Boat lab Report

Group 51
Are Fossli Viberg 756388
Johanne Kalland 768709
Karoline Halvorsen 765265

October 2017

NTNU – Trondheim
Norwegian University of
Science and Technology

Contents

Introduction

The purpose of this assignment was to partly model and simulate a continuous system influenced by stochastic signals. To do this we had to use basic identification techniques on parameters that were not specifically given, use basic control theory to design a simple autopilot to control the system and implement a discrete Kalman filter for wave filtering and estimation of disturbances by using both Matlab and Simulink.

This report is organized as follows: Section 1 contains the identification of the boat parameters, section 2 the identification of the wave spectrum model where we found the estimate PSD function and analytic PSD for ψ_w, plus an analytic expression of the transfer function of the wave response model. Section 3 contains the control system design where we designed an autopilot for the ship, such that it would be able to follow a desired course. Section 4 we calculate the observability matrices for different variations of the system and in section 5 we implemented a discrete Kalman filter to estimate the bias, the heading and the high-frequency wave induced motion on the heading. In addition to this the Appendix contains the Matlab code, Simulink diagrams and the plots.
The conclusion for the lab report is that the Kalman filter gives us good enough estimations, that we can make a good autopilot that works in rough weather conditions.

Preliminary

Summary of the complete system

In the navigation of the cargo ship several reference frames are used. Here we only consider two coordinate systems 'NED' and 'BODY'. NED is a coordinate system which lies on top of earths longitudinal and latitudinal axes with the z axis pointed downward into the center of the earth. BODY is places along the ship, with the x-axis from aft to fore, and y-axis from port to starboard side and z-axis from top to bottom. fig. 1 illustrates the BODY and NED reference frames.

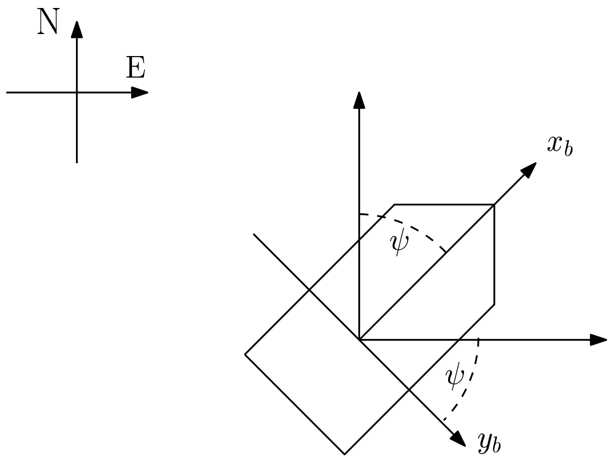


Figure 1: BODY and NED reference frames

A dynamical model of the ship is represented by eq. (0.1a) and eq. (0.1b) where the speed is low, such that some nonlinear terms as negligible

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{0.1a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{\theta} + \mathbf{w} \tag{0.1b}$$

where

- $\mathbf{M}$ - is the systems inertia matrix.
- $\mathbf{C}$ - is the Coriolis-centripetal matrix.
- $\boldsymbol{\tau}$ - is the vector of control inputs.
- $\mathbf{w}$ - is the vector of environmental disturbances.
- $\boldsymbol{\eta}$ - is the vector of NED positions $[x\ y\ \psi]$, where x is the position in the north direction, y is the position in the east-direction, and ψ is the angle between the north direction and the x_b axis. ψ is positive clockwise.
- $\boldsymbol{\nu}$ - is the vector of BODY velocities $[u\ v\ r]$. Where u is the velocity in the x-direction, v the velocity in the y-direction and r is rotation velocity about the z-axis.

The disturbances we will implement in our system are waves and current. The waves are considered to be high-frequency disturbances (white noise), while the current is a slowly varying disturbance. The waves representation corresponds to a spectral factorization of the wave spectrum and cam be modelled as a damped harmonic oscillator

$$\begin{bmatrix}\dot{x}_{w1}\\ \dot{x}_{w2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_0^2 & -2\lambda\omega_0\end{bmatrix}\begin{bmatrix}x_{w1}\\ x_{w1}\end{bmatrix} + \begin{bmatrix}0\\ K_w\end{bmatrix} w_w$$

$$y_w = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_{w1}\\ x_{w1}\end{bmatrix}$$

For the current, we will assume the only effect acting on it is the rudder angle bias. Which is modelled as in eq. (0.3e) where w_b is the Gaussian white noise. Here it is important to note that the ship can only deviate from the reference heading with a limited number of degrees.

Making the system into a state-space form we get

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w} \tag{0.2a}$$

$$y = \mathbf{C}\mathbf{x} + v \tag{0.2b}$$

with $\mathbf{x} = [\xi_w \ \psi_w \ \psi \ r \ b]^T$, $u = \delta$ and $\mathbf{w} = [w_w \ w_b]^T$. ψ is the average heading, i.e. without wave disturbance. ψ_w is a high-frequency component due to the wave disturbance, $\dot{\xi_w} = \psi_w$, r is the rotation velocity about the z-axis in the BODY coordinate system, and b is the bias to the rudder angle. T is the systems time constant and K is the gain constant. The model used can be stated as

$$\dot{\xi_w} = \psi_w \tag{0.3a}$$
$$\dot{\psi_w} = -\omega_0^2\xi_w - 2\lambda\omega_w\psi_w + K_w w_w \tag{0.3b}$$
$$\dot{\psi} = r \tag{0.3c}$$
$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - b) \tag{0.3d}$$
$$\dot{b} = w_b \tag{0.3e}$$
$$y = \psi + \psi_w + v \tag{0.3f}$$

1 Identification of the boat parameters

1.1 Transfer function from δ to ψ

To start out this assignment we want to find the transfer function from δ to ψ, parameterized by T and K. By assuming no disturbances ($b = 0$), the Laplace transformation of equation eq. (0.3c) and eq. (0.3d) gives the following

$$\mathscr{L}\{\ddot{\psi} = -\frac{1}{T}\dot{\psi} + \frac{K}{T}(\delta)\} \tag{1.1}$$

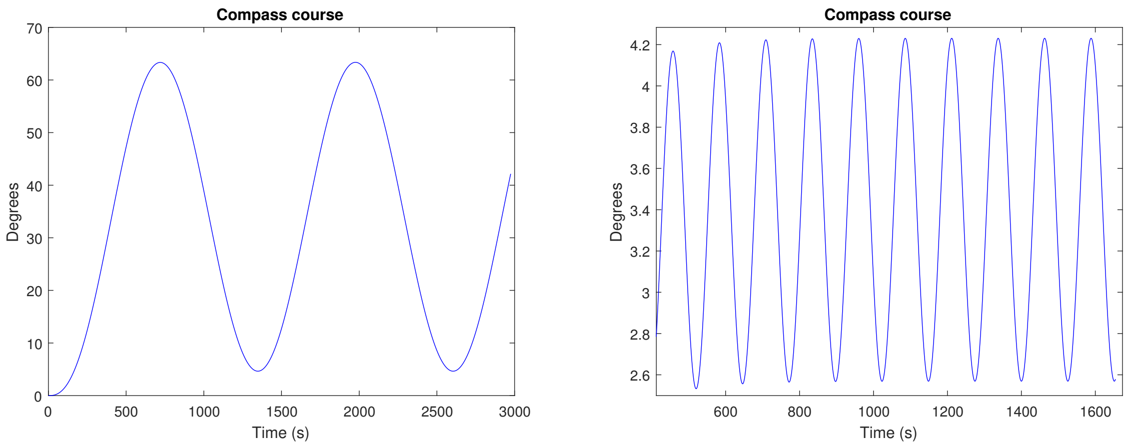
$$\Rightarrow s^2\psi(s) + s\psi(0) + \psi(0) + \frac{s}{T}\psi(s) + \psi(0) = \frac{K}{T}\delta$$

and with using the zero initial condition we find

$$H(s) = \frac{\psi(s)}{\delta(s)} = \frac{K}{s(Ts+1)} \tag{1.2}$$

1.2 Parameters in smooth weather conditions

We now want to identify the parameters T and K in smooth weather conditions, this is with all disturbances turned off. The input will be a sine function with amplitude 1 and frequency $\omega_1 = 0.005$ and $\omega_2 = 0.05$.



(a) ω_1 **(b)** ω_2

Figure 2: Output of the Compass with ω_1 and ω_2

After plotting the system, we calculated the average amplitude and got $A_1 = 29.354$ and $A_2 = 0.831$. This put us in a position where we could solve the following equation for K and T.

$$\begin{aligned} |H(j\omega)| &= \left|\frac{K}{j\omega(Tj\omega+1)}\right| \\ &= \frac{K}{\omega\sqrt{T^2\omega^2+1}} \\ &= A \end{aligned} \tag{1.3}$$

where $j = \sqrt{-1}$ for ω_1 and ω_2. From this we find by, isolating K, an expression for T

$$K = A_1\omega_1\sqrt{T^2\omega_1+1} = A_2\omega_2\sqrt{T^2\omega_2+1} \tag{1.4a}$$

$$T^2 = \frac{A_2^2\omega_2^2 - A_1^2\omega_1^2}{A_1^2\omega_1^4 - A_2^2\omega_2^4} \tag{1.4b}$$

By changing the constants for their numbers the boat parameters are $T \approx 72.4264$ and $K \approx 0.1561$.

1.3 Parameters with waves and measurement noise

In this problem we try to find the parameters K and T in rough weather condition. This means the wave and measurement noise are turned on. To solve this task we followed the same procedure as in the previous task.

