NTNU
Norwegian University of
Science and Technology

Auto-Docking of Vessel

Simon Bårslett (10016)

Martin Nikolai Longva (10056)

Thor-Inge Nygård (10044)

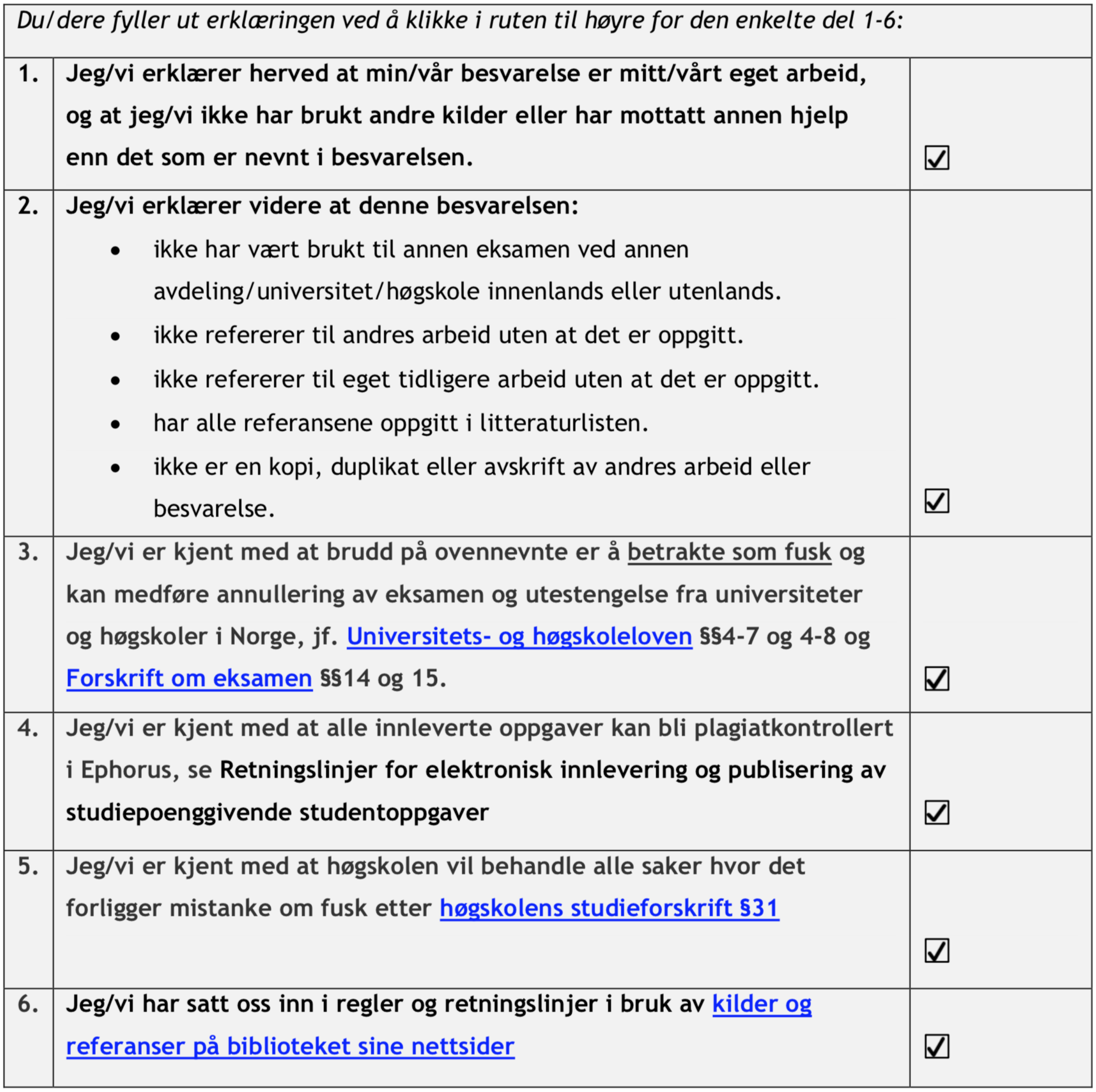
June 2018

Bachelor Thesis
Department of ICT and Natural Sciences
Norwegian University of Science and Technology

Supervisors: Ottar L. Osen & Robin T. Bye

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. **Manglende erklæring fritar ikke studentene fra sitt ansvar.**



Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	**Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.**	☑
2.	**Jeg/vi erklærer videre at denne besvarelsen:** • ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • ikke refererer til andres arbeid uten at det er oppgitt. • ikke refererer til eget tidligere arbeid uten at det er oppgitt. • har alle referansene oppgitt i litteraturlisten. • ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	☑
3.	Jeg/vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§14 og 15.	☑
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se **Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver**	☑
5.	Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens studieforskrift §31	☑
6.	**Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider**	☑

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven §2).
Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage HiM med forfatter(ne)s godkjennelse.
Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering: ☑**ja** □**nei**

Er oppgaven båndlagt (konfidensiell)? □**ja** ☑**nei**
(Båndleggingsavtale må fylles ut)
- Hvis ja:
Kan oppgaven publiseres når båndleggingsperioden er over? □**ja** □**nei**

Er oppgaven unntatt offentlighet? □**ja** ☑**nei**
(inneholder taushetsbelagt informasjon. Jfr. Offl. §13/Fvl. §13)

Dato:

Preface

The main objective of this thesis is to find possible solutions for autonomous docking of a vessel, by researching different sensors and simulating them. It was carried out by three Automation engineering students at NTNU Ålesund, and the assignment "Auto-Docking of Vessel" was given by Rolls Royce Marine. It was written during the spring of 2018.

As the third year of our bachelor degree in automation comes to an end, we can utilize the knowledge we have obtained during these six semesters to make a working simulation for autonomous docking of a vessel. This assignment is especially relevant for our type of study, where the offshore business have been in focus. All three members have relevant working experience from both shipbuilding and electrical maintenance on offshore vessels from before we started this bachelor degree.

The assignment was given because Rolls-Royce Marine is delivering a auto-docking system during 2018, and they wanted input from an external source. This assignment is primarily written for them and the readers that find autonomous ship technology interesting.

Ålesund, 01.06.2018

Simon Vågeskar Bårslett

Martin Nikolai Longva

Thor-Inge Eidsvik Nygård

Acknowledgment

We would like to thank our supervisors for their invaluable help and guidance during this thesis, associate professors Ottar L. Osen and Robin T. Bye at NTNU in Ålesund. We would also give a special thanks to Solveig Bjørneset, R&D Engineer at Rolls-Royce Marine, for helping us with vessel kinematics, simulation and general advice during this thesis.

S.V.B.

M.N.L

T.I.E.N

Abstract

The maritime industry have in recent years started to focus more on autonomous vessels. There is already technology like auto-crossing for ferries, and dynamic positioning for offshore vessels. One of the next steps regarding this subject is to make the docking process autonomous.

When it comes to auto-docking of a vessel, the requirements are demanding and high precision is required. The vessel will need to be able to manoeuvre with centimeter position accuracy, for the process to become safe and comfortable. Since many vessels already have dynamic positioning systems, the simulations are based on a DP-Controller. The controller will use position and distance measurements as inputs from an observer that estimates the position. The different simulations will use one or two sensors in combination with GNSS technology, and will be benchmarked by their deviation between estimated and actual position. The simulation is carried out in Matlab and Simulink with the use of the Marine Systems Simulator toolbox. The different simulations proved that LiDAR and RTK GPS was the most accurate solution, with accuracy down to a couple of centimeters.

Table of Contents

List of Figures

List of Tables

Terminology

Notation

$A_L(m^2)$	Lateral projected area
A_{SS}	Lateral projected area of superstructure
$A_t(m^2)$	Transverse projected area
B	Beam
C	Distance from bow of centroid of lateral projected area
C_N	Moment coefficient
C_X and C_Y	Empirical force coefficients
$L(m)$	The overall length of the ship
M	Number of sitinct groups of masts or king posts seen in lateral position
S	Length of perimeter of lateral projection of model excluding waterline and slender bodies such as masts and ventilators
u_r	Component of V_r in x direction
V_ω	Denotes the wind speed
V_r	Wind speed relative to the vessel in m/s
v_r	Component of V_r in y direction
ψ_ω	Deotes the wind direction relative to North/East
ψ	The angle of the vessel relative to North/East
ψ_r	Wind angle relative to the vessel
γ_r	Angle of the wind relative to the ship bow
$\rho_a(kg/m^3)$	The density of air

Control Variables

α	Thruster angle
η	Position vector
ν	Velocity vector
$R(\psi)$	Rotation matrix in yaw
τ_R	Radiation-Induced force vector
τ_{PID}	PID force vector
τ_{thr}	Thruster commanded force vector

Abbreviations

BHP	Break Horse Power
BP	Bollard Pull
COG	Center of Gravity
DGPS	Differential Global Positioning System
DOF	Degrees of Freedom
DP	Dymamic Poistioning
ESA	European Space Agency
GA	Genetic Algorithm
GLONASS	Global Orbiting Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GYRO	Gyro Compass
LiDAR	Light detection and range.
MIMO	Multiple-input and multiple-output.
MSS	Marine Systems Simulator
MUSIC	MUSIC (MUltiple SIgnal Classification)
NED	North-East-Down
NPO	Nonlinear Passive Observer
PID	Proportional Integral Derivative
RADius	Relative position reference system.
RSSI	Received signal strength indication.
RTK	Real Time Kinematic
UHF	Ultra High Frequency
UTC	A timezone. UTC+01:00 can be used in Norway.
VHF	Very High Frequency

Chapter 1

Introduction

When mentioning autonomous vehicles most people think about self-driving cars, yet autonomous vessels is currently a hot topic in the maritime industry. The possibility that in the near future we might see fully autonomous vessels, even use them in our everyday life is mind-blowing. This could make vessels more cost-effective, environmental friendly, and safer than what they are today.

However, people might be skeptical about self-driving vehicles, and autonomous vessels are no exception. In order to make companies, crew and possibly the passengers feel comfortable with such a vessel it is important to have a reliable system which performs well at all time.

A system responsible for automatically controlling a vessel must therefore be redundant and functional to the set limitations. In order to make a safe, trustworthy autonomous vessel there is a lot of internal and external variables to keep in mind. From keeping the desired position, balancing thrust and deciding what sensors to prioritize at any given time, to waves, wind, weather and other vessels. For a system to be able to take all of this into account, the sensors used can be the difference between a good and a bad system.

Therefore, this thesis will take on some sensors which could be used to realize autonomous docking of a vessel. A few of these sensors will be simulated along with GNSS to verify their accuracy in order to see if they are precise enough to be integrated with a dynamic positioning system and auto-docking process.

While doing this, the thesis will also explain some important topics for auto-docking, like mathematical modeling for low speed control and position keeping of a vessel. How sensors work and why they can be used.

1.1 Background

Automation have changed many industries. The ferry and cargo industries are two which comes to mind when thinking about what industries might be changed by autonomous vessels, and auto-docking is a crucial part of this process. The vessels in these industries often follows a predetermined route, this makes it easier to tailor a system for each vessel. Since the system can be tailored to each vessel and situation, these industries makes for good testing grounds to try out and introduce autonomous vessels.

There are not many vessels developed yet which can sail autonomously. Those who can transit automatically, can not yet dock by themselves. Fully autonomous vessels are still being developed and tested, maybe we will see them in the near future. Kongsberg, Wärtsilä and Rolls-Royce are some companies invested in this kind of technology, and it will be exciting to see their solutions be used and how they may differ.

Problem Formulation

When operating a vessel that travels the same route several times in one day, the process of docking is performed quite often. This is especially true with ferries. It is a repetitive task which can become tedious to perform again and again throughout the day. Fuel usage and docking time may vary from captain to captain. This is to be expected from humans, as everyone is different and might handle a vessel differently. It should therefore be of interest to most companies which revolves around ships with many short routes to have this process automated. This could make the whole process more cost-effective and smoother.

Related work

Dynamic positioning (DP)

Dynamic positioning systems was first developed in the 1960's when the first controllers managed to simultaneously control a vessel in three horizontal motions surge, sway & yaw. The first systems utilized a single-input/output PID-controller combined with a low pass/notch filter. The systems was continuously under development, and in the 1970's Balchen introduced the theory of multivariable optimal control and Kalman filter theory[4]. Then a nonlinear control was introduced later on.

When controlling a vessel using DP, it is necessary to be able to affect the vessels motion in 3 degrees of freedom(DOF). Here the thrusters are the source of force for keeping the vessel at the given position. For this system to work under harsh and challenging condition, it is expected that the sensors provide the needed data. The main ones are position measurement like GNSS & gyro readings, in addition to wind sensors and accelerometers for evaluating the force acting on the vessel.

Traditionally DP systems is used for keeping a given position, low speed motion or keeping a certain heading. When it comes to high speed applications the autopilot is the most common control system. It is custom these days is to merge the DP system with a high speed application, which creates a very agile system. This opens up the opportunity to use a way-point guidance system to get to a destination, and then switch over to DP when the desired position is reached and then keep it. [21] DP systems are also used for safety and redundancy on board ships. DP has 4 classes: [9]

- DP 0: Loss of position may occur in the event of a single fault.
- DP 1: Loss of position may occur in the event of a single fault. With some redundancy requirements. See table 2.1 section 2 in DNV GL [9]
- DP 2: a loss of position shall not occur in the event of a single fault in any active component or system. Static components will not be considered to fail where adequate protection from damage is demonstrated and reliability is deemed acceptable by DNV GL.
- DP 3: a loss of position shall not occur in the event of a single fault in any active or static component or system. This applies also for the total failure of one compartment due to fire or flooding

1.2 Objectives

The main objectives of this Bachelor project is

1. To research a variety of sensors which might be used in an automatic docking operation.
2. To create a simulation environment for the different combinations of sensors.
3. To simulate the automatic docking operations using a 2D visualization.
4. To make one or more examples of a redundant system for automatic docking.

1.3 Approach

The main issues in this project is first and foremost to make a realistic simulation of different types of sensors that will display the functionality and how different sensors work in different ways. Since there will be no physical testing, it is necessary to make the simulation as close to reality as possible. The simulations will be carried out in Matlab and Simulink, with the MSS toolbox as foundation for the simulation environment. Where the most relevant sensors and systems will be tested.

Marine Systems Simulator (MSS) is a toolbox built in Matlab and Simulink, for simulation of a series of marine applications. The toolbox covers most of the relevant subjects when it comes to marine applications. There are different vessel models, with possibility to choose different observers, controllers and external forces, i.e. wind. The toolbox includes the mathematical models for marine applications, with examples for simulating DP-systems, autopilots, way-point guidance and so on. There is also a possibility for user defined Matlab-files, for the user to define the application as desired. [6]

1.4 Limitations

This thesis is limited to simulation, as no physical model will be built. The task is to research what kind of solutions could be used in a autonomous docking process of a vessel. The systems will not be physically tested, but simulations of some sensors will be carried out using Simulink and Matlab. This theses will describe some possible solution for autonomously docking a vessel, and provide insight into what kind of sensors could be implemented in such a system and a basic explanation of how the sensors work and how they might be implemented.

1.5 Outline

- **Chapter 2, Mathematical Methods:**
  In this chapter we present the mathematical methods relevant for simulating marine applications, and which can be applied in a auto-docking application
- **Chapter 3, Sensors:**
  In this chapter we present the sensors relevant for a auto-docking system.
- **Chapter 4, Simulation:**
  In this chapter we present the simulation model used to simulate the auto-docking process and the sensors.
- **Chapter 5, Results:**
  In this chapter we present the results from the different test cases ran in the simulation model.
- **Chapter 6: Discussion:**
  In this chapter we discuss the results regarding the sensors that can be used in auto-docking, and present the concluding remarks from the test cases.
- **Bibliography**
- **Appendix A:**
  The appendix contains further explanation of the Matlab functions, Simulink/MSS models used for simulation, and images from the eta deviations from the simulations. It also contains all the weather data from Meteorologisk Institutt, visualized.
- **Appendix B:**
  Contains the attachments: DNV GL table, Preliminary report, GANTT Diagram and meeting minutes

Chapter 2

Mathematical methods

The following equations and mathematical methods are collected from the books Fossen, Marine Control Systems [7] and Sørensen, Marine Control Systems [21]

2.1 Vessel Kinematics

The study of dynamics can be split in to two parts: Kinematics and kinetics. **Kinematics** which only focuses on the geometrical aspects of motion and **kinetics** that is analysis of the forces creating motion. [7]

Degrees Of Freedom
For describing the motion of a marine vessel six independent variables are used, this is the six degrees of freedom. They are used to define position and orientation of the vessel. The first three, surge, sway, and heave are used to describe position along the x, y, and z-axes. The last three, roll, pitch, and yaw describes rotation and orientation around the x, y, and z-axes.

The notation of SNAME is used to describes the motion components for a six DOF vessel, see Table 2.1

Table 2.1: The notation of SNAME (1950) for marine vessels

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

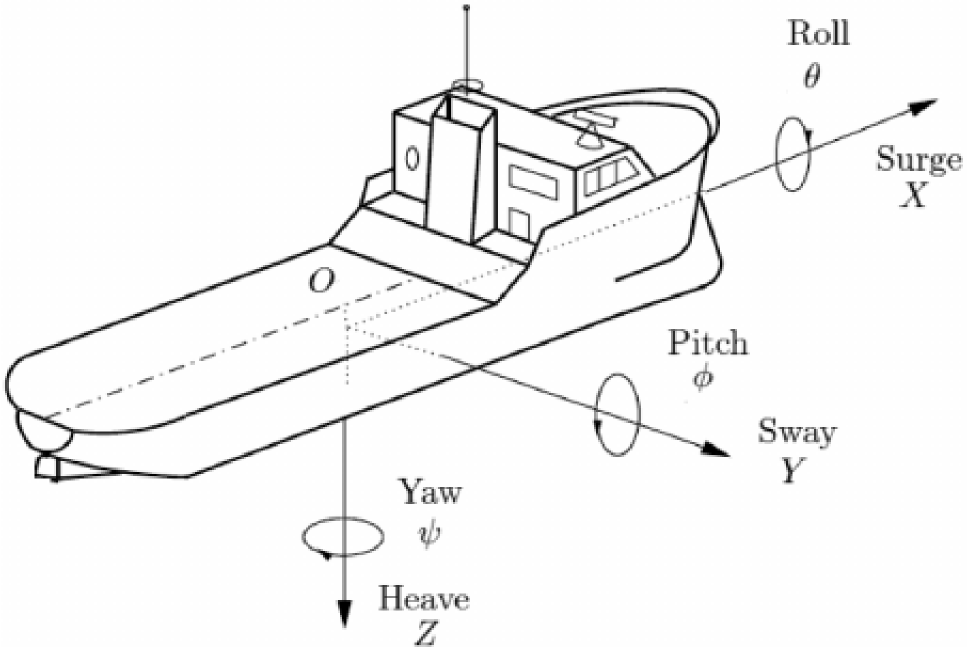


Figure 2.1: Illustration of the six degrees of freedom

From Table 2.1 the velocity and position vectors can be extracted.

$$v = [u, v, w, p, q, r]^T \tag{2.1}$$

$$\eta = [x, y, z, \phi, \theta, \psi]^T \tag{2.2}$$

2.1.1 Reference Frames

For describing and analyzing the motion of a marine vessel in six DOF, more than one coordinate frame have to be defined. There are two earth-centred frames and one BODY-frame.

Earth-centred inertial frame (ECI) is an inertial terrestrial navigation, with origin at the center of the Earth in fixed space. Its also a nonaccelerating reference where Newton's laws of motion apply using the coordinates x_i, y_i, z_i.

Earth-centred Earth-fixed reference frame (ECEF) is using the coordinates x_e, y_e, z_e. Similar to ECI it has its origin fixed to the center of the Earth, but rotates relative to the inertial frame ECI. The angular rate for determine the rotation is $\omega_e = 7.2921 \times 10^{-5}$ (Figure 2.2)

In additional the geographical reference frames are the ones describing the motion of a marine vessel.

North-East-Down (NED) coordinate system is relative to the Earths reference ellipsoid. It's the coordinate system used in everyday life. It can be seen as a tangent plane on the surface of the earth moving with the object. For this reference frame the x-axis aims against true NORTH, the y-axis against EAST and the z-axis points downwards to the Earth surface. It is using the coordinates x_n, y_n, z_n, where the location of the n-frame is relative to the e-frame. It's determined by using the angles l and μ for indicating the *longitude* and *latitude*. It is used for defining the position vector μ.

BODY-fixed reference frame. It is a coordinate frame that follows the vessel, and is using the coordinates x_b, y_b, z_b. The origin of this frame is usually located at the center of gravity of the given vessel. It is used for defining the velocity vector v.[7]

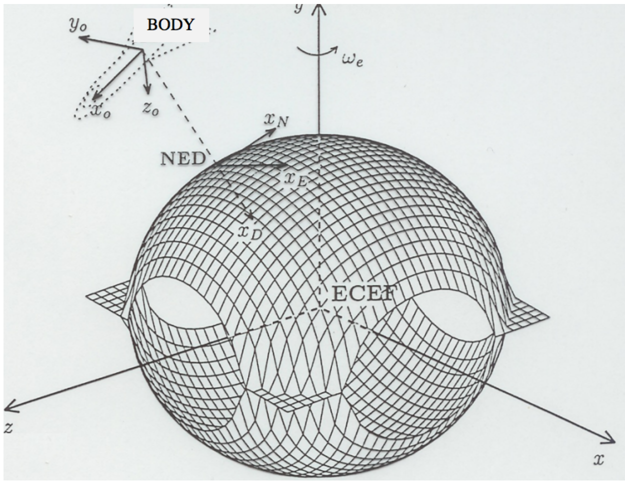


Figure 2.2: Illustrating the different reference frames

2.1.2 3 DOF Horizontal Model

DP applications is usually based on a 3 DOF Horizontal model. This means the motion is described by the motion components in surge, sway and yaw. The 3 other DOF- heave, pitch and roll is considered neglected. Then $\mathbf{v} = [u, v, r]^T$ and $\eta = [n, e, \psi]^T$ is chosen from the vectors (Equation 2.1 & 2.2)

2.1.3 Rigid-Body Dynamics

The rigid-body dynamics can be described using Equation 2.3, which comes from Newtons mechanics. It is used for describing the forces and moments acting on a rigid-body model. M_{RB} (Equation 2.4) is the system inertia matrix for a rigid-body, ν is the generalized velocity vector (Equation 2.1), C_{RB} is the Coriolis matrix and $\tau_{RB} = [X, Y, Z]^T$ is representing the external forces and moments as a vector in BODY-coordinates.

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB} \tag{2.3}$$

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{2.4}$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \tag{2.5}$$

2.1.4 Hydrodynamic Forces and Moments

When modeling a low speed application hydrodynamic forces and moments are acting on the vessel. The main topics are Radiation-Induced Forces and Environmental Disturbances. **Radiation-Induced Forces** and moments is divided in to three components.

- Added mass due to the inertia of the surrounding fluid
- Radiation-induced potential damping due to the energy carried away by generated surface waves
- Restoring forces due to Archimedes (weight and buoyancy).

The mathematical representation of these forces can be written as Equation 2.6

$$\tau_R = -\underbrace{M_A\dot{\nu} - C_A(\nu)\nu}_{\text{added mass}} - \overbrace{D_P(\nu)\nu}^{\text{potential damping}} - \underbrace{g(\eta) + g_o}_{\text{restoring forces}} \tag{2.6}$$

Environmental Disturbances

It's not only hydrodynamic forces that will affect the vessel. Environmental forces like wind, waves and currents will also contribute.

System inertia matrix

The system inertia matrix divided in to two parts. M_{RB} (2.4) which represents the rigid-body dynamics and M_A (2.7) which represents added mass. Combining these will result in the total system inertia matrix $M = M_{RB} + M_A$ (Equation 2.8)

$$M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix} \tag{2.7}$$

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.8}$$

Hydrodynamic damping is dependent on a series of different parameters. The main ones are potential damping, skin friction, wave drift damping and vortex shedding damping. All the different terms are a part of both linear and quadratic damping. It is normal to represent hydrodynamic damping as $D(v) = D + D_n(v)$. D represents the linear damping matrix (Equation 2.9), and $D_n(v)$ is the nonlinear damping matrix.

The linearized damping matrix for low speed vessels with xz-symmetry where the surge mode can be decoupled from the steering models in sway and yaw. It focuses on the moments in 3 DOF (surge,sway and yaw) where heave, pitch and roll are neglected.

$$D = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \tag{2.9}$$

2.2 Wind Mathematics

Let V_ω denote the wind speed, while ψ_ω denotes the wind direction.

$$V_\omega(h) = V_\omega(10) * (h/10)^{1/7} \tag{2.10}$$

Let $V_\omega(10)$ be the relative wind velocity 10(m) above the sea surface.

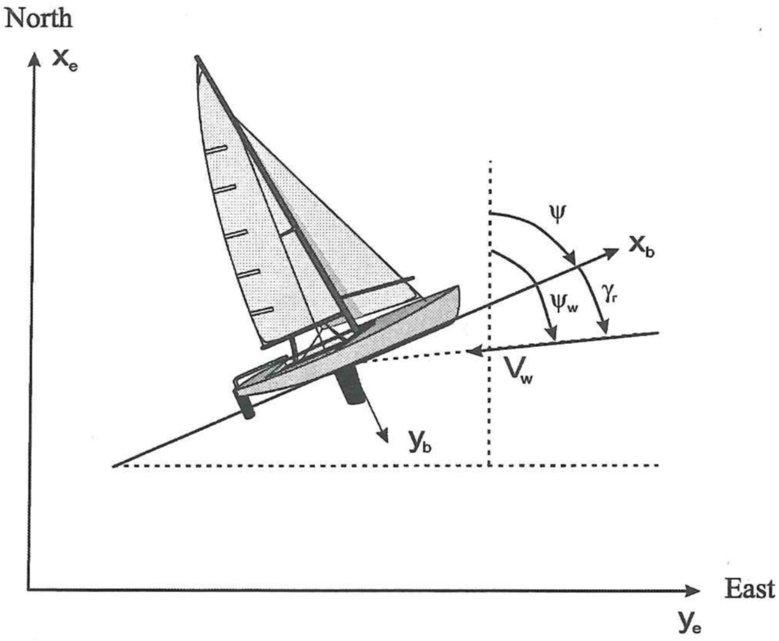


Figure 2.3: Definition of wind speed V_ω and direction γ_r

The wind forces and moments acting on a vessel can be defined as seen in Equation 2.11 in terms of relative wind speed V_r and angle γ_r.

$$V_r = \sqrt{u_r^2 + v_r^2}, \qquad \gamma_r = \tan^{-1}(v_r/u_r) = \psi_\omega - \psi \tag{2.11}$$

Equation 2.12 shows the components of V_r in the x- and y-directions.

$$u_r = V_\omega \cos(\gamma_r) - u, \qquad v_r = V_\omega \sin(\gamma_r) - v \tag{2.12}$$

$\gamma_r = \psi_\omega - \psi$ can be interpreted as the angle of the wind relative to the ship bow as seen in Figure 2.3. To be able to implement wind feedforward compensation for a surface vessel, a 3 DOF wind model as a function of relative wind speed and direction, V_r and γ_r, is needed. The generalized force vector $W_{\text{wind}} = [X_{\text{wind}}, Y_{\text{wind}}, N_{\text{wind}}]^T$ is used for this purpose.

It have been suggested by Isherwood (1973)[14] that one can write the wind forces (surge and sway) and moment (yaw) as seen in Equation 2.13.

$$\begin{aligned}
X_{\text{wind}} &= \frac{1}{2} C_X(\gamma_r)\rho_a V_r^2 A_T \quad (N) \\
Y_{\text{wind}} &= \frac{1}{2} C_Y(\gamma_r)\rho_a V_r^2 A_L \quad (N) \\
N_{\text{wind}} &= \frac{1}{2} C_N(\gamma_r)\rho_a V_r^2 A_L L \quad (Nm)
\end{aligned} \tag{2.13}$$

Where C_X and C_Y are empirical force coefficients, C_N is a moment coefficient, $\rho_a(kg/m^3)$ is the density of air, $A_T(m^2)$ and $A_L(m^2)$ are the transverse and lateral projected areas, and $L(m)$ is the overall length of the vessel. V_r, the speed, is given in m/s.

Table 2.2: Wind force parameters in surge

$\gamma_r(deg)$	A_0	A_1	A_2	A_3	A_4	A_5	A_6	S.E.
0	2.152	-5.00	0.243	-0.164	-	-	-	0.086
10	1.714	-3.33	0.145	-0.121	-	-	-	0.104
20	1.818	-3.97	0.211	-0.143	-	-	0.033	0.096
30	1.965	-4.81	0.243	-0.154	-	-	0.041	0.096
40	2.333	-5.99	0.247	-0.190	-	-	0.042	0.115
50	1.726	-6.54	0.189	-0.173	0.348	-	0.048	0.109
60	0.913	-4.68	-	-0.104	0.482	-	0.052	0.082
70	0.457	-2.88	-	-0.068	0.346	-	0.043	0.077
80	0.341	-0.91	-	-0.031	-	-	0.032	0.090
90	0.355	-	-	-	-0.247	-	0.018	0.094
100	0.601	-	-	-	-0.372	-	-0.020	0.096
110	0.651	1.29	-	-	-0.582	-	-0.031	0.090
120	0.564	2.54	-	-	-0.748	-	-0.024	0.100
130	-0.142	3.58	-	0.047	-0.700	-	-0.028	0.105
140	-0.677	3.64	-	0.069	-0.529	-	-0.032	0.123
150	-0.723	3.14	-	0.064	-0.475	-	-0.032	0.128
160	-2.148	2.56	-	0.081	-	1.27	-0.027	0.123
170	-2.707	3.97	-0.175	0.126	-	1.81	-	0.115
180	-2.529	3.76	-0.174	0.0128	-	1.55	-	0.112
							Mean S.E.	0.103

Table 2.3: Wind force parameters in sway

$\gamma_r(deg)$	B_0	B_1	B_2	B_3	B_4	B_5	B_6	S.E.
10	0.096	0.22	-	-	-	-	-	0.015
20	0.176	0.71	-	-	-	-	-	0.023
30	0.225	1.38	-	0.023	-	-0.29	-	0.030
40	0.329	1.82	-	0.043	-	-0.59	-	0.054
50	1.164	1.26	0.121	-	-0.242	-0.95	-	0.055
60	1.163	0.96	0.101	-	-0.177	-0.88	-	0.049
70	0.916	0.53	0.069	-	-	-0.65	-	0.047
80	0.844	0.55	0.082	-	-	-0.54	-	0.046
90	0.889	-	0.138	-	-	-0.66	-	0.051
100	0.799	-	0.155	-	-	-0.55	-	0.050
110	0.797	-	0.151	-	-	-0.55	-	0.049
120	0.996	-	0.184	-	-0.212	-0.66	0.34	0.047
130	1.014	-	0.191	-	-0.280	-0.69	0.44	0.051
140	0.784	-	0.166	-	-0.209	-0.53	0.38	0.060
150	0.536	-	0.176	-0.029	-0.163	-	0.27	0.055
160	0.251	-	0.106	-0.022	-	-	-	0.036
170	0.125	-	0.046	-0.012	-	-	-	0.022
							Mean S.E.	0.044

Table 2.4: Wind force parameters in yaw.

$\gamma_r(deg)$	C_0	C_1	C_2	C_3	C_4	C_5	C_6	S.E.
10	0.0596	0.061	-	-	-	-0.074	0.0048	
20	0.1106	0.204	-	-	-	-0.170	0.0074	
30	0.2258	0.245	-	-	-	-0.380	0.01015	
40	0.2017	0.457	-	0.0067	-	-0.472	0.0137	
50	0.1759	0.573	-	0.0118	-	-0.523	0.0149	
60	0.1925	0.480	-	0.0115	-	-0.546	0.0133	
70	0.2133	0.315	-	0.0081	-	-0.526	0.0125	
80	0.1827	0.254	-	0.0053	-	-0.443	0.0123	
90	0.2627	-	-	-	-	-0.508	0.0141	
100	0.2102	-	-0.0195	-	0.0335	-0.492	0.0146	
110	0.1567	-	-0.0258	-	0.0497	-0.457	0.0163	
120	0.0801	-	-0.0311	-	0.0740	-0.396	0.0179	
130	-0.0189	-	-0.0488	0.0101	0.1128	-0.420	0.0166	
140	0.0256	-	-0.0422	0.0100	0.0889	-0.463	0.0162	
150	0.0552	-	-0.0381	0.0109	0.0689	-0.476	0.0141	
160	0.0881	-	-0.0306	0.0091	0.0366	-0.415	0.0105	
170	0.0851	-	-0.0122	0.0025	-	-0.220	0.0057	
						Mean S.E.	0.0127	

Isherwood (1973)[14] have concluded that the three Equations 2.14 gives the best representation for the wind coefficients of data measured during his work on this matter.

$$C_X = A_0 + A_1\frac{2A_L}{L^2} + A2\frac{2A_T}{B^2} + A3\frac{L}{B} + A_4\frac{S}{L} + A5\frac{C}{L} + A_6 M$$

$$C_Y = -(B_0 + B_1\frac{2A_L}{L^2} + B2\frac{2A_T}{B^2} + B3\frac{L}{B} + B_4\frac{S}{L} + B5\frac{C}{L} + B_6\frac{A_{SS}}{A_L}) \tag{2.14}$$

$$C_N = -(C_0 + C_1\frac{2A_L}{L^2} + C2\frac{2A_T}{B^2} + C3\frac{L}{B} + C_4\frac{S}{L} + C5\frac{C}{L})$$

In these equations, the 8 parameters as seen below were measured from the ship and used.

- L - length overall
- B - beam
- A_L - lateral projected area
- A_T - transverse projected area
- A_{SS} - lateral projected area of superstructure
- S - length of perimeter of lateral projection of model excluding waterline and slender bodies such as masts and ventilators
- C - distance from bow of centroid of lateral projected area
- M - number of distinct groups of masts or king posts seen in lateral projection. King posts close against the bridge front are not included

More about these mathematical methods can be read about in the book Marine Control Systems[7].

2.3 Ferry

The Ferry is the vessel modeled to perform the auto-docking. Since auto-docking is a low speed application based on a DP system, the motion components are described as a 3 DOF horizontal model (Section 2.1.2).

The ferry specifications are based on parameters set by Rolls-Royce Marine with a typical ferry in mind.

