Precision Control of an Autonomous Surface Vessel

2nd Semester Master's Program in Control and Automation
Department of Electronic Systems
Aalborg University

Alejandro Alonso García, Anders Egelund Kjeldal, Himal Kooverjee,
Niels Skov Vestergaard and Noelia Villarmarzo Arruñada
Group 832

AALBORG UNIVERSITY
STUDENT REPORT

2nd Semester Project
Master's Program in
Control and Automation
Department of Electronic Systems
Fredrik Bajers Vej 7C, 9220 Aalborg

Title:
Precision Control of an Autonomous Surface Vessel

Theme:
Multivariable Control

Project Period:
Spring 2017

Project Group:
832

Participants:
Alejandro Alonso García
Anders Egelund Kjeldal
Himal Kooverjee
Niels Skov Vestergaard
Noelia Villarmarzo Arruñada

Supervisor:
Jesper Abildgaard Larsen

Pages: 106
Appendices: 10
Attachments: 1

Concluded: 30/06/2017

Synopsis:

Autonomous surface vessels (ASVs) have a wide range of applications that goes from marine research to surveillance. The aim of this project is to design a control strategy for an ASV with the purpose of acquiring bathymetric measurements. The vessel at hand is equipped with two thrusters, a Real Time Kinematic (RTK) GPS module, an Inertial Measurement Unit (IMU) and two processing units. The control system is divided in two levels, an inner and an outer controller. The former is designed using both a Linear Quadratic Regulator (LQR) and robust control theory, and handles the heading and speed of the vessel. The performance and robustness of these two approaches is also compared and analyzed. The outer controller, on the other hand, makes the vessel track a path by computing its appropriate heading and speed using a enclosure based steering algorithm. This control system receives data from the vessel sensors through two Kalman filters, which estimate the attitude and translational variables of the vessel.

Preface

The focus of this project is to design a control system for an autonomous surface vessel, such that it can navigate autonomously over a predetermined area in the water.

This report has been written by a group of students on the second semester of the Master in Control and Automation at Aalborg University in the spring semester of 2017. It has been supervised by Jesper Abildgaard Larsen, associate professor at the Institute of Electronic Systems at Aalborg University.

The group would like to thank Jens Frederik Dalsgaard Nielsen for his guidance during the setup of the GPS, and Palle Andersen for his help in the robust control design, both associate professors at the Institute of Electronic Systems Aalborg University.

The reader is expected to have a basic knowledge within physics and mathematics, as well as in modeling and linear control theory.

Reading Instructions

- The report is divided in three parts. Part I deals with the analysis of the system, which includes a description of the setup and the derivation of the dynamic model. Part II includes the control design, which contains the two approaches for an inner controller, the outer controller, the sensor fusion and the implementation. Part III includes the results of the project, the discussion and the conclusion.
- The report also includes appendixes that contain the journals for the different tests and other relevant information.
- The bibliography is written using ISO 690, noted as [x], and it is included at the end of the report, after the appendixes.
- An attachment is included as part of the report, and contains MATLAB scripts and simulations files, test data files and the ROS workspace files used in the project.

Text by:

Alejandro Alonso García

Anders Egelund Kjeldal

Himal Kooverjee

Niels Skov Vestergaard

Noelia Villarmarzo Arruñada

Contents

Part I

Pre-Analysis

1 | Introduction

Autonomous surface vessels (ASV) have a wide range of applications such as environmental monitoring [1, p. 745], meteorological data collection, marine biological research and surveillance [2, p. 8-10]. The design of an ASV raises some interesting control challenges, varying between the intended use case.

One such use case is shown in Figure 1.1. This is an ASV designed to aid with search and rescue missions in floods. It can be dangerous and difficult for rescuers to search in flooded areas. A drone is able to reach otherwise inaccessible places and provides better vision horizons than an ASV would. The drone however has short battery life, which limits the reach and duration of the mission.

Figure 1.1: An air surface system for search in flooded areas under search and rescue missions. [3]

Here the ASV provides long battery life and by carrying the drone, until needed for improved overview, extends reach and duration of each mission. [3]

Another application is for automated survey of an area. Bathymetric measurements can be used for efficient and safe guidance of marine vessels in shallow waters. It is also interesting when studying biological oceanography where it i.e. can help in deciding which areas to protect for preservation of sea life [4].

In volcanic countries observations are carried out by volcanologists to provide forecasts and warnings. One observation target is crater lakes. Flash floods and hydrovolcanic explosions can be caused by such lakes. Observing these can help providing information, such as when precautions must be taken and a potential evacuation planned. [5]

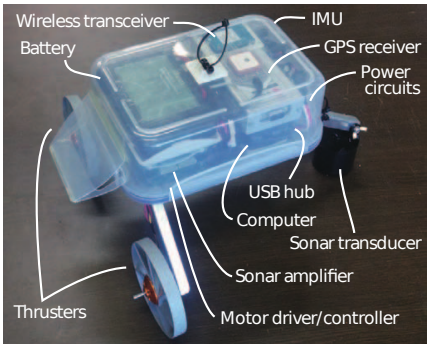


Figure 1.2: Small ASV for taking bathymetric measurements in the Mt. Zao Okama Crater Lake in Japan. [5]

The small ASV seen in Figure 1.2 is designed to take bathymetric measurements of such lakes and eliminate the need to endanger humans in the process. [5]

The ASV is made specifically for the Mt. Zao Okama Crater Lake in Japan, where it replaces the need for a manned canoe to enter the volcanic lake situated in high altitude, strong wind, and restricted area. The bathymetric measurements are used to indicate the amount of water, crater wall caving, and volcanic upthrusts in the lake. [5]

An example of a system designed to perform such a task is described in [6]. This system is able to detect and avoid obstacles in the water, while performing precision measurements of water quality and greenhouse gases. The vessel uses solar cells, which gives the added challenge of controlling the vessel at low speeds, to prevent battery usage during operation for longevity. This allows the vessel to autonomously survey relatively large areas without needing to recharge.

Regarding the control perspective, the problem of controlling the vessel autonomously has been studied in the recent years. The researches published provide different solutions to this problem. In [7], an Sliding Mode Control is used for controlling a catamaran. Yang Yang *et al.* [8] desing a disturbance observer and a feedback control law based on a vectorial backstepping approach in order to obtain a robust vessel behavior. Reyhanoglu and Bommer, [9], also base their control desing on a backstepping technique and they include a switched control law to achieve global tracking. In [10], the vessel model is obtained with system identification techniques and a comparison of different control theories is presented. The controllers compared are a PID controller with gain scheduling, a Model Reference Adaptive Controller and an L_1 adaptive controller.

In this project, the design and implementation of a control design with focus on the use of an ASV to perform bathymetric measurement is described.

2 | Problem Analysis

The goal of this project is to develop a control strategy that can make an ASV suitable for survey tasks in water. More specifically, it should be able to perform bathymetric measurements.

In order to set up the requirements for the control system it is helpful to study a use case in which bathymetric measurements are already in use and where improvements in measurement techniques are needed.

Port of Aalborg provides such a case along with previously used bathymetric measurements, see Figure 2.1. These measurements are used by Port of Aalborg to guide ships safely through the port without grounding.

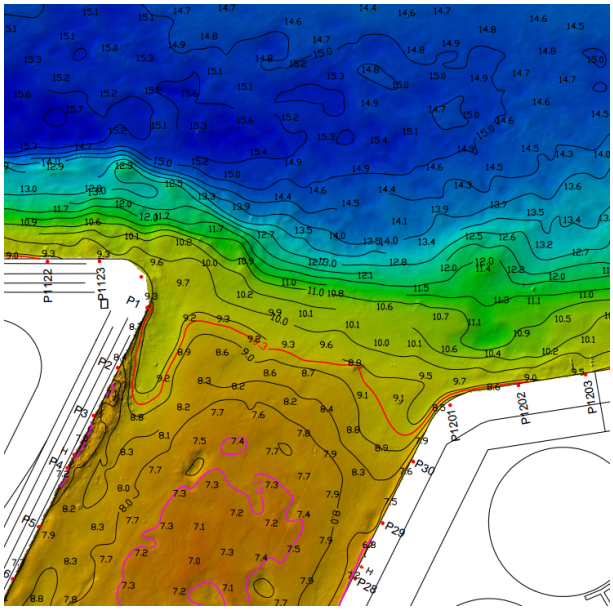


Figure 2.1: A cut from the full bathymetric map found in Appendix A provided by Port of Aalborg.

The depths of the port are in constant change due to shifting sands. So while the bathymetric measurements help in guiding the ships by the safest route they are not currently provided frequently enough that this can be done in the most optimal manner. If two ships are heading towards each other, one of them is forced to wait in places where there is sure to be enough space for both in order to allow them to pass each other. However, this is sometimes an unnecessary precaution, if there was more recent knowledge of the depths in the port.

The measurements are currently performed by a manned vessel on which a multibeam echo sounder is installed. Contrary to single beam echo sounders, the multibeam can sweep a wider area. Nevertheless, this is still a time consuming task.

It is therefore desired to automate the process, preferably with a smaller unmanned vessel. This allows more frequent bathymetric measurements and improves the efficiency of guiding ships through Port of Aalborg.

The vessel must be able to perform bathymetric measurements within an area autonomously. To do so it must be able to plan and follow a route, such that the entire area is measured. The route is dependent

on the swath angle of the sensor and the precision of the controller. In order to ensure the entire area is covered, the path planner should include some overlap of the scanned areas, as this accounts for potential inaccuracies of the system.

2.1 Design Considerations

Following the S-44 IHO standards for hydrographic mapping, the measurements fall into the *Special order* category, which sets a maximum total horizontal uncertainty (THU) of 2 m, 95% confidence level for the position measurements. This standard is intended for under keel measurements of the sea floor, which aligns with the scope of this project. [11]

The Canadian Hydrography Service (CHS) includes a stricter category, the *Exclusive order*. This category extends the *Special order* category to be more focused on shallow waters, such as harbors. This category sets the maximum THU to be 30 cm with a 95% confidence interval. It has been decided to use this standard as this is the one that best resembles the intended use case of the system. [12]

The system design assumes that the sensor used to measure the depth of the port is based on the current multibeam echosounder used by Port of Aalborg, the multibeam echosounder SeaBat 7125, which has a swath angle of 140° [13].

Based on the bathymetric map found in Appendix A, a minimum depth is considered to estimate the width of the beam when it reaches the bottom. This width is used to plan the path that the vessel needs to follow to reach all the points in a given area, see chapter 7. In Figure 2.2 a diagram of the echosounder's beam can be seen.

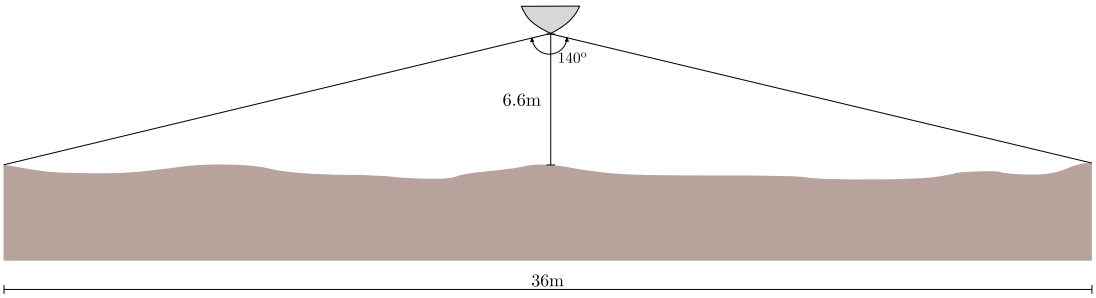


Figure 2.2: Diagram of the echosounder's beam.

As it is desired to have remote access to the vessel at all times, some form of communication between a operator and the vessel must be established. This is crucial for the implementation of safety features such as emergency stops, redirect the vessel or steer it back to land in case of system failures.

2.2 Control Analysis

For the vessel to autonomously survey an area, a control system is needed. The performance of the controller is crucial for how well the system performs overall. One of the challenges when designing a controller is how well it handles disturbances. In the case of the vessel the disturbances are mostly represented as wind, current and wave disturbances.

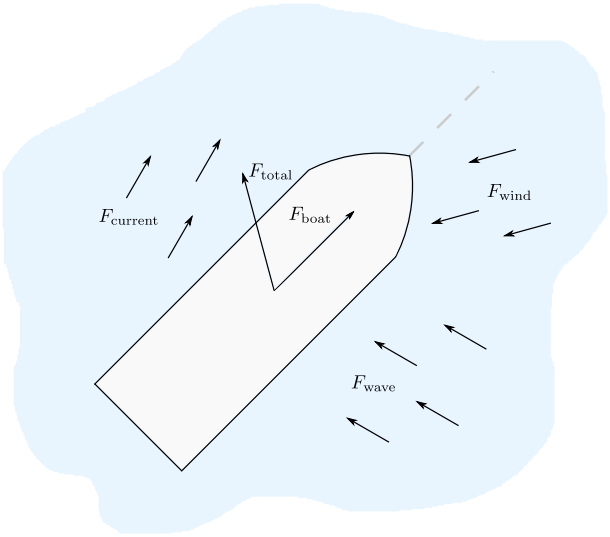


Figure 2.3: Illustration of how the vessel is affected by disturbances.

As illustrated on Figure 2.3, these external forces alter the force vector of the vessel, influencing its trajectory. This results in a loss of operating range, as the vessel has to overlap a larger area to make sure the path covers the entirety of the survey area.

Additionally the controller could experience model inaccuracies. It is not possible to model a system perfectly, as it is always an approximation. These model variations influence the controller's performance and robustness as its design is based upon them.

Another aspect of controller design is the energy consumption. The controller could be optimized such that it spends as little energy as possible reaching its destination. This feature gives the vessel a larger possible survey area as the ASV is a mobile platform and has a limited power supply.

2.3 Functional Requirements

The design of a working prototype requires to set some functional requirements, and verify them once the design has been carried out.

A: It shall be possible to select the area in which the bathymetric measurements are to be performed.

B: The ASV shall be able to plan a route, such that the entire survey area is mapped.

C: The ASV shall be able to follow the planned route.

D: The controller shall be robust to external disturbances.

E: The THU shall not exceed 30 cm with a 95% confidence interval.

F: The ASV shall record and store data locally for extraction at the end of the survey.

G: It shall be possible to give the ASV a command to stop and steer it back to land.

3 | System Description

A surface vessel is provided for the project, [14]. As seen in Figure 3.1 the vessel is equipped with actuators, sensors and control electronics.

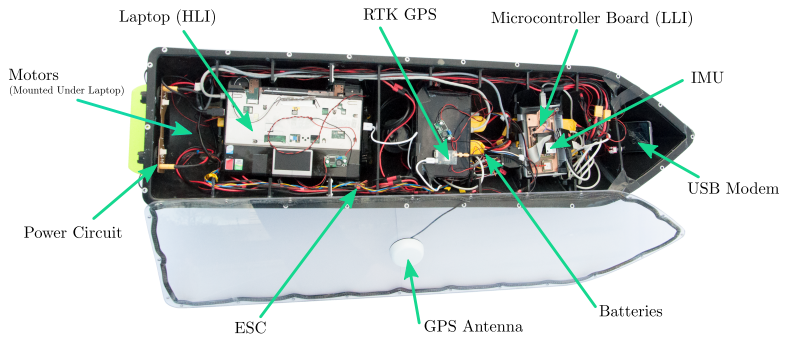


Figure 3.1: System picture. The arrows point to the components used in the project.

The surface vessel at hand is a complex system composed by several subsystems. These are shown in Figure 3.2, in which the link between the different subsystems is also depicted.

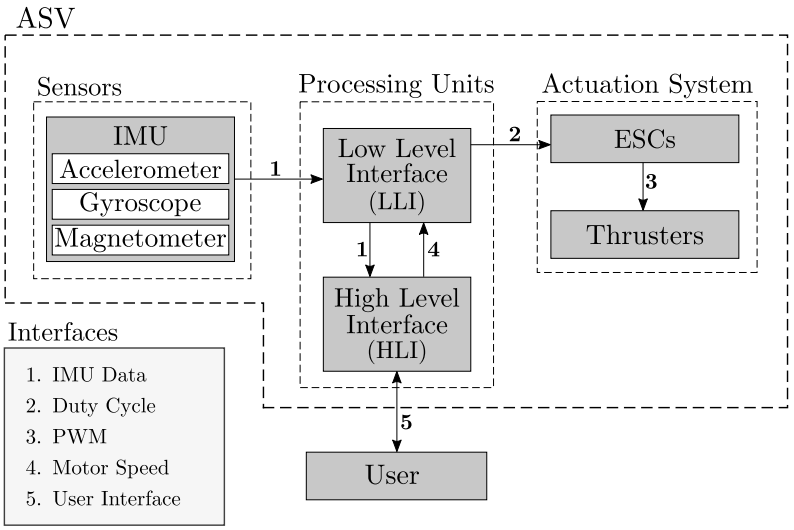


Figure 3.2: Functional diagram of the given system.

The main parts of the system are the actuation system, the sensors and the processing units, see Figure 3.2. The inertial measurement unit (IMU) information is gathered by the low level interface (LLI), and then transferred to the high level interface (HLI). In the HLI, the control algorithms are implemented, and the calculated commands are sent back to the actuation system through the LLI. The electronic speed controllers (ESCs) then calculate the required signal to make the motors turn at the requested speed.

This chapter briefly describes the main components of the surface vessel used in this project. Some additions to the existing systems are also presented.

3.1 Processing Units

The processing units are structured in two entities that run in different devices. These are the LLI and the HLI.

3.1.1 Low Level Interface

The LLI implemented on the vessel, seen in Figure 3.1, runs on an Arduino Mega Development board with an Atmel microcontroller ATMEGA 2560 [15]. It is in charge of extracting the sensor data from the IMU and send it to the HLI that runs on the computer. This includes managing the serial communication from the sensors to the Arduino Board and from the Arduino Board to the HLI.

The LLI also handles the actuators as it receives the command for the thrusters from the HLI and transmits the appropriate PWM so the ESCs ensure that the motors turn at the desired speed.

3.1.2 High Level Interface

The HLI takes care of the control, sensor data processing and path planning algorithms. This includes communicating with the LLI to get sensor data and send commands to the thrusters.

The HLI is implemented in an ASUS Eee Computer [16], seen in Figure 3.1, that runs Ubuntu 14.04 [17]. In order to implement the aforementioned algorithms, the Robotic Operating System (ROS) is used. ROS allows programming the different tasks without considering the transmission of data between threads, as this is managed by ROS, using a structure based on nodes an topics [18].

3.2 Actuation System

The actuation system present on the vessel is constituted by the thrusters and the ESCs.

3.2.1 Thrusters

The thrusters are the main actuators present on the vessel and they provide a forward force depending on the rotational speed of the motors.

The relationship between the command and the force that they exert have been obtained through an experimental test described in Appendix F. It is as follows

$$\text{PWM} = 6.6044\ F + 70.0168 \quad . \tag{3.1}$$

If a negative force is required, they can also rotate in the opposite direction, producing a backwards force that is calculated as

$$\text{PWM} = 8.5706\ F - 91.9358 \quad . \tag{3.2}$$

The thrusters are actuated with brushless motors INLINE 750 14.8 V from Graupner. They have two poles with a velocity constant of 1035 rpm·kV^{-1} and can operate within 7.4 and 22.2 V, being the nominal voltage 14.8 V. [19]

Figure 3.3: INLINE 750 14.8 V motor used to produce the thrust in the surface vessel [19].

3.2.2 Electronic Speed Controllers

In order to have the motors turning to the desired rotational speed, electronic speed controllers (ESCs) +70 G3.5 from Graupner are used. The supply voltage ranges from 6 to 25 V and they can handle up to 70 A in continuous current. The reference PWM that comes from the microcontroller translates into a 32 kHz PWM signal to the motors. [20]

Figure 3.4: Speed controllers +70 G3.5 used to control the thrusters in the surface vessel [20].

3.3 Sensors

The control system designed in the vessel requires the presence of sensor data that provides information about the vessel's motion. This is handled by an IMU.

The IMU installed in the vessel is formed by a triaxial gyroscope with a digital range scaling between $\pm 300° \cdot s^{-1}$, a triaxial accelerometer with a range of ± 18 g and a triaxial magnetometer with a range of ± 2.5 G. It also contains a serial peripheral interface (SPI) to obtain the data. [21]

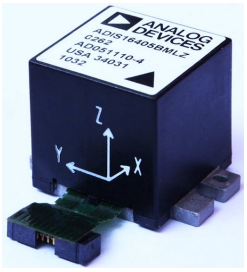


Figure 3.5: ADIS16405BMLZ IMU module mounted in the vessel [22]

The data provided by the IMU is used to estimate both the position and the attitude of the vessel.

3.4 System Additions

Some changes, including a real time kinematic (RTK) global positioning system (GPS) and communication setup, has been added to the system. A full diagram can be seen in Figure 3.6.

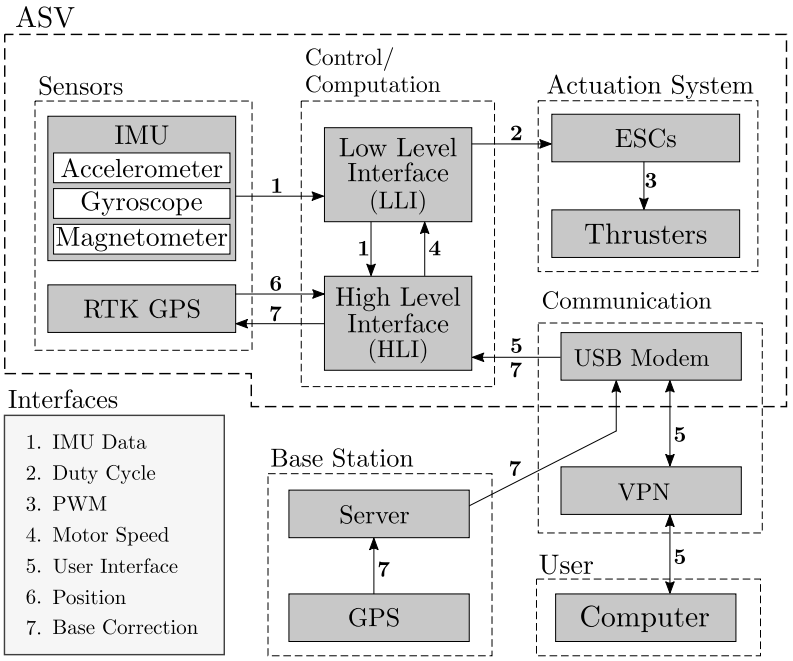


Figure 3.6: A functional diagram of the full system with additions.

The RTK GPS system, described in further detail below, is added to obtain better positioning of the ASV. This feature is necessary to accurately map the seabed of Port of Aalborg with the standards mentioned in section 2.1.

Additionally a virtual private network (VPN) server and a USB modem is used to provide user input when the ASV and user are not on the same closed network. This makes it possible to access the ASV through the cellular network and thus eliminates potential problems with regards to range between user and ASV.

3.4.1 RTK GPS

An RTK GPS system consists of two parts, a base station, and a mobile unit called rover.

The base station is set up at a stationary location with a known, precise GPS position. The rover GPS, mounted on the ASV, measures its location, based on GPS satellites and measurements from the base station. [23]

The modules used in this project, both for the base and the rover, are Emlid Reach RTK GPS, see Figure 3.7.

Figure 3.7: Emlid Reach RTK GPS module.[24]

An RTK GPS is able to achieve a higher precision than an ordinary GPS by receiving correction data from a base station, improving from 2-5 m down to a theoretical precision of a few centimeters. [23]

Figure 3.8 shows how the RTK GPS system is set up.

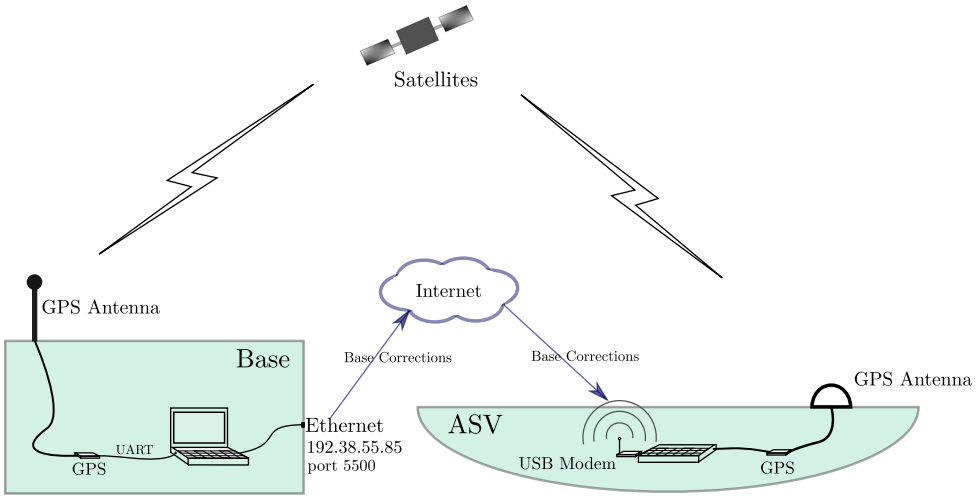


Figure 3.8: Overall set up of the GPS.

The base computer forwards the message to a TCP socket, making it accessible through the Internet. The HLI on the vessel connects to this socket and feeds the data to the rover located on the vessel. For a more detailed description of an RTK GPS and the setup see Appendix C.

4 | System Model

The model of the surface vessel is based in the methods presented in [25], where the main dynamic effects regarding the behavior of the vessel are taken into account and described in order to generate a model that serves as a basis for control design and simulations.

4.1 Reference Frames

The attitude and position of the vessel is described using two coordinate frames, a body frame and an inertial frame. For operations in a local area, with longitude and latitude approximately constants (flat navigation) a North-East-Down (NED) system can be assumed as an inertial frame where Newtonian mechanics apply [25, p. 17].

To distinguish between the two frames, the body frame is denoted with a subindex "$_{\mathrm{b}}$", and the inertial frame with subindex "$_{\mathrm{n}}$". In Figure 4.1, a diagram of the surface vessel with the notation used can be seen.

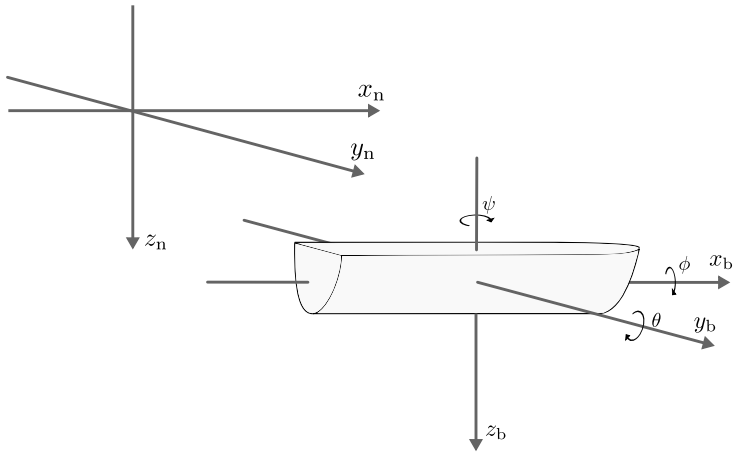


Figure 4.1: x_{b}, y_{b} and z_{b} refer to the position with respect to the body coordinate frame, while x_{n}, y_{n} and z_{n} describe it with respect to the inertial frame. ϕ, θ and ψ refer to the rotation around x_{b}, y_{b} and z_{b}, respectively.

The transformation from the body frame to the inertial can be done through a rotation matrix, (4.1), which describes a total rotation in terms of three consecutive rotations. Note that due to the size of the matrices sine and cosine are denoted s and c respectively.

