



DEPARTMENT OF MARINE TECHNOLOGY - IMT

TMR4240 - PROJECT PART 1

Marine Cybernetics

DP positioned supply vessel - Design and Control

Group:
Group 15

Authors:
Elias Strømmen Ravnestad
Mathias Netland Solheim
Johan Bakken Sørensen

November 2, 2020

Table of Contents

1	Introduction	1
2	Reference Frame	2
3	Process Plant Model	2
3.1	Vessel Dynamics	2
3.2	Low-frequency Process Plant Model	2
3.3	Wave-frequency Process Plant Model	3
4	Control Plant Model	4
4.1	Low-frequency Control Plant Model	4
4.2	Wave-frequency Control Plant Model	5
4.3	Complete Control Plant Model	6
5	Environmental Models	7
5.1	Current model	7
5.2	Wind model	8
5.3	Wave model	9
6	Reference Model	10
6.1	Implementation	10
6.2	Tuning	10
7	Controller Design	12
7.1	Control Algorithm	12
7.2	Implementation	12
7.3	Tuning	13
8	Thrust Allocation	15
8.1	Thrusters	15
8.2	Linear Effector Model	15
8.3	Extended Thrust Approach	16
8.4	Unconstrained control allocation	17
8.5	Optimization efforts	18
9	Observer Design	19
9.1	Extended Kalman Filter	19

9.2 Non-Linear Passive Filter	22
10 Simulation Results	27
10.1 Simulation 1 - Environmental Loads	27
10.2 Simulation 2 - DP and Thrust Allocation	27
10.3 Simulation 3 - DP and Environmental Forces	33
10.4 Simulation 4 - Observer selection	35
10.5 Simulation 5 - Complete simulation	37
10.6 Simulation 6 - Capability plot	40
10.7 Simulation 7 - Observer robustness	41
11 Discussion	43
11.1 Simulation 1	43
11.2 Simulation 2	43
11.3 Simulation 3	44
11.4 Simulation 4	44
11.5 Simulation 5	45
11.6 Simulation 6	45
11.7 Simulation 7	45
12 Conclusion	46
Bibliography	47

1 Introduction

This report details the work done for Part Two of the semester project in the course TMR4240 Marine Control Systems I. The overall purpose of the project is to design and implement a dynamic positioning system for a supply vessel. The goal of part one of the project was to design and validate the Current Velocity Model, DP Reference Model and the Controller for a supply vessel DP system. In part two of the project the system is extended to be more realistic, including more dynamics, observer implementation and thrust allocation procedures.

The mathematical model of the vessel, with all equations of motion and corresponding hydrodynamic coefficients used in the simulations are provided in SIMULINK. In addition, environmental loads, as current, wind and wave forces, are modeled and simulated, making the simulations more realistic. A reference frame is presented and used in the simulations. The theory behind process plant modelling and control plant modelling is provided. A control plant model of the vessel, is proposed in mathematical terms. Based on the control plant model, two observers are presented, implemented and tested. The reference model implemented in the simulations is presented. A controller used in calculating the desired thrust is chosen and implemented. To achieve optimal DP functionality, thrust allocation is handled and the controller is tuned.

To verify the functionality of the implemented systems several simulations run in different scenarios were tested. First, the vessel behavior under influence of Environmental forces, was shown by simulating the vessel with no active thrusters. Further, the system was tested in typical DP-positioning scenarios. The first scenario consisted of simulating a classic box-test with no acting environmental loads. The box-test tested the systems ability to maneuver to and stabilize around four different set-points, changing both position and heading references. In addition the system was tested with two faulty thrusters. Next, a box-test with external environmental loads acting, was carried out. In order to select which observer to use in further simulations, the observers was tested and compared under the same conditions as the first simulation with a constant thrust. With the selected observer, another box-test including the full DP system and environmental loads were attempted. The capability of the system was attempted to be simulated in a utilization plot. Finally, the observer robustness was tested by running with some of the environmental factors being changed to more extreme proportions.

2 Reference Frame

As a reference frame for the vessels motions, the North-East-Down- (NED), and body-fixed frame were used.

The NED-frame is a tangent plane to the earths surface and is denoted as:

$$\{n\} = [x_n, y_n, z_n]^T$$

where x_n follows true north, y_n follows east and z_n is positive pointing downwards. The body-fixed frame corresponds to the vessels different directions and is denoted as:

$$\{b\} = [x_b, y_b, z_b]^T$$

where x_b corresponds to the vessels longitudinal direction, y_b corresponds to transverse directions and z_b points in a direction normal to x_b and y_b .

3 Process Plant Model

The process plant model, or simulation model, is a mathematical model of the physical process. In the case of a supply vessel there is a desire to simulate the real world dynamics. The process plant model gives the necessary detailed descriptions of the vessels dynamics as well as external forces and moments from thrusters and environmental loads.

3.1 Vessel Dynamics

When modeling vessel dynamics, it is common to separate the total model into two components using superposition. The first is a wave-frequency model and the second is a low-frequency model, hereby referred to as WF- and LF-model respectively. Thus, the total motion becomes a sum of the WF- and LF-models.

The WF-motions are caused by first order wave loads, while the LF-motions are assumed to be the result of second-order mean and slowly varying wave loads, current loads, wind loads, and thrust forces.

3.2 Low-frequency Process Plant Model

In the lecture slides the nonlinear LF process plant model for a 6 DOF DP system is expressed by equation 3.1.

$$M\dot{v} + C_{RB}(v)v + C_A(v_r)v_r + D(v_r) + G(\eta) = \tau_{wind} + \tau_{wave2} + \tau_{thrust} \quad (3.1)$$

Where M is the mass matrix, including low frequency added mass coefficients. $C_{RB}(v)$ and $C_A(v_r)$ are the Coriolis matrix for respectively rigid body motion and added mass. $D(v_r)$ is the linear and nonlinear damping matrix and $G(\eta)$ is the restoring force matrix. τ_{wind} and τ_{wave2} is wind forces and second order wave forces(mean and slowly varying). The τ_{thrust} are the thrust forces. In case of ice and mooring it is possible to add τ_{ice} and $\tau_{mooring}$, on the right side of the equation. v is the speed of the vessel. The relative velocity vector is defined by equation 3.2.

$$v_r = [u - u_c, v - v_c, w, p, q, r]^T \quad (3.2)$$

Where \mathbf{u} is the speed in surge and \mathbf{u}_c is current the speed in surge, \mathbf{v} is the speed in sway and \mathbf{v}_c is the current speed in sway, w is the speed in heave, and \mathbf{p} , \mathbf{q} , and \mathbf{r} are the angular speed in respectively roll, pitch and yaw.

3.3 Wave-frequency Process Plant Model

The behavior of first order linear waves is solved as two separate sub-problems, wave reaction and wave excitation. Which is respectively forces due to the vessel being forced to oscillate with the waves and forces due to the vessel being restrained from oscillation. The lecture notes (Sørensen [2018]) express the solution to the respective problems by equation 3.4 and equation 3.3.

$$\mathbf{M}(\omega)\ddot{\boldsymbol{\eta}}_{Rw} + \mathbf{D}_p(\omega)\dot{\boldsymbol{\eta}}_{Rw} + \mathbf{G}\boldsymbol{\eta}_{Rw} = \boldsymbol{\tau}_{wave1} \quad (3.3)$$

$$\boldsymbol{\eta}_w = \mathbf{J}(\bar{\boldsymbol{\eta}}_2)\dot{\boldsymbol{\eta}}_{Rw} \quad (3.4)$$

Where

$$\bar{\boldsymbol{\eta}}_2 = [\mathbf{0}, \mathbf{0}, \psi_d]^T \quad (3.5)$$

Where ψ_d is the heading. $\mathbf{M}(\omega)$ is the inertia matrix, including frequency dependent added mass coefficients. $\boldsymbol{\eta}_{Rw}$ is the WF motion vector in the hydrodynamic frame. $\boldsymbol{\eta}_w$ is the earth fixed wave induced position. $\mathbf{D}_p(\omega)$ is the wave radiation damping matrix. \mathbf{G} is the linearized restoring coefficient matrix. $\boldsymbol{\tau}_{wave1}$ is the first order wave excitation vector and the \mathbf{J} is the rotation matrix.

4 Control Plant Model

In DP systems it may not be possible or too expensive to measure many of the states of interest concerning control. Therefore there is a desire to calculate or reconstruct those states based on the measurements available. This is called state estimation and is done in the observers. An observer also provides filters for noise and wave induced motions such that only the LF states affect the controller. The red line in Figure 4.1 provides an example of how an observer uses a low pass filter to remove high frequencies and a notch filter to remove resonance. In order to compute a state estimation, it is necessary to implement a control plant model. A control plant model is a simplified mathematical model of the vessel. The control plant model will be used in construction and design of the observers. The observer should also be able to handle temporary loss of position and heading measurements by estimating the motion in the feedback loop. This kind of state estimation is called dead reckoning, see Figure 4.1. Dead reckoning allows increased operability by fault-tolerant control and smooth transitions to the actual position and heading when signals come back.

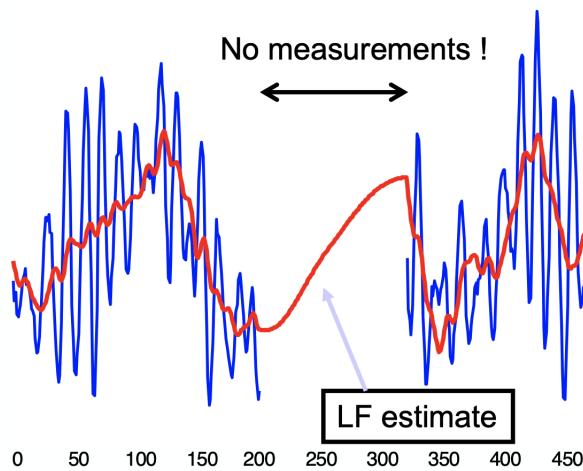


Figure 4.1: Dead reckoning(Lecture slides: DP Control CPM)

4.1 Low-frequency Control Plant Model

A LF control plant model is a simplified model of the actual LF process plant model, equation 3.1. In the case of a supply vessel, the first step of simplifying is reducing degrees of freedom, from a 6 DOF system to a 3 DOF system. This is done by using model reduction matrices, \mathbf{H}_{ixj} , that reduces the degrees of freedom considered in the matrices in the equation, \mathbf{M} , $\mathbf{C}_{RB}(v)$, $\mathbf{C}_A(v_r)$, $\mathbf{D}(v_r)$ and $\mathbf{G}(\eta)$. Next step is assuming small velocities, v_r and linearity. Then the Coriolis components can be neglected, reducing the equation 3.1 into equation 4.1.

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}\mathbf{v} + \mathbf{R}^T(\psi)\mathbf{G}\eta = \boldsymbol{\tau} + \mathbf{R}^T(\psi)\mathbf{b} \quad (4.1)$$

Where

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \quad (4.2)$$

and

$$\mathbf{R}(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

and the new velocity vector becomes

$$\mathbf{v} = [\mathbf{u}, \mathbf{v}, r]^T \quad (4.4)$$

This system is linear, except for the kinematics represented by $\mathbf{R}(\psi)$. The $\mathbf{G} \cdot \boldsymbol{\eta}$ component represents mooring, and can therefore be removed from the equation in this case. The \mathbf{b} term is the bias model. The bias model represents slowly varying excitation loads due to wind, current and second order waves, and take care of unmodeled dynamics. The bias model contributes with an integral effect, which contradicts steady state deviations. In the observer there are two different commonly used bias models, the first order Markov model, equation 4.5, and the Wiener process, equation 4.6.

$$\dot{\mathbf{b}} = -\mathbf{T}_b^{-1}\mathbf{b} + \mathbf{E}_b\mathbf{w}_b \quad (4.5)$$

$$\dot{\mathbf{b}} = \mathbf{E}_b\mathbf{w}_b \quad (4.6)$$

\mathbf{w}_b is a zero-mean Gaussian white noise vector, \mathbf{T}_b is a diagonal matrix of the bias time constants, and \mathbf{E}_b is a diagonal scaling matrix. Equation 4.5 will be used further in this report.

4.2 Wave-frequency Control Plant Model

In order to create a simplified model of the WF individual motions, it is not necessary to make as much of assumptions and simplifications as for the LF model. The WF model can fully be represented by a harmonic oscillator, written in state space form, equation 4.7 and equation 4.8.

$$\dot{\zeta}_w = \mathbf{A}_w\zeta_w + \mathbf{E}_w\mathbf{w}_w \quad (4.7)$$

$$\eta_w = \mathbf{C}_w\zeta_w \quad (4.8)$$

$\dot{\zeta}_w$ is the wave induced motion vector, \mathbf{A}_w is the system matrix, \mathbf{E}_w is the disturbance matrix, \mathbf{w}_w is a zero-mean Gaussian white noise vector and \mathbf{C}_w is the measurement matrix. All these matrices are mathematically defined in equation 4.9.

$$\mathbf{A}_w = \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ -\Omega^2 & -2\Lambda\Omega \end{bmatrix}, \mathbf{E}_w = \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{K}_w \end{bmatrix}, \mathbf{C}_w \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \end{bmatrix} \quad (4.9)$$

Where $\Omega = diag\{\omega_1, \omega_2, \omega_3\}$, $\Lambda = diag\{\zeta_1, \zeta_2, \zeta_3\}$ and $\mathbf{K}_w = diag\{K_{w1}, K_{w2}, K_{w3}\}$. For further details see Sørensen [2018].

4.3 Complete Control Plant Model

The proposed complete control plant model for a supply ship is

Low-frequency model:

$$M\dot{v} = \tau + R^T(\psi)b - Dv \quad (4.10)$$

Bias model:

$$\dot{b} = E_b w_b - T_b^{-1} b \quad (4.11)$$

Kinematics:

$$\dot{\eta} = R(\psi)v \quad (4.12)$$

Wave frequency model:

$$\dot{\zeta}_w = A_w \zeta_w + E_w w_w \quad (4.13)$$

$$\eta_w = C_w \zeta_w \quad (4.14)$$

Measurements:

$$y = \eta + \eta_w + v \quad (4.15)$$

Where the v is the measurement noise term.

5 Environmental Models

The environmental models are mathematical models applied to the vessel model in order to create a realistic simulation, similar to what a supply ship would experience during operations. Three different types of environmental loads were modeled and implemented in this project part: current, wind and waves.

5.1 Current model

The current model applies a disturbance to the vessel similar to that of a surface current in the real world. The current model used can be modeled in the NED-frame as shown in Equation 5.1

$$\nu_c = [V_c \cos(\psi_c), V_c \sin(\psi_c)]^\top \quad (5.1)$$

Where ν_c is the current velocity in NED-frame, V_c is the current velocity amplitude, and ψ_c is the current direction, expressed as an angle in the NED-frame. The model features a variation in the current velocity amplitude $V_c s$ and current direction ψ_c , expressed with a Gauss-Markov process, as shown in Equation 5.2

$$\dot{V}_c + \mu V_c = w_1 \quad \dot{\psi}_c + \mu \psi_c = w_2 \quad (5.2)$$

Where w_1 and w_2 are distinct Gaussian white noise terms, and μ is a positive constant. This causes the current velocity and direction to vary slightly, but not suddenly over time, akin to a realistic current variation. For part 2 of the project this was added to the model with limitations on the variations.

The current model was implemented in Simulink using a pre-defined block from the MSS toolbox, which receives an angle ψ_c and a current velocity V_c , and applies the mathematical model in Equation 5.1. The variation terms were applied in their own subsystems, using the slow-variation generator developed in Assignment 2. This is shown in Figure 5.1 for current velocity (current direction is identical). The entire current model is shown in Figure 5.2, where the MSS North-East current block is used, before the signal is converted to the body frame of the ship.