- Length between perpendiculars (Lpp) = 90 m
- Lenght over all (Loa) = 100 m
- Breadth = 15 m
- Draft = 3 m
- Block coefficient (Cb) = 0.33
- Wind side area = 800m^2
- Wind front area = 220m^2
- Current side area = 200m^2

Thrusters:

- 2x azimuth thrusters
- Variable rpm
- Electrical
- Max power = 950 kW
- Diameter = 2.5 m
- Max rpm = 220
- x-pos = ± 45 m
- z-pos = 2 m

2.3.1 Vessel Matrices

Since the ferry is symmetrical around its COG it becomes a diagonal matrix where the forces $m - Y_{\dot{v}}$ in the system inertia matrix(2.8) and Y_r, N_v in the Damping matrix(2.9) can be neglected, this is shown in the matrices below. (Equation 2.15 & 2.16)

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.15}$$

$$D = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{2.16}$$

2.3.2 Thruster Control

The commanded forces from the thrusters is given by the Equation 2.17.

$$\tau_{thr} = T_{3\times r}(\alpha) K u_d \tag{2.17}$$

Where $T_{3\times r}(\alpha) \in \mathbb{R}^{3\times r}$ is the thrust configuration matrix, and the angle $\alpha \in \mathbb{R}^{r}$ is the orientation vector for the thurster. Since the vessel in this project is a ferry with two azimut thrusters, the thurst configuration matrix ends up as a 3 × 2 matrix as seen in Equation 2.18.

$$T_{3\times 2}(\alpha) = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 \\ \sin\alpha_1 & \sin\alpha_2 \\ l_{ba}\sin\alpha_1 & l_p \sin\alpha_2 \end{bmatrix} \tag{2.18}$$

l is the length from the COG of the vessel to the position of the azimut thruster in x-position. l_{ba} is for the thruster located in the front, and l_p is for the one located at the stern of the vessel. The angle α_i is representing the orientation of the given thruster. Where $\alpha_i = 0°$ gives the maximum thrust in positive surge direction, and $\alpha_i = 90°$ maximum thrust in positive sway direction.

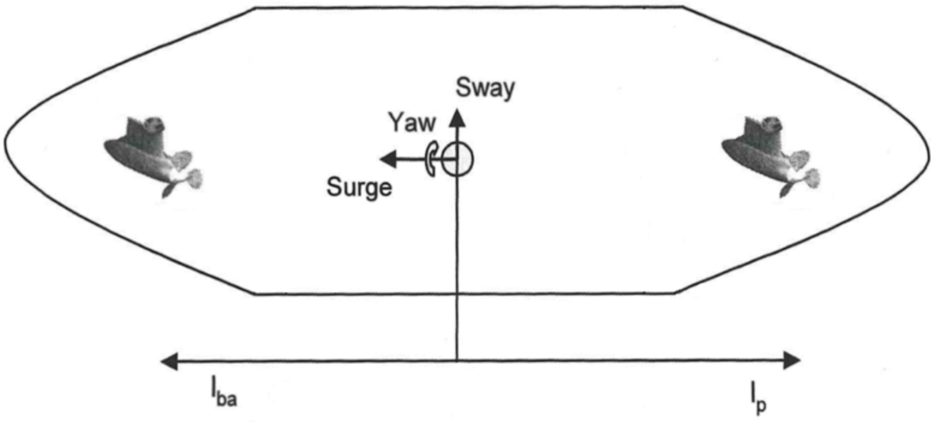


Figure 2.4: Ferry thruster location

2.3.3 Bollard Pull

Since the thrusters are producing 950 kW of power they are able to deliver a certain amount of bollard pull. It is the amount of thrust the azimuth thrusters can affect the vessel in a given direction. As a rule of thumb to calculate bollard pull the Equation 2.19 can be used. This equation takes the assumption that the thruster has a kort-nozzle and with a controllable pitch propeller. The result is in metric tons. [5]

$$\mathrm{BP} = \frac{\mathrm{BHP} * 0.9 * 1.4}{100} \tag{2.19}$$

The calculated BHP for the ferry is $950kW/0.736 = 1290\mathrm{BHP}$, and with an assumption that the thruster have a kort-nozzle. This gives the result in Equation 2.20.

$$\frac{1290 * 0.9 * 1.4}{100} = 16.254\mathrm{MT} \tag{2.20}$$

This result then needs to be converted into kilo newtons for simulation purposes. $16.254 * 9.80665 = 159.4kN$

2.4 Control System

2.4.1 Rotation Matrix in Yaw

The PID-controller and observer is operating in the horizontal plane, and therefor it is needed to use the rotation matrix in the horizontal plane. This yields the rotation matrix in yaw (Equation 2.21).

$$R(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

2.4.2 PID Controller

The PID controller is a regular PID, with proportional, derivative and integral components. The proportional component have the responsibility to make sure that the output changes proportional with the error and determines the ratio of output response. The integral component sums the error over time, and is used to eliminate stationary deviation. The stationary deviation is caused by low frequent forces, in this case wind force. The derivative component is proportional to the velocity (v) and counteracts the oscillations in the system, often caused by the integral component.

The commanded force is calculated using Equation 2.22, where τ_{PID}(Equation 2.23) is used for scaling the gains through the transposed rotation matrix in yaw (Equation 2.21).

$$\tau = R(\psi)^T \tau_{\text{PID}} \tag{2.22}$$

$$\tau_{\text{PID}} = -K_p\tilde{\eta} - R(\psi)K_d v - K_i \int_0^t \tilde{\eta}(\tau)d\tau \tag{2.23}$$

2.4.3 Nonlinear Passive Observer

The observer is a nonlinear passive observer, with a possibility of wave filtering created by Fossen in MSS(Section 1.3). It's designed for filtering low frequent noise. The advantage with a nonlinear observer is that there are no needs for linearizing the yaw kinematics and still obtain a global stability result, and it is less time-consuming than a Kalman filter based observer. The drawback with a observer like this and Kalman filter based observers is that there is a large amount of parameters that is needed to be determined through many sampling cycles of the vessel. The NPO guarantees global convergence of all estimation errors to zero. The tuning parameters are also reduced, there is only one set of observer gains to cover the state space model.

With this observer there are two assumptions taken, P1 & P2.
Assumtion P1: w = 0 and v = 0.
Assumtion P2: $R(y_3) = R(\psi)$ implying that $y_3 = \psi + \psi_w \approx \psi$

The model properties of the inertia and damping matrices will be used in the passive part of the observer design: $M = M^T > 0 \qquad \dot{M} = 0, D > 0$
With the application of the assumptions P1 & P2 to the DP model in page 197 in Fossen, Marine Control Systems[7] results in the DP observer model given from the Equations 2.24 to 2.28.

$$\dot{\xi} = A_w \xi \tag{2.24}$$

$$\dot{\eta} = R(y_3)\nu \tag{2.25}$$

$$\dot{b} = T^{-1}b \qquad \text{(alternatively } \dot{b} = 0\text{)} \tag{2.26}$$

$$M\dot{v} = -Dv + R^T(y_3)b + \tau \tag{2.27}$$

$$y = \nu + C_w\xi \tag{2.28}$$

Chapter 3

Sensors

3.1 Sensors

There are a lot of sensors available in the maritime industry. They are made for harsh environments, and need to be fail safe. Many of the existing solutions comes with high accuracy and reliability. So which of them could be implemented in a auto-docking system? This section will contain the basics around the relevant sensors.

3.1.1 Laser Sensor

Laser sensors can be used for distance measurement and sends out a laser beam to measure the distance to an object. The most used method for laser rangefinders is the time of flight principle. This principle is based on sending a laser pulse in a narrow beam towards the object and measure the time it takes for the pulse to be reflected off the targeted object and return to the sender [22]. The maximum range of laser sensors varies. There are military issued sensors that can measure with a range up to 10 km, and there are cheaper alternatives with maximum ranges around 1000m.

3.1.2 LiDAR

LiDAR is a type of laser sensor, its high power and highly directional rays makes laser systems relevant for data collection, such as distance measurement. Common applications are distance measurement to point (land survey) and environmental measurements from satellite. For autonomous vessels one may use LiDAR for obstacle detection and avoidance to navigate safely through environments, using rotating laser beams.

LiDAR compiles all the results and measurements into a point cloud which works like a 3-D map of the environment in real-time. The map spots objects from distances up to 100m [24] depending on the model, and can identify what kind of object it is. Once an object is identified the system can determine what actions to perform.[12]

"LiDAR is also being tested on vessels for detection and classification of oil. By analyzing the spectrum of the reflected light pulse, one could determine with a high degree of accuracy, the type of oil in the sea". [15]

3.1.3 Wifi Positioning

In areas with a lot of mountains or other objects obscuring signals between a vessel and satellites, an alternative could be to use a wifi positioning system. There are several methods which can be applied to use wifi as a positioning system, some more accurate than others.

RSSI and Lateration

Received signal strength indication (RSSI) is a technique which measures the signal strength between a client device and several access points. By combining this information with a propagation model, it is possible to find the distance between the client device and the APs (access points). Trilateration techniques can then be applied to calculate the position of the client device relative to the APs. This is a fairly cheap and easy method to implement. However it is not the most accurate method, with a mean value of 2-4m. [26]

Angle of Arrival

Angle of arrival (AoA) uses several antennas (2-8 or more, a higher number of antennas gives better accuracy) placed equally far apart in a column to calculate the angle in which a signal arrives. With the advent of MIMO[1] wifi interface, which uses multiple antennas, one can estimate the AoA of the multipath signals received at the antenna arrays in the APs. Triangulation can then be applied to calculate the location of a client device. SpotFi is one method which utilize this technique. One way to compute the AoA is with the use of the MUSIC[13] algorithm. [20]

SpotFi

SpotFi is accurate with a mean value of 40cm and is supposed to be used indoors where GPS signal could be weak or non existing. This is quite accurate considering it only needs three antennas and is made to "work with everything" since most existing APs comes with three antennas already. Because of this, SpotFi can be used on almost every existing APs, which makes it quite suitable for tracking mobile phones, and other everyday devices. [16]

ArrayTrack

ArrayTrack works similar to SpotFi, but with a higher number of antennas. From each antenna the AoA is being calculated to determine the most direct path to the client. With triangulation a position of the client device can be determined with about 30cm accuracy. [25]

3.1.4 Global Navigation Satellite Systems (GNSS)

Satellite systems uses satellites with radio signals to triangulate the position of receivers. The position is given in longitude, altitude and elevation to pinpoint a receivers position with an accuracy from a couple of meters to a couple of centimeters or even less depending on the system. Two GNSS which is operational today, is the American GPS and the Russian GLONASS (Global Orbiting Navigation Satellite System). In 1999 the ESA (European Space Agency) decided to develop their own system, the Galileo. It was scheduled to be globally operational within 2008, but due to financial problems it got delayed. It is now (2018) operational, and is supposed to be fully operational on a global scale with 30 satellites within 2020.

Since a receiver can know when a signal was sent and when it was received, the time difference can be calculated by subtracting the time it was sent from the receiving time, leaving the travel time. It is also known that radio waves travel at the speed of light in vacuum, and close to the speed of light with air resistance. By multiplying the traveling speed with the speed of light the distance between the receiver and a satellite can be calculated. Since the time a signal was sent and received is important for these calculations, it is important to have precise clocks which is why atomic clocks is used in satellites. Most receivers does not have an atomic clock however, and the quality of a receiver's clock could have an impact on its ability to precisely calculate its position over time, as the clock might go too slow or too fast. When the distance from the receiver to four satellites is known, the receiver can use trilateration to calculate its position. Below is a representation of what is known depending on how many satellites are in sight.[11][23]

- 1 satellite , the receiver's location is known within a sphere.
- 2 satellites, the receiver's location is known within 3D ring
- 3 satellites, the receiver's location is somewhere on at most two 3D regions
- 4 satellites , the region gets smaller because of the sphere of the new satellite

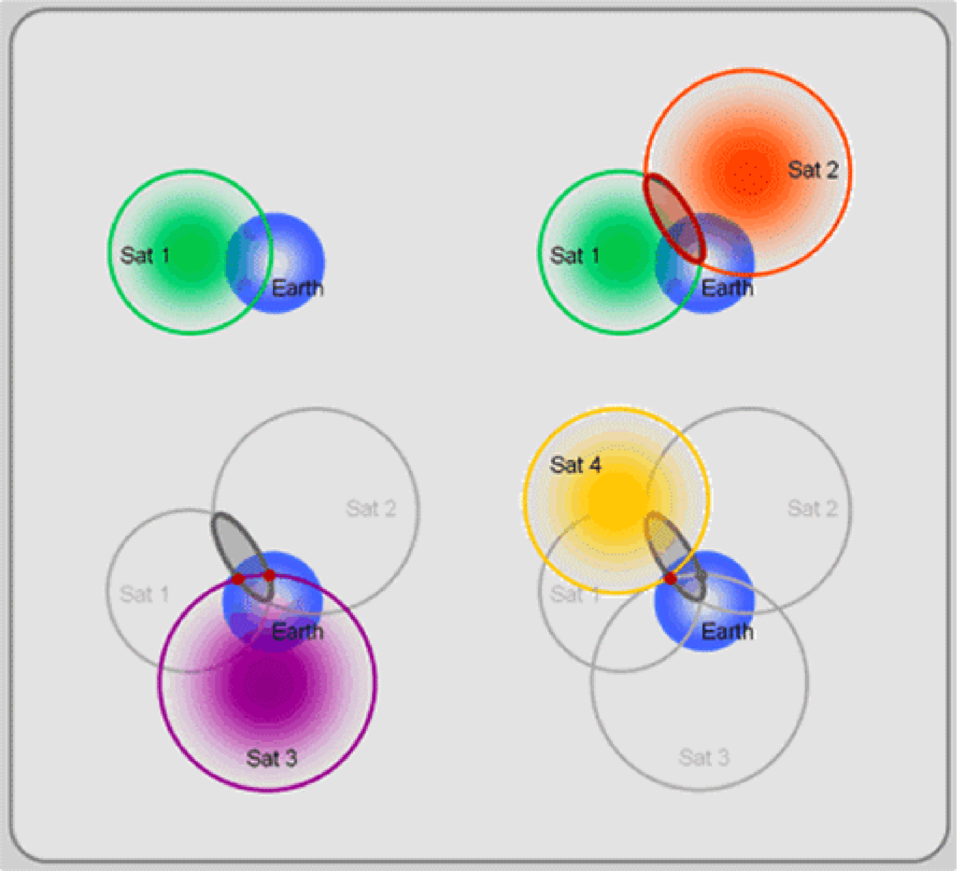


Figure 3.1: A rough representation of how GNSS uses four satellites.[19]

While GPS, GLONASS and soon Galileo themselves have a pretty good coverage, some receivers can use more than one of these systems to get even better redundancy. This can be helpful in areas where a receiver might not be able to get a clear view to four satellites of one system. Then it might be able to see two or three satellites from the GPS system, and the fourth one can be a GLONASS satellite. GNSS is a precise and reliable system for navigation, and it is also very robust in weather. GNSS does not show any visible effect by weather, and is built to serve as an all-weather, 24/7 navigation tool[10]. Unless you are going for some high precision navigation close to either of the earths poles, you should experience close to zero interference because of the weather.

RTK

While GNSS can give a pretty accurate position, the Real Time Kinematic (RTK) system can improve the accuracy to a few cm, or even mm precision. This method utilizes a base station which has a locked position. With the use of UHF (Ultra High Frequency) or VHF (Very High Frequency) radio signals the base station can assist a mobile receiver to reduce the margin of error which can occur when using only GNSS. This is done with the use of a correctional signal sent from the base station to the mobile receiver. If the position of the base station is known and the relative position can be found, the position of the mobile receiver can be found with centimeters precision.

The distance between the receiver can be increased by using a relay link. Normally, differential phase calculations can be made 20 km from the base station, but experimental tests have also shown that the method can be used over significantly longer distances.[15]

Differential GPS

By placing a GPS receiver in a well defined known position on land, usually a trigonometric point, it is possible to compare the deviation with the observed and the calculated distance to the satellite. This receiver is the main unit of the reference station. The reference station can further transfer the deviation directly through radio link or to a database for post processing. [15]

The receiver at the reference station must be able to track all satellites above the horizon and calculate deviations in the distance to all satellites. At the same time, the receiver can calculate how quickly the deviation changes. If the deviation and the distance rate are transmitted via radio link, the mobile receiver(on the vessel) must have a receiver that can decode the information and automatically correct the distance measured on the mobile GPS receiver, as well as predict the position until the next message is received.

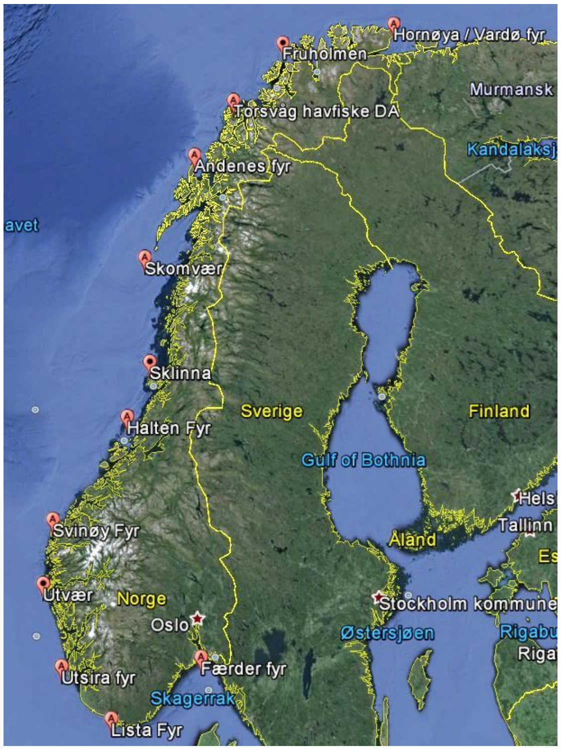


Figure 3.2: Location of Norwegian maritime DGPS stations. [3]

The closest reference station at this moment is on Svinøy, approximately 47 km from Sulesund (Figure 3.3). There are a total of 12 reference stations located along the coast of Norway, and the Norwegian Coastal Administration is responsible for operating and monitoring the system. As the reliability of the system was significantly reduced over the 2000s, it was decided to renew the entire Norwegian system. The upgrade was completed in 2012. [3]

The main factors limiting the range of a DGPS system is the signal strength and signal-to-noise ratio of the user. Too low signal strength and/or low signal noise conditions will result in loss of broadcasted data. In case of loss of too many subsequent messages, DGPS corrections will gradually become older, resulting in gradual deterioration of position accuracy. The receiver will eventually switch to "GPS mode only" and ignore DGPS corrections.

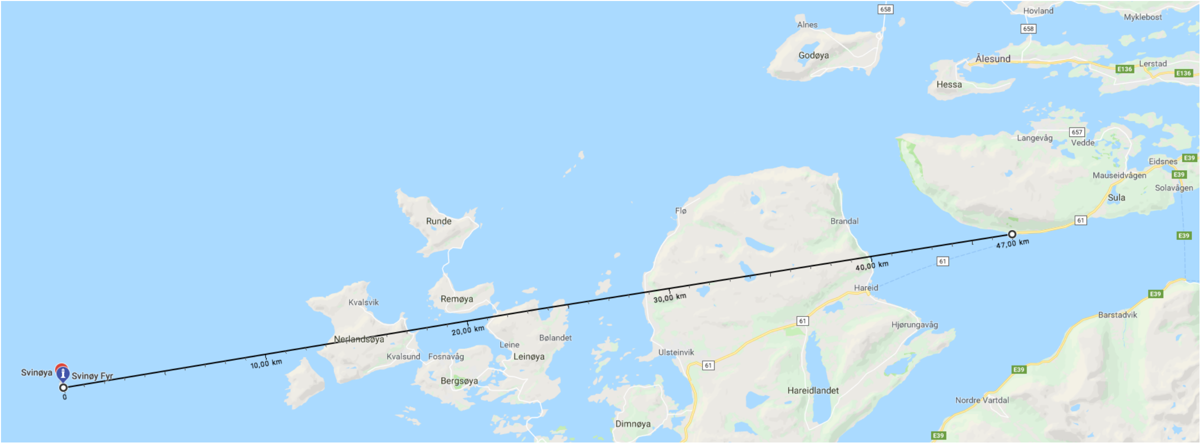


Figure 3.3: Distance from Svinøy to Sulesund.

3.1.5 RADius

In 2004 Kongsberg Seatex released a system called RADius. The system is based on measurement of bearing and distance to passive transponders mounted in known positions [17]. Bearing from RADius is relative to the vessel's heading. RADius is a robust system that operates in a 5.5 - 5.6 GHz frequency band, the system operates well in all weather and extreme cold conditions down to -40°C [18]. The system is designed to be used in distances up to 1100m, with accuracy as shown in Table 3.1.

Table 3.1: Accuracy and Distance [15]

Distance	Accuracy
500-1100m	A few meters
200-600m	1m
0-200m	<1m

A typical RADius system consists of an interrogator, one to four transponders, a computer and a display. Having more transponders available increases the accuracy of the positioning. To increase the working angle of the system, more interrogators can be added. The interrogator sends a signal to the transponder, which reflects it, so that the interrogator can calculate a range and a relative bearing.

RADius has a vertical and horizontal opening angle of 90°. This secures stable handling in close proximity operations where the difference in mounted transponder and interrogator can be considerable. Transponders are mounted on installations that a vessel with an interrogator approaches. Transponders are either battery operated (more than 1 year operation) or connected to a power source from the installation or vessel. Several interrogators can be easily deployed on suitable places on the outside of the vessel. Each interrogator contains antenna elements, a receiver, a transmitter and a signal processing front end [18].

3.1.6 Camera

It is reasonable to think that image processing can be used to determine when the vessel is docked. Most ships have cameras around the vessel so the crew can observe what is happening on the outside of the vessel at all times. If images provided by these cameras are good enough for image processing to be used, it could be a cheap solution. However, it could be that better cameras or different camera types than the ones in use are better. Tidefjord provided a few images from their cameras. These images proved to be not too useful, it was therefore hard to give any good examples. It is however imagined that given proper images, one can determine the distance between the vessel and the quay, or at least determine if the two are touching or not. This might have been easier if the camera was placed looking down upon the front of the vessel and the quay.

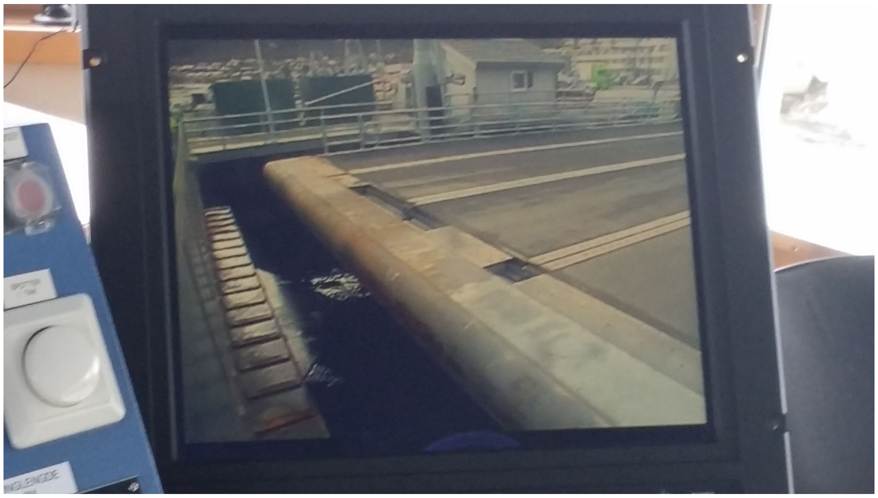


Figure 3.4: Tidefjord is almost docked to quay.

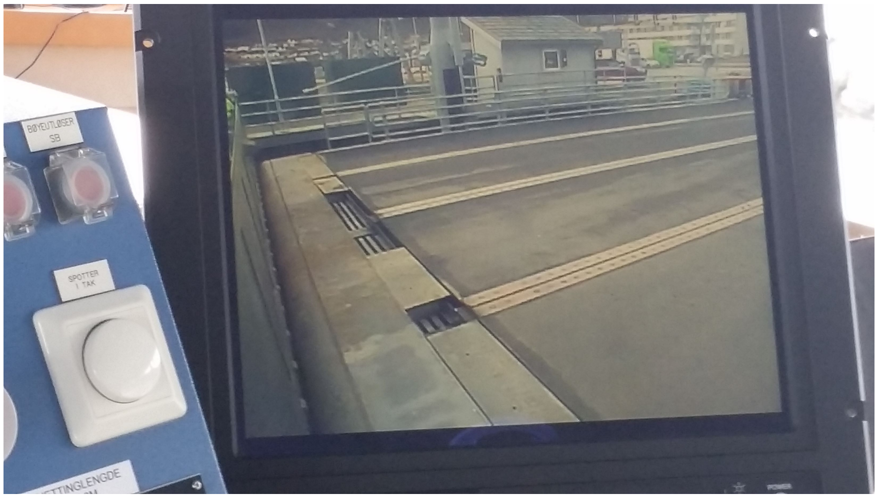


Figure 3.5: Tidefjord is docked to quay.

3.2 Sensor Placement

Sensor placement is crucial for a operational system. The placement of sensors is determined by the type of sensors, the range of the sensor and visibility from where the sensor is placed. Some sensors need to be placed on the side of the vessel while others can be placed on the top or even inside.

3.2.1 LiDAR

Since LiDAR utilizes the time of flight principle and therefore needs to be placed on the outside of the vessel to have a clear line of sight. This type of sensor will be used to estimate how close the vessel is to the quay. Placement of the sensor depends on the shape of the quay, in this case there is a pier on one of the sides. So the best placement for LiDAR would be on the front left or right side of the vessel. From that position the sensor can detect the pier and the quay.

3.2.2 WiFi

The WiFi requires two access points, and several routers placed on the quay. How many depends on the size of the quay and the range of each router. It would be preferable to place the routers in locations where they have as few obstacles between them as possible. This means that a few metres above ground could be ideal, if possible, on the quay.

3.2.3 RADius

RADius uses an interrogator placed on the vessel, and one to four transponders that are placed on the quay. The interrogator is typically placed on the top of the vessel, the transponders are placed strategically on the quay. The number of transponders depends on the shape of the quay. In this case two transponder located on the quay, or one transponder in the center of the quay would be ideal to calculate the position of the vessel.

3.2.4 Camera

Cameras need a clear and direct view of what is supposed to be analyzed or measured. If cameras are to be effectively used to determine whether or not a vessel is docked, they need a clear view of the vessel side, the quay and the front of the vessel. Exactly where cameras should be placed may differ form vessel to vessel and what image processing technique is being used.

3.3 Weather Data

Meteorologisk Institutt shared weather data from a weather station located on Vigra, approximately 20km away from the area used as base for testing and simulation. The table received contain data from the past five years (01.01.2013-31.12.2017). These data include the following:

- Wind Direction at 06:00 UTC.
- Wind Direction at 18:00 UTC.
- Strongest windshield.
- Strongest mean wind.
- Precipitation at 07:00 UTC.
- Snow depth at 07:00 UTC.

- Mean temperature.

In this thesis the most interesting data is the strongest mean wind. These data have been visualized in Figure 3.6. The rest of the data can be seen visualized in the Appendix A.4.

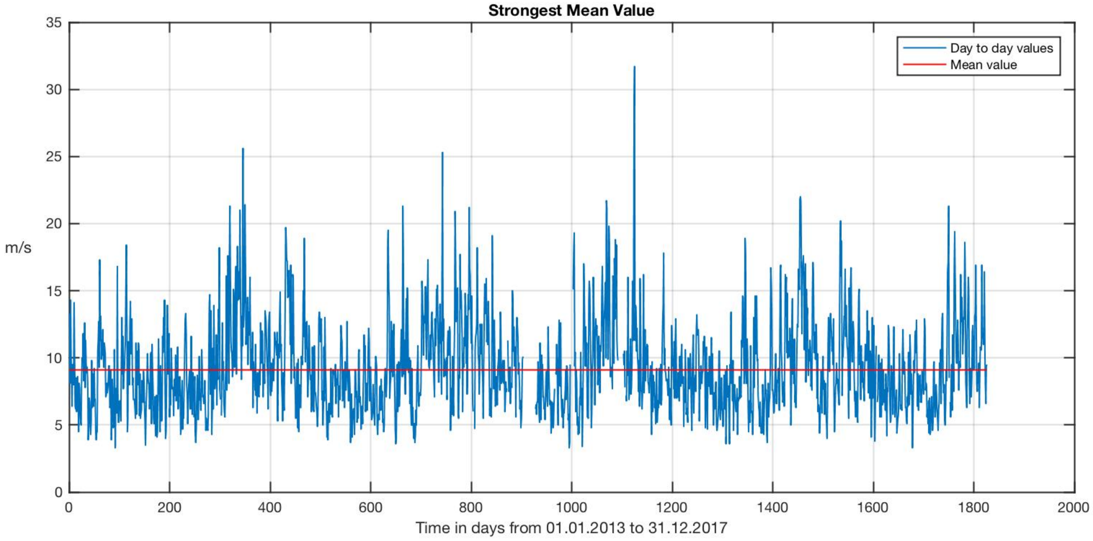


Figure 3.6: Strongest mean value of wind each day.

Chapter 4

Simulation

4.1 Simulation Environment

The entire simulation of this project is based on a dynamic positioning simulation application. It is built in Matlab-Simulink, with the foundation from a MSS DP-application. The vessel is represented as a ferry with two azimut thrusters. For measuring the position there are several options inside the sensor block for choosing which sensors to be active at the given simulation. When it comes to GNSS systems only one can be active at any given time, with the option to choose between Galileo, DGPS & RTK-GPS. In addition to GNSS systems, it is possible to use RADius, LiDAR or both for assisting and improving the measurements.

The sensor block sends the position measurements into the nonlinear passive observer for estimating position and velocity using position measurements and the forces acting on the vessel. These values are then sent to the nonlinear DP-controller for calculating the necessary force and angle the thrusters need to produce to follow the planned route, and eventually reach the final position. When the vessel hits the quay, it will stop by the quay "pushing back". In addition to thruster force acting on the vessel, there is a wind force block for calculating the wind force acting on the vessel from a given angle and wind speed. The motion of the vessel is visualized through the Position-Plot block.

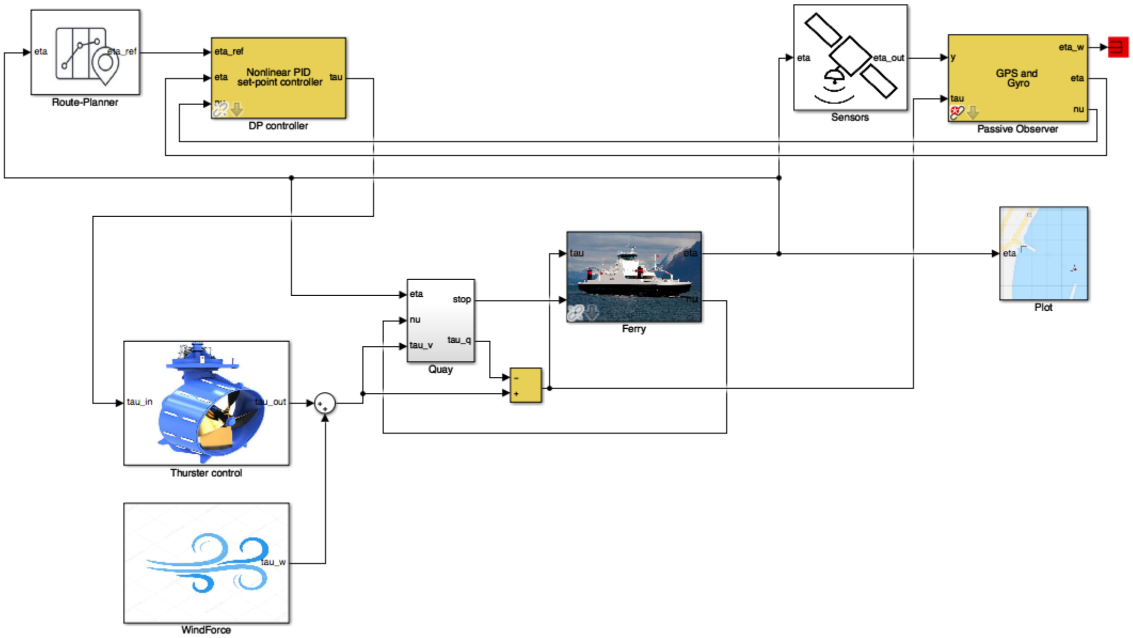


Figure 4.1: Simulation environment overview

4.2 The Vessel/Ferry

The ferry block is originally a supply vessel from the MSS toolbox, but it is modified to fit the specifications of a typical ferry. The way this block works is, it takes the force acting on the vessel and by using the inverse of the system inertia matrix and the linear damping matrix it can calculate the velocity of the vessel. When the velocity is calculated it's possible to calculate the position using a rotation matrix around the yaw position of the vessel.

The vessel block has two adjustable parameters, it is the inverse system inertia matrix (Equation 2.15) and the linear damping matrix (Equation 2.16). These parameters were given by RR Marine with an average ferry in mind. Then the Minv(system inertia matrix) was set to (Equation4.1) and D(linear damping matrix) was set to (Equation 4.2).

$$M = \begin{bmatrix} 136991250 & 0 & 0 \\ 0 & 136991250 & 0 \\ 0 & 0 & 7.5 * 10^8 \end{bmatrix}^{-1} \tag{4.1}$$

$$D = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 15221.25 & 0 \\ 0 & 0 & 21794062.5 \end{bmatrix} \tag{4.2}$$

4.3 Control System

The control system of this application is based on a nonlinear PID controller, a nonlinear passive observer and a thruster control/allocation algorithm. The PID controller and NPO origins from the MSS library, and is tuned to fit the given system and vessel.

4.3.1 DP-Controller

The DP controller has three inputs: eta_ref, eta and nu. The eta_ref is the reference position, this is used to set the final destination for the vessel. The vessel will start from its initial position, and move towards the given reference position. The initial position of the vessel can be set inside the ferry-blocks eta integrator. Eta is the given position of the vessel. This combined with the eta_ref is the regulation error of the controller. The controller uses the transposed rotation matrix in yaw for scaling the surge, sway and yaw gains as a function of the yaw angle. The output of this controller is a three dimensional τ-vector in surge, sway and yaw. This computes the necessary commanded force that the thrusters need to produce in order for the vessel to move to its reference position.

Tuning of the controller

This PID-controller needs to be tuned individually in all three DOF's. This is presented in Table 4.1.

Table 4.1: PID-controller gains

DOF	Kp	Ki	Kd
Surge	10^5	$10^{5.3}$	10^8
Sway	10^7	$10^{7.3}$	$10^{9.3}$
Yaw	10	11	10^4

4.3.2 Nonlinear Passive Observer

The Nonlinear passive observer (Passive DP wave filter) is used to estimate position and velocity based on position readings from the sensors and the force acting on the vessel. It uses the transposed rotation matrix in yaw to give a estimation of position and velocity through the observer gains. Since the observer needs a model of the vessel, the system inertia matrix(Equation 4.1) and linear damping matrix(Equation 4.2) are some of the possible tuning parameters for the observer. There is an option for the observer to compensate for low frequent wave motion. In this project wave motion is considered neglected, so the wave filter is disabled by setting the lambda value to "eye(3)" in Simulink.

