Boat Lab Report

Group 27
Student 749226
Student 768155
Student 768150

November, 2018

Department of Engineering Cybernetics

Contents

1 Problem 5.1: Boat Parameters

1.1 Task a) Transfer Function from δ to ψ

1.1.1 Method

Assuming there are no disturbances in the system. To find a transfer function from the rudder angle relative to the BODY frame, δ, to the heading ψ of the ship, one must look at the model of the ship as seen in eq. (1).

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

From eq. (1c) one can see that r can be written as a function of ψ, and by differentiating with respect to time, one gets eq. (2).

$$\dot{r} = \ddot{\psi} \tag{2}$$

By inserting eq. (2) into eq. (1d), one gets an expression of ψ and δ. Since there is no disturbance, the bias to the rudder angle, b, becomes zero. Laplace transforming the resulting equation and solving for $\frac{\psi}{\delta}$, yields:

$$\frac{\psi}{\delta} = \frac{K}{Ts^2 + s} \tag{3}$$

1.1.2 Results

The transfer function from δ to ψ is given by eq. (4).

$$\frac{\psi}{\delta} = \frac{K}{Ts^2 + s} \tag{4}$$

1.2 Task b) Boat Parameters in Smooth Weather Conditions

1.2.1 Method

To find the boat parameters T and K in smooth weather conditions, one starts by turning off all disturbances in the model. To determine the boat parameters, the boat is simulated with two different input signals sent in as step functions in Simulink, see fig. 1.

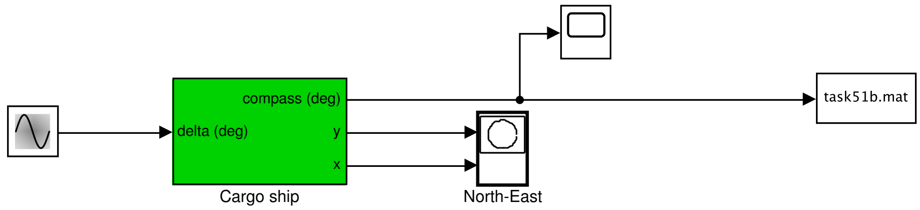

Figure 1: Simulink model for sine function as input.

The first input is a sine function with amplitude 1 and frequency $\omega_1 = 0.005$ [rad s^{-1}]. The second input is a sine function with the same amplitude, but with frequency $\omega_2 = 0.05$ [rad s^{-1}]. The amplitude of the output (here, heading ψ) from the two different signals, which equals $|H(j\omega)|$ for the corresponding frequency, is read from a plot of the heading. With both amplitudes from the two frequencies known, yields the two expressions for the boat parameters T and K given by eq. (5).

$$|H(j\omega_1)| = \left|\frac{K}{T(j\omega_1)^2 + j\omega_1}\right| \tag{5a}$$

$$|H(j\omega_2)| = \left|\frac{K}{T(j\omega_2)^2 + j\omega_2}\right| \tag{5b}$$

The magnitude of the transfer function $H(jw)$ is computed using the formula presented below.

$$|a + bj| = \sqrt{a^2 + b^2} \tag{6}$$

Using this gives an expression for the amplitude, presented in eq. (7).

$$|H(j\omega| = \sqrt{\frac{K^2}{T^2\omega^4 + \omega^2}} \tag{7}$$

The boat parameters in smooth weather conditions are found by solving eq. (7) for K and T, using ω_1 and ω_2, which yields two equations with two unknowns.

1.2.2 Results

The heading of the ship during a simulation using $\omega_1 = 0.005$ [rad s^{-1}] and a simulation using $\omega_2 = 0.05$ [rad s^{-1}] are shown in fig. 2 and fig. 3 respectively.

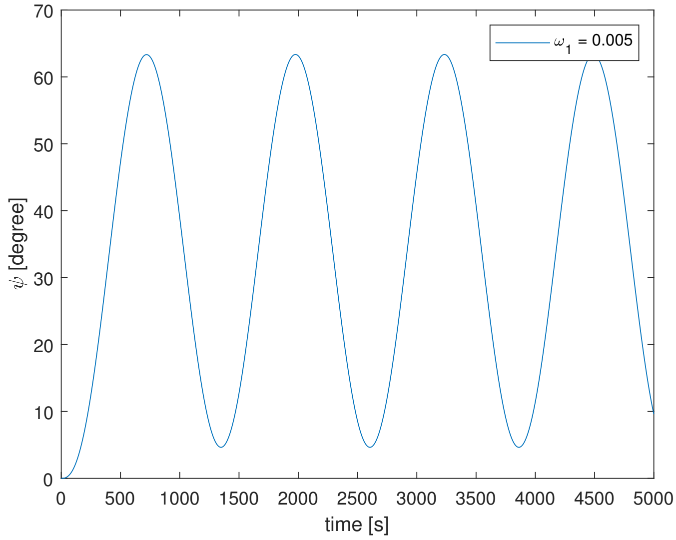

Figure 2: Heading of the ship for ω_1.

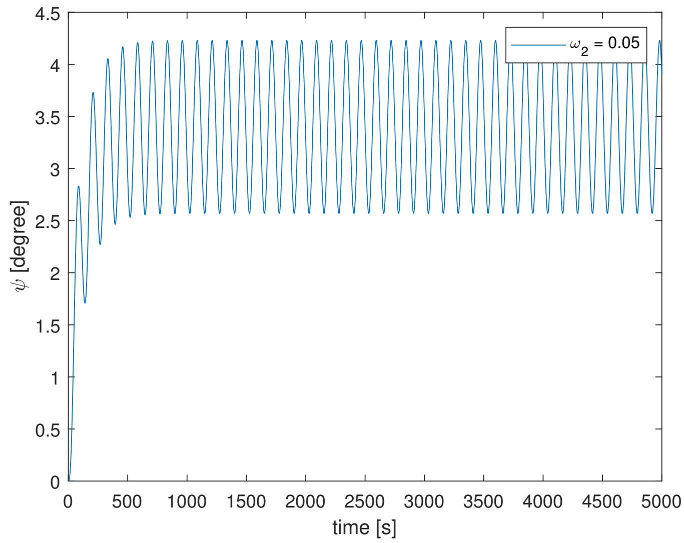

Figure 3: Heading of the ship for ω_2.

A reading from the output of the heading for the two frequencies gives amplitudes as presented in eq. (8).

$$|H(j\omega_1)| = 29.36 \qquad [-] \tag{8a}$$
$$|H(j\omega_2)| = 0.89 \qquad [-] \tag{8b}$$

The boat parameters in smooth weather conditions presented in the form of the Nomoto time and gain constants, T and K, are shown in eq. (9).

$$K = 0.15 \qquad [\mathrm{s}^{-1}] \tag{9a}$$
$$T = 66.15 \qquad [\mathrm{s}] \tag{9b}$$

1.3 Task c) Boat Parameters in Rough Weather Conditions

1.3.1 Method

The boat parameters T and K in rough weather conditions are found by applying waves and measurement noise to the system. The parameters are found by solving in the same manner as in section 1.2.

1.3.2 Results

The heading of the ship during a simulation using $\omega_1 = 0.005$ [rad s^{-1}] and a simulation using $\omega_2 = 0.05$ [rad s^{-1}] are show in fig. 4 and fig. 5 respectively.

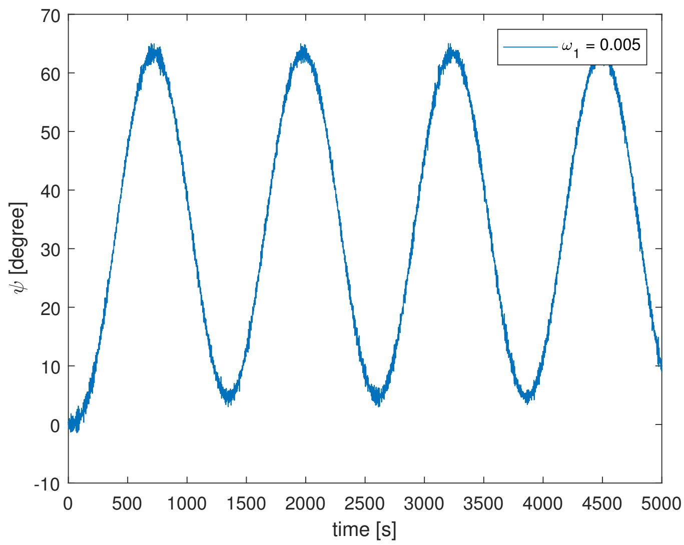

Figure 4: Heading of the ship for ω_1 with waves and measurement noise.

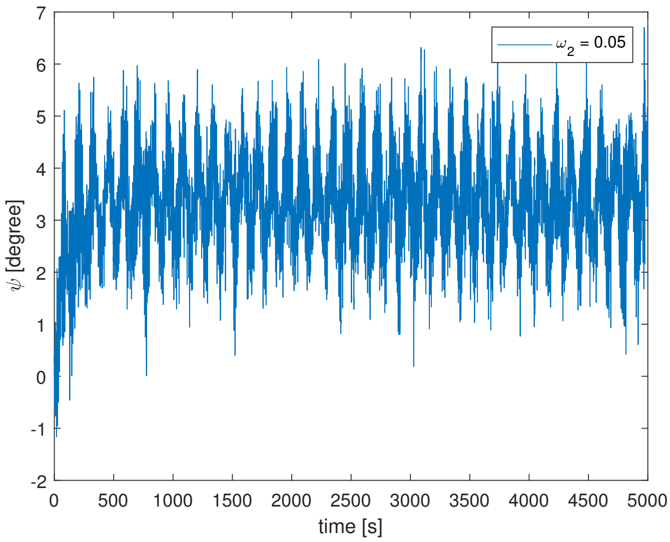

Figure 5: Heading of the ship for ω_2 with waves and measurement noise.

A reading from the output of the heading for the two frequencies gives amplitudes as presented in eq. (10).

$$|H(j\omega_1)| = 29 \qquad [-] \tag{10a}$$
$$|H(j\omega_2)| = 1.5 \qquad [-] \tag{10b}$$

The boat parameters in rough weather conditions presented in the form of the Nomoto time and gain constants, T and K, are shown in eq. (11).

$$K = 0.12 \qquad [\mathrm{s}^{-1}] \tag{11a}$$
$$T = 33 \qquad [\mathrm{s}] \tag{11b}$$

1.3.3 Discussion

Comparing the heading of the ship in section 1.2 with section 1.3, it is clear that the output is affected by the influence of the measurement noise and waves. The output signals in fig. 4 and fig. 5 contain a significant amount of noise compared to the output signals from section 1.2. This makes it difficult to estimate the amplitudes of the sine waves on the output. The values of the amplitudes given in eq. (10) are an approximation with a large uncertainty. The uncertainty in the reading affects the estimation of the boat parameters, as T and K are functions of the amplitude. This leads to less accurate estimates of the boat parameters for rough weather conditions.

1.4 Task d) Model Compared to Ship Response

1.4.1 Method

In order to compare the model with the ship, a block containing the transfer function $H(s)$ in eq. (3) is added to the Simulink model. Both the simulations of the ship and the model of the ship are given a step input of 1 [deg] to the rudder at $t = 0$ [s]. The Simulink model is shown in fig. 6.

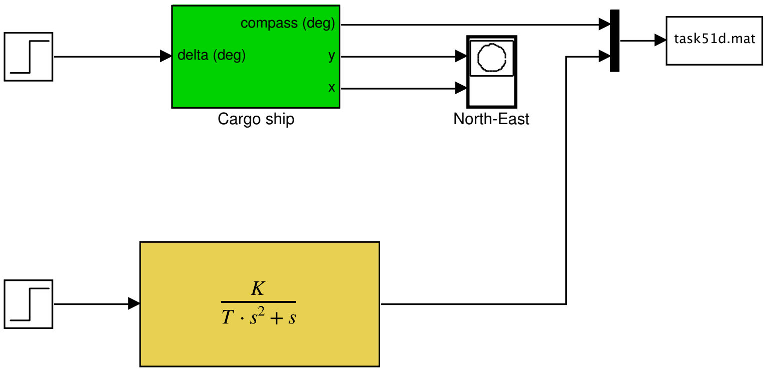

Figure 6: Simulink model of ship and model.

1.4.2 Results

Figure 7 shows the step response of the model and response of the ship.

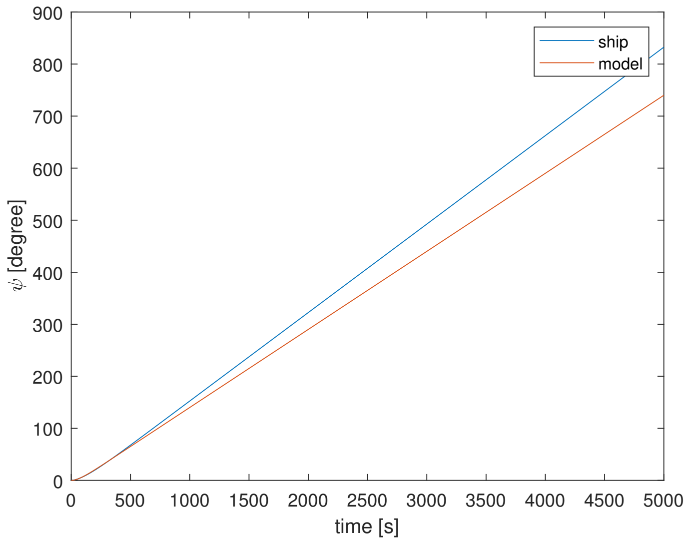

Figure 7: Comparing the step response of the model with the response of the ship.

