TTK4115 Linear System Theory
Boat Lab report

Group 62

Fan Gao 502425
Hongyu Zhou 502423
Shuyuan Shen 502427

NTNU

Department of Engineering Cybernetics
Norwegian University of Science and Technology

November 24, 2018

Contents

1 Introduction

In the project, we are to model and simulate a vessle influenced by measurement noise, current and waves disturbances. The vessle is modeled as a continuous system and its parameters should be found first. The parameters of disturbances, stochastic signals, are also found for implementation of observor. A simple PD controller will be designed, working with Kalman filter which is able to filter out the noise based on estimations of disturbance. Matlab and Simulink are used for simulatons. The system for modelling the boat is given as follows:

$$
\begin{aligned}
\dot{\xi_w} &= \psi_w \\
\dot{\psi_w} &= -{\omega_0}^2\xi_w - 2\lambda_0\psi_w + K_w w_w \\
\dot{\psi} &= r \\
\dot{r} &= -\frac{1}{T}r + \frac{K}{T}(\delta - b) \\
\dot{b} &= w_b \\
y &= \psi + \psi_w + v
\end{aligned}
\tag{1.1}
$$

In this system, ψ is the average heading of the boat, ψ_w is the high-frequency component in the heading caused by wave disturbances, r is the rotation velocity about z-axis, b is bias to the rudder angle, δ is the rudder angle relative to the BODY frame, w_b is white noise distubances from current and w_w is a zero mean white precess with unity variance from the waves. In summary, the system can be written as

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} + \mathbf{Ew} \\
y &= \mathbf{Cx} + v
\end{aligned}
\tag{1.2}
$$

where $\mathbf{x}$, u and $\mathbf{w}$ are given as

$$
\begin{aligned}
\mathbf{x} &= \begin{bmatrix} \xi_w & \psi_w & \psi & r & b \end{bmatrix}^T \\
u &= \begin{bmatrix} \delta \end{bmatrix}^T \\
\mathbf{w} &= \begin{bmatrix} w_w & w_b \end{bmatrix}^T
\end{aligned}
\tag{1.3}
$$

2 Identication of the boat parameters

2.1 Transfer function from δ to ψ

To find the transfer function from δ to ψ, the equations are given by

$$\begin{aligned} \dot{\psi} &= r \\ \dot{r} &= -1/T * r + K/T * (\delta - b) \end{aligned} \tag{2.1}$$

After Laplace Tranformation,

$$\begin{aligned} s\psi(s) &= r(s) \\ sr(s) &= \frac{1}{T}r(s) + \frac{K}{T}(\delta(s) - b(s)) \end{aligned} \tag{2.2}$$

Thus the transfer function is

$$H(s) = \frac{\psi(s)}{\delta(s)} = \frac{K}{s(Ts+1)} \tag{2.3}$$

In the transfer function, we can see that it is characterized by pameters T and K, and it gives the relation bewteen rudder input δ and compass course. Therefore, the next task is to find parameters T and K.

2.2 Boat parameters T and K

In this task, all disturbances in the model are turned off, so that the boat parameters can be identified. Two sine signals with amplitude 1 and different frequencies are implemented in the Simulink model. The frequencies are $\omega_1 = 0.005$ and $\omega_2 = 0.05$. The amplitude of the output will be equal to $|H(j\omega_1)|$ and $|H(j\omega_2)|$.
For $\omega_1 = 0.005$ ans $\omega_2 = 0.05$, amplitudes are obtained as

$$\begin{aligned} A_1 = |H(j\omega_1)| &= \frac{63.3582 - 4.6417}{2} = 29.3582 \\ A_2 = |H(j\omega_2)| &= \frac{4.2308 - 2.5687}{2} = 0.8310 \end{aligned} \tag{2.4}$$

So the boat parameters T and K can be calculated as

$$\begin{aligned} T &= A_1\omega_1\sqrt{T^2{\omega_1}^2 + 1} = 72.4391 \\ K &= \frac{\sqrt{K^2 - A_2{\omega_2}^2}}{A_2{\omega_2}^2} = 0.1561 \end{aligned} \tag{2.5}$$

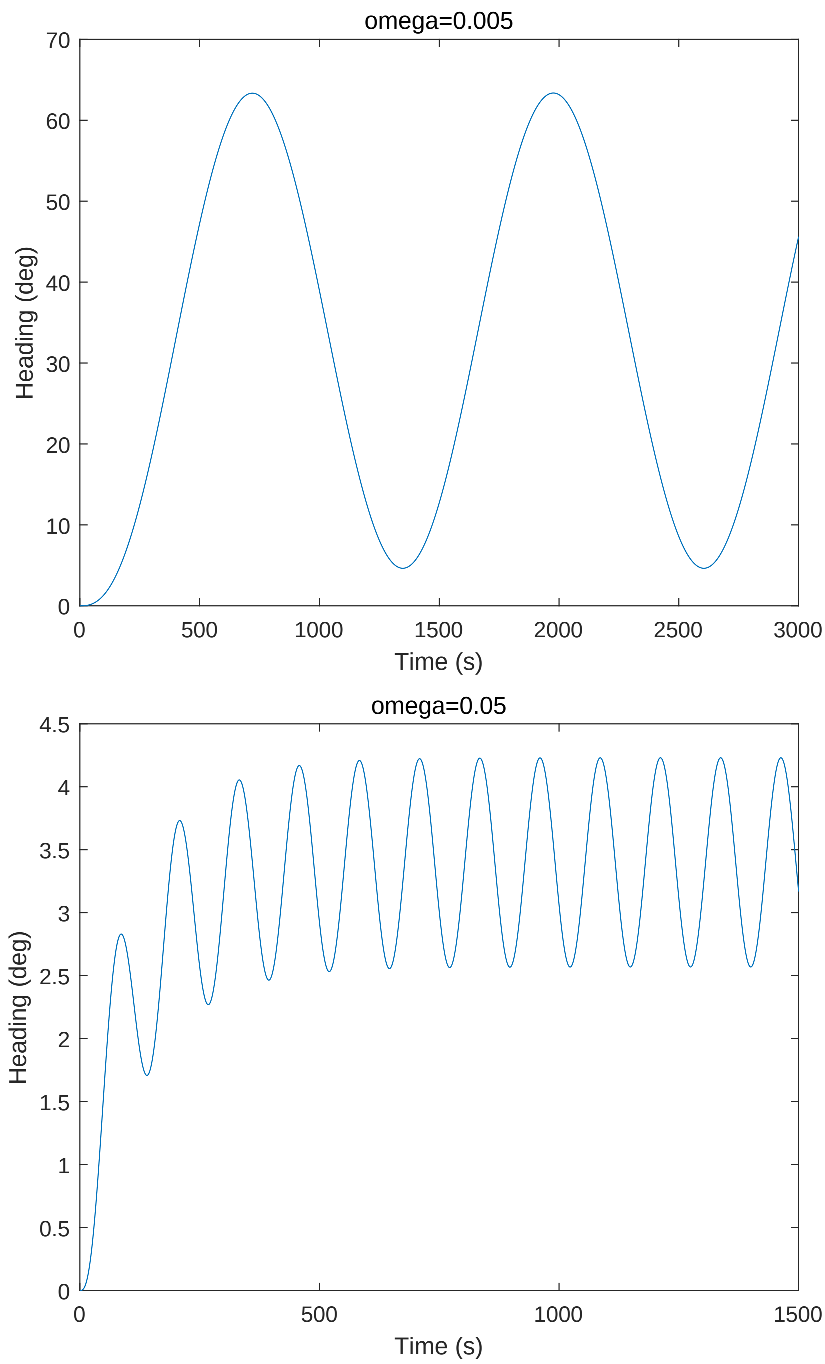

Figure 1: Simulation without disturbances

2.3 Waves and measurement noise

To find good estimates of the boat parameters is easy in good weather conditions, as shown in section 2.2. However, it's clear that parameter estimation becomes very difficult when there are wave and measurement disturbances, especially for higher frequency input, as shown in Figure 2.