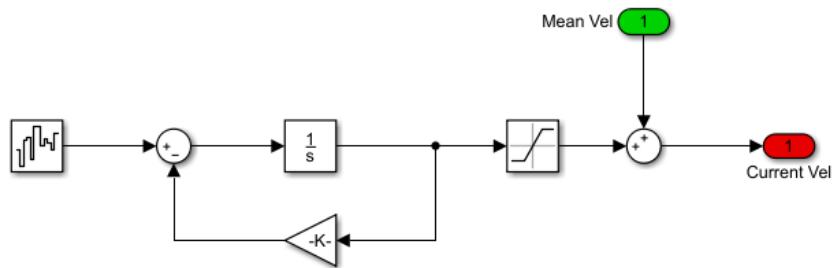


Figure 5.1: Current velocity variation in Simulink

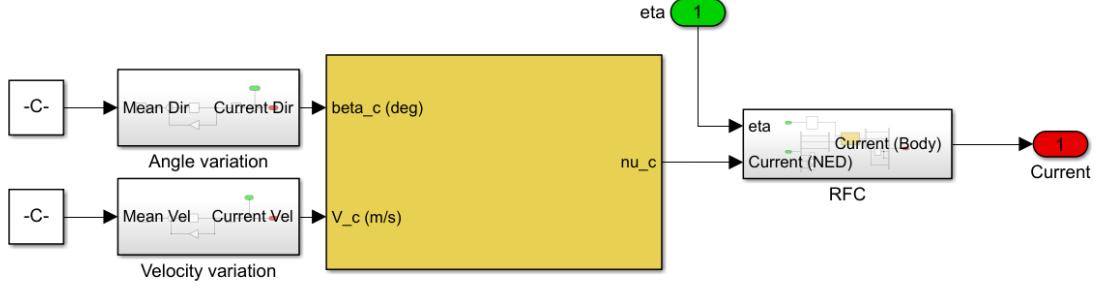


Figure 5.2: Current model in Simulink

5.2 Wind model

The implemented wind model yields a velocity that consists of three parts: The mean, slowly varying and gust components. The velocity from these three components are evaluated and added together to \mathbf{V}_w , and the wind force \mathbf{F}_{wind} can then be expressed as shown in Equation 5.3.

$$\mathbf{F}_{Wind} = |\mathbf{V}_w|^2 C_w(\alpha_{rw}) \quad (5.3)$$

C_w is a table of wind coefficients for each degree of freedom, dependent on the incident angle of the wind force α_{rw} . This table was given, with wind coefficients for each 10 degrees between 0 and 360°. The incident angle is calculated based on the current state of the ship, and the wind angle, which also has a slowly-varying component, set not to exceed 5 degrees.

The MSS toolbox features a Simulink block for wind speed, including all three components of mean, slowly-varying and gust. The mean wind speed and mean wind angle are set beforehand, whilst the gust component is calculated using a wind spectrum. The block utilizes the NORSOOK wind spectrum, which is based on measurements over the North Sea (Sørensen [2018]). The given parameters were used for the spectrum, including the numbers used in Assignment 2. For any other parameters, the default values were used.

The Simulink model for wind can be seen in Figure 5.3. The model uses the MSS wind block and the current position of the ship to evaluate the wind loads. The wind force coefficients are handled using a custom-written Matlab function and a lookup-table block to find the correct coefficients for the relevant angle.

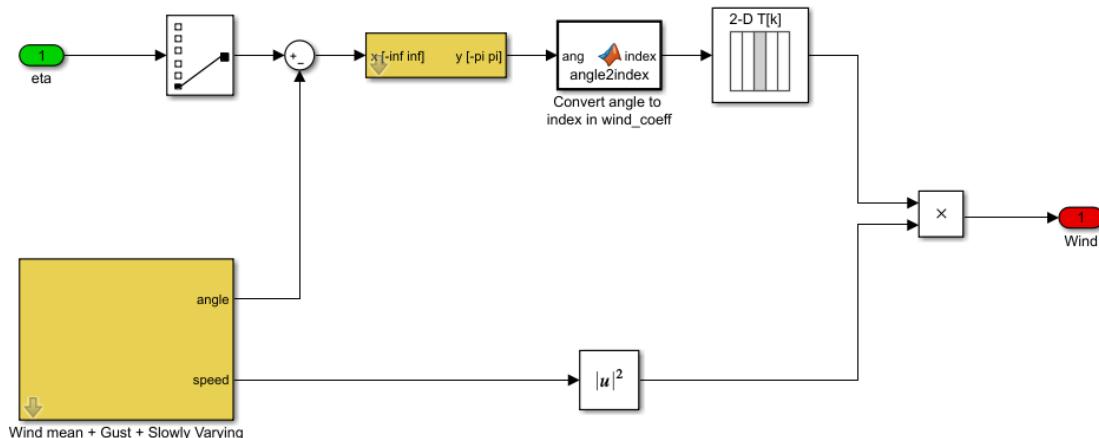


Figure 5.3: Wind model in Simulink

5.3 Wave model

The waves loads on the supply ship are modeled using a wave spectrum, namely the ITTC spectrum. This wave spectrum uses a Significant wave height H_s and a peak wave period T_p as parameters to describe the sea state. The wave loads are completely handled by the MSS Wave block, that calculates the relevant wave load given the parameters for the ITTC spectrum.

6 Reference Model

When an operator designates a new set-point for a vessel, the distance might be considerable. This means that the deviation becomes quite large, causing the forces generated by the vessels actuators to become saturated. This results in large accelerations causing sudden motions of a higher magnitude than desired. Working in saturation can also cause instabilities in the system, especially for integrators.

To compensate for this we want to implement a method to generate a trajectory between two set points. This is the reference model. Its purpose is to synthesize the desired position, velocity, and acceleration from set points, and feed these filtered global set-points into the control-algorithm. The result is a much smoother transition between set points.

6.1 Implementation

For this project a third-order filter as described in the lecture notes (Sørensen [2018]) and in Fossen [2011], is implemented as the reference model. The "Position and Attitude Reference Model", as Fossen refers to it as, is composed of a low-pass filter cascaded with a mass-damper-spring system. The transfer-function between the set-point position η and reference-value r is described in equation 6.1:

$$\frac{\eta_{d_i}}{r_i} = \frac{1}{1 + Ts} \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2} = \frac{\omega_{n_i}}{\omega_{n_i} + s} \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2}, \quad i = 1, 2, 3 \quad (6.1)$$

In equation 6.1 the relation of $T = 1/\omega_n$ is used. The reference model is implemented as its own subsystem in Simulink:

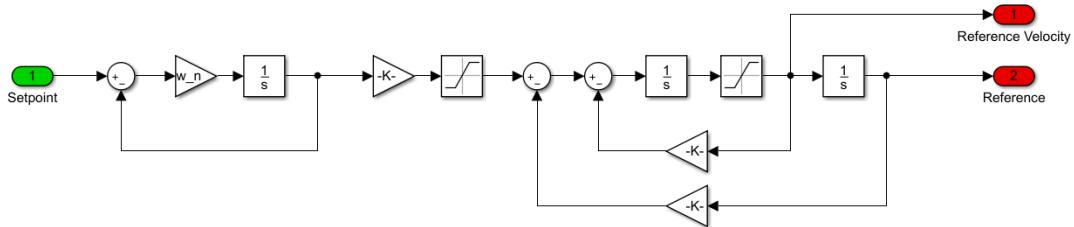


Figure 6.1: Implementation of Reference model in Simulink

To prevent the reference model from feeding unfeasible references to the controller, saturation was implemented on the acceleration- and velocity-terms.

6.2 Tuning

The reference model contains six variables that must be tuned for the system to function optimally. These consist of the damping ratio ζ_i and the natural frequency ω_{n_i} , each in all three degrees of motion.

The tuned damping ratios are presented in equation 6.2.

$$\zeta_i = [1 \ 1 \ 1]^T \quad (6.2)$$

for all degrees of motion, as it was proved that a critically damped system was easy to work with. It provided reasonably fast and accurate response.

The reference systems bandwidth needs to be lower than that of the motion control system. To tune the reference system we found the actual vessels natural frequencies, and identified the lowest one. This was taken as a starting point, and tuned downwards.

When tuning, a couple of protocols were followed: Considering the supply ships natural shape, it is clear that the vessels maneuverability is much better in longitudinal direction, than in the lateral. It is therefore reasonable to tune in a more aggressive behaviour, i.e higher ω , in surge than in sway.

It is also desirable to keep the heading of the ship as stable as possible, so the ω_3 was tuned with the least aggressive behaviour. Thus the parameters were tuned after the following relationship $\omega_1 > \omega_2 > \omega_3$. The final parameters are presented in equation 6.3.

$$\boldsymbol{\omega}_{n_i} = [0.08 \quad 0.04 \quad 0.03]^T \quad (6.3)$$

The parameters in equation 6.2 and equation 6.3 resulted in the desired behaviour; a smooth trajectory between set points.

7 Controller Design

The purpose of the controller is to achieve correspondence between the state variables and the desired setpoint. In this case, the controller achieves this by use of the feedback-loop. By comparing the current state of the control plant with the desired state, it generates an input to the control plant in order to reduce the difference between state and setpoint.

In this case, the controller receives a desired trajectory for the ship, given by the reference model. In addition; it receives the current value of the state variables we wish to control, the surge, sway and yaw, as well as their rates of change. The algorithm inside then generates a control plant input τ , which is fed into the ship model, after correcting for the reference frame.

7.1 Control Algorithm

The control algorithm decides how the deviation between state and setpoint are manipulated in order to give a good input to the control plant. Some different control algorithms were considered, such as the *Linear Quadratic Regulator* (LQR) algorithm. In the end, the standard Proportional-Integral-Derivative, or PID-algorithm, was chosen. The PID-controller was chosen due to its relatively easy implementation and usage, as well as the group members' familiarity with its behaviour.

The PID-algorithm can be expressed mathematically as in Equation 7.1. It manipulates the deviation in three different ways. The proportional part adds an output directly proportional to the deviation. The integral part will apply an output based on all previous deviation, and can be thought of as a long-term corrector. The derivative part will generate a controller output based on how the deviation changes in that moment, and can therefore be thought of as a predictor.

$$u(t) = K_p e(t) + K_i \int_0^t e(t') dt' + K_d \frac{de(t)}{dt} \quad (7.1)$$

In Equation 7.1, $u(t)$ is the controller output/control plant input, and $e(t) = \eta_0 - \eta_{now}$ is the deviation between set-point and current position. The controller gains K_p , K_i and K_d are all constants, and determine the effect of each part. These constants can be evaluated by tuning the control system.

7.2 Implementation

The Simulink model for the controller is shown in Figure 7.1. The deviations between current state and the requested behavior from the reference model is calculated and fed into the PID-algorithm. In addition to performing the PID-algorithm, the states is also converted from the NED reference frame that the setpoint is specified in, into the body frame of the ship.

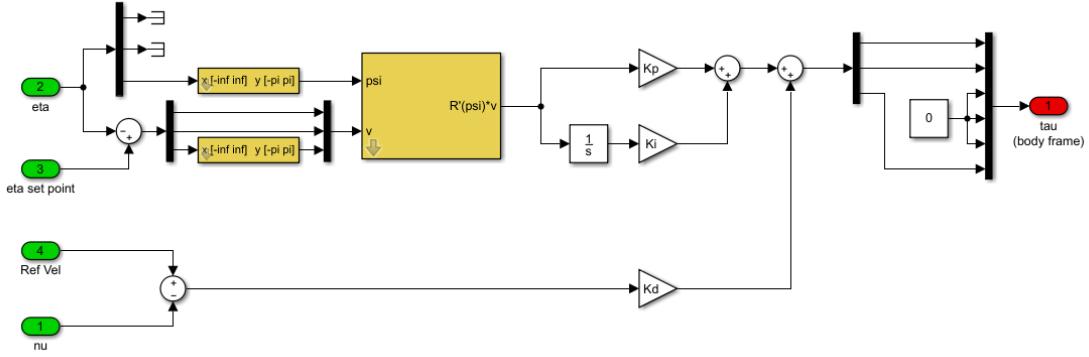


Figure 7.1: PID-controller implemented in Simulink

7.3 Tuning

Different values for the controller gains will yield different types of behaviour from the system. Some sets of constants will cause a calm and steady convergence to the setpoint, whilst others might cause overshoot and/or oscillatory behaviour. Some combinations of controller gains might destabilize the system altogether. There are a multitude of tuning methods to find a decent set of controller gains. For this project, a tuning scheme from Fossen [2011] was used.

7.3.1 Tuning by Fossen's algorithm

The tuning scheme starts by assuming a reasonable bandwidth for the control plant, as well as a damping ratio, which will determine the shape of the response. For the damping ratio, a value of 1 was chosen, as this would yield a response similar to that of a critically damped mass-spring-damper system. For the bandwidth, Fossen [2011] specifies that a value of 0.01 is reasonable for large tankers, whilst a value of 0.1 was suitable for smaller vessels. A value of 0.1 was chosen to go on with.

From the bandwidth ω_b and a damping ratio ζ a natural frequency can be determined by Equation 7.2

$$\omega_n = \frac{1}{\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}} \omega_b \quad (7.2)$$

Together with the mass m , one can then find the gain constants by the following scheme

Table 7.1: Gain constants by Fossen's algorithm

K_p	$m\omega_n^2$
K_i	$\frac{1}{10} \omega_n K_p$
K_d	$2\zeta\omega_n m$

The constants produced by this algorithm were applied to the Simulink model. Because the algorithm is based for single-input single-output (SISO) systems, the same constants were applied to correct all three degrees of freedom (surge, sway, yaw). With this in mind, optimal controller behaviour was not expected. The gain constants turned out to be quite a bit lower than what was needed, especially in yaw. The controller gains from the Fossen algorithm did however serve as a decent benchmark from which manual testing could be applied to find a better tuning result.

7.3.2 Manual tuning

Starting out with the suboptimal result from the Fossen algorithm, it was decided to attempt manual tuning to improve the performance. A PID-controller is relatively easy to tune, since each gain value has a clear response behaviour associated with it. Thus, the manual tuning process yielded some decent insights into the system's dynamics.

After some back and forth manual tuning attempts, the gain constants presented in Table 7.3 were the final values. These are by no means final numbers, and will likely be further changed around as the project develops. The constants were first chosen to ensure a decent performance on all the four simulations that were conducted for Part 1.

Table 7.2: Controller gain constants for part 1

	Surge	Sway	Yaw
K_p	$5e + 5$	$5e + 5$	$2e + 7$
K_d	$1e + 7$	$1e + 7$	$2e + 8$
K_i	$1e + 2$	$3e + 3$	$1e + 2$

Since the performance in Part 1 was lacking somewhat, and significant changes were made to the system the controller was re tuned. This was also done manually by experimentation until a desired behaviour was achieved.

Table 7.3: Controller gain constants for part 2

	Surge	Sway	Yaw
K_p	$1e + 5$	$5e + 5$	$1e + 7$
K_d	$1e + 7$	$1e + 7$	$2.01e + 8$
K_i	$2e + 2$	$2e + 2$	$1e + 2$

8 Thrust Allocation

The control-algorithm only outputs the net forces needed to move the vessel to the given reference. This can not be applied directly to the ship, as it has no way of distributing the generalized forces to its thrusters. Thus we have to implement a system that decomposes the general forces computed by our PID into individual commands for each of the supply vessels thrusters. This is known as Thrust Allocation.

The system is over-actuated, and has several valid approaches regarding thrust allocation. Initially, an attempt was made at an optimization-problem using quadratic programming but this was scrapped so that more focus could be put towards report-writing and tuning of the whole system. Instead, an extended thrust algorithm was implemented.

8.1 Thrusters

The supply-vessel has a total of five thrusters. One tunnel-thruster below the bow, and four azimuth-thrusters. Two parallel at the stern, and two in the bow.

As with every physical system, the thrusters have some limitations that should be taken into account. These include the maximum thrust, the rate at which the thrusters reach maximum thrust, and rotation-rate on the azimuth thrusters. All thruster specifications are presented in table 8.1

Thruster no.	X [m]	Y [m]	Angle [rad]	Rotation rate [rpm]	Max Thrust [kN]	0 100% [s]
1	39.5	0	$\pi/2$	[·]	125	8
2	35.3	0	α_2	2	150	8
3	31.3	0	α_3	2	150	8
4	-28.5	5	α_4	2	320	10
4	-28.5	-5	α_5	2	320	10

Table 8.1: Thruster specifications and limitations

8.2 Linear Effector Model

The relationship between the exerted thrust of the actuators and the generalized exerted thrust can be expressed mathematically by the Linear Effector Model model(8.1).

$$\boldsymbol{\tau} = \mathbf{B}\mathbf{u} \quad (8.1)$$

Here $\boldsymbol{\tau}$ is the produced force vector, \mathbf{u} is the vector containing the thrusters command output and $\mathbf{B} \in R^{m \times p}$ is the effectiveness matrix which describes how the thrusters inputs produce $\boldsymbol{\tau}$. The dimensions of \mathbf{B} are determined by the dimensions, m , the actuators produce forces in, along with the number of actuators, p . Our vessel has $p = 5$ actuators, producing forces in $m = 3$ degrees of freedom; surge, sway and yaw.