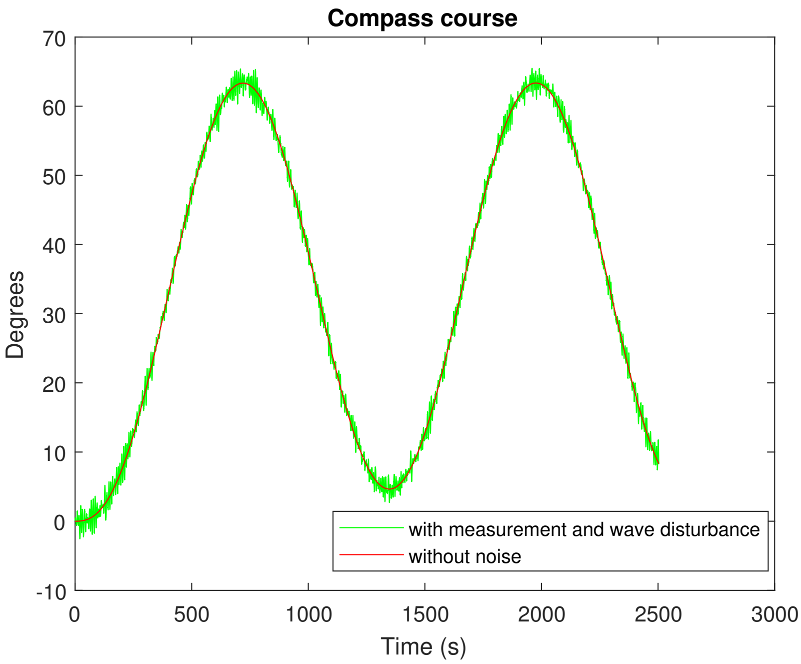


Figure 3: Compass comparison with ω_1 and added noise, $\omega_1 = 0.005$

For finding the average amplitude:

$$A_{1,1} = \frac{65.34 - 2.702}{2} = 31.344$$

$$A_{1,2} = \frac{65.47 - 2.495}{2} = 31.4875$$

The average of $A_{1,1}$ and $A_{1,2}$ then becomes 31.41575. With the noise added to the model it will not be possible to obtain equally good estimations as it was in the previous task. This because it is more challenging to get the right values for the amplitude.

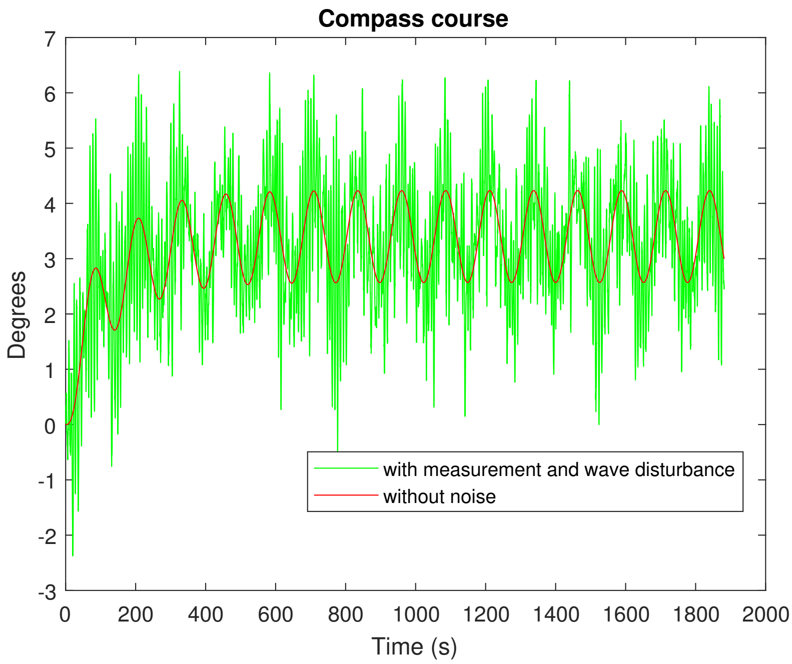


Figure 4: Compass comparison with ω_2 and added noise

For ω= 0.05 it is nearly impossible to approximate good parameters. Figure 4 shows how difficult it is to separate the noise from the desired signal. This will not make the model a good approximation of the ship. A solution for this is to run the signals through a low-pass filter before finding the values for the parameters.
After trying to find the average so that we can find a value for the amplitude we got

$$A_{1,1} = \frac{5.384 - 1.942}{2} = 1.721$$

$$A_{1,2} = \frac{6.36 - 1.451}{2} = 2.4545$$

$$A_{1,3} = \frac{6.321 - 0.2752}{2} = 3.0229$$

Giving the average for A_1 as 2.39946 which gave T_c = 290.67 and K = 0.1576. The new T varied quite a bit from section 1.2. So we can conclude that tuning the ship in rough weather conditions is not ideal.

1.4 Step response

For testing how good the approximation is, a step input of 1 degree to the rudder at t = 0 is implemented. See fig. 19 in appendix B for the simulink implementation.

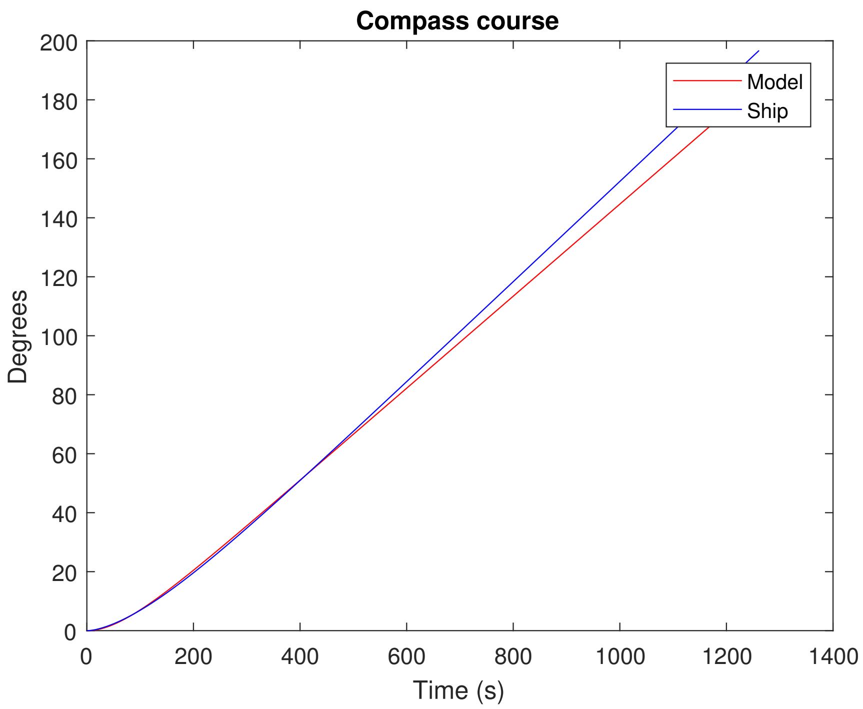


Figure 5: Step response of the model compared to the step response of the ship

The comparison tells us that the model are close to the ship in the fist 600 second. But after this the difference of the models increases. The bigger the deviations, will result in a non optimal regulator. The result is acceptable, because an approximation will never exactly be like the original system.

2 Identification of wave spectrum model

2.1 Power Spectral Density estimate

In this subsection we want to find the Power Spectral Density (PSD) of the wave,ψ_w, $S_{\psi_w}(\omega)$. To calculate the PSD the matlabfunction *pwelch* was used. This function returns the Welch's PSD estimate, pxx, and a frequency vector, [7].

The following version was used $[p_{xx}, f]$ =`pwelch`(x_2, window, noverlap, nfft, fs). The noverlap is used to overlap from segment to segment. The default of the noverlap is 50% of the window length. Nfft is for specifies the numbers of discrete Fourier transform to use in the estimation. In the Matlab code we made the input sequence to: $psi_w(2,:)$, this to get the data from wave.mat that influence the waves have on the compass measurement. This data was given in degrees, so to get it in radians it was multiplied with pi/180 since the fs is given in radians. The sample frequency was given as 10 Hz and the window was 4096. The code used can be seen in listing 3.

2.2 Analytic expression for transfer function and PSD

In this subsection the task was to find an analytic expression for the transfer function of the wave response model (from w_w to ψ_w). Ans also find an analytic expression for the Power Spectral Density function of ψ_w, that is $P_{\psi_w}(\omega)$.

Using the same techniques as earlier to find the transfer function from w_w to ψ_w using eq. (0.3a)

$$\xi_w(s) = \frac{\psi_w(s)}{s}$$

with $\xi_w(0) = 0$, inserting into eq. (0.3b) and using some algebra skills, we get

$$H(s) = \frac{\psi_w(s)}{w_w(s)} = \frac{sK_w}{s^2 + 2\lambda w_w s + \omega_0^2} \tag{2.1}$$

Then H(s) is a stable transfer function, and the PSD for ψ_w can be found through stationary analysis by using the formula under from [1]

$$\begin{aligned} P_{\psi_w}(\omega) &= |\hat{H}(j\omega)|^2 P_{\omega_w}(\omega) \\ &= H(j\omega)H(-j\omega)P_{\omega_w}(\omega) \end{aligned} \tag{2.2}$$

The amplitude of w_w is $P_w(j\omega)$, which is white noise, that is a constant and here is equal to 1.

$$\begin{aligned} H(j\omega)H(-j\omega) &= \frac{j\omega K_w}{(j\omega)^2 + 2\lambda w_0 j\omega + \omega_0^2} * \frac{-j\omega K_w}{(-j\omega)^2 - 2\lambda w_0 j\omega + \omega_0^2} \\ &= \frac{(\omega K_w)^2}{\omega^4 - 2\omega^2\omega_0^2 + 4\lambda^2\omega_0^2\omega^2 + \omega_0^4} \end{aligned} \tag{2.3}$$

Thus the PSD becomes

$$P_{\psi_w}(\omega) = \frac{(\omega K_w)^2}{\omega^4 - 2\omega^2\omega_0^2 + 4\lambda^2\omega_0^2\omega^2 + \omega_0^4} \tag{2.4}$$

2.3 Resonance frequency from estimated PSD

In this subsection the task was to find ω_0, from the estimated PSD found in Power Spectral Density estimate.