1.4.3 Discussion

The response of the model aligns well with the response of the ship in the first 500 [s] of the simulation. The deviation from the ship increases linearly as time increases. At the end of the simulation the response of the model is approximately 90 [deg] off from the ship's response. This is a large deviation from the actual response, and the model is therefore not a good approximation of the ship.

2 Problem 5.2: Wave Spectrum Model

2.1 Task a) Estimate of Power Spectral Density Function

2.1.1 Method

In order to estimate the power spectral density function (PSD), $S_{\psi_w}(\omega)$, of the compass disturbances from the waves, ψ_w, the Welch power spectral density estimate is utilized. The estimate is calculated by the `MATLAB` function `[pxx,f] = pwelch(x,window,noverlap,nfft,fs)`. `pxx` is the PSD estimate $S_{\psi_w}(f)$ for the frequencies `f`. `x` is the sampled measurement of the wave disturbances on the compass, ψ_w, and `fs` is the sampling frequency, while `window`, `noverlap` and `nfft` are algorithmic parameters. `window`, `noverlap` and `nfft` are given by the assignment as 4096, `[]`, and `[]`. The sampling frequency is given as 10 [Hz], thus `fs` is set equal to 10. Since the measurement of the wave disturbance on the compass is given in degrees, ψ_w is converted to radians with the `MATLAB` function `deg2rad(D)`, where the input `D` is equal to the measured compass disturbances, ψ_w. The code can be seen in appendix A.2.

The PSD estimate and the frequencies returned by `pwelch` are given in [$\mathrm{W\,Hz^{-1}}$] and [Hz] respectively. The PSD estimate and the corresponding frequencies are converted to [$\mathrm{W\,s\,rad^{-1}}$] and [$\mathrm{rad\,s^{-1}}$], respectively, by using the following transforms:

$$S_{\psi_w}(\omega) = \frac{1}{2\pi} S_{\psi_w}(f) \tag{12a}$$

$$\omega = 2\pi f \tag{12b}$$

2.1.2 Results

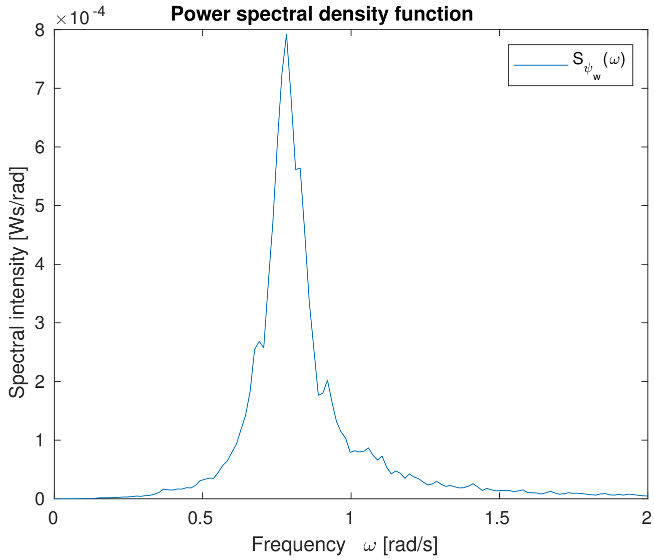

Figure 8: Numerically calculated estimate of the PSD of the compass disturbances from waves.

Using the MATLAB function `pwelch` with the measured wave disturbances in the file `wave.mat`, and transforming the result with the transforms in eq. (12), the PSD estimate in fig. 8 was obtained.

2.2 Task b) Transfer Function and Analytical PSD

2.2.1 Method

To find an analytical expression for the power spectral density function, P_{ψ_w}, the transfer function $\frac{\psi_w}{w_w}(s)$ is needed. By Laplace-transforming eq. (1a) and assuming a zero-state initial condition, the following result is obtained:

$$s\xi(s) = \psi_w(s)$$

This gives the following relation between $\xi(s)$ and $\psi_w(s)$:

$$\xi(s) = \frac{1}{s}\psi_w(s) \tag{14}$$

Then transforming eq. (1b) with a zero-state initial condition and inserting the relation in eq. (14) yields:

$$
\begin{aligned}
s\psi_w(s) &= -\omega_0^2\xi(s) - 2\lambda\omega_0\psi_w(s) + K_w w_w \\
s\psi_w(s) &= -\omega_0^2\frac{1}{s}\psi_w(s) - 2\lambda\omega_0\psi_w(s) + K_w w_w \\
s^2\psi_w(s) &= -\omega_0^2\psi_w(s) - 2\lambda\omega_0 s\psi_w(s) + K_w w_w s
\end{aligned}
$$

This results in the following transfer function:

$$
\frac{\psi_w}{w_w}(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \tag{15}
$$

The power spectral density function P_{ψ_w} is then calculated as:

$$
P_{\psi_w}(j\omega) = \frac{\psi_w}{w_w}(j\omega)\frac{\psi_w}{w_w}(-j\omega)P_{w_w}(j\omega) \tag{16}
$$

The power spectral density function of a stochastic, stationary signal is equal to the Fourier-transform of the autocorrelation function of the signal, also known as the Wiener-Khinchine relation [1]:

$$
P(j\omega) = \mathfrak{F}[R(\tau)] \tag{17}
$$

Since w_w is zero-mean Gaussian white noise, its autocorrelation function is equal to the Dirac-delta function, $\delta(\tau)$ [1]. Applying eq. (17) to this yields:

$$
\begin{aligned}
P_{w_w}(j\omega) &= \mathfrak{F}[R_{w_w}(\tau)] \\
&= \mathfrak{F}[\delta(\tau)]
\end{aligned}
$$

which results in

$$
P_{w_w}(j\omega) = 1 \tag{18}
$$

Inserting the results in eq. (18) and eq. (15) into eq. (16) gives

$$
\begin{aligned}
P_{\psi_w}(j\omega) &= \frac{K_w j\omega}{-\omega^2 + 2\lambda\omega_0\omega j + \omega_0^2} \cdot \frac{-K_w j\omega}{-\omega^2 - 2\lambda\omega_0\omega j + \omega_0^2} \cdot 1 \\
&= \frac{K_w \omega j}{(\omega_0^2 - \omega^2) + 2\lambda\omega_0\omega j} \cdot \frac{-K_w \omega j}{(\omega_0^2 - \omega^2) - 2\lambda\omega_0\omega j} \cdot 1
\end{aligned}
$$

which simplifies to the analytical expression for the power spectral density function of the wave disturbance:

$$
P_{\psi_w}(j\omega) = P_{\psi_w}(\omega) = \frac{K_w^2\omega^2}{(\omega_0^2 - \omega^2)^2 + (2\lambda\omega_0\omega)^2} \tag{19}
$$

2.2.2 Results

The analytical expression for the compass wave disturbances PSD was obtained by evaluating the transfer function of the wave disturbance and the power spectra of white noise exciting them. The analytical expression for the wave disturbance PSD was found to be:

$$P_{\psi_w}(\omega) = \frac{K_w^2 \omega^2}{(\omega_0^2 - \omega^2)^2 + (2\lambda\omega_0\omega)^2} \tag{20}$$

2.3 Task c) Natural Frequency and Intensity

2.3.1 Method

In order to find the natural frequency, ω_0, and its corresponding intensity σ^2, eq. (19) is differentiated as follows:

$$\begin{aligned}
\frac{dP_{\psi_w}(\omega)}{d\omega} &= \frac{2K_w^2\omega\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big] - 4K_w^2\omega^3\big[2\lambda^2\omega^2 - (\omega_0^2-\omega^2)\big]}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2} \\
&= \frac{2K_w^2\omega\big[(\omega_0^2-\omega^2)^2 + 4\lambda^2\omega_0^2\omega^2\big] - 8K_w^2\lambda^2\omega_0^2\omega^3 + 4K_w^2\omega^3(\omega_0^2-\omega^2)}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2} \\
&= \frac{2K_w^2\omega(\omega_0^2-\omega^2)^2 + 4K_w^2\omega^3(\omega_0^2-\omega^2) + \cancel{8K_w^2\lambda^2\omega_0^2\omega^3} - \cancel{8K_w^2\lambda^2\omega_0^2\omega^3}}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2} \\
&= \frac{2K_w^2\omega\big[\omega_0^4 - 2\omega_0^2\omega^2 + \omega^4\big] + 2K_w^2\omega\big[2\omega_0^2\omega^2 - 2\omega^4\big]}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2} \\
&= \frac{2K_w^2\omega\big[\omega_0^4 - \cancel{2\omega_0^2\omega^2} + \omega^4 + \cancel{2\omega_0^2\omega^2} - 2\omega^4\big]}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2}
\end{aligned}$$

This simplifies to:

$$\frac{dP_{\psi_w}(\omega)}{d\omega} = \frac{2K_w^2\omega(\omega_0^4 - \omega^4)}{\big[(\omega_0^2-\omega^2)^2 + (2\lambda\omega_0\omega)^2\big]^2} \tag{21}$$

Setting eq. (21) equal to zero, a non-trivial solution is that ω_0 is equal to the peak frequency of $P_{\psi_w}(\omega)$:

$$\frac{dP_{\psi_w}(\omega)}{d\omega} = 0 \implies \omega = \omega_0$$

In order for $P_{\psi_w}(\omega)$ and $S_{\psi_w}(\omega)$ to have similar shapes, the peak frequency of $P_{\psi_w}(\omega)$, ω_0, is set equal to the peak frequency of $S_{\psi_w}(\omega)$, denoted as ω_p. Similarly, the intensity of $P_{\psi_w}(\omega_0)$, σ^2, is set equal to the intensity of $S_{\psi_w}(\omega_p)$. The intensity, $S_{\psi_w}(\omega)$ and its corresponding index are found by using the MATLAB function `[M,I] = max(A)`. `M` and `I` are $S_{\psi_w}(\omega_p)$ and its corresponding index respectively, and `A` is the PSD estimate, $S_{\psi_w}(\omega)$. ω_p is then found by picking the frequency ω with the same index as that of $S_{\psi_w}(\omega_p)$. The code can be seen in appendix A.2.

2.3.2 Results

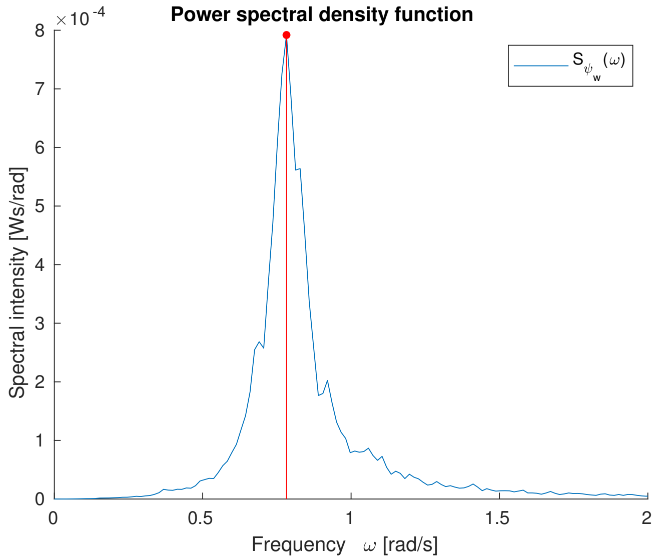

Figure 9: Numerically calculated estimate of the PSD of the wave disturbances on the compass, with illustration of ω_p and $S_{\psi_w}(\omega_p)$.

By finding the peak of the numerical power spectral density function estimate calculated in section 2.1 the natural frequency ω_0 and corresponding intensity σ^2 are approximately given by:

$$\omega_0 = \omega_p \approx 0.7823 \qquad [\mathrm{rad\,s^{-1}}] \tag{22a}$$

$$\sigma^2 = S_{\psi_w}(\omega_p) \approx 7.92 \cdot 10^{-4} \qquad [\mathrm{W\,s\,rad^{-1}}] \tag{22b}$$

2.4 Task d) Damping Factor

2.4.1 Method

The compass wave disturbance gain, K_w, is defined as:

$$K_w = 2\lambda\omega_0\sigma \tag{23}$$

where λ is the damping factor, ω_0 is the natural frequency, and σ is the square root of the intensity from the PSD function.

The analytical expression for the power density spectral function in eq. (19) then becomes:

$$P_{\psi_w}(\omega) = \frac{(2\lambda\omega_0\sigma\omega)^2}{(\omega_0^2 - \omega^2)^2 + (2\lambda\omega_0\omega)^2} \tag{24}$$

The only unknown in eq. (24) is the damping factor λ. In order to find λ, $P_{\psi_w}(\omega)$ needs to be curve fitted to the numerical estimate of the PSD, $S_{\psi_w}(\omega)$. The curve fitting is done by the built-in `MATLAB` optimization function `x = lsqcurvefit(fun, x0, xdata, ydata)`. `lsqcurvefit` returns the parameter `x` that minimizes the squared error between the function `fun(x, xdata)` and the data `ydata` [5]. In this case `fun` is an in-line function implementation of eq. (24). `x` is the damping factor λ, `x0` is the initial guess λ_0, `xdata` are the frequencies ω and `ydata` is the PSD estimate $S_{\psi_w}(\omega)$. The code can be found in appendix A.2.

