TTK4115 Linear System Theory

Group 37

Boat Lab Report

Authors	*Student numbers*
Tonja Joseph	768455
Atussa Koushan	768632
Lone Bekkeheien	768706

January 8, 2018

Abstract

This is a report on the boat lab assignment in the course TTK4115 Linear System Theory. The purpose of this assignment is to use basic control theory to design an autopilot for a simulated ship.

Contents

1 Introduction

An autopilot for this ship is designed using an analytic model of the ship. Deriving parts of this model using basic identification techniques is a part of the assignment. For the autopilot, a Kalman filter is used to estimate the bias caused by disturbances in the system. It is also used for filtering waves. This report covers the results from the lab work and discusses the applied control system using theory from the curriculum of TTK4115 [6].

The report is divided in five main parts corresponding to the five parts of the assignment [1]. The control system and Kalman filter is implemented using MATLAB and Simulink. The Simulink models, plots and MATLAB scripts are in separate appendixes.

2 Description of the system

The state vector of the system is given by: $\boldsymbol{x} = [\xi_\omega \quad \psi_\omega \quad \psi \quad r \quad b]^T$. The following model is used to describe the behaviour of the given states:

$$\dot{\xi_\omega} = \psi_\omega \tag{1a}$$

$$\dot{\psi_\omega} = -\omega_0^2 \xi_\omega - 2\lambda\omega_0\psi_\omega + K_\omega \omega_\omega \tag{1b}$$

$$\dot{\psi} = r \tag{1c}$$

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

$$\dot{b} = \omega_b \tag{1e}$$

$$y = \psi + \psi_\omega + v \tag{1f}$$

The system can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u + \boldsymbol{E}\boldsymbol{w}, \quad y = \boldsymbol{C}\boldsymbol{x} + v \tag{2}$$

where $u = \delta$ and $\boldsymbol{w} = [w_w \quad w_b]^T$. An explanation of the different symbols can be found in table 1.

Table 1

Symbol	*Definition*
ψ	Average heading
ψ_ω	Heading component due to wave disturbance
$\dot{\xi}_\omega$	ψ_ω
r	Rotation velocity about the z-axis
b	Rudder angle bias
$\omega_\omega, \omega_b, v$	White noise processes
δ	Rudder input
T	Nomoto time constant
K	Nomoto gain constant

3 Part I - Identification of the boat parameters

3.1 Transfer function from δ to ψ

Assuming that there are no disturbances ($b = 0$) affecting the system, the transfer function $H(s)$ from δ to ψ is calculated using equations 1c and 1d.

$$
\begin{aligned}
\ddot{\psi} = \dot{r} &= -\frac{1}{T}\dot{\psi} + \frac{K}{T}(\delta - b) \\
\mathscr{L}\{\ddot{\psi}\} &= \mathscr{L}\{-\frac{1}{T}\dot{\psi} + \frac{K}{T}(\delta - b)\} \\
s^2\psi(s) &= -\frac{s}{T}\psi(s) + \frac{K}{T}\delta(s)
\end{aligned}
\tag{3}
$$

$$
H(s) = \frac{\psi(s)}{\delta(s)} = \frac{K}{s(Ts+1)} \tag{4}
$$

The transfer function $H(s)$ in equation 4 shows how the output ψ varies in response to the input variable δ, in the frequency domain. In this case the transfer function will make the amplitude of the output ψ smaller for higher frequencies as the input will be divided by s (which equals $j\omega$). The opposite will happen for low frequencies, similar to a low pass filter. This becomes clear in the next section, where the sine wave response of the ship is measured.

3.2 Boat parameters

The model contains two unidentified parameters. These are not explicitly given, and must be identified using basic identification techniques. In this part of the assignment, the value of these parameters are derived by observing the behaviour of the system when a sine wave is applied as input, and by observing how the system responds to different frequencies on this input. This is done in both an environment with low disturbance corresponding to smooth weather conditions, and in an environment with high amounts of disturbance.

