Linear Systems TTK4115 – Boat lab
Group 71

Martin Tysseland – 759474
Ernst Torsgård – 759517

Autumn 2016

NTNU – Trondheim
Norwegian University of
Science and Technology

Contents

5 Assignment **1**

5.1 Identification of the boat parameters 1

a) 1

b) 1

c) 4

d) 7

5.2 Identification of wave spectrum model 9

a) 9

b) 9

c) 10

d) 11

5.3 Control system design 13

a) 13

b) 14

c) 15

d) 16

5.4 Observability 18

a) 18

b) 19

c) 19

d) 20

e) 21

5.5 Discrete Kalman filter 22

a) 22

b) 22

c) 23

d) 24

e) 25

References **28**

Matlab scripts **29**

Simulink models **38**

5 Assignment

5.1 Identification of the boat parameters

The Matlab code for this task is shown i Listing 1, whilst the Simulink models are shown in Figure 15 and 16.

a)

From the description, the relationship between the rudder angle δ and the heading ψ are described by equation 13c and equation 13d. Then we have

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

Assuming that there are no disturbances, i.e. b = 0, Laplace transformation with zero initial conditions yields

$$\begin{aligned}
s^2\psi &= -\frac{1}{T}s\psi + \frac{K}{T}\delta \\
\Rightarrow \psi(s^2 + \frac{1}{T}s) &= \frac{K}{T}\delta
\end{aligned}$$

Hence we obtain the transfer function

$$H_{ship}(s) = \frac{\psi}{\delta}(s) = \frac{K}{Ts^2 + s}$$

b)

We want to identify the boat parameters K and T. Turning off all the disturbances and the measurement noise, and applying sine inputs to the system gives the following plots of the ship heading

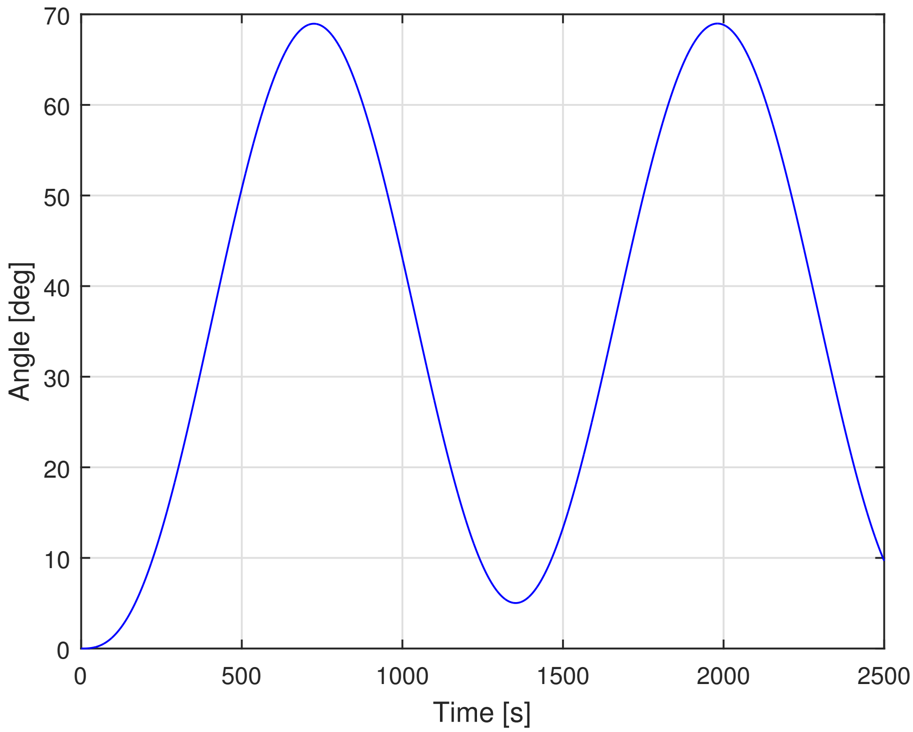

Figure 1: Ship heading with a sine input with amplitude 1 and frequency $\omega_1 = 0.005$

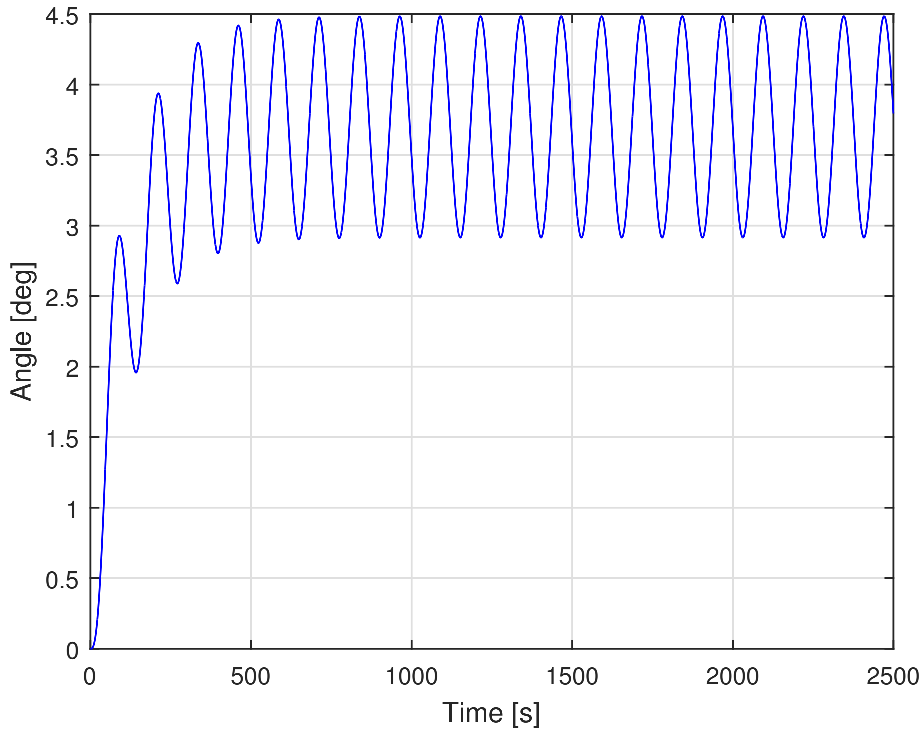

Figure 2: Ship heading with a sine input with amplitude 1 and frequency $\omega_2 = 0.05$

By the matlab commands $max()$ and $min()$, we find the amplitudes to be

$$\left|H(j\omega_1)\right| \approx \frac{68.98 - 5.02}{2} = 31.98$$
$$\left|H(j\omega_2)\right| \approx \frac{4.48 - 2.92}{2} = 0.78$$

From this we get two equations with two unknowns

$$
\begin{aligned}
&\frac{K}{\sqrt{T^2\omega_1^4 + \omega_1^2}} = 31.98 \\
&\Rightarrow K = 31.98^2(T^2\omega_1^4 + \omega_1^2) \\
&\frac{K}{\sqrt{T^2\omega_2^4 + \omega_2^2}} = 0.78 \\
&\Rightarrow K = 0.78^2(T^2\omega_2^4 + \omega_2^2)
\end{aligned}
$$

where $\omega_1 = 0.005$ and $\omega_2 = 0.05$. Solving for the two unknowns yields $K \approx 0.175$ and $T \approx 88.697$.

c)

Now we turn on both wave disturbance and measurement noise, and apply the same sine inputs as in part b.

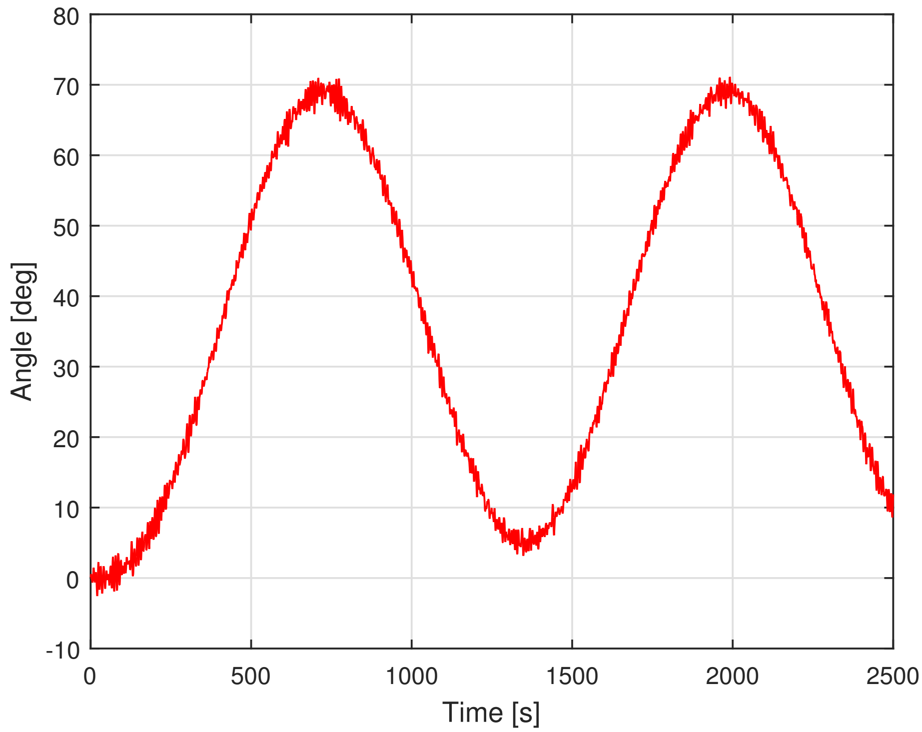

Figure 3: Ship heading with a sine input with amplitude 1 and frequency $\omega_1 = 0.005$

