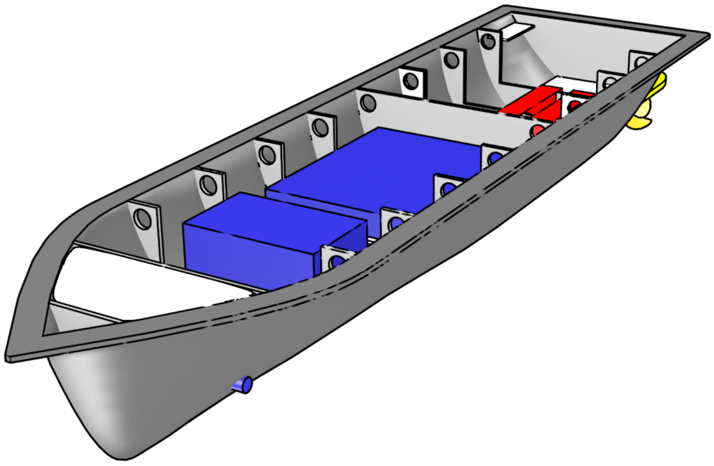
Precision Control of an Autonomous Surface Vessel



2^{nd} 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

This report is compiled in LaTeX, originally developed by Leslie Lamport, based on Donald Knuth's TeX. The main text is written in *Latin Modern* pt 12, designed by Bogusław Jackowski and Janusz M. Nowacki. Flowcharts and diagrams are made using Inkscape.

AALBORG UNIVERSITY
STUDENT REPORT

2^{nd} Semester Project

Master's Program in

Control and Automation

Department of Electronic Systems

Fredrik Bajers Vej 7C, 9220 Aalborg

Title:
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: 100
Appendices: 6
Attachments: 1

Concluded: 30/06/2017

Synopsis

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
FiXme Note: Write synopsis

Preface

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

[1]

Reading Instructions

- Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.
  [2]

Text by:

Alejandro Alonso García

Anders Egelund Kjeldal

Himal Kooverjee

Niels Skov Vestergaard

Noelia Villarmarzo Arruñada

[1] FiXme Note: Write prephase
[2] FiXme Note: Write reading instructions

Contents

Part I

Preanalysis

1 | Introduction

2 | Problem Analysis

This chapter analyses the scope of the project as well as the requirements that need to be fulfilled.

2.1 Scope of the Project

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

The controller design must be able to track references provided by the trajectory planer as well as rejecting disturbances such as possible wind or the effect of the waves. This requires to include a model of the disturbances in the controller design as well as a robust controller capable of handling model uncertainties.

The trajectory planer needs to be able to design a route in the form of waypoints, to send to the controller, to reach all the position needed to perform the different measurements required for the survey.

One important aspect to be taken into account is the precision of the route tracking, that should be below 10 cm. This requires a positioning system able of providing position data with more precision that the GPS module mounted on the vessel. A test that shows the distribution of the sample from this GPS can be seen in [1] For that reason, an RTK GPS is more suitable to enhance the precision of position data.

2.2 Requirements

To be able to design a working product some functional requirements need to be set and verified at the end of the project once all the design has been carried out.

- tracking of reference position with a precision below 10 cm

-

[1] FiXme Note: Refer to test with GPS to show it is not enough

3 | System Description

The Surface Vessel at hand is a complex system composed by several subsystems. These are shown in Figure 3.1, in which the link between the different subsystems is also depicted.

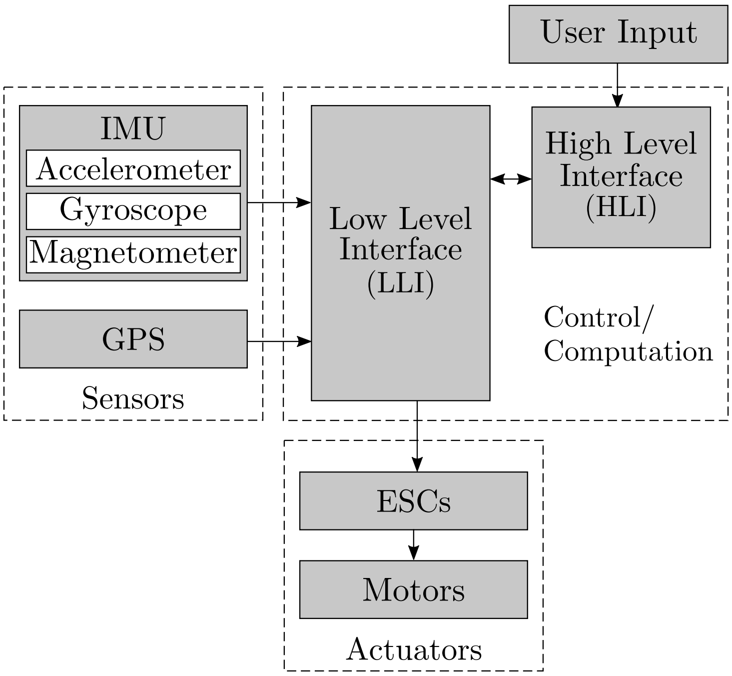


Figure 3.1

The main parts of the system are the actuation, the sensors and the control units, see Figure 3.1. The control unit gathers information from the sensors, in particular, the IMU and the GPS. This is done by the Low Level Interface which then sends it to the high level interface where the control algorithms run. The calculated actuation action is then sent back to the Low Level Interface that sends the command to the actuation system. The ESCs calculate then the required signal to make the motors turn at the right speed.

