

INITIAL DEVELOPMENT OF A COURSE KEEPING VESSEL



Nick Østergaard
Simon Als Nielsen
Brian Bach Nielsen
Rasmus Lundgaard Christensen
Christoffer Stagsted Andreassen



12gr630
Aalborg University 2012
Department of Electronic Systems

Institute of Electronic Systems

Control Engineering

Fredrik Bajers vej 7

9220 Aalborg Øst

Phone 99 40 86 00

<http://es.aau.dk>

Title:

Development of a course keeping vessel

Theme:

Control engineering

Projectperiod:

P6, spring semester 2012

Projectgroup:

630

Participants:

Nick Østergaard

Simon Als Nielsen

Brian Bach Nielsen

Rasmus Lundgaard Christensen

Christoffer Stagsted Andreassen

Supervisor:

Jesper Abildgaard Larsen

Number of printed copies: 7

Number of pages: 132

Appendices: 8 + 1 CD

Finished on: June 26, 2012

Synopsis:

This project documents the development of an autonomous course keeping vessel. The project is developed as a tool to help surveyors map confined waters such as lakes or harbours. The design is based on a remote controlled speedboat that is modified to navigate a predetermined route. The software controller is based on an IGEP board running a Linux operating system, while the necessary sensory data is obtained through two GPS receivers and a 3G Internet connection. A model of the physical system has been created and a control system for this has been simulated and verified.



Preface

This report documents the development of a vessel for depth mapping of confined waters, in the form of a model sized autonomous vessel, that can automatically undertake this task, using various sensors. This project has been researched and written by group 630 at 6th semester control engineering at Aalborg University (AAU). The theme of the semester is “control engineering”. The project has been researched and written between the 1st of February and the 30th of May 2012.

Thanks to

Throughout this project a company have been specially helpful, and therefore a special thanks is given to:

- Nellemann & Bjørnkjær for providing a test vessel, and insight into the practical concerns when surveying water depths.

Nick Østergaard

Simon Als Nielsen

Brian Bach Nielsen

Rasmus Lundgaard Christensen

Christoffer Stagsted Andreassen

Reading guide

The following report is divided into parts, related to different phases of the project. The parts are divided into chapters, the chapters describe different aspects of the project. The chapters are subdivided a number of times to further split up the content into specific topics. The report is ended with an appendix part, that contains all the material that is relevant to the project, but not necessarily interesting to the reader, such as measurement journals and transcripts of meetings.

Citations in the report is done according to the Harvard method, the list of references can be found on page 120. The elements on the list of references are sorted by author.

Acronyms are written to their full extend, the first time they are used, with the acronym in parentheses, thereafter only the acronym is used. The list of acronyms can be found on page xi.

Notation of vectors are written in bold font with lower case letters (**v**), matrices are written in bold font with upper case letters (**M**). Single variables and constants are typeset in normal math (*x*).

Attached to the report is a CD, which contains copies of web references and other digital files (source code, scripts and raw measurement data) that could be of interest to the reader. In some places in the report there will be a reference to the CD; this will look like this: /path-to-file.

Terminology

In this project the the “confined water depth measuring vessel” will be referred to simply as the “vessel”.

When the heading is mentioned throughout the report, it is the heading in relation to true north also known as geodetic north, unless otherwise informed. Magnetic north will not be used.

Contents

Preface	v
Reading guide	vi
Terminology	vi
Contents	vii
I Preliminary analysis	1
1 Introduction	3
1.1 Initial problem	3
1.2 Initial requirements	3
2 Analysis	5
2.1 Surveyor interview	5
2.2 Positioning	6
2.3 Disturbances	6
2.4 Maneuvering	7
2.5 Hardware platform	8
2.6 Limitations	9
2.7 Conclusion	10
3 Requirements	11
3.1 Requirement specification	11
3.2 Acceptance testing	11
4 System overview	13
4.1 Software layers	13
4.2 On board computer	15
4.3 Wireless data link	15
4.4 Microcontroller	15
4.5 GPS communication module	15
4.6 Route keeping	15
II Modelling	19
5 Reference frames	21
5.1 Earth-fixed frame	21
5.2 Surface-fixed frame	22
5.3 Body-fixed frame	23
6 Dynamics	25
6.1 Forward motion	25

6.2	Turning motion	26
7	Wind and water disturbances	29
7.1	Simplifications	29
7.2	Model of wind vector	29
7.3	Model of wind force	30
8	DC motor	35
8.1	Electrical part	35
8.2	Mechanical part	35
8.3	Transfer function of the DC motor	37
9	Linearization	39
10	State space representation	43
11	Summation	45
III	Controller design	49
12	Multivariable frequency control	51
12.1	General MIMO system	51
12.2	Forward control	52
12.3	Turn control	52
13	Model based control	57
13.1	Controllability	58
13.2	Observability	60
13.3	Optimal feedback gain	60
13.4	Adding the reference signal	62
13.5	Adding an integral part	65
13.6	Summation	67
13.7	Discretization of the state space equations	68
14	Software	71
14.1	Software structure	71
14.2	Interfaces	72
14.3	GPS	74
14.4	Conversion of data to surface frame	75
14.5	Route planner	76
14.6	Controller	77

IV Closing	81
15 Acceptance test	83
15.1 Acceptance testing simulations	83
15.2 Result overview	85
16 Conclusion and future perspective	87
16.1 Conclusion	87
16.2 Further development	87
V Appendix	89
A Meeting with Nellemann & Bjørnkjær	91
B RTK validation	93
B.1 Purpose	93
B.2 Measurements	93
B.3 Conclusion	94
C Measurements on test vessel	97
C.1 Purpose	97
C.2 Tools	97
C.3 Measurement setup and procedure	97
C.4 Measurements	99
C.5 Results	99
D Measurements of propulsion motor	101
D.1 Purpose	101
D.2 Tools	101
D.3 Measurement setup and procedure	101
D.4 Measurements and calculations	103
D.5 Results	105
E Vessel Dimensions	107
F Vessel tests	109
G ARGO datasheet	113
G.1 Operating ARGO	113
G.2 Technical information	114
H Transfer function implementation	117
H.1 State Space to Laplace domain	117
H.2 Laplace domain to discrete time	117

Bibliography	119
---------------------	------------

List of acronyms

AAU	Aalborg University
ABS	Acrylonitrile Butadiene Styrene
ARGO	Autonomous Research and Geo-Survey Oceanographer
ECEF	Earth-centred Earth-fixed
GPS	Global Positioning System
IMU	Inertial Measurement Unit
MIMO	Multiple-Input Multiple-Output
PWM	Pulse-Width Modulation
RTK	Real Time Kinematic
SISO	Single-Input Single-Output
TCP	Transmission Control Protocol
WGS84	World Geodetic System 84

Part I

Preliminary analysis

This part describes the preliminary analysis leading up to the design of Autonomous Research and Geo-Survey Oceanographer (ARGO). This includes analysis of the initial problem, requirement specification and hardware.

1

Introduction

Today mapping of harbours and other under water areas relatively close to the shore is done manually, a task that is both time consuming and expensive, as it is typically carried out by multiple surveyor who by boat sails to the desired position uses sensory equipment to map the seabed, and then sails back to shore. It is desirable to automate this task, and carry it out with an autonomous vessel, that is programmed to navigate to a certain area, carry out the mapping of a predetermined area – and then return to the point of departure.

To do this however, the vessel should be able to navigate to the site, and whilst there – map the area in the most efficient way possible, whilst under the influence of external disturbances such as wind, waves and currents. Of course, these disturbances are not as influential when the vessel is mapping confined waters, as they would be when mapping the open sea – but they do still play a role in the design of the vessel.

An interview with a surveyor has been conducted, to both validate these hypothesis and to get information about further needs and requirements for the vessel. The outlines of the meeting can be found in appendix A on page 91.



Figure 1.1: The vessel platform used in this project. On the image two GPS antennas and a 3G modem have been retrofitted.

1.1 Initial problem

Depth mapping of confined waters is a time consuming task so the main focus of this project is to develop an autonomous surface vessel to automate this. As this is to be used by surveyors, some initial requirements will be set up in the following section.

1.2 Initial requirements

To systematize the development process of the vessel, several requirements are made from the initial problem, and listed in the following table.

The vessel should be able to:

- Fit in the trunk of a medium sized car.
- Operate autonomously.
- Navigate along a predefined route.
- Recitative an absolute position reference.
- Operate under mild to moderate weather conditions.
- Log the depth measurements.

Further analysis of these requirements is conducted in the following chapter.

2

Analysis

Throughout this chapter the parameters affecting the design of the vessel will be analysed leading up to a requirement specification.

2.1 Surveyor interview

The surveyor who was interviewed about the requirements for an autonomous surveying vessel, was Peter Eistrup from Nelleman & Bjørnkjær.

Today's methods for surveying lakes, harbour areas and small rivers consists of a team of two or three employees. For surveying harbours and lakes they use a small vessel equipped with a single beam echo sounder for depth measuring. While the locations for the individual depth registrations are either found via Global Positioning System (GPS) or prism together with fixed datums near the site. The locations where the depth is measured is defined as some predetermined run lines which runs back and forth over the appointed area.

What Nelleman & Bjørnkjær expect of an autonomous surveying vessel is listed in the appendix A on page 91.

- An autonomous vessel which can navigate using GPS coordinates in quiet weather conditions.
- Obstacle avoidance.
- That the vessel can be operated by a single employee.
- A single beam echo sounder mounted under the vessel.
- Depth measuring accuracy of 15 cm.
- Capable of measuring depth down to 50 metre.
- A GPS positioning accuracy of 10 cm.
- Real time data transfer to the operator, with information on depth and position, with the purpose to verify that the data collected is usable.
- Depth measuring for every metre the vessel travels.
- Primarily be used in lakes and rivers
- The vessel must return to the launching point, if the battery is running low, or the mission has been completed.

The requirements for the vessel set by Nelleman & Bjørnkjær will effect the following sections in the analyses and thus shape the design of the vessel and the outcome of the project.

2.2 Positioning

As stated in the initial problem in section 1.1 on page 3, the task is to do depth mapping from a surface vessel. This requires a precise positioning system, which in turn can be used when autonomously controlling the vessel as well.

Today surveyors use more than one kind of positioning in their daily work like kinematic GPS and relative positioning with lasers. This is known through the meeting with Nelleman & Bjørnkjær. Since these positioning technologies are used by surveyors, these are to be considered the options for positioning available to the vessel.

In order to perform depth mapping autonomously it is required that the vessel can follow a predefined route, so that it will cover all of the targeted area. To make the vessel follow this route, it is required for the control system to have a feedback system, making it able to compensate for external disturbances such as currents, wind and waves.

The route is defined as a set of waypoints, hence the vessel's task is only to reach the next waypoint. Therefore it is necessary to determine the current location of the vessel and the current heading in order to get the vessel back on course when disturbances occur.

According to the initial requirements, see section 1.2 on page 3, the vessel should stay within 1 metre of the set route, the dotted lines in desired route in figure 2.2 on page 8, which means that some measure must be taken to secure the precision of the location.

Therefore an absolute position system must be chosen. The solution is to use a kinematic GPS which has a high relative precision to a base station near the work area. The kinematic GPS works by having two GPS receivers that are capable of outputting raw demodulated data, one on the vessel (rover) and one on the base station, with a piece of computer software used to calculate a more accurate position than a regular single GPS unit can do. This works by having the base station on a static position, hence its exact position is always known. The rover has the same setup, except that it does not have a known position. By using data from the rover and the base station, it is possible for the rover to compensate for different disturbances that affect the positioning calculations and it is possible to obtain sub decimetre accuracy. This means that the vessel should have a constant communication link with the base station.

2.3 Disturbances

In the initial problem the vessel is specified to be used in confined waters. For this purpose Port of Aalborg has been chosen as the basis for the analysis. Parameters for modeling and design purposes include an average wind estimation, currents and average depth.

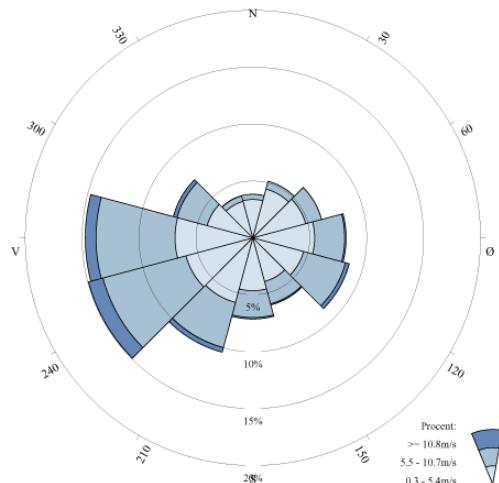


Figure 2.1: Overview of the average windspeed direction in Denmark. Data collected by the Danish Meteorological Institute (DMI) between 1973–2002.

When sailing at a certain course the course has to be continuously corrected, due to external forces acting upon the vessel.

These external forces consists of wind, water current, waves and to a small extend the torque roll from the propeller. The forces will not effect the vessel evenly on the whole surface, which will cause the vessel to diverge from the route.

It is important to determine how much noise these forces will generate and how much of an impact it will have on the vessel at the most. This will give a starting point, for the design of the control algorithm for the vessel.

Due to practical reasons, the area which will be used as testing grounds will be Limfjorden which runs through Aalborg, as this is confined, and should provide disturbances, which complies with the limitations. As the vessel is relatively small, there are some limitations that should be taken into account, as strong winds could knock the vessel over, and high waves could flood it. This is specified in the limitations section 2.6 on page 9. Extreme scenarios will not be dealt with since surveying will not normally take place under these conditions.

The wind speeds in Denmark has been mapped as shown on figure 2.1. See [Danish Meteorological Institute (DMI), 2012] for further details.

2.4 Maneuvering

Maneuvering is the act of making the vessel move from one waypoint to another. A sequence of waypoints is called a route, and this route is straight lines between two consecutive waypoints. This line between two consecutive

waypoints is used as heading reference for the vessel. When a constant disturbance is induced on the vessel, it should be able to maintain its centre point on the line, the vessel itself can have a heading that differ from the route line, as the disturbance may not work equal on the whole surface of the vessel.

When considering the movement of the vessel in autonomous mode, it is also important to analyze the case of an abrupt large disturbance that will move the vessel beyond the margins. This could be handled in two ways, one is to get the vessel to move as fast as possible back onto the route while the other is to change the path to the shortest distance to the next waypoint. These situations can be seen in figure 2.2.

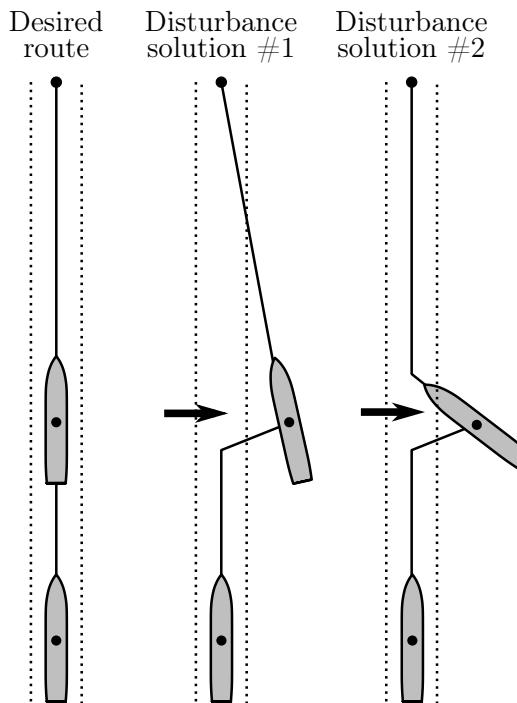


Figure 2.2: Abrupt disturbances to an ideal route, and two solutions of how to recover from this, and get back on route.

As the vessel is supposed to map the seabed, it is important that the vessel follows the predetermined route to make sure all the intended area is covered. With this in mind solution number 2 is the best fit for the intentions of this project.

2.5 Hardware platform

For this project a hardware platform is needed in the form of an empty hull to design the vessel around. The hull should be large enough to contain all the relevant parts, while being relatively easy to transport and use in the field.

This sets some constraints on how big the vessel can be and how it should be designed.

The hull design of the vessel should facilitate a fairly stable platform, that can provide acceptable measurements even with small waves on the water surface, while being small enough to be transportable by a medium sized car.

To make it a safe measuring platform, it is imperative that the vessel is large enough to contain all the necessary equipment, i.e. a depth measuring device, a computer, antennas, motors cooling system, wiring and mounts. Figure 2.3 is a sketch of where the equipment is placed on the vessel. The cooling system is placed around the propulsion motor and the computer is added to the vessel in the form of an IGEP and ARM board. The depth measuring device is not implemented on the sketch.

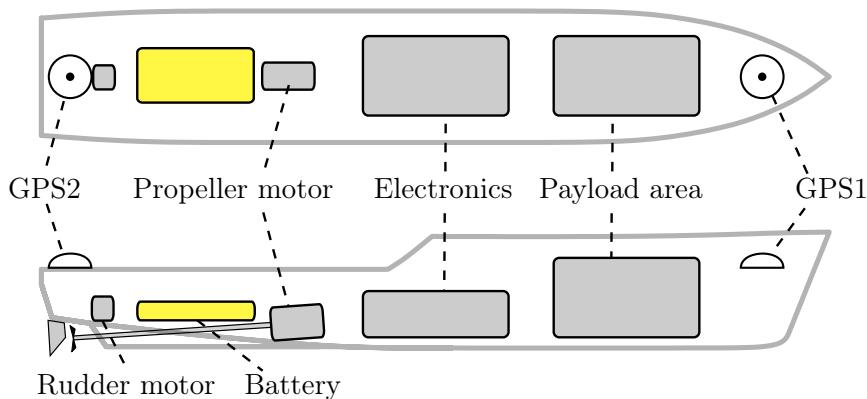


Figure 2.3: Sketch of hardware platform.

Taking all of the above into account, it has been decided that the hull should have an approximate length of one metre and a width of about 0.25 metres.

The vessel used is a model boat provided by Nellemann & Bjørnkjær. The vessel can be seen on figure 1.1 on page 3. Throughout this report the sketch model seen in figure 2.3 will be used.

The general locations of the GPS receivers on the vessel is placed as shown on figure 2.3. The vessel uses two GPS receivers, making it possible to calculate an absolute heading. These GPS receivers are capable of outputting the raw GPS data for computing the accurate position with the aid of the data from the base station.

2.6 Limitations

As described in the previous sections, there are some limitations and assumptions that will be used throughout the development process of the vessel. As small vessels are affected much the same way as larger ones (meaning that the disturbances scale linearly), the wind speed needs to be accounted for as well

as the wave size, as these are the two main external parameters that affects the vessels as it is mapping the sea bed.

As wave size and wind speed are dependent on one another, a lookup in the Beaufort scale, shows that if the waves are to be no bigger than 0.1 metres high, the wind speed should be between 3 and 5 metres per second. This is therefore used as a reference from now on.

The reason for choosing 0.1 metres as a maximum wave height is solely dependent on the size of the vessel. If the waves are much higher, they rise above the stern of the craft and greatly increases the chances of flooding. Therefore the following can be put up as external parameters:

- Wave size: 0.1 metre
- Wind speed: 3–5 metre per second

As for confined waters, it is assumed that the port of Aalborg is sufficient. The water is relatively confined, and is deep enough to measure.

As the scope of this project is to develop an autonomous platform and the time received for completing the project is limited, some functionality limitations have had to be made. Further on, obstacle avoidance and depth measurements will not be considered.

2.7 Conclusion

In conclusion to the analysis, an autonomous platform is to be developed using GPS positioning.

Positioning will consist of the GPS coordinates of the two GPS receivers mounted on the bow and stern of the vessel.

Navigation will be carried out by the use of waypoints. The waypoints will be GPS coordinates and determine the route which vessel has to follow. The vessel must move in a direct line from one waypoint to another. The vessel is not to be straying more than 1 metre to either side from the route. In case of disturbances the vessel must use solution 2 from figure 2.2 on page 8, as this method will give the surveyor the planned depth measurements.

External disturbances that the vessel must be capable of withstanding is; 0.1 metre waves and wind speeds of up to 5 metre per second.

3

Requirements

Throughout this chapter the requirements for the project will be listed, according to the information gathered in the analysis. Tests will be set up to verify the requirements. These tests are described in section 3.2.

3.1 Requirement specification

To take a systematic approach to the development process, the analysis in chapter 2 on page 5 have been condensed into three main requirements, which are divided into sub-requirements. The function of these requirements is to clarify the goals of this project, and are listed as the functionality the vessel should display at the end of the development.

- The vessel must navigate along a straight line with the following constraints:
 - a) Stay within 1 metre of the route (2σ).
 - b) Maximum winds of ± 5 m/s (at an arbitrary angle).
 - c) Keep a velocity of 1 m/s $\pm 10\%$ (2σ).
- The vessel must navigate using an absolute positioning system:
 - a) Log own position within ± 0.1 m, (2σ).
- The vessel must report the current position to the ground station, with the following constraints:
 - a) Reports to the ground station must be in real-time.
 - b) Have an operational range up to 1000 m away from the ground station.

In the acceptance testing in section 3.2 it is defined how the requirements are tested, to see if the vessel fulfills these.

3.2 Acceptance testing

To verify the requirements specified in section 3.1, corresponding tests have been set up to verify whether the vessel complies with these.

Test 1: The vessel must navigate along a straight line:

The vessel is programmed to navigate along a 20 metre straight route, during the movement along the route the vessel will be subjected to wind at varying angles at speeds up to 5 m/s, throughout the test the vessel must keep a

velocity in accordance with the requirement specification. The test will be conducted five times, to make sure sufficient wind angles and speeds have been covered.



Figure 3.1: Path definition

Test 2: The vessel must navigate using an absolute positioning system:

Two reference points are set up, where the absolute position is known. The vessel is placed at the first point, while logging it's position, and then at the second point while logging it's positions.

Test 3: The vessel must report the current position to the ground station:

The vessel is programmed to continuously send out data, and the ground station is set to receive mode. The vessel is moved away from the ground station to a distance of 1000 metres.

4

System overview

This chapter describes how the different layers of hardware and software together make up the entire control system and the payload handling.

Based on the requirements described in section 3 on page 11 a overview of the subsystems is constructed as shown in figure 4.1.

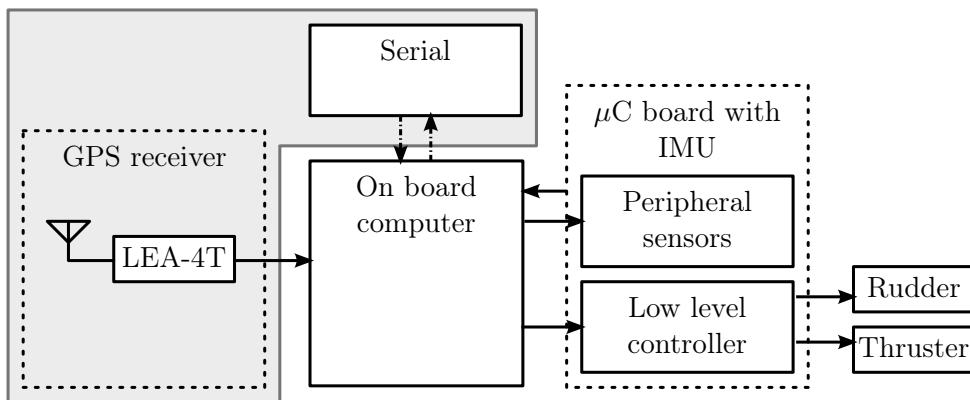


Figure 4.1: Overview of subsystems, differentiating the hardware parts as blocks.

4.1 Software layers

On figure 4.2 on the next page is the overview of the layers in the software structure. These software nodes will be run on an IGEP platform, using a Linux OS. When these tasks are performed, they are done from the top down, and are repeated until the measuring of the designated route is completed.

Receiver

The receiver is the node handling the GPS modules and the correction signal from the ground station. This means that the receiver have to establish and maintain a TCP connection to receive the correction signal from the ground station. Whenever the receiver node is run it uses RTKLib to compare the incoming signals from the GPS receivers and the ground station. This way the accuracy of the vessel's position and orientation is much higher than with only the GPS signals. These corrected GPS signals, which are received in degrees in the earth frame, are then send to the route planner for further processing.

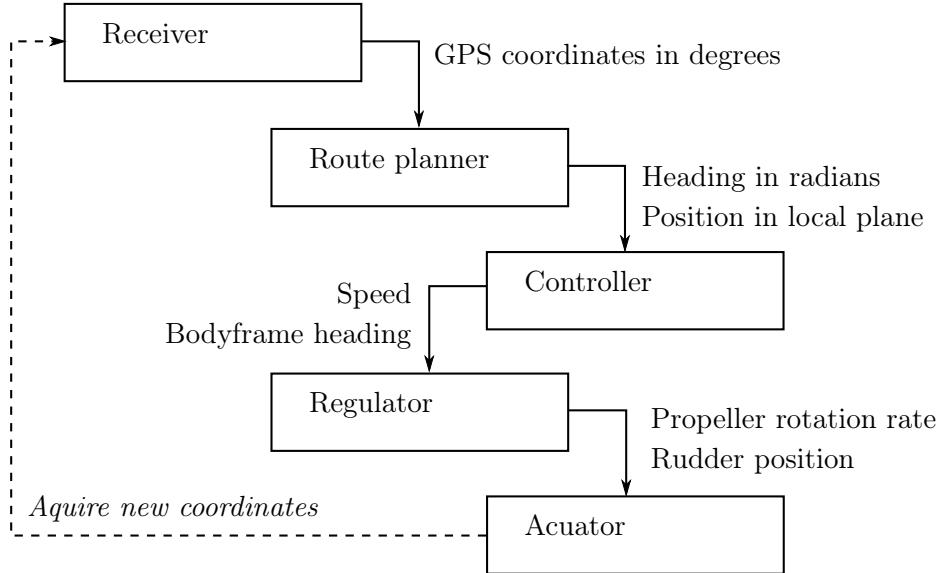


Figure 4.2: Overview of the software structure.

