Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Brage Sæther

Development and Testing of Navigation and Motion Control Systems for milliAmpere

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Breivik, Edmund Førland Brekke, Egil Eide
June 2019

NTNU
Norwegian University of
Science and Technology

Brage Sæther

Development and Testing of Navigation and Motion Control Systems for milliAmpere

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Breivik, Edmund Førland Brekke, Egil Eide
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Preface

This thesis is the culmination of my work in the five-years integrated MSc program in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). The work with the thesis was performed during the spring of 2019 under the supervision of Morten Breivik, Edmund Førland Brekke and Egil Eide. Through the entirety of the thesis, I have been aided by valuable discussions and insights to be applied in my work. I would also like to thank Torleiv H. Bryne and Erik F. Wilthil for insightful knowledge about the error-state Kalman filter (ESKF) and its implementation. I would like to thank my fellow graduate students who have been valuable partners in discussions about topics relevant to my thesis. Lastly, I would like to thank my family for all the support throughout the years.

The thesis is linked to the vessel milliAmpere, which is a prototype for an autonomous passenger ferry for use in urban waterways. The need for a new, improved guidance, navigation and control (GNC) system on the vessel served as motivation. The thesis is aimed at an audience with previous knowledge in topics in control theory and estimation theory. The algorithms used in the thesis rely heavily on knowledge of the feedback loop, Lyapunov theory, and the Kalman filter, among others. Additionally, practical experience on how to apply them in practice is preferred. The thesis assumes knowledge in linear algebra and modelling of stochastic noise and their distributions. Even though prior knowledge is expected, the thesis introduces concepts before applying them, providing the possibility to follow along without extensive previous experience with the algorithms that are being implemented.

The main contribution of this thesis is the design and implementation of a new navigation system and motion control system. The navigation system performs sensor fusion of the available sensor data, and the new motion control system uses an adaptive backstepping control approach. The new implementation uses the existing guidance system and thrust allocation. The system is implemented in Python, with the additional use of the Robot Operating System framework. Additionally, the library "numpy" is used to enhance the innate abilities for the programming language for use towards linear algebra. MATLAB is used for the analysis of the data obtained. The implementations have been tested, verified and validated through testing in simulated environments and an experimental environment.

The work on the adaptive control is based on an on-line paper co-authored by my supervisor Morten Breivik. The navigation system is based on recent papers available on-line. Bi-weekly meetings with the advisers were held during the entirety of the thesis. The meetings sought to serve as follow-up meetings where recent progress and issues were discussed, and as a means for general guidance with my work.

The bumpy time-consuming road of implementing the ESKF was aided by guidance from Edmund F. Brekke, Erik F. Wilthil and Torleiv H. Bryne. Their advice consisted of dif-

ferent suggestions to deal with debugging issues. Their guidance helped me overcome the implementation issues I faced. The rest of the work has been independent through the entirety of the thesis. The previous working experience I have on the experimental platform helped significantly during the experimental tests. The well-rounded prior knowledge of the platform, in conjunction with earlier involvement with the software development, made it possible to obtain the presented results during both the simulations and the experimental tests with no outside help.

Brage Sæther
June 10th, 2019
Trondheim, Norway

Summary

Marine vessels operating in a confined area demands a robust and accurate navigation and control system. This thesis presents the implementations of a navigation system and a motion control system. The implementations in the thesis explore two state of the art methods in their respective fields and their implementations on an experimental platform.

In the implementation of the new motion control system, the $\mathcal{L}_1$ adaptive control approach utilizes adaptive backstepping for faster convergence of the unmodeled dynamics. The latest motion control approach is coupled with an ESKF that performs sensor fusion and state estimation. Together these approaches serve as an upgrade over the prior systems on the experimental platform milliAmpere.

The new navigation system and motion control system are both tested through computer-simulated environments and experimental tests in the harbor basin by Brattøra in Trondheim. The new implementations use the existing implementations on the experimental platform for a comparison of performance. Various performance metrics ensure an objective assessment of performance between the new and old implementations.

The new implementations are tested individually at first to get a grasp of how they differ in performance in comparison to the prior system. Afterwards, they are joined together to show the performance with their combined efforts. The new implementations outperforms the prior solution. The new motion controller greatly increases the ability to estimate the unmodeled dynamics, and the new navigation system successfully eliminates several unwanted characteristics present in the previous implementation. The new implementations especially improves the directional stability of the vessel. Lastly, final experimental tests with a collision avoidance module, which is developed by a fellow student, are successfully carried out with virtual objects. The experimental tests with the use of the collision avoidance system mark the complete upgrade of the new GNC system.

Sammendrag

Marine fartøy som opererer i trange omgivelser krever et robust og nøyaktig navigasjonssystem og kontrollsystem. Denne oppgaven presenterer nye implementasjoner av navigasjonssystem og kontrollsystem i detalj. Implementasjonene utforsker det siste og beste innen hver av sine respektive fagområder, samt implementasjon av disse på en eksperimentell plattform.

Det nye kontrollsystemet tar i bruk $\mathcal{L}_1$ adaptiv kontroll basert på adaptiv backstepping for raskere konvergeringstid for estimeringen av dynamikk som ikke er modellert. Det nye kontrollsystemet fungerer sammen med et ESKF som utfører sensor fusjon og tilstandsestimering. Disse systemene fungerer som oppgraderinger av det gamle navigasjonssystemet og kontrollsystemet på den eksperimentelle plattformen milliAmpère.

Det nye navigasjonssystemet og kontrollsystemet er testet i simulatorer og gjennom eksperimentelle tester i havnebassenget ved Brattøra i Trondheim. De nye implementasjonene bruker de eksisterende systemene for å foreta sammenligninger i ytelse. En rekke ytelses metrikker er tatt i bruk for en objektiv evaluering av de ulike implementasjonene.

De nye systemene er testet både individuelt og samlet for å skaffe et bilde av hvilke konsekvenser de ulike systemene har på overordnet ytelse. De nye implementasjonene fungerer bedre enn de eksisterende implementasjonene. Det nye kontroll systemet øker evnen til å estimere dynamikk som ikke er modellert betraktelig, og det nye navigasjonssystemet fjerner flere uønskede egenskaper som var tilstede i det gamle navigasjonssystemet. Spesielt oppleves det en økning i retningsstabilitet som en konsekvens av de nye systemene. Til slutt er de nye systemene sydd sammen med en kollisjonsunngåelsesmodul som er utviklet av en felles student. Suksessfulle full skala tester med disse tre systemene som jobber sammen er utført med virtuelle objekter. Disse testene markerer en fullstendig oppgradering av planlegging, navigasjon og kontroll systemet på den eksperimentelle plattformen.

Table of Contents

List of Tables

List of Figures

Abbreviations

BLDC brushless direct current. 57

BODY body fixed. i, xi, xii, xxii, 9, 11–15, 31, 32, 37, 38, 46, 60, 83, 110, 129

CG center of gravity. 15, 85, 87

CLF control Lyapunov function. 24–26

CO center of origin. xi, 13, 15, 31, 38, 53, 56, 73

COG course over ground. 49

COLAV collision avoidance. vii, xv, 6, 122, 123, 126, 128, 129, 133–140

DOF degree of freedom. vi, xii, xiii, 9, 12–16, 21, 22, 28, 29, 31, 33, 36, 38, 42, 45, 58, 62, 63, 71, 73, 74, 76, 78, 80–82, 89, 112

DP dynamic positioning. 20, 29

ECEF earth-centered earth-fixed. xxii, 9–11, 43

EOM equations of motion. 13, 42, 45

ESKF error-state Kalman filter. i, iii, vii, xii, xiv, xv, 6, 37–39, 42, 43, 45, 46, 54, 65, 67, 69, 70, 74, 83, 85, 87–89, 92, 98, 101, 104, 110, 115–124, 126–128, 130, 153, 154

FOG fiber optic gyros. 37

GNC guidance, navigation and control. i, iii, vi, vii, xi, 5, 6, 16, 17, 19, 35, 41, 43, 60, 64, 83, 120, 123, 124, 126–128, 154

GNSS global navigation satellite system. vi, xi, 5, 6, 10, 36, 37, 39–42, 49, 50, 53, 56, 58, 65, 70, 72, 73, 110, 127, 128

GPS global positioning system. 40

GUI graphical user interface. 58, 171

HMI human-machine interface. 128

I/O input/output. 57

IAE integral of the absolute error. 75

IAEW integral of the absolute error multiplied by energy consumption. 75, 113, 114

IMU inertial measurement unit. vii, xiii, xiv, 5, 6, 36–40, 42, 46, 47, 49, 53, 56, 58, 65, 85–87, 92, 98, 101, 104–108, 110, 126–130, 154

INS inertial navigation system. 54

IR infrared. 57

ISE integral of the square of the error. 75

ITAE integral of the absolute error multiplied by time. 75

L1-AB $\mathcal{L}_1$ adaptive backstepping. 81

LiDAR light detection and ranging. vii, 57, 128

MEMS microelectromechanical systems. 37, 46

MRAC model reference adaptive control. 22

NED north-east-down. xi, xxii, 9–11, 13–15, 29, 36, 41, 46, 75, 83, 110

NEES normalized estimation error squared. xii, 66, 67, 69, 71, 73

NIS normalized innovation squared. xiii, xiv, 66, 67, 93, 98, 103–105

NTNU Norwegian University of Science and Technology. i

OBC on board computer. 63, 74

PD proportional–derivative. vii, 30, 76, 81, 112, 114, 115, 155

PID proportional–integral–derivative. vi, vii, xii, xiv, xv, 19–21, 28, 30, 33, 58, 62, 74, 76, 77, 79, 81, 82, 111–116, 125–127

QP quadratic programming. 32

RADAR radio detection and ranging. 57

RLG ring laser gyros. 37

ROS Robot Operating System. 74, 88, 157

RPM rotations per minute. 120

RTK real time kinematics. xi, 36, 40, 41, 58

SOG speed over ground. 49

UGAS uniformly globally asymptotically stable. 25, 27

WGS84 World Geodetic System 1984. 36

XKF3i Xsens Kalman filter. 92, 98, 108

Nomenclature

$\approx$	Approximately
$\boldsymbol{\Delta}$	Relative damping factor matrix
$\boldsymbol{\eta}$	State vector containing position and angles
$\boldsymbol{\nu}$	State vector for linear and angular velocities
$\boldsymbol{\Omega}$	Natural frequency matrix
$\boldsymbol{\omega}$	Vector containing angular velocities
$\boldsymbol{\tau}$	Force and moment vector
$\boldsymbol{\Theta}$	Attitude expressed by Euler angles
$\boldsymbol{b}$	Disturbance vector
$\boldsymbol{e}$	Error between actual state and desired state
$\boldsymbol{N}\{0, \sigma\}$	Gaussian distribution with standard deviation σ
$\boldsymbol{T}^{\dagger}$	Pseudoinverse of $\mathbf{T}$
Δt	timestep
$\in (\cdot)$	Defined in set $(\cdot)$
$[\cdot]_{\times}$	Skew symmetric matrix of $(\cdot)$
$\mathbb{R}^{3x3}$	3x3 matrix with real numbers
$\dot{\mathbf{x}}$	$\mathbf{x}$ differentiated with regards to time
$\hat{\mathbf{x}}$	Estimated $\mathbf{x}$
$\tilde{\mathbf{x}}$	Error between estimated and actual $\mathbf{x}$
$\mathbf{C}$	Coriolis-centripetal matrix

$\mathbf{C}_A$	Added mass coriolis-centripetal matrix
$\mathbf{D}_n$	Non-linear damping matrix
$\mathbf{D}_v$	Linear damping Matrix
$\mathbf{I}$	Identity matrix
$\mathbf{M}$	Inertia matrix
$\mathbf{M}_a$	Added mass matrix
$\mathbf{p}$	Position vector
$\mathbf{r}$	Vector containing lever arm between frames
$\mathbf{R}_b^n$	Rotation matrix
$\mathbf{S}(\cdot)$	Skew symmetric matrix of vector $(\cdot)$
$\mathbf{v}$	Linear velocity vector
$\mathbf{x}(\cdot)$	Matrix depending on vector $(\cdot)$
$\mathbf{x}^*$	True value of $\mathbf{x}$
$\mathbf{x}^{-1}$	Inverse of $\mathbf{x}$
$\oplus$	General composition
$\otimes$	Quaternion product
ϕ, θ, ψ	Euler angles about the x, y and z axes
$\times$	Cross product
$\{b\}$	BODY frame
$\{d\}$	Reference vector
$\{e\}$	Earth-centered earth-fixed coordinate frame
$\{m\}$	Sensor measurement frame
$\{n\}$	NED frame
p, q, r	Angular velocity about the x, y and z axes
$s\psi, c\psi$	Sine and cosine of ψ
diag$\{\cdot\}$	Diagonal matrix with diagonal entries given by vector $(\cdot)$
K, N, M	Moments around the x, y and z axes in the BODY frame
u, v, w	Velocities along the x, y and z axes in the BODY frame
X, Y, Z	Forces along x, y and z axes in the BODY frame
x, y, z	Position

Chapter 1

Introduction

1.1 Motivation

A drastic increase in automation has taken place during the last decades. The introduction of everything from industrial robots on assembly lines, to small automated lawnmowers, have seen the light of day. Lately, autonomous systems able to solve more complex tasks in more complex environments have emerged. The emergence of more complex autonomous systems can serve as a significant advance in personal transportation.

Figure 1.1: Concept for the final version of the autonomous passenger ferry intended to operate in the channel in Trondheim. Image courtesy of [Mustvedt, 2019].

There is an ever-increasing number of cars and buses on the roads. Bigger cities especially see an increasing problem with traffic. Autonomous passenger ferries can help revitalize the urban waterways and, alleviate the pressure on the city roads. Other areas can also benefit significantly from the use of autonomous passenger ferries. Such a solution can serve as an alternative mode of transportation across rivers. These ferries will easily tie together communities, and different parts of cities separated by water. The realization of a digitized ferryman will contribute towards the possibility of an intelligent, synchronous transportation network consisting of autonomous cars, buses, trains, and marine vessels.

Many older solutions such as cable ferries or a ferryman have traditionally had such a role. The main idea behind milliAmpere is to provide an inexpensive, environmentally friendly, and non-intrusive solution to this problem. By delivering the same logistical service as bridges, smaller ports, harbors, and canals are kept open and provides the essential waterways for large and small vessels alike. The channel between Fosenkaia and Ravnkloa in Trondheim is the intended area for the final iteration of the ferry to operate.

Figure 1.2: Map of the area for the intended use of the autonomous passenger ferry. Image courtesy of Egil Eide with Google Maps as base.

This service has in Trondheim historically been carried out by a local ferryman in a rowing boat. The project focuses on the continued development of the prototype milliAmpere.

The finished ferry is to serve as an on-demand solution. Its main goal is to provide passengers with a smooth, fast, and reliable transportation option across urban waterways. The full-sized vessel will have the capability to transport 12 passengers at any given time, as well as bikes, strollers and similar equipment. A scalable solution of this kind ensures an easily deployable solution where it is needed. Not only in Norway is this solution desirable, but it can be used all over the world. The development of milliAmpere also brings with it substantial attention, given its autonomous nature. A ferry of this kind will also provide the city of Trondheim with a new attraction of a unique kind.

1.2 Problem Description

The main objective of this thesis is to research and develop an improved GNC system for milliAmpere. The thesis researches and develops a new navigation system based on GNSS and IMU sensor fusion, and it explores a new adaptive motion control system for improved tracking performance. The guidance system is left as is in the thesis. However, the new GNC system needs to be compatible with a collision avoidance module developed in [Thyri, 2019]. The problem description summarizes in the bullet points as:

- Literature study for state estimation using GNSS and IMU.
- Literature study for adaptive motion control.
- Design and implement an observer for real time sensor fusion of IMU and GNSS sensor data.
- Design and implement a robust adaptive motion control system to account for modelling uncertainties and external disturbances.
- Test, verify and validate the system through simulations and experiments.
- Implement an interface such that the low-level GNC system can be interfaced with a high-level COLAV system.

1.3 Contributions

The project during the fall of 2018 explored the potential use of multiple controllers to remedy some of the shortcomings of the implementation at that time. The results did not pan out as intended. However, the lessons learned during the project were valuable. Consequently, this thesis pursues other ideas not related to the project. The main contributions for this thesis summarize as

- Designed and implemented an ESKF for real time sensor fusion of GNSS and IMU sensor data.
- Designed and implemented an adaptive motion control system using the $\mathcal{L}_1$ adaptive control approach.

- Verified the new navigation and control systems through extensive tests in a simulated environment.
- Validated the new navigation and control systems through experimental testing.
- Implemented software for a stacking beacon to ease experimental testing.
- Ensure integration of the navigation and motion control systems with a COLAV module.

1.4 Outline

This thesis organizes its contents in chapters and sections. Chapter 2 explores the vessel modelling. This chapter will focus mostly on kinematic and kinetic details for describing a marine vessel in motion. The use of correct notation in mathematical modelling is essential. This chapter also discusses the notation used throughout the thesis.

Chapter 3 explains the details surrounding the new navigation system. This chapter goes into detail about sensor preliminaries, the proposed observer, and other nuances to the newly implemented navigation system.

Chapter 4 examines the different motion control algorithms and their respective approaches. The chapter describes all the theory necessary for the implementation of the controllers. Further, the theory that forms the basis of the thrust allocation is detailed here.

Chapter 5 will elaborate more on the experimental platform. Previous work and previous test results are detailed. The chapter also presents the hardware used on milliAmpere. Chapters 6 and 7 goes more in-depth to present, analyze and discuss the results from the simulated environment and the experimental tests. Finally, Chapter 8 will wrap up and conclude the thesis.

Chapter 2

Modelling

This chapter describes a six DOF mathematical modelling of a marine surface vessel. Both the kinematics and kinetics are described in detail to obtain an accurate vessel model. The chapter also establishes the conventional notation and the reduced three DOF vessel model for further use in the motion control system.

2.1 Kinematics

When analyzing the motion of a vessel, it is important to differentiate between different reference frames. The different measurements are given in a specific frame, and it is important to transform the measurement to the correct frame before utilizing its measurement in the model. The different frames need to be separated, and the relationship between them established. The frames of interest are the earth-centered earth-fixed (ECEF) frame given in coordinate system $\{e\}$, NED frame given in coordinate system $\{n\}$ and BODY frame given in coordinate system $\{b\}$.

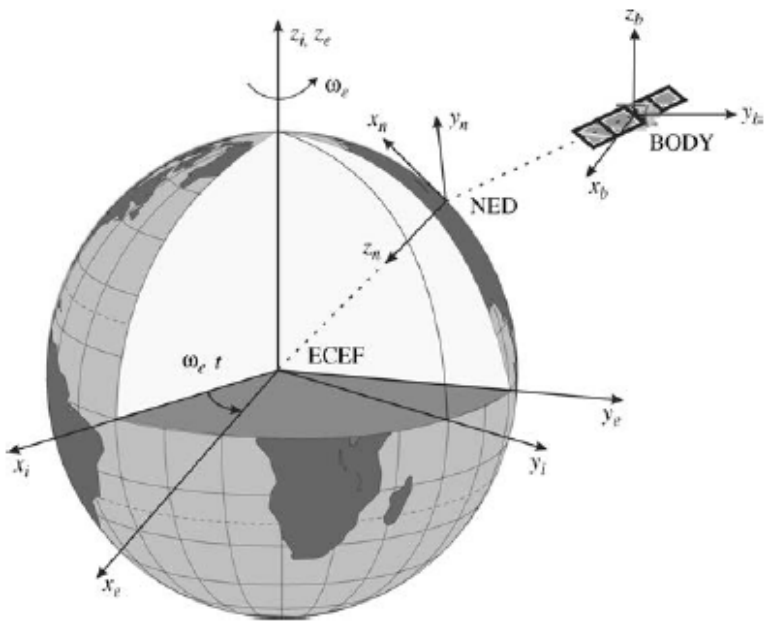


Figure 2.1: Relationship between different relevant reference frames used in the navigation and motion control systems. Courtesy of [Fossen, 2011].

2.1.1 ECEF frame

The ECEF frame is a fixed inertial frame with the origin at the Earth's center [Fossen, 2011]. Given that it is an inertial frame, Newton's second law of motion applies. The GNSS gives it's coordinates relative to the ECEF frame. The coordinates given are in the standard longitude, latitude, and altitude format. By using a transformation, the ECEF frame is transformed into the NED frame.

2.1.2 NED frame

The NED frame describes a tangent plane to the Earth's surface. The components of the NED frame is given as $[x_n, y_n, z_n]^\mathsf{T}$. The coordinates denote the position in the true north, true east and down respectively. The NED frame is a relative frame with origin in Earth's reference ellipsoid defined by [WGS84]. For vessels operating in a small area with small deviations in the longitude and latitude coordinates, the NED frame representation is simplified to be inertial. This simplification refers to the flat Earth approximation. This simplification ensures the validity of Newton's laws of motion in the NED frame.

The transformation between the ECEF frame and the NED frame is described in [Fossen, 2011] as

$$\dot{\mathbf{p}}^e_{b/e} = \mathbf{R}^e_n(\mathbf{\Theta}_{\text{en}})\dot{\mathbf{p}}^n_{b/e} = \mathbf{R}^e_n(\mathbf{\Theta}_{\text{en}})\mathbf{R}^n_b(\mathbf{\Theta}_{\text{nb}})\mathbf{v}^b_{b/e} \tag{2.1}$$

where $\mathbf{\Theta}_{\text{en}}$ is defined as $[l, \mu]^\mathsf{T} \in \boldsymbol{S^2}$ by longitude l and latitude μ. This means that $\mathbf{S^2} \rightarrow SO(3)$ is given by the rotation matrix $\mathbf{R}^e_n(\mathbf{\Theta}_{\text{en}})$. This rotation matrix is given in [Fossen, 2011] and is defined as

$$\mathbf{R}^e_n(\mathbf{\Theta}_{\text{en}}) = \mathbf{R}_{z,l}\mathbf{R}_{y,-\mu-\frac{\pi}{2}} \tag{2.2a}$$

$$\mathbf{R}^e_n(\mathbf{\Theta}_{\text{en}}) = \begin{bmatrix} -\cos(l)\sin(\mu) & -\sin(l) & -\cos(l)\cos(\mu) \\ -\sin(l)\sin(\mu) & \cos(l) & -\sin(l)\cos(\mu) \\ \cos(\mu) & 0 & -\sin(\mu) \end{bmatrix} \tag{2.2b}$$

2.1.3 BODY frame

The BODY frame is the transformation from the NED frame about the Euler angles given by roll, pitch, and yaw. For simplicity, these are from here on out referred to as ϕ, θ and ψ. The rotation matrix about the Euler angles is given in [Fossen, 2011] as

$$\mathbf{R}^n_b(\mathbf{\Theta}_{\text{nb}}) = \begin{bmatrix} \text{c}\psi\text{c}\theta & \text{-s}\psi\text{c}\phi + \text{c}\psi\text{s}\theta\text{s}\phi & \text{s}\psi\text{s}\phi + \text{c}\psi\text{c}\phi\text{s}\theta \\ \text{s}\psi\text{c}\theta & \text{c}\psi\text{c}\phi + \text{s}\phi\text{s}\theta\text{s}\psi & -\text{c}\psi\text{s}\phi + \text{s}\theta\text{s}\psi\text{c}\phi \\ -\text{s}\theta & \text{c}\theta\text{s}\phi & \text{c}\theta\text{c}\phi \end{bmatrix} \tag{2.3}$$

where $\mathbf{R}^n_b(\mathbf{\Theta}_{\text{nb}})$ denotes the rotation $\mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\phi}$, c = cos and s = sin. These angles represent the angular position of a vessel. A visualization of their derivatives, $[\dot{\phi}, \dot{\theta}, \dot{\psi}] = [p, q, r]$, is shown in Figure 2.2.

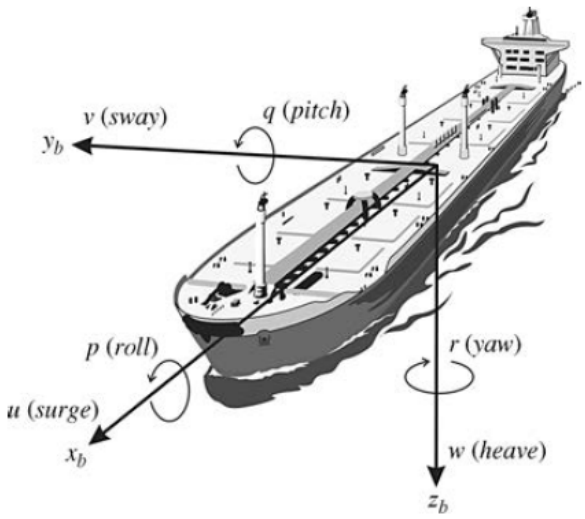


Figure 2.2: Relationships between the Euler angles and the relations for angular velocities around their respective axes. Courtesy of [Fossen, 2011].

This rotation refers to the BODY frame. In the BODY frame, the six motion components are described using the notations defined in [SNAME]. The motions and rotations are given by Table 2.1.

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

Table 2.1: Motion and rotation notation for marine vessels defined in [SNAME]

Figure 2.3 shows the notation for different velocities and different rotations and the various relationships in the BODY frame.

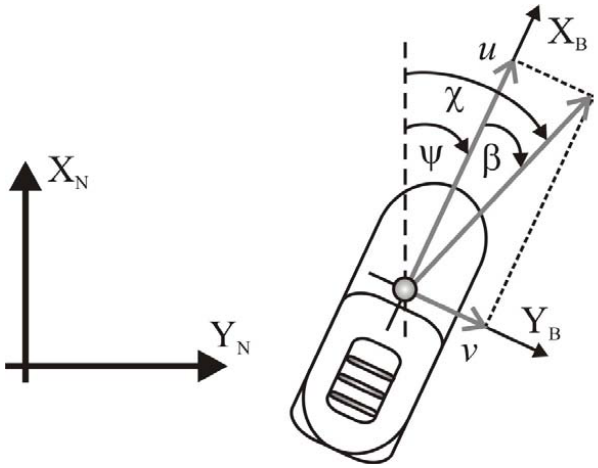


Figure 2.3: The BODY frame denoted by subscript b, on a marine vessel in relation to the NED frame denoted by subscript n. χ and ψ are the course and heading of the vessel, u and v denotes the surge and sway, and β is the sideslip. Figure courtesy of [Breivik, 2010].

When describing the equations of motion (EOM) for a marine vessel, proper vector notation is needed. In [Fossen, 2011], the six DOF rigid-body EOM according to the [SNAME] notation, is written in component form as

$$\begin{aligned}
\boldsymbol{f}_b^b &= [X, Y, Z]^\mathsf{T} && \text{– force through } o_b \text{ expressed in } \{b\} \\
\boldsymbol{m}_b^b &= [K, M, N]^\mathsf{T} && \text{– moment about } o_b \text{ expressed in } \{b\} \\
\boldsymbol{v}_{b/n}^b &= [u, v, w]^\mathsf{T} && \text{– linear velocity of } o_b \text{ relative } o_n \text{ expressed in } \{b\} \\
\boldsymbol{w}_{b/n}^b &= [p, q, r]^\mathsf{T} && \text{– angular velocity of } \{b\} \text{ relative to } \{n\} \text{ expressed in } \{b\} \\
\boldsymbol{r}_o^b &= [x_g, y_g, z_g]^\mathsf{T} && \text{– vector from } o_b \text{ to CO expressed in } \{b\} \\
\boldsymbol{p}_{n/b}^n &= [N, E, D]^\mathsf{T} && \text{– distance from } \{n\} \text{ to } \{b\} \text{ expressed in } \{n\} \\
\boldsymbol{\Theta}_{nb} &= [\phi, \theta, \psi]^\mathsf{T} && \text{– Euler angles between } \{n\} \text{ and } \{b\}
\end{aligned}$$

The six DOF kinematic equations for a vessel in motion are expressed in [Fossen, 2011] as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.4a}$$

or equivalently

$$\begin{bmatrix} \dot{\boldsymbol{p}}^n_{b/n} \\ \dot{\boldsymbol{\Theta}}_{nb} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^n_b(\boldsymbol{\Theta}_{nb}) & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{T}_\Theta(\boldsymbol{\Theta_{nb}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^b_{b/n} \\ \boldsymbol{\omega}^b_{b/n} \end{bmatrix} \tag{2.4b}$$

where $\boldsymbol{p}^n_{b/n} = [x, y, z]^\mathsf{T}$ and $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\mathsf{T}$ are the positions and Euler angles respectively, and $\boldsymbol{J}_\Theta$ is the transformation matrix between the BODY and NED frame.

For a marine vessel in calm waters, the assumption is that the ϕ and θ angles are negligible. This assumption reduces the transformation between NED and BODY frame to the rotation around the z-axis with the angle ψ. For a marine surface vessel in calm waters, z is also negligible. As a result, the surface vessel restricts its motion to horizontal motion and rotation around the z-axis. These motions refer to surge, sway, and yaw. The simplification of neglecting heave, roll, and pitch simplifies the model of the vessel to movement in three DOF. As a result, the state vector reduces to three DOF defined by the transformation matrix

$$\mathbf{J}_\Theta(\boldsymbol{\eta}) = \mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

The kinematic equation for the three DOF system becomes

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.6}$$

where the state vectors are reduced to $\boldsymbol{\eta} = [x, y, \psi]^\mathsf{T}$ and $\boldsymbol{\nu} = [u, v, r]^\mathsf{T}$.

2.2 Kinetics

A six DOF system is described with the kinetics in [Fossen, 2011] as

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) + \mathbf{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.7}$$

where

$$\mathbf{M} = \underbrace{\begin{bmatrix} m\mathbf{I}_{3x3} & m\mathbf{S}(\mathbf{r}^b_g) \\ m\mathbf{S}(\mathbf{r}^b_g) & \mathbf{I}_b \end{bmatrix}}_{\mathbf{M}_{RB}} + \mathbf{M}_A \tag{2.8a}$$

$$\mathbf{C} = \underbrace{\begin{bmatrix} m\mathbf{S}\left(v_2\right) & -m\mathbf{S}\left(v_2\right)\mathbf{S}\left(r^b_g\right) \\ m\mathbf{S}\left(r^b_g\right)\mathbf{S}\left(v_2\right) & -\mathbf{S}\left(I_b v_2\right) \end{bmatrix}}_{\mathbf{C}_{RB}(\boldsymbol{\nu})} + \mathbf{C}_A(\boldsymbol{\nu}) \tag{2.8b}$$

$$\mathbf{D} = \mathbf{D}_l(\boldsymbol{\nu}) + \mathbf{D}_n(\boldsymbol{\nu}) \tag{2.8c}$$

where $\mathbf{M}$, $\mathbf{C}(\boldsymbol{\nu})$, and $\mathbf{D}(\boldsymbol{\nu})$ are the inertia, coriolis and centripetal, and damping matrices respectively, and $\mathbf{g}(\boldsymbol{\eta})$ denote the generalized gravitation and buoyancy forces. Static restoring forces and moments are collected in the term $\mathbf{g}_0$. The system matrices satisfies the properties $\mathbf{M} > 0$, $\mathbf{D} > 0$ and $\mathbf{C} = -\mathbf{C}^\mathsf{T}$. The property of $\mathbf{C}$ refers to the skew-symmetric property of a matrix.

The three DOF model describes the motions in surge, sway and yaw. The states of interest are the North-East-Heading positions given by $\boldsymbol{\eta} = [N, E, \psi]^\mathsf{T}$ in the NED frame, and velocities surge, sway and yaw described by $\boldsymbol{\nu} = [u, v, r]^\mathsf{T}$ in the BODY frame. The complete kinematic and kinetic model for a marine vessel in three DOF becomes

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.9a}$$

$$\mathbf{M}\dot{\nu} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.9b}$$

where $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$, $\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu}$, and $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{D}_l\boldsymbol{\nu} + \mathbf{D}_n(\boldsymbol{\nu})\boldsymbol{\nu}$. The element wise definition of the reduced model matrices in three DOF are shown in (2.10) - (2.17). The inertia matrix $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ is

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{2.10}$$

and

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{r}} \\ -Y_{\dot{u}} & -Y_{\dot{v}} & -Y_{\dot{r}} \\ -N_{\dot{u}} & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{2.11}$$

The assumption for the vessel modelling is that the CO and the center of gravity (CG) coincide such that $x_g = 0$. Coupled with this the assumption, the parameters in (2.10) and (2.11) are joined together to form the matrix

$$\mathbf{M}_A = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{2.12}$$

The centripetal matrix $\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu})$ is given by

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \tag{2.13}$$

and

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{r}}r + Y_{\dot{v}}v \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{r}}r - Y_{\dot{v}}v & X_{\dot{u}}u & 0 \end{bmatrix} \tag{2.14}$$

Lastly, the definition of the damping matrix $\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D}_l + \mathbf{D}_n(\boldsymbol{\nu})$ with the linear damping matrix

$$\mathbf{D}_l = \begin{bmatrix} -X_u & -X_v & -X_r \\ -Y_u & -Y_v & -Y_r \\ -N_u & -N_v & -N_r \end{bmatrix} \tag{2.15}$$

and non-linear damping matrix

$$\mathbf{D}_n(\boldsymbol{\nu}) = \begin{bmatrix} D_{11}(\boldsymbol{\nu}) & 0 & 0 \\ 0 & D_{22}(\boldsymbol{\nu}) & D_{23}(\boldsymbol{\nu}) \\ 0 & D_{33}(\boldsymbol{\nu}) & D_{32}(\boldsymbol{\nu}) \end{bmatrix} \tag{2.16}$$

where the individual entries in the non-linear damping matrix are given by

$$D_{11}(\boldsymbol{\nu}) = -X_u - X_{|u|u}|u| - X_{uuu}u^2 \tag{2.17a}$$
$$D_{22}(\boldsymbol{\nu}) = -Y_v - Y_{|v|v}|v| - Y_{vvv}v^2 \tag{2.17b}$$
$$D_{23}(\boldsymbol{\nu}) = -Y_v - Y_{|v|v}|v| - Y_{vvv}v^2 \tag{2.17c}$$
$$D_{32}(\boldsymbol{\nu}) = -N_{|v|v}|v| - N_{|r|v}|r| \tag{2.17d}$$
$$D_{33}(\boldsymbol{\nu}) = -N_r - N_{|r|r}|r| - N_{rrr}r^2. \tag{2.17e}$$

This thesis uses the model parameters from [Pedersen, 2019]. Appendix F further details the model parameters. The simulated environment and the model-based control in the new motion control system use the model parameters.

2.3 Chapter Summary

This chapter outlined the fundamental theory for accurate vessel modelling of a marine surface vessel. The chapter discussed the transformations between the relevant reference frames and its applications in the vessel modelling. Roll angle, pitch angle and heave position are negligible such that the six DOF vessel model was reduced to the simplified three DOF model. The three DOF kinematic model is coupled with the vessel kinetics to form an accurate, complete vessel model for use in the GNC system.

Chapter 3

Navigation

A navigation system is a system that uses measured sensor data and performs data processing to utilize the data for motion control. The navigation system has the critical role of providing the motion control system with reliable state estimates. The navigation system uses different techniques such as wild point filtering to discard erroneous sensor information. The navigation system typically uses a state observer to minimize the stochastic noise present in the sensor measurements and to estimate the entire state vector. The navigation system is an essential part of the GNC system and provides the state information needed to perform tracking operations. The circled portion of Figure 3.1 shows the elements making up the navigation system, along with the measurements supplied by the sensors available, and the estimated states.

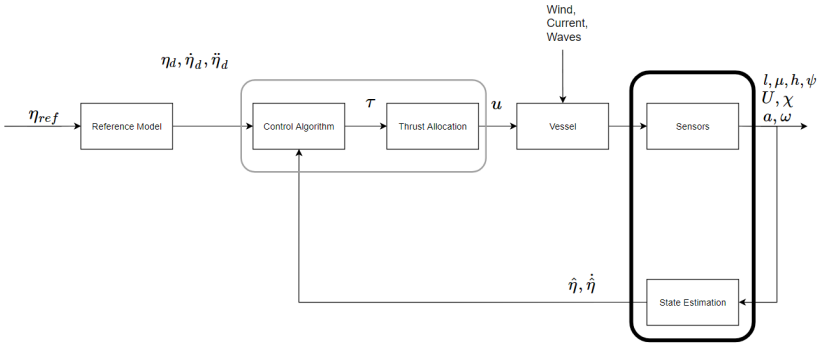


Figure 3.1: Navigation node in the GNC system consisting of the sensors and the state estimator. Illustration by author.

This chapter presents the state estimation algorithm used in the navigation system. The algorithm estimates the pose, velocity, acceleration, angular velocity, and attitude. The navigation system will utilize the measurements from a strap-down IMU and a RTK-GNSS.

3.1 Sensor Preliminaries

For any six DOF vessel model, it is required 12 differential equations to be able to describe the trajectory of a vessel in 3 dimensions fully. Simple reasons like expensive sensors, or the fact that the measurements are hard to perform accurately, lead to non-measured states. By using a GNSS and an IMU, a 6 DOF vessel state can be estimated despite lack of measurements of all states. The main drawback of using an IMU is the measurement errors that occur due to biases on the angular velocity and linear acceleration measurements.

3.1.1 Inertial Measurement Unit

An IMU is a measurement unit that measures the acceleration and angular velocity for a rigid body. Additionally, the IMU can estimate the attitude based on the Earth's magnetic field and the gravity constant. An IMU uses three orthogonally mounted accelerometers and gyroscopes to measure the acceleration and angular velocity. The angular velocity and acceleration measurements are integrated once and twice receptively to predict the system states in attitude, velocity, and position.

Accelerometer

An accelerometer measures specific force. A simple directional accelerometer takes the form of a directional pendulum attached to a spring [Fossen, 2011]. Application of a specific force to the proof mass in the accelerometer displaces the proof mass relative to the casing. This displacement can be measured, and the resulting acceleration can be related to the resulting extension of the spring attached to the pendulum. Other types of accelerometers utilizing vibrations also exist.

For an accelerometer, the gravity will be present in the measurements. Subsequently, compensation for the gravitational pull in the measurements is a must. For a marine vessel, the assumption is that the gravity will only be present downward, or along the z-axis expressed in the NED frame. The gravitational acceleration varies based on the location on Earth. The gravity can be calculated based on the World Geodetic System 1984 (WGS84) reference ellipsoid and the measured latitude and altitude. For a surface vessel in calm waters, the elevation is $h \approx 0$. The WGS84 calculates the gravity constant with high accuracy using the ellipsoidal gravity formula given by

$$g(\phi) = 9.7803253359\text{m/s}^2 \left[\frac{1 + 0.00193185265241\sin^2(\phi)}{\sqrt{1 - 0.00669437999013\sin^2(\phi)}} \right] \tag{3.1}$$

By using the latitude measured by the GNSS, the correct gravity constant is calculated accurately and compensated for in the ESKF.

Gyroscope

A gyroscope is an inertial sensor used for measuring orientation and angular velocity. A gyroscope used to measure angular velocity is referred to as a rate-gyroscope. The traditional gyroscope is a spinning wheel that uses the physical law of conservation of angular momentum to assume its orientation. Other gyroscopes are the microelectromechanical systems (MEMS) gyroscopes, the ring laser gyros (RLG) and fiber optic gyros (FOG) [Fossen, 2011]. The MEMS gyros are especially cheap, and lower cost applications see the use of these sensors frequently.

Roll/Pitch Inclination Estimate

If the measurement of the gravitational acceleration is in the BODY frame, it can be used to estimate the roll and pitch of the vessel. For a resting object at an inclination, gravity will affect the acceleration measurements. The calculation of the roll and pitch angles are according to

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \approx -\boldsymbol{R}_n^b(\Theta) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} g\,\sin(\theta) \\ g\,\cos(\theta)\sin(\phi) \\ g\,\sin(\theta)\cos(\phi) \end{bmatrix} \tag{3.2}$$

which gives the relationship

$$\phi \approx \ \text{atan}\left(\tfrac{a_y}{a_z}\right) \tag{3.3a}$$

$$\theta \approx \ -\text{atan}\left(\tfrac{a_x}{\sqrt{a_y^2+a_z^2}}\right) \tag{3.3b}$$

As there is no way to distinguish the biases and the specific force exerted on the vessel to the gravitational force, using this estimation as a measurement over time is not recommended. However, the inclination estimate, together with compass measurements, provide an initial estimate of the attitude to be used in the navigation system.

IMU Errors

All sensors show some sign of measurement noise. The noise can be the results of a variety of reasons. Commonly, the lack of resolution from the analogue to the digital conversion results in some loss of information for the perceived measurement. More expensive sensors that offer higher resolution in the conversion will have less information loss.

In addition to the measurement noise, an IMU will have errors due to biases. Biases are slowly varying perturbations of the measurements performed by the IMU. By not accounting for the bias errors, the estimation of position, velocity and attitude will drift from the actual state over time. State estimation with different specific filters corrects for the biases in the measurements. More specifically, integration filters with absolute position data can render these biases observable. Later, an ESKF will be implemented to account for the errors due to these biases.

Summarizing all characteristics of an IMU, [Fossen, 2011] models the outputs as

$$\boldsymbol{a}_m^b = \boldsymbol{R}_n^b(\boldsymbol{\Theta})(\dot{\boldsymbol{v}}_{m/n}^n - \boldsymbol{g}^n) + \boldsymbol{b}_{acc}^b + \boldsymbol{w}_{acc}^b \tag{3.4a}$$

$$\boldsymbol{\omega}_m^b = \boldsymbol{\omega}_{m/n}^b + \boldsymbol{b}_{gyro}^b + \boldsymbol{w}_{gyro}^b \tag{3.4b}$$

$$\boldsymbol{m}_m^b = \boldsymbol{R}_n^b(\boldsymbol{\Theta})\boldsymbol{m}^n + \boldsymbol{b}_{mag}^b + \boldsymbol{w}_{mag}^b \tag{3.4c}$$

where $\boldsymbol{b}^b$ and $\boldsymbol{w}^b$ denotes the bias and noise in the measurements respectively. The measurements are here defined in the body-fixed IMU sensor frame $\{m\}$ relative to the BODY frame $\{b\}$. A strapdown system is when the IMU is mounted away from CO. A computer is then used to transform the IMU measurements into the coordinate frame CO.

Wild Point Filtering

For the given measurements from the IMU, it is crucial to perform some form of sanity check on the incoming measurements. The sanity checks pick up any unforeseen disturbances or measurement errors affecting the measurements. For example, an erroneous spike in any of the measurements should be discarded, and not used in the estimation of the state vector.

For measurements in acceleration, any sudden spikes can happen if the vessel bumps into another object. Dropping these spikes can lead to poor prediction and should not occur. The design of the wild point filter must account for these spikes. For any three DOF measurement vector of acceleration, the wild point filter keeps track of the difference between the current acceleration measurement and the previous acceleration measurement. If the difference between the two measurements is too significant, the wild point filter drops the current measurement. The following check handles the potential measurement drop

if $|a_i - a_{i_{prev}}| < a_{tolerance}$ or dropped measurements $>$ max dropped allowed
    $a_i \rightarrow$ approved
    dropped measurements $= 0$
else
    $a_i \rightarrow$ rejected
    dropped measurements $+= 1$

where the subscript $i = x, y, z$, and denotes the acceleration along the separate axes.