This chapter describes briefly the main components of the surface vessel used in this project. [1]

3.1 Actuators

The actuation system present on the vessel is constituted by the thrusters. These can be divided into two forward thrusters and two side thrusters.

3.1.1 Forward Thrusters

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

$$F(n_{\text{PWM}}) = 0.2567 \cdot n_{\text{PWM}} - 22.83. \tag{3.1}$$

They can also rotate in the opposite direction, producing a backwards force that can be calculated as

$$F(n_{\text{PWM}}) = 0.2567 \cdot n_{\text{PWM}} - 22.83. \tag{3.2}$$

These equations have been obtained through experimental tests in described in [2].

The forward 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. [3]

Figure 3.2: INLINE 750 14.8 V motor used to produced the forward thrust in the surface vessel [3].

In order to have the motors turning to the desired rotational speed, electronic speed controllers (ESCs) +70 G3.5 from Graupner are used. The operating voltage range goes 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 32kHz PWM signal to the speed controller. [4]

Figure 3.3: Speed controllers +70 G3.5 used to control the forward thrusters in the surface vessel [4].

3.1.2 Side Thrusters

The side thrusters move the vessel sideways but their influence in its motion is limited to low speed maneuvers and they are intended for fine positioning of the vessel. For this reason, they are not utilized in the controller design for the vessel.

3.2 Sensors

The control system designed in the vessel requires the presence of sensor data that provide information about the vessel's motion. These are an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS).

3.2.1 IMU

The Inertial Measurement Unit installed in the vessel is formed by a triaxial gyroscope with digital range scaling between ±75°/sec, ±150°/sec or ±300°/sec, a triaxial accelerometer with a range of ±18 g and a triaxial magnetometer with a range of ±2.5 gauss. It also contains SPI-compatible serial interface to be able to obtain the data. [5]

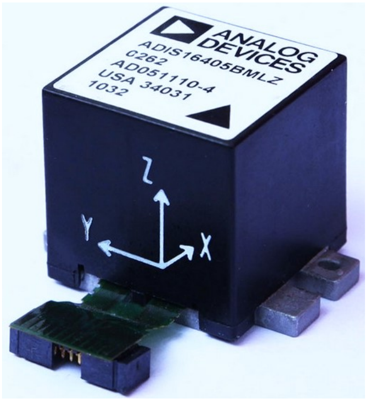


Figure 3.4: ADIS16405BMLZ IMU module mounted in the vessel [6]

The data provided by the IMU is used to estimate both the position and the attitude of the vessel and is extracted in the Low Level Interface though SPI serial communication.

3.2.2 GPS

The vessel has a UP-501 GPS Receiver installed. It operates with a update frequency up to 10 Hz and its trasmits the received position data through a serial communication to the Low Level Interface. [7]

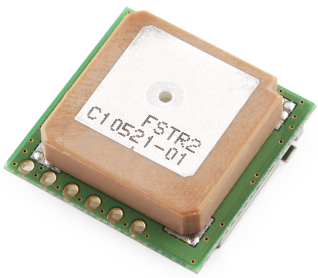


Figure 3.5: UP-501 GPS receiver mounted in the vessel [7].

As the UP-501 is a standard GPS receiver, its precision is in the range of meters [1]. This makes it cumbersome to control the position of the vessel using only GPS data, and thus, the GPS and the IMU data is combined to obtained the position of the vessel.

[1] FiXme Note: How to prove this?? Maybe make a test.

3.3 Control System

[2]The control system takes the sensor data and computes the required actuation output according to the desired motion for the vessel. It is structured in two entities that run in different devices. These entities are the Low Level Interface and the High Level Interface.

3.3.1 Low Level Interface

The Low Level Interface (LLI) implemented on the vessel runs in a Arduino Mega Development board with the Atmel microcontroller ATMEGA 2560. It is in charge of extracting the sensor data from the IMU and the GPS and send it to the HLI that runs in 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 handles also the actuators as it receives the rotational speed command for the thrusters from the HLI and calculates the appropiate PWM so the ESCs make the motors turn at the desired speed.

3.3.2 High Level Interface

The High Level Interface (HLI) takes care of the control, sensor fusion and trajectory planning algorithms. This includes communicating with the LLI to get sensor data and send commands to the thrusters.

The HLI is implemented in a ASUS Eee Computer [8] that runs with Ubuntu 14.04. 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 due to its structure of nodes an topics. Besides communication between tasks, it also includes multiple tools and capabilities useful for trajectory planning, computer vision and others [9].

[2]FiXme Note: Check the title os the section

4 | System Model

[1]

4.1 Background

The model of the surface vessel is based in the hydrodynamic modeling presented in [10], where the principal effects that need to be taken into account are described.

In this section, the different contributions are summarized and related to the vessel at hand.

4.1.1 Reference Frames

To describe the orientation and position of the surface vessel two coordinate frames are used, the body frame and an inertial frame. For operations in a local area, with longitude and latitude approximately constants (flat navigation) a NED system (North-East-Down) can be assumed as an inertial frame where Newton's physics will apply [10, p. 17].