4.3.3 Thruster Control

The thruster control block have three main parts for calculating force, and in which direction the force will affect the vessel. The first part is the thruster allocation block, it breaks down the commanding τ-vector from the PID-controller in thrust force and the angle for each thruster. These values are then sent to the "2 azimut thrusters" block which holds the thrust configuration matrix(Equation 2.18) and the method for calculating produced thrust (Equation 2.17). The thrust configuration matrix is modified so the thrusters are located 45 meters in each direction from the COG of the vessel. There is also a 3 second low-pass filter delay for producing thrust. Since the calculated maximum bollard pull is 159 kN(Equation 2.20), the signal out from the thruster allocation block is limited to 159 kN maximum value and -79 kN negative limit (50% negative pitch).

4.4 Sensor Block

The sensor block contains the LiDAR, RADius and the three GNSS. In this block one can easily control what kind of sensors and GNSS systems will be used in each simulation with the use of manual switches. The sensor fusion is done by taking the mean value of two or three position measurements, and combining them. The sensors output is weighted 50% each between the two sensors used, except when using LiDAR and RADius together. In that case the LiDAR and RADius is weighted 25% each and the GNSS system is weighted by 50%.

4.4.1 LiDAR

To simulate how a LiDAR measures the distance from the vessel to the quay, three points were chosen on the quay to help act as reference points. Using a line in Matlab called "linespace" allowed for several points to be made between these three points. These points will represent the physical quay, as seen in Figure 4.2. The position of the vessel is given to the block as input parameters. In the block, the distance between the ships coordinates and the closest point on the quay is calculated using Pythagoras theorem as seen in Equation 4.3 on all points using a for-loop, then choosing the closest one. pts1 is a matrix holding all the points in one of the lines. The block also plots a line between the ships coordinates and the points on the quay calculated to be the closest point at any given time in the simulation, as long as the vessel is within a given distance.

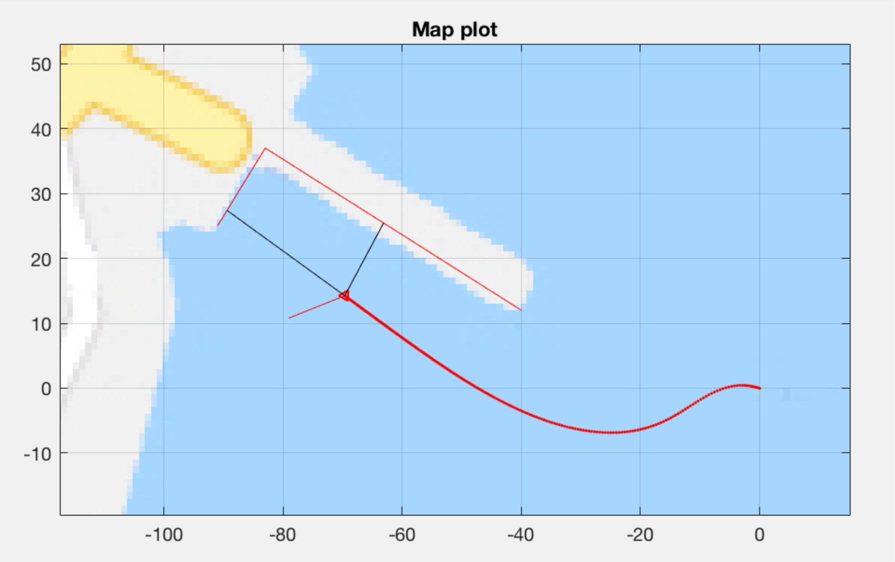


Figure 4.2: The simulation with the LiDAR lines visualized.

As seen in Figure 4.2, the red lines on the quay is where the points simulating the quay is. The black lines between the vessel and the quay is the closest points at that time in the simulation.

$$\text{distance} = \sqrt{(\text{pts1}(i,1) - x)^2 + (\text{pts1}(i,2) - y)^2} \tag{4.3}$$

In order to find the estimated position of the vessel, the distance between the vessel and the closest point on the quay is found, as well as the angle in radians for this line relative to the quay. These are then converted to Cartesian coordinates, and added to the coordinates of said point. This gives the estimated x and y coordinates of the vessel. More details about this can be read in the Appendix A.2.8.

LiDAR has an approximate error of 0-2cm. Therefore a white-noise block was inserted, with an integrator limiting the output to ±0.01 which is equivalent to a possible error up to 2cm.

4.4.2 RADius

RADius was simulated by creating a point on the vessel and the quay representing the transponder and interrogator respectively. With RADius, one can measure bearing and the distance between an interrogator and a transponder. In order to do this in Matlab a third point is needed. A point was therefore created which is always ahead of the vessel. By using the length between these three points as seen in Equation 4.4 the angle can be found. ’p1’ is the radius on the quay, ’p2’ is the vessel and ’p3’ is the point ahead of the vessel.

$$\text{angle} = \cos^{-1}((p21^2 + p23^2 - p13^2)/(2 * p21 * p23)) \tag{4.4}$$

’p21’ is the distance found using Pythagoras theorem between ’p1’ and ’p2’ etc. The RADius system has a 90 degree view angle for both interrogators and transponders, which is represented with black lines on the plot for both. To calculate the estimated x and y coordinates of the vessel, a point was made on the same y-coordinate as the transponder and a different x-coordinate. By doing this, the ships coordinates was estimated using the measured distance and angle between the transponder and interrogator. Read more about this in the Appendix A.2.9.

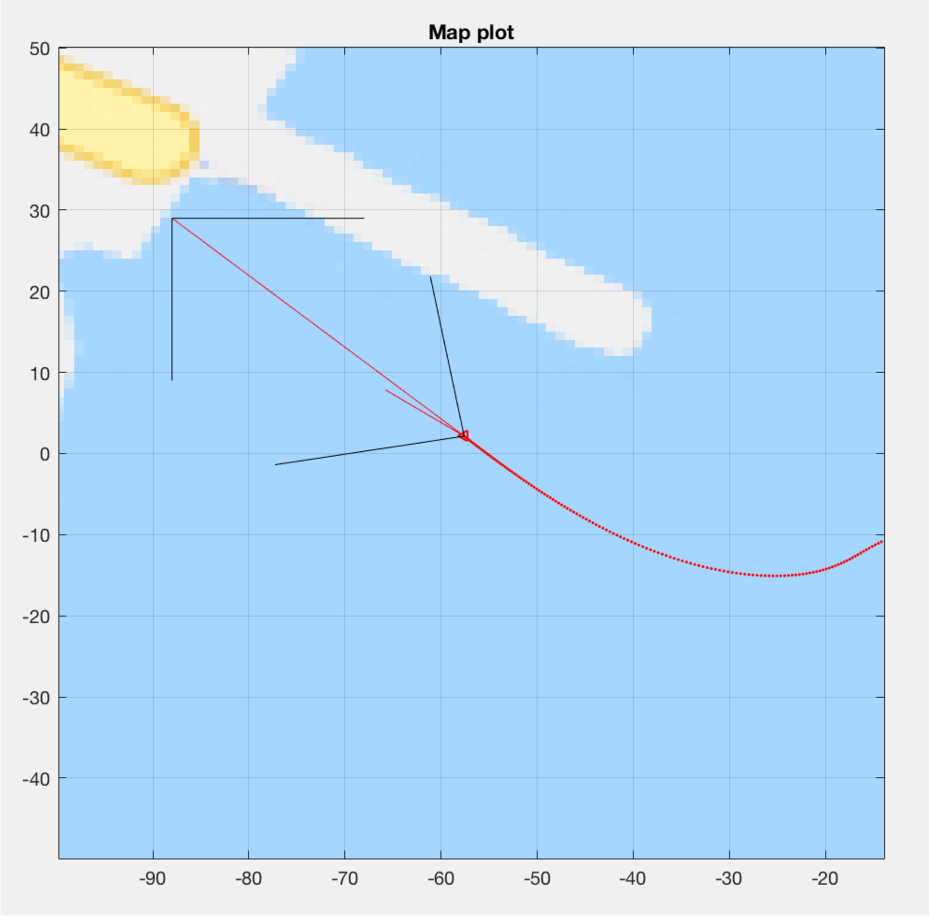


Figure 4.3: The simulation with the RADius lines visualized.

RADius has a possible error of 1m or less, depending on the distance see Table 3.1. It has been decided to use a 1m error in the simulations. As with LiDAR, a white-noise block and an integrator is being used where the integrator block is limited to ±0.5 which is equivalent with 0-1m error.

4.4.3 GNSS

To simulate GNSS systems, three blocks have been made. One block, simulating the Galileo system alone since the area used for simulations are located in Europe, one block simulating the use of RTK and one for the DGPS system. They take in the actual XY position of the vessel, and add noise depending on what kind of system is being used since each of the systems have different accuracy. The white noise is integrated with a limited output. The limited output will vary from system to system to limit the noise within the range of error in measurement which can be expected from each system.

For the GNSS system, an accuracy of 1m can be expected since the simulation is based on the Galileo system. The output limitation on the integrator is therefore set to ±0.5.

The differential GPS (DGPS) has an accuracy of 0-10cm. The output limitation on the integrator is therefore set to ±0.05.

The RTKgps system has an accuracy of 0-1cm. The output limitations on the integrator is therefore set to ±0.005.

4.5 Simulation of Wind

To simulate wind affecting the vessel a standard wind block was used. This block takes ten parameters in and gives an output vector containing the wind coefficient force for surge, sway and yaw relative to the vessel. The block uses the tables from Table 2.2, 2.3 and 2.4 as well as the equations in Equation 2.14. The data sets of Isherwood (1973)[14] are programmed in the Matlab function windcoeff.m (see Appendix A.2.10) which comes with the GNC toolbox. The inputs and outputs can be seen in Equation 4.5. The formulas and tables in this function has been discussed more in detail in Section 2.2, and the inputs used in the simulation can be found in Section 2.3.

$$\text{[w\_wind, cx, cy, cn]} = \text{windcoeff(gamma\_r, V\_r, L, B, A\_L, A\_T, A\_SS, S, C, M)} \tag{4.5}$$

The cx, cy and cn are optional outputs. The parameter gamma_r should be given in radians, the block will convert it using the following line: $gamma\_r = gamma\_r * 180/\pi$. The variables A0-A6, B0-B6 and C0-C5, which is used in the Equations 2.14, are made in the function using lines as seen in Equation 4.6.

$$\begin{aligned}
&\text{A0} = \text{interp1(CX\_data(:,1), CX\_data(:,2), gamma\_r);}\\
&\text{A1} = \text{interp1(CX\_data(:,1), CX\_data(:,3), gamma\_r);}\\
&\text{A2} = \text{interp1(CX\_data(:,1), CX\_data(:,4), gamma\_r);}\\
&\quad \ldots\\
&\text{C3} = \text{interp1(CN\_data(:,1), CN\_data(:,5), gamma\_r);}\\
&\text{C4} = \text{interp1(CN\_data(:,1), CN\_data(:,6), gamma\_r);}\\
&\text{C5} = \text{interp1(CN\_data(:,1), CN\_data(:,7), gamma\_r);}
\end{aligned} \tag{4.6}$$

where CX_data, CY_data and CN_data are the Tables 2.2, 2.3 and 2.4. The wind forces and moment is then calculated as seen in Equation 4.7, where rho_a is the density of air at 20°C and is 1.224.

$$\begin{aligned}
\text{tauX} &= 0.5 * \text{cx * rho\_a * V\_r}^2 * \text{A\_T;}\\
\text{tauY} &= 0.5 * \text{cy*rho\_a*V\_r}^2 * \text{A\_L;}\\
\text{tauN} &= 0.5 * \text{cn*rho\_a*V\_r}^2 * \text{A\_L*L;}
\end{aligned} \tag{4.7}$$

The final output for the function is tau_w = [tauX, tauY, tauN]$'$, and the optional wind coefficients cx, cy and cn.

4.6 Quay Force

Since the goal is to dock the vessel completely, it will eventually hit the quay. Then the quay will produce a force equal to what the vessel produces against it. So the quay force block is eliminating all force affecting the vessel and the velocity when it is inside the limited position where the quay is located. This was needed for stopping the vessel regardless of what happened during the simulation.

4.7 Route-Planner

In order to make a route for the vessel to follow, a function block was made which easily let the route be set and changed. The x- and- y-coordinates as well as the desired heading of the vessel are inserted in three separate vectors. The block is made to be adaptive to the number of desired points in the route. In Figure 4.4 the route that was planned can be seen in Equation 4.8, 4.9 and 4.10. The vessel start at (250, -100).

Since this simulation uses a distance-oriented PID regulator to control how much force is ordered from the thrusters, longer distance will lead to higher speed. In order to get a steady speed, the distance from the vessel and to the desired coordinate must also be steady at all times as the vessel is moving. In order to do this, a point is created which will be placed a given distance from the vessel, in the direction towards the next desired position point in the route, and this made up point will be given as the desired destination. Since the vessel will be moving towards the next desired point in the route, the made up point will move with the vessel, and always have the same distance relative to the vessel. This is called line-of-sight guidance, more on this in Fossen(2002)[7], chapter 5.3.

As the vessel reaches the final destination on the route, the point being made and placed ahead of the vessel will be placed closer and closer to the vessel, slowing it down. When the vessel is close enough, the final point in the matrices will be set as the destination. This will allow the PID to reduce how much force is asked for and thus slowing down the vessel as it reaches its final destination, the quay.

$$\text{routeX} = \begin{bmatrix} 100 & -48 & -87 \end{bmatrix} \tag{4.8}$$

$$\text{routeY} = \begin{bmatrix} -60 & 0 & 27 \end{bmatrix} \tag{4.9}$$

$$\text{heading} = \begin{bmatrix} 150 & 150 & 150 \end{bmatrix} \tag{4.10}$$

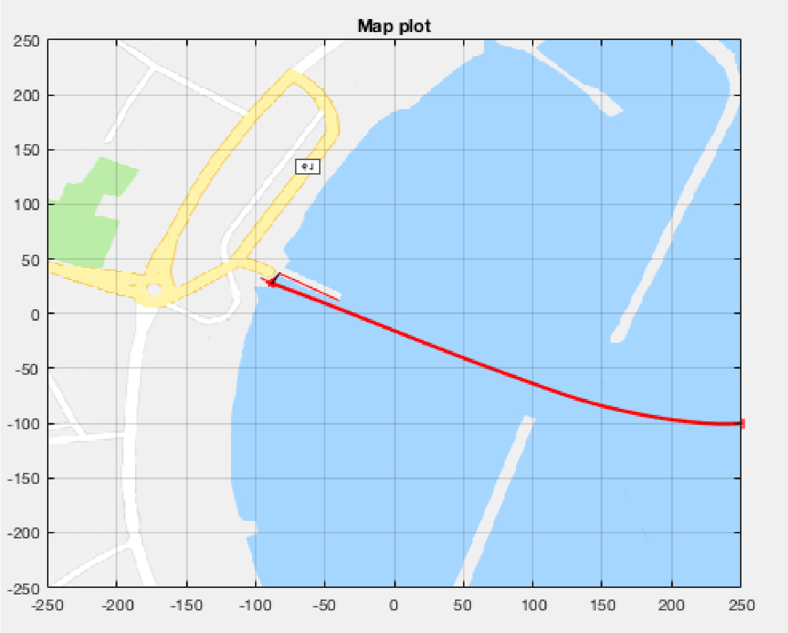


Figure 4.4: The plot with a route to follow

4.8 Plot

In order to have full control of how the plot for the simulation look, a custom plot block was created. As inputs the block gets the x and y coordinates of the vessel, as well as the yaw angle. In order to simulate the environment around the quay, a picture of a map have been chosen and inserted into the plot. The vessel is represented by a triangle, and the heading of the vessel is represented by a short red line pointing from the triangle. The path of the vessel is also mapped with a thin red line as the vessel moves through the plot. In the plot, one coordinate is equivalent to one meter.

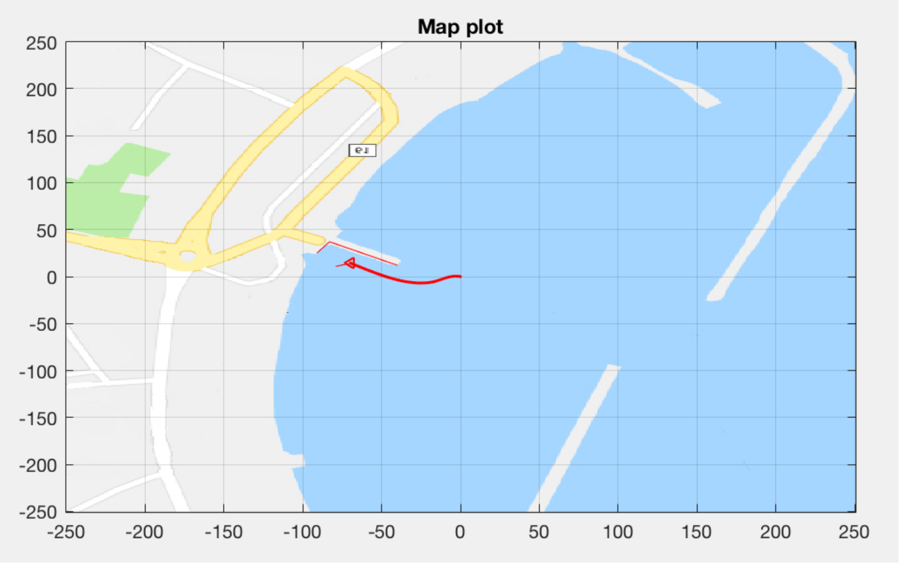


Figure 4.5: The plot used to visualize the simulation of the vessel

Chapter 5

Results

A total of nine cases have been simulated, one scenario for RADius, one for LiDAR and one with both of them. These three scenarios have then been simulated with three different GNSS systems: RTK, DGPS and Galileo. This should give some insight in error of measurement since each of the sensors and GNSS in the simulation have noise added to their signal representing and corresponding to the error of measurement expected from the real sensors and GNSS. The different test cases will be benchmarked by the position error(eta error), this shows the deviation between estimated and actual position.

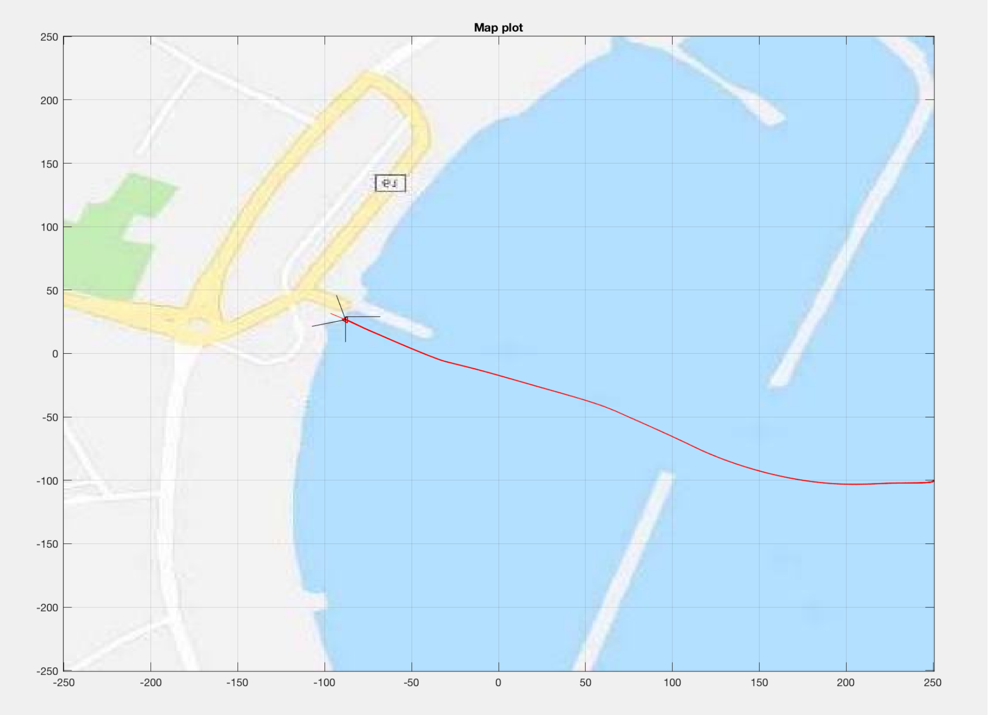


Figure 5.1: The vessels route at $10m/s$

For each case simulated, the low frequent noise is primarily wind. The wind speed is set to $5m/s$ with a direction of $\frac{\pi}{4}$. This angle is equal to south-west wind direction. The system performed adequate with a wind speed up to $10m/s$, see Figure 5.1. At $10m/s$ the system started to become a bit unstable, and the vessel did not follow the planned route as precisely. When the wind speed was set to $15m/s$ the vessel got out of control, see Figure 5.2. The data visualized in Figure 3.6 shows that there were 628 days with a mean wind value greater than $10m/s$, and 1198 days with a value equal or less than $10m/s$.

Ideally the auto-docking system takes control of the vessel from an auto-crossing system. Auto-crossing is used for transit between two ports. The auto-docking system should take control when the vessel is within the pier. In order to reach some velocity before the auto docking process begins, the vessel starts with some distance from the pier. In the simulations the vessel is within the pier at 380 seconds after the simulation starts.

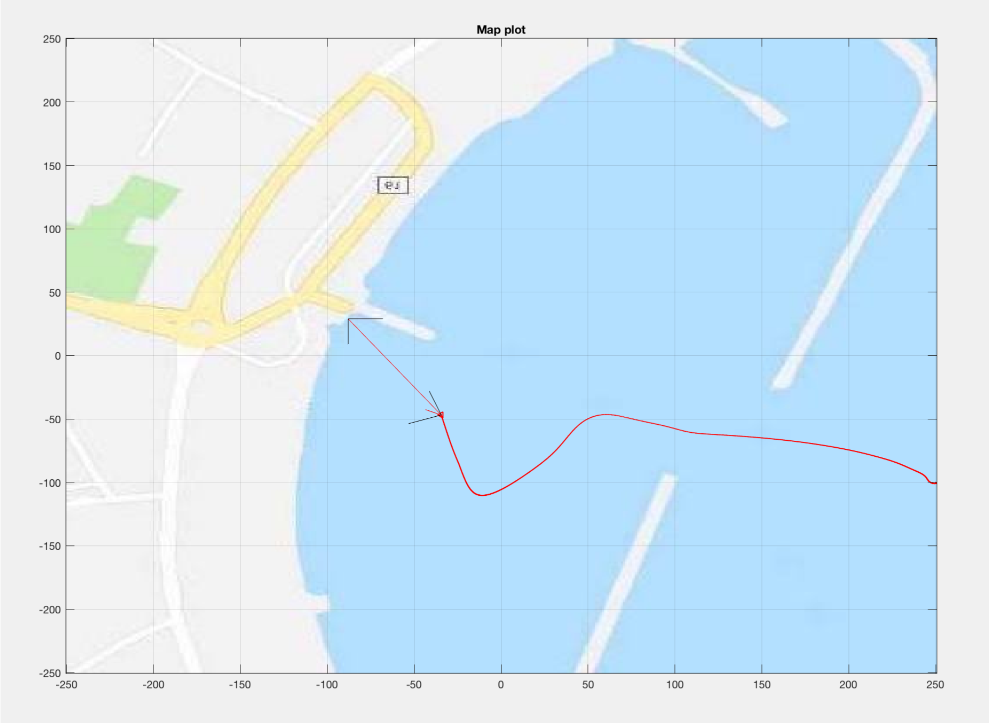


Figure 5.2: The vessels route at 15m/s

During the transit to the quay, the vessel follows the path set by the Route-Planner (section 4.7), this is visualized in Figure 5.3.

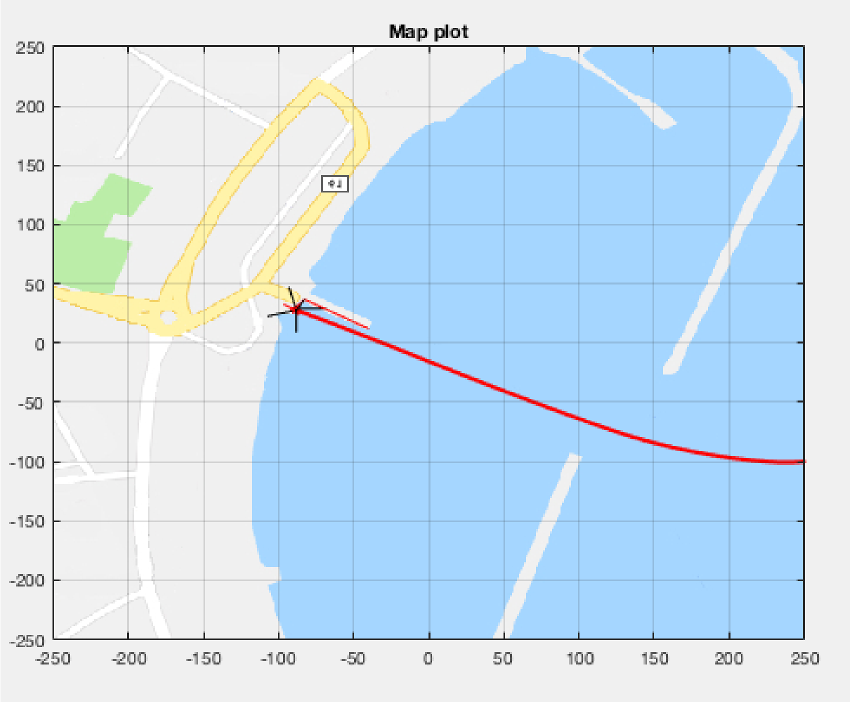


Figure 5.3: The vessels route at $5m/s$

When the simulation is at 380 seconds and the auto-docking system takes over, the vessel is closing in to its highest velocity at almost $0.9m/s$ which is equal to 1.75 knots. The vessel then starts to slow down, reducing the velocity as the vessel gets closer to the quay as seen in Figure 5.4. The vessel hits the quay with a surge speed of $0.02m/s$.

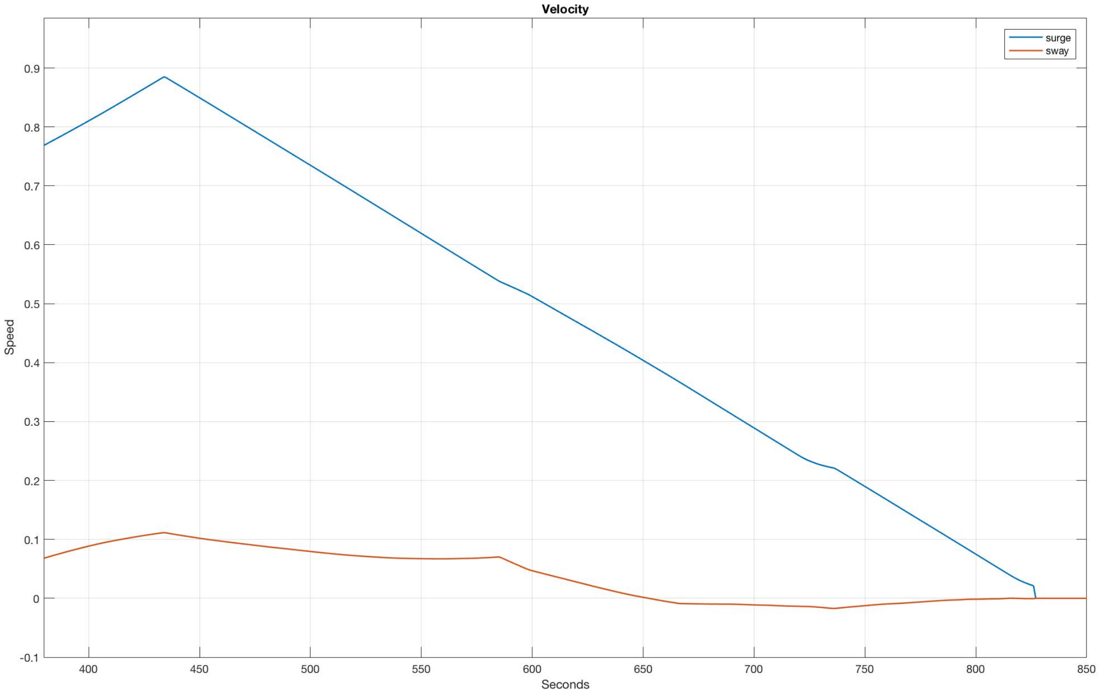


Figure 5.4: The vessels velocity

It is common for ferries to approach a quay diagonally, and not aim straight for it. The reason for this, is that the wind can then be used as an advantage. Either it is used as a "brake" or a "push" into the quay. As seen in Figure 5.4, the sway velocity slowly declines towards zero velocity. This means that the vessel behaves as described.

Figure 5.5 shows the traveled route in X and Y, and the final destination is reached in about 825 seconds into the simulation. This does not change significantly between the scenarios, since the route followed is the same for all of them.

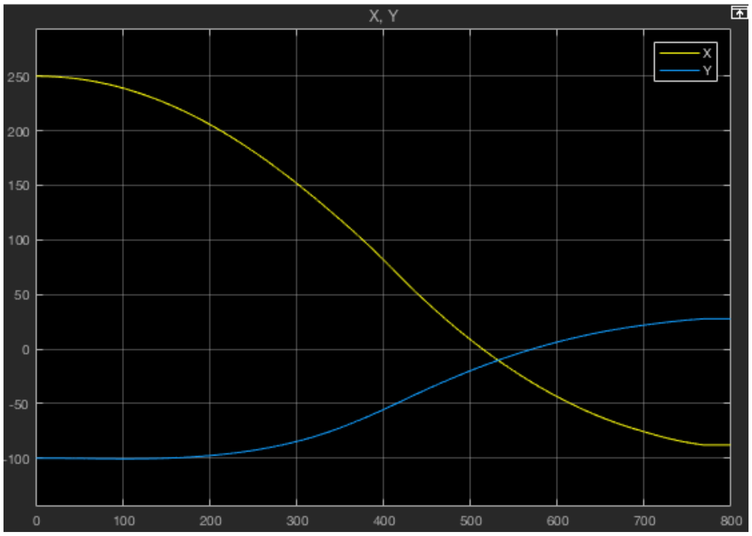


Figure 5.5: XY-Plot for travled route

The GNSS used for the simulations was chosen to show how their accuracy varies. As seen in Subsection 4.4.3, RTK should be the most accurate system because of the centimeter precision and Galileo the least accurate of the three chosen for the simulation. As mentioned in Section 3.1.4, there is a reference station for DGPS only 47km from Sulesund, and therefore no need to set up a station closer to the ferry. The Galileo system was chosen among the GNSS which does not have a reference point on the ground, since the location of simulation is in Europe. Galileo is operational(2018), and as mentioned in Subsection 3.1.4, should be fully operational within 2020.

Further, some results from the individual simulations is presented. The graphs shows the deviation between estimated position and actual position, and the data is gathered from 380 to 850 seconds into the simulations. This is the data which is interesting, from when the vessel is within the piers until right after it has docked to the quay. The minimum, maximum and mean errors have been marked with horizontal lines in each plot. These three values was gathered from the data within the time-frame of interest. Note that a deviation of 0.1 equals to 10cm. In each figure the error in surge and sway is displayed in individual graphs. The yaw error was not included, as there was no noticeable error of measurement.

The data gathered varies from negative to positive depending on if the deviation is positive or negative compared to the vessel. Therefore the data was changed to all be positive, to better show the maximum, minimum and mean error. However, in the actual simulations the deviation can vary from ± of the data seen here. The actual positive and negative deviations can be seen in the Appendix A.3.

5.1 RADius

RADius is originally designed for harsh and demanding environment where GNSS is degraded and safety is crucial. This makes RADius a great solution for any short distance scenarios. When implementing RADius one or more transponders must be placed in strategic locations on the quay, and one interrogator on the vessel. In the simulation a transponder is located in the middle of the quay. Having only one transponder should work for a moving vessel where the surroundings are known. The main advantage with RADius is that it works in almost all weather conditions.