The wild point filter for the angular velocity measurements uses a different tolerance methodology. An object hitting an unmovable object is considered as a fully elastic collision in a frictionless system. A fully elastic collision in this manner implies that an object will have the same velocity after the collision with a mirrored angle of direction. The

wild point filtering for the measured angular velocity uses this principle to set the filter tolerance. The wild point filter defines the tolerance for angular velocity measurements as

$$|\omega_i - \omega_{i_{prev}}| > \max(2 \cdot |\omega_{i_{prev}}|,\ \omega \text{ noise threshold})$$

where the ω noise threshold is some threshold set slightly larger than the largest observed peak-to-peak noise. The threshold is to keep the measurement noise from causing the wild point filter to drop measurements. Figure 3.2 demonstrates the ESKF dropping measurements for the a_z measurement.

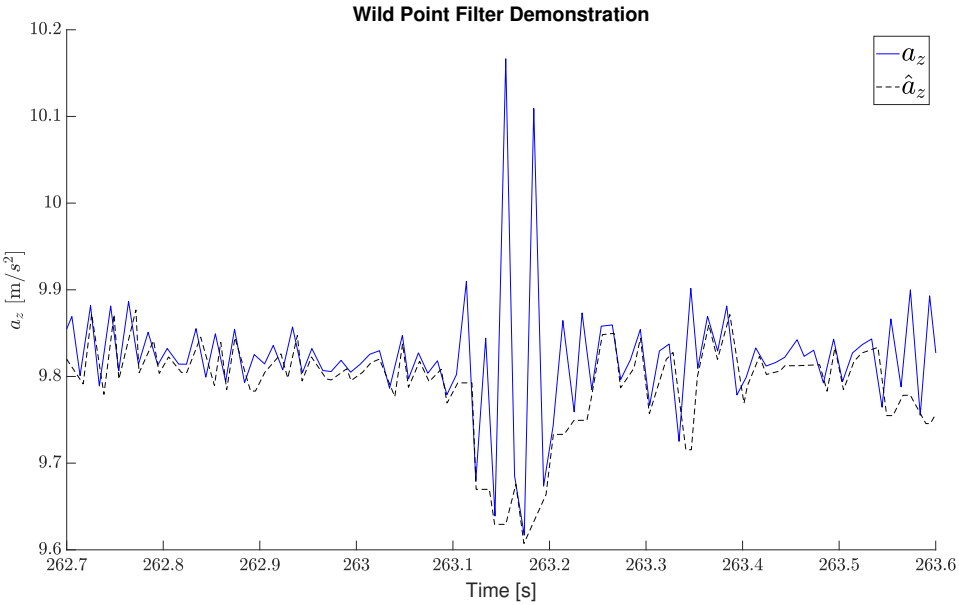


Figure 3.2: Acceleration wild point filter in action for measurements of a_z.

Prolonged Prediction

As mentioned in the introduction of this section, the measurements of an IMU can be integrated once and twice to obtain the states position, velocity and attitude. In case the aiding sensor drops out, in this scenario, a GNSS, the IMU can be used to predict the states of the system. When there are no external updates or corrections, the estimation algorithm will rely entirely on the IMU until any other sensor data measurements are received.

As will be discussed later in this chapter, the IMU measurements drive the implemented filter. Consequently, any measurement noise from the IMU will propagate during prediction. The result is that the measurement noise of the IMU directly alters the predicted state of position, velocity and attitude. For a slowly accelerating system like the one at

hand, the measurement noise will often be more dominant than the actual dynamics of the system. The domination of the stochastic noise will lead to inaccurate predictions over a small period, and drift will occur rapidly. Prolonged prediction based solely on IMU measurements is highly discouraged for the system at hand.

3.1.2 Global Navigation Satellite System

GNSS is a positioning system with global coverage. Several different positioning systems exist, such as the United States' global positioning system (GPS), Russia's GLONASS system and the European Union's Galileo positioning system. A positioning system uses several satellites to triangulate the position of a receiver. There are numerous error sources to the position measurements. A shortlist includes errors induced by ephemeris data, ionospheric delay, tropospheric delay, satellite clock error, multipath reception and receiver measurement timing [Beard and McLain, 2012]. However, this thesis will not go into these in detail.

To improve the overall accuracy of the position measurement, the GNSS measurement is aided by a base station placed in a known location, better known as RTK. RTK uses carrier-based ranging to determine the distance between the vessel receiving GNSS data and a base station. The ranging data is used to correct for the numerous error sources previously mentioned. This method dramatically increases the accuracy of the position measurements, and it achieves an accuracy as high as ± 1 cm. Figure 3.3 shows a visualization of the principle. The GNSS uses the same wild point filter technique discussed in Section 3.1.1 to drop any erroneous measurements.

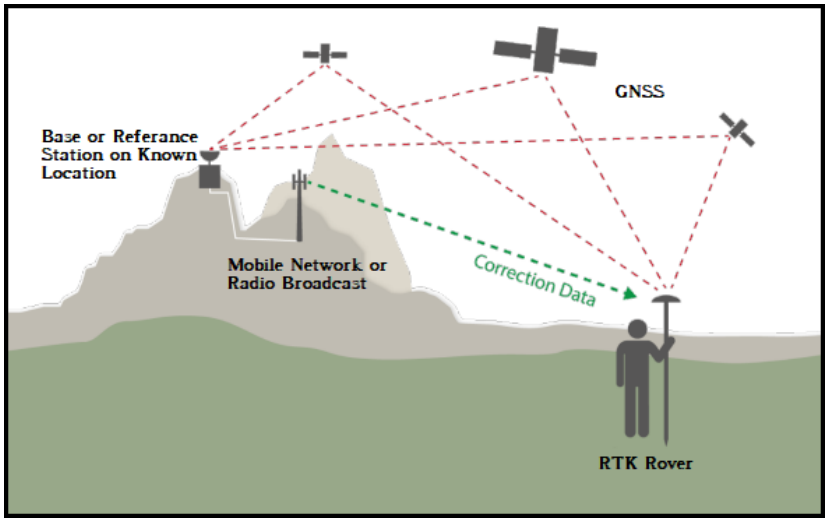


Figure 3.3: The principle of RTK-GNSS. Image courtesy of swisstopo.admin.com.

The GNSS provides the measurements position, velocity, and heading. The output from the GNSS is given in longitude and latitude, speed and course, and ψ. The converted measurements expressed in the NED frame is modelled by the following equations

$$\boldsymbol{p}_{b/n}^{n} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.5a}$$

$$\boldsymbol{v}_{b/n}^{n} = \mathbf{R}(\chi) \begin{bmatrix} U \\ 0 \\ 0 \end{bmatrix} \tag{3.5b}$$

$$\psi_b = \psi \tag{3.5c}$$

where χ is the course of the vessel and U is the speed of the vessel. Figure 2.3 shows the relation of these measurements to the relevant reference frames used in the GNC system.

3.2 Error State Kalman Filter

A state estimator is a dynamic system that estimates the full state of a system. The estimation algorithm uses available measurements of inputs and outputs to estimate other states that are not directly measured. The publication of [Kalman, 1960] in 1960, is the introduction of the well-known estimation algorithm called the Kalman filter. The Kalman filter has a statistical framework that deals with the uncertainty measurement noise and the like introduce. The statistical framework allows for the estimation of the state covariance. The state covariance is a metric used to quantify the uncertainty for the predicted state. The covariance estimation allows the Kalman filter to perform an educated estimate of the next state. Figure 3.4 shows a visual representation of this principle.

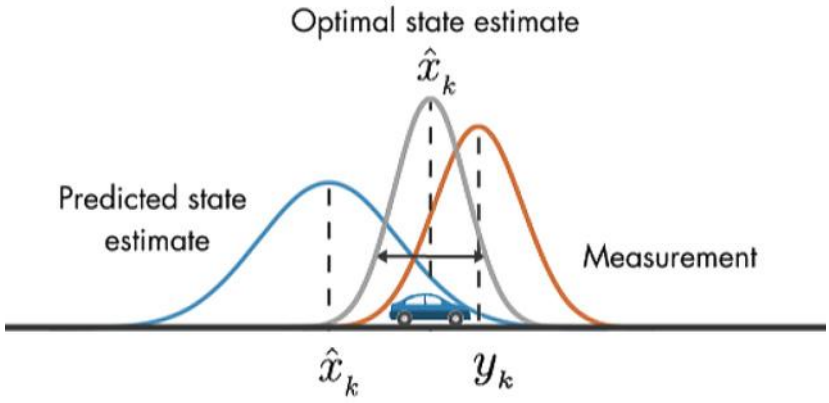


Figure 3.4: The principle of the Kalman filter. Image courtesy of MathWorks: Understanding Kalman Filters, Part 3: An Optimal State Estimator.

The Kalman Filter enables estimation of the non-measured states. The measurement of individual states renders other states observable. By using kinematic models of a system, the measured states in combination with the observable states are used to recreate the entire state space. The Kalman Filter is used in a variety of applications and has ever since its development had an enormous impact on control engineering.

For a system that has IMU measurements, the 6 DOF EOM can be estimated using complementary data from a GNSS. By using the IMU measurements for linear acceleration and angular velocity, an IMU driven ESKF can be developed. This section establishes the relevant theory for an ESKF. The approach in [Solà, 2017] forms the basis for the filter design.

3.2.1 Choices of Attitude Representation

There are numerous possible ways to represent the attitude of a rigid body system. One of the more conventional representations is the three-component parametrization Euler angles. The Euler angles decomposition describes three rotations about the axes of the relevant coordinate system. Depending on the coordinate system, the definition of the representation for the rotation is either intrinsic or extrinsic. Extrinsic rotations for the Euler angle representation refer to a rotation in an inertial ECEF frame. For any coordinate system attached to a moving body frame coordinate system, the rotation refers to an intrinsic rotation. Other three-component parametrizations such as the rotation vector exist. However, three-component parametrizations suffer from singularities.

Different alternatives to the three-component representations exist. Among these are the rotation matrix and the quaternion. The rotation matrix is a representation that uses a

matrix to perform rotations in Euclidean space. The quaternion, on the other hand, is a 4-component parameterization of a general rotation. Both representations are non-minimalistic, and thus do not suffer from the singularity present in the three-component parametrizations. However, the relation between the different parametrizations are useful and can be applied when necessary.

Further, the frequent use of the different conventions of the Euler representation can cause confusion. The differential equations that describe the behavior of the Euler angles, given in (3.6) and (3.7), are cluttered and contains many non-linearities. Thus, the mathematical representation of both the rotation matrix and the quaternion are less complicated. To avoid the singularity, and the cluttered mathematical expressions, quaternion parameterization is used as the primary representation of the attitude in the ESKF.

3.2.2 Quaternion Kinematics

As mentioned, the Euler angle representation is problematic since it has a singularity. In GNC applications, the commonly used convention of the Euler angle representation is the zyx-convention. The relation of the Euler rates and angular velocity showcases the singularity. In [Fossen, 2011], the relation is specified as

$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{T}_{\Theta}\left(\boldsymbol{\Theta}_{nb}\right)\boldsymbol{\omega}_{b/n}^{b} \tag{3.6}$$

where

$$\mathbf{T}_{\Theta}^{-1}\left(\boldsymbol{\Theta}_{nb}\right) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \quad \Longrightarrow \quad \mathbf{T}_{\Theta}\left(\boldsymbol{\Theta}_{nb}\right) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{3.7}$$

By inspection of the matrix $\mathbf{T}_{\Theta}\left(\boldsymbol{\Theta}_{nb}\right)$, it is clear that a singularity occurs for any angle that results in $cos(\theta) = 0$, or more specifically $\theta = 90^{\circ} + 180^{\circ} \cdot k$ where $k \in \mathbb{Z}$. This is known as gimbal lock.

The quaternion is a four-parameter parametrization of a general rotation. The parametrization is defined as a complex vector with one real part, q_w, and three imaginary parts given by the vector $\mathbf{q}_v = [q_x, q_y, q_z]$. Together they form the quaternion representation given by the 4 - dimensional vector

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{3.8}$$

This thesis uses the Hamilton convention of the quaternion. The definition of the quaternion product in component form in [Solà, 2017] is

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \tag{3.9}$$

In terms of the scalar and vector parts, the product is given by

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^\top \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} \tag{3.10}$$

The quaternion product is noncommutative, which means that

$$\mathbf{p} \otimes \mathbf{q} \neq \mathbf{q} \otimes \mathbf{p} \tag{3.11}$$

Another important kinematic detail for the quaternion is its conjugate. The quaternion conjugate is defined by

$$\mathbf{q}^* = q_w - \boldsymbol{q}_v = \begin{bmatrix} q_w \\ -\boldsymbol{q}_v \end{bmatrix} \tag{3.12}$$

The quaternion representation is significant because there are no singularities in its representation. Unfortunately, the quaternion is a non-intuitive representation. However, conversion of the attitude given by the quaternion into Euler angles or the rotation matrix is possible. The use of the different relationships between quaternions, Euler angles, and the rotational matrix is applied whenever needed. The quaternion expresses the rotation matrix in component form as

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2\left(q_x q_y - q_w q_z\right) & 2\left(q_x q_z + q_w q_y\right) \\ 2\left(q_x q_y + q_w q_z\right) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2\left(q_y q_z - q_w q_x\right) \\ 2\left(q_x q_z - q_w q_y\right) & 2\left(q_y q_z + q_w q_x\right) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.13}$$

In terms of the scalar and vector part of the quaternion, the quaternion defines the rotation matrix as

$$\mathbf{R}(\mathbf{q}) = \left(q_w^2 - \mathbf{q}_v^\top \mathbf{q}_v\right) \mathbf{I} + 2\mathbf{q}_v \mathbf{q}_v^\top + 2q_w \mathbf{S}(\mathbf{q}_v) \tag{3.14}$$

The quaternion defines the Euler angles through the relationships given by

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_w q_x + q_y q_z)}{1 - 2\left(q_x^2 + q_y^2\right)} \\ \arcsin\left(2\left(q_w q_y - q_z q_x\right)\right) \\ \arctan \frac{2(q_w q_z + q_x q_y)}{1 - 2\left(q_y^2 + q_z^2\right)} \end{bmatrix} \tag{3.15}$$

The relationship between Euler angles and the rotation matrix is described by

$$\begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{3.16}$$

where $c = cos$ and $s = sin$. One solution to this equality is given in [Fossen, 2011] as

$$\begin{aligned} \phi &= \operatorname{atan2}(R_{32}, R_{33}) \\ \theta &= -\sin^{-1}(R_{31}) = -\tan^{-1}\left(\frac{R_{31}}{\sqrt{1 - R_{31}^2}}\right); \quad \theta \neq \pm 90^\circ \\ \psi &= \operatorname{atan2}(R_{21}, R_{11}) \end{aligned} \tag{3.17}$$

The angular velocity measurements are integrated once with quaternion integration to predict the attitude. [Solà, 2017] uses a Taylor expansion to define a backward integration method for angular velocity measurements arriving in real-time. The definition of the zeroth-order backward quaternion integration is

$$\mathbf{q}_n = \mathbf{q}_{n-1} \otimes \mathbf{q}\{\boldsymbol{\omega_n} \Delta t\} \tag{3.18}$$

where $\mathbf{q}_n$ is the updated quaternion, $\mathbf{q}_{n-1}$ is the previous quaternion, and $\mathbf{q}\{\boldsymbol{\omega_n}\Delta t\}$ represents the quaternion of the incremental rotation in the attitude given by $\Delta\boldsymbol{\theta} = \boldsymbol{\omega}_n \Delta t$.

3.2.3 Nominal State Kinematics

The ESKF uses a nominal state that tracks the error corrected state vector. The state vector for the system is defined by the EOM in 6 DOF described by (2.4) with the addition of the estimated acceleration. The definition of the state vector is

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \boldsymbol{\Theta} \\ \boldsymbol{a} \\ \boldsymbol{\omega} \end{bmatrix} \tag{3.19}$$

where $\boldsymbol{p}$ is in NED frame, and $\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{a}$ and $\boldsymbol{\omega}$ are in the BODY frame. The nominal state, which tracks the error corrected estimated states, is defined by

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{\boldsymbol{p}} \\ \hat{\boldsymbol{v}} \\ \hat{\boldsymbol{\Theta}} \\ \hat{\boldsymbol{a}}_b \\ \hat{\boldsymbol{\omega}}_b \end{bmatrix} \tag{3.20}$$

where $\hat{\boldsymbol{p}}$ is the estimated position and $\hat{\boldsymbol{v}}$ is the estimated velocity in the NED frame, and $\hat{\boldsymbol{\Theta}}$ is the estimated attitude , $\hat{\boldsymbol{a}}_{\boldsymbol{b}}$ is the estimated acceleration bias and $\hat{\boldsymbol{\omega}}_{\boldsymbol{b}}$ is the estimated angular velocity bias in the BODY frame.

The measured angular velocity and linear acceleration from the IMU in the BODY frame performs the prediction of the nominal state using integration. Ideally, the angular velocity measurement model, given by (3.4b), should incorporate the Earth's rotation rate. such that the angular velocity measurement model is

$$\boldsymbol{\omega}_m^b = \boldsymbol{\omega}_{m/n}^b + \mathbf{R}^\mathsf{T}\boldsymbol{\omega}_\varepsilon^n + \boldsymbol{b}_{gyro}^b + \boldsymbol{w}_{gyro}^b \tag{3.21}$$

where $\boldsymbol{\omega}_\varepsilon$ is the Earth's rotation rate. However, as will be discussed in Chapter 5, the experimental platform uses a MEMS IMU prone to measurement noise and biases. Consequently, the implementation in this thesis neglects the Earth's rotation in the formulation of the ESKF.

[Solà, 2017] gives the equations for prediction of the nominal state in both continuous and discrete-time. This thesis focuses on the discrete-time implementation, and subsequently, the discrete-time equations. The nominal state in discrete time is predicted by

$$\hat{\boldsymbol{p}} \leftarrow \hat{\boldsymbol{p}} + \hat{\boldsymbol{v}}\Delta t + \frac{1}{2}(\hat{\mathbf{R}}(\boldsymbol{a}_m - \hat{\boldsymbol{a}}_b) + \boldsymbol{g})\Delta t^2 \tag{3.22a}$$

$$\hat{\boldsymbol{v}} \leftarrow \hat{\boldsymbol{v}} + (\hat{\mathbf{R}}(\boldsymbol{a}_m - \hat{\boldsymbol{a}}_b) + \boldsymbol{g})\Delta t \tag{3.22b}$$

$$\hat{\boldsymbol{q}} \leftarrow \hat{\boldsymbol{q}} \otimes \boldsymbol{q}\{(\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b)\Delta t\} \tag{3.22c}$$

$$\hat{\boldsymbol{a}}_b \leftarrow \hat{\boldsymbol{a}}_b \tag{3.22d}$$

$$\hat{\boldsymbol{\omega}}_b \leftarrow \hat{\boldsymbol{\omega}}_b \tag{3.22e}$$

$$\tag{3.22f}$$

where $\hat{\mathbf{R}} = \mathbf{R}(\hat{\boldsymbol{q}})$ is the estimated rotation matrix based on the estimated attitude $\hat{\mathbf{q}}$ represented by the quaternion, $\boldsymbol{a}_m$ is the measured acceleration, and $\boldsymbol{\omega}_m$ is the measured angular velocity.

3.2.4 Error State Kinematics

The error state kinematics bears many similarities to the nominal state kinematics. However, the error state introduces the random perturbations and noise found in the measure-

ments of the IMU. The error-state kinematics is approximated as a linear system by

$$\delta\hat{\mathbf{p}} \leftarrow \delta\hat{\mathbf{p}} + \delta\hat{\mathbf{v}}\Delta t \tag{3.23a}$$

$$\delta\hat{\mathbf{v}} \leftarrow \delta\hat{\mathbf{v}} + \left(-\hat{\mathbf{R}}\left[\mathbf{a}_m - \hat{\mathbf{a}}_b\right]_\times \delta\hat{\boldsymbol{\theta}} - \hat{\mathbf{R}}\delta\hat{\mathbf{a}}_b\right)\Delta t + \hat{\mathbf{v}}_\mathbf{i} \tag{3.23b}$$

$$\delta\hat{\boldsymbol{\theta}} \leftarrow \hat{\mathbf{R}}^\top\left\{\left(\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b\right)\Delta t\right\}\delta\hat{\boldsymbol{\theta}} - \delta\hat{\boldsymbol{\omega}}_b\Delta t + \boldsymbol{\theta}_\mathbf{i} \tag{3.23c}$$

$$\delta\hat{\mathbf{a}}_b \leftarrow \delta\hat{\mathbf{a}}_b + \mathbf{a}_\mathbf{i} \tag{3.23d}$$

$$\delta\hat{\boldsymbol{\omega}}_b \leftarrow \delta\hat{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_\mathbf{i} \tag{3.23e}$$

where $\mathbf{v_i}$, $\boldsymbol{\theta}_\mathbf{i}$, $\mathbf{a_i}$ and $\boldsymbol{\omega}_\mathbf{i}$ are random Gaussian white noise processes for the velocity, attitude, acceleration bias and angular velocity bias respectively.

3.2.5 Error State Jacobian

Given the error state kinematics in (3.23) without noise, the whole set of equations is on matrix form as

$$\delta\dot{\hat{\boldsymbol{x}}} = \mathbf{A}\delta\hat{\boldsymbol{x}}(t), \quad \mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\hat{\mathbf{R}}\left[\mathbf{a}_m - \hat{\mathbf{a}}_b\right]_\times \Delta t & -\hat{\mathbf{R}}\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{R}}^\mathsf{T}\{(\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b)\Delta t\} & \mathbf{0} & -\mathbf{I}\Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3.24}$$

The key here is that for the error state, this equation is linear. The linearity in the error state simplifies the entire operation, and the solution to (3.24) in discrete time is given by

$$\mathbf{x}_{n+1} = e^{\mathbf{A}\Delta t}\mathbf{x}_n = \boldsymbol{\Phi}\mathbf{x}_n \tag{3.25}$$

where $\boldsymbol{\Phi}$ is called the transition matrix for the system. To approximate the solution to (3.24), the Taylor expansion for e^t is used. This Taylor expansion is given by

$$\boldsymbol{\Phi} = e^{\mathbf{A}\Delta t} = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 + \frac{1}{6}\mathbf{A}^3\Delta t^3 + \cdots = \sum_{k=0}^{\infty}\frac{1}{k!}\mathbf{A}^k\Delta t^k \tag{3.26}$$

In Appendix B in [Solà, 2017], the matrices $\mathbf{A}^k, k = 1, 2, 3, 4$, are given. These are given

by

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{P_v} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V}_\theta & \mathbf{V_a} & 0 \\ 0 & 0 & \mathbf{\Theta}_\theta & 0 & \mathbf{\Theta}_\omega \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.27a}$$

$$\mathbf{A}^2 = \begin{bmatrix} 0 & 0 & \mathbf{P}_v\mathbf{V}_\theta & \mathbf{P}_v\mathbf{V}_a & 0 \\ 0 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{\Theta}_\theta^2 & 0 & \mathbf{\Theta}_\theta\mathbf{\Theta}_\omega \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.27b}$$

$$\mathbf{A}^3 = \begin{bmatrix} 0 & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\theta & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta^2 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{\Theta}_\theta^3 & 0 & \mathbf{\Theta}_\theta^2\mathbf{\Theta}_\omega \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.27c}$$

$$\mathbf{A}^4 = \begin{bmatrix} 0 & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\theta^2 & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\theta\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta^3 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta^2\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{\Theta}_\theta^4 & 0 & \mathbf{\Theta}_\theta^3\mathbf{\Theta}_\omega \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.27d}$$

By following the pattern in the increasing orders of the matrices, the general expression for the k-order matrix $\mathbf{A}^k$ is

$$\mathbf{A}^k = \begin{bmatrix} 0 & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\theta^{k-2} & 0 & \mathbf{P}_v\mathbf{V}_\theta\mathbf{\Theta}_\theta^{k-3}\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta^{k-1} & 0 & \mathbf{V}_\theta\mathbf{\Theta}_\theta^{k-2}\mathbf{\Theta}_\omega \\ 0 & 0 & \mathbf{\Theta}_\theta^k & 0 & \mathbf{\Theta}_\theta^{k-1}\mathbf{\Theta}_\omega \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.27e}$$

where $k \in \{\mathbb{N} \mid k \neq 0, 1, 2\}$ denotes the order of the matrix. The different entries in the matrix are found by comparing the entries in (3.24) and (3.27a). This further gives the expressions for the entries in the matrix as

$$\mathbf{P}_v = \mathbf{I} \tag{3.28a}$$

$$\mathbf{V}_a = -\hat{\mathbf{R}} \tag{3.28b}$$

$$\mathbf{V}_\theta = -\hat{\mathbf{R}}\left[\mathbf{a}_m - \hat{\mathbf{a}}_b\right]_\times \tag{3.28c}$$

$$\mathbf{\Theta}_\theta = (\hat{\mathbf{R}}^\mathsf{T}\{(\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b)\Delta t\} - \mathbf{I})/\Delta t \tag{3.28d}$$

$$\mathbf{\Theta}_\omega = -\mathbf{I} \tag{3.28e}$$

However, this is the transition matrix for the set of equations without noise. The integration also needs to account for the perturbation and noise signals present. The various noise and perturbation signals in the system are due to the measurement noise in the different sensors used, as well as the bias in the IMU measurements in angular velocity and linear acceleration. A GNSS measures position, speed over ground (SOG), course over ground (COG) and heading. The GNSS measurement models are defined as

$$\mathbf{p}^n = \mathbf{p}^n + \boldsymbol{w}_p^c \tag{3.29}$$
$$\mathbf{v}^b = \mathbf{v}^b + \boldsymbol{w}_v^c \tag{3.30}$$
$$\boldsymbol{\psi}^b = \boldsymbol{\psi}^b + \boldsymbol{w}_\psi^c \tag{3.31}$$
$$\tag{3.32}$$

where $\mathbf{p}^n = [x, y, z]^\mathsf{T}$, $\mathbf{v}^b = [\dot{x}^b, \dot{y}^b]^\mathsf{T}$, ψ^b is the compass measurement, and $\boldsymbol{w}^c$ represents the measurement noise for the given measurement. The IMU measurements are given by

$$\mathbf{a}^b = \mathbf{R}_n^b(\boldsymbol{\Theta})(\dot{\boldsymbol{v}}^n - \boldsymbol{g}^n) + \boldsymbol{b}_{acc}^b + \boldsymbol{w}_{acc}^b \tag{3.33}$$
$$\boldsymbol{\omega}^b = \boldsymbol{\omega} + \boldsymbol{b}_{gyro}^b + \boldsymbol{w}_{gyro}^b \tag{3.34}$$
$$\dot{\boldsymbol{b}}^b = \boldsymbol{v}^c \tag{3.35}$$

where $\boldsymbol{w}$ is the measurement noise for the given measurement, and $\boldsymbol{b}$ is the bias for the given measurement. The models for noise and perturbations signals are white Gaussian processes defined by

$$\boldsymbol{w}^c \sim \boldsymbol{N}\{0, \sigma_n\} \tag{3.36}$$
$$\boldsymbol{v}^c \sim \boldsymbol{N}\{0, \sigma_b\} \tag{3.37}$$

where σ_n and σ_b denotes the covariance for the measurement noise and perturbation signals respectively. By using the noise and perturbation covariances, the noise covariance matrix $\boldsymbol{Q}_d$ is defined by

$$\boldsymbol{Q}_d = \mathbf{BQB}^\mathsf{T} \tag{3.38}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{R} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \sigma_{a_b}^2 \Delta t^2 \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{w_b}^2 \Delta t^2 \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{a_n}^2 \Delta t \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{w_n}^2 \Delta t \mathbf{I} \end{bmatrix}, \tag{3.39}$$

and $\sigma_{a_b}^2, \sigma_{w_b}^2, \sigma_{a_n}^2$ and $\sigma_{w_n}^2$ denotes the covariance for the noise and perturbations in the measurements.

The system will then transition into the next state by using the transition matrix $\mathbf{\Phi}$ and noise covariance matrix $\mathbf{Q}_d$ by

$$\mathbf{P} \leftarrow \mathbf{\Phi P \Phi} + \mathbf{Q}_d \tag{3.40}$$

$$\tag{3.41}$$

3.2.6 Error Correction Using Secondary Sensor Data

Secondary sensor data is used to render the biases in the IMU measurements observable. The filter performs error correction by observation based on secondary sensor data given by the GNSS. The GNSS sensor provides a generic measurement given by

$$\boldsymbol{z} = h(\boldsymbol{x}_t) + v \tag{3.42}$$

where $\boldsymbol{z}$ denotes the measured data, $h(\boldsymbol{x}_t)$ is the true state and $\boldsymbol{v}$ is the measurement noise. The filter is estimating the error state, so the filter correction equations become

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T} + \mathbf{V})^{-1} \tag{3.43}$$

$$\hat{\delta\boldsymbol{x}} = \mathbf{K}(\boldsymbol{z} - h(\hat{\boldsymbol{x}}_t)) \tag{3.44}$$

$$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}(\mathbf{I} - \mathbf{K}\mathbf{H})^\mathsf{T} + \mathbf{K}\mathbf{V}\mathbf{K}^\mathsf{T} \tag{3.45}$$

$$\tag{3.46}$$

where $\mathbf{K}$ is the Kalman gain, $\mathbf{H}$ is the measurement matrix, $\mathbf{V}$ is the measurement covariance matrix, $\hat{\delta\boldsymbol{x}}$ is the error state and $\mathbf{P}$ is the covariance matrix. This form of the covariance update refers to the symmetric and positive Joseph form. It is important to realize that the definition of the measurement matrix $\mathbf{H}$ is for the error state. The measurement matrix H is defined by

$$\mathbf{H} = \left.\frac{\partial h}{\partial \delta \mathbf{x}}\right|_{\mathbf{x}} = \left.\frac{\partial h}{\partial \mathbf{x}_t}\right|_{\mathbf{x}} \left.\frac{\partial \mathbf{x}_t}{\partial \delta \mathbf{x}}\right|_{\mathbf{x}} = \mathbf{H}_\mathbf{x}\mathbf{X}_{\delta\mathbf{x}} \tag{3.47}$$

where $\mathbf{H}_x$ is the Jacobian to its argument, and

$$\mathbf{X}_{\delta\mathbf{x}} = \begin{bmatrix} \frac{\partial(\mathbf{p}+\delta\mathbf{p})}{\partial\delta\mathbf{p}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial(\mathbf{v}+\delta\mathbf{v})}{\partial\delta\mathbf{v}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial(\mathbf{a}_b+\delta\mathbf{a}_b)}{\partial\delta\mathbf{a}_b} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial(\omega_b+\delta\omega_b)}{\partial\delta\omega_b} \end{bmatrix} \tag{3.48}$$

Given that the entries in the $\mathbf{X}_{\delta\mathbf{x}}$ are for the error state, all entries except $\frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}}$ are trivial. The non-trivial term is defined in [Solà, 2017] as

$$\frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix} \tag{3.49}$$

Since the error state for the position and velocity measurements are linear, a single observation containing the measurements $\boldsymbol{p}^n \in \mathbb{R}^{3x1}$ and $\boldsymbol{v}^n \in \mathbb{R}^{2x1}$, such that the innovation is

$$\mathbf{x} - \hat{\mathbf{x}} = \begin{bmatrix} \boldsymbol{p} - \hat{\boldsymbol{p}} \\ \boldsymbol{v} - \hat{\boldsymbol{v}} \end{bmatrix},$$

the definition of $\mathbf{H}$ is

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.50}$$

For observations of the yaw angle given by the compass, the error correction is a bit trickier. The first step is to make sure that any subtraction between two angles is unique on the unit circle. The second step is the calculation of the Jacobian. A series of steps make up

the ultimate expression of the compass measurement Jacobian. These are given by

$$w_1 = 2 \cdot (q_x \cdot q_y + q_w \cdot q_z) \tag{3.51a}$$
$$w_2 = q_w^2 + q_x^2 - q_y^2 - q_z^2 \tag{3.51b}$$
$$\rightarrow \tag{3.51c}$$
$$\frac{\partial w_1}{\partial \boldsymbol{q}} = 2 \cdot [q_z, q_y, q_x, q_w]^\mathsf{T} \tag{3.51d}$$
$$\frac{\partial w_2}{\partial \boldsymbol{q}} = 2 \cdot [q_w, q_x, -q_y, -q_z]^\mathsf{T} \tag{3.51e}$$
$$\rightarrow \tag{3.51f}$$
$$\mathbf{J}_\psi = \frac{[w_2, -w_1] \begin{bmatrix} \frac{\partial w_1}{\partial \boldsymbol{q}} \\ \frac{\partial w_2}{\partial \boldsymbol{q}} \end{bmatrix}}{w_1^2 + w_2^2} \tag{3.51g}$$

The Jacobian $\mathbf{H}_x$ for the compass measurement becomes

$$\mathbf{H}_x = \begin{bmatrix} \mathbf{0}_{1\times 6} & \mathbf{J}_\psi & \mathbf{0}_{1\times 6} \end{bmatrix} \tag{3.52}$$

$\mathbf{H}_x$ is post multiplied with $\mathbf{X}_{\delta x}$ to get the measurement matrix $\mathbf{H}_\psi = \mathbf{H}_x \mathbf{X}_{\delta x}$. For a pure yaw angle measurement provided by the compass, the definition of the innovation is

$$\delta\psi = \psi - \hat{\psi} \tag{3.53}$$

where the innovation, $\delta\psi$, is a uniquely mapped solution on the circle manifold.

3.2.7 Filter Update

After observing the error through secondary sensor data, use of the general compositions inject the error state $\hat{\delta \boldsymbol{x}}$ into the nominal state vector $\hat{\boldsymbol{x}}$

$$\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} \oplus \delta\hat{\boldsymbol{x}},$$

alternatively,

$$\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} + \delta\hat{\mathbf{p}} \tag{3.54a}$$
$$\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}} + \delta\hat{\mathbf{v}} \tag{3.54b}$$
$$\hat{\mathbf{q}} \leftarrow \hat{\mathbf{q}} \otimes \mathbf{q}\{\delta\hat{\boldsymbol{\theta}}\} \tag{3.54c}$$
$$\hat{\mathbf{a}}_b \leftarrow \hat{\mathbf{a}}_b + \delta\hat{\mathbf{a}}_b \tag{3.54d}$$
$$\hat{\boldsymbol{\omega}}_b \leftarrow \hat{\boldsymbol{\omega}}_b + \delta\hat{\boldsymbol{\omega}}_b \tag{3.54e}$$

After the injection of the error state, the expected error is zero again. The error state is subsequently reset, and the covariance matrix updated. The expressions for the reset and update are

$$\delta\hat{\boldsymbol{x}} \leftarrow 0 \tag{3.55a}$$

$$\mathbf{P} \leftarrow \mathbf{GPG}^{\mathsf{T}}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - S(\frac{1}{2}\delta\hat{\boldsymbol{\theta}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_6 \end{bmatrix} \tag{3.55b}$$

where $\mathbf{S}(\boldsymbol{x})$ is the skew symmetric matrix of vector $\boldsymbol{x}$.

3.2.8 Lever Arm Compensation

For any error correction containing the position and velocities, it is necessary with a transformation of the measurement from the GNSS sensor frame to the IMU sensor frame. For position lever-arm compensation, the definition of the transformation from position measurements in the GNSS sensor frame to the IMU sensor frame $\{m\}$ is

$$\boldsymbol{p}_m^n = \boldsymbol{p}_{gnss}^n - \mathbf{R}(\boldsymbol{\Theta})\boldsymbol{r}_{gnss}^m \tag{3.56}$$

The definition of the velocity transformation matrix $\mathbf{H}$ is

$$\begin{bmatrix} \hat{\boldsymbol{v}}_{m/n}^b \\ \hat{\boldsymbol{\omega}}_{m/n}^b \end{bmatrix} = \mathbf{H}^{-1}\left(\boldsymbol{r}_{gnss}^m\right) \begin{bmatrix} \hat{\boldsymbol{v}}_{gnss/n}^b \\ \hat{\boldsymbol{\omega}}_{gnss/n}^b \end{bmatrix} \tag{3.57}$$

where

$$\mathbf{H}^{-1}\left(\boldsymbol{r}_{gnss}^m\right) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{S}\left(\boldsymbol{r}_{gnss}^m\right) \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{3.58}$$

$\boldsymbol{r}_{gnss}^m$ is the lever arm between the respective sensor frames, and $\boldsymbol{S}(\boldsymbol{r}_{gnss}^m)$ is the skew-symmetric matrix of the lever arm.

After the error correction, the estimated states are expressed in the IMU sensor frame $\{m\}$. The state vector subsequently needs to be transformed into the CO before using the estimated states for motion control. The same transformation expressed in (3.58) and (3.56) are used to transform the state vector from the $\{m\}$ frame to CO with the lever-arm $\boldsymbol{r}_m^b$.

3.3 Chapter Summary

In this chapter, the new inertial navigation system (INS) is presented. The chapter introduced the different sensors that are in use and the necessary preliminaries for their continued use in the estimation algorithm. Further, the chapter presented the theory and development of the ESKF estimation algorithm. Lastly, the chapter presented the sensor fusion capabilities of the ESKF.

Chapter 4

Motion Control

In a GNC system, the motion control is used to produce the correct control input such that the vessel converges to the desired reference trajectory. The guidance system generates the desired reference trajectory for the motion control system. The motion control acts on the current reference signal and the current state to allocate a thrust input based on the error $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_d$ and $\dot{\boldsymbol{e}}$. An accurate motion control system ensures fast convergence to the desired path. The system needs to be robust enough to correct for any external disturbances and modelling uncertainties efficiently and accurately. Figure 4.1 shows the motion control system and how it interacts with the rest of the GNC system on milliAmpere.

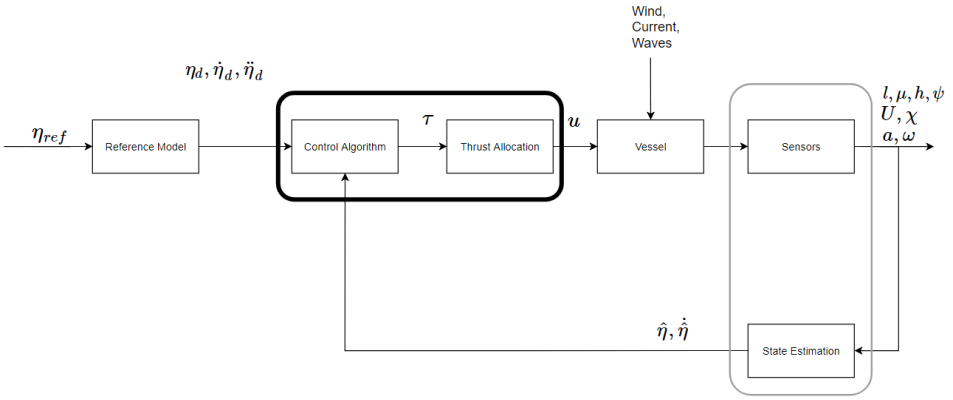


Figure 4.1: Motion control node in the GNC system consisting of the control algorithm and the thrust allocation. Illustration by author.

The thrust allocation converts the desired thrust force into conceivable setpoints for the actuators. Together they form the control system. The guidance system uses a pose reference filter. The already implemented thrust allocation is used to convert the desired control input into feasible actuator setpoints. This thesis utilizes both previous implementations to complement the new navigation and motion control implementations.

4.1 Pose Reference Model

Reference models are used to generate smooth trajectories based on the desired input. These generated trajectories from the reference filter serve as input in the motion control system. The reference model used in this system is a pose reference model.

By utilizing higher-order derivatives of the pose, the reference filter guarantees that the generated trajectories are smooth and feasible for the motion control system. It is essential to have all the constraints for the relevant controller incorporated in this reference model. In this case, the essential constraints are the maximum acceleration and the maximum velocity.

The pose reference model is given in [Fossen, 2011] in vectorial form as

$$\dddot{\boldsymbol{\eta}}_d + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}_d + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d + \boldsymbol{\Omega}^3\boldsymbol{\eta}_d = \boldsymbol{\Omega}^3\boldsymbol{r}^n \tag{4.1}$$

The state-space representation for this equation is

$$A_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -\boldsymbol{\Omega}^3 & -(2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2 & -(2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega} \end{bmatrix}, \qquad B_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\Omega}^3 \end{bmatrix} \tag{4.2}$$

where $\boldsymbol{\Omega} > 0$ and $\boldsymbol{\Delta} > 0$ in (4.1) and (4.2) are diagonal matrices containing the natural frequencies ω_{n_m} and the damping ratios ζ_m, and $\boldsymbol{r}^n$ is the input to the reference filter and denotes the desired pose at the end of the trajectory. The natural frequencies and damping ratios and are given by

$$\boldsymbol{\Omega} = \text{diag}\left\{\omega_{n_1}, \omega_{n_2}, ..., \omega_{n_m}\right\} \tag{4.3a}$$

$$\boldsymbol{\Delta} = \text{diag}\left\{\zeta_1, \zeta_2, ..., \zeta_m\right\} \tag{4.3b}$$

The vectorial reference filter equation (4.1) calculates the desired pose trajectory. Based on the desired jerk, $\dddot{\boldsymbol{\eta}}$, the desired values for any given step k, is calculated with the use of Euler integration

$$\boldsymbol{\eta}_d^n(k+1) = \boldsymbol{\eta}_d^n(k) + \boldsymbol{v}_d^n(k+1)\Delta t \tag{4.4a}$$

$$\boldsymbol{v}_d^n(k+1) = \boldsymbol{v}_d^n(k) + \boldsymbol{a}_d^n(k+1)\Delta t \tag{4.4b}$$

$$\boldsymbol{a}_d^n(k+1) = \boldsymbol{a}_d^n(k) + \dddot{\boldsymbol{\eta}}_d(k+1)\Delta t \tag{4.4c}$$

$$\dddot{\boldsymbol{\eta}}_d^n(k+1) = \boldsymbol{\Omega}^3\boldsymbol{r}^n - (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\boldsymbol{a}_d^n(k) - (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2\boldsymbol{v}_d^n(k) - \boldsymbol{\Omega}^3\boldsymbol{\eta}_d^n(k) \tag{4.4d}$$

While integrating to find the correct pose reference, it is important to enforce any constraints the system has. The saturation function upholds these constraints. The saturation function is given by (4.5) below.

$$sat(x) = \begin{cases} sgn(x)x_{max} & if\ |x| \geq x_{max} \\ x & else \end{cases} \tag{4.5}$$

By using the saturation function, a realistic reference pose trajectory, $\boldsymbol{\eta}_d$, is obtained for the motion control system. More details about the different controllers used in the motion control system will follow in Section 4.2. Figure 4.2 shows a visualization of the pose reference model block scheme.

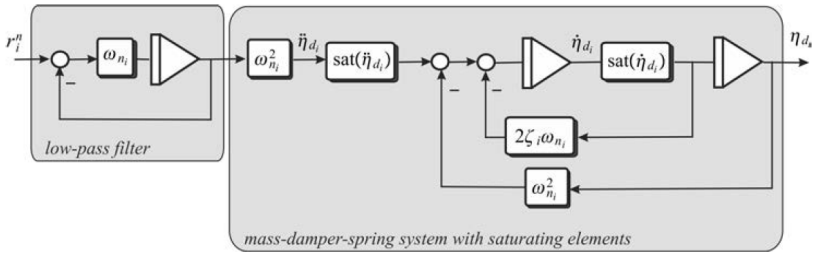


Figure 4.2: Reference model with saturation of $\boldsymbol{a}_d^n$ and $\boldsymbol{v}_d^n$ used in the guidance node of the GNC system. Image courtesy of [Fossen, 2011].