To distinguish between the two frames, the body frame is denoted by 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.

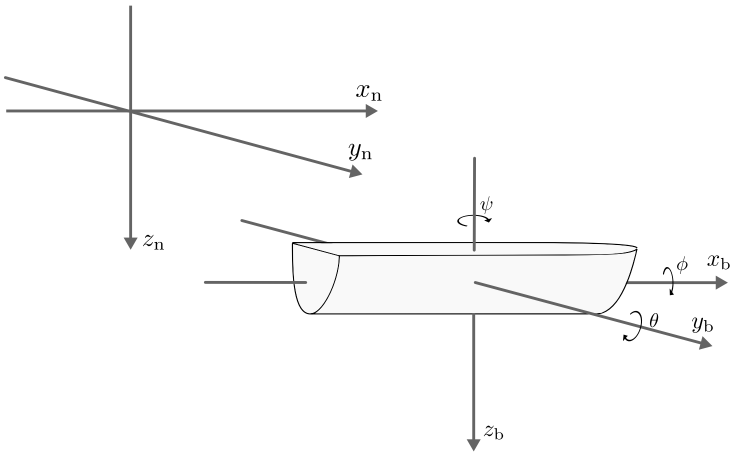


Figure 4.1: x_b, y_b and z_b refer to the position with respect to the body coordinate frame, while x_n, y_n and z_n describe it with respect to the inertial frame. ϕ, θ and ψ refer to the rotation around x_b, y_b and z_b, respectively.

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.

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 [10, p. 22].

[1] FiXme Note: Write header

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

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

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

Note that due to the size of the matrix sine and cosine are denoted s and c respectively.

To describe a vector in the inertial frame given its description in the body frame, a matrix-vector multiplication can be done as follows:

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

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 can be done following the same procedure using $\mathbf{R}_\mathrm{b}^{\mathrm{n}\ \mathrm{T}}$ as the rotation matrix.

4.1.2 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 boat 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.

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

In the case of the rotational movement, the motion is describe using the Newton's second law applied to rotational movement, where the torques applied to the system influence the angular acceleration in each axis.

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

This movement is subject to the Coriolis effect, which appear if the vessel is not rotating around the axis with least- or highest- inertial axis. The influence of this force is small if the vessel rotates at low speeds, hence it have been neglected in the model.

4.1.3 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	[kgm^{-3}]
g	is the gravitational acceleration	[ms^{-2}]
V	is the volume of fluid displaced by the surface vessel	[m^3]
ΔV	is the change in volume of fluid displaced by the surface vessel	[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

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

Where:

$F_{z_{\text{n}}}$	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 equation is applied.

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

Where:

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

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

$$F_{x_{\mathrm{b}}} = -\rho g A_{\mathrm{wp}} z_{\mathrm{n}}(-\sin\theta) \tag{4.8}$$

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

$$F_{z_{\mathrm{b}}} = -\rho g A_{\mathrm{wp}} z_{\mathrm{n}}(\cos\theta\cos\phi) \tag{4.10}$$

If the variations of ϕ and θ are small, the contribution of the buoyancy force in the x_{b} and y_{b} directions can be neglected and not included in the final model of the vessel. [10, pp. 62-67]

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 torques 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. [10, pp. 62-67]

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

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

Where:

T_{ϕ}	is the restoring torque due to the buoyancy force in the ϕ direction	[Nm]
T_{θ}	is the restoring torque due to the buoyancy force in the θ direction	[Nm]
$\overline{GM}_{\mathrm{T}}$	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 related with the position of the center of gravity and that of the center of buoyancy. [10, pp. 65-67]

4.1.4 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 forces that originate from the vessel imposing some energy in the surrounding fluid when it moves through it. The force generated is dependent on the acceleration of the vessel. [2]

The viscous damping is a combination of several factors, namely, skin friction, wave drift damping and vortex shedding [10, p. 122]. This type of damping appears in the equations as coefficients that multiply, with negative sign, the different translational and angular velocities that

[2] FiXme Note: what to do here??

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.

d_i is the viscous damping coefficient.

This equation consider that the viscous friction is linear, this assumption can be done for vessel speeds lower than 2 m/s [10, p. 138].

4.2 Model Equations

The final model equations are presented in this section, and they are described relative to the body frame, meaning that all movement is relative to the boat. Figure 4.2 and 4.3 show a diagram of the vessel.

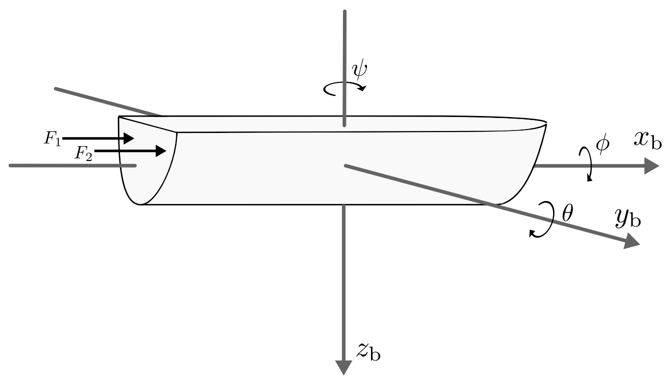


Figure 4.2: Diagram of the boat where the forces applied by the motors are shown.