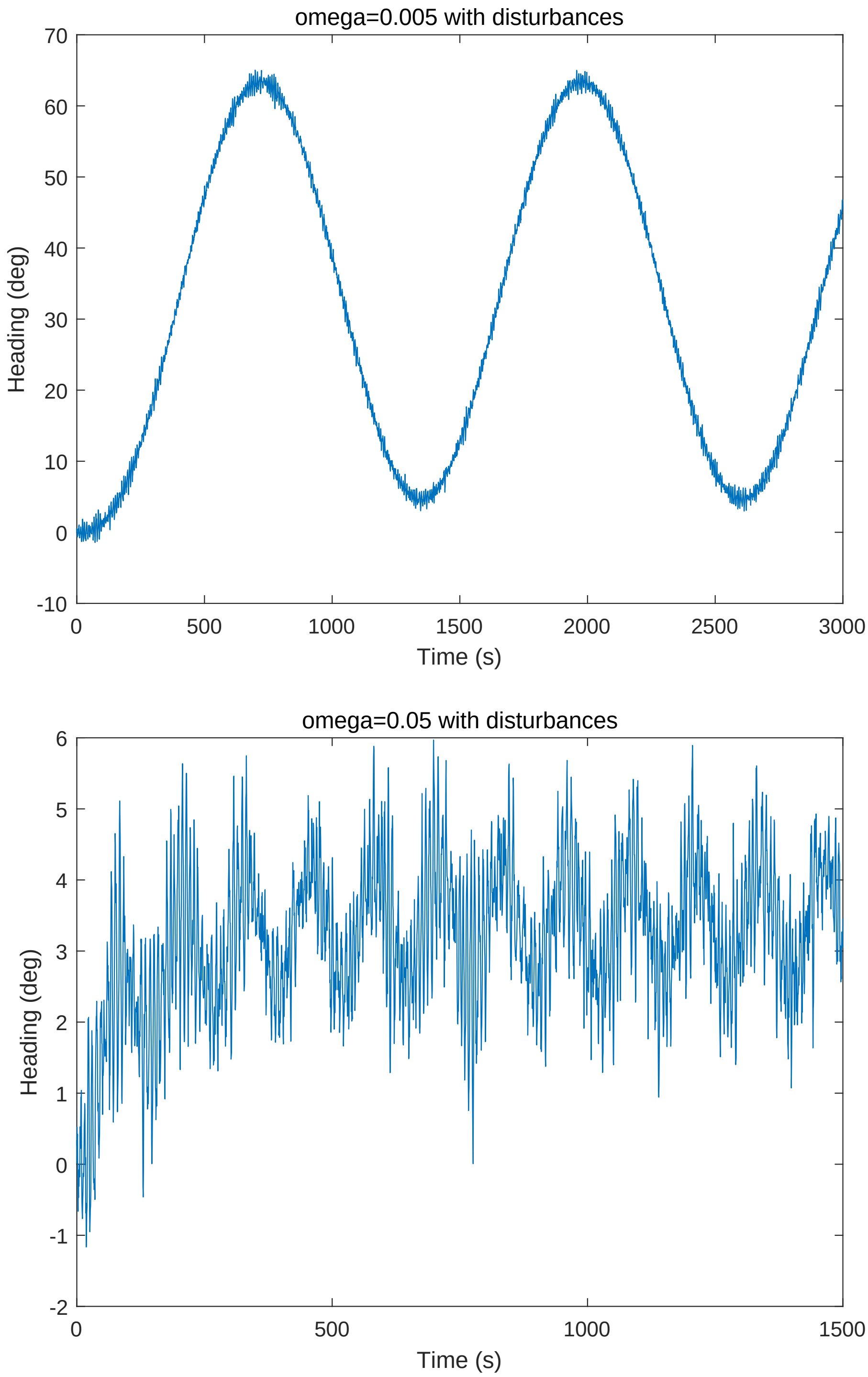

Figure 2: Simulation with disturbances

For $\omega = 0.005$, it is possible to have similar parameters with those obtained in last task, as the noise only cause small deviations from the correct heading. However, these small deviations would still have effect on the results. The measurement noise makes it harder to observe the correct amplitude of the oscillations, which will, in some degree, change the values of the parameters.
As the frequency of input signal becomes larger, it is nearly impossible to have a good estimation of boat parameters, as the noise can not be separated correctly from the

heading. Thus, to identify boat parameters, the best condition is the whether without disturbances or with low-frequency disturbances.

2.4 Step response

The plot in Figure 3 shows that the response of transfer function model shares the same tendency with the ship's response but as time increases, deviation increases as well. As a model, this resulting deviation is acceptable. Therefore the model will be used in following simulations.

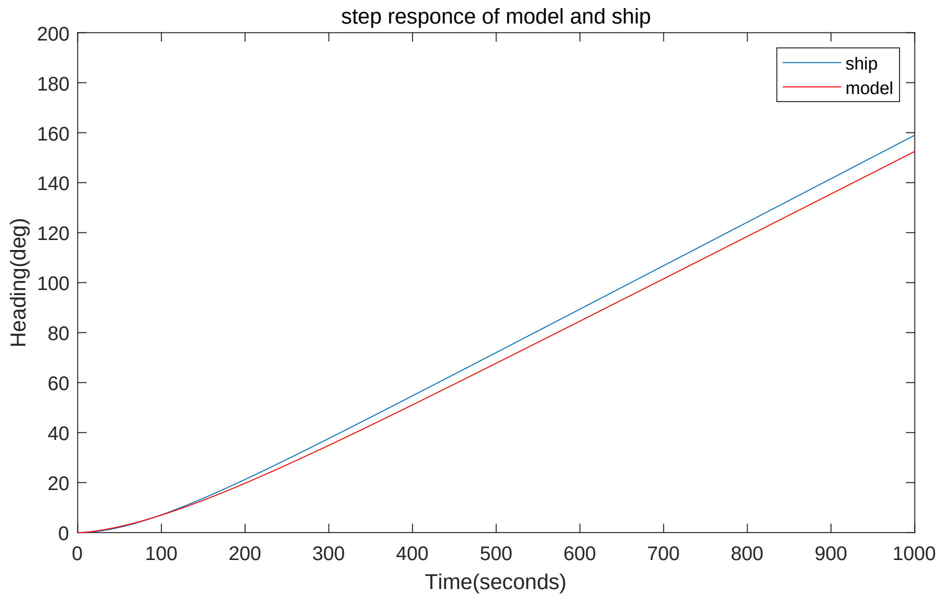

Figure 3: Step response of model and ship

3 Identification of wave spectrum model

In this part, the Power Spectral Density (PSD) of the wave system is calculated and used to identify the parameters of the wave spectrum.

3.1 Power spectral density

In order to get the Power Spectral Density of waves, the data file `wave.mat` is used as basic parameters. Use the MATLAB function `[pxx,f] = pwelch(x, window, [], [], fs)` to find the PSD intensity (pxx) for each frequency (f). The window size is set as 4096 and sampling frequency is 10Hz.

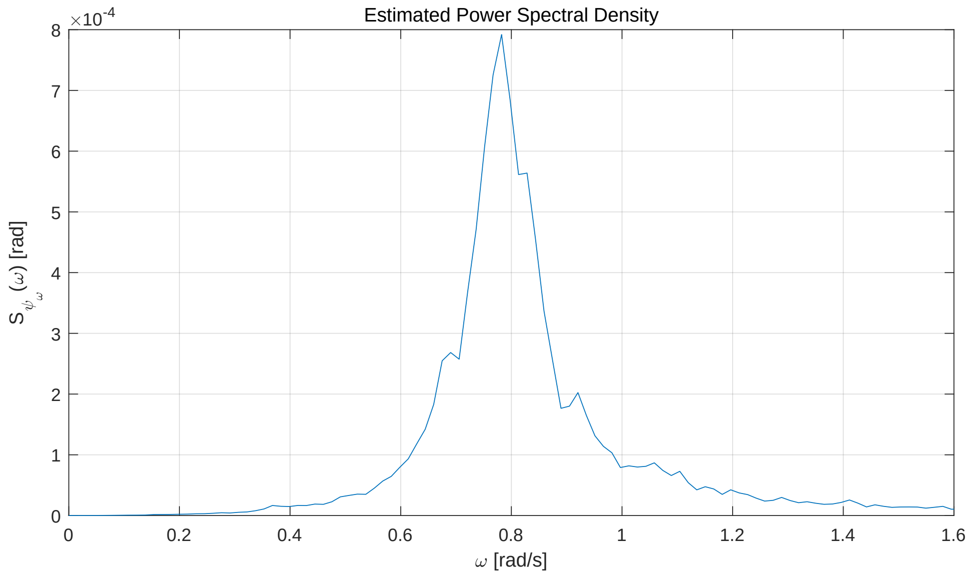

Figure 4: Plot of the estimated Power Spectral Density

3.2 Transfer function from ω_w to ψ_w

In this part, the relation between ω_ω, ψ_ω and ξ_ω are

$$\begin{aligned}\dot{\xi}_w &= \psi_w \\ \dot{\psi}_w &= -\omega_0^2 \xi_w - 2\lambda\omega_0\psi_w + K_w w_w\end{aligned} \tag{3.1}$$

Laplace transform is used and the transfer function is calculated

$$\begin{aligned}\xi_w(s) &= \frac{\psi_w(s)}{s} \\ s\psi_w(s) &= -\omega_0^2 \xi_w(s) - 2\lambda\omega_0\psi_w + K_w w_w(s)\end{aligned} \tag{3.2}$$

$$H_{w_w,\psi_w}(s) = \frac{\psi_w(s)}{w_w(s)} = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \tag{3.3}$$

To get the power density spectrum of the output signal, we need to find an analytical expression between $S_x(j\omega)$ and $S_f(j\omega)$ as

$$S_{\psi_\omega}(j\omega) = H_{\psi_\omega}(j\omega)H_{\psi_\omega}(-j\omega)S_f(j\omega) \tag{3.4}$$

In our case, assume $S_f(j\omega) = 1$ for white noise. Then the expression for the power spectrum density is

$$P_{\psi_\omega} = \frac{-K_\omega^2 s^2}{s^4 + \omega_0^2 s^2 - 4\lambda^2\omega_0^2 s^2 + \omega_0^4} \tag{3.5}$$

3.3 Resonance frequency

From the estimated PSD, the resonance frequency is $\omega_0 = 0.7823$ which is the peak frequency. This shows that the wave disturbance have the biggest signal influence around a frequency of $\omega_0 = 0.7823$.

3.4 Damping factor

Let $K_\omega = 2\lambda\omega_0\beta$, where β^2 corresponds to the peak of S_{ψ_ω}. The damping factor λ was found by fitting the analytical PSD to the estimated using the MATLAB-command `lsqcurvefit`.

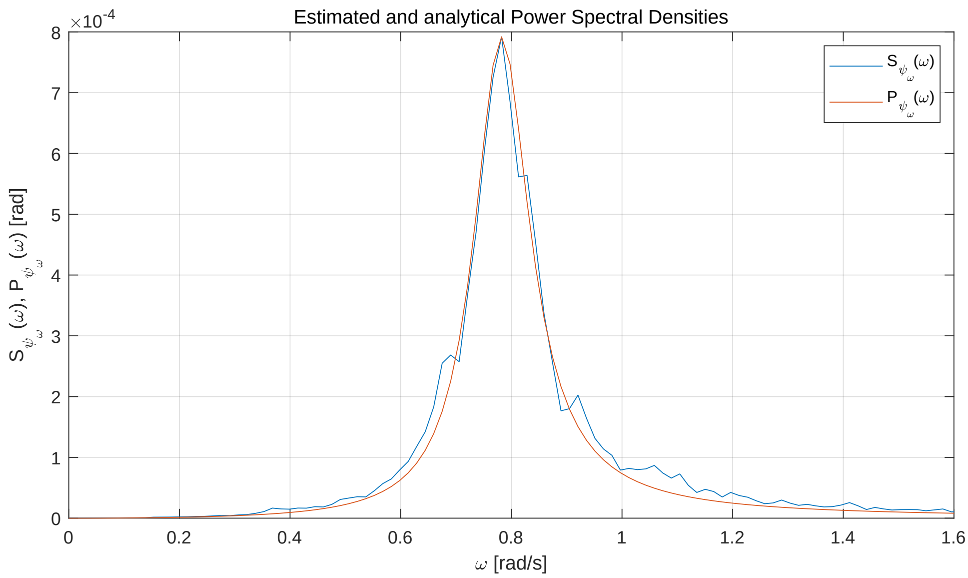

Figure 5: Plot of the estimated and analytical Power Spectral Densities

The fitted analytical Power Spectral Density is shown in Figure 5. The result is $\lambda = 0.0791$.