Route planner

The GPS coordinates are in the route planner calculated to a heading and location. If the vessel is straying from the intended route, the route planner have to correct this by having the vessel navigate towards the line of the planned route. This is done by sending the vessel heading as well as the position.

Controller

The controller will be designed to follow changing reference values, the changing location and heading of the vessel, and is thereby called a tracking controller or a servo. Processing these inputs will result in an input for the regulator, which will be the heading the vessel have to sail, to either further follow the route or get back to it. The same is done for the velocity of the vessel.

Regulator

The regulator will have the external disturbances and the output from the controller as input. The function of a regulator is to approximate a fixed output. In this case the fixed output is the path the vessel have to follow. To approximate the route the regulator have the two motors, one for the propeller and one for the rudder.

Actuator

For each of the motors there has to be a driver. The actuator is made so that the output from the regulator can be transformed into motion in the motors. As a reaction to the motors causing the vessel to move, the receiver has to find the new position of the vessel.

4.2 On board computer

It is needed to calculate the position and direction continually, based on navigational data collected from the base station and GPS antennas. Therefore a computer-on-board solution is used.

4.3 Wireless data link

Communication with the ground station is simplified significantly by using already existing technology in the form of a 3G modem.

4.4 Microcontroller

The microcontroller board is used to control the actuators and read input from the sensors. The implemented microcontroller is used, because it is easy to use and already known. Furthermore it is easy to design a shield to go on top of it for further handling of actuators and sensors like the Inertial Measurement Unit (IMU).

4.5 GPS communication module

The GPS module is constructed using two off-the-shelf GPS antennas of the type SM-1575 connected by an RG-174 cable to the GPS receivers. The GPS receiver are LEA-4T systems which can output raw GPS data directly to a PC. The precision has been found to be 3–4 cm. The method for reaching and measuring this precision is shown in appendix B on page 93.

4.6 Route keeping

Per the requirements in section 3 on page 11 it is necessary for the vessel to follow a route, hence it is needed to make the vessel converge to a straight path between two waypoints. The following text describes the solution to that problem.

A case could be to only regulate after the destination waypoint, but this does not make the vessel converge to the route leg nicely if the vessel somehow got

far away from the route. This is because the vessel does not try to get within the margin of the route, only tries to reach the destination waypoint.

A method to make the vessel converge to the route and follow the path to the destination waypoint, is to have a moving waypoint in the body fixed frame (the frame described in section 5 on page 21), with a fixed or speed dependent distance from the vessels centre of gravity, that the lower control system can use as a reference.

This reference is then moved (for each position sample) in a manner such that it satisfies how the vessel should move. This geometry in the surface frame is illustrated in figure 4.4 and described further in section 4.6.

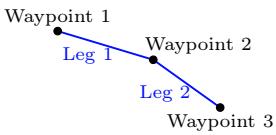


Figure 4.3:

How a route is defined from waypoints that defines legs.

Geometry of route keeping

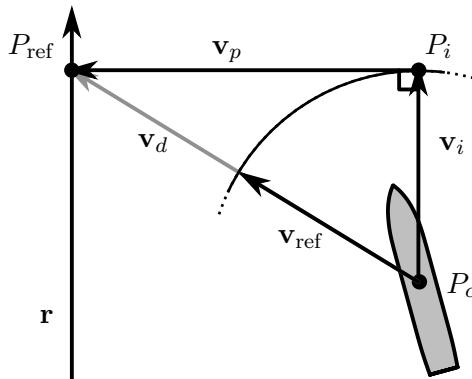


Figure 4.4: Shows the principle of route keeping, by a moving waypoint in the body fixed frame. The point P_c is the origin of the body frame.

\mathbf{v}_p = projected vector, $\mathbf{v}_p \perp \mathbf{v}_i$.

\mathbf{v}_i = intermediate projected vector, $\mathbf{v}_i \parallel \mathbf{r}$.

\mathbf{v}_{ref} = resultant vector that is given to the control system.

\mathbf{v}_d = vector from the vessel to the projected point.

P_{ref} = projected point.

P_i = intermediate projected point.

P_c = centre position of the vessel.

\mathbf{r} = vector representing the route leg.

When the vessel have strayed off the intended route, \mathbf{r} , as it has on figure 4.4, the method of converging back to \mathbf{r} is to follow \mathbf{v}_{ref} . The vector \mathbf{v}_i is a set length towards the next waypoint parallel to the route \mathbf{r} , the length of \mathbf{v}_i depends on the forward speed of the vessel.

The point P_c is found from the GPS coordinates, while the intermediate vector \mathbf{v}_i length is representing the speed of the vessel. By projecting P_i onto

\mathbf{r} , P_{ref} can be determined, as can \mathbf{v}_p since this is the vessels deviation from the route. \mathbf{v}_d is the heading back towards P_{ref} and is defined as shown in equation 4.1.

$$\mathbf{v}_d = \mathbf{v}_i + \mathbf{v}_p = \begin{bmatrix} 0 \\ \mathbf{v}_{i,y} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{p,x} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{p,x} \\ \mathbf{v}_{i,y} \end{bmatrix} \quad (4.1)$$

The length of the vector \mathbf{v}_{ref} is the length of \mathbf{v}_i upon \mathbf{v}_d towards P_{ref} . And as such to shorten the vector \mathbf{v}_d the equation 4.3 is used to find the heading of vector \mathbf{v}_{ref} .

$$|\mathbf{v}_d| = \sqrt{\mathbf{v}_{p,x}^2 + \mathbf{v}_{i,y}^2} \quad (4.2)$$

$$\mathbf{v}_{\text{ref}} = \begin{bmatrix} \mathbf{v}_{p,x} \\ \mathbf{v}_{i,y} \end{bmatrix} \cdot \frac{\mathbf{v}_i}{|\mathbf{v}_d|} \quad (4.3)$$

The vector \mathbf{v}_{ref} is the output of the route keeping, and the input of the controller where its heading is used to determine the thrust from the propeller and the angle of the rudder. Such that the vessel moves towards the intended route \mathbf{r} . Shown on figure 4.5 on the following page is the solution simulated in MATLAB.

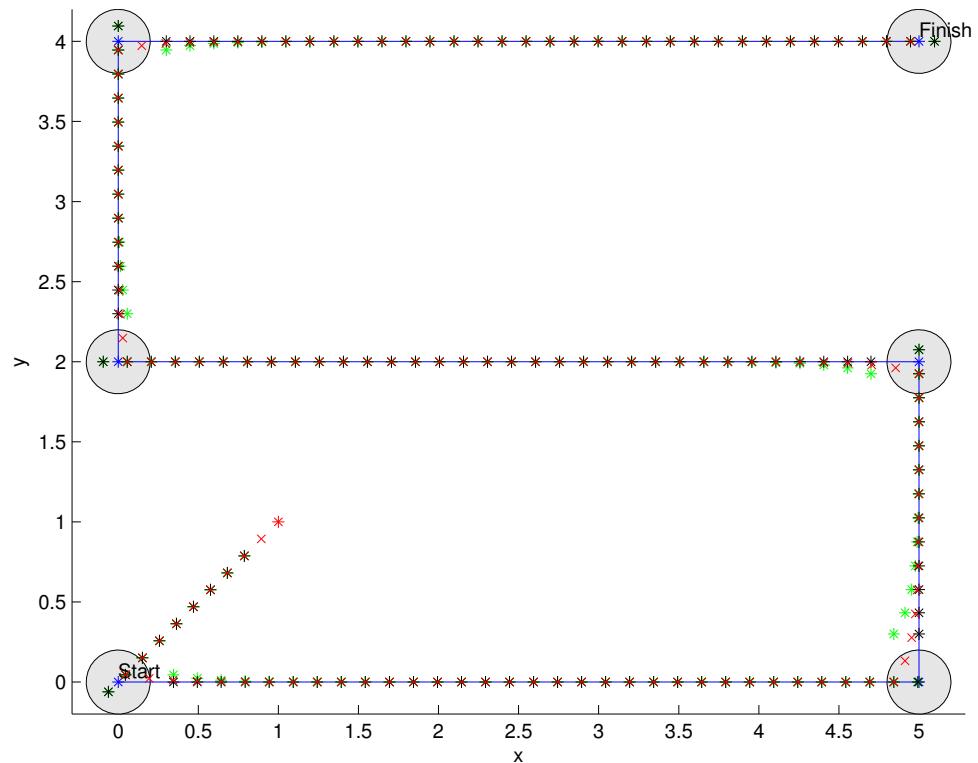


Figure 4.5: Simulation demonstrating the path keeping strategy. It do not take into consideration the vessels limitation to change direction, therefore the vessel can change direction very fast. The red x's is the vessel position (P_c), the green asterisk is the intermediate projected point (P_i) and the black asterisk is the projected point (P_{ref}).

Part II

Modelling

To be able to make a sufficient controller for the system, it is necessary to model the vessel and external disturbances. The full model is divided into several cases, where the models have different approximations regarding the relevant case of movement. Relevant cases is to model the vessel itself and wind and water current disturbances is modeled with simplified assumptions.

5

Reference frames

To be able to make a sufficient controller for the system, it is necessary to model the vessel and external disturbances. The full model is divided into several cases, where the models have different approximations regarding the relevant case of movement.

As of the requirement specification it is the goal to make a vessel that is able to navigate between several waypoints. Since the model and the control is working in different frames of reference, it is important for the design process, to decide in which frames to model in and in which the control system is referenced.

As the GPS tells the absolute position, these coordinates has to be mapped onto a surface frame to use in navigation and maneuvering.

To define the forces acting on the vessel locally, a frame denoted “body-fixed frame” is used, this is written with a prescribed superscript “ b ” as in the following example for a force in the body-fixed frame: $^b\mathbf{F}$. This is used to generate the models that affects the vessel directly, but does not take into account how far the vessel has moved in the navigation frame. To be able to describe the vessel on the surface of the earth another frame called “surface-fixed frame” is defined, and this frame is absolutely defined by the “earth-fixed frame”. These frames is described further in this section and is illustrated on figure 5.1.

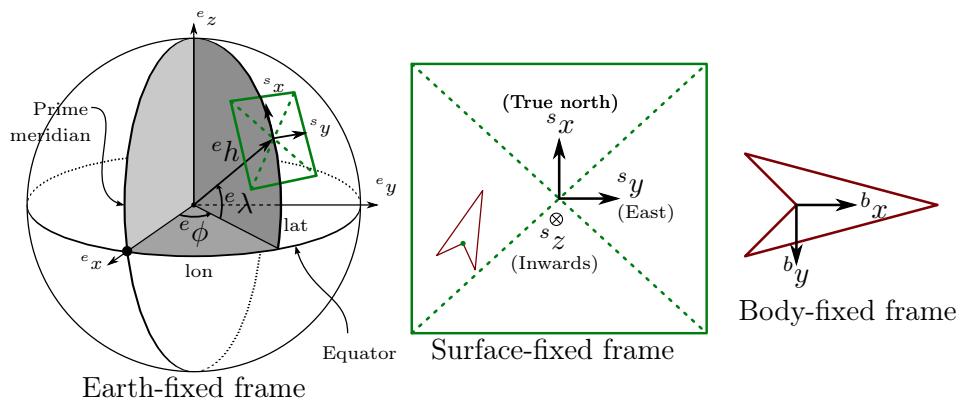


Figure 5.1: Depiction of the reference frames in relation to each other. The labels; lon ($^e\phi$) and lat ($^e\lambda$), is the shorthand form for longitude and latitude, respectively.

5.1 Earth-fixed frame

This frame is a three dimensional polar coordinate system, with its origin in the centre of the earth. It uses the World Geodetic System 84 (WGS84) ellip-

soid to approximate the shape of the earth. This is the bare coordinates that is generated by the GPS devices, that is the latitude and longitude as well as the height i.e. the distance from centre of the earth. This frames also contains an other format to describe points on the earth in a 3 dimensional Cartesian coordinate system called Earth-centred Earth-fixed (ECEF). The relation between ECEF and WGS84 is as depicted on figure 5.1 on the preceding page.

According to the WGS84 geoid the earth is approximated by an ellipsoid of revolution about its minor axis, with the minor axis poles as the north and south pole of the earth. [see Farrel and Barth, 1998, p. 26]

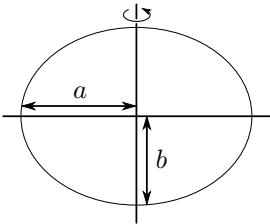


Figure 5.2:
Geodetic ellipse

Semimajor axis length:	$a = 6378137.0$	[m]
Semiminor axis length:	$b = 6356752.3142$	[m]
Eccentricity:	$e = 0.08182$	[–]

Table 5.1: Dimensions of the WGS84 ellipsoid [see Farrel and Barth, 1998, p. 26–27]

5.2 Surface-fixed frame

This frame is a cartesian 2 dimensional frame in respect to navigation, though it might be considered as 3 dimensional cartesian system, when depth samples are added. This frame is a tangent plane to the earth in the middle of the operating area, which is a fair approximation because the area only spans a few hundred metres, whereby the height error between the tangent plane and the curved face of the earth is negligible.

Transformation from earth to surface fixed frame

To be able to transform coordinates in the different reference frames a transformation has to be constructed. For the earth frame to the surface frame, which is a tangent plane on the earth at a fixed point. This point is fixed in relation to the specific mission area the vessel should operate at.

The transformation used is as in [see Farrel and Barth, 1998, p.34–35]. It consists of two rotation matrices that rotates around the z -axis and then the y -axis (\mathbf{R}_z and \mathbf{R}_y). This rotates the points on to the tangent plane. All ECEF points have been translated into the centre of the earth with the tangent point in the centre of the earth. Then this space is rotated with the rotation matrices and the resulting navigation plane is the x - and y -axis of the resulting coordinate system.

$$\mathbf{R}_z = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$\mathbf{R}_y = \begin{bmatrix} -\sin(\lambda) & 0 & \cos(\lambda) \\ 0 & 1 & 0 \\ -\cos(\lambda) & 0 & -\sin(\lambda) \end{bmatrix} \quad (5.2)$$

$$\mathbf{R}_{e2s} = \mathbf{R}_y \mathbf{R}_z = \begin{bmatrix} -\sin(\lambda) \cos(\phi) & -\sin(\lambda) \sin(\phi) & \cos(\lambda) \\ -\sin(\phi) & \cos(\phi) & 0 \\ -\cos(\lambda) \cos(\phi) & -\cos(\lambda) \sin(\phi) & -\sin(\lambda) \end{bmatrix} \quad (5.3)$$

When the ECEF set of coordinates (x, y, z) and the tangent point (x_0, y_0, z_0) is determined, together with the angles for the tangent point, the transformation from the earth to surface plane can be performed as in equation 5.4 when these is known.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_s = \mathbf{R}_{e2s} \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}_e - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_e \right) \quad (5.4)$$

WGS84 to ECEF

The ECEF system is derived from WGS84 using the following formulas:

$$N(\phi) = \frac{a}{\sqrt{1 - e^2 \cdot \sin^2(\phi)}} \quad (5.5)$$

$$X = (N(\phi) + h) \cdot \cos(\phi) \cdot \cos(\lambda) \quad (5.6)$$

$$Y = (N(\phi) + h) \cdot \cos(\phi) \cdot \sin(\lambda) \quad (5.7)$$

$$Z = (N(\phi) \cdot (1 - e^2) + h) \cdot \sin(\phi) \quad (5.8)$$

The constants (a and e) are listed in the table 5.1 on the facing page.

5.3 Body-fixed frame

The body fixed frame is based on a coordinate system centred in the centre of gravity of the vessel. This makes way for a coordinate system depicted in the following figure 5.3 on the next page.

Transformation between surface - and body fixed frame

Since the body frame simply is a rotations of the the surface frame, this simply equates to a clockwise rotation matrix about the z-axis.

$$\mathbf{R}_{s2b} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

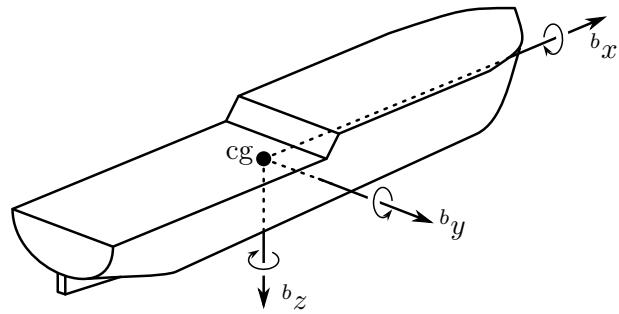


Figure 5.3: Depiction of the body frame

The transformation from the body frame to surface frame is the surface frame to body frame transposed, as seen in equation 5.10.

$$\mathbf{R}_{b2s} = \mathbf{R}_{s2b}^T = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

6

Dynamics

The body frame makes the basis for the mathematical models computed throughout this chapter as this frame is inertial, thus making it possible to map the different forces acting on the vessel.

6.1 Forward motion

The first force calculated, are the one acting on the vessel along the x-axis. Ideally there are only two forces acting along this axis, namely the thrust generated by the vessel, and the skin-friction it experiences as it is moving through the water. A general formula for the thrust generated by the vessel has been found in [Soerensen, 2005] and is given as:

$$F_{\text{thrust}} = \rho_{\text{water}} \cdot D^4 \cdot K_T \cdot |n| \cdot n \quad (6.1)$$

Where:

ρ_{water} = the density of water

D = the diameter of the propeller

K_T = a propeller coefficient

n = the number of revolutions of the propeller

The only unknown in this equation is K_T which is a quality factor for the propeller. However, as the formula to calculate this requires a lot of constants which can not be measured with the available equipment, the formula 6.1 can be rewritten to the following which can give an estimate of the propeller quality at full thrust.

$$K_T = \frac{F_{\text{thrust}}}{\rho_{\text{water}} \cdot D^4 \cdot |n| \cdot n} \quad (6.2)$$

This measurement is carried out in the appendix on page C on page 97. The value for K_T is then calculated to be 0.2527. As the vessel experiences skin-friction when it moves through the water, this needs to be calculated as well, so an expression for the total force acting along the x-axis can be computed. The formula for calculating friction is given as:

$$F_{\text{drag}} = \frac{C_D \cdot \rho_{\text{water}} \cdot A_{\text{hull}} \cdot \dot{x}^2}{2} \quad (6.3)$$

Where:

C_D = a drag coefficient

ρ_{water} = the density of the fluid

A_{hull} = the cross sectional area of the submerged part of the vessel

\dot{x} = the velocity of the vessel

In this formula, the only unknown is the drag coefficient, which is assumed to be 1.25 as a measurement test for this have not been carried out, the assigned value have been set as per [NASA]. These two formulas are then combined and the total force acting on the vessel along the x-axis can then be given as:

$$F_x = -F_{\text{drag}} + F_{\text{thrust}} \quad (6.4)$$

As it is sought to develop a controller for the speed, the force is converted into an acceleration using Newtons second law $F = m \cdot \ddot{x}$. Which gives the following function:

$$\ddot{x} = \frac{F_y}{m_{\text{vessel}}} = \frac{1}{m_{\text{vessel}}}(-F_{\text{drag}} + F_{\text{thrust}}) \quad (6.5)$$

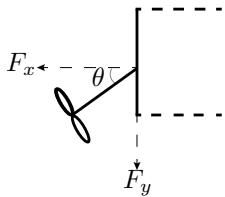
Where:

m_{vessel} = the mass of the vessel

Equation 6.5 is the differential model of the vessel in the local body frame calculating accelerations in the x-direction.

6.2 Turning motion

As the propeller is mounted directly on the rudder (a term called thrust vectoring) the total force acting on the vessel can be seen as the resultant vector of the forces acting on the vessel in the x- and y-direction, which can be written as:



$$F_{\text{total}} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} F_{\text{thrust}} \quad (6.6)$$

As the first component in equation 6.6 does not generate any contribution to the yaw-motion of the vessel, this can be omitted. And the only term generating a torque can be given as:

$$F_{\text{yaw}} = \sin(\theta) \cdot F_{\text{thrust}} \quad (6.7)$$

As turning motions are given as torques equation 6.7 is converted to a torque by multiplying the force with the distance from the centre of gravity (l_{cog}), giving:

$$\tau_{\text{yaw}} = l_{\text{cog}} \cdot \sin(\theta) \cdot F_{\text{thrust}} \quad (6.8)$$

Figure 6.1:
Depiction of forces and angle on the rudder

Which then is converted to an angular acceleration by rewriting $\tau = I \cdot \ddot{\varphi}$, which gives the following:

$$\ddot{\varphi} = \frac{\tau_{yaw}}{I} = \frac{1}{I} (l_{cog} \cdot \sin(\theta) \cdot F_{thrust}) \quad (6.9)$$

Where the inertia of the vessel (I) is given as:

$$I = \frac{1}{12} m_{vessel} (w_{vessel}^2 + l_{vessel}^2) \quad (6.10)$$

Where:

w_{vessel} = the width of the vessel

l_{vessel} = the length of the vessel

Which makes the differential equation for the vessel. The last thing to model is the rotational drag which prevents the vessel from keep rotating. This is caused by one of two things. One is the wind that pushes the vessel the other is when the vessel is meant to turn. The rotational drag is a torque around the z-axis opposite of the torque generated by the rudder (the wind torque can be positive though).

The formula used to determine the amount of force acting opposite (later the torque) is based on the general formula for skin-frictional drag, which is given in equation 6.3 on page 25 and gives the drag as a function of the velocity of the vessel \dot{x} . The velocity for rotating bodies, can be equated to the length timed the angular velocity giving:

$$F_\varphi = \frac{C_D \cdot \rho_{water} \cdot A_{water} \cdot (r \cdot \dot{\varphi})^2}{2} \quad (6.11)$$

Where:

C_D = The drag coefficient

ρ_{water} = The density of water

A_{water} = The area of the vessel under water

r = The distance to the hull

$\dot{\varphi}$ = The rotational velocity of the vessel

As the frictional drag in this case is given as a force, it is multiplied with the distance to the point of attack, as this converts it to a torque. This gives the following equation:

$$\tau_\varphi = \frac{C_D \cdot \rho_{water} \cdot A_{water} \cdot (r \cdot \dot{\varphi})^2}{2} \cdot r \quad (6.12)$$

As seen in equation 6.12 the vessels area is seen as one piece, however this is not a useful approximation in this case so the vessels side is divided into

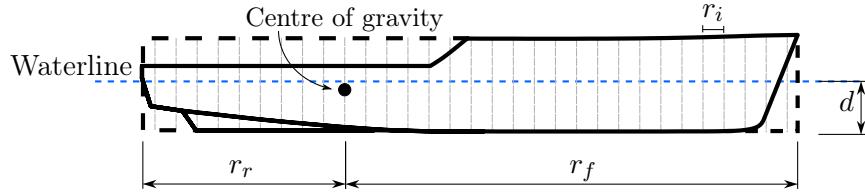


Figure 6.2: Sketch of how the vessel is divided into an infinite number of vertical strips below the waterline

infinitely many small strips, each one giving an addition to the frictional drag. The approximation is that the vessel is considered to be a box. See figure 6.2 for an illustration of the division of the vessel.

The formula for calculating the frictional drag reduces to two integrals ranging from zero to the radius. The radius is divided into two, as the force acting on the vessel is different in the front and the rear end. The integrals are given as:

$$\begin{aligned}\tau_\varphi = & \int_0^{r_f} \left(\frac{1}{2} \cdot C_D \cdot (r_f \cdot \dot{\varphi})^2 \cdot \rho_{\text{water}} \cdot d \cdot r_f \right) dr \\ & + \int_0^{r_r} \left(\frac{1}{2} \cdot C_D \cdot (r_r \cdot \dot{\varphi})^2 \cdot \rho_{\text{water}} \cdot d \cdot r_r \right) dr\end{aligned}\quad (6.13)$$

Where:

r_f = The radius to the front section

r_r = The radius to the rear section

d = The depth of the vessel below the waterline

As the vessel is approximated by a box, solving the integrals give a constant and a variable term, thus giving:

$$\tau_\varphi = \underbrace{\frac{1}{8} \cdot C_D \cdot \rho \cdot d(r) \cdot (r_f^4 + r_r^4) \cdot \dot{\varphi}^2}_{\text{constant}} \quad (6.14)$$

Which makes to rewrite the equation as a constant term multiplied with the variable, making the torque a function of the rotational velocity::

$$\tau_\varphi(\dot{\varphi}) = C_R \cdot \dot{\varphi}^2 \quad (6.15)$$

Where:

$$C_R = \frac{1}{8} \cdot C_D \cdot \rho \cdot d(r) \cdot (r_f^4 + r_r^4)$$

As the other forces are converted into accelerations equation 6.15 needs to be as well. This is done using $\tau_\varphi = I \cdot \ddot{\varphi}$, and gives the following differential equation for the rotational drag:

$$\ddot{\varphi}(\dot{\varphi}) = \frac{1}{I} (C_R \cdot \dot{\varphi}^2) \quad (6.16)$$

Wind and water disturbances

7

In order to accurately model the environmental influence on the vessel it is necessary to model the wind and water current disturbances. This chapter will describe how this is done considering the simplifications used. Models will be constructed so that the wind and water can be simulated. The physical dimensions of the vessel is described in appendix E on page 107

7.1 Simplifications

To model the wind effects on the vessel from different angles of attack and different wind speeds some assumptions and simplifications are made, since the actual vessel shape is hard to described precisely.