2.4.2 Results

By curve fitting the analytical expression for $P_{\psi_w}(\omega)$ in eq. (24) to the numerical estimate, $S_{\psi_w}(\omega)$, found in section 2.1 and using the values for ω_0 and σ found in section 2.3, an estimate for the damping factor λ was found. With a functional tolerance of $1 \cdot 10^{-6}$ and an initial guess $\lambda_0 = 0.1$ $[-]$, `lsqcurvefit` returned the following value for λ:

$$\lambda \approx 0.083 \qquad [-] \tag{25}$$

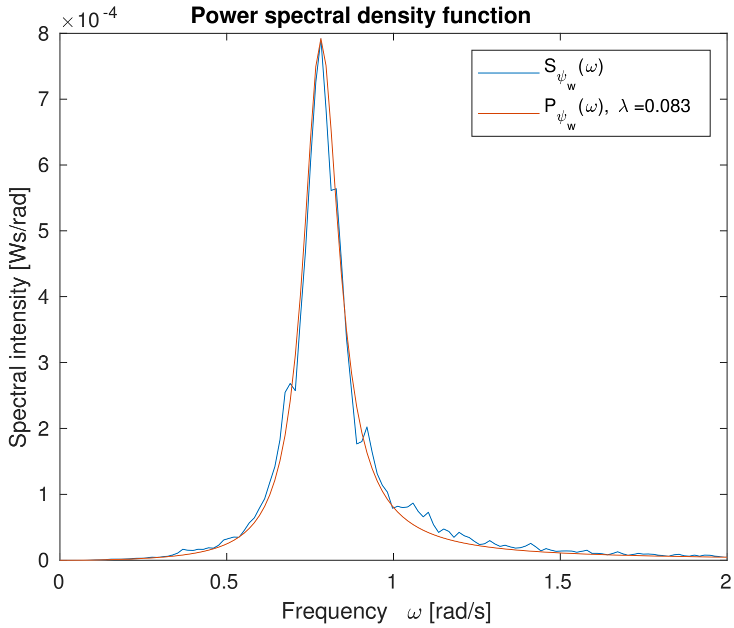

Figure 10: Numerically calculated estimate and curve fitted analytical expression of the PSD of wave disturbances.

3 Problem 5.3: Control system design

3.1 Task a) PD Controller

3.1.1 Method

A PD controller with transfer function

$$H_{pd}(s) = K_{pd}\frac{1 + T_d s}{1 + T_f s} \tag{26}$$

should be made based on the transfer function from δ to ψ, which is presented in eq. (3). The cutoff frequency, ω_c, and the phase margin, ϕ, of the open loop system $H_{pd}(s) \cdot H_{ship}(s)$ are determined by tuning the gain parameter K_{pd} and time constant T_f. The transfer function of the open loop system $H_{pd}(s) \cdot H_{ship}(s)$ is presented in eq. (27) below.

$$H_{pd}(s) \cdot H_{ship}(s) = K_{pd}\frac{1 + T_d s}{1 + T_f s}\frac{K}{Ts^2 + s} = K_{pd}\frac{1 + T_d s}{1 + T_f s}\frac{K}{s(1 + Ts)} \tag{27}$$

Choosing the derivative time constant T_d such that it cancels out the time constant from the transfer function $H_{ship}(s)$, i.e. setting $T_d = T$, yields the following transfer function for the open loop system:

$$H_{pd}(s) \cdot H_{ship}(s) = \frac{K_{pd}K}{s(1 + T_f s)} \tag{28}$$

Given that the cutoff frequency, w_c, is 0.1 [rad s^{-1}] and the phase margin, ϕ, is 50 [deg], the gain parameter K_{pd} and time constant T_f are found by setting up expressions for the cutoff frequency and the phase margin. The phase margin is given by:

$$\phi = \angle(H_{pd}(j\omega_c) \cdot H_{ship}(j\omega_c)) - (-180°) \qquad \text{[deg]} \tag{29}$$

$\angle(H_{pd}(j\omega_c) \cdot H_{ship}(j\omega_c))$ can be expressed as

$$\angle(H_{pd}(j\omega_c) \cdot H_{ship}(j\omega_c)) = \angle\frac{1}{s} + \angle\frac{1}{1 + T_f s}$$

which in turn implies that

$$\begin{aligned}\phi &= \angle\frac{1}{s} + \angle\frac{1}{1 + T_f s} - (-180°) \\ &= -90° - \arctan(T_f \omega_c) + 180°\end{aligned} \qquad \text{[deg]} \tag{30}$$

Given that the phase margin should be 50 [deg], eq. (30) gives an expression for T_f:

$$T_f = -\frac{\tan(50^\circ + 90^\circ - 180^\circ)}{\omega_c} \tag{31}$$

The cutoff frequency is where the amplitude of the open loop transfer function is 0 dB, as stated in eq. (32).

$$\left| \frac{K_{pd}K}{j\omega_c(1 + T_f j\omega_c)} \right| = 0\text{dB} = 1 \tag{32}$$

Equation (32) written out becomes:

$$\left| \frac{K_{pd}K}{j\omega_c - T_f\omega_c^2} \right| = \frac{\sqrt{K_{pd}^2 K^2}}{\sqrt{(-T_f\omega_c^2)^2 + \omega_c^2}} = \frac{K_{pd}K}{\omega_c\sqrt{1 + T_f^2\omega_c^2}} = 0\text{dB} = 1$$

This in turn yields:

$$K_{pd} = \frac{\omega_c\sqrt{1 + T_f^2\omega_c^2}}{K} \tag{33}$$

To validate the results, the Bode plot for the open loop system is created by using the `MATLAB` function `bode(sys)`, with `sys` being a `MATLAB` transfer function variable of the open loop transfer function in eq. (27). The code can be seen in appendix A.4.

3.1.2 Results

From eq. (31), T_f is found as $T_f = 8.39$ [s]. Given that $\omega_c = 0.1$ [rad s^{-1}], and finding K from eq. (7) and T_f from eq. (31), using eq. (33) gives $K_{pd} = 0.87$ [$-$]. The resulting gain parameter and time constant are presented in eq. (34).

$$K_{pd} = 0.87 \qquad [-] \tag{34a}$$

$$T_f = 8.39 \qquad [\text{s}] \tag{34b}$$

The resulting Bode plot of the open loop transfer function in eq. (27), with the calculated gain parameters and time constants, is presented in fig. 11.

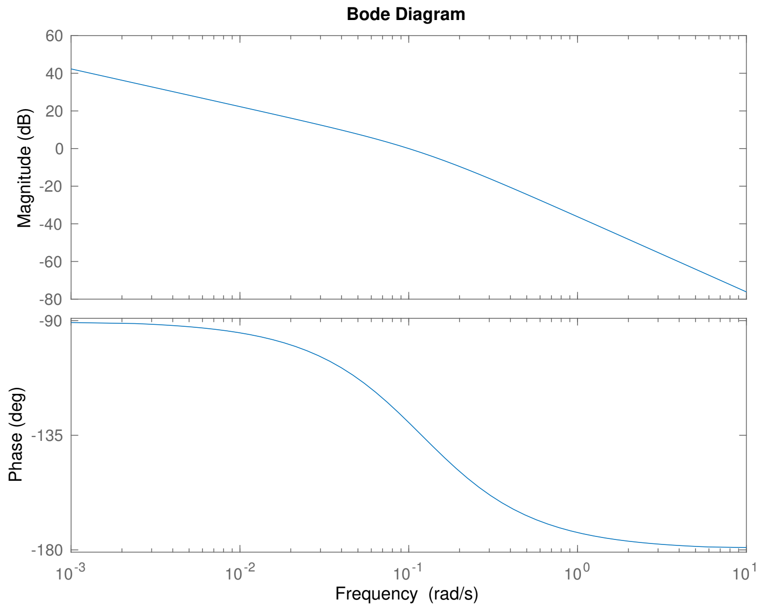

Figure 11: Bode plot of the open loop transfer function $H_{pd}(j\omega) \cdot H_{ship}(j\omega)$

3.1.3 Discussion

The heading of the ship must be less than ± 35 [deg]. When the heading reference is set to 30 [deg], the heading of the ship turned out to always be less than the boundary. Thus, there was no need to implement a saturation element.

3.2 Task b) Autopilot without Disturbances

3.2.1 Method

Implementation of the PD controller in Simulink was done by implementing the transfer function as a transfer function block, and adding the gain parameter K_{pd}. The error was calculated by subtracting the measured heading from the desired heading. The implementation of the controller in Simulink is shown in section 3.2.2.

The system is simulated by turning off the disturbances (waves and current), and turning on the measurement noise in the cargo ship block.

3.2.2 Results

The implementation of the PD controller in Simulink is shown in fig. 12.

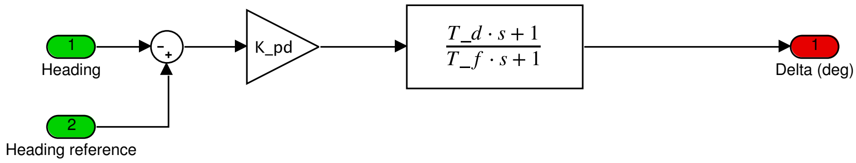

Figure 12: Implementation of PD controller in Simulink

The response of the system and the rudder input when running the simulation with measurement noise, without disturbances, are presented in fig. 13 and fig. 14 respectively.

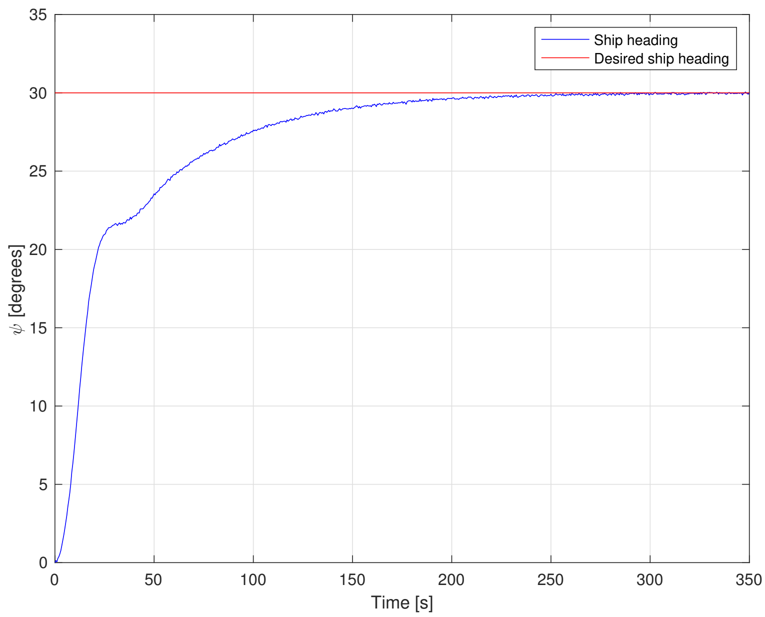

Figure 13: Desired heading vs. actual heading.

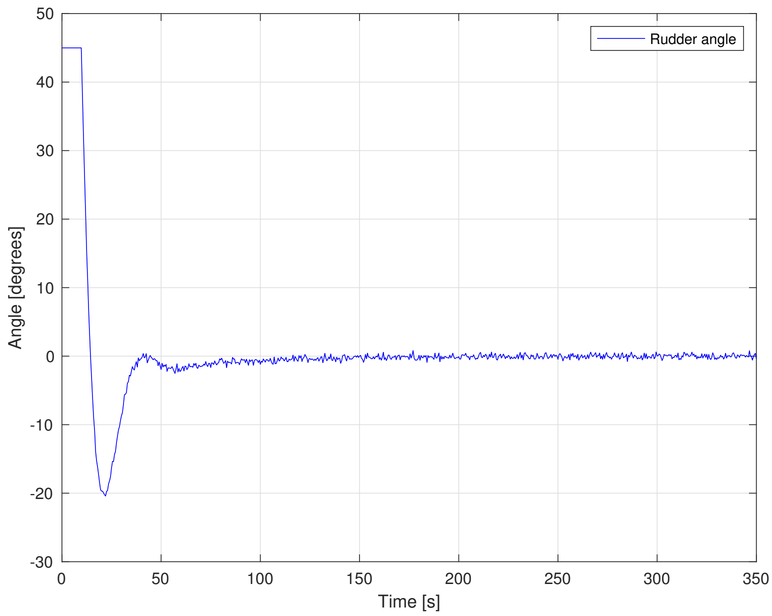

Figure 14: Input to rudder.

3.2.3 Discussion

As seen in fig. 13 the ship reaches the reference angle within approximately 200 [s] and manages to keep the desired heading with no steady state error. From the results one can see that the derivative term in the PD controller leads to a negative input signal to the rudder around 25 [s] into the simulation. This leads to a minor halt in the response at around 35 [s], slowing down the overall response, but in exchange the response does not overshoot the reference. This is one of the advantages that a PD controller has over a regular P controller.