4 Control system design

4.1 PD controller design

The transfer function of PD controller and the transfer function H_{ship} from $\delta(s)$ to $\psi(s)$ are defined respectively as

$$H_{pd} = K_{pd}\frac{1 + T_d s}{1 + T_f s} \tag{4.1}$$

$$H_{ship} = \frac{K}{s(T + 1)} \tag{4.2}$$

The derivative time constant, T_d, is chosen to cancel the time constant in the transfer function of boat, which means $T_d = T$. Thus, the transfer function of the system is

$$H_s = H_{pd}H_{ship} = \frac{KK_{pd}}{s(T_f s + 1)} \tag{4.3}$$

Since the phase margin is $\psi = 50°$ and $\omega_c = 0.1rad/s$, the derivative time constant T_f is

$$\begin{aligned} \varphi = 50° &= \angle H_s(j\omega_c) - (-180°) \\ \angle H_s(j\omega_c) &= \angle \frac{KK_{pd}}{j\omega_c(1 + jT_f\omega_c)} = -130° \\ 0 - 90° - arctan(T_f\omega_c) &= -130° \\ T_f &= \frac{tan40°}{\omega_c} = 8.3910 \end{aligned} \tag{4.4}$$

The parameter K_{pd} is calculated as

$$\begin{aligned} |H_s(j\omega_c)| &= 1 \\ \sqrt{\frac{K^2K_{pd}^2}{\omega_c^2|1 + jT_f\omega_c|^2}} &= 1 \\ K_{pd} = \frac{\omega_c|1 + jT_f\omega_c|}{K} &= 0.8363 \end{aligned} \tag{4.5}$$

The bode plot of the system transfer function with PD controller is shown in Figure 6.

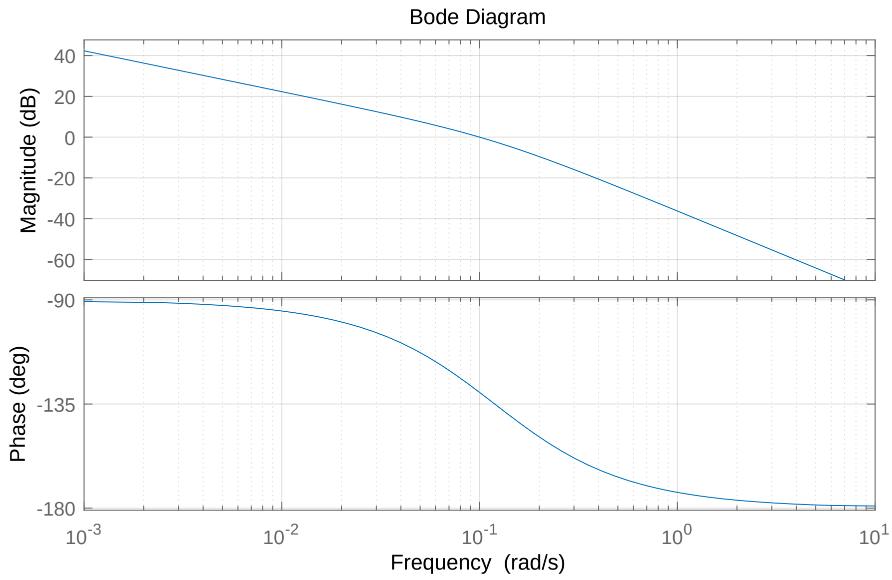

Figure 6: Bode plot of the system transfer function with PD controller

4.2 Simulation without disturbances

First, simulate the system using the designed PD controller without disturbances, only measurement noise. The result is shown in Figure 7. It can be seen from the figure that in around 300s, the vessel reached the reference course angle at 30° and keep it stable in the following time. This is a good result and the PD controller works well without environment loads disturbances. The position of rudder shows that the rudder reach to 0 fast with small variations.

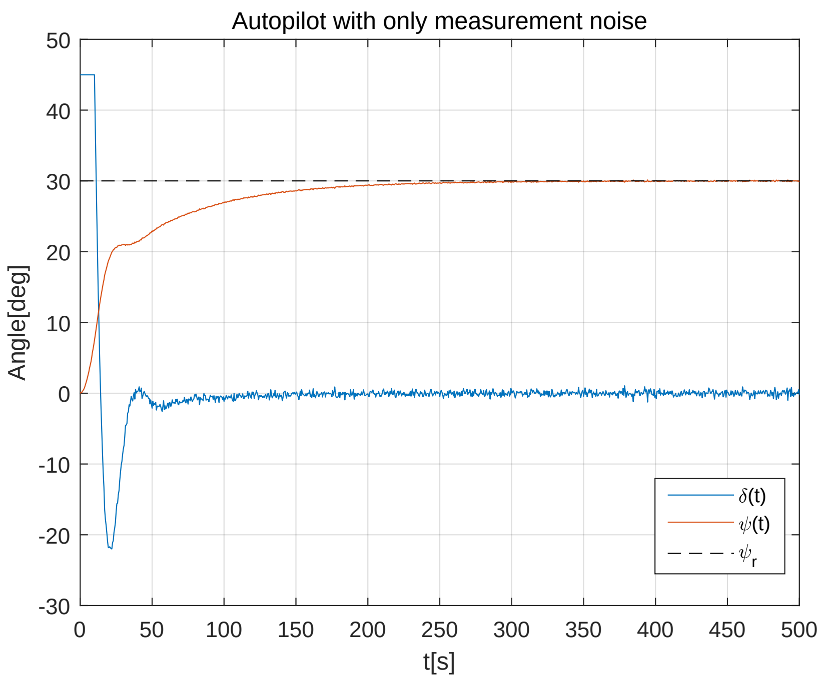

Figure 7: Autopilot with only measurement noise

4.3 Simulation with current disturbance

In this case, autopilot is test with current disturbance and without wave disturbance. It turns out to be a fairly poor response in Figure 8. The heading fails to reach the reference course angle at 30°, because of the current disturbance, which causes a steady state error of approximately 3°. This steady state error can be removed by implementing a PID controller. The integrating part would accumulate the error and correct it.

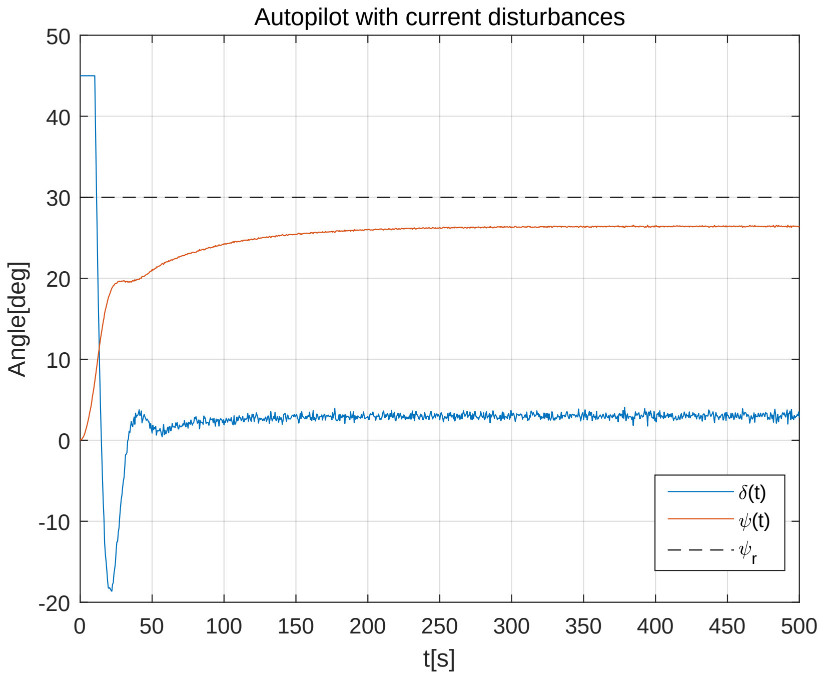

Figure 8: Autopilot with only current disturbance

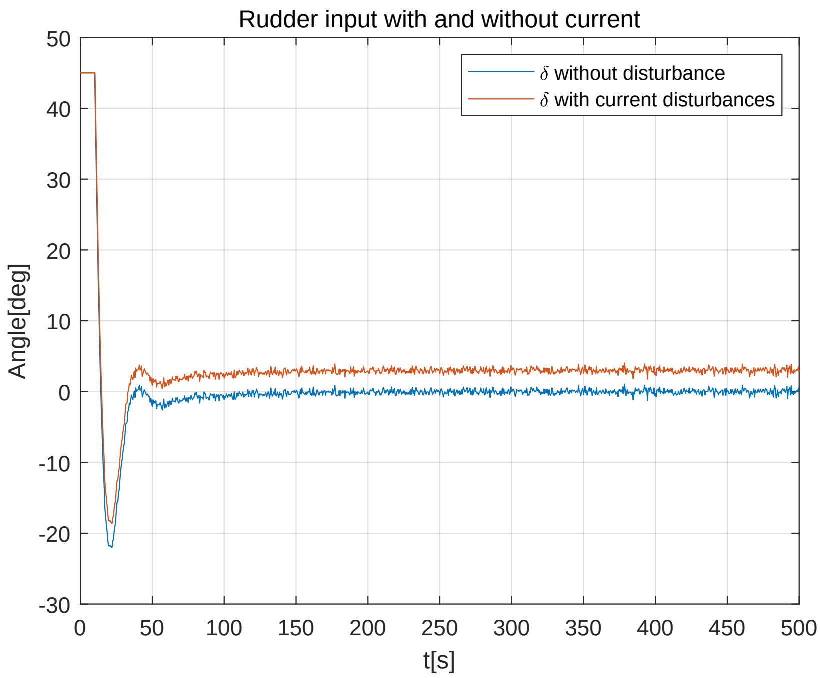

Figure 9: Rudder input without disturbance and with current disturbance

The rudder input with and without current disturbance is shown in Figure 9. The current disturbance contribute to rudder angle bias, nearly 3°.

