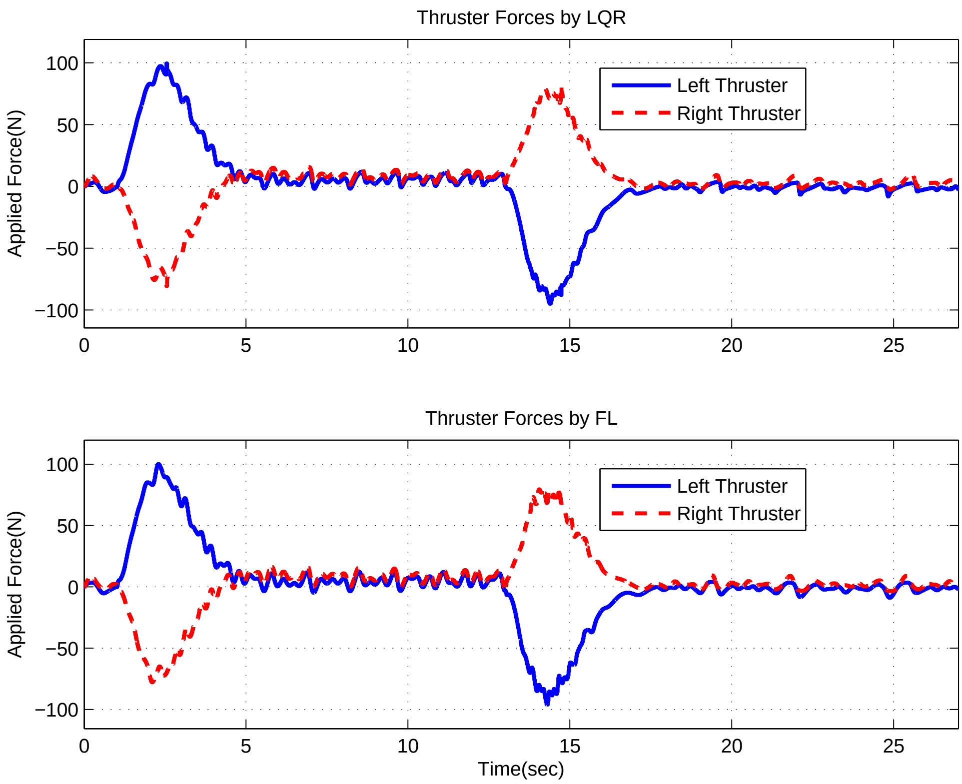


Figure 5.9: Thruster commands of the controllers in Ref_1 with disturbances.

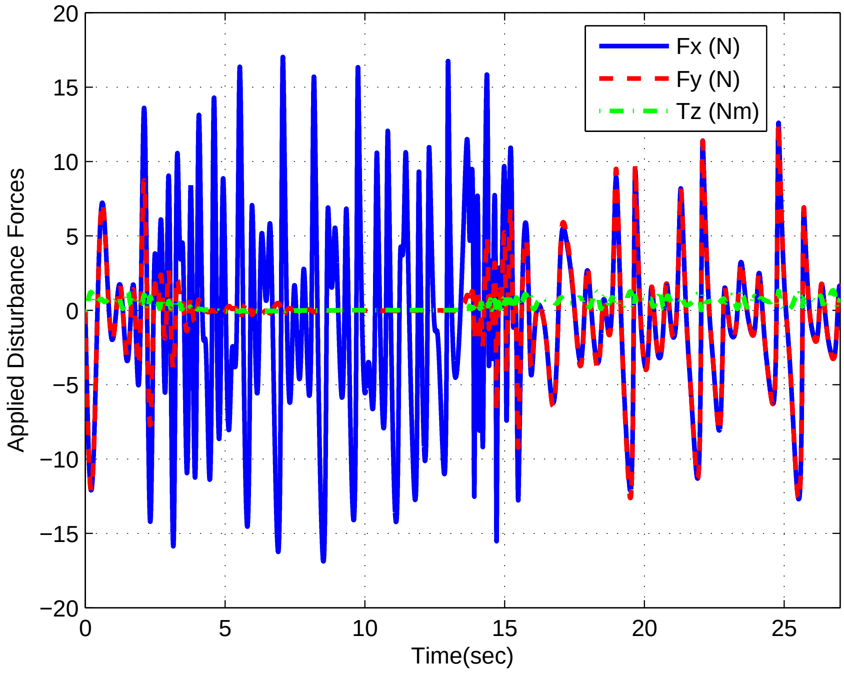


Figure 5.10: Disturbance forces exerted on the body.

Table 5.1: RMS yaw position errors.

$RMS\ Err \times 10^3$	Ref_1	Ref_2	$Ref_1 + Dist$
LQR	8.08	22.12	8.11
FL	3.37	6.37	3.37

Table 5.2: RMS surge speed errors.