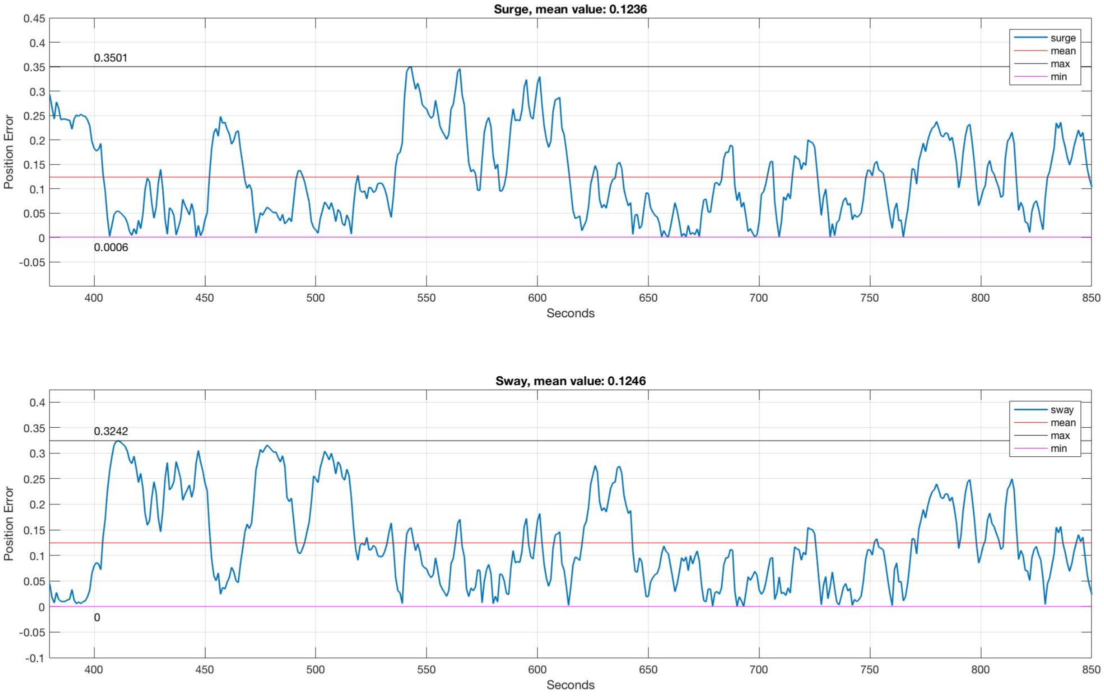


Figure 5.6: Position error. RADius and RTK

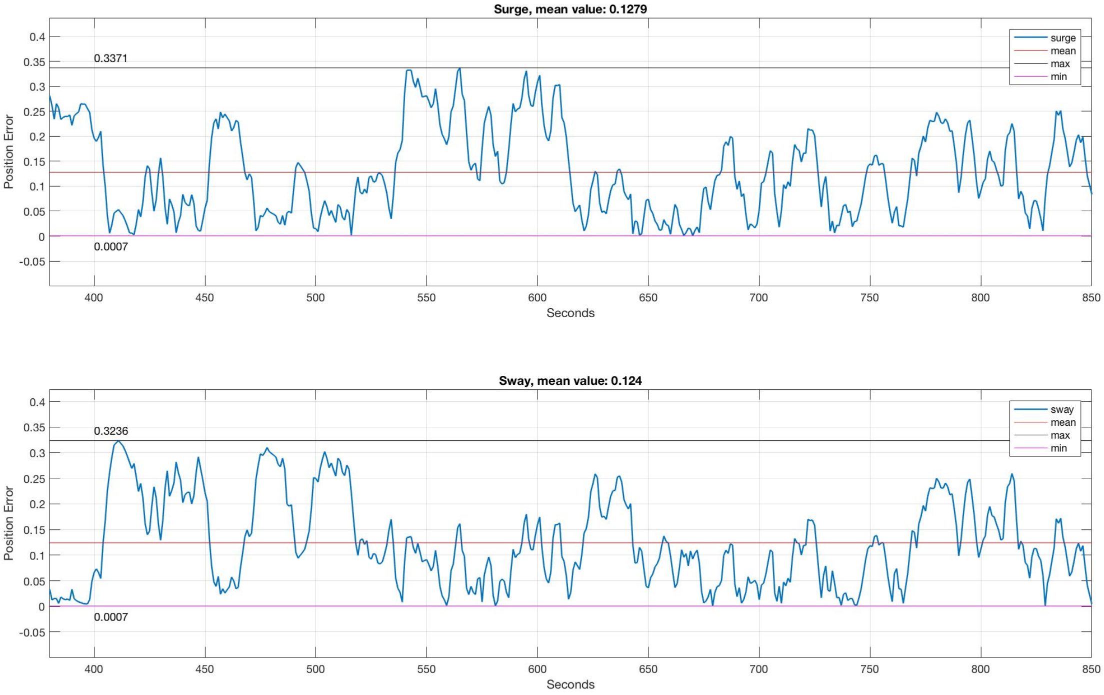


Figure 5.7: Position error. RADius and DGPS

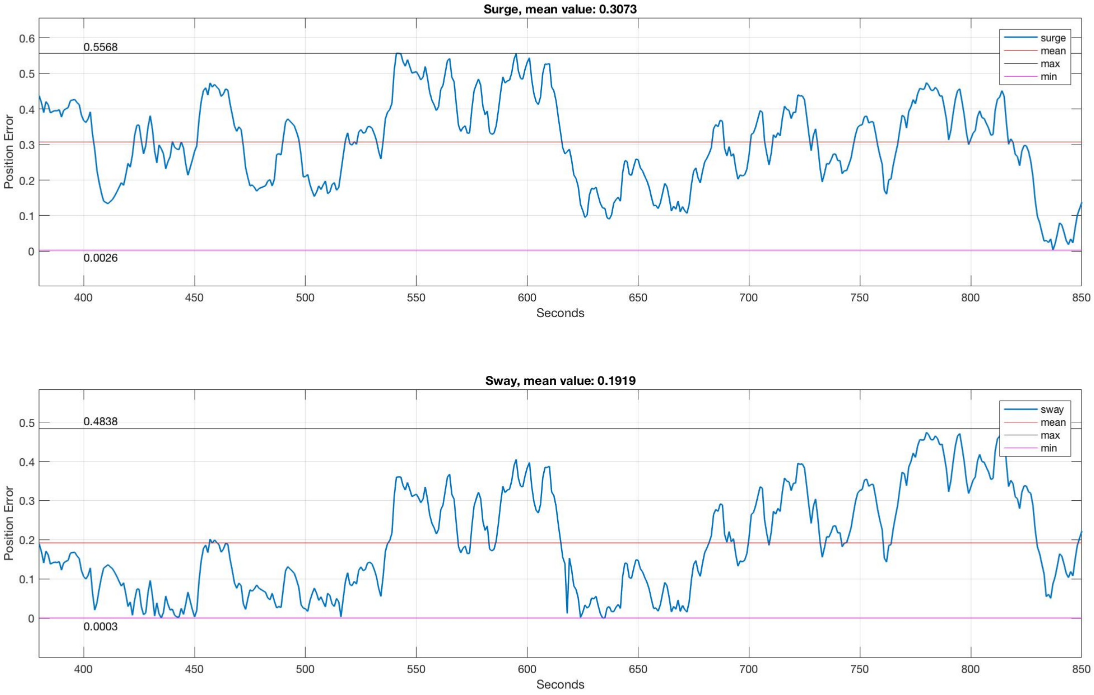


Figure 5.8: Position error. RADius and Galileo

5.2 LiDAR

LiDAR uses light in the form of a pulsed laser to measure distance to objects. LiDAR can be used on a vessel close to shore/quay to determine where potential obstacles are, and where the vessel is in relation to these potential obstacles. LiDAR is very accurate at short distances, and can therefore be very good for measuring distance to quay and obstacles which could enter the vessels path.

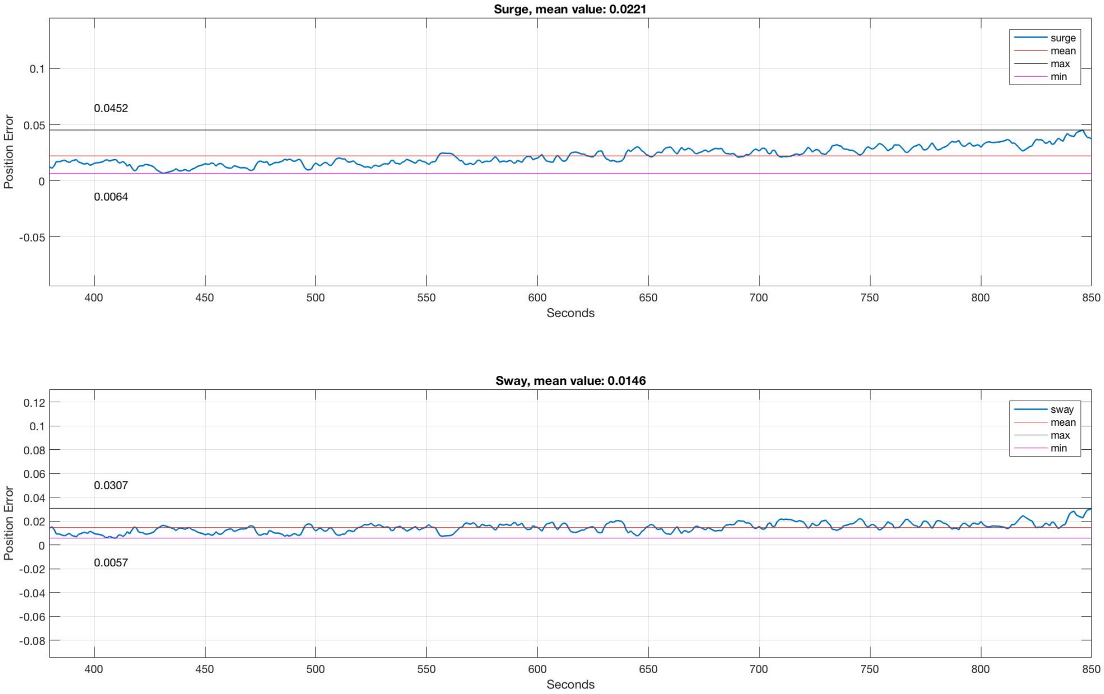


Figure 5.9: Position error. LiDAR and RTK

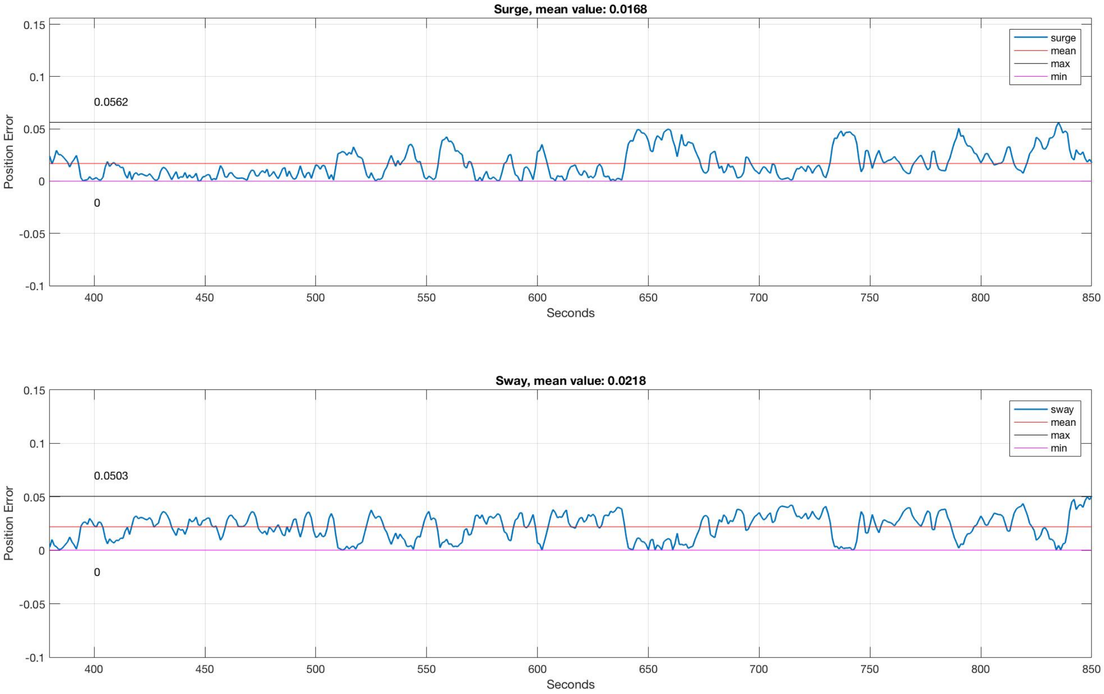


Figure 5.10: Position error. LiDAR and DGPS

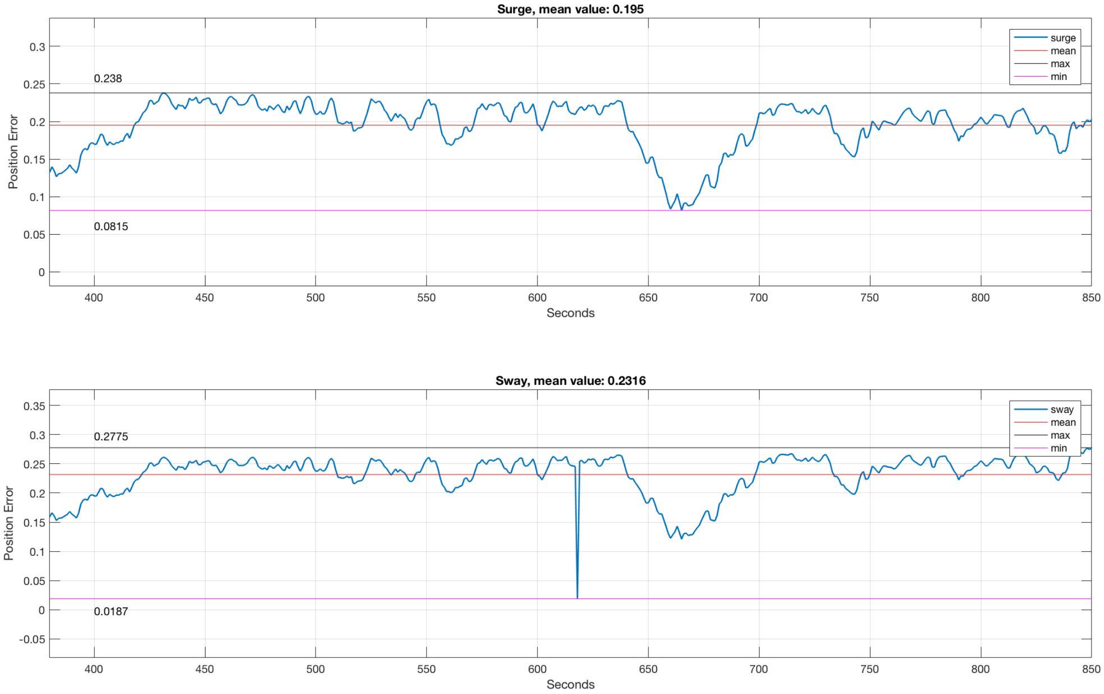


Figure 5.11: Position error. LiDAR and Galileo

5.3 LiDAR & RADius

As mentioned earlier both LiDAR and RADius are systems that works great independently for distance measurement. By combining these two systems the redundancy can be improved. With two independent systems which could work great on their own, the program can with better certainty calculate distance, position and heading.

RADius is based on measurement of bearing and distance to passive transponders mounted in known positions. By measuring both the distance and the bearing which is relative to the vessel´s heading, RADius can control that the vessel is on the right course to the quay and how far from its position the vessel is. When it gets closer, LiDAR with its cm precision can take the vessel safely to quay. It is interesting to see if there is any major difference between using only LiDAR or RADius and using both.

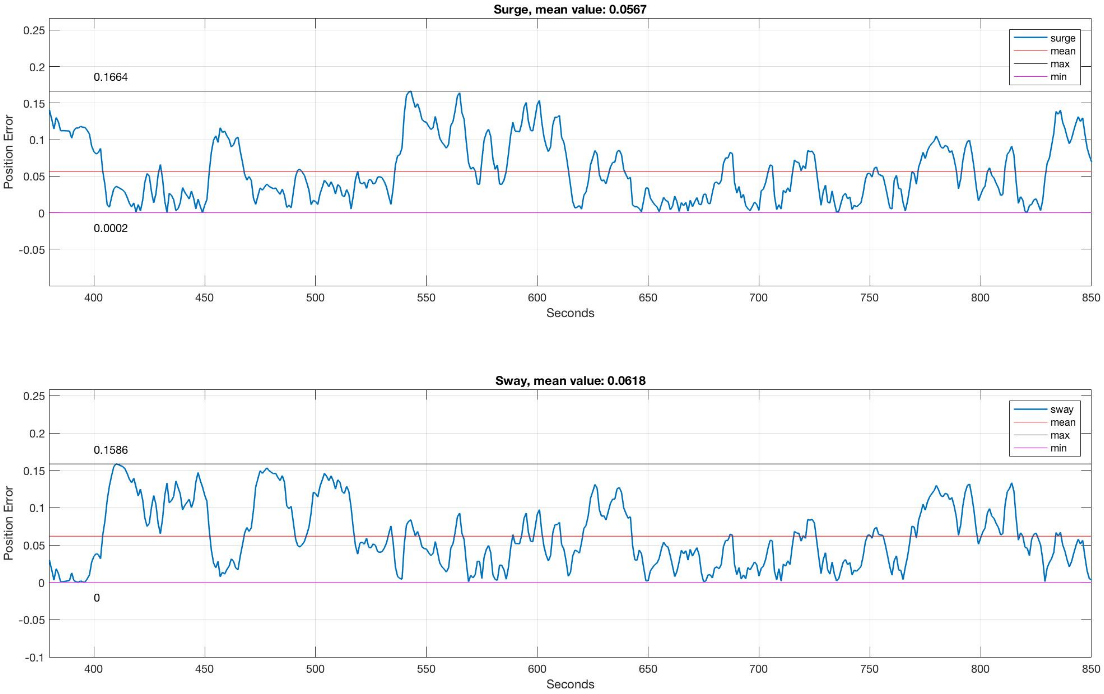


Figure 5.12: Position error. RADius, LiDAR and RTK

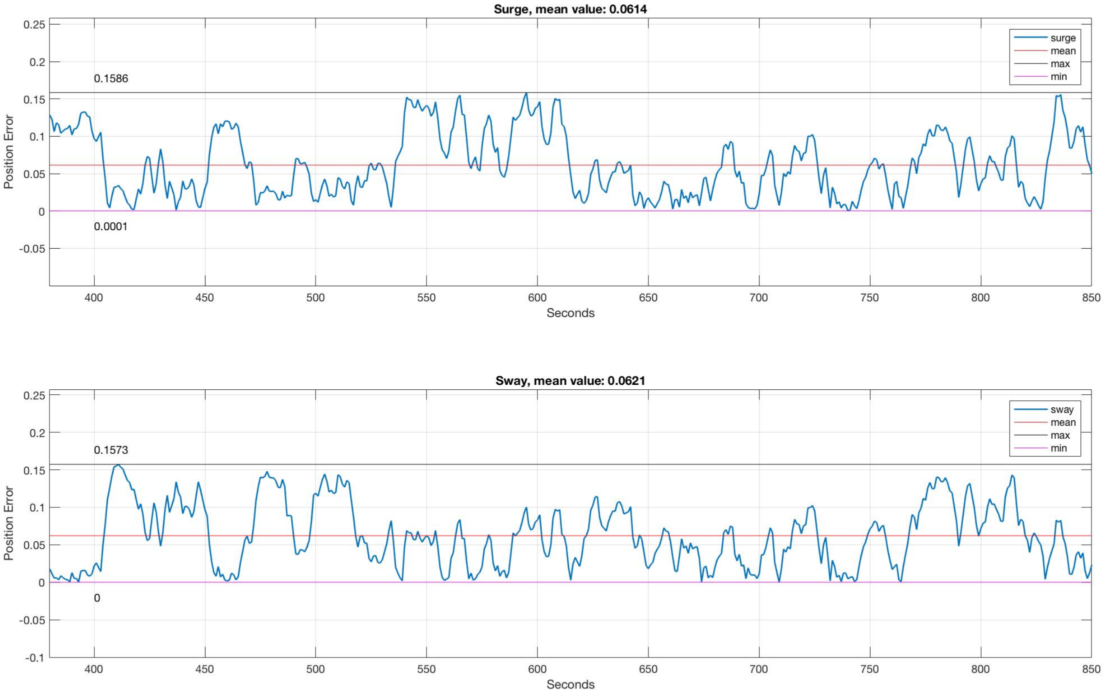


Figure 5.13: Position error. RADius, LiDAR and DGPS

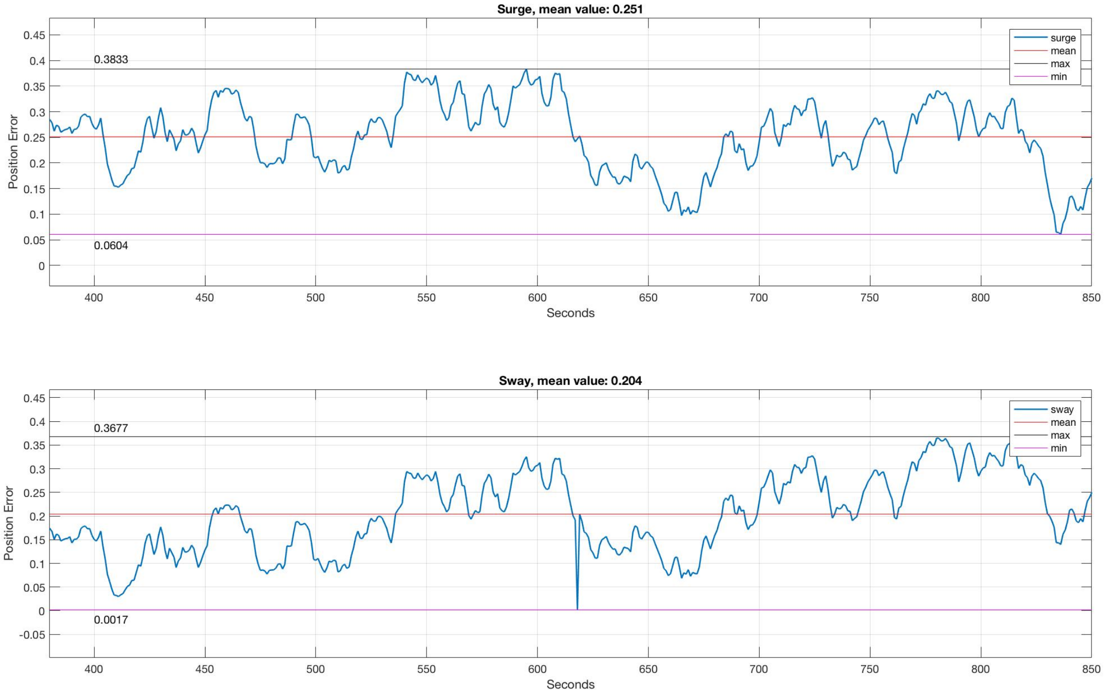


Figure 5.14: Position error. RADius, LiDAR and Galileo

5.4 Conclusions

The different sensor combinations performed quite differently. LiDAR with RTK and DGPS provided the least deviation between estimated and actual position, as seen in Figure 5.9 and 5.10.As seen in Table 5.1 LiDAR with RTK had a maximum error of $\pm 4.52cm$ and $\pm 3.07cm$ in surge and sway, while LiDAR with DGPS shows a maximum error of $\pm 5.62cm$ and $\pm 5.03cm$ in surge and sway, respectively. However, the mean deviation of the two cases is shown to be $\pm 2.21cm$ in surge and $\pm 1.46cm$ in sway for LiDAR with RTK, and $\pm 1.68cm$ and $\pm 2.18cm$ in surge and sway for LiDAR with DGPS. From these data it would seem that LiDAR with RTK provides the least amount of maximum deviation from actual and estimated position, while they both provide similar mean deviations.

Table 5.1: Deviations in estimated position from the simulations in cm.

	Surge		Sway	
GNSS	Max	Mean	Max	Mean
RADius				
RTK	35.01	12.36	32.42	12.46
Galileo	55.68	30.73	48.38	19.19
DGPS	33.71	12.79	32.36	12.40
LiDAR				
RTK	4.52	2.21	3.07	1.46
Galileo	23.80	19.50	27.75	23.16
DGPS	5.62	1.68	5.03	2.18
LiDAR & RADius				
RTK	16.64	5.67	15.86	6.18
Galileo	38.33	25.10	36.77	20.40
DGPS	15.86	6.14	15.73	6.21

LiDAR provided good results in two of three scenarios. When LiDAR was combined with Galileo, the maximum deviation was $\pm 23.80cm$ and $\pm 27.75cm$ in surge and sway, with a mean deviation as high as $\pm 19.50cm$ and $\pm 23.16cm$ in surge and sway. From Table 5.1 it would seem that in every case where Galileo is being used, there is a large maximum deviation. The largest of which are in the scenario where RADius is combined with Galileo, with a maximum error of $\pm 55.68cm$ and $\pm 48.38cm$ in surge and sway. The mean deviation in that case is $\pm 30.73cm$ and $\pm 19.19cm$ in surge and sway. However, the scenarios with RADius combined with RTK and DGPS seem to have a maximum deviation at around $\pm 32-35cm$ in both surge and sway and a mean deviation at $\pm 12-13cm$.

When both LiDAR and RADius was used together, the deviations seems to be somewhere in the middle compared to the cases with LiDAR and RADius separately. In these scenarios it still seems to be Galileo who brings the highest deviations. RTK and DGPS both seem to perform about the same in this scenario, with a maximum deviation of $\pm 15.8-16.6cm$ in surge and $\pm 15.7-15.8cm$ in sway. The mean deviation in these scenarios is in the range of $\pm 5.6-6.2cm$ in surge and $\pm 6.1-6.2cm$ in sway.

Chapter 6

Discussion

6.1 Sensors

In this section the pros and cons of each sensor will be discussed to give more insight to how they may be utilized in a working solution.

LiDAR

As mentioned in Subsection 3.1.2, LiDAR is a laser sensor that utilizes the time of flight principle to measure distance to objects. LiDAR requires a free line of sight, so any object that comes in the way of the point of interest will make the signals noisy. There is software for creating different clusters of object so that the system knows how to classify them [27]. By utilizing advanced algorithms these systems can detect different objects with high accuracy.

Laser sensors work great in perfect weather conditions. In heavy rain or snow the sensor will not perform as well as it would in fair weather conditions. A study in Spain have tested how LiDAR performs in rainy conditions and found that "Range measurements appear stable even under important rain affectation. The variations are always lower than 20 cm." [2].

RADius

RADius is designed for harsh environments and can withstand most weather conditions. The systems are commonly used in DP handling of vessels near oil rigs, where the weather can be tough. The downside with RADius compared to LiDAR is the accuracy. While RADius has an accuracy of up to 1m within 200m of the transponder as shown in Table 3.1, systems like LiDAR have an accuracy of 1-2cm within a distance of around 100m [24].

One of the main advantages with RADius is that it calculates the relative bearing, so the vessel can stay on the right course. The interrogator on the vessel can detect the transponder from a range up to 1100m, the accuracy of the system is not as good at this range field but gives a good indication that the vessel is on the right course for the quay. As it gets closer the accuracy will increase as seen in Table 3.1.

Advantages of RADius compared to competing laser systems are: [15]

- There are no moving parts(low maintenance)
- Not as affected by weather conditions
- More transponders and multi-user performance
- Longer range

GNSS

There are several global navigation satellite systems. They wary in accuracy depending on what system is being used and where they are being used. Since these systems provide close to global coverage and a very precise position estimate, it is likely that GNSS will be a big part of any autonomous docking processes. It could be argued that in some areas a GNSS is all that is needed to perform autonomous docking, however in order to heighten the redundancy of the process more sensors should be included. Especially in areas with bad reception, for example areas surrounded by tall mountains, other systems must be in place to support the GNSS.

WiFi

WiFi positioning have been used mostly for indoors positioning, where GNSS signals are weak or non-existing. Outdoor WiFi positioning have not been researched much, but it could provide a cheap solution for companies with not so much to spend on an automated docking process. A number of routers, depending on the size of the quay, could be placed with even distances to provide a somewhat accurate position of the vessel. Indoor accuracy with normal household routers are around 20-40 cm depending on which technique is being used, obstacles and how spread the routers are. It could be interesting to research if it could become more accurate outdoors without walls and furniture as obstacles. It is also reasonable that if this solution should prove itself useful, specialized routers for this purpose could be developed to increase range and accuracy. Combined with a GNSS solution, this may be a cheap yet efficient approach for automated docking.

Camera

When we got the images from Tidefjord as mentioned in Subsection 3.1.6, we tried to use some methods in Matlab to extract information from them. The images provided were not good enough quality, and from an angle compared to the quay made it harder than first thought. We therefore decided not to spend too much time on it, as we wanted to spend more time on creating the simulation environment, and researching other sensors. That being said, cameras could definitely be used for distance-measuring, finding obstacles or to check if a vessel is properly aligned with a quay.

6.2 Integrating Auto-Docking in Dynamic Positioning Systems

Many vessels these days have a DP system integrated. A system like this usually utilize GNSS, and a gyro or satellite compass to keep its desired position and heading. It should be possible to use this when adding auto-docking, making it an extension of the DP system. This will not be enough for an autonomous vessel but it is a good start. The advantage of integrating auto-docking with a DP system is the good control system. Combining this with more high-precision sensors, it might just be a good enough solution.

6.3 Limitations with Auto-Docking Systems

Auto-docking systems may be one of the first steps for unmanned vessels, but the system in this case is thought to be more like an assistant for the crew on a vessel. To make a fully unmanned vessel, it would have to work in any circumstances that can occur. There are many factors affecting this, it would have to handle all possible weather situations, and be classed as at least a DP3 system like explained in Section 1.3. A sensible wind speed limit for using the auto docking system would be around $10m/s$. At this wind speed the vessel is affected by a large amount of force, so it may be an advantage to know how the vessel is affected by this force and maneuver it in a way that will not cause a dangerous situation.

6.4 Camera for Safe Docking

The use of camera to measure the distance between the vessel and the quay could be used in all of the cases. With suitable cameras and good calibration this would provide accurate measurements, and could also detect when the vessel is docked which is crucial for knowing when to stop the process of docking. One way to do this would be to have a camera above the front of the vessel and on the sides. This would ensure a good clear view of the quay and vessel.

There are many possible solutions for how to extract the interesting information from the images provided by these cameras. One way could be to transform the images to binary images, and use a method for edge detection. This would allow us to find the horizontal edges of the quay and vessel. With the right filters this could give two lines to focus on. With calibrated cameras the distance at two or three points along the lines could be measured as seen in Figure 6.1. This could give information about how close the vessel is to the quay, as well as how aligned they are. There are probably many ways to ensure two clear lines and filter out data which is not of interest, depending on the camera type. One way to do this when using "normal" color cameras could be to paint the front of the vessel and the end of the quay a particular color. Of course this could get dirty, and it might not be perfect. Heat-sensitive cameras with heated cables along the edges of the vessel and quay might be another way to handle it.

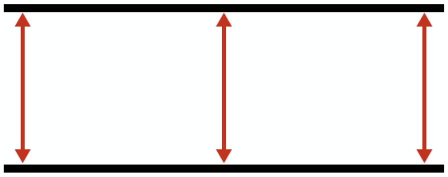


Figure 6.1: A simple representation of camera distance measuring

6.5 Results

The different test cases shown in Chapter 5 gave a wide specter of results, where it varied from $\pm 3.07cm$ to $1m$ in deviation from actual position. It was proven that the LiDAR was the most accurate distance measurement and RADius could not deliver the same accuracy. When these two were put together the results actually got worse than by using LiDAR alone. The reason for this lies propably in the sensor fusion part. The two sensors are weighted 50% between them, and combined with a GNSS system the distribution ends up as 25% on each distance measurement and 50% on the GNSS system. Then the RADius may have provided more noise to the position measurement rather than to actually improve it. This could be turned into an improvement by using a more advanced type of sensor fusioning, where the sensors where weighted according to their accuracy. Some of the sensors would maybe prove to be more useful when the vessel is closer to the quay than it would when the vessel is approaching the quay.

The system were able to keep its position and heading, with a wind speed up to $10m/s$. Since the ferry port in Hareid is in focus during this thesis, we collected the weather data from the closest weather station. As mentioned in Chapter 5 and seen in Figure 3.6, the data from this station showed us that during a five year period there would be a total of 628 days were the system would not have performed adequate. From a total of 1826 days this means the system would have performed well 52.42% of the time. This could have been solved by using a controller which is optimized for handling such strong external forces, or with more powerful thrusters.

From Table 2-1 in "Assessment of station keeping capability of dynamic positioning vessels" [8] presented by DNV GL, this auto-docking system would gain a DP capability number of 5. This means the systems would be operational for a maximum wind speed classified as "Fresh Breeze", with the assumption that forces gained by waves and currents are considered neglected. (See Appendix B.1)

6.6 Concluding Remarks

We created a simulation environment for the purpose of simulating the docking process of a vessel in 2D. We feel that the simulation environment, although not completely realistic, managed to display the precision of the systems we tested.

There is a lot of systems and sensors which probably can be used for autonomous docking of a vessel. We have simulated a few of them, and in the simulations they performed as expected. From the results, the best solution of those we simulated is a combination of LiDAR and RTK. They provided the most accurate results, and should in theory perform well in most situations.

LiDAR has advantages besides its precision, it is not as affected by different weather and if used right can even map out the surroundings of the vessel. This would be ideal for detecting obstacles, and could even be used to place the vessel on a graphical user interface which shows the surroundings of the vessel in real-time as mentioned in Subsection 3.1.2.

As mentioned, LiDAR with RTK seemed to be the system with the least deviation. However, LiDAR with DGPS provided almost as precise results with only one or two centimeter difference. It could be assumed that where it is not possible to place a transponder for the RTK, the DGPS could get the job done just as effectively since stations for DGPS can be placed as far as 50km away from the desired area and still perform adequate.

Galileo is said to be able to provide up to $1cm$ precision encrypted, but in the simulations we decided to use the public guaranteed accuracy of $1m$. It could be that Galileo is accurate enough if the $1cm$ precision were accomplished, but the public guaranteed precision of $1m$ is probably not good enough when docking autonomously.

RADius is a good sensor for measuring distance and heading between the transponder and interrogator. Although in the simulations, the accuracy did not perform as well as LiDAR even with a very precise GNSS. It could be argued that with good sensor fusion, RADius could be part in controlling the heading, and act as a backup or fail-safe for a more precise system.

We did not quite manage to test cameras for this process. It is still something we believe could benefit an autonomous process like this if the right camera type and software is being used. We imagine camera could be used to either see when the vessel is docked, to map the surrounding area, or both.

In order to have a redundant system one should probably use more than only LiDAR. It would probably be more suitable to use a different sensor type than RADius, a sensor which can match the precision of the LiDAR. It is not easy to specify exactly how many types of sensors should be used in a system for autonomous docking. Too few and the system is not redundant enough, too many and the system gets more complicated than necessary. It could be that a good GNSS, LiDAR and maybe some cameras combined with good software would be enough for most vessels and situations.

The subject of auto docking is complex and comes with extensive demands for safety and precision that is difficult to match. In order to reach the demands for a fully autonomous docking operation one is dependant on robust, accurate and versatile sensors, and software which utilizes these sensors to the fullest with little to no room for failure. Since this is a DP0 system it is not necessary to be completely fail-proof, there should always be one or more personnel on the bridge ready to take control of the vessel should anything happen. That being said, safety is never a subject to underestimate. It is important to show the industry, the crew and the passengers that autonomous docking is safe, beneficial and the future.

6.6.1 Recommendations for Further Work

This project offers satisfying results from the simulation, but there could be some improvements, especially when it comes to the control system. The tuning of the PID-Controller could be done by using a Genetic Algorithm. The reason GA could be used is because it can be tuned without any training or existing data set. With an improvement like this, the system can be optimized with a series of different factors in mind: Like optimal thrust control for better fuel usage, time used for docking the vessel automatically and position accuracy.

Another control system improvement would be to use the distance measurement sensors as actuators. This means to maybe add another controller to the system, and weight the values between them according to how close the vessel is to the quay. With a change like this, the system would become more redundant, agile, and the possibility to extend the system with different types of sensors.

Since the PID is dependent on an observer, the existing observer could be changed with a extended or unscented kalman filter. This is an observer that will work with nonlinear systems, and give an accurate estimation of position from a more advanced type of sensor fusion than used in this project. Kalman filter technology is relevant because it is one of the observers that will manage different sensors and evaluate them with their accuracy in mind, and give a estimated position using the force acting on the system and the measurements from the sensors.

The thursters in the current simulation application is working together and splitting the load 50% between them. There should be a thruster optimization algorithm, for using the thrusters individually. Ferries usually use the front thruster to pull, and the stern thruster to brake. This is not the most optimal solution if hydrodynamics are considered, but it does not affect the quay or the surroundings that much. With a more optimal thruster control, the system would may be able to handle stronger environmental disturbances. This would improve the DP capability number of the system, and yield more operational time of the auto-docking system.