4.4 Simulation with wave disturbance

Simulate the system with only wave disturbance. From Figure 9, it can be seen that the heading reaches the reference angle with slight oscillation. But rudder changes rapidly because of the high frequency component due to the wave disturbance. The oscillation can be overcome by applying a low pass filter, like a Kalman Filter in section 6.

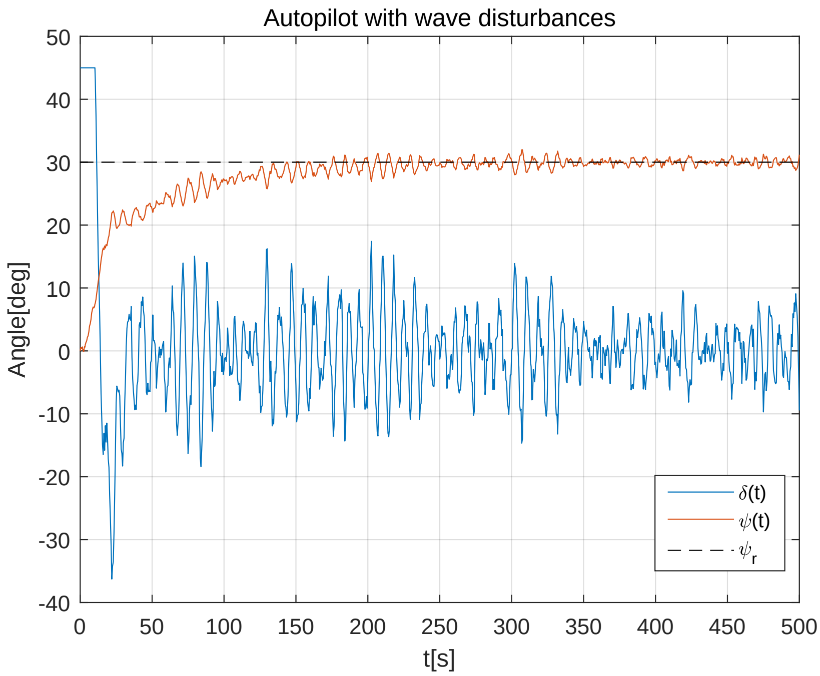

Figure 10: Autopilot with only wave disturbance

5 Observability

5.1 State space model

The state space model can be constructed through the series of equations below

$$\begin{aligned}
\dot{\xi}_w &= \xi_w \\
\dot{\psi}_w &= -\omega_0^2 \xi_w + 2\lambda\omega_0\psi_w + K_w w_w \\
\dot{\psi} &= r \\
\dot{r} &= -\frac{1}{T} r + \frac{K}{T}(\delta - b) \\
\dot{b} &= w_b \\
y &= \psi_w + \psi + v
\end{aligned} \tag{5.1}$$

The variable vector of state-space model is

$$\mathbf{x} = \begin{bmatrix} \xi_w \\ \psi_w \\ \psi \\ r \\ b \end{bmatrix} \tag{5.2}$$

Thus the state space model can be constructed with specified **A**, **B**, **E** and **C**

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w} \\
y &= \mathbf{C}\mathbf{x} + v
\end{aligned} \tag{5.3}$$

The full model is shown as

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}}_{\mathbf{B}} u + \underbrace{\begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{E}} \mathbf{w} \tag{5.4}$$

$$y = \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} + v \tag{5.5}$$

The observability matrix is defined as

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{5.6}$$

In MATLAB, the function `obsv(A,C)` is used to calculate the matrix. The system is observable if $\mathcal{O}$ is full rank.

5.2 Without disturbances

When there are no disturbances in the system, the state vector is reduced to

$$\mathbf{x} = \begin{bmatrix} \psi \\ r \end{bmatrix} \tag{5.7}$$

The matrices in state space model are shown as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad E = 0 \tag{5.8}$$

Thus the observability matrix is

$$\mathcal{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.9}$$

The matrix has $rank(\mathcal{O}) = 2$, which is full rank. Therefore, the system without disturbances is observable.

5.3 Current disturbance

In this case, only current disturbance is included in the system. the state vector is

$$\mathbf{x} = \begin{bmatrix} \psi \\ r \\ b \end{bmatrix} \tag{5.10}$$

The matrices in state space model are shown as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.11}$$

The observability matrix is

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.0138 & -0.0022 \end{bmatrix} \tag{5.12}$$

The matrix has $rank(\mathcal{O}) = 3$, which is full rank. Thus the system with current disturbance is observable.

5.4 Wave disturbance

In this problem only wave disturbance is included in the system. the state vector becomes

$$\mathbf{x} = \begin{bmatrix} \xi_w \\ \psi_w \\ \psi \\ r \end{bmatrix} \tag{5.13}$$

And the matrices in state space model are shown as

$$
\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{K}{T} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \end{bmatrix},
$$
$$
\mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ K_w \\ 0 \\ 0 \end{bmatrix} \tag{5.14}
$$

The observability matrix is calculated as

$$
\mathcal{O} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -0.6120 & -0.1238 & 0 & 1 \\ 0.757 & 0.5967 & 0 & -0.0138 \\ -0.3652 & -0.1496 & 0 & 0.0002 \end{bmatrix} \tag{5.15}
$$

The matrix has $rank(\mathcal{O}) = 4$, which is full rank. Thus the system with wave disturbance is observable.

5.5 Current and wave disturbance

For this part, both current and wave disturbances are take into account. The state space model is the same as (5.4). The observability matrix is

$$
\mathcal{O} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -0.6120 & -0.1238 & 0 & 1 & 0 \\ 0.0757 & -0.5967 & 0 & -0.0138 & -0.0022 \\ 0.3652 & 0.1496 & 0 & 0.0002 & 0 \\ -0.0915 & 0.3466 & 0 & 0 & 0 \end{bmatrix} \tag{5.16}
$$

The matrix has $rank(\mathcal{O}) = 5$, which is full rank. Thus the system with current and wave disturbances is observable. In this part, it has been proved that the system is observable in all cases. This means that for measurements, it is possible to determine the initial states as well as the behavior of the system just from the measured outputs and the input during the period of time. On the other hand, system's observability make it possible to implement estimator, which will be used in the next part of the assignment, to improve the performance of PD controller.

6 Discrete Kalman filter

6.1 Discretization

To improve the performance of model, a Kalman filter is implemented. By using Kalman filter, it is possible to filter out the disturbances and have good estimations of states.
For the first step, exact discretization is implemented based on the model from section 5.1. The state-space equations are transferred from continuous time into discrete time by MATLAB function `c2d`, with the sample time setting to be 0.1s, corresponding to sample frequency equals 10Hz.

$$\mathbf{A}_d = \begin{bmatrix} 0.9970 & 0.0993 & 0 & 0 & 0 \\ -0.0608 & 0.9847 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0999 & 0 \\ 0 & 0 & 0 & 0.9986 & -0.0002 \\ 0 & 0 & 0 & 0.0001 & 1 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.0002 \\ 0 \end{bmatrix},$$
$$\mathbf{C}_d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad D_d = 0, \quad \mathbf{E}_d = \begin{bmatrix} 0 & 0 \\ 0.0003 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1000 \end{bmatrix} \tag{6.1}$$

6.2 Estimate of the measurement noise variance

In order to find an estimation of the variance of the measurement noise, the MATLAB function `var` is used. To do this, the rudder input is set to zero while the compass is measured. Thus, only the measurement noise is recorded, and can be used to calculate the variance of the measurement noise R. From the calculation, the variance of the measurement noise is given as

$$\sigma^2 = 6.0666 \cdot 10^{-7} \tag{6.2}$$

As $E[v^2] = R$ equals the measurement noise variance divided by the sample interval, R is calculated as

$$R = \frac{\sigma^2}{T_{sample}} = 6.0666 \cdot 10^{-6} \tag{6.3}$$

6.3 Implementation of discrete Kalman filter

For the implementation of discrete Kalman filter, the following matrices are defined

$$\mathbf{P}_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-3} \end{bmatrix}, \quad \mathbf{x}_0^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$\mathbf{Q} = E[ww^T] = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix}, \quad R = 6.0666 \cdot 10^{-6} \tag{6.4}$$

In Simulink models, Matlab function is used to implement Kalman filter. The inputs for the Kalman filter are rudder and compass, which should be taken through *ZeroOrderHold* block, so that they are discreted before they are used in Kalman filter. Memory blocks are used for the outputs of the observor. One thing should be taken care is that the unit we used in the observor is *rad*, while out of the observor *deg* is used. Therefore, conversion between differnent units should be done.
The Kalman gain matrix is

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}) \tag{6.5}$$

Corrector equations are given by

$$\begin{aligned} \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{C}_K) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \end{aligned} \tag{6.6}$$

Predictor equations are given by

$$\begin{aligned} \mathbf{P}_{k+1} &= \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{E} \mathbf{Q} \mathbf{E}^T \\ \hat{\mathbf{x}}_{k+1} &= \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k \end{aligned} \tag{6.7}$$

where $\mathbf{y}_k$ is compass course measured, and $\mathbf{u}_k$ is rudder input. Through correction and prediction, recursive method, Kalman filter is able to find optimal estimated values of bias and heading, in the meantime filtering out most of noise. Firstly, the estimations are calsulated based on the measurement, states and Kalman filter gain. Then these estimations are corrected using covariance. The covariance is calculated and updated at every time step.