The application of the linear effector model in thrust allocation revolves around finding the inverse \mathbf{B} such that we can express the actuator commands as:

$$\mathbf{u} = \mathbf{B}^{-1}\boldsymbol{\tau} \quad (8.2)$$

\mathbf{B} is generally not a square matrix, so one has to resort to a pseudo-inverse instead. To write the equation on its full form we denote:

$$\mathbf{u} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3 \quad \mathbf{u}_4 \quad \mathbf{u}_5]^T \quad (8.3)$$

and study how each of the five thrusters produce thrust. The connection is then described in the \mathbf{B} matrix. First of is our tunnel thruster, which only exerts forces in sway direction, with a corresponding moment in yaw. The connection for tunnel thrusters can thus be given as:

$$\mathbf{b}_1 = [\mathbf{0} \quad \mathbf{1} \quad l_{x,1}]^T \quad (8.4)$$

The azimuth-thruster however can be exert forces in both surge and sway, as well as a corresponding moment. The forces depends on the angular-position of the azimuth. We therefore express them as decomposed body-fixed forces in 8.5

$$\mathbf{b}_i = [\cos(\alpha_i) \quad \sin(\alpha_i) \quad l_{x,i} \sin(\alpha_i) - l_{y,i} \cos(\alpha_i)]^T, \quad i = \{2, 3, 4, 5\} \quad (8.5)$$

Thus the full matrix \mathbf{B} is denoted:

$$\mathbf{B} = \begin{bmatrix} 0 & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) & c(\alpha_5) \\ 1 & s(\alpha_2) & s(\alpha_3) & s(\alpha_4) & s(\alpha_5) \\ l_{x,1} & l_{x,2}s(\alpha_2) - l_{y,2}c(\alpha_2) & l_{x,3}s(\alpha_3) - l_{y,3}c(\alpha_3) & l_{x,4}s(\alpha_4) - l_{y,4}c(\alpha_4) & l_{x,5}s(\alpha_5) - l_{y,5}c(\alpha_5) \end{bmatrix} \quad (8.6)$$

The expanded form of 8.1 for our vessel is then:

$$\begin{bmatrix} \mathbf{F}_X \\ \mathbf{F}_Y \\ M \end{bmatrix} = \begin{bmatrix} 0 & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) & c(\alpha_5) \\ 1 & s(\alpha_2) & s(\alpha_3) & s(\alpha_4) & s(\alpha_5) \\ 39.5 & 35.3s(\alpha_2) & 31.3s(\alpha_3) & -28.5s(\alpha_4) - 5c(\alpha_4) & -28.5s(\alpha_5) + 5s(\alpha_5) \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \end{bmatrix} \quad (8.7)$$

8.3 Extended Thrust Approach

As can be seen from equation 8.7, \mathbf{B} is varying with the thruster angle α . This is unfortunate, since the thruster angles is one of the parameters we are trying to compute in our thrust allocation algorithm and not something we are provided. To circumvent the need for the thruster-angle we use an extended thrust approach. This means we decompose the control input to our azimuth thrusters into a longitudinal component and a lateral component and treat each component as if they were a separate thrusters in our computations.

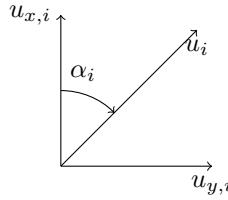


Figure 8.1: Decomposition of azimuth-thrust

The expanded thrust vector becomes:

$$\mathbf{u}_{exp} = [u_1 \quad u_{2,x} \quad u_{2,y} \quad u_{3,x} \quad u_{3,y} \quad u_{4,x} \quad u_{4,y} \quad u_{5,x} \quad u_{5,y}]^T \quad (8.8)$$

$$= [u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7 \quad u_8 \quad u_9]^T \quad (8.9)$$

The \mathbf{B} -matrix must also be expanded. Now each x-component exerts a force in surge along with a moment and each y-component exerts a force in sway, also with a corresponding moment. This can be expressed:

$$\begin{aligned}\mathbf{b}_i &= [\mathbf{0} \quad 1 \quad l_{x,i}]^T, \quad i = \{1, 2, 4, 6, 8\} \\ \mathbf{b}_i &= [1 \quad 0 \quad l_{y,i}]^T, \quad i = \{3, 5, 7, 9\}\end{aligned}$$

This gives us the the expanded \mathbf{B} matrix for our system:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 39.5 & 35.3 & 0 & 31.3 & 0 & -28.5 & 5 & -28.5 & -5 \end{bmatrix} \quad (8.10)$$

$$\Rightarrow \begin{bmatrix} \mathbf{F}_X \\ \mathbf{F}_Y \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 39.5 & 35.3 & 0 & 31.3 & 0 & -28.5 & 5 & -28.5 & -5 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \\ \mathbf{u}_6 \\ \mathbf{u}_7 \\ \mathbf{u}_8 \\ \mathbf{u}_9 \end{bmatrix} \quad (8.11)$$

8.4 Unconstrained control allocation

One thing to note is that this is an unconstrained control allocation, so we can put weights on the different thruster-components to limit the angular-rate and magnitude of the output. The unconstrained control-problem is expressed as:

$$\min_{\mathbf{u} \in \mathbb{R}^p} \frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u} \quad (8.12)$$

$$\text{subject to } \boldsymbol{\tau}_c = \mathbf{B} \mathbf{u} \quad (8.13)$$

If \mathbf{B} has a full rank, the above problem has the solution of:

$$\mathbf{u} = \mathbf{C} \boldsymbol{\tau}_c \quad (8.14)$$

where \mathbf{C} is:

$$\mathbf{C} = \mathbf{W}^{-1} \mathbf{B}^T \left(\mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T \right)^{-1} \quad (8.15)$$

This is a damped least squares inverse. Since there are some discrepancies like thrust loss or singularities that may cause the matrix \mathbf{B} to become rank deficient we want to ensure that the inverse $\mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T$ exists by adding a safety measure $\epsilon \mathbf{I}$. This is an arbitrary small variable that makes sure the pseudo inverse can be calculated in such cases.

$$\mathbf{C} = \mathbf{W}^{-1} \mathbf{B}^T \left(\mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T + \epsilon \mathbf{I} \right)^{-1} \quad (8.16)$$

The weight matrix was tuned to avoid rapid fluctuations in the azimuths. As such a cost of 10 was applied to them, while the tunnel thruster was given a cost of 1. Assuming all elements in the \mathbf{W} are uncoupled, it can be expressed as a diagonal matrix:

$$\mathbf{W} = \text{diag}\{[1 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10]\}$$

When the extended thrust matrix is computed, one can trivially convert the thrust-components to the full thrust and thruster-angle using the relationships expressed in 8.17:

$$u_i = \sqrt{u_{i,x}^2 + u_{i,y}^2} \quad (8.17)$$

$$\alpha_i = \text{atan2}(u_{i,y}, u_{i,x}) \quad (8.17)$$

8.5 Optimization efforts

Since this is an unconstrained approach to the problem, the solution is sub-optimal. To improve on the performance some measures were taken in the implementation, as described in the following section paragraphs.

When the azimuth-thruster are made to rotate from one angle to another, we ideally want to choose the shortest rotational path. To handle this, a basic algorithm was implemented in the thrust allocation system. It compares the previous iterations angle to the angle calculated by the extended thrust approach, and calculates the shortest path. The optimal angle then becomes a sum of the previous angle and the shortest path.

Another rotation optimization was also attempted, based on the algorithm presented in Section 3 of Sørdalen [1997]. This algorithm uses vector projection of the desired thrust and low-pass filtering in order to determine the optimal azimuth angles. Unfortunately, the source material fell a bit short, and it proved difficult to implement and gain any form of performance boost. It was therefore elected to move on without it.

The allocation algorithm does not consider the effects of forces exerted while the thrusters rotating from one angle to a new one. To minimize potential unwanted effects when the rotational distance of the azimuth is far, a quick test is implemented to investigate if the rotation is larger than $\pi/2$. If the criteria is met, then the azimuth is set to rotate the opposite direction, such that it points at an angle of π opposite to the commanded. The thrust sign is then flipped. This is possible due to the fact that azimuths are able to produce thrust in both directions. It is however important to note that most propellers are designed to rotate more efficiently one way, so forcing the thruster to rotate the other way might not be optimal in regards to wear and tear, but for this project it yielded better performance.

Finally a check was added for the thrusters physical limitations. This checks if the rate of change in thrust force and angle does not supersede the real limits. Although this is technically also handled in the "Thruster dynamics"-block it yielded a much more accurate behaviour for the allocation algorithm, as can be viewed in section 10.

9 Observer Design

When trying to control a real world process, one is reliant on a plethora of measurements to determine the systems current states. In marine applications this can be anything from positional measurements from GPS, heading position, different directional accelerations or rotational velocities. It is however in most cases, not possible to get accurate measurements of the process-states, or in some cases, any measurements at all. This may be because no viable sensors exist, or said sensor is too expensive. The measurements from the available sensor are also always contaminated by noise from external sources. To properly use these measurements, they must be processed and put together to provide an *estimate* of the system states. This is the job of an observer.

In DP-application the observer mainly functions for wave-filtering. As explained previously in section 3 and 4, the motions of a marine vessel can be split into a low-frequency and a wave-frequency part. In regards to control, the WF motion is of no concern. Thus the observer estimates the LF-motions, and makes sure the WF motions are not taken into account in the controller. Additionally as mentioned in section 4, the observer can act as a backup in case of sensory fail. It will then be able to predict the vessels position based on previous measurements. This is known as *dead reckoning*.

Two types of observers were implemented and compared for this project. Based on the lecture notes Sørensen [2018] and Fossen [2011] the Extended Kalman Filter and a Non-Linear Passive Filter were chosen.

9.1 Extended Kalman Filter

The Extended Kalman Filter filter is applicable for nonlinear systems on the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{Bu} + \mathbf{Ew} \quad (9.0)$$

$$\mathbf{y} = \mathbf{Hx} \quad (9.0)$$

This is a fifteenth-order state-space model, where $\mathbf{x} = [\zeta^T, \eta_p^T, b_p^T, v^T]^T$, $\mathbf{w} = [w_w^T, w_b^T]$ and $\nu = \tau$.

$$\mathbf{f}(\mathbf{x})$$

denotes the non-linear function of system, given as:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} A_w \zeta \\ R(\phi) \nu \\ -T_b^{-1} b \\ -M^{-1} D \nu + M^{-1} R^T(\phi) b \end{bmatrix} \quad (9.0)$$

The EKF linearizes the system in the current position. In our case it is the rotational matrix $R(\phi)$ which is non-linear, and we must therefore linearize the system for different heading-angles.

For this linearization, the continuous state space matrices were used:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_w & \mathbf{0}_{6x3} & \mathbf{0}_{6x3} & \mathbf{0}_{6x3} \\ \mathbf{0}_{3x6} & \mathbf{0}_{3x6} & \mathbf{0}_{3x6} & \mathbf{I}_{3x3} \\ \mathbf{0}_{3x6} & \mathbf{0}_{3x3} & -\mathbf{T}^{-1} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x6} & \mathbf{0}_{3x3} & \mathbf{M}^{-1} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \quad (9.0)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{6x3} \\ \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} \\ \mathbf{M}^{-1} \end{bmatrix} \quad (9.0)$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_w & \mathbf{0}_{6x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{E}_b \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix} \quad (9.0)$$

$$\mathbf{H} = [\mathbf{C}_w \quad \mathbf{I}_{3x3} \quad \mathbf{0}_{3x3} \quad \mathbf{0}_{3x3}] \quad (9.0)$$

Since measurements are sampled in discrete-time, we must discretize our model. This is done by means of forward difference, resulting in the followin discrete time system model:

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Delta \mathbf{u}(k) + \Gamma \mathbf{w}(k) \quad (9.0)$$

$$\mathbf{y}(k) = \mathbf{H} \mathbf{x} + \mathbf{v}(k) \quad (9.0)$$

where:

$$\Phi = \exp(\mathbf{A}h) \approx \mathbf{I} + (\mathbf{f}h) \quad (9.0)$$

$$\Delta = \mathbf{A}^{-1}(\Phi - \mathbf{I})\mathbf{B} \quad (9.0)$$

$$\Gamma = \mathbf{A}^{-1}(\Phi - \mathbf{I})\mathbf{E} \quad (9.0)$$

With the parameters of the EKF established, the computations in the filter are done with a *corrector-predictor*-approach.

Predictor:

$$\bar{\mathbf{P}}_{k+1} = \Phi_k \hat{\mathbf{P}}_k \Phi_k^T + \Gamma_k \mathbf{Q} \Gamma_k \Gamma_k^T \quad (9.0)$$

$$\bar{\mathbf{x}}_{k+1} = \Phi \hat{\mathbf{x}}_k + \Delta \mathbf{u} \quad (9.0)$$

Kalman gain:

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}^T [\mathbf{H} \bar{\mathbf{P}}_k \mathbf{H}^T + \mathbf{R}]^{-1} \quad (9.0)$$

Corrector:

$$\hat{\mathbf{P}} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \bar{\mathbf{P}}_k (\mathbf{I} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (9.0)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H} \bar{\mathbf{x}}_k) \quad (9.0)$$

9.1.1 Implementation

The Kalman filter was implemented in MATLAB as shown in Figure 9.1. The linearization is handled beforehand, computing a value for the discretized-time matrices used in the predictor-corrector algorithm, for different heading angles.

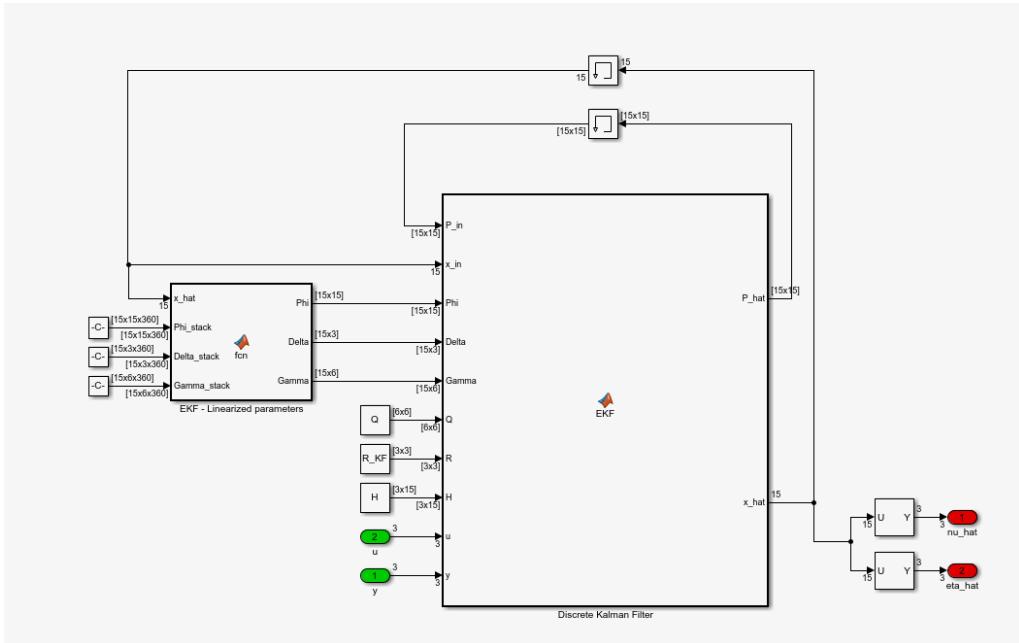


Figure 9.1: Extended Kalman Filter implementation is Simulink

It was elected to create one set of linearization variables at each whole degree of rotation, meaning in total, there were 360 different sets of variables. These were applied depending on the heading angle at the previous time-step and used to create a state estimate. The continuous-time parameters were calculated before the Matlab function `c2d` was used to find the discrete-time versions.

9.1.2 Tuning

Tuning the EKF turned out to be a very difficult task. The tuning parameters are not as intuitive as in the controller, so the presented tuning parameters might not be optimal. The bias model, consisting of \mathbf{T}_b and \mathbf{E}_b , ended up as

$$\begin{aligned}\mathbf{T}_b &= \text{diag}(10^3, 10^3, 10^3) \\ \mathbf{E}_b &= \text{diag}(10^2, 10^5, 10^5)\end{aligned}$$

The time constants were chosen by iteration and what seemed to work well. The \mathbf{E}_b was tuned based on comparison between estimate and actual state, where it seemed that sway and yaw were more difficult to estimate, hence the higher value in \mathbf{E}_b .