Forces simulating waves and currents in the ocean can be implemented for creating a more realistic simulation of environmental forces.

Further research on camera types, and how to utilize cameras would be interesting. It would also be interesting to see if camera and LiDAR could be combined to map the surrounding area of a vessel, maybe display it in 3d using images from cameras or only the measurements from several LiDARs.

Bibliography

[1] Intel (2017). Learn about multiple-input multiple-output. https://www.intel.com/content/www/us/en/support/articles/000005714/network-and-i-o/wireless-networking.html.

[2] S. Lagüela L. Díaz-Vilariño P. Arias A. Filgueira, H. González-Jorge. Quantifying the influence of rain in lidar performance. https://www.sciencedirect.com/science/article/pii/S0263224116305577.

[3] Richard Aase. Dgps. http://www.kystverket.no/Maritime-tjenester/Meldings--og-informasjonstjenester/Radionavigasjon-DGPS/.

[4] N.A. Jensen Balchen, J. G. and S. Sælid. Dynamic positioning using kalman filtering and optimal control theroy. *Technical report*, 1976.

[5] Association of Hanseatic Marine Underwriters Capt. P. Zahalka. Bollard pull. *Technical report*, 93:1–10, 2008.

[6] T. I Fossen and T. Perez (2004). Marine systems simulator (mss). http://www.marinecontrol.org/.

[7] Thor I. Fossen. *Marine Control Sysyems: Guidance, Navigation and Control of ships, Rigs and Underwater Vehicles.* Marine Cybernetics AS, Trondheim, Norway, 2002.

[8] DNV GL. Assessment of station keeping capability of dynamic positioning vessels. https://rules.dnvgl.com/docs/pdf/DNVGL/ST/2016-07/DNVGL-ST-0111.pdf.

[9] DNV GL. Dp. http://rules.dnvgl.com/docs/pdf/gl/maritimerules/gl_i-1-15_e.pdf.

[10] Galileo GNSS. Galileo. http://galileognss.eu/do-winter-weather-conditions-have-an-effect-

[11] J. Grafarend, E. & Shan. Gps solutions (2002). https://doi.org/10.1007/PL00012897.

[12] Erik Gregersen. Lidar. https://www.britannica.com/technology/lidar.

[13] Pooja Gupta and S.P. Kar (2015). Music and improved music algorithm to estimate direction of arrival. https://www.mathworks.com/matlabcentral/answers/uploaded_files/48180/gupta2015.pdf.

[14] R. M. Isherwood. *Wind Resistance of Merchant Ships*. Royal Institution of Naval Architects, England, 1973.

[15] Norvald Kjerstad. *Elektroniske og Akustisje Navigasjonssystemer*. Fagbokforlaget, Ålesund, Norway, 2015.

[16] Dinesh Bharadia Manikanta Kotaru, Kiran Joshi and Sachin Katti. Spotfi. https://dl.acm.org/citation.cfm?doid=2785956.2787487.

[17] Kongsberg Maritime. Radius. https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/57E47945903AB766C12570F3003061A8?OpenDocument.

[18] Kongsberg Maritime. Radius. https://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/DD5597E3D4A1D779C1256F540034A762/$file/ProductSheet_RADius_jan17.pdf?OpenElement.

[19] Michael Schmandt. Gis commons: An introductory textbook on geographic information systems. http://giscommons.org/chapter-2-input/.

[20] Martin Schüssel. Angle of arrival. http://www3.uah.es/ipin2016/usb/app/descargas/223_WIP.pdf.

[21] Asgeir J. Sørensen. *Marine Control Systems*. Department of Marine Technology, NTNU, Trondheim, Norway, 2013.

[22] Terrabee. Time-of-flight. https://www.terabee.com/time-of-flight-principle/.

[23] Keith. Thomassen. How gps works. http://avionicswest.com/Articles/howGPSworks.html.

[24] Velodyne. Datasheet velodyne hdl-32e. http://velodynelidar.com/lidar/hdldownloads/HDL-32E%20datasheet_web.pdf. Datasheet.

[25] Jie Xiong and Kyle Jamieson. Arraytrack. https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final51.pdf.

[26] Jie Yang and Yingying Chen. Rssi and lateration. https://ieeexplore.ieee.org/document/5425237/?arnumber=5425237&tag=1.

[27] Yin Zhou and Oncel Tuzel. End-to-end learning for point cloud based 3d object detection. https://arxiv.org/pdf/1711.06396.pdf.

Appendix A

A.1 Introduction

The appendix contain the layout in Simulink and Matlab code used for the simulation. Some explanations are included for most parts. The scopes with the deviations are also included, where the deviation from actual position and estimated position is displayed, and weather data collected from Vigra.

A.1.1 More Details

A.2 Simulation

Persistent variables are used throughout the custom-made functions. A persistent variable will carry its content from one function call to the next, whereas a normal variable will not remember its state from the last time the function was called.

A.2.1 Position plot

The position plots task is to visualize the vessels position and route on a map within a coordinate system. The vessel is represented with a red triangle, and a short line pointing out from the vessel/triangle is the heading of the vessel in NED-Coordinates. Several dots is drawn where the vessel is moving, which on the plot looks like a line. This is to see the traveled path of the vessel. The two variables h3 and h4 holds the triangle representing the vessel and the line representing the heading, respectively. This allows for the triangle and line to be deleted before they are plotted again, creating an illusion of the two moving continuously when in fact they are deleted and plotted again every time the function is called. The if-statement on line 11 that checks if the variable h3 is empty, is there to make sure the plot is set up, and that h3 and h4 is defined when first trying to delete them, since it would make an error if they were not. This if statement will only result in true the first time the function is called. The heading of the vessel is found using the lines from 31 to 33. A coordinate created on the x-axis at (10,0), then converted to Polar coordinates. The vessels yaw-angle is then added to the Polar coordinates ’THETA’, and the polar coordinates is then converted back to Cartesian coordinates. The coordinate representing the heading is in NED coordinates. By adding the vessels coordinates, it becomes relative to the BODY coordinates. When plotting the line that represents the vessels heading, a line is plotted between the vessels position and the coordinate that represents the heading.

```
1  function fnc(x,y, a_rad)
2  %#codegen
3  coder.extrinsic('plot', 'imagesc', 'delete')
4
5  % Initiate persistent variables
6  persistent h3;
7  persistent h4;
8
9  figure(1)
10 % If first run, initialize flag and set up the plot configurations.
11 if isempty(h3)
12      i = imread('Hareid_map.jpg');
13      imagesc([-250 250],[-250 250],i); hold on
14      set(gca,'YDir','normal');
15      grid on
16     title('Map plot');
17     % h3 and h4 must be given values so they are not empty.
18     h3 = plot(x,y,'<','Markersize',5, 'Color', 'r');
19     h4 = plot(x,y,'<','Markersize',5, 'Color', 'r');
20 end
21
22 % Plot the ships route on the plot.
23 plot(x,y,'.','Markersize',0.5, 'Color', 'r')
24
25 % Delete the triangle representing the ship, then plots the new position.
26 delete(h3);
27 h3 =  plot(x,y,'<','Markersize',5, 'Color', 'r');
28
29 % Creates a short line, and rotate it to point the heading of the vessel
30 % relative to the ship.
31 [THETA,R] = cart2pol(10,0);      % Convert to polar coordinates
32 THETA=THETA+a_rad;               % Add a_rad to theta
33 [xr,yr] = pol2cart(THETA,R);     % Convert back to Cartesian coordinates
```

```

% Deletes the heading line, then plot it again at the ships position.
delete(h4)
h4 = plot([x xr+x], [y yr+y], 'Markersize', 1, 'Color', 'r');
```

A.2.2 Overview

A.2.3 DP Controller

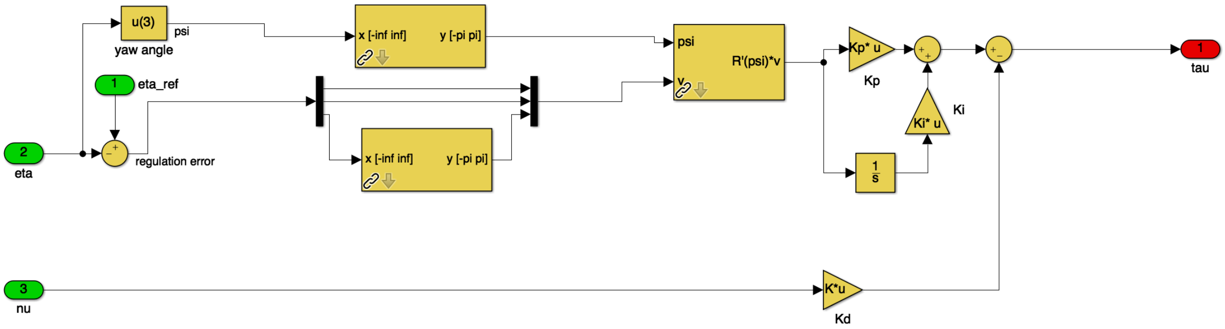


Figure A.1: Nonlinear DP Controller

The Nonlinear DP Controller comes from the MSS toolbox. It is a nonlinear PID controller that uses the rotation matrix in yaw to find the surge,sway and yaw gains as a function of the yaw angle.

A.2.4 Observer

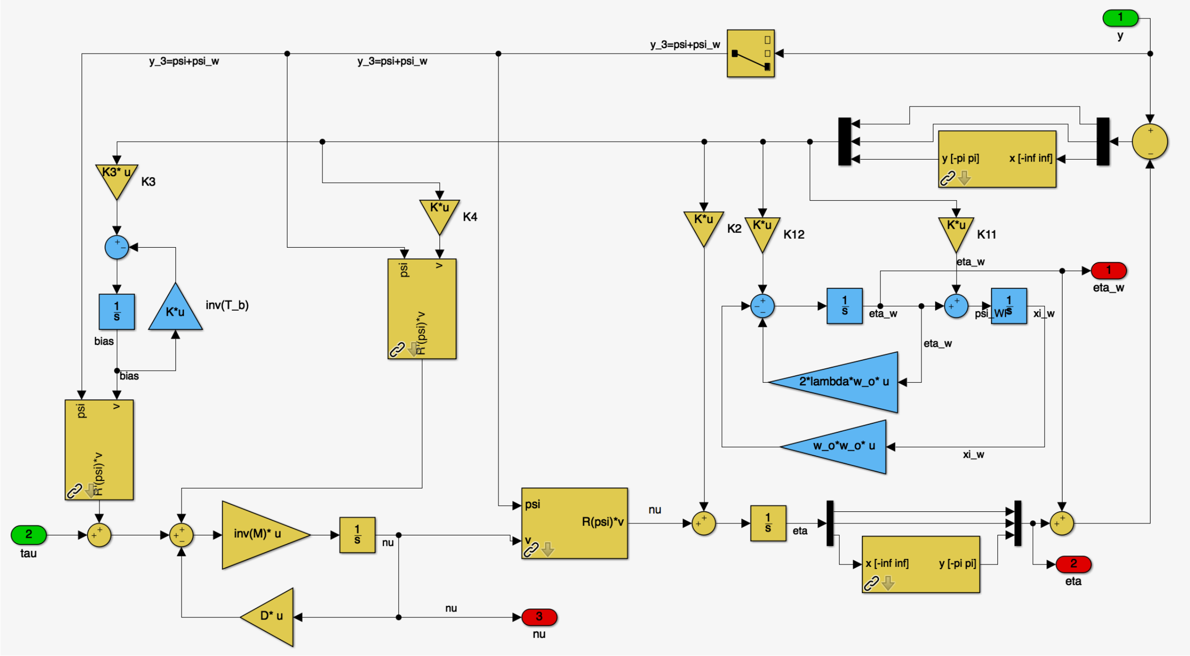
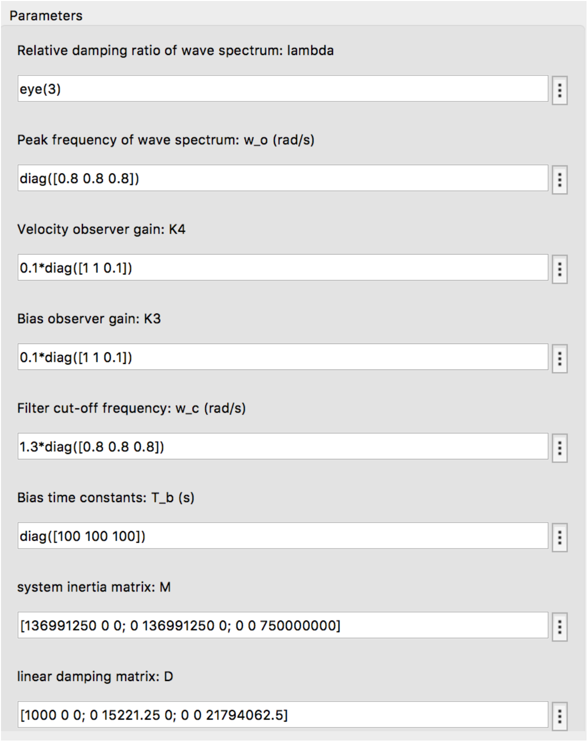


Figure A.2: Observer overview

Nonlinear passive observer from the MSS toolbox.



Parameters

Relative damping ratio of wave spectrum: lambda

eye(3)

Peak frequency of wave spectrum: w_o (rad/s)

diag([0.8 0.8 0.8])

Velocity observer gain: K4

0.1*diag([1 1 0.1])

Bias observer gain: K3

0.1*diag([1 1 0.1])

Filter cut-off frequency: w_c (rad/s)

1.3*diag([0.8 0.8 0.8])

Bias time constants: T_b (s)

diag([100 100 100])

system inertia matrix: M

[136991250 0 0; 0 136991250 0; 0 0 750000000]

linear damping matrix: D

[1000 0 0; 0 15221.25 0; 0 0 21794062.5]

Figure A.3: Observer tuning parameters

A.2.5 Thruster Control

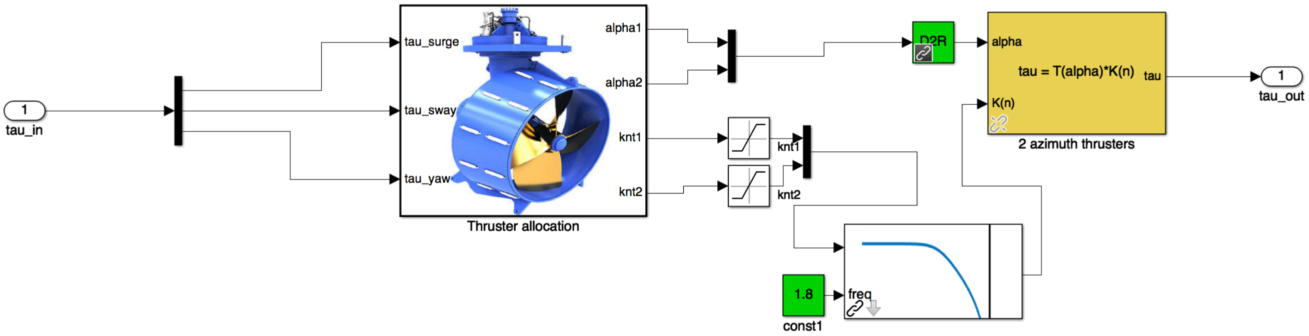


Figure A.4: Thruster control overview

The thruster control block is built to calculate the commanded thrust from the DP controller. The first block Thruster allication breaks down the τ_{PID} vector and calculates the given force the thrusters need to produce, and the angle they need to be in for affecting the vessel in a way that will move it to the reference position. For calculating the force the τ vector is parted in

$[\tau_{surge} \;\; \tau_{sway} \;\; \tau_{yaw}]$ this vector represents the forces in the 3DOF that the simulation is based on. In the ThrustAllication function the total force is calculated as a product of these three vectors (knt1 & knt2).

The angle of the thrusters have a orbit in ±180°. Where 0° gives the maximum positive thrust in surge direction. The commanded angle of the thrusters is calculated using Equation A.1.

$$\alpha = \pm 1 \times \frac{180}{\pi} \times \cos^{-1}(\frac{\tau_{surge}}{knt1}) \tag{A.1}$$

```
function [alpha1,alpha2,knt1,knt2] = ThrustAllocation(tau_surge,tau_sway,tau_yaw)

x1=45;
x2=-45;

%tau_yaw_force
tyf = tau_yaw/x1;

%tau_yaw_force pr. thruster
tyfpt = tyf/2;

%thruster_surge
t_surge = tau_surge/2;

%thruster_sway
t_sway = tau_sway/2;

%Total sway pr. thruster
tst1 = t_sway+tyfpt;
tst2 = t_sway-tyfpt;

%Force (Tau) pr. thruster

knt1 = sqrt((t_surge.^2)+(tst1.^2));
knt2 = sqrt((t_surge.^2)+(tst2.^2));

%Thruster angle (alpha)
    %Thruster 1
    if tst1 < 0
        ang = -1;
    else
        ang = 1;
    end

    alpha1 = ang*(180/pi)*acos(t_surge/knt1);

    %Thruster 2
    if tst2 < 0
        ang2 = −1;
    else
        ang2 = 1;
    end

    alpha2 = ang2*(180/pi)*acos(t_surge/knt2);
```

From the output of the Thruster allocation block there is two saturnation block for limiting the power the thrusters can produce by the calculated bollard pull. There is also a discrete varying low pass filter for delaying the produced thrust by three seconds for making the simulation more realistic.

The block called "2 azimut thruster" handles the produced thrust and the given angle of the thrusters to calculate the total τ vector that will affect the vessel. This is done according to Equation 2.17. This block then holds the thruster configuration matrix (Equation 2.18) as represented in Figure A.5.

Thruster configuration matrix

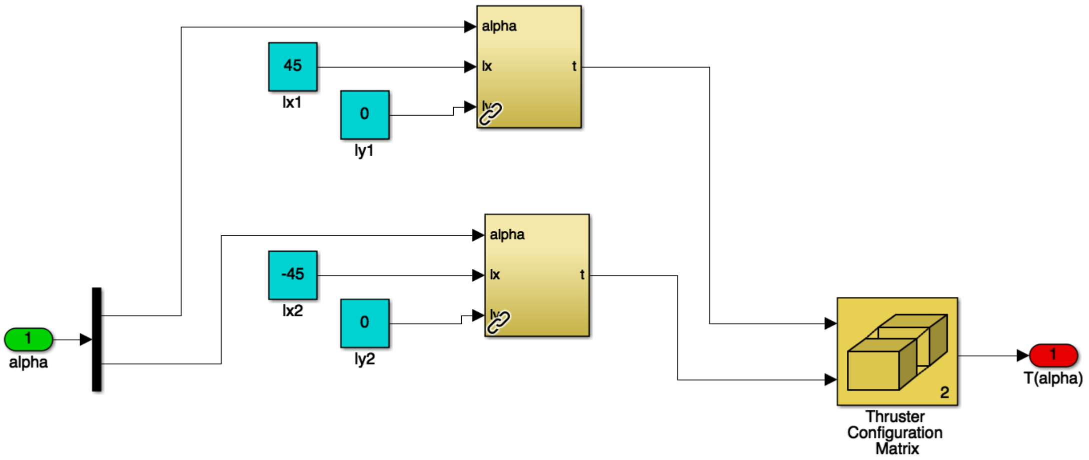


Figure A.5: Thruster configuration matrix

The thruster configuration matrix are calculating how the amount of force will affect the vessel according to their location on the vessel. "lx" is the position in x-direction, and "ly" is the poistion in y-direction each has their origin from the COG of the vessel.

A.2.6 Ferry/Vessel

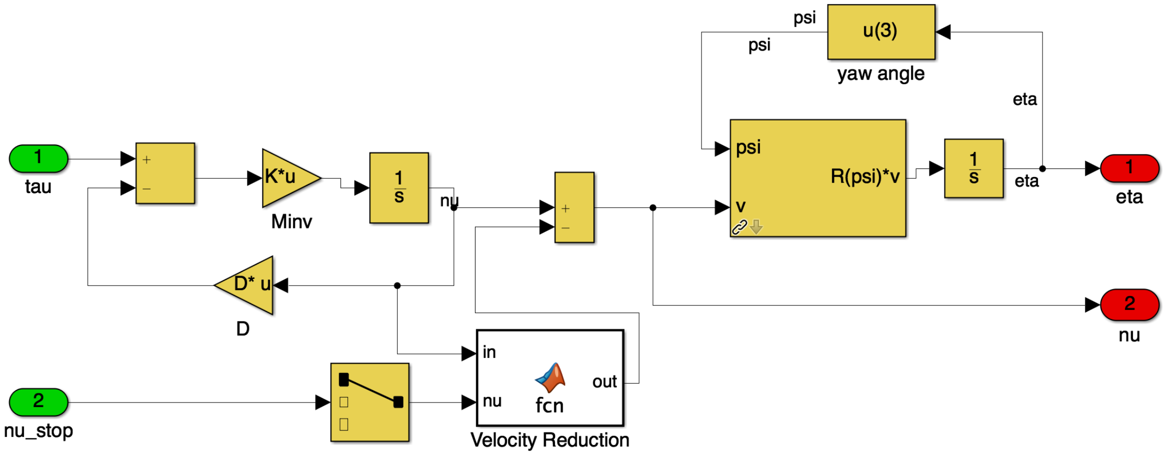


Figure A.6: Ferry block overview

Low frequent control plant model for the Ferry. This was originally designed to be a supply vessel, but its modified to represent a ferry by tuning.

A.2.7 Sensor Block

The sensor block contains the GNSS and sensors for the simulations. The input to this block contains the vessels position in X and Y coordinates and the heading in radians. The output will contain the same data, but with slightly different X and Y coordinates. This is because some noise is added to simulate the error of measurement in GNSS and the sensors. By tapping the switches, one can select what sensors and GNSS to include in the simulation. By weighing the output from the GNSS and sensors used 50/50, the output will be equally estimated by the GNSS and the sensors.

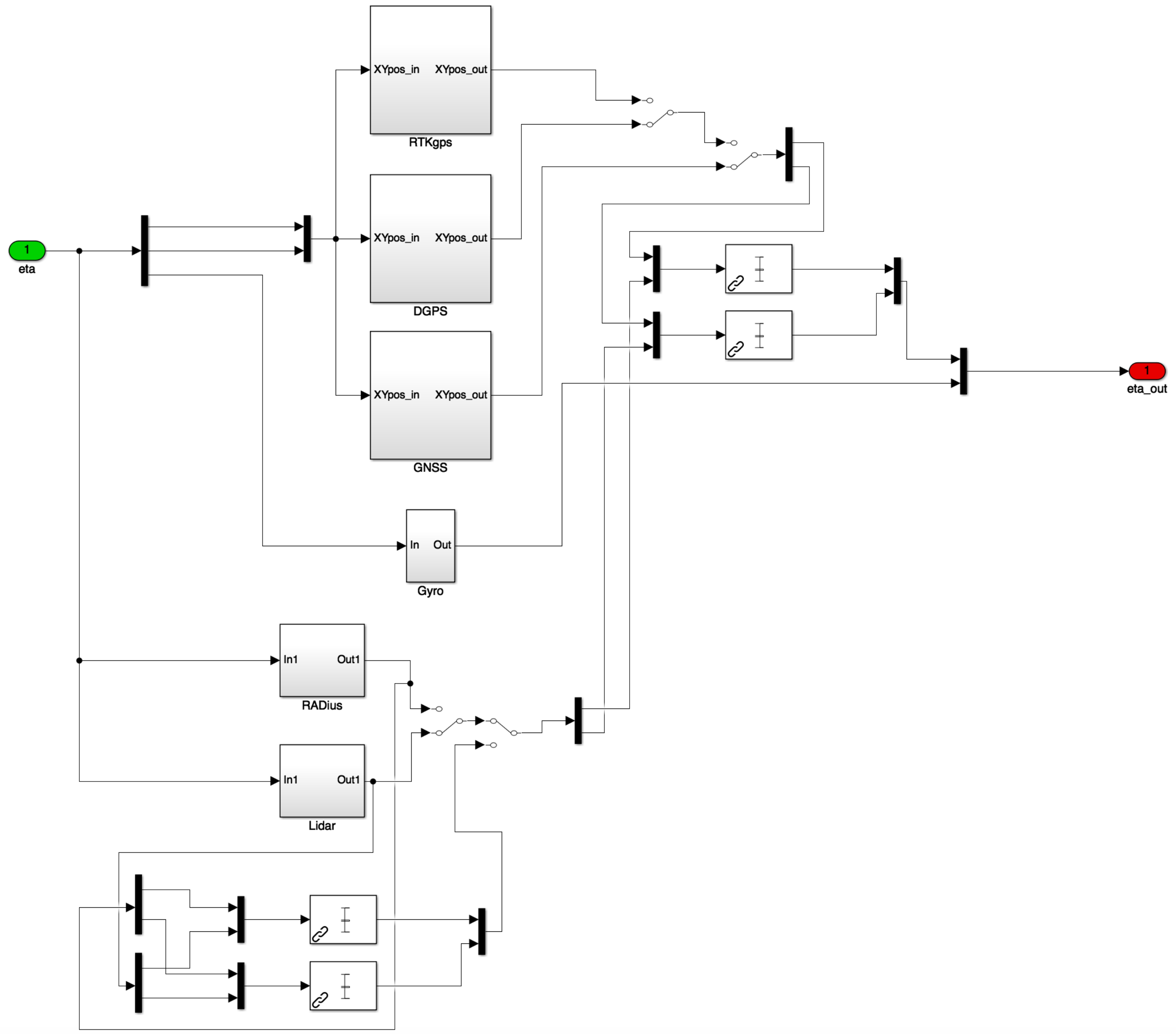


Figure A.7: The sensor block.

A.2.8 LiDAR

The LiDAR function visualizes a LiDAR-system, and give an estimated output of the vessels position. Since the ships actual position is used in this process, the estimated position is very precise. It was decided to do this since the PID being used need an X/Y position. While it is not the most realistic representation of a real LiDAR, it was decided it was good enough. By adding some noise to the output coordinates, a more realistic range of error is created. Three points is made on the quay, and a line is drawn between the two points furthest out to the one in "the middle", creating a ’L’ shape. These lines represents the physical quay. 250 points is then created along these lines, between the existing edges of the lines. This will make the resolution of the LiDAR higher, and it will look a lot smoother when visualizing the lines of the LiDAR representing the shortest distance from the vessel to the quay. The for-loops on line 50 to 69 finds the points on the lines that is closest to the vessel. This is done by creating two extremely high variables, dist1 and dist2. The first for-loop checks the distance between the vessel and the long-side of the quay. Should the measured distance between the vessel and the point of interest at any point in the for-loop be shorter than dist1, then this point on the line will be marked as the closest point (q1 and q2), and dist1 will be set to that distance. When the for-loop is done, q1 and q2 will therefore end up being the points closest to the vessel. This is the same for the next for-loop, except here dist2, q3 and q4 is being used. q1 through q4 is being used to store the coordinates of the closest point on the two lines. The distance between the vessel and the point on the long side quay is then measured, and two lines will be drawn from the vessel to the closest point on both of the quay-sides if the distance between the vessel and the two points is lower than 60 for the long-side quay and less than 50 for the short-side quay. The vessels position is then calculated using the closest point on the short-side quay, and a fictional helping-point on the same y coordinate as that point, but 50 points added to the x-coordinate. It was decided to do it this way, since it proved hard to do it without a helping point. This is because in order to calculate the position of the vessel from these points, the angle between two lines must be known, and the angle of one of those lines must be known as well. Since the helping point is on the same y-coordinate as the point on the short side quay, it is known that the angle of that line is 0 degrees, and the vessels position can therefore be calculated. Doing it this way should make the output very similar to the input.

```
1  function [estX, estY] = fcn(x, y, yaw)
2  % x and y is the coordinates of the ship.
3  %#codegen
4  coder.extrinsic('plot', 'imagesc', 'delete')
5  figure(1)
6  % Initiate persistent variables.
7  persistent h1;
8  persistent h2;
9
10 % Draw the lines of the quay on the plot.
11 plot([-40 -83], [12 37], 'Color', 'r', 'Markersize', 1);
12 plot([-83 -91], [37 25], 'Color', 'r', 'Markersize', 1);
13
14 % 1. Coordinates, furthest out on the quay (east).
15 x1 = -40;
```

```
y1 = 12;

% 2. Coordinates, on the quay, north.
x2 = -83;
y2 = 37;

% 3. Coordinates, on the quay, south.
x3 = -91;
y3 = 25;

% Initializing the variables with meaningless data.
q1 = 0;
q2 = 0;
q3 = 0;
q4 = 0;

% Make several points between first and second point. (longside of quay)
NumberNewPoints = 250;
xvals = linspace(x1, x2, NumberNewPoints+2);
yvals = linspace(y1, y2, NumberNewPoints+2);
pts1 = [xvals(:), yvals(:)];

% Make several points between second and third point. (Short side, where
% the cars will drive on and off the ferry.)
NumberNewPoints = 250;
xvals = linspace(x2, x3, NumberNewPoints+2);
yvals = linspace(y2, y3, NumberNewPoints+2);
pts2 = [xvals(:), yvals(:)];

% Set two impossible long distances so we can find the shortest.
dist1 = 10000000; % Distance from ship to  long quayside
dist2 = 10000000; % Distance from ship to short quayside

% Find the point in pts1 with shortest distance.
for i = 1 : 252 %size(pts1)
    thisDist1 = sqrt((pts1(i,1)-x).^2+(pts1(i,2)-y).^2);

    if thisDist1 < dist1
        dist1 = thisDist1;
        q1 = pts1(i,1);
        q2 = pts1(i,2);
    end
end

% Find the point in pts2 with shortest distance
```

```
for i = 1 : 252 %size(pts2)
    thisDist2 = sqrt((pts2(i,1)-x).^2+(pts2(i,2)-y).^2);

    if thisDist2 < dist2
        dist2 = thisDist2;
        q3 = pts2(i,1);
        q4 = pts2(i,2);
    end
end

% Plot lines from ship to quay ------------------------------------------
if isempty(h1)
    h1 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
    h2 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
end
% removes then re-plots the lines visualizing the shortest distance from
% the ship to the quayside.
delete(h1);
delete(h2);

% Only plot the lines if the distance is shorter than the given value.
if dist1 < 60
    h1 = plot([x q1], [y q2], 'Color', 'k', 'Markersize', 1);
end
if dist1 < 50
    h2 = plot([x q3], [y q4], 'Color', 'k', 'Markersize', 1);
end

%------------------------------------------------------------------------
% Calculate estimated X and Y position from the lidars readings.


% pts1 = longside quay, q1 and q2 = x and y
% pts2 = shortside quay, q3 and q4 = x and y
% dist1 = distance to longside quay, dist2 = distance to shortside quay

% Calculate lengths between ship and the points on quay closest to the
% ship. For reference; p1 = point on shortside quay, p3 point on longside
% quay and p2 is the ship. p21 is the distance between ship and shortside
% quay.
q5 = q3+50;
q6 = q4;
p31 = sqrt((q5-q3).^2 + (q6-q4).^2);
p21 = sqrt((q3-x).^2 + (q4-y).^2);
p32 = sqrt((q5-x).^2 + (q6-y).^2);
```

```

% Calculate angle in radians between p31 and p21
angleRad = acos((p21.^2 + p31.^2 - p32.^2 )/ (2 * p21 * p31));

% Find coordinate of ship as if p1 were Origo.
estX = p21 * cos(-angleRad);
estY = p21 * sin(-angleRad);

% Add the p1 coordinates to these coordinates to make the position relative
% to the coordinate system.
estX = estX + q3;
estY = estY + q4;

end
```

A.2.9 RADius

RADius has a transponder on land and a interrogator on the vessel. In the function, the distance between these two points is calculated using Pythagoras theorem, as well as the angle between the heading of the vessel and the direct line from the vessel to the transponder using geometry. When the distance between the interrogator and the transponder is less than 120, a line will be plotted to show the direct line of view. The function also plots two lines at both the interrogator and transponder to visualize their viewing angle, which is 90 degrees. As with the LiDAR, in order to estimate the vessels position, a helping point is created on the same y-coordinate, but with a different x-coordinate than the transponder. The angle between the direct line between the transponder and interrogator, and the transponder and the helping point is then found. With that angle and the length between the interrogator and transponder, the Cartesian coordinates is found, as if the transponder is origin. By adding the transponders coordinates to these estimated coordinates, they become relative to the NED coordinate system, giving a very accurate estimate of the ships actual position.

```
function [estX, estY] = fcn(x, y, yaw)
%#codegen
coder.extrinsic('plot', 'imagesc', 'delete', 'degtorad', 'fill', 'radtodeg')
figure(1)
% Initialize persistent variables.
persistent line1;
persistent radiusLine1;
persistent radiusLine2;
persistent radiusLine3;
persistent radiusLine4;

% Ships position
p2x = x;
p2y = y;

% Transsponder position on land
p1x = -88;
p1y = 29;

% Distance between ship and transsponder.
distance = sqrt((p1x-x).^2+(p1y-y).^2);

[THETA,R] = cart2pol(20,0); %Convert to polar coordinates
THETA=THETA+yaw; %Add yaw to theta
[xr,yr] = pol2cart(THETA,R); %Convert back to Cartesian coordinates

% Create the point ahead of the ship.
p3x = x+xr;
p3y = y+yr;

% Define lines between ship, transsponder and helping point ahead of ship.
p21 = sqrt((p1x-p2x).^2 + (p1y-p2y).^2);
p23 = sqrt((p3x-p2x).^2 + (p3y-p2y).^2);
p13 = sqrt((p3x-p1x).^2 + (p3y-p1y).^2);

% Find the angle. "p2 = skip"
angleRad = acos((p21.^2 + p23.^2 - p13.^2 )/ (2 * p21 * p23) );
angleDeg = radtodeg(angleRad);

% Plot lines from ship to quay ------------------------------------------------
if distance < 120
    if isempty(line1)
        line1 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
    end
    delete(line1);

    if (angleRad < 0.8)
        line1 = plot([p1x p2x], [p1y p2y], 'Color', 'r', 'Markersize', 1);
    end
end


%------------------------------------------------------------------------------
% Add view range of interrogator and transsponder

% -45 degrees from the heading of the ship
[THETA,R] = cart2pol(20,0);      %Convert to polar coordinates
THETA=THETA+yaw-(pi/4);          %Add a_rad to theta
[xr,yr] = pol2cart(THETA,R);     %Convert back to Cartesian coordinates
% Create the point ahead of the ship.
radius1x = x+xr;
radius1y = y+yr;

% +45 degrees from the heading of the ship
[THETA,R] = cart2pol(20,0);      %Convert to polar coordinates
THETA=THETA+yaw+(pi/4);          %Add a_rad to theta
[xr,yr] = pol2cart(THETA,R);     %Convert back to Cartesian coordinates
% Create the point ahead of the ship.
radius2x = x+xr;
radius2y = y+yr;

% Transsponders viewing angle helping points position.
radius3x = -68;
radius3y = 29;
radius4x = -88;
radius4y = 9;

% Make sure persistent variables are not empty before they are being used.
if isempty(radiusLine1)
    radiusLine1 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
    radiusLine2 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
    radiusLine3 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
    radiusLine4 = plot([x 1], [y 1], 'Color', 'g', 'Markersize', 1);
end

% delete old lines and create new ones.
delete(radiusLine1);
delete(radiusLine2);
delete(radiusLine3);
delete(radiusLine4);
radiusLine1 = plot([x radius1x], [y radius1y], 'Color', 'k', 'Markersize', 1);
radiusLine2 = plot([x radius2x], [y radius2y], 'Color', 'k', 'Markersize', 1);
radiusLine3 = plot([p1x radius3x], [p1y radius3y], 'Color', 'k', 'Markersize', 1);
radiusLine4 = plot([p1x radius4x], [p1y radius4y], 'Color', 'k', 'Markersize', 1);

% ------------------------------------------------------------------------
% Find coordinate of ship based on RADius measurements.
p4x = -60;
p4y = 29; % Choosing same y-coordinate for helping point as the transponder.
```

```

% p21 is already defined.
p42 = sqrt((p2x-p4x).^2 + (p2y-p4y).^2);
p41 = sqrt((p1x-p4x).^2 + (p1y-p4y).^2);

% Find angle between p31 line and p21 line.
angleRad2 = acos((p21.^2 + p41.^2 - p42.^2 )/ (2 * p21 * p41));

% Find coordinate of ship as if RADius were Origo.
estX = p21 * cos(-angleRad2);
estY = p21 * sin(-angleRad2);

% Add the transsponders coordinates to these coordinates to make them
% relative to the coordinate system.
estX = estX + p1x;
estY = estY + p1y;

end
```