4.2 Control Algorithms

In this system, various controllers serve different purposes. This section establishes these controllers and provides the theory behind them. Figure 4.1 shows the relationship between the controllers and the rest of the GNC system.

4.2.1 PID Controller

One controller forms the basis for the controllers in this system. This controller is called a PID controller. A PID controller can have certain additives such as integrator anti-windup and negation of derivative kick.

The PID controller is used to minimize an error given between the actual state and the reference state. The PID controller utilizes the error between these states and calculates a

desired output. The continuous-time PID controller is given by

$$\tau_{pid} = K_p e + K_i \int_0^t e(\tau)d\tau + K_d \dot{e} \tag{4.6a}$$

$$e = x - x_d \tag{4.6b}$$

where e is the error state, $\dot{e}$ is the differentiated error state and $K_p > 0, K_i > 0$ and $K_d > 0$ are the gains in the controller. The controller used in this system is the discrete PID controller. The discrete version of the PID controller is given by

$$\tau_{pid} = K_p e(k) + K_i \sum_{k=0}^{k} e(k) + K_d \dot{e}(k) \tag{4.7}$$

This controller is applied in the motion control system, and for the controller that sets the correct azimuth angle.

Integrator Anti-Windup

For any control algorithm that uses the traditional control algorithm discussed in Section 4.2.1, specific considerations are necessary. In this case, the one of interest is the integrator anti-windup. Integrator anti-windup is a prudent step taken when the control algorithm wants to output the maximum or minimum allowed output over time. Maximum or minimum allocated trust can result in the error growing over time when the system has no way of acting on it. This accumulated error can propagate to a scenario where the integral effect is not desirable. This course of action can result in undesirable behavior or worst case, instability. Integrator anti-windup mitigates that by disabling the integrator term in specific predefined scenarios.

4.2.2 Dynamic Positioning Controller

Dynamic positioning started as an exotic control technology in the early 1960s [Breivik et al., 2015]. The technology became more advanced and widespread during the 1970s and saw extended use in Norway's effort during the discovery of oil and gas. A dynamic positioning (DP) system is a fully actuated control system, and its intended use is for low-speed maneuvering and station keeping. By driving the state of the system to a given reference, or equivalently $\boldsymbol{e}(t) \to 0$, the desired trajectory for the system is achievable. $\boldsymbol{e}(t)$ is given by

$$\boldsymbol{e}(t) = \boldsymbol{R}^{\mathsf{T}}(\psi)\big(\boldsymbol{\eta}(t) - \boldsymbol{\eta_d}(t)\big) = \boldsymbol{R}^{\mathsf{T}}(\psi) \begin{bmatrix} N(t) - N_d(t) \\ E(t) - E_d(t) \\ \psi(t) - \psi_d(t) \end{bmatrix} \tag{4.8}$$

To establish a control plant for the motion control system, the low-frequency control plant defined in [Sørensen, 2018] is employed. The definition of the plant is

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{4.9a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}(\nu) + \boldsymbol{R}^{\mathsf{T}}(\psi)\boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{R}^{\mathsf{T}}(\psi)\boldsymbol{w} \tag{4.9b}$$

where $\boldsymbol{\nu} = [u, v, r]^{\mathsf{T}}$, $\boldsymbol{\eta} = [x, y, \psi]^{\mathsf{T}}$, $\boldsymbol{w} \in \mathbb{R}^3$ is the bias vector and $\boldsymbol{\tau} = [F_x, F_y, \tau_\psi]$ is the force vector.

The controller should be designed to set the desired $\boldsymbol{\tau}$ correctly. $\boldsymbol{\tau}$ can be written as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{pid} + \boldsymbol{\tau}_{ref_{FF}} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{current} \tag{4.10}$$

where $\boldsymbol{\tau}_{pid}$ is a convential PID controller in three DOF given by (4.7), $\boldsymbol{\tau}_{ref_{FF}}$ is the reference feed forward, and $\boldsymbol{\tau}_{wind}$ and $\boldsymbol{\tau}_{current}$ are the external disturbances. $\boldsymbol{\tau}_{pid}$ is written as

$$\boldsymbol{\tau}_{pid} = -\mathbf{K}_p\boldsymbol{e} - \mathbf{K}_i\int_0^t \boldsymbol{e}d\tau - \mathbf{K}_d\dot{\boldsymbol{e}} \tag{4.11}$$

$$\boldsymbol{e} = \mathbf{R}^{\mathsf{T}}(\psi)\big(\boldsymbol{\eta} - \boldsymbol{\eta}_d\big) \tag{4.12}$$

$$\dot{\boldsymbol{e}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d \tag{4.13}$$

where $\boldsymbol{\eta} = [x, y, \psi]^{\mathsf{T}}$ is the pose, $\boldsymbol{\eta}_d = [x_d, y_d, \psi_d]^{\mathsf{T}}$ is the desired pose, $\boldsymbol{\nu} = [u, v, r]^{\mathsf{T}}$ is the velocity, $\boldsymbol{\nu}_d = [u_d, v_d, r_d]^{\mathsf{T}}$ is the desired velocity, and $\mathbf{K}_p > 0$, $\mathbf{K}_i > 0$ and $\mathbf{K}_d > 0$ are the gain matrices in $\mathbb{R}^{3x3}$.

4.2.3 Model Reference Feedforward

The three DOF vessel model defined by (2.9) can be used to improve transient behavior. By using the knowledge of the model coupled with the desired velocity and acceleration from the reference filter, the calculation of a predictive force is possible. This feedforward force vector is defined by

$$\boldsymbol{\tau}_{FF} = \mathbf{M}\mathbf{a}_d^b + \mathbf{C}(\boldsymbol{\nu}_d^b)\boldsymbol{\nu}_d^b + \mathbf{D}(\boldsymbol{\nu}_d^b)\boldsymbol{\nu}_d^b \tag{4.14}$$

where $\mathbf{a}_d^b = \mathbf{R}^{\mathsf{T}}(\psi)\ddot{\boldsymbol{\eta}}_d$ and $\boldsymbol{\nu}_d^b = \mathbf{R}^{\mathsf{T}}(\psi)\dot{\boldsymbol{\eta}}_d$. It is unrealistic for any model to be 100% accurate, so the feedforward is used alongside the traditional PID controller to correct for any error in the predicted force to ensure tracking of the desired trajectory.

4.2.4 Adaptive Control

The history of adaptive control dates back to the 1950s. Adaptive control is a potent tool that enables estimation of modelling uncertainties and external disturbances. In 1958 the model reference adaptive control (MRAC) first appeared, and in 1965 Lyapunov theory was introduced to address stability issues for the controller [Anavatti et al., 2015]. This section outlines the theoretical approach surrounding the adaptive controller used in this thesis.

Modelling Uncertainties

For a three DOF model of a ship, there will always be unmodelled dynamics. By using adaptive backstepping, these dynamics can be estimated and accounted for in the controller. In [Sørensen and Breivik, 2015], a modified model changed slightly from the one given in (2.9) relates the uncertainties to the model matrices and the control vector. The relationship enables estimation of these dynamics. The new model becomes

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{4.15a}$$

$$\delta\mathbf{M}\dot{\nu} + \delta\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \sigma\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \rho\boldsymbol{\tau} + \mathbf{R}(\psi)^{\mathsf{T}}\boldsymbol{\omega} \tag{4.15b}$$

where $\dot{\mathbf{M}} = 0$ and $\mathbf{C} = -\mathbf{C}^{\mathsf{T}}$ holds. For notational simplicity, the system matrices will be used as $\mathbf{C} = \mathbf{C}(\boldsymbol{\nu})$, $\mathbf{D} = \mathbf{D}(\boldsymbol{\nu})$ and $\mathbf{R} = \mathbf{R}(\psi)$. Using this notation, the correlation from (2.9) to (4.15) is

$$\delta\mathbf{M} = \mathbf{M}^* \tag{4.16a}$$

$$\delta\mathbf{C} = \mathbf{C}^* \tag{4.16b}$$

$$\sigma\mathbf{D} = \mathbf{D}^* \tag{4.16c}$$

$$\rho\boldsymbol{\tau} = \boldsymbol{\tau}^* \tag{4.16d}$$

where the * notation represent the real system matrices and control input, δ, σ and ρ represent the uncertainty connected to the system matrices and the control input considered in the implementation, and $\mathbf{R}^{\mathsf{T}}\boldsymbol{\omega}$ are the external disturbances.

$\mathcal{L}_1$ Adaptive Backstepping

The publication of the $\mathcal{L}_1$ adaptive backstepping controller was in 2010 [Hovakimyan and Cao, 2010]. The new design decouples the adaptation from the robustness of the controller. The approach successfully guarantees transient performance for the vessel states, and it ensures robustness margins. The decoupling of the adaptation from the controller allows for fast adaptation, and subsequently the possibility for aggressive maneuvering.

The $\mathcal{L}_1$ adaptive backstepping controller relies on state prediction to correctly estimate the unmodeled dynamics. The first stage consists of designing a state predictor and consequently, the adaptation laws with the use of the technique backstepping. After the design

of the state predictor, backstepping in combination with variable change is applied to develop a motion controller for the vessel. In this section, the hat notation describes the predicted state by the state predictor used in the estimation of the unmodeled dynamics. The thesis names this controller the $\mathcal{L}_1$-AB controller from here on.

State Prediction and Adaptation Laws

In [Sørensen and Breivik, 2015], an $\mathcal{L}_1$ adaptive controller was designed. The design starts with the state predictor. The state predictor uses the error between a predicted state and the actual state to correct for the prediction errors. The predictor uses the following errors as the basis for the predictions

$$\tilde{\boldsymbol{\eta}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta} \tag{4.17a}$$
$$\tilde{\boldsymbol{\nu}} = \hat{\boldsymbol{\nu}} - \boldsymbol{\nu} \tag{4.17b}$$

where $\tilde{\boldsymbol{\eta}}$ and $\tilde{\boldsymbol{\nu}}$ are the prediction errors for pose and velocity, $\hat{\boldsymbol{\eta}}$ and $\hat{\boldsymbol{\nu}}$ are the predicted pose and velocity, and $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are measured pose and velocity. To make sure that the origins in the prediction errors are exponentially stable, the error dynamics are defined as

$$\dot{\tilde{\boldsymbol{\eta}}} = -\mathbf{L}_1\tilde{\boldsymbol{\eta}} \tag{4.18a}$$
$$\dot{\tilde{\boldsymbol{\nu}}} = -\mathbf{L}_2\tilde{\boldsymbol{\nu}} \tag{4.18b}$$

where $\mathbf{L}_1 > 0$ and $\mathbf{L}_2 > 0$ to ensure stability. By combining the differentiation of (4.17) with regards to time with (4.15), the prediction dynamics becomes

$$\dot{\hat{\boldsymbol{\eta}}} = \dot{\tilde{\boldsymbol{\eta}}} + \dot{\boldsymbol{\eta}} \tag{4.19a}$$
$$\dot{\hat{\boldsymbol{\eta}}} = -\mathbf{L}_1\tilde{\boldsymbol{\eta}} + \mathbf{R}\boldsymbol{\nu} \tag{4.19b}$$

and

$$\dot{\hat{\boldsymbol{\nu}}} = \dot{\tilde{\boldsymbol{\nu}}} + \dot{\boldsymbol{\nu}} \tag{4.20a}$$
$$\dot{\hat{\boldsymbol{\nu}}} = -\mathbf{L}_2\tilde{\boldsymbol{\nu}} + \frac{1}{\hat{\delta}}\mathbf{M}^{-1}(\hat{\rho}\boldsymbol{\tau} - \hat{\delta}\mathbf{C}\boldsymbol{\nu} - \hat{\sigma}\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^{\mathsf{T}}\hat{\boldsymbol{\omega}}) \tag{4.20b}$$
$$\dot{\hat{\boldsymbol{\nu}}} = -\mathbf{L}_2\tilde{\boldsymbol{\nu}} + \mathbf{M}^{-1}(\hat{\rho_\delta}\boldsymbol{\tau} - \mathbf{C}\boldsymbol{\nu} - \hat{\sigma_\delta}\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^{\mathsf{T}}\hat{\boldsymbol{\omega}}_{\boldsymbol{\delta}}) \tag{4.20c}$$

where $\hat{\rho_\delta} = \frac{\hat{\rho}}{\hat{\delta}}$, $\hat{\sigma_\delta} = \frac{\hat{\sigma}}{\hat{\delta}}$, and $\hat{\boldsymbol{\omega}}_{\boldsymbol{\delta}} = \frac{\hat{\rho}}{\hat{\delta}}$. These values consists of the estimates of the uncertainty terms given in (4.15).

Given the uncertainty terms in the state predictor dynamics, it must be designed adaptation laws that correctly estimates the uncertainty variables such that $\hat{\delta} = \delta$, $\hat{\sigma} = \sigma$, and $\hat{\rho} = \rho$. To achieve this, (4.18) is adjusted to

$$\dot{\tilde{\boldsymbol{\eta}}} = -\mathbf{L}_1\tilde{\boldsymbol{\eta}} \tag{4.21a}$$

$$\dot{\tilde{\boldsymbol{\nu}}} = -\mathbf{L}_2\tilde{\boldsymbol{\nu}} + \mathbf{M}^{-1}\big(\tilde{\rho}_\delta\boldsymbol{\tau} - \tilde{\sigma}_\delta\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^\mathsf{T}\tilde{\boldsymbol{\omega}}_{\boldsymbol{\delta}}\big) \tag{4.21b}$$

where $\tilde{\rho}_\delta = \hat{\rho}_\delta - \rho_\delta$, $\tilde{\sigma}_\delta = \hat{\sigma}_\delta - \sigma_\delta$, and $\tilde{\boldsymbol{\omega}}_\delta = \hat{\boldsymbol{\omega}}_\delta - \boldsymbol{\omega}_\delta$. To derive the adaptation laws, the following positive definite control Lyapunov function (CLF) for the prediction law is chosen

$$V_{pred} = \frac{1}{2}\left(\frac{1}{\gamma_{\rho_\delta}}\tilde{\rho}_\delta^2 + \frac{1}{\gamma_{\sigma_\delta}}\tilde{\sigma}_\delta^2 + \frac{1}{\gamma_{\omega_\delta}}\tilde{\boldsymbol{\omega}}_\delta^\mathsf{T}\tilde{\boldsymbol{\omega}}_\delta\right) + \frac{1}{2}\tilde{\boldsymbol{\nu}}^\mathsf{T}\mathbf{M}\tilde{\boldsymbol{\nu}} + \frac{1}{2}\tilde{\boldsymbol{\eta}}^\mathsf{T}\tilde{\boldsymbol{\eta}} \tag{4.22}$$

By differentiation with regards to time, the CLF becomes

$$\begin{aligned}\dot{V}_{pred} =& \frac{1}{\gamma_{\rho_\delta}}\tilde{\rho}_\delta(\dot{\hat{\rho}}_\delta - \dot{\rho_\delta}) + \frac{1}{\gamma_{\sigma_\delta}}\tilde{\sigma}_\delta(\dot{\hat{\sigma}}_\delta - \dot{\sigma_\delta}) + \frac{1}{\gamma_{\omega_\delta}}\tilde{\boldsymbol{\omega}}_\delta^\mathsf{T}(\dot{\hat{\boldsymbol{\omega}}}_\delta - \dot{\boldsymbol{\omega}}_\delta) \\ &+ \tilde{\boldsymbol{\nu}}^\mathsf{T}\mathbf{M}\dot{\tilde{\boldsymbol{\nu}}} + \tilde{\boldsymbol{\eta}}^\mathsf{T}\dot{\tilde{\boldsymbol{\eta}}}\end{aligned} \tag{4.23}$$

Here it is assumed that all the uncertainties are constant or slowly varying compared to the vessel dynamics, meaning that $\dot{\rho_\delta} = 0, \dot{\sigma_\delta} = 0$ and $\dot{\omega_\delta} = 0$. Using this assumption while inserting (4.21) into the equation yields

$$\begin{aligned}\dot{V}_{pred} =& \frac{1}{\gamma_{\rho_\delta}}\tilde{\rho}_\delta\dot{\hat{\rho}}_\delta + \frac{1}{\gamma_{\sigma_\delta}}\tilde{\sigma}_\delta\dot{\hat{\sigma}}_\delta + \frac{1}{\gamma_{\omega_\delta}}\tilde{\boldsymbol{\omega}}_\delta^\mathsf{T}\dot{\hat{\boldsymbol{\omega}}}_\delta - \tilde{\boldsymbol{\eta}}^\mathsf{T}\mathbf{L}_1\tilde{\boldsymbol{\eta}} \\ &+ \tilde{\boldsymbol{\nu}}^\mathsf{T}\big(-\mathbf{M}\mathbf{L}_2\tilde{\boldsymbol{\nu}} + \tilde{\rho}_\delta\boldsymbol{\tau} - \tilde{\sigma}_\delta\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^\mathsf{T}\tilde{\boldsymbol{\omega}}_{\boldsymbol{\delta}}\big)\end{aligned} \tag{4.24}$$

By collecting similar terms, V_{pred} becomes

$$\begin{aligned}\dot{V}_{pred} =& \tilde{\rho}_\delta(\frac{1}{\gamma_{\rho_\delta}}\dot{\hat{\rho}}_\delta + \tilde{\boldsymbol{\nu}}^\mathsf{T}\boldsymbol{\tau}) + \tilde{\sigma}_\delta(\frac{1}{\gamma_{\sigma_\delta}}\dot{\hat{\sigma}}_\delta - \tilde{\boldsymbol{\nu}}^\mathsf{T}\mathbf{D}\boldsymbol{\nu}) + \tilde{\boldsymbol{\omega}}_\delta^\mathsf{T}(\frac{1}{\gamma_{\omega_\delta}}\dot{\hat{\boldsymbol{\omega}}}_\delta + \mathbf{R}\tilde{\boldsymbol{\nu}}) \\ &- \tilde{\boldsymbol{\eta}}^\mathsf{T}\mathbf{L}_1\tilde{\boldsymbol{\eta}} - \tilde{\boldsymbol{\nu}}^\mathsf{T}\mathbf{M}\mathbf{L}_2\tilde{\boldsymbol{\nu}}\end{aligned} \tag{4.25}$$

The adaptation laws are then chosen as

$$\dot{\hat{\rho}}_\delta = -\gamma_{\rho_\delta} \tilde{\boldsymbol{\nu}}^\mathsf{T} \boldsymbol{\tau} \tag{4.26a}$$
$$\dot{\hat{\sigma}}_\delta = \gamma_{\sigma_\delta} \tilde{\boldsymbol{\nu}}^\mathsf{T} \mathbf{D} \boldsymbol{\nu} \tag{4.26b}$$
$$\dot{\hat{\boldsymbol{\omega}}}_{\boldsymbol{\delta}} = -\gamma_{\omega_\delta} \mathbf{R} \tilde{\boldsymbol{\nu}} \tag{4.26c}$$

where $\gamma_{\rho_\delta} > 0, \gamma_{\rho_\delta} > 0$ and $\gamma_{\rho_\delta} > 0$ are the adaptation gains. With these adaptation laws, (4.25) becomes

$$\dot{V}_{pred} = -\tilde{\boldsymbol{\eta}}^\mathsf{T} \mathbf{L}_1 \tilde{\boldsymbol{\eta}} - \tilde{\boldsymbol{\nu}}^\mathsf{T} \mathbf{M} \mathbf{L}_2 \tilde{\boldsymbol{\nu}} < 0, \quad \forall \boldsymbol{\eta}, \boldsymbol{\nu} \neq 0 \tag{4.27}$$

By theorem A.6 in [Fossen, 2011], the system of error states $(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{\nu}}, \tilde{\rho}_\delta, \tilde{\sigma}_\delta, \tilde{\omega}_\delta)$ becomes uniformly globally asymptotically stable (UGAS).

Control Law

The next step is to use a change of variables to define the error variables $\mathbf{z}_1$ and $\mathbf{z}_2$. The definition of the error variables is

$$\mathbf{z}_1 = \mathbf{R}^\mathsf{T}(\boldsymbol{\eta} - \boldsymbol{\eta}_d) \tag{4.28a}$$
$$\mathbf{z}_2 = \boldsymbol{\nu} - \boldsymbol{\alpha} \tag{4.28b}$$

where $\boldsymbol{\alpha}$ is the stabilizing function to be designed later. The next phase of the design consists of two steps to finish the complete backstepping controller.

Step 1

In step one, the first CLF, V_1, is chosen as

$$V_1 = \frac{1}{2} \mathbf{z}_1^\mathsf{T} \mathbf{z}_1 \tag{4.29}$$

To continue the design, differentiation w.r.t. time of the positive definite CLF yields

$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T} \dot{\mathbf{z}}_1 \tag{4.30a}$$
$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T} \big(\mathbf{S}(r)^\mathsf{T} \mathbf{R}^\mathsf{T} (\boldsymbol{\eta} - \boldsymbol{\eta}_d) + \mathbf{R}^\mathsf{T} (\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d) \big) \tag{4.30b}$$
$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T} \big(\mathbf{S}(r)^\mathsf{T} \mathbf{z}_1 + \mathbf{R}^\mathsf{T} (\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d) \big) \tag{4.30c}$$

where

$$\mathbf{S}(r) = \begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.31}$$

Given that $\mathbf{S}(r)$ is skew symmetric, which means $\mathbf{z}_1^\mathsf{T}\mathbf{S}^\mathsf{T}(r)\mathbf{z}_1 = 0$, (4.30c) further simplifies to

$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T}\left(\mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}} - \mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d\right) \tag{4.32a}$$
$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T}\left(\boldsymbol{\nu} - \mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d\right) \tag{4.32b}$$

By inserting (4.28b), $\dot{V}_1$ takes the form

$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T}\left(\mathbf{z}_2 + \boldsymbol{\alpha} - \mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d\right) \tag{4.33a}$$
$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T}\mathbf{z}_2 + \mathbf{z}_1^\mathsf{T}\left(\boldsymbol{\alpha} - \mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d\right) \tag{4.33b}$$

The stabilizing function $\boldsymbol{\alpha}$ is now chosen as

$$\boldsymbol{\alpha} = -\mathbf{K}\mathbf{z}_1 + \mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d \tag{4.34}$$

giving $\dot{V}_1$ the form

$$\dot{V}_1 = \mathbf{z}_1^\mathsf{T}\mathbf{z}_2 - \mathbf{z}_1^\mathsf{T}\mathbf{K}_1\mathbf{z}_1 \tag{4.35}$$

where $\mathbf{K}_1 > 0$.

Step 2

Since $\dot{V}_1 < 0 \quad \forall \mathbf{z}_1, \mathbf{z}_2 \neq 0$ is not true, a second CLF is used. The second CLF, V_2, is chosen to be

$$V_2 = \frac{1}{2}\mathbf{z}_2^\mathsf{T}\mathbf{M}\mathbf{z}_2 + V_1 \tag{4.36a}$$

By following the same procedure as in Step 1, differentiation of V_2 w.r.t. results in

$$\dot{V}_2 = \mathbf{z}_2^\mathsf{T}\mathbf{M}\dot{\mathbf{z}}_2 + \dot{V}_1 \tag{4.37a}$$
$$\dot{V}_2 = \mathbf{z}_2^\mathsf{T}\mathbf{M}\big(\dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\alpha}}\big) + \dot{V}_1 \tag{4.37b}$$
$$\dot{V}_2 = \mathbf{z}_2^\mathsf{T}\big(\hat{\rho}_\delta\boldsymbol{\tau} - \mathbf{C}\boldsymbol{\nu} - \hat{\sigma}_\delta\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta - \mathbf{M}\dot{\boldsymbol{\alpha}}\big) + \mathbf{z}_1^\mathsf{T}\mathbf{z}_2 - \mathbf{z}_1^\mathsf{T}\mathbf{K}_1\mathbf{z}_1 \tag{4.37c}$$
$$\begin{aligned}\dot{V}_2 = \mathbf{z}_2^\mathsf{T}\big(&\mathbf{z}_1 + \hat{\rho}_\delta\boldsymbol{\tau} - \mathbf{C}\boldsymbol{\alpha} - \hat{\sigma}_\delta\mathbf{D}\boldsymbol{\alpha} + \mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta - \mathbf{M}\dot{\boldsymbol{\alpha}}\big) - \mathbf{z}_1^\mathsf{T}\mathbf{K}_1\mathbf{z}_1 \\ &- \mathbf{z}_2^\mathsf{T}\mathbf{C}\mathbf{z}_2 - \mathbf{z}_2^\mathsf{T}\hat{\sigma}_\delta\mathbf{D}\mathbf{z}_2\end{aligned} \tag{4.37d}$$

where

$$\dot{\boldsymbol{\alpha}} = \mathbf{S}(r)^\mathsf{T}\mathbf{R}^\mathsf{T}\dot{\boldsymbol{\eta}}_d + \mathbf{R}^\mathsf{T}\ddot{\boldsymbol{\eta}}_d - \mathbf{K}_1\dot{\mathbf{z}}_1 \tag{4.38}$$

and $\mathbf{z}_2^\mathsf{T}\mathbf{C}\mathbf{z}_2 = 0$ because of the skew symmetric property of the $\mathbf{C}$ matrix. The control law is then chosen as

$$\hat{\rho}_\delta\boldsymbol{\tau} = -\mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta + \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}\boldsymbol{\alpha} + \hat{\sigma}_\delta\mathbf{D}\boldsymbol{\alpha} - \mathbf{z}_1 - \mathbf{K}_2\mathbf{z}_2 \tag{4.39}$$

where $\mathbf{K}_2 > 0$. This ensures that

$$\dot{V}_2 = -\mathbf{z}_1^\mathsf{T}\mathbf{K}_1\mathbf{z}_1 - \mathbf{z}_2^\mathsf{T}(\mathbf{K}_2 + \hat{\sigma}_\delta\mathbf{D})\mathbf{z}_2 < 0 \quad \forall \mathbf{z}_1, \mathbf{z}_2 \neq 0 \tag{4.40}$$

Once again, using theorem A.6 in [Fossen, 2011], the error system $(\mathbf{z}_1, \mathbf{z}_2)$ is UGAS.

The addition of a low pass filter removes any high-frequent oscillatory effects that the adaptation laws introduce. The definition of the low-pass filter is

$$C(s) = \frac{k}{s+k} \tag{4.41}$$

where k is the time constant tuning parameter for the low-pass filter. The low pass filter constant needs to be small enough not to cause a significant time delay on the control input. On the other hand, it needs to be sufficiently large to filter out most of the high-frequency signals coming from the adaptation law. The final control law for the $\mathcal{L}_1$-AB controller is

$$\boldsymbol{\tau}_c = C(s)(-\mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta + \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}\boldsymbol{\alpha} + \hat{\sigma}_\delta\mathbf{D}\boldsymbol{\alpha} - \mathbf{z}_1 - \mathbf{K}_2\mathbf{z}_2) \tag{4.42}$$

The adaptive control scheme is displayed in Figure 4.3.

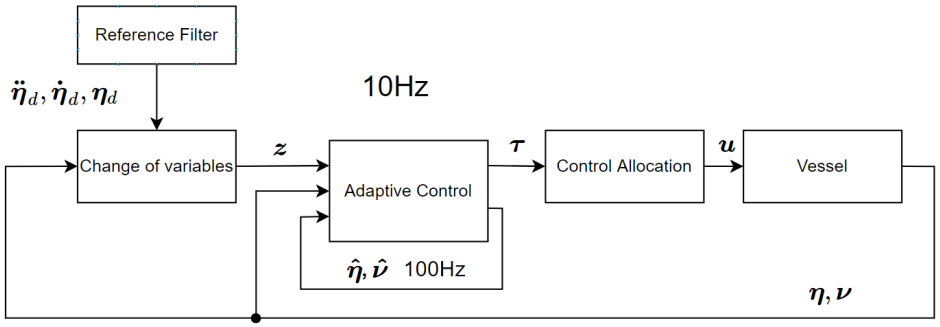


Figure 4.3: The new adaptive motion control scheme. The state prediction of the adaptive control runs at 100Hz, while the rest of the control algorithm runs at 10Hz. Illustration by author.

4.3 Control Parameters

One of the main tasks in this thesis is the development of a new controller that improves tracking performance. All the controllers rely on state feedback to compute the error state. Different approaches to adaptive control are explored to minimize the effect of unmodelled dynamics. The distinct design aspects and their implementations are discussed further in this section.

4.3.1 3 DOF PID Controller

The 3 DOF PID controller is a discrete nonlinear controller. The important elements to consider in the implementation of the controller, are saturating elements, yaw wrapping and integrator anti-windup.

The handling of saturating elements is done indirectly through the reference filter and the generation of the reference trajectory. Yaw wrapping is the action of defining ψ and ψ_d within a set of allowed values given by $-\pi \leq \psi, \psi_d \leq \pi$. Section 4.2.1 discusses the integrator anti-windup.

The 3 DOF PID controller is based on the low frequency plant given in (4.9). See Section 4.2.2 for a more in-depth description of the plant and the controller. The implemented controller has three uncoupled PID controllers in each DOF. The gain parameters for the individual DOF are defined by the entries in the diagonal matrices

$$\mathbf{K}_p = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 800 \end{bmatrix}, \quad \mathbf{K}_d = \begin{bmatrix} 700 & 0 & 0 \\ 0 & 700 & 0 \\ 0 & 0 & 1600 \end{bmatrix} \mathbf{K}_i, \quad = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 15 \end{bmatrix} \tag{4.43}$$

The values of the gain matrices in (4.43) can be related to the force outputs from the thrusters. During the DP controller tests, the thrusters had a max force output of $500N$ per thruster. The arm l_x given in Figure 4.4, is $1.8\,m$. This means that the thrusters can output a maximum of $\boldsymbol{\tau}_{max} = [1000\text{N}, 1000\text{N}, 1800\text{Nm}]^T$ for an uncoupled motion for each separate DOF. For a deviation of

$$\boldsymbol{e_1} = \begin{bmatrix} 0.5\,m \\ 0.5\,m \\ 0.5\,rad \end{bmatrix}$$

the desired force output from the proportional gain will be

$$\boldsymbol{\tau}_p = [100N, 100N, 400Nm]$$

The same logic applies for $\boldsymbol{e}_2$ and $\boldsymbol{\tau}_d$. The $\boldsymbol{\tau}_i$ term will continue to accumulate proportionally with the integrated error state $\int_0^t \boldsymbol{e}_1(\tau)d\tau$ in each DOF with a factor of the relevant entry in the $\boldsymbol{K}_i$ matrix.

4.3.2 $\mathcal{L}_1$ Adaptive Backstepping Controller

The $\mathcal{L}_1$-AB Controller follows a backstepping approach. First, the adaptation gains need to be tuned. The control law is defined in (4.39) as

$$\hat{\rho}_\delta \boldsymbol{\tau} = -\mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta + \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}\boldsymbol{\alpha} + \hat{\sigma}_\delta \mathbf{D}\boldsymbol{\alpha} - \mathbf{z}_1 - \mathbf{K}_2\mathbf{z}_2 \tag{4.44}$$

However, to reduce the number of adaptive variables in the control law, all the system uncertainties and external disturbances are collected into one term. This is done by setting $\hat{\rho}_\delta = \hat{\sigma}_\delta = 1$ or equivalently $\dot{\hat{\rho}}_\delta = \dot{\hat{\sigma}}_\delta = 0$. This simplifies the control law to

$$\boldsymbol{\tau} = -\mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta + \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\alpha} - \mathbf{z}_1 - \mathbf{K}_2\mathbf{z}_2 \tag{4.45}$$

where all the modelling uncertainties and external disturbances expressed in the NED frame are collected in the term $\hat{\boldsymbol{\omega}}_\delta$.

The adaptation law for the adaptive parameter $\hat{\boldsymbol{\omega}}_\delta$ is defined in (4.26). The tuning of the adaptation gain γ_{ω_δ} needs to be with care. Sensor measurements contain measurement noise, and the adaptation gain amplifies the noise with a large γ_{ω_δ}. On the other hand, choose γ_{ω_δ} too small and the estimation might not converge fast enough to ensure accurate tracking. Another challenge is the delay introduced by the thruster dynamics. If the adaptation gain is too large, it may allocate a thrust vector with unfeasible rate change. If this is the case, the delay in the thruster dynamics might cause instability. By using the simulated environment containing measurement noise, the tuning of the adaptation gain results in

$$\gamma_{\omega_\delta} = 25000 \tag{4.46}$$

The tuning of the adaptation law needs to be investigated further in the experimental tests to adjust for the measurement noise and the thruster dynamics.

The next step is the tuning of $\mathbf{K}_1$ and $\mathbf{K}_2$ in (4.40). After iterative tuning in simulations, these are found to be

$$\mathbf{K}_1 = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}, \quad \mathbf{K}_2 = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 2000 \end{bmatrix} \tag{4.47}$$

4.3.3 $\mathcal{L}_1$ Adaptive Feedforward

The $\mathcal{L}_1$-AB controller introduces a different way of compensating for modelling uncertainties and external disturbances in comparison to the PID controller. The design of the $\mathcal{L}_1$-AB controller, allows for usage of the adaptive feedforward terms together with the PID controller. By using the adaptive feedforward terms from (4.42), combined with the P and D terms of (4.7), the definition of a proposed PD controller with adaptive feedforward is

$$\boldsymbol{\tau} = -\mathbf{R}^\mathsf{T}\hat{\boldsymbol{\omega}}_\delta + \mathbf{M}\dot{\boldsymbol{\nu}}_d + \mathbf{C}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d + \mathbf{D}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d + \boldsymbol{K_p}\boldsymbol{e} + \boldsymbol{K_d}\dot{\boldsymbol{e}} \tag{4.48}$$

The estimation of the external disturbances and modelling uncertainties improves the accuracy of the predicted force necessary to comply with the desired path. This design offers reusability of the previously tuned PID controller. Iterative tuning is often time-consuming, and the ability to reuse the already tuned controller simplifies the tuning process significantly.

The introduction of the estimation of unmodelled dynamics through the $\mathcal{L}_1$ controller framework negates the need for the integral term in the traditional PID controller. The adaptation law allows for fast convergence of the estimation of unmodelled dynamics. Faster convergence of the unmodeled dynamics will, in turn, ensure faster path convergence. This controller will be referred to as the proportional–derivative (PD) adaptive feedforward controller, or PD + $\mathcal{L}_1$-FF from here on. The thesis also collects the variations of the PID controller into the term non-adaptive controllers, and the controllers utilizing the adaptive backstepping approach into the term adaptive controllers to separate the two approaches easily.

4.3.4 Magnitude Saturation

A magnitude saturation model ensures that the control algorithm outputs a feasible desired force vector. The magnitude saturation model is a function based on the maximum thrust

for each thruster given by

$$\boldsymbol{\tau}_{ms} = sat_m(\boldsymbol{\tau}_c) \tag{4.49a}$$

where $sat_m = [sat_x, sat_y, sat_\psi]$ are the maximum allowed outputs for the azimuth thrusters. The static saturation limits are set to

$$\boldsymbol{\tau}_{ms} = [500\text{N}, 500\text{N}, 1000\text{Nm}]$$

for the current implementation.

Figure 5.3 shows the non-linear relationship between thrust input and thrust force. Based on the non-linear relationship and the lack of an accurate thruster model, this saturation model includes no rate saturation. During the experimental tests, if the desired rate change in thruster force appears not to be feasible, it is imperative that the tuning parameters in (4.46) and (4.47) are changed.

4.4 Thrust Allocation

The vessel milliAmpere operates in three DOF. It is necessary to map the desired generalized forces, $\boldsymbol{\tau} = [F_x, F_y, \tau_\psi]^\mathsf{T}$ from the controller, to feasible actuator inputs. For milliAmpere, the actuators in question are two azimuth thrusters.

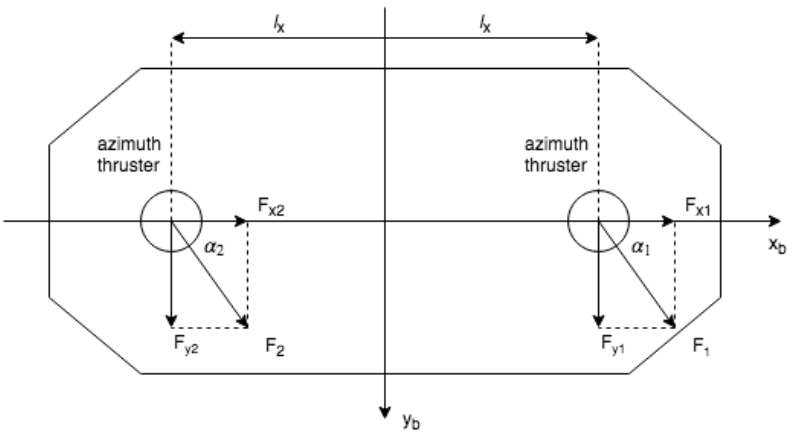


Figure 4.4: Azimuth actuator configuration used on milliAmpere. Here, α denotes the angle, F_x and F_y denotes the decomposed thrust forces in the BODY frame, F_1 and F_2 are the total force and direction produced by each azimuth thruster, and l_x is the moment arm from the thrusters to the CO. Illustration by author.

For the current actuator setup, the relationship between the output force $\boldsymbol{\tau}$ and the actuator inputs $\boldsymbol{u}$ is described by

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u} \tag{4.50}$$

where $\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}$ is called the thrust configuration matrix. $\boldsymbol{K}$ is the relationship between the thruster input $\boldsymbol{u}$ and thruster force output $\boldsymbol{f}$. $\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}$ describes the relationship between the angles and thrust of the azimuth thrusters and the resulting force and torque in the BODY frame. For this vessel the thrust configuration matrix becomes

$$\boldsymbol{\tau} = \begin{bmatrix} \cos(\alpha_1) & \cos(\alpha_2) \\ \sin(\alpha_1) & \sin(\alpha_2) \\ l_x \sin(\alpha_1) & -l_x \sin(\alpha_2) \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.51}$$

The pseudoinverse expresses the explicit solution to the equation. By use of the pseudoinverse, the actuator input vector $\boldsymbol{u}$ is defined in [Fossen, 2011] as

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha})\boldsymbol{\tau} \tag{4.52a}$$

$$\boldsymbol{T}_w^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{T}^{\mathsf{T}}(\boldsymbol{\alpha})[\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^{\mathsf{T}}(\boldsymbol{\alpha})]^{-1} \tag{4.52b}$$

where $\boldsymbol{T}_w^{\dagger}$ is the pseudoinverse and $\boldsymbol{W}$ is the weighting matrix.

The process of finding the constrained feasible actuator input $\boldsymbol{u}$ from the desired output force $\boldsymbol{\tau}$ is described as a model-based optimization problem. The optimization problem is a constrained optimization problem as the thrusters have a constrained thruster force, and the rotation of the thrusters have a constrained rotation speed. For any rotational thruster, there are numerous possible solutions to the desired output force. The optimization problem then becomes non-convex, which is harder to solve. The formulation of this optimization problem is called a quadratic programming (QP) problem. The definition of a simple QP problem formulation of the current optimization problem is

$$J = \min_{\boldsymbol{f},\boldsymbol{s}} \left\{ \sum_{i=1}^{r} \boldsymbol{f}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{f} + \boldsymbol{s}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{s} + (\boldsymbol{\alpha} - \boldsymbol{\alpha_0})^{\mathsf{T}}\boldsymbol{\Omega}(\boldsymbol{\alpha} - \boldsymbol{\alpha_0}) \right\} \tag{4.53a}$$

subject to:

$$\begin{aligned} &\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} = \boldsymbol{\tau} + \boldsymbol{s} \\ &\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \\ &\Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha}_{max} \end{aligned}$$

where $\mathbf{W}$ is the weighting matrix for the force vector, $\boldsymbol{s}$ are the slack variables, $\boldsymbol{f}$ are the thruster forces, $\boldsymbol{\Omega}$ is the weighting matrix for the servo angles, and $\Delta\boldsymbol{\alpha}$ are the changes in the servo angles.

4.5 Chapter Summary

This chapter summarized the theory behind the applied motion control algorithms. First, the chapter discussed the implementation of the reference filter serving as the guidance system on the vessel. Additionally, the 3 DOF PID controller is discussed in further detail. The chapter went on to discuss an adaptive backstepping approach for a new non-linear controller and the use of model reference feedforward. Additionally, the chapter detailed the relevant control parameters for their respective control algorithms. Lastly, a brief introduction to the thrust allocation rounds out the chapter.

Chapter 5

Autonomous Passenger Ferry Platform

The prototype milliAmpere is a small autonomous passenger ferry designed and developed by NTNU. milliAmpere serves as an essential experimental platform for research in the field of autonomous marine vessels. The development of milliAmpere started in 2017 and is built in a 1:2 scale of the intended final design.

Figure 5.1: The experimental platform milliAmpere during experimental tests in May 2019.

5.1 milliAmpere

The ferry currently relies on two separate sensors for the navigation system. The main sensor is the Vector VS330 GNSS Compass [Hemisphere]. This sensor measures the position, velocity, and heading data used in the navigation system. The second sensor is an IMU from XSENS [Xsens]. The IMU provides acceleration and angular velocity data for the navigation system. The GNSS and IMU are mounted at approximately $\boldsymbol{r}^b_{gnss} \approx [0, 0.975, -2.33]^\mathsf{T}$ and $\boldsymbol{r}^b_{imu} \approx [-0.78, 0, 0.07]^\mathsf{T}$ in relation to the vessel CO.

Component	**Make and Model**	**Specs**
IMU	Xsens MTi-20 VRU	Inclination stabilization
GNSS	Hemisphere Vector VS330 GNSS Compass	RTK enabled, dual reciever
Arduino	Arduino Mega	Microcontroller
Thruster	Torqeedo Pod Cruise 2.0FP	500N force output, reverse rotation capabilities
Servo	Intecno BLDC	32deg/s rotation speed
Angle encoder	Wachendorff WDGA 36a SSI	Absolute angle measurement
Servo signal generator	ICPDAS ET-7224	-10V - 10V signal generator

Table 5.1: Main components on milliAmpere used during tracking operations.

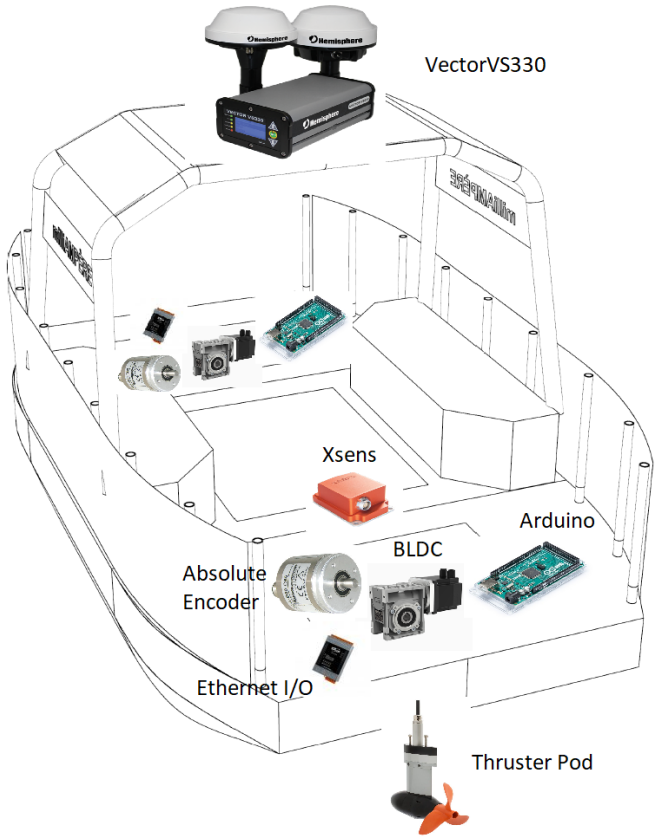


Figure 5.2: The design of milliAmpere and the placement of the various components. Design sketch of milliAmpere courtesy of Jon Boye Andersen, Innovation JBA.