The ω_0 is the maximum of the power spectral density function, $S_{\psi_w}(\omega)$, and describes the frequency of waves that has the greatest impact on the head of the ship.

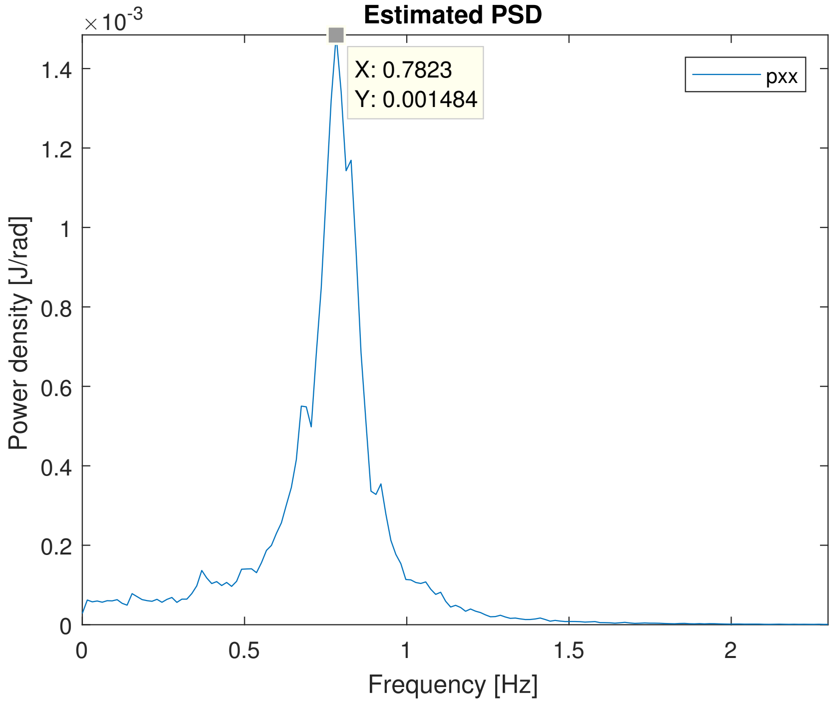


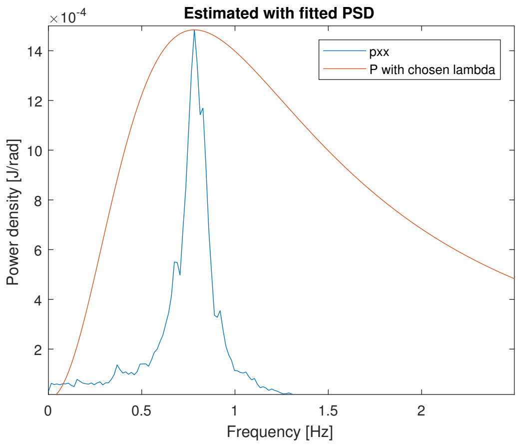
Figure 6: Reading off ω_0 from the estimated PSD found in 5.2.a

From the figure it was easy to determine that ω_0 was

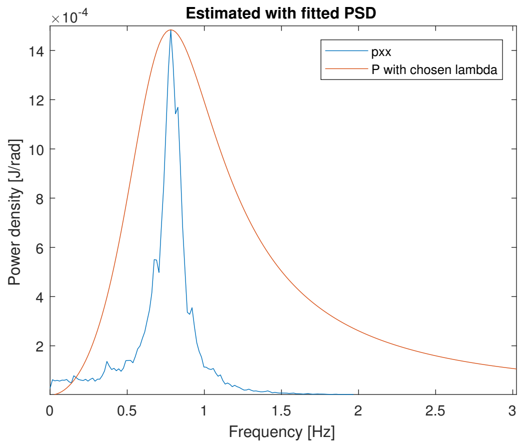
$$\omega_0 = 0.7823 \tag{2.5}$$

2.4 Damping factor λ

For completing the model of the wave response, the damping factor was to be determent, λ. In the task K_w was defined as $2\lambda\omega_0\sigma$ where σ^2 was the peak value of $P_{\psi_w}(\omega)$. To solve this task we decided to use the trial and error method. By trying different values of λ that fitted the $P_{\psi_w}(\omega)$ to the estimate of the PSD, we have whats show in fig. 7 - 8. The final pick for λ was 0.07. In listing 3 it is shown how the PSD function was plotted.

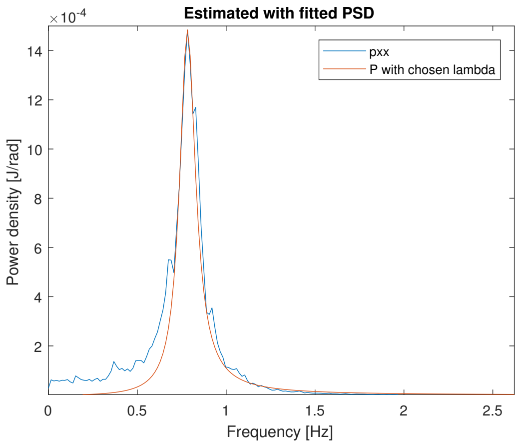


(a) $\lambda = 1$

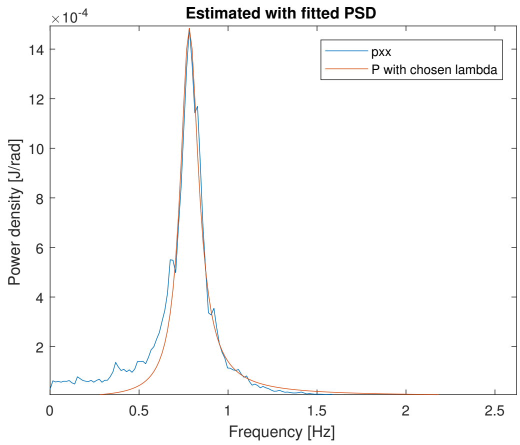


(b) $\lambda = 0.5$

Figure 7: To large values for λ



(a) $\lambda = 0.07$



(b) $\Lambda = 0.08$

Figure 8: The best found values for λ

3 Control System Design

In this section we want to design an autopilot for the ship. Which means we want to be able to choose an angle ψ_r and the ship will follow this course. The model of the boat holds only for small deviations for the compass value. This means the compass value cannot be more than $\pm 35°$. We used $\psi_r = 30$ in all the following simulations. We chose not to put a saturation on the output of the compass degree signal, this because we realized that the model always was within a boundary of ± 35 degrees.

3.1 PD controller

In this subsection we start by designing a PD-controller in the form,

$$H_{pd}(s) = K_{pd}\frac{1+T_d s}{1+T_f s} \tag{3.1}$$

we base (3.1) on the transferfunction from δ to ψ and assume that the disturbances are negligible. We let ω_c and the phase margin, φ, of the open loop system, $H_{pd}(s) \cdot H_{ship}(s)$, be approximately 0.10 (rad/s) and 50 degrees respectively. The systems transferfunction then becomes

$$\begin{aligned} H_0(s) &= H_{pd}(s) \cdot H_{ship}(s) \\ &= K_{pd}\frac{K + KT_d s}{s^3 T T_f + s^2(T+T_f) + s} \end{aligned} \tag{3.2}$$

We want T_d to cancel out the transfer function time constant

$$\begin{aligned} 1 + T_d s &= 1 + Ts \\ T_d &= T \end{aligned}$$

The transferfunction becomes,

$$H_0(s) = K_{pd}\frac{K}{(1+T_f s)s} \tag{3.3}$$

To find the coefficient T_f and K_{pd}, the cut-off frequency was given $\omega_c = 0.10$ (rad/s), we solved the equation

$$\begin{aligned} 1 &= |H_0(j\omega_c)| \\ \varphi &= 180 - \angle H_0(j\omega_c) \end{aligned} \tag{3.4}$$

giving the expression for T_f as

$$T_f = \frac{1}{\omega_c \tan(\varphi * pi/180)} = 8.39099 \tag{3.5}$$

The value for K_{pd} is easily calculated from

$$K_{pd} = \frac{\sqrt{\omega_c^2 + T_f^2\omega_c^4}}{K} = 0.836263 \tag{3.6}$$

The open loop system can be show in a bode diagram, see fig. 9, with the phase margin approximately 50 degrees. T_f affects the magnitude and phase of the system, while the K_{pd} only affects the magnitude. This means that a lower T_f make the response slower, while a bigger T_f would make the response oscillate, maybe giving an overshoot. This is because T_f limits the derivative effect.