- It is assumed the wind speed is constant and only affecting the vessel in one direction at one specific time.
- The wind direction is considered to be constant in one direction to find the newton force affecting the vessel at each time step. The direction is defined as the heading the wind is progressing
- The wind direction is considered to be a vector for each time step, given by a angle of attack and a velocity.
- The wind is considered to be 10 cm above sea level and the water is considered to be a smooth surface
- Air density is considered to be constant and the same everywhere
- The vessel is considered to be a box with smooth surfaces
- The wind is considered to be a laminar flow
- It is assumed that the area affected by the wind do not change due to listing

7.2 Model of wind vector

There are two parameters given as inputs, these are the wind speed and angle of attack. These are denoted $|\mathbf{w}|$ and θ respectively. To distinguish these parameters in the different frames, they are marked with a prescribed superscript. Therefore the following variables will be used.

- $^l\mathbf{w}$ = wind vector in the local frame
- $^b\mathbf{w}$ = wind vector in the body frame
- $^l\theta_w$ = wind angle of attack in the local frame
- $^b\theta_w$ = wind angle of attack in the body frame

${}^b\theta_v$ = vessel angle in reference to the ${}^l x$ -axis

As the wind vector in the local frame is defined by a wind speed (length = $|{}^l \mathbf{w}|$) and an angle (according to the ${}^l x$ -axis = ${}^l \theta_w$), it can be written in vector form as:

$${}^l \mathbf{w} = \begin{bmatrix} \cos({}^l \theta_w) \\ \sin({}^l \theta_w) \end{bmatrix} \cdot |{}^l \mathbf{w}| \quad (7.1)$$

Equation 7.1 can be transformed from the local frame to the body frame with a rotation matrix (counter clock wise rotation), as follows:

$${}^b \mathbf{w} = \begin{bmatrix} \cos({}^b \theta_v) & -\sin({}^b \theta_v) \\ \sin({}^b \theta_v) & \cos({}^b \theta_v) \end{bmatrix} \cdot {}^l \mathbf{w} \quad (7.2)$$

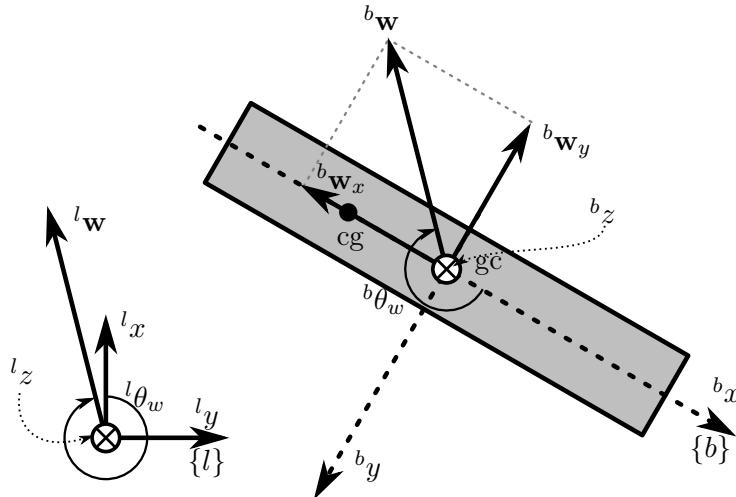


Figure 7.1: Relation of wind vectors on the local frame and body frame.

7.3 Model of wind force

The force applied to the vessel is the wind pressure exerted on an area of the vessel, which can be described as $F = A \cdot P_{wind}$ where A is the affected area and P is the wind pressure. This pressure is the pressure at a specific wind speed. The pressure exerted on the vessel can be described as a function of wind speed, density of air and the shape of the affected area (the drag coefficient): $P = \frac{1}{2} \cdot \rho_{air} \cdot V^2 \cdot C_d$. Shown in equation 7.3 on the facing page is the formula

for the drag load in the x -direction. This formula applies to sway and yaw as well with the respective constants.

$$F_x = \frac{1}{2} \cdot C_x \cdot \rho \cdot V^2 \cdot A_x \quad (7.3)$$

Where:

F_x = force applied to the surface in the x direction

C_x = drag coefficient in the x direction

ρ = density of air

V = relative velocity of air past the vessel

A_x = affected area in the x direction

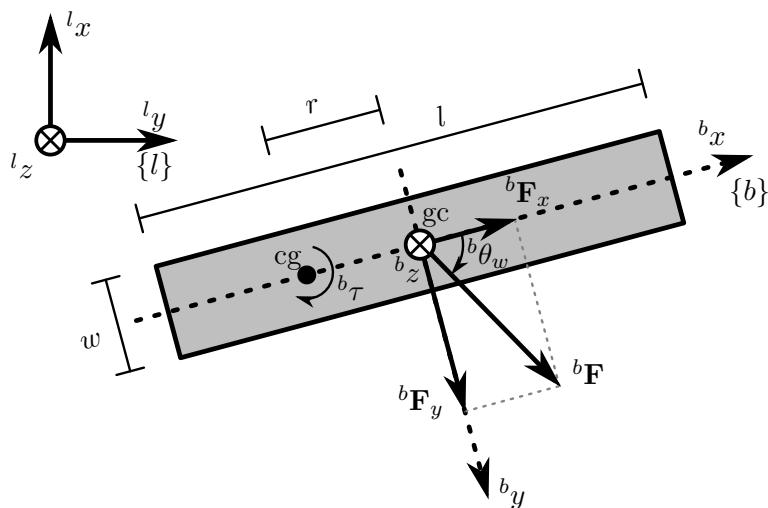


Figure 7.2: Forces acting upon the simplified vessel. Illustrating all geometric related variables in the body-fixed-frame, together with the relation with the local-fixed-frame.

The density ρ is defined as $1.225 \frac{kg}{m^3}$ at sea level and at 15°C , see [ISO, 1975]. $C_x \approx 2.0$ for flat plates perpendicular to the flow, see [The Engineering Toolbox, 2012].

As the vessel is considered to be a box, See figure 7.1 on page 29, the force affecting the vessel can be divided into two smaller components as shown in figure 7.2 resulting in the equations shown in equation 7.4 respectively the x - and y -direction of the vessel. A wind perpendicular to the vessel will however also cause a rotation around the centre of gravity resulting in a torque.

$$\begin{bmatrix} {}^b\mathbf{F}_x \\ {}^b\mathbf{F}_y \\ {}^b\mathbf{F}_\tau \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}_x|^2 \cdot A_x \\ \frac{1}{2} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}_y|^2 \cdot A_y \\ \frac{1}{2} \cdot C_d \cdot \rho \cdot (|{}^b\mathbf{w}_y| \cdot r_{gc-cg})^2 \cdot A_x \end{bmatrix} \quad (7.4)$$

Where:

- ${}^b\mathbf{F}$ = Force applied to the surface in the x , y and rotational direction
- C_d = Drag coefficient of a box ≈ 2
- ρ = Density of air
- $|{}^b\mathbf{w}|$ = Wind speed
- $A_{x(y)}$ = Affected area in the given direction
- r_{gc-cg} = The arm from the centre of geometry to the centre of gravity

Change in acceleration due to wind

Because the wind is continuously broken down into x - and y -components which is rotated into the body frame see section 7.3 on page 30, only ${}^b\mathbf{F}_y$ is relevant when determining the torque exerted by the wind. Using $\tau = F \cdot r$ the torque caused by the wind can be described as equation 7.5

$$\tau = \frac{1}{2} \cdot C_d \cdot \rho \cdot (|{}^b\mathbf{w}| \cdot r_{gc-cg})^2 \cdot A_y \cdot r_{gc-cg} \quad (7.5)$$

As the torque is dependent on the length of the arm r_{gc-cg} this is integrated along the length of r_{gc-cg} as shown in equation 7.6 resulting in equation 7.7.

$$\tau = \int_0^r \frac{1}{2} \cdot C_d \cdot \rho \cdot (|{}^b\mathbf{w}| \cdot r)^2 \cdot A_y \cdot r_{gc-cg} \, dr \quad (7.6)$$

$$\tau = \frac{1}{8} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}|^2 \cdot A_y \cdot r_{gc-cg}^4 \quad (7.7)$$

The acceleration is found as $\ddot{\varphi} = \tau/I$, where the moment of inertia can be calculated as $I = \frac{1}{12} \cdot m \cdot (w_{vessel}^2 + l_{vessel}^2)$, which becomes a constant. Similar the movement in the x direction of the body coordinate system can be described according to Newtons law: $F = m \cdot a$, which gives the equations shown in equation 7.8

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{\frac{1}{2} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}_x|^2 \cdot A_x}{m} \\ \frac{\frac{1}{2} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}_y|^2 \cdot A_y}{m} \\ \frac{\frac{1}{8} \cdot C_d \cdot \rho \cdot |{}^b\mathbf{w}_y|^2 \cdot A_y \cdot r_{gc-cg}^4}{\frac{1}{12} \cdot m \cdot (w_{vessel}^2 + l_{vessel}^2)} \end{bmatrix} \quad (7.8)$$

Inserting the following values into the equations in equation 7.8 reduces the equations to equation 7.9 on the facing page.

$$\begin{aligned} C_D &= 2 & \rho &= 1.225 & r &= 0.08 & A_x &= 0.0182 \\ A_y &= 0.097 & m &= 3.7 & w_{vessel} &= 0.19 & l_{vessel} &= 0.95 \end{aligned}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{0.0222950}{3.7} \cdot |^b\mathbf{w}_x|^2 \\ \frac{0.118825}{3.7} \cdot |^b\mathbf{w}_y|^2 \\ \frac{0.000001}{0.289402} \cdot |^b\mathbf{w}_y|^2 \end{bmatrix} \quad (7.9)$$

Equation 7.9 describes the velocity added or subtracted from the vessel's velocity depending on wind angle and velocity, and is simulated in section 7.3.

Water current disturbance

The current is very similar to wind in regards of the way that it acts on the vessel. As with the wind, the current is considered to be laminar and constant. As such the acceleration caused by the current in a random direction can be estimated in the same way as the wind giving the equations seen in 7.10.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot C_d \cdot \rho \cdot |^b\mathbf{v}_{current,x}|^2 \cdot A_x \\ \frac{1}{2} \cdot C_d \cdot \rho \cdot |^b\mathbf{v}_{current,y}|^2 \cdot A_y \\ \frac{1}{8} \cdot C_d \cdot \rho \cdot |^b\mathbf{v}_{current,y}|^2 \cdot A_y \cdot r_{gc-cg}^4 \\ \frac{1}{12} \cdot m \cdot (w_{vessel}^2 + l_{vessel}^2) \end{bmatrix} \quad (7.10)$$

Inserting the following values in the equations reduces them to 7.11:

$$\begin{aligned} C_D &= 2 & \rho &= 1000 & r &= 0.08 & A_x &= 0.0095 \\ A_y &= 0.04 & m &= 3.7 & w_{vessel} &= 0.19 & l_{vessel} &= 0.80 \end{aligned}$$

It can be seen that the constant part of the reduced equations are greater than the ones shown in 7.9, and as such a current should have far greater effect on the heading of the vessel.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{9.5}{3.7} \cdot |^b\mathbf{v}_x|^2 \\ \frac{40}{3.7} \cdot |^b\mathbf{v}_y|^2 \\ \frac{0.00041}{0.20846} \cdot |^b\mathbf{v}_\varphi|^2 \end{bmatrix} \quad (7.11)$$

Wind simulations

Through this section the wind and water currents are simulated. The simulations are based on equation 7.9 for wind and 7.11 for water current. The simulation is constructed in Simulink in MATLAB.

Shown on figure 7.3 on the following page is the simulation of the change in position caused by the wind. Only the simulation of position in x -direction is shown as the model is similar, only with different constants.

The input is a representation of the x -component of the wind vector that are rotated from surface frame to the body frame. Similar is the output position rotated back into the local frame. On figure 7.4 on the next page the position in the x - and y - direction in the local frame is shown at a wind speed of 5 m/s and at a wind angle of 0.15π rad.

Due to the short rotational arm from centre of gravity and centre of geometry the rotational acceleration and velocity are negligible.

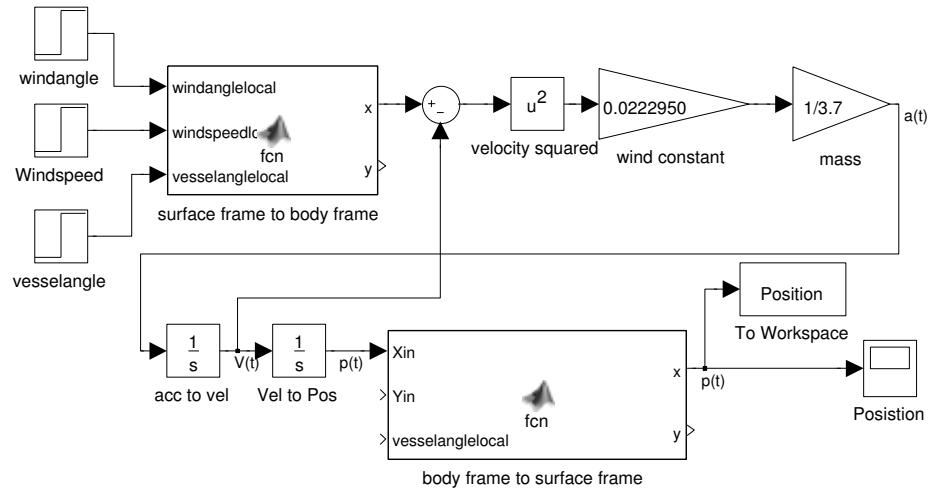


Figure 7.3: Simulation model of the velocity in the x -direction with a step in wind velocity

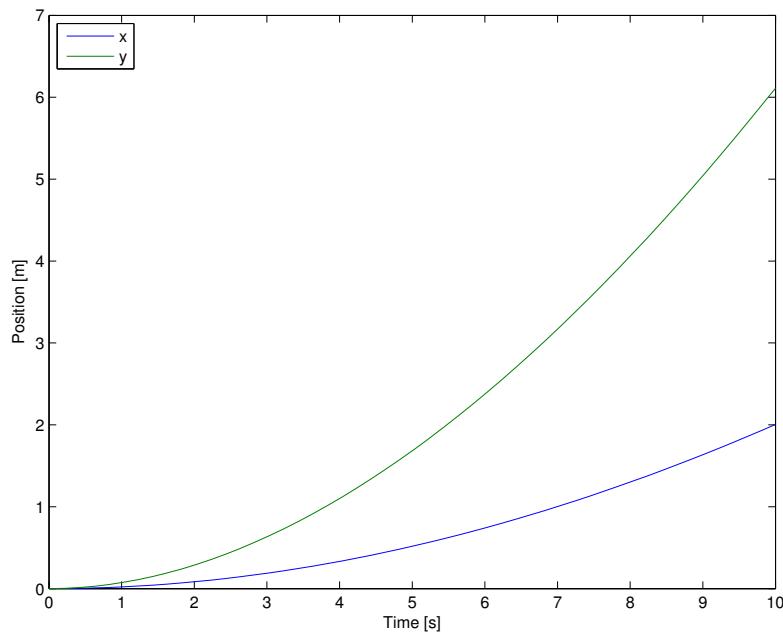


Figure 7.4: Vessel position when affected by wind in both the x - and y -direction of the surface frame.

8

DC motor

As the system thrust is generated by a DC motor, this is modeled in this chapter. This is based around the voltage as an input, and the rotational velocity as an output. The modeling is split into two parts, one regarding the electrical aspects and another one about the mechanical aspects of the motor.

8.1 Electrical part

The electrical model of the DC motor is made of a circuit consisting of a resistor, an inductor and a generator. The system is depicted on figure 8.1.

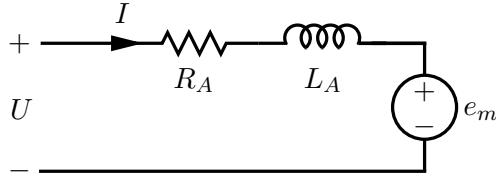


Figure 8.1: Electric model of the DC motor

Applying Kirchoffs laws on the electrical system, yields the following equation for the voltage U .

$$U(t) = R_A \cdot I(t) + L_A \frac{\partial I(t)}{\partial t} - e_m \quad (8.1)$$

Where:

e_m = the electromotoric force, given by: $K_m \cdot \omega(t)$ given in [?]

K_m = a motor constant

$\omega(t)$ = the rotational velocity [rad/s]

Inserting the electromotoric force in equation 8.1 gives a full expression for the voltage as described in equation 8.2.

$$U(t) = R_A \cdot I(t) + L_A \frac{\partial I(t)}{\partial t} - K_m \cdot \omega(t) \quad (8.2)$$

This concludes the electrical modeling of the system and the mechanical is now modelled.

8.2 Mechanical part

The mechanical modeling of the DC motor can be viewed as the free-body diagram on figure 8.2 on the next page To make the engine turn, the engine

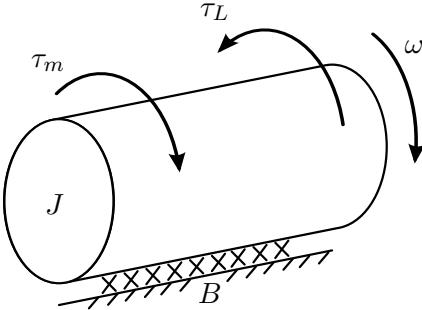


Figure 8.2: Mechanical freebody diagram of the DC motor

must overcome the load torque τ_L and the friction B . The function can be written as:

$$J \frac{d\omega(t)}{dt} = \tau_m - (\tau_f + \tau_L) \quad (8.3)$$

Where:

τ_m = The “positive” torque generated by the DC motor

τ_f = The friction inside the DC motor B on figure 8.2

τ_L = The torque generated by the propeller rotating in the water

J = The friction of the DC motor

The load generated by the propeller can using [Soerensen, 2005] be given as a function of the rotational as seen in equation 8.4

$$\tau_L = \rho_{\text{water}} \cdot D^5 \cdot K_q \cdot |n| \cdot n \quad (8.4)$$

Where:

ρ_{water} = the density of the water

D = the diameter of the propeller

K_q = a torque constant

n = the rotational velocity [rps]

In equation 8.4 the only variable is n thus reducing the equation to $|n| \cdot n \cdot C_q$. C_q has in [Dam et al., 2011] been estimated to 0.0128. This equates the function for τ_L to equation 8.5:

$$\tau_L(n) = 0.000004 \cdot n^2 \quad (8.5)$$

As this gives a nonlinear system, the system is linearized as described in section 9 on page 39 which gives the following expression for the load:

$$\tau_L(n) \approx 0.00044n - 0.0121 \quad (8.6)$$

The torque generated by the motor can be [Close et al., 2002, page 347] be written as the motor constant K_m multiplied with the current $I(t)$, and the friction can be given as a constant multiplied with the rotational velocity, thus stating:

$$\tau_m = K_m \cdot I(t) \quad (8.7)$$

$$\tau_f = B \cdot \omega(t) \quad (8.8)$$

When inserted in equation 8.3 on the preceding page the differential equation for the mechanical part of the DC motor becomes:

$$J \frac{\partial \omega(t)}{\partial t} = K_m \cdot I(t) - (B \cdot \omega(t) + 0.00044 \cdot \omega(t)) \quad (8.9)$$

The motor parameters are measured and calculated in appendix D on page 101.

8.3 Transfer function of the DC motor

To establish a transfer function for the system, the two equations are Laplace transformed. As the propeller is not moving when no current is applied, and the propeller is not revolving at time 0, the initial conditions are zero, thus giving the two transformed functions as:

$$U(s) = R_A \cdot I(s) + L_A \cdot s \cdot I(s) - K_m \cdot \omega(s) \quad (8.10)$$

$$J \cdot s \cdot \omega(s) = K_m \cdot I(s) - (B \cdot \omega(s) + 0.00044 \cdot \omega(s)) \quad (8.11)$$

The system can now be modeled using block diagrams, giving the system seen on figure 8.3. For the sake of simplicity the figure contains two boxes named

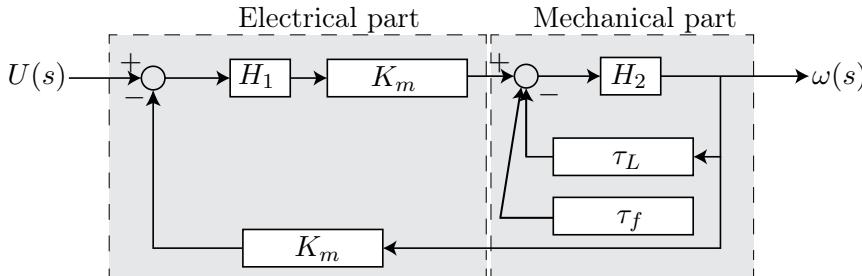


Figure 8.3: Combined model for the system

H_1 and H_2 . The function in H_1 converts an input voltage to a torque and H_2 converts an input torque to a rotational velocity (in revolutions per second).

$$H_1 = \frac{1}{R_A + L_A \cdot s} \quad (8.12)$$

$$H_2 = \frac{1}{J \cdot s \cdot 2\pi} \quad (8.13)$$

The system on figure 8.3 on the preceding page can then be converted to a single transfer function by calculating the closed loop transfer functions from the inside out. The closed loop transfer function for the mechanical part (H_2 and τ_{friction}) is calculated by inserting the known values of J and B and then gives:

$$H_{\text{mech}}(s) = \frac{\frac{H_2}{1+(H_2 \cdot \tau_L)}}{1 + (\frac{H_2}{1+H_2 \cdot \tau_L} \cdot \tau_f)} \quad (8.14)$$

The last thing to do is to add the closed loop transfer function for the mechanical part, this giving:

$$H_{\text{motor}}(s) = \frac{H_1(s) \cdot K_m \cdot H_{\text{mech}}(s)}{1 + (H_1(s) \cdot K_m^2 \cdot H_{\text{mech}}(s))} \quad (8.15)$$

Which, when numbers are inserted, reduces to the following transfer function.

$$H_{\text{motor}}(s) = 327.94 \frac{1}{0.0000576s^2 + 3.644s + 33.341} \quad (8.16)$$

9

Linearization

As the system is nonlinear (i.e. the $n \cdot |n|$, \dot{x}^2 , $\sin \theta$ and $\cos \theta$) the system needs to be linearized. The linearizations is described throughout this chapter.