$RMS\ Err \times 10^3$	Ref_1	Ref_2	$Ref_1 + Dist$
LQR	6.74	2.81	9.25
FL	0.10	0.045	1.81

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The work contained in this thesis essentially represents the investigation and development of navigation and autopilot algorithms for sea surface vehicles. To that end, the mathematical model of a marine craft is examined in the beginning of the study.

In the wake of a comprehensive literature review, the vectorial model proposed by Fossen [2] is decided to be based on. Meanwhile, possible simplifications and modifications are suggested with regard to the model boat that will be utilized in the experimental tests.

A number of necessary mathematical tools, such as rigid body dynamics, coordinate systems and transformations, added mass dynamics, hydrodynamic damping, restoring, thruster and air drag forces are discussed to be able to construct the model. The equations expressing the motion of the vessel are founded on Newton's laws of motion, while they comprise of separate vectors. Consequently, the implementation tasks are also performed in a modular structure to render the model user/debug friendly and maintainable.

After the mathematical model is constructed, various navigation algorithms are considered to provide a reliable basis for system identification and autopilot applications for surface vehicles. For this purpose, three distinct integrated navigation algorithms making use of the solutions of INS, GNSS and an external attitude estimation system are developed. Subsequently, comparative tests with synthetic data sets, which are generated regarding the characteristics of the sensors used during experimentation, are conducted between these methods. The technique executing the integration of the mentioned three navigation solutions in a loosely coupled, closed loop fashion is observed to exhibit the best performance in terms of the RMS position, attitude and velocity errors. As a result, the measurements collected from the model boat are also processed by this algorithm to obtain the navigation solution, and the outcomes are illustrated.

Next, to be able to bridge the gap between the theory and practice, a model boat, namely Pacific Islander Tugboat, is modified with the addition of necessary electrical and mechanical equipments to be employed in the experiments. Thereafter, numerous experiments are carried out, and the results, which are treated by the navigation algorithm, are analyzed within a parallel study by Erünsal [6] aiming to identify the parameters of the boat included in the mathematical model. In the end of this phase, a model taking the parameters with improved accuracy into account is attained to be exploited in the autopilot design.

In the end, two autopilot algorithms based on LQR and FL controllers are studied to realize the controlled motion of the vessel. In the design procedure of LQR controllers, the mathematical model is firstly linearized at various linearization points, and the controllable subspaces of the corresponding representations are acquired. Then, the optimal control theory is applied to calculate the parameters of LQR controllers. On the other hand, FL controller makes use of the original nonlinear system model. Reference tracking performances for yaw angle/surge speed and disturbance rejection characteristics of the controllers are investigated by simulations, and the results are presented. For all of the tests, FL is confirmed to have higher performance compared to LQR according to the RMS errors and time response characteristics.

6.2 Future Work

In this study, the autopilot algorithms are only tested in simulation environment. Consequently, further investigations need to be carried out in order to validate the performances of LQR and FL controllers using the same experimental setup utilized in this work. These could leave room for further research about the possible combinations of controllers or applications of more sophisticated ones.

A tightly coupled integrated navigation algorithm making use of raw GNSS range measurements may also be constructed, and the results can be compared with the current performance in a further study.

It would also be very interesting to render the surface vehicle fully autonomous by designing guidance and path planning algorithms with obstacle avoidance feature.

Meanwhile, two more Pacific Islander Tugboats are already ordered, and under construction at present. These boats will be delivered in the forthcoming months. Hence, an experimental setup for investigating the coordination of multiple surface vehicles which are assigned missions such as, reconnaissance, detection and object identification will be possible.

REFERENCES

[1] SNAME. The Society of Naval Architects and Marine Engineers. *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*. 1950.

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

[3] P.D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2nd edition, 2013.

[4] M.S. Grewal, R.W. Weill, and A.P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley, 2nd edition, 2007.

[5] Pacific Islander Tugboat. `https://scalespeed.com`. Accessed: 2014-10-08.

[6] I.K. Erünsal. System Identification and Control of an Unmanned Sea Surface Vehicle. Master's thesis, Middle East Technical University, September 2015.

[7] Pixhawk Autopilot Main Page. `https://pixhawk.org/start`. Accessed: 2015-08-24.

[8] Pixhawk Peripherals. `http://copter.ardupilot.com/wiki/common-pixhawk-wiring-and-quick-start/`. Accessed: 2015-08-24.

[9] Structure of a MAVLink Message. `http://qgroundcontrol.org/mavlink/start`. Accessed: 2015-08-23.

[10] S.J. Corfield and J.M. Young. Unmanned Surface Vehicles- Game Changing Technology for Naval Operations. *IEEE Control Engineering Series*, 69, 2006.

[11] M. Caccia, M. Bibuli, R. Bono, and G. Bruzzone. Basic Navigation, Guidance and Control of an Unmanned Surface Vehicle. *Autonomous Robots*, 25(4):349–365, 2008.