6.4 Feed forward from estimated bias

Under current disturbance, the autopilot have a better performance than the equivalent simulation.

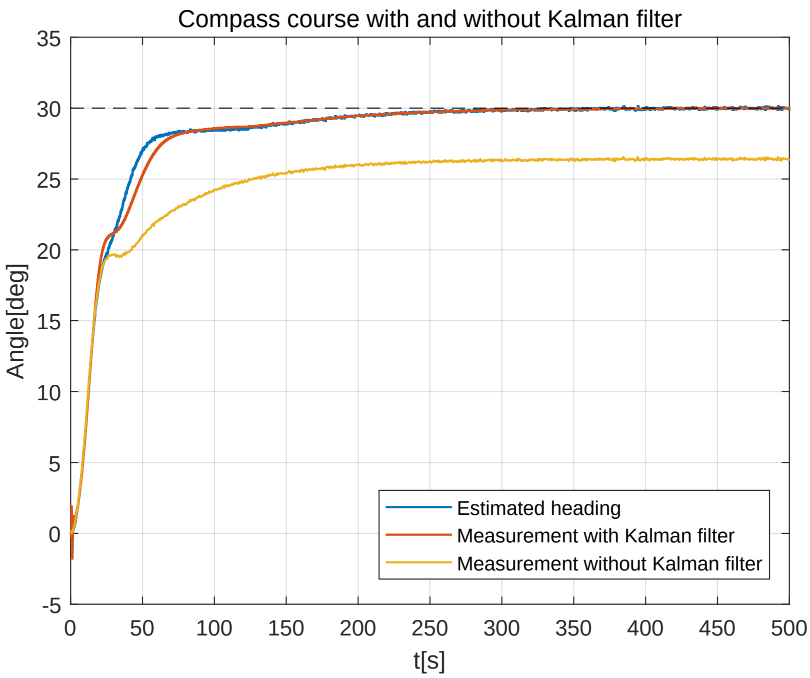

Figure 11: Compass course with and without Kalman filter

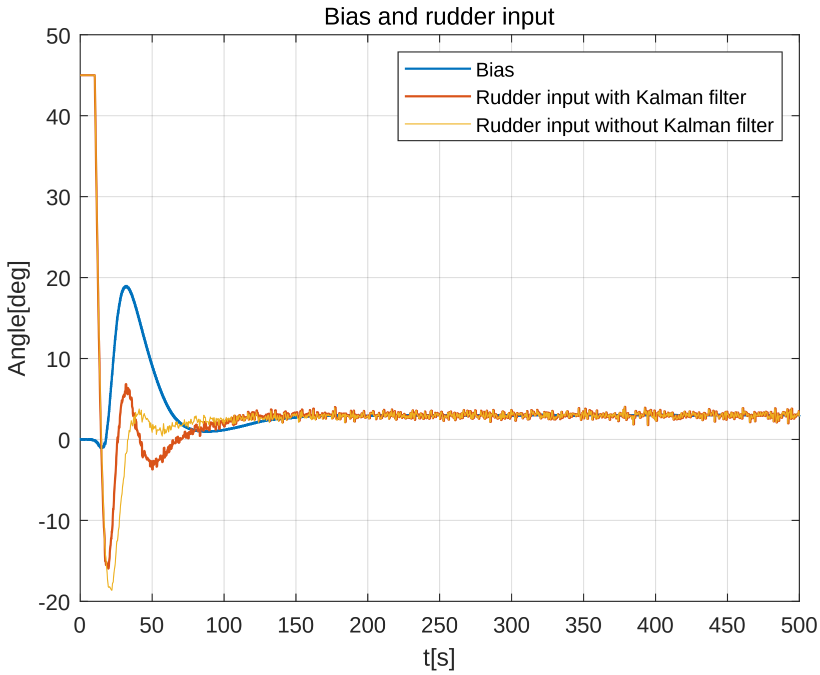

Figure 12: Bias and rudder input

The result in 4.3 shows that the autopilot has a stationary error. While in Figure 11, we can observe that the value of compass course is able to reach the reference heading value, and estimated heading is very close to the actual heading. The reason is that estimated bias generated by Kalman filter is accurate so that it cancels out the real rudder bias.

6.5 Wave filtering

The parameters of PD controller are slightly changed into

$$\begin{aligned} T_f &= 8.3901 \\ K_{pd} &= 0.6363 \end{aligned} \tag{6.8}$$

The reason is that the rudder input would have significant oscillations if original parameters are used. In order to decrease the oscillations that are harmful to the rudder machinery, a better controller is necessary. From Figure 13, the crossover frequency and phase margin satisfy the requirement given in the assignment. Therefore, in the following simulations, this PD controller is used.

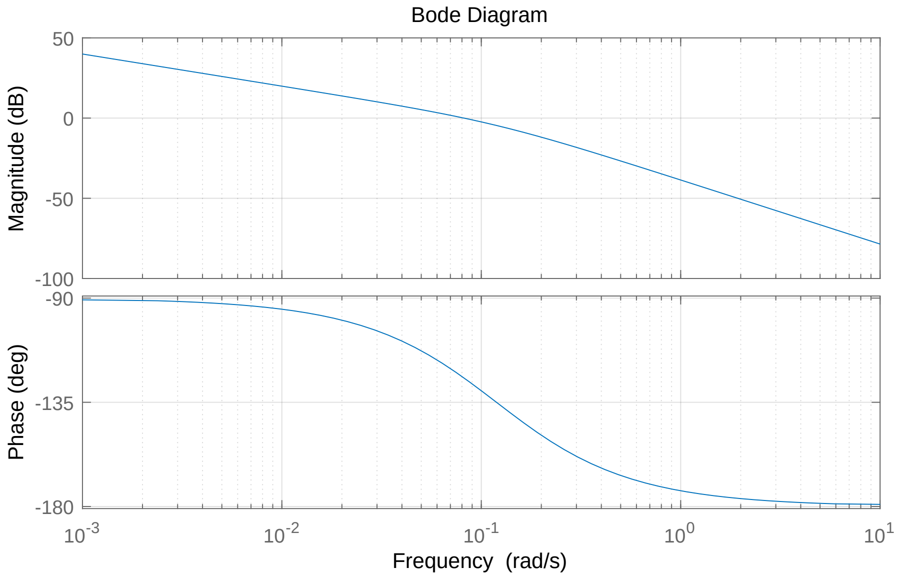

Figure 13: Bode diagram

As seen from Figure 14, it can be seen that estimated heading follows the measurement without excessive oscillations. The measurement noise and high frequency component of the wave disturbance is filtered out. Compared with the result in section 4.3 and 4.4, the autopilot is improved much better.

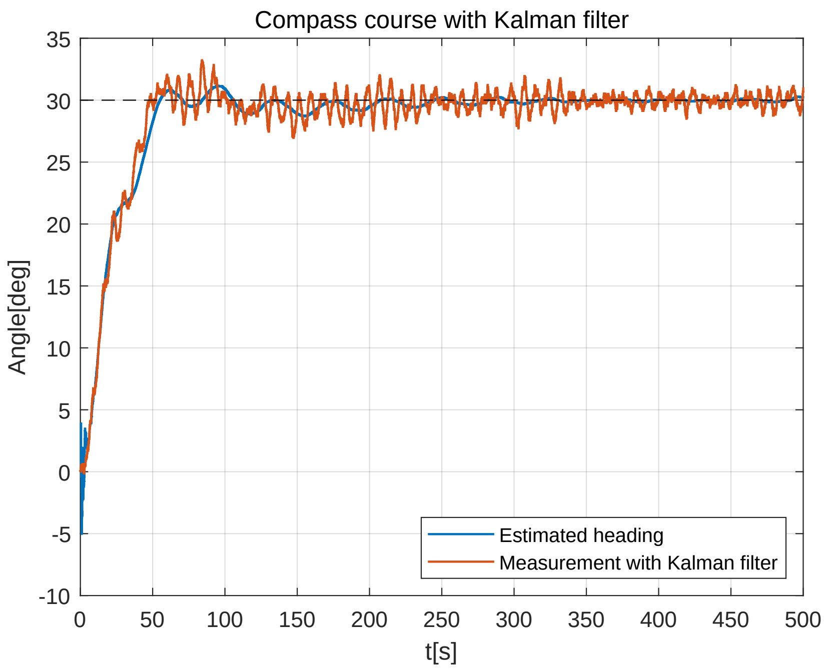

Figure 14: Compass course with Kalman filter

The comparision between the measurement with and without Kalman filter is shown in Figure 15. Without Kalman filter, the vessel reaches the reference angle at about 200s. while with Kalman filter, the vessel reaches that much faster, at 50s. After reaching the reference course, the vessel in both two simulations moves with a constant compass course equal to the reference course with small oscillation due to the wave.

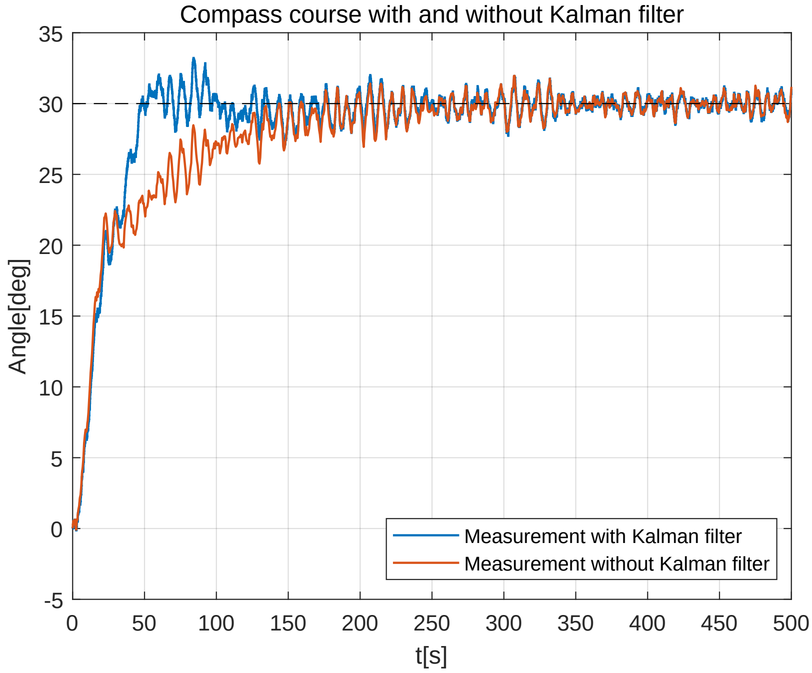

Figure 15: Compass course with and without Kalmen filter

Estimated bias and rudder input with Kalman filter and without Kalman filter are shown is Figure 16 and Figure 17 respectively, it is obvious the rudder input angle in the situation with filtered reduces more oscillation than that without filtered because the high frequency component is filtered out.