In this case the rotation matrix is composed with a 1-2-3 convention, that is, first a rotation around x_{b}, then around y_{b} and finally around z_{b} [25, p. 22].

$$\mathbf{R}_{\mathrm{X}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \qquad \mathbf{R}_{\mathrm{Y}} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \qquad \mathbf{R}_{\mathrm{Z}} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_{\mathrm{b}}^{\mathrm{n}} = \mathbf{R}_Z \mathbf{R}_Y \mathbf{R}_X = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}. \tag{4.1}$$

Where:

$\mathbf{R}_{\mathrm{X}}$	is the matrix describing a rotation around the x_{b} axis
$\mathbf{R}_{\mathrm{Y}}$	is the matrix describing a rotation around the y_{b} axis
$\mathbf{R}_{\mathrm{Z}}$	is the matrix describing a rotation around the z_{b} axis
$\mathbf{R}_{\mathrm{b}}^{\mathrm{n}}$	is the total rotation matrix

To describe a vector in the inertial frame given its description in the body frame, it is left multiplied by the rotation matrix as

$$v_{\mathrm{n}} = \mathbf{R}_{\mathrm{b}}^{\mathrm{n}} v_{\mathrm{b}} \ . \tag{4.2}$$

Where:

v_{n}	is a column vector that contains the description with respect to the inertial frame
v_{b}	is a column vector that contains the description with respect to the body frame

If the inverse computation is needed, it is done following the same procedure using $\mathbf{R}_{\mathrm{b}}^{\mathrm{n}\ \mathrm{T}}$ as the rotation matrix.

4.1.1 Rigid Body Dynamics

The first step to model the motion of the surface vessel is to look at its rigid body dynamics. They are described assuming that the center of gravity of the vessel coincides with the origin of the body coordinate frame.

The translational movement can be analyzed using Newton's second law, where the acceleration of the vessel is related to the applied forces as

$$\sum F = m\ddot{x} \ . \tag{4.3}$$

For rotational movements, the motion is described using the Newton's second law applied to rotational movement, where the torques applied to the system influence the angular acceleration around each axis as

$$\sum \tau = I\ddot{\theta} \ . \tag{4.4}$$

The rotational movement is affected by the Coriolis effect, which appears if the vessel is not rotating around the axis with least or highest inertial axis. However, the influence of this force is small if the vessel rotates at low speeds, hence it has been neglected in the model. [25, p. 170]

4.2 Hydrostatics

The hydrostatics describe what forces and torques are applied on the surface vessel by the volume of fluid displaced when floating on water. The force induced upon the vessel is called buoyancy force and it is applied to the center of buoyancy.

The buoyancy force acts in the negative z_{n} direction as seen in

$$B = \rho g(V + \Delta V(z)) \ . \tag{4.5}$$

Where:

ρ	is the density of the fluid in which the vessel floats	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$
g	is the gravitational acceleration	$[\mathrm{m} \cdot \mathrm{s}^{-2}]$
V	is the volume of fluid displaced by the surface vessel	$[\mathrm{m}^3]$
ΔV	is the change in volume of fluid displaced by the surface vessel	$[\mathrm{m}^3]$
B	is the buoyancy force	[N]

When the vessel floats, the gravity force cancels out $\rho g V$ of the buoyancy force, making the contribution of the latest along x_{b}, y_{b} and z_{b} directions dependent only on the variation with respect to the equilibrium flotation point. This result is seen in

$$F_{z_{\mathrm{n}}} = mg - \rho g V - \rho g \Delta V(z) = -\rho g \Delta V(z) \ . \tag{4.6}$$

Where:

$F_{z_{\mathrm{n}}}$	is the summation of forces along the z_{n} direction	[N]

The change in volume can be expressed as in Equation 4.7. The water plane of the vessel is not considered to vary significantly with change in vertical position, thus the approximation seen in the following equation is applied

$$\Delta V(z) = \int_0^{z_{\mathrm{N}}} A_{\mathrm{wp}}(\zeta) d\zeta \approx A_{\mathrm{wp}} z_{\mathrm{n}} \ . \tag{4.7}$$

Where:

A_{wp}	is the water plane area of the vessel	$[\mathrm{m}^2]$

The contribution along the body frame directions is calculated as a function of the ϕ and θ angles in

$$F_{x_{\mathrm{b}}} = -\rho g A_{\mathrm{wp}} z_{\mathrm{n}}(-\sin\theta) \ , \tag{4.8}$$
$$F_{y_{\mathrm{b}}} = -\rho g A_{\mathrm{wp}} z_{\mathrm{n}}(\cos\theta \sin\phi) \ , \tag{4.9}$$
$$F_{z_{\mathrm{b}}} = -\rho g A_{\mathrm{wp}} z_{\mathrm{n}}(\cos\theta \cos\phi) \ . \tag{4.10}$$

The buoyancy force also contributes with some torques around the different axis in the body coordinate frame. This occurs as the center of buoyancy in general is not aligned with the center of gravity, generating

some restoring torques on the vessel. These are dependent on the gravity and the buoyancy force. As the contribution of the term $\rho g \Delta V$ is small compared to that of $\rho g V$, only the latter is considered in the model [25, pp. 62-67]. These torques are expressed as

$$T_\phi = -\rho g V \overline{GM}_\mathrm{T} \sin\phi(\cos\theta\cos\phi) \ , \tag{4.11}$$

$$T_\theta = -\rho g V \overline{GM}_\mathrm{L} \sin\theta(\cos\theta\cos\phi) \ . \tag{4.12}$$

Where:

T_ϕ	is the restoring torque due to the buoyancy force in the ϕ direction	$[\mathrm{N}\cdot\mathrm{m}]$
T_θ	is the restoring torque due to the buoyancy force in the θ direction	$[\mathrm{N}\cdot\mathrm{m}]$
$\overline{GM}_\mathrm{T}$	is the transverse metacentric height	[m]
$\overline{GM}_\mathrm{L}$	is the longitudinal metacentric height	[m]

The metacentric heights are the distances between the center of gravity and the metacenter of the vessel. The metacenter position is located in the intersection of imaginary vertical lines that go through the different centers of buoyancy originated when tilting or displacing the vessel. This crossing point, together with the metacentric heights can be seen in Figure 4.2.

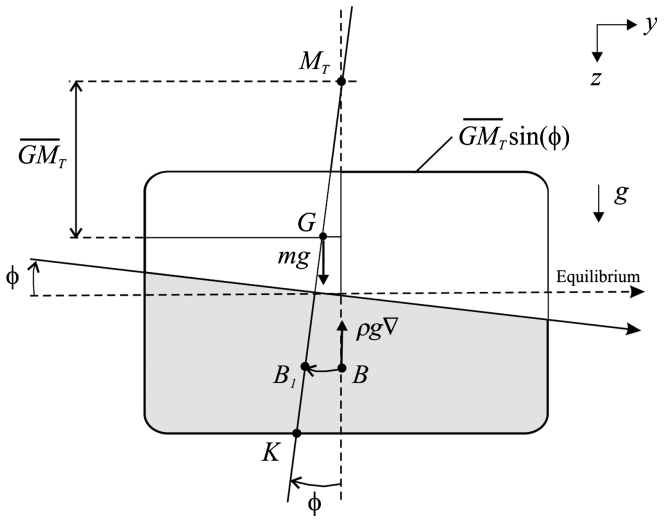


Figure 4.2: Representation of the metacentric heigths and the metacenter location in a surface vessel [25, p. 62].

4.3 Hydrodynamics

The hydrodynamic forces induced in the surface vessel are mainly caused by two terms. The added mass and the viscous damping.

The added mass induces a force that originate from the vessel imposing some energy in the surrounding fluid when the vessel moves through it, which is dependent on the acceleration of the vessel. However, this effect is neglected since most of the effective mass of the vessel comes from the nominal mass, and any variation is handled by the controller.

The viscous damping is a combination of several factors, namely, skin friction, wave drift damping and vortex shedding [25, p. 122]. This type of damping appears in the equations as coefficients that multiply, with negative sign, the different translational and angular velocities that define the movement of the vessel. For each degree of freedom it is expressed as

$$D_{\dot{x}_\mathrm{b}} = -d_{\dot{x}_\mathrm{b}}\dot{x}_\mathrm{b} \ , \tag{4.13}$$
$$D_{\dot{y}_\mathrm{b}} = -d_{\dot{y}_\mathrm{b}}\dot{y}_\mathrm{b} \ , \tag{4.14}$$
$$D_{\dot{z}_\mathrm{b}} = -d_{\dot{z}_\mathrm{b}}\dot{z}_\mathrm{b} \ , \tag{4.15}$$
$$D_{\dot{\phi}} = -d_{\dot{\phi}}\dot{\phi} \ , \tag{4.16}$$
$$D_{\dot{\theta}} = -d_{\dot{\theta}}\dot{\theta} \ , \tag{4.17}$$
$$D_{\dot{\psi}} = -d_{\dot{\psi}}\dot{\psi} \ . \tag{4.18}$$
$$\tag{4.19}$$

Where:

D_i	is the damping force or torque due to viscous damping	$[\mathrm{N},\ \mathrm{N}\cdot\mathrm{m}]$
d_i	is the viscous damping coefficient	$[\mathrm{N}\cdot\mathrm{m}^{-1}\cdot\mathrm{s},\ \mathrm{N}\cdot\mathrm{m}\cdot\mathrm{rad}^{-1}\cdot\mathrm{s}]$

These equations consider that the viscous friction is linear, since this assumption can be done for vessel speeds lower than 2 m·s [25, p. 138].

4.4 Model Equations

The final model equations are presented in this section, and are described relative to the body frame, meaning that all movement is relative to the vessel. Figure 4.3 and 4.4 show a diagram of the vessel.

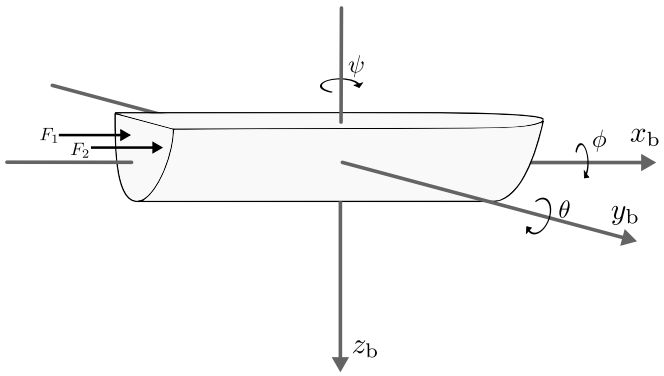


Figure 4.3: Diagram of the vessel, where the forces applied by the thrusters are shown.

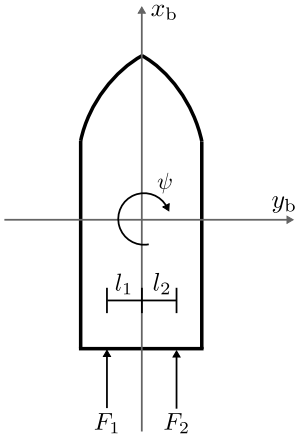


Figure 4.4: Top view of the vessel, where the distances needed for the model equations are also presented.

The translational movement of the vessel is described by Equation 4.20, 4.21 and 4.22. The model includes the forces applied by the thrusters and the damping, that create an acceleration in the vessel as

$$m\ddot{x}_\mathrm{b} = F_1 + F_2 - d_{\dot{x}_\mathrm{b}}\dot{x}_\mathrm{b} + F_{x_\mathrm{b}} \ , \tag{4.20}$$
$$m\ddot{y}_\mathrm{b} = -d_{\dot{y}_\mathrm{b}}\dot{y}_\mathrm{b} + F_{y_\mathrm{b}} \ , \tag{4.21}$$
$$m\ddot{z}_\mathrm{b} = -d_{\dot{z}_\mathrm{b}}\dot{z}_\mathrm{b} + F_{z_\mathrm{b}} \ . \tag{4.22}$$

Where:

m	is the mass of the vessel	[kg]
$\ddot{x}_\mathrm{b}$	is the acceleration in the x_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-2}]$
$\ddot{y}_\mathrm{b}$	is the acceleration in the y_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-2}]$
$\ddot{z}_\mathrm{b}$	is the acceleration in the z_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-2}]$
$\dot{x}_\mathrm{b}$	is the velocity in the x_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-1}]$
$\dot{y}_\mathrm{b}$	is the velocity in the y_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-1}]$
$\dot{z}_\mathrm{b}$	is the velocity in the z_b direction	$[\mathrm{m} \cdot \mathrm{s}^{-1}]$
$F_{1,2}$	are the forces applied by each thuster	[N]

The rotational movement of the vessel is described by Equation 4.23, 4.24 and 4.25.

$$I_\mathrm{x}\ddot{\phi} = -d_{\dot{\phi}}\dot{\phi} + T_\phi \ , \tag{4.23}$$

$$I_\mathrm{y}\ddot{\theta} = -d_{\dot{\theta}}\dot{\theta} + T_\theta \ , \tag{4.24}$$

$$I_\mathrm{z}\ddot{\psi} = F_1 l_1 - F_2 l_2 - d_{\dot{\psi}}\dot{\psi} \ . \tag{4.25}$$

Where:

I_x	is the inertia around the x_b axis	$[\mathrm{kg} \cdot \mathrm{m}^2]$
I_y	is the inertia around the y_b axis	$[\mathrm{kg} \cdot \mathrm{m}^2]$
I_z	is the inertia around the z_b axis	$[\mathrm{kg} \cdot \mathrm{m}^2]$
$\ddot{\phi}$	is the angular acceleration around the x_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-2}]$
$\ddot{\theta}$	is the angular acceleration around the y_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-2}]$
$\ddot{\psi}$	is the angular acceleration around the z_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-2}]$
$\dot{\phi}$	is the angular velocity around the x_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-1}]$
$\dot{\theta}$	is the angular velocity around the y_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-1}]$
$\dot{\psi}$	is the angular velocity around the z_b axis	$[\mathrm{rad} \cdot \mathrm{s}^{-1}]$
l_1	is the perpendicular distance from thruster 1 to the center of gravity	[m]
l_2	is the perpendicular distance from thruster 2 to the center of gravity	[m]

Similar to the translational equations, only one axis is controllable. This is, however, not a problem in practice, since the vessel is stable by nature and it can be controlled even being an underactuated vehicle [25, pp. 235-239].

4.5 Linearization of Model Equations

The model equations need to be linearized to design a controller using linear techniques. This is done using the first order Taylor approximation around an equilibrium point as seen in

$$f(x) \approx f(\overline{x}) + f'(\overline{x})(x - \overline{x}) \rightarrow \ \tilde{f}(x) \approx f'(\overline{x})\tilde{x} \ . \tag{4.26}$$

In this equation, $\overline{x}$ represents the equilibrium point and $\tilde{x}$ the change from the equilibrium point.

The equilibrium point must fulfill that all the derivatives of the states are zero, in this case the velocities and accelerations. This implies that the resulting forces and torques must be zero.

To apply the approximation, the function must be differentiated with respect to each of the present variables, and once linearized, the function is expressed in terms of variations from the equilibrium point.

$$f = f(x_1, x_2, ..., x_\mathrm{n}) \ ,$$
$$\tilde{f} = \left.\frac{\partial f}{\partial x_1}\right|_{\overline{x}_1,\overline{x}_2,...,\overline{x}_n} \tilde{x}_1 + \left.\frac{\partial f}{\partial x_2}\right|_{\overline{x}_1,\overline{x}_2,...,\overline{x}_n} \tilde{x}_2 + ... + \left.\frac{\partial f}{\partial x_n}\right|_{\overline{x}_1,\overline{x}_2,...,\overline{x_n}} \tilde{x}_\mathrm{n} \ .$$

In the case of the vessel, the only nonlinear terms are the restoring forces and torques. They can be linearized, giving the following results

$$\tilde{F}_{x_\mathrm{b}} = 0 \ , \tag{4.27}$$
$$\tilde{F}_{y_\mathrm{b}} = 0 \ , \tag{4.28}$$
$$\tilde{F}_{z_\mathrm{b}} = -\rho g A_\mathrm{wp} \tilde{z}_\mathrm{n} \ , \tag{4.29}$$
$$\tilde{T}_\phi = -\rho g V \overline{GM_T} \cdot \tilde{\phi} \ , \tag{4.30}$$
$$\tilde{T}_\theta = -\rho g V \overline{GM_L} \cdot \tilde{\theta} \ . \tag{4.31}$$

From now on, the linearized variables are represented without the symbol "$\sim$", to avoid excessive notation, even though they refer to changes with respect to the the equilibrium point.

The model equations including these linearized terms end up being

$$m\ddot{x}_\mathrm{b} = F_1 + F_2 - d_{\dot{x}_\mathrm{b}} \dot{x}_\mathrm{b} \ , \tag{4.32}$$
$$m\ddot{y}_\mathrm{b} = -d_{\dot{y}_\mathrm{b}} \dot{y}_\mathrm{b} \ , \tag{4.33}$$
$$m\ddot{z}_\mathrm{b} = -d_{\dot{z}_\mathrm{b}} \dot{z}_\mathrm{b} - \rho g A_\mathrm{wp} \tilde{z}_\mathrm{n} \ , \tag{4.34}$$
$$I_\mathrm{x}\ddot{\phi} = -d_{\dot{\phi}} \dot{\phi} - \rho g V \overline{GM_T} \cdot \phi \ , \tag{4.35}$$
$$I_\mathrm{y}\ddot{\theta} = -d_{\dot{\theta}} \dot{\theta} - \rho g V \overline{GM_L} \cdot \theta \ , \tag{4.36}$$
$$I_\mathrm{z}\ddot{\psi} = F_1 l_1 - F_2 l_2 - d_{\dot{\psi}} \dot{\psi} \ . \tag{4.37}$$

4.6 Model Verification

The parameters used in the final model are taken form previous work on the vessel [26].

Parameter	Value	Units
m	13	kg
$d_{\dot{x}_\mathrm{b}}$	2.86	$\mathrm{N{\cdot}m^{-1}{\cdot}s}$
$d_{\dot{y}_\mathrm{b}}$	32.5	$\mathrm{N{\cdot}m^{-1}{\cdot}s}$
I_x	0.0654	$\mathrm{kg{\cdot}m^2}$
I_y	1.0892	$\mathrm{kg{\cdot}m^2}$
I_z	1.1067	$\mathrm{kg{\cdot}m^2}$
$d_{\dot{\phi}}$	0.1094	$\mathrm{N{\cdot}m{\cdot}rad^{-1}{\cdot}s}$
$d_{\dot{\theta}}$	7.2030	$\mathrm{N{\cdot}m{\cdot}rad^{-1}{\cdot}s}$
$d_{\dot{\psi}}$	0.2228	$\mathrm{N{\cdot}m{\cdot}rad^{-1}{\cdot}s}$
l_1, l_2	0.05	m
$\rho g V \overline{GM_T}$	6.9736	N·m
$\rho g V \overline{GM_T}$	131.8316	N·m

As it can be seen, the parameters corresponding to the z_b direction are not presented, since Equation 4.34 is not used neither for the control design nor for the sensor fusion.

To verify the model a test is carried out. Two constant but different forces are applied to see the turning behavior of the real vessel, as seen in Appendix G. The data is then compared to the simulated model when applying the same inputs, and the results can be seen in Figure 4.5 and 4.6.

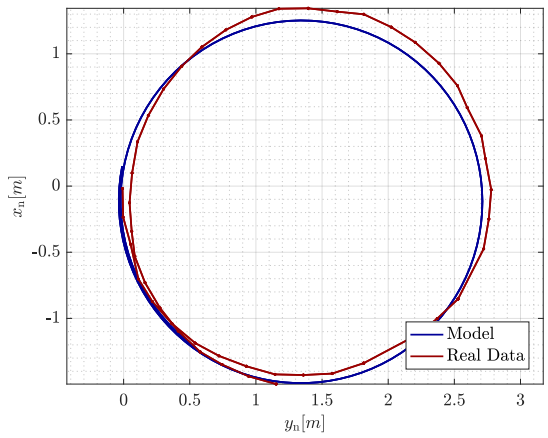


Figure 4.5: Position of the vessel in the x_n-y_n plane given by the model simulation and the GPS data.

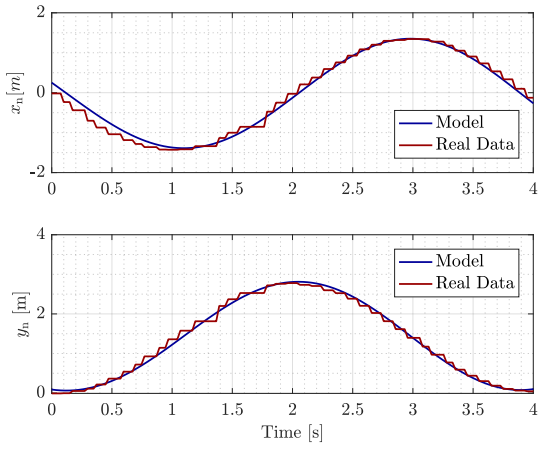


Figure 4.6: x_n and y_n with respect to time, both simulated and real.

The results show that the behavior of the real vessel is close to that of the simulated model. It is noticeable that the error between the model and the real behavior is mainly due to the quantization coming from the sampling of the GPS data. Considering this comparison, the model of the vessel is deemed sufficient both for simulation and control design purposes.

In Part I, the concept and possible applications of ASVs have been introduced with focus on the design of a control system. The possible implications in the design performing bathymetric measurements have also been considered. In order to characterize the system at hand, the vessel and its components have been described and the vessels behavior has been represented by means of a mathematical model, which constitutes the basis of the control design.

Part II

Design & Implementation

5 | Design Approach

Based on the requirements mentioned in Section 2.3 the approach shown in Figure 5.1 is designed to autonomously survey an area specified by four pairs of coordinates given in the NED frame, $\{x_\mathrm{n};\, y_\mathrm{n}\}$.

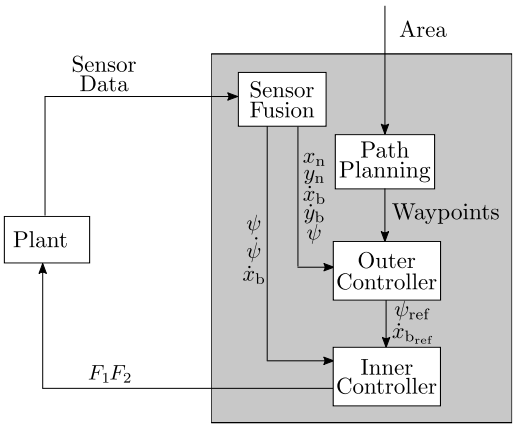


Figure 5.1: Diagram of the control approach.

This is achieved by computing a path within the specified area, based on a predefined width, to perform bathymetric measurements. The path is used as input into the controllers, making the vessel follow it.

The control design consists of two controllers, an inner controller and an outer controller. These work in the two coordinate frames, described in section 4.1.

As shown in Figure 5.1 the outer controller is in charge of following the path by changing the reference of the inner controller. Using the position of the vessel in the NED frame, x_n and y_n, as well as its velocity in the body frame, x_b and y_b, it is able to compute the ψ_ref and $\dot{x}_\mathrm{b,ref}$ needed to follow the path.

The inner controller uses these references to control $\dot{x}_\mathrm{b}$ and $\dot{\psi}$ by asking for the appropriate F_1 and F_2 commands. Two design approaches are tested for the inner controller, an H_∞ controller and a linear quadratic regulator (LQR).

The H_∞ controller is designed to be robust towards wind, current and wave disturbances and towards model errors. The LQR, on the other side, is designed to minimize a cost function that depends on the states and on the usage of inputs. The two approaches for the inner controller are then tested in simulation, to compare the robustness and performance of both of them.

Additionally, to improve the precision of the measurements, the sensor data is fused together before it is used as input for the controllers. Two Kalman filters, based on the model of the vessel, are used to filter the measurements. One estimates the attitude variables, ψ and $\dot{\psi}$, and the other the translational variables, x_n, y_n, $\dot{x}_\mathrm{b}$ and $\dot{y}_\mathrm{b}$.

In chapter 6 the inner controller design is included while the outer control is described in chapter 7. The sensor fusion is then presented in chapter 8.

6 | Inner Controller Design

The control system is designed in different control levels, as explained in chapter 5. The inner controller receives reference signals in ψ and $\dot{x}_\mathrm{b}$ and calculates the forces that the thrusters need to apply in order to track the given references.

This part of the control system is solved with an LQR and an $\mathcal{H}_\infty$ controller. The two approaches are then compared, checking their ability to reject disturbances and model uncertainties while tracking a step reference. The controller cost, which is calculated based on the state errors and usage of inputs, is also used as a comparative measure of the efficiency of the control approaches.

6.1 Linear Quadratic Regulator

The first design approach is obtained using an LQR. First, a state space model of the system is created based on the equations derived in chapter 4. Then, a cost function based on the state errors and the input usage is minimized in order to calculate the controller gains.

6.1.1 State Space Model

The linearized model derived in section 4.5, consisting of Equation 4.32 to 4.37, needs to be represented in state space form to design a state space controller. In order to do that, the three degrees of freedom model used for the control of the vessel is represented as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \ , \tag{6.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \ . \tag{6.2}$$

Where:

$\mathbf{x}$	is the state vector
$\mathbf{u}$	is the input vector
$\mathbf{y}$	is the output vector
$\mathbf{A}$	is the state matrix
$\mathbf{B}$	is the input matrix
$\mathbf{C}$	is the output matrix
$\mathbf{D}$	is the feedforward matrix

The state vector is constituted by the angle and angular velocity in yaw as well as the velocity in x_b. The system outputs are the yaw angle and velocity in x_b. The input vector is composed of the two forces applied in the body frame.

$$\mathbf{x(t)} = \begin{bmatrix} \psi \\ \dot{\psi} \\ \dot{x}_b \end{bmatrix} \ , \qquad \mathbf{y(t)} = \begin{bmatrix} \psi \\ \dot{x}_b \end{bmatrix} \ , \qquad \mathbf{u(t)} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \ . \tag{6.3}$$

The resulting **A**, **B**, **C** and **D** matrices are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d_\psi}{I_z} & 0 \\ 0 & 0 & -\frac{d_x}{m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{l_1}{I_z} & -\frac{l_2}{I_z} \\ \frac{1}{m} & \frac{1}{m} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.4}$$

while, as for most mechanical systems, the **D** matrix is zero.

6.1.2 Controller Design

The design in carried out in the discrete domain. To do so, it is necessary to discretize the system. A discrete state space model is expressed as,

$$\mathbf{x}(k+1) = \mathbf{A}_\mathrm{z}\mathbf{x}(k) + \mathbf{B}_\mathrm{z}\mathbf{u}(k) \ , \tag{6.5}$$
$$\mathbf{y}(k) = \mathbf{C}_\mathrm{z}\mathbf{x}(k) + \mathbf{D}_\mathrm{z}\mathbf{u}(k) \ , \tag{6.6}$$

where the z subindexes indicate the matrices being discrete and k is the sample index.

The model is discretized using the zero order hold method, as it does not introduce a $\mathbf{D}_\mathrm{z}$ matrix, with a sampling time, $T_s = 0.05$ s. This sampling time is suitable as it is faster than the system's dynamics, which ensures the controller is able to react faster to changes in the system. In Figure 6.1 the discrete system is shown in a block diagram.

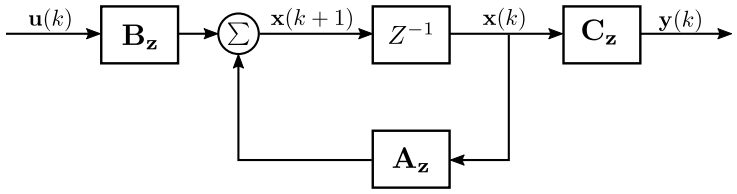


Figure 6.1: Block diagram of the discrete system.

In order to track a reference and handle input disturbances, it is chosen to also include integral action in the design. The final control structure is seen in Figure 6.2.

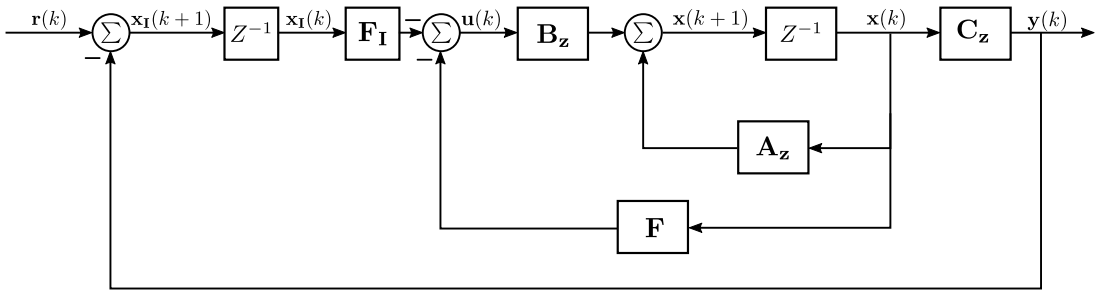


Figure 6.2: Block diagram of the control structure in the discrete domain. **F** correspond to the feedback gain matrix while $\mathbf{F}_\mathrm{I}$ is the integral gain matrix.