Initially, when running the simulation, the input signal to the rudder angle was instantly very large, about 200 [deg. This is due to the proportional term and the large error in the heading when the system starts. By examination of the cargo ship block, the rudder angle was found to be limited within an interval of ±45 [deg]. A saturation element was then added to the input signal of the rudder angle, making sure that the measured input is coinciding with what is actually happening in the ship process. Given the results from fig. 14 as well as the response from fig. 13, the autopilot of the ship has been proven to work.

3.3 Task c) Autopilot with Current

3.3.1 Results

Plots of the response of the system and the input signal to the rudder, when running the simulation with measurement noise and current disturbance, are presented in fig. 15 and fig. 16 respectively.

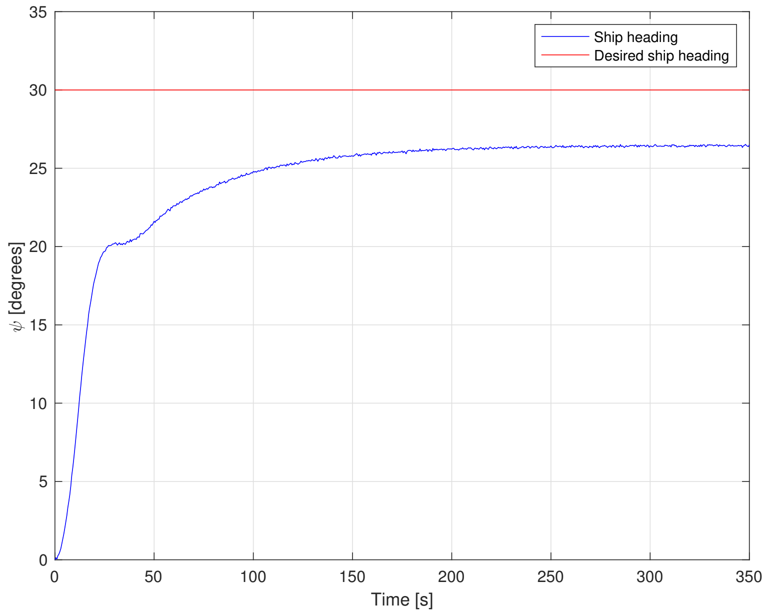

Figure 15: Desired heading vs. actual heading.

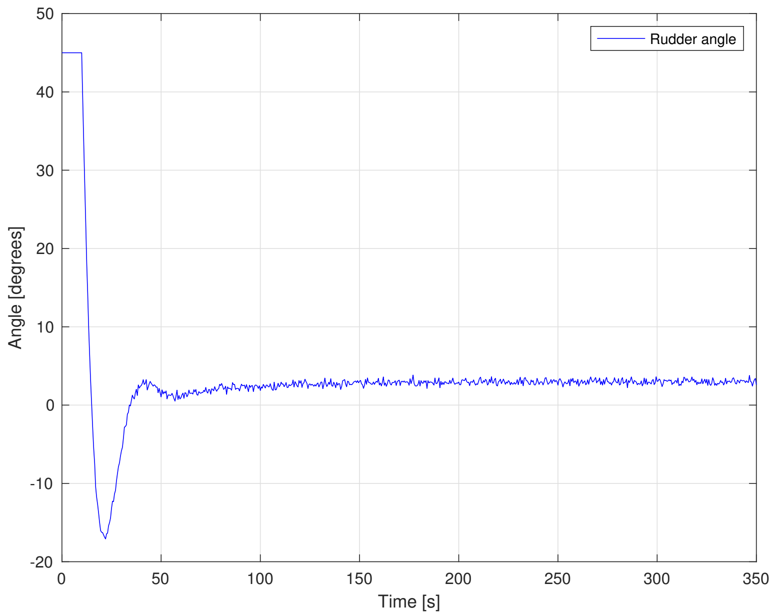

Figure 16: Input to rudder.

3.3.2 Discussion

In fig. 15, one can clearly see a steady state error. This is due to the fact that the system now includes disturbances, in the form of current. There was no steady state error in section 3.2 because there was no disturbance present. The ship stabilizes after approximately 200 [s], with a steady state error of around 4 [deg]. With current applied in the simulation, fig. 16 shows a similar response for the input signal of the rudder angle as for the simulation containing only noise.

Since there is a steady state error present of significant size, the autopilot system does not work satisfactory with these environmental loads. By including an integral effect in the controller the steady state error could have been eliminated, but might have lead to the response overshooting the reference.

3.4 Task d) Autopilot with Waves

3.4.1 Results

Plots of the response of the system and the input signal to the rudder when running the simulation with measurement noise and wave disturbance are presented in fig. 17 and fig. 18 respectively.

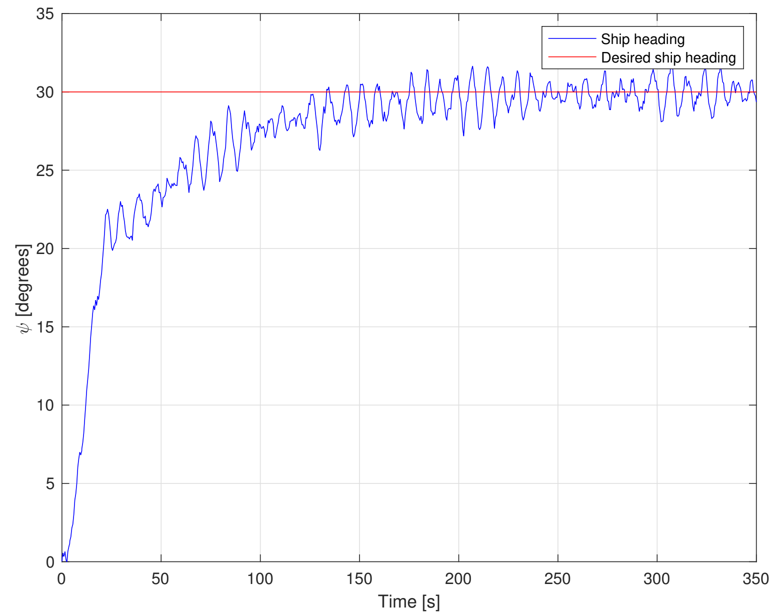

Figure 17: Desired heading vs. actual heading.

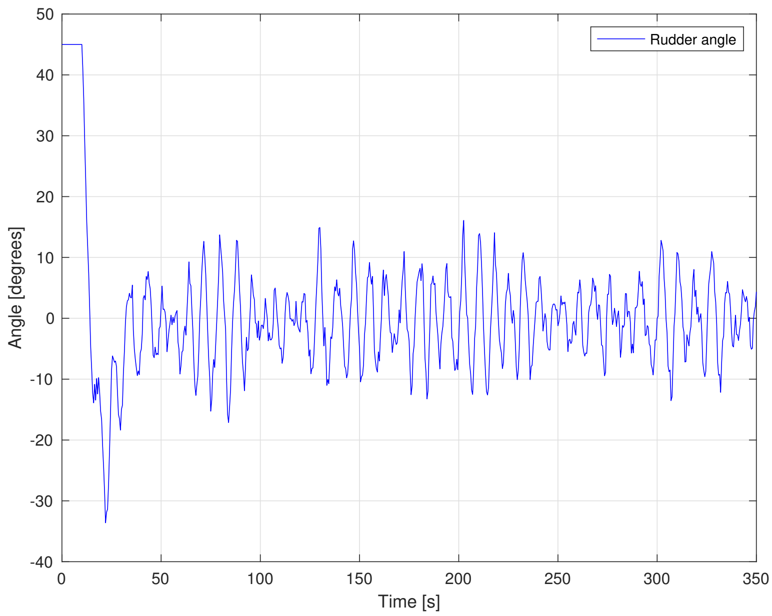

Figure 18: Input to rudder.

3.4.2 Discussion

From fig. 17, one can see that the ship heading is oscillatory. This is because of the high frequency waves affecting the ship model. The ship reaches the reference angle after approximately 200 [s], however there are some considerable fluctuations. An explanation for why there seems to be no steady state error in this case, even though there are disturbances in form of waves, could be that the waves are constantly changing, not giving a resulting force in the same direction, as the constant current in the previous task most likely did. Had the mean drift forces or other nonlinear wave effects been included, there might have been a steady state error.

In fig. 18, the rudder angle is clearly oscillatory. Again, this is due to the high frequency motions on the ship caused by the waves, which the rudder tries to compensate for. Calculating control inputs in order to counteract these high frequency motions seriously increases the wear and tear on the actuators, may they be thrusters or rudders, and does not increase performance in the control system. Therefore, the wave induced response of the ship should be filtered out from the measured response, so that the control system does not try to counteract the high frequency motions.

Because of the high wear and tear on the actuators when the high frequency waves are included in the calculations of the control outputs, the system does not work satisfactory despite the fact that the reference heading is reached.

4 Problem 5.4: Observability

4.1 Task a) State Matrices

4.1.1 Method

The system described by eq. (1) is written in a state space model as seen in eq. (35).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w} \tag{35a}$$
$$y = \mathbf{C}\mathbf{x} + v \tag{35b}$$

where $\mathbf{x} = [\xi_w \; \psi_w \; \psi \; r \; b]^T$, $u = \delta$, $\mathbf{w} = [w_w \; w_b]^T$, and v is the measurement noise.

4.1.2 Results

The matrices **A**, **B**, **C**, and **E** are given by eq. (36).

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0{}^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} \tag{36a}$$

$$\mathbf{E} = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{36b}$$

4.2 Task b) Observability without Disturbances

4.2.1 Method

To figure out if the system is observable without disturbances, the system is reduced to eq. (37).

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} \mathbf{u} \tag{37}$$

where $\mathbf{x} = [\psi \; r]^T$ and $u = \delta$. The output is given by eq. (38).

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} + v \tag{38}$$

Thus,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{39}$$

The observability matrix is given by eq. (40)

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{40}$$

where n is the number of states.

If the observability matrix has full rank, i.e. $rank(\mathcal{O}) = 2$, the system is observable. Observability means that for a finite time $t_1 > 0$ it is possible to to determine the unknown initial state $\mathbf{x}(0)$ using the input u and output $\mathbf{y}$ over $[0, t_1]$ [2].

Using the formula in eq. (40), the observability matrix is computed, as seen in eq. (41).

$$\mathcal{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{41}$$

The rank of the observability matrix equals the number of linearly independent columns of $\mathcal{O}$. For the system without disturbances, the rank of $\mathcal{O}$ equals 2.

4.2.2 Results

The rank of the observability matrix is 2, implying that the matrix has full rank. This means that the system is observable without disturbances.

4.3 Task c) Observability with Current

4.3.1 Method

With only current disturbances working on the system, the system is reduced to eq. (42).

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \tag{42}$$

where $\mathbf{x} = [\psi \; r \; b]^T$, $u = \delta$, and $w = w_b$. The output is given by eq. (43).

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} + v \tag{43}$$

Thus,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{1}{K} \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{44}$$

The observability matrix is computed using the MATLAB function `Ob = obsv(A,C)`, with the matrices $\mathbf{A}$ and $\mathbf{C}$ as input and $\mathcal{O}$ as output. Equation (45) shows the resulting observability matrix.

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \end{bmatrix} \tag{45}$$

To find the rank of the observability matrix, the MATLAB function `k = rank(A)` is used. Given $\mathcal{O}$ as input, the function returns the column rank of $\mathcal{O}$, which is 3 for the given system.

4.3.2 Results

The system with current disturbances has an observability matrix with rank 3, i.e. full rank, which means that the system is observable.

4.4 Task d) Observability with Waves

4.4.1 Method

With only wave disturbances working on the system, the system is reduced to eq. (46).

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\xi}_w \\ \dot{\psi}_w \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -{\omega_0}^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \end{bmatrix} u + \begin{bmatrix} 0 \\ K_w \\ 0 \\ 0 \end{bmatrix} w \tag{46}$$

where $\mathbf{x} = [\xi_w \ \psi_w \ \psi \ r]^T$, $u = \delta$, and $w = w_w$. The output is given by eq. (47).

$$y = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \mathbf{x} + v \tag{47}$$

Thus,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \tag{48}$$

The observability matrix is computed using the MATLAB function `Ob = obsv(A,C)`, which takes the matrices $\mathbf{A}$ and $\mathbf{C}$ as input and returns $\mathcal{O}$. Equation (49) shows the resulting observability matrix.

$$\mathcal{O} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -{\omega_0}^2 & -2\lambda\omega_0 & 0 & 1 \\ 2\lambda{\omega_0}^3 & 4\lambda^2{\omega_0}^2 - {\omega_0}^2 & 0 & -\frac{1}{T} \\ -{\omega_0}^2(4\lambda^2{\omega_0}^2 - {\omega_0}^2) & -8\lambda{\omega_0}^3 - 4\lambda{\omega_0}^3 & 0 & \frac{1}{T^2} \end{bmatrix} \tag{49}$$

Using the function `k = rank(A)` in `MATLAB`, the rank of $\mathcal{O}$ is computed to 4 for the given system.

4.4.2 Results

The system with wave disturbances has an observability matrix with rank 4, i.e. full rank, which implies that the system is observable.