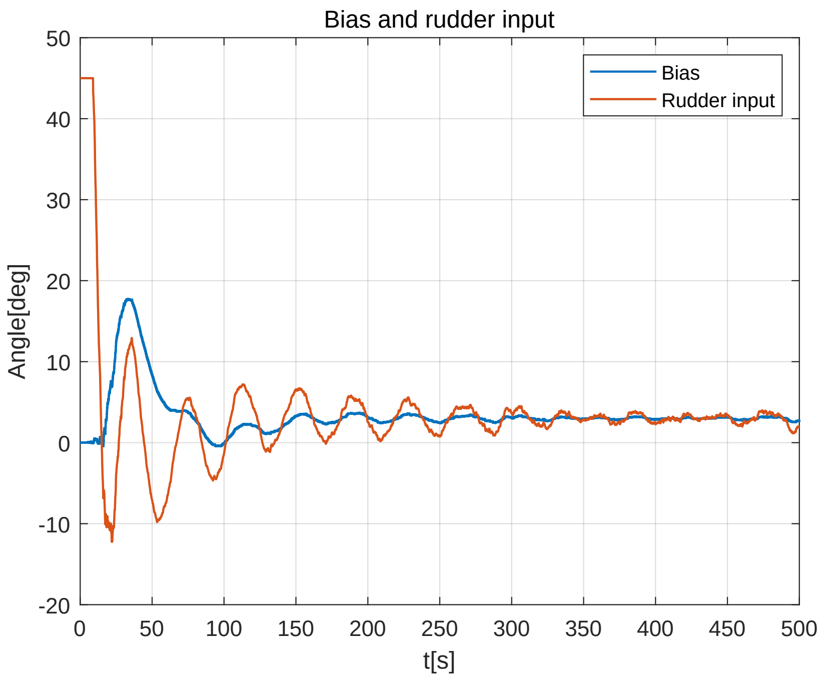

Figure 16: Bias and rudder input

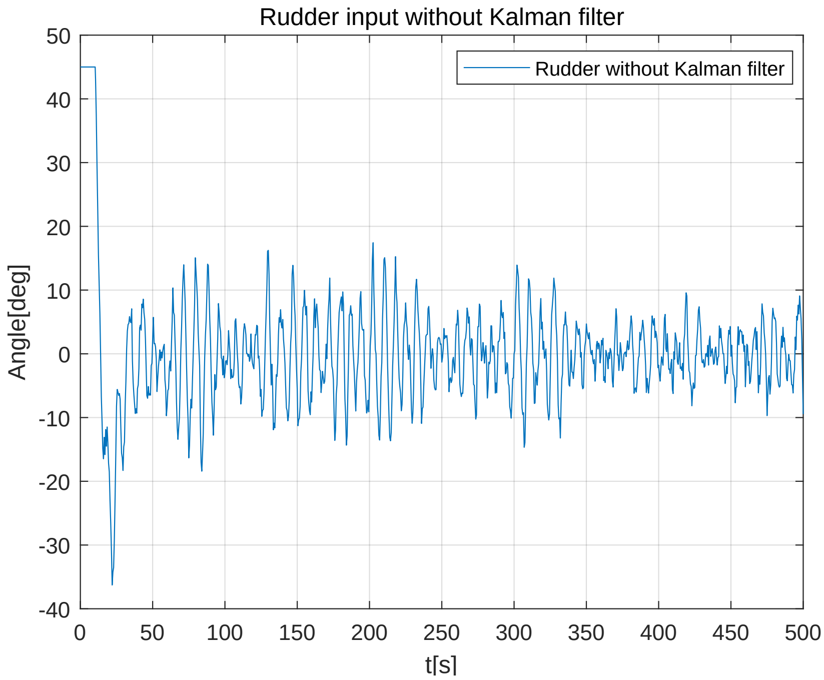

Figure 17: Rudder input without Kalman filter

From Figure 18, it can be seen that the estimated high frequency wave influence is similar to the actual wave influence, except the first 100s. Therefore, the estimator is aceptable.

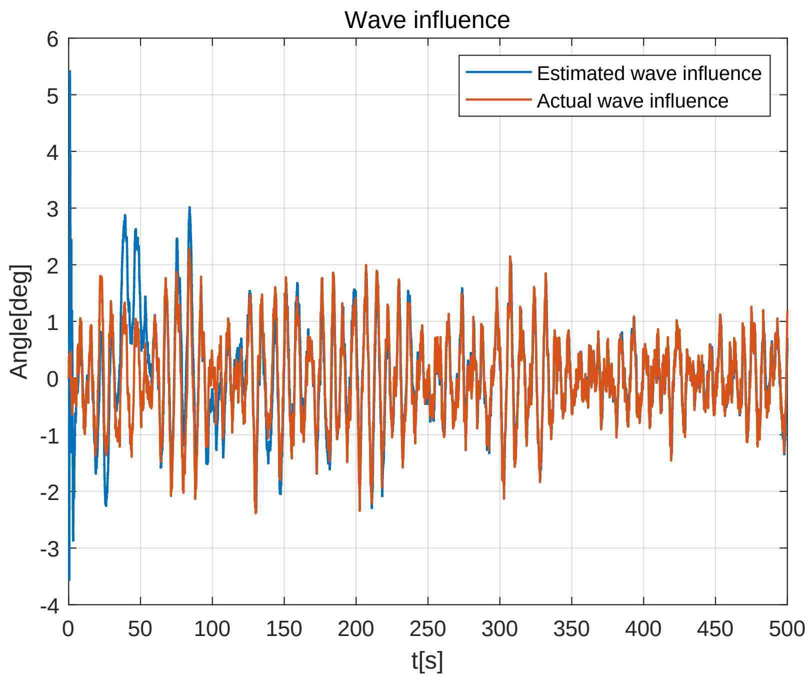

Figure 18: Wave influence

6.6 Response affection of Q matrix

In this task, $\mathbf{Q}$ is changed to be larger or smaller in order to investigate the affection of $\mathbf{Q}$. First we increase the values of $\mathbf{Q}$, which is given as