A.2.10 Wind Force

The block which creates the wind forces are in the MSS GNC toolbox. The mathematical methods are written about in Section 2.2. The inputs to the function are specifications of the ship being used, what angle one want the wind to come from and how strong the wind is. More about this is written in Section 4.5. The output of the function is a vector containing the forces affecting the ship in surge, sway and yaw direction.

```
function [tau_w,cx,cy,cn] = windcoef(gamma_r,V_r,L,B,A_L,A_T,A_SS,S,C,M)
% [w_wind,cx,cy,cn] = windcoef(gamma_r,V_rL,B,A_L,A_T,A_SS,S,C,M) returns the the wind
% force/moment vector w_wind = [tauX,tauY,tauN] and the optionally wind coeffisients
% cx,cy and cn for merchant ships using the formulas of Isherwood (1972).
%
% INPUTS:
% gamma_r = relative wind angle (rad)
% V_r     = relative wind speed (m/s)
% L           = length overall (m)
% B           = beam (m)
% A_L     = lateral projected area (m^2)
% A_T     = transverse projected area (m^2)
% A_SS    = lateral projected area of superstructure (m^2)
% S           = length of perimeter of lateral projection of model (m)
%                 excluding waterline and slender bodies such as masts and ventilators
```

```
    (m)
% C           = distance from bow of centroid of lateral projected area (m)
% M           = number of distinct groups of masts or king posts seen in lateral
%                    projection; king posts close against the bridge front are not
    included
%
% Author:     Thor I. Fossen
% Date:       10th September 2001
% Revisions: 19th April 2004, changed velocity from knots to m/s. This was a bug
% _______________________________________________________________
%
% MSS GNC is a Matlab toolbox for guidance, navigation and control.
% The toolbox is part of the Marine Systems Simulator (MSS).
%
% Copyright (C) 2008 Thor I. Fossen and Tristan Perez
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program.  If not, see <http://www.gnu.org/licenses/>.
%
% E-mail: contact@marinecontrol.org
% URL:     <http://www.marinecontrol.org>

 if nargin~=10, error('the number of inputs must be 10');end

% conversions and constants
rho_a = 1.224;            % density of air at 20 C
gamma_r = gamma_r*180/pi;  % rad2deg

% CX_data = [gamma_r    A0      A1      A2      A3      A4      A5      A6      ]
CX_data= [...
0       2.152   -5.00   0.243   -0.164  0           0       0
10      1.714   -3.33   0.145   -0.121  0           0       0
20      1.818   -3.97   0.211   -0.143  0           0       0.033
30      1.965   -4.81   0.243   -0.154  0           0           0.041
40      2.333   -5.99   0.247   -0.190  0       0           0.042
```

```
50      1.726   -6.54   0.189   -0.173  0.348   0           0.048
60      0.913   -4.68   0           -0.104      0.482   0           0.052
70      0.457   -2.88   0           -0.068      0.346   0           0.043
80      0.341   -0.91   0           -0.031      0           0           0.032
90      0.355   0           0       0           -0.247  0           0.018
100     0.601   0           0       0       -0.372      0           -0.020
110     0.651   1.29    0           0       -0.582      0           -0.031
120     0.564   2.54    0           0       -0.748      0           -0.024
130     -0.142  3.58    0           0.047       -0.700  0           -0.028
140     -0.677  3.64    0           0.069       -0.529  0           -0.032
150     -0.723  3.14    0           0.064       -0.475  0           -0.032
160     -2.148  2.56    0           0.081       0           1.27        -0.027
170     -2.707  3.97    -0.175  0.126   0           1.81        0
180     -2.529  3.76    -0.174  0.128   0       1.55    0             ];

% CY_data = [gamma_r B0 B1      B2      B3      B4      B5      B6]
CY_data = [...
0   0       0       0       0       0       0       0
10      0.096   0.22    0           0       0           0       0
20      0.176   0.71    0           0       0       0       0
30      0.225   1.38    0           0.023       0           -0.29       0
40      0.329   1.82    0           0.043   0           -0.59   0
50      1.164   1.26    0.121   0           -0.242      -0.95   0
60      1.163   0.96    0.101   0           -0.177      -0.88   0
70      0.916   0.53    0.069   0           0           -0.65   0
80      0.844   0.55    0.082   0           0           -0.54   0
90      0.889   0           0.138       0           0           -0.66   0
100     0.799   0           0.155       0           0           -0.55   0
110     0.797   0           0.151       0           0       -0.55       0
120     0.996   0           0.184       0           -0.212      -0.66   0.34
130     1.014   0           0.191       0           -0.280      -0.69   0.44
140     0.784   0           0.166       0           -0.209      -0.53   0.38
150     0.536   0           0.176       -0.029  -0.163  0           0.27
160     0.251   0           0.106       -0.022  0           0       0
170     0.125   0           0.046       -0.012  0           0       0
180 0       0       0       0       0       0       0     ];

% CN_data = [gamma_r C0 C1      C2      C3      C4      C5]
CN_data = [...
0   0       0       0       0       0       0
10      0.0596  0.061   0           0       0       -0.074
20      0.1106  0.204   0           0       0       -0.170
30      0.2258  0.245   0       0           0       -0.380
40      0.2017  0.457   0           0.0067      0           -0.472
50      0.1759  0.573   0           0.0118      0           -0.523
```

```
60      0.1925  0.480   0           0.0115      0           -0.546
70      0.2133  0.315   0           0.0081      0            -0.526
80      0.1827  0.254   0           0.0053      0           -0.443
90      0.2627  0           0       0       0       -0.508
100     0.2102  0           -0.0195     0           0.0335      -0.492
110     0.1567  0           -0.0258     0           0.0497      -0.457
120     0.0801  0           -0.0311     0           0.0740      -0.396
130     -0.0189 0           -0.0488     0.0101  0.1128  -0.420
140     0.0256  0           -0.0422     0.0100  0.0889  -0.463
150     0.0552  0           -0.0381     0.0109  0.0689  -0.476
160     0.0881  0           -0.0306     0.0091  0.0366  -0.415
170     0.0851  0           -0.0122     0.0025  0       -0.220
180 0       0       0       0       0       0       ];


% interpolate in the tables
A0 = interp1(CX_data(:,1),CX_data(:,2),gamma_r);
A1 = interp1(CX_data(:,1),CX_data(:,3),gamma_r);
A2 = interp1(CX_data(:,1),CX_data(:,4),gamma_r);
A3 = interp1(CX_data(:,1),CX_data(:,5),gamma_r);
A4 = interp1(CX_data(:,1),CX_data(:,6),gamma_r);
A5 = interp1(CX_data(:,1),CX_data(:,7),gamma_r);
A6 = interp1(CX_data(:,1),CX_data(:,8),gamma_r);

B0 = interp1(CY_data(:,1),CY_data(:,2),gamma_r);
B1 = interp1(CY_data(:,1),CY_data(:,3),gamma_r);
B2 = interp1(CY_data(:,1),CY_data(:,4),gamma_r);
B3 = interp1(CY_data(:,1),CY_data(:,5),gamma_r);
B4 = interp1(CY_data(:,1),CY_data(:,6),gamma_r);
B5 = interp1(CY_data(:,1),CY_data(:,7),gamma_r);
B6 = interp1(CY_data(:,1),CY_data(:,8),gamma_r);

C0 = interp1(CN_data(:,1),CN_data(:,2),gamma_r);
C1 = interp1(CN_data(:,1),CN_data(:,3),gamma_r);
C2 = interp1(CN_data(:,1),CN_data(:,4),gamma_r);
C3 = interp1(CN_data(:,1),CN_data(:,5),gamma_r);
C4 = interp1(CN_data(:,1),CN_data(:,6),gamma_r);
C5 = interp1(CN_data(:,1),CN_data(:,7),gamma_r);

% wind coeffisients
cx =   A0 + A1*2*A_L/L^2 + A2*2*A_T/B^2 + A3*(L/B) + A4*(S/L) + A5*(C/L) + A6*M;
cy = -(B0 + B1*2*A_L/L^2 + B2*2*A_T/B^2 + B3*(L/B) + B4*(S/L) + B5*(C/L) + B6*A_SS/A_L);
cn = -(C0 + C1*2*A_L/L^2 + C2*2*A_T/B^2 + C3*(L/B) + C4*(S/L) + C5*(C/L));

% wind forces and moment
```

```
tauX = 0.5*cx*rho_a*V_r^2*A_T;
tauY = 0.5*cy*rho_a*V_r^2*A_L;
tauN = 0.5*cn*rho_a*V_r^2*A_L*L;

tau_w = [tauX,tauY,tauN]';
```

A.2.11 Quay Force

In order to simulate the vessel hitting the quay, and thus stopping, the forces will be set to 0 if the vessel is at the given boundaries x<-88 or y>28. This is not the most ideal way to check if the vessel is at the quay, but for these simulations it sufficed.

```
function [ tau_q, nu_s] = fcn(tau_qi,x,y, nu)
if ( x < -88 || y > 28)
    tau_q = tau_qi;
    nu_s = nu;

else
    tau_q = [0 0 0]';
    nu_s = [0 0 0]';
end

end
```

A.2.12 Route Planner

The route planner is responsible for where the vessel will be moving. Three matrices as described in Section 4.7 controls where the vessel will move. In order to control the speed, a point is being placed in the direction the vessel should move. The brake value decides how far ahead of the vessel this point is being placed. A value of 100 will result in a velocity of about 1. As seen from line 37 to 51, three points are set, and the lines between them are being defined. The helping point is made on the same y-coordinate, but with a different x-coordinate than the vessel since it will make it easier to find the angle between the lines defined from the vessel to the helping point and goto-coordinate in body coordinates. On line 55, this angle is being found using Pythagoras. Since Pythagoras is used to create a triangle in order to find the angle, this can result in the wrong angle being found if the angle in reality is greater than 180 degrees. A test to see if the goto-coordinates y-coordinate is less than the vessels y-coordinate is being performed, and if this is the case then the angle will be subtracted from pi, then pi is added, changing it to the correct angle. Note, all angles are calculated in radians. When creating Polarized coordinates to control the speed, the angle is set to 0 degrees in order to easily change it to the desired angle. The distance and distanceLast variables holds the distance to the next point in the route and the last point in the route, respectively. When the vessel is within a given distance to the next point in the route, the function will start moving to the next point in the route until the last point is reached. As the ship gets closer to the final destination, it will start slowing down. This is done by shortening the distance between the vessel and the point ahead of the vessel. With the use of the command "car2pol(50,0)" a line with the x/y coordinates of (50, 0) is changed to polarized coordinates. By adding a value in radians to the THETA variable, one can rotate this line. Then, by using the pol2cart command, this rotated line is then changed back to an x/y coordinate with the appropriate rotation relative to the vessel. In order to place this point directly ahead of the vessel, one can add "yaw" to THETA. 'angleRad' in this function is the angle one need to add to THETA in order to change the rotation to point in the direction of the next destination in the route.

```
function [gotoX, gotoY, gotoYaw] = fcn( x, y, yaw)
coder.extrinsic('degtorad');

% Add values to these matrixes in order to set new points ot the route.
% These matrixes MUST be the same size.
routeX = [100 -48 -87]; %x and y are coordinates
routeY = [-60 0 27];
heading = [150 150 150]; % yaw is desired angle in degrees.
brake = 100; % Change brake distance to control speed. 20 will give a speed of 0.2 in the scope.

% Some values had to be given, or we get an error.
gotoX = 1;
gotoY = 1;

% Make sure i has a value when the function is first called.
persistent i;
if isempty(i)
    i = 1;
end

% Calculating distance between the ships position and the next desired
% point in the planned route.
distance = sqrt((routeX(1,i)-x).^2+(routeY(1,i)-y).^2);
distanceLast = sqrt((routeX(1,length(routeX))-x).^2+(routeY(1,length(routeY))-y).^2);

% If distance is less than the given value and it is not the last point in
```

```matlab
% the route, increase i by one.
if (i == (length(routeX)-1))
    if (distance < 20)
        i = i+1;
    end
elseif (distance < 20)
    if (i < length(routeX))
        i = i+1;
    end
end
%--------------------------------------------------------------------------
% Ships position
p2x = x;
p2y = y;
% goto coordinate
p1x = routeX(1,i);
p1y = routeY(1,i);
% Create a helping point.
p3x = x+60;
p3y = y;
% Define lines between ship, transsponder and helping point.
p21 = sqrt((p1x-p2x).^2 + (p1y-p2y).^2);
p23 = sqrt((p3x-p2x).^2 + (p3y-p2y).^2);
p13 = sqrt((p3x-p1x).^2 + (p3y-p1y).^2);

% create speed-controlling coordinates.
[THETA,R] = cart2pol(brake,0); % Convert to polar coordinates
% Find the angle of the goto coordinate. "p2 = skip"
angleRad = acos((p21.^2 + p23.^2 - p13.^2 )/ (2 * p21 * p23) );

% Check if angle should be greater or less than pi
if (p1y < y)% if goto y-coordinate is less than ships y-coordinate:
    angleRad = (pi - angleRad) + pi;
end
% apply the change in angle and convert back to Cartesian coordinates.
THETA2=THETA+angleRad;
[xr2,yr2] = pol2cart(THETA2,R);
%--------------------------------------------------------------------------
% Set output variables.
gotoX = routeX(1,i);
gotoY = routeY(1,i);
gotoYaw = heading(1, i);

% If the distance is larger than brake, set the goto coordinate to a set
% distance to control speed.
```

```
if (distanceLast > brake)
    gotoX = x + xr2;
    gotoY = y + yr2;
end

% ------------------------------------------
% As the ship get closer to the final destination, shorten the set distance
% in order to slow down.
if (distanceLast < 10)
    gotoX = routeX(1,i);
    gotoY = routeY(1,i);

elseif (distanceLast < 30)
    [THETA,R] = cart2pol(20,0);
    THETA4=THETA+angleRad;
    [xr3,yr3] = pol2cart(THETA4,R);
    gotoX = x + xr3;
    gotoY = y + yr3;

elseif (distanceLast < 60)
    [THETA,R] = cart2pol(30,0);
    THETA4=THETA+angleRad;
    [xr3,yr3] = pol2cart(THETA4,R);
    gotoX = x + xr3;
    gotoY = y + yr3;

elseif (distanceLast < 130)
    [THETA,R] = cart2pol(40,0);
    THETA4=THETA+angleRad;
    [xr3,yr3] = pol2cart(THETA4,R);
    gotoX = x + xr3;
    gotoY = y + yr3;

elseif (distanceLast < 175)
    [THETA,R] = cart2pol(50,0);
    THETA4=THETA+angleRad;
    [xr3,yr3] = pol2cart(THETA4,R);
    gotoX = x + xr3;
    gotoY = y + yr3;

end
end
```

A.3 Eta Deviations

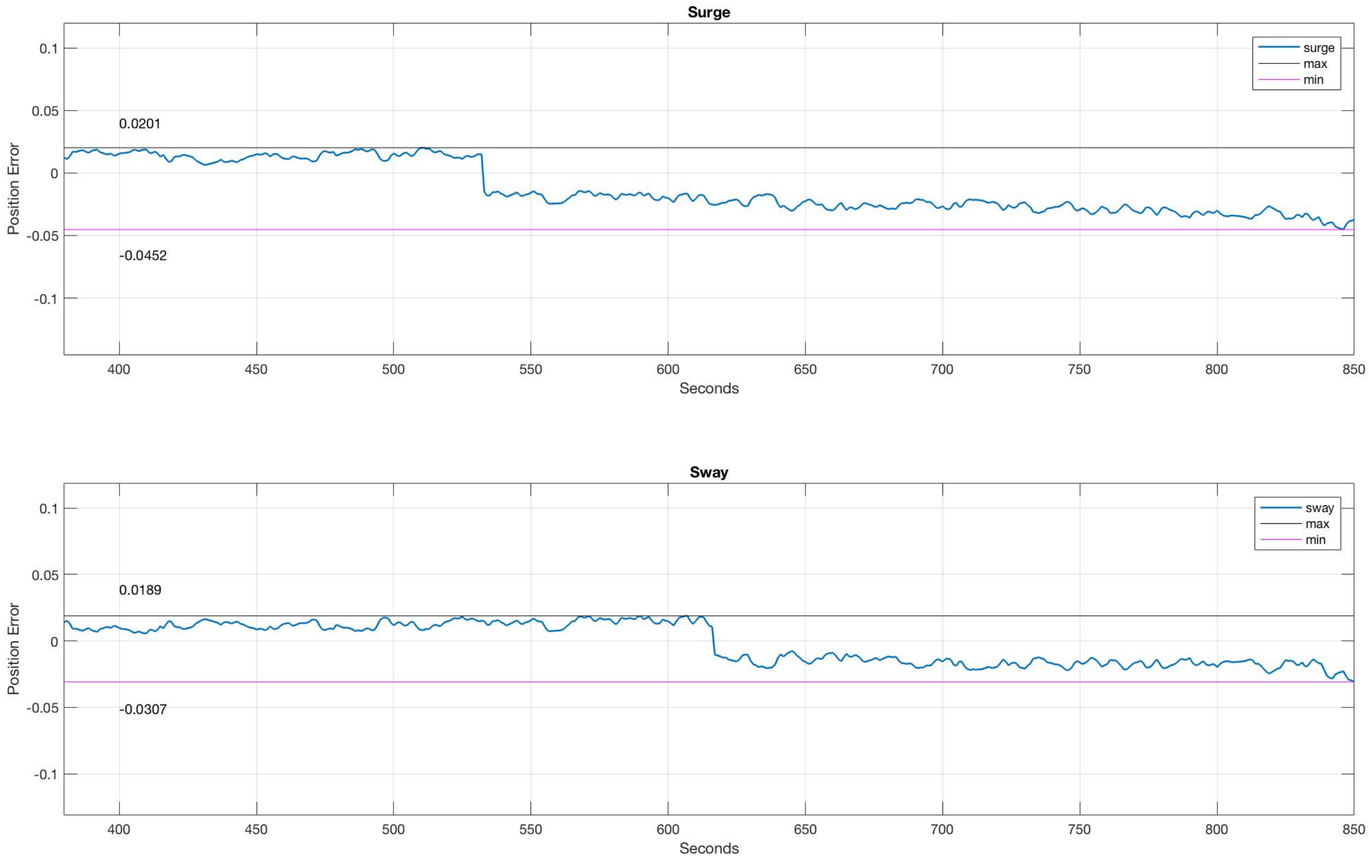


Figure A.8: Eta error. LiDAR and RTK.

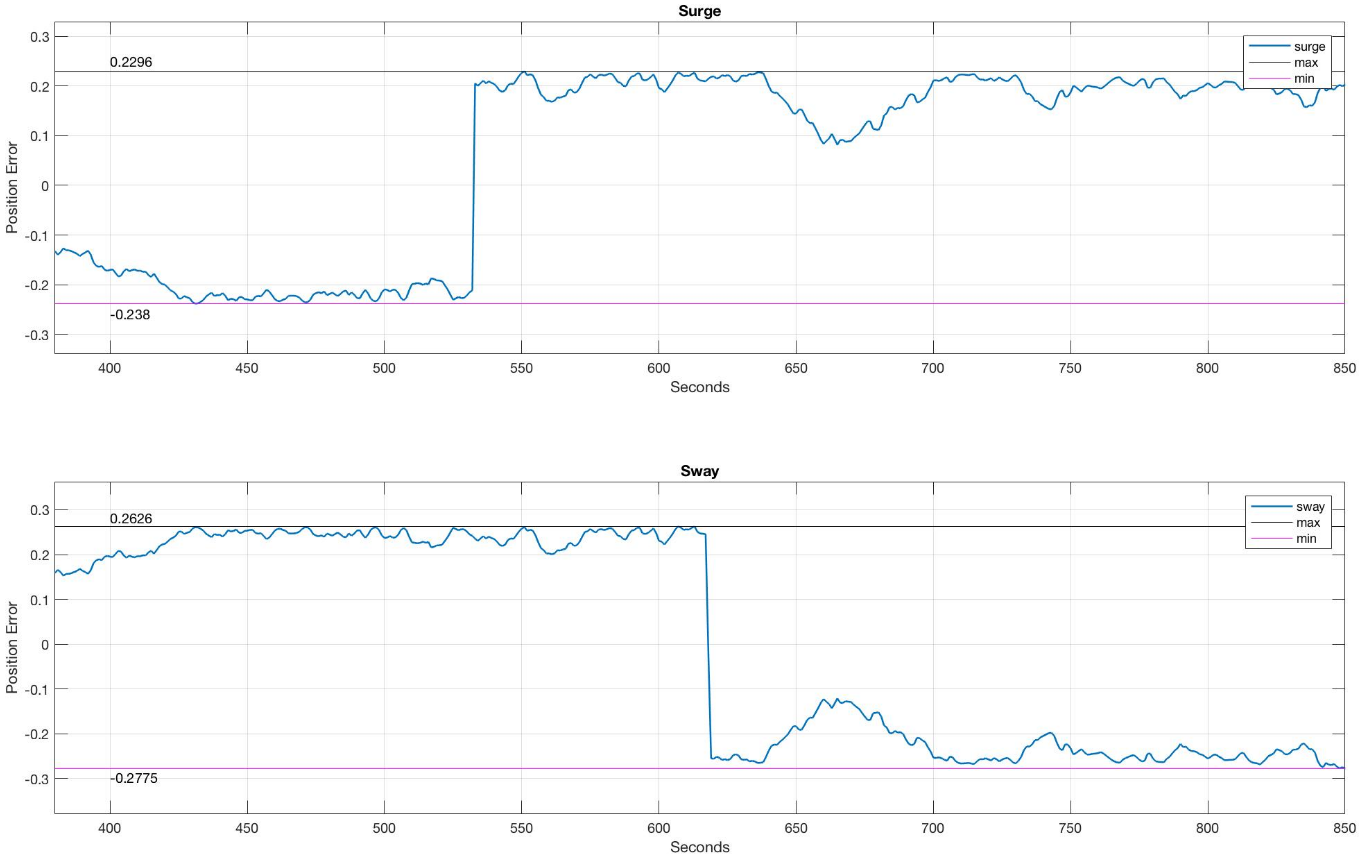


Figure A.9: Eta error. LiDAR and Galileo.

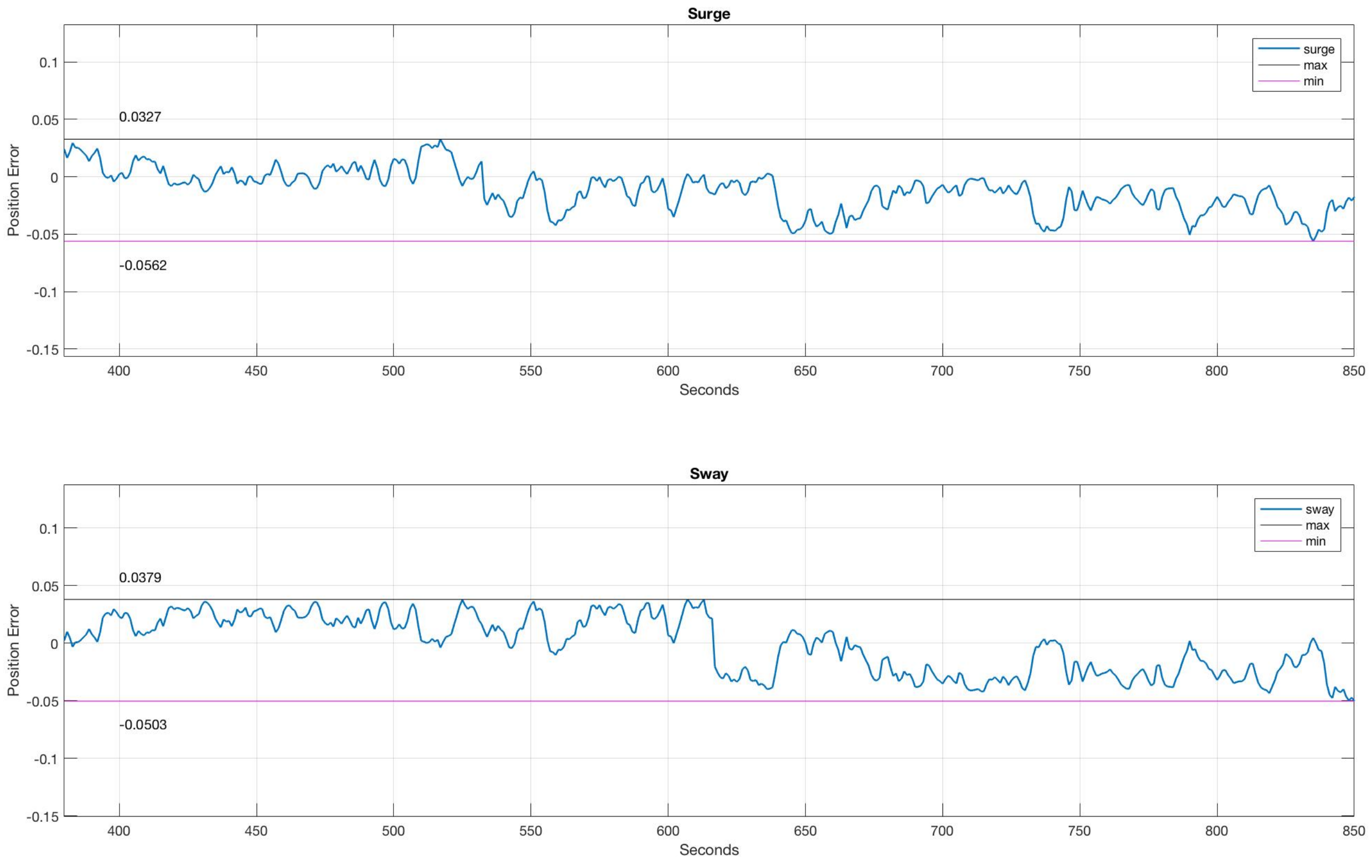


Figure A.10: Eta error. LiDAR and DGPS.

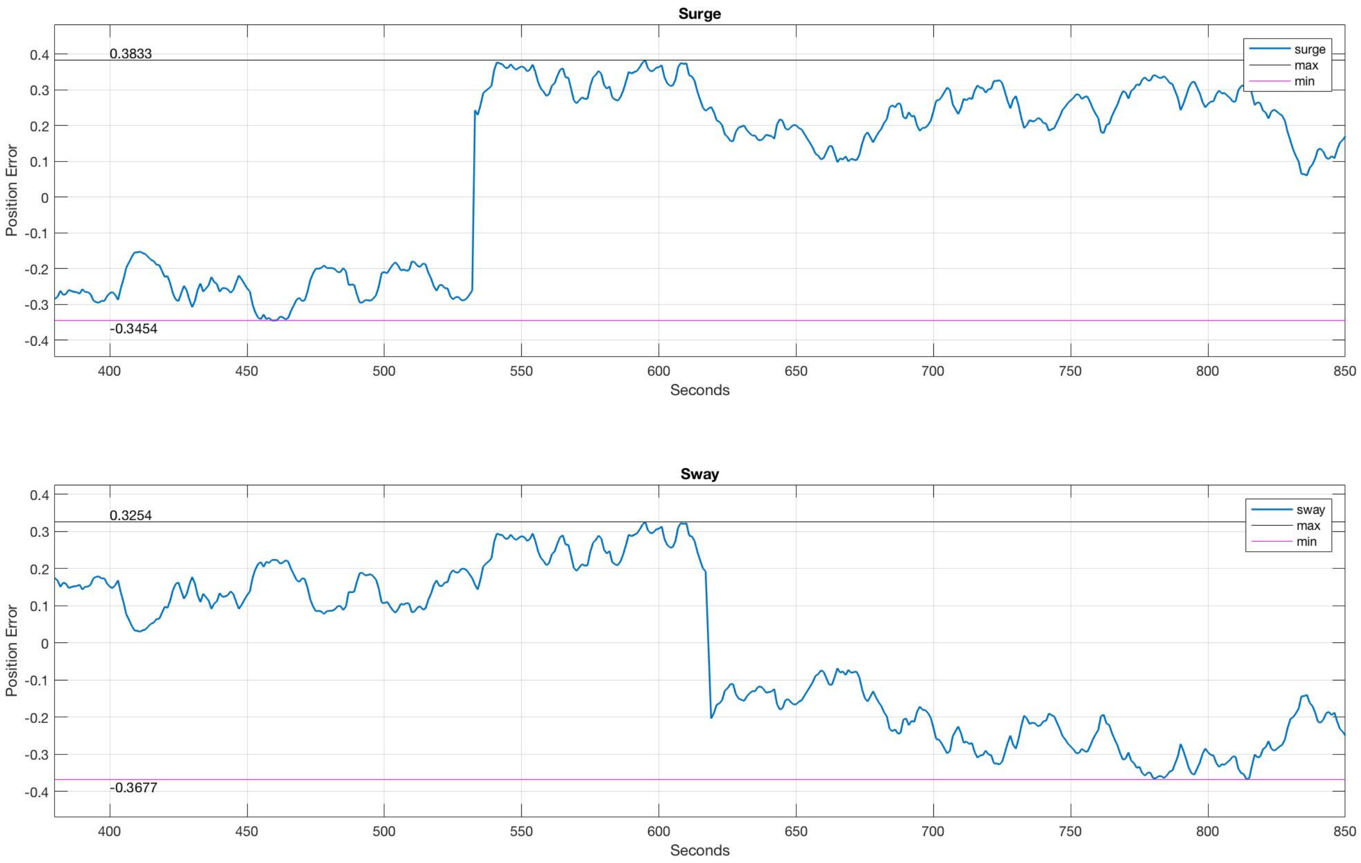


Figure A.11: Eta error. LiDAR, RADius and Galileo.

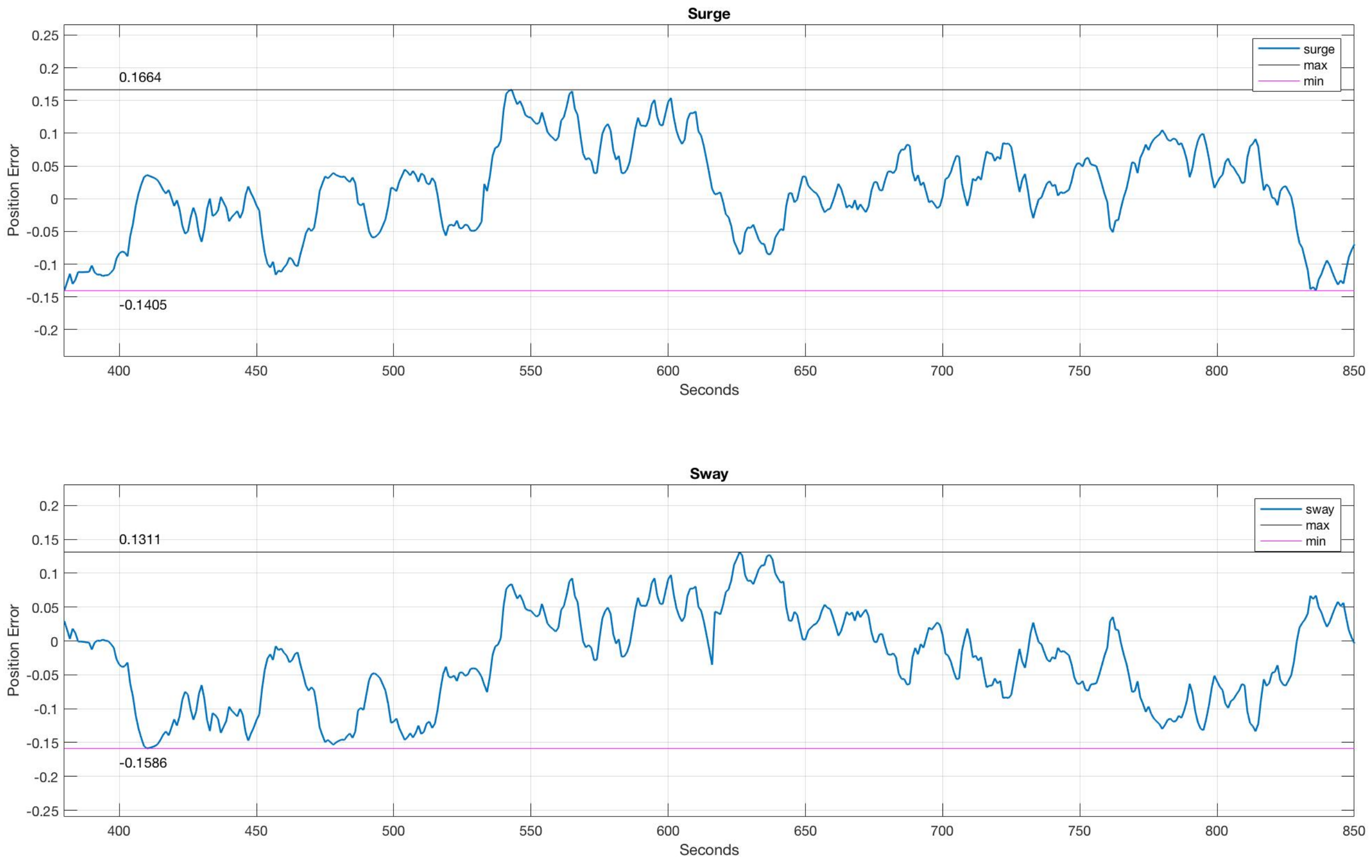


Figure A.12: Eta error. LiDAR, RADius and RTK.