4.5 Task e) Observability with Current and Waves

4.5.1 Method

With both wave and current disturbances in the system, the **A** matrix and **C** matrix are the same as those found in problem 5.4 a. Using these matrices as input to the `MATLAB` function `Ob = obsv(A,C)`, the observability matrix is computed. This yields eq. (50).

$$\mathcal{O} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -{\omega_0}^2 & -2\lambda\omega_0 & 0 & 1 & 0 \\ 2\lambda{\omega_0}^3 & 4\lambda^2{\omega_0}^2 - {\omega_0}^2 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ -{\omega_0}^2(4\lambda^2{\omega_0}^2 - {\omega_0}^2) & -8\lambda^3{\omega_0}^3 + 4\lambda{\omega_0}^3 & 0 & \frac{1}{T^2} & \frac{K}{T^2} \\ -{\omega_0}^2(-8\lambda^3{\omega_0}^3 + 4\lambda{\omega_0}^3) & -12\lambda^2{\omega_0}^4 + {\omega_0}^4 + 16\lambda^4{\omega_0}^4 & 0 & -\frac{1}{T^3} & -\frac{K}{T^3} \end{bmatrix} \tag{50}$$

The rank of the observability matrix is calculated using the `MATLAB` function `k = rank(A)`, which gives $rank(\mathcal{O}) = 5$. Since the system has a state vector with five dimensions, the observability matrix has full rank.

4.5.2 Results

With waves and current in the system, the rank of the observability matrix is 5, i.e. full rank. With full rank of the observability matrix, the system is observable.

4.5.3 Discussion

The results from the subproblems in task 5.4 shows that the system is observable both with and without wave and current disturbances. This implies that the initial state can be determined from the input u and the output **y**. In order to implement an observer, the system must be observable. One such observer is the Kalman filter. Thus, a Kalman filter can be used to estimate the states regardless of what environmental loads are present in the process.

5 Problem 5.5: Discrete Kalman Filter

5.1 Task a) Discretization

5.1.1 Method

The model found in section 4.1 is discretized using exact discretization with a sampling frequency of 10 Hz. The model is transformed to eq. (51)

$$\mathbf{x}[k+1] = \bar{\boldsymbol{A}}\mathbf{x}[k] + \bar{\boldsymbol{B}}u[k] + \bar{\mathbf{E}}w[k] \tag{51a}$$

$$y[k] = \boldsymbol{C}\mathbf{x}[k] + \bar{v}[k] \tag{51b}$$

where $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{E}}$, and $\bar{v}$ are given by eq. (52). The discretized $\mathbf{C}$ matrix $\bar{\mathbf{C}}$ is equal to the $\mathbf{C}$ matrix in the continuous time domain [2, p.110] found in section 4.1. Given that the input $u[k]$ is produced by a zero-order hold and assuming that the disturbance $w[k]$ can be considered constant over the time interval $[kT, (k+1)T]$, which is merely an approximation, the matrices $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, and $\bar{\mathbf{E}}$ and noise $\bar{v}$ are given by:

$$\bar{\mathbf{A}} = e^{\mathbf{A}T} \tag{52a}$$

$$\bar{\mathbf{B}} = \int_0^T e^{\mathbf{A}\alpha}\mathbf{B}d\alpha \tag{52b}$$

$$\bar{\mathbf{E}} = \int_0^T e^{\mathbf{A}\alpha}\mathbf{E}d\alpha \tag{52c}$$

$$\bar{v} = \frac{1}{T}\int_0^T v(kT-\alpha)d\alpha \tag{52d}$$

The matrices $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, and $\bar{\mathbf{E}}$ are found using the `MATLAB` function `sysd = c2d(sys,Ts)` which transform a system `sys` from continuous to discrete time with the sampling time `Ts` using a zero-order hold discretization [4]. The matrices are found using the principle of superposition, where we first ignore the disturbance w and consider only u as input. $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are computed by providing `c2d` with the matrices $\mathbf{A}$ and $\mathbf{B}$ and the sampling time T. Secondly, the input u is ignored and the disturbance w is considered the only input to the system. $\bar{\mathbf{E}}$ is then computed by providing `c2d` with the matrices $\mathbf{A}$ and $\mathbf{B}$ and the sampling time T. The code is shown in appendix A.6.

5.1.2 Results

The matrices in the discrete system are presented in eq. (53).

$$\bar{\mathbf{A}} = \begin{bmatrix} 0.997 & 0.099 & 0 & 0 & 0 \\ -0.061 & 0.984 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.099 & -10^{-5} \\ 0 & 0 & 0 & 0.998 & -2 \cdot 10^{-4} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{53a}$$

$$\bar{\mathbf{B}} = \begin{bmatrix} 0 \\ 0 \\ 10^{-5} \\ 2 \cdot 10^{-4} \\ 0 \end{bmatrix} \tag{53b}$$

$$\bar{\mathbf{E}} = \begin{bmatrix} 1.8 \cdot 10^{-5} & 0 \\ 3.6 \cdot 10^{-4} & 0 \\ 0 & -3.8 \cdot 10^{-7} \\ 0 & 1.1 \cdot 10^{-5} \\ 0 & 0.100 \end{bmatrix} \tag{53c}$$

5.2 Task b) Variance of Measurement Noise

5.2.1 Method

In order to find the covariance matrix R used in the Kalman filter, the variance of the measurement noise needs to be estimated. Since the measurement noise is a scalar, R is also a scalar equal to the variance of the measurement noise.

In order to find the variance of the measurement noise, the rudder input is set to zero, and the system is run without wave and current disturbances. By doing this the measurement only consists of the measurement noise. Using the `MATLAB` function `V = var(A)` with the measured heading ψ gives the variance in [deg^2].
Since most of the variables in the Kalman filter are presented in radians, the calculated variance is converted from [deg^2] to [rad^2].

5.2.2 Results

The variance of the measurement noise from the simulation was found to be:

$$\sigma_v^2 \approx 0.002 \qquad [\text{deg}^2] \tag{54}$$

Converting the variance from [deg^2] to [rad^2] yielded:

$$\sigma_v^2 \approx 6.09 \cdot 10^{-7} \qquad [\text{rad}^2] \tag{55}$$

5.3 Task c) Discrete Kalman Filter

5.3.1 Method

In order to implement a discrete Kalman filter for the system in eq. (51) the process noise covariance $\bar{\boldsymbol{Q}}$ and measurement noise covariance $\bar{R}$ for the discrete system are needed. For the continuous time system with the disturbance vector $\mathbf{w} = [w_w, w_b]^T$, $\boldsymbol{Q}$ is given in the assignment as:

$$\boldsymbol{Q} = E[\boldsymbol{w}\boldsymbol{w}^T] = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix} \tag{56}$$

Since the observer is discrete, the covariance matrices for the discretized system are need. The discrete measurement noise covariance $\bar{R}$ equals the variance found in section 5.2, divided by the sampling time, see eq. (57).

$$\bar{R} = \frac{R}{T} \tag{57}$$

The discrete process noise covariance $\bar{\mathbf{Q}}$ is computed using eq. (58) [3, p.297].

$$\bar{\mathbf{Q}} = \bar{\mathbf{E}}\mathbf{Q}\bar{\mathbf{E}}^T \tag{58}$$

where $\bar{\mathbf{E}}$ is the disturbance matrix for the discretized system found in section 5.1.

The initial conditions for the a priori estimate $\hat{x}_0^-$ and covariance matrix P_0^- are given in the assignment as:

$$P_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-1} \end{bmatrix} \quad \hat{x}_0^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{59}$$

The Kalman matrix gain is calculated as:

$$\boldsymbol{K}[k] = \boldsymbol{P}^-[k]\boldsymbol{C}^T\big(\boldsymbol{C}\boldsymbol{P}^-[k]\boldsymbol{C}^T + \bar{R}\big)^{-1} \tag{60}$$

The updated estimates for the states and error covariance are calculated as:

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\big(y[k] - \boldsymbol{C}\hat{\boldsymbol{x}}^-[k]\big) \tag{61}$$

$$\boldsymbol{P}[k] = \big(\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{C}\big)\boldsymbol{P}^-[k]\big(\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{C}\big)^T + \boldsymbol{K}[k]\bar{R}\boldsymbol{K}[k] \tag{62}$$

The discrete time a priori state and error covariance estimates are calculated as:

$$\hat{\boldsymbol{x}}^-[k+1] = \bar{\boldsymbol{A}}\hat{\boldsymbol{x}}[k] + \bar{\boldsymbol{B}}u[k] \tag{63}$$

$$\boldsymbol{P}^-[k+1] = \bar{\boldsymbol{A}}\boldsymbol{P}[k]\bar{\boldsymbol{A}}^T + \bar{\boldsymbol{Q}} \tag{64}$$

The Kalman filter is implemented using the `MATLAB` function block in Simulink, see fig. 19. The corresponding `MATLAB` script is shown in appendix A.7. When predicting the a priori state $\hat{\boldsymbol{x}}^-[k+1]$ and error covariance $\boldsymbol{P}^-[k+1]$ in the Kalman filter, values of the estimated state $\hat{\boldsymbol{x}}^[k]$, input $u[k]$, and error covariance $\boldsymbol{P}[k]$ from the last iteration are needed. In order to keep these variables in the Kalman filter for the next iteration, the variables are declared `persistent`. Persistent variables are retained in memory between calls to the function.

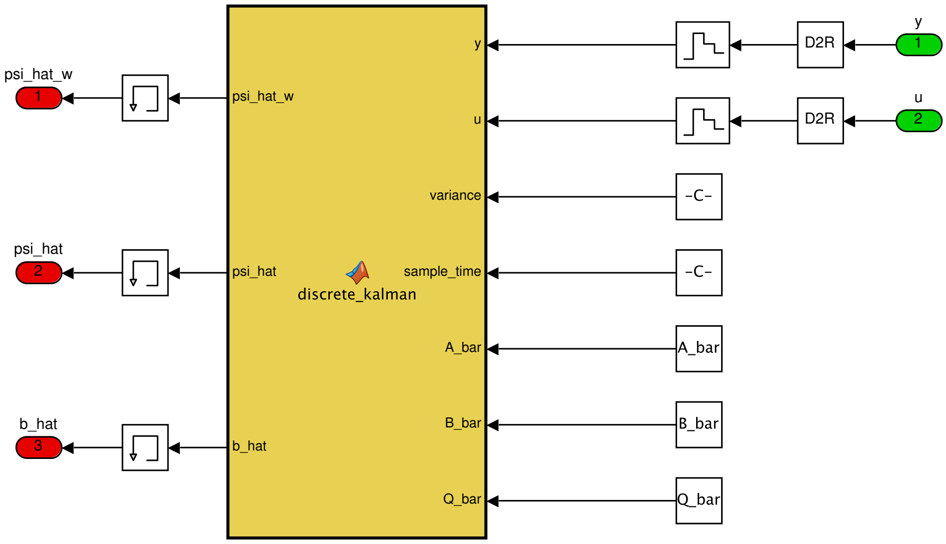

Figure 19: Simulink model of Kalman filter

The parameters needed in the Kalman function block are initialized in the script `task5.m` and passed to the Kalman block using constant blocks in Simulink. The inputs to the Kalman filter block are the measured heading y and the rudder angle u. The input variables are both measured in degrees, so they need to be converted to radians in order to match the units of the Kalman filter. Additionally, these two variables are continuous and need to be discretized. This is implemented using a zero-order hold block with the same sampling frequency as in section 5.2. The outputs from the Kalman filter block are the estimated high-frequency wave induced motion on the heading $\hat{\psi_w}$, the estimated heading $\hat{\psi}$, and the estimated bias $\hat{b}$. To avoid an unsolvable algebraic loop in the simulation, three memory blocks are implemented on the output signals. The memory block delays its input by one iteration. The memory block's sample time is inherited from the Kalman filter.

5.3.2 Discussion

With a memory block implemented in the Kalman filter, a time delay is introduced in the system equal to the sampling time. However, since the sampling time is chosen to be much smaller than the time constant of the process, the effect of the time delay is considered to be insignificant.

5.4 Task d) Bias Estimation

5.4.1 Method

The system is simulated with current disturbances using a reference heading $\psi_r = 30$ [deg]. The rudder angle contains a bias due to current disturbances. The bias is estimated in the Kalman filter. Using feed forward, the bias is cancelled by adding the estimated bias to the rudder input, as seen in fig. 20.