To design this system, it is convenient to express it on the following form

$$\mathbf{x}_\mathrm{e}(k+1) = \mathbf{A}_\mathrm{e}\mathbf{x}_\mathrm{e}(k) + \mathbf{B}_\mathrm{e}\mathbf{u}(k) + \mathbf{r}(k) \ , \tag{6.7}$$
$$\mathbf{y}(k) = \mathbf{C}_\mathrm{e}\mathbf{x}_\mathrm{e}(k) \ . \tag{6.8}$$

where the subindex "e" denotes the extended system.

To describe the control design in this form, the $\mathbf{A}_\mathrm{e}$, $\mathbf{B}_\mathrm{e}$ and $\mathbf{C}_\mathrm{e}$ matrices must be constructed. From Figure 6.2, the discrete state space model for the integral is derived and shown in Equation 6.10.

$$\mathbf{x}_\mathrm{I}(k+1) = \mathbf{x}_\mathrm{I}(k) + \mathbf{y}(k) + \mathbf{r}(k) \ , \tag{6.9}$$

$$\mathbf{x_I}(k+1) = \mathbf{x}_\mathrm{I}(k) - C_\mathrm{z}\mathbf{x_I}(k) + \mathbf{r}(k) \ . \tag{6.10}$$

where the subindex "I" stands for integral .

This leads to the discrete state space model extended with the integral states expressed as

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_\mathrm{I}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathrm{z,3x3}} & \mathbf{0}_{\mathrm{3x2}} \\ -\mathbf{C}_{\mathrm{z,2x3}} & \mathbf{I}_{\mathrm{2x2}} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_\mathrm{I}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{\mathrm{z,3x2}} \\ \mathbf{0}_{\mathrm{2x2}} \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{0}_{\mathrm{3x2}} \\ \mathbf{I}_{\mathrm{2x2}} \end{bmatrix} \mathbf{r}(k) \ , \tag{6.11}$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{C}_{\mathrm{z,2x3}} & \mathbf{0}_{\mathrm{2x2}} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_\mathrm{I}(k) \end{bmatrix} \ , \tag{6.12}$$

which corresponds to Equation 6.7 and 6.8.

A discrete time infinite horizon LQR is used in the design of the feedback, $\mathbf{F}_\mathrm{e} = [\ \mathbf{F} \quad \mathbf{F}_\mathrm{I}]$, which works by minimizing the cost function,

$$\mathcal{J}_\mathrm{z} = \sum_{k=0}^{\infty} \mathbf{x}(k)^\mathrm{T}\mathbf{Q}_\mathrm{z}\mathbf{x}(k) + \mathbf{u}(k)^\mathrm{T}\mathbf{R}_\mathrm{z}\mathbf{u}(k) \ . \tag{6.13}$$

Where:

$\mathbf{Q}_\mathrm{z}$	is the discrete time symmetric positive semidefinite state cost matrix
$\mathbf{R}_\mathrm{z}$	is the discrete time symmetric positive definite input cost matrix

The $\mathbf{Q}_\mathrm{z}$ matrix contains the penalties for the states, such that a higher cost is generated for more critical states, thus driving these states faster to zero. The $\mathbf{R}_\mathrm{z}$ matrix contains the penalties for the inputs. This helps to ensure the inputs are not driven towards saturation.

It is necessary for all states to be stable and controllable. Otherwise the performance index, $\mathcal{J}_\mathrm{z}$, becomes infinite [27, p. 125].

The controllability is determined by

$$\mathcal{C} = \begin{bmatrix} \mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^2\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^3\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^4\mathbf{B}_\mathrm{e} \end{bmatrix} \ , \tag{6.14}$$

which has full rank, thus the system is controllable [28, p. 169].

The eigenvalues of $\mathbf{A}_\mathrm{e}$ are all on or within the unit circle in the z-plane, thus, no states are unstable and the LQR design is feasible.

The design approach taken to describe the cost function, $\mathcal{J}$, is done by defining weights on the states and inputs for the continuous time infinite horizon LQR cost function,

$$\mathcal{J} = \int_0^\infty \mathbf{x}(t)^\mathrm{T}\mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^\mathrm{T}\mathbf{R}\mathbf{u}(t)\ dt \ . \tag{6.15}$$

Where:

$\mathbf{Q}$	is the continuous time symmetric positive semidefinite state cost matrix
$\mathbf{R}$	is the continuous time symmetric positive definite input cost matrix

Bryson's rule is used as an initial design method to determine sensible values for the state and input penalties of the $\mathbf{Q}$ and $\mathbf{R}$ matrices, as described in Equation 6.16

$$Q_{\mathrm{ii}} = \frac{1}{{x_{\mathrm{i}_{\max}}}^2} \ , \qquad R_{\mathrm{ii}} = \frac{1}{{u_{\mathrm{i}_{\max}}}^2} \ . \tag{6.16}$$

Where:

$x_{\mathrm{i}_{\max}}$	are the maximum acceptable state values
$u_{\mathrm{i}_{\max}}$	are the maximum acceptable input values

The requirements stated in section 2.3 must be taken into account when determining the values of $x_{\mathrm{i}_{\max}}$ and $u_{\mathrm{i}_{\max}}$. As the ASV needs a high accuracy for bathymetric measurements, the weights of $\mathbf{Q}$ are set higher than $\mathbf{R}$ to ensure priority is focused on driving the states down to zero. The individual integral states are penalized higher than the system states to set further importance on driving the integral states to zero. This ensures the reference signals are closely tracked. Higher weights for $\mathbf{R}$ also ensure the actuators are not overexerted. This means the actuators are not driven to saturation. This is also ideal for the mobile vessel as the actuators use less energy, thus the ASV is able to perform longer surveys.

These $\mathbf{Q}$ and $\mathbf{R}$ matrices must be discretized, as the state feedback design is done in the discrete time domain. This is done by using the procedure presented in [29].

From this, the state feedback is calculated as

$$\mathbf{F}_{\mathrm{e}} = -(\mathbf{B}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{B}_{\mathrm{e}} + \mathbf{R}_{\mathrm{z}})^{-1}\mathbf{B}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{A}_{\mathrm{e}} \ , \tag{6.17}$$

where $\mathbf{P}$ can be found as the solution of the infinite horizon algebraic discrete time Riccati equation [30, p. 42],

$$\mathbf{P} = \mathbf{A}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{A}_{\mathrm{e}} + \mathbf{Q}_{\mathrm{z}} - \mathbf{A}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{B}_{\mathrm{e}}(\mathbf{B}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{B}_{\mathrm{e}} + \mathbf{R}_{\mathrm{z}}^{-1}\mathbf{B}_{\mathrm{e}}^{\mathrm{T}}\mathbf{P}\mathbf{A}_{\mathrm{e}} \ . \tag{6.18}$$

Once $\mathbf{F}_{\mathrm{e}}$ is obtained, it is split into the two feedback matrices, $\mathbf{F}_{\mathrm{e}} = [\ \mathbf{F} \ \ \mathbf{F}_{\mathrm{I}}\ ]$, and implemented, following the control structure provided in Figure 6.2. The final values are

$$\mathbf{F} = \begin{bmatrix} 589.1908 & 275.9541 & 166.0624 \\ -589.1908 & -275.9541 & 166.0624 \end{bmatrix} \ , \tag{6.19}$$

$$\mathbf{F}_{\mathrm{I}} = \begin{bmatrix} -507.9954 & -309.8882 \\ 507.9954 & -309.8882 \end{bmatrix} \ . \tag{6.20}$$

This design has been simulated together with the model of the system. Its performance is also compared to the $\mathcal{H}_\infty$ controller designed in section 6.2 when disturbances, measurement noise and parameter uncertainties are present. The simulations are presented in section 6.3.

6.2 $\mathcal{H}_\infty$ Design

The model in section 4.5 has varying parameters, such as the mass or damping coefficients. The vessel may also experience external disturbances, such as wind, wave or current forces. During surveying, it is convenient for the vessel to be robust to these model variations and it must be able to sufficiently reject disturbances. Using the $\mathcal{H}_\infty$ design technique, a robust controller for the vessel can be synthesized. In this case, the design of model and controller is done simultaneously and can not be as clearly separated as for the LQR.

The $\mathcal{H}_\infty$ problem is solved by finding an internally stabilizing controller that provides a closed loop $\mathcal{H}_\infty$ norm less than some bound, γ, [31, p. 835], [32, pp. 92-93]. Such a controller is also called suboptimal $\mathcal{H}_\infty$ controller, as there might be smaller γ yielding an internally stabilizing controller.

A more detailed mathematical formulation of the $\mathcal{H}_\infty$ problem and its solution is given in [32, pp. 91-119].

The state space model from Equation 6.1 and Equation 6.2 needs to be remodeled into a state space form suitable for solving the suboptimal $\mathcal{H}_\infty$ control problem [32, pp. 95], [33, p. 64]. This form is

$$\dot{\mathbf{x}}_\infty(t) = \mathbf{A}_1\mathbf{x}_\infty(t) + \mathbf{B}_1\mathbf{w}(t) + \mathbf{B}_2\mathbf{u}(t) \ , \tag{6.21}$$

$$\mathbf{z}(t) = \mathbf{C}_1\mathbf{x}_\infty(t) + \mathbf{D}_{11}\mathbf{w}(t) + \mathbf{D}_{12}\mathbf{u}(t) \ , \tag{6.22}$$

$$\mathbf{y}_\infty(t) = \mathbf{C}_2\mathbf{x}_\infty(t) + \mathbf{D}_{21}\mathbf{w}(t) + \mathbf{D}_{22}\mathbf{u}(t) \ , \tag{6.23}$$

Where:

$\mathbf{x}_\infty$	is the state vector
$\mathbf{w}$	is the uncontrolled input vector
$\mathbf{u}$	is the controlled input vector
$\mathbf{z}$	is the performance output vector
$\mathbf{y}_\infty$	is the measured output vector
$\mathbf{A}_1$	is the state matrix
$\mathbf{B}_1$	is the uncontrolled input matrix
$\mathbf{B}_2$	is the controlled input matrix
$\mathbf{C}_1$	is the performance output matrix
$\mathbf{D}_{11}$	is the direct feedforward matrix from $\mathbf{w}$ to $\mathbf{z}$
$\mathbf{D}_{12}$	is the direct feedforward matrix from $\mathbf{u}$ to $\mathbf{z}$
$\mathbf{C}_2$	is the measured output matrix
$\mathbf{D}_{21}$	is the direct feedforward matrix from $\mathbf{w}$ to $\mathbf{y}_\infty$
$\mathbf{D}_{22}$	is the direct feedforward matrix from $\mathbf{u}$ to $\mathbf{y}_\infty$

The $\mathcal{H}_\infty$ model representation can also be seen in Figure 6.3, where all signals and matrices involved in the design process are represented.

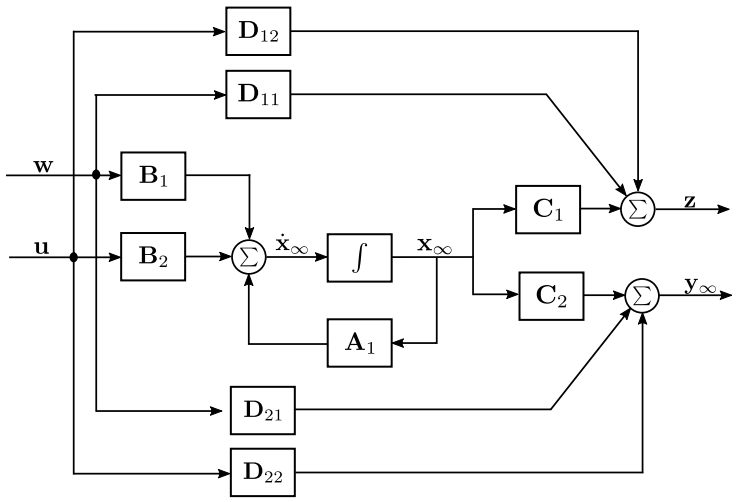


Figure 6.3: Block diagram used in the $\mathcal{H}_\infty$ controller design.

The method used to achieve the solution to the problem requires assuming certain conditions on the matrices in the model [31, p. 835]. These conditions are

1. $(\mathbf{A}_1, \mathbf{B}_1)$ and $(\mathbf{A}_1, \mathbf{B}_2)$ are stabilizable.
2. $(\mathbf{C}_1, \mathbf{A}_1)$ and $(\mathbf{C}_2, \mathbf{A}_1)$ are detectable.
3. $\mathbf{D}_{12}^{\mathrm{T}}[\mathbf{C}_1 \ \mathbf{D}_{12}]$ is $[\mathbf{0} \ \mathbf{I}]$.
4. $\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_{21} \end{bmatrix} \mathbf{D}_{21}^{\mathrm{T}}$ is $[\mathbf{0} \ \mathbf{I}]$.
5. $\mathbf{D}_{11}$ and $\mathbf{D}_{22}$ are zero.

For obtaining the matrices present in Equation 6.21, 6.22 and 6.23, the content of the state vector and signal vectors needs to be defined.

6.2.1 State Vector

The state vector construction starts with the three states that define the basic dynamics of the system. Namely, ψ, $\dot{\psi}$ and $\dot{x}_\mathrm{b}$. As some reference tracking is desired, integral states need to be included in the state vector, these depend on the measured output and the reference signal as

$$\dot{\mathbf{x}}_\mathrm{I}(t) = \begin{bmatrix} x_{\mathrm{I}_\psi} \\ x_{\mathrm{I}_{\dot{x}_\mathrm{b}}} \end{bmatrix} = \begin{bmatrix} \psi_\mathrm{ref} - \psi \\ \dot{x}_\mathrm{b,ref} - \dot{x}_\mathrm{b} \end{bmatrix} . \tag{6.24}$$

The state vector also includes the states coming from the reference, disturbance and noise models. These extra states show the dynamics of the uncontrolled inputs, to which some weighting functions are applied. This process is carried out in order to modify how the uncontrolled inputs affect the states, and this is normally done through transfer functions, [34]. In order to include weights in the state space representation, some states for each uncontrolled input need to be defined. Figure 6.4 shows an example of how an uncontrolled input is weighted so it can be included in the state space representation.

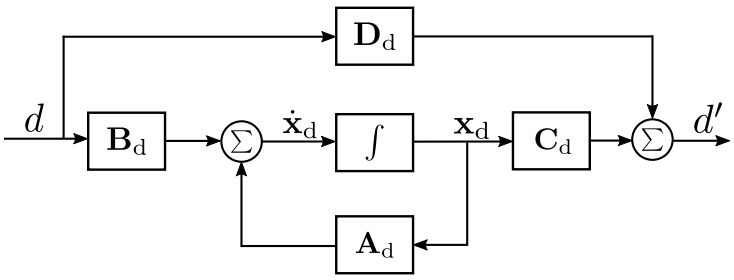


Figure 6.4: Block diagram illustrating how an uncontrolled input is weighted in the $\mathcal{H}_\infty$ controller design. d is the uncontrolled input and d' is the weighted uncontrolled input. The states $\mathbf{x}_\mathrm{d}$ are included in the state vector of the $\mathcal{H}_\infty$ state space representation.

The $\mathbf{A}_\mathrm{d}$, $\mathbf{B}_\mathrm{d}$, $\mathbf{C}_\mathrm{d}$ and $\mathbf{D}_\mathrm{d}$ in Figure 6.4 can be calculated from a weighting function as in the example given in Equation 6.25, where the uncontrolled input d is weighted by means of a first order transfer function with low pass characteristics.

$$\frac{d'}{d} = \frac{a}{s+a} \rightarrow \dot{d}' = -ad' + ad \rightarrow \begin{cases} \dot{x}_\mathrm{d} = -ax_\mathrm{d} + ad \\ d' = x_\mathrm{d} \end{cases} \tag{6.25}$$

Where:

a	is a parameter defining the pole position of the weighting function

Another example with a high pass weighting function is shown in Equation 6.26.

$$\frac{n'}{n} = \frac{s}{s+a} = \frac{-a}{s+a} - 1 \rightarrow \begin{cases} \dot{x}_\mathrm{n} = -ax_\mathrm{n} + n \\ n' = -ax_\mathrm{n} + n \end{cases} \tag{6.26}$$

This process also entails defining the weights for each uncontrolled input. The weight on the reference is set to be a first order transfer function with a very fast pole so the step input does not get distorted. The weights on the input disturbances are also low pass filtered as most of them appear at low frequencies.

The noise, on the other hand, is weighted according to a high pass filter, as it is stronger in the high frequency range. In this way, the controller design focuses on achieving a robust controller with respect to the uncontrolled inputs in their particular frequency ranges. The chosen frequencies are 1, 20 and 20 rad·s^{-1} for the references, wind, current and wave disturbances low pass filters, respectively, and 100 rad·s^{-1} for the noise high pass filter. The reference frequency is used as a tuning parameter. The disturbances frequencies are calculated to be more conservative than the value obtained in Appendix H, while the noise frequency is assumed to be present as higher frequency than 100 rad·s^{-1}. The weights for each uncontrolled input are

$$W_{\psi_\mathrm{ref}} = \frac{1}{s+1}\ , \qquad W_{\dot{x}_\mathrm{b,ref}} = \frac{1}{s+1}\ ,$$
$$W_{F_\mathrm{wc}} = \frac{20}{s+20}\ , \qquad W_{\tau_\mathrm{wc}} = \frac{20}{s+20}\ ,$$
$$W_{F_\mathrm{wave}} = \frac{20}{s+20}\ , \qquad W_{\tau_\mathrm{wave}} = \frac{20}{s+20}\ ,$$
$$W_{n_\psi} = \frac{s}{s+100}\ , \qquad W_{n_{\dot{x}_\mathrm{b}}} = \frac{s}{s+100}\ .$$

The states coming from the weighting functions, together with the system states and the integral states, constitute the state vector as

$$\mathbf{x}_\infty(t) = \begin{bmatrix} \psi & \dot{\psi} & \dot{x}_\mathrm{b} & x_{int_\psi} & x_{int_{\dot{x}_\mathrm{b}}} & x_{F_\mathrm{wc}} & x_{\tau_\mathrm{wc}} & x_{F_\mathrm{wave}} & x_{\tau_\mathrm{wave}} & x_{n_\psi} & x_{n_{\dot{x}_\mathrm{b}}} \end{bmatrix}^\mathrm{T} . \tag{6.27}$$

6.2.2 Controlled Input Vector

The controlled input are the two forces provided by the thrusters of the vessel, that is,

$$\mathbf{u}(t) = \begin{bmatrix} F_1 & F_2 \end{bmatrix}^\mathrm{T} . \tag{6.28}$$

6.2.3 Uncontrolled Input Vector

The uncontrolled inputs include the references to be tracked, the input disturbances, and the measurement noises. The size of this vector depends on the amount of reference signals, the input disturbances considered and the amount of measured outputs, as these are normally affected by noise. The references for the inner controller are two, the heading, ψ, and the translational speed along the x_b direction. The input disturbances are coming from wind, current and waves. These three elements potentially generate both a force along the x_b and a torque in ψ. The noise vector affects measured outputs considered for the inner controller, which are the outputs, ψ and $\dot{x}_\mathrm{b}$. The uncontrolled input vector is formed as

$$\mathbf{w}(t) = \begin{bmatrix} \psi_\mathrm{ref} & \dot{x}_\mathrm{b,ref} & F_\mathrm{wc} & \tau_\mathrm{wc} & F_\mathrm{wave} & \tau_\mathrm{wave} & n_\psi & n_{\dot{x}_\mathrm{b}} \end{bmatrix}^\mathrm{T} . \tag{6.29}$$

Where:

$F_\mathrm{wc/wave}$	is the force of the wind,current/waves along the x_b direction	[N]
$\tau_\mathrm{wc/wave}$	is the torque of the wind,current/waves in ψ	$[\mathrm{N}\cdot\mathrm{m}]$
n_x	is noise affecting measured output x	$[\mathrm{rad},\ \mathrm{m}\cdot\mathrm{s}^{-1}]$

6.2.4 Measurement Output Vector

The measurement output vector includes the outputs that track a reference. It also includes the integral states that represent the error between the outputs and the references and are affected by noise. Consequently, the output vector contains four elements and is represented as

$$\mathbf{y}_\infty(t) = \begin{bmatrix} \psi & \dot{x}_\mathrm{b} & \mathbf{x}_\mathrm{I}^\mathrm{T} \end{bmatrix}^\mathrm{T} . \tag{6.30}$$

6.2.5 Performance Output Vector

The performance output contains all variables whose performance should be taken into account by the controller. As the design entails a state feedback control, all the states are considered performance outputs. The controlled inputs are also part of this vector as it is desired to set some limitations or constant weights in order to account for the saturation of these inputs in the real system. This is also needed to fulfill condition 3.

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{x}_\infty^\mathrm{T} & \mathbf{u}^\mathrm{T} \end{bmatrix}^\mathrm{T} . \tag{6.31}$$

Now that the state, input and output vectors have been defined, the model matrices can be derived.

6.2.6 Model Matrices

The matrices present in Equation 6.21, 6.22 and 6.23 are derived from the relations between the different signals and states.

The design is started with Equation 6.21, which contains the $\mathbf{A}_1$, $\mathbf{B}_1$ and $\mathbf{B}_2$ matrices.

The matrix $\mathbf{A}_1$ describes the dynamics of the system states and all the other added states, which account for the references and disturbances. Its structure is

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{3x2} & \mathbf{B}_{\text{dist}} & \mathbf{B}_{\text{dist}} & \mathbf{0}_{3x2} \\ -\mathbf{C} & \mathbf{A}_{\text{I}} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x3} & \mathbf{0}_{2x2} & \mathbf{A}_{\text{wc}} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x3} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{A}_{\text{wave}} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x3} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{A}_{\text{noise}} \end{bmatrix} . \tag{6.32}$$

It can be seen that the matrix is composed by submatrices that correspond to the different parts of the system, namely for the original states in the first three rows, the integral states for reference tracking in the next two rows and the uncontrolled inputs in the last six rows.

$\mathbf{B}_1$ relates the uncontrolled inputs with the state derivatives. Its structure in terms of the different submatrices is

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}_{3x2} & \mathbf{0}_{3x2} & \mathbf{0}_{3x2} & \mathbf{0}_{3x2} \\ \mathbf{I}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & \mathbf{B}_{\text{wc}} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{B}_{\text{wave}} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{B}_{\text{noise}} \end{bmatrix} . \tag{6.33}$$

The matrix $\mathbf{B}_2$ relates the controlled inputs to the states, in this case, the thrusters only affect the system states. The $\mathbf{B}_2$ matrix is constructed as

$$\mathbf{B}_2 = \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} \end{bmatrix} . \tag{6.34}$$

Next, the matrices appearing in Equation 6.22 are derived. These are $\mathbf{C}_1$, $\mathbf{D}_{11}$ and $\mathbf{D}_{12}$ and they can be considered to be weighting matrices where the importance of each of the performance outputs and the usage of the controlled inputs can be specified.

The $\mathbf{C}_1$ matrix relates the states and the performance outputs, it is formed by a diagonal matrix, where weights are applied to each state, and some zero rows corresponding to the controlled inputs of the system. The matrix is constructed as

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{W}_{\text{x}} & \mathbf{0}_{3\text{x}2} & \mathbf{0}_{3\text{x}2} & \mathbf{0}_{3\text{x}2} & \mathbf{0}_{3\text{x}2} \\ \mathbf{0}_{2\text{x}3} & \mathbf{W}_{\text{I}} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}2} & \mathbf{W}_{\text{wc}} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{W}_{\text{wave}} & \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{W}_{\text{noise}} \\ \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} \end{bmatrix} . \tag{6.35}$$

Where the weighting submatrices values are chosen as according to the importance of each state in the controller design. The numbers are found through an iterative process starting with an initial design that weighted more the most important states in the system, which are the system states and the reference states. The design process is carried out by simulating the controller together with the model of the system. The final designed weights are

$$\mathbf{W}_{\text{x}} = diag(2,3,2)\ , \mathbf{W}_{\text{I}} = diag(5,2)\ , \mathbf{W}_{\text{wc}} = diag(1,1)\ , \tag{6.36}$$

$$\mathbf{W}_{\text{wave}} = diag(1,1)\ , \mathbf{W}_{\text{noise}} = diag(1,1)\ . \tag{6.37}$$

$\mathbf{D}_{11}$ is a zero matrix as there is no relation between the uncontrolled inputs and the performance outputs in the $\mathcal{H}_\infty$ design. It has as many rows as the number of elements in the performance output and as many columns as uncontrolled inputs.

The $\mathbf{D}_{12}$ has nonzero elements only in the entries that weight the inputs. The matrix is constructed as

$$\mathbf{D}_{12} = \begin{bmatrix} \mathbf{0}_{2\text{x}3} \\ \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}2} \\ \mathbf{0}_{2\text{x}2} \\ \mathbf{W}_{\text{u}} \end{bmatrix} . \tag{6.38}$$

The weights for the inputs are also chosen through an iterative process such that the inputs are not driven to saturation. The final designed weights for the inputs are

$$\mathbf{W}_{\text{u}} = diag(0.001, 0.001)\ . \tag{6.39}$$

Finally, the matrices present in Equation 6.23, $\mathbf{C}_2$, $\mathbf{D}_{12}$ and $\mathbf{D}_{22}$, are derived below, they are not part of the design as they describe how the measured outputs of the system are affected.

The $\mathbf{C}_2$ matrix relates the states with the measured outputs, selecting the ψ and $\dot{x}_{\text{b}}$ states and the integral states to form the output, as seen in Equation 6.40.

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{C} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{C}_{\text{noise}} \\ \mathbf{0}_{2\text{x}3} & \mathbf{I}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{C}_{\text{noise}} \end{bmatrix} . \tag{6.40}$$

The $\mathbf{D}_{21}$ matrix relates uncontrolled inputs with the measurement outputs, mainly adding the noise to the outputs as

$$\mathbf{D}_{21} = \begin{bmatrix} \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{D}_{\text{noise}} \\ \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{0}_{2\text{x}2} & \mathbf{D}_{\text{noise}} \end{bmatrix} . \tag{6.41}$$

6.2.7 Controller Gains

After setting up the model, which includes part of the controller design, the $\mathcal{H}_\infty$ controller can be found by solving a Riccatti equation defined as

$$\mathbf{X}_\infty = Ric \begin{bmatrix} \mathbf{A} & \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^{\mathrm{T}} - \mathbf{B}_2\mathbf{B}_2^{\mathrm{T}} \\ -\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1 & -\mathbf{A}^{\mathrm{T}} \end{bmatrix} , \tag{6.42}$$

and calculating the feedback gain matrix as

$$\mathbf{F}_\infty = -\mathbf{B}_2^{\mathrm{T}}\mathbf{X}_\infty . \tag{6.43}$$

The state feedback matrix $\mathbf{F}$ then corresponds to the three first columns of $\mathbf{F}_\infty$, while the integral gain $\mathbf{F}_\mathrm{I}$ corresponds to the next two columns. Similarly to the LQR design, the final diagram with the placement of the gains can be seen in Figure 6.5.

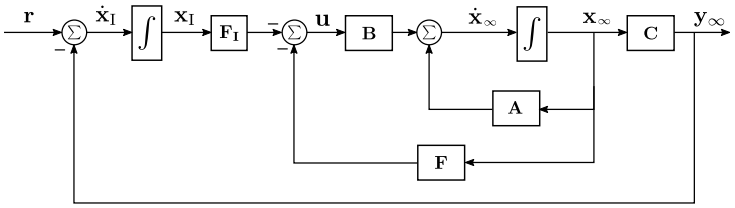


Figure 6.5: Final block diagram of the $\mathcal{H}_\infty$ controller design.

The value of γ in the Riccatti equation is also a design parameter that can be modified making the controller more or less conservative. The value chosen should be bigger than the largest singular value of the system. In this design, the minimum possible γ value is 2.7832. The final value is chosen higher than the minimum in order to have a robustness margin. The value used is 5.