This is done by using the first order Taylor-series expansion. The Taylor expansion is given as:

$$f(x) \approx f(x_0) + \frac{\partial}{\partial x} f(x_0)(x - x_0) \quad (9.1)$$

Which can be rewritten to:

$$f(x) \approx \underbrace{f(x_0) - \frac{\partial}{\partial x} f(x_0) \cdot x_0}_{\text{constant}} + \underbrace{\frac{\partial}{\partial x} f(x_0) \cdot x}_{\text{variable}} \quad (9.2)$$

As seen in equation 9.2 the linearized function now contains a constant and a variable part. The next step is to insert the previously calculated expressions for the forward acceleration. As the vessel should sail forward and be able to turn, the expression is broken down into two parts. One evaluating the forward motion and one evaluating the turning motion of the vessel.

$$\begin{bmatrix} \ddot{x}(n, \theta, \dot{x}) \\ \ddot{\varphi}(n, \theta, \dot{\varphi}) \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cdot n \cdot |n| \cdot C_F - C_M \cdot \dot{x}^2 \\ \sin(\theta) \cdot n \cdot |n| \cdot C_T - C_R \cdot \dot{\varphi}^2 \end{bmatrix} \quad (9.3)$$

Where:

$$\begin{aligned} C_F &= \frac{1}{m_{\text{vessel}}} (\rho_{\text{water}} \cdot D^4 \cdot K_T) = 0.000036081 \\ C_T &= \frac{1}{I} (l_{\text{cog}} \cdot \rho_{\text{water}} \cdot D^4 \cdot K_T) = 0.0016 \\ C_M &= \frac{C_D \cdot A_{\text{hull}} \cdot \rho_{\text{water}}}{2} = 3.7500 \\ C_R &= \frac{\frac{1}{8} \cdot C_D \cdot \rho \cdot d(r) \cdot (r_f^4 + r_r^4)}{I} = 0.0784 \end{aligned}$$

As the linearization is done around a linearization point, these have to be estimated. As the requirements specifies that the vessel should be able to sail in a straight line with a velocity of 1 m/s - these constants can be given as: $n = n_0 = 55$, and $\theta = \theta_0 = 0$ and $\dot{x} = \dot{x}_0 = 1$. The turning motion however, is calculated with other constants, as the vessel hereby is influenced by the rudder not being in the centre e.g. $\theta = \theta_0 \neq 0$. The rudder is measured to have a maximum angle of 20 degrees, and the linearization will therefore be done around half of this, giving $\theta = \theta_0 = 10^\circ$. The angular velocity is then simulated to be around 10.07, which is then used as a linearization point in the equation, thus: $\dot{\varphi} = \dot{\varphi}_0 = 10.07^\circ/\text{s}$. As these constants are different from each linearization, the final result should yield two results, one that is linearized around the forward motion for the ship, and one that is linearized

around the turning motion. When the general motion equations are inserted into equation 9.2 on the preceding page the expressions become:

$$\begin{bmatrix} \ddot{x}(n, \theta, \dot{x}) \\ \ddot{\varphi}(n, \theta, \dot{\varphi}) \end{bmatrix} \approx \underbrace{\begin{bmatrix} \ddot{x}(n_0, \theta_0, \dot{x}_0) \\ \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \end{bmatrix}}_{\text{constant}} - \nabla \begin{bmatrix} \ddot{x}(n_0, \theta_0, \dot{x}_0) \\ \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \end{bmatrix} \cdot \begin{bmatrix} n_0, \theta_0, \dot{x}_0 \\ n_0, \theta_0, \dot{\varphi}_0 \end{bmatrix} \\ + \underbrace{\nabla \begin{bmatrix} \ddot{x}(n_0, \theta_0, \dot{x}_0) \\ \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \end{bmatrix}}_{\text{variable}} \cdot \begin{bmatrix} n, \theta, \dot{x} \\ n, \theta, \dot{\varphi} \end{bmatrix} \quad (9.4)$$

The two equations given in 9.4 can be re-written to:

$$\begin{bmatrix} \ddot{x}(n, \theta, \dot{x}) \\ \ddot{\varphi}(n, \theta, \dot{\varphi}) \end{bmatrix} \approx \underbrace{\begin{bmatrix} \ddot{x}(n_0, \theta_0, \dot{x}_0) \\ \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \end{bmatrix}}_{\text{constant}} - \mathbf{K} \cdot \begin{bmatrix} n_0, \theta_0, \dot{x}_0 \\ n_0, \theta_0, \dot{\varphi}_0 \end{bmatrix} \\ + \underbrace{\mathbf{K} \cdot \begin{bmatrix} n, \theta, \dot{x} \\ n, \theta, \dot{\varphi} \end{bmatrix}}_{\text{variable}} \quad (9.5)$$

Where the vector \mathbf{K} is given by:

$$\begin{aligned} \mathbf{K} &= \nabla \begin{bmatrix} \ddot{x}(n_0, \theta_0, \dot{x}_0) \\ \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \end{bmatrix} \\ &= \left[\frac{\partial}{\partial n_0} \ddot{x}(n_0, \theta_0, \dot{x}_0) + \frac{\partial}{\partial \theta_0} \ddot{x}(n_0, \theta_0, \dot{x}_0) + \frac{\partial}{\partial \dot{x}_0} \ddot{x}(n_0, \theta_0, \dot{x}_0) \right. \\ &\quad \left. \frac{\partial}{\partial n_0} \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) + \frac{\partial}{\partial \theta_0} \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) + \frac{\partial}{\partial \dot{\varphi}_0} \ddot{\varphi}(n_0, \theta_0, \dot{\varphi}_0) \right] \end{aligned} \quad (9.6)$$

The constant values are now computed to be:

$$\mathbf{m}_f = \begin{bmatrix} 209.7817 \\ 4.3904 \end{bmatrix} \quad (9.7)$$

$$\mathbf{m}_t = \begin{bmatrix} 210.4428 \\ 1.8913 \end{bmatrix} \quad (9.8)$$

For the purpose of linearizing the functions these constant terms can be omitted. In these functions there is an added y component, as the wind affecting the vessel will affect it in this direction. As seen the 3-by-2 matrix does not contain any value in the second row, which is because this motion cannot be controlled by any of the inputs. To control this the propulsion device could be changed to an azipod thruster. The calculations for the variable terms are put

on matrix form, this is done to ease the state space transformation used later.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix}_f = \begin{bmatrix} -7.3909 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.1468 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 3.7460 & 3.75 \\ 0 & 0 \\ 0.0784 & -4.7616 \end{bmatrix} \begin{bmatrix} n \\ \theta \approx 0 \end{bmatrix} \quad (9.9)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \end{bmatrix}_t = \begin{bmatrix} -7.3925 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.6837 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 3.7461 & 3.769 \\ 0 & 0 \\ 0.0478 & -4.6881 \end{bmatrix} \begin{bmatrix} n \\ \theta \neq 0 \end{bmatrix} \quad (9.10)$$

The two matrix systems 9.9 and 9.10 gives the linearized versions of the motion systems that the vessel uses. Equation 9.9 expresses the movement when the ship is moving forward (hence $\theta \approx 0$), while the equation in 9.10 gives the motion when the ship is turning ($\theta \neq 0$).

The calculations for these constants are carried out in the MATLAB file found in [matlab/linearization/lineraization.m](#)

State space representation

10

This chapter is describing how a system can be modelled when written in a specific format called state space. This notation is widely used in modern control.

The state space representation can generally be given as:

$$\dot{\mathbf{x}} = \underbrace{\mathbf{A} \cdot \mathbf{x}}_{\text{states}} + \underbrace{\mathbf{B} \cdot \mathbf{u}}_{\text{inputs}} + \underbrace{\mathcal{G}_{\text{water}} \cdot \mathbf{w} + \mathcal{G}_{\text{current}} \cdot \mathbf{v}}_{\text{wind and current disturbances}} \quad (10.1)$$

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{x} \quad (10.2)$$

Where the matrix \mathbf{A} (the state matrix) contains the different states the system can change between, being forward motion, sideways motion or turning motion.

The systems are the same as computed in section 9 on page 39.

$$\mathbf{A}_{\text{forward}} \cdot \mathbf{x} = \begin{bmatrix} -7.3909 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.1468 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (10.3)$$

$$\mathbf{A}_{\text{turning}} \cdot \mathbf{x} = \begin{bmatrix} -7.3925 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.6837 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (10.4)$$

The matrix \mathbf{B} is the controllable elements in the system, meaning the inputs that is used to make the vessel go forward or turn. These are also the same as previously calculated.

$$\mathbf{B}_{\text{forward}} \cdot \mathbf{u} = \begin{bmatrix} 3.7460 & 3.75 \\ 0 & 0 \\ 0.0784 & -4.7616 \end{bmatrix} \cdot \begin{bmatrix} n \\ \theta \end{bmatrix} \quad (10.5)$$

$$\mathbf{B}_{\text{turning}} \cdot \mathbf{u} = \begin{bmatrix} 3.7461 & 3.769 \\ 0 & 0 \\ 0.0478 & -4.6881 \end{bmatrix} \cdot \begin{bmatrix} n \\ \theta \end{bmatrix} \quad (10.6)$$

The disturbance matrices for the wind and currents are given as:

$$\mathcal{G}_{\text{wind}} \cdot \mathbf{w} = \begin{bmatrix} 0.0060 \\ 0.0321 \\ 0.00000345 \end{bmatrix} \cdot \begin{bmatrix} {}^b\mathbf{w}_x^2 \\ {}^b\mathbf{w}_y^2 \\ {}^b\mathbf{w}_y^2 \end{bmatrix}, \quad \mathcal{G}_{\text{current}} \cdot \mathbf{v} = \begin{bmatrix} 2.5676 \\ 10.8108 \\ 0.0020 \end{bmatrix} \cdot \begin{bmatrix} {}^b\mathbf{v}_{\text{current}}_x^2 \\ {}^b\mathbf{v}_{\text{current}}_y^2 \\ {}^b\mathbf{v}_{\text{current}}_y^2 \end{bmatrix} \quad (10.7)$$

The output matrix \mathbf{C} is used to scale the outputs but as the outputs in this system should not be scaled, the output matrix therefore becomes equal to the identity matrix:

$$\mathbf{C} \cdot \mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (10.8)$$

There are several control strategies that can be used to control a multivariable system. As the theme for this semester is to analyse systems and develop controllers in different domains, the systems is looked at from different perspectives.

Summation

11

In the modeling chapter, the final functions for the motion was divided into two main equations. One giving the forward motion for the vessel and one giving the turning. To comply with the study regulations, this project must also use classical methods of regulation, so this chapter sums up the computed functions from earlier, as well as converting these to transfer functions in the frequency domain.

In general, a state space to laplace transformation can according to [Stoustrup, 2012] be given as:

$$\mathbf{G}(s) = \mathbf{C} \cdot (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \quad (11.1)$$

Where:

- $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the state space matrices, see equation 10.1 on page 43
- \mathbf{I} an identity matrix with same dimensions as \mathbf{A}
- $\mathbf{G}(s)$ the transfer functions in a matrix.
- s the laplace variable (representing frequency)

In equation 9.9 on page 41 and 9.10 on page 41 the state space equation where computed. Using the algorithm given in equation H.1 on page 117 the transfer functions for the two systems are given as:

$$\mathbf{G}_{\text{forward}}(s) = \begin{bmatrix} \frac{3.746}{s+7.391} & \frac{3.75}{s+7.391} \\ 0 & 0 \\ \frac{0.0784}{s+0.1568} & \frac{0.0784}{s+0.1568} \end{bmatrix} \quad (11.2)$$

$$\mathbf{G}_{\text{turn}}(s) = \begin{bmatrix} \frac{3.746}{s+7.393} & \frac{3.769}{s+7.393} \\ 0 & 0 \\ \frac{0.04784}{s+0.6837} & \frac{-4.688}{s+0.6837} \end{bmatrix} \quad (11.3)$$

As this model does not take into account the dynamics of the DC motor, each part involving it is multiplied with the transfer function of the DC motor, to generate the final transfer function for the system from input voltage to output forward acceleration the algorithm is shown in equation 11.4.

$$\mathbf{G}(s) = \begin{bmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \\ G_{3,1} & G_{3,2} \end{bmatrix} \cdot \begin{bmatrix} H_{\text{motor}}(s) \\ 1 \end{bmatrix} \quad (11.4)$$

As $G_{i,1}(s)$ represents the transfer function for the forward motion of the vessel – this is to be multiplied with the function found for the DC motor to compute the open loop response of the system. To recap the modelling, the

system has two sets of transfer functions, the forward motion transfer functions are given in 11.5 through 11.10:

$$G_{1,1} = \frac{1228}{0.0000576s^3 + 3.644s^2 + 60.27s + 246.4} \quad (11.5)$$

$$G_{1,2} = \frac{3.75}{s + 7.391} \quad (11.6)$$

$$G_{2,1} = 0 \quad (11.7)$$

$$G_{2,2} = 0 \quad (11.8)$$

$$G_{3,1} = \frac{25.71}{0.0000576s^3 + 3.644s^2 + 33.91s + 5.228} \quad (11.9)$$

$$G_{3,2} = \frac{0.0784}{s + 0.1568} \quad (11.10)$$

The turning motion transfer functions are given in equation 11.11 through 11.16:

$$G_{1,1} = \frac{1228}{0.0000576s^3 + 3.644s^2 + 60.28s + 246.4} \quad (11.11)$$

$$G_{1,2} = \frac{3.769}{s + 7.393} \quad (11.12)$$

$$G_{2,1} = 0 \quad (11.13)$$

$$G_{2,2} = 0 \quad (11.14)$$

$$G_{3,1} = \frac{15.69}{0.0000576s^3 + 3.644s^2 + 35.83s + 22.79} \quad (11.15)$$

$$G_{3,2} = \frac{-4.688}{s + 0.6837} \quad (11.16)$$

The state space \mathbf{A} for both forward and turning motion is, as described in state space representation.

$$\mathbf{A}_{\text{forward}} \cdot \mathbf{x} = \begin{bmatrix} -7.3909 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.1468 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (11.17)$$

$$\mathbf{A}_{\text{turning}} \cdot \mathbf{x} = \begin{bmatrix} -7.3925 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.6837 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (11.18)$$

For the state space matrices $\mathbf{B}_{\text{forward}}$ and $\mathbf{B}_{\text{turning}}$ the interior is:

$$\mathbf{B}_{\text{forward}} \cdot \mathbf{u} = \begin{bmatrix} 3.7460 & 3.75 \\ 0 & 0 \\ 0.0784 & -4.7616 \end{bmatrix} \cdot \begin{bmatrix} n \\ \theta \end{bmatrix} \quad (11.19)$$

$$\mathbf{B}_{\text{turning}} \cdot \mathbf{u} = \begin{bmatrix} 3.7461 & 3.769 \\ 0 & 0 \\ 0.0478 & -4.6881 \end{bmatrix} \cdot \begin{bmatrix} n \\ \theta \end{bmatrix} \quad (11.20)$$

The state space matrix \mathbf{C} is the same for both forward and turning motion.

$$\mathbf{C} \cdot \mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (11.21)$$

Part III

Controller design

This part describes the development of traditional and modern controllers. In the end, the results are compared, to choose the most suitable controller for this project that fulfills the requirements. The methods used are; modern state space and Multiple-Input Multiple-Output (MIMO) frequency control.

Multivariable frequency control **12**

A classical strategy to design controllers on a multivariate system, is to analyze all the inputs to the system, and see how they influence the outputs of the system individually. This can be done by using a “Relative Gain Array” which tells how much the individual inputs influences the output, which effectively tells what input/output pairs that can be considered as multiple Single-Input Single-Output (SISO) systems. This means that classic SISO frequency domain methods can be used.

12.1 General MIMO system

This section is upon [Tham, 1999]. Initially it is determined that the MIMO system to be considered is a feedforward system as depicted on figure 12.1.

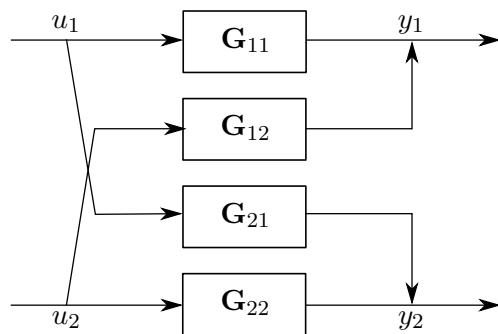


Figure 12.1: Illustration of a feedforward MIMO system

As a feedforward system both inputs u have a direct coupling on both outputs y . The output y can thereby be described as:

$$\mathbf{y} = \mathbf{G} \cdot \mathbf{u} \quad (12.1)$$

$$y_1 = u_1 \cdot \mathbf{G}_{11} + u_2 \cdot \mathbf{G}_{12} \quad (12.2)$$

$$y_2 = u_1 \cdot \mathbf{G}_{21} + u_2 \cdot \mathbf{G}_{22} \quad (12.3)$$

As a special case for 2x2 relative gain array matrices Λ can be given as:

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix} \quad (12.4)$$

Where the $\lambda_{i,j}$ elements represents the coupling between the inputs (columns) and outputs (rows) of the system. If the value is one it means there is full

coupling and therefore the output is independent of other inputs. If $\lambda_{i,j}$ is zero there is no coupling from input to output. Its desired to have a $\lambda_{i,j}$ as close to one as possible to minimize cross coupling. The matrix is computed as follows:

$$\Lambda = \mathbf{G} \circ ((\mathbf{G})^{-1})^T \quad (12.5)$$

The “ \circ ” means that the matrices have to be added element by element (The Schur product).

12.2 Forward control

The two transfer function matrices \mathbf{G}_f (forward model) and \mathbf{G}_t (turn model) as described in section 11 on page 45, have two inputs and three outputs when looking at the dimensions of the matrices. As the middle row in the function is not controllable, it can be removed, and the matrix becomes a 2x2 matrix. The relative gain array for the forward control will be:

$$\Lambda_f = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (12.6)$$

As the inputs are independent of each other, as shown be the relative gain array, it is not necessary to design a MIMO system for forward control. There will instead be made two SISO systems.

$$\mathbf{y} = \mathbf{G} \cdot \mathbf{K} \quad (12.7)$$

Through `rltool` the forward model gain matrix \mathbf{K}_f becomes :

$$\mathbf{K}_f = \begin{bmatrix} 1.6781 \cdot \frac{0.23 \cdot s + 1.6781}{s} & 0 \\ 0 & 50.32 \cdot \frac{0.19 \cdot s + 50.32}{s} \end{bmatrix} \quad (12.8)$$

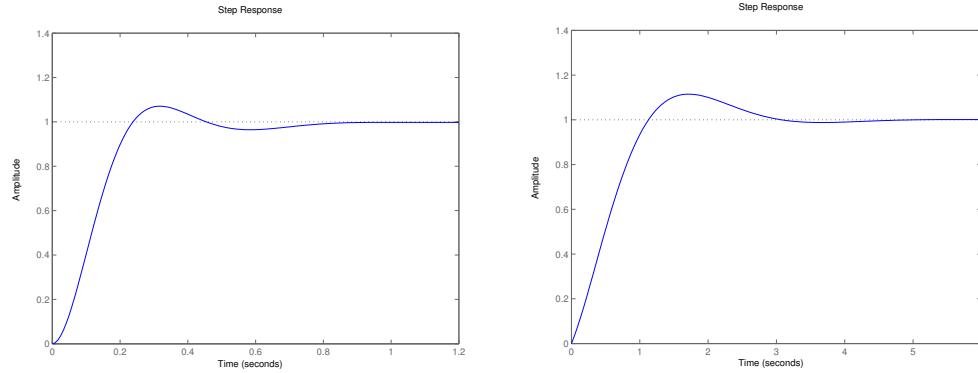
The graphs in figure 12.2 on the facing page show the outputs of the two SISO systems.

12.3 Turn control

Turning control is as the relative gain array indicates a MIMO system, as both inputs have an effect on both outputs.

$$\Lambda_t = \begin{bmatrix} 0.9898 & 0.0102 \\ 0.0102 & 0.9898 \end{bmatrix} \quad (12.9)$$

As the relative gain array for the turning motions transfer function is not an identity matrix, a MIMO system will be designed through decoupling \mathbf{G}_t .



(a) Rotations a second of the propeller.
The output y_0 .

(b) Angle of the rudder. The output y_1

Figure 12.2: The outputs for the input $n = 1$ and $\theta = 1$.

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \quad (12.10)$$

$$G_{11} = \frac{1228}{0.0000576s^3 + 3.644s^2 + 60.28s + 246.5} \quad (12.11)$$

$$G_{12} = \frac{3.769}{s + 7.393} \quad (12.12)$$

$$G_{21} = \frac{15.68}{0.0000576s^3 + 3.644s^2 + 35.83s + 22.8} \quad (12.13)$$

$$G_{22} = \frac{-4.688}{s + 0.6837} \quad (12.14)$$

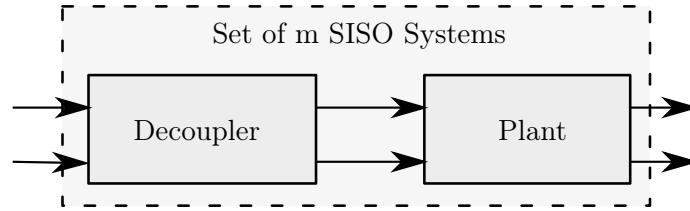


Figure 12.3: Decoupling figure of a MIMO system from [Albertos and Sala, 2004, p. 137].

To decouple the MIMO system a forward decoupling as seen on figure 12.3 will be performed. By the approach described in [Albertos and Sala, 2004, p. 137], the following formula will be used to decouple the system.

$$\mathbf{G} = \mathbf{MN} \quad (12.15)$$

$$M_{11} = \frac{1}{(0.0000576s^3 + 3.644s^2 + 60.28s + 246.5) \cdot (s + 7.393)} \quad (12.16)$$

$$M_{12} = 0 \quad (12.17)$$

$$M_{21} = 0 \quad (12.18)$$

$$M_{22} = \frac{1}{(0.0000576s^3 + 3.644s^2 + 35.83s + 22.8) \cdot (s + 0.6837)} \quad (12.19)$$

$$N_{11} = 1228 \cdot (s + 7.393) \quad (12.20)$$

$$N_{12} = 3.769 \cdot (0.0000576s^3 + 3.644s^2 + 60.28s + 246.5) \quad (12.21)$$

$$N_{21} = 15.68 \cdot (s + 0.6837) \quad (12.22)$$

$$N_{22} = -4.688 \cdot (0.0000576s^3 + 3.644s^2 + 35.83s + 22.8) \quad (12.23)$$

$$\mathbf{K} = \begin{bmatrix} K_{11} & 0 \\ 0 & K_{22} \end{bmatrix} \quad (12.24)$$

The obtained \mathbf{M} is the “transformed” plant of the SISO system and \mathbf{K} is the diagonal regulator. \mathbf{N} will be used in the form of the inverse \mathbf{N} .

$$\mathbf{y} = \frac{\mathbf{G} \cdot \mathbf{N}^{-1} \cdot \mathbf{K}}{1 + \mathbf{G} \cdot \mathbf{N}^{-1} \cdot \mathbf{K}} \quad (12.25)$$

Using MATLAB’s root locus function `rltool` on M_{11} and M_{22} separately, the gain matrix for the turn model \mathbf{K}_t was found through automatic robust tuning of a PI controller.

$$\mathbf{K}_t = \begin{bmatrix} 6509.3 \cdot \frac{0.29 \cdot s + 1}{s} & 0 \\ 0 & 9.2526 \cdot \frac{3.1 \cdot s + 1}{s} \end{bmatrix} \quad (12.26)$$

After having found all the matrices, the result for the turn model is the Simulink system in figure 12.4 on the facing page. The Simulink model can be found in `matlab/mimo/mimosystem.mdl`. The step response for the system can be seen in figure 12.5 on the next page.

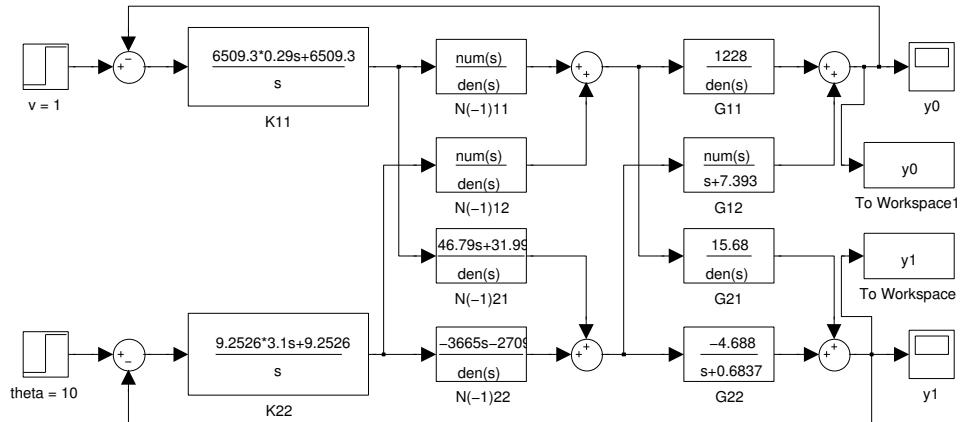


Figure 12.4: MIMO system model in Simulink. Showing the regulator \mathbf{K} with the multiplier \mathbf{N}^{-1} and transfer function \mathbf{G} .

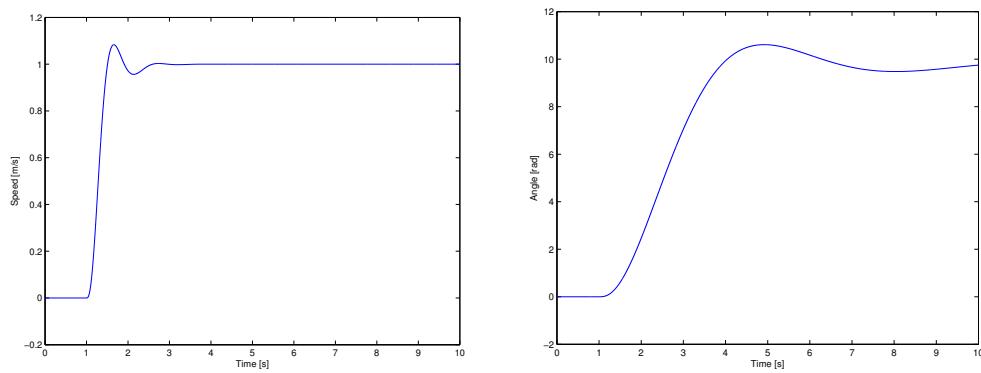


Figure 12.5: The outputs for the input $v = 1$ and $\theta = 10$ steps responses after 1 seconds delay.

13

Model based control

This chapter describes the design of controllers via a modern method. This includes utilizing the state space representation and the model of the vessel dynamics.

In the previous section 10 on page 43, the state space equations for the system were derived. As the system was linearized around two different sets of values – the state space controller design will also be developed around two different values. To recap, the matrices for the forward motion system was given as in equation 13.1.

$$\begin{aligned}\mathbf{A}_f &= \begin{bmatrix} -7.3909 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.1568 \end{bmatrix} \\ \mathbf{B}_f &= \begin{bmatrix} 3.7460 & 3.7500 \\ 0 & 0 \\ 0.0784 & -4.7616 \end{bmatrix}\end{aligned}\tag{13.1}$$

And for the turning system the equations were as in equation 13.2

$$\begin{aligned}\mathbf{A}_t &= \begin{bmatrix} -7.3925 & 0 & 0 \\ 0 & -2.4375 & 0 \\ 0 & 0 & -0.6837 \end{bmatrix} \\ \mathbf{B}_t &= \begin{bmatrix} 3.7461 & 3.7690 \\ 0 & 0 \\ 0.0478 & -4.6881 \end{bmatrix}\end{aligned}\tag{13.2}$$

In both of the cases the output matrix \mathbf{C} is given as a 3x3 identity matrix as none of the outputs needs scaling. As the two systems are dependent on the desired angle of the rudder (θ), if this approximately equals zero then the forward controller is used, but if the angle grow, then the system shifts to the turning controller. The basic state space system is depicted on figure 13.1 on the next page. As this only takes one input to the system, the reference angle is examined, and determines what system to use. Throughout this chapter two controller systems will be developed to control the individual subsystems. The θ block on figure 13.1 on the following page determines which system to use from:

$$\theta = \begin{cases} \text{forward} & \text{if } \theta \approx 0 \\ \text{turning} & \text{if } \theta \neq 0 \end{cases}\tag{13.3}$$

As the basis for developing the controllers are in place, the following will examine if the systems are controllable.