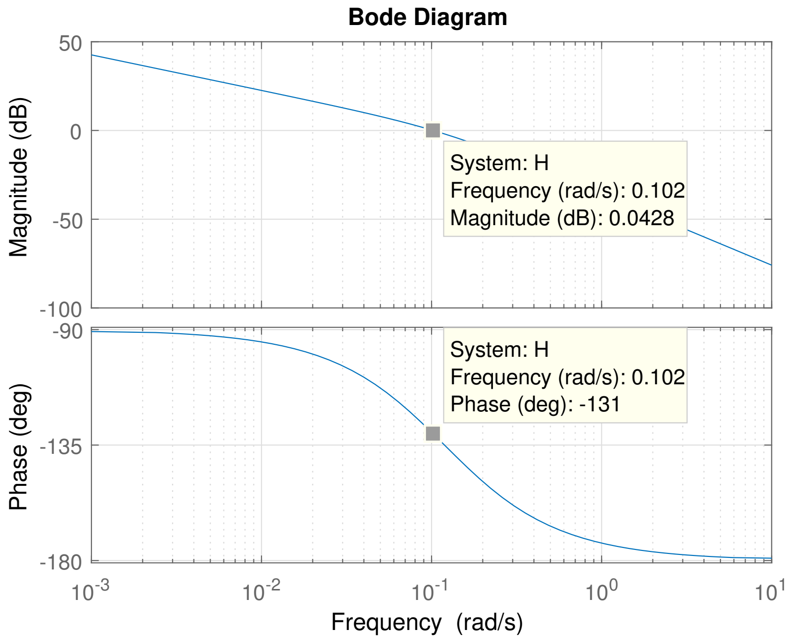


Figure 9: Bode diagram with $T_f = 8.391$ and $K_{pd} = 0.8363$

3.2 Simulation with measurement noise

Simulating the system without disturbances, except for measurement noise, we see that the autopilot manages to keep the reference, at 30°. For the Simulink- diagrams, see appendix B.3

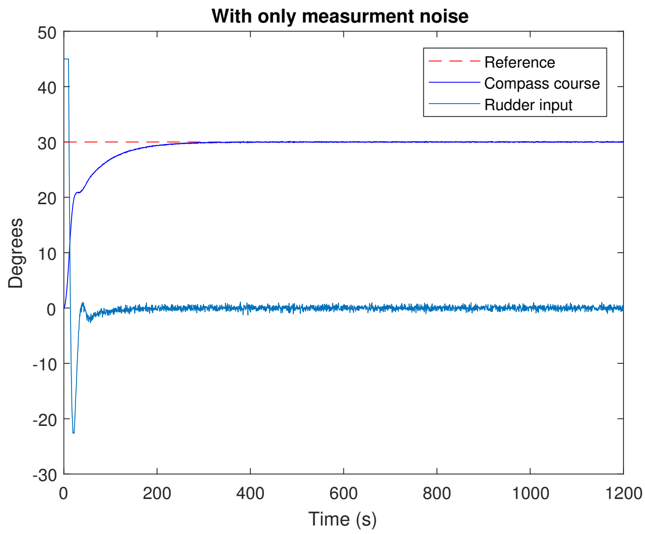


Figure 10: Rudder and compass output of the system with measurement noise

In figure 10 we can observe that the compass course settles to 30 degrees pretty fast. The same goes for the rudder input, it holds approximately 0 degrees. In the beginning of the plot we can observe that the rudder input goes in saturation. The autopilot work in the desired way in smooth weather conditions. This was also expected because the system is an ideal model and behave like the mathematical model.

The ship use 300 seconds to achieve 30 degrees, if the system should be faster then the K_{pd} must be increased. But if this is done the phase margin will decrease, this can make the system less stable. Notice that the rudder changes direction even though the system has reached its reference. This is because the course changes so fast and would end in an overshoot, so the derivative effect makes the system calm down by turning slower.

3.3 Simulation with current disturbance

The system simulated with a current disturbance, but no wave disturbances is shown in fig. 11

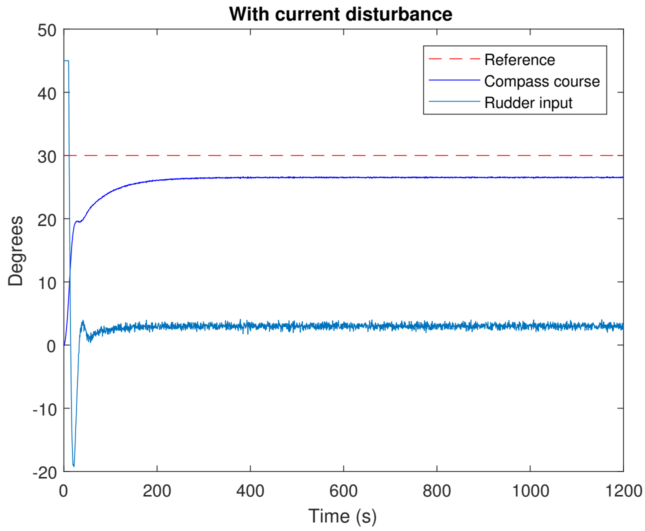


Figure 11: Rudder and compass output of the system with measurement noise and current disturbance

In the figure 11, there is a standard deviation equal 3.5 degrees from the reference on the compass course, and the rudder input is approximated 2.5 degrees. This is because the current is model as a constant disturbance. The error in the compass course is not desirable for an autopilot. The standard derivation in the compass is because the system has en PD regulator. For an better autopilot, we would make an PID to integrate up the error and reach the desired heading. But then we would have an other problem. Since the phase of the PID would bean at 180○ and the system is not longer stable. An alternatives could be having another differentiation, but we don't see this as a optimal solution.

3.4 Simulation with wave disturbance

Simulating the system with a wave disturbance, but no current disturbances, we have a lot of noise, making it harder for the autopilot. The rudder tries to counter the effect each wave has on the ship. which means if will end up rapidly changing back and forth for each wave, unnecessarily compensating for the wave noise.

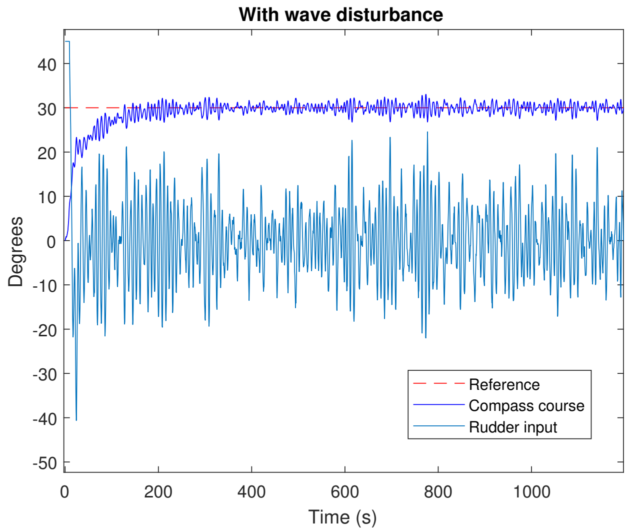


Figure 12: Rudder and compass output of the system with measurement noise and wave disturbance

The wave disturbance is high frequency so it make the compass course oscillating round the reference point. In average the compass signal and rudder input, does not vary to much of the reference, so the ship would come to the desired destination. But the signals of the rudder input is not so precise in regards to the mechanical in the motor. To solve this we would recommend to make state estimators for the internal states.

4 Observability

Here we look at the matrices of the system to check the observability in different scenarios, such as with and without different disturbances. The matrix's was calculated in Matlab (4).

4.1 State space model matrices A, B, C and E

Using the system in eq. (0.2) with $\mathbf{x}$, u and $\mathbf{w}$ as given, we get

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{4.2}$$

4.2 Without disturbances

For section 4.1 - section 4.5, the observability matrix is used to study observability. It is defined as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA^2} \\ \vdots \\ \boldsymbol{CA^{n-2}} \\ \boldsymbol{CA^{n-1}} \end{bmatrix}$$

where n is the number of state variables. In listing 4, is the code used to calculate the observer matrix observability using `obsv(A,C)` and `rank(O)`. Examining the observability without disturbances, we see that all states affected by the disturbances disappear, and the remaining are ψ and r. This gives the states space matrices as

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.3}$$

which gives the observability matrix as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.4}$$

From that we can see that the rank is equal to 2, and thus the system without disturbances is observable.

4.3 Current disturbance

Examining the observability only counting the disturbances from the current, we see that all states affected by the disturbances disappear and we are left with ψ, b and r giving

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0138 & -0.0022 \\ 0 & 0 & 0 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ 0.0022 \\ 0 \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.6}$$

The observability matrix the becomes

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.0138 & -0.0022 \end{bmatrix} \tag{4.7}$$

which has a rank equal to 3, meaning it is observable with full rank.

4.4 Wave disturbance

Examining the observability only counting the disturbances from the waves, we see that all states affected by the disturbances disappear and we are left with ψ, ψ_w, ξ_w and r giving

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.612 & -0.1095 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.0138 \end{bmatrix}, \; \boldsymbol{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \tag{4.8}$$

with observability matrix as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -0.612 & -0.1095 & 0 & 1 \\ 0.067 & -0.6 & 0 & -0.0138 \\ 0.3672 & 0.1327 & 0 & 0.0002 \end{bmatrix} \tag{4.9}$$

which from we can easily observe that the rank is equal to 4, making the system observable with full rank.

4.5 Wave and current disturbances

Using both disturbances and the state space matrices in eq. (4.1) and eq. (4.2) we have the observability matrix as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -0.612 & -0.1095 & 0 & 1 & 0 \\ 0.067 & -0.6 & 0 & -0.0138 & -0.0022 \\ 0.3672 & 0.1327 & 0 & 0.0002 & 0 \\ 0.0812 & 0.3527 & 0 & 0 & 0 \end{bmatrix} \tag{4.10}$$

and the rank equal to 5, making the system observable with both wave and current disturbances. Having all there observability matrices with full rank means that its is possible to estimate the state from the output for each of them. This in turn means that making a Kalman filter is possible for all weather conditions.

5 Discrete Kalman Filter

In this part a discrete Kalman filter is implemented to estimate the bias b and the heading ψ. The high-frequented wave induced motion ψ_m is also estimated, but this must be removed from the control loop to avoid wear and tear on the actuator system. We continue to be careful not to make ψ bigger than $\pm 35\circ$ and small deviations in compass value. Since the rudder angle is constrained to $\pm 45\circ$ an saturation block was placed on the input of the rudder signal. (See 13).

Kalman filtering applies a recursive method for estimation of a random process. The optimization criterion used is minimization of the mean-squared estimation error of the random variable x.