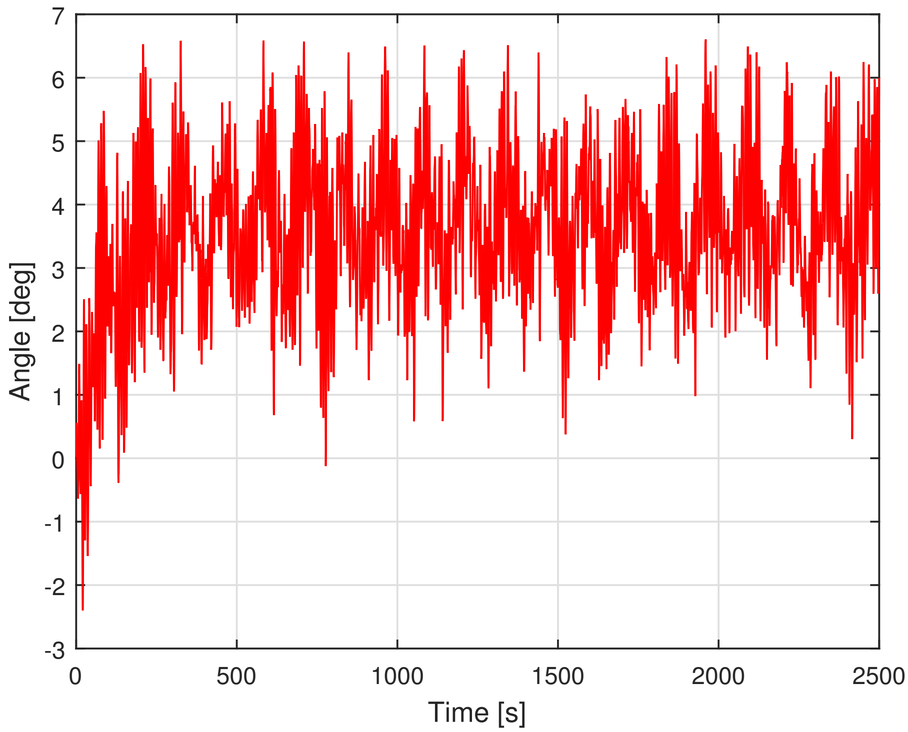

Figure 4: Ship heading with a sine input with amplitude 1 and frequency $\omega_2 = 0.05$

We see from the plots that you could get reasonable estimates of the parameters with the sine input with frequency $\omega_1 = 0.005$ because the amplitude of the heading is a lot greater than the amplitude of the disturbance. However, with the sine input with frequency $\omega_2 = 0.05$ it is difficult to get good estimates because the amplitude of the heading is closer to the amplitude of the disturbance, making the combined signal very distorted. Hence it is not possible to get good estimates of the boat parameters in rough weather conditions.

d)

A step input of 1 degree on the rudder at t = 0 gives the following ship response

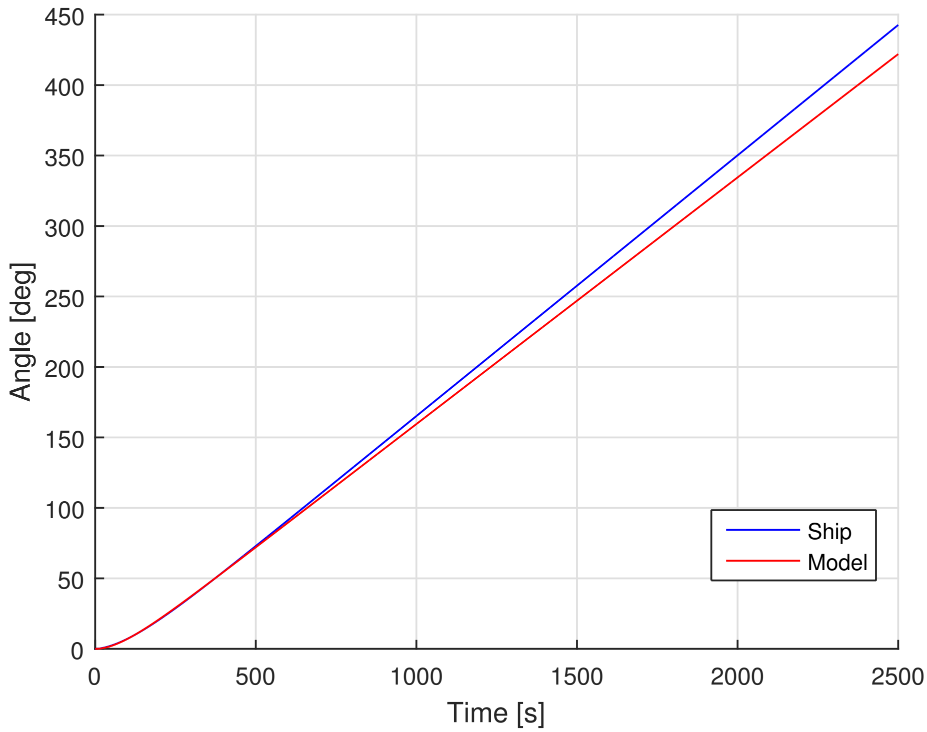

Figure 5: Response of ship (blue) vs. model (red) with a step input of 1 degree

We see that the model is a very god approximation in a certain time interval, but it looks like the deviation increases with time. The reason could be that the model has a smaller derivative than the ship, so increasing the K with 0.0075 results in

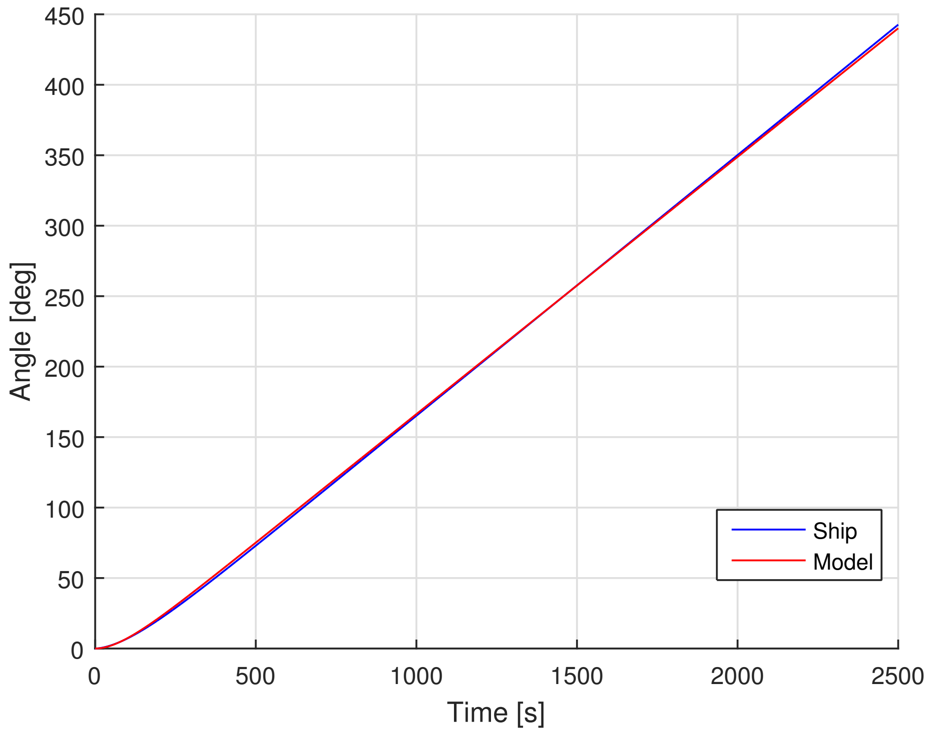

Figure 6: Response of ship (blue) vs. model (red) with a step input of 1 degree and $K = 0.1825$

Now we see that the model is a better approximation for a wider time interval.

In the rest of the problems the measurement noise is turned on.

5.2 Identification of wave spectrum model

The Matlab code for this task is shown i Listing 2.

a)

We use the Matlab function $[pxx, f] = pwelch(x, window, noverlap, f, fs)$ to calculate an estimate of the Power Spectral Density function of ψ_ω, $S_{\psi_\omega}(\omega)$. This function returns the two-sided Welch PSD estimates at the frequencies specified in the vector f. The frequencies in f are in cycles per unit time. The sampling frequency fs is the number of samples per unit time.