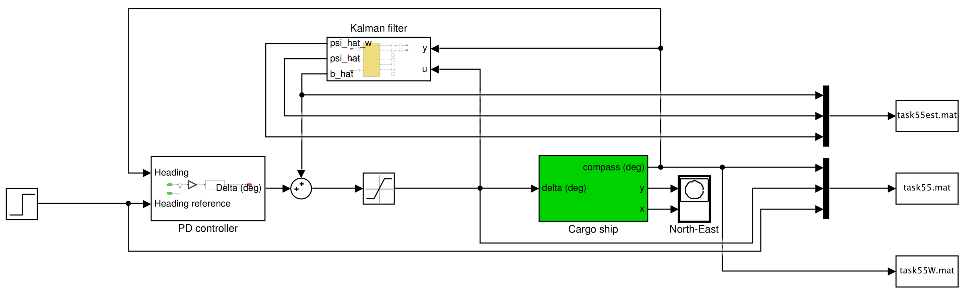

Figure 20: Simulink model with bias estimation

This introduces a new term in eq. (1d), which becomes:

$$
\begin{aligned}
\dot{r} &= -\frac{1}{T}r + \frac{K}{T}(\delta - b + \hat{b}) \\
&= -\frac{1}{T}r + \frac{K}{T}(\delta - (b - \hat{b}))
\end{aligned}
$$

Introducing the bias estimate error $\widetilde{b} = b - \hat{b}$ yields the following equation for the yaw acceleration:

$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - \widetilde{b}) \tag{65}$$

5.4.2 Results

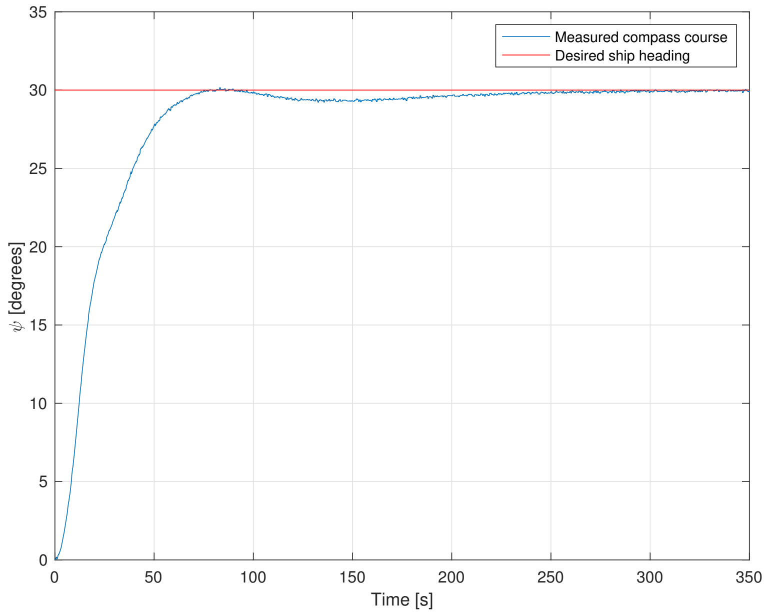

Figure 21: Desired heading vs. actual heading

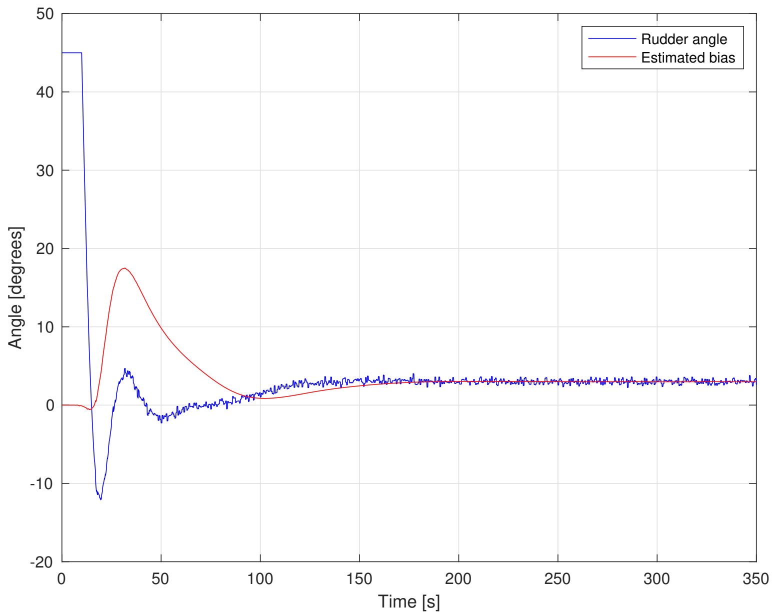

Figure 22: Estimated bias vs. rudder input

5.4.3 Discussion

Comparing this simulation with the equivalent simulation from section 3.3, fig. 21 shows that the steady state error in fig. 15 has been eliminated. This is due to the fact that the feed forward of the estimated bias cancels out the bias of the rudder angle generated by the current. Having eliminated the steady state error of approximately 4 [deg] from the simulation from section 3.3, the autopilot clearly has a better performance when the estimated bias is fed forward to cancel the bias of the rudder angle than without bias feed forward.

When modelling the ship, the heading was assumed to have small deviations from the reference heading. This was taken into account by making sure that ψ was between ± 35 [deg]. The results showed that implementing a saturation element in the Simulink model was excessive, as the heading was always within the boundaries.

5.5 Task e) Autopilot with waves and current

5.5.1 Method

The system is simulated with wave and current disturbances using the same heading reference as in the latter task. The high-frequency wave induced component of the heading motion must be removed from the control loop, since the controller should not counteract for the high-frequency component of the heading as this leads to wear and tear of the actuator system. The wave filtering is implemented by estimating the heading ψ using the Kalman filter and feeding the filtered heading back to the autopilot. The resulting Simulink model is shown in fig. 23.

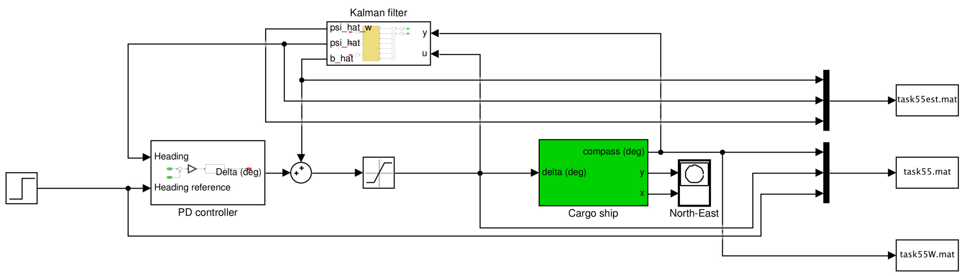

Figure 23: Simulink model with wave filtering

5.5.2 Results

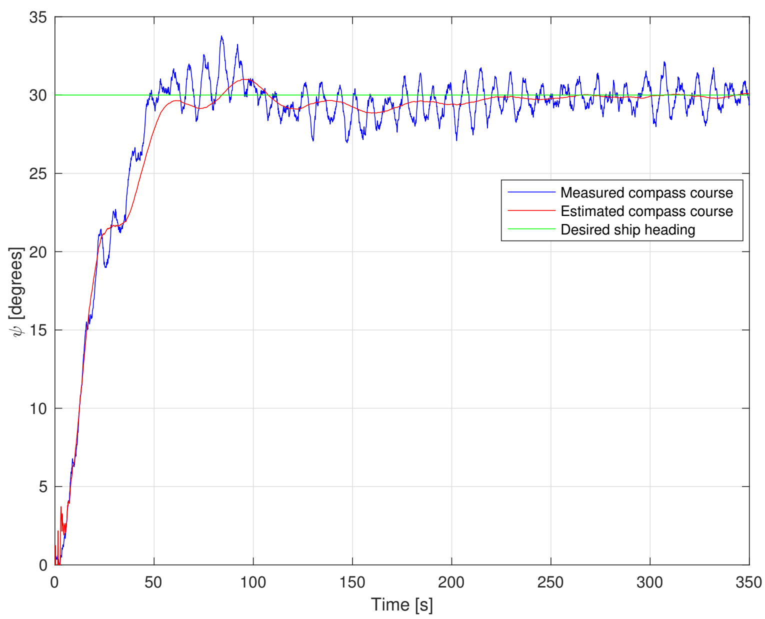

Figure 24: Desired heading vs. actual heading

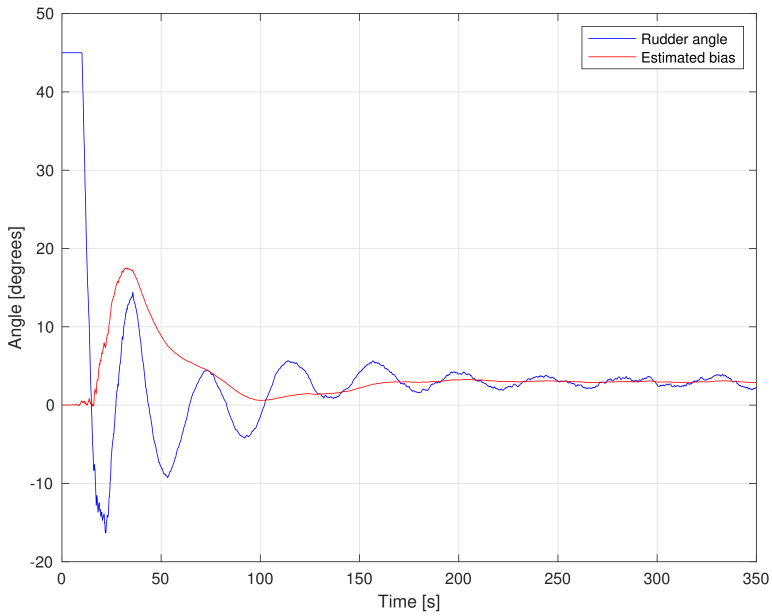

Figure 25: Estimated bias vs. rudder input

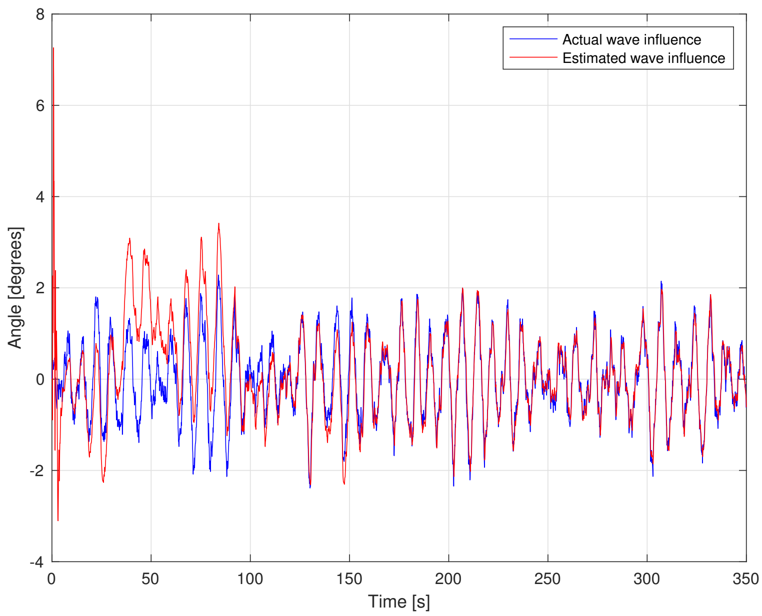

Figure 26: Wave influence vs. estimated wave influence

5.5.3 Discussion

Comparing the simulation to the equivalent simulation in section 3.4, fig. 24 shows a quicker response than the response in fig. 17, despite the fact that the simulation section 3.4 does not include current. The heading reference is reached after around 50 seconds, while in the simulation in section 3.4 the reference was reached after about 200 seconds. Figure 24 also shows that there is no steady state error between the heading reference and the heading, despite the fact that the system has current disturbance. The steady state error is eliminated due to the feed forward of the bias estimate to the rudder input signal, which can be seen in fig. 25.

Figure 25 shows that the high frequency components in the rudder input signal , which can be seen in fig. 18, are filtered out by the Kalman filter.
As seen in fig. 26, the estimated wave influence is generally quite close to the measured wave influence, except from some deviations within the initial 70 seconds. The error in the wave disturbance estimate in the initial 70 seconds may be the reason why the response has excessive low frequency oscillations with periods of about 50 seconds. The error in the estimate most likely causes the rudder to either over- or undercompensate for the motions of the ship, leading to it overshooting the reference and thus creating oscillations. This argument can be backed further by the fact that the oscillations get smaller as the wave disturbance estimate error gets smaller. Apart from in the initial 70 seconds, the estimated wave influence is generally a good approximation of the measured wave influence and the Kalman can thus filter out most of the high frequency wave-induced motions. This highly reduces the wear and tear on the actuators and generally leads to a better performance of the system.

Having shown that the desired heading is reached faster, as well as the rudder not accounting for the high-frequency motions, the conclusion is that the autopilot with the Kalman filter and bias feed forward has better performance than the one in section 3.4.

5.6 Task f) Process Noise Covariance Matrix

5.6.1 Method

In the previous tasks, the process noise covariance matrix $\mathbf{Q}$ given in the problem description has been used. The original $\mathbf{Q}$ is given in eq. (66).

$$\mathbf{Q} = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix} \tag{66}$$

To see the effect of the process noise covariance matrix, the simulations from section 5.4 and section 5.5 are simulated using two new matrices, $\mathbf{Q}_{increased}$ and $\mathbf{Q}_{reduced}$ given in eq. (67). The elements in the $\mathbf{w} = [w_w \ w_b]^T$ vector are assumed uncorrelated, which means that the wave and current disturbance do not affect each other. This implies that the $\mathbf{Q}$ matrix is a diagonal matrix, with diagonal elements representing the variance of the wave and current disturbance respectively.

$$\mathbf{Q}_{increased} = \begin{bmatrix} 3000 & 0 \\ 0 & 10^{-4} \end{bmatrix} \tag{67a}$$

$$\mathbf{Q}_{reduced} = \begin{bmatrix} 0.3 & 0 \\ 0 & 10^{-8} \end{bmatrix} \tag{67b}$$

5.6.2 Results

The heading, rudder angle and bias from doing the same simulation as in section 5.4 using $\mathbf{Q}_{increased}$ are plotted in fig. 27 and fig. 28.