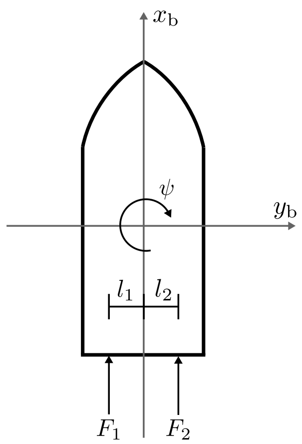


Figure 4.3: Above perspective of the vessel, where the distances needed for the model equations are presented.

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 motors and the damping, that create and acceleration in the boat.

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

Where:

$m_{\mathrm{x,y,z}}$	are the virtual masses in their respective direction	[kg]
$\ddot{x}_{\mathrm{b}}$	is the acceleration in the x_{b} direction	[ms^{-2}]
$\ddot{y}_{\mathrm{b}}$	is the acceleration in the y_{b} direction	[ms^{-2}]
$\ddot{z}_{\mathrm{b}}$	is the acceleration in the z_{b} direction	[ms^{-2}]
$\dot{x}_{\mathrm{b}}$	is the velocity in the x_{b} direction	[ms^{-1}]
$\dot{y}_{\mathrm{b}}$	is the velocity in the y_{b} direction	[ms^{-1}]
$\dot{z}_{\mathrm{b}}$	is the velocity in the z_{b} direction	[ms^{-1}]
$F_{1,2}$	are the forces applied by each motor	[N]

From the equations it can be seen that only the x_{b} axis is controllable, as this is the axis containing the main thrusters. While the vessel do contain side thrusters, they are only intended for fine maneuvering, as their strength is relatively weak, compared to the main thrusters.

The virtual mass components are the added mass from the displaced water in the respective direction plus the mass of the vessel. This value is different for each axis, as the difference in shape drags a different amount of water.

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	[kg]
I_{y}	is the inertia around the y_{b} axis	[kg]
I_{z}	is the inertia around the z_{b} axis	[kg]
$\ddot{\phi}$	is the angular acceleration around the x_{b} axis	[rads^{-2}]
$\ddot{\theta}$	is the angular acceleration around the y_{b} axis	[rads^{-2}]
$\ddot{\psi}$	is the angular acceleration around the z_{b} axis	[rads^{-2}]
$\dot{\phi}$	is the angular velocity around the x_{b} axis	[rads^{-1}]

$\dot{\theta}$	is the angular velocity around the y_b axis	[rads^{-1}]
$\dot{\psi}$	is the angular velocity around the z_b axis	[rads^{-1}]
l_1	is the perpendicular distance from motor 1 to the center of gravity	[m]
l_2	is the perpendicular distance from motor 2 to the center of gravity	[m]

Similar to the translational equations, only one axis is controllable. These are, however, not a problem in practice, since the boat is stable by nature and it can be controlled even being and under-actuated vehicle if the appropriate control objectives are chosen [10, pp. 235-239].

4.3 Linearization of Model Equations

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

The equilibrium point must fulfill that all the derivatives of the states are zero, in this case the velocities and accelerations. This also result in the restoring forces and torques, as well as the motor forces equal to 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.

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

In the case of the boat, the only non-linear terms are the restoring forces and torques. They can be linearized and give the result seen in

$$\tilde{F}_{x_\mathrm{b}} = 0 \tag{4.27}$$

$$\tilde{F}_{y_\mathrm{b}} = 0 \tag{4.28}$$

$$\tilde{F}_{z_\mathrm{b}} = -\rho g A_\mathrm{wp} \tilde{z}_\mathrm{n} \tag{4.29}$$

$$\tilde{T}_\phi = -\rho g V \overline{GM_L} \cdot \tilde{\phi} \tag{4.30}$$

$$\tilde{T}_\theta = -\rho g V \overline{GM_L} \cdot \tilde{\theta} \tag{4.31}$$

The first order Taylor approximation is only applicable in areas close to the operating point, however as the vessel should not deviate much from this during operation.

From now on, the linearized variables are represented without the symbol Δ, 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