The estimate is shown in Figure 7.

b)

From the description, the relationship between the the white noise process ω_ω and ξ_ω is described by the equation 13a and equation 13b. Thus we have

$$\begin{aligned}\dot{\xi}_\omega &= \psi_\omega \\ \dot{\psi}_\omega &= -\omega_0^2\xi_\omega - 2\lambda\omega_0\psi_\omega + K_\omega\omega_\omega\end{aligned}$$

Laplace transformation with zero initial conditions yields

$$\begin{aligned}s\xi_\omega &= \psi_\omega \\ s\psi_\omega &= -\omega_0^2\xi_\omega - 2\lambda\omega_0\psi_\omega + K_\omega\omega_\omega \\ \Rightarrow s\psi_\omega &= -\omega_0^2\frac{\psi_\omega}{s} - 2\lambda\omega_0\psi_\omega + K_\omega\omega_\omega\end{aligned}$$

Hence we obtain the transfer function

$$H_{wave}(s) = \frac{\psi_\omega}{\omega_\omega}(s) = \frac{sK_\omega}{s^2 + 2s\lambda\omega_0 + \omega_0^2}$$

Next we find the Power Spectral Density function of ψ_ω, $P_{\psi_\omega}(j\omega)$. ψ_ω is a high-frequency component due to the wave disturbance ω_ω, where ω_ω is a zero mean white noise process with unity variance. From [1, p. 106-107], we have

$$P_{\psi_\omega}(j\omega) = H_{wave}(j\omega)H_{wave}(-j\omega)S_{\omega_\omega}(jw) = \left|H_{wave}(j\omega)\right|^2 S_{\omega_\omega}(jw)$$

where $S_{\omega_\omega}(jw)$ is the spectral density function of the white noise.

White noise is defined to be a stationary random process having a constant spectral density function [1, p. 75]. Denoting the amplitude as A yields

$$S_{\omega_\omega}(jw) = A$$

Thus the corresponding autocorrelation function of the white noise is

$$R_{\omega_\omega}(\tau) = \mathcal{F}^{-1}(S_{\omega_\omega}(jw)) = A\delta(\tau)$$

At $\tau = 0$ the autocorrelation function is the mean square value of the white noise process, i.e. 1, since the white noise process has unity variance and zero mean. This implies that $A = 1$ and therefore $S_{\omega_\omega}(jw) = 1$. Then we have

$$\begin{aligned} P_{\psi_\omega}(j\omega) = \left|H_{wave}(j\omega)\right|^2 S_{\omega_\omega}(jw) &= \frac{\left|jK_\omega\omega\right|^2}{\left|\omega_0^2 - \omega^2 + 2j\omega\omega_0\lambda\right|^2} \\ &= \frac{(K_\omega\omega)^2}{(\omega_0^2 - \omega^2)^2 + (2\omega\omega_0\lambda)^2} \\ &= \frac{K_\omega^2\omega^2}{\omega^4 + \omega_0^4 + 2\omega^2\omega_0^2(2\lambda - 1)} \end{aligned}$$

c)

Plotting the estimatet Power Spectral Density function from part a yields

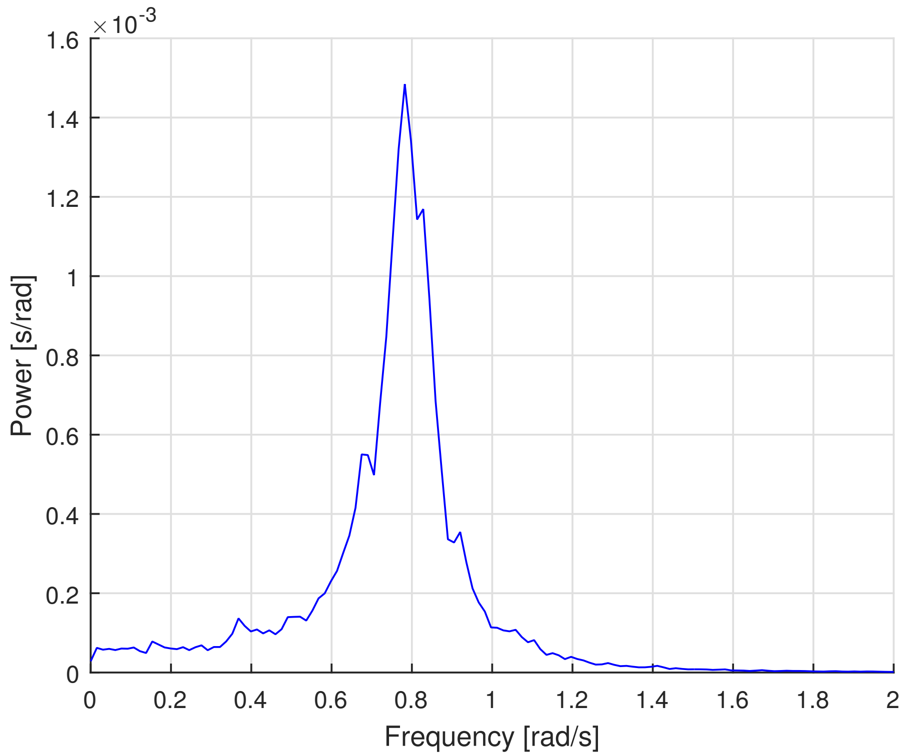

Figure 7: Estimate of the Power Spectral Density (PSD) function, $S_{\psi_\omega}(\omega)$

The resonance frequency ω_0 is the frequency where the amplitude is at its maximum. By the Matlab function $max()$, we find that $\omega_0 \approx 0.78$, which could also be seen from Figure 7.

d)

Now we need to identify the damping factor to have a complete model of the wave response. We define $K_\omega = 2\lambda\omega_0\sigma^2$, where σ^2 is the peak value of P_{ψ_ω}. Again we use the Matlab command $max()$, and we find the peak of the estimate to be approximately 0.015, so $\sigma^2 \approx 0.015$. This could also be seen from Figure 7. Substituting $K_\omega = 2\lambda\omega_0\sigma$ into $P_{\psi_\omega}(\omega)$ gives

$$P_{\psi_\omega}(\omega) = \frac{(2\lambda\omega_0\sigma\omega)^2}{\omega^4 + \omega_0^4 + 2\omega^2\omega_0^2(2\lambda - 1)}$$

Plotting this for different values of λ and comparing the result with the estimate $S_{\psi_\omega}(\omega)$, we see that the curve fits well with $\lambda \approx 0.09$. See Figure 8.

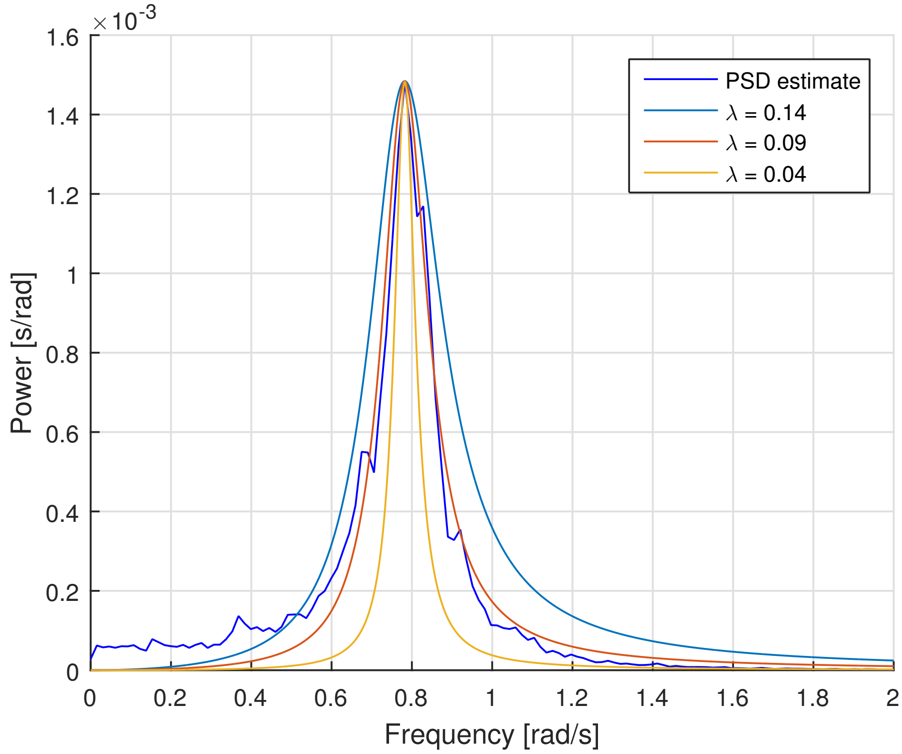

Figure 8: S_{ψ_ω} (blue) vs. P_{ψ_ω} with different values of λ

The curve with $\lambda = 0.09$ is very close to the estimate for frequencies above ≈ 0.7 rad/s, which is desirable, since it is above these frequencies most of the spectral content is concentrated.

5.3 Control system design

The Matlab code for this task is shown i Listing 3, whilst the Simulink model is shown in Figure 17.

a)

We want to design an autopilot for the ship. To do this, we design a PD controller based on the transfer function from the rudder angle δ to the ship heading ψ without disturbances.