The matrix \mathbf{R} , relating to certainty in measurements, was set to $\mathbf{0}_{3 \times 3}$ since we know our measurements are spot-on. This would change if some measurement noise model was added, something that was planned but not implemented. In real-life situations, the measurements are not as precise as in this model, so that \mathbf{R} must be taken into account.

The matrix \mathbf{Q} , dealing with uncertainty in the models, was iteratively tuned in order to find a good correspondence between estimate and actual position, along with eliminating the variations caused by environmental forces. The final value of \mathbf{Q} that was reached, was

$$\mathbf{Q} = \text{diag}(1, 1, 1, 10^4, 10^4, 10^4)$$

9.2 Non-Linear Passive Filter

This entire section is based on Sørensen [2018], Fossen and Strand [1999], and the lecture slides.

9.2.1 Motivation

The Kalman filter has to perform typically 36 linearizations in steps of 10° , to cover the whole heading envelope. This results in 36 tuning matrices. The logistics regarding which gain matrix to select are demanding, and require a lot computational power. Another drawback of the Kalman Filter, is that it does not guarantee global stability. The motivation of the non-linear passive observer design, is creating a global exponentially stable alternative with less tuning parameters. This is done by utilizing the passivity property.

9.2.2 The Passivity Property

Linear systems have the property that they have phase angles between 90° and -90° . This feature is called the passivity property. When a system has the passivity property, the system is stable. Systems consisting of only passive blocks, like the system represented in figure 9.2, are passive as well. Hence, systems consisting of only passive subsystems are stable.

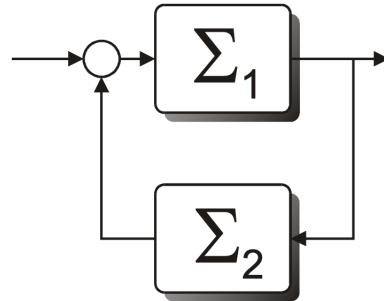


Figure 9.2: Σ_1 and Σ_2 are linear systems, and therefore passive. The complete system is passive, since all the subsystems are passive. (Lecture slides: Passive nonlinear observer)

In order to design a non-linear observer, there is a desire to extend the system represented by figure 9.2, to contain rotations as represented by figure 9.3.

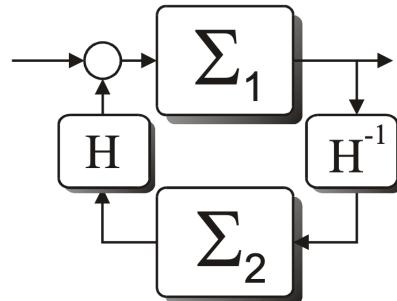


Figure 9.3: Σ_1 and Σ_2 are linear systems, and therefore passive. H^{-1} and H are the rotations expressed by sine and cosine functions. (Lecture slides: Passive nonlinear observer)

The sine and cosine functions are bounded between +1 and -1, and can therefore be considered as passive. Hence, the complete system in figure 9.3 is passive, since all the subsystems are passive.

9.2.3 Design

SPR-Lyapunov analysis is used to prove passivity and stability of the non-linear observer, see Sørensen [2018] for further details. When designing the observer it is assumed that position and heading sensor noise can be neglected, since the term is negligible compared to the wave-induced motion. The amplitude of the wave-induced yaw motion is assumed to be small. Adding the injection term to the control plant model in addition to these assumptions results in the observer represented by equation 9.1.

$$\dot{\hat{\zeta}} = A_w \hat{\zeta} + K_1 \tilde{y} \quad (9.1a)$$

$$\dot{\hat{\eta}} = R(\psi) \hat{v} + K_2 \tilde{y} \quad (9.1b)$$

$$\dot{\hat{b}} = T_b^{-1} \hat{b} + K_3 \tilde{y} \quad (9.1c)$$

$$M \dot{\hat{v}} = -D \hat{v} + \tau + R^T(\psi_d) \hat{b} + R^T(\psi_d) K_4 \tilde{y} \quad (9.1d)$$

$$\hat{y} = \hat{\eta} + \hat{\eta}_w \quad (9.1e)$$

$$\hat{\eta}_w = C_w \hat{\zeta}_w \quad (9.1f)$$

$\tilde{y} = y - \hat{y}$ are the estimation errors and the symbols with hats is the estimated values of the corresponding symbols in the complete control plant model proposed in section 4. $K_1 \in \mathbb{R}^{6 \times 3}$, and $K_2, K_3, K_4 \in \mathbb{R}^{3 \times 3}$ are observer gain matrices. The wiener bias model, equation 4.11 is replaced with

$$\dot{\hat{b}} = K_3 \tilde{y} \quad (9.1)$$

Equation 9.1 results in following Simulink block diagram:

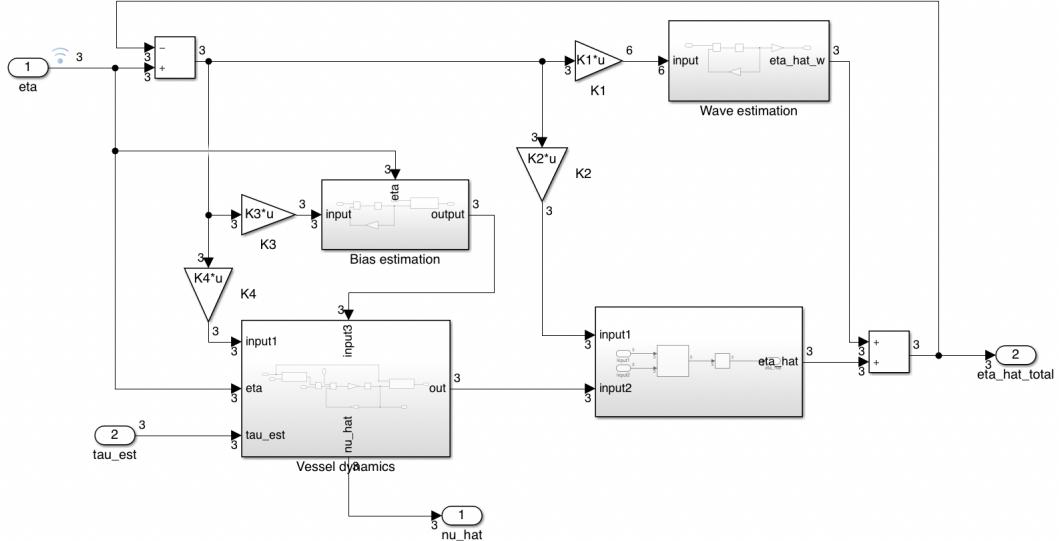


Figure 9.4: The complete Non-Linear Passive Observer.

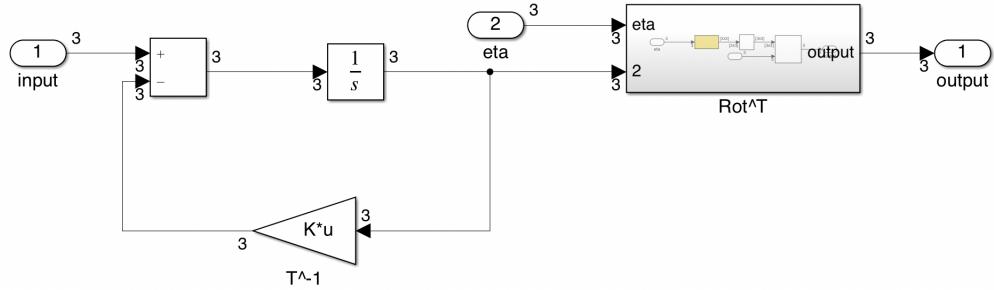


Figure 9.5: The Bias Estimator in the Non-Linear Passive Observer.

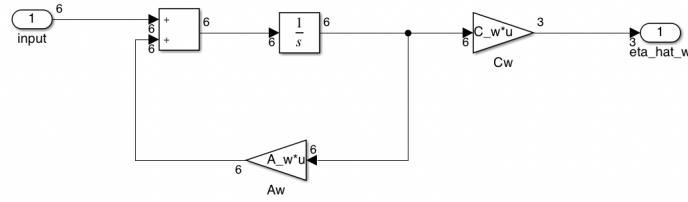


Figure 9.6: The Wave Estimator in the Non-Linear Passive Observer.

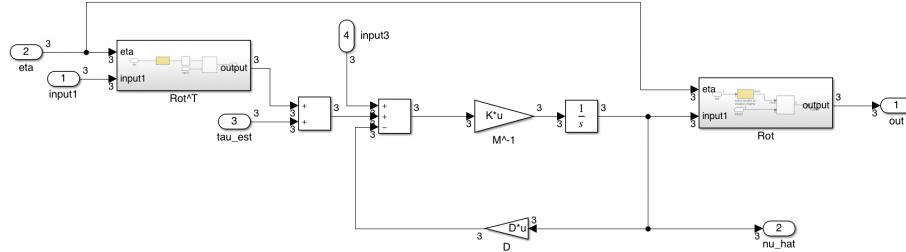


Figure 9.7: The slow varying vessel dynamics in the Non-Linear Passive Observer.

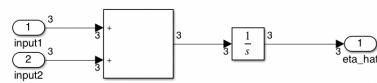


Figure 9.8: Integration of the slow varying components resulting in the estimated position as a result by the slow varying forces.

9.2.4 Passive observer error dynamics

The error dynamics of the observer is found subtracting the equations of the control plant model by the equations of the observer model. The estimation errors are defined as $\tilde{\eta}_w = \eta_w - \hat{\eta}_w$, $\tilde{\eta} = \eta - \hat{\eta}$, $\tilde{y} = y - \hat{y}$, $\tilde{v} = v - \hat{v}$ and $\tilde{b} = b - \hat{b}$. The resulting error dynamics is expressed in equation 9.2.

$$\dot{\tilde{\zeta}} = A_w \tilde{\zeta} - K_1 \tilde{y} + E_w W_w \quad (9.2a)$$

$$\dot{\tilde{\eta}} = R(\psi) \tilde{v} - K_2 \tilde{y} \quad (9.2b)$$

$$\dot{\tilde{b}} = -T_b^{-1} \tilde{b} - K_3 \tilde{y} + E_b W_b \quad (9.2c)$$

$$M \dot{\tilde{v}} = -D \tilde{v} + R^T(\psi_d) \tilde{b} - R^T(\psi_d) K_4 \tilde{y} \quad (9.2d)$$

$$\tilde{y} = \tilde{\eta} + \tilde{\eta}_w \quad (9.2e)$$

$$\tilde{\eta}_w = C_w \tilde{\zeta}_w \quad (9.2f)$$

The error dynamics can be written on a more compact state space form, equation 9.3. For further details regarding symbols and properties of equation 9.3 see Sørensen [2018].

$$M \dot{\tilde{v}} = -D \tilde{v} - R^T(\psi) C_0 \tilde{x}_0 \quad (9.3a)$$

$$\dot{\tilde{x}}_0 = A_0 \tilde{X}_0 + B_0 R(\psi) \tilde{v} + E_0 w \quad (9.3b)$$

The resulting system consist of a series of passive sub systems, resulting in a complete passive system. The resulting state space form of the error dynamics, can be represented by the block diagram in figure 9.9.

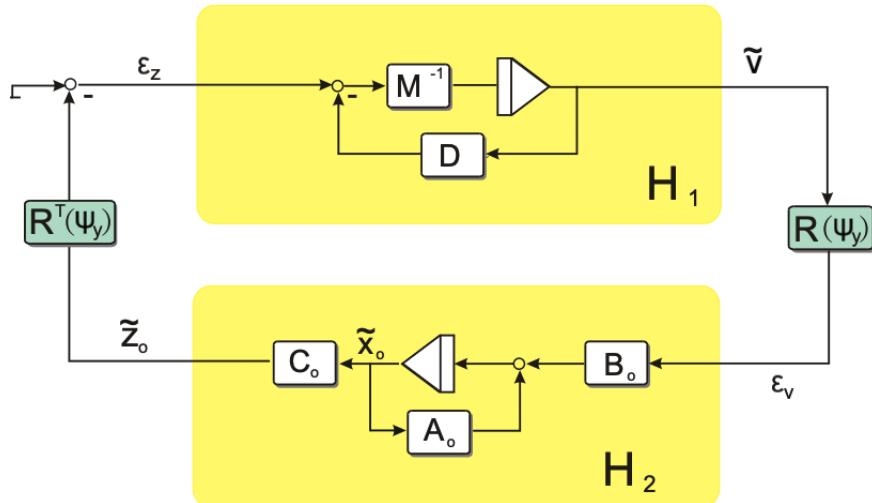


Figure 9.9: Resulting block diagram of the error dynamics of the non-linear passive observer. \mathbf{H}_1 are \mathbf{H}_2 linear systems interconnected through the bounded rotation matrix, $R(\psi)$, resulting in the complete system being passive. The figure is fetched from Sørensen [2018].

9.2.5 Tuning

The tuning is based on this paraphrased version of Theorem 7.1 in Sørensen [2018]:

"The nonlinear observer error dynamics (equation 9.1) is passive if the observer gain matrices $K_i (i = 1, \dots, 4)$ are chosen such that (9.3b) satisfies the KYP-Lemma."

For more details about the KYP-lemma see Sørensen [2018]. The gain matrices is presented in

equation 9.4.

$$\mathbf{K}_1 = \begin{bmatrix} \text{diag}\{k_1, k_2, k_3\} \\ \text{diag}\{k_4, k_5, k_6\} \end{bmatrix} \quad (9.4a)$$

$$\mathbf{K}_2 = \text{diag}\{k_7, k_8, k_9\} \quad (9.4b)$$

$$\mathbf{K}_3 = \text{diag}\{k_{10}, k_{11}, k_{12}\} \quad (9.4c)$$

$$\mathbf{K}_4 = \text{diag}\{k_{13}, k_{14}, k_{15}\} \quad (9.4d)$$

Further, in order to achieve passivity, \mathbf{K}_1 , \mathbf{K}_2 , \mathbf{K}_3 and \mathbf{K}_4 must be tuned accordingly. Sørensen [2018] using results from Fossen and Strand [1999], propose that equation 9.5 is used in tuning.

$$k_i = -2(\zeta_{ni} - \zeta_i) \frac{\omega_{ci}}{\omega_i}, i = 1, 2, 3 \quad (9.5a)$$

$$k_i = -2\omega_i(\zeta_{ni} - \zeta_i), i = 4, 5, 6 \quad (9.5b)$$

$$k_i = \omega_{ci}, i = 7, 8, 9 \quad (9.5c)$$

$$1/T_i \ll K_{3i}/K_{4i} < \omega_i < \omega_{ci}, i = 1, 2, 3 \quad (9.5d)$$

ω_{ci} is the filter cut-off frequency and ω_i is the dominating wave frequency. ω_i is often set equal to the wave peak frequency, $\omega_i \approx 2\pi/T_{pi}$. $\zeta_{ni} > \zeta_i$ are tuning parameters set between **0.1 – 1.0** (usually set to $\zeta_{ni} = 1$ and $\zeta_i = 0.1$). $T_i \gg 1$ is the bias time constant used to specify the limited integral effect in the bias estimator. The resulting observer gain matrices is presented in equation 9.2.5, 9.2.5, 9.2.5 and 9.2.5.

$$\mathbf{K}_1 = \begin{bmatrix} -2.160 & 0 & 0 \\ 0 & -2.160 & 0 \\ 0 & 0 & -2.160 \\ 1.566 & 0 & 0 \\ 0 & 1.566 & 0 \\ 0 & 0 & 1.566 \end{bmatrix} \quad (9.5)$$

$$\mathbf{K}_2 = \begin{bmatrix} 0.837 & 0 & 0 \\ 0 & 0.837 & 0 \\ 0 & 0 & 0.837 \end{bmatrix} \quad (9.5)$$

$$\mathbf{K}_3 = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix} \quad (9.5)$$

$$\mathbf{K}_4 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \quad (9.5)$$

10 Simulation Results

10.1 Simulation 1 - Environmental Loads

The first simulation aims to test the implemented environmental models, and show vessel behaviour when influenced by these loads. All thrusters and controllers are turned off for this simulation, so the supply-ship will only drift. The simulation was run for 300 seconds with current from east at average speed of 0.2 [m/s], wind from the north with velocity 10 [m/s] and waves from northeast with significant wave height $H = 2.5\text{m}$ and peak wave period $T_p = 9[\text{s}]$