The prototype has two thruster pods for propulsion, one in the stern and one in the bow. Two brushless direct current (BLDC) motors from Intecno controls the servo angles. The servos use a signal voltage of $-10V - 10V$ to control the rotation speeds for the servos. Two separate ICPDAS ethernet input/output (I/O) modules generate the signal for the respective servos. The servo control uses the angle measurements from the absolute encoders as feedback.

The absolute encoder measures the angular position as well as the number of full rotations performed. Figure 5.2 shows all the relevant components and their placements on the ferry. The setup of the Arduino, servo motor, absolute encoder and thruster is identical in the stern and the bow.

During the spring of 2019, the installation of a sensor rig served as an important upgrade for milliAmpere. The new sensor rig has radio detection and ranging (RADAR), a $360°$ panorama view infrared (IR) camera, and light detection and ranging (LiDAR). The instalment of the sensor rig enables milliAmpere with abilities of perception and other means

of localization.

5.2 Previous Work

NTNU has developed the ferry through student summer projects during the summers of 2017 and 2018. The author has taken part in the project during both summers. During the summer of 2017, the focus was on the construction of the vessel. All the different components, wires and accessories were installed. The intended design of most of the components is not suitable for the specified usage on milliAmpere. The design of custom software drivers enabled the usage of the different components for their purposes on the vessel. At the end of the summer, the ferry was fully operational with manual remote control. The initial set of sensors needed for transit was installed and interfaced. As discussed in the previous section, this included high precision RTK-GNSS and an IMU.

For the modelling equations of these measurements, see Section 3.1. To see more information on these measurements in regards to notation, see Table 2.1. For more information on relevant reference frames for use in transformations, see Section 2.1. These measurements provide the necessary input for autonomy without collision avoidance.

During the summer of 2018, the focus shifted to software development and software architecture. During that time, the thrust allocation, a basic 3 DOF PID controller, a basic navigation system using measurements directly and low-pass filtered differentiation, and a graphical user interface (GUI) were developed and implemented on the ferry. See Appendix E for a detailed view of the GUI.

5.2.1 Bollard Pull Test

A bollard pull test is a test that maps the thruster input to output force. The mapping from the bollard pull test ensures that the thrusters and servos get the correct input based on the desired force. Different steps on the input to the thrusters result in the ferry exerting a specific force. A weight tied to the stern of the ferry records the resulting force. The result of the test is plotted in Figure 5.3. The parameters found to fit this mapping was of a 5^{th} order curve fit.

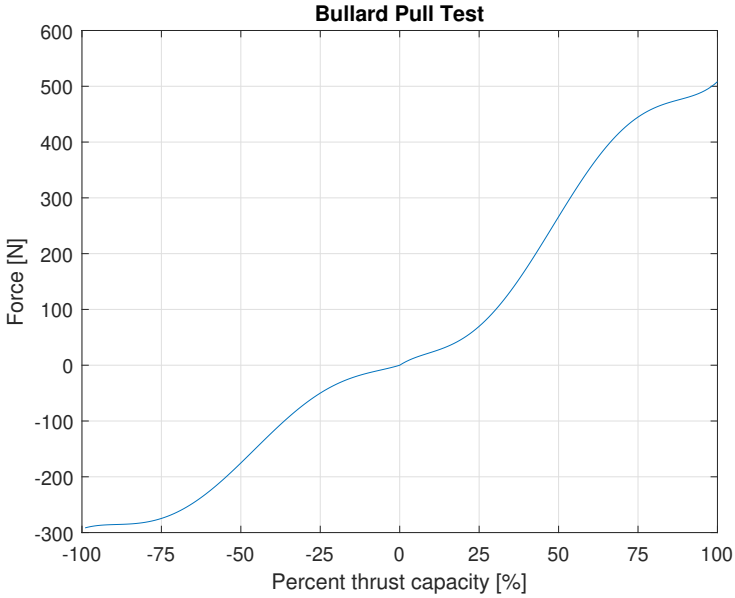


Figure 5.3: Bollard pull test performed during the summer of 2018. The test data results in a mapping from percentage of thrust capacity to generated thrust force. The mapping is a 5^{th} order curve fit.

The ability of the servos to rotate 360 degrees enables the vessel to set a force vector at any given angle with each of its 500N thrusters. Figure 5.4 shows a visualization of all the force vector setpoints the ferry is capable of producing with a coarse resolution.

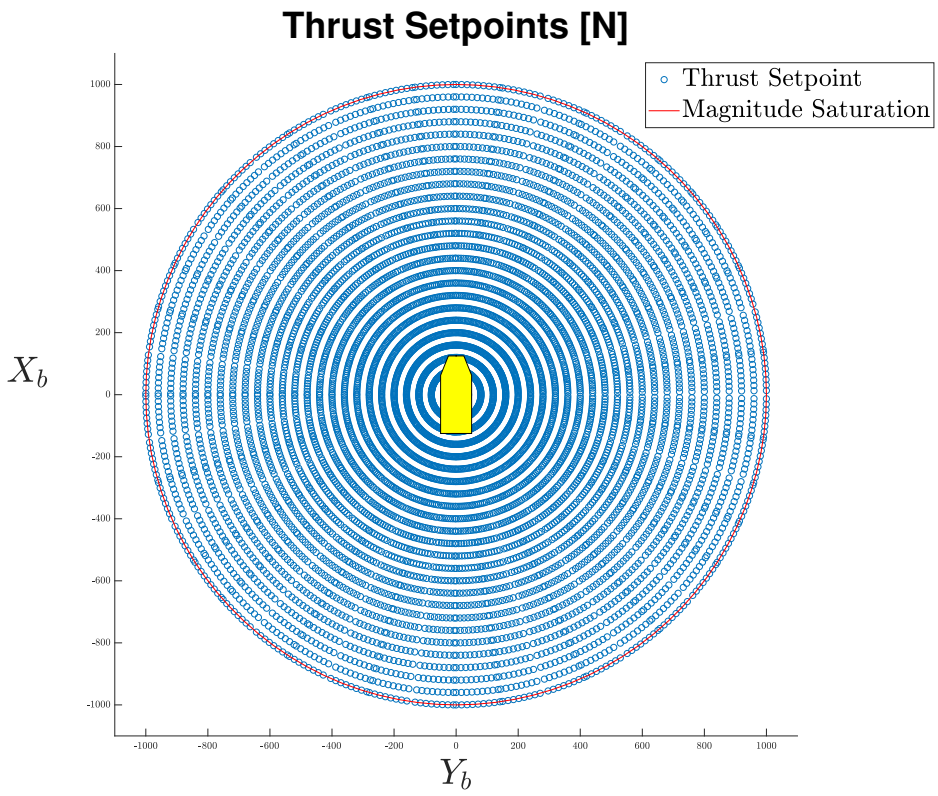


Figure 5.4: Force vectors in the BODY frame the vessel is capable of producing with both thrusters. The angle resolution is 1° and the thrust resolution is 40N.

5.2.2 GNC System

The ferry has gone through extensive testing, and experimental results were promising. The GNC system implemented over the summer of 2018 was a basic system with a simple reference filter, no observer, and no model-based feedforward or no wind feedforward. The summer of 2018 also saw the implementation of the thrust allocation. Despite the simplicity of the implementation, the GNC system performs accurate tracking.

A 4-corner test is commonly used to test the accuracy of a motion control system. A 4-corner test is conducted by having a square-shaped positional reference with different attitude references. Figure 5.5 shows a visualization of the concept of a 4-corner test.

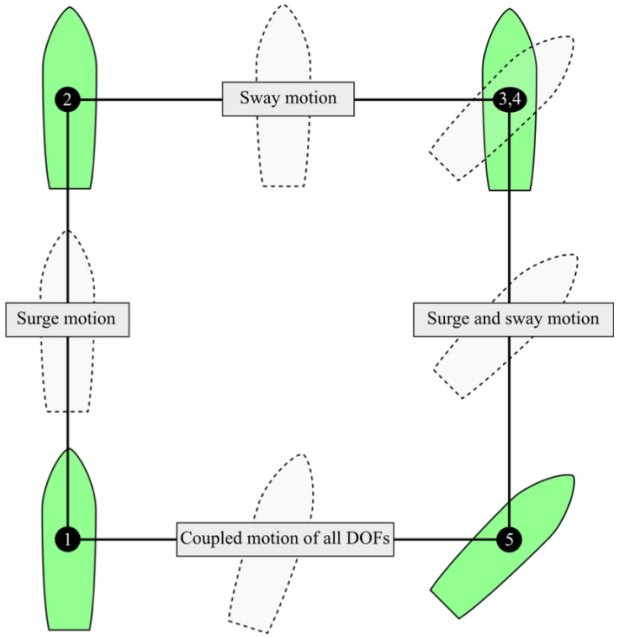


Figure 5.5: Concept of the 4-corner test. The vessel performs a surge motion, a sway motion, a yaw rotation, a coupled surge and sway motion, and a fully coupled motion. Image Courtesy of [Lyngstadaas, 2018].

Figure 5.6 shows the result from the 4-corner test. The main issue with the motion control system was the lack of heading stability.

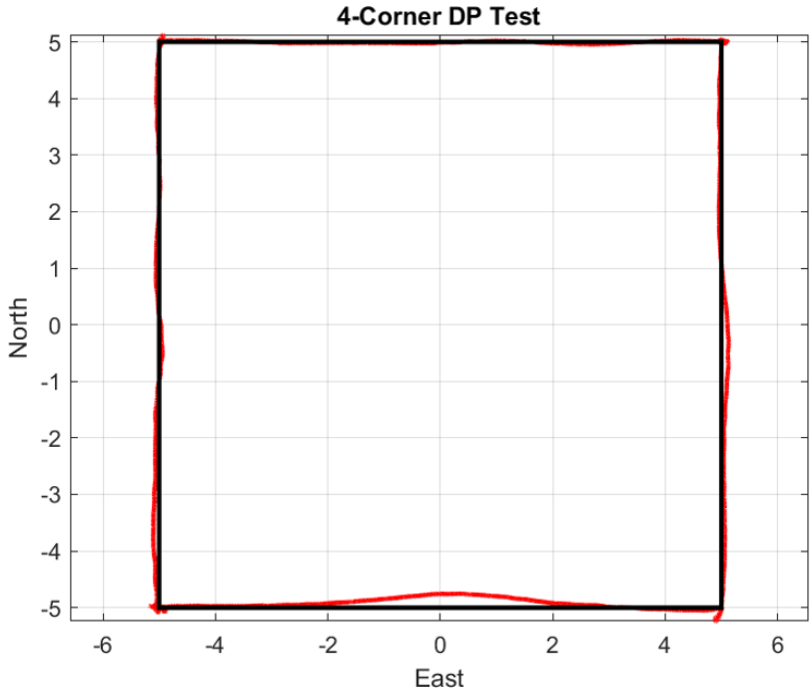


Figure 5.6: 4-corner test for the 3 DOF PID controller performed in pristine conditions during the summer of 2018.

In Figure 5.7, the measurements are separated into individual plots. The positional accuracy in north and east were almost spot on, but the heading tracking struggled and suffered from oscillations. It is important to note that the conditions during the 4-corner test were pristine.

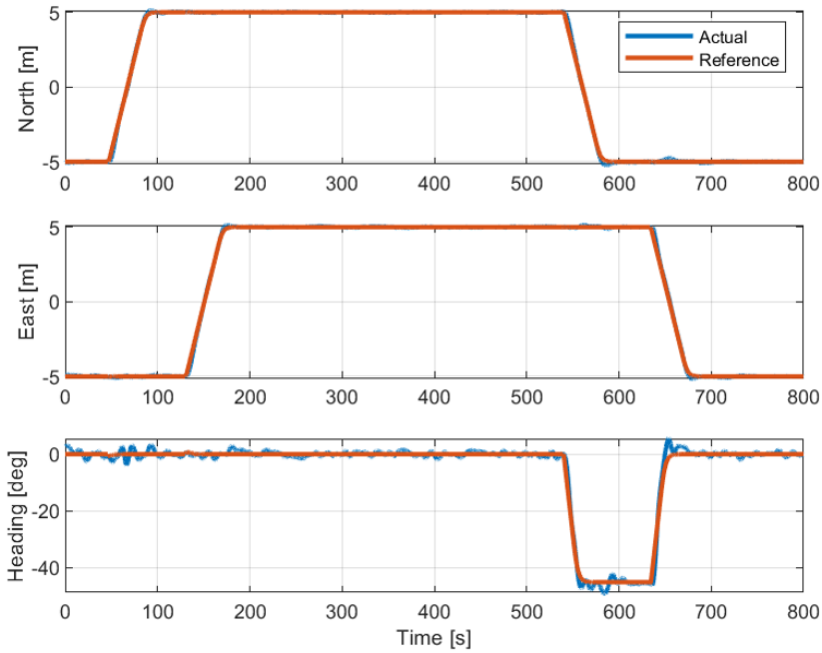


Figure 5.7: Responses for individual DOF during 4-corner test performed in pristine conditions the summer of 2018.

5.3 Platform Improvement Potential

The platform has areas of improvement to enhance overall performance during tracking operations. The most substantial restriction is slow thruster dynamics. The delay caused by the thrusters ramping up and down, coupled with the slow rotational speed of the servos, cause the platform performance to suffer when the need for more considerable thrust rate change is needed.

Due to the platform's unique design with a shallow draft, the vessel is very susceptible to wind forces. The shallow draft means less wind force is necessary to dislocate the vessel from its desired trajectory. Another design feature that negatively impacts the vessel characteristics is the flat-bottomed hull and the small ratio between the length and width of the hull. The flat-bottomed, square-like design of the vessel makes it directionally unstable.

The vessel also suffers from a quite substantial message delay on the servo control. With the current set up, it takes 0.5 seconds for the servo to act on the setpoint the on board

computer (OBC) requests. This delay further adds to the overall delay in the thruster dynamics. Removing the delay will improve the thruster dynamics and, in turn, the overall trajectory tracking accuracy.

5.4 Chapter Summary

This chapter presented the experimental platform used in the thesis. The chapter presented the design, the components and the previous work. The previous work consists of a previous implementation of a GNC system and includes some experimental results. The chapter also discussed some platform improvement potential for improving overall tracking performance for the vessel.

Chapter 6

Simulation Results

Two separate simulated environments are used to verify the different implementations. For the verification of the performance of the ESKF, simulated measurement data is used to mimic the actual sensor data. For the new controllers, the simulated environment uses a vessel model developed in [Pedersen, 2019]. These separate environments enable verification of both the ESKF and the new adaptive controllers.

6.1 Navigation Results

The navigation system evaluates the performance with the use of a node that mimics sensor data from the GNSS and IMU sensors. The imitation node provides data in real-time to test the ESKF in a realistic setting. The simulated environment enables the ability for necessary debugging and performance tests for the ESKF. A discussion in Section 6.1.1 details the performance tests for the ESKF. The filter is made to function with the same sensor data provided on the vessel. This approach will simplify the process of filter integration with the rest of the system on milliAmpere. The simulated sensor data includes simulated noise and perturbations. The added noise in the simulations ensures that the simulations are as close to any real-life scenario as possible.

The ESKF can estimate states based on absolute position data alone. This variation of the filter serves as a benchmark for performance. Other sensor data, such as velocity and compass measurement updates, are implemented in the ESKF. These filter variations are used as a comparison to evaluate precision and accuracy compared to the filter variation with only position error correction.

6.1.1 Filter Consistency

An essential aspect of the ESKF is the tuning of the $\mathbf{Q}$ and $\mathbf{R}$ matrices. It is beneficial to be able to confirm that the values in these matrices optimize filter performance. The

confirmation method is especially useful when the characteristics of the sensors are not necessarily known. Filter consistency is a way of validating the tuning of these matrices. Two tests often used to check for filter consistency are outlined in [Brekke, 2018] and [Bar-Shalom et al., 2002]. The first is the NEES test

$$\varepsilon_k = (\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T} \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \mathbf{x}_k) \tag{6.1}$$

where $\hat{\boldsymbol{x}}_k$ is the estimated state, $\boldsymbol{x}_k$ is the true state and $\mathbf{P}_k$ is the covariance matrix for iteration k. The second is the NIS test given by

$$\varepsilon_k^v = \boldsymbol{\nu}_k^\mathsf{T} \mathbf{S}_k^{-1} \boldsymbol{\nu}_k \tag{6.2}$$

where $\boldsymbol{\nu}_k$ is the innovation for measurement update k, and $\mathbf{S}_k^{-1}$ is the innovation matrix in the Kalman gain.

The NEES test is a tool for testing the consistency of the estimates in comparison to the ground truth. This test is dependent on knowing the actual state. The NIS test is a consistency test based on the innovation between the estimated state and the measured correction data. The latter is suitable for real-life scenarios where the true state is not known. The results from these tests should mimic that of a series of the norm of a three dimensional zero-mean unit-variance Gaussian random vector. If this is the case, the estimation error is modelled as white noise, indicating optimal estimation. Figure 6.1 shows a comparison between a NEES distribution and a 3 dimensional Gaussian distribution.

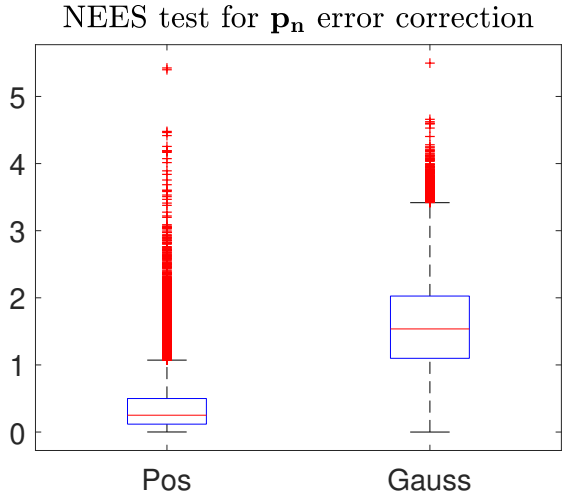


Figure 6.1: NEES test concept for position error between estimated position and true position in the ESKF.

As shown in the example, the test results do not mirror the Gaussian distribution completely. However, this case is satisfactory. The NEES/NIS test results that have larger values than the Gaussian distribution itself are far worse as this can lead to divergence. Both cases will lead to suboptimal estimation, but the former is preferred. Unfortunately, the non-linearities in the ESKF make it hard to make guarantees regarding filter consistency.

6.1.2 Position Data Error Correction

Figures 6.2 and 6.3 shows the filter estimates with position error corrections.

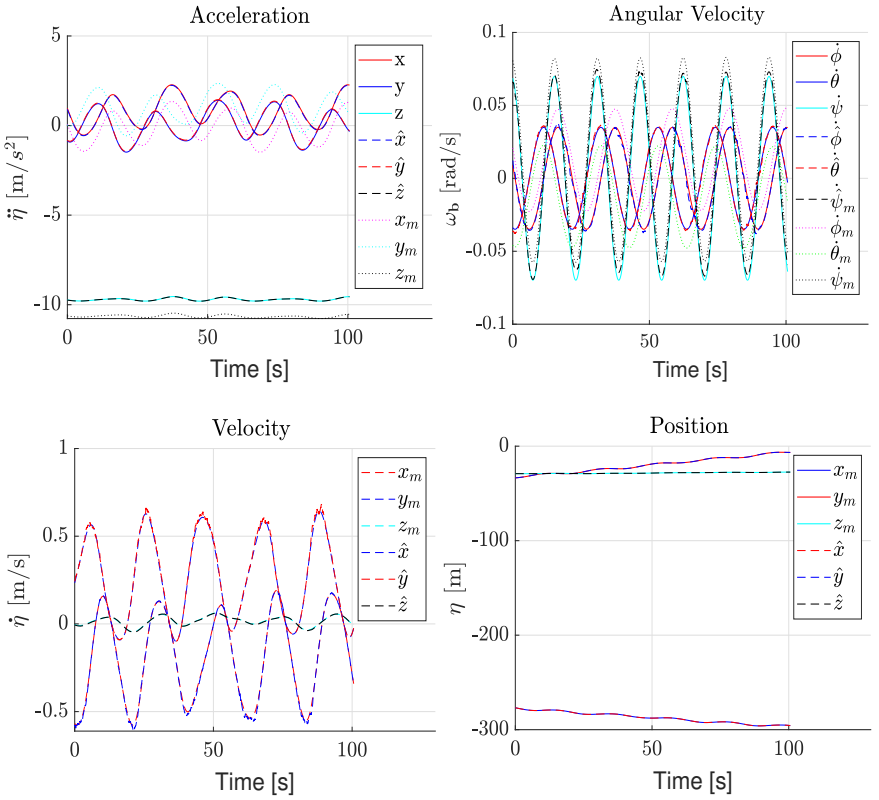


Figure 6.2: Estimation of acceleration, angular velocity, velocity, and position with position error correction during a simulation. The clean variable is the true state, the variable with hat notation is the estimated state, and the variable with the subscript m is the measured state.

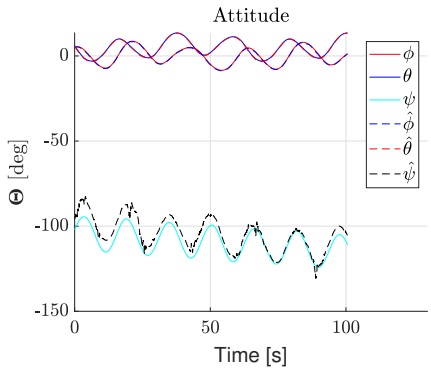


Figure 6.3: Attitude estimation with position error correction during simulation. The clean variable is the true state, the variable with hat notation is the estimated state.

With just the position error correction, the estimation of the non-measured states is performed well, except for the heading ψ. Even though the estimation of ψ is somewhat inaccurate, the estimation never diverges and is kept close to the actual ψ during simulations. The consistency test shown in Figure 6.4 shows that the filter behaves as desired by having an estimation error distribution smaller than the Gaussian distribution, thus supporting that the filter will not diverge. The introduction of compass measurements will resolve the struggle with the estimation of the heading.

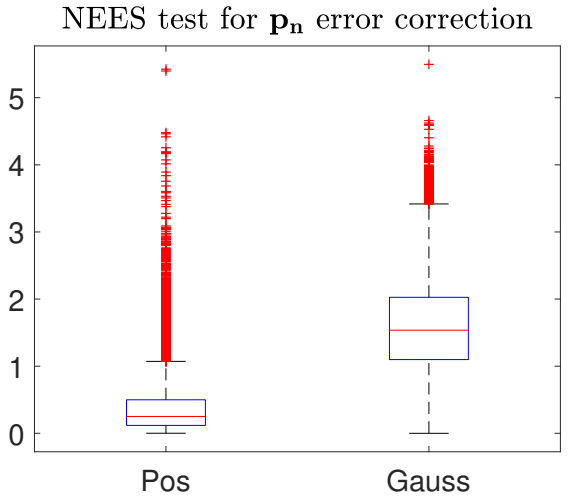


Figure 6.4: NEES test result for error between position estimates and true position compared to a three dimensional Gaussian during simulation.

The NEES test gives satisfying results with position data error correction. The estimation

error distribution is smaller than that of the Gaussian distribution. The estimation error distribution satisfies the remarks outlined in Section 6.1.1, indicating well-tuned parameters.

Heading Observability with Position Data Error Correction

It is not straightforward how the heading is observable with only the use of position data error correction. The heading is observable through the acceleration measurements [Gade, 2016]. In the case of position data error correction only, the ESKF relies on an accelerated system to achieve heading observability. The larger the accelerations are, the better the observability, and thus more accurate heading estimate. The heading is possible to estimate with this approach only if the obtained position measurement has sufficient accuracy and rate. Unfortunately, this method requires a sufficiently large vessel acceleration to be applicable. In the simulations, the accelerations are unreasonably large in comparison to the real-life application of milliAmpere. A correct estimate of the heading with only position data error correction is unlikely to be achievable in a real-life scenario with milliAmpere.

Successful estimation of the heading is also possible with velocity measurements only. The required vessel acceleration is then less in comparison to position measurements. The lesser need for acceleration is a result of the separation of velocity measurement noise and the integrated acceleration is quicker in comparison to the position measurement. The quicker separation is due to the further integration of the acceleration needed to obtain the position. Estimation of heading with velocity measurements only will not be examined in more detail in this thesis as the ESKF utilizes compass measurements.

6.1.3 Position and Compass Data Error Correction

The onboard GNSS provides yaw measurements. As mentioned in the previous section, the orientation estimate lacks accuracy in the yaw position. Compass measurements are introduced to remedy this shortcoming.

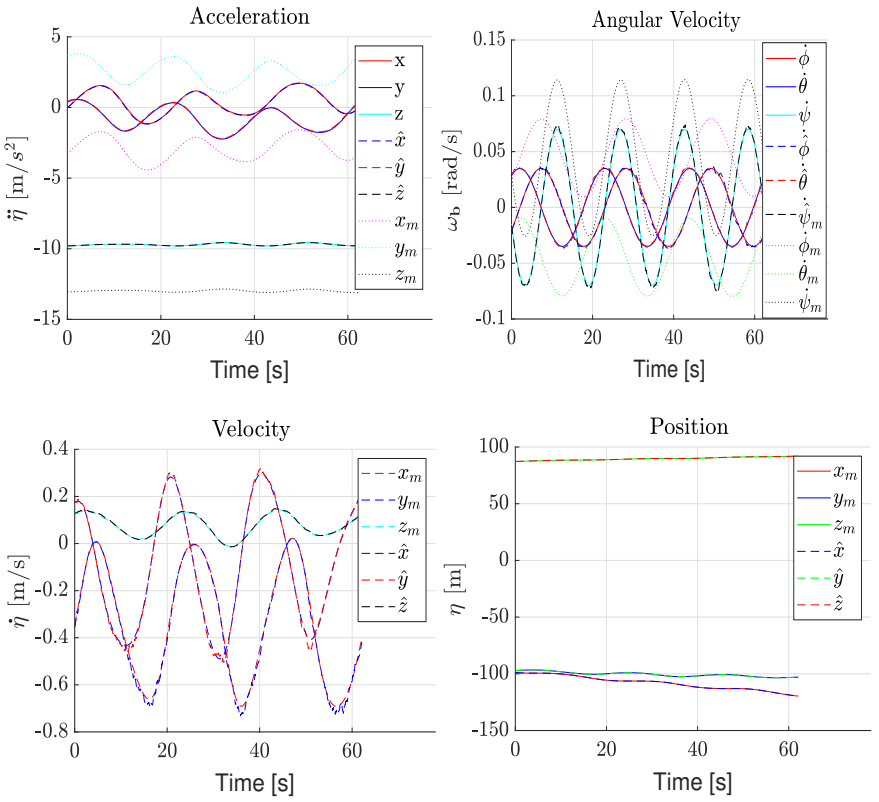


Figure 6.5: Estimation of Acceleration, angular velocity, velocity, and position with position and compass error correction during simulation. Subscript m denotes the measured state, hat notation denotes the estimated state, and the clean variable is the true state.

The introduction of compass measurements increases the accuracy of the estimated angular velocity. The increased observability, because of the compass measurements, ensure impeccable estimation of all states.

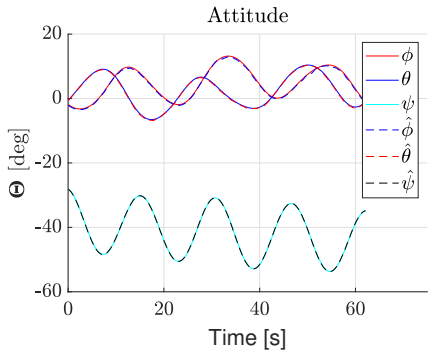


Figure 6.6: Attitude estimation with position and compass error correction during simulation. The clean variable is the true state, the variable with hat notation is the estimated state.

The introduction of the compass measurement has the desired effect. The heading now coincides with the actual heading, and the attitude is correctly estimated. The correct estimate of attitude means an accurate estimation of the complete six DOF vessel model. The estimation of the six DOF state vector is valuable not only for the motion control system, but it also allows for accurate transformations between the different reference frames. The results from the NEES test in Figure 6.7 further supports stability of the filter by abiding to the rules of filter consistency in Section 6.1.1.

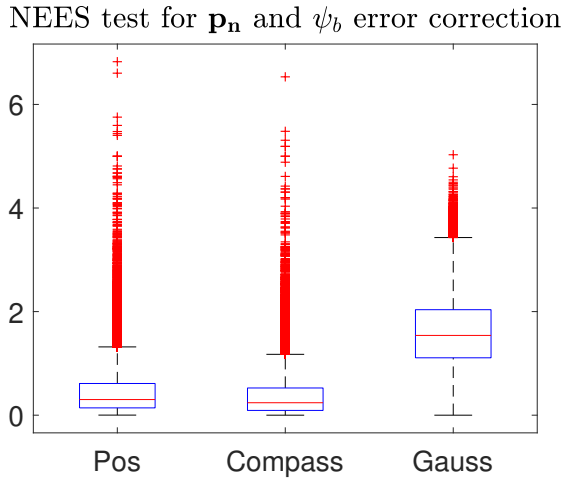


Figure 6.7: NEES test with position and compass estimation error during simulation.

6.1.4 Position, Velocity and Compass Data Error Correction

The addition of planar velocity error correction further improves the state estimation accuracy. The planar velocity measurement is a transformation of the speed and course measurements provided by the GNSS. Together with the rest of the estimated states, Figure 6.8 shows the improvement in the velocity estimate.

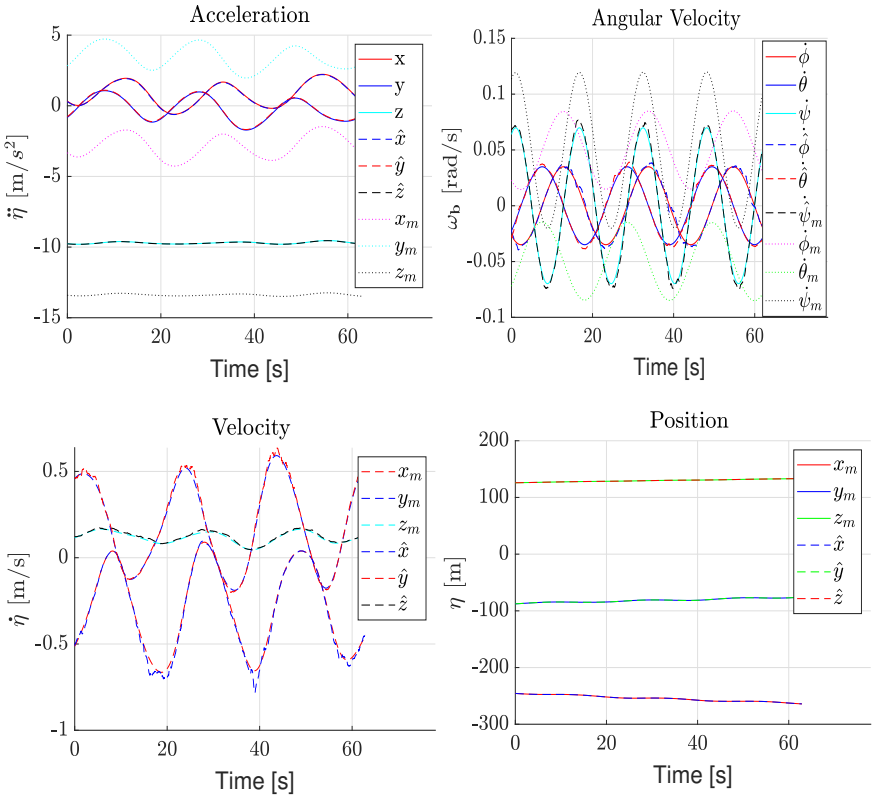


Figure 6.8: Estimation of Acceleration, angular velocity, velocity, and position with position, velocity, and compass error correction during simulation. Clean variable denotes true state, hat notation are estimated states, and subscript m is the measured states.

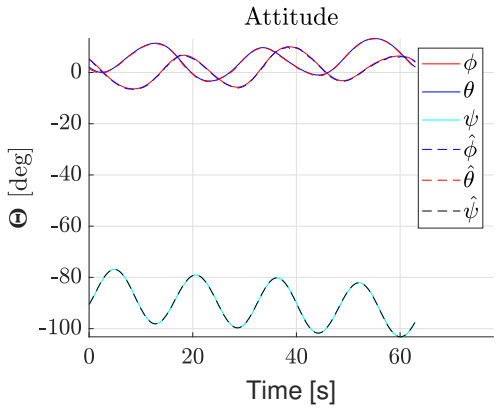


Figure 6.9: Attitude estimation with position, velocity, and compass error correction during simulation.

The introduction of the velocity measurement, $\mathbf{v}^n = [\dot{\mathbf{x}}^n, \dot{\mathbf{y}}^n]^\mathsf{T}$, does not have a profound effect on the filter performance. However, there are some subtle improvements. The attitude estimation is now more accurate and more consistent, allowing for more consistent transformation of the GNSS measurement to the CO. The velocity estimates are also smoother, although not perfect. As the true state is known, all errors in the estimates are available in three DOF. Figure 6.10 shows the comparison between the estimation error and a three DOF Gaussian distribution.

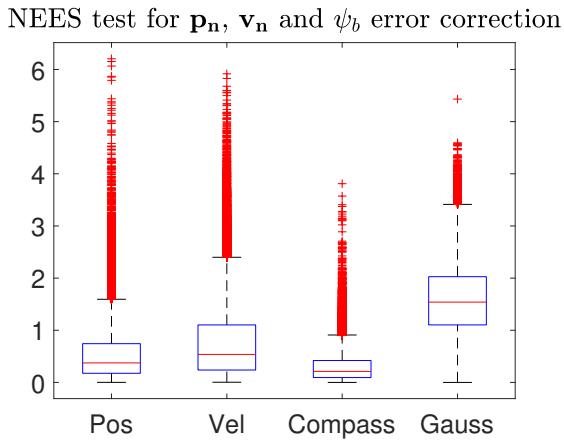


Figure 6.10: NEES test with position, velocity, and compass error correction during simulation.

The NEES test results are satisfactory. All the NEES tests for the different measurements

have smaller distributions than the Gaussian distribution. These tests further support the achievement of filter stability.

6.1.5 Discussion

The simulations of the ESKF show that the filter behaves as expected. The ESKF successfully estimates the state vector for all three variations of the filter. The only issue is that the filter appears to have some issues with the observability of ψ when there is only position error correction. One reason is that the added stochastic noise reduces the overall observability. Larger accelerations will also improve heading estimation accuracy.

The inability to accurately estimate ψ gives cause to concern for the filter running in a real scenario with less acceleration and with more dominant noise. However, this is for the variation that only uses position data as error correction. The inability to estimate ψ accurately does not cause concern for overall filter accuracy, as the final implementation uses more measurements for error correction.

Another critical remark is that the estimation of the state vector is stable in real-time. State estimation in real-time is an important milestone as it is crucial for any application in a real-life. The filter has a runtime which allows for very high order approximation of the transition matrix. The ESKF runs successfully in real-time with an approximation of the transition matrix as high as 7^{th} order. 7^{th} order approximation might seem overkill, but increased approximation accuracy is highly valued to ensure stability.

Additionally, the maximum runtime for one iteration of the filter, which occurs when there is an update for all three measurements during one iteration, is clocked at $2.6 \cdot 10^{-3}s$. A 5^{th} order approximation of the transition matrix was running with the achievement of this runtime. This runtime was clocked on a virtual Ubuntu machine on a laptop with a minimum amount of allocated processing power. The relatively low computational cost of running the filter will come in handy down the line with the introduction of more nodes to the software stack on the OBC.

6.2 Motion Control Results

The control algorithm currently being used is a 3 DOF PID controller. The introduction of feedforward will easily improve the transient response in comparison to the response shown in Section 5.2.2. However, the model is never 100% accurate. The introduction of adaptive control is an attempt to improve the performance of the current controller. The principles of adaptive control will be used to estimate the model uncertainties and external disturbances. The simulated environment used for verification is set up as a Robot Operating System (ROS) node to interact with the entire system in real-time.

6.2.1 Simulations With Unmodelled Dynamics

It is essential to have a simulation environment as close to reality as possible. A realistic simulated environment will ensure that the real-life applications of the implementation are

attainable with little to no tweaking. The measurements from the different sensors are subject to measurement noise. The simulated environment models the measurement noise as a Gaussian process based on the datasheet for the particular measurement. Table 6.1 details the standard deviation of the added measurement noise in the simulated environment.

Measurement	Standard Deviation
$\boldsymbol{x}$	10 mm
$\boldsymbol{y}$	10 mm
ψ	0.05 deg
$\dot{\boldsymbol{x}}$	0.01 m/s
$\dot{\boldsymbol{y}}$	0.01 m/s
r	0.05 deg/s

Table 6.1: Simulated measurement noise in the simulated environment for the measurements used in the motion control algorithms on milliAmpere.

The measurement noise for the measurements are random samples drawn from a Gaussian distribution defined by the probability density function

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{6.3}$$

where the mean $\mu = 0$, and the standard deviation σ for the respective measurement is according to Table 6.1. The simulated environment applies the measurement noise to the correct measurement before use.

To further make the simulated environment more accurate towards a real-life scenario, an external force vector is added to interact with the vessel. The external disturbance mimics that of a force induced by irrotational ocean currents. The definition of the force vector is in the NED frame. All the simulations performed from here on out are all subjected to the external force $\boldsymbol{\omega}_d^n = [200\text{N}, 200\text{N}, 50\text{Nm}]$. Furthermore, all model matrices are perturbed by $\pm 40\%$ to act as the inherently imperfect model. The sum of all unmodeled dynamics is then collectively expressed in the term $\boldsymbol{\omega}^n$.

The simulations use a static magnitude saturation function based on the maximum thrust output from a single thruster. The static magnitude saturation defines the thruster force saturation as

$$\boldsymbol{\tau}_{ms} = [500\text{N}, 500\text{N}, 1000\text{Nm}]^\mathsf{T}$$

6.2.2 Performance Metrics

A set of performance metrics are used to evaluate the performance of the different control algorithms objectively. [Sørensen and Breivik, 2015] uses four performance metrics for the evaluation of the performance of a motion control algorithm. This thesis uses the same performance metrics. These performance metrics are the integral of the absolute error (IAE)

$$IAE(e) = \int_0^t |e| dt, \tag{6.4}$$

which is purely the error over time with no other considerations. The IAE metric indicates how well the vessel tracks the desired path over time. The integral of the square of the error (ISE)

$$ISE(e) = \int_0^t e^2 dt, \tag{6.5}$$

which weigh larger errors more than smaller errors and gives a metric for how well an algorithm eliminates larger deviations from the intended trajectory. The integral of the absolute error multiplied by time (ITAE)

$$ITAE(e) = \int_0^t t|e| dt, \tag{6.6}$$

which penalize errors that are present over more extended periods. This metric will heavily weigh stationary errors that the tracking operation does not eliminate. The last metric is the integral of the absolute error multiplied by energy consumption (IAEW)

$$IAEW(e) = \int_0^t |e| dt \int_0^t P dt, \tag{6.7}$$

which combines the tracking error and the power consumption to see which algorithm has the best tracking vs power consumption. In these metrics the variables e and P are given by

$$e = -sin(\psi)(x - x_d) + cos(\psi)(y - y_d) \tag{6.8}$$

and

$$P = ||\boldsymbol{\nu}^T \boldsymbol{\tau}|| \tag{6.9}$$

6.2.3 Non-Adaptive Controllers

The first simulations show the individual responses from the non adaptive controllers. The controllers in question are a 3 DOF PID controller without feedforward, a 3 DOF PD controller with feedforward, and a 3 DOF PID controller with feedforward.

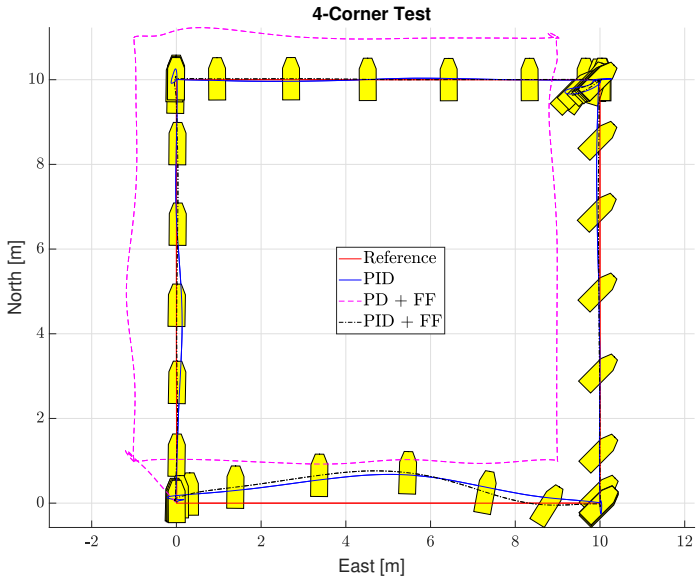


Figure 6.11: 4-corner test for non-adaptive controllers during simulations.

As expected, the PD controller with feedforward performs poorly. Without compensation for unmodelled dynamics in any way, the controller falls short and never converges to the desired state. The flaws for the other controllers are also apparent. The fully coupled motion, described in Figure 5.5 as the $5 \rightarrow 1$ stage, causes large deviations from the desired path. With $\pm 40\%$ model perturbation, the feedforward causes more harm than good at this stage.

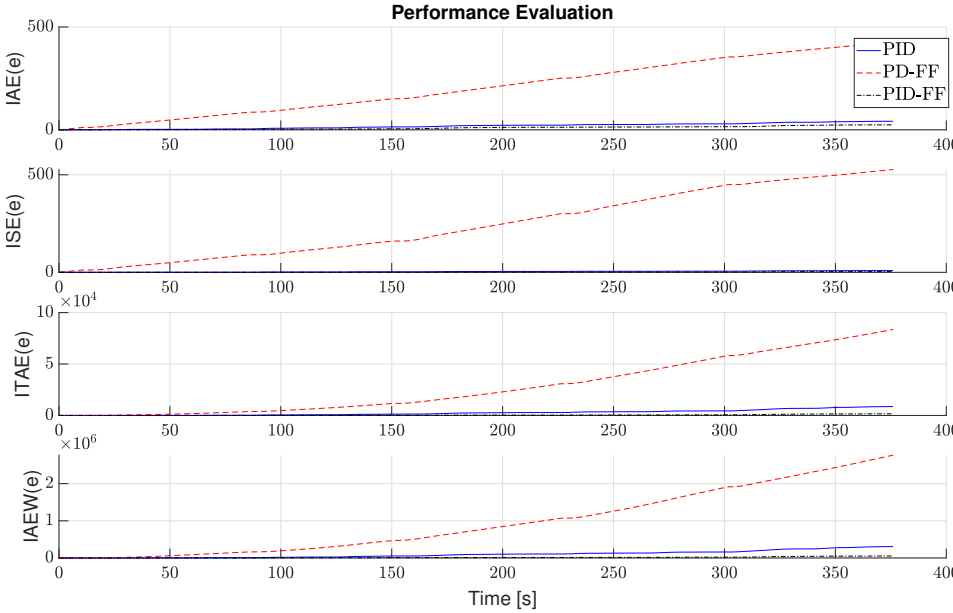


Figure 6.12: Performance metrics for non-adaptive controllers during simulations.