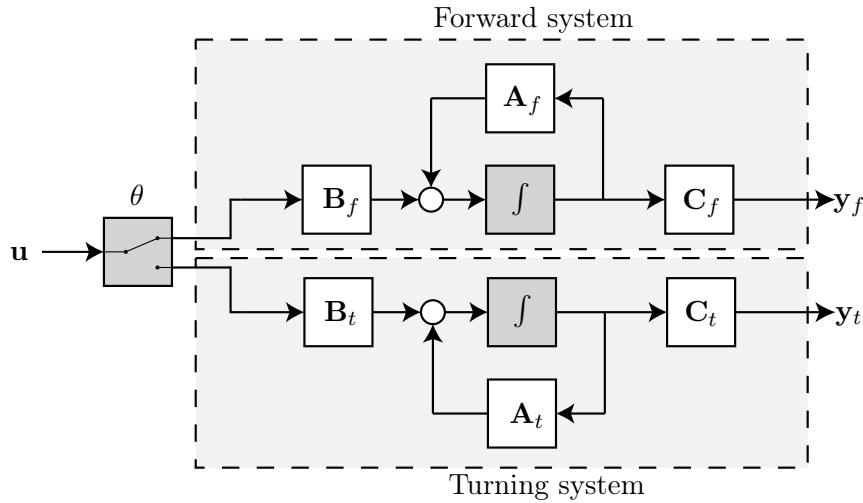


Figure 13.1: General state space controller

13.1 Controllability

To check if either of the systems can be controlled using state feedback, it is first examined whether the systems are controllable. A system is said to be controllable if there exists any input that can output a value different from zero, to any given time. The controllability matrix is defined by [Stoustrup, 2012, MM2 - Controllability] as:

$$\mathcal{C} = [\mathbf{B} \quad \mathbf{A} \cdot \mathbf{B} \quad \dots \quad \mathbf{A}^n - 1 \cdot \mathbf{B}] \quad (13.4)$$

For a system to be controllable, the rank of \mathcal{C} should equal the order of the system (in this case three). As this is not the case, (the rank equals two) the system is uncontrollable. If the state space equation is examined closer, the input making the system uncontrollable is the motion of the vessel in the y -direction \dot{y} . The system can however be made controllable by dividing the system into two parts. The controllable part becomes a system with the inputs given as the input to the engine and the rudder angle. This operation is done by removing the second row of all the matrices, and the rest of the system is then given as:

$$\hat{\mathbf{A}}_f = \begin{bmatrix} -7.3909 & 0 \\ 0 & -0.1568 \end{bmatrix}, \quad \hat{\mathbf{B}}_f = \begin{bmatrix} 3.7469 & 3.7690 \\ 0.0784 & -4.7616 \end{bmatrix} \quad (13.5)$$

$$\hat{\mathbf{A}}_t = \begin{bmatrix} -7.3925 & 0 \\ 0 & -0.6837 \end{bmatrix}, \quad \hat{\mathbf{B}}_t = \begin{bmatrix} 3.7561 & 3.7690 \\ 0.0478 & -4.6881 \end{bmatrix} \quad (13.6)$$

Which indeed is controllable, as the rank has been reduced by one, giving a second order system, and the controllability matrix also has rank two. The

output matrix \mathbf{C} is still given as an identity matrix, but has now have its dimension reduced to 2x2. As this system is controllable, a state feedback loop can be implemented as seen on figure 13.2. The feedback loop changes the input of the system to $\mathbf{u} = \mathbf{F} \cdot \mathbf{x}$ where \mathbf{F} is a gain matrix that scales the states. As seen on \mathbf{B} the system has two inputs, and the approach by calculating the T-transform of the system does not work. However, the feedback gain places the poles of the system at desired positions, and when in the state space domain, the poles of the feedback system can according to [Stoustrup, 2012, MM2 - State Feedback] be given as the eigenvalues of equation 13.7:

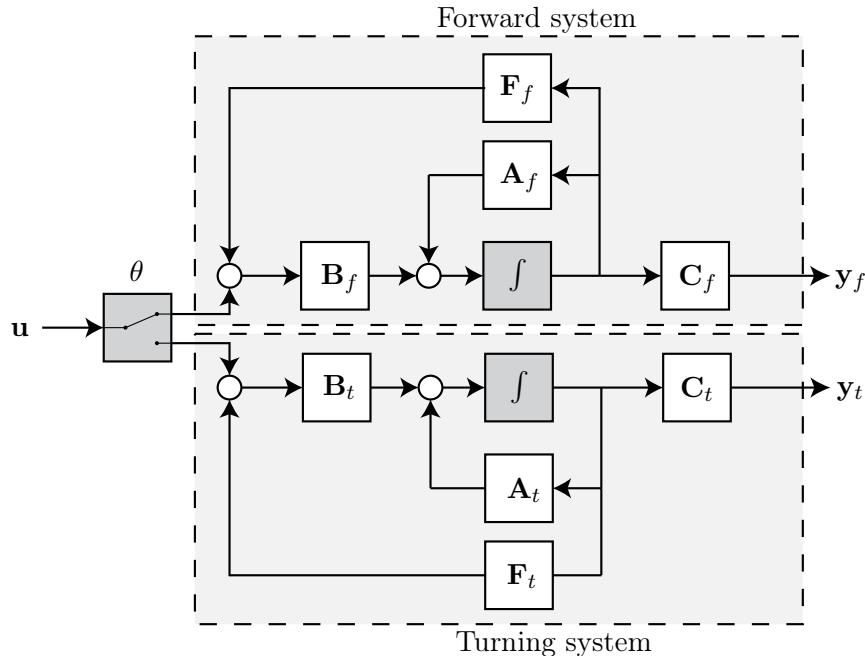


Figure 13.2: State space controller with state feedback loop inserted

$$[\mathbf{A} + \mathbf{BF}] \quad (13.7)$$

As it is desired to have poles that are faster (giving a faster response). The initial pole position for the two systems are calculated to be:

$$\begin{aligned} \det(\lambda\mathbf{I} - \mathbf{A})_f &= (\lambda + 7.3909)(\lambda + 0.1568) \\ \text{Initial forward poles in } &= \{-7.3909, -0.1568\} \end{aligned} \quad (13.8)$$

$$\begin{aligned} \det(\lambda\mathbf{I} - \mathbf{A})_t &= (\lambda + 7.3925)(\lambda + 0.6837) \\ \text{Initial turning poles in } &= \{-7.3925, -0.6837\} \end{aligned} \quad (13.9)$$

As the poles are to be faster, a choice could be to place the poles at $\{-10, -1\}$ for the forward system and $\{-10, -7\}$ for the turning system. The gain can

now be calculated using MATLAB's `-place` function. This solves equation 13.7 on the previous page for \mathbf{F} when given two poles in equation 13.8 on the preceding page or 13.9 on the previous page to give the following feedback gain matrices:

$$\mathbf{F}_f = \begin{bmatrix} 0.6965 & 0 \\ 0.0115 & -0.1771 \end{bmatrix} \quad (13.10)$$

$$\mathbf{F}_t = \begin{bmatrix} 0.6890 & 1.3418 \\ 0.0070 & -1.3336 \end{bmatrix} \quad (13.11)$$

The poles could also be calculated “by hand”, using the controllability matrix. The controllability matrix is given in equation 13.4 on page 58. And this can be used to compute the transformation matrix \mathbf{T} , that ensures \mathbf{A} and \mathbf{B} is on canonical form. As computation of MIMO system canonical matrices are out of the scope of this semester, the answers from MATLAB's `-place` will be used.

13.2 Observability

As calculated, the reduced system is controllable (having $\text{rank}(\mathcal{C}) = 2$) and it is now examined if this system is also observable. If the system is fully observable, a state feedback matrix is enough to control the system. The observability matrix is given as:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (13.12)$$

The observability matrix takes the \mathbf{A} and \mathbf{C} as inputs, and if this matrix has full rank the system is said to be observable. The observability matrix for both of the systems are again not square (due to the multiple inputs and multiple outputs) and is also out of the scope of this semester, but as all the states can be measured (the rotational velocity by a gyrometer and the velocity by the GPS signals), the system is considered to be observable.

13.3 Optimal feedback gain

Another strategy to determine pole placement of the system, is to minimize the systems cost function. The cost function for state space systems is by [Stoustrup, 2012, MM5 - Optimal Control] defined as:

$$\mathcal{J} = \int_0^{\infty} \rho \cdot y^T \cdot y + u^T \cdot u \, dt \quad (13.13)$$

Which can be re-written to an optimal control problem, by substituting the two matrices into the equation (**Q** and **R**). Equation 13.13 on the preceding page is a special case, when $\mathbf{Q} = \rho \mathbf{C}^T \mathbf{C}$ and $\mathbf{R} = 1$. The formulas are based upon lectures given by [Stoustrup, 2012].

$$\mathcal{J} = \int_0^\infty x^T \mathbf{Q} x + u^T \mathbf{R} u \, dt \quad (13.14)$$

The parameters **Q** and **R** can be estimated using Bryson and Ho's rule, [Franklin et al., 2009, Page 481] which state that the matrices can be estimated by:

$$\begin{aligned}\mathbf{Q}^{2 \times 2} &= \text{the constraints of the state vector } \mathbf{x} \\ \mathbf{R}^{2 \times 2} &= \text{the constraints of the input vector } \mathbf{u}\end{aligned}$$

The matrices **Q** and **R** are calculated by using the formulae 13.15 and 13.16.

$$Q_{ii} = \frac{1}{\text{maximum value of } \mathbf{x}_{ii}^2} \quad (13.15)$$

$$R_{ii} = \frac{1}{\text{maximum value of } \mathbf{u}_{ii}^2} \quad (13.16)$$

The physical constraints of the two systems are except the angular input, identical, and are given as:

$$\begin{aligned}Q_{1,1} &= \frac{1}{(\max\{\dot{x}(t)\})^2} = 1 \\ Q_{2,2,f} &= \frac{1}{(\max\{\dot{\varphi}(t)\})^2} \approx 0 \\ Q_{2,2,t} &= \frac{1}{(\max\{\dot{\varphi}(t)\})^2} = 3.642 \\ R_{1,1} &= \frac{1}{(\max\{u\})^2} = 0.0193 \\ R_{2,2,f} &= \frac{1}{(\max\{\theta\})^2} \approx 0 \\ R_{2,2,t} &= \frac{1}{(\max\{\theta\})^2} = 3.642 \quad (13.17)\end{aligned}$$

Using the inputs computed in equation 13.17 gives the following constraints matrices. And MATLAB's “-lqr” solver, gives two optimal pole placements. If the maximum rudder angle for the forward system should ≈ 0 , the system becomes undefined and can not be solved. Therefore the maximum angle for the forward system is defined as 1 degree, as this makes the equation solvable (and also generates the limits for the switch between the controllers). The solver calculates two feedback gains, one for the forward system given in

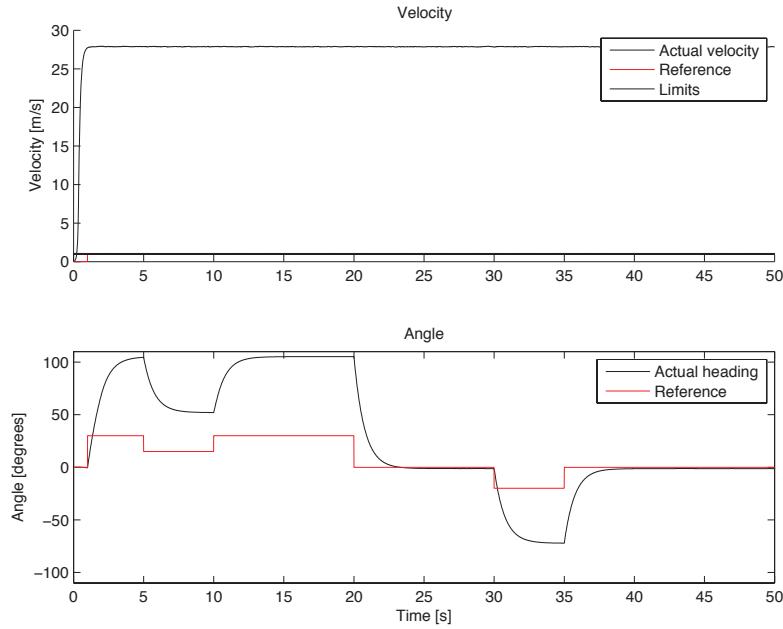


Figure 13.3: Simulation of the system with optimal state feedback gain, as seen the system is stable throughout the entire simulation, the system have added noise.

equation 13.18 and one for the turning given in 13.19.

$$\mathbf{F}_{\text{opt}}_f = \begin{bmatrix} 5.4890 & 0.3065 \\ 0.0127 & -0.9667 \end{bmatrix} \quad (13.18)$$

$$\mathbf{F}_{\text{opt}}_t = \begin{bmatrix} 5.4771 & 1.1600 \\ 0.0325 & -0.8616 \end{bmatrix} \quad (13.19)$$

The optimal pole placement strategy places poles in $\{13.2197, 4.4218\}$ and $\{13.2599, 3.3988\}$ for the forward and turning system respectively. As seen the optimal pole placement (for a minimized cost function) places poles far out in the right half plane of the system. This makes the system highly unstable and have a large error. As seen on figure 13.3 the system has a large deviation from the reference signal. Another simulation have been made with the state feedback gain calculated in equation 13.10 on page 60, these poles give the response depicted on figure 13.4 on the next page.

As the self assigned state feedback gives a better response, these poles are chosen as the ones to use in the rest of the system.

13.4 Adding the reference signal

As the vessel needs an input to decide whether it should sail with full speed or brake, it is desirable to introduce an error signal. As this error signal is desired

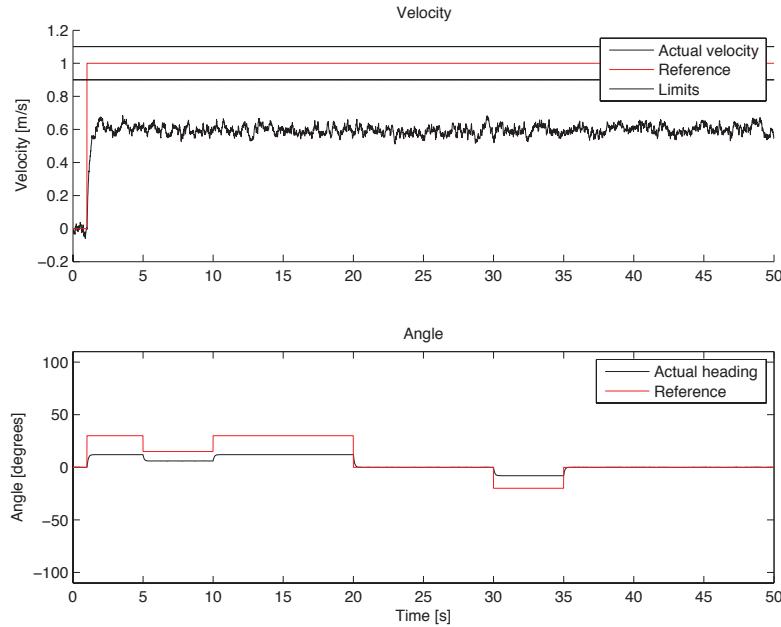


Figure 13.4: Simulation of the system with self assigned state feedback gain, as seen the system is stable throughout the entire simulation but there is a large steady state error, the system have added noise.

to be the error between the states and the desired reference. The obvious way to do this would be to implement the error signal as: $u = \mathbf{Fx} + \mathbf{r}$. But this will generate a steady state error. One way to fix this is to find the steady state for a variable, and make sure that when the input vector reaches steady state, then this error equals zero. This can according to [Franklin et al., 2009, page 470-471] rewrite the input equation to:

$$\mathbf{u} = \mathbf{u}_{ss} + \mathbf{F}(\mathbf{x} - \mathbf{x}_{ss}) \quad (13.20)$$

It is desirable to solve this system, so the system's steady state error on the system, can be given as a rewrite of the general state space equation and the following steady state state space equations for the controllable system can be given as the following:

$$\mathbf{0} = \mathbf{Ax}_{ss} + \mathbf{Bu}_{ss} \quad (13.21)$$

$$y_{ss} = \mathbf{Cx}_{ss} + \mathbf{0} \quad (13.22)$$

Now what is left to do, is to solve for values where the output of the system y_{ss} is equal to the reference signal \mathbf{r}_{ss} for all values of \mathbf{r}_{ss} . If the state vector is re-defined as $\mathbf{x}_{ss} = \mathbf{N}_x \mathbf{r}_{ss}$ and the input vector is re-defined to $\mathbf{u}_{ss} = \mathbf{N}_{ss} \mathbf{r}_{ss}$ then these can be introduced and the system reduces to a set of matrix equations

as given by [Franklin et al., 2009, page 470-471]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{N}_x \\ \mathbf{N}_u \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (13.23)$$

The last equation says that for all \mathbf{r}_{ss} there should be an \mathbf{N}_u that makes it equal to the output. The above is to ensure that for all inputs and states there exists an \mathbf{N}_x that gives a steady state error of 0. When the system is solved for $[\mathbf{N}_x \ \mathbf{N}_u]^T$ the two input gains become:

$$\begin{bmatrix} \mathbf{N}_{x,f} \\ \mathbf{N}_{u,f} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2.6695 & 0 \\ 0.0325 & -0.0329 \end{bmatrix} \quad (13.24)$$

$$\begin{bmatrix} \mathbf{N}_{x,t} \\ \mathbf{N}_{u,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2.6423 & 1.4870 \\ 0.0270 & -1.4780 \end{bmatrix} \quad (13.25)$$

And to compute the reference gain in equation 13.28 the matrices are split into two:

$$\mathbf{N}_{x,f} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{N}_{u,f} = \begin{bmatrix} 2.6695 & 0 \\ 0.0325 & -0.0329 \end{bmatrix} \quad (13.26)$$

$$\mathbf{N}_{x,t} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{N}_{u,t} = \begin{bmatrix} 1.9533 & 0.1452 \\ 0.0199 & -0.1443 \end{bmatrix} \quad (13.27)$$

Now, rewriting these equations in 13.20 on the previous page gives the following equations for the reference signal:

$$u = \mathbf{F}\mathbf{x} + (\mathbf{N}_u - \mathbf{F}\mathbf{N}_x)r \quad (13.28)$$

All left to do now, is to calculate the paranthesis, and the reference gain matrix can be determined as seen in equation 13.32.

$$\hat{\mathbf{N}}_{forward} = \mathbf{N}_{u,f} - \mathbf{F}\mathbf{N}_{x,f} \quad (13.29)$$

$$= \begin{bmatrix} 2.6695 & 0 \\ 0.0440 & -0.2100 \end{bmatrix} \quad (13.30)$$

$$\hat{\mathbf{N}}_{turning} = \mathbf{N}_{u,t} - \mathbf{F}\mathbf{N}_{x,t} \quad (13.31)$$

$$= \begin{bmatrix} 2.6423 & 1.4870 \\ 0.0270 & -1.4780 \end{bmatrix} \quad (13.32)$$

The simulation on figure 13.5 on the next page illustrates the system when the reference gain is inserted. As seen on the figure, the heading tracking is a lot more accurate, but when the vessel is meant to keep turning, the system generates an error. To remove this, an integrator can be inserted into the system.

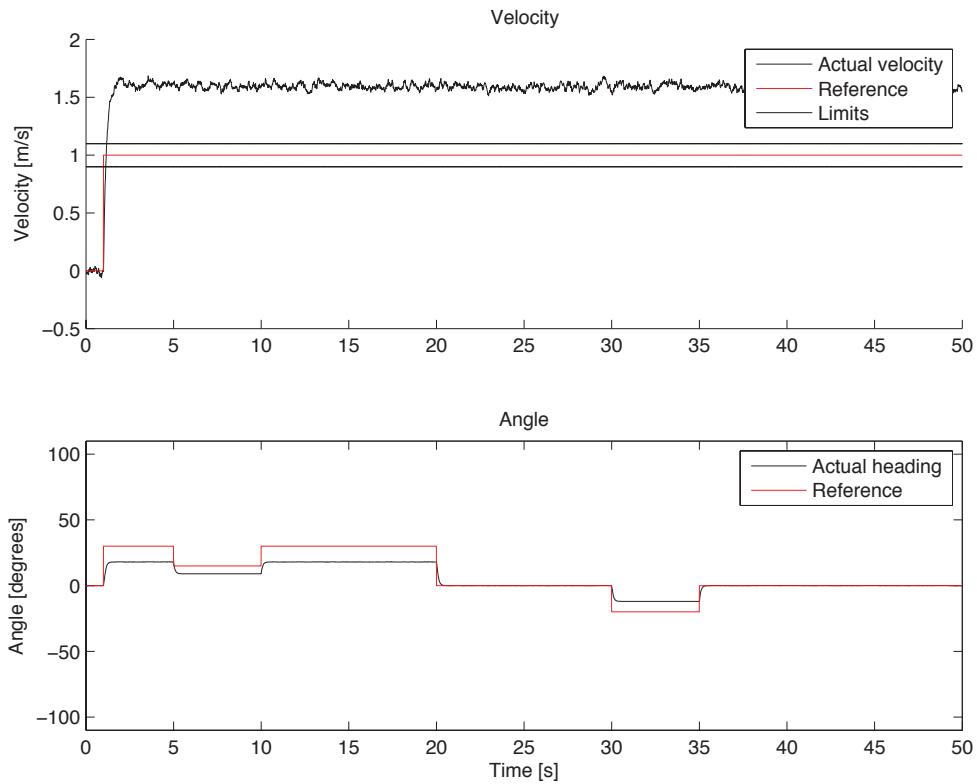


Figure 13.5: Simulation of the system with a state feedback loop and added reference gain.

13.5 Adding an integral part

To further improve the system and remove the steady state error an integral part is introduced as the purpose of this, is to add dynamics to the controller, and eliminate the steady state error, as the reference gain matrix \hat{N} is designed to eliminate the steady state error for the given system. The implementation can be seen on figure 13.7 on page 68. When this part is added an extra feedback loop is created, and the feedback becomes: $\mathbf{u} = \mathbf{Fx} + \mathbf{F}_I \mathbf{x}_I$, thus extending the state space equations to 13.35. [Stoustrup, 2012, MM4 - Integral Control]:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (13.33)$$

$$\dot{\mathbf{x}}_I = \mathbf{y} - \mathbf{r} \quad (13.34)$$

$$\mathbf{y} = \mathbf{Cx} \quad (13.35)$$

Which can be rewritten as a set of matrix equations:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_I \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_I \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{r} \quad (13.36)$$

$$\mathbf{u} = [\mathbf{C} \quad \mathbf{0}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_I \end{bmatrix} \quad (13.37)$$

Where the state feedback gain is changed to 13.38:

$$\mathbf{y} = [\mathbf{F} \quad \mathbf{F}_I] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_I \end{bmatrix} \quad (13.38)$$

As this extends the model to include an integral state as well, the state matrices used in the computation of the integral state feedback becomes:

$$\mathbf{A}_{\text{ext}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \quad (13.39)$$

$$\mathbf{B}_{\text{ext}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \quad (13.40)$$

As the system now is extended, the `-place` function is used again, and through simulation, the system shows the best response with poles placed at $\{-4, -7, -4, -7\}$ and at $\{-3, -2, -5, -2\}$ for the forward and turning system respectively. When the `place` function is used the integral state feedback matrices for the forward system becomes equation 13.41:

$$\mathbf{F}_{I,f} = \underbrace{\begin{bmatrix} F & 0 \\ -0.9635 & 0 \\ -0.0159 & 2.2772 \end{bmatrix}}_{\text{and}} \quad \underbrace{\begin{bmatrix} F_I & 0 \\ -7.4746 & 0 \\ -0.1231 & 5.8804 \end{bmatrix}}_{\text{and}} \quad (13.41)$$

And for the turning motion of the state feedback system the integral state feedback matrices becomes as given in equation 13.42:

$$\mathbf{F}_{I,t} = \underbrace{\begin{bmatrix} F & 0 \\ 0.1037 & -0.9169 \\ 0.0011 & 0.9114 \end{bmatrix}}_{\text{and}} \quad \underbrace{\begin{bmatrix} F_I & 0 \\ -2.6432 & -1.2746 \\ -0.0270 & 1.2668 \end{bmatrix}}_{\text{and}} \quad (13.42)$$

As seen in 13.41 and 13.42 the systems consists of two feedback gains, one to be implemented in the feedback loop designed previously and one in the integral part. As this changes the state feedback gain of the system, the \mathbf{N} reference gain matrices change as well. When using equations 13.28 on page 64 the reference gains become:

$$\hat{\mathbf{N}}_f = \begin{bmatrix} 0.9877 & 0 \\ 0.0268 & -2.3101 \end{bmatrix} \quad (13.43)$$

$$\hat{\mathbf{N}}_t = \begin{bmatrix} 1.8496 & 1.0621 \\ 0.0189 & -1.0557 \end{bmatrix} \quad (13.44)$$

The final system have been simulated and gives the response depicted on figure 13.6 As seen on figure 13.6 the system behaves nicely and thus a state space controller has been derived for the system.