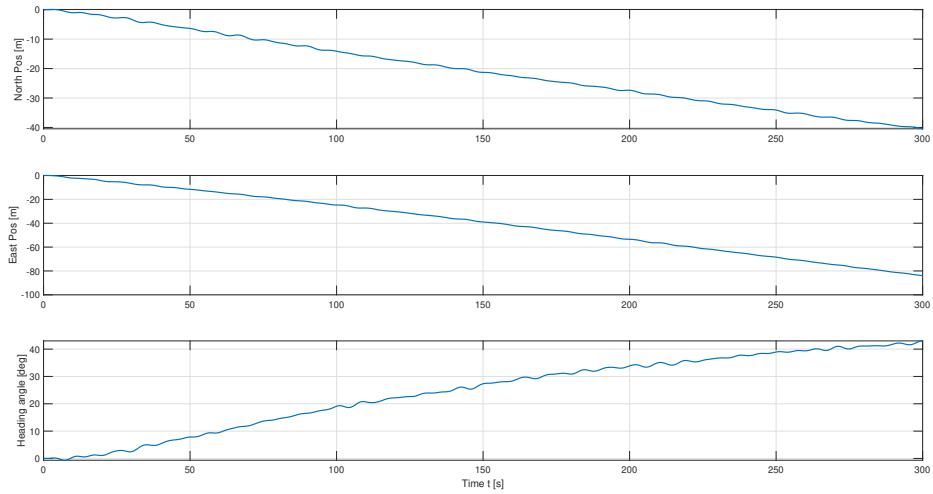


Figure 10.1: Position and heading over time for Simulation 1

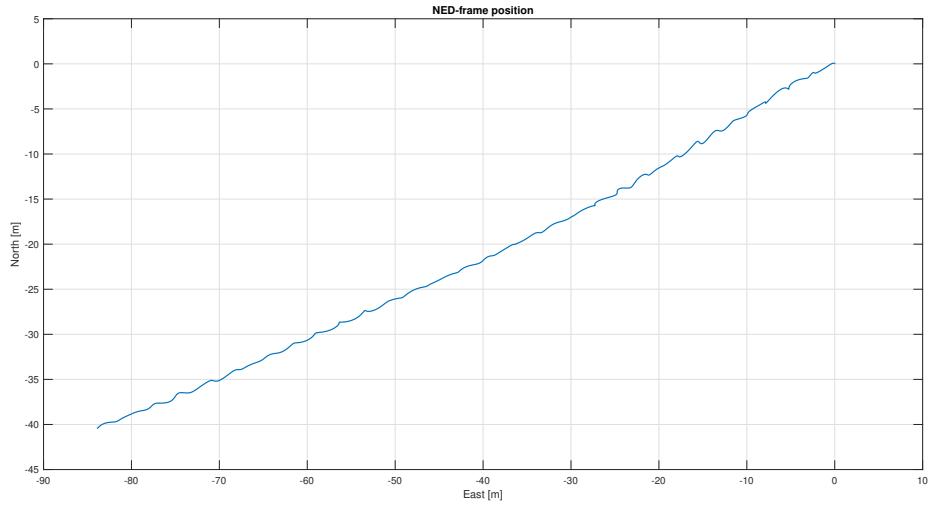


Figure 10.2: Position in NED-frame for Simulation 1

10.2 Simulation 2 - DP and Thrust Allocation

The second simulation tests the Thrust Allocation Scheme of the DP-system. All environmental forces are to be disabled, and a box test is to be run first with all thrusters active and then with

failure in thrusters 2 and 5.