5.1 Exact discretization

To use a discrete Kalman filter, a discretized system is needed. The matrices A, B and E found in 4.1 was put into Matlab and the function `c2d` with sampling frequency of 10 Hz was used to get an exact discretization of the system A_d, B_d and E_d (see code below). The `c2d` function uses zero-order hold to get a discrete counterpart to the continous system, and had to be used twice as the function only discretize two matrices at the time. The matrices C_d and D_d are the same as C and D as shown in page 110 in [2].

```
Fs = 10; %Hz
Ts = 1/Fs;

% discretization
[Ad, Bd] = c2d(A,B,Ts);
[Ad, Ed] = c2d(A,E,Ts);
```

5.2 Estimate of measurement noise variance

To find the estimate of the variance of the measurement noise we used the Matlab function `var` on the measured compass course. This data was imported from Matlab and transformed from degrees to radians before taking the mean variance. The variance was found to be $= 6.0813{*}10^{-7}$.

```
% variance of measurement noise
load ('data.mat');
compass_data = data(2,:); %compass(deg)
mes_var = var(compass_data*pi/180); % Measurment noice in radians
```

5.3 Implementation of discrete Kalman filter

To implement the discrete Kalman filter we chose to use a Matlab function block in Simulink. To initialize the filter the following was given from the assignment

$$\boldsymbol{w} = [w_w \; w_b]^T \; , \; E\{\boldsymbol{w}\boldsymbol{w^T}\} = \boldsymbol{Q} = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix} \tag{5.1}$$

$$\boldsymbol{P_0^-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-3} \end{bmatrix} \; , \; \boldsymbol{x_0^-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.2}$$

$\mathbf{w}$ is the process noise, $\mathbf{Q}$ is the process noise covariance, $\hat{\boldsymbol{x}}_{\mathbf{0}}^{-}$ is the initial a priori state estimate and $\boldsymbol{P}_{\mathbf{0}}^{-}$ is the initial a priori estimate error covariance. $E(v^2) = R = \frac{variance}{T_s}$ where the variance was found in the previous section and T_s was $\frac{1}{F_s}$. This is because the process is sampled.

Now with the initial phase defined the equations from [2] was used to create the Kalman filter. The first step of the Kalman filter is to calculate the new Kalman gain, K.

$$K = \boldsymbol{P}^{-}\boldsymbol{C}_d^T(\boldsymbol{C}_d\boldsymbol{P}^{-}\boldsymbol{C}_d^T + R)^{-1}; \tag{5.3}$$

With this new gain it is now possible to move to the second step which is to update to our new state $\hat{\boldsymbol{x}}$ and our new estimate error covariance $\boldsymbol{P}$, also known as $\hat{\boldsymbol{x}}$ and $\boldsymbol{P}$ a posteriori. The y used here is the measured compass course.

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}^{-} + K(\boldsymbol{y} - \boldsymbol{C}_d\hat{\boldsymbol{x}}^{-}) \tag{5.4}$$

$$\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C}_d)\boldsymbol{P}^{-}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C}_d)^T + KRK^T; \tag{5.5}$$

The third step is to project ahead and create what will become the new $\boldsymbol{x}$ and $\boldsymbol{P}$ a priori.

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \boldsymbol{A_d}\hat{\boldsymbol{x}} + \boldsymbol{B_d}u \tag{5.6}$$

$$\hat{\boldsymbol{P}}_{k+1}^{-} = \boldsymbol{A_d}\boldsymbol{P}\boldsymbol{A_d}^T + \boldsymbol{EdQ}\boldsymbol{E_d}^T; \tag{5.7}$$

All of these functions plus the initialization was implemented in Matlab as seen 6. The new estimated compass course ψ, here called y_est, and the bias b was updated at the end to be used in the controll loop.

The Simulink implementation of this part can be seen in 22.Here a *Zero Order Hold* block was used to make the signal discrete. This was necessary to do since the design was a discrete Kalman filter.
The Kalman filter depends on the control input, but the input of the Kalman filter depends on the output for the Kalman filter. This does that Simulink is not able to determine the initial values and blame on an *algebraic loop*. This is solved by adding a *Memory* block on the output of the Kalman filter.

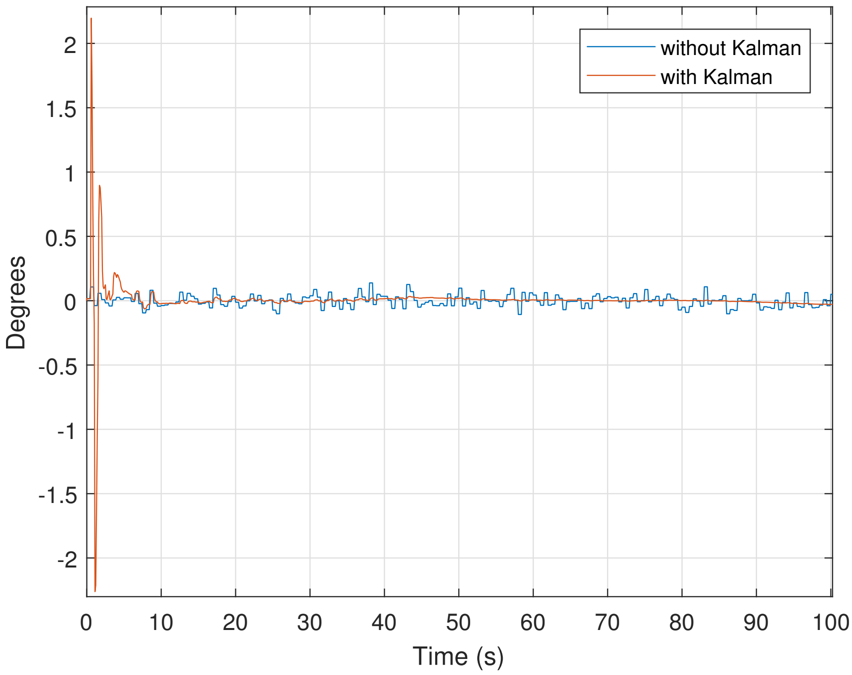


Figure 13: Signal with and without Kalman filtering

As seen in the fig. 13 the signal that is filtered form the Kalman filter use some seconds to adjust to the correct values. This is not a big concern when the deviation is approximately 2 degrees, and over some seconds.

5.4 Simulation with current disturbance with feed forward

For testing the Kalman filter and the performance of the autopilot a simulation with $\psi_r = 30$ was used. The figure 14 shows the values in the simulation.

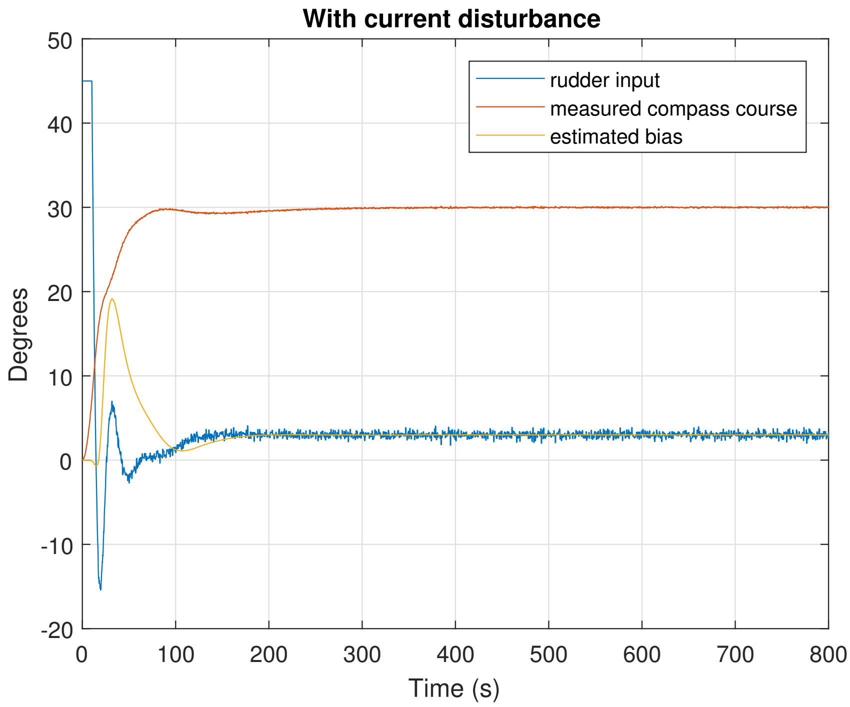


Figure 14: Autopilot with Kalman filtering and current disturbance

The biggest difference is that the compass course contains 30 degrees. This is because the rudder bias that is estimates is accurate enough to cancel out the real rudder bias. Hence, we have a better performance of the autopilot then in section 3.3.

5.5 Simulation with wave and current disturbance and wave filtering

In this subsection we are simulatin with both wave and current disturbance. Instead of the measured compass couse we use the wave filtered ψ signal.