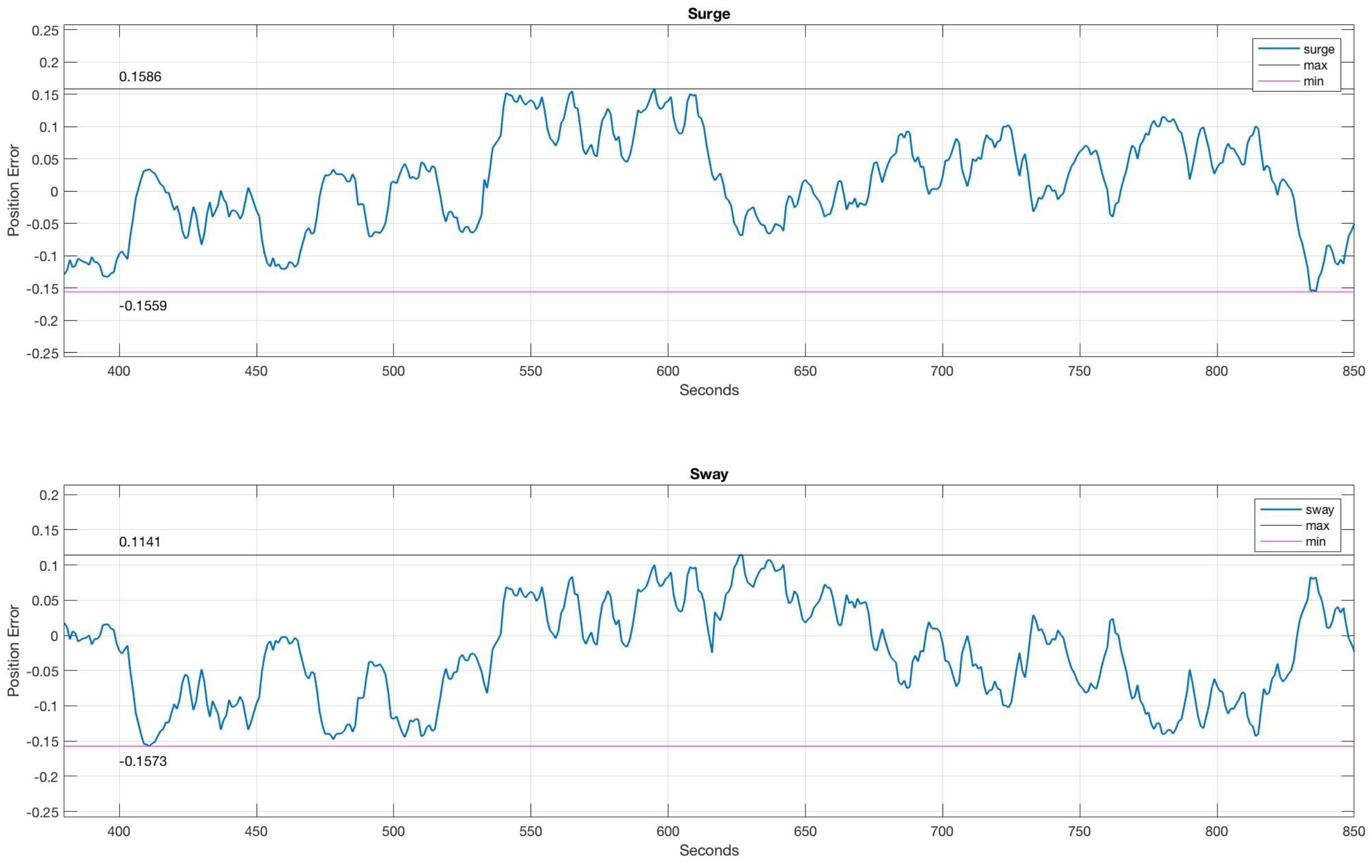


Figure A.13: Eta error. LiDAR, RADius and DGPS.

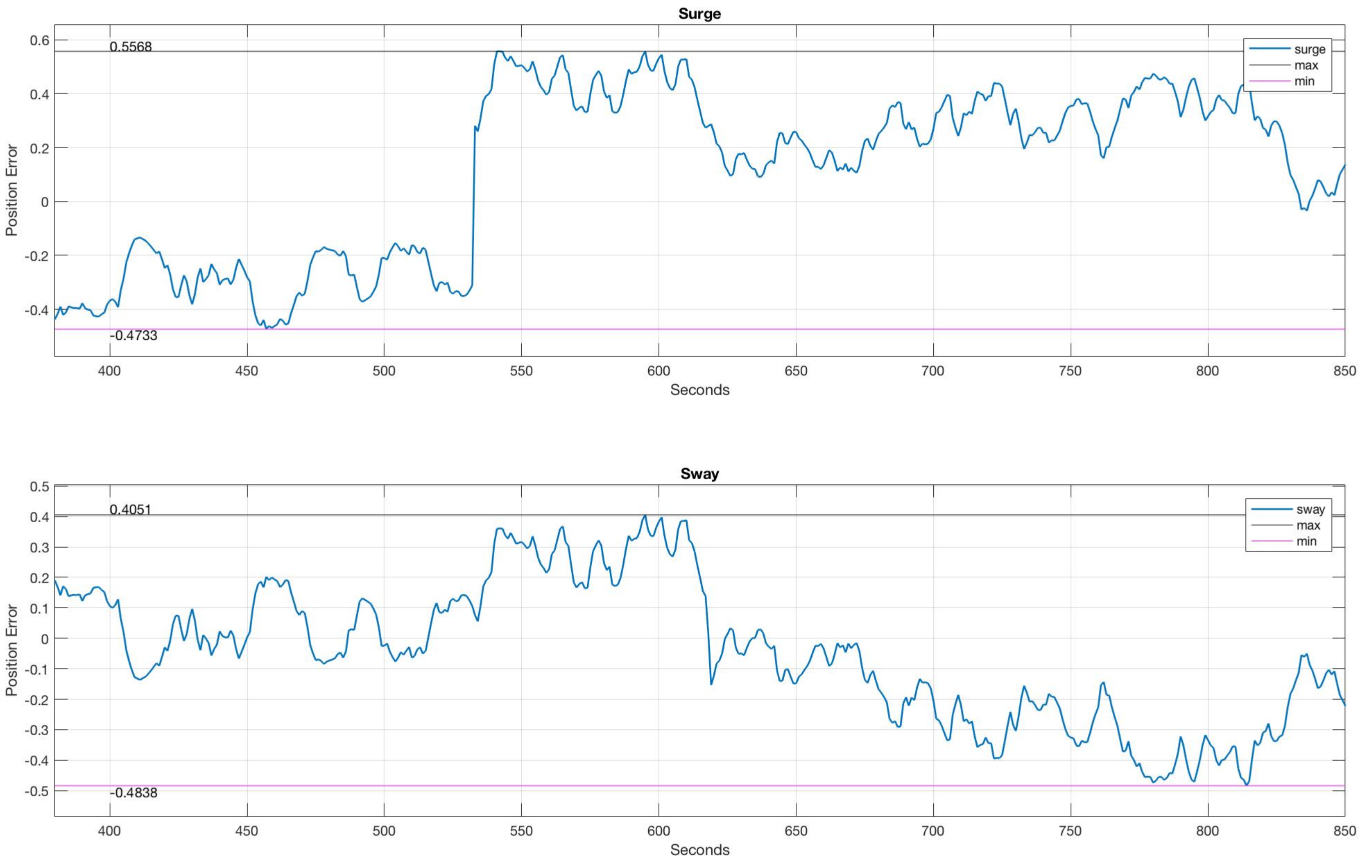


Figure A.14: Eta error. RADius and Galileo

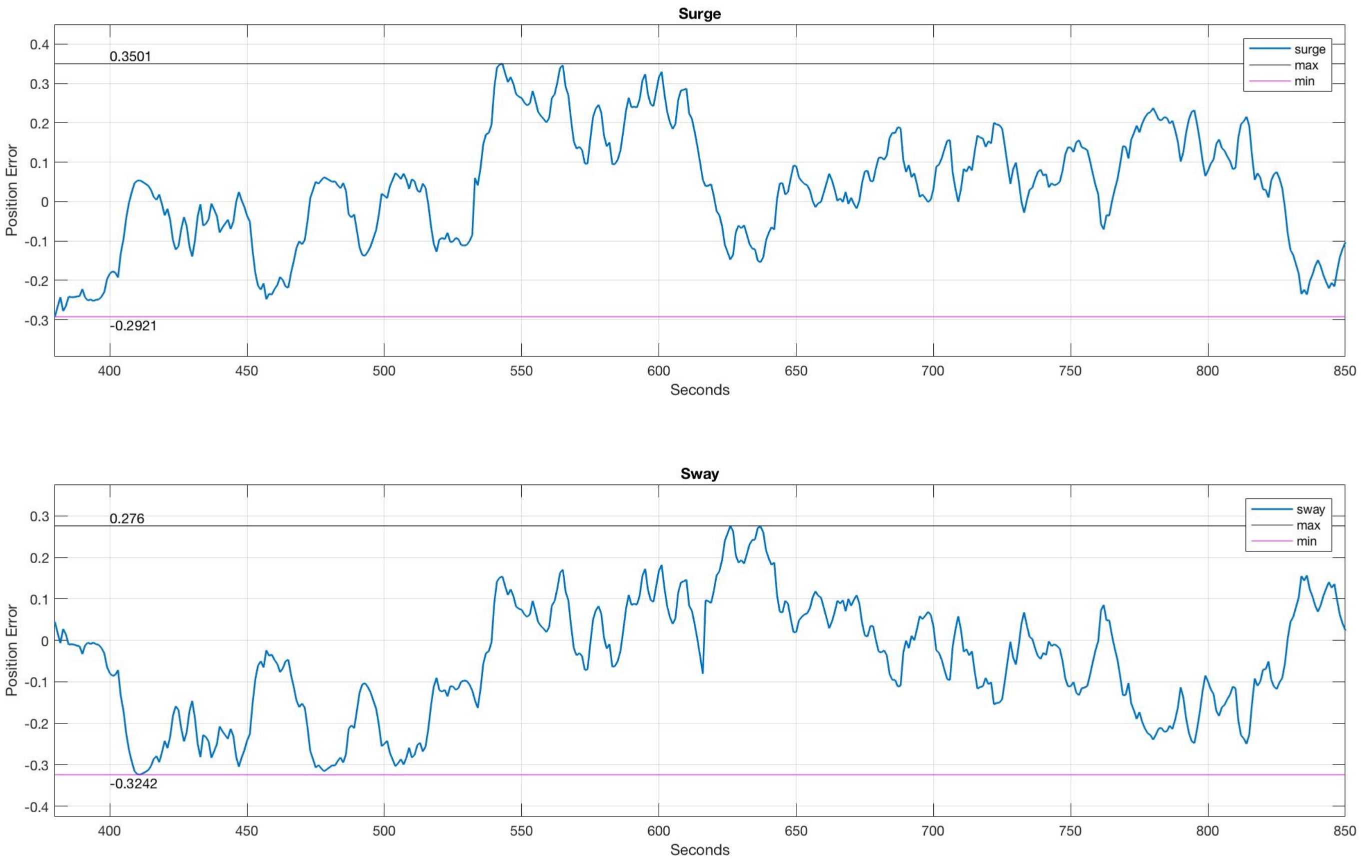


Figure A.15: Eta error. RADius and RTK.

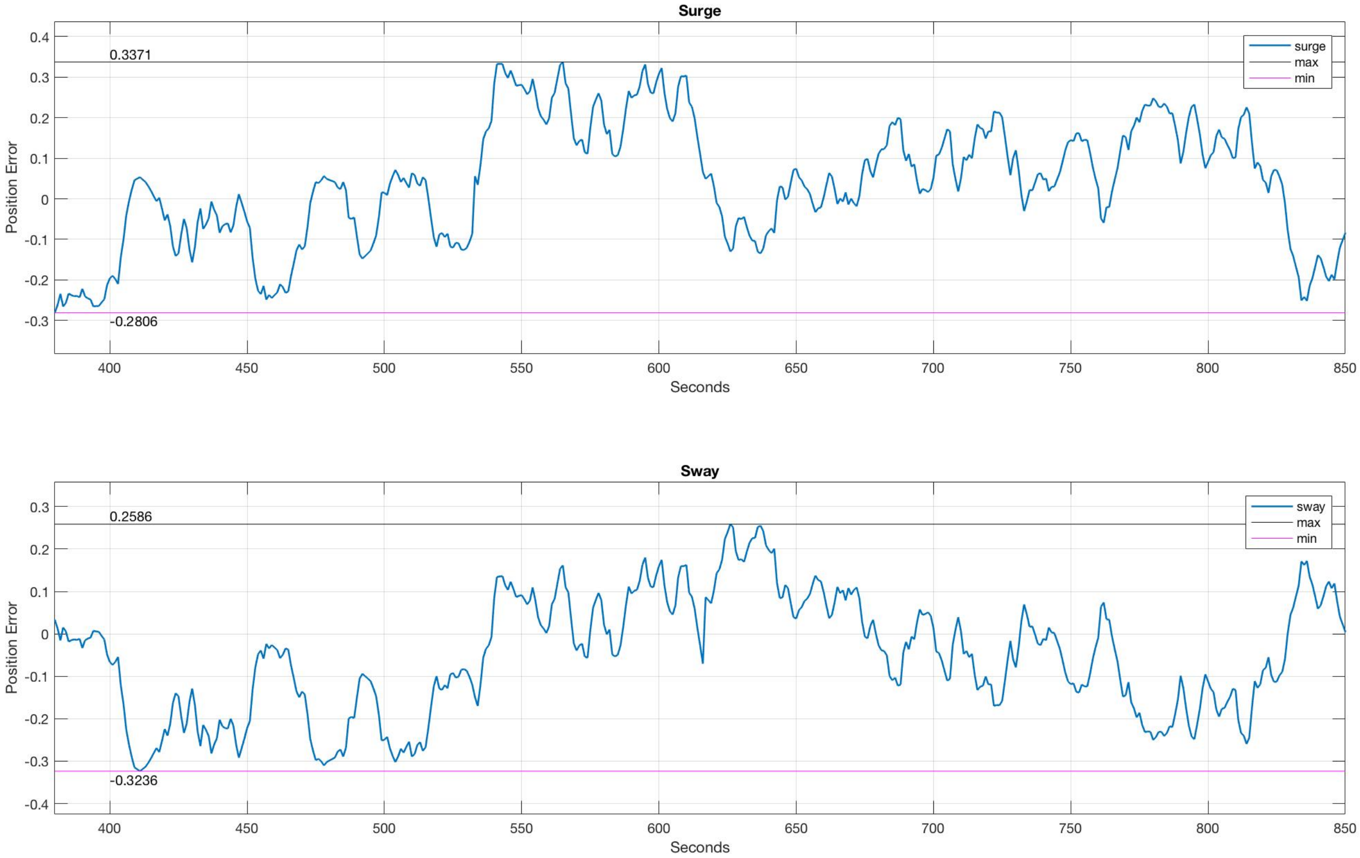


Figure A.16: Eta error. RADius and DGPS.

A.4 Weather Data

In figure 2.1-2.7 the weather data have been visualized using MatLab to make it easier to observe and understand. In some cases missing values have occurred. This happened because of miss-readings or unreliable readings from the sensors, and were therefore neglected. Where this have happened a "nan" vale would appear. For calculations on the average, these have been set to 0.

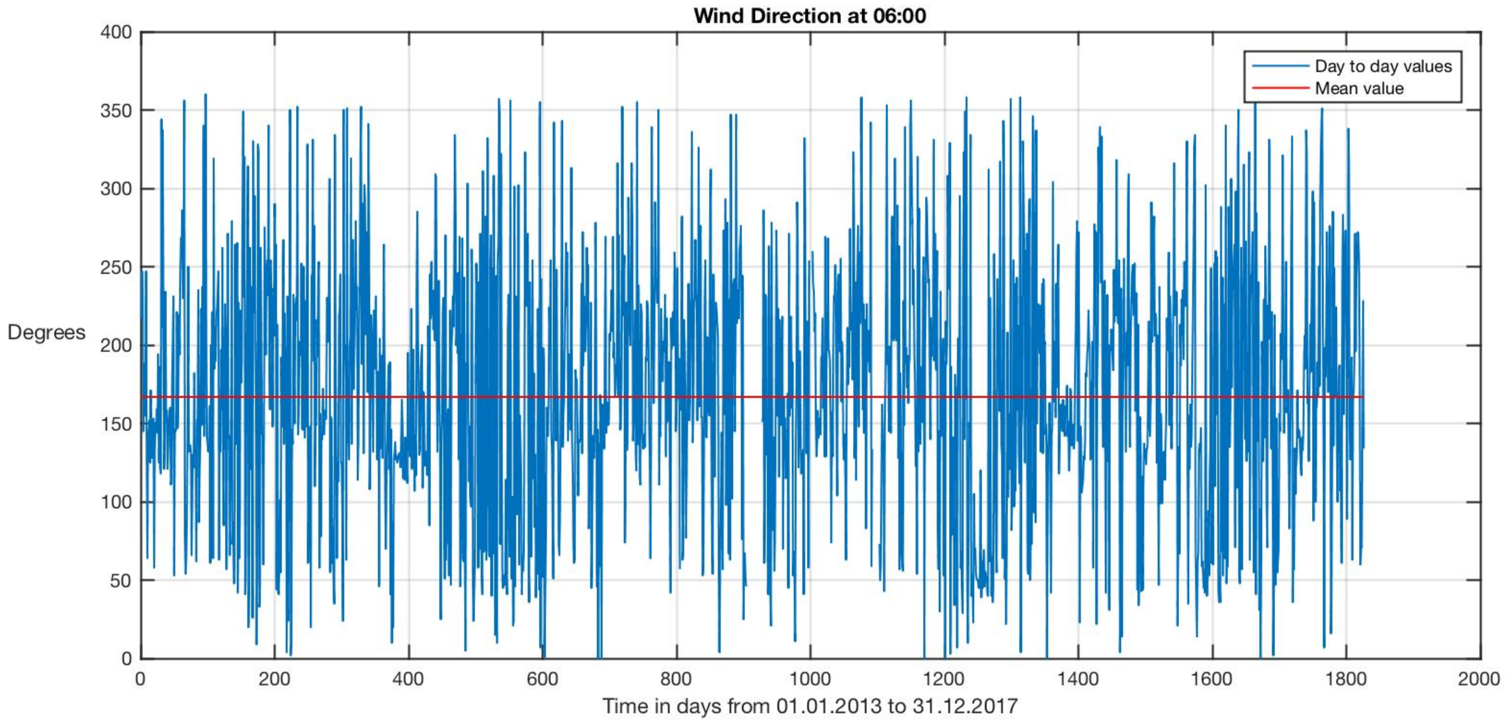


Figure A.17: Wind direction each day at 06:00.

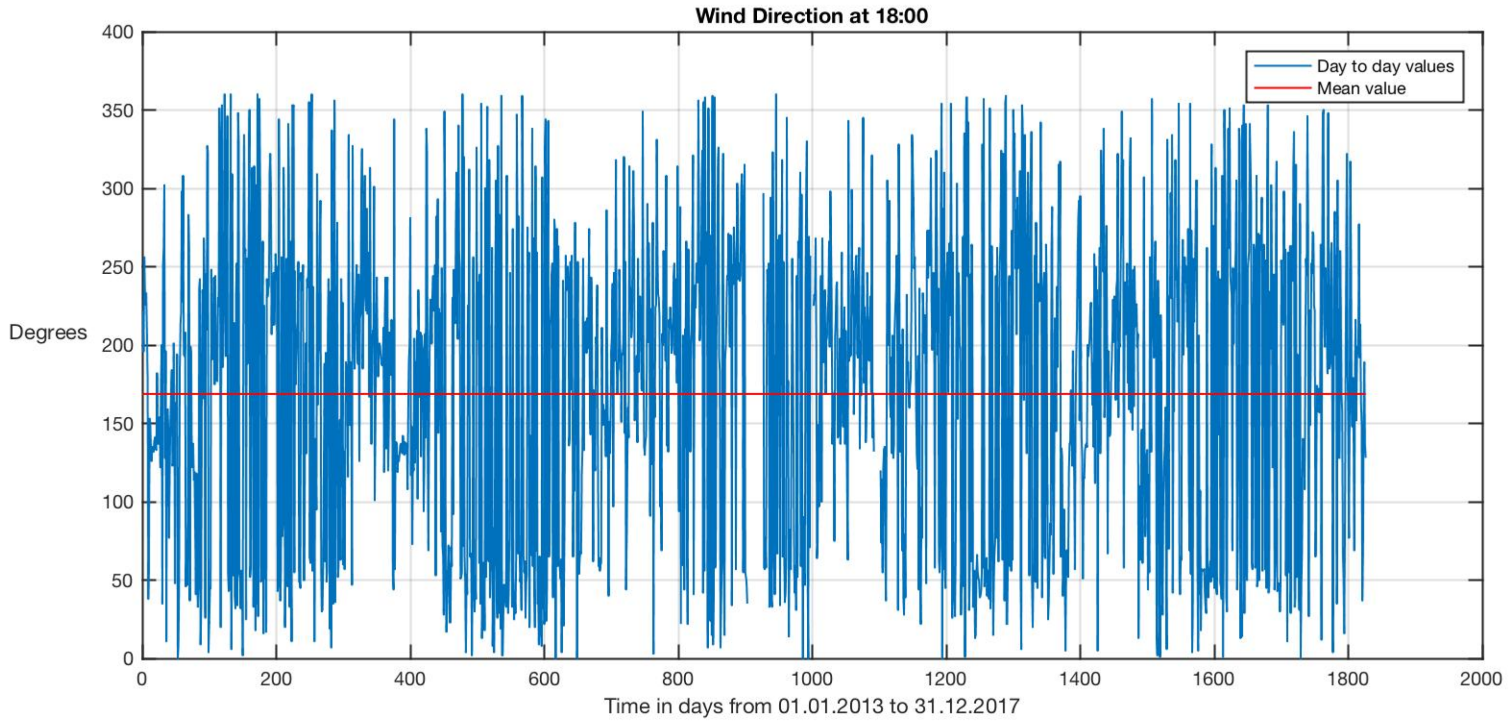


Figure A.18: Wind direction each day at 18:00.

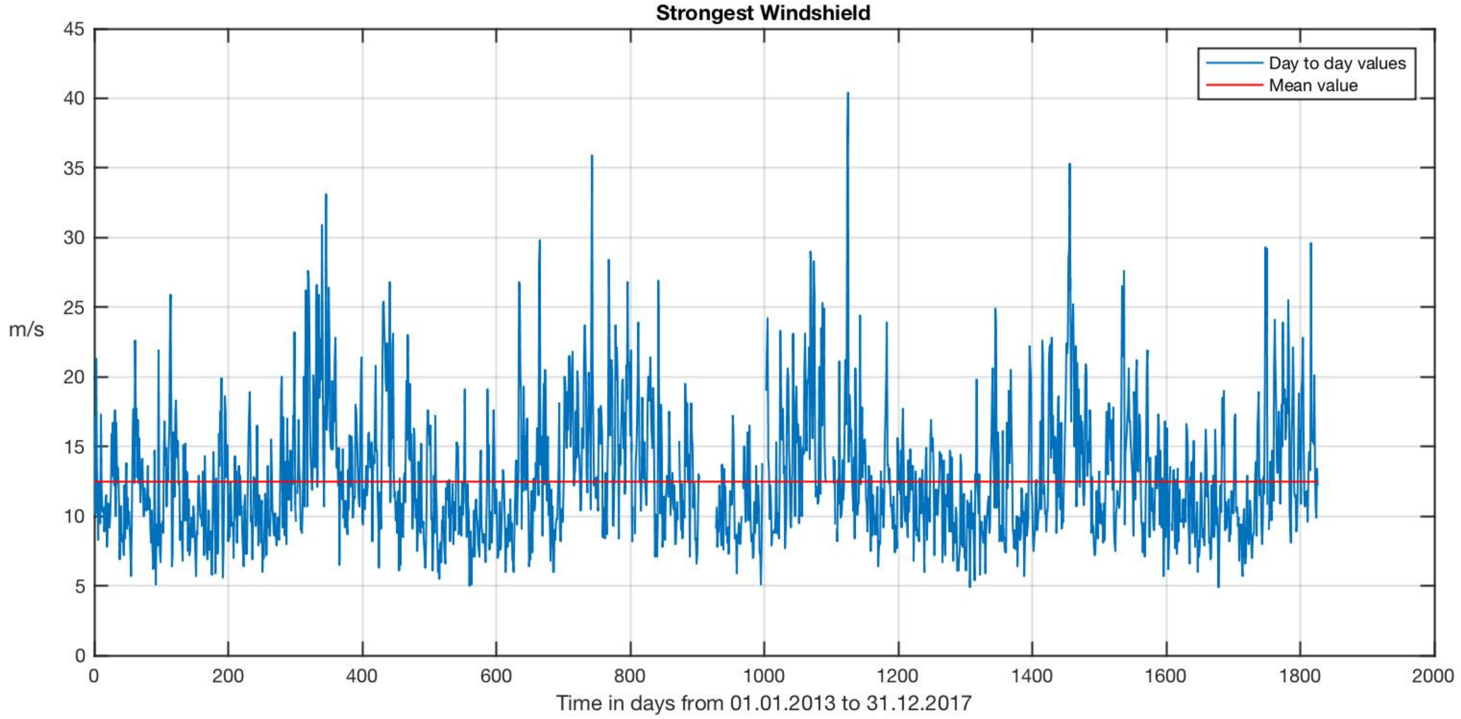


Figure A.19: Strongest windshield each day.

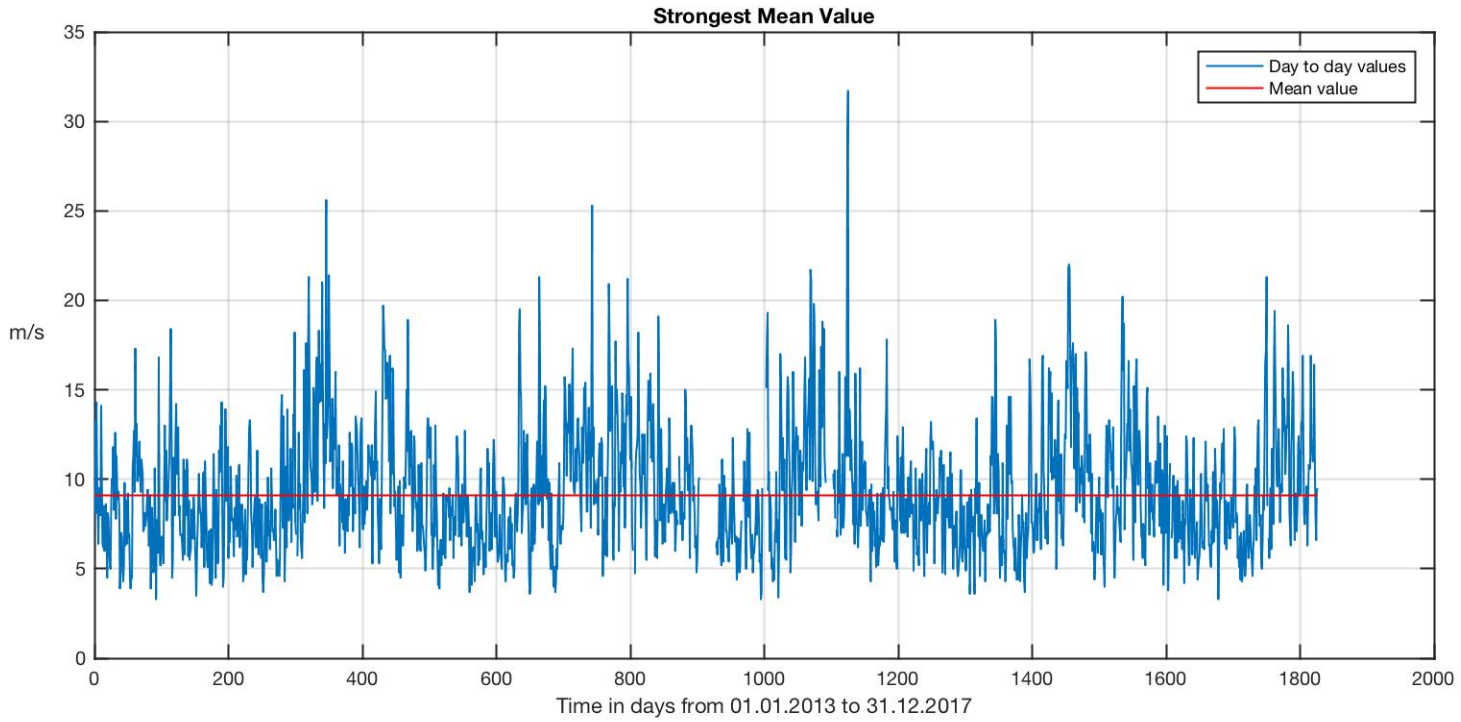


Figure A.20: Strongest mean value of wind each day.

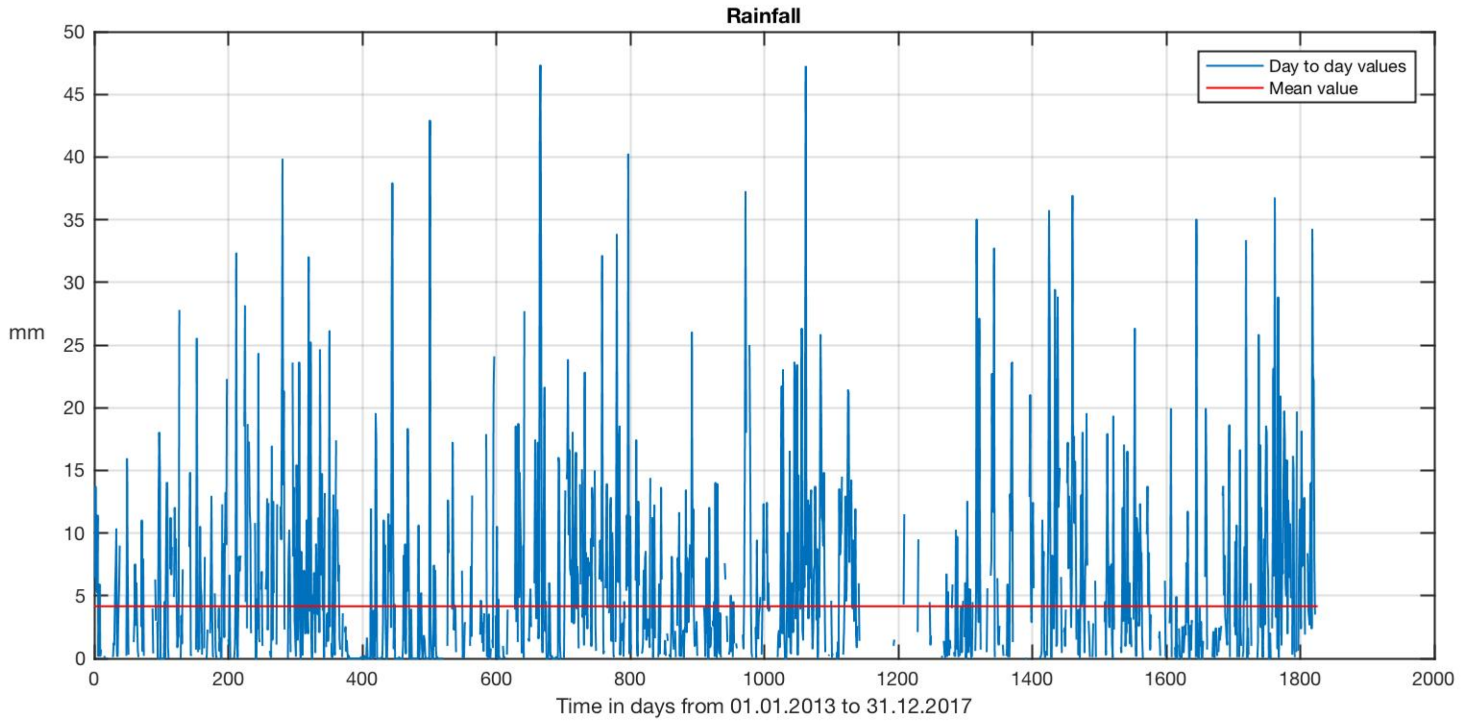


Figure A.21: Rainfall each day.

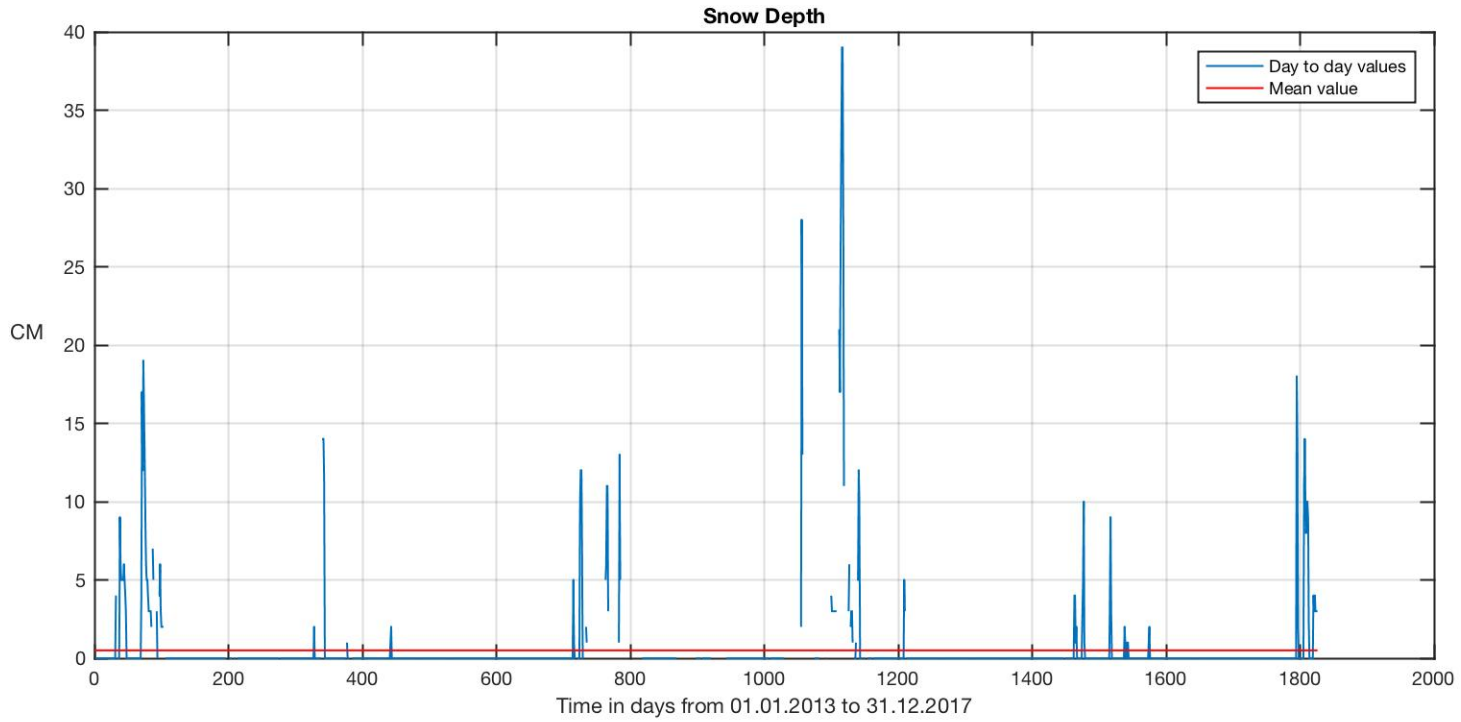


Figure A.22: Snow depth each day.