10.2.1 All actuators active:

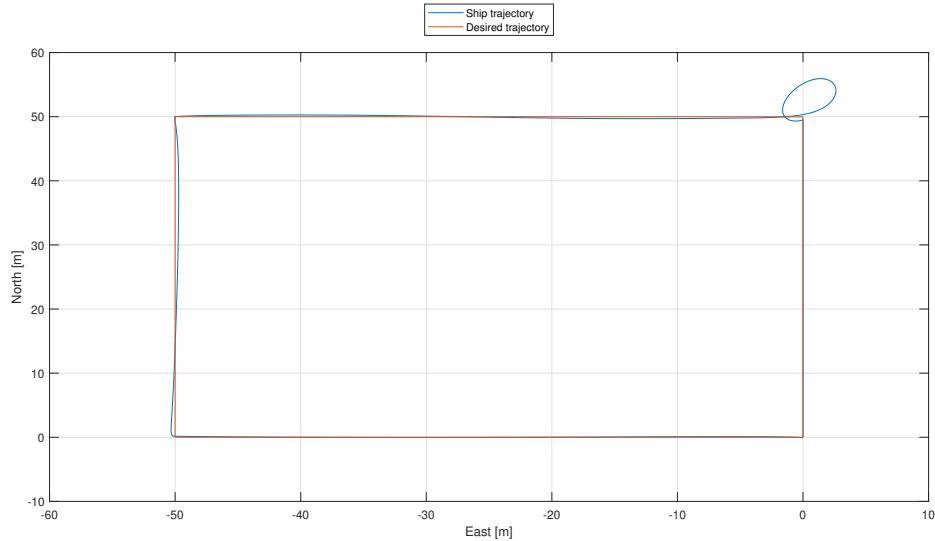


Figure 10.3: Position in NED-frame for Simulation 2.1

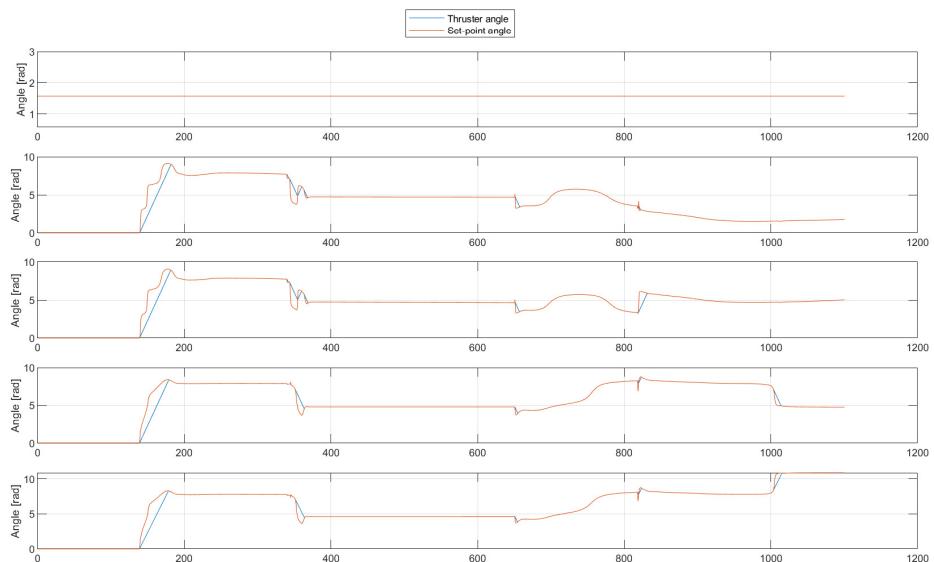


Figure 10.4: Thruster angle for Simulation 2.1

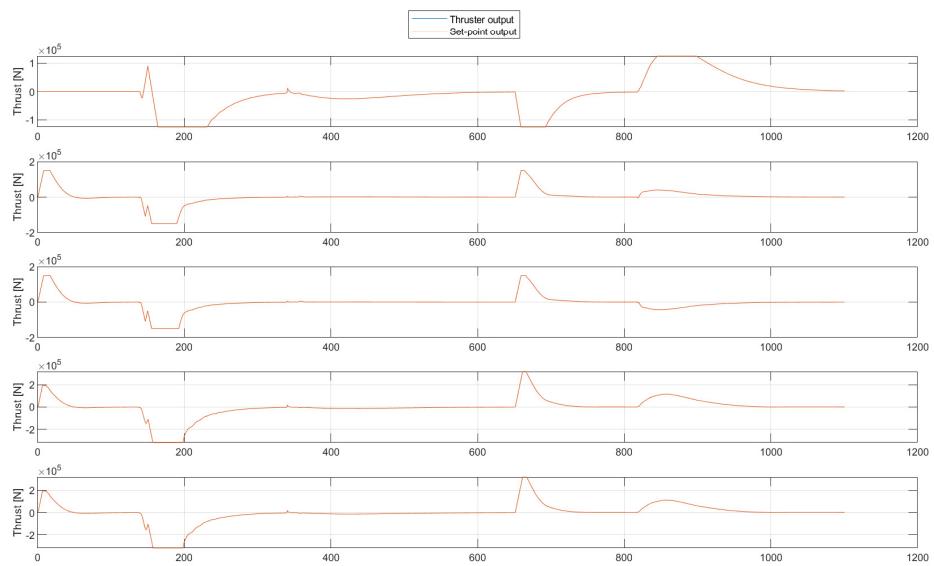


Figure 10.5: Thruster output for Simulation 2.1

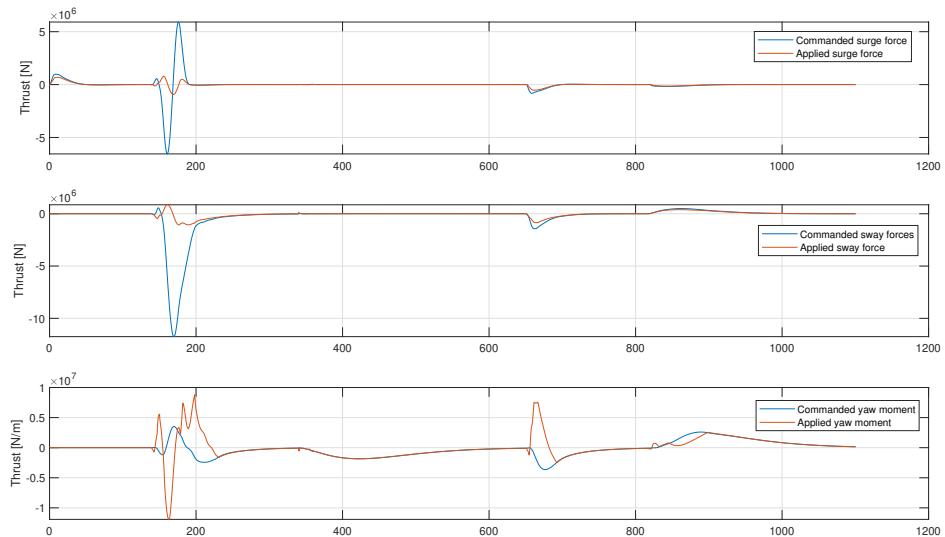


Figure 10.6: Thruster forces for Simulation 2.1

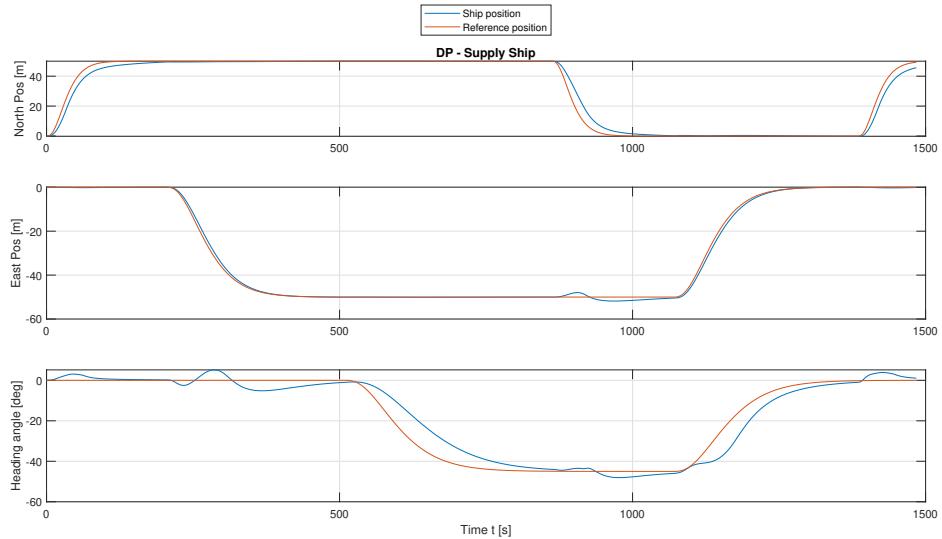


Figure 10.7: Position and heading over time for Simulation 2.1

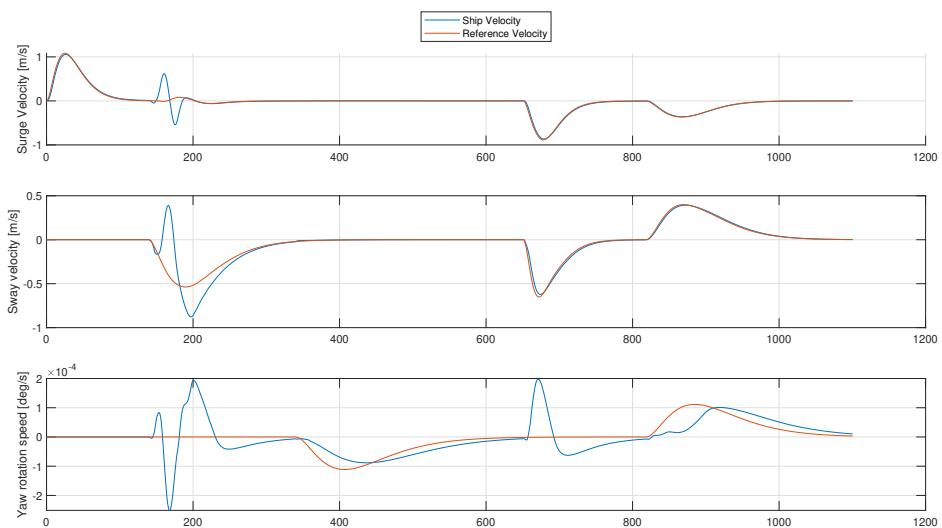


Figure 10.8: Velocities over time for Simulation 2.1

10.2.2 Failure in thrusters 2 and 5

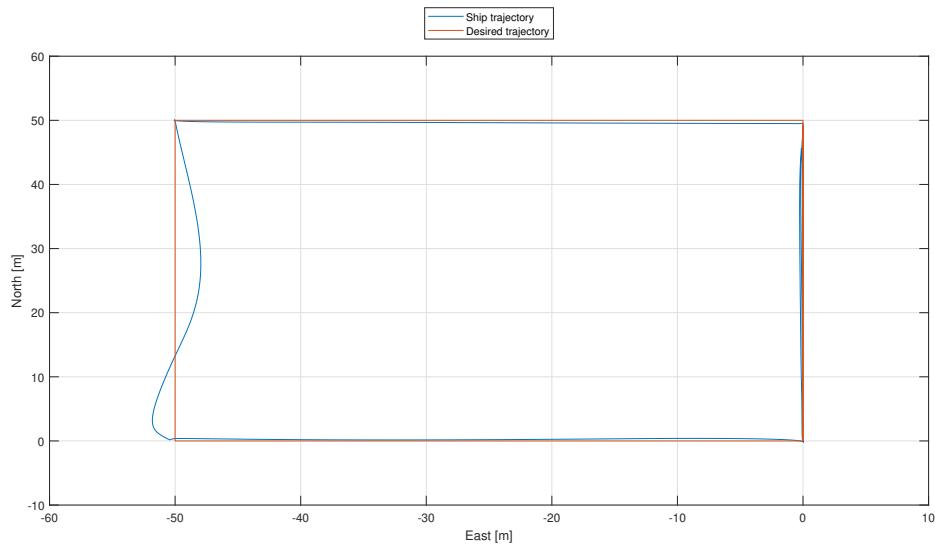


Figure 10.9: Position in NED-frame for Simulation 2.2

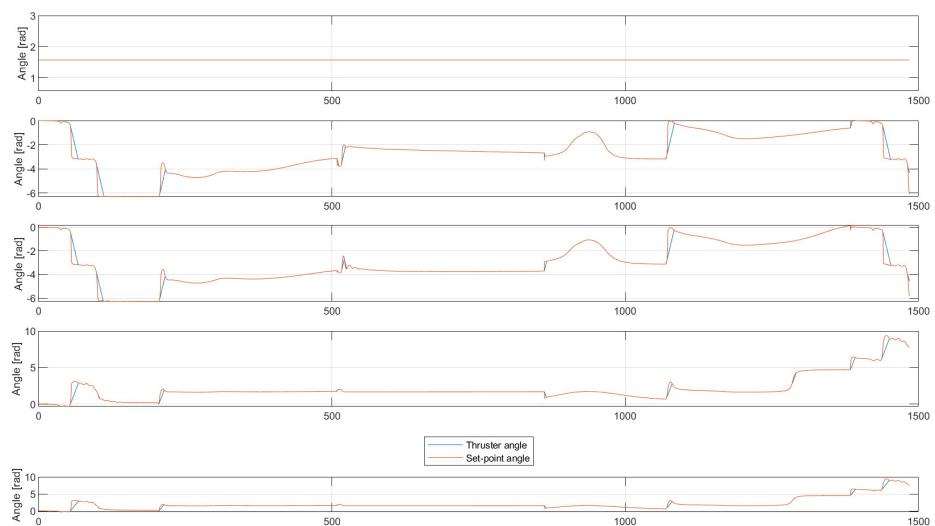


Figure 10.10: Thruster angle for Simulation 2.2

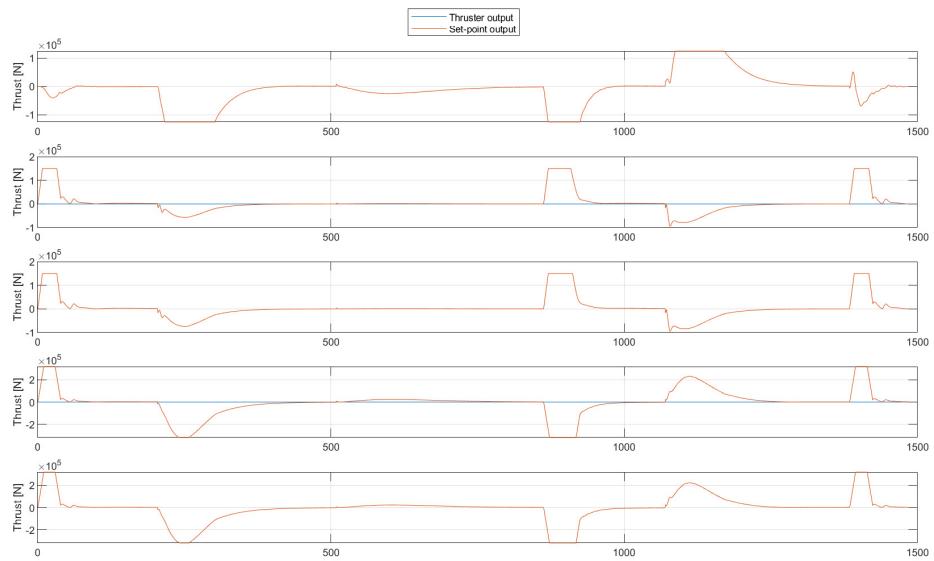


Figure 10.11: Thruster output for Simulation 2.2

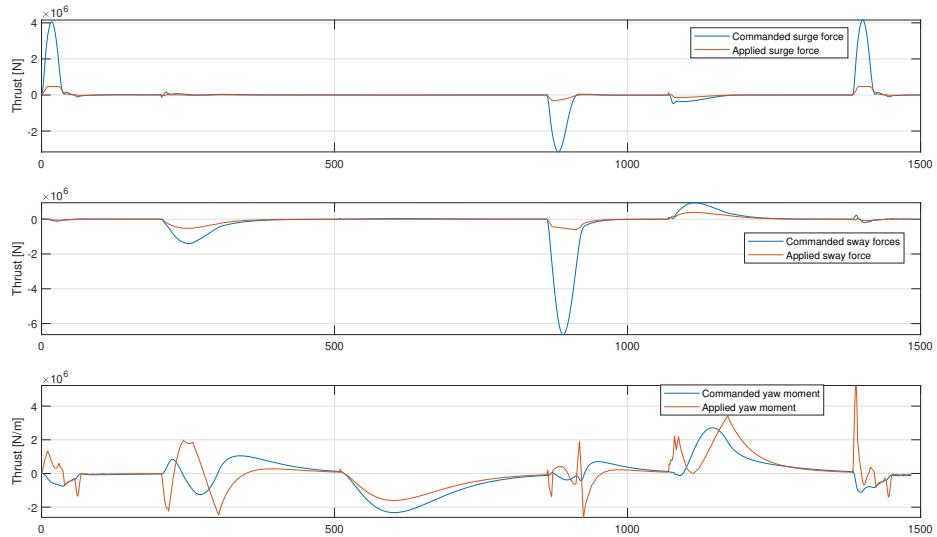


Figure 10.12: Thruster forces for Simulation 2.2

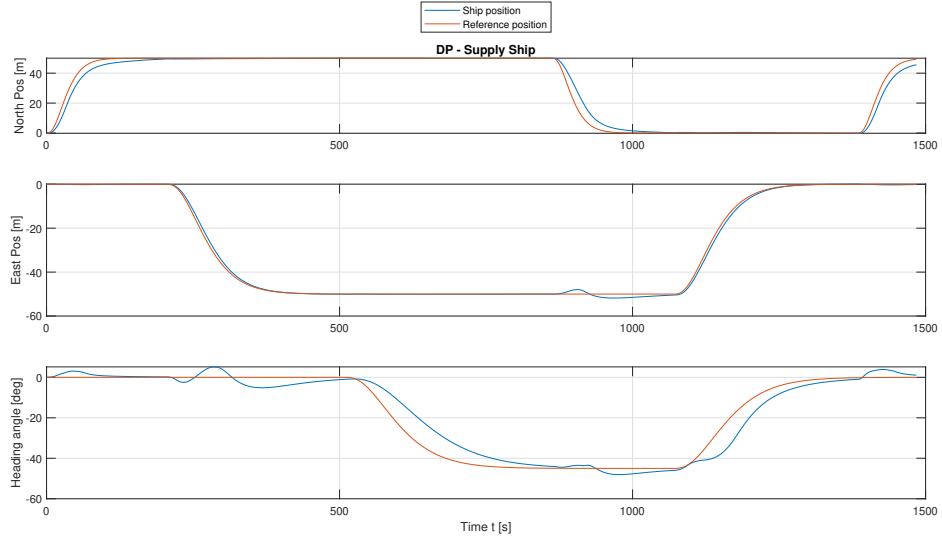


Figure 10.13: Position and heading over time for Simulation 2.2

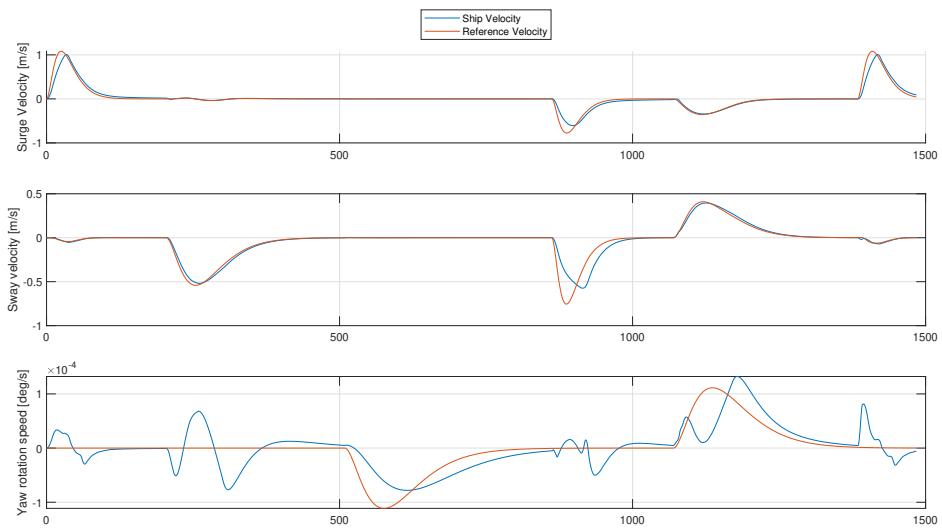


Figure 10.14: Velocities over time for Simulation 2.2

10.3 Simulation 3 - DP and Environmental Forces

These simulation aims to examine the effects of environmental forces on the DP-system. No observer is to be used, and environmental forces are activated with the same parameters as in 10.1.

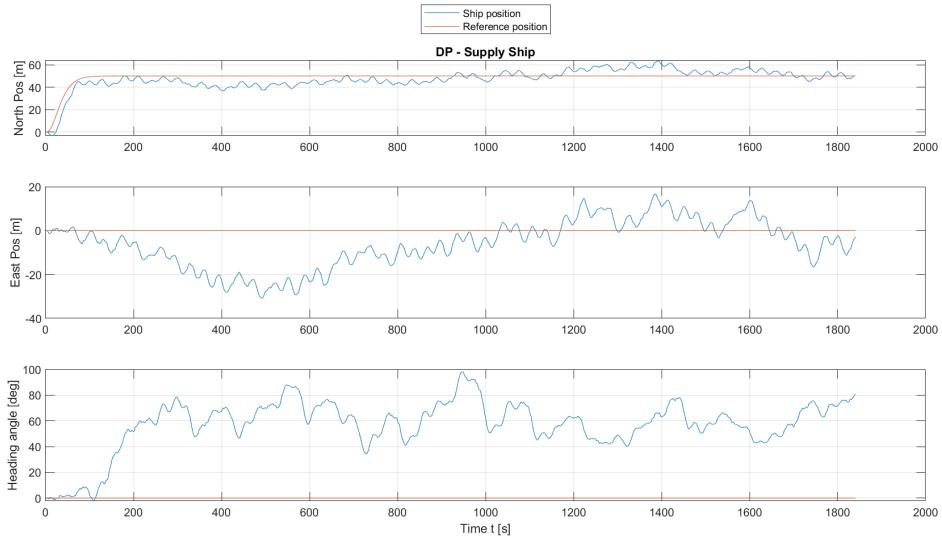


Figure 10.15: Position and heading over time for Simulation 3

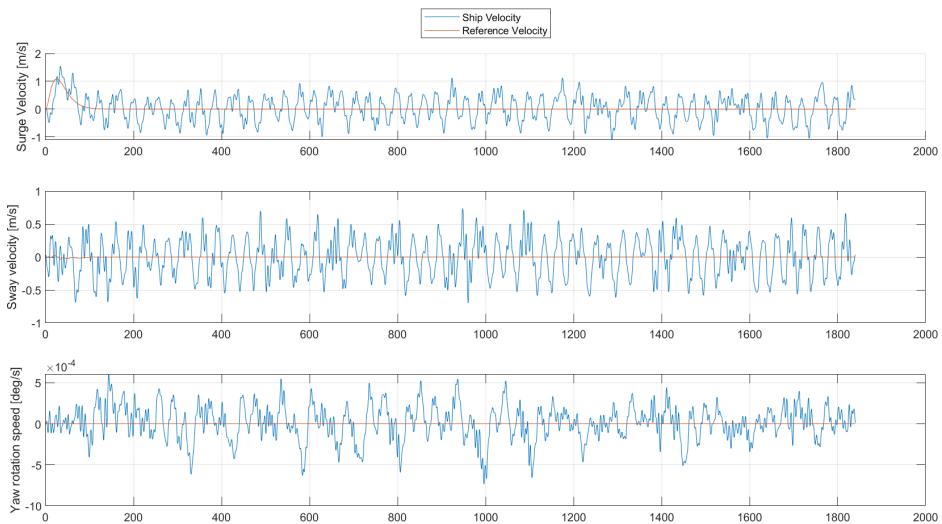


Figure 10.16: Velocities over time for Simulation 3

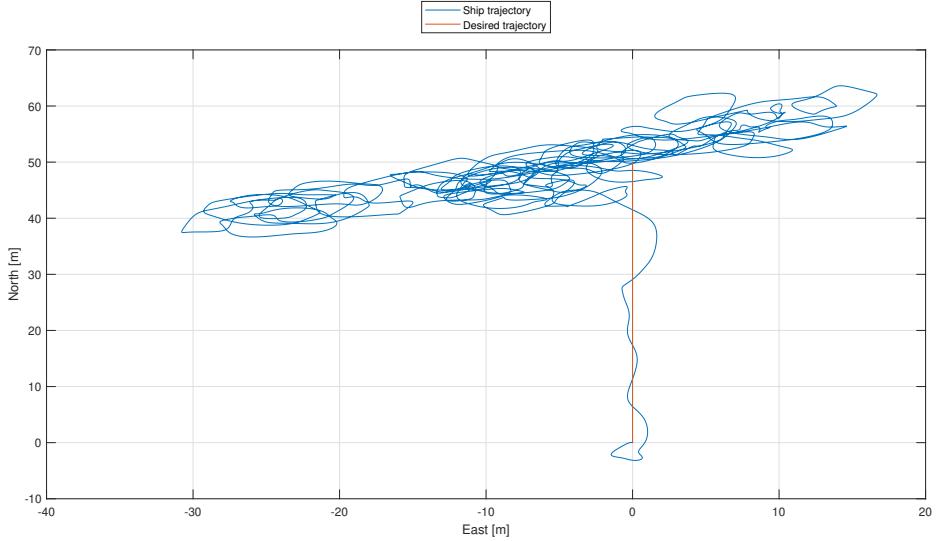


Figure 10.17: Position in NED-frame for Simulation 3

10.4 Simulation 4 - Observer selection

Simulation 4 compares the two implemented observers. Using the same environmental forces, as in 10.1, we lock the DP-forces to $\tau = [1 \ 1 \ 1] \cdot 10^4$. Each observer is subject to two tests, one with all environmental forces and one without wave forces so that we can also compare observer output with measurements. Both tests were run at a duration of 150 seconds.

10.4.1 Extended Kalman Filter

A comparison with and without wave forces enabled is shown in the Figures

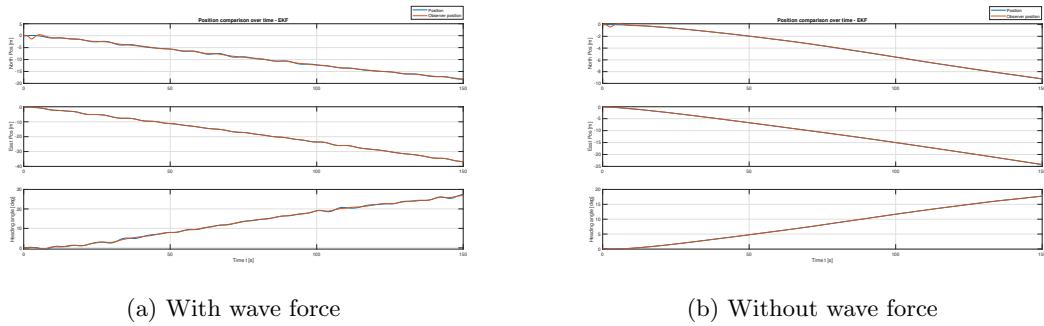
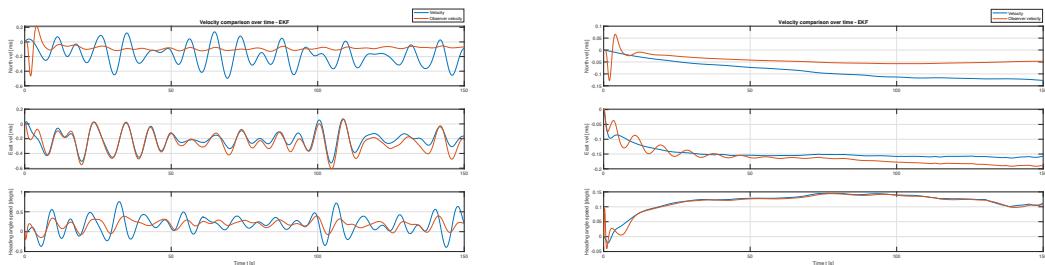