The final values for the feedback gains result as follows

$$\mathbf{F} = \begin{bmatrix} 2233.8 & 1514.2 & 1001.3 \\ -2233.8 & -1514.2 & 1001.3 \end{bmatrix} , \tag{6.44}$$

$$\mathbf{F}_\mathrm{I} = \begin{bmatrix} -1150.2 & -421.8 \\ 1150.2 & -421.8 \end{bmatrix} . \tag{6.45}$$

Once the controller is designed, its performance is evaluated in simulations, in which the compensator controls the model derived in chapter 4.

6.3 Comparison between Controller Designs

The two controller designs are now compared in order to analyze the robustness and the performance. The control inputs applied by each controller are also compared. The simulations carried out in this section include input disturbances, both from wind, current and waves, and measurement noise in the outputs. The measurement noise are modeled to have a power spectral density similar to the sensor noise that comes out of the sensor fusion. Parameter variations are also included.

The performance comparison is evaluated by looking at the response of the model of the system when tracking a step in the reference inputs, $\dot{x}_{\mathrm{b,ref}}$ and ψ_{ref}.

The disturbances applied to the system range from ±1.5 N in the force along the $\dot{x}_\mathrm{b}$ axis and ±1.5 N·m in the torque around the z_b axis. These disturbances come from wind and waves. The frequency of the latter is also varied from 0 to 10 Hz.

In the simulation, the parameters are assumed to vary ±20% from their nominal value. The parameters varied in the simulation are the mass, m, the moment of inertia around the z_b axis, I_z, the damping coefficients, d_x and d_ψ, and the lengths where the forces are applied, l_1 and l_2.

Figure 6.6 and 6.7 show the step response of the velocity along the x_b direction from 1000 simulations where the model parameters and the disturbances were varied randomly within the defined ranges. These simulations also include a reference step in ψ at 10 seconds that causes a perturbation in the $\dot{x}_\mathrm{b}$ response.

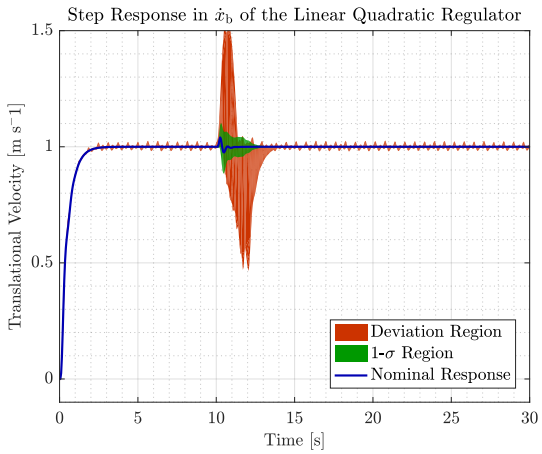


Figure 6.6: Step response in x_b of the LQR, where the nominal response, the maximum variation region and the 1σ region are shown.

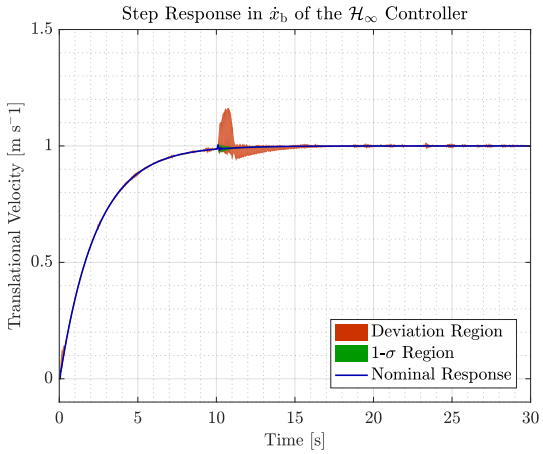


Figure 6.7: Step response in x_b of the $\mathcal{H}_\infty$ controller, where the nominal response, the maximum variation region and the 1σ region are shown.

As it can be seen, both controllers are able to track the given reference in $\dot{x}_\mathrm{b}$. The controller designed using LQR theory yields a fast controller, but the change in reference in the ψ angle leads to a great perturbation in the vessel speed, reaching 50% of deviation, while the 1σ values go up to approximately 10%. There is no steady state error once the perturbation has been compensated by the controller.

The $\mathcal{H}_\infty$ controller on the other hand, is slower in terms of settling time. The perturbation introduced when changing ψ_ref has a 1σ value that is almost zero and a maximum value around 15%. This is acceptable as the forward velocity of the vessel does not require a fast response and it is not a critical parameter for the outer controller.

It can be seen that the LQR response has a settling time of 1 s, with an error band of 5%, while the $\mathcal{H}_\infty$ controller takes around 5 s to settle. These responses can be approximated with first order systems with time constants of 0.33 s and 1.66 s, respectively, which give bandwidths 3 rad·s^{-1} and 0.6 rad·s^{-1}.

In Figure 6.8 and 6.9, a closed look to the difference of the step responses with respect to the nominal behavior is shown.

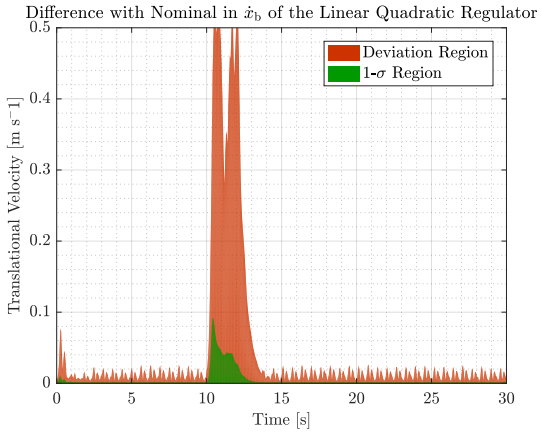


Figure 6.8: Difference with the nominal behavior in x_b of the LQR, where the maximum variation region and the 1σ region are shown.

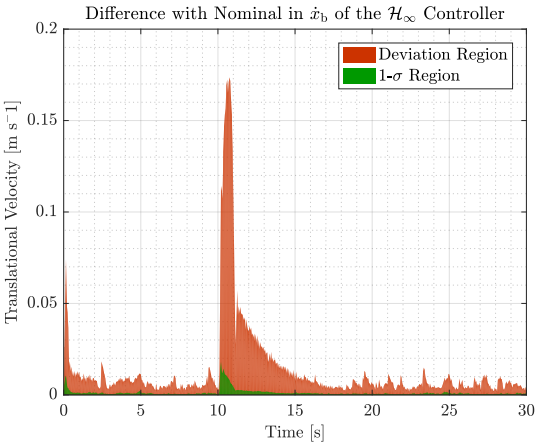


Figure 6.9: Difference with the nominal behavior in x_b of the $\mathcal{H}_\infty$ controller, where the maximum variation region and the 1σ region are shown.

It can be seen that the error keeps at a low value until the step for ψ is applied and the highest disturbance occurs, being the $\mathcal{H}_\infty$ controller the one that is able to reject better the disturbance and the one with less variation from the nominal performance.

In Figure 6.10 and 6.11, the performance of the controllers is analyzed by looking at the step response when tracking a reference in ψ. These plots are part of the same simulations depicted in Figure 6.6 and 6.7, and therefore cope with the same disturbances and parameter variations. The step is applied after 10 seconds of simulation.

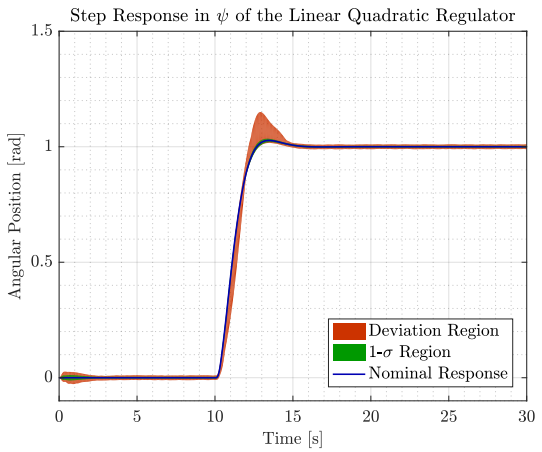


Figure 6.10: Step response in ψ of the LQR, where the nominal response, the maximum variation region and the 1σ region are shown.

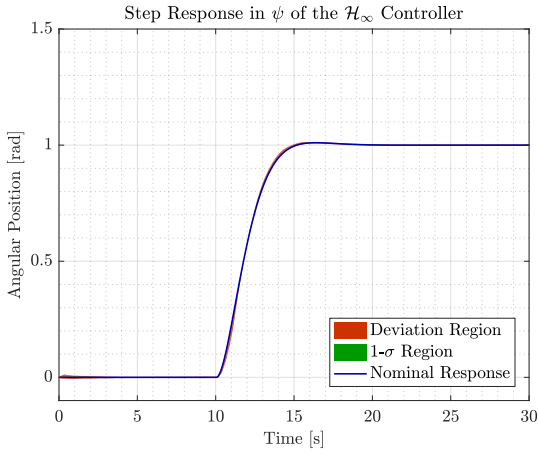


Figure 6.11: Step response in ψ of the $\mathcal{H}_\infty$ controller, where the nominal response, the maximum variation region and the 1σ region are shown.

The behavior seen in these figures resembles that for Figure 6.6 and 6.7. The $\mathcal{H}_\infty$ controller shows less overshoot and variability in the different simulations but it is also slightly slower than the LQR. In both cases, the 1σ value is very closed to zero, but the response of the LQR shows a maximum variation of 20%.

Additionally, it can be seen that the LQR response has a settling time of 2 s, while the $\mathcal{H}_\infty$ controller takes around 3.5 s to settle. These responses can be approximated with first order systems with time constants of 0.66 s and 1.16 s, respectively, which give bandwidths 1.5 rad·s^{-1} and 0.86 rad·s^{-1}.

As in the case of $\dot{x}_\mathrm{b}$, a close look at the difference with respect to the nominal response can be seen in Figure 6.12 and 6.13.

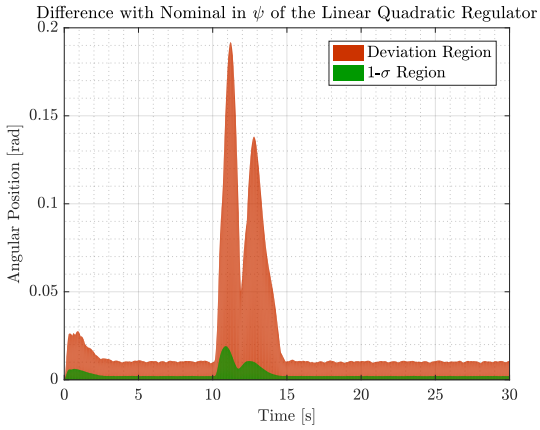


Figure 6.12: Difference with the nominal behavior in ψ of the LQR, where the maximum variation region and the 1σ region are shown.

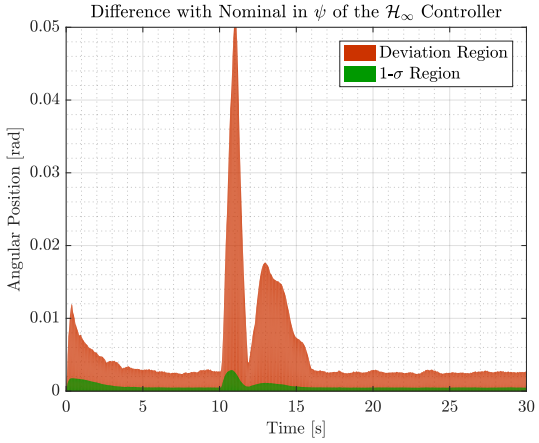


Figure 6.13: Difference with the nominal behavior in ψ of the $\mathcal{H}_\infty$ controller, where the maximum variation region and the 1σ region are shown.

In this case, it is also the $\mathcal{H}_\infty$ controller the one that is able to handle better the disturbances and with less variations with respect to the nominal behavior.

These controller designs are also compared by their usage of inputs. The mean force applied at each time by the thrusters when simulating the LQR is 17.5793 N, while that of the $\mathcal{H}_\infty$ controller is 17.9281 N. This results as expected since the LQR is designed to optimize the use of inputs to the system.

Finally, to combine performance and usage of inputs, the cost used in the LQR design is calculated using Equation 6.13. The mean cost among the simulations is $7.1441 \cdot 10^5$ and $3.4227 \cdot 10^6$ for the LQR and $\mathcal{H}_\infty$ controller designs, respectively. As expected, the cost is lower for the LQR design as minimizing it is the basis of the LQR approach.

7 | Outer Controller Design

The vessel's functionality, as stated in section 2.3, requires it to follow a path along which the bathymetric measurements are taken. The path is generated from the given area in straight lines as described in section 7.1. The generated path is followed using an enclosure based steering algorithm [25, pp. 258-265] that uses waypoints sampled along the path. The outputs for this controller are the reference for the yaw angle and the velocity along the x_b axis, which are inputs to the state space controller designed in chapter 6.

7.1 Path Generation Algorithm

The path generation algorithm creates a list of waypoints for the vessel to follow. The area that needs to be surveyed is used to generate the waypoints. The desired area is a bounded rectangle and is given as an input to the algorithm as four points, in Equation 7.1, that specify the limits of the area.

$$[x_1; y_1], \quad [x_2; y_2], \quad [x_3; y_3], \quad [x_4; y_4]. \tag{7.1}$$

Where:

$[x_k; y_k]$	is the coordinate of the corner of the bounded rectangle

The path generation algorithm is confined to generate waypoints in an area that is a rectangle. Thus, the four input coordinates need to be verified to ensure the area is a rectangle. This is done by checking

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} \ , \tag{7.2}$$

$$\sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2} = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \ , \tag{7.3}$$

$$\sin \frac{y_1 - y_2}{x_1 - x_2} - \sin \frac{y_1 - y_4}{x_1 - x_4} = \frac{\pi}{2} \ . \tag{7.4}$$

Now that the desired area is well defined, as a rectangle, the path generation algorithm is able to generate a list of waypoints for the ASV to follow.

The path is generated with straight lines and turns which the vessel shall follow. To determine the width between these lines, the dynamics of the vessel, the swath angle of the multibeam echosounder and the minimum depth of the area to survey were taken into consideration. The swath angle and minimum depth of the Port of Aalborg survey in Appendix A are used to calculate the width between straight lines, as described in section 2.1,

$$w_\mathrm{beam} = 36 \text{ m} \ . \tag{7.5}$$

Where:

w_beam	is the minimum required beamwidth	[m]

The dynamics of the vessel are considered to ensure the vessel can perform smooth turns between straight lines. It is determined that the vessel is capable of using the turning radius

$$R_\mathrm{turn} = \frac{1}{2} w_\mathrm{beam} = 18 \text{ m} \ . \tag{7.6}$$

Where:

R_turn	is the turning radius of the waypoints	[m]

The path generation algorithm first determines along which side of the rectangle to generate waypoints. This is done to ensure the vessel performs the least number of turns and is determined by calculating which side of the rectangle is larger. This is done by checking if

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \geq \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2} \ . \tag{7.7}$$

Once the sailing direction is chosen, the first waypoint is generated using the starting point's coordinates and the direction of the straight lines. The starting point in the transversal direction is offset by R_{turn} so the multibeam starts surveying from the edges of the area.

The number of waypoints on both the straight lines and on the turns are generated using

$$d_{\text{wps}} = 50 \text{ m} \ , \tag{7.8}$$
$$n_{\text{wps}} = 10 \ . \tag{7.9}$$

Where:

d_{wps}	is the distance between waypoints on the straight line	[m]
n_{wps}	is the number of waypoints on each turn	

As the area to survey may vary, the waypoints are generated every d_{wps} to ensure the vessel converges faster if it diverges from the path. Thus, the vessel does not rely only on the waypoints that define the area. The path has a fixed turning radius, R_{turn}, and therefore the number of waypoints is fixed to n_{wps}.

In Figure 7.1 an example of the desired area to survey and the generated waypoints are shown. It can be seen that the vessel is offset by R_{turn} and the minimum area the multibeam sensor surveys is also shown.

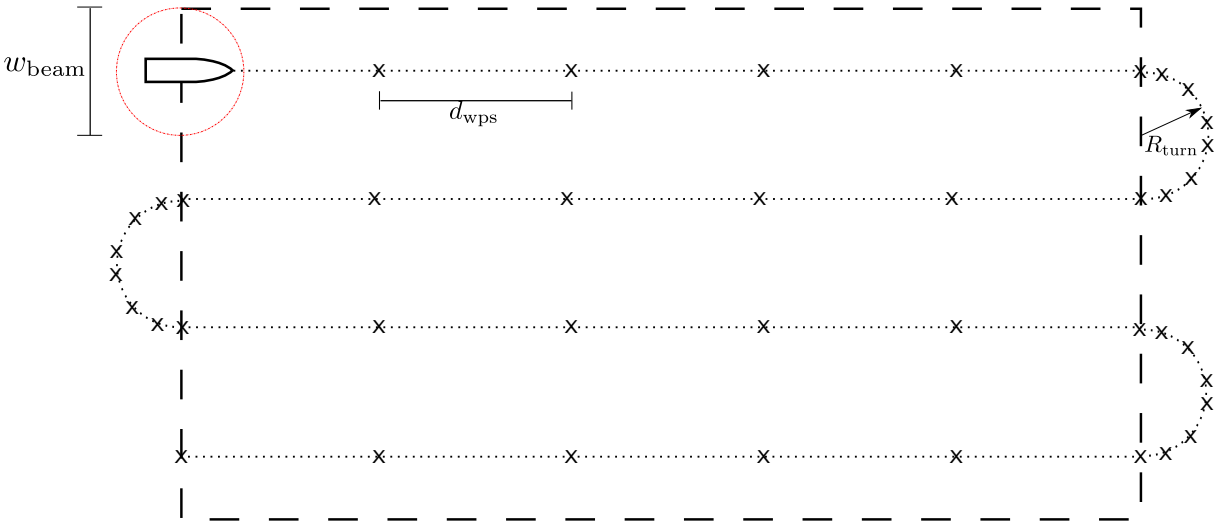


Figure 7.1: Area wanted to survey with waypoints.

7.2 Path Following Algorithm

The path generated is approximated by straight line segments connected by the calculated waypoints. The vessel then follows the segments in order to track the path and cover the area in which the measurements are to be taken. This approximation is suitable as the bathymetric measurements are usually taken in straight line paths. Straight line segments are sufficient on the curved paths as the curved paths are outside the area of interest. If curved paths were required, a solution would be to sample the path with

higher frequency in curved sections. Figure 7.2 shows an example of how a path is approximated by straight line segments and waypoints.

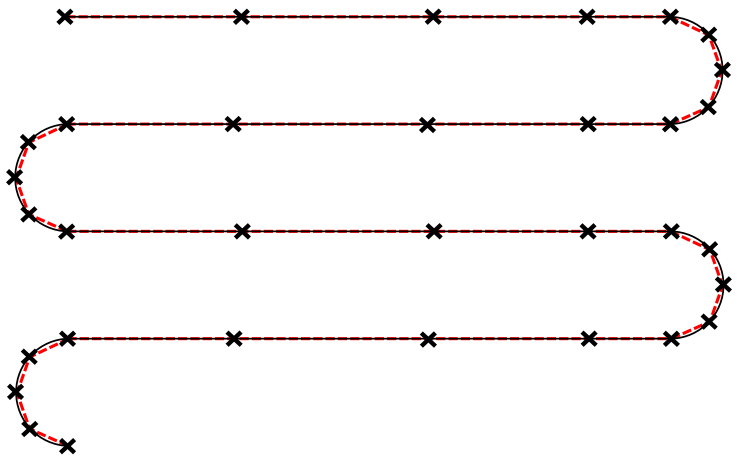


Figure 7.2: A predefined path and its approximation as straight line segments using waypoints.

The algorithm starts by considering the first two waypoints in the path. The yaw reference given to the state space controller is calculated based on the crossing point between the straight line segment that joints the waypoints and a circle centered in the position of the vessel. Figure 7.3 shows how this crossing point is obtained. There may be two intersections between the straight line and the circle, thus the chosen crossing point is the point closer to second waypoint. The crossing point is also called Line of Sight (LOS) point and is found using the equations of the circle and of the straight line as

$$(x_{\text{LOS}} - x_{\text{n}})^2 + (y_{\text{LOS}} - y_{\text{n}})^2 = R^2 \ , \tag{7.10}$$

$$y_{\text{LOS}} - y_{\text{k}} = \frac{y_{\text{k+1}} - y_{\text{k}}}{x_{\text{k+1}} - x_{\text{k}}}(x_{\text{LOS}} - x_{\text{k}}) \ . \tag{7.11}$$

Where:

R	is the radius of the circle centered at the vessel position	[m]
$[x_{\text{LOS}}, y_{\text{LOS}}]$	is the crossing point between the circle around the vessel and the straight line that joins the waypoints	[m]
$[x_{\text{k}}, y_{\text{k}}]$	is the first waypoint in the currently followed path segment	[m]
$[x_{\text{k+1}}, y_{\text{k+1}}]$	is the second waypoint in the currently followed path segment	[m]
$[x_{\text{n}}, y_{\text{n}}]$	is the position of the vessel in the NED frame	[m]

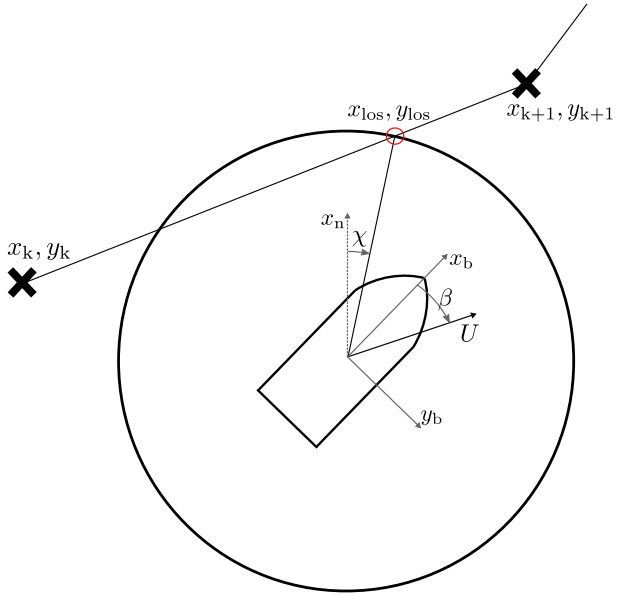


Figure 7.3: Algorithm used to find the yaw reference for the state space controller in order to follow a path.

The LOS point is then used to calculate χ as the angle from the x_{n} axis and the line joining the position of the vessel and the LOS point. See Equation 7.12. This can be directly used as the reference for yaw, ψ_{ref}, in the state space controller. This disregards the possibility of disturbances and assumes that the velocity vector of the vessel is aligned with the x_{b} axis. This is in general not true as disturbances like wind, current or waves would generate some force also in the y_{b} axis direction. The reference for yaw is then adjusted by subtracting the angle that the velocity vector has with respect to the x_{b} axis as seen in Equation 7.13 and Equation 7.14.

This approach tries to make the vessel velocity vector point towards the LOS point.

$$\chi = \arctan\left(\frac{y_{\text{LOS}} - y_{\text{n}}}{x_{\text{LOS}} - x_{\text{n}}}\right), \tag{7.12}$$

$$\beta = \arctan\left(\frac{\dot{y}_{\text{b}}}{\dot{x}_{\text{b}}}\right), \tag{7.13}$$

$$\psi_{\text{ref}} = \chi - \beta. \tag{7.14}$$

Where:

χ	is the angle between the x_{n} axis and the LOS point	[rad]
β	is the angle between the velocity vector of the vessel and the x_{b} axis	[rad]

The algorithm relies on the path and the circle defined around the vessel to cross at the LOS point.

If the vessel is positioned far from the path such that the circle does not intersect it, then the algorithm uses the next waypoint as LOS point. Once the vessel gets closer to the path, the LOS point is calculated as described above.

The yaw reference, ψ_{ref}, given to the controller in chapter 6 ensures that the vessel follows the desired LOS. While following the LOS the distance between the vessel and the path, e, in Figure 7.4, decreases. At each time sample ψ_{ref} is recalculated using the algorithm. This is done to ensure e converges to the zero, meaning the vessel converges onto the path.

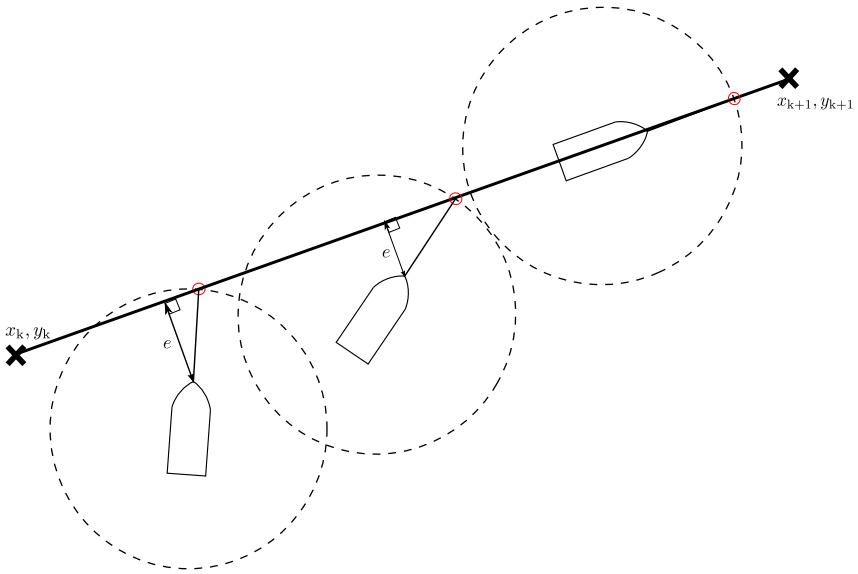


Figure 7.4: Path following algorithm minimizing the error, e, between the path and vessel.

This convergence can also be seen in the initial condition response of the controller, shown in Figure 7.5

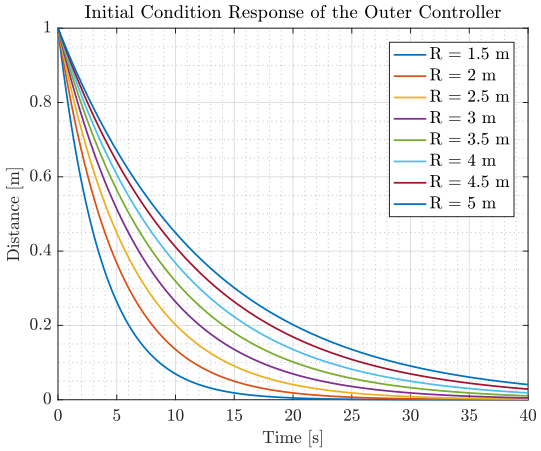


Figure 7.5: Initial condition response of the outer controller, when considering an ideal inner controller, that is, the inner dynamics are not considered. The speed is set to 0.4 m·s^{-1}.

As described above, the distance to the path converges to zero when an initial position outside the path is set. The speed of convergence depends on the radius of the circle chosen as can be seen in Figure 7.5. This is the tuning parameter that is used to design the controller.

For this particular system, and considering that the bandwidth of the LQR is 1.5 rad·s^{-1}, a radius of 2 m is chosen as this gives settling time of 15 s and a bandwidth of 0.2 rad·s^{-1}, seven times slower than the inner controller. For the $\mathcal{H}_\infty$ controller, the bandwidth is 0.6 rad·s^{-1}, so the radius is chosen to be 4.5 m, giving a bandwidth of 0.085 rad·s^{-1}.

In order to follow the path, a way to change which two waypoints define the current path segment needs to be established. Several possibilities can be considered but all of them change active waypoints when the vessel gets close enough to the waypoint that defines the end of the segment. In the project at hand, the distance to the waypoint is evaluated as the distance from the waypoint to the intersection point of

the path segment and a perpendicular line to the segment that passes through the vessel position. This distance is depicted in Figure 7.6.