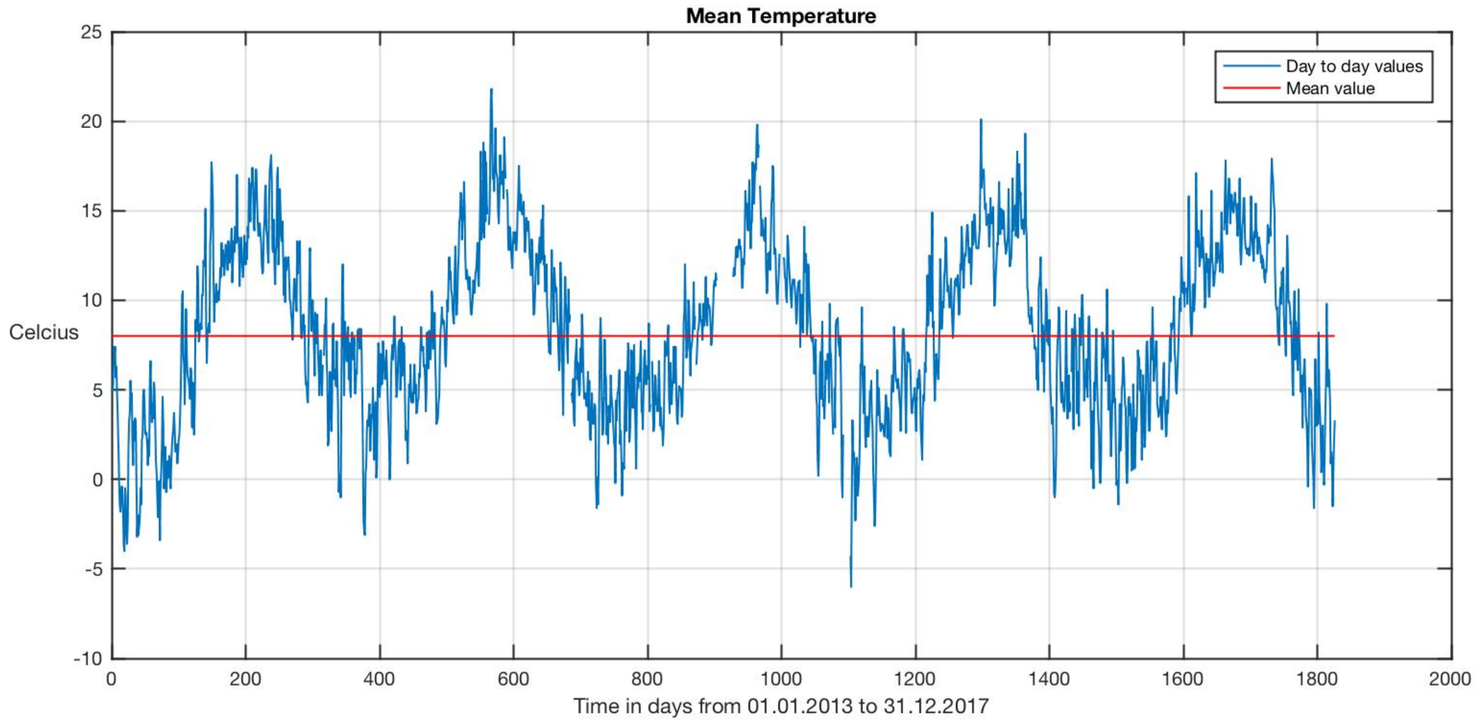


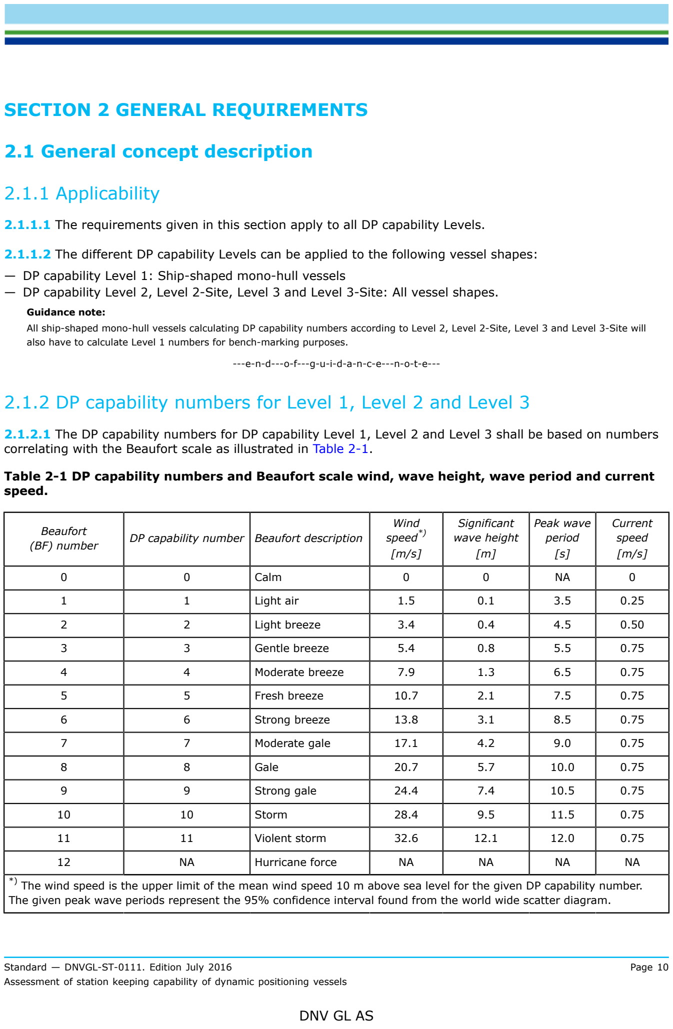
Figure A.23: Mean temperature each day.

Appendix B

Attachments

This appendix contains a table used for determining the DP capability number of the system. It also contains the preliminary report, gantt diagram and meeting minutes.

B.1 DNV GL



SECTION 2 GENERAL REQUIREMENTS

2.1 General concept description

2.1.1 Applicability

2.1.1.1 The requirements given in this section apply to all DP capability Levels.

2.1.1.2 The different DP capability Levels can be applied to the following vessel shapes:

— DP capability Level 1: Ship-shaped mono-hull vessels
— DP capability Level 2, Level 2-Site, Level 3 and Level 3-Site: All vessel shapes.

Guidance note:

All ship-shaped mono-hull vessels calculating DP capability numbers according to Level 2, Level 2-Site, Level 3 and Level 3-Site will also have to calculate Level 1 numbers for bench-marking purposes.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

2.1.2 DP capability numbers for Level 1, Level 2 and Level 3

2.1.2.1 The DP capability numbers for DP capability Level 1, Level 2 and Level 3 shall be based on numbers correlating with the Beaufort scale as illustrated in Table 2-1.

Table 2-1 DP capability numbers and Beaufort scale wind, wave height, wave period and current speed.

Beaufort (BF) number	*DP capability number*	*Beaufort description*	*Wind speed*[*)] *[m/s]*	*Significant wave height [m]*	*Peak wave period [s]*	*Current speed [m/s]*
0	0	Calm	0	0	NA	0
1	1	Light air	1.5	0.1	3.5	0.25
2	2	Light breeze	3.4	0.4	4.5	0.50
3	3	Gentle breeze	5.4	0.8	5.5	0.75
4	4	Moderate breeze	7.9	1.3	6.5	0.75
5	5	Fresh breeze	10.7	2.1	7.5	0.75
6	6	Strong breeze	13.8	3.1	8.5	0.75
7	7	Moderate gale	17.1	4.2	9.0	0.75
8	8	Gale	20.7	5.7	10.0	0.75
9	9	Strong gale	24.4	7.4	10.5	0.75
10	10	Storm	28.4	9.5	11.5	0.75
11	11	Violent storm	32.6	12.1	12.0	0.75
12	NA	Hurricane force	NA	NA	NA	NA

[*)] The wind speed is the upper limit of the mean wind speed 10 m above sea level for the given DP capability number. The given peak wave periods represent the 95% confidence interval found from the world wide scatter diagram.

Standard — DNVGL-ST-0111. Edition July 2016 Page 10
Assessment of station keeping capability of dynamic positioning vessels

DNV GL AS

B.2 Preliminary Report

FORPROSJEKT - RAPPORT

FOR BACHELOROPPGAVE

TITLE:

Preliminary report

Auto-Docking of Vessel

CANDIDATE(S):

Martin Longva
Simon Bårslett
Thor-Inge Nygård

DATE: **19.01.2018**	COURSE CODE: **IE303612**	COURSE: **Bachelor thesis**	RESTRICTION: - Open

STUDY PROGRAMME: **AUTOMATION ENGINEERING**	PAGES/ATTACHMENTS: 13 / 1	LIBRARY NO:

EMPLOYER(S)/SUPERVISOR(S):

Rolls Royce Marine/ Ottar Osen, Robin Bye

SUMMARY:

The assignment is given by Rolls Royce Marine, Auto-Docking of Vessel. The task is to find out how and what equipment/sensors that can be used for automatically dock a vessel in to a quay. The plan is to simulate how the vessel will behave with inputs from virtual sensors, and find the best solution possible.

The group is three Automation engineering students, with practical experience from the maritime industry. It's an assignment that's going to be executed by these students, with guidance from Rolls Royce and the control group from NTNU.
The project is planned to be executed using the scrum development method, where good cooperation and strict group norms is an important part of the whole project.

This report is submitted by students for evaluation at *NTNU Ålesund.*

TABLE OF CONTENTS

1 INTRODUCTION

We have chosen the Auto-Docking of Vessel assignment given by Rolls Royce Marine. The base line of the assignment is to study how to dock vessels automatically in various quay configurations. There will be a need to access how all external forces will affect the whole process, and which sensors and measurements are needed to compensate for this.
The main topics given by Rolls Royce Marine is

- Path planning for optimal docking in different weather situations
- Optimal sensor geometry to ensure maximum coverage
- Camera usage and possible image processing

2 CONCEPTS/NOTION

MSS	– Marine Systems Simulator
RR	– Rolls Royce

3 PROJECT ORGANIZATION

3.1 Project group

Student number(s)
460004 - Simon 460010 – Thor-Inge 460008 – Martin

3.1.1 Tasks for the project group – organization

Simon Bårslett – Project Manager

Martin Nikolai Longva - Secretary

Thor-Inge Nygård – Member

3.1.2 Tasks for the project manager

The project manager has the responsibility to plan the project according to the specs given by the employer, and distribute the workload evenly between the members. He also has the main responsibility for resolving conflicts if they were to occur, and follow up that a solution is reached. It is also the project managers responsibility to make sure that all members are on time for the deadline set for each part of the project.

3.1.3 Tasks for the secretary

The secretary has the main responsibility to assist the project manager in different administrative tasks. He is also responsible for informing the project manager, the employer and the control group about the progression of the project. Another important task is to make sure that the project is documented thoroughly through the whole process.

3.1.4 Tasks for other member(s)

Main responsibility for programming, and assist the project manager and secretary when needed. And collect the necessary external data.

3.2 Control group (Supervisor and Employer)

Ottar Osen

Robin Bye

Rolls Royce Marine

4 AGREEMENTS

4.1 Agreement with employer

On 15.01.2018 the group and the supervisors had a meeting with the employer where all parties sat down and discussed the project work and defined the assignment.

The employer wants the group to focus on sensors, primarily with ferries in mind. They already have a project on the subject, but wish for someone to look at the case with “fresh eyes”. The group are given freedom to choose how they go through with the task, to either look thoroughly at few sensors or to look a little more lightly on several sensors. Figure out the pro’s and con’s, look at how weather can affect the process of docking the vessel and at least briefly look at the cost for the different solutions and keep them as low as possible.

The employer wants the group to simulate the different solutions, and compare them to each other.

4.2 Workplace and resources

Workplace:

The school area will be used as workplace, and room L167 will primarily be used for this project.

Resources:

Matlab will be the main tool used for simulation and calculations. The MSS toolbox in Matlab will primarily be utilized in this thesis. MSS uses Simulink to simulate different sensors and input to the vessel.

Access to people:

Ottar Osen

Robin Bye

Rolls Royce Marine/Solveig Bjørneset

Data security/classified information:

At this time, there is no classified information involved in the project. Primarily it will be shared between the control group and the employer during the project.

Scheduled reports:

There should be progress reports about every fortnight.

4.3 Group norms – cooperation rules – countenance

For a project like this to work, strict and fair group norms should be set. These should be decided together, and all members should agree on these. For starters, a set work time is quite important. This is so all members can be a part of what is going on, and work together when needed. It will make the whole process more effective.

Since this is a group project, all members will gather a lot of information from different places. All relevant documents that is needed throughout the project should be kept and uploaded to a place where everyone can access it. This will make it easier for completing the final report/documentation, and it may explain to other members why certain decisions were made.

- Working hours will be between 08:15 and 16:00. If one of the members cannot hold up the set working hours, all the other members has to be notified.
- If any big changes occur in the project, all members have to be notified.
- The main report should be updated evenly through the project

This is norms and manners important for a successful cooperation between colleagues/students in an engineering project.

5 PROJECT DESCRIPTION

5.1 Issue- objective - purpose

Issue: The everyday life of the average human depends on ships. Ships transport materials, goods and passengers. When a ship is involved in an accident it is expensive, could be hazardous for the environment and a danger to human life. A lot of fuel (or energy, if electrical) is used to operate larger vessels. These things taken into account, it is important that ships are as safe and cost effective as possible. One issue to be addressed is the docking of vessels, an automatic docking system could be the answer. This eliminates the risk of human error and could make the process a lot more effective. The process of docking a vessel automatically comes with a series of challenges. One is to know when the vessel is docked to the quay, another is how to handle the approach to the quay during rough conditions.
The weather conditions can affect sensor data and how the vessel behaves.

Objective: The objective of this project is to research a variety of sensors which could be used for automatic docking of a vessel, primarily ferries. They need to compensate for external forces in almost all possible scenarios. The final result should provide information about a combination of sensors and equipment that will perform an automatic docking process seen from a theoretical perspective.

Purpose: The purpose of the project is to research what kind of sensors can achieve a satisfying result with all demands fulfilled, and finally utilized in the process of automatically docking a ferry.

5.2 Requirements for solution or project result - Specification

The employer does not expect a physical prototype showing that auto docking can be done. Instead they want simulations and documentation for what kind of sensors and equipment that might work well for the process of docking a vessel automatically. The project is expected to be well documented.

5.3 Planned procedure for development - method

The first step in this project is to start investigating what sensors, measurements and data is needed. Since this is a fresh subject for the group, the development process for the final solution will be based on a trial and error process. One of the drawbacks with this method is that it is time consuming and may produce a series of bad solutions. In a worst-case scenario, the group will not come up with a usable solution at all. The plan is to consult with Rolls Royce through the process and see if they have any experience with the ideas the group comes up with.

The general development method will be the scrum method. Where the members of the group will manage themselves, but the project manager/scrum master will have the responsibility to follow up that the given tasks is completed within the deadline set.

5.4 Information collection – done and planned

Information and data will be the foundation of the decisions made during this project. There will be a need to access the accuracy and relevancy of every piece of information collected. Some of the points of information needed for this project.

- The group plans to collect data about technology already in use, and research what other sensors might be useable for safe and effective auto-docking of a vessel.
- There will be a need for weather data around the quay in Hareid, and maybe on Sulesund.
- Different possible areas to place sensors on the ferry and on land will be researched.
- Different types of GNSS.

Mainly information and data about a given product should be collected from the producer or a trustable source.

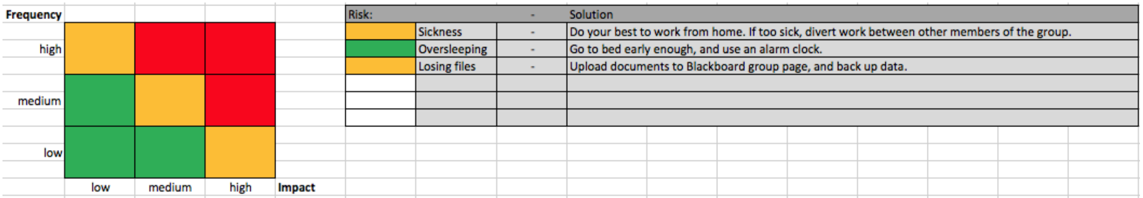
5.5 Assessment – risk analysis

In order to succeed in completing this project, it is important that the group can stay in the same room when the project is under development, which will provide better communication. This will help keeping daily meetings between each group member easier and make sure that everyone knows what to do from day to day. Bad communication can lead to reduced efficiency and ensure that daily targets are not completed.

In the risk chart (Figure 1) there are some elements that may arise through the projects development. It is provided an indication of the likelihood that an incident may occur and how much it effects the progress.

The risk chart is divided into frequency and impact that something can occur. Where some of the risks are acceptable for the project and some are fatal for further development.

- Green is a risk that is not than common, and the impact is small.
- Yellow are medium risks that may have a larger impact on the project.
- Red are risks that may have a devastating consequence for the project.



Frequency				
high	🟨	🟥	🟥	
medium	🟩	🟨	🟥	
low	🟩	🟩	🟨	
	low	medium	high	Impact

Risk:		-	Solution
🟨	Sickness	-	Do your best to work from home. If too sick, divert work between other members of the group.
🟩	Oversleeping	-	Go to bed early enough, and use an alarm clock.
🟨	Losing files	-	Upload documents to Blackboard group page, and back up data.

Figure 1: Risk chart for execution of the project

5.6 Main activities in further work

The tasks set up for each member of the group stretches over different time periods. From one day to three weeks. It all depends on the magnitude of the task itself. Research and testing is a big part of this project, but the final solution should be finished by the fourth of May.

The final report should be finished by first of June, which is the largest part of the project. That is the reason it stretches over the whole project.

Since the project is based on research and testing through simulation, no financial plan is needed. At this time there is no need for any equipment, only the data from the equipment is needed.

5.7 Progress plan – management of the project

5.7.1 Main objective

The plan for the project is to design a solution that has its origin in the ferry crossing between Hareid and Sulesund in Norway. It is going to be an agile solution, which can be implemented with other quay setups. The first step will be to setup a simulation environment and make a model of a typical ferry. Next step will be to find the best solutions when it comes to safety, functionality, cost and how it can be implemented in a typical ferry. Since this is a scientific project were the goal is to find the most cost effective and functional solution, the end result for the project is hard to predict. But the goal is defiantly to come up with a suggestion that satisfies the employers request. The decision-making process will mainly be between the group members and Rolls Royce with inputs from the supervisors.

Final testing should begin early April, and a final solution at the beginning of May. The final report should be finished by first of June.

5.7.2 Management utilities

Management utilities which will be utilized in this project is Asana, and Instagantt. These tools can be used to map, plan and keep up with task deadlines. They provide a visual representation of the planned tasks, with colors indicating if tasks are in route, finished, or past deadline. It can sometimes be hard to plan exactly when a task will be done. Sometimes a task could be finished way ahead of planned deadline and other times, a planned task may take longer to finish than anticipated. A visual representation of our deadline status will then be easier to acknowledge than if they are just typed down dates on a paper, or verbally planned.

5.7.3 Development Utilities

The employer suggested to use Maritime Systems Simulator.

This is a toolbox for Matlab which can be used with Simulink to simulate a vessel with sensors, and input to the vessel determined by the sensors (Fossen & Perez, 2004) . The group can use this toolbox to explore if and how different sensors work with the vessel. With MSS, the group can choose whether or not the sensors should have any form of noise. In the beginning, no noise might be best just to test if a solution could work. If it doesn't work without noise, it will most likely fail with noise.

5.7.4 Internal control - evaluation

Whether or not a task is complete must be discussed within the group. Only when the entire group agrees that a task is done may it be marked as finished. The criteria for a task to be deemed finished is when the group feels that the task have been completed in an acceptable manner, documented well and a conclusion for the task have been made. It should be well documented what was done and why the conclusion for the specific task was made. Even if the task is deemed not feasible, documentation on why that conclusion were made should be written.

5.8 Decisions – decision making

For mayor decisions, the group should manage to come to an agreement of sorts. Should it however appear a decision the group cannot agree on, the majority of the group decides what choice should be made. If no one in the group agrees, a consultation with the control group is needed.

6 DOCUMENTATION

6.1 Reports and technical documents

The project should be well documented. All test results and findings should be documented. The group should document evenly throughout the project. During the project, the final report will be kept online using ShareLaTeX.

7 SCHEDULED MEETINGS AND REPORTS

7.1 Meetings

7.1.1 Meetings with control group

There should be a meeting between the project group and the control group approximately every fortnight. The employer may also participate if possible. Before every meeting, a short summary should be written to fully inform the control group on how fare the project has come. The project group will prepare questions if there are any for the other participants before each meeting.

7.1.2 Project meetings

The project group will have a meeting every Monday morning for discussing status, progress and eventual obstacles that may have occurred. This is to have an overview of what has been done, and what is the next step in the process. If there are any obstacles the whole group needs to discuss it and try to find a solution for the problem.

7.2 Periodic reports

7.2.1 Progress reports

Every other week a progress report will be written before meeting with the control group. The progress report must include all changes done since last meeting. It will also contain if a task is finished. The secretary will take responsibility for this to be done in time.

8 PLANNED DEVIATION TREATMENT

In the case of a deviation from the planned progress, the members of the group should consult each other and decide how to proceed to fix this. The group must ask themselves if the deviation is unavoidable or if it could be handled immediately. If the member responsible for a task have too much to do at a given time and it fails, it should be considered that another member takes over the given task.

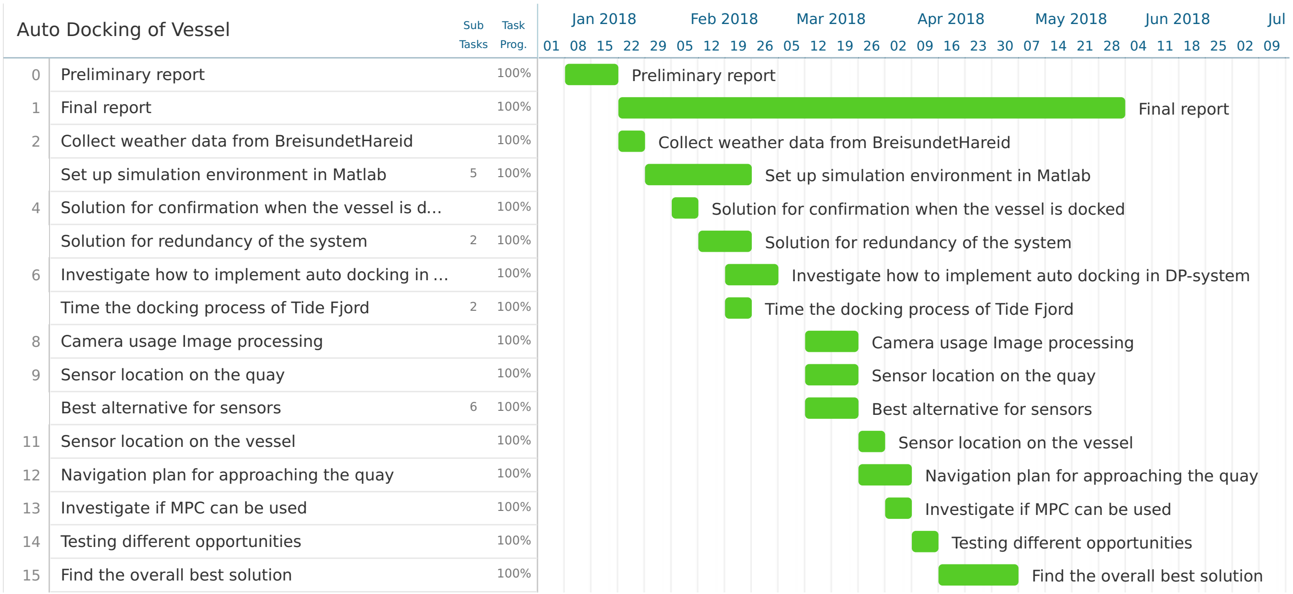
9 EQUIPMENT REQUIREMENTS/PREREQUISITES FOR IMPLEMENTATION

At this time of the project there is no need for any special equipment or software.

10 REFERENCES

Danielsen, A. L., 2003. *Mathworks.* [Internett]
Available at: https://se.mathworks.com/matlabcentral/fileexchange/4217-the-marine-visualization-toolbox
[Funnet 15 January 2018].
Fossen, T. & Perez, T., 2004. *Marinecontrol.* [Internett]
Available at: http://www.marinecontrol.org
[Funnet 15 January 2018].

B.3 GANTT Diagram



Auto Docking of Vessel

		Sub Tasks	Task Prog.
0	Preliminary report		100%
1	Final report		100%
2	Collect weather data from BreisundetHareid		100%
	Set up simulation environment in Matlab	5	100%
4	Solution for confirmation when the vessel is d...		100%
	Solution for redundancy of the system	2	100%
6	Investigate how to implement auto docking in ...		100%
	Time the docking process of Tide Fjord	2	100%
8	Camera usage Image processing		100%
9	Sensor location on the quay		100%
	Best alternative for sensors	6	100%
11	Sensor location on the vessel		100%
12	Navigation plan for approaching the quay		100%
13	Investigate if MPC can be used		100%
14	Testing different opportunities		100%
15	Find the overall best solution		100%

Jan 2018
Feb 2018
Mar 2018
Apr 2018
May 2018
Jun 2018
Jul
01 08 15 22 29 05 12 19 26 05 12 19 26 02 09 16 23 30 07 14 21 28 04 11 18 25 02 09
Preliminary report
Final report
Collect weather data from BreisundetHareid
Set up simulation environment in Matlab
Solution for confirmation when the vessel is docked
Solution for redundancy of the system
Investigate how to implement auto docking in DP-system
Time the docking process of Tide Fjord
Camera usage Image processing
Sensor location on the quay
Best alternative for sensors
Sensor location on the vessel
Navigation plan for approaching the quay
Investigate if MPC can be used
Testing different opportunities
Find the overall best solution

B.4 Meeting Minutes

Project Meeting

Minutes for January 15, 2018

Present: Simon Vågeskar Bårslett (Chair), Martin Nikolai Longva, Thor-Inge Eidsvik Nygård, Ottar Laurits Osen, Robin Trulssen Bye, Solveig Bjørneset

Agenda

1. Discuss what sensor that may be useful.
2. Define the objective of the project.

Discussion

We had a meeting to help us define the project, and also get some input from Rolls Royce to find out what they wanted out of the assignment. We agreed to firstly focus on making a simple solution with an ideal situation without any wind or waves, and after that we could start making the problem more advanced. At first we will base our solution on a common ferry that has two destinations where we will focus on one of these. That makes it possible for us to maybe place sensors on the quey if that comes in to consideration.

As of sensor we got a lot of good suggestions, the main ones that we will focus on are laser sensors like LiDAR and others, the use of camera and GNSS/RTK for positioning of the vessel.

Simulation of the project was also something that we had questions about, Rolls Royce gave us some suggestions on some of the toolboxes in Matlab that they use. Where MSS(marine systems simulator) is the one that we will start looking into first, the other one is MVT(Marine Visualization Toolbox) witch may be used later in the project.

New Business

1. We will start looking into the new toolboxes in Matlab.
2. We will start researching sensors that may be of use.

Next Meeting: Tuesday, January 31, at 09:15

Project Meeting

Minutes for January 31, 2018

Present: Simon Vågeskar Bårslett (Chair), Martin Nikolai Longva, Thor-Inge Eidsvik Nygård, Ottar Laurits Osen, Robin Trulssen Bye, Solveig Bjørneset

Agenda

1. Introduction to Marine Systems Simulator.
2. Discuss more sensors that may be of use.

Discussion

We asked Solveig for an introduction in how to set up a simulation in Simulink, and how to use the MSS toolbox. She also gave an introduction on how thruster and weather forces will affect the vessel.

Solveig told us about a project in Finland called Advanced Autonomous Waterborne Applications. Rolls-Royce together with the other partners in the AAWA project, DNV GL, Inmarsat, Deltamarin, NAPA, Brighthouse Intelligence, Finferries and ESL Shipping - and with the support of Tekes Rolls- Royce, works together to produce the specification and preliminary designs for the next generation of advanced ship solutions. This is a very exiting project that we will look more into.

As of new sensors and navigation possibilities Ottar mentioned wifi mapping. This sounds like something that may work in our case, and we will research how it works and how relevant it is in our case. SLAM(simontanious localization and mapping) is used for constructing or updating a map of an unknown environment while simultaneously keeping track of location within it. we are not sure if this is something that we may use but we will look into it.

New Business

1. We will start making a simulation environment in Simulink.
2. We will do research on alternative sensors.

Next Meeting: Tuesday, February 13, at 13:30

Project Meeting

Minutes for February 13, 2018

Present: Simon Vågeskar Bårslett (Chair), Martin Nikolai Longva, Thor-Inge Eidsvik Nygård, Ottar Laurits Osen, Robin Trulssen Bye, Solveig Bjørneset, Norvald Kjerstad

Agenda

1. Get some input from Solveig on the simulation
2. Get a presentation on different sensors from Norvald

Discussion

We had a small meeting with Solveig to show her what has been done since our last meeting. We also had some question on how to go forward with the simulation. Solveig had sent us an mail with some information and input values for surge, sway and yaw that we can use to calculate force. The meeting was short(about 30 min) but we now have a better idea on how to go forward.

After the meeting with Solveig we had an appointment with Norvald to go through some navigation solutions and also some common sensors to see how they work. We meet Norvald in a Navigation lab on NMK2, where he showed us some of the equipment that they have for navigation. There also was some lasersensors so we got some input of how they are best used.

New Business

1. Make changes to the simulation based on the information from Solveig
2. Do more research on satellites and GNSS

Next Meeting: Thursday ,March 8, at 10:00

Project Meeting

Minutes for March 8, 2018

Present: Simon Vågeskar Bårslett (Chair), Martin Nikolai Longva, Thor-Inge Eidsvik Nygård, Solveig Bjørneset

Absent: Ottar Laurits Osen, Robin Trulssen Bye

Agenda

1. Get some input from Solveig on the simulation.
2. Go over what has been done since last meeting.

Discussion

We had a meeting with Solveig to show her what has been done since our last meeting. We had some questions about how to go forward with the observer, and we were told that a simple kalman filter will work just fine in our case. Since last time we have made some cases of two or more sensors that can work well together for a final solution in our project. And following that we had some questions about how to simulate specific sensor to make the solutions realistic. The way we came up with was to make it as simple as possible, and set a desired destination in the simulation and put some disturbance to the readings to make it more realistic.

New Business

1. Research more about kalman filter
2. Make the simulation more realistic for each sensor

Next Meeting: Wednesday, April 18, at 09:00

Project Meeting

Minutes for April 18, 2018

Present: Simon Vågeskar Bårslett (Chair), Thor-Inge Eidsvik Nygård, Robin Trulssen Bye, Solveig Bjørneset

Absent: Martin Nikolai Longva, Ottar Laurits Osen

Agenda

1. See how far we have come. Show simulation and report.
2. Ask for help with Kahlman filter.

Discussion

In this meeting we showed and talked about the simulation. We got some input and will try to tune the PID to better slow down the vessel when approaching the quay, as we noticed this was a problem when we gave the vessel an initial speed at simulation startup. We talked a bit about Kahlman filter, and Solveig suggested that we could send the entire matlab file to her, and she will send it to a man in RR to see if he can help us to implement the filter.

Solveig asked us if we were interested in holding a presentation at one of their regular meetings, Thursday 26.04.2018 was proposed. A PPT presentation and maybe a video of the simulations should be prepared for this. Try to include things we want more input on.

In the report we should mention that Simulink have more ideal conditions than real life. Also, the sensors and GPS systems might update their values more often in the simulation than in real life. It should also be specified why we choose the white noise for the signals. How do we determine how much noise should be applied to the system? We should also specify what we focus on in the simulation. We should not have a conclusion in the simulation chapter, only have the data and results. We should have a chapter specifically for discussing the results of the simulations.

We should discuss in the report how to determine when we are docked. When is the ship close enough to quay? Cameras? Capacitive or inductive sensors? Robin want a copy of our report how it is now, to give us some pointers.

In the future we should be better at sending status reports and gantt diagram. What did we want to do and what did we actually did. Send this before the meetings.

New Business

1. We will try to tune the PID to slow down the vessel when it approaches the quay.
2. 1. Hold a presentation next week. Thursday? Make video and a PPT presentation.

Next Meeting: Tuesday ,May 15, at 10:00

Project Meeting

Minutes for May 15, 2018

Present: Simon Vågeskar Bårslett (Chair), Martin Nikolai Longva, Thor-Inge Eidsvik Nygård, Robin Trulssen Bye, Solveig Bjørneset

Absent: Ottar Laurits Osen

Agenda

1. Show the final solution of the simulation.
2. Ask for some advice on the structure of the report.

Discussion

In this meeting we showed and talked about the simulation in Simulink. In detail we opened some of the blocks and described how the vessel-kinematics works. Then we ran the simulation once to show the results of it.

We had some questions about the structure of the report. We went trough some main topics and asked for some feedback on that. Robin and Solveig had some good pointers of what we should do and write in the report, and what should be the main focus. After that, Robin showed us some helpful commands in Sharelatex that would help us with the look of our mathematical formulas.

The main topic of this meeting was to present the simulation, and go somewhat into detail of how each step of it works.

New Business

1. Implement some of the feedback into the report.