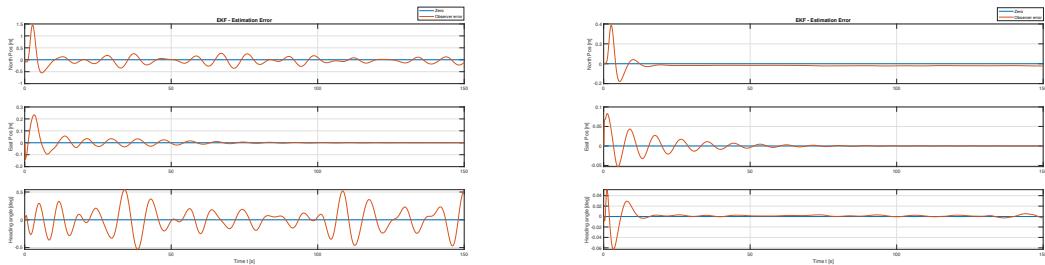
Figure 10.18: Position comparison



(a) With wave force

(b) Without wave force

Figure 10.19: Velocity comparison

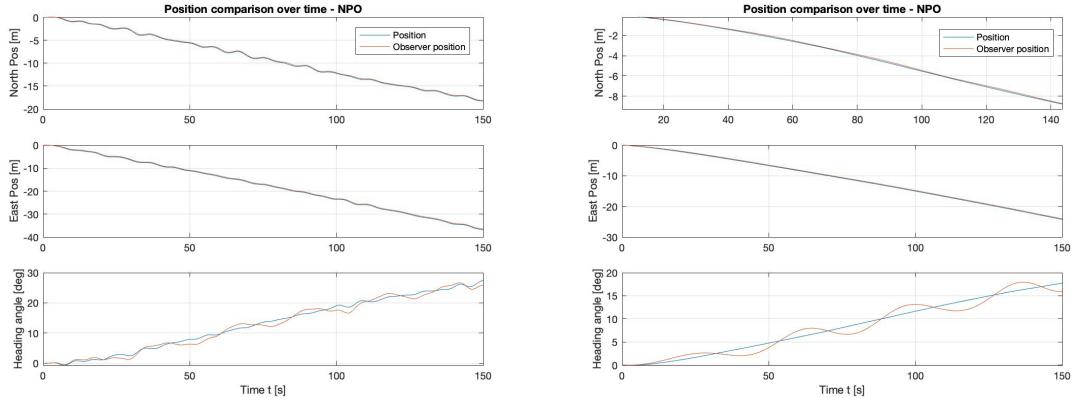


(a) With wave force

(b) Without wave force

Figure 10.20: Position Error comparison

10.4.2 Non-Linear Passive Observer



(a) With wave force

(b) Without wave force

Figure 10.21: Position comparison

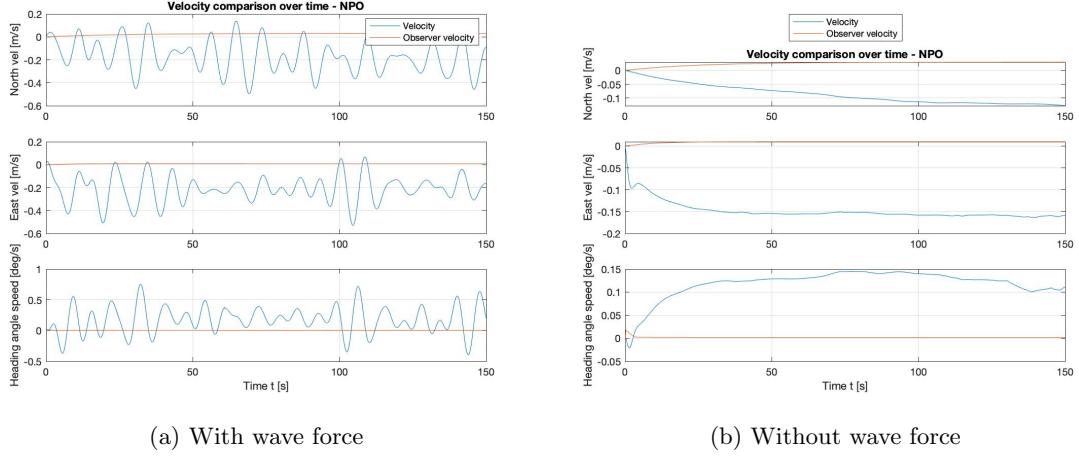


Figure 10.22: Velocity comparison

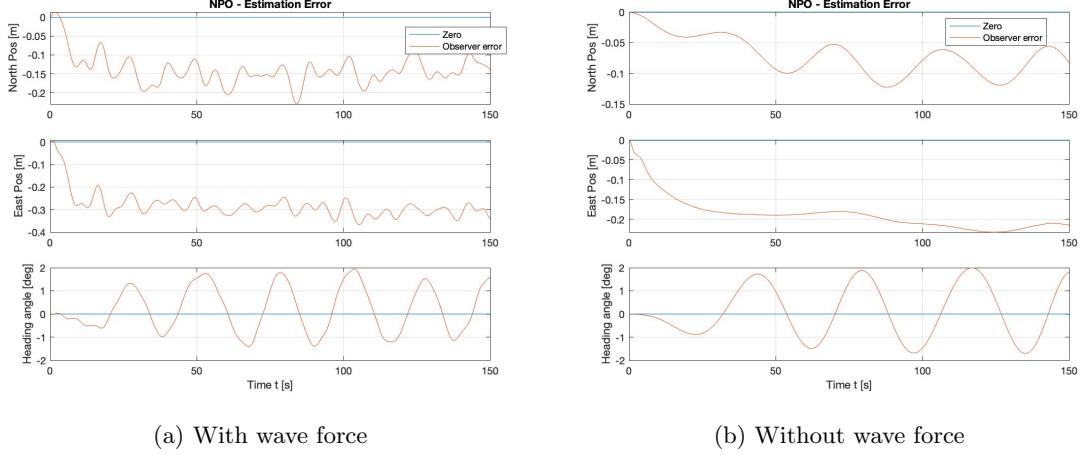


Figure 10.23: Position Error comparison

10.5 Simulation 5 - Complete simulation

This simulation runs the four corner test with a complete DP-system and environmental forces.

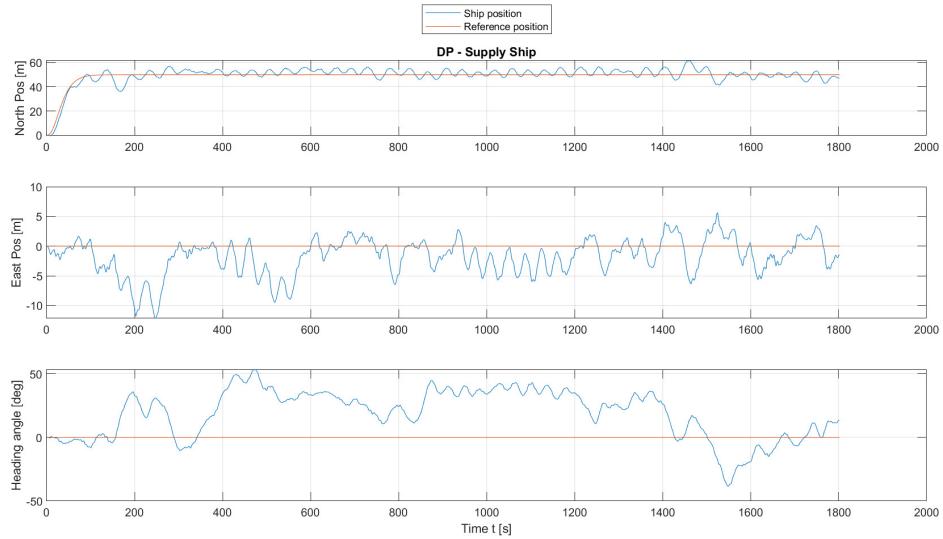


Figure 10.24: Position and heading over time for Simulation 5

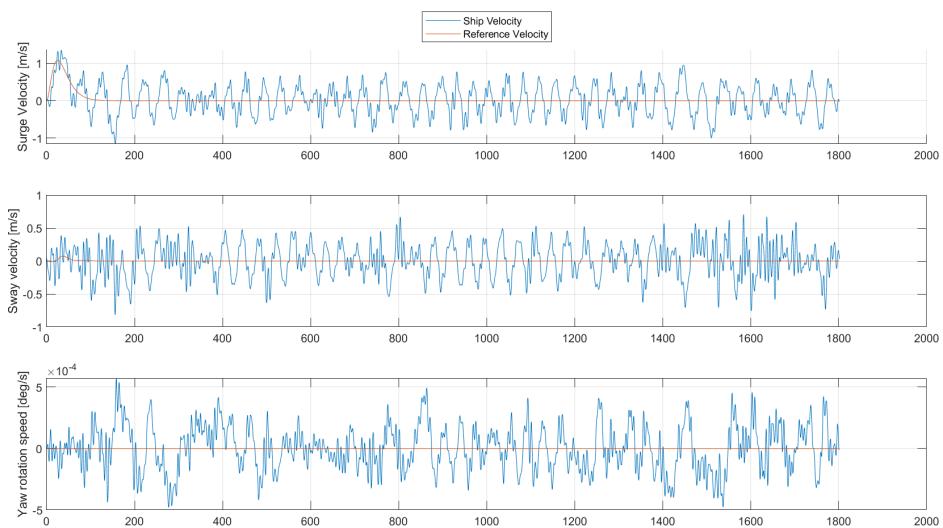


Figure 10.25: Velocities over time for Simulation 5

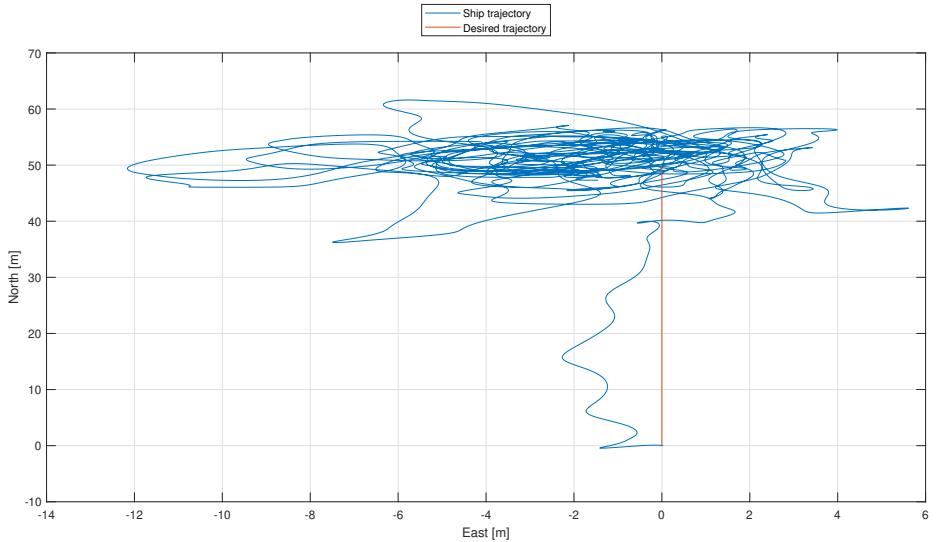


Figure 10.26: Position in NED-frame for Simulation 3

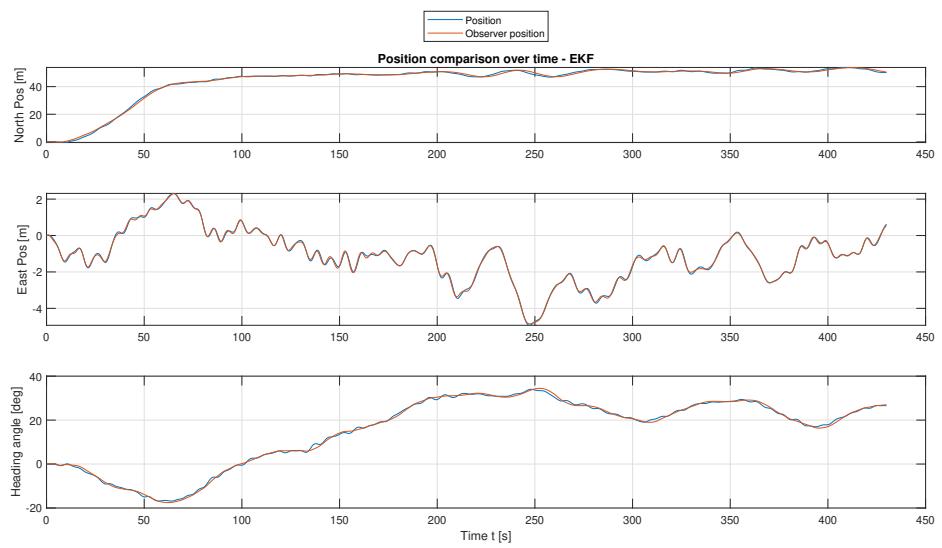


Figure 10.27: Observer position for Simulation 5

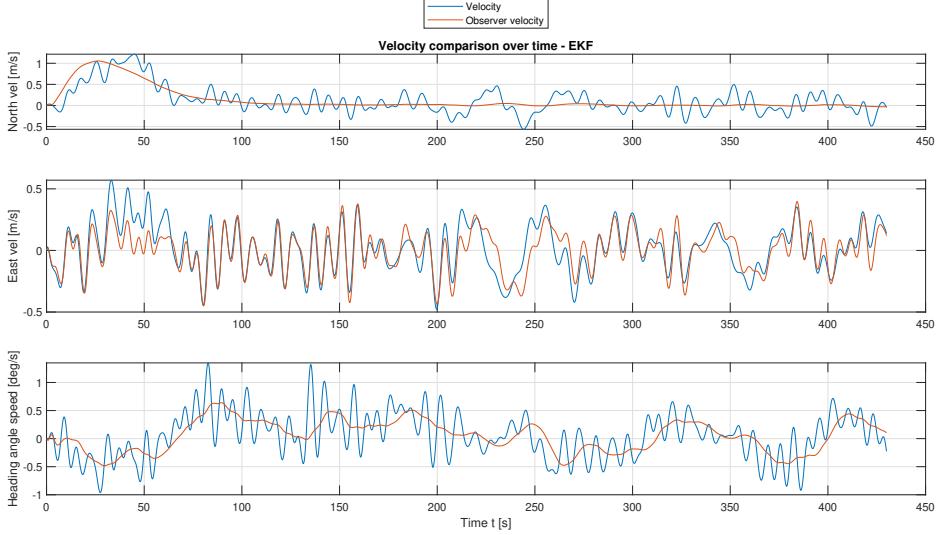


Figure 10.28: Observer velocity for Simulation 5

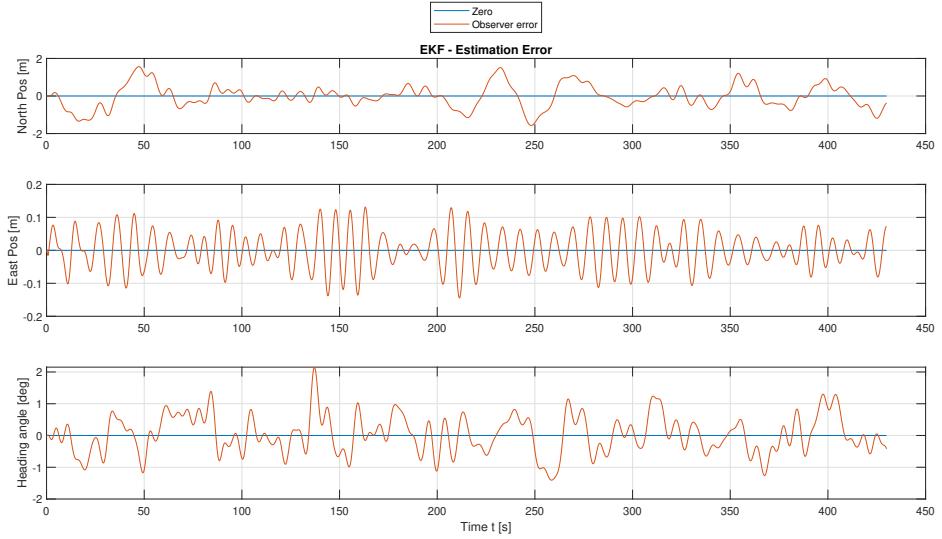


Figure 10.29: Observer position error for Simulation 5

10.6 Simulation 6 - Capability plot

The simulation is here run a total of 36 times. The environmental forces are all applied from one direction, starting from north. Then each simulation they are incremented by 10 degrees and the simulation is repeated. This way, it is possible to see how the system reacts with environmental forces coming in from specific angles.

The implementation consisted of saving the thrust data for a range of simulations run in series. Once one was complete, the angles would be incremented, and the simulation would run again until the whole circle was complete.

Unfortunately, some issues were encountered at this step. The environmental models did not seem

to take kindly to the changing parameters, and the simulation would halt, throwing an indexation error, when the indices in the wind-model would reach some large negative value, likely an overflow error. It was tried to find a few past this issue, but no solution was uncovered. The capability plot code is still submitted alongside the report, but no plot could be produced from the simulations.

10.7 Simulation 7 - Observer robustness

To check the robustness of the observer, the environmental forces are changed to more extreme conditions. Significant wave height is changed to 8 meters and the period to 13 seconds. The vessel is to keep stationary at $\eta = [0, 0, 0]$ with a simulation time of 1000 seconds.