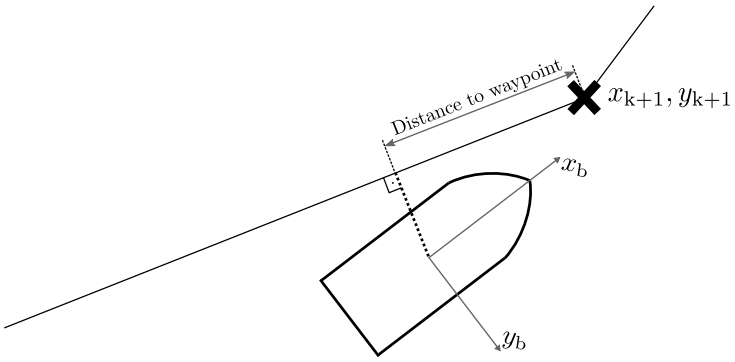


Figure 7.6: The distance considered when defining the criterion to change to new waypoints.

With this approach, the vessel tries to move forward in the path although a waypoint position has not been precisely attained.

This could be caused by a sudden disturbance experienced by the vessel and, in general, it is desired to keep following the path rather than turning around to hit the waypoint. In most cases, the vessel itself is going to be close to the waypoint when the change occurs. This can be seen in the simulation plots presented below.

7.3 Simulations

The path following algorithm is tested in the same path and considering different settings for the algorithm. In all cases, the distance in which the active waypoints are changed is 1 m. For seeing the results, the model of the system has been simulated with the path following algorithm and the two inner controller designs. In the plots presented, the data from 100 simulations is depicted, where the disturbances, model uncertainties and noise affect the system.

The wind and current disturbance varies randomly from ± 1.5 N in force along x_b and so does along y_b, and from ± 1.5 N·m in torque in ψ. The waves are assumed to be sinusoidal with amplitude of 1 N and a frequency that goes from 0 to 10 Hz. This disturbance is applied along x_b and y_b.

The uncertainty considered is of 20% in all parameters of the model, that is, the mass, m, the moment of inertia around the z_b axis, I_z, the damping coefficients, d_x, d_y and d_ψ, and the vessel lengths, l_1 and l_2.

In Figure 7.7, 7.8, 7.9 and 7.10 the results of the algorithm are presented by depicting the path taken by the vessel and the distance to the target path. In these figures, the path following algorithm corresponds to the simpler case in which $\psi_{ref} = \chi$, that is, assuming the velocity of the vessel is pointing along the x_b direction. The simulations are performed with both inner controller designs.

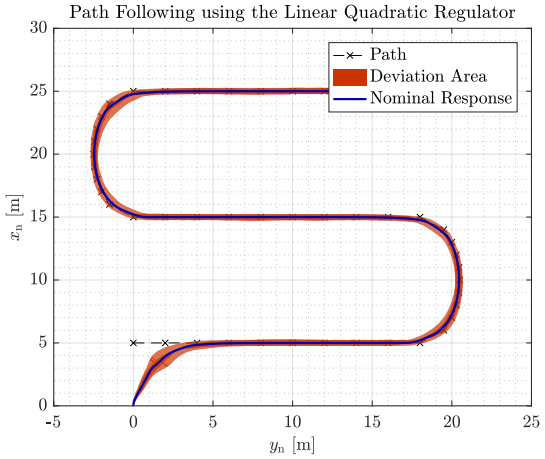


Figure 7.7: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi$ and using the LQR inner controller. The reference for velocity is set to 0.4 m·s^{-1}. and the system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

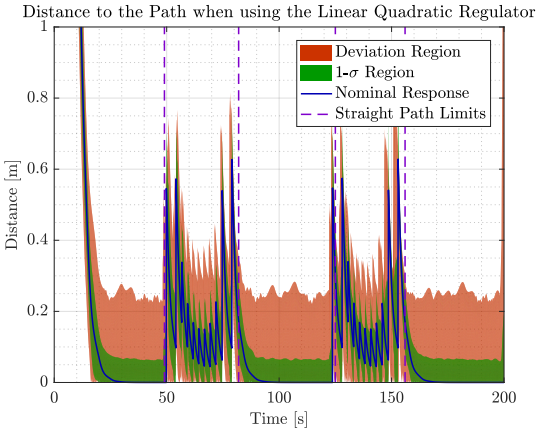


Figure 7.8: Distance to the path when using the algorithm based on $\psi_{\text{ref}} = \chi$ and the LQR inner controller.

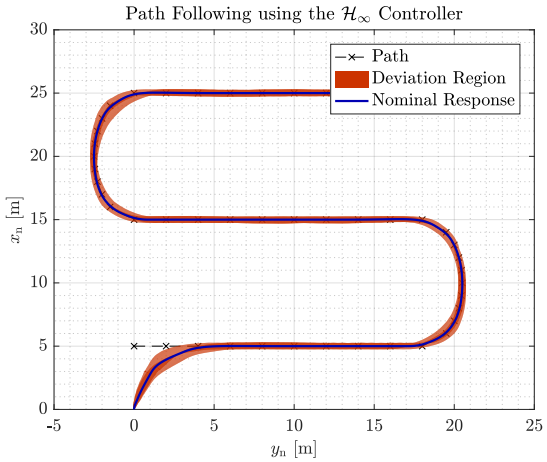


Figure 7.9: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi$ and using the $\mathcal{H}_\infty$ inner controller. The reference for velocity is set to 0.4 m·s^{-1}. and the system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

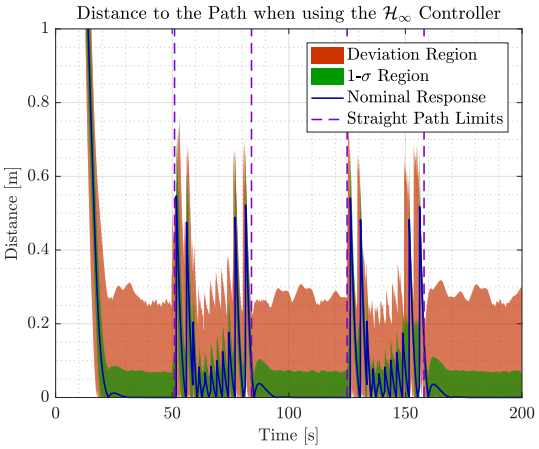


Figure 7.10: Distance to the path when using the algorithm based on $\psi_{\text{ref}} = \chi$ and the $\mathcal{H}_\infty$ inner controller.

In these graphs, it is clear that the path is not precisely followed when disturbances are introduced in the system. This offset is expected as the assumption for this simpler algorithm to work does not hold with disturbances like wind, current and waves. The latter is specially visible in the straight line parts of the path, where the vessel movement shows a 1σ value that goes up to 10 cm and a maximum value that reaches up to approximately 30 cm.

When the information of the vessel velocity is used to calculate the reference angle, ψ_{ref}, the disturbance is rejected. This is seen in Figure 7.11, 7.12, 7.13 and 7.14, where the vessel is experiencing disturbances and model perturbations in the same range as in the previously shown figures. Both the vessel x_n-y_n movement and the distance to the target path are depicted.

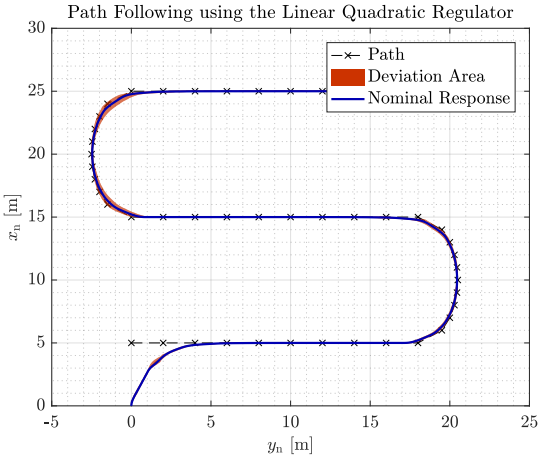


Figure 7.11: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi - \beta$ and using the LQR inner controller.The reference for velocity is set to 0.4 m·s^{-1}. and the system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

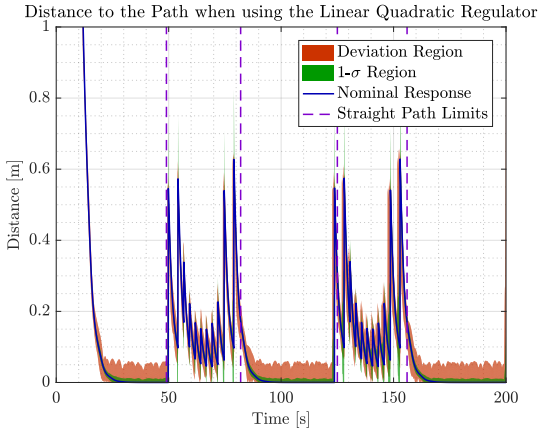


Figure 7.12: Distance to the path when using the algorithm based on $\psi_{\text{ref}} = \chi - \beta$ and the LQR inner controller.

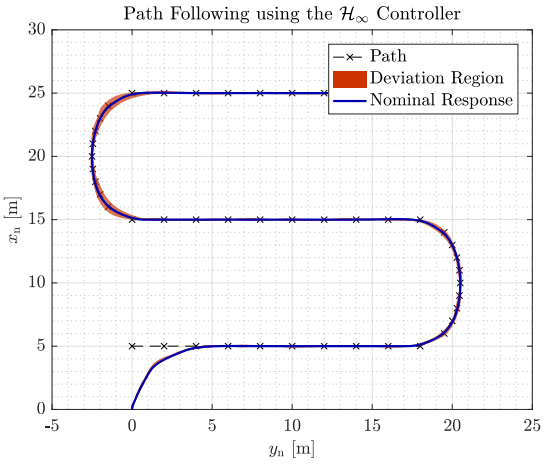


Figure 7.13: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi - \beta$ and using the $\mathcal{H}_\infty$ inner controller. The system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

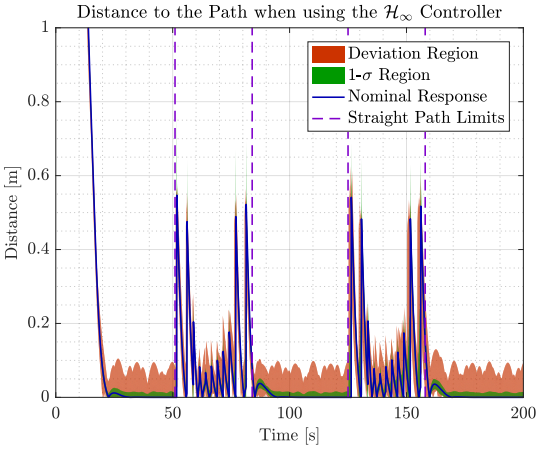


Figure 7.14: Distance to the path when using the algorithm based on $\psi_{\text{ref}} = \chi - \beta$ and the $\mathcal{H}_\infty$ inner controller.

In this case, the offset in position is corrected and the path is followed within the desired precision in the straight line segments. There are still some small deviations due to the wave disturbance present in the system, but the 1σ value is under 5 cm and the maximum variations stay below 10 cm with respect to the reference path.

When comparing the two inner controllers, different behaviors are seen. In the turns, the variations experienced by the LQR are larger than those seen with the $\mathcal{H}_\infty$ controller. This is expected as it is consistent with the response seen in the step responses presented in section 6.3, where a higher deviations appeared in the LQR when the references were applied to the system.

In the straight line segments, the behavior is the opposite, the LQR shows a smaller maximum and 1σ error than those shown by the $\mathcal{H}_\infty$ controller. This is due to the speed of the controller in terms of settling time, the faster response of the LQR makes it able of correcting the constant wind disturbances and low frequency wave disturbances slightly better, making the error lower by approximately 4 cm.

8 | Sensor Fusion

The sensors placed in the vessel are, as presented in section 3.3, an IMU and a GPS module. The information provided by these sensors is combined in order to obtain useful information to use in the controllers. This process is achieved using two Kalman filters.

The Kalman filter is a linear recursive estimator that can be applied to systems with both linear and nonlinear dynamics. The algorithm consists of two steps, a prediction step and an update step [35]. The reason for using two of them is to make the structure of the system more modular and gain advantage of the faster update rate of the IMU compared with that of the GPS.

The first filter estimates the attitude, the angular velocity and the angular acceleration from the data provided by the IMU. The second one estimates the position in the NED frame and the translational velocity and acceleration in the x_b and y_b directions, using the IMU acceleration data and the GPS measurements.

In this chapter, a more detailed description of the two filters are presented, as well as simulation results to analyze their performance.

8.1 Attitude Estimation

The estimation contains attitude, angular velocity and angular acceleration information. For constructing the Kalman filter, a process model and a measurement model need to be created.

The process model describes the dynamics of the system and how the inputs affect its states. The process model also includes noise, which is assumed to be normally distributed. The measurement model describes how the measurements taken from the IMU relate with the states of the system represented in the process model. Measurement noise is also included in the model. The process and measurement models are

$$\hat{\mathbf{x}}_\mathrm{att}(k+1) = \mathbf{A}_\mathrm{att}\hat{\mathbf{x}}_\mathrm{att}(k) + \mathbf{B}_\mathrm{att}\mathbf{u}(k) + \mathbf{w}_\mathrm{att}(k)\ , \tag{8.1}$$

$$\mathbf{y}_\mathrm{att}(k) = \mathbf{C}_\mathrm{att}\hat{\mathbf{x}}_\mathrm{att}(k) + \mathbf{v}_\mathrm{att}(k)\ . \tag{8.2}$$

Where:

$\hat{\mathbf{x}}_\mathrm{att}$	is the system state vector for the attitude Kalman filter
$\mathbf{u}$	is the input vector
$\mathbf{w}_\mathrm{att}$	is the process noise vector
$\mathbf{y}_\mathrm{att}$	is the measurement vector
$\mathbf{v}_\mathrm{att}$	is the measurement noise vector
$\mathbf{A}_\mathrm{att}$	is the system matrix for the attitude Kalman filter model
$\mathbf{B}_\mathrm{att}$	is the input matrix for the attitude Kalman filter model
$\mathbf{C}_\mathrm{att}$	is the output matrix for the attitude Kalman filter model

The noise vectors $\mathbf{w}_\mathrm{att}$ and $\mathbf{v}_\mathrm{att}$ are independent, and follow a zero mean normal distribution. The covariance matrices of each distribution are $\mathbf{Q}_\mathrm{att}$ and $\mathbf{R}_\mathrm{att}$, respectively. These are diagonal matrices as

all the errors are considered independent, and are calculated as

$$\mathbf{Q}_{\text{att}} = diag\left(\sigma_{\phi}^2, \sigma_{\theta}^2, \sigma_{\psi}^2, \sigma_{\dot{\phi}}^2, \sigma_{\dot{\theta}}^2, \sigma_{\dot{\psi}}^2, \sigma_{\ddot{\phi}}^2, \sigma_{\ddot{\theta}}^2, \sigma_{\ddot{\psi}}^2\right) \ , \tag{8.3}$$

$$\mathbf{R}_{\text{att}} = diag\left(\sigma_{\phi,\text{acc}}^2, \sigma_{\theta,\text{acc}}^2, \sigma_{\psi,\text{mag}}^2, \sigma_{\dot{\phi},\text{gyro}}^2, \sigma_{\dot{\theta},\text{gyro}}^2, \sigma_{\dot{\psi},\text{gyro}}^2\right) \ . \tag{8.4}$$

The states variances have been found iteratively in order to obtain the a good estimation of the states. The measurements variances have been obtained from the real sensors and can be seen in Appendix D.

The states chosen for the Kalman model are

$$\hat{\mathbf{x}}_{\text{att}} = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} & \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^{\text{T}} \ . \tag{8.5}$$

Even though the inner state space controller only considers ψ and $\dot{\psi}$, the other Euler angles and angular velocities are included to allow a better overall estimation, especially when using the rotation matrix elements as described in section 8.2. The angular accelerations are normally not part of a state vector in a state space representation. However, in the attitude Kalman filter state vector, the accelerations are included as they are an important part of the dynamics of the vessel as it can be seen in the last tree rows in A_{att}, Equation 8.11.

The output vector elements depend on the measurements given by the IMU. These include the angular velocities provided by the gyroscope and the attitude measurements. The latter are obtained from a direct calculation using the accelerometer data to compute ϕ_{acc} and θ_{acc} and the magnetometer data projected on to the plane to compute ψ_{mag} [36], as

$$\phi_{\text{acc}} = -\arctan\left(\frac{\ddot{y}_{\text{b,acc}}}{\sqrt{\ddot{x}_{\text{b,acc}}^2 + \ddot{z}_{\text{b,acc}}^2}}\right) \ , \tag{8.6}$$

$$\theta_{\text{acc}} = -\arctan\left(\frac{\ddot{x}_{\text{b,acc}}}{\sqrt{\ddot{y}_{\text{b,acc}}^2 + \ddot{z}_{\text{b,acc}}^2}}\right) \ , \tag{8.7}$$

$$\psi_{\text{mag}} = \arctan\left(\frac{M_{y_{\text{b}}}\cos(\phi) + M_{z_{\text{b}}}\sin(\phi)}{M_{x_{\text{b}}}\cos(\theta) + M_{y_{\text{b}}}\sin(\phi)\sin(\theta) + M_{z_{\text{b}}}\cos(\phi)\sin(\theta)}\right) \ . \tag{8.8}$$

Where:

$\ddot{x}_{\text{b,acc}}$	is the measured acceleration along the x_{b} direction	$[\text{m}\cdot\text{s}^{-2}]$
$\ddot{y}_{\text{b,acc}}$	is the measured acceleration along the y_{b} direction	$[\text{m}\cdot\text{s}^{-2}]$
$\ddot{z}_{\text{b,acc}}$	is the measured acceleration along the z_{b} direction	$[\text{m}\cdot\text{s}^{-2}]$
$M_{x_{\text{b}}}$	is the magnetic field strength along the x_{b} direction	[G]
$M_{y_{\text{b}}}$	is the magnetic field strength along the y_{b} direction	[G]
$M_{z_{\text{b}}}$	is the magnetic field strength along the z_{b} direction	[G]

Leading to the measurement vector

$$\mathbf{y}_{\text{att}} = \begin{bmatrix} \phi_{\text{acc}} & \theta_{\text{acc}} & \psi_{\text{mag}} & \dot{\phi}_{\text{gyro}} & \dot{\theta}_{\text{gyro}} & \dot{\psi}_{\text{gyro}} \end{bmatrix}^{\text{T}} \ . \tag{8.9}$$

The input vector, in this case stays the same as in the state space controller design, containing the two forces provided by the thrusters. The input vector is

$$\mathbf{u} = \begin{bmatrix} F_1 & F_2 \end{bmatrix}^{\text{T}} \ . \tag{8.10}$$

With this information, the process and measurement model matrices are built. The $\mathbf{A}_{\text{att}}$ matrix describes the dynamics of the model. In this system, the sampling time of the model, T_{S}, is chosen to be 0.05 s as it is the same sampling time as the controller. This allows the controller to react to the updated estimates from the Kalman filter, as it is calculated at each time step.

The same dynamics describe the angular velocities. The angular accelerations depend on the angular velocities through the damping coefficients. In order to consider the most recent value of the angular velocities, the three last rows of $\mathbf{A}_{\text{att}}$ are just a copy of the three middle rows of $\mathbf{A}_{\text{att}}$ multiplied by the corresponding damping coefficient and divided by the matching moment of inertia. The elements on these last rows also include the minus sign coming from the damping. The $\mathbf{A}_{\text{att}}$ matrix is

$$\mathbf{A}_{\text{att}} = \begin{bmatrix} 1 & 0 & 0 & T_{\text{s}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_{\text{s}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_{\text{s}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & T_{\text{s}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & T_{\text{s}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & T_{\text{s}} \\ 0 & 0 & 0 & -\frac{d_\phi}{I_{\text{x}}} & 0 & 0 & -T_{\text{s}}\frac{d_\phi}{I_{\text{x}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{d_\theta}{I_{\text{y}}} & 0 & 0 & -T_{\text{s}}\frac{d_\theta}{I_{\text{y}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{d_\psi}{I_{\text{z}}} & 0 & 0 & -T_{\text{s}}\frac{d_\psi}{I_{\text{z}}} \end{bmatrix}. \tag{8.11}$$

The matrix $\mathbf{B}_{\text{att}}$ is mostly formed by zeros, the only nonzero elements are placed in the last row, indicating how the forces contribute to the angular acceleration in the yaw angle. The $\mathbf{C}_{\text{att}}$ matrix is also straightforward to obtain as the measurement are just the first six states in $\hat{\mathbf{x}}_{\text{att}}$. The $\mathbf{B}_{\text{att}}$ and $\mathbf{C}_{\text{att}}$ matrices are

$$\mathbf{B}_{\text{att}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{l_1}{I_{\text{z}}} & -\frac{l_2}{I_{\text{z}}} \end{bmatrix}, \qquad \mathbf{C}_{\text{att}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{8.12}$$

Once the model is constructed, the Kalman filter equations are obtained. They are divided in two steps, prediction and update [35]. To start the process, the estimate, $\hat{\mathbf{x}}_{\text{att}}$, and error covariance matrix, $\mathbf{P}_{\text{att}}$, need to be initialized. The state estimation is initialized with its mean, which is zero and the error covariance matrix is initialized with the covariance matrix for the states. The initialization is done as

$$\hat{\mathbf{x}}_{\text{att}}(0|0) = \mathbf{0}_{6\text{x}1}\ , \tag{8.13}$$

$$\mathbf{P}_{\text{att}}(0|0) = \mathbf{Q}_{\text{att}}\ . \tag{8.14}$$

In the prediction step, the sensor's most recent measurement has not been acquired yet and the model of the system is used to predict the state values in the next period. The error covariance is also predicted using the system matrix, $\mathbf{A}_{\text{att}}$, and the covariance matrix, $\mathbf{Q}_{\text{att}}$, of the process noise vector included in the model. The prediction step of the Kalman filter is

$$\hat{\mathbf{x}}_{\text{att}}(k+1|k) = \mathbf{A}_{\text{att}}\hat{\mathbf{x}}_{\text{att}}(k|k) + \mathbf{B}_{\text{att}}\mathbf{u}(k) \ , \tag{8.15}$$

$$\mathbf{P}_{\text{att}}(k+1|k) = \mathbf{A}_{\text{att}}\mathbf{P}_{\text{att}}(k|k)\mathbf{A}_{\text{att}}^{\text{T}} + \mathbf{Q}_{\text{att}} \ . \tag{8.16}$$

The update step corrects the estimate using the prediction obtained in the previous step and the innovation [35, p. 7], that is, the error between the new measurement and the predicted new measurement. This error updates $\hat{\mathbf{x}}_{\text{att}}$ weighted by the Kalman gain $\mathbf{K}(k)$. The update step of the Kalman filter calculates the updated state estimation and the updated error covariance as

$$\hat{\mathbf{x}}_{\text{att}}(k+1|k+1) = \hat{\mathbf{x}}_{\text{att}}(k+1|k) + \mathbf{K}(k+1)\left[\mathbf{y}_{\text{att}}(k+1) - \mathbf{C}_{\text{att}}\hat{\mathbf{x}}_{\text{att}}(k+1|k)\right] \ , \tag{8.17}$$

$$\mathbf{P}_{\text{att}}(k+1|k+1) = \left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}_{\text{att}}^{\text{T}}\right]\mathbf{P}_{\text{att}}(k+1|k) \ . \tag{8.18}$$

where the Kalman gain is given by

$$\mathbf{K}(k+1) = \mathbf{P}_{\text{att}}(k+1|k)\mathbf{C}_{\text{att}}^{\text{T}}\left[\mathbf{C}_{\text{att}}\mathbf{P}(k+1|k)\mathbf{C}_{\text{att}}^{\text{T}} + \mathbf{R}_{\text{att}}\right]^{-1} \ , \tag{8.19}$$

which weights how much the measurements and the prediction affect the estimate of the states. It is calculated using the prediction error covariance matrix, $\mathbf{P}_{\text{att}}$, the output matrix, $\mathbf{C}_{\text{att}}$, and the covariance matrix for the noise vector included in the measurement model. A more detailed derivation of this gain can be found in [35, pp. 5-8].

8.1.1 Simulation of the Filter

The performance of the Kalman filter is tested through simulations. These are performed by applying some arbitrary inputs to the model of the system derived in chapter 4. The signals obtained are transformed into measurements by adding noise whose variance is the present in the physical sensors and shown in Appendix D. The simulations are also used to tune the filters so a good estimation of the states is obtained. The tuning parameters are the elements in the covariance matrix for the states, $\mathbf{Q}_{\text{att}}$.

The procedure starts by focusing on the estimation of the angular velocities and the angular accelerations as these states are mostly dependent on the measurement given by the gyroscope and act as input for the attitude state estimation.

The result is shown in Figure 8.1 and 8.2, where the estimations for $\dot{\psi}$ and $\ddot{\psi}$ are plotted. As it can be seen, both signals are correctly estimated. It is especially remarkable in $\ddot{\psi}$, as there are no direct measurements for this state.

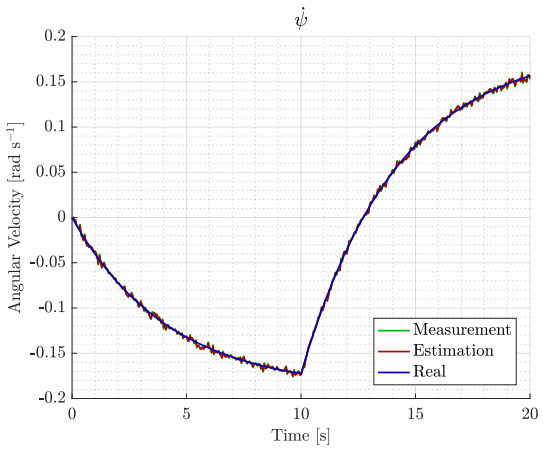


Figure 8.1: Result of the estimation of the angular velocity around z_b, compared to the real value in the simulation and the measurements.

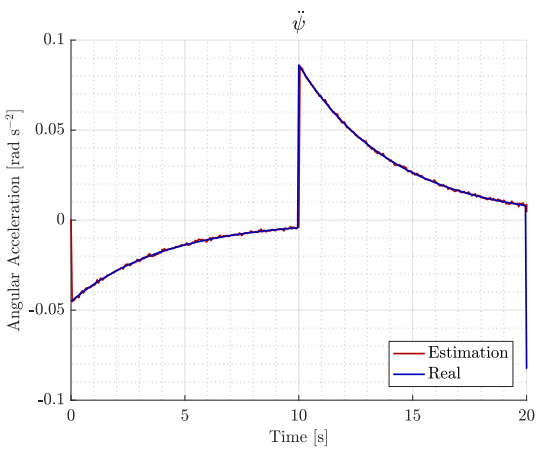


Figure 8.2: Estimation of the angular acceleration around z_b, compared to the real value.

Figure 8.3 shows the estimation of the heading, ψ as an example of the attitude estimation of the Kalman filter. This plot is obtained with the final values for the state covariance matrix elements.

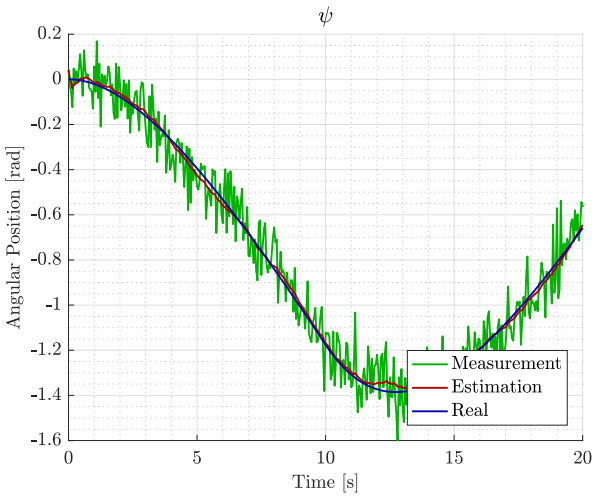


Figure 8.3: Result of the estimation of the heading, compared to the real value in the simulation and the measurements.

In this case, the filter is also able to remove most of the noise present in the measurement and provide an estimation close to the real value.