The performance metrics show that adding feedforward to the PID controller increases overall performance. The model reference feedforward achieves improved transient response and better overall performance for the controller. The feedforward ensures that the controller more accurately follows the desired path during phases of acceleration. The difference in transient performance is most notable in the corners of the 4-corner test in Figure 6.11. The same improvement is visible in Figure 6.13.

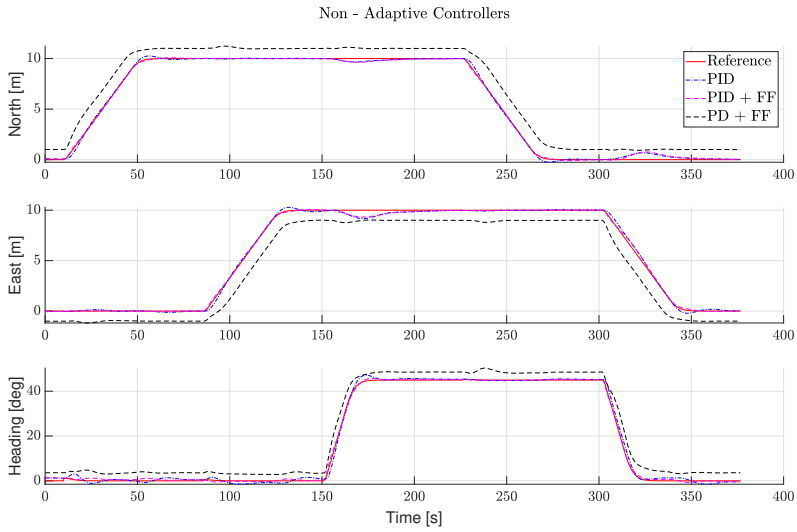


Figure 6.13: Responses for the individual DOF for non-adaptive controllers during simulations.

Due to the addition of stochastic measurement noise in the simulated environment, the desired control outputs have a high-frequency oscillation. This is shown in Figure 6.14.

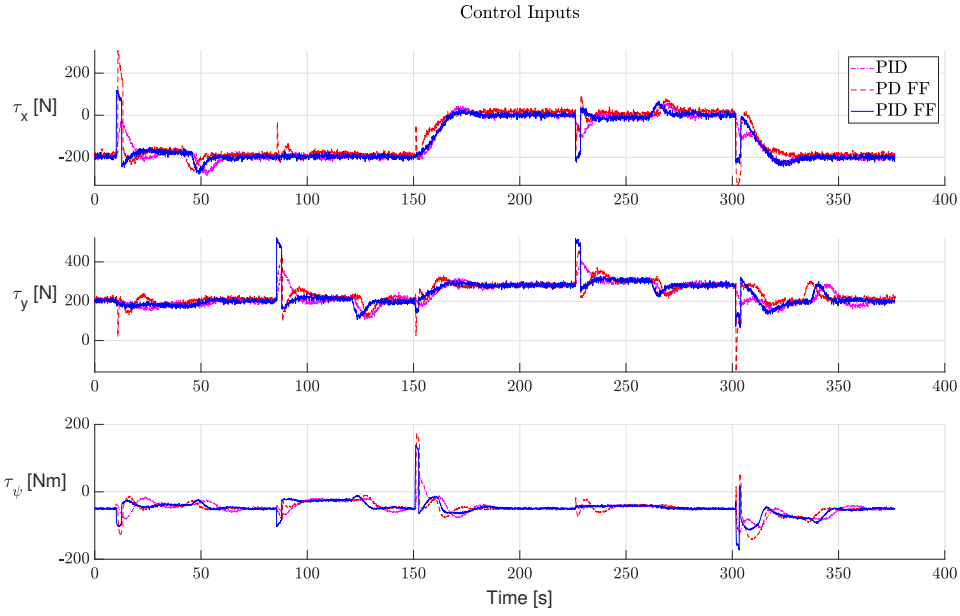


Figure 6.14: Control input for non-adaptive controllers during simulations.

The control inputs showcase the effect of the feedforward. With feedforward, a predicted force is immediately added based on the desired acceleration and velocity calculated by the reference filter. This predicted force causes the control input to have a powerful surge in phases of acceleration. In Figure 6.14, this is easily seen by inspection of the desired control input from the PID controller and how it differs from the others.

6.2.4 Adaptive Controllers

As noted in Section 4.3.2, the adaptive term enables an alternate way of estimating the uncertainties and external disturbances. This new design has a significant advantage compared to the traditional PID controller. The adaptive control can estimate all the unmodeled dynamics quickly, whereas the integral term in the traditional PID controller needs to build up over time. The rapid convergence of unmodeled dynamics provided by the adaptive controller gives an advantage over the PID controller and will ensure improved tracking.

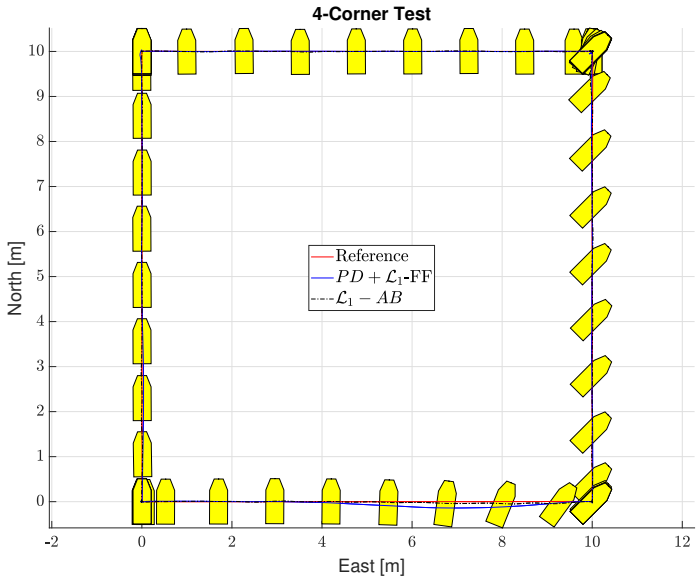


Figure 6.15: 4-Corner test for adaptive controllers during simulations.

The results of the 4-corner test with the introduction of adaptive control are dramatically improved. The adaptive control ensures almost perfect tracking, even with 40% model perturbation and an external disturbance vector of $\boldsymbol{\omega}_d^n = [200\text{N}, 200\text{N}, 50\text{Nm}]^\mathsf{T}$. The $\mathcal{L}_1$-AB controller has slightly better tracking, but at the cost of increased actuator usage. Figure 7.34 further shows the increased actuator usage.

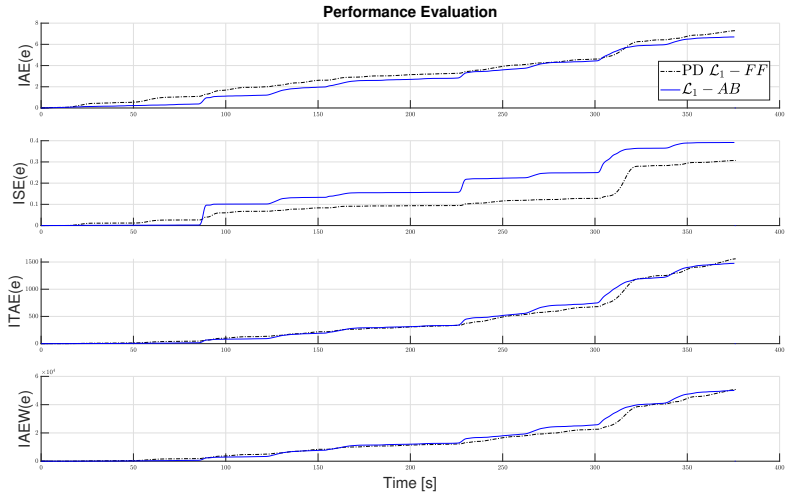


Figure 6.16: Performance metrics during the 4-corner test for adaptive controllers during simulations.

The dramatic improvement in performance is evident by comparing figures 6.12 and 6.16.

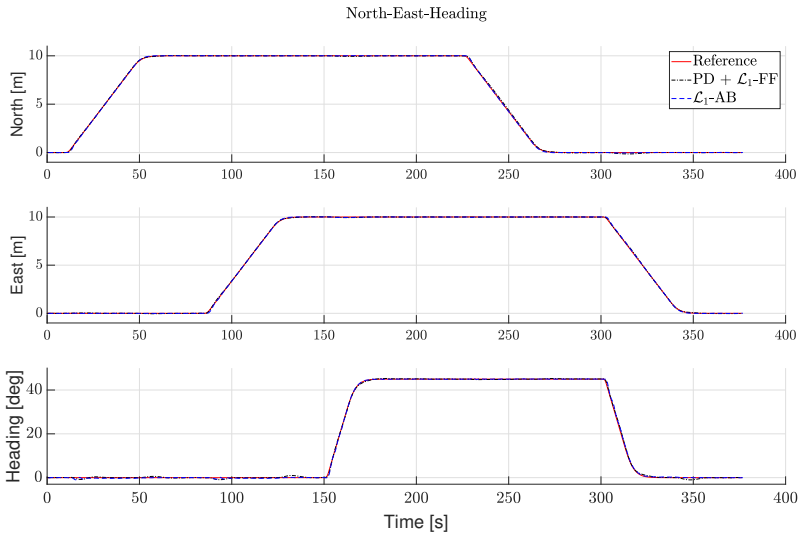


Figure 6.17: Responses for the individual DOF in the 4-corner test for adaptive controllers during simulations.

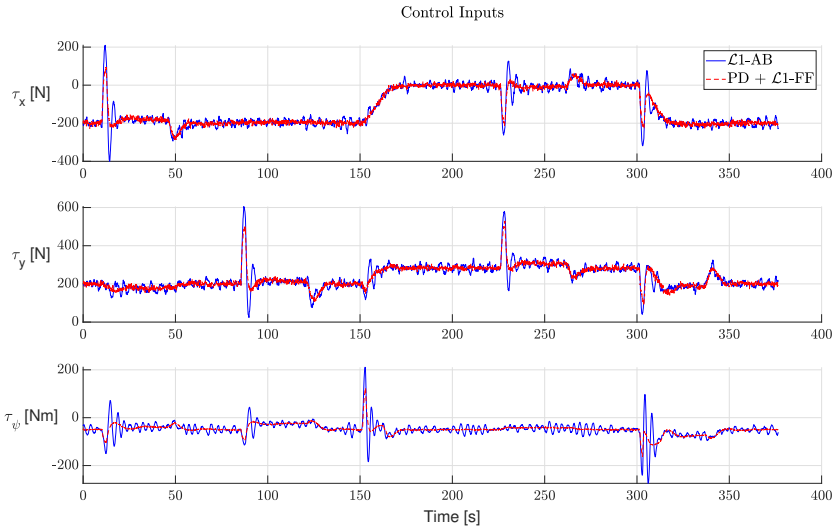


Figure 6.18: Control input for adaptive controllers during the 4-corner test in simulations.

The usage of the $\mathcal{L}_1$-AB controller increases the actuator wear and tear. However, reducing the gains given in (4.47) will reduce the aggressive behavior evident in the actuator usage by the $\mathcal{L}_1$-AB controller. The response from the $\mathcal{L}_1$-AB controller does show that it is capable of more aggressive maneuvering. More aggressive maneuvering is useful in some instances and should be part of the consideration for the final choice of the controller. The high-frequency oscillations in the desired control output are due to the measurement noise present for $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$.

6.2.5 Discussion

All the simulations of the new controllers show improvement over the 3 DOF PID controller that currently reside on milliAmpere. Especially are the results that were performed using principles of adaptive control promising. The PD controller with adaptive feedforward show massive improvement over the PID controller with feedforward.

What is also essential to notice is how the adaptive controllers differ in actuator usage. The different actuator usage provides some flexibility regarding the desired behavior of the ferry. For a situation demanding aggressive maneuvering and accuracy, the $\mathcal{L}_1$ adaptive backstepping (L1-AB) controller is more suitable. If less actuator wear and tear and less power consumption outweigh the need for a slight improvement in accuracy, the PD controller with adaptive feedforward is the better choice. Regardless, both the controllers with adaptive control shows great promise and outperforms the traditional PID based controllers

by a considerable margin. The increased performance in the simulations is auspicious for the final performance these controllers will have on milliAmpere.

6.3 Chapter Summary

Chapter 6 presented all the results acquired from the simulations. The simulations of the estimation algorithm used a node that mimics the sensor data available. The filter ran in three different variations in the simulations, and the chapter presented a discussion for each of the variations along with the results.

This chapter also presented the simulations for the control algorithms. The new adaptive controller variations are compared to the already implemented 3 DOF PID controller using the 4-corner test. The results from the 4-corner tests are presented with the use of performance metrics and individual responses for the separate DOF for positions and control input.

Chapter 7

Experimental Results

This chapter outlines all the experimental results. The experimental results present the performance for the newly implemented controllers and the ESKF. The new algorithms serve as an improvement to the GNC system on milliAmpere, and it is imperative that both the ESKF and the controllers have stable and predictable behaviors to ensure safety on board. The filter estimates are from here on out compared to raw position measurements and the estimated velocities obtained through low pass filtered differentiation of the position measurements. These measurements and estimates will be referred to as the old navigation node and are denoted by the measured pose $\boldsymbol{\eta}$, and the velocity in body $\boldsymbol{\nu}$ achieved by transforming the low pass filtered differentiation of the measured pose from the NED frame to the BODY frame.

7.1 Test Area

The declaration of Trondheimsfjorden as a test site for autonomous ships in 2016 [SaeKhow, 2016] marked it as the first of its kind in the world. A suitable place to test the new implementations on the ferry is Havnebassenget by Brattøra. It is vital to have large safety margins as the ferry is a prototype and has no backup solution. Given the severity of the situation, if a crash is to occur, a big open area is favorable. As the traffic in this area is minimal, it serves as a perfect testing area for the ferry. The experimental tests performed are further detailed in Appendix C.

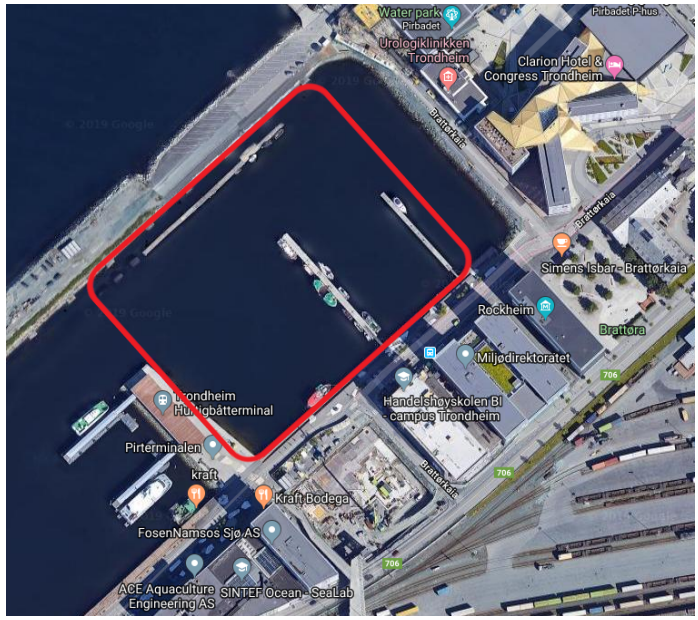


Figure 7.1: Test area for milliAmpere. Map provided from Google Maps.

7.1.1 Testing on the Experimental Platform

Testing implementations on an experimental platform is both a burden and a joy. The experimental platform allows for real-life testing that showcases the results in a much more satisfying way than through simulations. The ability to see actual real-life applications is both rewarding and exciting. On the other hand, the platform introduces difficulties that are not associated with simulations. The time consumption of set up, experimental testing, unforeseen events and other time thieves, further constrain an already constrained time frame. The time constraint became more and more apparent towards the end of the thesis as experimental tests were pushed repeatedly for the variety of reasons previously mentioned.

The test platform can also provide some logistical issues regarding performing experiments. Given that milliAmpere is a prototype, there are extra precautions concerning the operation of the vessel. Safety guidelines dictate that at least two people should always be on board the ferry in case of malfunction. The need for multiple people on board puts limitations on available time slots to conduct experiments. Another issue is that milliAmpere serves as a testing platform for other students. As many students rely on the platform, available testing time is further restricted. Other limitations such as scheduled demos, weather conditions and hardware issues mean that available time for testing purposes is limited.

A discovery of hardware issues on milliAmpere during Ocean Week in Trondheim meant pushing the majority of the experimental tests from the 9^{th} of May to 21^{st} of May 2019.

7.2 Navigation Results

The ESKF in its current state can utilize measurements for all states if available. Most of the initialization parameters in the filter can be set easily due to reliant accurate measurements given by the sensors in question. The nominal states in the ESKF that are not easily measured pose more of a significant impact on the initialization of the filter. More specifically, the nominal states for velocity, roll and pitch. The biases in the acceleration and angular velocity measurements are expected to be small, and the initial error values for the biases reflect that. The roll and pitch can not be reliably measured, and the velocity measurement is very noisy when at a standstill.

Proper initialization of the filter is essential as it decreases the convergence time. Filter initialization is necessary every time the system needs to reboot. During a reboot, it is reasonable to assume a moored state of the vessel. Thus, this state of the vessel is the basis for the filter initialization.

7.2.1 Inertial Measurement Unit Mounting

The IMU is currently mounted below deck with the lever arm $\boldsymbol{r}_m^b = [-0.78, 0, 0.07]^\mathsf{T}$. With the current hardware setup, there are no better alternatives to mount the IMU regarding CG. However, mounting the IMU close to CG in this case, comes at a cost. The mounting of the sensor is directly below deck on the inside of the hull. Any movement on deck causes spikes in the measurements. Consequently, the wild point filter drops many IMU measurements. Figure 7.2 shows the effects on the acceleration measurements induced by walking and stomping above on deck.

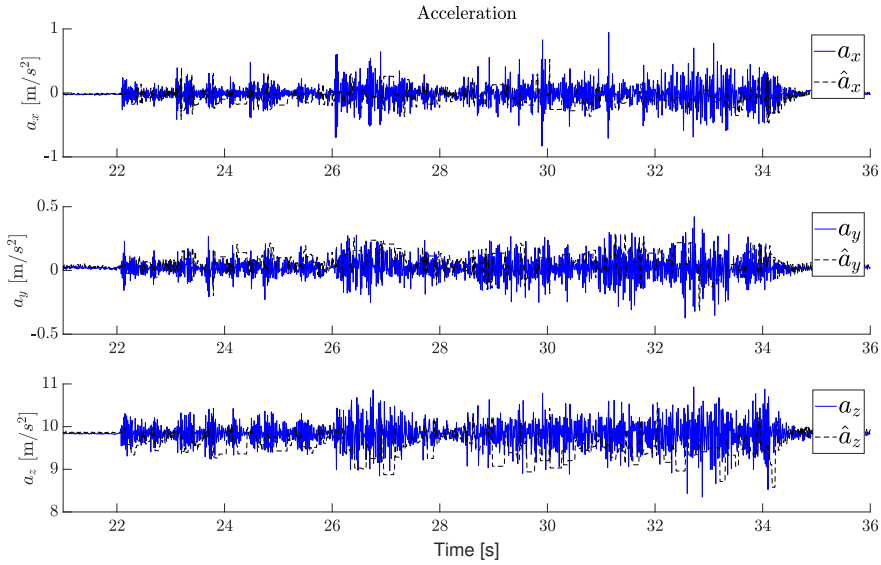


Figure 7.2: Induced acceleration in IMU measurements due to stomping and walking on deck.

The induced accelerations are quite large, and some spikes even cause deviations larger than $1m/s^2$. The wild point filter picks up a lot of the spikes, but if the vibrations occur over a prolonged period, the measurements approved by the wild point filter will not be satisfactory.

Another issue with the current mounting is that after inspection of the IMU, it shows that the IMU has a slight offset in the roll position. Figure 7.3 shows this offset. The offset is not ideal as the provided measurements from the IMU will be entangled with the gravity vector. The offset also causes the roll estimate provided by the IMU to have a slight offset. This inaccuracy is important to consider for the attitude comparisons later.

Figure 7.3: The mounting of the IMU. The sensor has a slight offset in negative roll.

The previous mounting of the IMU had the lever arm $\boldsymbol{r}_m^b = [0, 0, -2.33]^\mathsf{T}$. To illustrate another aspect of the importance of the IMU mounting, Figure 7.4 shows the resulting measured accelerations by the IMU with the old mounting. These are recordings during the 4-corner test in pristine conditions discussed in Section 5.2.2. Still, the measurements are greatly affected by the angular acceleration. The angular acceleration causes oscillations of the acceleration measurements with amplitudes as large as $0.15m/s^2$ with a period of two seconds.

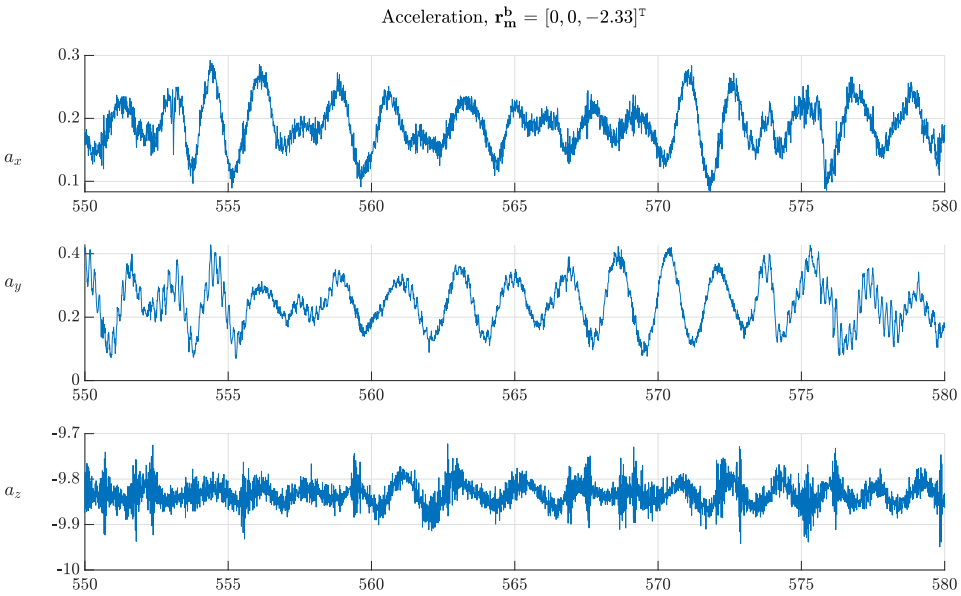


Figure 7.4: IMU acceleration measurements corrupted due to angular acceleration as a result of the lever arm $\boldsymbol{r}_m^b = [0, 0, -2.33]^\mathrm{T}$.

A solution that remedies both current shortcomings is a contraption rooted in the base of the hull. The contraption will extend the position of the IMU as close to CG as possible. With this solution, the IMU is minimally susceptible to be influenced by small vibrations and material deflections caused by people walking around on deck. Additionally, the acceleration measurements will be less affected by angular accelerations that are caused by being mounted away from CG. A new mounting solution also alleviates the observed offset in the roll angle. A new, improved mounting of the sensor will minimize the effects of both factors. Eliminating these factors will improve the overall performance of the ESKF.

7.2.2 Covariance Matrix Initialization

Before the state estimation algorithm can run in the current setup, all measurements have to be registered at least ones. These measurements are for position, velocity, compass, angular velocity and linear acceleration. Based on the initial measurement of the linear acceleration, the roll and pitch angles are estimated. See Section 3.1.1 for a more thorough explanation of this estimation. An accurate initial estimate of the state vector is then established based on these measurements.

The covariance matrix is initialized to express the uncertainty of the initial estimate of the state vector. The initial uncertainty estimates are set slightly larger than the initial measurement uncertainty to avoid any unwanted effects that might occur if the uncertainty estimates are set too small. For the states with no reliable measurements, the initial un-

certainty estimates are set slightly larger than strictly necessary to avoid any initialization issues. If the uncertainty in the initial estimates does not reflect the actual uncertainty of the measurements, the convergence time of the filter increases, or in worst case diverges. The entries in the initial diagonal covariance matrix are

$$\mathbf{P}_0 = \begin{bmatrix} \boldsymbol{p}_\sigma & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{v}_\sigma & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{\theta}_\sigma & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{a}_{b_\sigma} & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{\omega}_{b_\sigma} \end{bmatrix} \tag{7.1}$$

where

$$\boldsymbol{p}_\sigma = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \quad \boldsymbol{v}_\sigma = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \quad \boldsymbol{\theta}_\sigma = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \tag{7.2}$$

where the initial error estimate of the attitude $\boldsymbol{\theta}_\sigma$ is expressed in degrees. The initial values for the last entries in the covariance matrix is

$$\boldsymbol{a}_{b_\sigma} = \begin{bmatrix} 10^{-8} & 0 & 0 \\ 0 & 10^{-8} & 0 \\ 0 & 0 & 10^{-8} \end{bmatrix}, \quad \boldsymbol{\omega}_{b_\sigma} = \begin{bmatrix} 10^{-8} & 0 & 0 \\ 0 & 10^{-8} & 0 \\ 0 & 0 & 10^{-8} \end{bmatrix} \tag{7.3}$$

The bias values for the acceleration and angular velocity measurements are chosen to be equal along with the diagonal of the matrix. The biases are assumed to be minute and initialized as such.

7.2.3 Initial Filter Tuning Based on Recorded Data

The initial tuning of the ESKF is performed on recorded data by using the ROS framework and the rosbag feature. The rosbag feature allows for real-time playback of the recorded information flow on the vessel. From the observer point of view, this is equivalent to performing live estimation in a real-life scenario. This powerful feature allows for "online" tuning while not being on the vessel. For more information about this feature and ROS in general, see appendix D.

Position Correction Data

With only position data error correction, accurate estimation of the heading proved difficult in the simulated environment. It is doubtful that the filter will be able to correctly estimate the full state vector accurately with only position error correction with real data. As stated in Section 6.1.2, the vessel operates with low acceleration, and the heading estimate will suffer significantly due to the lack of observability of the state. Additionally, the

introduction of more dominant stochastic noise in the measurements will negatively affect the observability. The ESKF experimental test with only position data for error correction proves this suspicion.

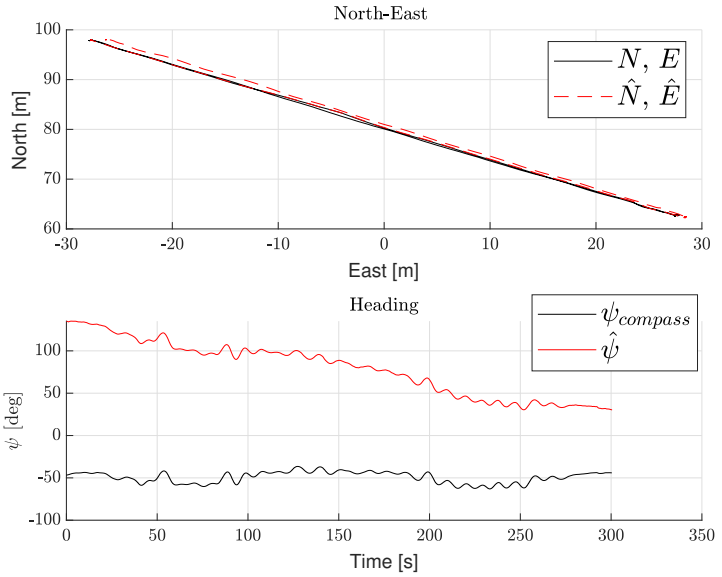


Figure 7.5: Estimated pose $\hat{\boldsymbol{\eta}}$ vs measured pose $\boldsymbol{\eta}$ with position error correction on experimental data.

With only position data for error correction, the heading estimate does not converge, and the ESKF fails to estimate the full six DOF state vector. The unacceptable error in the estimation of the heading propagates further for the estimation of any states that are dependent on the rotation matrix. Case in point is the estimated velocity in body $\hat{\boldsymbol{\nu}}$ shown in Figure 7.6.

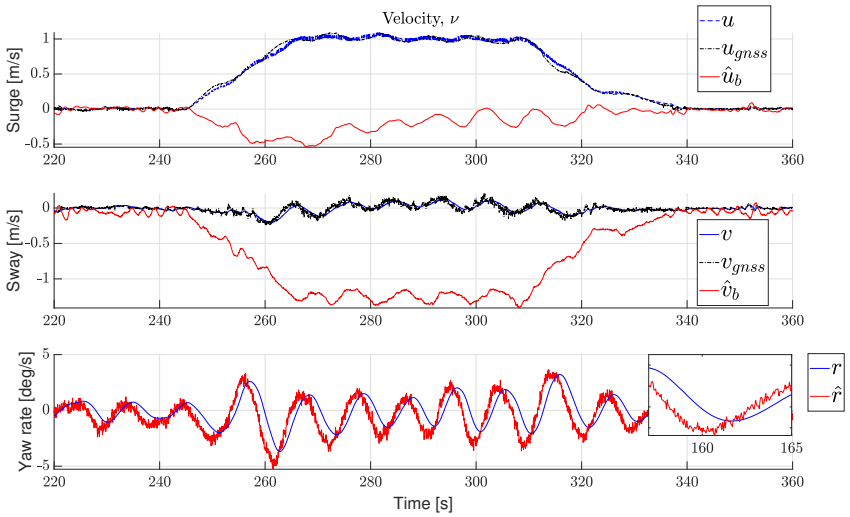


Figure 7.6: Estimated velocity $\hat{\boldsymbol{\nu}}$ vs low pass filtered velocity $\boldsymbol{\nu}$ vs raw velocity measurement $\boldsymbol{\nu}_{gnss}$ with position error correction on experimental data. No subscript is the velocity estimated with low pass filtered differentiation of position, gnss subscript is the measured velocity from the GNSS, and hat notation is the estimated velocity.

Due to the lack of accuracy in the heading estimate, the velocity estimate $\hat{\boldsymbol{\nu}}$ is way off. This is unacceptable and will produce unwanted behavior from the motion control system. This chapter refers to the estimated acceleration as $\hat{\boldsymbol{a}} = \boldsymbol{a}_m - \hat{\boldsymbol{a}}_b$, and the estimated angular velocity as $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b$, where subscript m denotes the measurement and subscript b denotes the bias. The rest of the estimated states in comparison to the measured states are shown in figures 7.7 - 7.9.

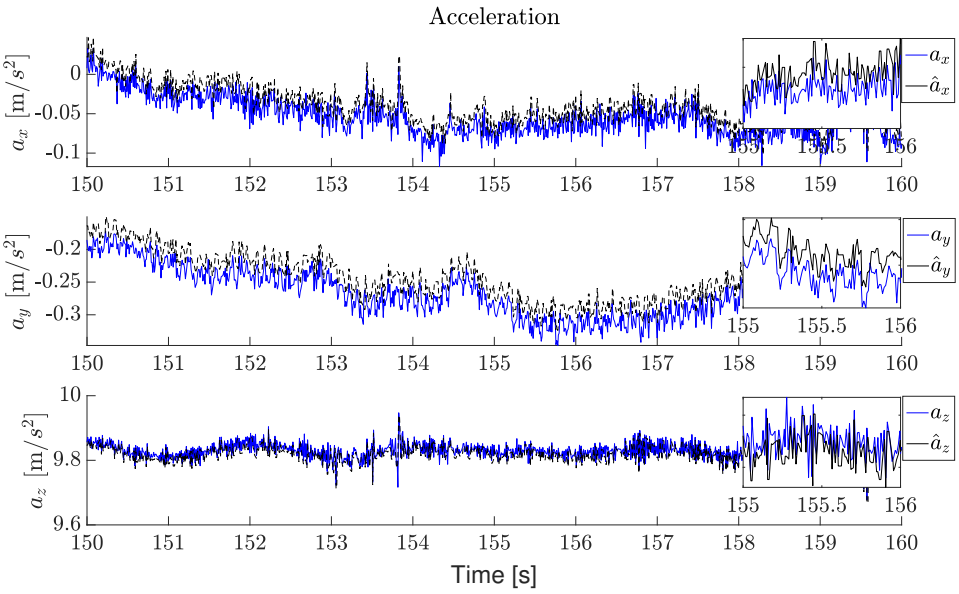


Figure 7.7: Estimated acceleration $\hat{\boldsymbol{a}}$ vs the measured acceleration $\boldsymbol{a}$ with position error correction on experimental data. The boxes on the individual plots are plots of the same states only zoomed in for better inspection of differences between the estimations and the measurements.

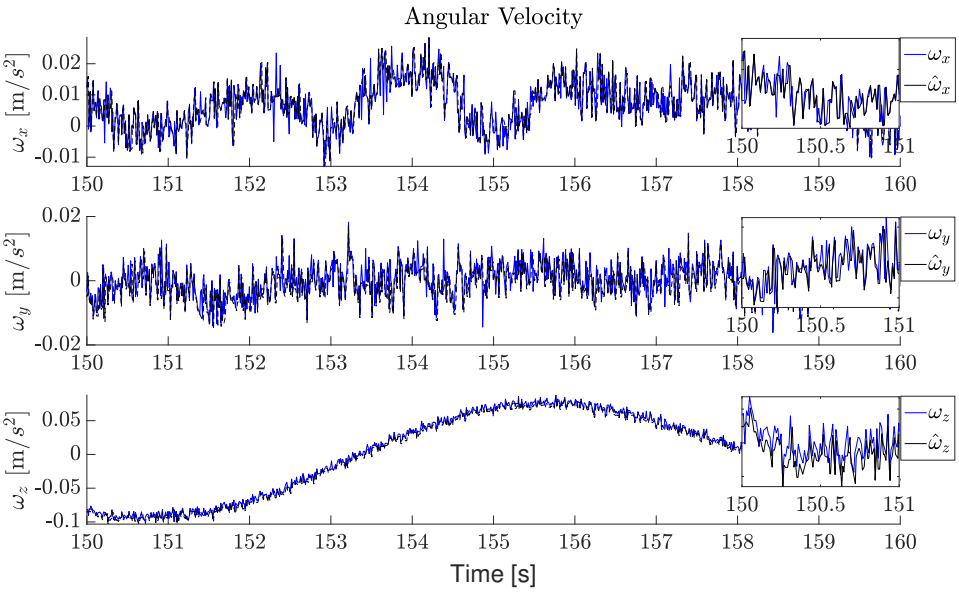


Figure 7.8: Estimated angular velocity $\hat{\boldsymbol{\omega}}$ vs measured angular velocity $\boldsymbol{\omega}$ with position error correction on experimental data. The boxes on the individual plots are plots of the same states only zoomed in for better inspection of differences between the estimations and the measurements.

Section 7.2.1 discusses the mounting of the IMU and especially mentions the slight offset in the roll position. The roll offset manifests itself into the measured acceleration along the y-axis. Figure 7.21 shows that the filter correctly adds a bias to the acceleration measurements to correct offset in the roll.

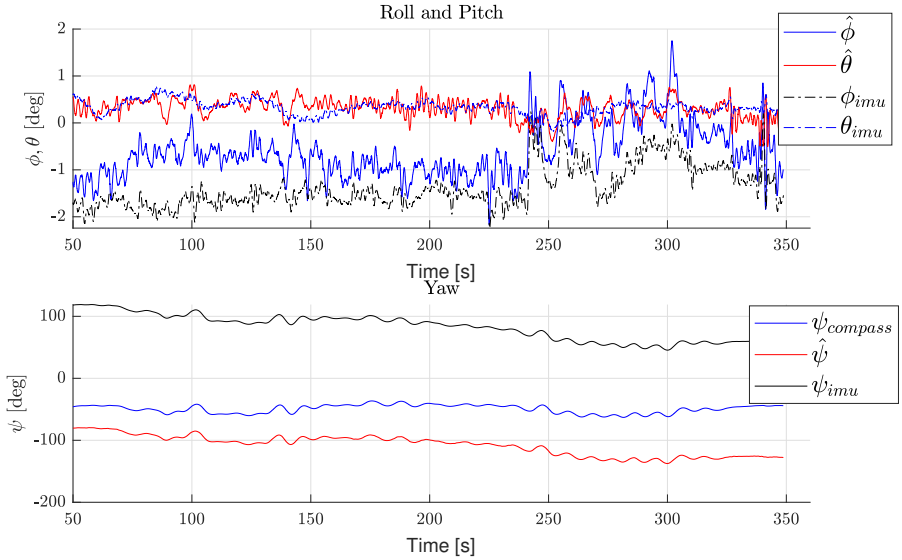


Figure 7.9: Estimated attitude $\hat{\mathbf{\Theta}}$ vs measured attitude $\mathbf{\Theta}$ with position error correction on experimental data. The yaw plot show the compass measurement $\psi_{compass}$, the estimated yaw $\hat{\psi}$ and the IMU yaw measurement ψ_{imu}.

The IMU uses the method discussed in Section 3.1.1 to estimate the roll and pitch. Additionally, the IMU uses a specially designed Xsens Kalman filter (XKF3i) to estimate roll and pitch with better accuracy. The XKF3i averages the measured acceleration over an extended period to estimate the gravity acceleration. As summarized in [Xsens], the estimation of the gravitational acceleration is used to stabilize the estimation of roll and pitch. The limitation for this approach is that the acceleration must be averaged over an extended period to give an accurate estimate of the direction of the gravity vector.

Additionally, the averaged acceleration will never be zero, so the expectation is for the estimated attitude to have some slight deviation from the actual attitude. However, in this case of a slow, low accelerating marine vessel, this is not an issue. Subsequently, the roll angle and pitch angle estimates from the IMU are entirely accurate. The yaw measurement from the IMU is unreferenced, but more accurate than simple integration of the angular velocity. The estimate for the attitude is used to compare to the ESKF attitude estimate.

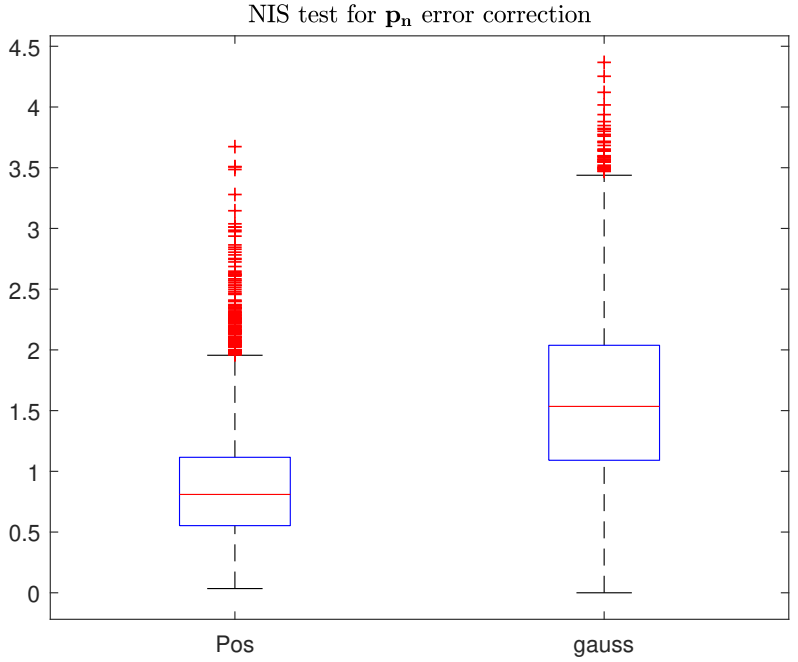


Figure 7.10: NIS test results for error between position estimates and position correction data on experimental data.

The NIS test indicates that the tuning parameters are decent. Still, the filter is not able to correctly estimate the heading accurately. As will be discussed next, this is due to the reduced observability caused by the low acceleration, and the introduction of more dominant stochastic noise in the system.

Observability Issues

In [Bageshwar et al., 2009], a simple test of stochastic observability in discrete Kalman filters is proposed. The test uses the singular value decomposition of the covariance matrix to determine if the system is stochastically observable. This simple test inspects the maximum singular value for the covariance matrix. The definition of the test parameter is

$$\sigma_{max}(\mathbf{P}_k) < T_v, \qquad t_k \geq t_N \tag{7.4}$$

where σ_{max} is the largest singular value of the covariance matrix $\mathbf{P}_k$, and $T_v, t_N < \infty$. The test dictates that after convergence, the maximum singular value should be below a threshold limit defined by the convergence value of the maximum singular value. For the

system to be stochastically observable, once $\sigma_{max}(\mathbf{P}_k)$ has converged to a finite limit for $t_k = t_N$, $\sigma_{max}(\mathbf{P}_k)$ must be less than the threshold value for $t_k \geq t_N$. As seen in Figure 7.11, this is not the case.

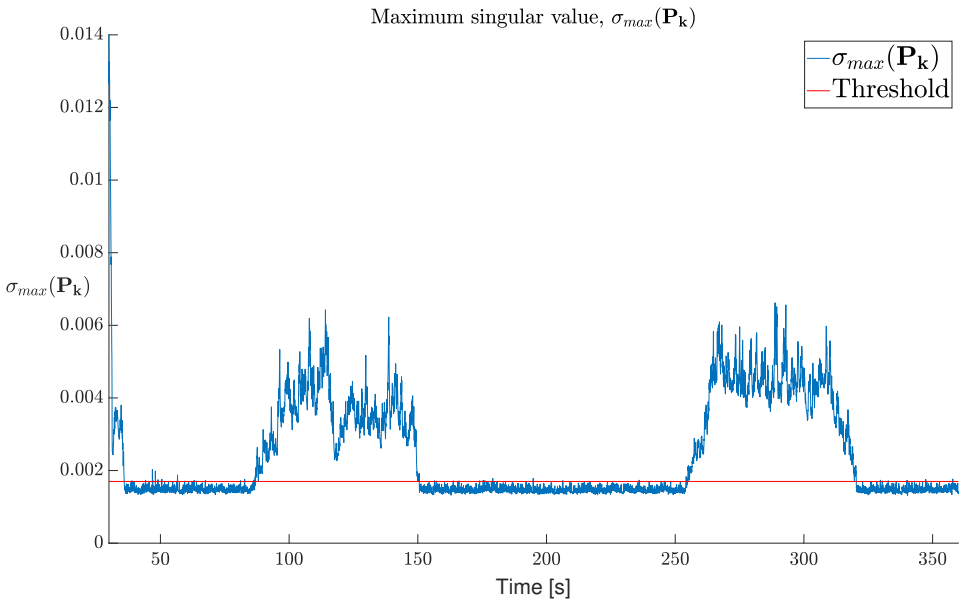


Figure 7.11: Stochastic observability test results with position data error correction on experimental data.

The conclusion is that the system, with such a low bounded maximum acceleration, is not fully observable when only using position data for error correction. Thus, the filter is not expected to converge while performing normal crossing operations. The filter needs more measurements to achieve satisfying estimation.

Position and Compass Correction Data

The introduction of compass measurements fixes the issue with an inaccurate estimate of the heading angle.