[12] A.W. Browning. *A Mathematical Model to Simulate Small Boat Behaviour*. PhD thesis, Bournemouth Polytechnic, October 1990.

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

[14] T.I. Fossen. *Nonlinear Modeling and Control of Underwater Vehicles*. PhD thesis, Norwegian University of Science and Technology, 1991.

[15] J.J. Craig. *Introduction to Robotics*. Prentice-Hall, 3rd edition, 2005.

[16] S.I. Sagatun and T.I. Fossen. Lagrangian Formulation of Underwater Vehicles' Dynamics. *Proceedings of the IEEE International Conference on Systems*, 1991.

[17] T.I. Fossen and O.E. Fjellstad. Nonlinear Modelling of Marine Vehicles in 6 Degrees of Freedom. *Journal of Mathematical Modelling of Systems*, 1(1):17–27, 1995.

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

[19] S.P. Berge and T.I. Fossen. On the Properties of the Nonlinear Ship Equations of Motion. *Journal of Mathematical and Computer Modeling of Dynamical Systems*, 6(4):365–381, 2000.

[20] J. Larson, M. Bruch, and J. Ebken. Autonomous Navigation and Obstacle Avoidance for Unmanned Surface Vehicles. In *Defense and Security Symposium*. International Society for Optics and Photonics, 2006.

[21] L. Paull, S. Saeedi, M. Seto, and H. Li. AUV Navigation and Localization: A Review. *Oceanic Engineering, IEEE Journal of*, 39(1):131–149, 2014.

[22] Y. Liu, W. Liu, R. Song, and R. Bucknall. Predictive Navigation of Unmanned Surface Vehicles in a Dynamic Maritime Environment When Using the Fast Marching Method. *International Journal of Adaptive Control and Signal Processing*, 2015.

[23] W. Naeem, R. Sutton, and T. Xu. An Integrated Multi-Sensor Data Fusion Algorithm and Autopilot Implementation in an Uninhabited Surface Craft. *Ocean Engineering*, 39:43–52, 2012.

[24] N. Minorsky. Directional Stability of Automatically Steered Bodies. *Journal of ASNE*, 42(2):280–309, 1922.

[25] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[26] T. Koyama. On the Optimum Automatic Steering System of Ships at Sea. *J.S.N.A*, 122:18–35, 1967.

[27] N.H. Norrbin. On the Added Resistance Due to Steering on a Straight Course. *Proceedings of the 13th ITTC*, 1972.

[28] D.C. Donha, D.S. Desanj, M.R. Katebi, and M.J. Grimble. H_∞ Adaptive Controllers for Auto-pilot Applications. *International Journal of Adaptive Control and Signal Processing*, 12(8):623–648, 1998.

[29] K.D. Do, Z.P. Jiang, and J Pan. Universal Controllers for Stabilization and Tracking of Underactuated Ships. *Systems and Control Letters*, 47(4):299–317, 2002.

[30] J. Guo, Chiu F.C., and C.C. Huang. Design of a Sliding Mode Fuzzy Controller for the Guidance and Control of an Autonomous Underwater Vehicle. *Ocean Engineering*, 30(16):2137–2155, 2003.

[31] Z. Zwierzewicz. The Design of Ship Autopilot by Applying Observer-Based Feedback Linearization. *Polish Maritime Research*, 22(1):16–21, 2015.

[32] K. Ahıska. Control and Guidance of an Unmanned Sea Surface Vehicle. Master's thesis, Middle East Technical University, September 2012.

[33] A. Noureldin, T.B. Karamat, and J. Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and Their Integration*. Springer, 2013.

[34] D. Titterton and John L. H. Weston. *Strapdown Inertial Navigation Technology*. The Instution of Electrical Engineers, 2nd edition, 2004.

[35] Anon. *Department of Defense World Geodetic System 1984*. National Imagery and Mapping Agency, 3rd edition, 1997.

[36] J.A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. The McGraw-Hill, 2008.

[37] Pixhawk Attitude Estimator. `https://pixhawk.org/firmware/apps/attitude_estimator_ekf`. Accessed: 2015-08-15.

[38] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[39] J. Ryu and J.C. Gerdes. Integrating Inertial Sensors With Global Positioning System (GPS) for Vehicle Dynamics Control. *Journal of Dynamic Systems, Measurement, and Control*, 126(2):243, 2004.

[40] MAVLink Definition. `https://en.wikipedia.org/wiki/MAVLink`. Accessed: 2015-08-25.

[41] D.E. Kirk. *Optimal Control Theory: An Introduction*. Dover, 1970.

[42] H.H. Rosenbrock. *State Space and Multivariable Theory*. John Wiley, 1970.

[43] R.E. Kalman. Contributions to the Theory of Optimal Control, 1960.

[44] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[45] J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.