The final values obtained for $\mathbf{Q}_{\text{att}}$ are

$$\mathbf{Q}_{\text{att}} = diag\left(10^{-5}, 10^{-5}, 10^{-5}, 10^{-4}, 10^{-4}, 10^{-10}, 10^{-2}, 10^{-2}, 10^{-2}\right) \ . \tag{8.20}$$

8.2 Position Estimation

As for the attitude Kalman filter, a process model and a measurement model need to be created. These are

$$\hat{\mathbf{x}}_{\text{pos}}(k+1) = \mathbf{A}_{\text{pos}}(k)\mathbf{x}_{\text{pos}}(k) + \mathbf{B}_{\text{pos}}\mathbf{u}(k) + \mathbf{w}_{\text{pos}}(k) \ , \tag{8.21}$$

$$\mathbf{y}_{\text{pos}}(k) = \mathbf{C}_{\text{pos}}\hat{\mathbf{x}}_{\text{pos}}(k) + \mathbf{v}_{\text{pos}}(k) \ . \tag{8.22}$$

Where:

$\hat{\mathbf{x}}_{\text{pos}}$	is the system state vector
$\mathbf{w}_{\text{pos}}$	is the process noise vector
$\mathbf{y}_{\text{pos}}$	is the measurement vector
$\mathbf{v}_{\text{pos}}$	is the measurement noise vector
$\mathbf{A}_{\text{pos}}$	is the system matrix for the position Kalman filter
$\mathbf{B}_{\text{pos}}$	is the input matrix for the position Kalman filter
$\mathbf{C}_{\text{pos}}$	is the output matrix for the position Kalman filter

The noise, $\mathbf{w}_{\text{pos}}(k)$ and $\mathbf{v}_{\text{pos}}(k)$, are independent zero mean white Gaussian noise vectors with covariance matrices $\mathbf{Q}_{\text{pos}}$ and $\mathbf{R}_{\text{pos}}$, respectively. These are

$$\mathbf{Q}_{\text{pos}} = \text{diag}\left(\sigma^2_{\text{x}_\text{n}}, \sigma^2_{\text{y}_\text{n}}, \sigma^2_{\dot{\text{x}}_\text{b}}, \sigma^2_{\dot{\text{y}}_\text{b}}, \sigma^2_{\ddot{\text{x}}_\text{b}}, \sigma^2_{\ddot{\text{y}}_\text{b}}\right) \ , \tag{8.23}$$

$$\mathbf{R}_{\text{pos}} = \text{diag}\left(\sigma^2_{x_{\text{n,GPS}}}, \sigma^2_{y_{\text{n,GPS}}}, \sigma^2_{\ddot{x}_{\text{b,acc}}}, \sigma^2_{\ddot{y}_{\text{b,acc}}}\right) \ . \tag{8.24}$$

As well as in the attitude filter, the state variances are found iteratively using the simulations while the measurement variances come from the real sensors and can be seen in Appendix E and D.

The state vector for this Kalman filter is formed by the position in the NED frame, the velocity in the body frame and the acceleration in the body frame. These variables are either needed in the outer path follower controller or they describe an important part of the dynamics of the system. The position and velocity are in the former group while the acceleration is in the latter. It can also be seen from the state vector, Equation 8.25, that the z coordinate is not considered in this filter as it is not needed in the controllers and it does not play a relevant role in the dynamics of the system.

$$\hat{\mathbf{x}}_{\text{pos}} = \begin{bmatrix} x_\text{n} & y_\text{n} & \dot{x}_\text{b} & \dot{y}_\text{b} & \ddot{x}_\text{b} & \ddot{y}_\text{b} \end{bmatrix}^\text{T} \ . \tag{8.25}$$

The measurement vector is formed by the position data provided by the GPS module and the accelerations in the body frame provided by the IMU's accelerometer. This is represented as

$$\mathbf{y}_{\text{pos}} = \begin{bmatrix} x_{\text{n,GPS}} & y_{\text{n,GPS}} & \ddot{x}_{\text{b,acc}} & \ddot{y}_{\text{b,acc}} \end{bmatrix}^\text{T} \ . \tag{8.26}$$

Finally, as for the attitude Kalman filter, the input vector is formed by the two thruster forces.

$$\mathbf{u} = \begin{bmatrix} F_1 & F_2 \end{bmatrix}^\text{T} \ . \tag{8.27}$$

The next step is to build the matrices involved in the models. The sampling time T_s is chosen as 0.2

seconds as this is the rate that the RTK GPS updates its position with. The $\mathbf{A}_{\text{pos}}$ matrix written as

$$\mathbf{A}_{\text{pos}}(\phi(k),\theta(k),\psi(k)) = \begin{bmatrix} 1 & 0 & T_{\text{s}}\,\mathbf{R}_{\text{b}}^{\text{n}}(1,1) & T_{\text{s}}\,\mathbf{R}_{\text{b}}^{\text{n}}(1,2) & 0 & 0 \\ 0 & 1 & T_{\text{s}}\,\mathbf{R}_{\text{b}}^{\text{n}}(2,1) & T_{\text{s}}\,\mathbf{R}_{\text{b}}^{\text{n}}(2,2) & 0 & 0 \\ 0 & 0 & 1 & 0 & T_{\text{s}} & 0 \\ 0 & 0 & 0 & 1 & 0 & T_{\text{s}} \\ 0 & 0 & -\frac{d_{\text{x}}}{m} & 0 & -T_{\text{s}}\frac{d_{\text{x}}}{m} & 0 \\ 0 & 0 & 0 & -\frac{d_{\text{y}}}{m} & 0 & -T_{\text{s}}\frac{d_{\text{y}}}{m} \end{bmatrix} . \tag{8.28}$$

As it can be seen from the matrix, some terms are not constant, but depend on the attitude state of the vessel. These terms come from the transformation of the translational velocity from the body frame to the NED frame when using the rotation matrix in Equation 4.1. They are also shown below for repetition purposes.

$$\begin{aligned} \mathbf{R}_{\text{b}}^{\text{n}}(1,1) &= \cos(\theta(k))\cos(\psi(k)) \ , \\ \mathbf{R}_{\text{b}}^{\text{n}}(1,2) &= \sin(\phi(k))\sin(\theta(k))\cos(\psi(k)) - \cos(\phi(k))\sin(\psi(k)) \ , \\ \mathbf{R}_{\text{b}}^{\text{n}}(2,1) &= \cos(\theta(k))\sin(\psi(k)) \ , \\ \mathbf{R}_{\text{b}}^{\text{n}}(2,2) &= \sin(\phi(k))\sin(\theta(k))\sin(\psi(k)) + \cos(\phi(k))\cos(\psi(k)) \ . \end{aligned} \tag{8.29}$$

These elements are used by evaluating the matrix with the most recent attitude estimate provided by the attitude Kalman filter. In this way, the $\mathbf{A}_{\text{pos}}$ matrix is known at each Kalman filter cycle.

The $\mathbf{B}_{\text{pos}}$ matrix has nonzero elements only in the row related with the acceleration in x_{b} direction. The $\mathbf{C}_{\text{pos}}$ has a similar structure as for the attitude Kalman filter, with ones in the diagonal when the measurements and the states are directly correlated. This occurs for the position in the NED frame and the accelerations in the body frame. The $\mathbf{B}_{\text{pos}}$ and $\mathbf{C}_{\text{pos}}$ are

$$\mathbf{B}_{\text{pos}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ 0 & 0 \end{bmatrix} , \qquad \mathbf{C}_{\text{pos}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{8.30}$$

With these matrices, the prediction and update steps of the Kalman filter can be calculated as described below. As for the attitude Kalman filter, the estimate and the error covariance matrix need to be initialized. This is done as

$$\hat{\mathbf{x}}_{\text{pos}}(0|0) = \mathbf{0}_{4\text{x}1} \ , \tag{8.31}$$

$$\mathbf{P}_{\text{pos}}(0|0) = \mathbf{Q}_{\text{pos}} \ . \tag{8.32}$$

Before performing the prediction step, the elements of the $\mathbf{A}_{\text{pos}}$ matrix that depend on the attitude state of the vessel are calculated using the last attitude estimation. Then, the state vector estimation, $\hat{\mathbf{x}}_{\text{pos}}$, and error covariance matrix, $\mathbf{P}_{\text{pos}}$ are predicted as

$$\hat{\mathbf{x}}_{\text{pos}}(k+1|k) = \mathbf{A}_{\text{pos}}(k)\hat{\mathbf{x}}_{\text{pos}}(k|k) + \mathbf{B}_{\text{pos}}\mathbf{u}(k) \ , \tag{8.33}$$

$$\mathbf{P}_{\text{pos}}(k+1|k) = \mathbf{A}_{\text{pos}}(k)\mathbf{P}_{\text{pos}}(k|k)\mathbf{A}_{\text{pos}}(k)^{\text{T}} + \mathbf{Q}_{\text{pos}} \ . \tag{8.34}$$

Once the new measurement data, $\mathbf{y}(k)$, is received, the estimate is corrected according to the Kalman gain, which weighs the importance of the measurement and the prediction for updating the estimate. The equations for the update step are

$$\hat{\mathbf{x}}_{\text{pos}}(k+1|k+1) = \hat{\mathbf{x}}_{\text{pos}}(k+1|k) + \mathbf{K}\left[\mathbf{y}_{\text{pos}}(k+1) - \mathbf{C}_{\text{pos}}\hat{\mathbf{x}}_{\text{pos}}(k+1|k)\right] \ , \tag{8.35}$$

$$\mathbf{P}_{\text{pos}}(k+1|k+1) = \left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}_{\text{pos}}^{\text{T}}\right]\mathbf{P}_{\text{pos}}(k+1|k) \ . \tag{8.36}$$

The Kalman gain is calculated as

$$\mathbf{K}(k+1) = \mathbf{P}_{\text{pos}}(k+1|k)\mathbf{C}_{\text{pos}}^{\text{T}}\left[\mathbf{C}_{\text{pos}}\mathbf{P}(k+1|k)\mathbf{C}_{\text{pos}}^{\text{T}} + \mathbf{R}_{\text{pos}}\right]^{-1} \ . \tag{8.37}$$

8.2.1 Simulation of the Filter

The position Kalman filter is tuned and tested utilizing simulations. These are performed by applying some inputs to the simulation model of the system. The signals coming out of the model are then extracted and some noise is added to them. The noisy signals are used as the input to the position Kalman filter in order to evaluate its performance. The amount of noise added is the same as that present in the real sensors, whose variances are seen in Appendix E and D.

As in the case of the attitude, the tuning is done by choosing the appropriate values for the covariance matrix of the states, $\mathbf{Q}_{\text{pos}}$.

This procedure starts by obtaining a good estimation of the accelerations in the body frame, $\ddot{x}_{\text{b}}$ and $\ddot{y}_{\text{b}}$, since the other estimations depend on these. The result can be seen in Figure 8.4, where the measurement, the estimation and the real value of $\ddot{x}_{\text{b}}$ are depicted.

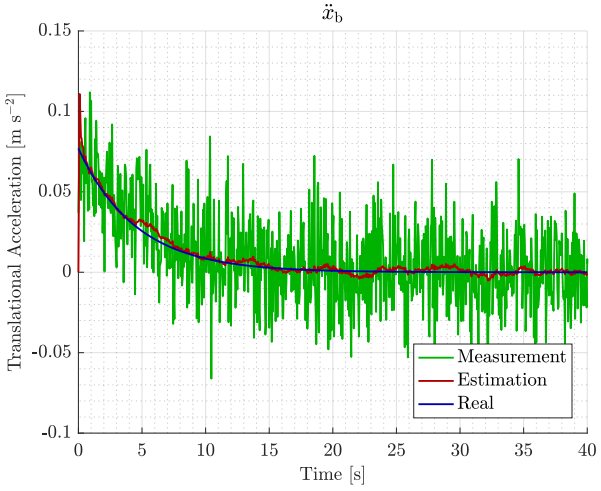


Figure 8.4: Measurement, real value and estimation of $\ddot{x}_{\text{b}}$.

It is noticeable that the filter reduces most of the noise present on the measurement and gives a good estimation of the acceleration in the body frame.

Then, the velocities, $\dot{x}_{\text{b}}$ and $\dot{y}_{\text{b}}$ estimations are tuned. These do not rely on any measurements and are mainly the result of integrating the acceleration as seen in the model used for the position Kalman filter, see Equation 8.28. The result for $\dot{x}_{\text{b}}$ is seen in Figure 8.5, where

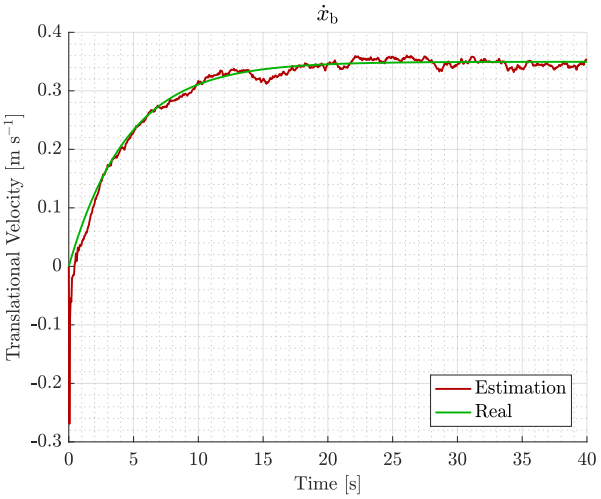


Figure 8.5: Real value and estimation of $\dot{x}_b$

The estimation of $\dot{x}_b$ gives a good result even though there is no direct measurement of the velocity.

Finally, the estimation of the positions in the NED frame, x_n and y_n, are tuned, these depend on the accelerations and velocities estimations and on the position measurements coming from the RTK GPS module. Figure 8.6 shows the result in an $x_n - y_n$ plot of the position of the vessel.

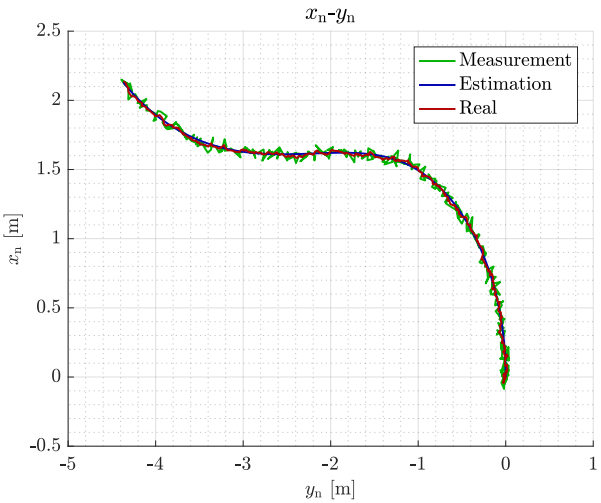


Figure 8.6: Measurement, real value and estimation of the $x_n - y_n$ position of the vessel in the NED frame.

As it can be observed, the estimation approximates the real value well and the Kalman filter removes most of the noise generated in the GPS.

The final covariance matrix for the states obtained after the tuning process is

$$\mathbf{Q}_{\text{pos}} = diag\left(10^{-3}, 10^{-3}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}\right) \ . \tag{8.38}$$

9 | Implementation

The implementation is based on Robot Operating System (ROS), which provides a communication infrastructure for a project, allowing multiple programs to communicate between each other.

Each program runs in individual threads, called nodes, such that their timing is independent from each other, except for hardware limitations. This allows each node to run in parallel in multiple threads while still being able to share data between them.

The data sharing is done through topics, onto which the nodes can publish and subscribe to. Each topic contains the data stored in a predefined data structure, specified as a message, such that each node publishes or reads data of the same type.

A diagram of the particular implementation, including nodes and topics can be seen in Figure 9.1.

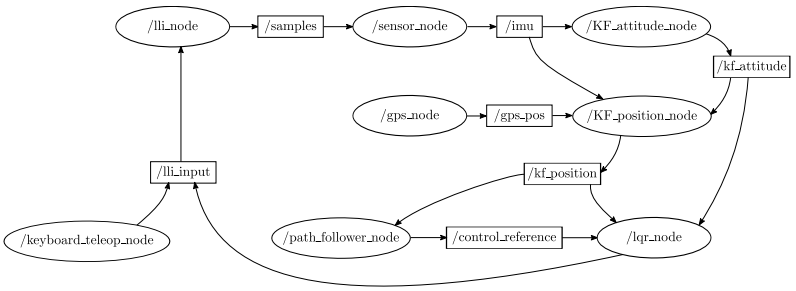


Figure 9.1: Diagram of the implementation in ROS, which includes the relations between nodes and topics.

A more detailed explanation of the functionality of each node and the data contained in each topic is presented in this chapter.

9.1 Nodes

Each node uses the information coming from a topic or from a device connected to the computer and publishes its results in another topic, which is then used by one or more nodes. The functionality of each node is described below.

`/lli_inode`

This node is in charge of reading the messages that come from the LLI and publishing them in the `/samples` topic. It is also responsible of sending the commands, which are published in the `/lli_input` topic, to the LLI. These commands need to be coded in a message, such that the LLI can read them.

`/sensor_node`

The purpose of this node is to decode the information of the IMU that is packed in the messages of the `/samples` topic. This is done extracting the data from a string and converting it in the measurements of the accelerometer, the magnetometer and the gyroscope by transforming them into the correct units. This information is then published in the `/imu` topic.

/gps_node

This node has two main functionalities.

It reads the position information that comes from the GPS from the same serial port and decodes it to know the latitude and longitude of the vessel. With this information it is able to compute the relative distance of the vessel with the chosen origin of the NED frame, given by its latitude and longitude. The latitude and longitude of the vessel, together with its relative position to the origin is published in the /gps_pos topic.

Additionally, it parses the correction data from the RTK base to the GPS in the vessel using the serial port. A more detailed description of the RTK base can be found in Appendix C.

/KF_attitude_node

The attitude Kalman filter is implemented in this node using the information that comes from the /imu topic. This node uses that data to estimate the angular position, velocity and acceleration of the vessel as described in section 8.1. Finally, the estimation is published in the /kf_attitude topic.

/KF_position_node

The estimation of the position of the vessel is done in this node, as described in section 8.2. The information of the GPS is fused with the measurements of the accelerometer and the estimated attitude to give a better estimate of the translational position, velocity and acceleration of the vessel. This estimation is then published in the /kf_position topic.

/path_follower_node

This node implements the path follower algorithm described in section 7.2. It reads the waypoints generated as described in section 7.1 from a .txt file, and the estimated position and attitude from the Kalman filters. With this information it is able to compute the required heading reference for the vessel to reach the desired path, which is published in the /control_reference topic.

/controller_node

The inner controller node uses the information from both filters as well as the reference published in the /control_reference topic to apply the gains, both state feedback and integral control, and computes the required force in each motor. These forces are finally translated to PWM values to be published in the /lli_input topic.

/keyboard_teleop_node

This node send commands to the motors using the /lli_input topic. The commands depend on the keys that are pressed:

- '1': Enable the commands.
- '0': Disable the commands and stop the motors.
- 'W'/'A': Increase/decrease the speed of the motors.

- 'S'/'D': Turn left/right. One of the motors rotates with half of its speed.
- 'H': Gives the motors a step in speed in the forward direction.
- 'T': Gives the motors a step in speed to turn counterclockwise.

9.2 Topics

Each topic is predefined to contain specific data that need to be exchanged between the nodes. This data can be anything from sensor data to commands for the motors.

The `/samples` topic contains a message that comes from the LLI.

The data decoded by the `/sensor_node` from the previous message, is published in the `/imu` topic, such as accelerometer, magnetometer and gyroscope data.

The `/gps_pos` topic includes the latitude and longitude of the vessel, as well as its relative position to the chosen origin of the NED frame.

Each Kalman filter node, `/KF_attitude_node` and `/KF_position_node`, publishes the estimated states in two topics, `/kf_attitude` and `/kf_position`, respectively. These states include angular and translational positions, velocities and accelerations.

The `/path_follower_node` publishes the references for speed and heading in the `/control_reference` topic, to whom the `/controller_node` is subscribed.

Finally, the `/controller_node` publishes the commands, in the form of PWM, that the motor controllers need to apply in the `/lli_input` topic.

Additionally, the `/keyboard_teleop_node` is used to manually control the ASV by publishing commands in the `/lli_input` topic.

In Part II, the required elements to construct the control system for an ASV have been designed. First, an inner controller design has been made based on two different control theories, namely, LQR and $\mathcal{H}_\infty$. Then, the outer controller design has been presented as a path following algorithm called enclosure based steering. The part also includes the sensor data processing, which obtains the necessary states for the different controllers, using two Kalman filters. Finally, the implementation of the designs in ROS has been described by presenting the nodes and topics involved.

Part III

Results & Conclusion

10 | Results

The designed system has been tested in order to check that the functional requirements stated in section 2.3 are fulfilled. The results include both simulations of the full system and plots obtained after performing real tests with the vessel.

10.1 Controller Requirements

A: It shall be possible to select the area in which the bathymetric measurements are to be performed.

B: The ASV shall be able to plan a route, such that the entire survey area is mapped.

C: The ASV shall be able to follow the planned route.

D: The controller shall be robust to external disturbances.

The area to survey is selected by giving the four corner points. Based on these points, the path generation algorithm selects the needed waypoints to cover it, depending on the beamwidth, fulfilling requirements **A** and **B**.

Once these waypoints are selected, the outer controller sets the appropriate references for both speed, $\dot{x}_b$, and heading, ψ, to the inner controller. The performance of the outer controller when using both inner controllers can be seen in section 7.3. The figures are included here as well, in order to check that requirements **C** and **D** are met.

Figure 10.1 shows the simulation results of the path followed by the vessel in the survey area when using the LQR as inner controller. The results are also shown in Figure 10.2 by plotting the error to the reference path.

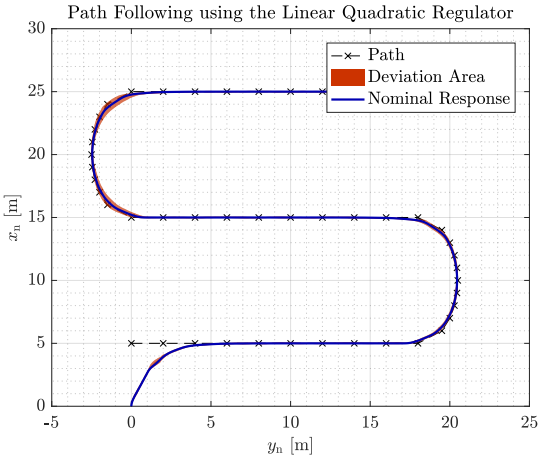


Figure 10.1: Performance of the path following algorithm using the LQR as inner controller. The system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

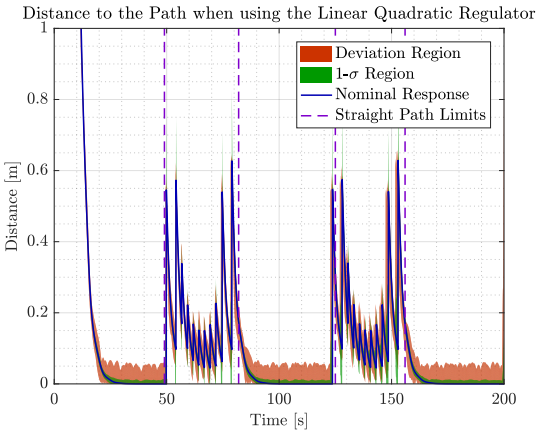


Figure 10.2: Distance to the path when using the following algorithm and the LQR inner controller.

It can be seen that the distance to the path is kept below 10 cm in the straight parts of the path, which

correspond to the given area, while the distance in the curved sections is not relevant for the survey task. This accuracy is deemed sufficient for the control system of a survey vessel as the tracking accuracy is not critical for this type of application.

The path has also been tracked using the $\mathcal{H}_\infty$ inner controller as seen in Figure 10.3 and 10.4.

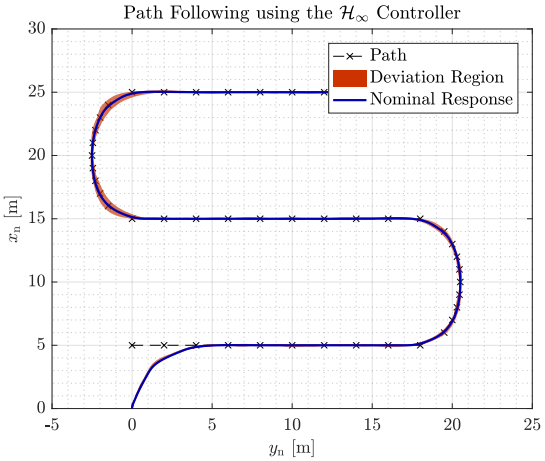


Figure 10.3: Performance of the path following algorithm using the $\mathcal{H}_\infty$ as inner controller. The system is experiencing wind, current and wave disturbances, model perturbations and measurement noise.

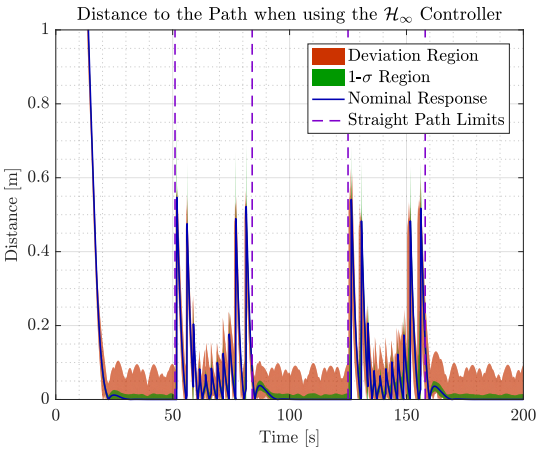


Figure 10.4: Distance to the path when using the algorithm based on $\psi_{\text{ref}} = \chi - \beta$ and the $\mathcal{H}_\infty$ inner controller.

In this case, the path is also adequately tracked as the maximum deviation in the straight line segments is approximately 15 cm, staying below 10 cm for most of the survey path segments.

According to the results of the simulations, it can be said that the vessel follows the path through the waypoints when they are part of a straight line section of the path. In curved sections, the vessel joins smoothly the straight line segments that approximate the curve.

Since the simulation includes disturbances, noise and parameter variation, the robustness of the controller against them has also been tested. In many cases, and especially for bathymetric measurements, the algorithm can be considered suitable.

10.2 Implementation Requirements

E: The THU shall not exceed 30 cm with a 95% confidence interval.

F: The ASV shall record and store data locally for extraction at the end of the survey.

G: It shall be possible to give the ASV a command to stop and steer it back to land.

As it can be seen in Appendix E, the position measurements of the GPS have a 95 percentile at 0.02 m, that is, 95% of the samples measured are within 0.02 m from the mean. The measurements from the test are also shown in Figure 10.5. In this figure, the measurements shows that an accuracy error occurs. It is especially clear in the altitude plot, where the position acquired indicates the vessel is below sea level. From this, requirement **E** is not validated. Possible reasons for this issue can be found in chapter 11.

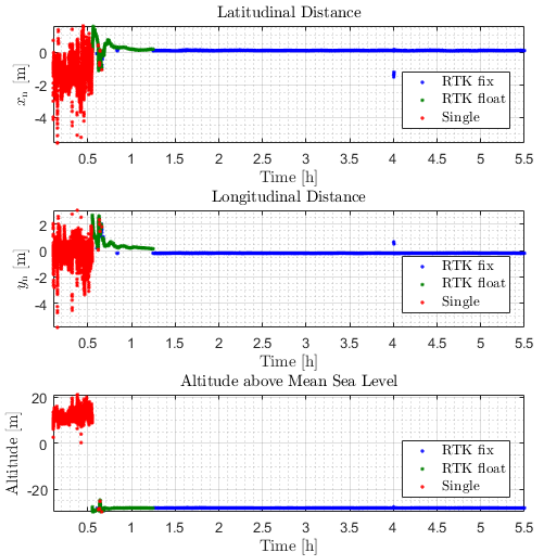


Figure 10.5: GPS measurements using the three modes.

The implementation in ROS allows to record all the topics, which contain all the sensor, estimation and inputs needed, in order to process and analyze the data afterwards. This ensures that requirement **F** is achieved.

A VPN has been set up such that it is possible to connect to the vessel remotely, allowing to run and stop the nodes when needed. A node for manual control has been added as well, such that it is possible to remotely move and control the vessel, satisfying requirement **G**.