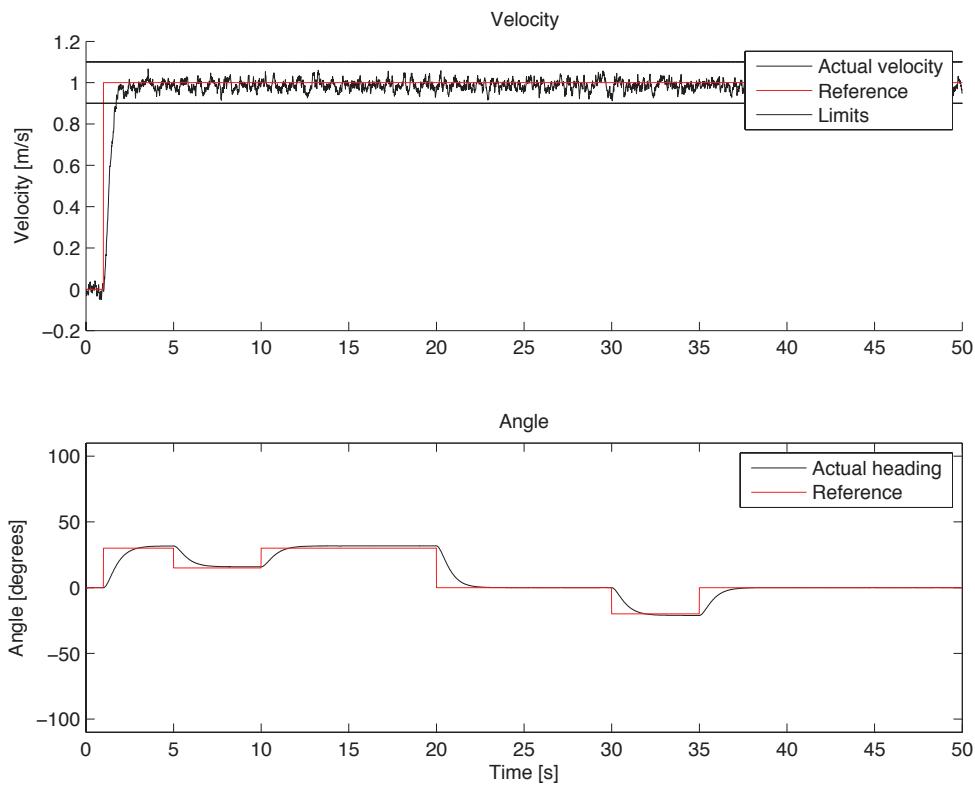


Figure 13.6: Simulation of the system with a state feedback loop, added reference gain and integral feedback gain.

13.6 Summation

The final system is depicted on figure 13.7 on the following page. To recap, a state feedback controller using two different models have been designed and simulated to give the response depicted on figure 13.6. The next step is to develop the controller for implementation in the system.

The calculations carried out in this chapter can be found in the MATLAB file found in [matlab/model/ss_model.m](#) and the simulation used to generate the figure, is found in: [matlab/model/ss_observ.mdl](#).

13.7 Discretization of the state space equations

To develop a controller that can be implemented on the onboard computer, the system is first discretized. The general state space equations are computed in the time domain, and the discrete equivalents are not necessarily equal. The system is converted using zero order hold, meaning that the system only has one sample per time interval (meaning that the sampling time and the time

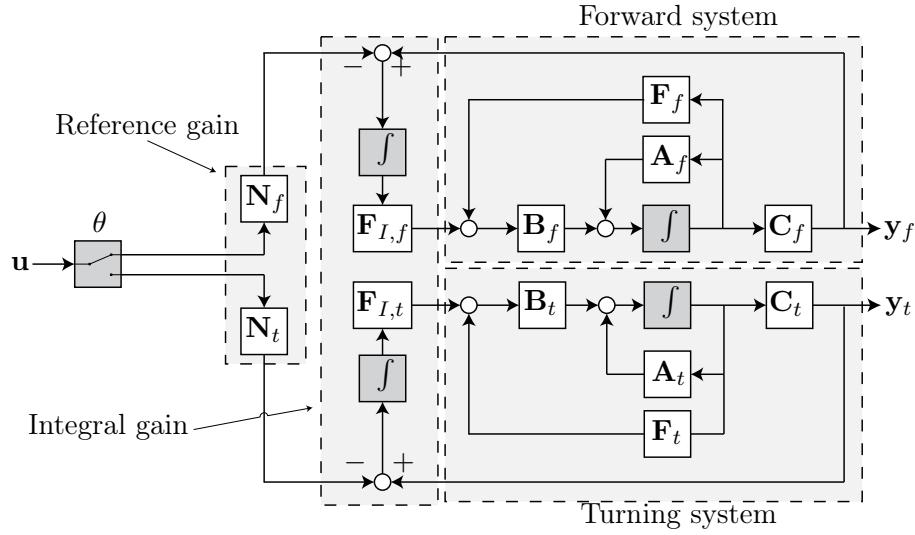


Figure 13.7: State space controller

interval should be the same). The discrete equivalent to the state space system is given as in equation 13.45, [Stoustrup, 2012, MM1 - Discrete State Space], (the subscript \mathcal{D} denotes that the signal is discretized):

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}_{\mathcal{D}}\mathbf{x}[k] + \mathbf{B}_{\mathcal{D}}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}_{\mathcal{D}}\mathbf{x}[k] \end{aligned} \quad (13.45)$$

Where:

- $\mathbf{A}_{\mathcal{D}}$ = The discrete equivalent of \mathbf{A}
- $\mathbf{B}_{\mathcal{D}}$ = The discrete equivalent of \mathbf{B}
- $\mathbf{C}_{\mathcal{D}}$ = The discrete equivalent of \mathbf{C}

To calculate the discrete equivalents the following formulae from [Haugen, 2005, Page 40] which converts the matrices to their discrete equivalences. The sampling frequency is 10 Hz, so the sampling time is given as $T_s = f^{-1} = 0.1$. Writing the equations give:

$$\mathbf{A}_{\mathcal{D}} = e^{\mathbf{A} \cdot T_s} \quad (13.46)$$

$$\mathbf{B}_{\mathcal{D}} = \left(\int_{\tau=0}^{T_s} e^{\mathbf{A}\tau} d\tau \right) \mathbf{B} \quad (13.47)$$

As the computations of the discrete equivalences and conversions from continuous to discrete time has not been covered by lectures, the MATLAB command `c2d` is used to discretize the system. When used on the forward system, the

discrete matrices becomes equation 13.48:

$$\begin{aligned}\mathbf{A}_{\mathcal{D},f} &= \begin{bmatrix} 0.4776 & 0 \\ 0 & 0.9844 \end{bmatrix} \\ \mathbf{B}_{\mathcal{D},f} &= \begin{bmatrix} 0.2648 & 0.2651 \\ 0.007779 & -0.4724 \end{bmatrix}\end{aligned}\quad (13.48)$$

And the discrete state and input matrices for the turning system becomes equation 13.49:

$$\begin{aligned}\mathbf{A}_{\mathcal{D},t} &= \begin{bmatrix} 0.4775 & 0 \\ 0 & 0.9339 \end{bmatrix} \\ \mathbf{B}_{\mathcal{D},t} &= \begin{bmatrix} 0.2648 & 0.2664 \\ 0.004624 & -0.4531 \end{bmatrix}\end{aligned}\quad (13.49)$$

As the different gains computed in the above is not the same for the discrete system, all of these are to be recalculated using the same formulae (but with different pole placements). When this is done, the state feedback gain matrices for the discrete forward and turning system is given in equation 13.50 and 13.51.

$$\mathbf{F}_{\mathcal{D},f} = \begin{bmatrix} -0.4142 & 0 \\ -0.0068 & -3.6699 \end{bmatrix}\quad (13.50)$$

$$\mathbf{F}_{\mathcal{D},t} = \begin{bmatrix} -0.6012 & -0.4244 \\ -0.0061 & 0.4218 \end{bmatrix}\quad (13.51)$$

When two poles are inserted into the system it gives the plot given on figure 13.8 on the next page, as seen the system has a large steady state error for both the forward - and turning system. To try and remove this, the system is also simulated using the reference gain matrix. This is calculated the same way as for the continuous case, and gives the reference gains in equation 13.52 and 13.53 for the forward and turning system respectively.

$$\hat{\mathbf{N}}_{\mathcal{D},f} = \begin{bmatrix} -1.3893 & 0 \\ -0.0229 & 0.7787 \end{bmatrix}\quad (13.52)$$

$$\hat{\mathbf{N}}_{\mathcal{D},t} = \begin{bmatrix} -1.1837 & -1.6281 \\ -0.0121 & 1.6182 \end{bmatrix}\quad (13.53)$$

When the system is simulated with the reference gain matrices, the plots becomes as depicted in figure 13.9 on the following page. As seen on this figure the system converges towards the reference signals both in the case of the velocity model and the turning model. The calculations for the discrete system can be found in MATLAB file on [matlab/statefeedback/discrete.m](#) and the Simulink model on [matlab/statefeedback/ss_discrete.mdl](#)

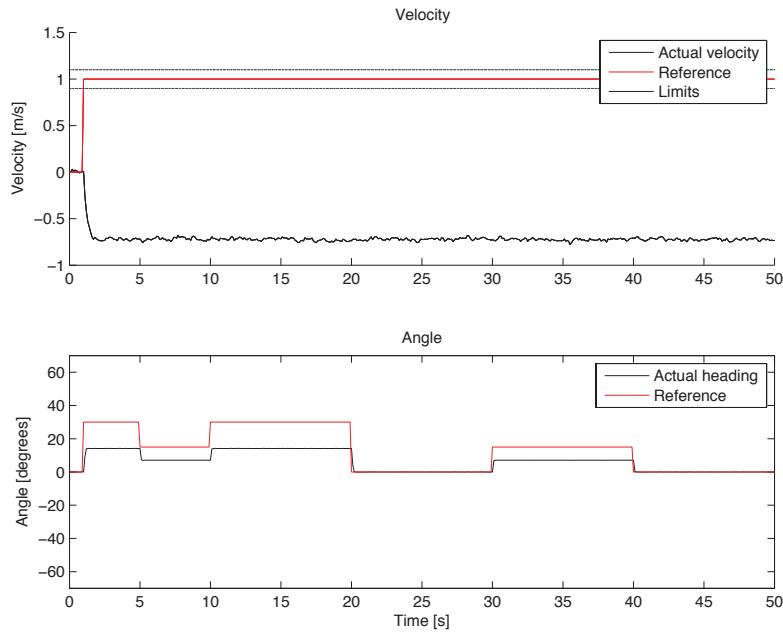


Figure 13.8: Simulated version of the discrete time system with the state feedback blocks $\mathbf{F}_{\mathcal{D},f}$ and $\mathbf{F}_{\mathcal{D},t}$ inserted.

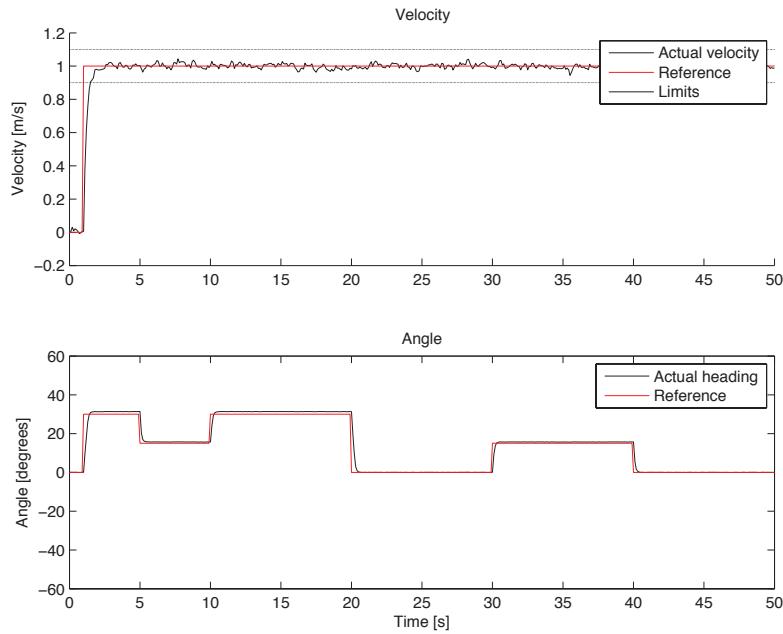


Figure 13.9: Simulated version of the discrete time system with the state feedback blocks $\mathbf{F}_{\mathcal{D},f}$ and $\mathbf{F}_{\mathcal{D},t}$ inserted as well as the reference gain matrices $\hat{\mathbf{N}}_{\mathcal{D},f}$ and $\hat{\mathbf{N}}_{\mathcal{D},t}$.

14

Software

The implementation is primarily done using software, this chapter describes the different functions and methods used to develop these algorithms.

The programming language used in this project is C for the targeted platform IGEP. The target platform uses a Linux operating system. Throughout this chapter a system overview is presented before the individual blocks presented in the software structure.

As there are used several programs operating simultaneously, it is necessary to have a real time system that can manage this. The system developed through this project is implemented on a Linux operating system, that supports multi threading and real time operations. The implemented system is a sequential system that runs in parallel with several other programs such that the underlaying operating system, Linux, handles all real time issues.

14.1 Software structure

This section describes how the software is designed in general, the following sections will describe the functions in more detail. In order to make the software more tangible it has been divided into subgroups as shown in figure 14.1.

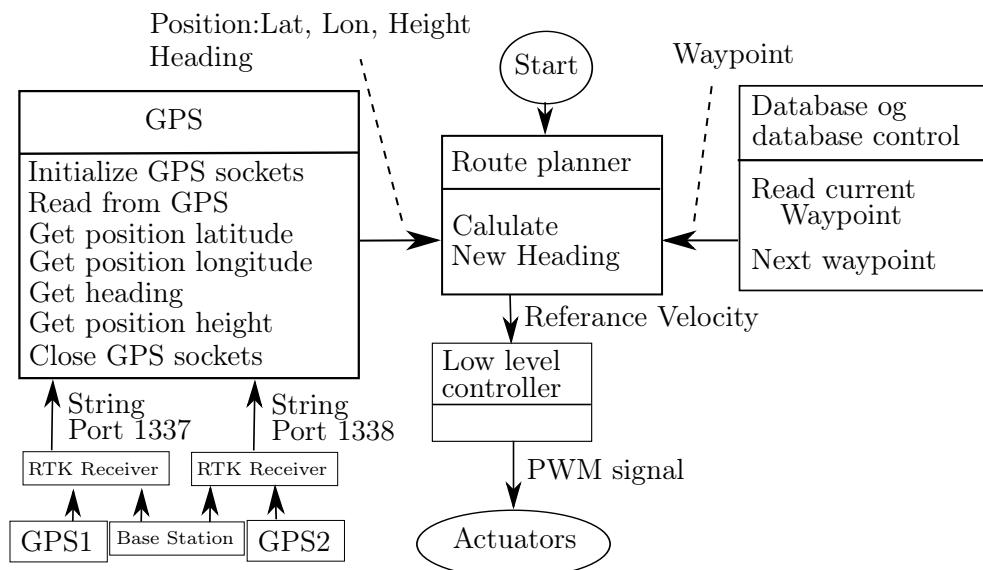


Figure 14.1: System overview. The functions depicted are the functions to be called from another section of the program and the arrows is the return value of these calls, where the parentheses indicate the type returned.

The software is based around the “route planner” (section 14.4 on page 75)

where all inputs are gathered, computed and new low level references for the actuators are chosen.

14.2 Interfaces

As the software consists of multiple parts it is necessary to develop interfaces between these. The interfaces being described is the client, server, writing to serial connection, writing to and reading from a .csv file and writing to .txt files to serve as log files.

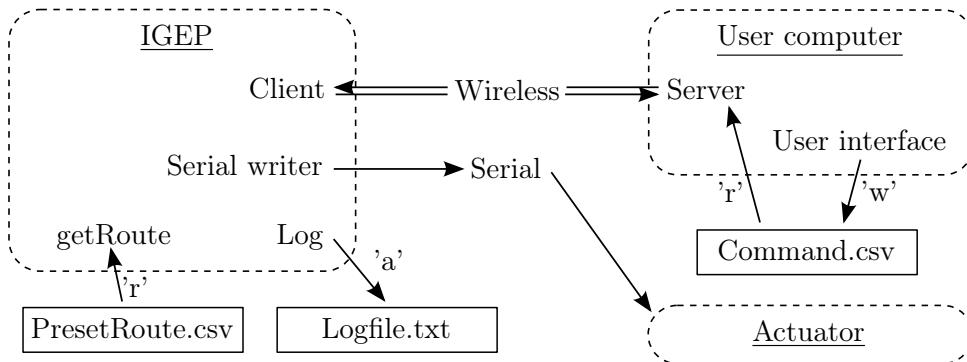


Figure 14.2: Schematic of the software interfaces.

Client

To be able to communicate with the vessel it has been chosen to establish a server-client connection, between the vessel and a computer. The application “client” is run on the vessel, because it is not possible to open an ssh connection to the vessel from a computer, since the mobile network provider does not allow incoming ssh connections on their network devices. The server will be running on a computer available to the user. The cause of having a server-client connection between the vessel and the user, is to be able to send commands to the vessel if necessary.

Function	Arguments	Return type	Unit
clientinit()	void	void	None
getCommand()	void	char	Character
closeClient()	void	void	None

Figure 14.3: Function calls in `client.c`.

clientinit(): Function for initializing a TCP socket connection on port 7734 to a server on the designated IP address.

getCommand(): When this function is called the character “A” is sent from the client to the server, indicating that the client is requesting a command from the server. The client then waits for a reply from the server. As this can cause a deadlock the client will only wait for 100 milliseconds, using the function **select()**.

closeClient(): Closes the socket connection to the client.

Server

The server is run on a user computer. On this computer two programs must be running for the user to communicate with the vessel. One of which is the Commander which writes user commands into a .csv file. The other program is the Server Socket, which will whenever it receives the character “A” from the client, read from the command .csv file containing the commands and send these via the opened socket connection to the client.

Write to files

A write to file function is needed on both the server and the IGEП. The one on the server is part of the user interface which writes to the .csv file, from which the server receives its commands. This writer uses the extension “w” when the function **open(FILE, ext)** is used. This extension causes the writer to replace whatever is written within the file, with the command character put into the **write()** function. The other writer have the same structure as the first, only it writes to .txt files which is used as a log file, its extension is “a”, which causes the writer to append the new data to the existing in the document.

Read from files

The function for reading from a csv file, uses **open(FILE, ext)** to open the file. The readers extension is “r” indicating that it reads from the file. **fgets(buffer, sizeof buffer, FILE)** fills its buffer with a line from the .csv file. Every time the command is called it reads a new line into the buffer. When reading from a .csv file every column is separated with a comma, this makes it simple to differentiate the values and characters from each other.

Serial writer

The only way for the IGEП to communicate with the actuators is via a serial connection, therefore it is necessary to implement a function for handling serial output. To open the serial connection, **open(FILE, ext)** is used, the **FILE** parameter specifies the file name and format to be opened. The extension is **O_RDWR | O_NOCTTY | O_NDELAY** which is respectively read-write mode, not controlling the terminal, which means that keyboard input will not be written

in the serial terminal and `O_NDELAY` means that the program will not care how the received data is interpreted, and in what spaces receiver will place the input. In the `write(open, buffer, sizeof buffer)` it is necessary to indicate the opened connection, the buffer which is a character array, and the length of the buffer.

14.3 GPS

The following gives a detailed description of the functions used to communicate with the GPS, where table G.1 on page 114 shows the argument needed to call the respective functions and the expected return value. This data is generated by the Real Time Kinematic (RTK) positioning software (RTKlib) that runs on the vessel, to receive data from the two GPS receivers and calculate corrections to achieve sub decimeter accuracy.

Function	Arguments	Return type	Unit
<code>sockinit()</code>	<code>void</code>	<code>void</code>	None
<code>closesock()</code>	<code>void</code>	<code>void</code>	None
<code>sock()</code>	<code>void</code>	<code>void</code>	None
<code>getPosLat()</code>	<code>void</code>	<code>Double</code>	Radians
<code>getPosLon()</code>	<code>void</code>	<code>Double</code>	Radians
<code>getPosHeight()</code>	<code>void</code>	<code>Double</code>	Radians
<code>llhheading()</code>	<code>void</code>	<code>Double</code>	Radians

Figure 14.4: Detailed description of the GPS-functions.

sockinit(): Initialization and opening of the Transmission Control Protocol (TCP) sockets at port 1337 and port 1338 and setting the server to localhost. The code is a modified version of a socket programming example found at [LinuxHowTos, 2012]. Once Initialized the sockets will remain open until actively closed.

closesock(): Close both TCP sockets.

sock(): Read a single String from the TCP stream from both TCP sockets. The string is separated into the latitude, longitude, height and quality. These are converted into doubles. The following is a random string extracted from the TCP sockets and interpreted as described in [Takasu, 2012, p. 72–75].

1688 113169 57.014596415 9.985928604 66.2552 5

The symbol “” indicates a whitespace. In the string from RTKlib there always needs to be at least two whitespaces between each parameter, this is the reason for there being three to four whitespaces in front of latitude and longitude as this value can be negative and have two decimals in front of the decimal separator. For height there are room for this value to have four decimals. Only latitude, longitude, height and quality are used for this project, therefore the rest of the string from RTKlib is not needed. Latitude and longitude are readouts in degrees and height is in metres. Quality is a number between 1-5 that indicates the solution quality, with 1 being a fixed solution with an accurate position and 5 being a single point positioning solution with a less accurate position.

getPosLat(): This function returns the latitude in radians. This function takes the latitude from both GPS receivers and simply use $\frac{\text{lat1}+\text{lat2}}{2}$ to find the central latitude position of the vessel.

getPosLon(): This function returns the longitude in radians. Similar to getPosLat() $\frac{\text{lon1}+\text{lon2}}{2}$ is used to find the central longitude position of the vessel.

getPosHeight: This function returns the height in meters and is returned as a double.

llhheading(): This function returns the heading in radians by first taking the individual positions received by both GPS-receivers and using azimuth calculations as shown in equation 14.1 to compute the heading. See [Mathworks, 2012] for further details.

$$\begin{aligned}\text{heading} &= \text{atan2}(x, y) \\ x &= (\cos(\text{lat2}) \cdot \sin(\text{lon2}-\text{lon1})) \\ y &= (\cos(\text{lat1}) \cdot \sin(\text{lat2}) - \sin(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\text{lon2}-\text{lon1}))\end{aligned}\tag{14.1}$$

This function uses true north as a reference meaning that at 0° the vessel is pointing towards true north and at 90° it is directly east and so forth.

14.4 Conversion of data to surface frame

The desired route is described in the WGS84 coordinate system and therefore it needs to be converted into the surface frame for the vessel to calculate heading and velocity.

Conversion WGS84 to ECEF

In this section the input and output functions of the data are described as well as how the conversions from one coordinate system to another is carried out.

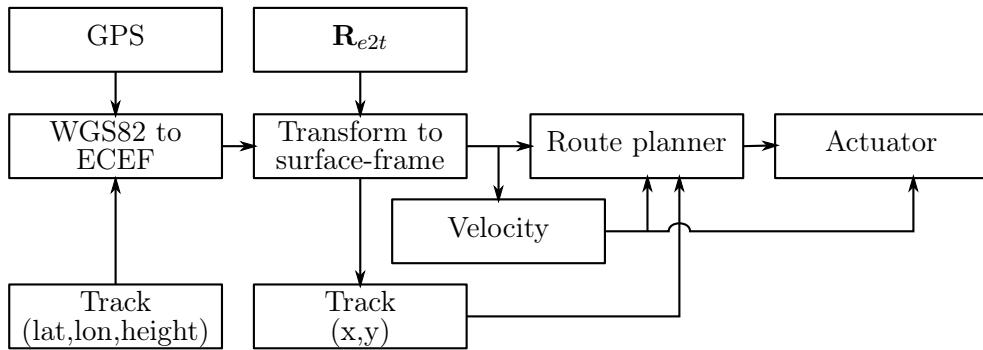


Figure 14.5: Detailed system overview, depicting how the different functions interact.

The figure 14.5 depicts the different functions and how they are interacting, which is used in the development of the software. Both current position obtained by GPS and the current list of waypoints are in WGS84 coordinates. See [Farrel and Barth, 1998, p. 27] for further details.

All coordinate are then converted to the ECEF system using the formulas described in section 5.2 on page 23.

Mapping to tangent plane

With both the desired location and the current position of the vessel, the rotational matrix is applied in order to transform the rotation to a tangent plane, called the surface frame. Similar the entire route is converted to the surface frame and stored for later reference. This is done by applying the rotational matrices described in section 5 on page 21.

Velocity

In the surface frame the velocity is calculated based on the change in position in this frame and used to control the input to the propeller controller. This is done by multiplying the distance between the last waypoint and the current by a time constant equal to the sampling frequency.

With the GPS coordinates and the desired route converted from WGS84 to ECEF it is possible for the route planer to calculate a heading and a position in the surface frame.

14.5 Route planner

In this section the algorithm for the route planner is described. See section 4.6 on page 15 for a simulation of the following implementation. This algorithm works on points along the route. Which are used as a reference for the controller. The route planner is developed as described in section 2.2 on page 6

and was at first simulated in MATLAB to make a base for the further development in C. The MATLAB code is developed to take an initial position and a list of way points in a fictive surface frame. The figure 14.6 shows the different functions of the implemented route planner.

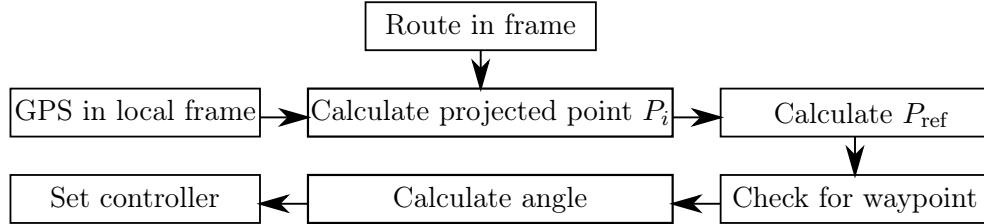


Figure 14.6: Detailed overview of the route planner, showing how the different functions interact.

As it can be seen on the flow and functionality of the routeplanner it is very similar to the flow described in section 4.6 on page 15, with a few changes. At first the intermediate point P_i parallel to the route is calculated. For this the intermediate vector v_i is found and then normalized. Using a calculated intermediate length of 0.2 m and then the current position can be obtained from the GPS and P_i can be found.