The open loop system is

$$H_{sys}(s) = H_{pd}(s)H_{ship}(s) = \frac{KK_{pd}}{s}\frac{1+T_d s}{1+T_f s}\frac{1}{1+Ts}$$

To cancel out the time constant T of the transfer function $H_{ship}(s)$, we choose $T_d = T$. Hence we obtain

$$H_{sys}(s) = \frac{KK_{pd}}{s}\frac{1}{1+T_f s}$$

Next we need to find the regulator parameters T_f and K_{pd}. With a cutoff frequency of 0.1 and a phase margin of 50°, we solve the following equation to find a value for T_f

$$\tan(50) = \frac{\Im H_{sys}(j\omega_c)}{\Re H_{sys}(j\omega_c)}$$

Separating the transfer function $H_{sys}(j\omega)$ into a real and imaginary part and ultimately solving for T_f yields

$$
\begin{aligned}
H_{sys}(j\omega_c) &= -j\frac{KK_{pd}}{w}\frac{1-T_f j\omega_c}{1+T_f^2\omega_c^2} \\
&= -\frac{KK_{pd}T_f\omega_c}{\omega_c(1+T_f^2\omega_c^2)} - j\frac{KK_{pd}}{\omega_c(1+T_f^2\omega_c^2)} \\
\Rightarrow \tan(50) &= \frac{\frac{KK_{pd}}{\omega_c(1+T_f^2\omega_c^2)}}{\frac{KK_{pd}T_f\omega_c}{\omega_c(1+T_f^2\omega_c^2)}} = \frac{1}{T_f\omega_c} \\
\Rightarrow T_f &= \frac{1}{\tan(50)\omega_c}
\end{aligned}
$$

The amplitude of the transfer function at the cutoff frequency ω_c is 1 dB. Hence we find K_{pd} by solving

$$
\left|H_{sys}(j\omega_c)\right| = 1
$$

$$
\begin{aligned}
\left|H_{sys}(j\omega_c)\right| &= \frac{KK_{pd}}{\omega_c}\frac{1}{\left|1+jT_f\omega_c\right|} \\
&= \frac{KK_{pd}}{\omega_c\sqrt{1+T_f^2\omega_c^2}} = 1 \\
\Rightarrow K_{pd} &= \frac{\omega_c\sqrt{1+T_f^2\omega_c^2}}{K}
\end{aligned}
$$

From this we obtain $T_f \approx 8.391$ and $K_{pd} \approx 0.746$.

b)

Simulation of the system with no disturbances gives

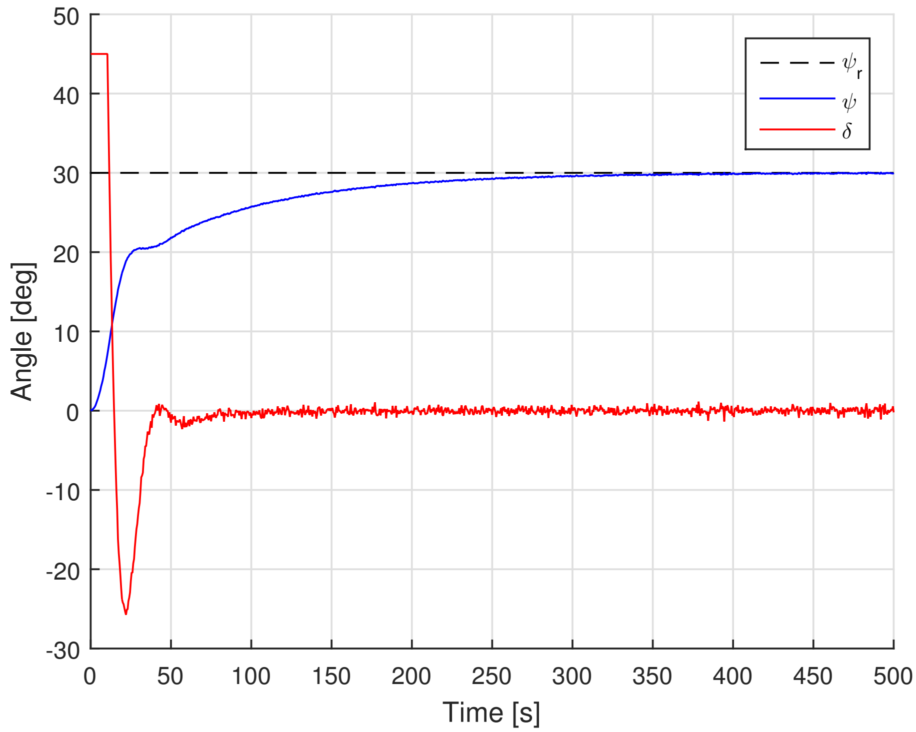

Figure 9: Simulation of the system with no disturbances

From Figure 9, we see that the autopilot works. The heading of the ship reaches and stays at the reference angle with no deviation. The oscillations in the rudder angle is due to the measurement noise, since it is trying to compensate for the errors in the measured heading. To avoid the high-frequency noise components, a low pass filter could be implemented in the measurement process.

c)

Simulation of the system with current disturbance gives

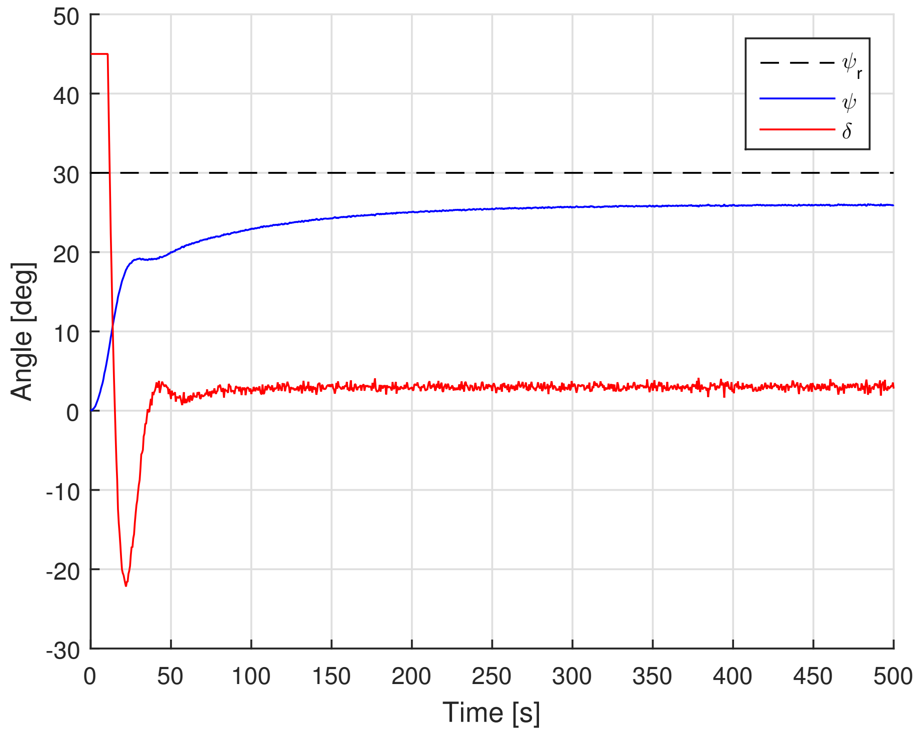

Figure 10: Simulation of the system with current disturbance

From Figure 10, we see that the ship cannot reach the reference angle. The current disturbance results in stationary deviation. This is because we use a PD regulator, which is unable to eliminate stationary deviation. The autopilot does not work.

d)

Simulation of the system with wave disturbance gives

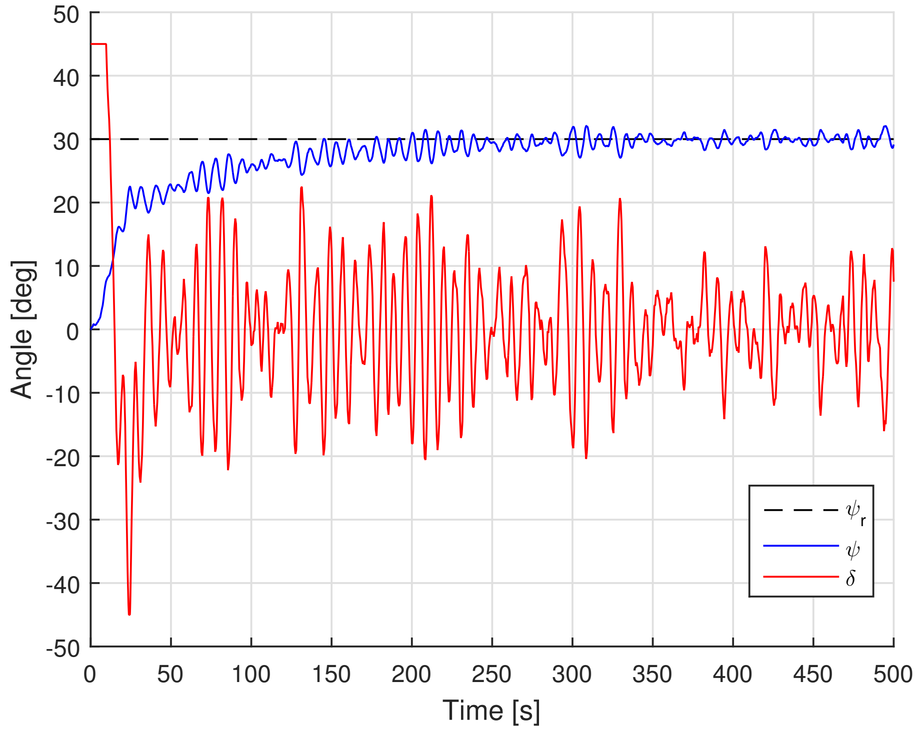

Figure 11: Simulation of the system with wave disturbance

From Figure 11, we see that the wave disturbance gives rise to oscillations in both heading and rudder, the latter being extreme. The heading will oscillate around the reference angle, that is, the autopilot will keep the ship on the desired angle with rather small variance. However, the the rudder is subject to extreme oscillations because it is the derivative of the oscillating heading. This is an undesirable response, as it will eventually wear down the rudder, and also increase both distance covered and travel time of the ship. Regarding only the heading of the ship, the autopilot works, but considering the rudder response as well, the system does not work satisfactory.