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

5 | Controller Design

Header of control chapter

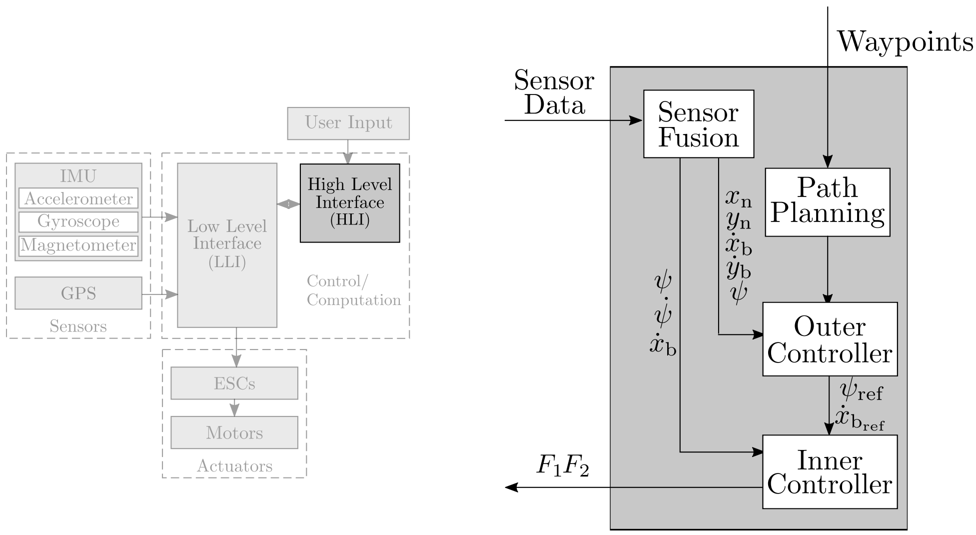


Figure 5.1

5.1 State Space Model

The linearized model derived in section 4.3, consisting of Equation 4.32 to 4.37, needs to be represented in state space form in order to design a state space controller. In order to do that, the 3 degree of freedom model used for the control of the vessel is represented as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{5.1}$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{5.2}$$

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 feed-forward matrix

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

$$\mathbf{x(t)} = \begin{bmatrix} \psi \\ \dot{\psi} \\ \dot{x}_b \end{bmatrix} \qquad \mathbf{y(t)} = \begin{bmatrix} \phi \\ \dot{x}_b \end{bmatrix} \qquad \mathbf{u(t)} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{5.3}$$

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

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d_\psi}{I_z} & 0 \\ 0 & 0 & -\frac{d_x}{m_x} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{l_1}{I_z} & -\frac{l_2}{I_z} \\ \frac{1}{m_x} & \frac{1}{m_x} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

and the **D** matrix is zero.

5.2 Initial Control Design

[1] [2] It is desired to design a state feedback using a linear quadratic regulator (LQR). As the controller eventually must be implemented the design is carried out in the discrete domain. To do so, it is necessary to discretize the system. A discrete state space model can be expressed as,

$$\mathbf{x}(k+1) = \mathbf{A_z}\mathbf{x}(k) + \mathbf{B_z}\mathbf{u}(k) \tag{5.5}$$

$$\mathbf{y}(k) = \mathbf{C_z}\mathbf{x}(k) + \mathbf{D_z}\mathbf{u}(k) \quad , \tag{5.6}$$

where the z subindexes indicate the matrices being in the discrete domain and k is the sample index. The model is discretized using zero order hold. In Figure 5.2 the discrete system is shown in a block diagram. The feed forward matrix is excluded as it is not present in this system.

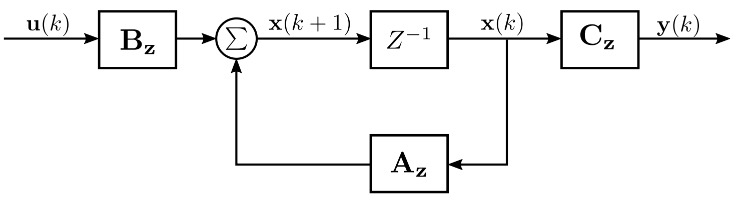


Figure 5.2: Block diagram of the discrete system without feed forward.

In order to track a reference and handle input disturbances, it is chosen to also include an integral controller in the design. The final control structure is seen in Figure 5.3.

[1] FiXme Note: Make Matrix Appendix with numbers
[2] FiXme Note: Refference to matrix appendix where appropriate in this section.

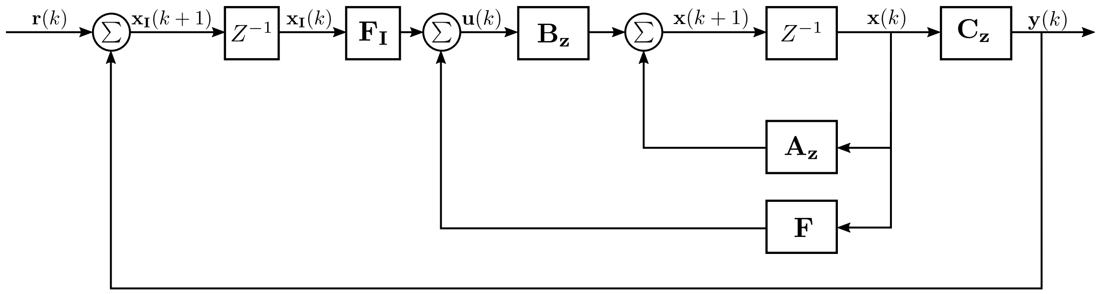


Figure 5.3: Block diagram of the control structure in the discrete domain.