This point is then projected onto the planned route using the vector to vector projection $\mathbf{r}_d = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|^2} \cdot \mathbf{b}$ such that the vector R_d is calculated as seen in equation 14.2.

$$\mathbf{r}_d = \frac{\mathbf{r}_l \cdot \mathbf{r}_i}{|\mathbf{r}_i|^2} \mathbf{r}_i \quad (14.2)$$

By adding the former P_i and P_i the speed can be calculated.

It is checked if the vessel is within the radius of the current waypoint, and if the vessel is within a specified waypoint radius the software changes heading to the next waypoint.

14.6 Controller

The software for the controller is built as a two part system composed of a system on the IGEP board and a system on the actuator board. The interface between the two can be seen on figure 14.7 on the next page.

IGEP

The primary function of the IGEP board is to continuously calculate and maintain a constant course and velocity based on the reference velocity and direction received from the route planner.

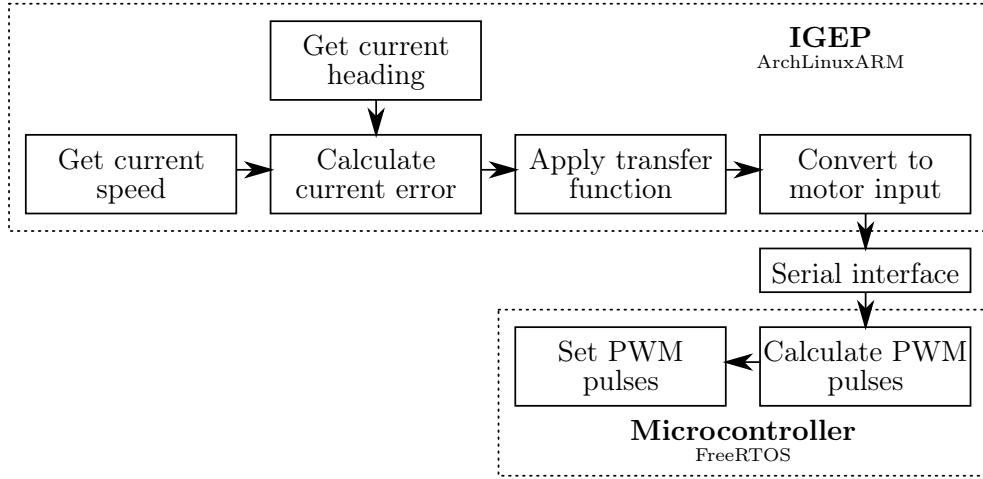


Figure 14.7: Software flow of the velocity and turning controller and the interface between IGEP and the actuator board.

The difference between the reference and the actual velocity and rudder angle is found by subtracting the current velocity by the reference (or the current angle from the reference angle) this is then used as the input for the controller. The reference is applied to the z -transformed transfer function. See appendix H on page 117 for details on the z -transformation.

Actuator board

The primary function of the actuator board is to create a Pulse-Width Modulation (PWM) pulse that drives the propulsion and rudder motor based on the input from the IGEP board. The precision of the rudder and propeller is determined by the precision of the PWM pulse and is limited by the hardware used. The software on the actuator board is reused software from a former project and implemented as described in [Dam et al., 2011].

In this project the PWM is limited by the microcontroller such that the precision for the rudder is 14 steps in total. Therefore the $\pm 20^\circ$ are scaled down to conform to these limitations and the PWM pulse is created accordingly. The PWM values and its corresponding rudder angles are listed in table 14.1.

Direction	Angle	PWM Value	Duty cycle
Left	-30°	930/1023	9.1 %
Centre	0°	937/1023	8.4 %
Right	30°	944/1023	7.7 %

Table 14.1: Data for the rudders physical servo positions

Similarly the PWM pulse corresponding to a velocity of the propeller is found as shown in table 14.2. It can be seen that the precision is better as the propeller motor is a DC-motor and therefore the PWM pulse has direct effect on the propeller speed. It has been decided that 58 steps between $\pm 100\%$ velocity is a sufficient resolution.

Direction	Speed	PWM Value	Duty cycle
Forward	100 %	907/1023	11.3 %
Stop	0 %	936/1023	8.5 %
Backward	-100 %	965/1023	5.6 %

Table 14.2: Data for the propeller speed

Part IV

Closing

This part concludes the report with an acceptance test, conclusion and discussion of further development.

15

Acceptance test

Throughout the project, one problem in particular has halted the process of getting the product finished, this being the lacking reliability of the RTK-GPS precision, this have hindered an “on location” test, instead the tests from section 3.2 on page 11 will be simulated, instead of running on a real lake.

15.1 Acceptance testing simulations

The tests have been conducted using MATLAB.

Test 1: The vessel must navigate along a straight line:

The vessel is programmed to navigate along a 20 metre straight route, during the movement along the route the vessel will be subjected to wind at varying angles at speeds up to 5 m/s, throughout the test the vessel must keep a velocity in accordance with the requirement specification. The test will be conducted five times, to make sure sufficient wind angles and speeds have been covered.

Test simulation

Shown on figure 15.1 on the following page are two simulations showing the controllers designed using respectively the classical pole-zero placement and the state space approach. The simulations show how the controllers handle varying reference values for heading and velocity. This reflects how the vessel would react in a real world scenario, only the reference signals have been simplified to make the graphs easier to understand.

Both controllers work, though the state space controller gives the best result, and would therefore be the one used on the vessel.

Test result

As it is seen on figure 15.1 on the next page the simulated outputs of the controllers clearly converges to the reference signals, both regarding speed and heading, thus the test is considered passed.

Test 2: The vessel must navigate using an absolute positioning system:

Two reference points are set up, where the absolute position is known. The vessel is placed at the first point, while logging it's position, and then at the second point while logging it's positions.

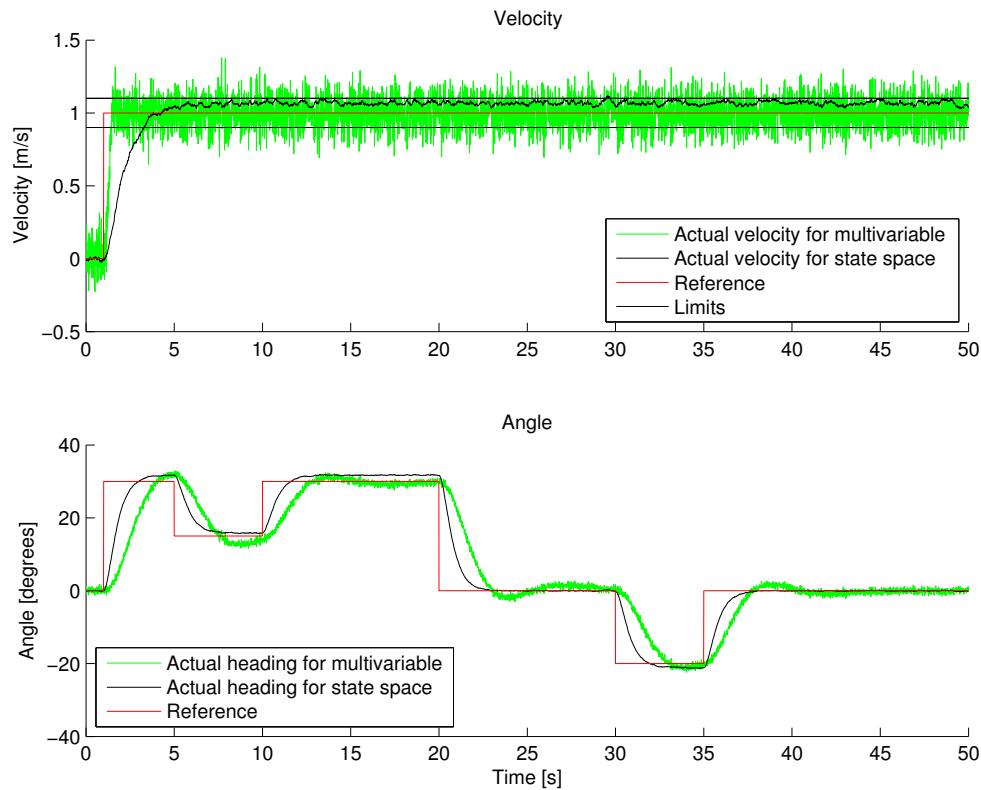


Figure 15.1: Simulations showing speed - and angle controller working. Designed by pole-zero placement - versus state space method. Both controllers sees 0.01 % noise on the feedback signal, relative to the maximum value of the reference signal.

Test simulation

Since the RTK GPS system has not worked as reliably as expected, an actual acceptance test has not been conducted with the GPS receivers on the vessel, however the RTK GPS system has unambiguously been shown to be able to produce reliable results in appendix B on page 93.

Test result

The test is not passed since the RTK GPS system was not successfully tested on the vessel, even though good results have previously been shown.

Test 3: The vessel must report it's position to ground station:

The vessel is programmed to continuously send out data, and the ground station is set to receive mode. The vessel is moved away from the ground station to a distance of 1000 m.

Test

The communication have been implemented through a 3G modem, that have been successfully tested on several occasions.

Test result

The test is considered passed, since the 3G modem makes the vessel able to communicate with the ground station whenever it is in an area with 3G coverage, regardless of the distance to the ground station.

15.2 Result overview

The following table is the result of the acceptance test described in previous subsections.

Test #	Summary	Result
Test 1	Navigation	Success
Test 2	Positioning	Failure
Test 3	Communication	Success

Table 15.1: Acceptance test overview

Conclusion and future perspective

16

This chapter will summarize the report by discussing what was achieved and what still needs further development, as well as discussing future applications for the product and technologies developed in this report.

16.1 Conclusion

This report documents the initial development of an autonomous vessel for depth mapping in confined waters, from the preliminary interview with Nellemann & Bjørnkjær and analysis of the parameters affecting the design to the final design. The focus of the project has been designing controllers to make the vessel autonomous, therefore there is not implemented any sensory equipment other than the ones needed for autonomous control. The design has been based on a remote controlled platform that has been retrofitted with an onboard computer, two GPS receivers, 3G internet modem and the necessary hardware to connect these. The two GPS receivers and the 3G modem are used for accurate position and heading acquisition. Unfortunately it proved difficult to obtain a satisfactory accuracy due to conditions out of the projects main scope. There has successfully been designed and simulated both a state-space and multivariate frequency controller using the root locus approach. The simulations have been used as basis for the acceptance testing, that have then been mostly successful, see tabel 15.1 on page 85.

16.2 Further development

To make the vessel functional, the positioning system needs work, to gain the wanted accuracy and an echo sounder needs to be implemented.

For further development of the vessel a number of refinements are interesting to look at:

Remote control is a feature that would be useful for surveyors, making them able to measure small confined waters without preparing waypoints, and making measuring of dynamic or inaccessible areas easier.

A deployable base station with a GPS receiver for making correction signals for the vessel, would be a great supplement to the vessel, giving the surveyor high RTK GPS precession anywhere in a matter of hours, this could be combined with an alternative data link between the base station and the vessel, making the vessel functional when no 3G coverage is available.

Obstacle avoidance system to ultimately make the vessel able to operate in dynamic environments, with no supervision.

Further autonomization with more sensorfeedback, e.g. from an IMU, ultimately making the vessel able to set and change it's own waypoints.

It is intended to continue work on the autonomous vessel system on later semesters, however not necessarily in the context of surveying applications, and using an other hardware platform, but utilizing the systems, ideas and experience gained throughout this project.

Part V

Appendix

The appendix includes chapters which are important for the project, but not necessarily interesting to the reader of the report.

Meeting with Nellemann & Bjørnkjær

A

This chapter summarize the important facts, wishes and requirements that the buisnesspartner, Peter Eistrup from Nelleman & Bjørnkjær had regarding an autonomous mapping vessel.

As this project is conducted in cooperation with Nellemann & Bjørnkjær, a meeting to understand the needs of the company for a depth mapping vessel portable by car, is summarized in this text as a basis for the requirement specification for this project. This appendix describes their needs and what was discussed during the meeting.

At the meeting Peter said that measuring water is currently a very time consuming task, that usually involves two or more people, and can take up to four days. When the depth of an area is measured, a single beam echo sounder is used. To obtain the precise position for the depth measurement, either a GPS or a laser/prism system is used. The GPS system they use is utilising RTK compensation here by making them able to obtain a accuracy in the order of sub decimetre. An other system they use is a laser/prism system, which is used in conjunction with dataum points.

To simplify this, an autonomous vessel could be developed, where a route can be plotted on a map at the office, and the vessel is then brought to the field, where it navigates along this route whilst collecting depth data. This reduces the work load on the surveyors, and makes it easier to measure these waters, and can be used to survey waters that previously was too expensive for the client to get mapped.

The summarisation specifications have been listed in the following table based on what was discussed at the meeting.

- Depth measurement: About one foot length precision and down to 50 m depth
- Deviation from route: Dependent on the job at hand, but about one foot length or a little bit more, if the positioning system is accurate enough.
- Route planning: To be done from home using waypoints on a map
- Surveying equipment: Single beam echo sounder
- Data transmission: Real-time to indicate if everything is okay.
- Vessel speed: Very slow (0.5 m/s)
- Frequency of measurements: 1 sample/m
- Length of measurements: Up to 1 km.
- Operating area size: Up to max one hectare and be done in one work day
- Single man operation and transportable by vehicle
- Should be able to operate in good weather

These specifications have been used as a basis for the initial development of the vessel — and are used as a reference to the requirement specification in chapter 3 on page 11.

B

RTK validation

This documents the accuracy of the RTK system in respect to the baseline between the base station and rover unit. Two tests have been conducted. One test with the rover unit close to the base station and one far away.

B.1 Purpose

To validate the feasibility of the RTK system, it is desirable to see how accurate the positioning system is at different locations away from the base station. Therefore some measurements has been conducted to obtain data for validation.

B.2 Measurements

The two tests have been conducted, one with a baseline of approximately 16 m (short baseline) and one with a baseline of 7 km (long baseline). The baseline is the distance between the rover and base station in a straight line. In this setup the RTK software system consists of a program library called RTKlib, which also contains some GUI windows to interact and present data to the user. The hardware is two GPS receivers from u-blox, model LEA-4T.

One was set up on the rooftop of a building connected to a computer that worked as a server (base station) to send data to the other receiver that was connected to a laptop (rover). The base station served the data stream from the university lab network which could then be tunneled out to the rover via the Internet on a wireless LAN connection at the rover end. The topography is illustrated on the figure B.1. All the measurements are conducted when the rover was in a fixed position, so it is possible to see the deviation over time. The positions have a sample frequency of 5 Hz and have been sampled for about one hour.

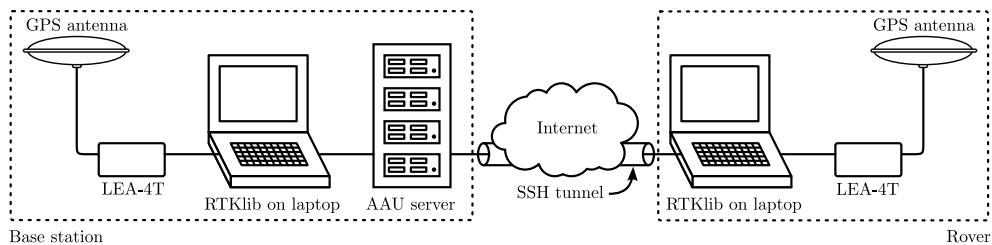


Figure B.1: Topography of RTK setup

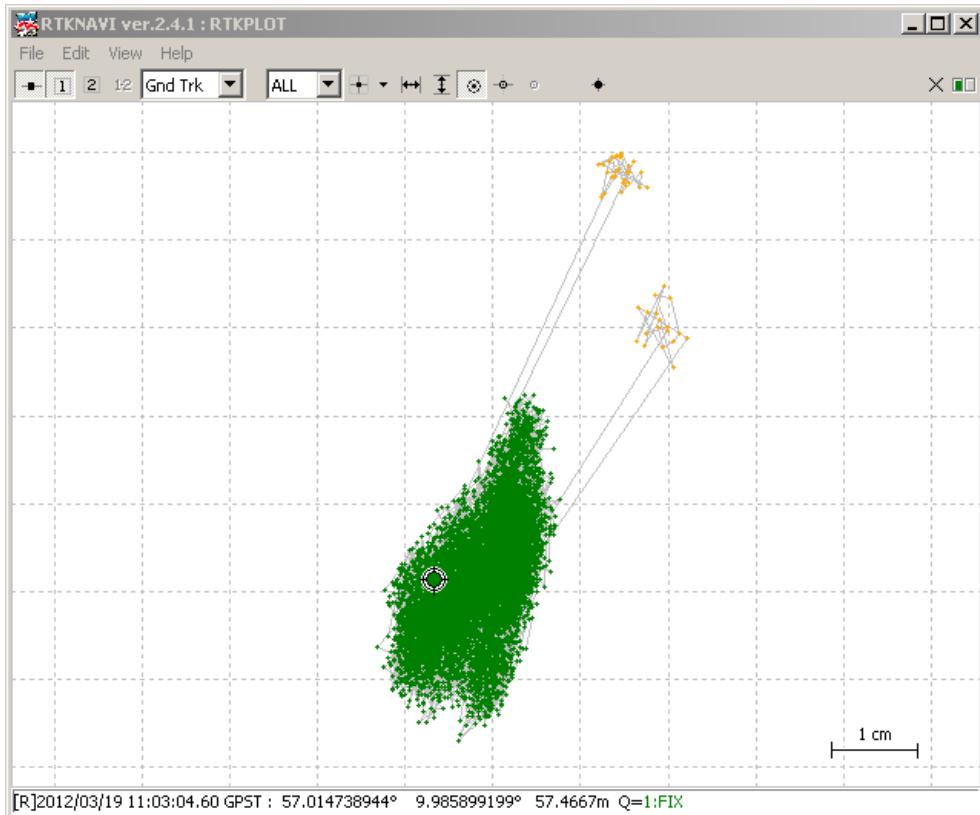


Figure B.2: Short baseline (16 m): This is a plot of how the data appeared most of the time, hence this is how the positioning accuracy is under a fixed solution, therefore it has a general accuracy of about 3 cm.

B.3 Conclusion

On figure B.2 it is seen that when the system has computed a fixed solution (green colours) it has an deviation of about 4 cm of samples (short baseline). But on the long baseline the snapshot was a little better as it is seen on figure B.3 on the next page, the deviation was about 3 cm of samples (long baseline).

When figure B.4 on page 96 is inspected it is seen that the positions have drifted a bit. This may have been caused by a bad satellite constellation at that point in time. It should be noted that the rover was placed in a garden about seven metre away from the end wall of a building which is 4 metre tall. This could contribute with reflections and shadow which may have caused the drift. The drift of the fixed solution is up to about 20 cm for this long baseline case.

The test also shows that the precision does not decrease considerably with the long baseline of about 7 km. The RTK system is considered accurate within

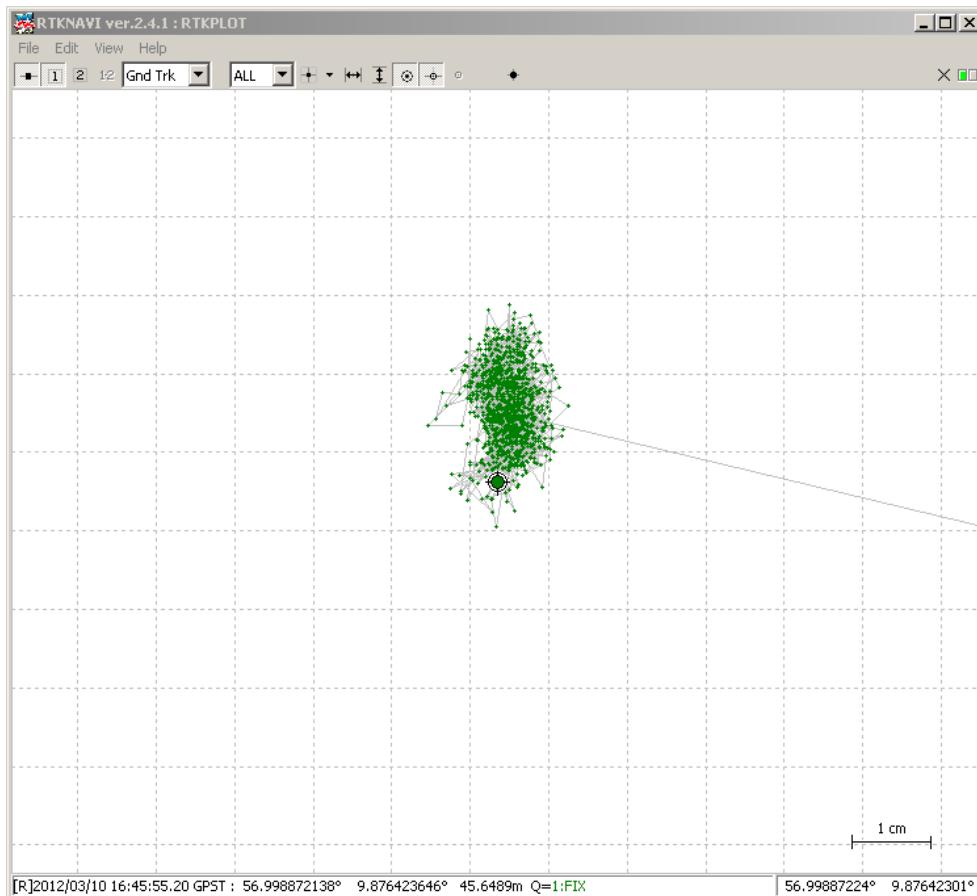


Figure B.3: Long baseline (7 km): This is a plot of how the data appeared most of the time, hence this is how the positioning accuracy is under a fixed solution, therefore it has a general accuracy of about 3 cm.

this range. However relatively large deviations can occur in some cases of bad satellite constellations, reflections and other sources of local noise, which is to be considered in the design of the navigation system.

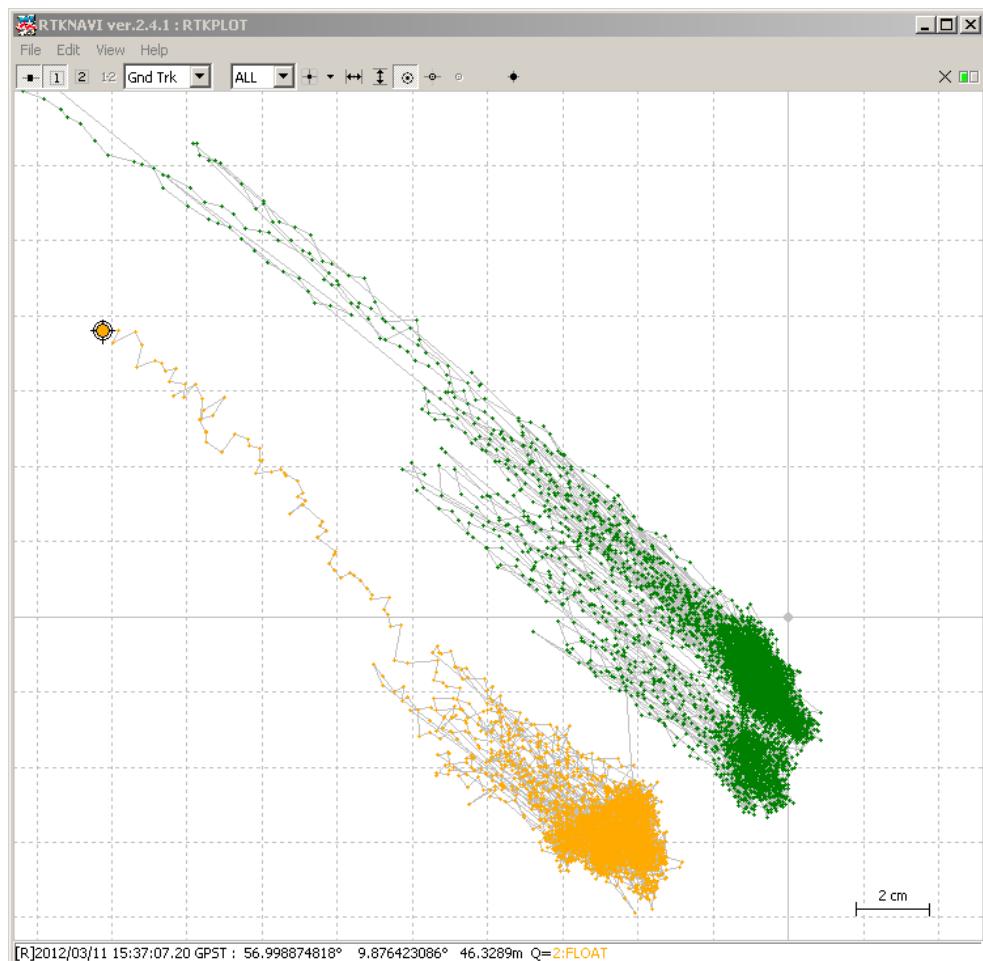


Figure B.4: Long baseline (7 km): This is a plot of the data suddenly drifting for a minor period of time during the hour long sampling time.

Measurements on test vessel

C

This journal documents the measurements concerning the propulsion system of the test vessel, and is comprised of three individual experiments, each focusing on different aspects of the propulsion system.

C.1 Purpose

The purpose of this measure journal is to determine the speed of the motor when running at full power with the propeller submerged, the forward force generated by the propeller, and the power consumed by the motor when running submerged. The thrust and speed is used to determine the propeller coefficient in section 6.1 on page 25 equation 6.2 on page 25.

C.2 Tools

Type	AAU no.	Manufacturer	Model
Hand digital tachometer	08246	SHIMPO	DT-205
Newton meter	—	—	10 N spring type
Multimeter	60760	Fluke	189
Multimeter	60764	Fluke	189
Oscilloscope	33941	Agilent	54621D

Table C.1: Tools used for this measurement.