5.4 Observability

The Matlab code for this task is shown i Listing 4.

a)

With $u = \delta$ and the given state- and distrubance vector

$$\mathbf{x} = \begin{bmatrix} \xi_\omega \\ \psi_\omega \\ \psi \\ r \\ b \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} \omega_\omega \\ \omega_b \end{bmatrix}$$

we have

$$\begin{aligned}
\dot{x_1} &= \dot{\xi}_\omega = x_2 \\
\dot{x_2} &= \dot{\psi}_\omega = -\omega_0^2 x_1 - 2\lambda\omega_0 x_2 + K_\omega \omega_\omega \\
\dot{x_3} &= \dot{\psi} = x_4 \\
\dot{x_4} &= \dot{r} = -\frac{1}{T} x_4 + \frac{K}{T}(\delta - x_5) \\
\dot{x_5} &= \dot{b} = \omega_b \\
y &= x_2 + x_3 + v
\end{aligned}$$

Then we can transform the system into the a state-space formulation of the form

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w} \\
y &= \mathbf{C}\mathbf{x} + v
\end{aligned}$$

with matrices

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

$$\mathbf{E} = \begin{bmatrix} 0 & 0 \\ K_\omega & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

b)

With no disturbances, i.e. disregarding ξ_ω, ψ_ω and b, the state vector becomes $\mathbf{x} = [\psi \quad r]^T$ and our $\mathbf{A}$ and $\mathbf{C}$ matrices reduces to

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

We find the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}$$

by the Matlab command $obsv(\mathbf{A}, \mathbf{C})$. The observability matrix has full rank, i.e. $rank(\mathcal{O}) = 2$, and the system is observable. From the C matrix, we see that ψ is observable directly through the output, whereas r is observable through the derivative of the output, since $\dot{\psi} = r$.

c)

With only current disturbance, i.e. disregarding ξ_ω and ψ_ω, the state vector becomes $\mathbf{x} = [\psi \quad r \quad b]^T$ and our $\mathbf{A}$ and $\mathbf{C}$ matrices reduces to

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

We find the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix}$$

by the Matlab command $obsv(\mathbf{A}, \mathbf{C})$. The observability matrix has full rank, i.e. $rank(\mathcal{O}) = 3$, and the system is observable. We see that b is observable through the double derivative of the output, since $\dot{\psi} = r$ and $\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - \underline{b})$.

d)

With only wave disturbance, i.e. disregarding b, the state vector becomes $\mathbf{x} = [\xi_\omega \quad \psi_\omega \quad \psi \quad r]^T$ and our $\mathbf{A}$ and $\mathbf{C}$ matrices reduces to

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

We find the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix}$$

by the Matlab command $obsv(\mathbf{A}, \mathbf{C})$. The observability matrix has full rank, i.e. $rank(\mathcal{O}) = 4$, and the system is observable. Here, ψ_ω is observable directly through the output, whereas ξ_ω is observable through the derivative of the output, since $\dot{\psi}_\omega = -\omega_0^2 \underline{\xi_\omega} - 2\lambda\omega_0\psi_\omega + K_\omega\omega_\omega$.

e)

With both wave and current disturbance, the state vector becomes $\mathbf{x} = [\xi_\omega \quad \psi_\omega \quad \psi \quad r \quad b]^T$ and our $\mathbf{A}$ and $\mathbf{C}$ matrices are

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

We find the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \mathbf{CA}^4 \end{bmatrix}$$

by the Matlab command $obsv(\mathbf{A}, \mathbf{C})$. The observability matrix has full rank, i.e. $rank(\mathcal{O}) = 5$, and the system is observable. Here, the connection between observability and the states is a combination of all those above.

5.5 Discrete Kalman filter

The Matlab codes for this task are shown i Listing 5 and 6, whilst the Simulink models are shown in Figure 18, 19, 20 and 21.

a)

We want to implement a discrete Kalman filter to estimate the bias b, the ship heading ψ and the high-frequency wave induced motion on the heading ψ_ω. First we need to discretize our model.

Using the Matlab function $c2d(sys, Ts)$, which discretizes a continuous-time dynamic system model using zero-order hold on the inputs and a sample time of Ts seconds, we obtain a discretized model with matrices

$$\mathbf{A}_d \approx \begin{bmatrix} 0.977 & 0.0992 & 0 & 0 & 0 \\ -0.0607 & 0.983 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0999 & -9.86 \cdot 10^{-6} \\ 0 & 0 & 0 & 0.999 & -1.97 \cdot 10^{-4} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B}_d \approx \begin{bmatrix} 0 \\ 0 \\ 9.86 \cdot 10^{-6} \\ 1.97 \cdot 10^{-4} \\ 0 \end{bmatrix}$$

$$\mathbf{E}_d \approx \begin{bmatrix} 2.70 \cdot 10^{-5} & 0 \\ 5.38 \cdot 10^{-4} & 0 \\ 0 & -3.29 \cdot 10^{-7} \\ 0 & -9.86 \cdot 10^{-6} \\ 0 & 0.1 \end{bmatrix} \qquad \mathbf{C}_d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad D_d = 1$$

b)

We need to know some statistical properties of the noise, as we want the Kalman filter to filter this out from our system. Plotting the compass course with only measurement noise with no input, we find the variance of the noise by using the Matlab command $var()$. Hence we find the variance to be approximately $6.274 \cdot 10^{-7}$ rad.

c)

Given the process noise covariance $\mathbf{Q}$, initial a priori state estimate $\mathbf{x}_0^-$ and the initial a priori estimate error covariance $\mathbf{P}_0^-$

$$\mathbf{w} = \begin{bmatrix} \omega_\omega \\ \omega_b \end{bmatrix} \qquad E\{\mathbf{w}\mathbf{w}^T\} = \mathbf{Q} = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix}$$

$$\mathbf{P}_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-4} \end{bmatrix} \qquad \hat{\mathbf{x}}_0^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

With the measurement error varians $R = \frac{\textit{measurement noise variance}}{\textit{sample time}} = \frac{6.274 \cdot 10^{-7}}{0.1} = 6.274 \cdot 10^{-6}$, we have all the information needed to create a discrete Kalman filter.

We choose to implement the Kalman filter with a normal Matlab function within the Matlab function Simulink block. The input of the Matlab function is ψ and δ, and the output is the estimate $\hat{\mathbf{x}}_k = [\hat{\xi}_\omega \quad \hat{\psi}_\omega \quad \hat{\psi} \quad \hat{r} \quad \hat{b}]^T$.

Explanation of the recursive process; the Kalman filter loop [1, p. 143-147]:

1. First we compute the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1}$$

2. Then we update the estimate with the new measurement ψ

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\psi - \mathbf{C}_k \hat{\mathbf{x}}_k^-)$$

3. Next we compute the error covariance for the updated estimate

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$$

4. At last we project ahead the new $\hat{\mathbf{x}}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ for the next iteration

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \delta$$

$$\mathbf{P}_{k+1}^{-} = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{E}_k \mathbf{Q}_k \mathbf{E}_k^T$$

This concludes one iteration of the Kalman filter loop.

d)

As the rudder is subject to current bias, we compensate by adding a feed forward from the estimated bias. Then, simulation of the system with current disturbance results in

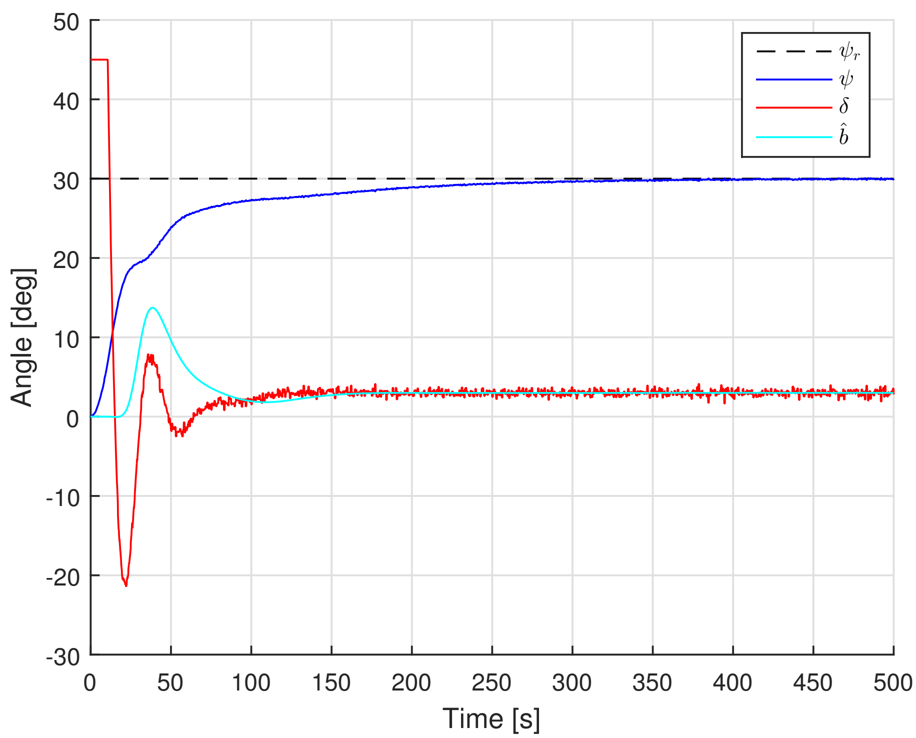

Figure 12: Simulation of the system with current disturbance

From Figure 12, we clearly see that the feed forward from the estimated bias cancels out the bias we experienced earlier with the PD regulator, effectively eliminating the stationary deviation. As a result, the autopilot now works; the heading reaches and stays at the reference angle with no deviation. We could also see this algebraically, as the ship heading $\dot{\psi} = r$ and with