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

$$\mathbf{x_e}(k+1) = \mathbf{A_e}\mathbf{x}(k) + \mathbf{B_e}\mathbf{u}(k) + \mathbf{r}(k) \tag{5.7}$$
$$\mathbf{y}(k) = \mathbf{C_e}\mathbf{x}(k) \quad . \tag{5.8}$$

To describe the control design in this form, the $\mathbf{A_e}$, $\mathbf{B_e}$ and $\mathbf{C_e}$ matrices must be constructed. From Figure 5.3, following relation is found:

$$\mathbf{x_I}(k+1) = \mathbf{x_I}(k) + \mathbf{y}(k) + \mathbf{r}(k)$$
$$\mathbf{x_I}(k+1) = \mathbf{x_I}(k) - C_z\mathbf{x}(k) + \mathbf{r}(k) \quad . \tag{5.9}$$

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

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x_I}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A_z}_{3x3} & \mathbf{O}_{3x2} \\ -\mathbf{C_z}_{2x3} & \mathbf{I}_{2x2} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x_I}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B_z}_{3x2} \\ \mathbf{O}_{2x2} \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{O}_{3x2} \\ \mathbf{I}_{2x2} \end{bmatrix} \mathbf{r}(k) \tag{5.10}$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{C_z}_{2x3} & \mathbf{O}_{2x2} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x_I}(k) \end{bmatrix} \quad , \tag{5.11}$$

which cooresponds to Equation 5.7 and 5.8.

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

$$J = \sum_{k=0}^{\infty} \mathbf{x}_k^\mathrm{T}\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\mathrm{T}\mathbf{R}\mathbf{u}_k \; dt \quad . \tag{5.12}$$

Where:

Q	is the symmetric positive semidefinite state cost matrix
R	is the symmetric positive definite input cost matrix

The Q matrix contains the penalties for the state, such that a higher cost is generated for more critical states, thus driving these states faster to zero. The R matrix contains the penalties for

the input. This helps to ensure that the actuators never enters saturation.
It is necessary for all states to be stable and controllable. Otherwise the the performance index, J, will become infinite [11, p. 125].
The controllability is determined by

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} \mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^{\mathbf{2}}\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^{\mathbf{3}}\mathbf{B}_\mathrm{e} & \mathbf{A}_\mathrm{e}^{\mathbf{4}}\mathbf{B}_\mathrm{e} \end{bmatrix} \quad , \tag{5.13}$$

of which the rank is 5, that is, the controllability matrix, $\boldsymbol{\mathcal{C}}$, has full rank, thus the system is controllable [12, 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.

Bryson's rule is used as an initial design method to determine sensible values for the state and input penalties in the Q and R matrices.

$$Q_{ii} = \frac{1}{{x_{i_\mathrm{max}}}^2} \qquad R_{ii} = \frac{1}{{u_{i_\mathrm{max}}}^2} \tag{5.14}$$

Where:

x_{i_max}	are the maximum acceptable state values
u_{i_max}	are the maximum acceptable input values

The requirements stated in section 2.2 must be taken into account when determining the values of x_{i_max} and u_{i_max}. [3]

From this the state feedback is calculated by [13, p. 42],

$$\mathbf{F}_\mathrm{e} = -(\mathbf{B}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{B}_\mathrm{e} + \mathbf{R})^{-1}\mathbf{B}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{A}_\mathrm{e} \quad , \tag{5.15}$$

where $\mathbf{P}$ can be found as the solution of the infinite horizon algebraic discrete-time Riccati equation [13, p. 42],

$$\mathbf{P} = \mathbf{A}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{A}_\mathrm{e} + \mathbf{Q} - \mathbf{A}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{B}_\mathrm{e}(\mathbf{B}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{B}_\mathrm{e} + \mathbf{R})^{-1}\mathbf{B}_\mathrm{e}^\mathrm{T}\mathbf{P}\mathbf{A}_\mathrm{e} \quad . \tag{5.16}$$

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 5.3.

[3] FiXme Note: Include arguments and numbers for final choice of Q and R.

6 | Path Following

The vessel functionality requires it to follow a path along which the bathymetric measurements are taken. This path should be followed as precisely as possible in the interest area. For doing that (... explain how we generate the path). The generated path is followed by using an enclosure based steering algorithm [10, 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 section 5.1. [1]

6.1 Path Following Algorithm

The path generated is approximated by straight line segments connected by waypoints. The vessel follows then this segments in order to follow 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. If curved paths were required, the solution would be to sample the path with higher frequency in curved sections. Figure 6.1 shows an example of how a path is approximated by straight line segments and waypoints.

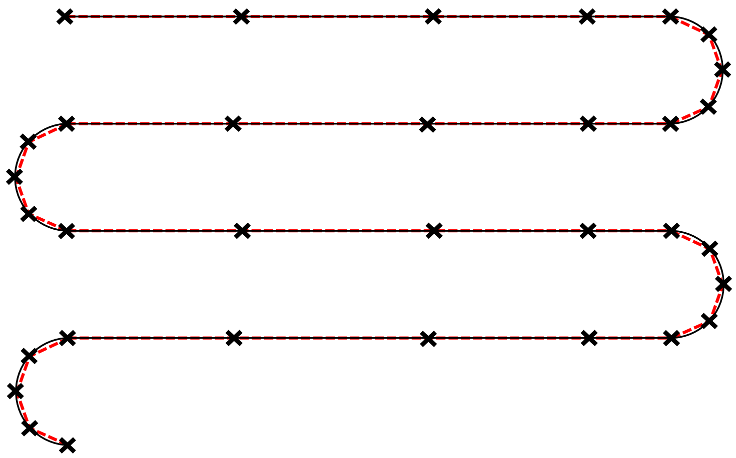


Figure 6.1: A predefined path and its approximation as straight line segments using waypoints.