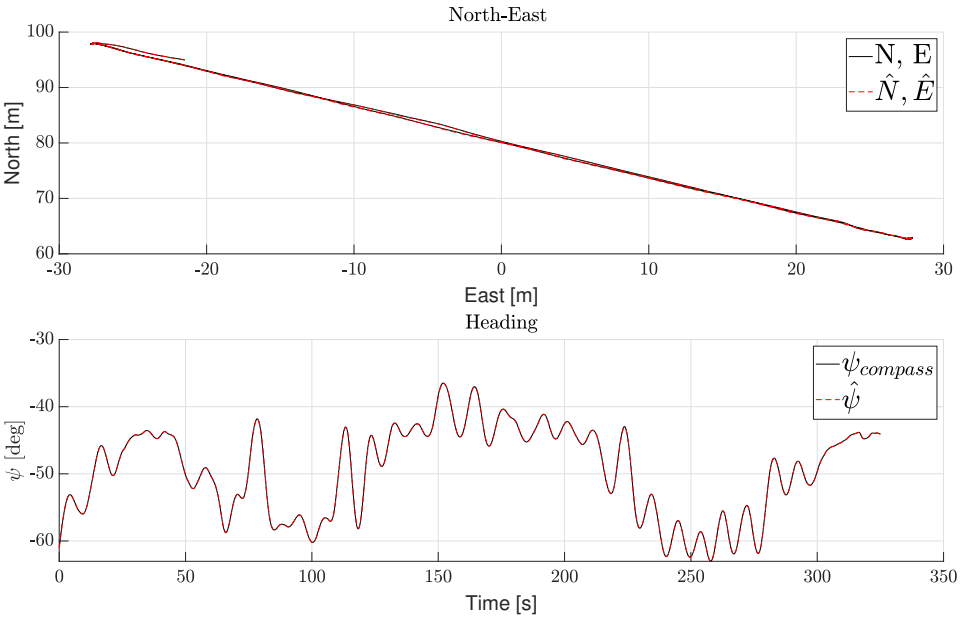


Figure 7.12: Estimated pose $\hat{\boldsymbol{\eta}}$ vs measured pose $\boldsymbol{\eta}$ with position, and compass error correction on experimental data.

The full state vector is estimated correctly with the introduction of compass error correction. However, the measurement noise in acceleration measurements affects the velocity estimates quite substantially. If the acceleration measurements are weighted heavily in the estimation, more noise propagates into the velocity estimate. On the other hand, as seen in Figure 7.13, non-satisfactory velocity estimates are a result of lower weights on the acceleration measurements. Obtaining both accurate and relatively noise-free estimates of the velocity with the current set up was not achieved.

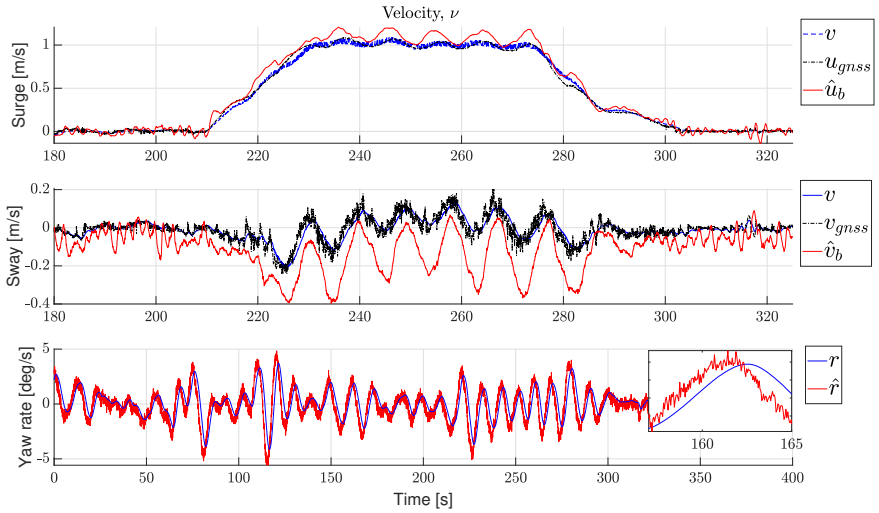


Figure 7.13: Estimated velocity $\hat{\boldsymbol{\nu}}$ vs low pass filtered velocity $\boldsymbol{\nu}$ with position and compass error correction on experimental data

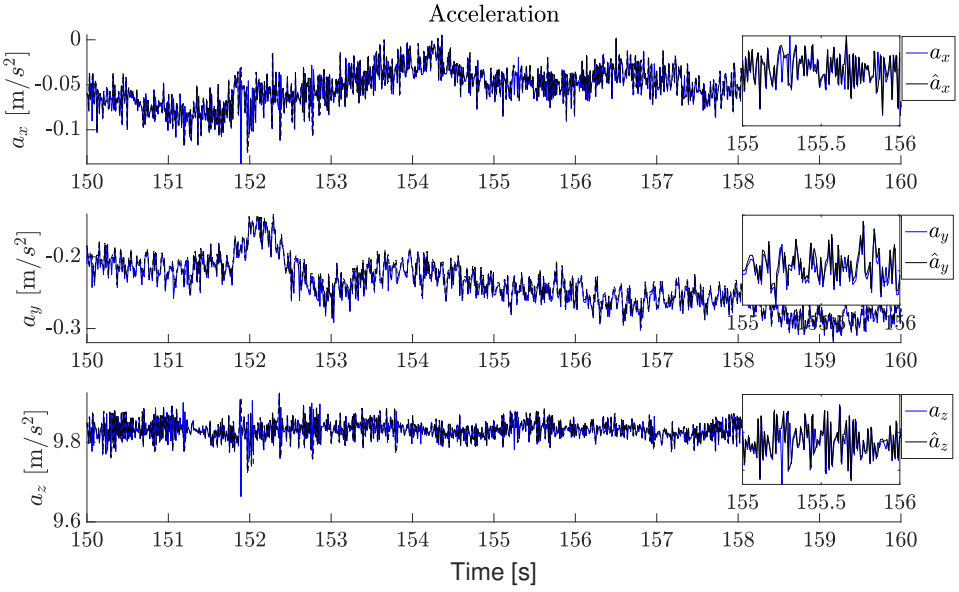


Figure 7.14: Estimated acceleration $\hat{\boldsymbol{a}}$ vs measured acceleration $\boldsymbol{a}$ vs raw velocity measurement $\boldsymbol{\nu}_{gnss}$ with position and compass error correction on experimental data.

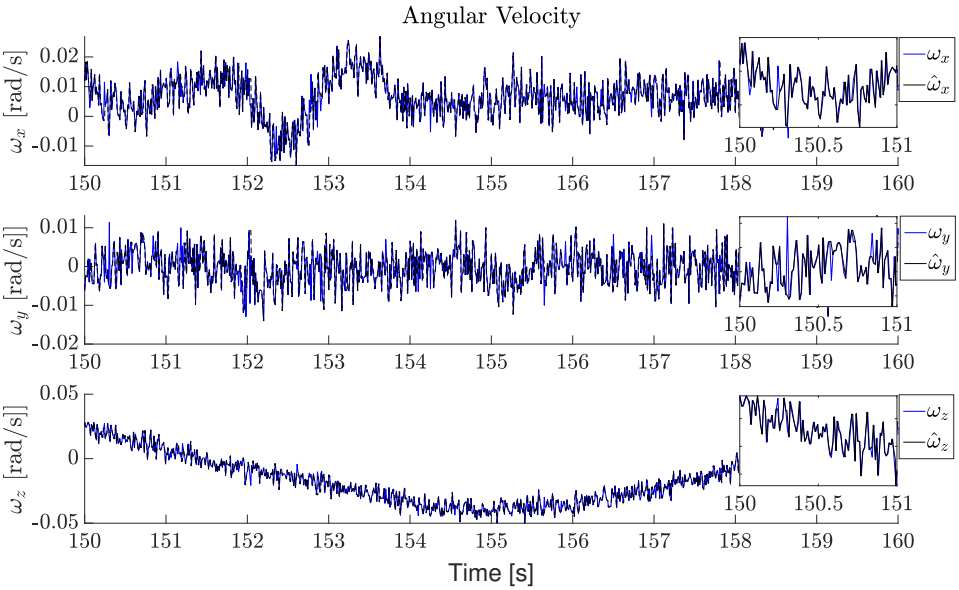


Figure 7.15: Estimated angular velocity $\hat{\boldsymbol{\omega}}$ vs measured angular velocity $\boldsymbol{\omega}$ with position and compass error correction on experimental data.

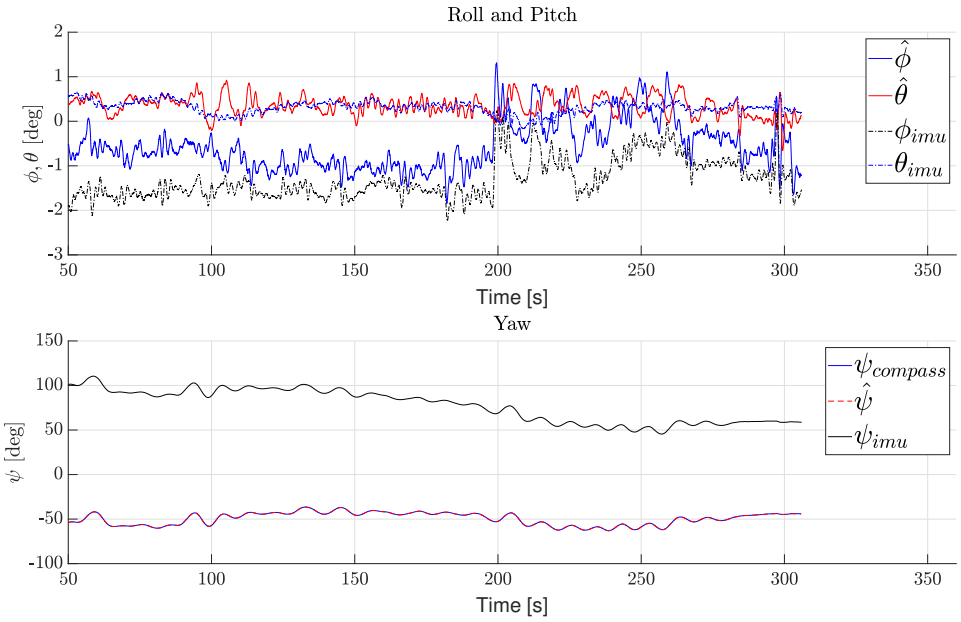


Figure 7.16: Estimated attitude $\hat{\mathbf{\Theta}}$ vs measured attitude $\mathbf{\Theta}$ with position and compass error correction on experimental data.

Following logic reasoning, the attitude estimate makes sense. Any marine vessel in relatively still waters will have roll and pitch angles approximately equal to zero. It is reassuring to have the attitude estimate backed up by another estimation algorithm with a proven record in the XKF3i algorithm. Similar estimates of attitude through the two respective methods give high confidence in the filter estimate.

Compass measurements aid the ESKF for the estimation of the yaw angle. These compass measurements provide the filter with very accurate yaw measurements. In comparison, the IMU estimate of ψ struggles due to the magnetic field induced by other components around the sensor. Figure 7.16 show that the estimated attitude from the filter is accurate.

For the NIS test of compass estimates, the compass measurement is one-dimensional, and subsequently, the Gaussian distribution used in the comparison is univariate.

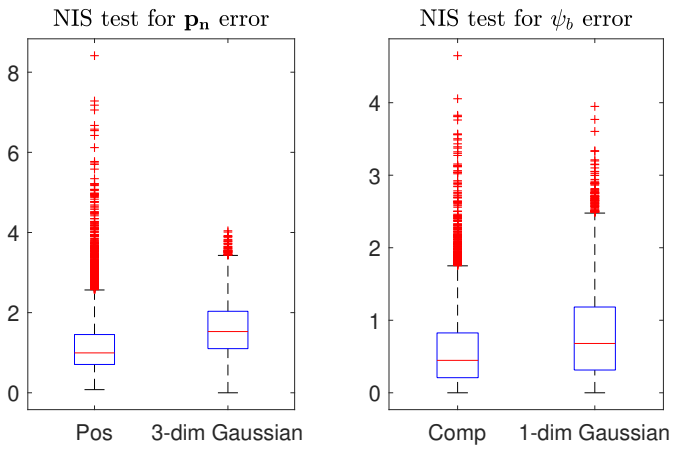


Figure 7.17: NIS test results with position, and compass error correction on experimental data vs the norm of their respective dimensional Gaussian distributions.

The consistency tests give further confidence in the estimated state vector. The filter shows sufficient estimation consistency and thus, reliant estimation.

Position, Velocity and Compass correction data

With the introduction of the velocity measurements, the estimated position and attitude does not change much. As expected, the improvement mainly comes in the estimated velocity. The estimated pose $\hat{\boldsymbol{\eta}}$ is the same as before, and the ESKF shows an accurate estimate of the position.

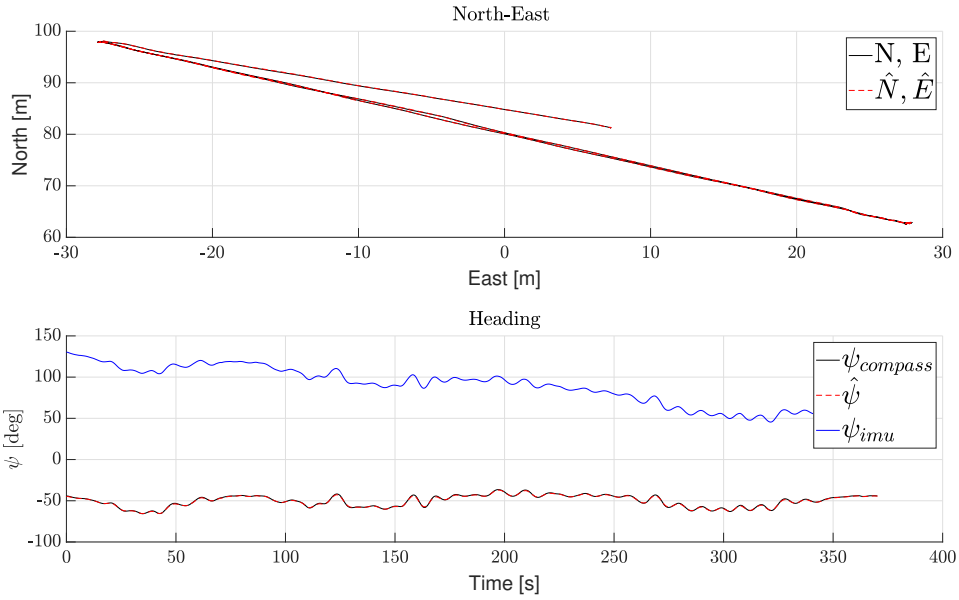


Figure 7.18: Estimated pose $\hat{\boldsymbol{\eta}}$ vs measured pose $\boldsymbol{\eta}$ with position, velocity, and compass error correction on experimental data.

The introduction of the velocity measurement corrects the error that occurs in the velocity estimate with only position and compass error correction. Figures 7.13 and 7.19 show the improvement as a result of velocity data error correction.

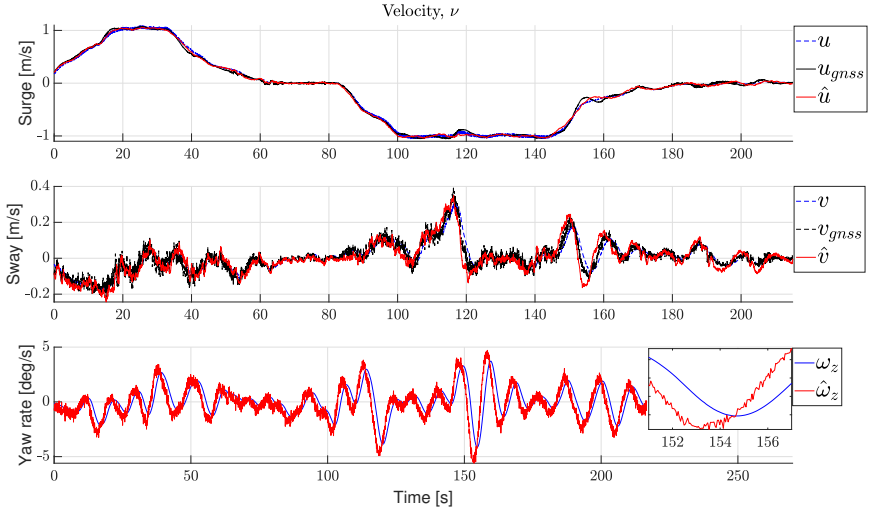


Figure 7.19: Estimated velocity $\hat{\boldsymbol{\nu}}$ vs low pass filtered $\boldsymbol{\nu}$ vs raw velocity measurement $\boldsymbol{\nu}_{gnss}$ with position, velocity, and compass error correction on experimental data. The small box plotted on the right side of the yaw rate measurements show the phase lag of 1.5 seconds caused by the low pass filtering.

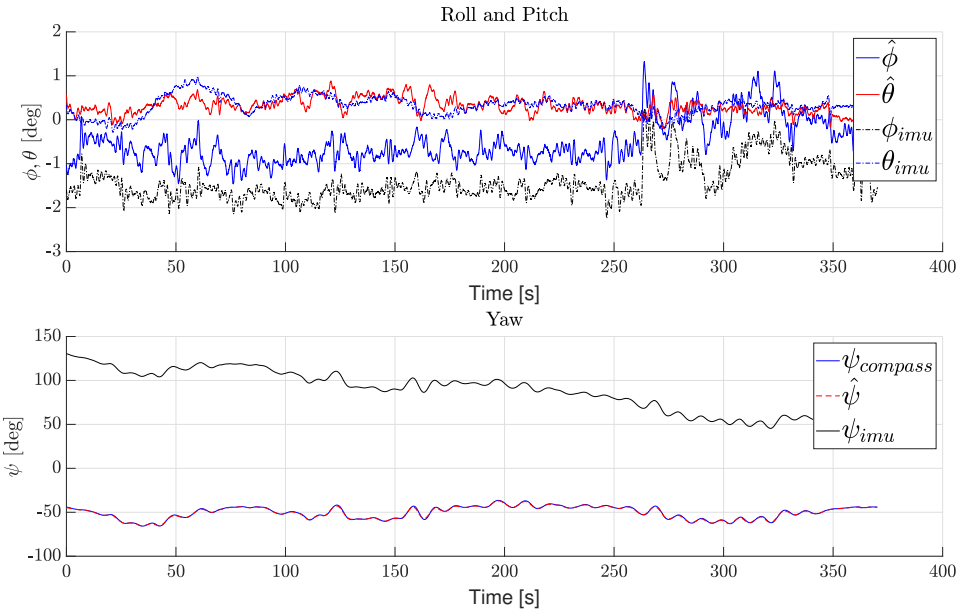


Figure 7.20: Estimated attitude $\hat{\boldsymbol{\Theta}}$ vs measured attitude $\boldsymbol{\Theta}$ with position, velocity, and compass error correction on experimental data.

With the addition of the velocity error correction in the ESKF, the attitude estimation further improves slightly. The mounting of the IMU means that the inclination estimate of the IMU has a small offset with a negative roll. As a result, the two individual estimates of the attitude are not expected to coincide directly. See Section 3.1.1 for more details on the IMU estimation of attitude.

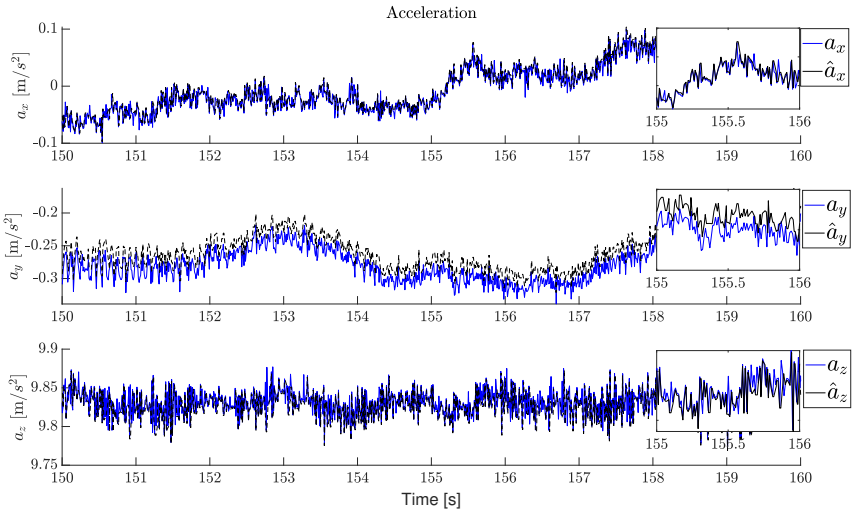


Figure 7.21: Estimated acceleration $\hat{\boldsymbol{a}}$ vs measured acceleration $\boldsymbol{a}$ with position, velocity, and compass error correction

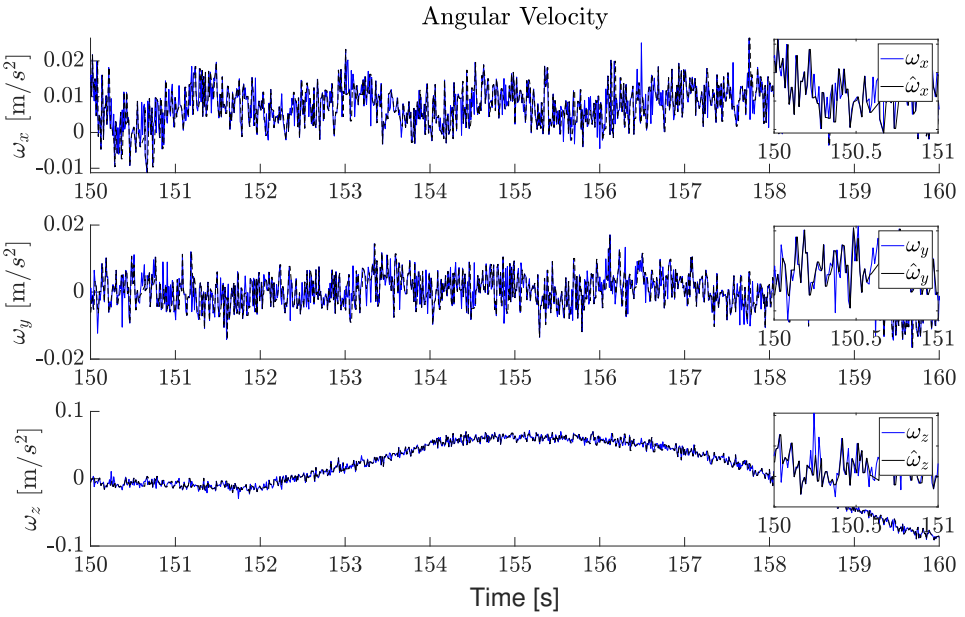


Figure 7.22: Estimated angular velocity $\hat{\boldsymbol{\omega}}$ vs measured angular velocity $\boldsymbol{\omega}$ with position, velocity, and compass error correction.

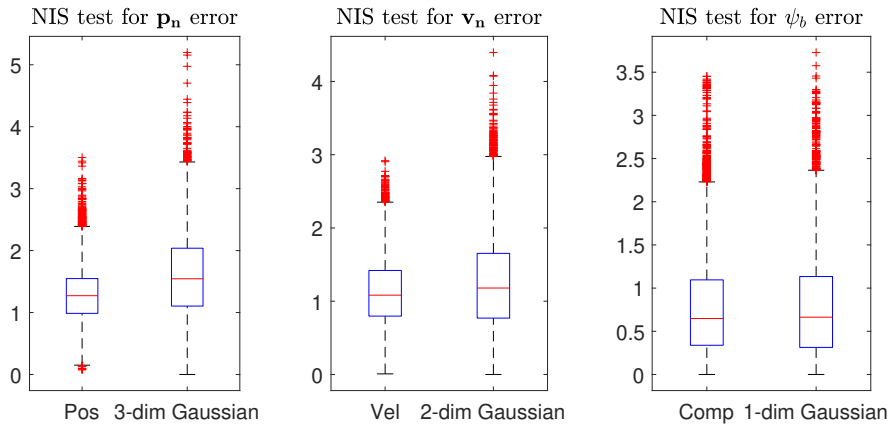


Figure 7.23: NIS test on real data with position, velocity, and compass error correction vs the norm of their respective dimensional Gaussian distributions.

7.2.4 Filter Consistency During Prolonged Run Time

Consistency Test When Moored

When in operation, the filter will run for a prolonged amount of time. It is vital to perform consistency tests over an extended period to verify that the filter is stable. The NIS test is yet again used to verify the filter performance. During prolonged operation, any weaknesses or faults are more likely to propagate and be exposed. Figure 7.24 shows the NIS test results during 1-hour runtime for a moored ferry.

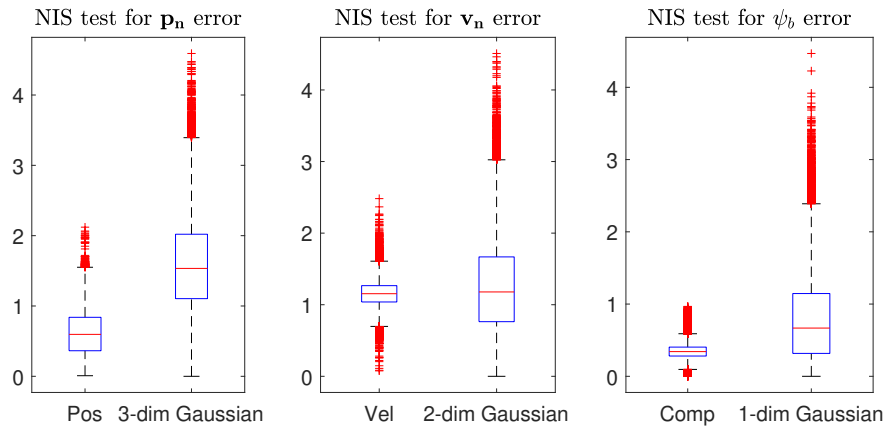


Figure 7.24: NIS test for position during 1 hour while vessel is moored vs norm of three dimensional Gaussian distribution.

The consistency test reveals that the tuning matrices for the position and velocity measurements are well-tuned and that the ESKF is stable over a prolonged period. However, the R_ψ is too large since the filter is overconfident for the compass measurements. At a standstill, the modelling of the velocity measurement noise is not as a white noise distribution. Additionally, the position measurement will have less measurement noise. These noise attributes are reflected through their respective distributions when at a standstill.

Consistency Test Under Normal Operation

The filter performs well under normal operation. Figure 7.25 shows the consistency test results during normal operation.

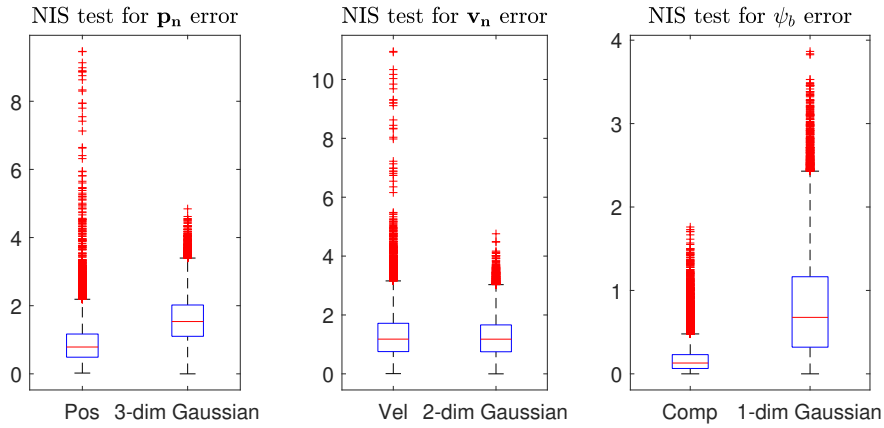


Figure 7.25: NIS test during 30 minutes of normal operation.

What can be observed is that there are a lot more outliers during regular operation than when moored. Vibrations in the hull cause the observed outliers. The thrusters ramping up and down, and people walking on deck, cause the vibrations in the hull. Given that the IMU is very sensitive to external vibrations in its current location, especially the acceleration measurements, these vibrations will propagate in the predictions the ESKF makes, and thus cause the outliers visible for the velocity and position NIS test results. The outliers aside, the NIS results for the position and velocity are excellent, and the tuning parameters should be left as is. For the compass, the ESKF is overconfident, and measurement covariance for the compass measurement R_{ψ} needs adjusting to a smaller value.

7.2.5 Robustness Test

The ESKF needs to be stable under demanding situations. By inducing large spikes on the IMU measurements, the corrupted measurements will give a clear measure of the filter robustness. As discussed previously, this is done by stomping and jumping on deck to cause material deflections and vibrations that manifest themselves in the IMU measurements. Figure 7.2 shows how these actions greatly affect the acceleration measurements. With the corruption of the measurements affecting the filter estimates, NIS consistency test results indicate the filter robustness. The resulting NIS test is shown in Figure 7.26.

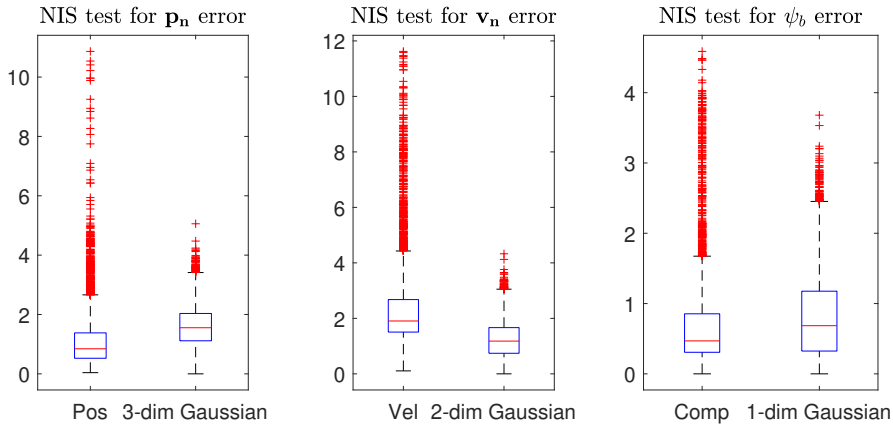


Figure 7.26: NIS test while IMU measurements are compromised.

The more significant number of outliers in the consistency test results is a result of the corrupted measurements. However, the consistency test performed while compromised measurements propagate through the filter reveal that the filter is highly capable of dealing with false measurements.

The induced deviations in the measurements are quite large, and it compromises the filter estimates somewhat. Compromised estimates are inevitable given the IMU driven nature of the filter. However, the filter withstands the compromised measurements and remains stable.

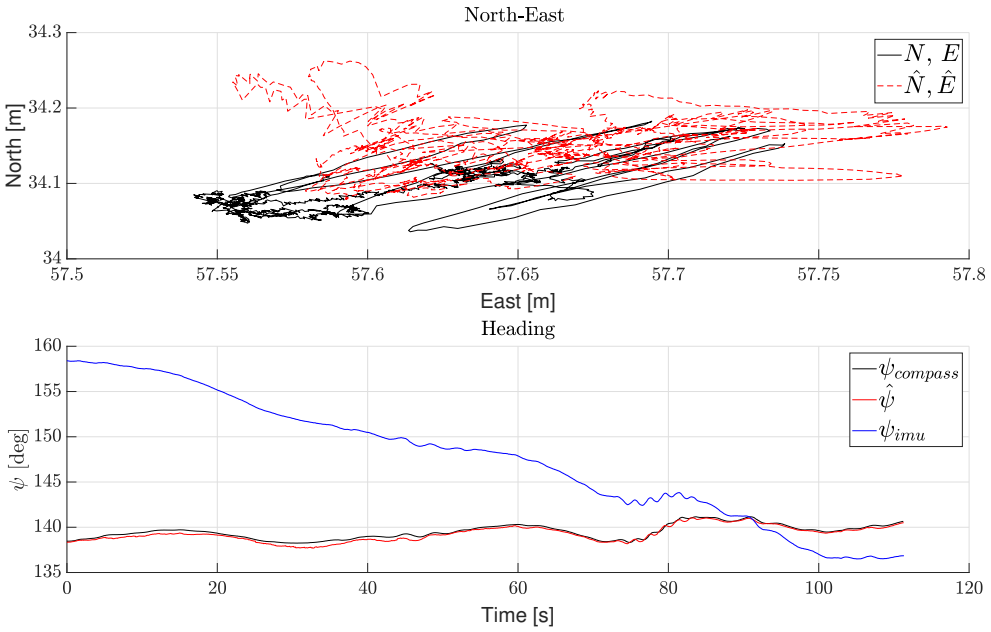


Figure 7.27: Estimated pose $\hat{\boldsymbol{\eta}}$ estimate while IMU measurements are compromised.

The corrupted IMU measurements induce almost no drift in the position estimate. The small position deviation is highly encouraging, and it shows that the filter stays true to the measured state even when the IMU measurements are highly unrealistic for an extended period. The same goes to show for the velocity and the attitude estimate. The filter keeps both the estimated velocity $\hat{\boldsymbol{\nu}}$ and the estimated attitude $\hat{\boldsymbol{\Theta}}$ close to the "true" state shown by the velocity estimate obtained through low pass filtered differentiation and the compass measurement.

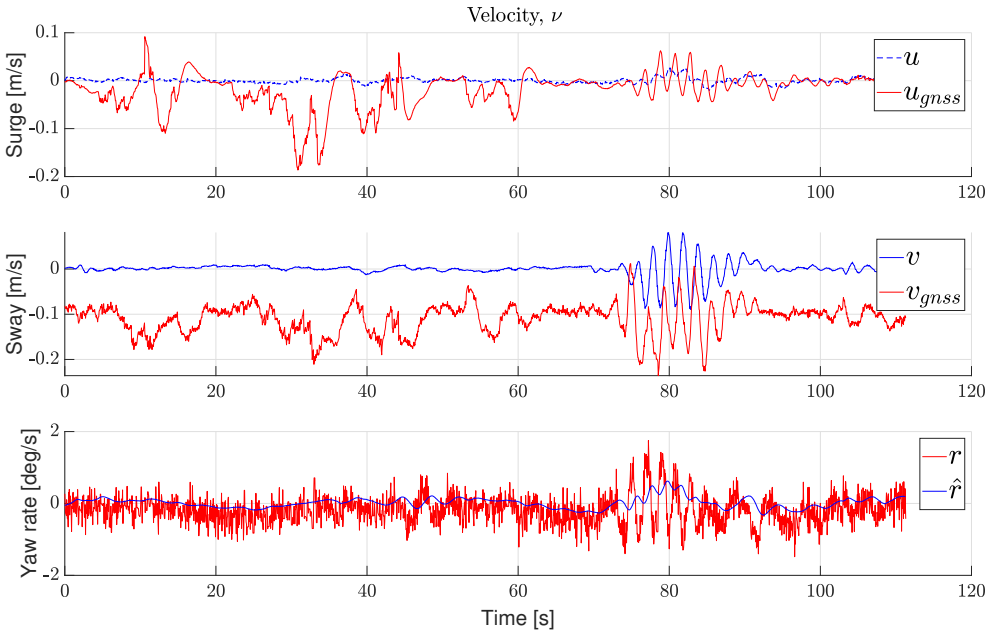


Figure 7.28: $\hat{\nu}$ estimate while IMU measurements are compromised.

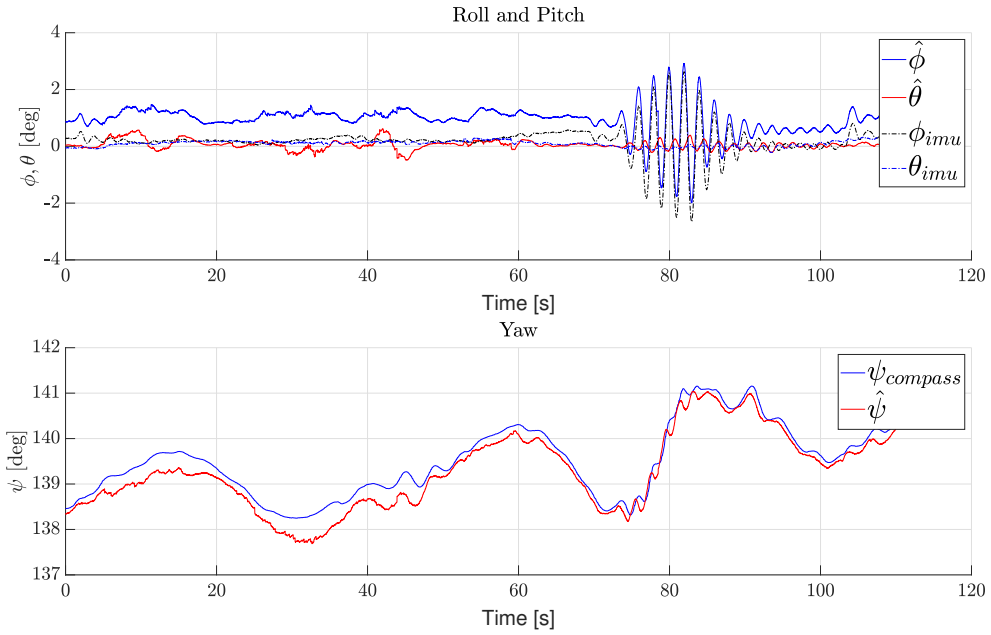


Figure 7.29: Estimated attitude $\hat{\Theta}$ while IMU measurements are compromised.

The low pass filtered differentiated states provided by the old navigation node are a good measure of the true state when the IMU measurements are compromised. As discussed in Section 3.1.1, the XKF3i averages the acceleration measurements over time to establish the inclination. Since the expected acceleration during this test over time is zero, the inclination estimate is expected to be relatively accurate even when the measurements are compromised, thus providing a good comparison for the estimated attitude, and north and east position.

7.2.6 Final Tuning Parameters

R matrix

The position measurement suffers from increased measurement noise for increasing velocity. Figure 7.30 shows the correlation between velocity and the measurement noise in the position measurement.

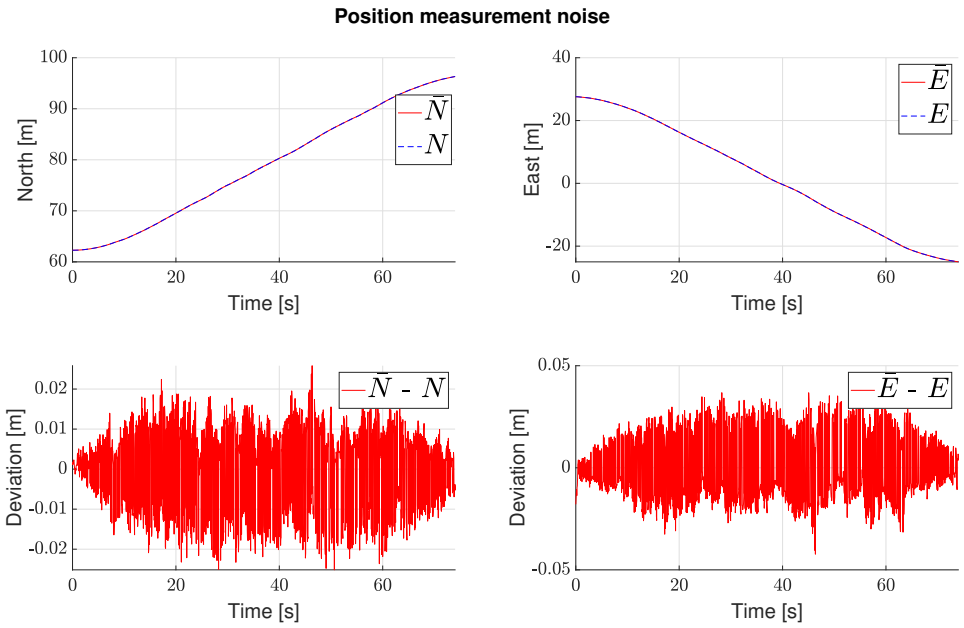


Figure 7.30: Correlation between the vessel velocity and the position measurement noise. N and E are the measured north and east positions, and $\bar{N}$ and $\bar{E}$ are the moving averages of the position measurements. $\bar{N} - N$ and $\bar{E} - E$ show the deviations from the means for the respective measurements.

The correlation between increased velocity and the resulting position measurement noise is apparent. For both the north and east position, the deviation from the mean is more significant when the change in position is more substantial. The tuning of the position measurement covariance matrix $\mathbf{R_p}$ needs to reflect the variable measurement noise of the position measurement. The measurement covariance matrix for the position updates is inspired by a tracking example using Kalman filters in [Brekke, 2018]. The covariance matrix uses the estimated velocity coupled with a scaling factor to increase the measurement covariance to a more suitable value for higher velocities. Additionally, the altitude measurements are less accurate than the lateral position measurements. The position measurement covariance matrix reflects these characteristics with the following parameters

$$\mathbf{R_p} = \begin{bmatrix} 0.015 & 0 & 0 \\ 0 & 0.015 & 0 \\ 0 & 0 & 0.025 \end{bmatrix} + \frac{1}{48} \begin{bmatrix} \hat{v}_x^2 & 0 & 0 \\ 0 & \hat{v}_y^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.5}$$

The measurement covariance matrices $\mathbf{R}$ for the other measurements are assumed constant. Through iterative tuning, the parameters are set to

$$\mathbf{R}_{\mathbf{v}} = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}, \quad \mathbf{R}_{\psi} = 0.0005 \tag{7.6}$$

Q Matrix

Continuous-time models are useful from a conceptual perspective because they are invariant to the discretization time interval [Brekke, 2018]. A continuous-time representation of the process noise matrix is

$$\mathbf{Q} = \begin{bmatrix} \sigma_a^2 \mathbf{I}_{3x3} & 0 & 0 & 0 \\ 0 & \sigma_\omega^2 \mathbf{I}_{3x3} & 0 & 0 \\ 0 & 0 & \sigma_v^2 \mathbf{I}_{3x3} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \mathbf{I}_{3x3} \end{bmatrix} \tag{7.7}$$

where σ^2 represents the process noise for the given state.

The diagonal representation of the process noise is not realistic for any discrete-time implementation, and typically the process noise matrix $\mathbf{Q}$ ought to be non-diagonal for a discrete implementation given the correlations that build up over time. The assumption for the tuning of the process noise matrix is that the high frequency of the IMU measurements renders the correlations between measurements insignificant. The high-frequency updates mean that the diagonal discrete-time process noise matrix successfully approximates the continuous-time process noise matrix with an acceptable error.

The high sampling rate causes all the non-diagonal terms in the discrete-time process noise matrix to be approximately 0 as no correlations are evident. The discrete-time approximation of the process noise matrix $\mathbf{Q}$ is then given by

$$\mathbf{Q} = \begin{bmatrix} 0.008^2 \Delta t^2 \mathbf{I}_{2x2} & 0 & 0 & 0 & 0 \\ 0 & 0.03^2 \Delta t^2 & 0 & 0 & 0 \\ 0 & 0 & 0.02^2 \Delta t^2 \mathbf{I}_{3x3} & 0 & 0 \\ 0 & 0 & 0 & (2.5 \cdot 10^{-4})^2 \Delta t \mathbf{I}_{3x3} & 0 \\ 0 & 0 & 0 & 0 & (3 \cdot 10^{-7})^2 \Delta t \mathbf{I}_{3x3} \end{bmatrix} \tag{7.8}$$

The more significant value for the process noise parameter of the acceleration along the z-axis $\sigma_{a_z}^2$ is due to the lack of velocity measurement in heave. As a consequence, more information is needed from the acceleration measurement to obtain a correct estimate of the heave velocity.