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

With the feed forward, we get

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

where $\hat{b}$ is the estimate bias to the rudder. Thus the bias is cancelled.

e)

As the heading of the ship is subject to high-frequency wave induced motion, we substitute the feedback from the measured heading with the wave filtered estimated heading. Simulation of the system with wave disturbance yields

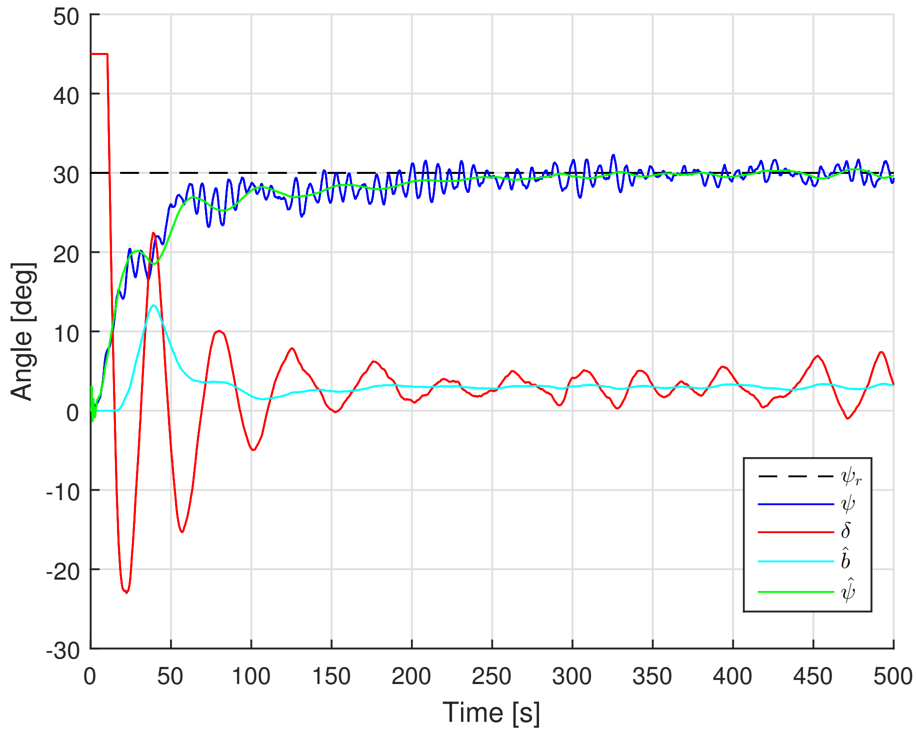

Figure 13: Simulation of system with wave disturbance

From Figure 13, we see that the oscillations in the rudder is significantly reduced, as we no longer use the oscillating heading as feedback. We also see that the estimation of the heading is very close to the measured heading for almost the whole time interval, making the overall performance of the autopilot much better than the one from Part 5.3 d). However, the estimated heading is subject to severe spikes initially. This is probably due to the non-optimal initial a priori estimate error covariance. One could avoid this by simulating the system and extract the more optimal error covariance from where the filter converges to its desired value, and substitute this with the initial a priori estimate error covariance.

Plotting the actual wave influence with estimated wave influence yields

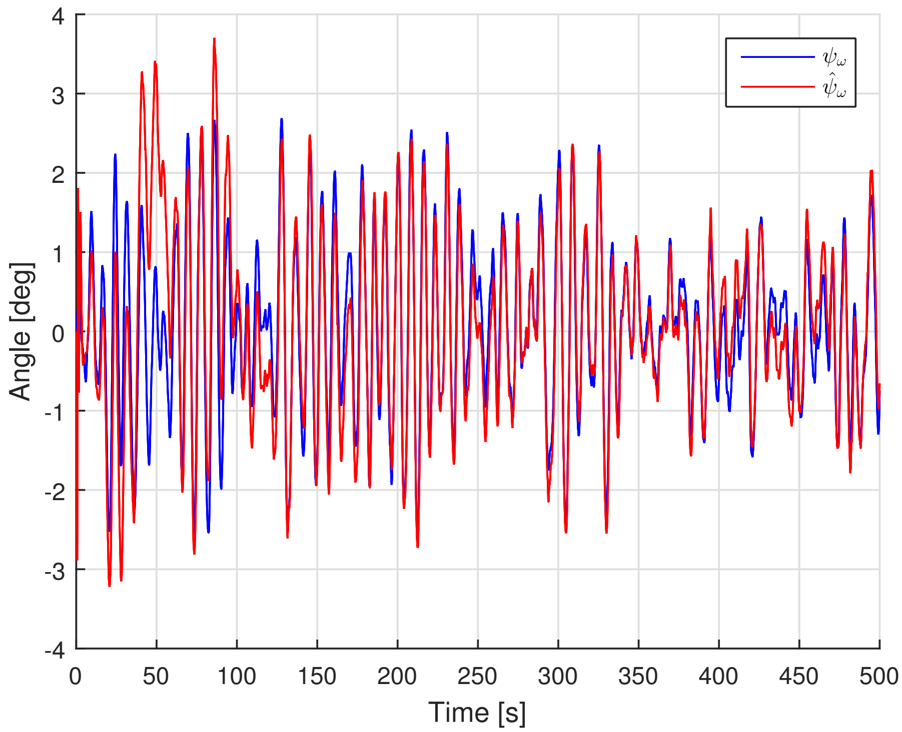

Figure 14: Estimated wave influence (red) vs. actual wave influence (blue)

From Figure 14, we see that the estimated wave influence is very close to the actual wave influence. However, as with the heading, the estimated wave influence is subject to severe spikes initially. This is due to the same reason as before.

References

[1] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition*. John Wiley & Sons Ltd, 2012.

Matlab scripts

```
%% 5.1 - Identification of the boat parameters

%% a - Constants
K = 0.175;  % K = 0.1825
T = 88.697;
simTime = 2500;

%% b
omega = 0.005;
sim('ship_5_1_b_disturbances_off', [0 simTime]);
fig1 = figure(1);
x1 = ship.time;
y1 = ship.signals.values;
plot(x1, y1, 'b');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
min_y1 = abs(findpeaks(-y1));
max_y1 = max(y1);
amplitude1 = min((max_y1 - min_y1) / 2);

omega = 0.05;
sim('ship_5_1_b_disturbances_off', [0 simTime]);
fig2 = figure(2);
x2 = ship.time;
y2 = ship.signals.values;
plot(x2, y2, 'b');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
min_y2 = abs(findpeaks(-y2));
max_y2 = max(y2);
amplitude2 = min((max_y2 - min_y2) / 2);

%% c
omega = 0.005;
sim('ship_5_1_c_disturbances_on', [0 simTime]);
fig3 = figure(3);
x3 = ship.time;
y3 = ship.signals.values;
plot(x3, y3, 'r');
grid on;
xlabel('Time [s]');
```

```
ylabel('Angle [deg]');

omega = 0.05;
sim('ship_5_1_c_disturbances_on', [0 simTime]);
fig4 = figure(4);
x4 = ship.time;
y4 = ship.signals.values;
plot(x4, y4, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');

%% d
sim('ship_5_1_d', [0 simTime]);
fig5 = figure(5);
x5 = ship.time;
y5 = ship.signals.values;
x6 = model.time;
y6 = model.signals.values;
hold on;
plot(x5, y5, 'b');
plot(x6, y6, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('Ship', 'Model');

%% Printing simulink
% print('-depsc','-s','ship_5_1_bc');
% print('-depsc','-s','ship_5_1_d');
%%
```

Listing 1: 5.1 – Identification of the boat parameters

```
%% 5.2 - Identification of wave spectrum model

%% a
load('wave.mat', 'psi_w');
window = 4096;
fs = 10;
x = psi_w(2,:)*pi/180;
[pxx,f] = pwelch(x,window,[],[],fs);

fig1 = figure(1);
hold on;
plot(f*2*pi, pxx/(2*pi), 'b');
```

```
grid on;
xlabel('Frequency [rad/s]');
ylabel('Power [s/rad]');
axis([0 2 0 0.0016]);

%% c
indexmax = find(max(pxx) == pxx);
y_max = pxx(indexmax)/(2*pi);
x_max = f(indexmax)*2*pi;
omega_0 = x_max;

%% d
sigma = sqrt(y_max);
for lambda = 0.14: -0.05: 0.04
    P = (2*sigma*lambda*omega_0.*f).^2 ./ ((omega_0^2 - f.^2).^2
    + (2.*f*omega_0*lambda).^2);
    plot(f, P);
end
legend('PSD estimate', '\lambda = 0.14', '\lambda = 0.09', '\
    lambda = 0.04');
%%
```