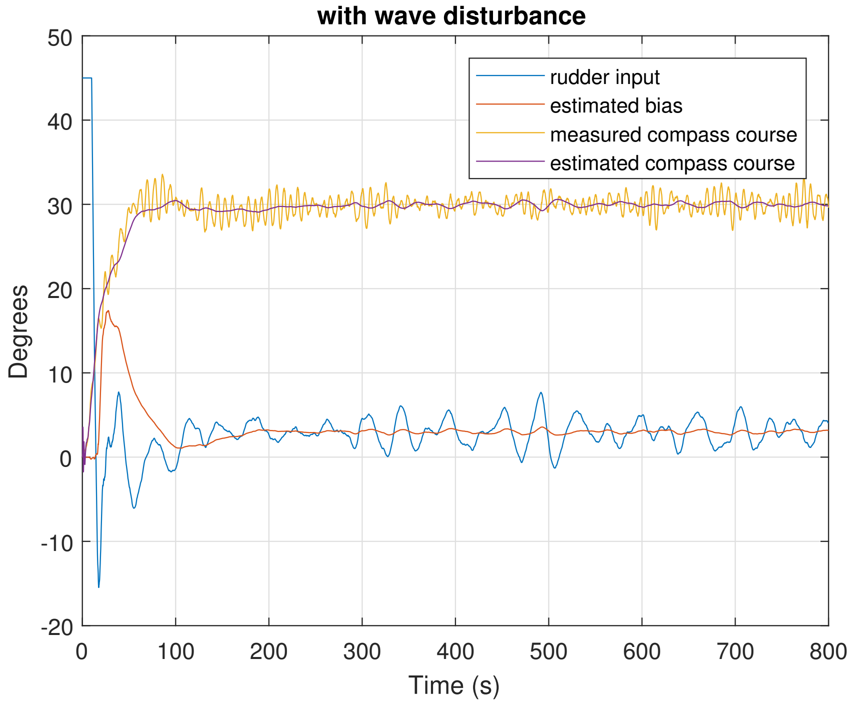


Figure 15: Autopilot with Kalman filtering with wave and current disturbance

In figure 15 is the simulation with wave and current disturbance. A big difference from the autopilot in section 3.4 is that the rudder input does not fluctuate as much. This is a desirable because the system would must likely be less damaged by the oscillation from the rudder signal. The estimated compass course is the filtered compass course from the Kalman filter, we see here that the signal is better that without the Kalman filter.
An alternative could we using a low-pass filter on the compass measurement. This would probably removed some of the ψ_ω, but not the wave disturbance. The conclusion is that with the Kalman filter we get a more robust and reliable system.

The last figure (16 represent the comparison of the actual wave influence and estimated wave influence on the ship. As the figure shows the estimation is good, even sometimes the estimation wave is to small.But we are happy with the result.

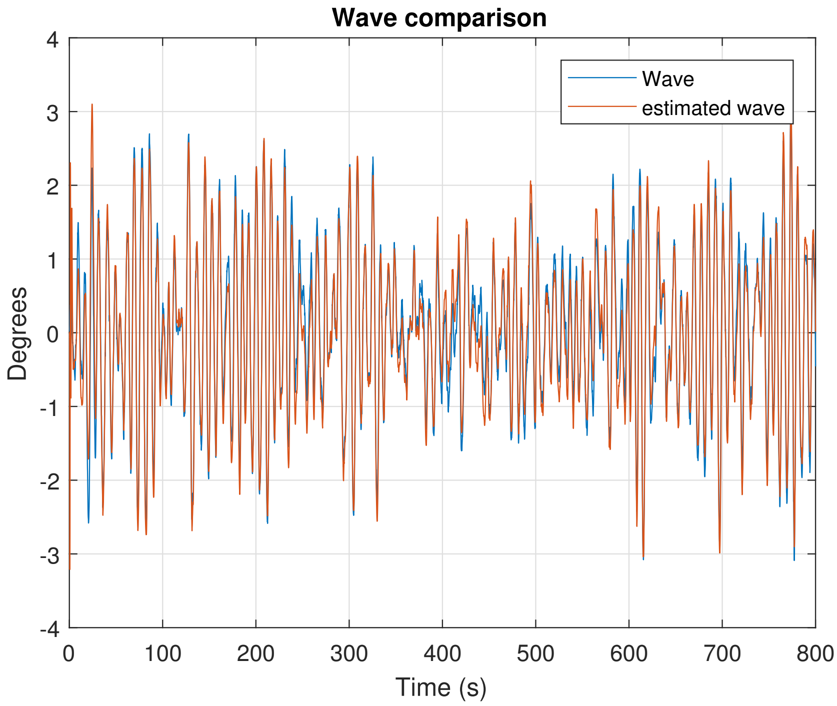


Figure 16: Comparison

References

[1] Robert Brown, *Introduction to random signals and applied Kalman filtering : with MATLAB exercises*, John Wiley & Sons Inc., Hoboken, NJ, 4th edition, 2012

[2] Chi-Tsong Chen, *Linear System Theory and Design*, Oxford University Press, 4th edition, 2014

[3] Kristoffer Gryte et al., *Boat lab assignment*, Department of Engineering Cybernetics, NTNU, Version 1.8, 2016

[4] Pedersen M.D, *The Kalman Filter*, Lecture presented at; 2017; Norwegian University of Science and Technology

[5] How to find phase angle from transfer function, *https://electronics.stackexchange.com/questions/269008/how-to-find-phase-angle-from-transfer-function*, 03.11.2017

[6] Ipe extensible drawing editor, downloaded at: *https://sourceforge.net/projects/ipe7/* for making figures, 10.11.2017

[7] *https://se.mathworks.com/help/signal/ref/pwelch.html* for information about the pwelch function in Matlab.

A MATLAB Code

A.1 Part 1

Listing 1: Matlab code showing how the tack in part 1 was solved

```
clc;
clear all;
close all;

% Initialization
om1 = 0.005;
om2 = 0.05;

% b) calculating paramteres K and T

% Observed amplitudes for omega_1 and omega_2:
A1 = 29.354;     A2 = 0.831;

%Calculating T and K with basic algebra
T_sqr = (A2^2*om2^2 - A1^2*om1^2) / (A1^2*om1^4 - A2^2*om2^4);
T = sqrt(T_sqr);

K = A1*om1*sqrt(T^2*om1^2 + 1);

% c) parameters with noise

% Average amplitudes
A1_C = 31.41575;     A2_C = 2.39946;

% New values for T and K
T_sqr_C = (A2_C^2*om2^2 - A1_C^2*om1^2) / (A1_C^2*om1^4 - A2_C^2*om2
    ^4);
T_C = sqrt(T_sqr_C);

K_C = A1_C*om1*sqrt(T_C^2*om1^2 + 1);
```

A.2 Part 2

Listing 2: Matlab code showing how the PSD function was calculated

```
% Part 2
load('wave.mat');
Fs = 10;
window_size = 4096;

[pxx,f] = pwelch(psi_w(2,:)*(pi/180), window_size,[],[],Fs);
omega= 2*pi*f; %rad/s]
pxx= pxx./(2*pi); %[s/rad]

figure
```

```
plot(omega, pxx); hold on;
%w_0 was identifyd by the plot
w_0 = 0.7823;

%Calculating sigma, sigma^2 was the peak value og the plot
sigma= sqrt(0.001484);

%Finding lamda
lambda = 0.08;
Kw = 2*lambda*w_0*sigma;

ss = (Kw^2*omega.^2)./(omega.^4+w_0^4 +2*omega.^2*w_0^2*(-1+2*lambda^2));
plot(omega,ss);
xlabel('Frequency [Hz]'); ylabel('Power density [J/rad]');
title('Estimated with fitted PSD'); legend('pxx','P with chosen lambda');
```

A.3 Part 3

Listing 3: Matlab code showing the code for part 3

```
%Part 3
K_pd = 0.8633;
T_f = 8.3909;
T_d = T;

%Bode
s = tf('s') ;
H = K_pd*K/ (T_f* s ^2 + s ) ;
```

A.4 Part 4

Listing 4: Matlab code showing the calculation for showing the observability

```
%Part 4
        A = [0 1 0 0 0; -w_0^2 -2*lambda*w_0 0 0 0 ;
                 0 0 0 1 0; 0 0 0 -1/T -K/T; 0 0 0 0 0;];
        B = [0 0 0 K/T 0]';
        E = [0 0; K_w 0; 0 0; 0 0; 0 1];
        C = [ 0 1 1 0 0];

% No distubance
        A_b = [0 1; 0 -1/T];
        B_b = [0 K/T]';
        C_b = [1 0];
        obs_b = obsv(A_b, C_b);
        rank(obs_b);

%With current
        A_c = [0 1 0; 0 -1/T -K/T; 0 0 0;];
        B_c = [0 K/T 0]';
```

```
            E_c = [ 0 0; 0 0; 0 1];
            C_c = [1 0 0];
            obs_c = obsv(A_c, C_c);
            rank(obs_c);

%With wave disturbance
            A_d =  [0 1 0 0; -w_0^2 -2*lambda*w_0 0 0 ;
                    0 0 0 1; 0 0 0 -1/T;];
            c_d = [0 1 1 0];
            obs_d = obsv(A_d, c_d);
            rank(obs_d);

%With current and wave disturbance
            obs_d = obsv(A,C);
            rank(obs_d);
```

A.5 Part 5

Listing 5: Matlab code showing the Kalman function

```
function [phi,b] = kalman(r,cm, ks)
persistent init x_pri P_pri y_pri Ad Bd C Ed Q I R

if isempty(init)
    %Initializing constants
    Ad=ks.Ad;    Bd=ks.Bd;    C=ks.C; Ed=ks.Ed;
    I=ks.I; R=ks.R; Q=ks.Q; y_pri=ks.y_pri;
    % Initializing first step
    P_pri = ks.P0_pri;
    x_pri = ks.x0_pri;
    % Making sure initializing only once
    init = 1;
end
% Updating the Kalman filter
K_post = P_pri*C'*inv(C*P_pri*C'+R);
x_post = x_pri+K_post*(cm-y_pri);
P_post = (I-K_post*C)*P_pri*(I-K_post*C)'+K_post*R*K_post';

%Project ahead
P_pri = Ad*P_post*Ad'+Ed*Q*Ed';
x_pri = Ad*x_post+Bd*r;
y_pri = C*x_post;

% Setting the output
phi = x_post(3);       b = x_post(5);
end
```