C.3 Measurement setup and procedure

The measurements were conducted in three separate setups. One for measuring the thrust, one for measuring the rotation of the propeller, and one for measuring the power consumption of the propeller.

Thrust setup

The vessel was lowered into a pond and a string was attached to the rear end, the other end of the string was attached to the newton meter.

Thrust measurement procedure

The measurement was conducted by setting the vessel to full throttle, keeping the newton meter steady and reading of the value of the newton meter, see figure C.4 on the next page.



Figure C.1: Setup of the thrust measurement test

Motor rotation setup

The propeller of the vessel was submerged in water and the vessel was held at a fixed position. The axle transferring power from the motor to the propeller was marked with non reflecting tape on one half.

Motor rotation procedure

The the motor was set to full throttle and the tachometer was pointed at the marked point of the axle, measurements were recorded with a camera. These values were noted for each second and a mean was calculated.



Figure C.2: Setup of the rotation measurement test

Power consumption setup

In this test the propeller was submerged into a bin filled with water the motor was connected to a power supply through an ammeter. The setup of the test can be seen in figure C.3.

Power measurement procedure

The power supply was set to 7.4 V, while it was noted how many ampere the motor was drew.

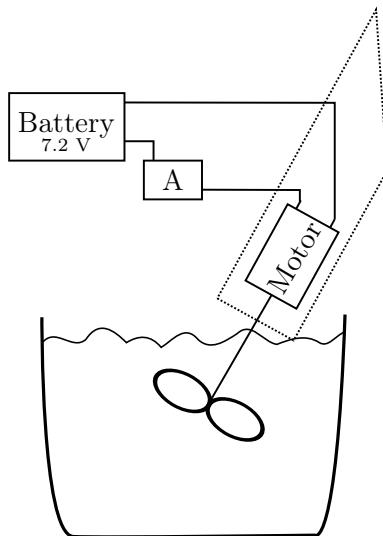


Figure C.3: Setup of the power consumption measurement



Figure C.4:
Thrust force measurement with spring newtonmeter

C.4 Measurements

The thrust of the vessel was measured to 4.85 N, as seen on figure C.4. The rotational speed data collected through one minute of the motor running submerged at full speed was put into MATLAB and the mean was calculated, giving the result: $3325 \text{ rpm} \approx 55.41 \text{ rps}$.

The current was measured during a separate measurement, and was measured to 15 ampere, when the motor was running at full speed with the propeller submerged.

C.5 Results

When the vessel is fixed in the water it can deliver 4.85 N of thrust, at a rotational speed of the motor of 55.41 Hz while drawing 15 ampere. Propeller diameter is 5 cm.

Measurements of propulsion motor

D

This documents the measurements concerning the propulsion motor of the test vessel.

D.1 Purpose

The purpose of this measure journal is to determine the motor parameters of the propulsion motor, to be able to make a better model of the motor.

D.2 Tools

Type	AAU no.	Manufacturer	Model
Tachometer probe	77087	Compact Instruments	A2108
Power supply	60770	HAMEG	HM7042-3
Multimeter	77057	Fluke	289
Multimeter	77056	Fluke	289
Multimeter	64540	Fluke	189
Oscilloscope	52773	Agilent	54621D
Foto camera	—	—	—

Table D.1: Tools used for this measurement.

D.3 Measurement setup and procedure

Two measurements were done on the motor, to determine different characteristics of the motor.

Measurement one setup

The measurement setup was a power supply connected through a multimeter, measuring ampere to the motor. With an other multimeter measuring the voltage on the power supply. The tachometer probe was connected to the last multimeter and held at close range to the shaft of the motor that had been marked with tape, in order to enable the tachometer probe to be able to read the rotations of the motor. The setup can be viewed on fig D.1 on the following page.

The three multimeters were placed next to each other, so that photos could be taken to ensure all measurements were simultaneous.

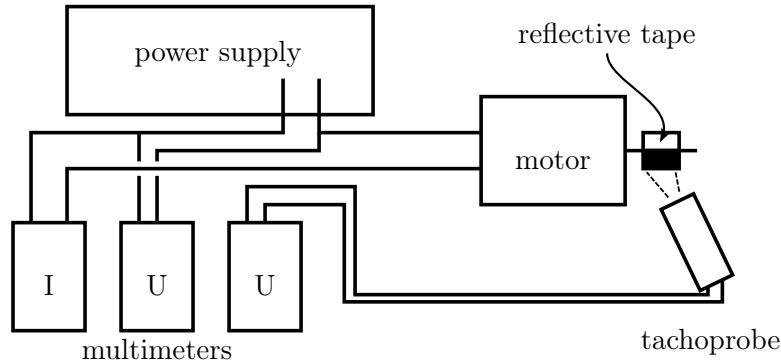


Figure D.1: Measurement setup showing connections in the setup of measurement one.

Measurement one procedure

The voltage on the power supply was set to 2 V and increased 0.5 V at a time, each time taking a picture of the three multimeters until reaching 7.5 volts.



Figure D.2: Measurement setup with three digital multimeters to measure the current, voltage and speed.

Measurement two setup

The motor connected to a resistor in series, an oscilloscope measures the voltage over the resistor. The setup can be seen in figure D.3 on the next page.

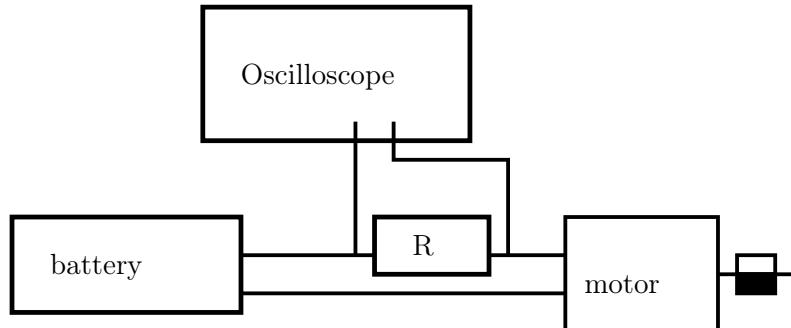


Figure D.3: Measurement setup showing connections in the setup of measurement two.

Measurement two procedure

A 7.2 volt battery is connected to the circuit and voltage over the resistor is logged until the motor has reached maximum speed.

D.4 Measurements and calculations

The measured data can be viewed at the dots on figure D.4.

It is seen that the data approximates two straight lines. In MATLAB the function `polyfit` is used to approximate a straight line in each case. The `polyfit` function takes two arrays, one with x-coordinates and one with y-coordinates, and an integer, describing the order of polynomial that is wanted, in this case 1, since a first order polynomial is wanted. From the straight lines generated, two points are taken out of each graph, one at 4 V and one at 7 V.

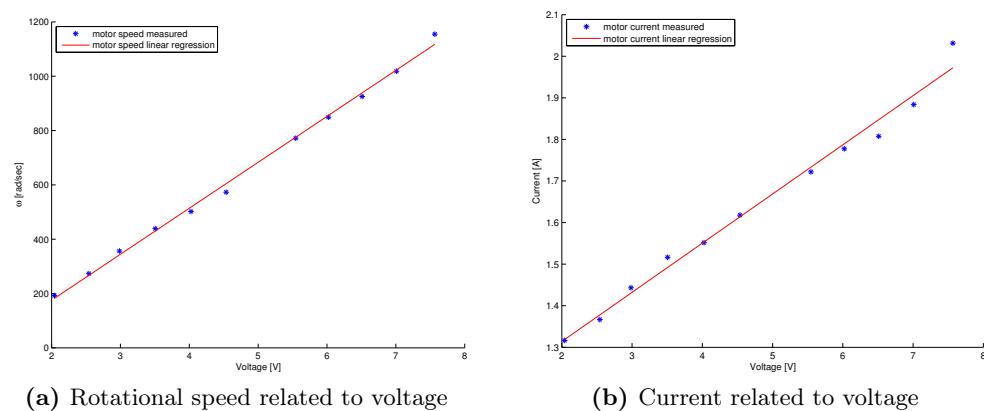


Figure D.4: Data and approximated lines describing the data

The data obtained can be inserted to the following matrix: [Pedersen and Andersen, 2011]

$$\begin{bmatrix} \omega_1 \cdot K + i_{a1} \cdot R \\ i_{a1} \cdot K - \tau_c - \omega_1 \cdot b \\ \omega_2 \cdot K + i_{a2} \cdot R \\ i_{a2} \cdot K - \tau_c - \omega_2 \cdot b \end{bmatrix} = \begin{bmatrix} u_{a1} \\ \tau_{b1} \\ u_{a2} \\ \tau_{b2} \end{bmatrix} \quad (\text{D.1})$$

ω_1 = rotational speed measured at 4 V = 513.7148 $\frac{\text{rad}}{\text{s}}$

ω_2 = rotational speed measured at 7 V = 1021.4 $\frac{\text{rad}}{\text{s}}$

i_{a1} = current at measurement at 4 V = 1.5504 A

i_{a2} = current at measurement at 7 V = 1.9051 A

u_{a1} = voltage at 4 V

u_{a2} = voltage at 7 V

τ_{b1} = load at measurement at 4 V = 0

τ_{b2} = load at measurement at 7 V = 0

K = motor constant

τ_c = friction torque

R = resistance in the windings of the motor

b = friction of the motor

The unknown K, τ_c , R and b can be calculated from the matrix shown above, resulting in:

$$K = 0.005343688804$$

$$\tau_c = 0.006366937622$$

$$R = 0.8093833687$$

$$b = 0.000003733428547$$

To measure the inertia of the motor, the following equation is utilized:

$$J = \frac{\tau_m \cdot (R \cdot b + K^2)}{R} \quad (\text{D.2})$$

J = inertia of motor

τ_m = mechanical time constant of motor

The mechanical time constant of the motor can be measured by looking at the results from measurement two, the data from this measurement is plotted as seen in figure D.5 on the next page

When the voltage over the resistor reaches a steady stage it means that the rotation of the motor has too, therefor measuring a time constant for the

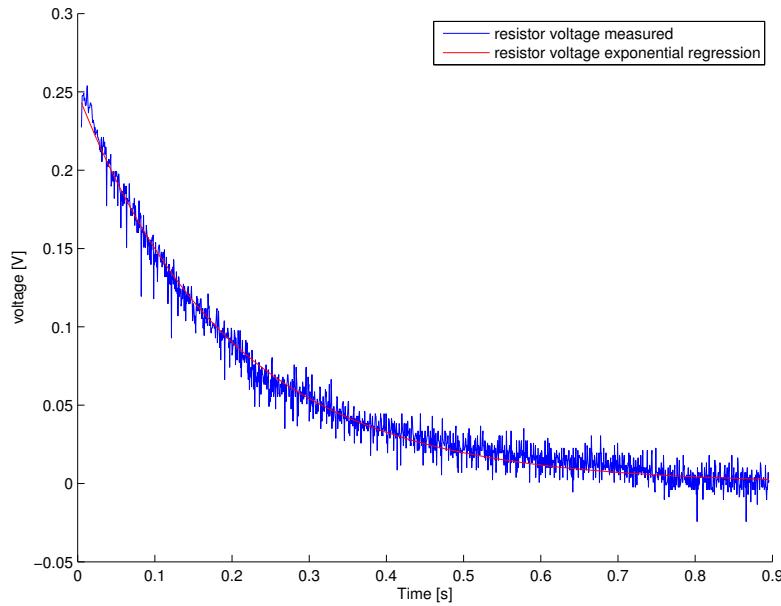


Figure D.5: Measured and approximated voltage over resistor, when motor is given a voltage step.

voltage over the resistor will get the same result as measuring it directly on the motor.

The time constant is set to when the voltage over the resistor has dropped to 37 % of the initial voltage, and is found by an iterative process through a MATLAB program. The time constant is found for three different measurements, and the mean of these are 0.208833 s.

Since τ_m has been found equation D.2 on the facing page can be evaluated, and the inertia of the motor is found to be: $J = 0.000008147282236$.

D.5 Results

Name	Symbol	Value	Unit
Motor constant	K	0.005343688804	[-]
Coulomb friction	τ_c	0.006366937622	[$N \text{ m}$]
Winding resistance	R	0.8093833687	[Ω]
Friction	b	0.000003733428547	[$\frac{N}{rad/s}$]
Mechanical time constant	τ_m	0.208833	[s]
Motor inertia	J	0.000008147282236	[$kg \text{ m}^2$]

Table D.2: List of motor parameters.

E

Vessel Dimensions

This chapter documents the vessels physical dimensions, which are used for the approximations and calculating various constants. The areas above and below the surface of the water has been approximated from the test vessel. Figure E.1 shows the vessels width and various lengths used throughout this project. The figure E.2 shows the lengths of the vessel as well as the draught.

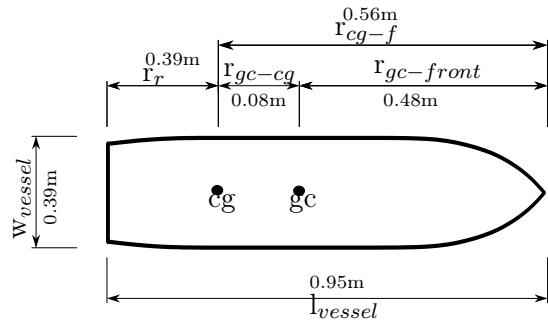


Figure E.1: Top view of measured lengths, where r is the arm from the the centre of gravity, cg .

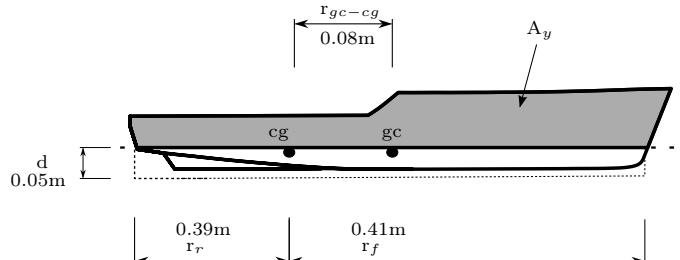


Figure E.2: Side view of measured lengths, where r is the arm from the the centre of gravity, cg .

Measurement	Measured value
Cross section area below the water line A_{xa}	0.0095 m^2
Cross section area above the waterline A_{xb}	0.0182 m^2
Side area below the water line A_{ya}	0.04 m^2
Side area above the water line A_{yb}	0.097 m^2
Weight	3.7 kg
draft	0.05 m

Figure E.3: Notation and units for the body fixed frame

F

Vessel tests

Before proceeding to the accept tests smaller test has been performed. Not taking into account the individual test for each of the modules in the software, two tests of the vessel have been performed.

First test

The first test, which was carried out the 17th of May, was primarily for testing whether or not the GPS receivers could get a fixed position and output the correct speed of the vessel. To test this the vessel was placed on a trolley with a computer connected to it. A computer external of the vessel was necessary at this point, as the IGE board did not support the rtk program, and could as such not run the software intended for the vessel.

For the program on the vessel to proceed from its initializing state into the main loop, the GPSs have to get a fix of a certain rate. As the test was being performed, this fix rate was not met at any given time. And as such, to test whether the vessel could received GPS coordinates and log these correctly, the GPS fix rate was lowered such that the vessel could enter the main loop of the software.

The vessel was able to log the coordinates which it registered. How ever these coordinates was, as the fix rate was low, far and wide from where the GPS receivers was located. In figure F.1 and F.2 on the following page shows the GPS coordinates from each of the receivers.

As it can be seen in both the figures (F.1 and F.2 on the following page) the coordinates seems to drift far off in the right hand side of where coordinates



Figure F.1: GPS1 coordinates first test.



Figure F.2: GPS2 coordinates first test.

are registered. The cause of the drifting is the initializing process where the GPS tries to get a fix. The points recorded under the walk, which was once around the parking lot, seems somewhat reasonable individually but if these coordinates are put together and used by the vessel, the vessel seems to move sideways.

Second test

The second test was done at the night of the 21st of May. The figures (F.3 on the next page and F.4 on the facing page) show that in the second test, as in the first the GPS drifts in the initializing phase. In this test the fix rate was kept low from the first test, as the fix rate was settling. It can be seen that from the three rounds around the parking lot, that the difference in the two GPSs coordinates varies is more than the metre that is the real length between the two. Further more the coordinates are beyond one metre from the traveled route.

As well as testing the GPS coordinate precision, the vessels ability to find its speed and using the thrust motor to obtain this speed was being look upon. The calculations of speed resulted in motion from the thrust motor. When the trolley moved slowly or was at a stop, the propeller rotated, and when the trolley moved faster than the planned speed, the propeller stopped.



Figure F.3: GPS1 coordinates second test.



Figure F.4: GPS2 coordinates second test.

ARGO datasheet

G

ARGO is a tool aimed at surveyors for small to medium sized confined water depth measurement tasks, e.g. depth measurement of a small lake or harbour. ARGO is able to operate with high RTK-GPS precision in areas with 3G coverage. This datasheet explains how to operate ARGO, how it is assembled and relevant technical specifications.

G.1 Operating ARGO

Setup

The predetermined route the vessel have to follow is made from WGS84 coordinates, the x- and y-coordinates are put into a csv file. The first column is for the x-coordinates, the second column is assigned for y-coordinates and in the entire third column there is written a semicolon, such that the code does not read any further in the buffer which is filled with a new row each loop. The csv file is placed in the same directory, on the on-board computer, as the main program.

Or at least that is how it is intended, the code has been written and is functional, but it has not been implemented in the main program, the coordinates for the route is of now hardcoded into the program.

Launching the vessel

Before launching the vessel, make sure that there is no obstacles in the predetermined route since the vessel has no obstacle detection/avoidance system, sealing off the area is also recommended, though the vessel can be called back to its starting point at any point.

To launch the vessel, the vessel is simply put into the water and turned on, the vessel will complete the predetermined route, if not interrupted, and return to where it was put into the water. Since the vessel returns directly to where it was put into the water when it reaches the last waypoint, make sure that there is no obstacles between the end point of the route and where the vessel is put into the water.

Post measurement

When the vessel has completed its task, make sure that any water that has leaked into the vessel is drained out and the battery is recharged.

G.2 Technical information

Specifications

Aspect	Value
Surveying speed	$1\frac{m}{s}$ (2σ)
Hull length	0.95 m
Hull width	0.39 m
Hull height	0.12 m
Hull draught	0.05 m
Rudder draught	0.09 m
Battery	NiCd, 3.6 Ah, 7.2 V
Weight	3.7 kg
Communication	mobile 3G USB modem
GPS antennas	SM-1575
GPS receiver modules	LEA-4T
RTK-GPS	within 0.05 m

Figure G.1: Vessel specifications list

Inner construction of vessel

The vessel is made up of three major parts:

Main electronics box

The computing in the vessel is handled by an IGEPv2 board and an ARM7 micro-controller, the IGEP board handles the communication and all the control algorithms while the microcontroller converts the control signals into PWM signals that are feed to the servo controlling the rudder, and the H-bridge controlling the motor driving the propeller. A 3G modem/antenna is mounted on top of the vessel and is connected to the vessel via USB cable.

On top of the microcontroller is a custom made shield [Dam et al., 2011, page 136], that interfaces the mirocontroller to the actuators, and the IMU to the microcontroller.

Inside the box is also a USB-hub that is connected to the IGEP board, and can be reached from outside the box.

All the circuit boards can be seen on figure G.2 on the next page. The IGEP board is the board on the left, the microcontroller sits on the right with the shield mounted on top. The USB-hub can be seen in the lid, to the left, together with various connectors connecting the actuators to the shield, power to the electronics and communication capability to the IGEP board and the microcontroller.

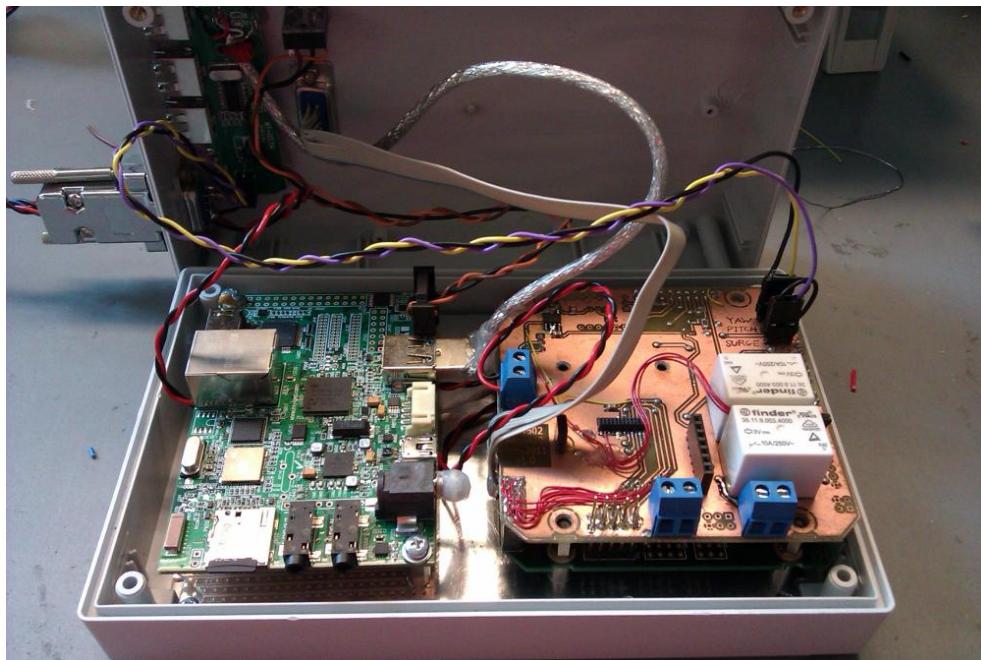


Figure G.2: Electronics box

The GPS receiving system

A GPS antenna is mounted in the front of the vessel, and another one in the back, the two receiver modules are stored in the front of the vessel and connected to the IGEP via USB cables.

The actuator system

The actuators consist of a brushed DC motor and a standard servomotor. The propeller is mounted on the rudder and is driven through a propeller shaft and a universal joint by the DC motor. The servo drives the rudder.

Transfer function implementation

H

This appendix describe the transformations from state space to the discrete time that can be used in the software implementation.

H.1 State Space to Laplace domain

In general, a state space to laplace transformation can according to [Stoustrup, 2012] be given as:

$$\mathbf{G}(s) = \mathbf{C} \cdot (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \quad (\text{H.1})$$

Where:

- $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the state space matrices, see equation 10.1 on page 43
- \mathbf{I} an identity matrix with same dimensions as \mathbf{A}
- $\mathbf{G}(s)$ the transfer functions in a matrix.
- s the laplace variable (representing frequency)

H.2 Laplace domain to discrete time

In order to use the transfer functions found in 6.1 on page 25 in the software they should be z -transformed to discrete time, where the output is dependent of the previous and current input and output. Converting from $H(s)$ to $H(z)$ is done using Tustin's Bilinear transformation and is described through this section.

Substituting s in the transfer function, $H(s)$, obtained when modeling the system, described in chapter 6.1 on page 25, with equation H.2, where T is the sampling time. The sampling time is 5 Hz meaning that T becomes $\frac{1}{5}$.

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad (\text{H.2})$$

The resulting function $H(z)$ is given as H.3.

$$H(z) = \frac{Y(z)}{X(z)} \quad (\text{H.3})$$

The inverse z -transformation is applied on this function and is solved for $y[n]$ in discrete time.

Bibliography

P. Albertos and A. Sala. *An Engineering approach.* Springer, 2004. ISBN 1-85233-738-9.

Charles M. Close, Dean K. Frederick, and Jonathan C. Newell. *Modeling and Analysis of Dynamic Systems.* John Wiley & Sons, INC, 2002. ISBN 0471394424.

Jeppe Dam, Nick Østergaard, Lasse Høj Nielsen, Simon Als Nielsen, Brian Bach Nielsen, and Rasmus Lundgaard Christensen. *Initial Development of an Autonomous Underwater Vehicle.* <http://kom.aau.dk/group/11gr502/master.pdf>, 2011.

Danish Meteorological Institute (DMI). *Danish Meteorological Institute (DMI).* <http://www.dmi.dk/>, 2012. Visited February 20 2012.

Jay. A. Farrel and Matthew Barth. *The Global Positioning System & Inertial Navigation.* McGraw-Hill, 1998. ISBN 0-07-022045-X.

Gene F. Franklin, J.David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems.* Pearson Prentice Hall, 2009. ISBN 9780135001509.

Finn Haugen. *Discrete-time signals and systems.* <http://www.scribd.com/doc/26420589/37/Discretization-of-linear-continuous-time-state-space-models>, 2005. Visited on 30/5/2012.

ISO. *ISO 2533:1975 Standard Atmosphere.* 1975.

LinuxHowTos. *LinuxHowTos Sockets Tutorial.* http://linuxhowtos.org/C_C%2B%2B/socket.htm, 2012. Visited May 5 2012.

Mathworks. *Matlab mapping toolbox.* www.mathworks.com/, 2012.

NASA. Drag coefficient.

Tom Pedersen and Palle Andersen. slidesengmm5. slides, 2011.

Asgeir J. Soerensen. *Marine Cybernetics Modelling and Control.* Department of Marine Technology NTNU, 2005. ISBN UK-05-76.

Jakob Stoustrup. Modern control theory lectures. 2012.

T. Takasu. *RTKLIB ver. 2.4.1 Manual.* www.rtklib.com/, 2012. Visited April 10 2012.

M. T. Tham. *An introduction to decoupling control.* <http://lorien.ncl.ac.uk/ming/mloop/mloop.pdf>, 1999. s. 4–9 of 19.

The Engineering Toolbox. *The Engineering Toolbox.* <http://www.engineeringtoolbox.com/>, 2012. Visited Feburary 4 2012.