Using the ROS implementation the controllers designed for the ASV were tested. This has been done as explained in Appendix J. It was seen that the designed LQR and $\mathcal{H}_\infty$ controllers were too aggressive and were not able to sufficiently stabilize the vessel. This led to a redesign of the LQR such that the weight matrices $\mathbf{Q}$ and $\mathbf{R}$ have been tuned such that the maximum force has been reduced while the maximum allowable value for the states has been increased to reduce the aggressiveness of the controller.

With these new weight matrices, a new controller was calculated such that the controller gains are

$$\mathbf{F} = \begin{bmatrix} 124.3988 & 113.7050 & 88.7435 \\ -124.3988 & -113.7050 & 88.7435 \end{bmatrix}, \tag{10.1}$$

$$\mathbf{F}_\mathrm{I} = \begin{bmatrix} -20.6709 & -17.7744 \\ 20.6709 & -17.7744 \end{bmatrix}. \tag{10.2}$$

It should also be noted that the Kalman filter for the position gave inaccurate measurements for the speed, $\dot{x}_\mathrm{b}$. Due to time constraints the problem was not be solved. The Kalman filter implementation from [26] was finally used to obtain a reliable $\dot{x}_\mathrm{b}$.

This new controller has been tested by tracking a reference in heading while keeping a constant speed. The results can be seen in Figure 10.6 and 10.7.

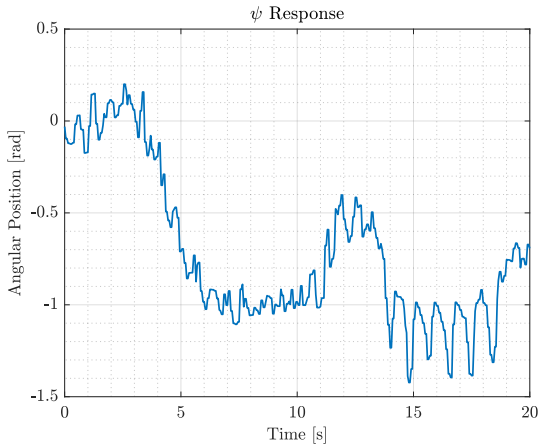


Figure 10.6: Step response in ψ with a reference of -1 rad.

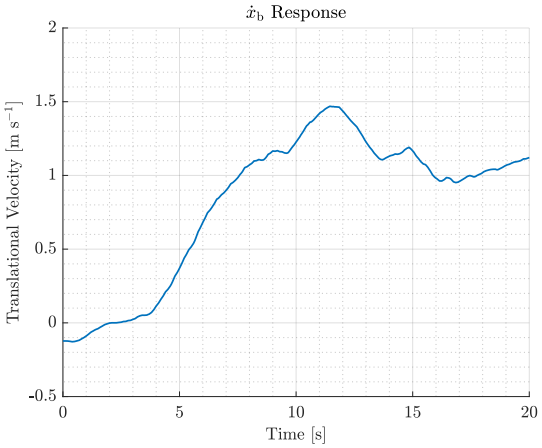


Figure 10.7: Step response in $\dot{x}_b$ with a reference of 1 m·s^{-1}.

The response in ψ shows that the controller is able to track a reference, especially in the first ten seconds. Afterwards, the vessel is disturbed due to the the security string used in the test setup, at approximately 11 seconds, and the yaw angle deviates from its references. Once the perturbation is compensated by the controller, the reference is tracked again. It is worth mentioning the oscillations seen from 14 to 19 seconds. These errors when estimating ψ possibly come from the IMU data. This can be deduced as the vessel dynamics do not allow an oscillation of 0.4 rad of approximately 1 Hz. With high probability, these oscillations are lower in the real behavior of the vessel.

The $\dot{x}_b$ response also shows a controller capable of tracking the reference. In this case, there is an overshoot in the response, and it goes to approximately 1.2 m·s^{-1}. It is important to disregard the deviation seen from 11 to 13 seconds as this corresponds to the perturbation suffered by the vessel while the test was running. After the controller recovers, it can be seen that the speed converges to the desired reference of 1 m·s^{-1}.

Due to time constraints, the performance of the outer controller could not be tested with success in the real vessel, but the implementation done in ROS suggests a working outer controller, as shown in Appendix I. In this appendix the code implemented is tested in order to check if the inner and outer controllers together as expected. The result is also shown in Figure 10.8.

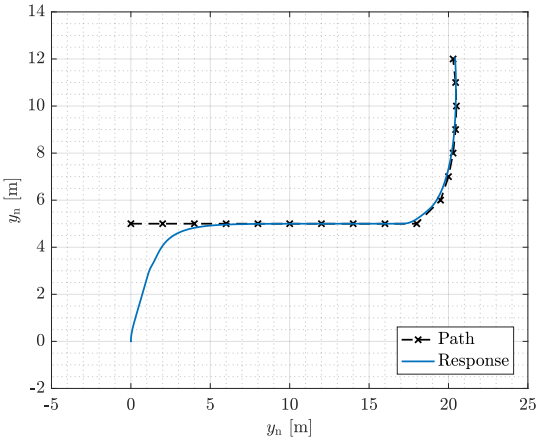


Figure 10.8: Path and response of the implemented code, both the inner and outer controllers together with the model node.

11 | Discussion

The results obtained shows that the controller requirements and the implementation requirements were sufficiently fulfilled, despite there being minor issues. Some requirements, such as **E**, were not fulfilled, but the concepts were proven.

The initial simulations indicated that the controllers performed satisfactory with regards to their respective design parameters. The LQR was shown to have a faster step response than the $\mathcal{H}_\infty$, while the $\mathcal{H}_\infty$ controller proved to be more robust towards noise and disturbances. From this it is further emphasized that the LQR is an optimal controller while the $\mathcal{H}_\infty$ controller is robust. It should be noted that the $\mathcal{H}_\infty$ design technique is known for producing a conservative controller with regards to performance.

The implementation of the controllers on the ASV was problematic. Both controllers in the real system were too aggressive and the vessel did not behave as simulated. From this it was determined that some dynamics of the system were not modeled. Even though the model was verified in section 4.6, upon further inspection it was noticed that only a constant force was applied by the motors, thus neglecting the transients of the motors. During testing a significant delay was noticed in the response time of the motors. It should also be noted that the motors have a dead zone in the lower range of the operation area. This meant it was difficult for the motors to perform minor corrections, which degraded the performance of the controller. Due to these reasonings it was decided that the system model lacks a more detailed description of the motors.

During testing it was also seen that vessel was sensitive to wave disturbances. Since the vessel has a narrow hull, it is difficult for the vessel to reject these disturbances when these affect the y_b axis. This effect was visible in Klingenberg pond where the waters can be considered calm. This is concerning as the intended use is to survey Port of Aalborg, which is assumed to have larger disturbances. Once again, it should be reiterated that the simulations proved the designed controller is capable of rejecting these larger disturbances.

The ROS implementation was deemed satisfactory as it was possible to successfully simulate the controller as described in Appendix I. Unfortunately, the Kalman filter used to estimate position did not work as intended and was not tested in ROS when simulating the controllers. It did not work due to an error in the implementation and was not corrected because of time restrictions. To overcome this problem, the implementation from [26] was used. Besides this minor error, the simulated Kalman filters estimate the attitude and position of the vessel when the measurements are subjected to the noise similar to the real system.

The RTK GPS implemented shows a 95 percentile which gives a precision well below 30 cm, when it receives base correction data. However, the base correction data in Appendix E indicates an accuracy error. This error may be due to reflections from surrounding buildings as well as the antenna on the vessel being highly directional. While the exact position of the ASV may not be accurate, the RTK GPS consistently gives precise data. This indicates that a THU under 30 cm is achievable given a correct base setup.

12 | Conclusion

In this project, a control strategy for an ASV is developed to navigate through a given area for the purpose of taking bathymetric measurements.

The first step has been to analyze the problem and which requirements are needed for the system. These include requirements for the robustness and precision of the controller as well as for the final implementation. The analysis also contains a dynamic model of the system to be used in the control design, which describes the behavior of the vessel.

Then, the control strategy has been divided into two cascaded controllers.

The inner controller is in charge of controlling the velocity and heading of the vessel, and it has been designed using and comparing two different approaches. The first approach has been an LQR, which is based on optimizing a cost function that includes the inputs and the convergence of the states. The second approach has been done using $\mathcal{H}_\infty$ theory, and it has been used to design a controller robust against disturbances and noise. Both controllers have been simulated and compared to analyze their performance.

In the area to survey, the path is generated by calculating the waypoints needed to cover it. Then, the outer controller calculates, using an enclosure based steering algorithm, the heading and speed references to send to the inner controller, such that the vessel follows the path. As in the case of the inner controller, its performance has also been analyzed though simulations that include disturbances, noise and varying parameters.

Finally, an estimator has been designed to fuse the data from both the IMU and the GPS. It consists of two Kalman filters, one for attitude and one for position, that estimate the needed variables for the controller to work such as heading, speed and position. The estimator has then been tuned and tested through simulation to check its performance.

Even though the simulated results have not been fully replicated in the real vessel, they show a promising behavior of the control system.

13 | Future Work

The main problem, experienced during the project, was not being able to replicate the simulated controllers on the ASV. A possible solution to this problem could be to include a better model of the motors in the system. This can be done by accurately calculating the nonlinear force to PWM curves of the thrusters. Recommended tests should include measuring not only the forward and backward forces but also the forces measured when turning. This happens when one thruster has a forwards force while the other has a backwards force.

Another solution could be to model the system more comprehensively by including more characteristics of the vessel. For example, nonlinear forces such as the Coriolis forces could be included. By including more nonlinear terms another recommendation is to design a nonlinear controller as it could improve the performance of the controller by reacting better to these nonlinear terms.

To be able to sufficiently reject disturbances in rougher waters, such as Port of Aalborg, there should be a redesign of the vessel by increasing the width of the hull, thus allowing the thrusters to be positioned further apart. This should improve the turning capabilities of the vessel and may allow the vessel to be more effective at rejecting these disturbances.

To ensure the RTK GPS gives an accurate THU below 30 cm, further testing should be done on both the base and rover modules to ensure that the base correction data does not give an error in position. Additionally the base station should be moved to a new location, which is not in close proximity to any buildings, such that the base does not experience issues with multipathing.

Bibliography

[1] M. Ani Hsieh, Oussama Khatib, and Vijay Kumar. *Experimental Robotics, The 14th International Symposium on Experimental Robotics*. 2016.

[2] Farbod Fahimi. *Autonomous Robots Modeling, Path Planning and Control*. 2009.

[3] Jimin Zhang et al. "Flooding Disaster Oriented USV & UAV System Development & Demonstration". In: *OCEANS 2016 - Shanghai* (Apr. 10, 2016).

[4] *National Ocean Service - How is Bathymetric Data Used*. Feb. 26, 2015. URL: http://oceanservice.noaa.gov/facts/bathyuses.html (visited on Mar. 30, 2017).

[5] Atsushi Watanabe, Miwa Kuri, and Keiji Nagatani. "Field Report Autonomous Lake Bed Depth Mapping by a Portable Semi-submersible USV at Mt. Zao Okama Crater Lake". In: *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (Oct. 23, 2016).

[6] Matthew Dunbabin, Alistair Grinham, and James Udy. "Field Report Autonomous Lake Bed Depth Mapping by a Portable Semi-submersible USV at Mt. Zao Okama Crater Lake". In: *Australasian Conference on Robotics and Automation (ACRA), December 2-4, 2009, Sydney, Australia* (Dec. 2, 2009).

[7] Pradya Prempraneerach. "Trajectory Tracking using Sliding Mode Control for Autonomous Surface Vessel". In: (2016).

[8] Yang Yang et al. "A Trajectory Tracking Robust Controller of Surface Vessels with Disturbance Uncertainties". In: *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY* (2013).

[9] Mahmut Reyhanoglu and Ard Bommer. "Tracking Control of an Underactuated Autonomous Surface Vessel Using Switched Feedback". In: (2006).

[10] Sean Kragelund et al. "Adaptive Speed Control for Autonomous Surface Vessels". In: ().

[11] Internal Hydrographic Bureau. *IHO STANDARDS FOR HYDROGRAPHIC SURVEYS (S-44)*. 2008.

[12] Canadian Hydrographic Service. *STANDARDS FOR HYDROGRAPHIC SURVEYS*. 2013.

[13] *Multibeam Echo Sounder SeaBat 7125*. Apr. 4, 2017. URL: http://www.teledyne-reson.com/products/echo-sounder-seabat/multibeam-seabat-7125/.

[14] *AAUSHIP*. Feb. 21, 2017. URL: http://www.auv.aau.dk/index.php?n=Main.AAUSHIP.

[15] *Arduino MEGA 2560*. Feb. 21, 2017. URL: https://www.arduino.cc/en/Main/arduinoBoardMega2560.

[16] *Eee PC 1000H*. May 11, 2017. URL: https://www.asus.com/us/Laptops/Eee_PC_1000H.

[17] *Ubuntu 14.04.5 LTS (Trusty Tahr)*. May 11, 2017. URL: http://releases.ubuntu.com/14.04.

[18] *Robot Operating System (ROS)*. Feb. 21, 2017. URL: http://www.ros.org/.

[19] *INLINE 750 14,8 V - Graupner*. Feb. 21, 2017. URL: https://www.graupner.com/INLINE-750-14-8-V./6608/.

[20] *Brushles Control +T70 G3,5*. Feb. 21, 2017. URL: https://www.graupner.com/BRUSHLESS-CONTROL-T70-G3-5/33770/.

[21] *Data Sheet ADIS16405 - Triaxial Inertial Sensor with Magnetometer*. Feb. 21, 2017. URL: http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16405.pdf.

[22] *Original authentic ADIS16405BMLZ MODULE GYRO/ACCEL/MAG 24M Sensors*. Feb. 21, 2017. URL: https://www.aliexpress.com/item/Original-authentic-ADIS16405BMLZ-MODULE-GYRO-ACCEL-MAG-24M-Sensors-Transducers-Multifunction-ADIS16405-Free-shipping/32389942714.html?spm=2114.40010408.3.1.bJVySu.

[23] Emlid. *How RTK Works*. Apr. 27, 2017. URL: https://docs.emlid.com/reach/common/tutorials/rtk-introduction/ (visited on May 4, 2017).

[24] Emlid. *Emlid Docs*. Apr. 27, 2017. URL: https://docs.emlid.com/reach/ (visited on May 4, 2017).

[25] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. First Edition. 2011. ISBN: 978-1-119-99149-6.

[26] Jeppe Dam and Nick Østergaard. "Formation Control of Autonomous Surface Vehicles for Surveying Purposes". MA thesis. Aalborg University, Jan. 8, 2015.

[27] Desineni Subbaram Naidu. *Optimal Control System*. 2003.

[28] Chi-Tsong Chen. *Linear System Theory and Design*. 3rd ed. 1999.

[29] *MathWorks: MATLAB lqrd*. May 20, 2017. URL: https://se.mathworks.com/help/control/ref/lqrd.html.

[30] Jerome Le Ny. *Linear-Quadratic Optimal Control: Full-State Feedback*. May 19, 2012. URL: http://www.professeurs.polymtl.ca/jerome.le-ny/teaching/DP_fall09/notes/lec4_LQR.pdf (visited on Mar. 24, 2017).

[31] John C. Doyle et al. "State-Space Solutions to Standard H2 and Hinf Control Problems". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 34, NO. 8, AUGUST 1989* (1989).

[32] A. A. Stoorvogel. *The Hinf control problem: a state space approach*. 2000.

[33] Steen Tøffner-Clausen, Palle Andersen, and Jakob Stoustrup. *Robust Control*. U96-4153. Aalborg Universitet, 2006.

[34] M. Salari Khaniki and M. J. Khosrowjerdi. "Multiobjective H2/Hinf Control Design for a VSTOL Flight Model". In: *Proceedings of ICEE 2010* (2010).

[35] Simon Haykin. *Kalman Filtering and Neural Networks*. 2001. ISBN: 978-0-471-36998-1.

[36] M. Bibuli et al. "Unmanned Surface Vehicles for Automatic Bathymetry Mapping and Shores' Maintenance". In: *OCEANS 2014 - TAIPEI* (2014).

[37] Novatel. *Real-Time Kinematic (RTK)*. May 25, 2017. URL: https://www.novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk/ (visited on May 4, 2017).

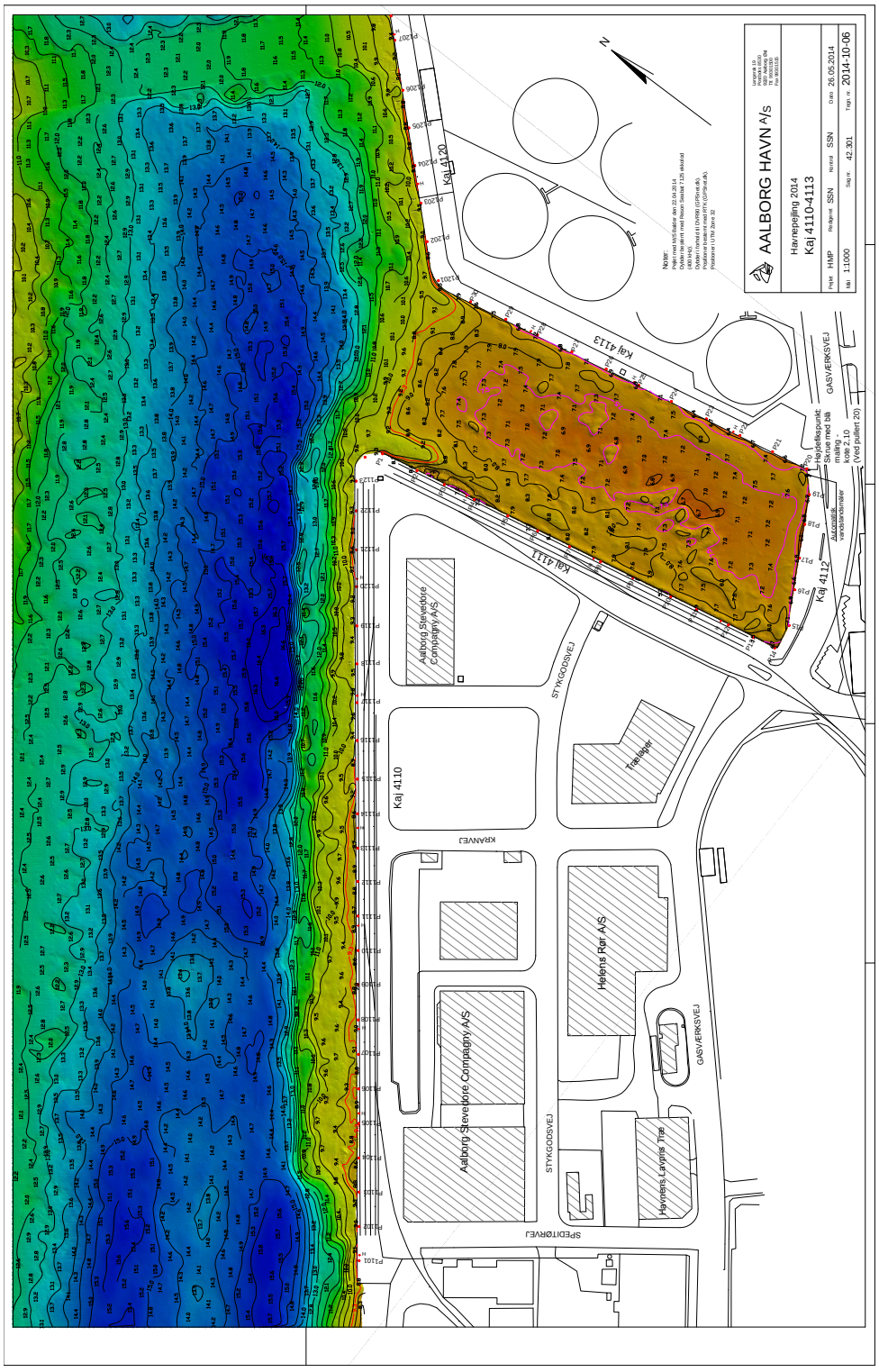
[38] e-education. *Real-Time Kinematic and Differential GPS*. May 25, 2017. URL: https://www.e-education.psu.edu/geog862/node/1828 (visited on May 4, 2017).

[39] What when How. *Ambiguity-Resolution Techniques (GPS)*. May 25, 2017. URL: http://what-when-how.com/gps/ambiguity-resolution-techniques-gps/ (visited on May 4, 2017).

[40] *GPS Accuracy, Errors & Precision*. May 30, 2017. URL: http://www.radio-electronics.com/info/satellite/gps/accuracy-errors-precision.php.

Appendix

A | Bathymetric Map from Port of Aalborg



B | Topic Description

Topic	Message Type	Data Type	Variable
/samples	Faps.msg	string	DevID
		string	MsgID
		string[]	Data
		float64	Time
/imu	ADIS13205.msg	float32	supply
		float32	xgyro
		float32	ygyro
		float32	zgyro
		float32	xaccl
		float32	yaccl
		float32	zaccl
		float32	xmagn
		float32	ymagn
		float32	zmagn
		float32	temp
		float32	adc
/gps_pos	RTKGPS.msg	string	timestamp
		float64	delx
		float64	dely
		float64	longitude
		float64	latitude
/lli_input	LLIinput.msg	uint8	DevID
		uint8	MsgID
		int16	Data
		float64	Time

Topic	Message Type	Data Type	Variable
/kf_attitude	AttitudeStates.msg	float64	roll
		float64	pitch
		float64	yaw
		float64	rolld
		float64	pitchd
		float64	yawd
		float64	rolldd
		float64	pitchdd
		float64	yawdd
/kf_position	PositionStates.msg	float64	xn
		float64	yn
		float64	xbd
		float64	ybd
		float64	xbdd
		float64	ybdd
/control_reference	Ref.msg	float32	speed
		float32	yaw

C | Base Station Implementation and Usage

This appendix contains a description how the RTK GPS base station is implemented and how to use it.

RTK GPS Basics

The inaccuracies of GPS receivers are mostly due to atmospheric changes, which add a verity of disturbances dependent on the weather, temperature etc. The most significant error to be corrected is the ambiguity resolution. Ambiguity resolution is a fixed offset that the GPS receiver experiences as the number of wavelengths from the satellite to the receiver is unknown. This results in a integer which needs to be resolved to improve the accuracy of the GPS measurements. If the ambiguity resolution is not found, the gained precision from other sources is mostly irrelevant due to the bias. [37] [38] [39]

Most modern GPS receivers have a precision of 2-5 m. For applications where this precision is insufficient, an RTK GPS can be used. By comparing the measurements received from a base station, the rover is able to compensate for these disturbances. The base station is a stationary GPS receiver with a known position, which streams correction data, to be used by the rover. The increased precision gained from including a base station is done by assuming the position errors are correlated. As long as the rover remains within a close distance to the base station, this is a good assumption. The rover uses the correction data received form the base to estimate errors. This results in the position being improved by orders of magnitude. The further the rover gets from the base, the more the noises start to become uncorrelated, resulting in a loss in accuracy. [23] [38]

Base Station Implementation and Usage

The correction data is used by the rover to estimate signal disturbances, caused be the signals entering the atmosphere. This enables it to increase the precision of its GPS measurements. This data is formatted as a RTCM3 message, which is a protocol designed for this purpose.

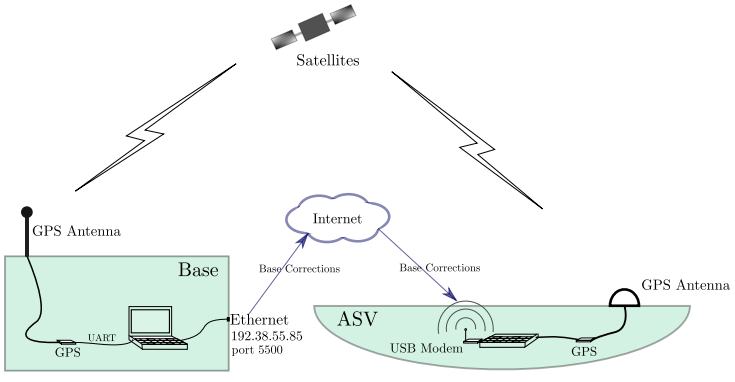


Figure C.1: Overall setup of the GPS.

Figure C.1 shows how the RTK gps system is setup. The computer forwards the message to a TCP socket, making it accessible through the Internet. The Reach is able to act both as a base station and as a rover depending on the settings. The settings pane is accessed by typing the IP of the module in a browser, which connects to a GUI server hosted by the module after initialization. In order to get the

best possible measurements, the base is set to "static mode" in "RTK settings". The computer runs a server capable of reading the serial data, and forwards in the correct package sizes.

The correction data from the base station consists of different message types, varying in length.

Each message contains header, data and a checksum. The header contains information on message type and packet size, which is used to forward the right amount of bytes each time. Through testing, it has been found that the header is initiated with a hex value of d3, indicating that a new package begins. As the data is transmitted binary, this pattern is not a guarantee that a message is transmitted. To ensure the correct package sizes is sent, an initialization sequence is ran on the server at startup, which searches for the initialization sequence, finds the packet size, then reads that amount of bytes, until a package is received with that length, followed by another start sequence. This can be validated using the checksum, however this feature is yet to be implemented.

The package size is obtained by locating the initialization sequence to find the header containing the package size. The base station is implemented to cast the RTCM3 message on a TCP socket on IP: 192.38.55.85 Port: 5500. Users with access to the Internet are able to access this socket from anywhere. Under the "Correction Input" pane, the method of input can be selected, if the module has Internet connection, these can be acquired by selecting the "TCP" and client mode, and typing in the IP and Port Number in their respective fields.

The Reach RTK is setup using the ReachView app, which is accessed as a html page, hosted by the reach itself. Throughout this project ReachView v. 2.3 is used. Under the RTK settings page following changes have been applied:

- Ambiguity Resolution (AR) mode is set to Fix and Hold
- GNSS selection is: GPS+Glonass+SBAS+QZSS
- SNR mask: 15
- Elevation Mask Angle: 15

Under Correction input, the following settings are used:

- TCP is selected for input
- Role is Client
- IP: 192.38.55.85
- PORT: 9000

These settings are only viable in cases where the Emlid Reach has access to the Internet. This will connect the rover to the base station, such that the correction data is received. In cases where it is not possible for the Reach to access the Internet directly, the GPS node is able to forward the correction input through USB. In this case following settings are used:

- USB-To-PC is selected for input
- Baud Rate: 115200

Appendix

D | Test Journal: IMU Variances

Date: 2017/04/06

Purpose

Find the variance of the accelerometer and the variance and bias of the gyroscope present in the IMU, as well as the variance when calculation the attitude using the accelerometer and the magnetometer as described in section 8.1 in Equation 8.6, 8.6 and 8.8.

Equipment

- Vessel with all its components.
- External laptop.

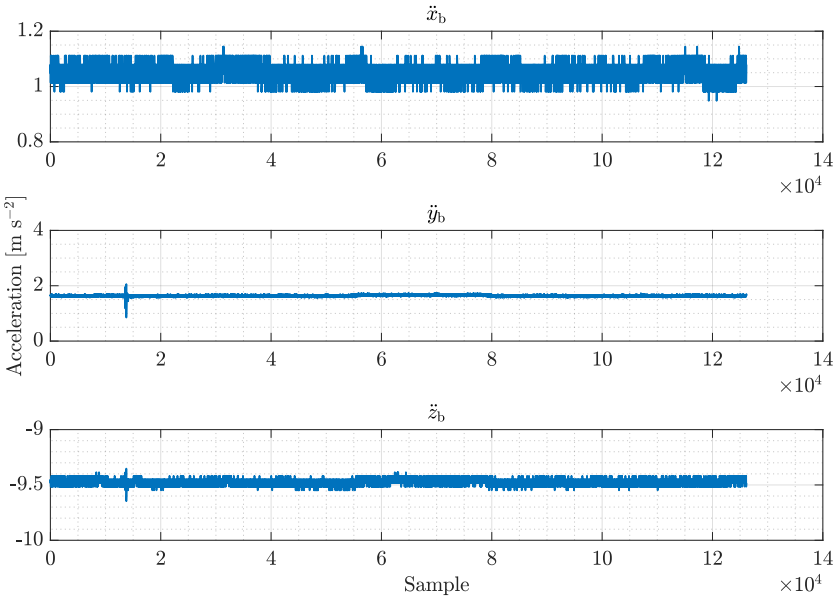
Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/lli_node`
    - `/sensor_node`
4. Leave the vessel in a fixed position and orientation.
5. Record the following topics
    - `/imu`
6. Stop the recording after some time has passed.
7. Turn off the equipment.
8. Process the data.