Listing 6: Matlab code showing implantation for Kalman filter

```
%Part 5
% c)
```

```
R = mes_var/Ts;
P0_pri = [1 0 0 0 0; 0 0.013 0 0 0; 0 0 pi^2 0 0;
      0 0 0 1 0; 0 0 0 0 2.5e-3];
x0_pri = [0 0 0 0 0]';
Q = [30 0; 0 1e-6];
I = eye(5);
ks = struct('Ad',Ad,'Bd',Bd,'C',C,'Ed',Ed,'R',R,'Q',Q,'I',I,'P0_pri',
    P0_pri,'x0_pri',x0_pri);
```

B Simulink Diagram

B.1 Part 1

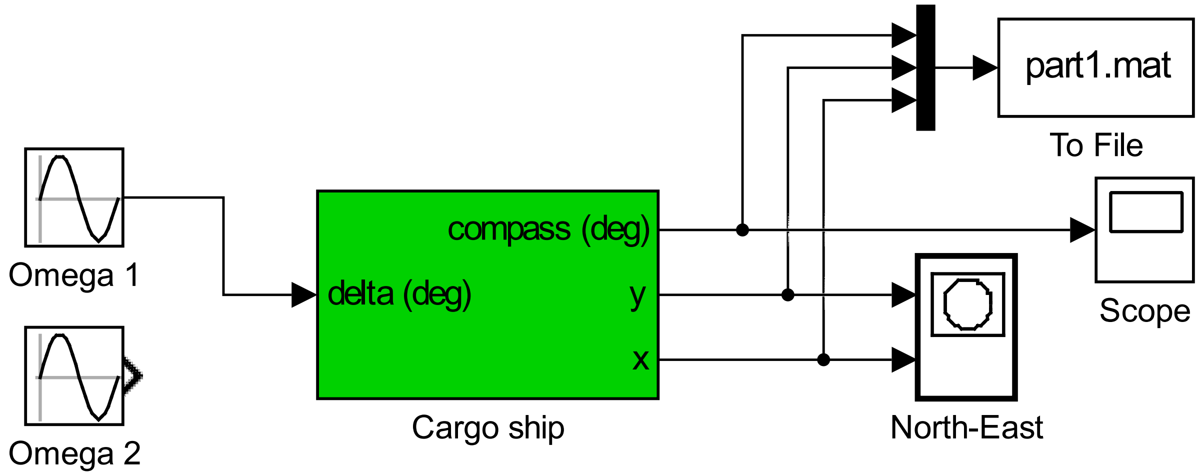


Figure 17: System for part 1 b)

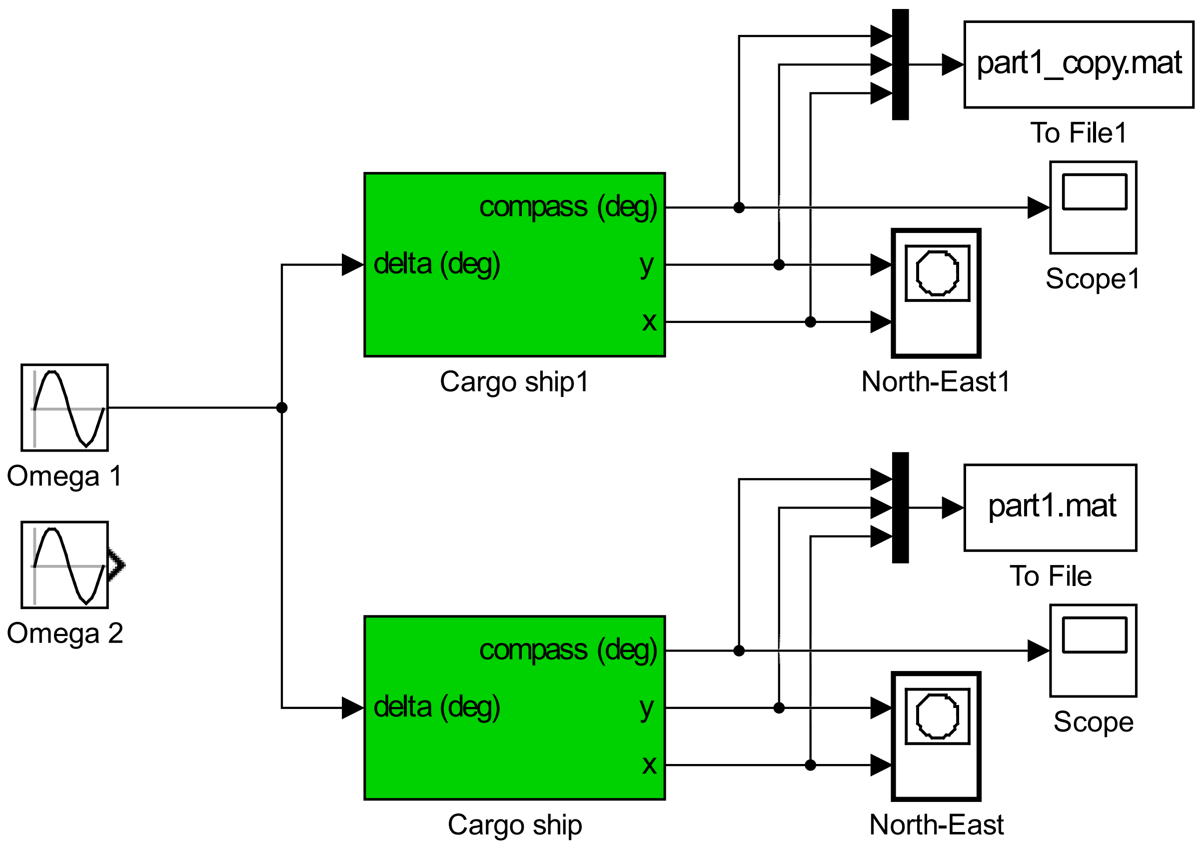


Figure 18: System for part 1 c)

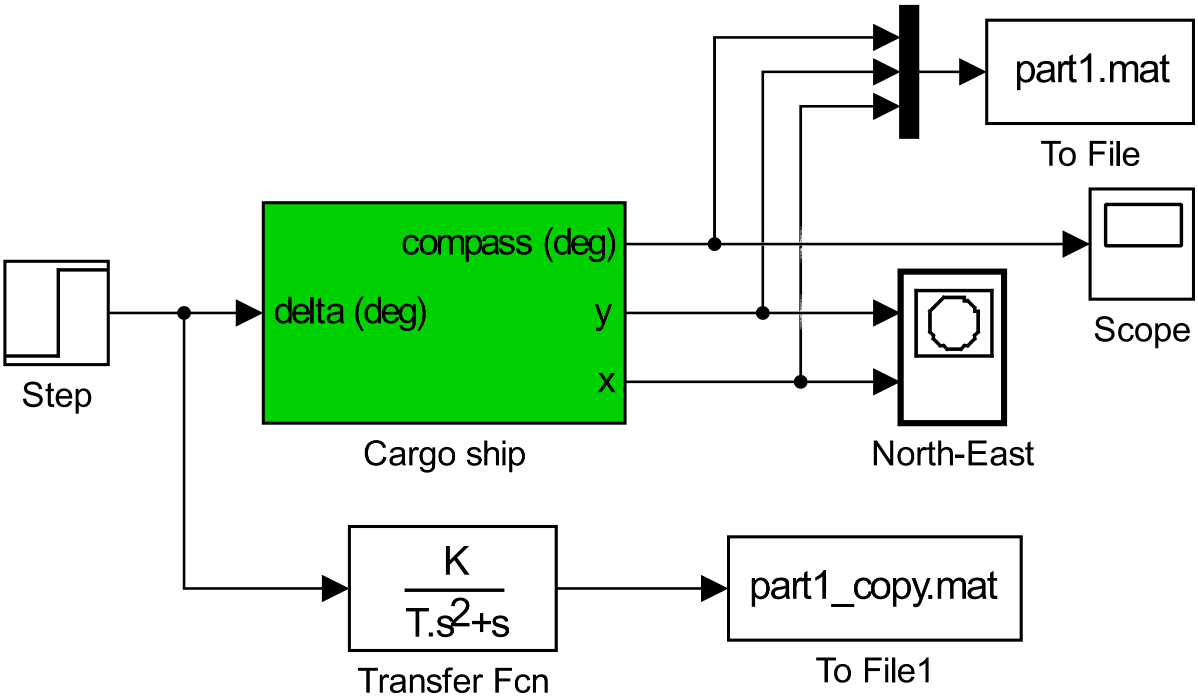


Figure 19: System for part 1 d)