7.2.7 Discussion

The ESKF velocity estimate $\hat{\boldsymbol{\nu}}$ reveals some critical aspects of the previous navigation node on the vessel. The old navigation node uses low pass filtered numerical differentiation to obtain the velocity estimate in the NED frame for $\dot{\boldsymbol{\eta}}$. The old node then uses the rotation

matrix to obtain velocity in the BODY frame $\boldsymbol{\nu}$. Figure 7.19 reveals three flaws with this approach. Firstly, the position data provided by the GNSS have substantial measurement noise when the vessel is operating at higher speeds. Any numerical derivation approach will amplify any measurement noise in the signal. The low pass filtering is supposed to minimize the noise, but the consequence of the filtering is a phase lag.

The consequence of the low pass filtering brings up the second flaw. The low pass filtering introduces an unwanted phase lag on the estimated velocity $\boldsymbol{\nu}$. To be able to minimize the effect the measurement noise has on the estimate of the surge velocity, the low pass filter time constant needs to be substantial. Increasing the time constant in the low pass filter will further increase the phase lag. This behavior is highly discouraged, and efforts to eliminate it is necessary.

The third problem with the old navigation system is the reduced amplitude caused by the low pass filtering. The low pass filter will weigh previous measurements to minimize noise. The extreme values for a given state will reduce by weighing previous measurements. This reduction of extreme values is apparent in both the sway and yaw plots shown in Figure 7.19. The reduction will affect how the PID controller allocates a control input. The relation between the velocity and the allocated control input is detailed in (4.6a). If the estimated state $\hat{\boldsymbol{\nu}}$ has a reduced value, the D term in the PID controller will not control the rate change of the control input as efficiently. This loss of accurate representation of the state will cause less accurate tracking.

The introduction of the filter successfully removes the phase lag in all estimates. The removal of the phase lag will affect the accuracy of the motion control system as the controller can allocate a control action based on estimated data with almost no delay. As shown in the zoomed-in plot for the yaw rate estimate in Figure 7.19, the yaw rate has a phase lag of 1.5 seconds. The phase lag is substantial and will affect the ability to control the heading accurately. For milliAmpere, this phase lag will have even more significant effects on the heading accuracy due to the small yaw moment needed to induce a rotation about its z-axis.

7.3 Motion Control Results

The experimental tests of the controllers shown were all conducted in as small a timespan as possible. The small timespan ensured that the conditions for each test were as similar as possible. However, there is no guarantee that the conditions for individual controllers are equal to the others. The external disturbances during the experimental tests were a combination of time-varying wind forces, wave forces, and ocean currents. The most significant factor for the unmodeled dynamics was the introduction of the thruster dynamics. All these factors make up a much more sophisticated testing environment than the simulated environment.

The pure backstepping controller, the $\mathcal{L}_1$-AB controller, proved to be very aggressive in the allocation of the desired control input. The thruster dynamics on milliAmpere is relatively slow, and the controller is deemed unsuitable for use on this vessel. As a consequence, the experimental results do not involve the $\mathcal{L}_1$-AB controller. Even though the limitations

on the current vessel restricts the usage of the $\mathcal{L}_1$-AB controller, it shows that a future platform with faster thruster dynamics can utilize the controller for a performance boost. The non-linear approach allows for more aggressive maneuvering than that of the linear PID controller and serves as a viable candidate for the next iteration of the ferry.

7.3.1 PD $\mathcal{L}_1$-FF vs PID-FF vs PID

The experimental tests unveiled some interesting problems with the initial tuning of the new control algorithm. The biggest issue is the relatively slow thruster dynamics coupled with the 1.5 seconds delay in the navigation node. These dynamics greatly limits the rate change of the control input vector. Reducing the tuning parameter compared to that in the simulated environment handles the introduction of these delays. After iterative tuning, the adaptation gain for the PD $\mathcal{L}_1$-FF controller is set to

$$\gamma_{\omega_\delta} = 3000 \tag{7.9}$$

The PID tuning parameters are left unchanged. The forced reduction of the γ_{ω_δ} will affect the allocation of the control input. The reduced ability for rate change in the thrust force vector, coupled with time-varying external disturbances, will reduce the overall accuracy of the controller.

7.3.2 4-Corner Test

Figure 7.31 shows the experimental 4-corner test for the new control algorithms compared to the old 3 DOF PID controller.

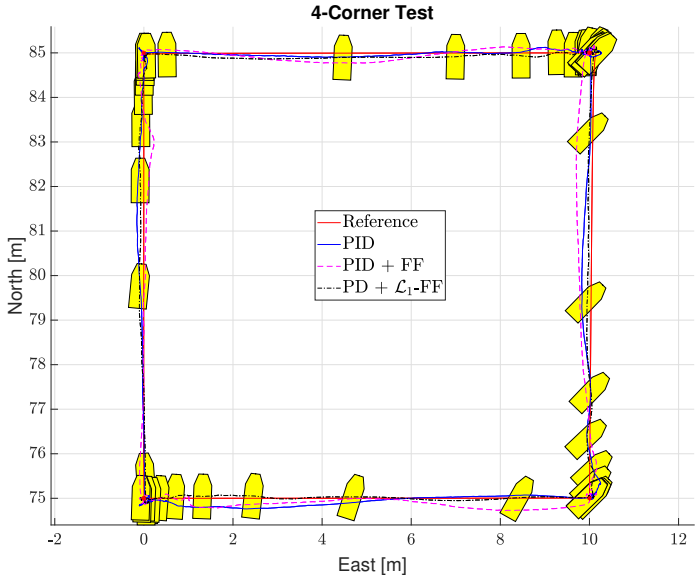


Figure 7.31: 4-Corner test with the new controllers compared to the old PID controller.

The significant reduction in the adaptation gain in the new PD controller with adaptive feedforward negatively affects its overall performance. Still, the new controller vastly outperforms the traditional PID controller. The improvement in performance is best shown in stage $5 \rightarrow 1$, as explained in Section 5.2.2 where the vessel performs a fully coupled motion. This stage is notorious for causing large deviations from the intended path, so the result showing increased accuracy is very encouraging. The performance metrics in Figure 6.12 further shows the improvement in performance. The improved accuracy does come at a cost. It is evident by the performance metric IAEW shown at the bottom in Figure 7.32, that better accuracy comes at the cost of increased actuator usage.

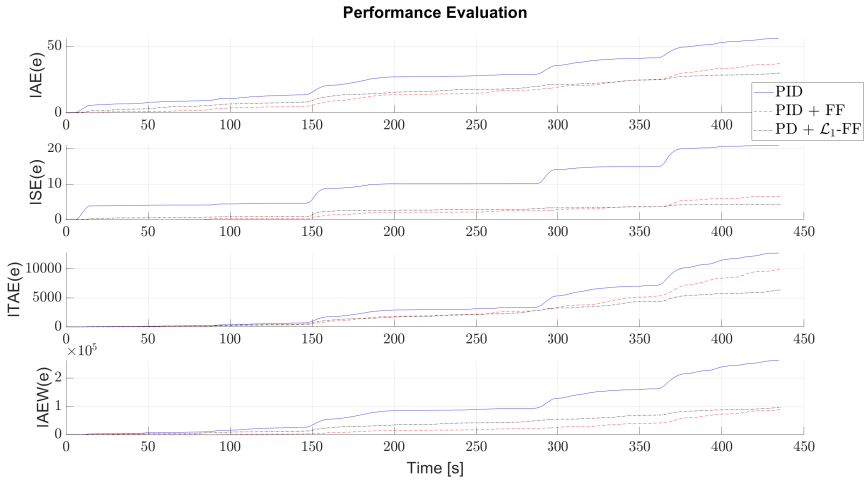


Figure 7.32: Performance metrics for the new controllers compared to the old PID controller.

However, this increase in actuator usage is minuscule, and the improved accuracy outweighs the cost of the small increase in actuator usage. It is crucial to keep in mind that the vessel is operating in a confined area with high traffic. As a result, higher accuracy is desired to ensure that the vessel adheres to the commands given as accurately as possible.

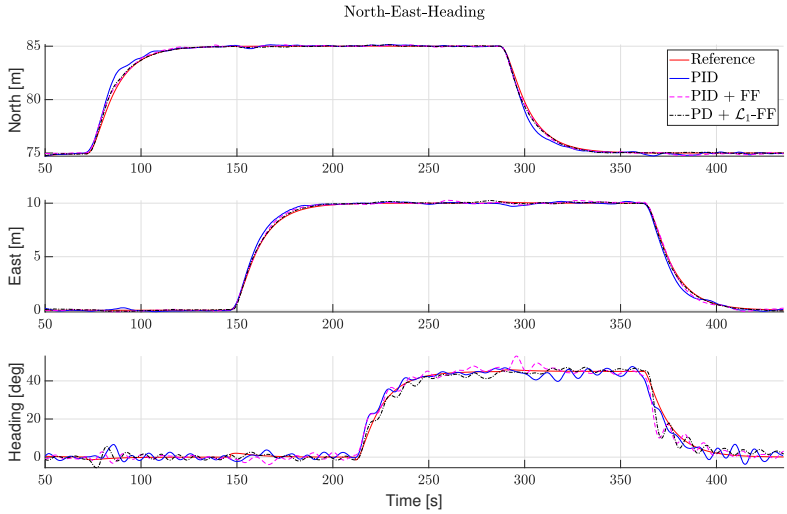


Figure 7.33: North-East-Heading for the new controllers compared to the old PID controller.

The North-East position is spot on during the 4-corner test with the new PD controller with adaptive feedforward. The vessel does, however, have trouble maintaining the desired heading. The vessel is especially susceptible to rotation about its z-axis. The susceptibility to wind forces is troublesome since the thruster dynamics are not fast enough to keep up with the rapid rate change needed to keep it stable during operation. The introduction of the ESKF will improve heading stability by removing the phase lag, but looking into other means for improvement is essential.

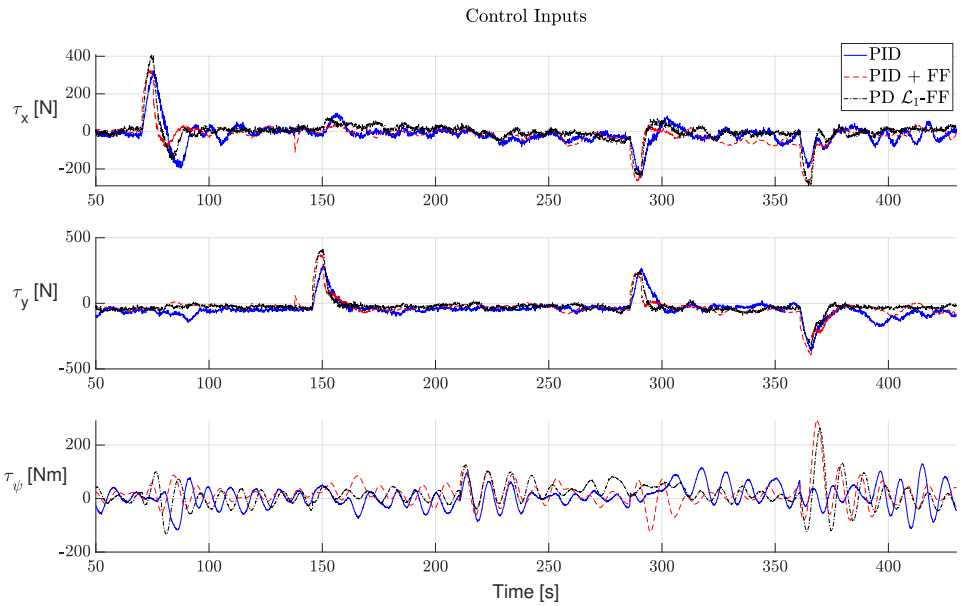


Figure 7.34: Control input for the new controllers compared to the old PID controller.

Figure 7.34 shows the desired control input from the individual controllers. In comparison to the IAEW metric, the actuator usage is almost identical for the different controllers, especially for the desired force in surge and sway direction. The desired control input for the yaw moment varies more. The PD $\mathcal{L}_1$-FF Controller, in general, has a steadier control input to converge to the desired pose. The decreased convergence time of unmodeled dynamics supported by the adaptive backstepping framework ensures the increase in performance.

7.3.3 Discussion

The introduction of adaptive control has the desired effect on the motion control system. The use of adaptive control and the backstepping framework increases the overall accuracy. The most significant improvement is due to the adaptive controller being able to estimate the unmodeled dynamics faster. The reduced convergence time, coupled with the feedforward, increases overall performance significantly. The introduction of adaptive backstepping control also somewhat improves the overall heading accuracy.

The new backstepping approach does not integrate over time to estimate the unmodeled dynamics. Subsequently, the approach mitigates the need for handling of any effects like integrator wind-up. The unconstrained behavior of the integrator term in the PID controller can lead to serious side effects if not handled properly. The integrator does also have a convergence time much slower than for the adaptive backstepping approach. The reduced

time needed for estimation of the unmodeled dynamics will ensure that the vessel is more robust against time-varying disturbances and other external factors.

One remark is that the controllers with feedforward have excessive actuator usage for yaw moment during the final stage of the 4-corner test. This excessive use of moment is evident by comparing the feedforward-based controllers vs the pure PID controller. By using figures 7.34 and 7.33, it is seen that the model is too aggressive in predicting the moment needed in yaw. The incorrect use of thrust force suggests that the coupled vessel model is somewhat inaccurate.

7.4 Combined Efforts of The New Navigation and Motion Control Systems

The testing of the ESKF revealed various unwanted characteristics in the old navigation node on the vessel. Most noticeable is the phase lag on the state estimate $\hat{\boldsymbol{\nu}}$ produced by the low pass filtering used in the old navigation node. Section 7.2.7 discusses this in more detail. The following section examines the performance with the combined efforts of the new controller and the new navigation node.

7.4.1 4-Corner Test

The conditions during the individual tests were adequate, with small amounts of wind. The conditions were stable and ensured testing under relatively similar circumstances.

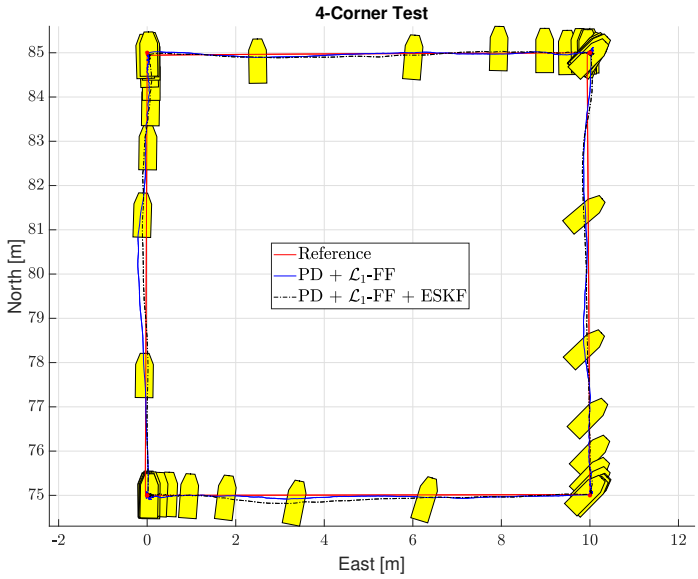


Figure 7.35: 4-corner test for adaptive control with and without the use of the ESKF.

Overall planar position accuracy is slightly worse with the use of the ESKF. The loss of planar position accuracy is apparent in the last stage of the 4-corner test. However, the difference in planar position accuracy between the controllers is insignificant.

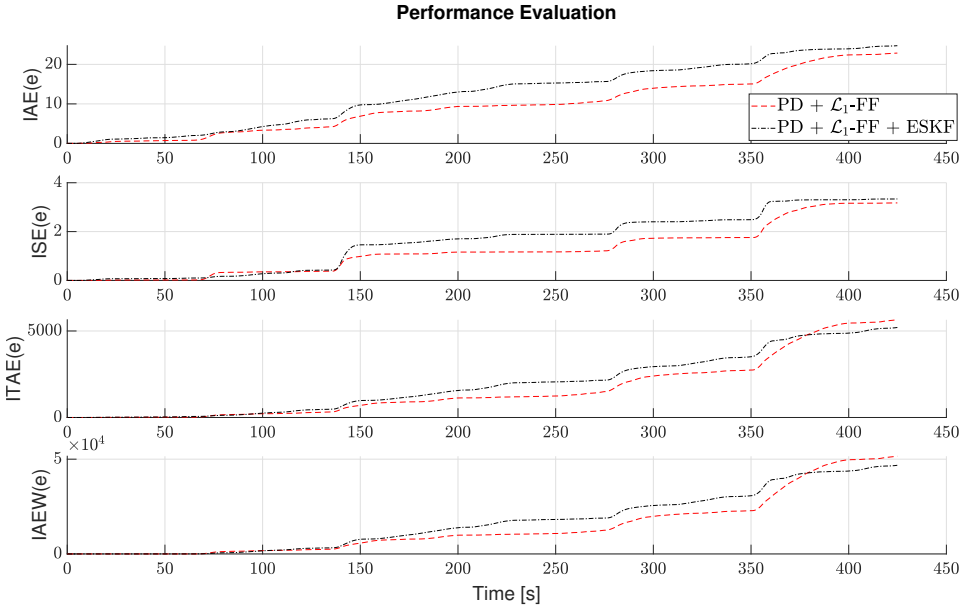


Figure 7.36: Performance metrics for adaptive control with and without the use of the ESKF.

The performance metrics show the loss of planar positional accuracy mentioned above. The minimal loss of positional accuracy is somewhat compensated for by less power consumption.

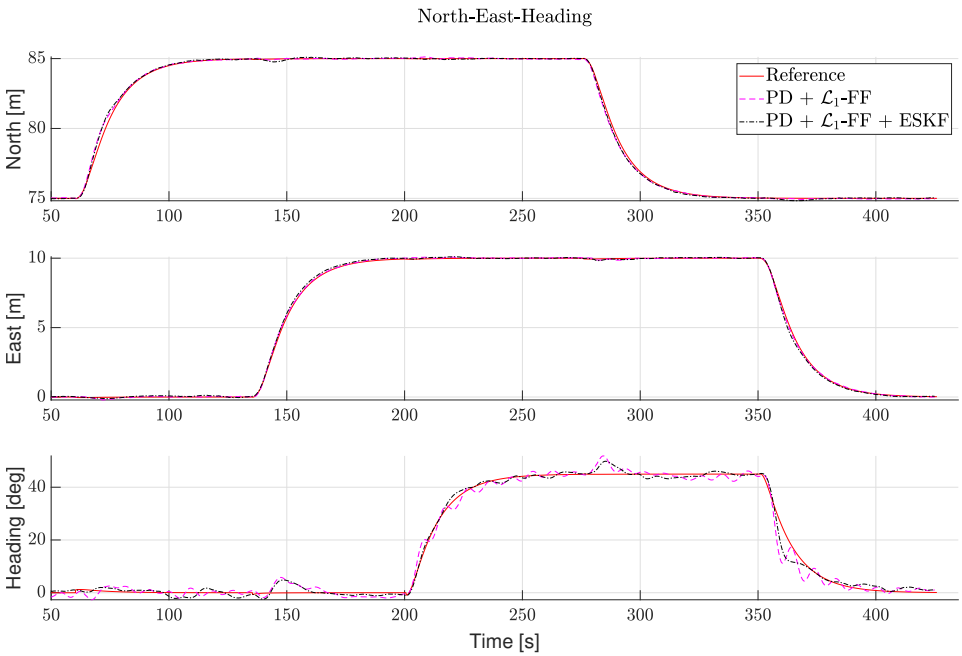


Figure 7.37: Individual responses for adaptive control with and without the use of the ESKF.

The plots of the individual responses show that the planar position is almost identical. The most notable difference is the vessel's ability to track the desired heading reference. The introduction of the ESKF drastically improves the tracking ability for the heading.

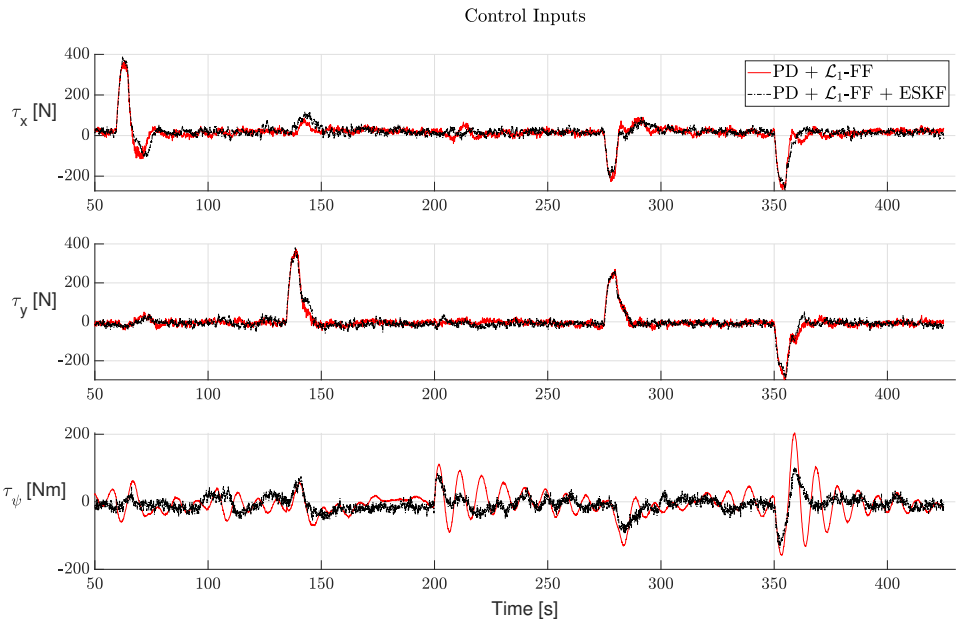


Figure 7.38: Control input for adaptive control with and without the use of the ESKF.

The visibly smoother control input for the yaw moment has removed almost all oscillations in the desired force. The smoother control input will reduce power consumption and ensure smoother heading trajectory tracking. The control input for the yaw moment does, however, show that more measurement noise has propagated through to the actuators. The desired yaw moment with the use of the ESKF is visibly noisier.

7.4.2 Crossing Test With Old vs New GNC system

During regular operation, the ferry will have a forward and backward direction dictated by the non-dynamic positive surge direction. For this definition, the thrust allocation has set the home angle for the thrusters to $0°$, which affects how the ferry responds during operation. The thrust allocation punishes servo movement substantially as the servo dynamics are slow. The reduced ability for fast servo angle changes causes the thrusters to rotate in reverse when the ferry is moving backwards.

When the propellers are operating in reverse, they generate less thrust because of cavitation. The cavitation causes the thrusters to need a more extensive range of rotations per minute (RPM) to generate the desired control input. Consequently, the negative rotation direction of the thrusters leads to slower thruster dynamics than positive rotation. Slower thruster dynamics will, in turn, cause more significant deviations from the desired pose. The heading stability especially suffers due to the need for quick adjustments to keep the vessel stable.

The crossing tests with the old vs the new GNC systems focus on the vessel operating in the defined positive surge direction. Figure 7.39 and 7.40 show the individual responses for the four possible variations consisting of the old and new motion controllers and navigation systems. A small remark is that the initial position plays a role in the generation of the reference trajectory for a crossing. Initializing a crossing at the same position every time is not possible. These initialization offsets cause the trajectories to differ slightly in comparison to each other.

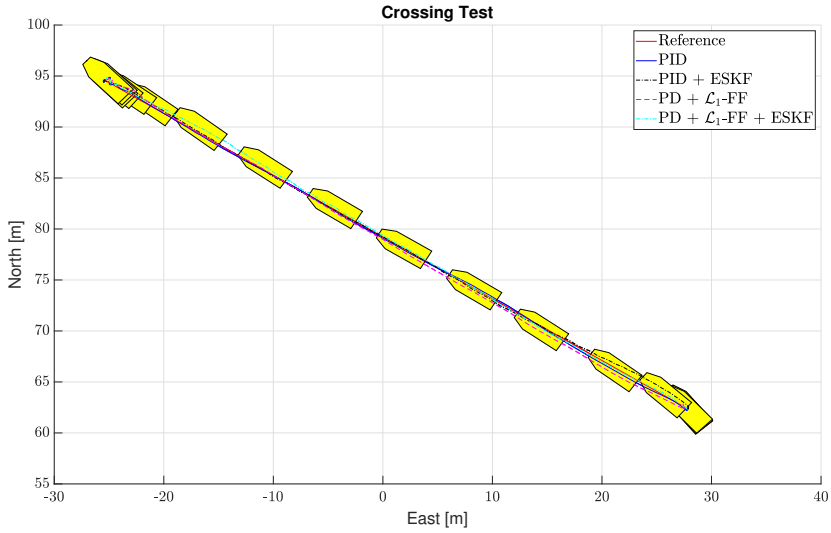


Figure 7.39: Pose $\boldsymbol{\eta}/\hat{\boldsymbol{\eta}}$ during normal crossing operation. $\hat{\boldsymbol{\eta}}$ is the pose from the estimator, while $\boldsymbol{\eta}$ is pose from raw measurements. Crossing tests that use the ESKF utilize the pose $\hat{\boldsymbol{\eta}}$ in the motion control algorithm.

The position accuracies for all variations are stellar. However, the heading stability suffers from oscillations with the use of the old navigation node.

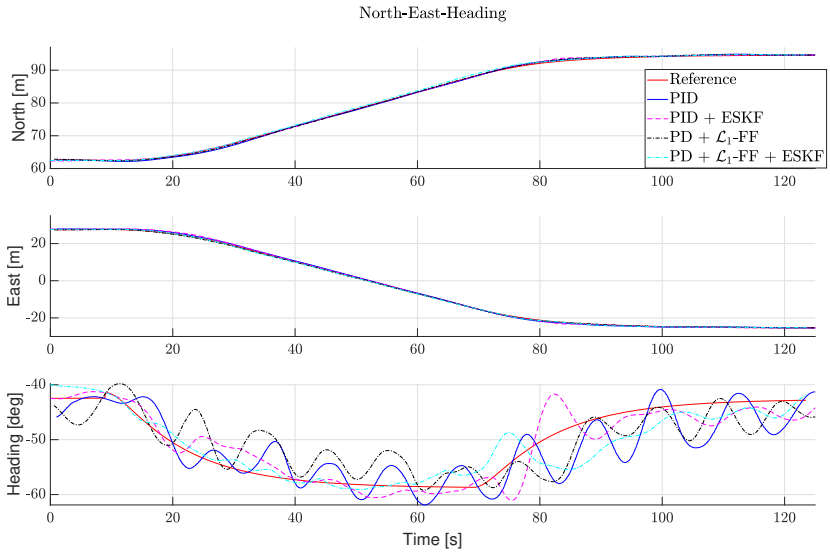
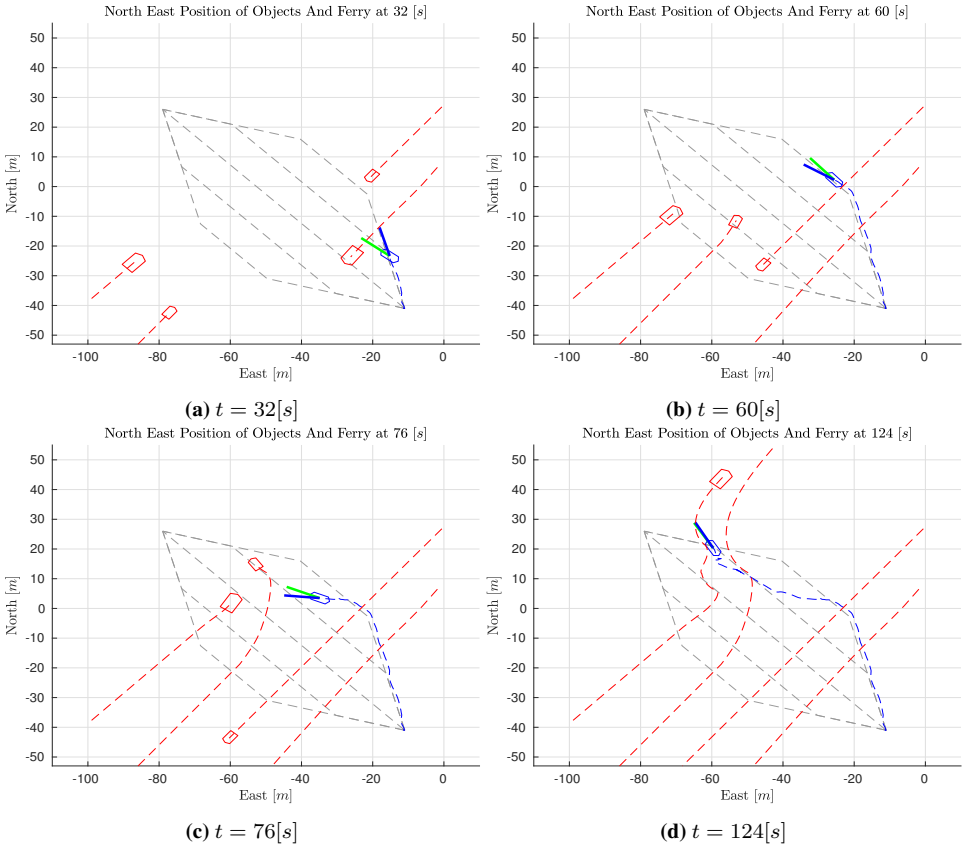


Figure 7.40: Comparison of the individual responses during normal crossing operation with four variations of navigation and motion control systems.

The combined efforts of the new navigation and control systems significantly improve the overall heading accuracy during crossing operations. By inspection of the two variations using the ESKF, the tracking accuracy is improved substantially with the use of the filter. The reduced convergence time of external dynamics with the use of the new controller further improves the heading accuracy.

7.4.3 Collision Avoidance

The new navigation and motion control systems interfaces with a COLAV module to enable the ferry with autonomous capabilities. The full details of the development of the COLAV module are in [Thyri, 2019]. The algorithm uses a path-velocity decomposition approach with predefined paths. A series of different transit scenarios with virtual objects established the ferry's ability to avoid collisions. Figure 7.41 shows one experimental test scenario performed in the harbor basin by Brattøra. More COLAV scenarios are available in Appendix B.



(a) $t = 32[s]$ **(b)** $t = 60[s]$

(c) $t = 76[s]$ **(d)** $t = 124[s]$

Figure 7.41: Snapshots at different time stamps of scenario 5 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

The grey lines are the possible paths the ferry can follow and switch between, the solid blue line is the course of the ferry, the green line is the heading for the ferry, and the dotted blue line is the path of the ferry. The red objects are the virtual objects and their respective paths. The successful integration of the COLAV module, together with the new navigation and motion control systems, marks the implementation of a new upgraded GNC system on milliAmpere capable of autonomous transit.

7.4.4 Discussion

In comparison to the old navigation system, the ESKF struggles to obtain an accurate estimate for the velocity during low speeds. The inaccurate estimate is due to substantial measurement noise in both the acceleration and velocity measurements. The somewhat inaccurate estimate leads to a minimal loss of planar positional accuracy during the 4-corner test. The old navigation node obtains very accurate velocity estimates during low speeds, though at the cost of a phase lag. For slow lateral movement, as is the case for the

4-corner test, this phase lag will have little to no effect, and thus very accurate tracking can be performed when the need for adaptation of external disturbances is small.

The conditions during the 4-corner test presented in this section were excellent. Thus, the need for estimation of the unmodeled dynamics was small. Achieving the same positional accuracy with the new system during low-speed operations in excellent conditions is difficult because of the less accurate velocity estimate. The yaw rate estimate will still suffer from the phase lag with the use of the old navigation node, and the heading stability suffers because of it.

The increased heading stability is the most encouraging improvement because of the introduction of the ESKF and the non-linear adaptive controller. The removal of the phase lag in $\hat{\boldsymbol{\nu}}$ is the main contributor for the improvement. The notorious problem with heading oscillations during transit is almost wholly gone. The increased ability to stabilize the heading is evident in figures 7.40 and 7.42. Figure 7.42 show how the vessel copes with the introduction of an induced time-varying yaw moment. The results further emphasize the improvement the non-linear motion controller and the new navigation system have on the vessel's heading stability. The heading stability and its oscillations have previously been the Achilles heel of the GNC system. The improvement is very encouraging.

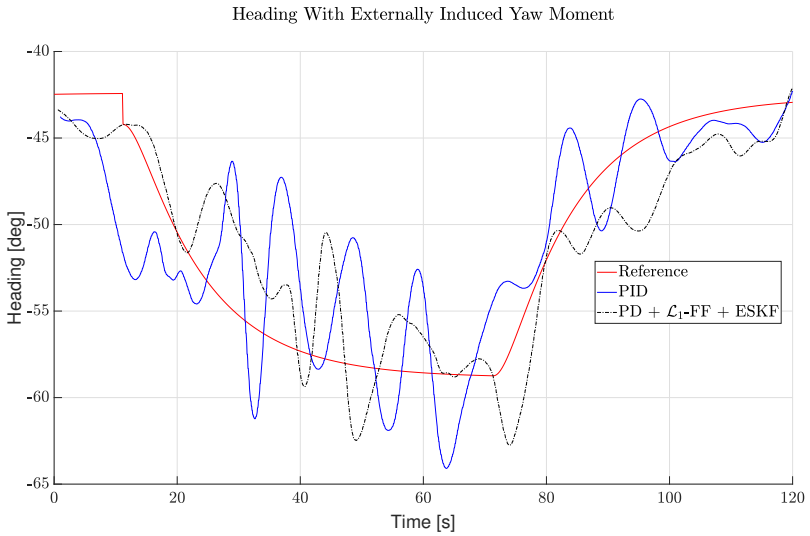


Figure 7.42: Pose during normal crossing operation with induced time-varying yaw moment.

Other means of stabilizing the heading could be through means of hardware. One hardware solution can be an instalment of a couple of fins to the bottom of the hull. The fins will make sure that more yaw moment is required to induce any angular velocity around the z-axis.

The introduction of the ESKF has another upside. The removal of the phase lag in the velocity estimate allows for faster adaptation. The phase lag caused the whole control system to destabilize with the use of more aggressive adaptation. The value for the adaptation gain, γ_{ω_δ}, increased through iterative tuning from 3000 to 20000. The increase in the value of the adaptation law improves the controller's ability to estimate the unmodeled dynamics. During demanding conditions, the faster convergence time for the estimation of the unmodeled dynamics will ensure a more robust controller and more accurate tracking with faster path convergence.

The simulated environment show the improvement in the ability to estimate the unmodeled dynamics. A time varying external disturbance of

$$\boldsymbol{\tau}_d = (\sin(0.05\mathrm{t}) + \frac{1}{20}\cos(\mathrm{t}))[200, -200, 50]^\mathsf{T}$$

is applied to interact with the vessel. Figure 7.43 shows the difference between the estimation of the unmodeled dynamics using the two different approaches.

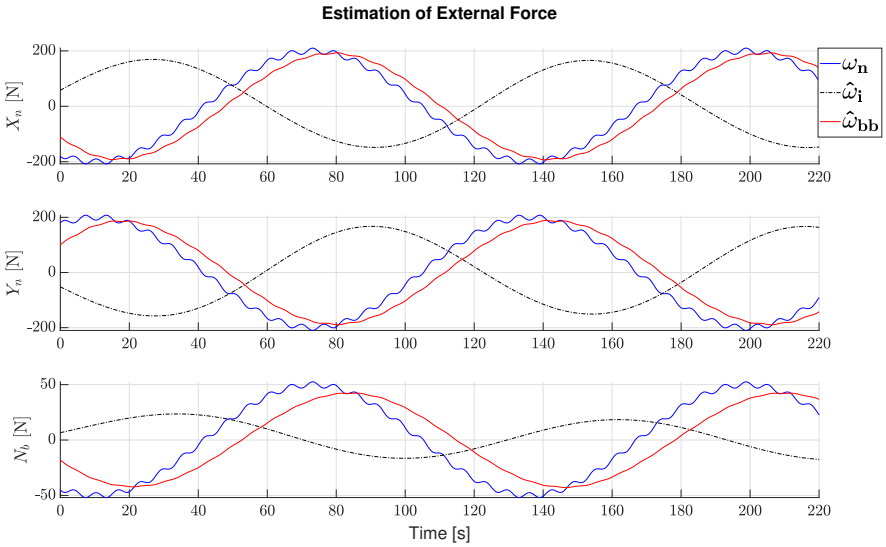


Figure 7.43: Estimation of the external disturbances. $\boldsymbol{\omega}_n$ is the external disturbance, $\hat{\boldsymbol{\omega}}_{bb}$ is the estimated disturbance by the backsteppping framework, and $\boldsymbol{\omega}_i$ is the estimated disturbance with the integrator in the PID controller. Estimation in the simulated environment is performed with the same tuning parameters as on the experimental platform.

The backstepping framework drastically improves the ability to estimate the external disturbances. The linear integrator in the PID controller is significantly worse with a substantial delay in the estimation of the unmodeled dynamics. Further, the new adaptation approach decouples the adaptation from the robustness of the controller, providing a more

robust control system as a whole. All in all, while being tested under scrutiny, the new adaptive control approach outperforms the old PID controller in every way.

There is, however, one drawback to the introduction of the ESKF. The IMU angular velocity measurement noise introduced in the feedback propagates through to the actuators. The increased noise corrupts the desired output force through the derivative component of the controller. The result of the noise is a constant high-frequency oscillation in the error term between the desired turn rate and actual turn rate. The actuators will continuously readjust to the perceived error. Constant readjustment caused by measurement noise will cause unnecessary actuator wear and tear and needs correcting.

7.5 Chapter Summary

This chapter presented and discussed all the experimental results. The experimental tests with the use of the ESKF follow the same procedure as in the simulations. The experimental results shown are three variations of the filter for the final implementation of the estimation algorithm. Testing of the ESKF in a challenging environment containing corrupted measurements further tested the extended abilities of the filter. The results obtained with corrupted measurements were presented along with tests over a prolonged period to establish the stability and robustness of the filter.

This chapter also presented the experimental tests of the control algorithms. The experimental tests were conducted using the same methodology as in the simulations to compare the performance of the individual controllers. The same performance metrics and visualizations, as in the simulations, were used to evaluate the performance of the controllers in the experimental tests.

Lastly, the chapter presented the difference in performance between the new and old implementations. Four different variations were presented to highlight the contributions the different aspects of the implementations had. At the end of the chapter, a COLAV scenario with virtual objects is presented to show the capabilities of the new GNC system.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The new controller introduces feedforward for better transient response, and an adaptive backstepping approach to rapidly estimate and counteract the unmodeled dynamics. These additions have shown to increase the overall accuracy and robustness for the motion control system. The main contributor to the performance boost is the faster convergence of the estimation of unmodeled dynamics. The linear nature of the PID controller puts limitations on the efficiency of the controller when in a non-linear environment. This behavior severely limits the controller's ability to deal with significant disturbances and fast time-varying disturbances. The new non-linear adaptive backstepping approach makes the new controller more suited for estimation of the unmodeled dynamics and increases accuracy.

The new implementation of the navigation system uses sensor fusion of GNSS and IMU data for robust state estimation. The new navigation system performs state estimation without relying on low pass filtered differentiation w.r.t time of the measured states. The new approach successfully removes the phase lag that is present in the old implementation. The removal of the phase lag increases the overall tracking performance of the motion control system. The main advantage is the significant increase in the vessel's ability to track the desired heading.

The new navigation system proves to be able to withstand significant amounts of compromised measurements. It proves to be robust during both normal and abnormal circumstances. The wild point filters implemented in the ESKF serves their purpose gracefully. The wild point filtering successfully detects false measurements and drops them from the estimation process. The different implementations and their combined efforts mark the achievement of the goal to implement a more robust GNC system through sensor fusion and adaptive motion control.

One drawback is apparent with the new state estimation algorithm. The current implemen-

tation does not have any form of filtering of the IMU measurements that drives the filter. Consequently, the measurement noise propagates through to the actuators. Future work should address this issue to reduce the actuator wear and tear.

The successful integration between the new navigation system and the motion control system with the COLAV module conclude a complete overhaul of the existing GNC system. The successful experimental tests show that the ferry can avoid collision given sufficient information about its immediate dynamic environment.

8.2 Future Work

The vessel serves a prototype for an autonomous passenger ferry giving grounds to an excessive amount of possibilities in future work. Different tasks for future work involve everything from cyber-security, human-machine interface (HMI), hardware and software redundancy and fallbacks, risk analysis and the like. Future work mentioned in the current thesis focuses on the different tasks related to the sensor fusion algorithm and the new adaptive control algorithm.

8.2.1 LiDAR Sensor Fusion

The current setup with the ESKF is highly dependent on the GNSS measurements. For any autonomous vessel, this is unacceptable. The GNSS is a vulnerable sensor that can cause severe malfunction due to outside efforts. Outside efforts such as either spoofing or jamming of the GNSS measurements render the sensor useless. Both of these occurrences would be catastrophic for the vessel. Additionally, if the measurements from the GNSS become unreliable or are lost, the measurement noise in the IMU will cause the position prediction to drift rapidly. Introducing other sensor data to the sensor fusion algorithm will minimize the effects of corrupted measurements and increase the robustness of the navigation algorithm.

Other onboard sensors can reduce the dependability on the GNSS. The LiDAR is an active sensor and offers localization data with high accuracy. The LiDAR localization data provides position data redundancy in the sensor fusion algorithm and is an essential step towards a more robust navigation system. The ferry is operating in a small predetermined area, and extensive knowledge of the surrounding environment enables the LiDAR to obtain highly accurate localization data. Using LiDAR for localization, will reduce the dependability of GNSS measurements. [Dalhaug, 2019] looks further into a method for LiDAR localization on milliAmpere. The use of other onboard sensors will downgrade the GNSS sensor to a secondary role.

8.2.2 IMU Sensor Upgrade

The ESKF makes better use of the IMU sensor data on the ferry. However, investing in a new, improved IMU less susceptible to measurement noise will further increase the accuracy of the state estimates. The acceleration measurements have an observed peak-to-peak noise as high as $0.08m/s^2$. The acceleration constraint on the reference filter with

the lowest constrained acceleration is at $0.1m/s^2$. Measurement noise of that magnitude makes it difficult to perform accurate predictions and leads to worse estimation. The IMU sensor should also be moved to make it less susceptible to external vibrations and material deflections caused by movement on deck and thruster usage.

8.2.3 Redundancy and Safety

The new navigation node is more robust against measurement errors and similar unwanted behavior than the old navigation node. However, there is still room for improvement regarding safety. Various safety features can include logging of measurements like that of a flight recorder, sensor redundancy, and different fall back methods in case of malfunction. These safety measures should be a combination of efforts from both hardware and software.

8.2.4 Thruster Dynamics

The various experiments conducted during this thesis revealed issues with the current azimuth thruster setup. The thruster dynamics are too slow and cause considerable delays when the desired change in thrust force is substantial. The problem with the rate saturation is especially troublesome during more aggressive maneuvering. The current thruster setup restricts any high accuracy operation to scenarios with low acceleration.

The most notable struggles are evident when significant deceleration is needed while operating at speed. Multiple of the experimental COLAV scenarios require sudden deceleration. The delay in reversing the thrusters cause the actual thrust force output to be close to zero during that span. This time span is as large as 3-4 seconds at most, which causes deviations from the intended trajectory. Due to the volatility of the orientation of the vessel, these delays cause considerable deviations from the intended heading.

The significant deviations from the desired heading set off a chain of events that cause unwanted oscillations in all degrees of freedom. Due to the large oscillations in the heading, the necessary control input in the BODY frame change rapidly in order to keep the vessel on the right course along the predefined trajectory. The thruster system's inability to track the rapid rate change in the control input causes substantial deviations from the desired pose.