$$\mathbf{Q} = \begin{bmatrix} 300 & 0 \\ 0 & 10^{-2} \end{bmatrix} \tag{6.9}$$

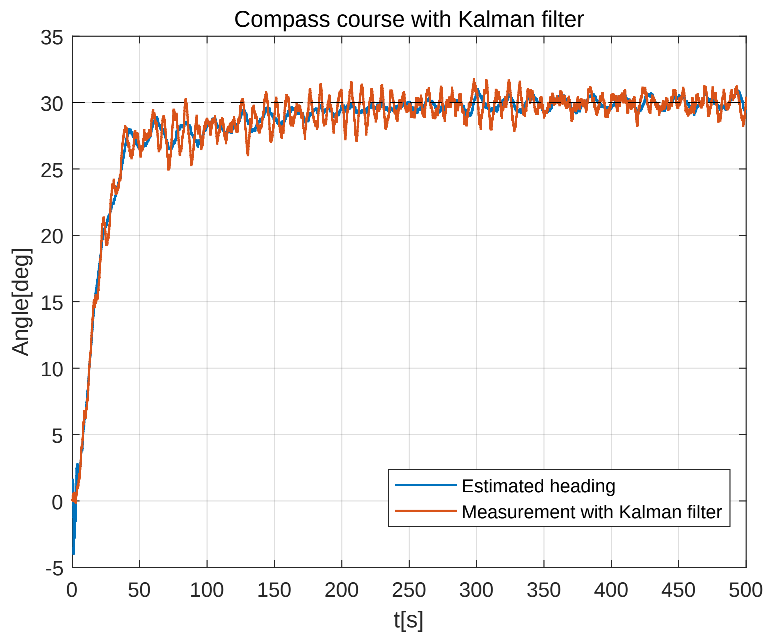

Figure 19: Compass course with Kalman filter with larger Q

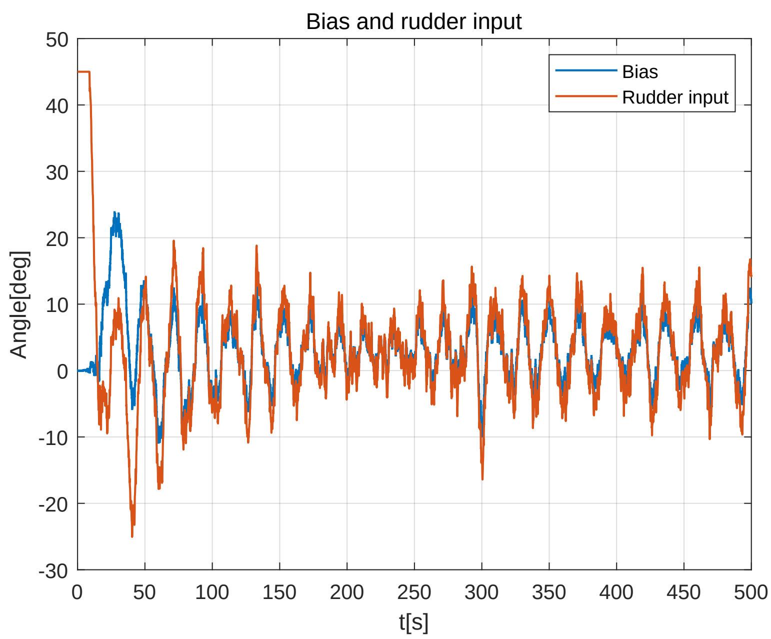

Figure 20: Bias and rudder input with larger Q

The smaller **Q** we use is

$$\mathbf{Q} = \begin{bmatrix} 3 & 0 \\ 0 & 10^{-10} \end{bmatrix} \tag{6.10}$$

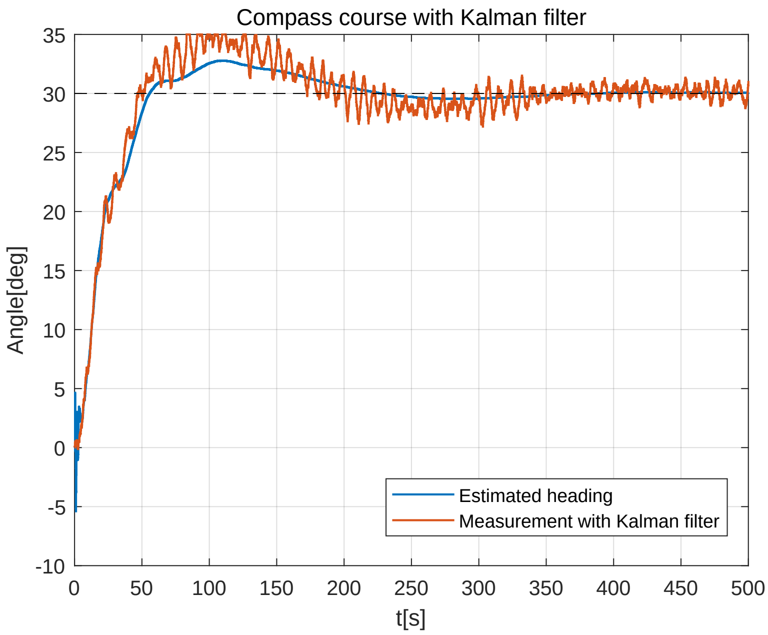

Figure 21: Compass course with Kalman filter with larger Q

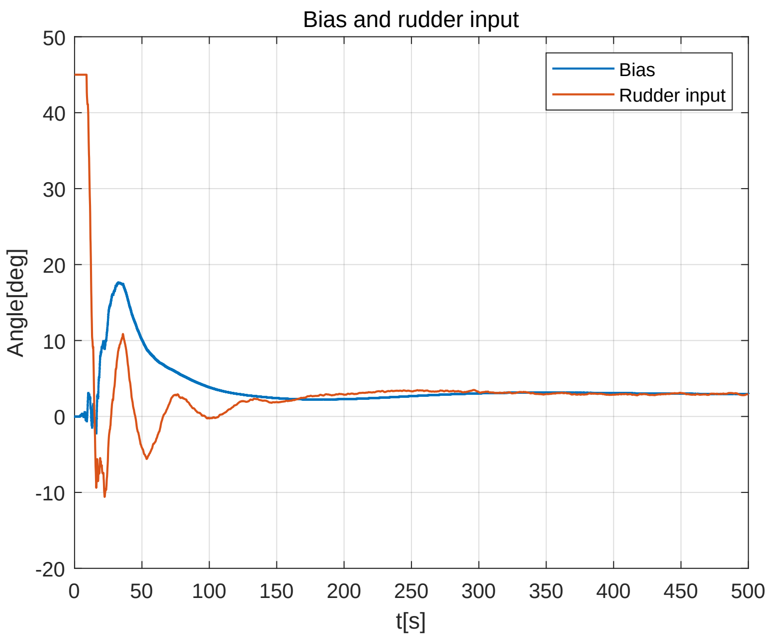

Figure 22: Bias and rudder input with larger Q

Generally speaking, the values of **Q** represent how much we trust the measurements. The larger the **Q** is, the more the Kalman filter rely on the measurements. Therefore, when increasing $\mathbf{Q}$, the estimated heading follows the compass closely. However, if the $\mathbf{Q}$ is too large, the estimation would include high-frequency component, which is shown as oscillations. As a result of this, the rudder input would oscillate as well, trying to change the heading quickly, which is harmful to the rudder machinery. On the other hand, the lower $\mathbf{Q}$ results in bad estimation. In Figure 21, the estimation has deviation from the compass, although the observor is able to filter out the wave influence perfectly. Because of filtering, the rudder input is smooth, which is a good feature. Lowering $\mathbf{Q}$ implies another way to produce smooth rudder input, besides changing the value of K_{pd}.

7 Conclusion

In this boat assignment, wave and current loads are added to the boat. By implementing a PD controller, the performance of the boat is tested under environmental loads and measurement noise. Also, discrete Kalman filter is designed to deal with high frequency disturbance from waves and thus to avoid rapid rudder angle change which can induce fast wear and tear. A feed forward loop canceled the current offset, and a filtered feedback smoothed the rudder operation.

8 Simulink Models

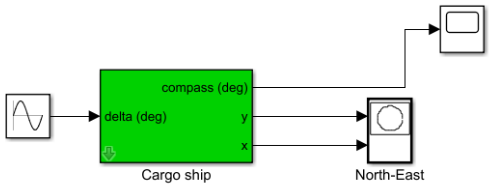

Figure 23: Simulink Model for section 2.2 and section 2.3

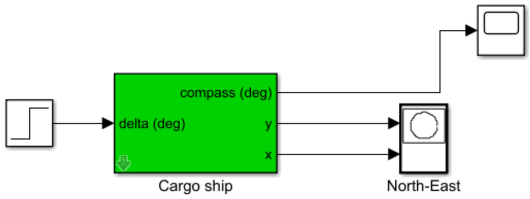

Figure 24: Simulink Model for section 2.4

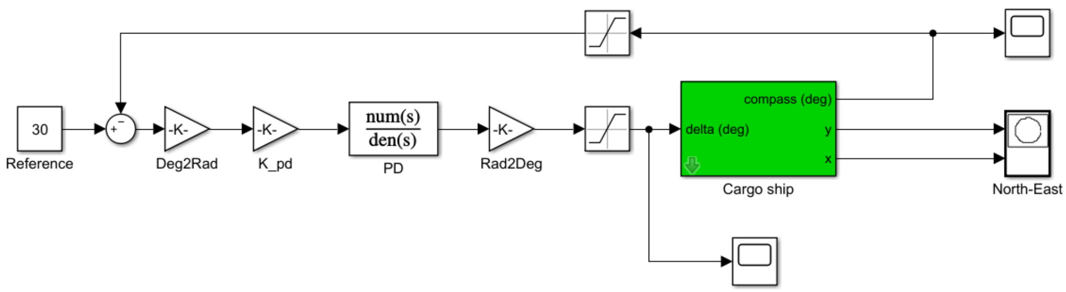

Figure 25: Simulink Model for section 4

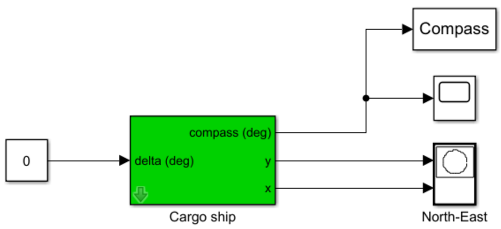

Figure 26: Simulink Model for section 6

9 MATLAB script

9.1 Part 1

```
load('omega0.005.mat')

% % % % omega=0.005
plot(omega1);
title('omega=0.005')
xlabel('Time(seconds)');
ylabel('Heading(deg)');

a=min(find(omega1.time>1000));
b=max(find(omega1.time<1500));

for i=1:1000
    x(i)=omega1.data(2007+i);
end

MIN=min(x);
MAX=max(omega1);
Amp=(MAX-MIN)/2;

% % % % omega=0.05
plot(omega2);
title('omega=0.05')
xlabel('Time(seconds)');
ylabel('Heading(deg)');

a=min(find(omega2.time>1000));
b=max(find(omega2.time<1500));

for i=1:1000
    x(i)=omega2.data(2007+i);
end

MIN=min(x);
MAX=max(omega2);
Amp=(MAX-MIN)/2;

% % % % with wave and noise
figure(1)
plot(dis_omega1);
title('omega=0.005 with disturbances');
ylabel('Heading(deg)');
figure(2)
plot(dis_omega2);
title('omega=0.05 with disturbances');
```

```
ylabel('Heading(deg)');

% % % % step response
K=0.174;
T=86.3;
syms s h;
h=K/(s*s*(s*T+1));
f=ilaplace(h);
ezplot(f,[0,2000]);
hold on;
plot(model,'r');
title('step responce of model and ship')
xlabel('Time(seconds)');
ylabel('Heading(deg)');
legend('ship','model');
axis([0,1000,0,200])
```

9.2 Part 2

```
load('wave.mat');

% parameters
fs=10;
window=4096;
noverlap=[];
nfft=[];

% fndd estimated PSD
[pxx,f]=pwelch(psi_w(2,:).*(pi/180),window,noverlap,nfft,fs
   );

% convert the outputs to the required units
pxx = pxx./(2*pi);
Omega = f.*2*pi;

% find maximun value and place of pxx
[PSDMax,place] = max(pxx);
Omega_0 = Omega(place);

% fitting curve
Sigma = sqrt(PSDMax);
Lambda = 1;

P_psi_w = @(Lambda,Omega)(4.*Lambda^2.*Omega_0^2.*Sigma^2.*
   Omega.^2)./(Omega.^4+Omega_0^4+2.*Omega_0^2.*Omega
   .^2.*(2.*Lambda^2-1));
x = lsqcurvefit(P_psi_w,Lambda,Omega,pxx);
P_psi = (4.*x^2.*Omega_0^2.*Sigma^2.*Omega.^2)./(Omega.^4+
   Omega_0^4+2.*Omega_0^2.*Omega.^2.*(2.*x^2-1));
```

```
%plot
plot(Omega,pxx)
hold on;
plot(Omega,P_psi);
ylabel('S_{\psi_\omega}(\omega),P_{\psi_\omega}(\omega)[rad
   ]');
xlabel('\omega[rad/s]');
legend('S_{\psi_\omega}(\omega)','P_{\psi_\omega}(\omega)')
   ;
title(['\lambda=',num2str(x)]);
axis([0 1.6 0 8e-4]);
grid on;
```