Listing 2: 5.2 – Identification of wave spectrum model

```
%% 5.3 - Control system design

%% Constants
K = 0.175;
T = 88.697;
simTime = 500;
load('wave.mat', 'psi_w');

%% a
psi_r = 30;
omega_c = 0.1;
phase_margin = 50*pi/180;
T_f = 1/ (tan(phase_margin)*omega_c);
T_d = T;
K_pd = omega_c*sqrt(1+(T_f*omega_c)^2)/K;
H_pd = tf([K_pd*T_d K_pd], [T_f 1]);

%% b
sim('ship_5_3_b', [0 simTime]);
fig1 = figure(1);
x1 = ship.time;
y1 = ship.signals.values;
```

```
x2 = rudder.time;
y2 = rudder.signals.values;
hold on;
plot([0 simTime],[psi_r psi_r],'k--');
plot(x1, y1, 'b');
plot(x2, y2, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('\psi_r', '\psi', '\delta');

%% c
sim('ship_5_3_c', [0 simTime]);
fig2 = figure(2);
x3 = ship.time;
y3 = ship.signals.values;
x4 = rudder.time;
y4 = rudder.signals.values;
hold on;
plot([0 simTime],[psi_r psi_r],'k--');
plot(x3, y3, 'b');
plot(x4, y4, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('\psi_r', '\psi', '\delta');

%% d
sim('ship_5_3_d', [0 simTime]);
fig3 = figure(3);
x5 = ship.time;
y5 = ship.signals.values;
x6 = rudder.time;
y6 = rudder.signals.values;
hold on;
plot([0 simTime],[psi_r psi_r],'k--');
plot(x5, y5, 'b');
plot(x6, y6, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('\psi_r', '\psi', '\delta');

%% Printing simulink
% print('-depsc','-s','ship_5_3_bcd');
```

```
%%
```

Listing 3: 5.3 – Control system design

```
%% 5.4 - Observability

%% Constants
K = 0.175;
T = 88.697;
omega_0 = 0.782330201821677;
lambda = 0.09;
sigma = 0.038525441820137;
K_omega = 2 * lambda * omega_0 * sigma;

%% a
A = [          0                    1  0     0     0;
     -omega_0^2  -2*lambda*omega_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]';
C = [0 1 1 0 0];
E = [0 0; K_omega 0; 0 0; 0 0; 0 1];

%% b
A_1 = [0 1; 0 -1/T];
C_1 = [1 0];
O_1 = obsv(A_1,C_1);
rank(O_1);

%% c
A_2 = [0 1 0; 0 -1/T -K/T; 0 0 0];
C_2 = [1 0 0];
O_2 = obsv(A_2,C_2);
rank(O_2);

%% d
A_3 = [          0                    1  0     0;
       -omega_0^2  -2*lambda*omega_0  0     0;
                 0                    0  0     1;
                 0                    0  0  -1/T];
C_3 = [0 1 1 0];
O_3 = obsv(A_3,C_3);
rank(O_3);

%% e
```

```
A_4 = [             0                         1  0      0      0;
            -omega_0^2  -2*lambda*omega_0  0  0      0      0;
                    0                         0  0      1      0;
                    0                         0  0   -1/T   -K/T;
                    0                         0  0      0      0];
C_4 = [0  1  1  0  0];
O_4 = obsv(A_4,C_4);
rank(O_4);

%%
```

Listing 4: 5.4 – Observability

```
%% 5.5 - Discrete Kalman filter

%% Constants
simTime = 500;
K = 0.175;
T = 88.697;
omega_0 = 0.782330201821677;
lambda = 0.09;
sigma = 0.038525441820137;
K_omega = 2 * lambda * omega_0 * sigma;
load('wave.mat', 'psi_w');
psi_r = 30;

%% PD-controller
omega_c = 0.1;
phase_margin = 50*pi/180;
T_f = 1/ (tan(phase_margin)*omega_c);
T_d = T;
K_pd = omega_c*sqrt(1+(T_f*omega_c)^2)/K;
H_pd = tf([K_pd*T_d K_pd], [T_f 1]);

%% a
A = [             0                         1  0      0      0;
          -omega_0^2  -2*lambda*omega_0  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]';
C = [0  1  1  0  0];
D = 1;
E = [0  0;  K_omega  0;  0  0;  0  0;  0  1];

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

```
[A_d, B_d] = c2d(A, B, Ts);
[A_d, E_d] = c2d(A, E, Ts);
C_d = C;
D_d = D;

%% b
sim('ship_5_5_b', [0 simTime]);
fig1 = figure(1);
x1 = ship.time;
y1 = ship.signals.values * pi/180;
plot(x1, y1, 'b');
grid on;
xlabel('Time [s]');
ylabel('Angle [rad]');
V_rad = var(y1);

%% c
Q = diag([30 10^(-6)]);
P_0 = diag([1 0.013 pi^2 1 2.5*10^(-4)]);
R_rad = V_rad/Ts;
X_0 = [0 0 0 0 0]';
kalmanFilter = struct('A_d', A_d, 'B_d', B_d, 'C_d', C_d, 'E_d',
    E_d, 'Q', Q, 'P_0', P_0, 'R_rad', R_rad, 'X_0', X_0);

%% d
sim('ship_5_5_d', [0 simTime]);
fig2 = figure(2);
x2 = ship.time;
y2 = ship.signals.values;
x3 = rudder.time;
y3 = rudder.signals.values;
x4 = bias_estimate.time;
y4 = bias_estimate.signals.values;
hold on;
plot([0 simTime],[psi_r psi_r],'k--');
plot(x2, y2, 'b');
plot(x3, y3, 'r');
plot(x4, y4, 'c');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('$\psi_r$', '$\psi$', '$\delta$', '$\hat{b}$');
set(legend,'Interpreter','latex');

%% e
```

```
sim('ship_5_5_e', [0 simTime]);
fig3 = figure(3);
x5 = ship.time;
y5 = ship.signals.values;
x6 = rudder.time;
y6 = rudder.signals.values;
x7 = bias_estimate.time;
y7 = bias_estimate.signals.values;
x8 = ship_estimate.time;
y8 = ship_estimate.signals.values;
hold on;
plot([0 simTime],[psi_r psi_r],'k--');
plot(x5, y5, 'b');
plot(x6, y6, 'r');
plot(x7, y7, 'c');
plot(x8, y8, 'g');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('$\psi_r$', '$\psi$', '$\delta$', '$\hat{b}$', '$\hat{\psi}$');
set(legend,'Interpreter','latex');

fig4 = figure(4);
x9 = wave.time;
y9 = wave.signals.values;
x10 = wave_estimate.time;
y10 = wave_estimate.signals.values;
hold on;
plot(x9, y9, 'b');
plot(x10, y10, 'r');
grid on;
xlabel('Time [s]');
ylabel('Angle [deg]');
legend('$\psi_\omega$', '$\hat{\psi}_\omega$');
set(legend,'Interpreter','latex');

%% Printing simulink
% print('-depsc','-s','ship_5_5_b');
% print('-depsc','-s','ship_5_5_c');
% print('-depsc','-s','ship_5_5_d');
% print('-depsc','-s','ship_5_5_e');
%%
```

Listing 5: 5.5 – Discrete Kalman filter

```
function X_hat  = fcn(ship, rudder, kalmanFilter)

persistent P_k X_k init_flag R_rad Q A_d B_d C_d E_d;

if isempty(init_flag)
     init_flag = 1;
     P_k = kalmanFilter.P_0;
     X_k = kalmanFilter.X_0;
     R_rad = kalmanFilter.R_rad;
     Q = kalmanFilter.Q;
     A_d = kalmanFilter.A_d;
     B_d = kalmanFilter.B_d;
     C_d = kalmanFilter.C_d;
     E_d = kalmanFilter.E_d;
end

%(1) Kalman gain
K_k = P_k*C_d'*inv(C_d*P_k*C_d'+R_rad);
%(2) Update estimate
X_hat = X_k + K_k*(ship-C_d*X_k);
%(3) Update covariance
P_k = (eye(5)-K_k*C_d)*P_k*(eye(5)-K_k*C_d)' + K_k*R_rad*K_k';
%(4) Project ahead
X_k = A_d*X_hat + B_d*rudder;
P_k = A_d*P_k*A_d'+E_d*Q*E_d';