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 boat. Figure 6.2 shows how this crossing point is obtained. The point is also called LOS point and is found using the equations of the circle and of the straight line as

$$(x_{\mathrm{los}} - x_{\mathrm{n}})^2 + (y_{\mathrm{los}} - y_{\mathrm{n}})^2 = R^2, \tag{6.1}$$

$$y_{\mathrm{los}} - y_{\mathrm{k}} = \frac{y_{\mathrm{k+1}} - y_{\mathrm{k}}}{x_{\mathrm{k+1}} - x_{\mathrm{k}}}(x_{\mathrm{los}} - x_{\mathrm{k}}) \tag{6.2}$$

Where:

[1] FiXme Note: Provisional header

R is the radius of the circle centered at the vessel position

$[x_{\text{los}}, y_{\text{los}}]$ is the crossing point between the circle around the vessel and the straight line that joins the waypoints

$[x_{\text{k}}, y_{\text{k}}]$ is the current waypoint followed in the path

$[x_{\text{k+1}}, y_{\text{k+1}}]$ is the following waypoint followed in the path

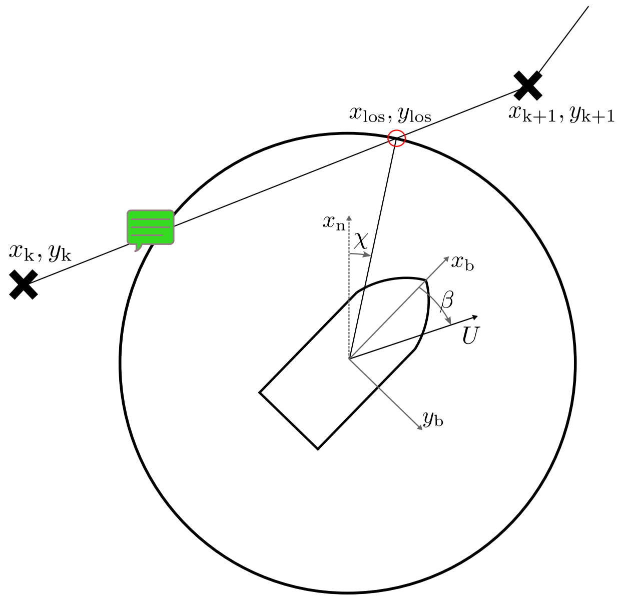


Figure 6.2: Algorithm used to find the yaw reference for the state space controller in order to follow a path.

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 (6.3). 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 or waves would generate some speed 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 (6.4) and (6.5).

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

$$\chi = \arctan\left(\frac{y_{\text{los}} - y_{\text{n}}}{x_{\text{los}} - x_{\text{n}}}\right), \tag{6.3}$$

$$\beta = \arctan\left(\frac{\dot{y}_{\text{b}}}{\dot{x}_{\text{b}}}\right), \tag{6.4}$$

$$\psi_{\text{ref}} = \chi - \beta. \tag{6.5}$$

Where:

χ is the angle between the x_{n} axis and the LOS point

β is the angle between the velocity vector of the vessel and the x_b axis

The algorithm relies on the path and the circle defined around the boat 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.

In order to follow all the path, a way to change the which two waypoints define the current path segment needs to be established and 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 6.3.

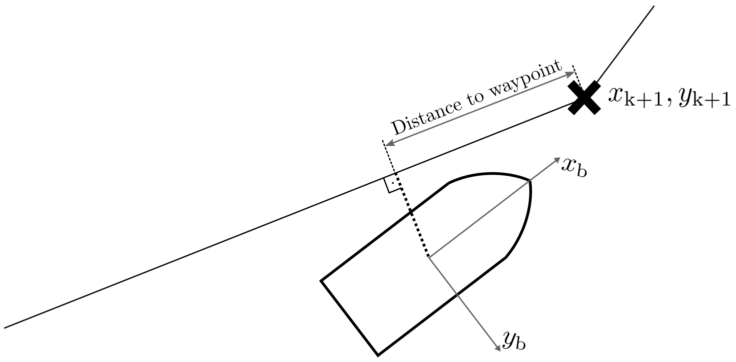


Figure 6.3: The distance considered when defining the criterion to change to new waypoints.

With this approach, the vessel will always try to move forward in the path although a waypoint position has not being 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 is going to be itself close to the waypoint when the change occurs. This can be seen in the simulation plots presented below.

6.1.1 Path Following Algorithm Simulation

[2] [3] The path following algorithm has been tested in the same path and considering different settings for the algorithm. In all cases, the radious of the circle defined around the vessel is 5 m and the distance in which the active waypoints are changed is 3 m.

In Figure 6.4 and 6.5 the result of the algorithm are presented when considering the simpler case in which $\psi_{\text{ref}} = \chi$, that is, assuming the velocity of the vessel is pointing along the x_b direction. In the first graph the path is followed precisely as the assumption regarding the velocity vector holds, whereas in the second, the constant disturbance imposes and offset in the position of the vessel.