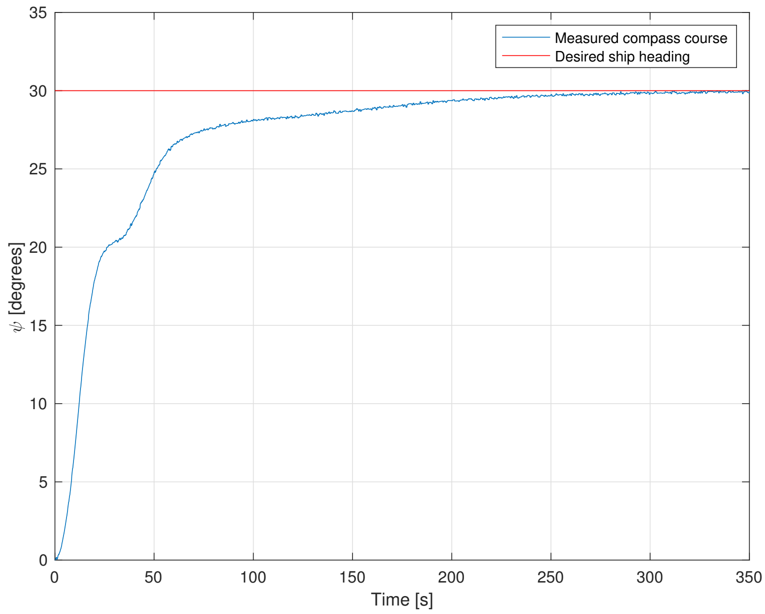

Figure 27: Desired heading vs. actual heading, Q increased.

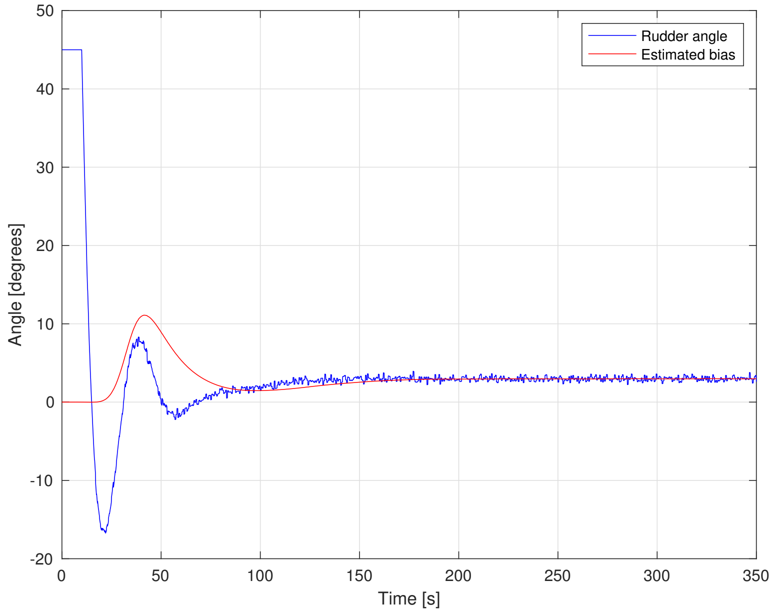

Figure 28: Estimated bias vs. rudder input, Q increased.

The heading, rudder angle and bias from doing the same simulation as in section 5.4 using $\mathbf{Q}_{reduced}$ are plotted in fig. 29 and fig. 30.

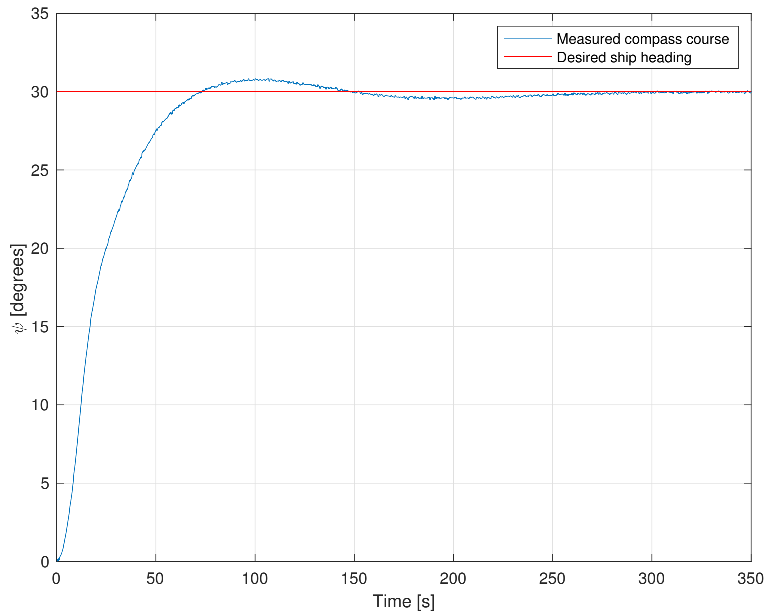

Figure 29: Desired heading vs. actual heading, Q decreased.

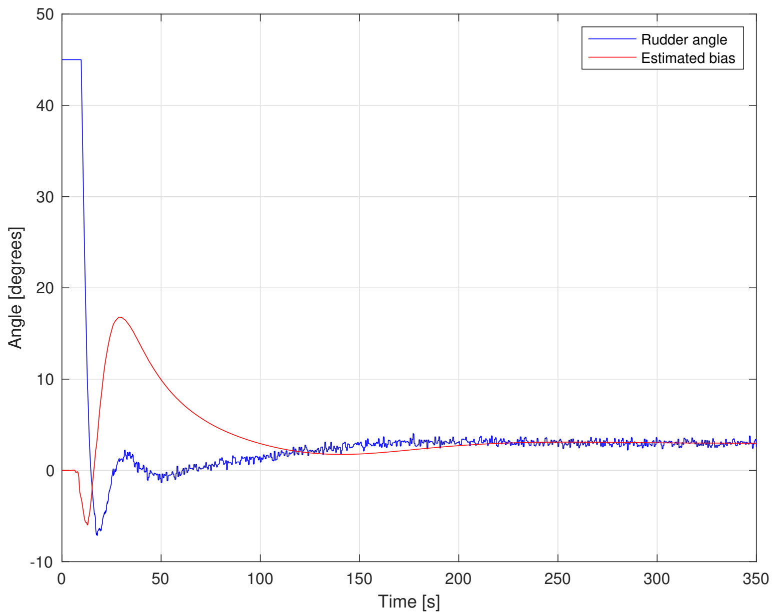

Figure 30: Estimated bias vs. rudder input, Q decreased.

The heading, rudder angle and bias from doing the same simulation as in section 5.5 using $\mathbf{Q}_{increased}$ are plotted in fig. 31 and fig. 32.

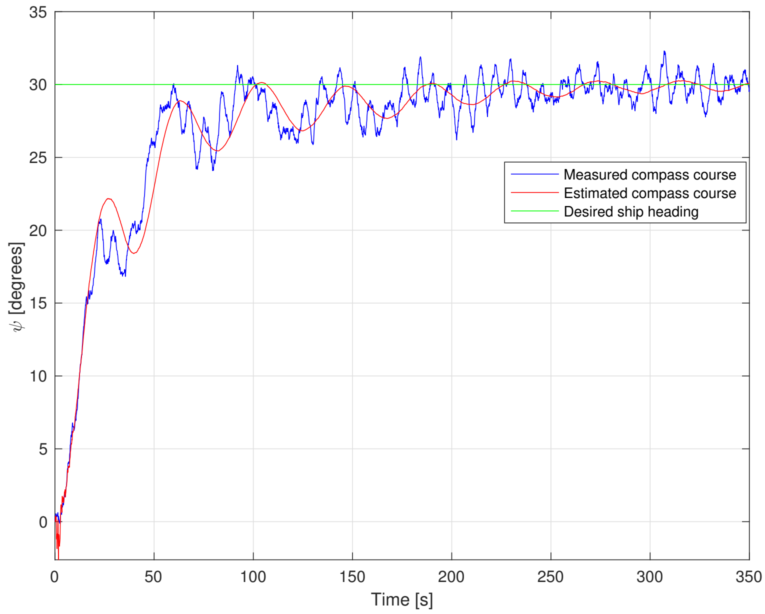

Figure 31: Desired heading vs. actual heading with current and waves, Q increased.

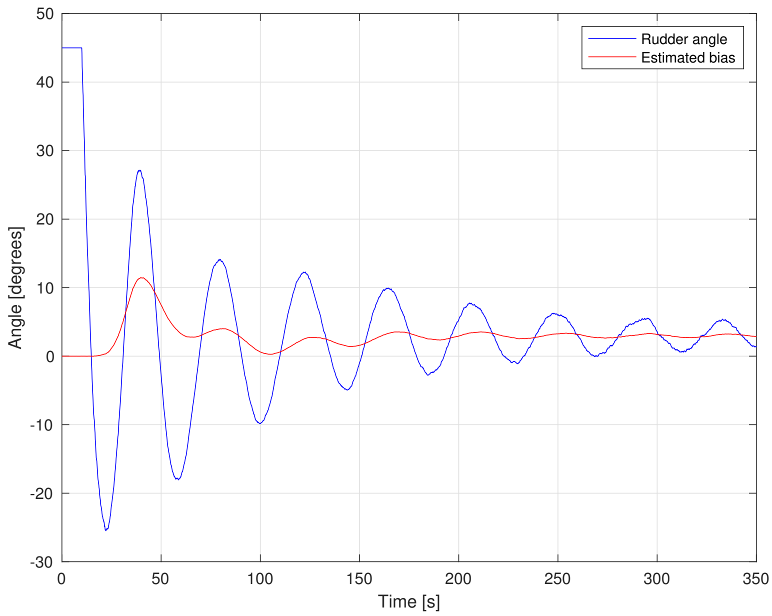

Figure 32: Estimated bias vs. rudder input with current and waves, Q increased.

The heading, rudder angle and bias from doing the same simulation as in section 5.5 using $\mathbf{Q}_{decreased}$ are plotted in fig. 33 and fig. 33.

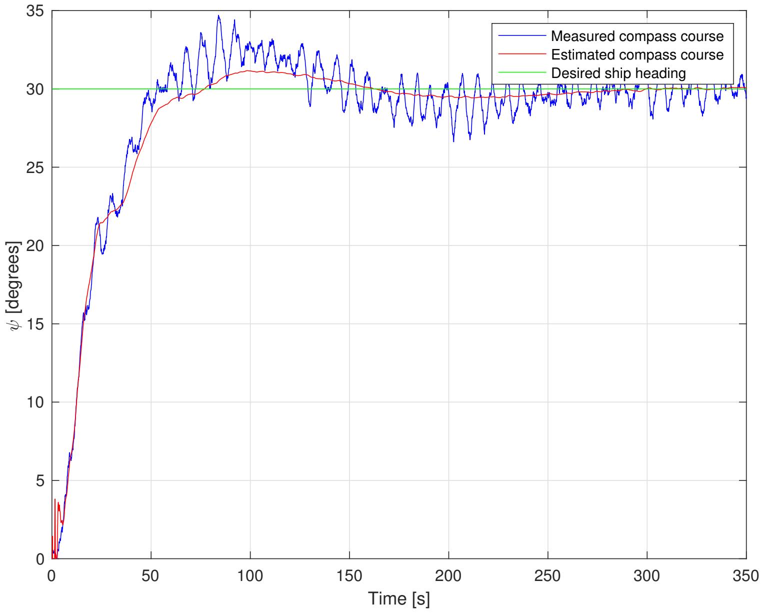

Figure 33: Desired heading vs. actual heading with current and waves, Q reduced.

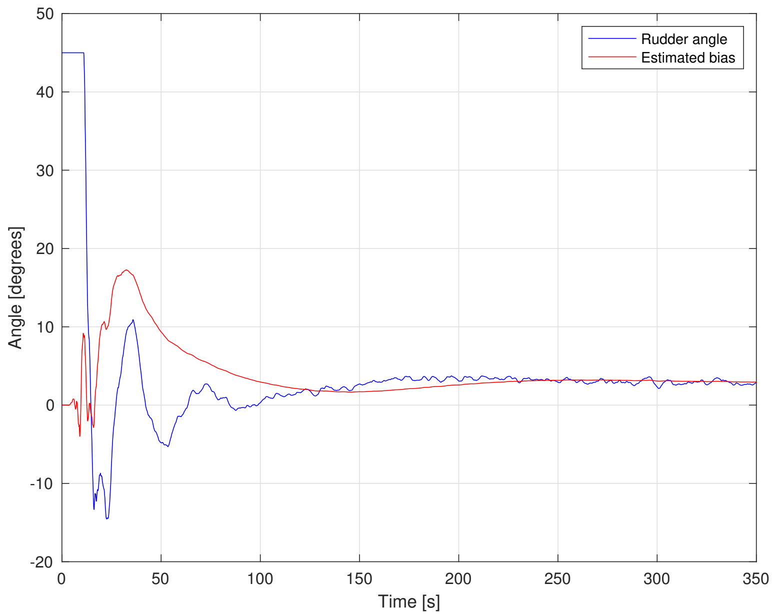

Figure 34: Estimated bias vs. rudder input with current and waves, Q reduced.