The slow thruster dynamics restrict the ability to perform accurate tracking in scenarios requiring more substantial acceleration. The slow dynamics also restrict the adaptation in the newly implemented controller and rules out the use of the $\mathcal{L}_1$-AB controller completely. A more dynamic thruster system capable of more substantial thrust rate change will significantly improve the ability to track the intended trajectory. The improved tracking ability is achieved through faster adaptation, but also by enabling other non-linear controllers, such as the $\mathcal{L}_1$-AB controller, to be put into use.

The $\mathcal{L}_1$-AB controller showed great promise in simulations and the ability for accurate tracking during more demanding situations. This controller should serve as a viable candidate as the motion control algorithm for a more capable thruster system. The next iteration

of the ferry should improve the overall thruster dynamics, and a thruster setup that allows for higher thrust rate change is a necessity.

8.2.5 Other Future Work

- Further tune the tuning parameters in the ESKF to ensure optimal estimation
- Use a filter to reduce the IMU measurement noise to alleviate unnecessary actuator wear and tear.
- Timing of messages from sensors with GPS time synchronization
- Add handling in case of sensor drop-outs, such as a navigation fallback method and visual warnings.
- Add more sensor data to the sensor fusion algorithm

Bibliography

S.G. Anavatti, F. Santoso, and M. Garratt. Progress in Adaptive Control Systems: Past, Present, and Future. *In Proceedings of the 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 15-17 October*, 2015. doi:10.1109/ICAMIMIA.2015.7537196.

V. L. Bageshwar, D. Gebre-Egziabher, W. L. Garrard, and T. T. Georgiou. A stochastic observability test for discrete-time kalman filters. *Journal of Guidance, Control, and Dynamics*, 32, 2009. doi:10.2514/1.38128.

Y. Bar-Shalom, T. Kirubarajan, and X. R. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, Inc., 2002. ISBN 9780471221272. doi:10.1002/0471221279.

R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press Princeton, NJ, USA, 2012. ISBN 9781400840601. doi:10.2514/1.61067.

M. Breivik. *Topics in Guided Motion Control of Marine Vehicles*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2010.

M. Breivik, S. Kvaal, and P. Østby. From Eureka to K-Pos: Dynamic positioning as a highly sucessful and important marine control technology. *In Proceedings of the 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC: Copenhagen, Denmark, 24–26 August*, 2015. doi:10.1016/j.ifacol.2016.01.001.

E. F. Brekke. *Fundamentals of Sensor Fusion: Target Tracking, Navigation and SLAM*. Unpublished, 2018.

N. Dalhaug. Lidar-based localization for autonomous ferry. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2019.

T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011. ISBN 9781119994138. doi:10.1002/9781119994138.

K. Gade. The seven ways to find heading. *Journal of Navigation*, pages 1–16, April 2016. doi:10.1017/S0373463316000096.

Hemisphere. Vector VS330 GNSS Compass (last accessed 10.06.2019). URL https://www.navtechgps.com/assets/1/7/VS330_DataSheet_WEB_09.11.2015.pdf.

N. Hovakimyan and C. Cao. *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM, 2010. ISBN 9780898717044. doi:10.1137/1.9780898719376.

R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering (ASME)*, 82D:35–45, 01 1960. doi:10.1115/1.3662552.

O. N. Lyngstadaas. Ship motion control concepts considering actuator constraints. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2018.

P. Mustvedt. Autonom ferge designet for å frakte 12 passasjerer trygt over nidelven. Master's thesis, Department of Design, Norwegian University of Science and Technology, Trondheim, Norway, 2019.

A. A. Pedersen. Optimization-based system identification for the milliAmpere ferry. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2019.

ROS. ROS tutorial (last accessed 10.06.2019). URL http://wiki.ros.org/ROS/Tutorials.

N. Sae-Khow. Klart for ubemannede skip i trondheimsfjorden (last accessed 10.06.2019). 2016. URL https://www.nrk.no/trondelag/klart-for-ubemannede-skip-i-trondheimsfjorden-1.13156068.

SNAME. Sname1950 (last accessed 10.06.2019). URL http://www.soi.wide.ad.jp/class/20070026/slides/04/27.html.

J. Solà. Quaternion kinematics for the error-state Kalman filter, Article (last accessed 10.06.2019). 2017. URL http://www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf.

A J. Sørensen. Marine Cybernetics. *Lecture Notes, Norwegian University of Science and Technology, Trondheim, Norway*, 2018.

M. E. N. Sørensen and M. Breivik. Comparing nonlinear adaptive motion controllers for marine surface vessels. *In Proceedings of the 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC: Copenhagen, Denmark 24–26 August*, 2015. doi:10.1016/j.ifacol.2015.10.295.

E. Thyri. A path-velocity decomposition approach to collision avoidance for autonomous passenger ferries. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2019.

WGS84. WGS84 (last accessed 10.06.2019). URL https://gisgeography.com/wgs84-world-geodetic-system/.

Xsens. Xsens MTi user manual (last accessed 10.06.2019). URL https://sensores-de-medida.es/wp-content/uploads/2017/08/mti_user_manual.pdf.

Appendix A

Dissemination

A.1 Coverage

During the project and thesis work, milliAmpere has generated a lot of media attention. The media outlets are an important part of the continued development of autonomous vessels. The media exposure serve as an important part in bringing attention to the field of autonomy the potential of this technology.

Figure A.1: Brage (left) on board milliAmpere during a demo at Sjøsikkerhetskonferansen in Haugesund. Photo courtesy of Teknisk Ukeblad

Articles and Media

- NRK TV, milliAmpère på distriktsnyheter (26.09.2018): *https://tv.nrk.no/serie/distriktsnyheter-rogaland/DKRO97092618/26-09-2018#t=2m43s*
- NRK Radio, Innslag i Haugesund (26.09.2018): *https://radio.nrk.no/serie/distriktsprogram-rogaland/DKRO02019318/26-09-2018#t=8m27.16s*
- Skipsrevyen, innslag med milliAmpère (26.09.2018): *https://www.skipsrevyen.no/article/autonomt-innslag-paa-sjoesikkerhetskonferansen/*
- TV Haugaland, visning av milliAmpère (27.09.2018): *https://hnytt.no/2018/09/27/her-kjorer-den-forerlose-ferga-i-smedasundet/*
- Teknisk Ukeblad, milliAmpère i Haugesund (28.09.2018): *https://www.tu.no/artikler/her-demonstrerer-de-selvkjorende-milliampere-na-skal-de-bygge-ferge-pa-12-meter/447398?key=aTPPxOqK*
- Stand på DN Havet Konferansen med besøk av kronprins Håkon (16.10.2018): *https://www.facebook.com/havetDN/videos/1130244403799605/*
- Dagens Bilde i Adresseavisa (20.11.2018)

A.2 Ocean Week Trondheim

During Ocean Week in Trondheim, the author held a pitch for his majesty the king and other excellences about the use of autonomous passenger ferries and their potential impact in the future. The presentation is shown below.

Autonomous Passenger Ferries – Revitalizing Urban Waterways

Brage Sæther, MSc Engineering Cybernetics, NTNU

Vision

- Take the load of city roads
- Efficient

Mobility-as-a-Service

- On-demand
- Connected Network


Future Impact
• Flexibility
• Sustainability
Image courtesy of Petter Mustvedt, NTNU


Technological Advances
• Technologically feasible
• Situational awareness


Norway as a Maritime Nation
• Leading role
• Research at NTNU
• Realize the future

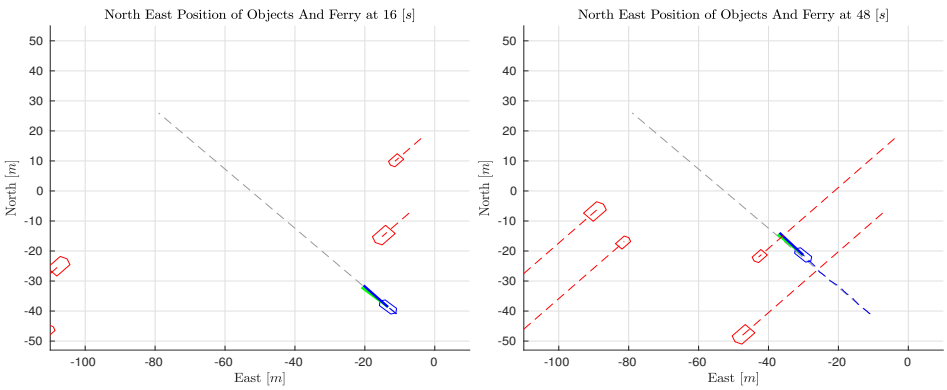

Thank You For
Your Time
https://www.ntnu.edu/autoferry/

Figure A.2: Pitch during Ocean Week May 7^{th} 2019. Photo courtesy of Morten Breivik.

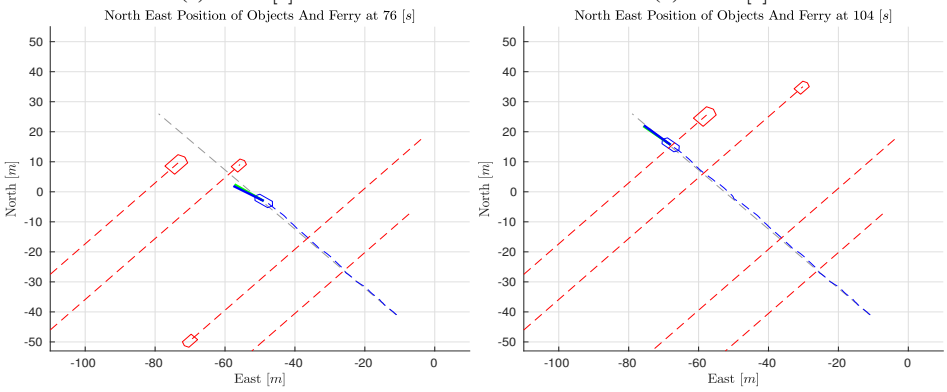
Appendix B

COLAV Experiments

Experimental test results with the new navigation and motion control systems interfaced with a COLAV system. Section 7.4.3 shows the scenario for experimental COLAV transit 5.

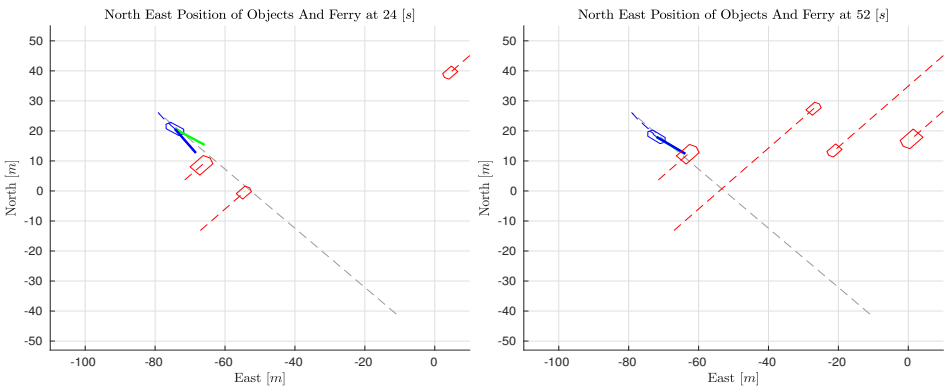


(a) $t = 16[s]$ **(b)** $t = 48[s]$

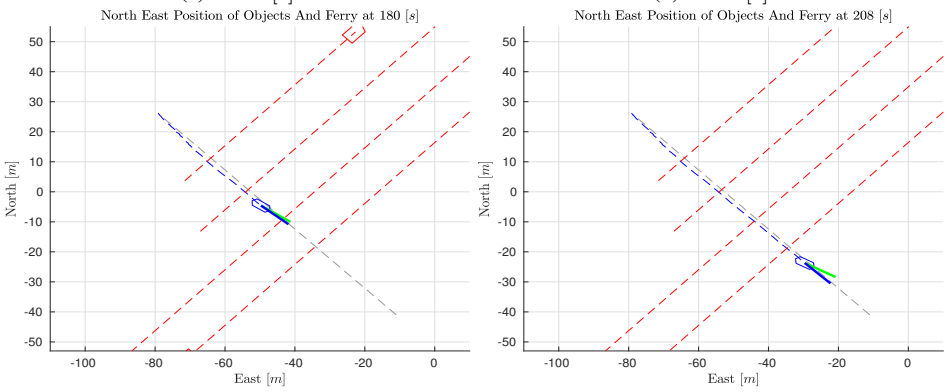


(c) $t = 76[s]$ **(d)** $t = 104[s]$

Figure B.1: Snapshots at different time stamps of scenario 1 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

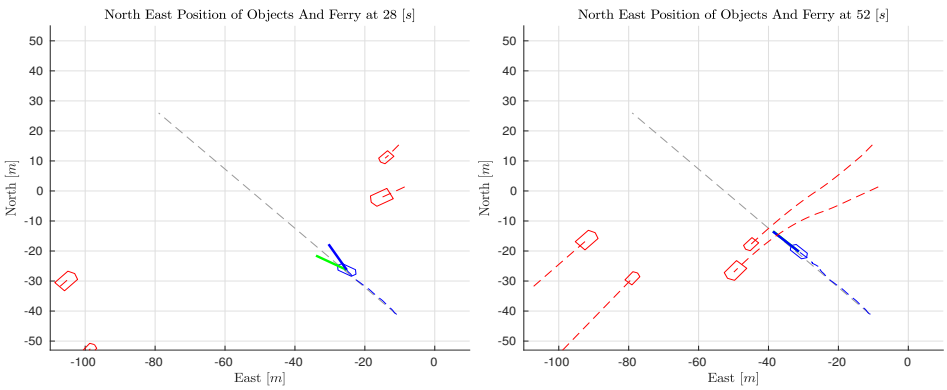


(a) $t = 24[s]$ **(b)** $t = 52[s]$

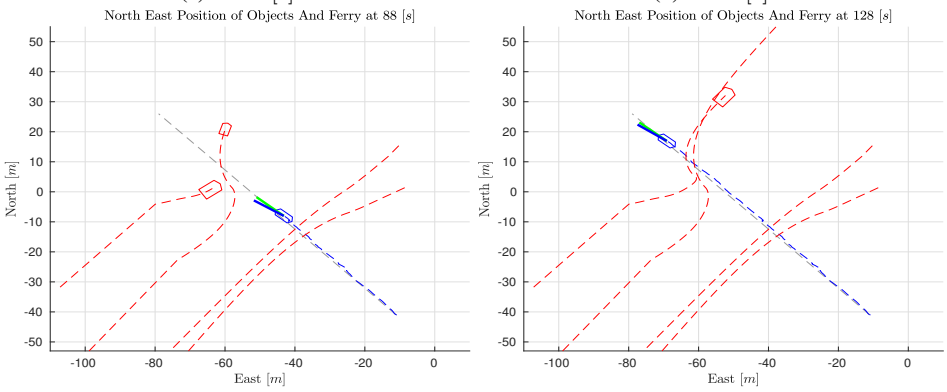


(c) $t = 180[s]$ **(d)** $t = 208[s]$

Figure B.2: Snapshots at different time stamps of scenario 2 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

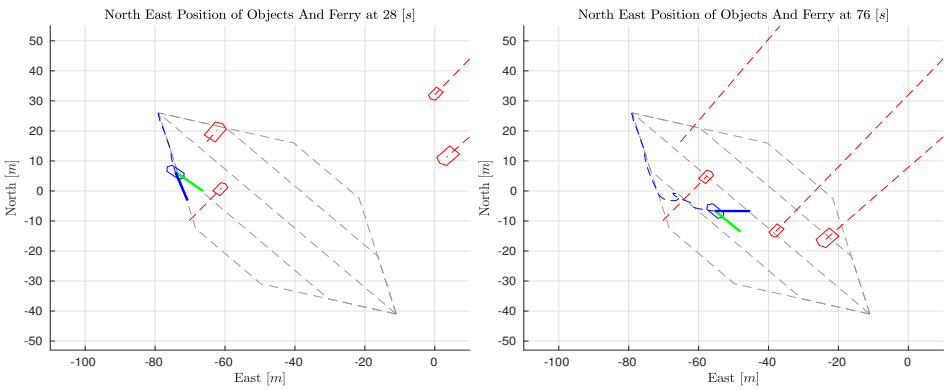


(a) $t = 84[s]$ **(b)** $t = 96[s]$

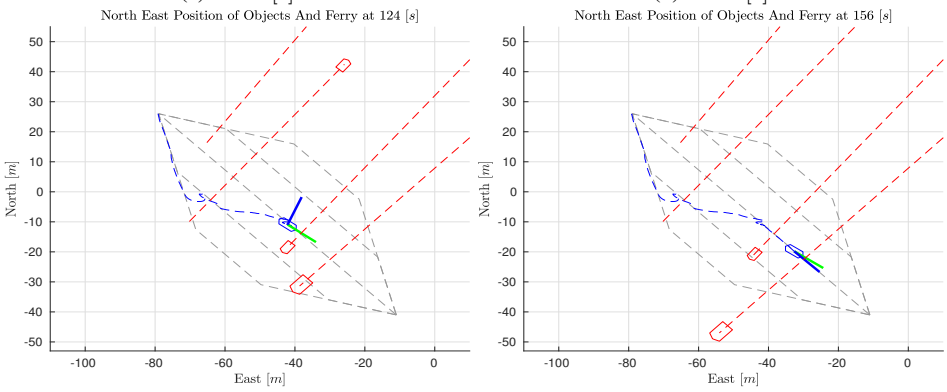


(c) $t = 128[s]$ **(d)** $t = 148[s]$

Figure B.3: Snapshots at different time stamps of scenario 3 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

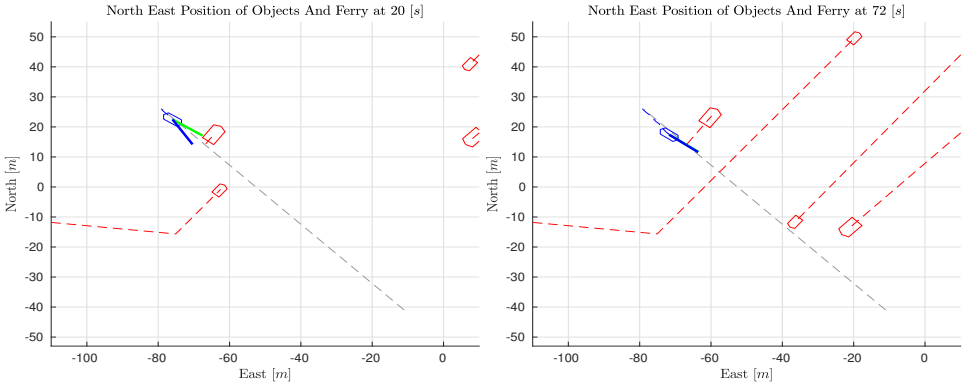


(a) $t = 28[s]$ **(b)** $t = 76[s]$



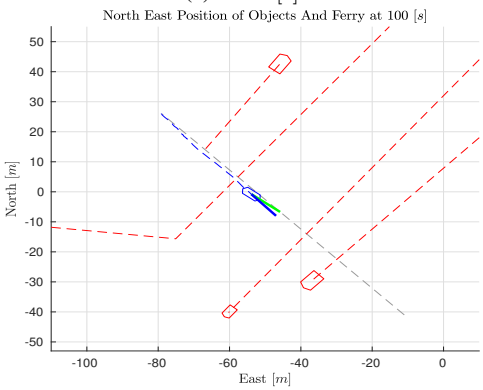
(c) $t = 124[s]$ **(d)** $t = 156[s]$

Figure B.4: Snapshots at different time stamps of scenario 4 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

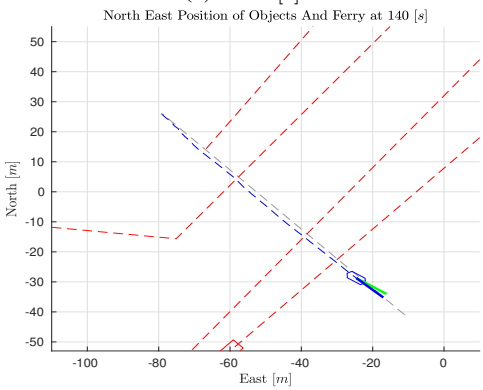


(a) $t = 20[s]$

(b) $t = 72[s]$

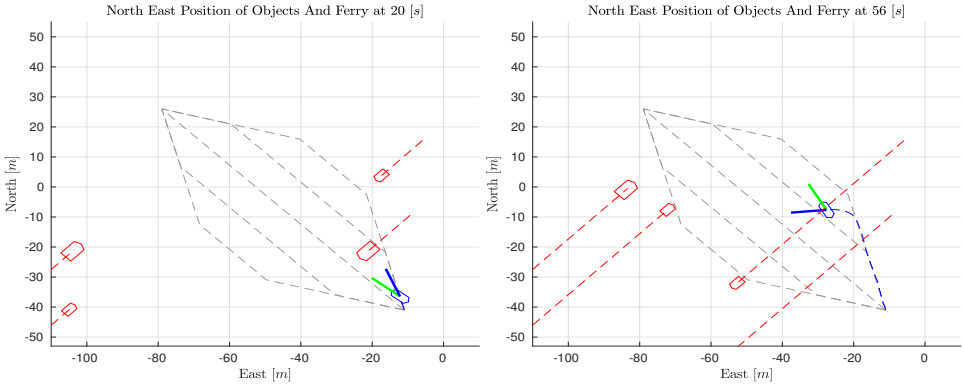


(c) $t = 100[s]$

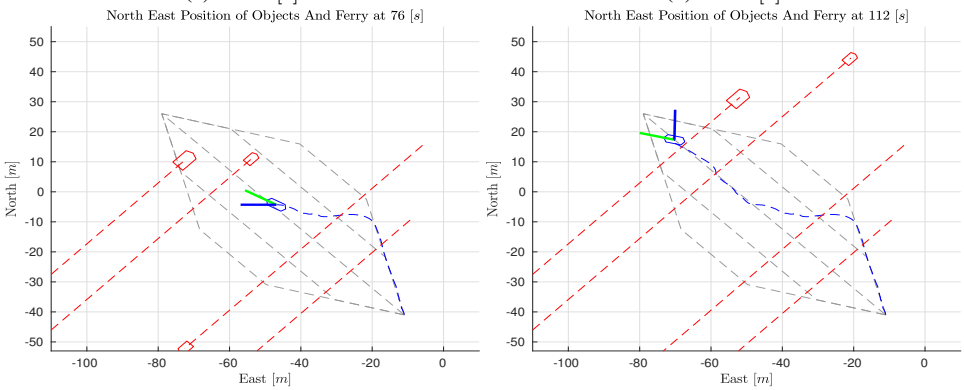


(d) $t = 140[s]$

Figure B.5: Snapshots at different time stamps of scenario 6 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

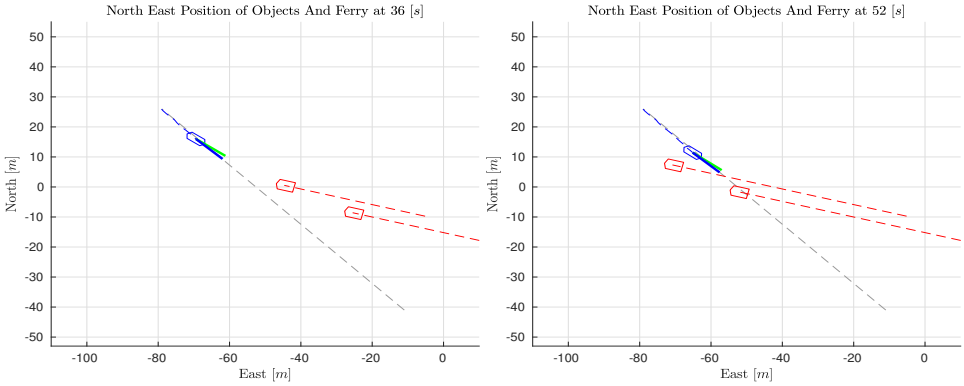


(a) $t = 20[s]$ **(b)** $t = 56[s]$



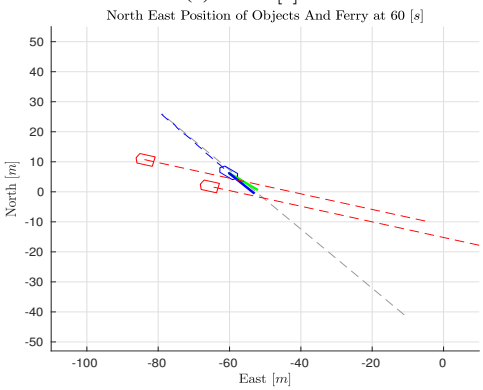
(c) $t = 76[s]$ **(d)** $t = 132[s]$

Figure B.6: Snapshots at different time stamps of scenario 7 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

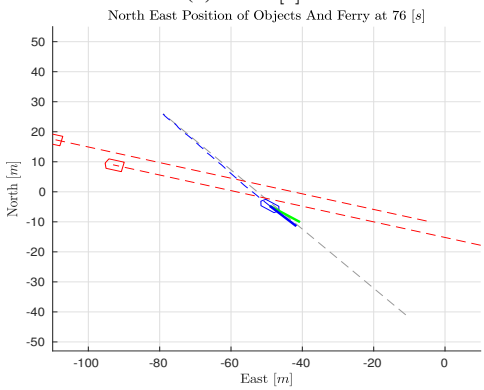


(a) $t = 36[s]$

(b) $t = 52[s]$



(c) $t = 60[s]$



(d) $t = 76[s]$

Figure B.7: Snapshots at different time stamps of scenario 8 during the experimental COLAV tests. Snapshots courtesy of [Thyri, 2019].

Appendix C

Test Plan for Experimental Tests

C.1 Experimental Test Plan for ESKF

The ESKF can be challenging to tune, as many parameters need to be adjusted. Especially tricky is the tuning of the process noise matrix Q. The matrix does not constitute any real quantity but is a weighting matrix for the acceleration and angular velocity measurements.

A systematic test plan is put in place to minimize the time and effort needed to obtain decent tuning parameters. A proper test plan hopefully results in the best tuning parameters possible.

The ROS framework has a convenient feature that enables the ability to record the data available in the various topics in the system. This feature allows for easy logging of all the data streams that are present on the OBC. The rosbag feature stores the recorded data in a rosbag. This rosbag can then be played back in real-time, appearing as for the observer that the data is coming in real-time in a real scenario. This feature is used heavily. By using rosbags to store data, the measurements and dynamics of the ferry in a real scenario can be played back while working in an office. Analyzing data in an office provides better working conditions and better opportunities to analyze the results and tune the parameters simultaneously.

C.1.1 Test 1, Initial Tuning

The first test will perform initial sensor checks and record sensor data during regular operation. The initial tuning of the filter uses the recorded data as input and error correction. Test 1 focuses on the following tasks

- Integrate the observer node with the rest of the system on milliAmpere
- Inspect all the sensor components for mounting issues

- Double check both sensor lever arms
- Record data during normal operation for use in "online tuning" using the rosbag feature in ROS
- Play back recorded data in the office at a proper workstation using the rosbag feature in ROS.
- Use the NIS test actively for judgement of estimation accuracy.

After testing according to test 1, the tuning parameters are expected to be well-tuned.

C.1.2 Test 2, Robustness and Corrupted Measurements

The second part of the ESKF tests will focus on prolonged estimation and robustness. After the initial tuning of the filter based on the recorded data from test 1, the filter parameters are expected to well-tuned. These tests investigate if the filter can withstand other outside factors and to be able to perform reliant estimation while under duress. The robustness of the filter is tested by doing the following

- While the ferry is moored, record data over an extended period with the observer in the loop performing live estimation. The extended period will give an indication of any error occurring after prolonged estimation
- Force spikes in IMU measurements by stomping and jumping on deck to induce false measurements.
- Record the ESKF's ability to cope with corrupted IMU measurements.
- Analyze data back at the office.
- Use NIS test results actively in the tuning process.
- Use data to tweak parameters slightly if needed

After test 1 and test 2, the tuning parameters should be close to final. At this stage, the expectation is excellent estimation performance.

C.1.3 Test 3, Live Estimation

Test 3 focuses on filter estimation during an operation where the ESKF is working as the navigation node. The main goal of the tests at this stage is to document the different effects the new navigation node has on the GNC system as a whole. No parameter tuning is expected at this stage, but parameters will be adjusted if necessary. The test is summarizing in these bullet points

- Perform crossing tests with the ESKF as the working navigation node
- Adjust parameters after initial runs if necessary
- Perform 4 corner tests both with the ESKF and with the old navigation node
- Investigate tracking performance with the new navigation node

At this stage, the tuning parameters for the filter should be final. A final consistency test will be performed during prolonged regular operation to ensure proper and reliable filter estimates. Any last changes to the tuning parameters will be conducted based on this consistency test.

C.2 Experimental Test Plan for Adaptive Controller

The intentions for the experiments are to tune the newly designed control algorithm on the ferry. By performing the 4 corner test, any weaknesses become apparent, and the individual non-coupled and coupled motions are easy to analyze. The bullet points summarize the test plan

- Integrate the new controller node into the existing system on milliAmpere

  The new ROS node is set up in such a way that it should communicate with the existing system already in place. The integration of the new node should be quick.

- Tune the adaptation law for the adaptive feedforward

  The PD controller with the adaptive feedforward utilizes the P and D terms in the previous controller already in place on milliAmpère. The adaptation gain needs to be tuned based on the measurement noise in the different sensors, as well as the systems ability to adhere to the commanded thrust vector.

  Initial tuning is carried out while performing station keeping. The adaptation gain is increased with small increments while observing the estimation parameter $\hat{\omega}$. The adaptation gain is slowly increased online until measurement noise manifests itself in the estimated parameter.

- Perform 4 corner test with the roughly tuned PD + $\mathcal{L}_1 - FF$ controller with small variations in parameter value.

  Perform multiple 4 corner test with varying adaptation gain to find the best parameter value. By performing multiple tests with variations in the adaptation gain based on maximum value found, the best possible tuning is found by analyzing the accuracy of the controller with the help of the performance metrics in Section 6.2.2.

- Test the impact of the low pass filter on the adaptive control input

  The low pass filter reduces high oscillatory effects introduced by the adaptation. The time constant needs to be sufficiently large to remove the unwanted effects, but also sufficiently low to cause minimal lag on the desired thrust force. Tune the filter constant to achieve the best possible results.

Appendix D

Robot Operating System

The implementations in this thesis utilize the ROS framework heavily. The features rostopics and rosbag are especially widely used. This appendix serve as a brief introduction to the essential aspects of the main ROS features used. The tutorials are taken from the ROS website [ROS].

Understanding ROS Topics

roscore

Let's start by making sure that we have roscore running, **in a new terminal**:

```
$ roscore
```

If you left roscore running from the last tutorial, you may get the error message:

```
- roscore cannot run as another roscore/master is already running.
- Please kill other roscore/master processes before relaunching
```

This is fine. Only one roscore needs to be running.

ROS Topics

The `turtlesim_node` and the `turtle_teleop_key` node are communicating with each other over a ROS **Topic**. `turtle_teleop_key` is **publishing** the key strokes on a topic, while `turtlesim` **subscribes** to the same topic to receive the key strokes. Let's use rqt_graph which shows the nodes and topics currently running.

Note: If you're using `electric` or earlier, `rqt` is not available. Use `rxgraph` instead.

Introducing rostopic

The `rostopic` tool allows you to get information about ROS **topics**.

You can use the help option to get the available sub-commands for `rostopic`

```
$ rostopic -h
```

```
- rostopic bw     display bandwidth used by topic
- rostopic echo   print messages to screen
- rostopic hz     display publishing rate of topic
- rostopic list   print information about active topics
- rostopic pub    publish data to topic
- rostopic type   print topic type
```

Or pressing `tab` key after `rostopic` prints the possible sub-commands:

```
$ rostopic
bw    echo  find  hz    info  list  pub   type
```

Let's use some of these topic sub-commands to examine turtlesim.

Using rostopic echo

`rostopic echo` shows the data published on a topic.

Usage:

```
rostopic echo [topic]
```

Let's look at the command velocity data published by the `turtle_teleop_key` node.

For ROS Hydro and later, this data is published on the `/turtle1/cmd_vel` topic. **In a new terminal, run:**

```
$ rostopic echo /turtle1/cmd_vel
```

For ROS Groovy and earlier, this data is published on the `/turtle1/command_velocity` topic. **In a new terminal, run:**

```
$ rostopic echo /turtle1/command_velocity
```

You probably won't see anything happen because no data is being published on the topic. Let's make `turtle_teleop_key`publish data by pressing the arrow keys. **Remember if the turtle isn't moving you need to select the `turtle_teleop_key` terminal again.**

ROS Messages

Communication on topics happens by sending ROS **messages** between nodes. For the publisher (`turtle_teleop_key`) and subscriber (`turtlesim_node`) to communicate, the publisher and subscriber must send and receive the same **type** of message. This means that a topic **type** is defined by the message **type** published on it. The **type** of the message sent on a topic can be determined using `rostopic type`.

rostopic continued

Now that we have learned about ROS **messages**, let's use rostopic with messages.

Using rostopic pub

`rostopic pub` publishes data on to a topic currently advertised.

Usage:

```
rostopic pub [topic] [msg_type] [args]
```

For ROS Hydro and later, example:

```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]'
'[0.0, 0.0, 1.8]'
```

For ROS Groovy and earlier, example:

```
$ rostopic pub -1 /turtle1/command_velocity turtlesim/Velocity  -- 2.0  1.8
```

The previous command will send a single message to turtlesim telling it to move with a linear velocity of 2.0, and an angular velocity of 1.8 .

This is a pretty complicated example, so lets look at each argument in detail.

For ROS Hydro and later,

- This command will publish messages to a given topic:

```
rostopic pub
```

- This option (dash-one) causes rostopic to only publish one message then exit:

```
 -1
```

- This is the name of the topic to publish to:

```
/turtle1/cmd_vel
```

- This is the message type to use when publishing to the topic:

```
geometry_msgs/Twist
```

- This option (double-dash) tells the option parser that none of the following arguments is an option. This is required in cases where your arguments have a leading dash -, like negative numbers.

```
--
```

- As noted before, a geometry_msgs/Twist msg has two vectors of three floating point elements each: `linear` and `angular`. In this case, `'[2.0, 0.0, 0.0]'` becomes the linear value with `x=2.0`, `y=0.0`, and `z=0.0`, and `'[0.0, 0.0, 1.8]'` is the `angular` value with `x=0.0`, `y=0.0`, and `z=1.8`. These arguments are actually in YAML syntax, which is described more in the YAML command line documentation.

```
'[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

For ROS Groovy and earlier,

- This command will publish messages to a given topic:

```
rostopic pub
```

- This option (dash-one) causes rostopic to only publish one message then exit:

```
 -1
```

- This is the name of the topic to publish to:

```
/turtle1/command_velocity
```

- This is the message type to use when publishing to the topic:

```
turtlesim/Velocity
```

- This option (double-dash) tells the option parser that none of the following arguments is an option. This is required in cases where your arguments have a leading dash -, like negative numbers.

```
--
```

- As noted before, a turtlesim/Velocity msg has two floating point elements : `linear` and `angular`. In this case, `2.0`becomes the linear value, and `1.8` is the `angular` value. These arguments are actually in YAML syntax, which is described more in the YAML command line documentation.

```
2.0 1.8
```

You may have noticed that the turtle has stopped moving; this is because the turtle requires a steady stream of commands at 1 Hz to keep moving. We can publish a steady stream of commands using `rostopic pub -r` command:

For ROS Hydro and later,

- `$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, -1.8]'`

For ROS Groovy and earlier,

- `$ rostopic pub /turtle1/command_velocity turtlesim/Velocity -r 1 -- 2.0  -1.8`

This publishes the velocity commands at a rate of 1 Hz on the velocity topic.

Understanding ROS Services and Parameters

ROS Services

Services are another way that nodes can communicate with each other. Services allow nodes to send a **request** and receive a **response**.

Using rosservice

`rosservice` can easily attach to ROS's client/service framework with services. `rosservice` has many commands that can be used on services, as shown below:

Usage:

```
rosservice list         print information about active services
rosservice call         call the service with the provided args
rosservice type         print service type
rosservice find         find services by service type
rosservice uri          print service ROSRPC uri
```

rosparam list

```
$ rosparam list
```

Here we can see that the turtlesim node has three parameters on the param server for background color:

- /background_b
- /background_g
- /background_r
- /rosdistro
- /roslaunch/uris/host_57aea0986fef__34309
- /rosversion
- /run_id

Recording and playing back data

Recording data (creating a bag file)

This section of the tutorial will instruct you how to record topic data from a running ROS system. The topic data will be accumulated in a bag file.

First, execute the following commands in separate terminals:

Terminal 1:

```
roscore
```

Terminal 2:

```
rosrun turtlesim turtlesim_node
```

Terminal 3:

```
rosrun turtlesim turtle_teleop_key
```

This will start two nodes - the turtlesim visualizer and a node that allows for the keyboard control of turtlesim using the arrows keys on the keyboard. If you select the terminal window from which you launched turtle_keyboard, you should see something like the following:

```
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
```

Pressing the arrow keys on the keyboard should cause the turtle to move around the screen. Note that to move the turtle you must have the terminal from which you launched turtlesim selected and not the turtlesim window.

Recording all published topics

First lets examine the full list of topics that are currently being published in the running system. To do this, open a new terminal and execute the command:

```
rostopic list -v
```

This should yield the following output:

```
Published topics:
 * /turtle1/color_sensor [turtlesim/Color] 1 publisher
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher
 * /rosout [rosgraph_msgs/Log] 2 publishers
 * /rosout_agg [rosgraph_msgs/Log] 1 publisher
```

```
 * /turtle1/pose [turtlesim/Pose] 1 publisher


Subscribed topics:
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 subscriber
 * /rosout [rosgraph_msgs/Log] 1 subscriber
```

The list of published topics are the only message types that could potentially be recorded in the data log file, as only published messages are recorded. The topic /turtle1/cmd_vel is the command message published by teleop_turtle that is taken as input by the turtlesim process. The messages /turtle1/color_sensor and /turtle1/pose are output messages published by turtlesim.

We now will record the published data. Open a new terminal window. In this window run the following commands:

```
mkdir ~/bagfiles
cd ~/bagfiles
rosbag record -a
```

Here we are just making a temporary directory to record data and then running **rosbag record** with the option -a, indicating that all published topics should be accumulated in a bag file.

Move back to the terminal window with turtle_teleop and move the turtle around for 10 or so seconds.

In the window running rosbag record exit with a Ctrl-C. Now examine the contents of the directory ~/bagfiles. You should see a file with a name that begins with the year, date, and time and the suffix .bag. This is the bag file that contains all topics published by any node in the time that **rosbag record** was running.

The limitations of rosbag record/play

In the previous section you may have noted that the turtle's path may not have exactly mapped to the original keyboard input - the rough shape should have been the same, but the turtle may not have exactly tracked the same path. The reason for this is that the path tracked by turtlesim is very sensitive to small changes in timing in the system, and rosbag is limited in its ability to exactly duplicate the behavior of a running system in terms of when messages are recorded and processed by rosrecord, and when messages are produced and processed when using rosplay. For nodes like turtlesim, where minor timing changes in when command messages are processed can subtly alter behavior, the user should not expect perfectly mimicked behavior.

Appendix E

GUI

The GUI was made to streamline online testing on milliAmpère. The GUI features interactive buttons providing the possibility of switching states in the state machine. It also visualizes the azimuth thruster angles such that it is easy to discover any potential malfunctions related to the azimuth thrusters. Along with the visualization of the angles of the azimuth thrusters, the GUI presents various data for easier debugging. The GUI has since been updated with a flashing alarm that triggers if any predefined events are to occur.

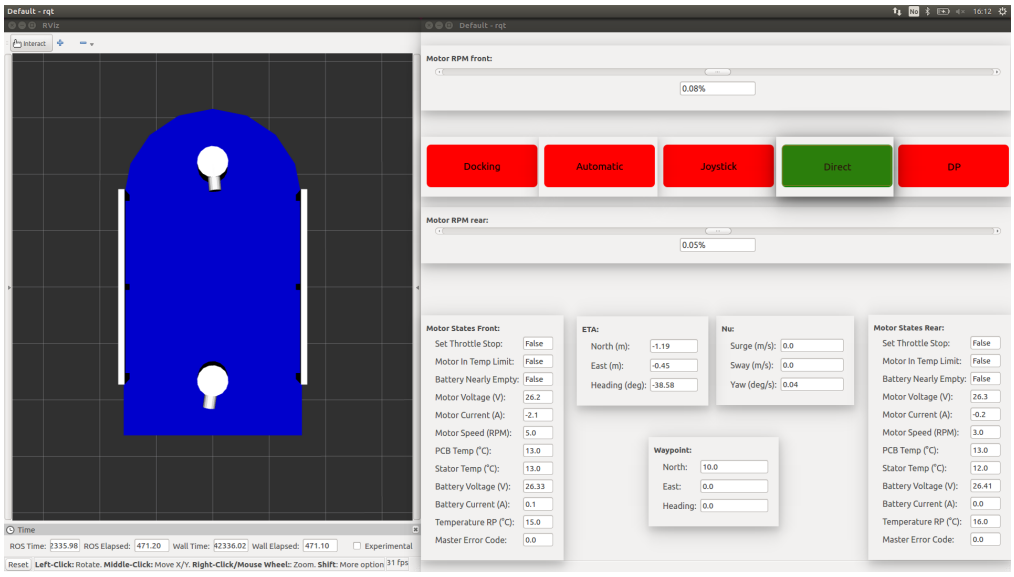


Figure E.1: GUI made with RVIZ, RQT and QT

Appendix F

Model Parameters

Parameter	Value	Unit
m_{11}	2131.80	kg
m_{12}	1.00	kg
m_{13}	141.02	kgm
m_{21}	-15.87	kg
m_{22}	2231.89	kg
m_{23}	-1244.35	kgm
m_{31}	-423.76	kgm
m_{32}	-397.64	kgm
m_{33}	4351.56	kgm^2
X_u	-68.676	$\frac{kg}{s}$
$X_{\|u\|u}$	-50.08	$\frac{kg}{m}$
X_{uuu}	-14.93	$\frac{kgs}{m^2}$
X_v	-25.20	$\frac{kg}{s}$
X_r	-145.30	$\frac{kgm}{s}$
Y_u	90.15	$\frac{kg}{s}$
Y_v	-8.69	$\frac{kg}{s}$
$Y_{\|v\|v}$	-189.08	$\frac{kg}{m}$
Y_{vvv}	-0.00613	$\frac{kgs}{s^2}$
$Y_{\|r\|v}$	–3086.95	kg
Y_r	-24.09	$\frac{kgm}{s}$
$Y_{\|v\|r}$	-338.32	kg
$Y_{\|r\|r}$	1372.06	kgm^2
N_u	-38.00	$\frac{kgm}{s}$
N_v	-97.26	$\frac{kgm}{s}$
$N_{\|v\|v}$	-18.85	kg
$N_{\|r\|v}$	5552.23	kgm
N_r	-230.19	$\frac{kgm^2}{s}$
$N_{\|r\|r}$	-0.0063031	kgm^2
N_{rrr}	-0.0006723	$kgms$
$N_{\|v\|r}$	-5888.89	kgm

Table F.1: Estimated model parameters for the milliAmpere ferry [Pedersen, 2019].

Brage Sæther

Development and Testing of Navigation and Motion Control Systems for milliAmpere

NTNU
Norwegian University of
Science and Technology