Results

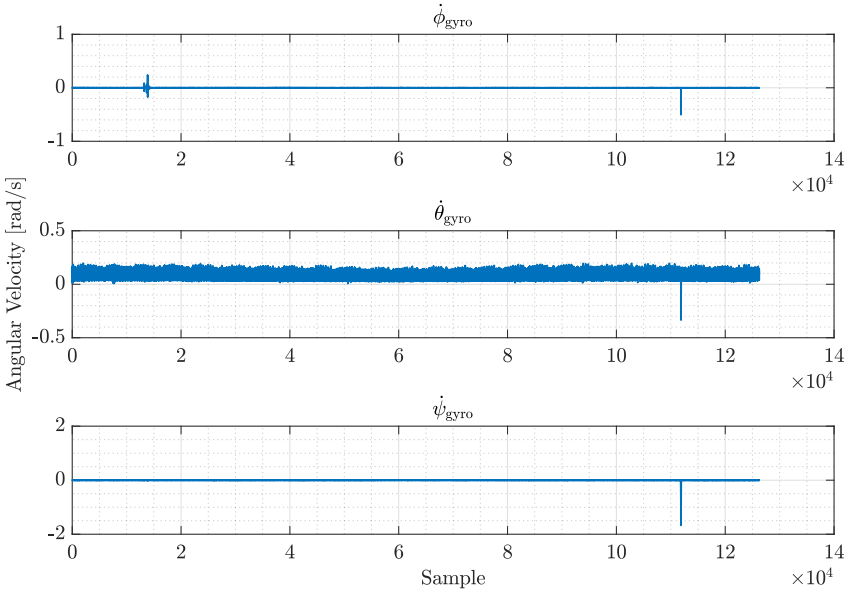
The resultance variance for each measurement can be seen below.

Accelerometer



$$\sigma^2_{\ddot{x}\mathrm{b,acc}} = 0.00050346 \text{ m}^2 \cdot \text{s}^{-4}$$
$$\sigma^2_{\ddot{y}\mathrm{b,acc}} = 0.00057036 \text{ m}^2 \cdot \text{s}^{-4}$$
$$\sigma^2_{\ddot{z}\mathrm{b,acc}} = 0.00043887 \text{ m}^2 \cdot \text{s}^{-4}$$

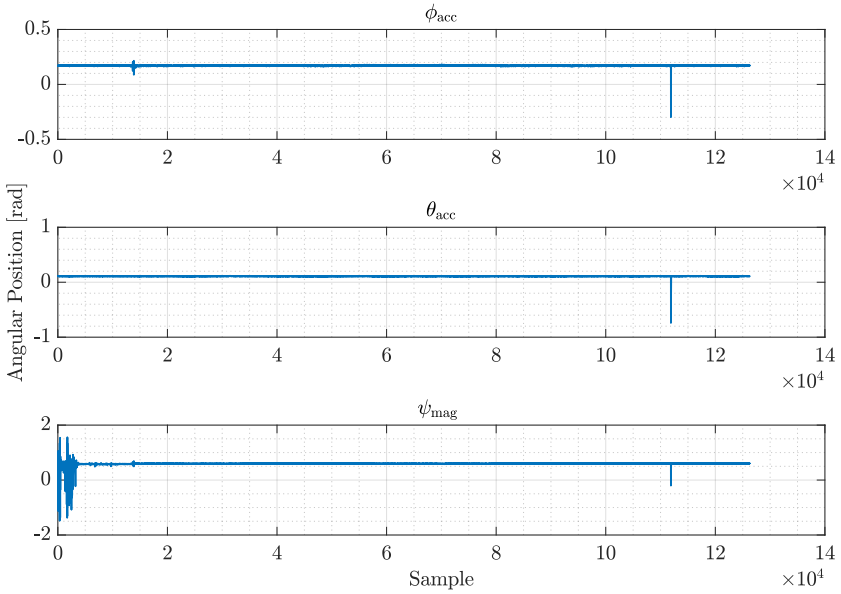
Gyroscope



$$\sigma^2_{\phi,\text{gyro}} = 0.0000103 \text{ rad}^2 \cdot \text{s}^{-2}$$
$$\sigma^2_{\dot{\theta},\text{gyro}} = 0.00120 \text{ rad}^2 \cdot \text{s}^{-2}$$
$$\sigma^2_{\dot{\psi},\text{gyro}} = 0.0.00007606 \text{ rad}^2 \cdot \text{s}^{-2}$$

Attitude Calculation

The attitude calculation is done using Equation 8.6, 8.7, 8.8 and the measurements from the accelerometer and magnetometer.



$$\sigma^2_{\phi,\text{acc}} = 0.00001165 \text{ rad}^2$$
$$\sigma^2_{\theta,\text{acc}} = 0.00002264 \text{ rad}^2$$
$$\sigma^2_{\psi,\text{mag}} = 0.00779021 \text{ rad}^2$$

E | Test Journal: GPS Performance

Date: 2017/05/24

Purpose

To determine the effectiveness of including base corrections when solving the location of the GPS. Additionally the variance of the GPS is found for the case where the GPS is getting RTK corrections with integer solution, as described in Appendix C.

Equipment

- Emlid Reach RTK GPS connected to the base station as described in Appendix C
- Water proof cover.
- Active directional GPS antenna with MCX connector.
- HTC Desire HD smartphone with access to a cellular network.

Theory

By comparing the mean and variance of a GPS with and without base correction, the theoretical improvement in precision and accuracy by including base corrections can be shown.
The Emlid Reach module has three different operational modes when receiving corrections from an RTK base. The *Single mode* is when it does not rely on the base. In this mode, it works as a regular GPS module. In *RTK Float mode*, the module is receiving corrections from the RTK base, but is not using a reliable fix for the integer solution. The last mode is *RTK Fix mode*, where it receives RTK correction data with reliable fix for the integer solution. Ideally, it should always operate in *RTK Fix mode* to map out Port of Aalborg within the requirements.
Preferably, the correct position of the GPS should be recorded over at least 24 hours by placing the GPS in *Single mode*. In this way, the average includes a lot of different satellite configurations and no potential bias is added by a base.
If the found location is assumed to be correct the accuracy of the proceeding tests can be evaluated with respect to this.
For practical reasons, this was not prioritized in the project. A 5 hour test was instead performed including all three modes in which the GPS can operate. This is a startup process that the GPS goes through before it receives base corrections and eventually finds a fix for the integer solution.
From these results the precision, relevant for the control system, is accessed.

Procedure

1. Set up the Emlid Reach to log its position in NMEA format.
2. Connect the antenna to the Emlid Reach.
3. Turn on hotspot on the phone (this should be known to the Emlid Reach module).
4. Through ReachView connect the Emlid Reach to the base station.
5. Waterproof the setup with the cover.

6. Place the GPS and phone with access to a power outlet and unobstructed view.
7. Wait five hours.
8. Retrieve the setup and extract the data from the logs recorded on the Emlid Reach.

Results

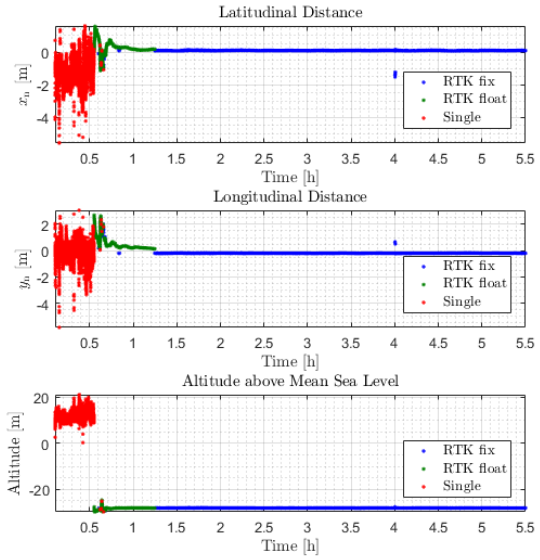


Figure E.1: All three modes of operation are shown here. The Emlid Reach normally goes through all three at startup. However in this case it took longer than observed during testing.

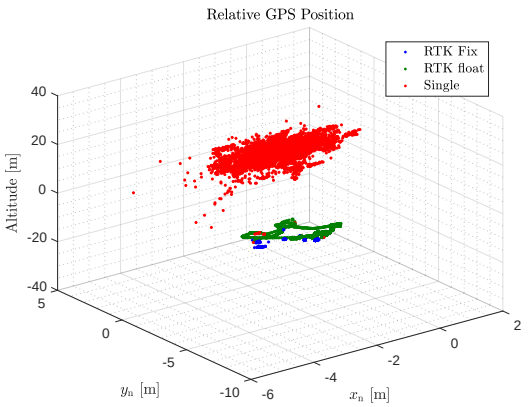


Figure E.2: This is the data from Figure E.1 plotted in the three dimensions. The x_n and y_n are positions relative to the mean of the Single mode measurements while the altitude is distance above mean sea level.

In Figure E.1 and E.2, the jump in altitude is presumed to be caused by the base although, given the limited duration in which single measurements are performed, no reliable conclusion can be drawn. It does however indicate the need for further testing.
If the offset is due to bias from the base this can be caused by signal multipath [40].
Note that the RTK fix solution seems very precise compared to the other modes. In the project, this is the mode in which the RTK GPS is presumed to operate. For this reason, this data is more closely analyzed.

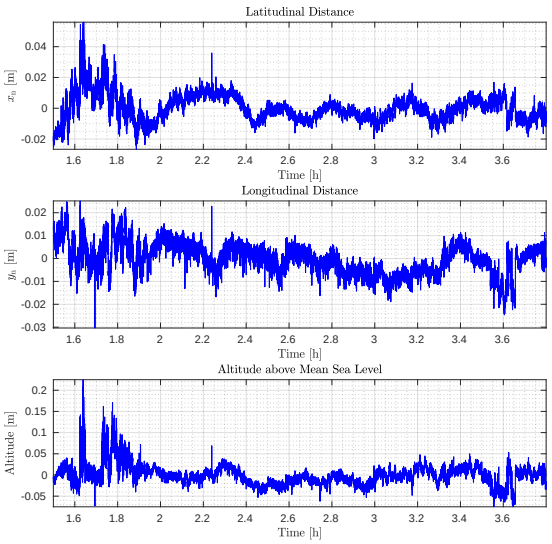


Figure E.3: Here all the data is recorded in *RTK Fix mode* and shown relative to its own mean to access the precision of the measurements.

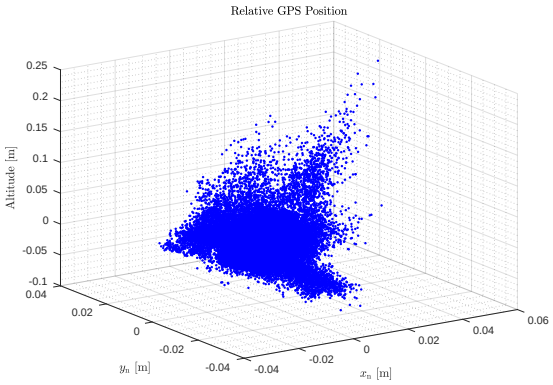


Figure E.4: This is the data from Figure E.3 shown in the three spacial dimensions.

In Figure E.3 and Figure E.4 the data recorded in *RTK Fix mode* is isolated and plotted around its mean. The variances extracted from Figure E.3 are used in the design of the position Kalman filter in section 8.2. These are

$$\sigma^2_{x_{\mathrm{n,GPS}}} = 0.00004395 \ , \tag{E.1}$$

$$\sigma^2_{y_{\mathrm{n,GPS}}} = 0.00007283 \ . \tag{E.2}$$

The following shows the distribution and presents analysis of the precision of the measurements.

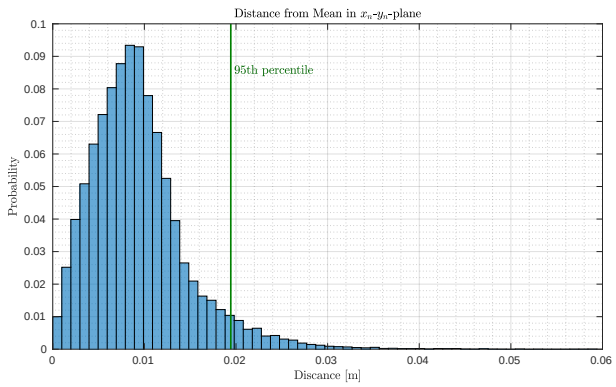


Figure E.5: A histogram of the distance from each point to the mean of the RTK fix measurements in the x_{n}-y_{n}-plane. The amount of measurements in each bin are scaled as probabilities.

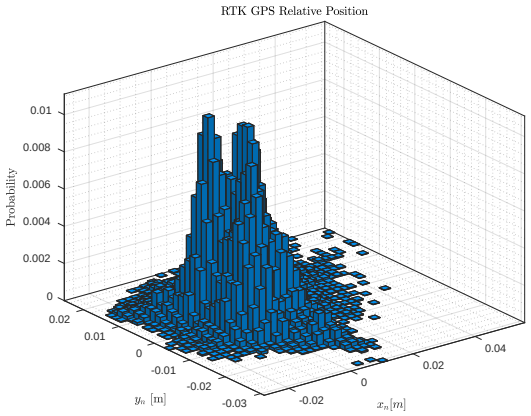


Figure E.6: A histogram of the RTK fix measurements in the x_n-y_n-plane. The amount of measurements in each bin are again scaled as probabilities.

In Figure E.6 the distribution of RTK fix measurements in the x_{n}-y_{n} plane relative to their mean is forming a Gaussian distribution. To verify the precision of the measurements shows the planar distances from the data to their mean. The 95th percentile is indicated to show that 95% of the data has a precision well within the requirement, **E**, *The THU shall not exceed 30 cm with a 95% confidence interval*, as stated in section 2.3.

For this requirement to be fulfilled, not only the precision but also the accuracy be evaluated. Due to the short duration of the test, no reliable conclusion can be drawn. However, there is some indication that the base might cause a bias, see Figure E.3, and that reflections from nearby buildings might play a role. If further testing supports these claims, the base can be moved to a more appropriate location with no immediate obstructions of the signals.

F | Test Journal: Force-PWM Relation

Date: 2017/05/10

Purpose

The purpose of this test journal is to find the relationship between the command sent to the LLI and the force that the propellers apply.

Equipment

- Vessel with all its components.
- External laptop.
- Analog newton meter, AAU number 02054-03.

Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/lli_node`
    - `/keyboard_teleop_node`
4. Send a PWM command to the LLI and measure the resultant force.
5. Move the vessel using the keyboard.
6. Repeat previous step with other commands, both positive and negative.
7. Process the data

Results

The relation between force and PWM command depends on the sign of the force, as can be seen in Figure F.1 and F.2.

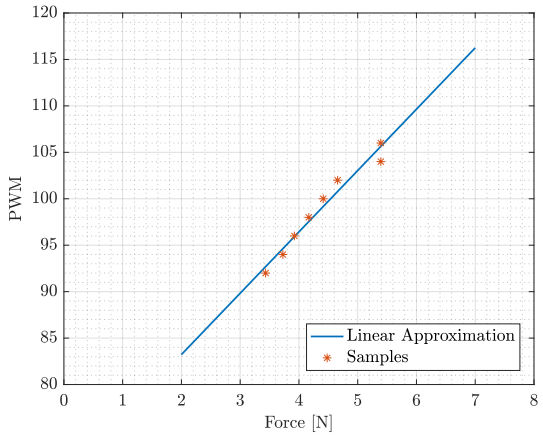


Figure F.1: Plot of the relation between positive force and PWM command.

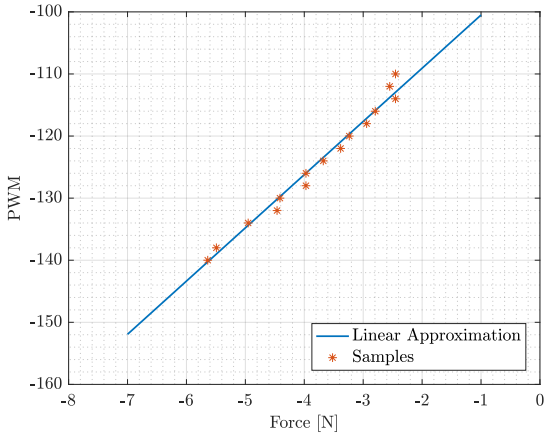


Figure F.2: Plot of the relation between negative force and PWM command.

For positive forces, the relation is as follow

$$\text{PWM} = 6.6044\ F + 70.0168\ \ . \tag{F.1}$$

For negative forces, the relation is as follows

$$\text{PWM} = 8.5706\ F - 91.9358\ \ . \tag{F.2}$$

G | Test Journal: Model Verification

Date: 2017/05/11

Purpose

The purpose of this test journal is to get real measurements to verify the model, when moving the vessel manually in the water.

Equipment

- Vessel with all its components.
- External laptop.

Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/lli_node`
    - `/sensor_node`
    - `/gps_node`
    - `/keyboard_teleop_node`
4. Record the following topics
    - `/imu`
    - `/gps_pos`
    - `/lli_input`
5. Move the vessel using the keyboard.
6. Stop the recording.
7. Bring the vessel back to land and turn off the equipment.
8. Process the data

Results

The results when applying two constant but different forces ($F1 = -1$ N and $F1 = 12$ N), which produces a counterclockwise turn, can be seen inFigure G.1 and G.2.

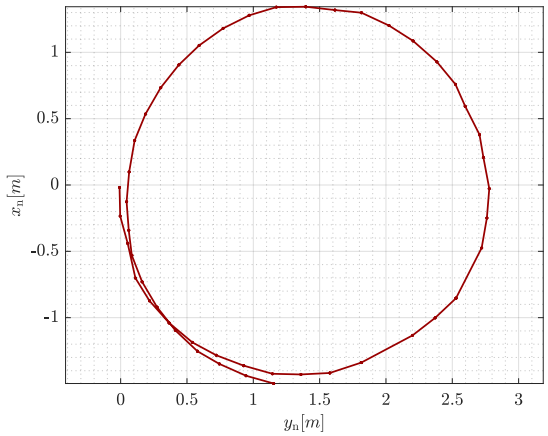


Figure G.1: Position of the vessel in the x_n-y_n plane given by the GPS data.

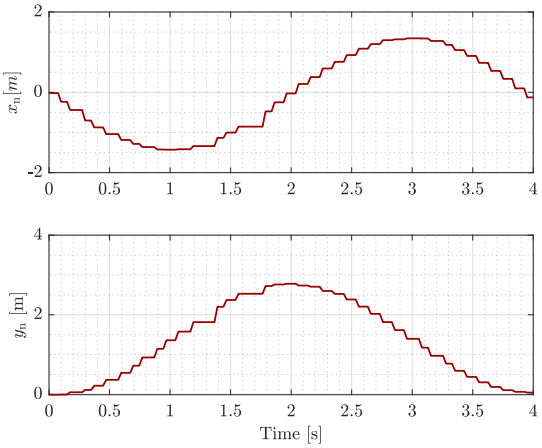


Figure G.2: x_n and y_n with respect to time.

H | Test Journal: Disturbance Frequency

Date: 2017/05/10

Purpose

Find the frequency of the wave forces that affects the vessel. The wave forces can be modeled as a sinusoidal curve with a given frequency, unlike wind forces which are modeled as a constant force. To do this the vessel is left stationary in water and the effects of the waves are recorded. The frequency of the waves are determined by processing the data.

Equipment

- Vessel with all its components.
- External laptop.

Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/sensor_node`
4. Place the vessel in water without actuating the vessel.
5. Record the following topics
    - `/imu`
6. Stop the recording after some time has passed.
7. Turn off the equipment.
8. Process the data.

Results

The Fast Fourier Transform (FFT) of x_{acc} was performed, as shown in Figure H.1 thus the frequency of the disturbances can be seen as $f = 0.833$ Hz.

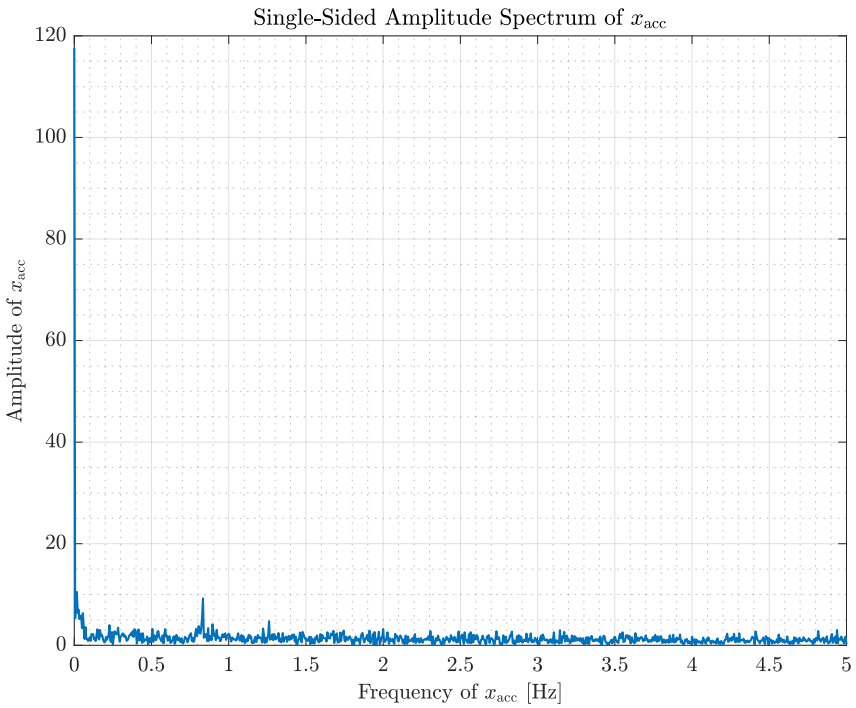


Figure H.1: FFT of x_{acc}

I | Test Journal: Controllers Implementation

Date: 2017/05/25

Purpose

Check the performance of the inner and outer controller code, when tested with a ROS node that simulates the dynamics of the vessel.

Equipment

- Vessel with all its components.
- External laptop.

Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/lqr_node`
    - `/model_node`
4. Record the following topics
    - `/kf_attitude`
    - `/kf_position`
    - `/lli_input`
5. Run the following topic with a step in $\dot{x}_\mathrm{b}/\psi$.
    - `/controller_node`
6. Stop the recording after some time has passed.
7. Repeat from 3 but including the `/path_follower_node`.
8. Process the data.

Results

The result of the implemented code are shown below

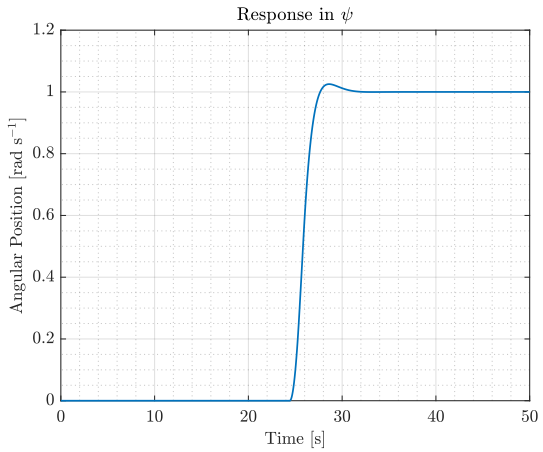


Figure I.1: Step response in ψ of the implemented controller code for the inner controller together with the model node.

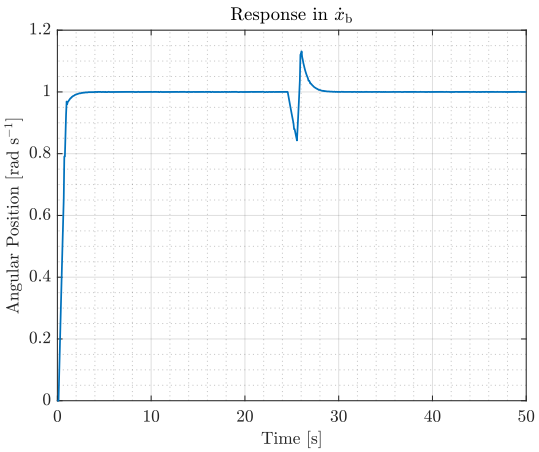


Figure I.2: Step response in $\dot{x}_b$ of the implemented controller code for the inner controller together with the model node.

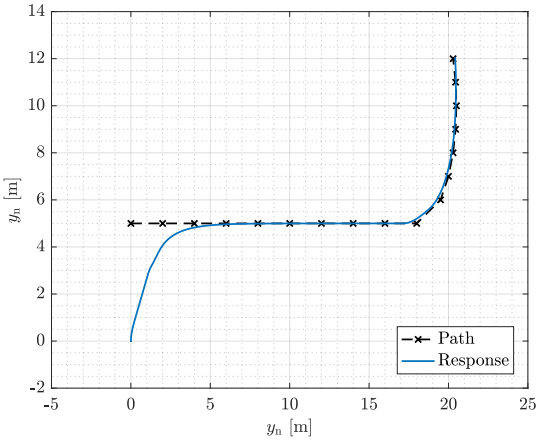


Figure I.3: Path and response of the implemented code, both the inner and the outer controllers together with the model node.

As it can be seen, both ψ and $\dot{x}_b$ converge to the reference when using the inner controller and the model follows the path when using the outer one. This proves that both controller nodes work and are coded properly.

J | Test Journal: Inner Controller

Date: 2017/05/26

Purpose

Check the performance of the inner controller, when tracking references in $\dot{x}_\mathrm{b}$ and ψ.

Equipment

- Vessel with all its components.
- External laptop.

Procedure

1. Turn on all equipment.
2. Remotely log into the vessel, when both, laptop and vessel's computer, are in the same network.
3. Run the following nodes
    - `/lli_node`
    - `/sensor_node`
    - `/gps1_node` (adapted to work with a previous implementation of the Kalman filter)
    - `/KF_attitude_node`
    - `/kalmanfilter_node` (from a previous implementation)
4. Record the following topics
    - `/imu`
    - `/gps1` (from a previous implementation)
    - `/kf_attitude`
    - `/KF_States` (from a previous implementation)
    - `/lli_input`
5. Run the `/controller_node` with a reference in ψ and $\dot{x}_\mathrm{b}$.
6. Stop the recording after some time has passed.
7. Turn off the equipment.
8. Process the data.

Results

The results of the test can be seen in Figure J.1 and J.2.

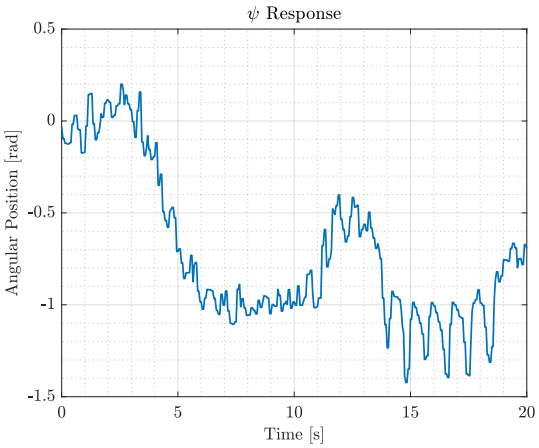


Figure J.1: Response in ψ with a reference of -1 rad.

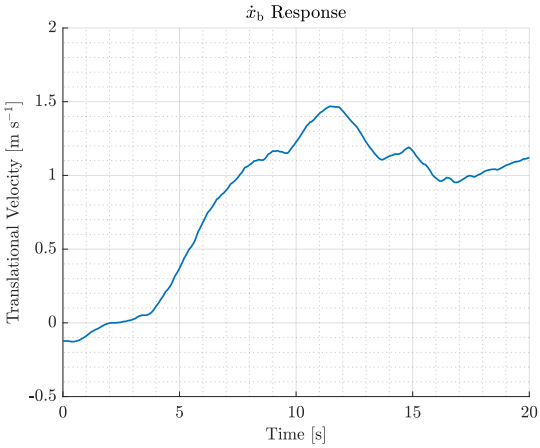


Figure J.2: Reponse in $\dot{x}_b$ with a reference of 1 m·s^{-1}.