B.2 Part 2

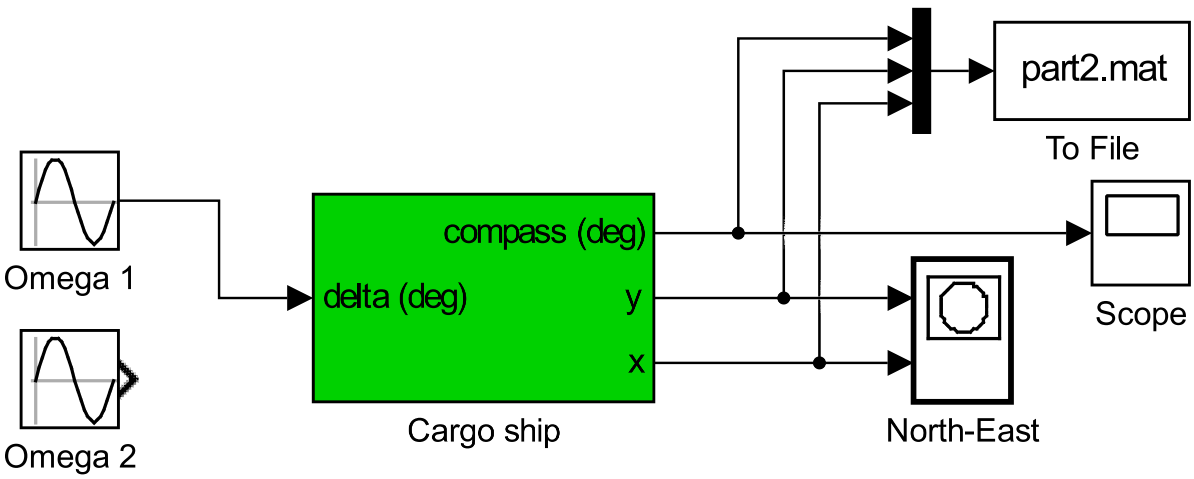


Figure 20: System for part 2

B.3 Part 3

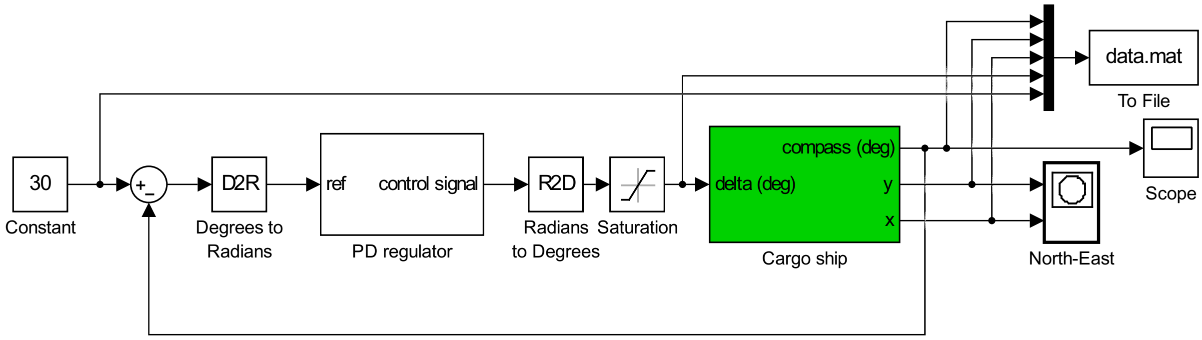


Figure 21: System for part 3

B.4 Part 5

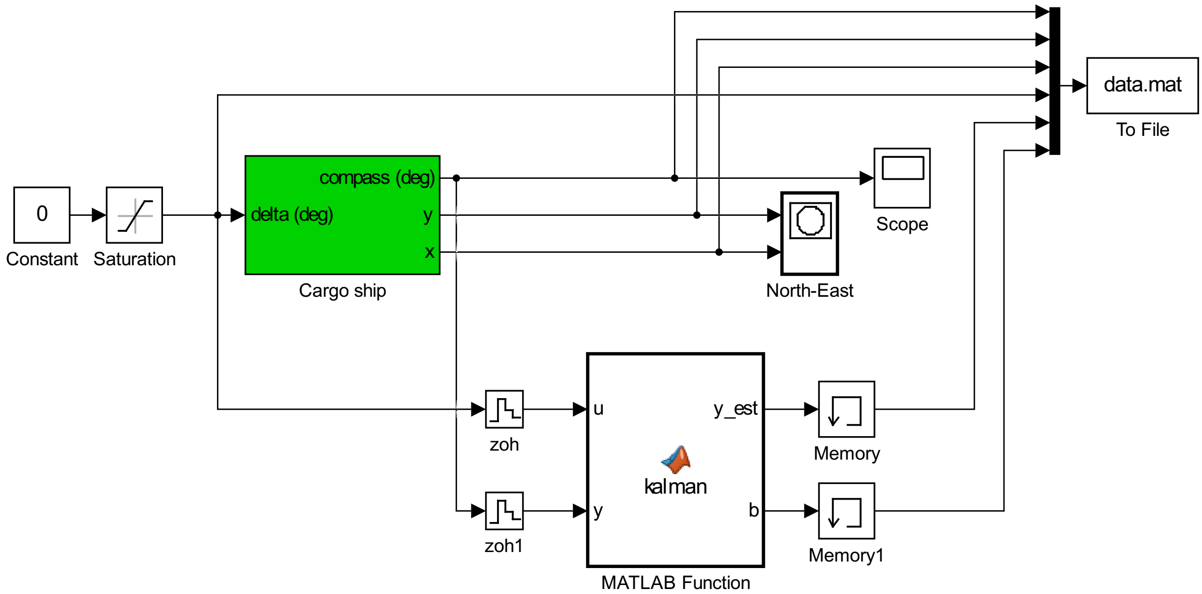


Figure 22: System for part 5 c)

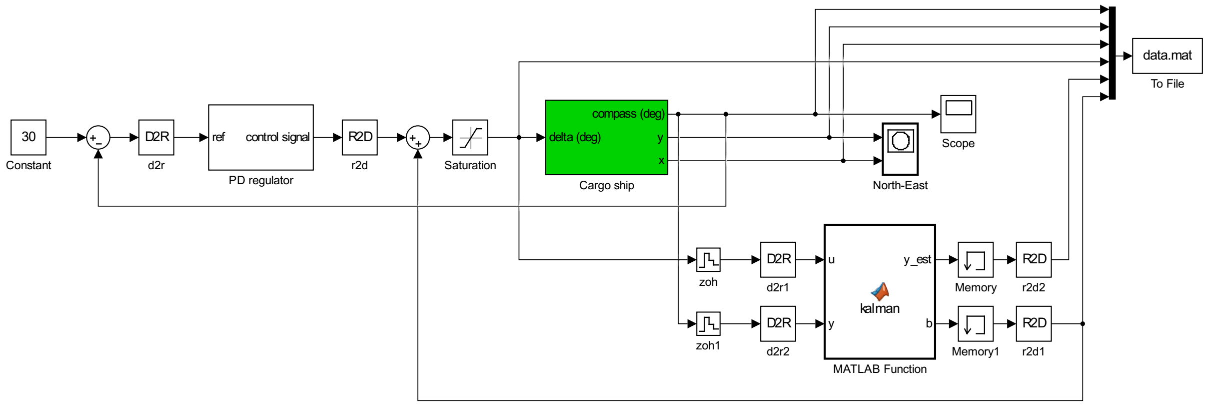


Figure 23: System for part 5 d)

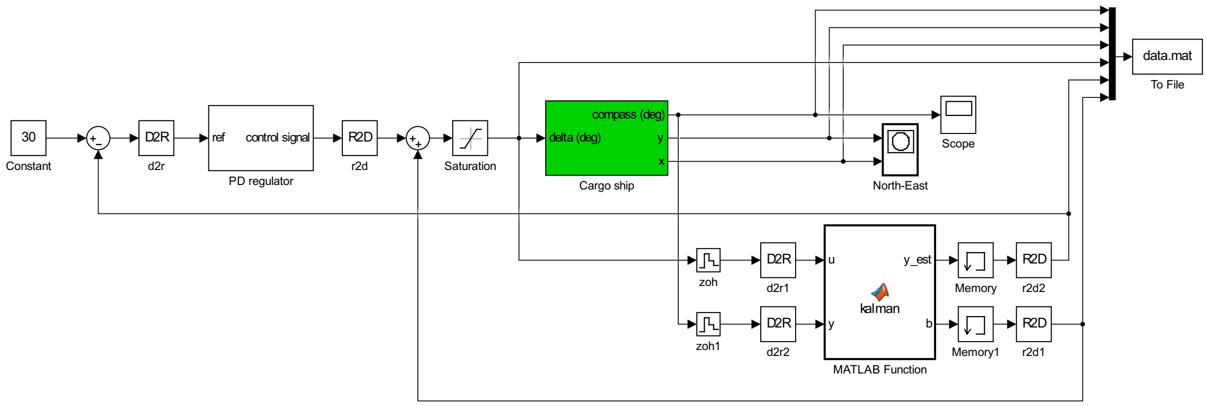


Figure 24: System for part 5 e)

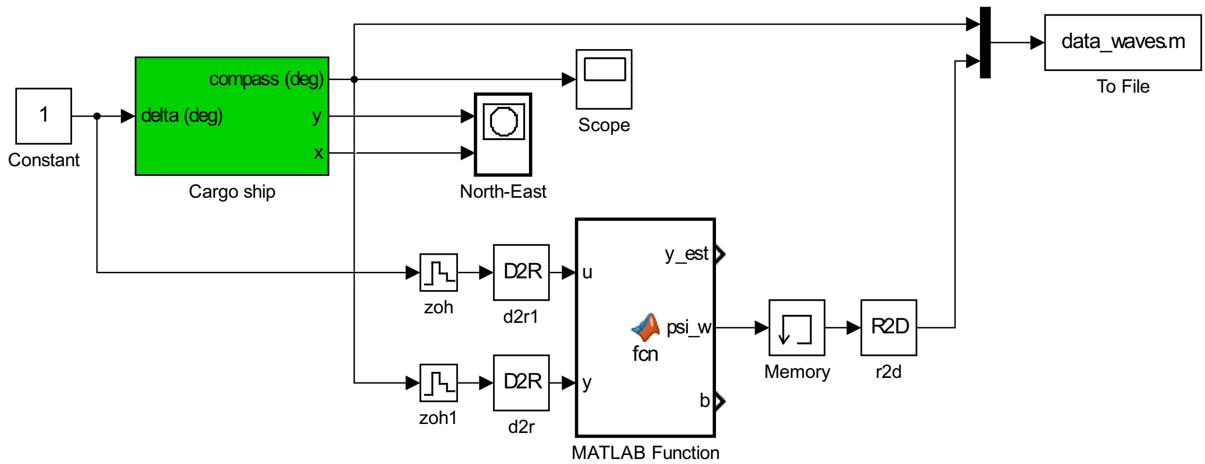


Figure 25: System for part 5 e, finding wave influence)