9.3 Part 3

```
load('wave.mat');

% Parameters of ship
T=72.4391;
K=0.1561;

% T_d cancles transfer time constant
T_d=T;

% Desired phase margine and crossover frequency;
w_c=0.1;
phi=50;
T_f=tan(40/180*pi)/w_c;
K_pd=w_c*abs(1+w_c*T_f*i)/K;
% K_pd=w_c*abs(1+w_c*T_f*i)/K-0.2;
num=[K*K_pd];
den=[T_f 1 0];
H=tf(num,den);
bode(H);
grid on;

sim('p5p3bx.mdl');
sim('p5p3cx.mdl');
sim('p5p3dx.mdl');

figure(1);
plot(b_Rudder);
hold on;
plot(b_Compass);
hold on;
plot(b_Reference,'--k');
hold on;
xlabel('t[s]');
```

```
ylabel('Angle[deg]');
grid on;
% ylabel('\psi_r(t),\psi(t),\delta(t)');
legend('\delta','\psi(t)','\psi_r(t)');
hleg=legend('\delta(t)','\psi(t)','\psi_r','Orientation','
   vertical');
set(hleg,'Position',[.75,.20,.14,.05]);
title('Autopilot with only measurement noise');

figure(2);
plot(c_Rudder);
hold on;
plot(c_Compass);
hold on;
plot(c_Reference,'--k');
hold on;
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
% ylabel('\psi_r(t),\psi(t),\delta(t)');
legend('\delta','\psi(t)','\psi_r(t)');
hleg=legend('\delta(t)','\psi(t)','\psi_r','Orientation','
   vertical');
set(hleg,'Position',[.75,.20,.14,.05]);
title('Autopilot with current disturbances');

figure(3);
plot(d_Rudder);
hold on;
plot(d_Compass);
hold on;
plot(d_Reference,'--k');
hold on;
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
% ylabel('\psi_r(t),\psi(t),\delta(t)');
legend('\delta','\psi(t)','\psi_r(t)');
hleg=legend('\delta(t)','\psi(t)','\psi_r','Orientation','
   vertical');
set(hleg,'Position',[.75,.20,.14,.05]);
title('Autopilot with wave disturbances');

figure(4);
plot(b_Rudder);
hold on;
plot(c_Rudder);
hold on;
```

```
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
% ylabel('\psi_r(t),\psi(t),\delta(t)');
legend('\delta without disturbance','\delta with current
   disturbances');
title('Rudder input with and without current');

figure(5);
plot(d_Rudder);
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('\delta with wave disturbances');
title('Rudder input with waves');
```

9.4 Part 4

```
% Parameters of ship
T = 72.4391;
K = 0.1561;
lam = 0.0791;
ome = 0.7823;

% part 4.2
A_2=[0 1;0 -1/T];
C_2=[1 0];
ob_2=obsv(A_2,C_2);
r_2=rank(ob_2);

% part 4.3
A_3=[0 1 0;0 -1/T -K/T;0 0 0];
C_3=[1 0 0];
ob_3=[C_3;C_3*A_3;C_3*A_3*A_3];
r_3=rank(ob_3);

% part 4.4
A_4=[0 1 0 0;-ome^2 -2*lam*ome 0 0;0 0 0 1;0 0 0 -1/T];
C_4=[0 1 1 0];
ob_4=[C_4;C_4*A_4;C_4*A_4*A_4;C_4*A_4*A_4*A_4];
r_4=rank(ob_4);

% part 4.5
A_5=[0 1 0 0 0;-ome^2 -2*lam*ome 0 0 0;0 0 0 1 0;0 0 0 -1/T
    -K/T;0 0 0 0 0];
C_5=[0 1 1 0 0];
ob_5=[C_5;C_5*A_5;C_5*A_5*A_5;C_5*A_5*A_5*A_5;C_5*A_5*A_5*
   A_5*A_5;];
r_5=rank(ob_5);
```

9.5 Part 5

5.a) - 5.d)

```
clear all;
clc;
close all;
load('wave.mat');
load('c_Compass');
load('c_Rudder');
load('d_Compass');
load('d_Rudder');

%% Parameters
T=72.4391;
K=0.1561;
omega_0=0.7823;
lamda=0.0791;
sigma=0.0281;
K_w=2*lamda*omega_0*sigma;

%% PD controller
% T_d cancles transfer time constant
T_d=T;

% Desired phase margine and crossover frequency;
w_c=0.1;
phi=50;
T_f=tan(40/180*pi)/w_c;
K_pd=w_c*abs(1+w_c*T_f*i)/K;

%% Discretization
% Matrices from Task 5.4.a
A=[0 1 0 0 0;-omega_0^2 -2*lamda*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];
D=0;
E=[0 0; K_w 0; 0 0; 0 0; 0 1];

% Sample time
F_s=10;
T_s=1/F_s;

% Discretization
[Ad,Bd]=c2d(A,B,T_s);
[Ad,Ed]=c2d(A,E,T_s);
Cd=C;
Dd=D;
```

```
%% Variance R & Q
% sim('VarCal.slx');
% R=var(Compass.data*pi/180)/T_s;
R=6.0666e-6;

% Q is the Process noise covariance
Q = [30 0; 0 10e-6];

% Initial value
P_0 = diag([1 0.013 pi^2 1 2.5e-3]);
x_0 = [zeros(5,1)];
I = eye(5);
data = struct('Ad',Ad,'Bd',Bd,'Cd',Cd,'Ed',Ed,'Q',Q,'R',R,'
   P_0',P_0,'x_0',x_0,'I',I);

sim('p5p5dx.mdl');

figure(1);
plot(bias,'LineWidth',1);
hold on;
plot(rudder,'LineWidth',1);
hold on;
plot(c_Rudder,'LineWidth',0.5);
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Bias','Rudder input with Kalman filter','Rudder
   input without Kalman filter');
title('Bias and rudder input');

figure(2);
plot(compass,'LineWidth',1);
hold on;
plot(psi_est,'LineWidth',1);
hold on;
plot(c_Compass,'LineWidth',1);
hold on;
plot(reference,'--k');
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Estimated heading','Measurement with Kalman filter'
   ,'Measurement without Kalman filter');
title('Compass course with and without Kalman filter');
```

5.e)

```
clear all;
clc;
close all;
load('wave.mat');
load('c_Compass');
load('c_Rudder');
load('d_Compass');
load('d_Rudder');

%% Parameters
T=72.4391;
K=0.1561;
% K=0.17;
omega_0=0.7823;
lamda=0.0791;
sigma=0.0281;
K_w=2*lamda*omega_0*sigma;

%% PD controller
% T_d cancles transfer time constant
T_d=T;

% Desired phase margine and crossover frequency;
w_c=0.1;
phi=50;
T_f=tan(40/180*pi)/w_c;
K_pd=w_c*abs(1+w_c*T_f*i)/K-0.2;
num=[K*K_pd];
den=[T_f 1 0];
H=tf(num,den);
bode(H);
grid on;

%% Discretization
% Matrices from Task 5.4.a
A=[0 1 0 0 0;-omega_0^2 -2*lamda*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];
D=0;
E=[0 0; K_w 0; 0 0; 0 0; 0 1];

% Sample time
F_s=10;
T_s=1/F_s;
```

```
% Discretization
[Ad,Bd]=c2d(A,B,T_s);
[Ad,Ed]=c2d(A,E,T_s);
Cd=C;
Dd=D;

%% Variance R & Q
% sim('VarCal.slx');
% R=var(Compass.data*pi/180)/T_s;
R=6.0666e-6;

% Q is the Process noise covariance
% Q = [30 0; 0 10e-6];

% Larger Q
Q = [300 0; 0 10e-2];
% Smaller Q
% Q = [3 0; 0 10e-10];

% Initial value
P_0 = diag([1 0.013 pi^2 1 2.5e-3]);
x_0 = [zeros(5,1)];
I = eye(5);
data = struct('Ad',Ad,'Bd',Bd,'Cd',Cd,'Ed',Ed,'Q',Q,'R',R,'
   P_0',P_0,'x_0',x_0,'I',I);

sim('p5p5ex.mdl');

figure(1);
plot(bias,'LineWidth',1);
hold on;
plot(rudder,'LineWidth',1);
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Bias','Rudder input');
title('Bias and rudder input');

figure(2);
plot(d_Rudder);
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Rudder without Kalman filter');
title('Rudder input wiouth Kalman filter');

figure(3);
plot(psi_est,'LineWidth',1);
```

```
hold on;
plot(compass,'LineWidth',1);
hold on;
plot(reference,'--k');
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Estimated heading','Measurement with Kalman filter'
   );
title('Compass course with Kalman filter');

figure(4);
plot(compass,'LineWidth',1);
hold on;
plot(d_Compass,'LineWidth',1);
hold on;
plot(reference,'--k');
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Measurement with Kalman filter','Measurement
   without Kalman filter');
title('Compass course with and without Kalman filter');

% Wave influence
for i=1:5005
    time(i)=psi_w(1,i);
    wave_inf(i)=psi_w(2,i);
end

figure(5);
plot(psi_w_est,'LineWidth',1);
hold on;
plot(time,wave_inf,'LineWidth',1);
xlabel('t[s]');
ylabel('Angle[deg]');
grid on;
legend('Estimated wave influence','Actual wave influence');
title('Wave influence');
```

References

[1] Chi-Tsong Chen, *Linear System Theory and Design*. Oxford University Press, international fourth edition, 2013.

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