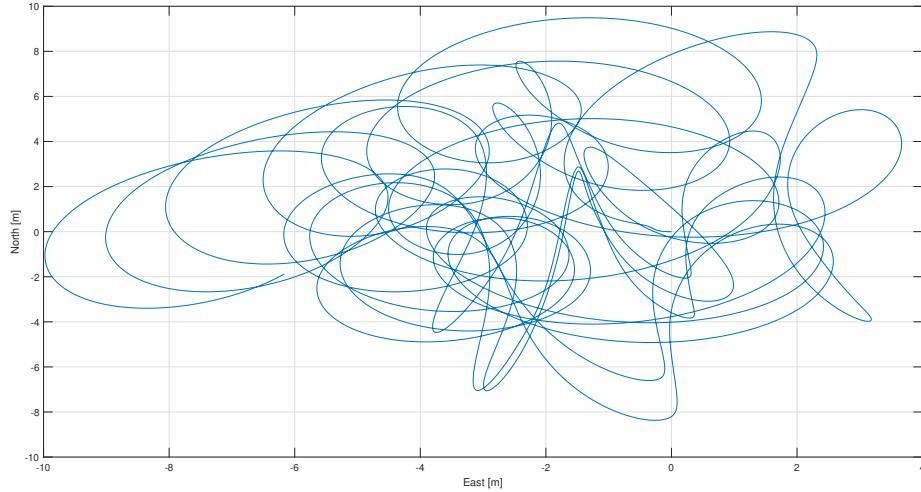


Figure 10.30: Position in NED-frame

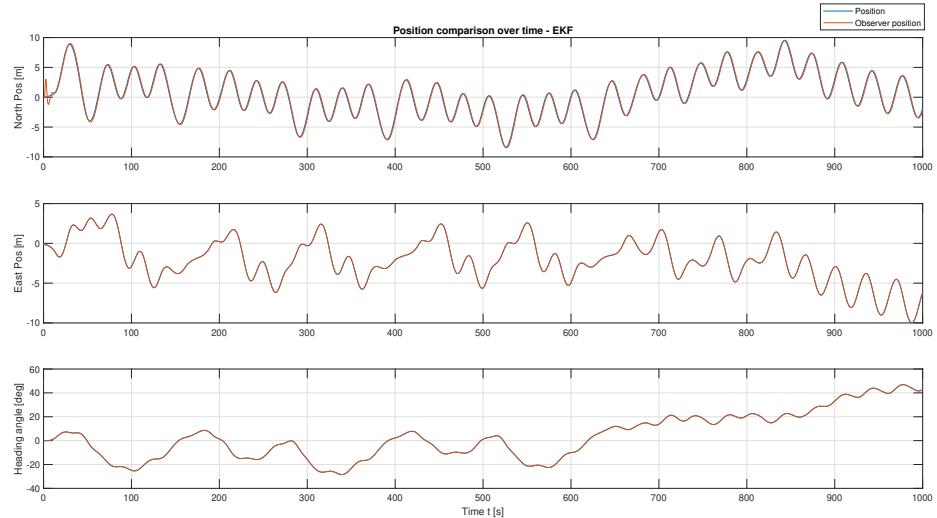


Figure 10.31: Position and heading changes, EKF and actual

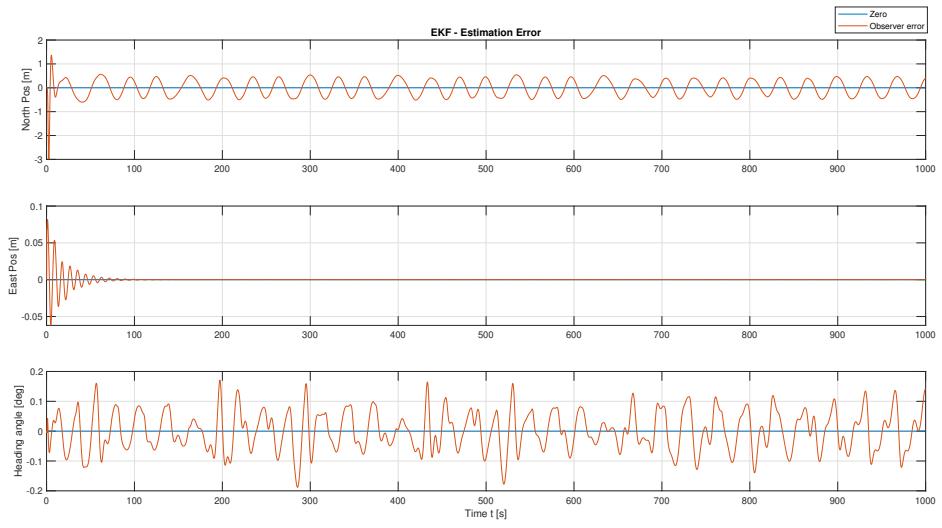


Figure 10.32: Position error over time, EKF and actual

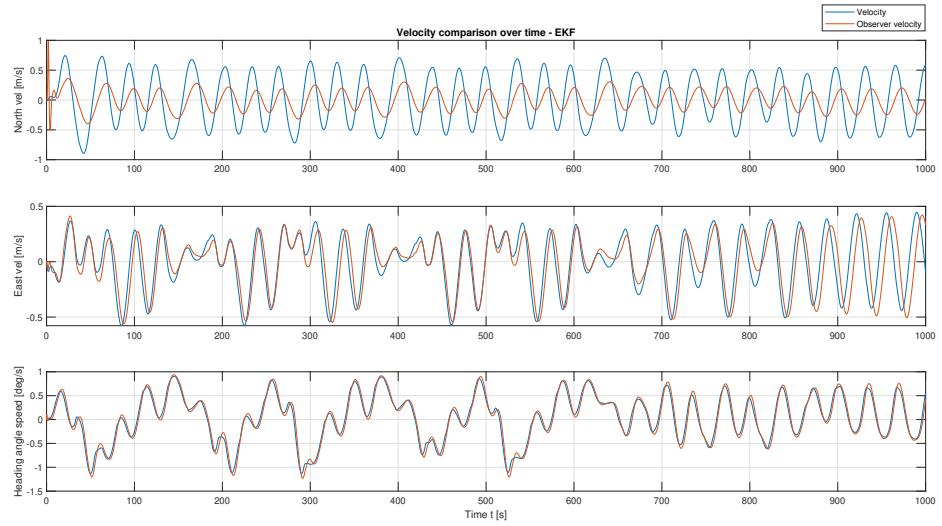


Figure 10.33: Velocity and heading changes, EKF and actual

11 Discussion

11.1 Simulation 1

Without any thruster or controller action, it is clear to see that the environmental loads have a solid impact on the vessel's position. In the course of five minutes, the vessel drifts around 80 meters westward and 40 meters southward. This is accompanied by a rotation in yaw that reaches up to around 40 degrees. Taking the directions of the environmental loads into account, this trajectory makes sense. The vessel drifts a distance approximately equal to its own length in the span of 300 seconds, which seems reasonable, assuring that the magnitude of the environmental loads is realistic as well.

The slowly-varying parts of the environmental models can also be observed in the plots, especially for heading angle in Figure 10.1 and in Figure 10.2.

11.2 Simulation 2

At a first glance at the results, some things stick out as concerning. First of these is the "pirouette" the vessel makes at the when reaching the first corner of the box test. The exact cause of this is not completely clear, but we suspect it might be due to thruster-thruster interaction. As can be observed in figure 10.4 around the same time the "loop" happens, the setpoint changes and all azimuths experience an identical rotation of almost 8 radians. This is also a discrepancy as the angle should be mapped top $[-\pi, \pi]$. Most likely this is due to the system not correctly mapping angles on this interval.

The reason thruster-thruster interaction is suspected is due to the results from the second simulation, where 2 azimuths are disabled. As observed in figure 10.9 the ship no longer behaves oddly at the top right corner. In this case each of the bow and stern azimuths have the thruster beside it disabled which might hinder the thrust generated from one interrupting the other. If indeed thruster-interaction is the cause, a solution may be to implement a dead-zones for the thrusters or constraints such that the thrusters can not work in parallel. I.e $\alpha_2 = \alpha_3 + \frac{\pi}{6}$

Despite these discrepancies, the remaining results are quite satisfactory. The first thing to note is that the box test is completed in around 1250 seconds with all actuators active. Compared to part one that is almost 600 seconds faster. This is likely due to a better tuned controller. Despite the loop in the right corner we see in 10.3 that the vessel completes the journey while tightly holding to the set-points generated by the reference model. It struggles slightly more with disabled thruster as is observable 10.9, but the test is finished at almost the same time.

In figures 10.4 and 10.10 the thruster angles are plotted against the commanded angles. The topmost is the tunnel thruster so it is always constant. The actual angles follow the commanded angles nearly perfectly for the entirety of the simulations for both cases. This is expected since the allocation-algorithm checks the angular rate constraints before outputting the commands.

The commanded thrust outputs are identical to the actual outputs of the thrusters, as is displayed in 10.5 and 10.11. The stern azimuths seem to act quite identical as well, and the same for the bow thrusters.

Figures 10.6 and 10.12 represent the total forces applied on the vessel, and plot the commanded outputs from the PID against the actual applied. We observe spikes in both case-studies, especially when the setpoint changes. For the simulation with all actuators enabled the PID-controller commands large fluctuating force in all directions. This is around the time the vessel does the "loop", and may be a contribute to the unwanted behaviour. Despite the spikes the applied forces mirror those of the commanded for the majority of the simulation.

The same goes for the velocity tracking, illustrated in figures 10.8 and 10.14. Here we also see a clear effect of the disabled thrusters, as the vessel ship velocity struggles to follow the reference

velocity, especially in yaw.

11.3 Simulation 3

Figure 10.15 to 10.17 show the results for simulation 3. In total the simulation was run for 1800 s, or 30 real life minutes. It is abundantly clear from 10.17 that the ship is not able to complete the box test. The velocity and positional plots 10.15 and 10.16 show large fluctuations and the vessel struggling to follow the reference path. This is due to both the low frequency and the wave frequency components entering the feedback loop. The system is trying to react to every tiny wave induced disturbance, resulting in the oscillating movements. Considering there is no observer to filter out the WF-component, this result is as expected.

11.4 Simulation 4

11.4.1 Extended Kalman Filter

It can be seen in Figure 10.21 that the positional deviations are not very considerable in the grand scheme of things. Apart from a few discrepancies right at the start, the Kalman filter estimate corresponds with the actual position quite closely for all the three states. This bodes well, since comparatively to the magnitudes we expect the states to take, the EKF yields a great estimate.

In order to better study the deviations between estimate and reality, the error between the position and position estimate are plotted in Figure 10.20, making it easier to see the small patterns of discrepancy between the two. From there it can be seen that the slight deviation in north position is around 1.5m, and seems to be mostly caused by the waves, evident by its reduction in the non-wave case. The estimation error in the heading angle also seems to depend mostly on the waves. In the non-wave case (Figure 10.20b), the current and wind errors seem to be quickly filtered out, leaving the states very well estimated after only a minute's time.

The velocity estimates in Figure 10.22 are a little more messy, but overall a decent result. It was attempted to reach a sort of middle ground between good estimates and oscillation filtering, which proved difficult when the waves were involved. Without waves in Figure 10.19b it seems the estimator has some trouble working out the velocity in north direction, but not by a devastating margin, so this was ignored. Otherwise a decent correspondence was achieved between the estimator and reality. Since there are no velocity measurements available, the velocity estimates are harder to determine with good precision, and depends more on the model implemented in the Kalman Filter.

11.4.2 Non-linear Passive Filter

In Figure 10.21a and 10.21b are the true position of the vessel and the estimated position plotted. The positional deviations does not seem to be very big. Apart from the estimated heading angle, the Non-Linear Observer estimate corresponds with the actual position. The system is sensitive to small variations in the heading angle. Therefore, the estimation error in the implemented Non-Linear Observer is not considered to be sufficient enough.

The error between the position and position estimate are plotted in Figure 10.23. The observer seemed to estimate the North Position with a greater accuracy when the system was under the influence of wave forces, than when the system was not. This could be due to the fact that the observer is a wave estimator, or due to implementation or tuning errors. However, this is not the case when estimating the east position or the heading angle, then the observer performs marginally better without acting wave forces.

In Figure 10.22a and 10.22b are the velocity estimates presented. In both cases, with and without influence from wave forces, the observer seem to neglect the speed and estimate it to be zero. In conclusion he velocity estimation does not seem to be sufficient.

Overall, this implementation of the Non-linear observer is not sufficient enough for further simulations. The reason for why the observer is failing, may be insufficient tuning or implementation error. A possible implementation error may be that the observer send the sum of slow varying estimations and wave estimations out as output, and this is used as input in the controller. This way the observer does not filter out high frequency wave forces, but may increase the magnitude of the impact that the high frequency wave forces has on the system.

11.4.3 Observer Choice

Considering the results achieved and time constraints on tuning and troubleshooting, it was decided to move on using the Extended Kalman Filter. Although it has some difficult tuning properties, it demonstrated decent correspondence with the actual state variables.

11.5 Simulation 5

As is clear form the figures 10.24, 10.25 and 10.26, the observer is not doing a good enough job at filtering out the WF-components. This results in the vessel not being able to complete the box-test, and a behaviour akin to simulation 3. At first the observer was suspected of faults, but figure 10.27 and the following plots show the observer having an actual effect. Thus this likely a problem of tuning. With finer tuning of the filter the performance would be better. More specifically tuning the Q-matrix values for wave-forces might have an effect.

11.6 Simulation 6

The simulation was not completed, so no thrust capability plot was produced. There was however an attempt at a capability plot, which is included with the SIMULINK and Matlab files to illustrate the thought process that went into this problem.

11.7 Simulation 7

It can clearly be seen that the control system cannot keep up with the extreme environmental loads it is subject to in this case. The high frequency waves, combined with the large significant wave height means that the system becomes far too destabilized to recover. In Figure 10.30 one can see that the vessel seems to be looping around in some chaotic pattern, continuously trying to correct itself.

This experiment clearly demonstrates that rough sea states are difficult to predict and model with the tools at hand. The DP-system (both observers and controller) could possibly be tuned and calibrated to handle this sea-state, but through some experimentation, no such configuration was found, and the vessel seemed to have a hard time station-keeping.

The observer data seems to indicate that the observer is following the real state quite well. In Figure 10.32 we see that the error between estimated and actual state is quite small, and seems to hold decently over time, with little discrepancy observed in Figure 10.31. There are still some error present however, especially in surge. The velocities in Figure 10.33 also seem to correspond quite well. This indicates that the Kalman filter functions as intended in measuring and estimating, even with this seastate.

12 Conclusion

Based on the mathematical model of the vessel provided in SIMULINK, a DP system model was created and simulated. In the first part a PID-controller, reference model and basic environmental model were implemented. In the second part this was expanded with a number of new components and configured to work together with acceptable performance.

First a more extensive environmental model was created, describing a much more realistic sea-state. This included loads from wind, waves and current

Then a thrust-allocation system was developed as part of the controller. This utilized an unconstrained extended thrust-allocation method, although several other methods were initially explored. The thrust allocation system provided satisfactory results after some optimizations efforts were explicitly implemented. After the thrust allocation was finished, the controller was also returned, which yielded a much faster and accurate response than what was achieved in Part 1.

Next to different observers, the Extended Kalman filter and the Non-Linear Passive Filter, were implemented and tested to varying degrees of success. The latter observer, was faulty in implementation and therefore could not provide satisfactory results. The Kalman filter however functioned properly, but was unfortunately not tuned properly. The results were therefore also considered sub-optimal.

The reference model and controller were kept mostly unchanged, but were subject to tuning changes once the thrust-allocation was implemented.

While the model was implemented and tuned somewhat, some simulations were unfortunately not possible to complete with any meaningful results due to the faulty implementation. Still, the ones that were completed yielded great insights into the operation of DP-systems, both advantages and shortcomings. Additional tuning and experimentation could have helped mitigating some issues especially in the latter simulations, whilst more knowledge of the model and its internal workings might have eliminated some others. Despite this we have learned much from this project about the workings of DP-systems and its implementation and are overall satisfied, regardless of the outcomes.

Bibliography

- Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, 2011.
- Thor I. Fossen and Jann Peter Strand. Passive nonlinear observer design for ships using lyapounov methods: full-scale experiments with a supply vessel. *Automatica*, 48:3–16, 1999.
- O.J. Sørdalen. Optimal thrust allocation for marine vessels. 1997.
- Asgeir J. Sørensen. *Marine Cybernetics towards Autonomous Marine Operations and Systems - Lecture Notes*. Department of Marine Technology, NTNU, 2018.