[2] FiXme Note: Do not include many graphs now because we will probably redo them. Just some dummy ones.
[3] FiXme Note: Maybe we could include the result also with different low level controllers

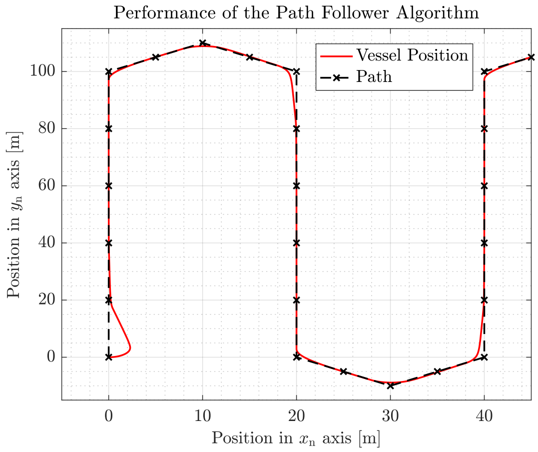


Figure 6.4: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi$.

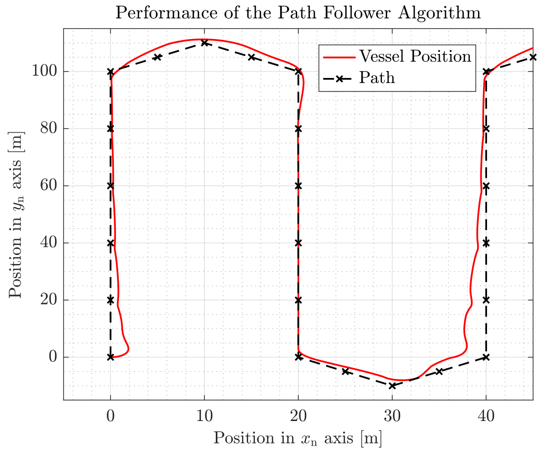


Figure 6.5: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi$. The vessel is experiencing a constant disturbance force of 1 N applied with an angle of $\pi/2$.

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 6.6, where the vessel is experiencing the same disturbance as in Figure 6.5. In this case, the offset in position has been corrected and the path is precisely followed.

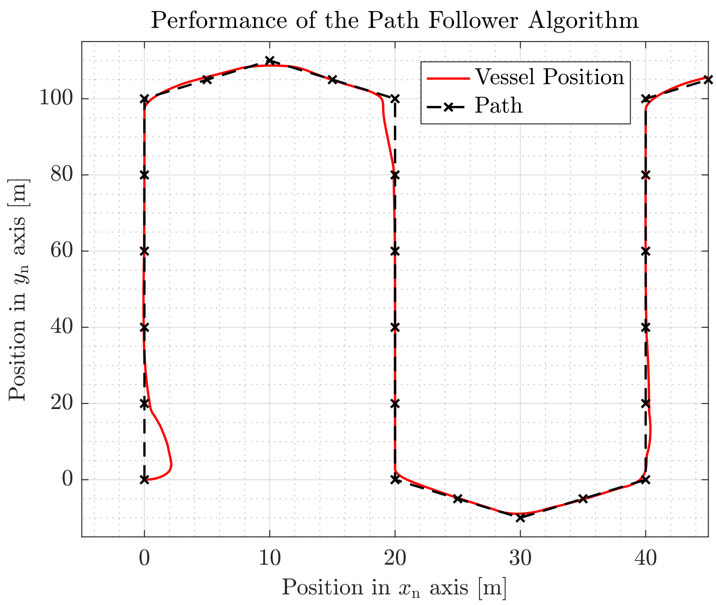


Figure 6.6: Performance of the path following algorithm based on $\psi_{\text{ref}} = \chi - \beta$. The vessel is experiencing a constant disturbance force of 1 N applied with an angle of $\pi/2$.

According to the results of the simulations, it can be said that the vessel hits the waypoints precisely 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. In many cases, and specially for bathymetric measurements the algorithm can be conseidred suitable.

Part II

Design & Implementation

7 | Implementation

7.1 ROS

ROS is designed to be a communication infrastructure for a project, which allows 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. The topic contains the data stored in a prespecified data structure, specified as a message, such that each node publishes data of the same type.

7.1.1 System overview

The system consistes of two subsystems, a Low-Level-Interface (LLI) and a High-Level-Interface (HLI). The LLI is a Harware interface, responsible for reading all the sensors and sending control signals to the motors. The HLI contains the sensor fusion, and the controllers, as well as a interface to the operator.

7.1.2 Route following node

The route following uses waypoints together with position data to generate a reference for the low level controller. The method described in section[1] describes the line between two waypoints as an affine linear line. This description have the issue of having singularity when describing vertical lines, making the slope infinite. To circumvent this, the implementation uses a different approach. While the general concept remains the same, the method for describing a line is different.

[1] FiXme Note: Ref to section describing Route following node

Part III

Test & Conclusion

Appendix

Bibliography

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

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

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

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

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

[6] *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.

[7] *66 Channel UP-501 GPS Receiver*. Feb. 21, 2017. URL: https://www.sparkfun.com/products/retired/10702.

[8] *ASUS Eee PC 1000H*. Feb. 21, 2017. URL: https://www.asus.com/us/Notebooks/Eee_PC_1000H/specifications/.

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

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

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

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

[13] 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).

List of Corrections