5.6.3 Discussion

The effects of changing the values of the process noise covariance matrix for the system simulated exposed to current, as in section 5.4, are discussed in the following. The response of the system and the rudder input signal with $\mathbf{Q}_{increased}$ are shown in fig. 27 and fig. 28 respectively. From fig. 27 one can see that the response is still relatively fast compared to the response with the original disturbance covariance matrix $\mathbf{Q}$, but that it takes a longer time for the steady state error to be removed. Figure 28 shows that the bias estimate is significantly smaller than the one in fig. 22. This can be explained by the fact that the $\mathbf{Q}$ quantifies the uncertainty in the process model, which can be seen in eq. (58) and eq. (63). Since the model becomes more uncertain, the Kalman filter will base its estimate more on the measurement than on the a priori estimate produced by the process model. Since the measurement does not include the bias, the bias estimate will not change that rapidly, and thus uses more time to converge.

The response of the system and the rudder input signal when the disturbance covariance matrix is set equal to $\mathbf{Q}_{decreased}$ can be seen in fig. 29 and fig. 30 respectively. The response is slightly faster than the response with the original disturbance covariance matrix, $\mathbf{Q}$, which can be seen in fig. 21.

The response does, however, overshoot the reference and is more oscillatory. This can be explained by looking at the bias in fig. 30, which changes faster and is slightly larger than the bias with the original disturbance covariance matrix, $\mathbf{Q}$. This is due to the Kalman filter basing more of its estimate on the a priori estimate due to the decrease in the model uncertainty.

The effects of changing the values of the process noise covariance matrix for the system simulated exposed to waves and current, as in section 5.5, are discussed in the following. The response of the system and the rudder input signal from the simulation with $\mathbf{Q}_{increased}$ can be seen in fig. 31 and fig. 32 respectively. Comparing the response to the response with the original disturbance covariance matrix $\mathbf{Q}$, seen in fig. 24, one can see that response is slightly slower and has more excessive oscillations. However, the response does not overshoot, but instead moves slowly towards the reference. The slightly slower response can be due to the bias estimate, which does not change that rapidly and thus do not become as large as in the original simulation, seen in fig. 25. As discussed earlier this is due to the estimate being based more on the measurements than the a priori estimates created with the process model. The oscillations can be due to the error in the wave disturbance estimate, causing the rudder to over- and under-compensate for the other ship motions, as discussed in section 5.5.3.

The response of the system and the rudder input signal from the simulation with $\mathbf{Q}_{decreased}$ can be seen in fig. 33 and fig. 34 respectively. Comparing the the response to the response of the system with the original disturbance covariance matrix, $\mathbf{Q}$, one can see that the response is slightly faster and less oscillatory. Figure 34 reveals that the initial changes in the bias estimate are quite volatile and that the peak value of the bias estimate is larger than the in the original simulation. As discussed earlier this is most likely due to the Kalman filter basing more of its estimate on the a priori estimate due to the reduced uncertainty in the model. This might also be the reason why the response overshoots the reference. The less oscillatory response might be due to the wave disturbance estimate converging faster towards the measured wave disturbances, such that the rudder compensates for the slowly-varying motions. However, one can still see that there are still some high frequency components in the rudder input signal, similarly to the signal in fig. 25, indicating that the wave filtering is not perfect.

The results that gave the best response of the ship, was the one using the process noise covariance matrix given in the assignment, both for the simulation with only current and the one exposed to both waves and current.

6 Conclusion

An autopilot for a ship has been designed using a PD controller. The ship was simulated exposed to measurement noise, wave and current disturbances, and the response of the ship was evaluated. At first, the controller computed the rudder angle using the measured heading in a feed back loop. The ship managed to reach the desired heading when there were no waves, but when it was exposed to current alone the ship did not reach the desired ship heading, as there was a steady state error between the reference and the heading. When the system was exposed to waves, the heading and rudder angle had a much more fluctuating behavior. To avoid these unwanted effects caused by the disturbances, a Kalman filter was included in the system. The filter estimated the bias generated by the current, which was used to cancel out the bias in the rudder angle. This removed the steady state error between the heading and the heading reference. To filter out the wave component in the measured heading, the estimated heading without the wave component was fed back to the controller. This reduced the fluctuating behavior of the rudder angle that had been observed without the wave filtering.

A MATLAB Code

A.1 Task1_init.m

```

%% Task 5.1
amp = 1;
omega_1 = 0.005; %[rad/s]
omega_2 = 0.05; %[rad/s]

%5.1d
%constants for transferfunction H(s)
K = 0.154;
T = 66.15;
```

A.2 task2_init.m

```
%% Compass wave distrubance power density spectrum

clear all;
close all;
clc;

load('wave.mat');

% Numerical estimate of the PSD with Welch's method
window_size = 4096;
sampling_freq = 10;
[S_freq, freqs] = pwelch(deg2rad(psi_w(2,:)), window_size, ...
                         [], [], sampling_freq);
S_freq = S_freq';
freqs = freqs';

% Convert to circular frequency
S_omega = S_freq ./ (2*pi);
omegas = 2*pi * freqs;

% Find spectral peak frequency and intensity
[sigma_squared, peak_index] = max(S_omega);
omega_0 = omegas(peak_index);
sigma = sqrt(sigma_squared);

% Curve fit analytical PSD to numerical estimate
lambda_0 = 0.10; % Initial guess
fun = @(lambda, omegas) ...
       (2 * lambda * omega_0 *  omegas * sigma).^2 ./ ...
       ( (omega_0^2 - omegas.^2).^2 ...
         + (2 * lambda * omega_0 * omegas).^2 );

lambda = lsqcurvefit(fun, lambda_0, omegas, S_omega);

% Calculate PSD from analytical expression
K_w = 2 * lambda * omega_0 * sigma;
P_omega = (K_w * omegas).^2 ./ ...
          ( (omega_0^2 - omegas.^2).^2 + ...
          (2 * lambda * omega_0 * omegas).^2 );
```

A.3 task2_plotting.m

```
%% Plotting

x_lim = 2;

% Numerically estimated PSD
figure
plot(omegas(1,:), S_omega(1,:))
legend('S_{\psi_{w}}(\omega)')
title('Power spectral density function')
xlabel('Frequency \omega [rad/s]')
ylabel('Spectral intensity [Ws/rad]')
xlim([0 x_lim])

% Numerically estimated PSD with illustration of omega_0 and
% sigma_squared
figure
hold on
plot(omegas(1,:), S_omega(1,:))
plot(omega_0, sigma_squared, 'r.', 'MarkerSize', 13)
line([omega_0 omega_0], [0 sigma_squared], 'Color', 'r')
legend('S_{\psi_{w}}(\omega)')
title('Power spectral density function')
xlabel('Frequency \omega [rad/s]')
ylabel('Spectral intensity [Ws/rad]')
xlim([0 x_lim])

% Numerically estimated PSD and curve fitted analytical PSD
figure
plot(omegas(1,:), S_omega(1,:), ...
     omegas(1,:), P_omega(1,:))
analytical_legend = strcat('P_{\psi_{w}}(\omega), \lambda = ', ...
                          num2str(lambda, 2));
legend('S_{\psi_{w}}(\omega)', ...
        analytical_legend)
title('Power spectral density function')
xlabel('Frequency \omega [rad/s]')
ylabel('Spectral intensity [Ws/rad]')
xlim([0 x_lim])
hold off
```

A.4 Task3_init.m

```
%% PD controller
% Process parameters
T = 66.147;
K = 0.154;
T_d = T;

% Controller parameters
w_c = 0.1;
T_f = -(tan((50+90-180)*(pi/180)))/(w_c);
K_pd = (w_c*sqrt(1 + (T_f^2)*w_c^2))/K;

heading_ref = 30;


% Bode plot of open loop transfer function
% Comment out when making Bode plot is not needed
num = [0 K_pd*K];
den = [T_f 1 0];

H = tf(num, den);

bode(H);
```

A.5 plotresults3.m

```
load('task53b.mat');
figure

plot(task53b(1,:), task53b(2,:), 'b', task53b(1,:), ...
    task53b(4,:), 'r');
grid on;
legend('Ship heading', 'Desired ship heading')
xlabel('Time [s]');
ylabel('\psi [degrees]');
ylim([0, 35]);

figure

plot(task53b(1,:), task53b(3,:), 'b');
grid on;
legend('Rudder angle');
xlabel('Time [s]');
ylabel('Angle [degrees]');
```

A.6 task5.m

```
% Process parameters
T = 66.147;
K = 0.154;
T_d = T;

% Controller parameters
w_c = 0.1;
T_f = -(tan((50+90-180)*(pi/180)))/(w_c);
K_pd = (w_c*sqrt(1 + (T_f^2)*w_c^2))/K;

heading_ref = 30;

% Wave disturbance parameters
w0 = 0.7823;
lambda = 0.083;
sigma = sqrt(7.9191*10^-4); %sqrt(2.5977);
Kw = 2*lambda*w0*sigma;
sample_time = 0.1;

% State space model

A = [0, 1, 0, 0, 0;
     -w0^2, -2*lambda*w0, 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;
     Kw, 0;
     0, 0;
     0, 0;
     0, 1];

C = [0, 1, 1, 0, 0];

% Discrete state space model

[A_bar, B_bar] = c2d(A, B, sample_time);

[A_bar, E_bar] = c2d(A, E, sample_time);
```

```
% Kalman parameters

variance = 0.002*(pi/180)^2;

Q = [30, 0;
     0, 10^(-4)];

Q_bar = E_bar*Q*E_bar';
```

A.7 Kalman filter - Matlab function

```
function [psi_hat_w, psi_hat, b_hat] = discrete_kalman(y, u, ...
                                        variance, sample_time, ...
                                        A_bar, B_bar, Q_bar)

C = [0, 1, 1, 0, 0];

persistent init_flag;
persistent x_hat_bar;
persistent P_bar;

R = variance/sample_time;

if isempty(init_flag)
    init_flag = 1;

    %Initial conditions:
    x_hat_bar_0 = [0; 0; 0; 0; 0];
    P_bar_0 = diag([1, 0.013, pi^2, 1, 2.5*10^-3]);

    x_hat_bar = x_hat_bar_0;
    P_bar = P_bar_0;
end

% Compute Kalman gain

L = P_bar*C'*inv((C*P_bar*C' + R));

% Update estimate with measurement

x_hat = x_hat_bar + L*(y - C*x_hat_bar);

% Update error covariance matrix

P = (eye(5) - L*C)*P_bar*(eye(5) - L*C)' + L*R*L';

% Project ahead

x_hat_bar = A_bar*x_hat + B_bar*u;

P_bar = A_bar*P*A_bar' + Q_bar;
```

```
psi_hat_w = rad2deg(x_hat(2));
psi_hat = rad2deg(x_hat(3));
b_hat = rad2deg(x_hat(5));
```

A.8 plotresults55d.m

```
load('task55.mat');
load('task55est.mat');
figure

plot(task55b(1,:), task55b(2,:), task55b(1,:), ...
    task55b(4,:), 'r');
grid on;
legend('Measured compass course', 'Desired ship heading')
xlabel('Time [s]');
ylabel('\psi [degrees]');
ylim([0, 35]);

figure

plot(task55b(1,:), task55b(3,:), 'b', task55best(1,:), ...
    task55best(2,:), 'r');
grid on;
legend('Rudder angle', 'Estimated bias');
xlabel('Time [s]');
ylabel('Angle [degrees]');
```

A.9 plotresults55e.m

```
load('task55.mat');
load('task55est.mat');
figure

plot(task55b(1,:), task55b(2,:), 'b', task55best(1,:), ...
    task55best(3,:), 'r', task55b(1,:), task55b(4,:), 'g');
grid on;
legend('Measured compass course', ...
    'Estimated compass course', 'Desired ship heading');
xlabel('Time [s]');
ylabel('\psi [degrees]');
ylim([0, 35]);

figure

plot(task55b(1,:), task55b(3,:), 'b', task55best(1,:), ...
    task55best(2,:), 'r');
grid on;
legend('Rudder angle', 'Estimated bias');
xlabel('Time [s]');
ylabel('Angle [degrees]');

figure
plot(task55W(1,:), task55W(2,:), 'b', task55best(1,:), ...
    task55best(4,:), 'r');
grid on;
legend('Actual wave influence', 'Estimated wave influence');
xlabel('Time [s]');
ylabel('Angle [degrees]');
```

References

[1] Robert G. Brown. *Introduction to Random Signlas and Applied Kalman Filtering, Fourth Edition*. John Wiley Sons, Inc., 2012.

[2] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Incorporated, 2014.

[3] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control, First Edition*. John Wiley & Sons Ltd., 2011.

[4] MathWorks. *c2d*. `https://se.mathworks.com/help/control/ref/c2d.html`. Accessed: 2018-11-18.

[5] MathWorks. *lsqcurvefit*. `https://se.mathworks.com/help/optim/ug/lsqcurvefit.html`. Accessed: 2018-10-19.