end
```

Listing 6: The Matlab function code for the discrete Kalman filter

Simulink models

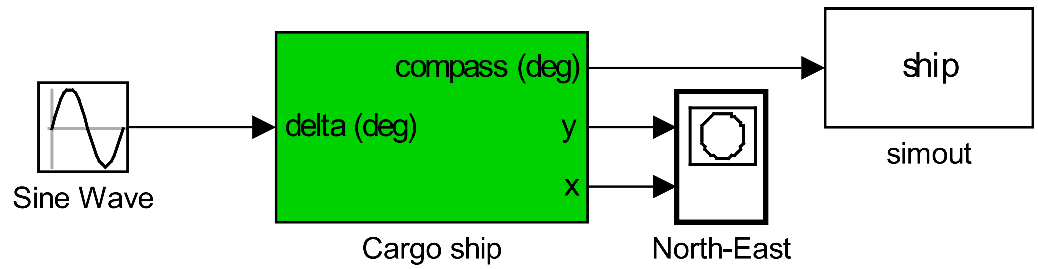

Figure 15: 5.1 b) and c) – Identification of the boat parameters

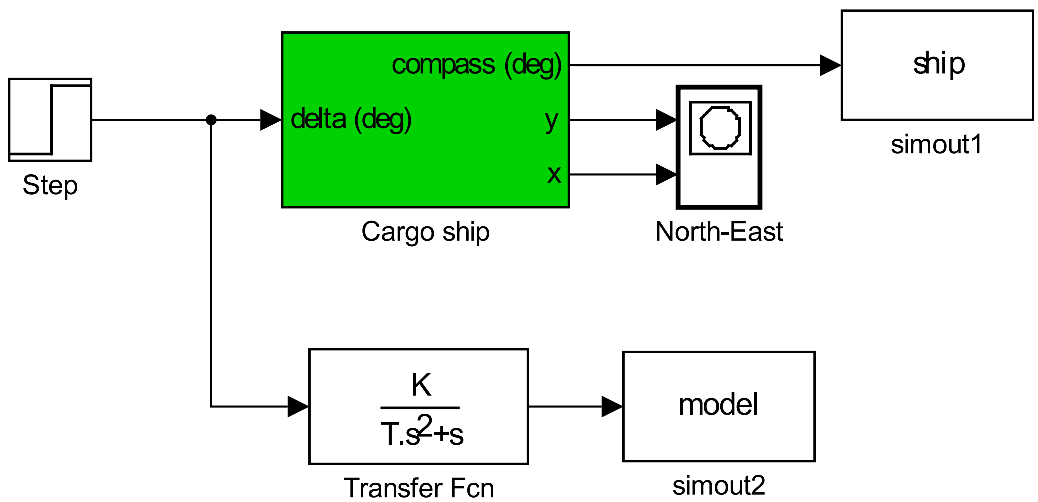

Figure 16: 5.1 d) – Step response

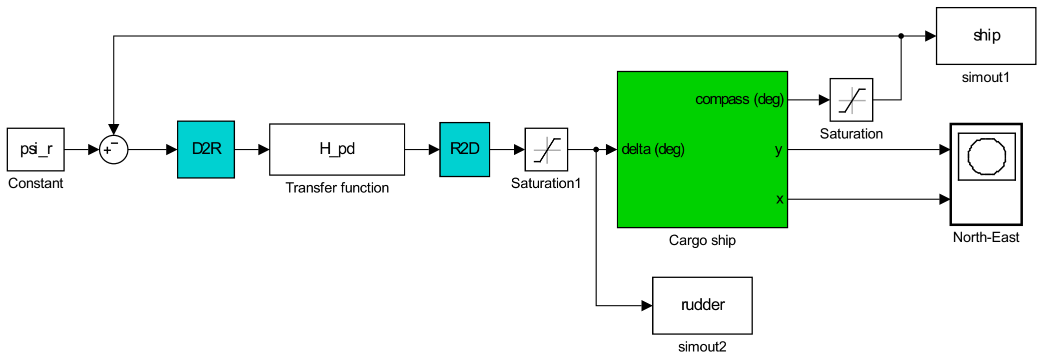

Figure 17: 5.3 b), c) and d) – Control system design

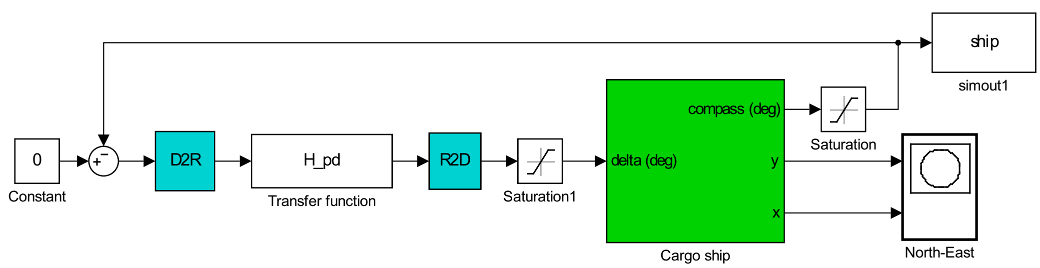

Figure 18: 5.5 b) – Simulating measurement noise

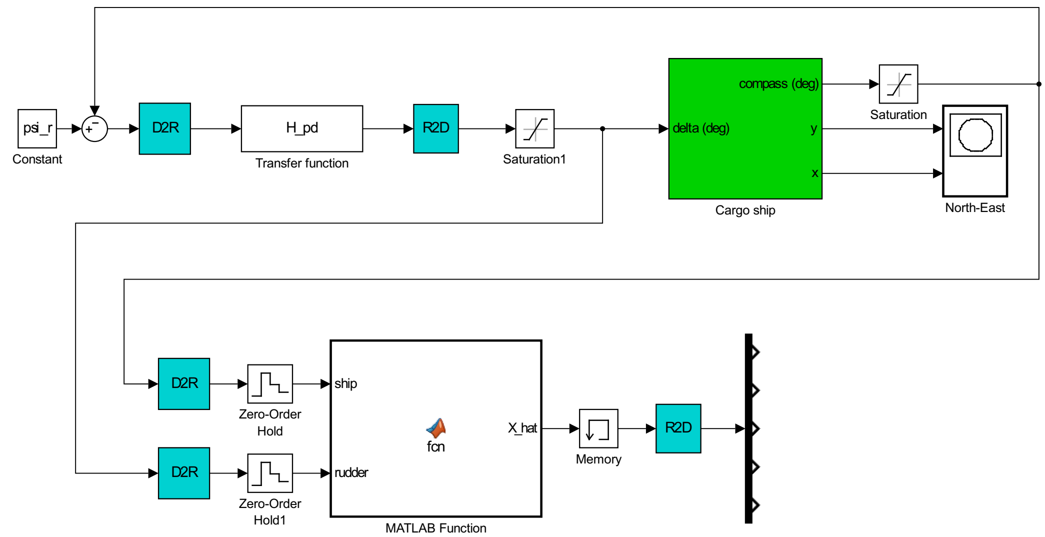

Figure 19: 5.5 c) – The simulink model for the discrete Kalman filter

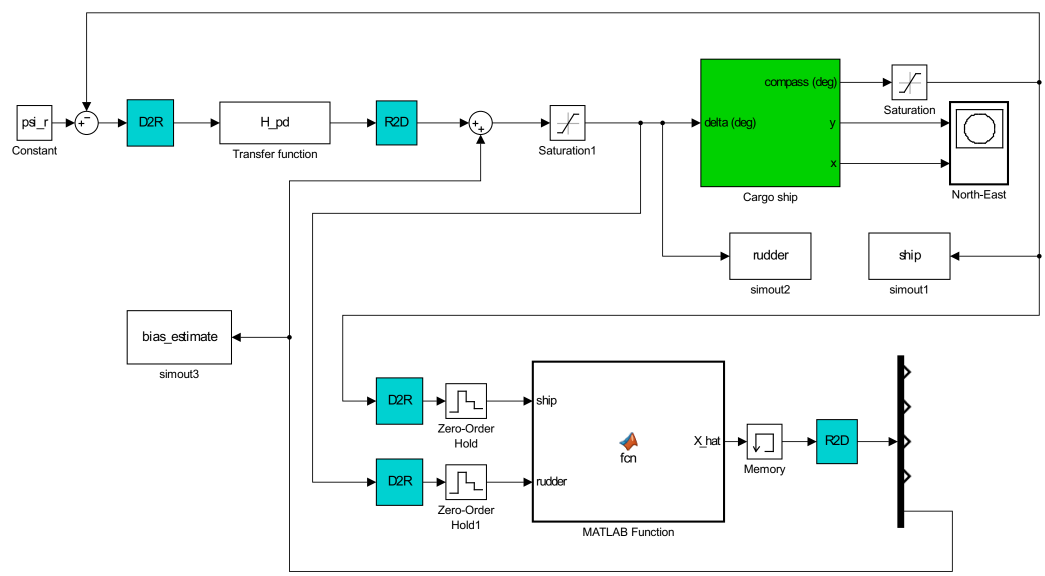

Figure 20: 5.5 d) – Discrete Kalman filter, feed forward from estimated bias

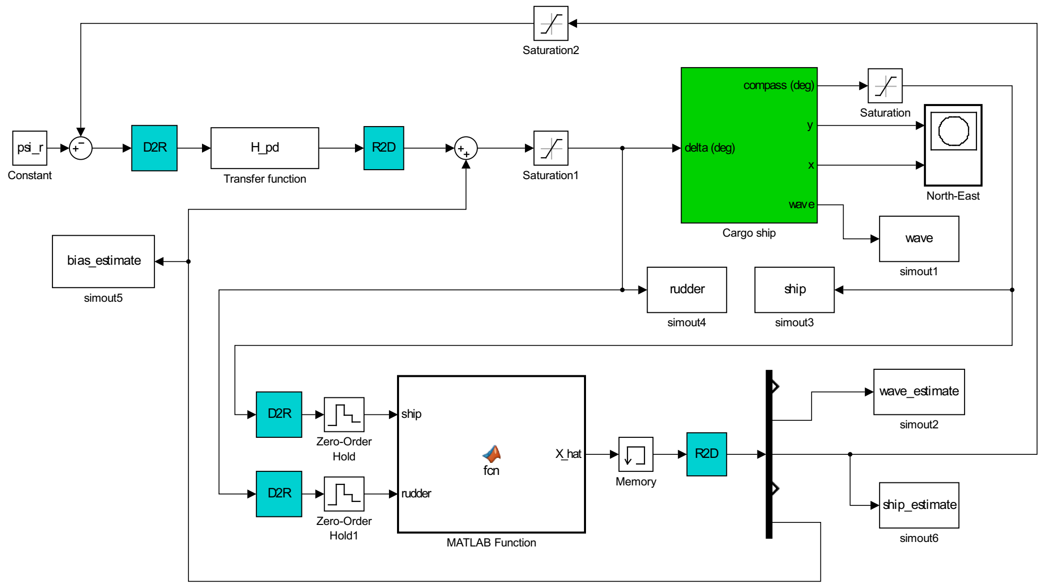

Figure 21: 5.5 e) – Discrete Kalman filter, feedback from estimated heading