In order to identify the parameters T and K in smooth weather conditions, all disturbances in the model are turned off. The input of the model is put to be a sine wave with amplitude 1 and frequencies equal to 0.005 rad/s and 0.05 rad/s. The resulting output signals are shown in figure 1. The amplitude of these output signals are denoted A_1 and A_2 and corresponds to $|H(s)|$ in each case. The measured values of A_1 and A_2 are given below:

$\omega_1 = 0.005$ results in $A_1 = 29.35$
$\omega_2 = 0.05$ results in $A_2 = 0.83$

Putting these values in equation 5, a set of equations is acquired where K and T are the only unknown parameters.

$$|H(s)| = |\frac{K}{j\omega(Tj\omega + 1)}| = \frac{|K|}{|j\omega||Tj\omega + 1|} = \frac{K}{T^2\omega^2 + 1} \tag{5}$$

Solving the equation set results in the following values for K and T:

$$\text{K} = 0.16 \quad \text{T} = 72.52 \text{ s} \tag{6}$$

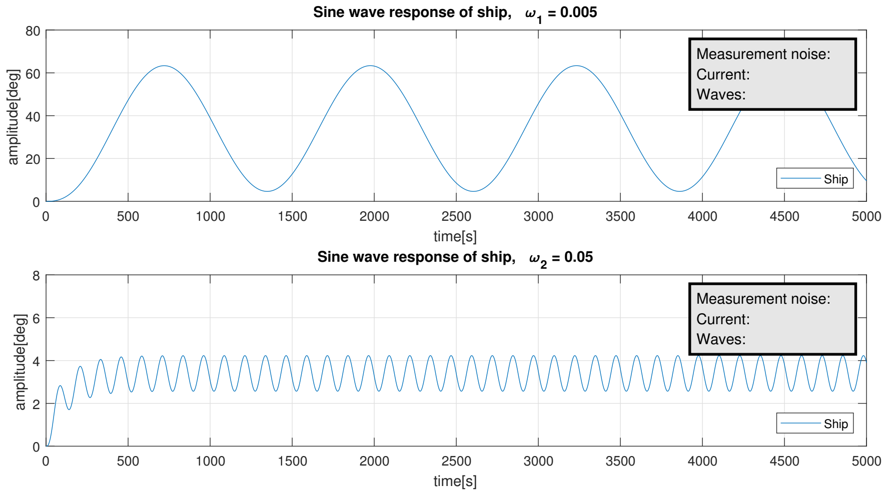

Figure 1: Simulation of output for w_1 and w_2.

3.3 Waves and measurement noise

Good estimates of the parameters are harder to find when the system is affected by waves and measurement noise, wich are both applied in this part of the assignment. However, looking at the plots in figure 2 the disturbances has a greater influence on the response of the system when the frequency of the input is increased. With a frequency equal to ω_1, relatively good estimates are still possible to derive. However, increasing the frequency by a factor of 10, makes the estimations remarkably poor. This can be explained by considering the physical system represented by this model. If the rudder of a ship varies back and forth rapidly, the ship would not have enough time to react to each small change in input. Thus, the disturbances would become more dominant. If the rudder changes slower and with higher amplitude, the ship would react easier and the reactions provoked by the disturbances would be small in comparison and thus less dominant.

This is the most important point expressed by the plots in figure 2. Using the same method as in the last section, the new values for the amplitudes and parameters K and T are found to be:

$$A_1 = 28.95$$
$$A_2 = 0.55$$

$$K = 0.17 \quad T = 121.86 \text{ s} \tag{7}$$

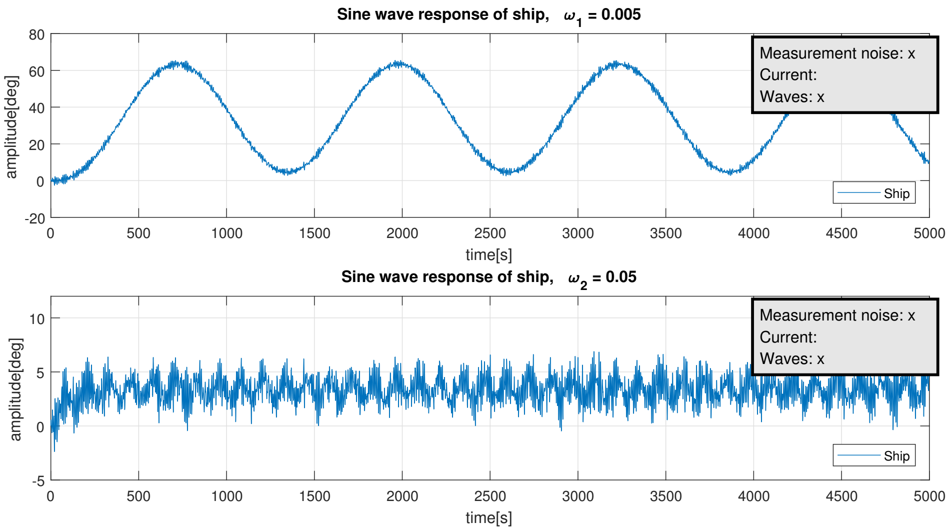

Figure 2: Simulation of output for w_1 and w_2.

3.4 Step response

The purpose of this part, is to compare the model derived in section 3.1 to the actual behaviour of the ship. To do this the model is implemented in Simulink. The implementation is seen in figure 16. A step input of 1 degree is applied to the rudder at $t = 0$ for both the ship and the model from section 3.1. The responses are shown in figure 3, where the the blue line is the step response of the ship and the red line is the step response of the model. The model is a good approximation of the ship up until around $t = 681$ seconds. After this point a linearly increasing error occurs. This implies that the deviations are minimal the first 10 minutes after the change in input. This is a relatively long time span, and the approximation of the model is concluded to be good.

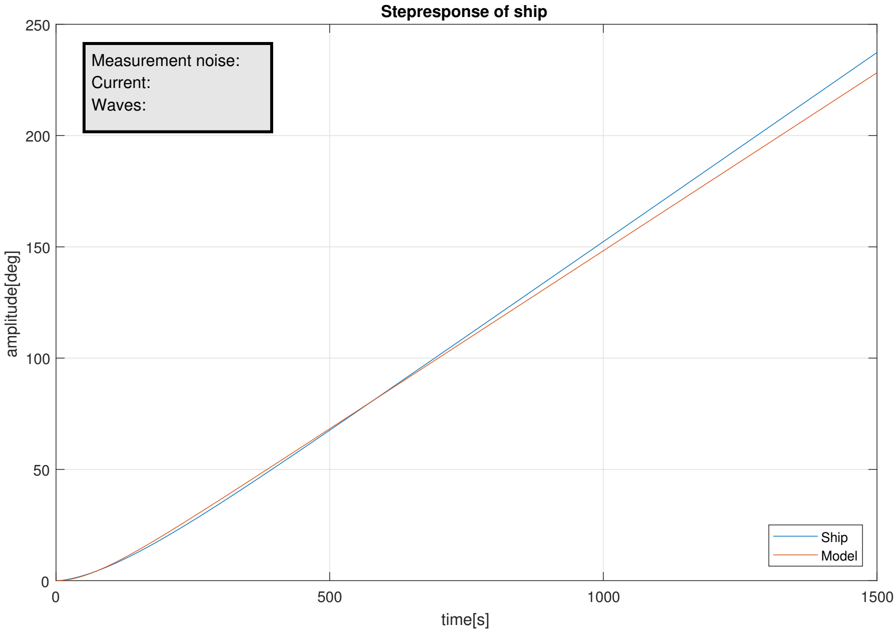

Figure 3: Step response of ship and model.

4 Part II - Identification of wave spectrum model

4.1 Power density spectrum

An estimate of the Power Spectral Density is found using the MATLAB function `pwelch`, showed in section 9.2 in appendix A. Because `fs` is given in Hz, the scaling factors $\frac{1}{2\pi}$ and 2π are applied to the outputs to convert `pxx` ($S_{\psi_\omega}(\omega)$) to power s/rad and `f` (ω) to rad/s.
PSD shows the strengths of the variations(energy) as a function of frequency. In other words, it shows at which frequencies variations are strong and at which frequencies variations are weak.[5] A plot of the PSD estimate is shown in figure 4. This plot indicates that the PSD is approxiamatly normal distributed with expected value equal to about 0.8. This means that ψ_ω has strongest variations at frequency approximately 0.8. Therefore the peak of the PSD is also known as resonance frequency, which is the frequency where the system oscillates with the highest amplitude.[2] A more exact resonance frequency is found in section 4.3.

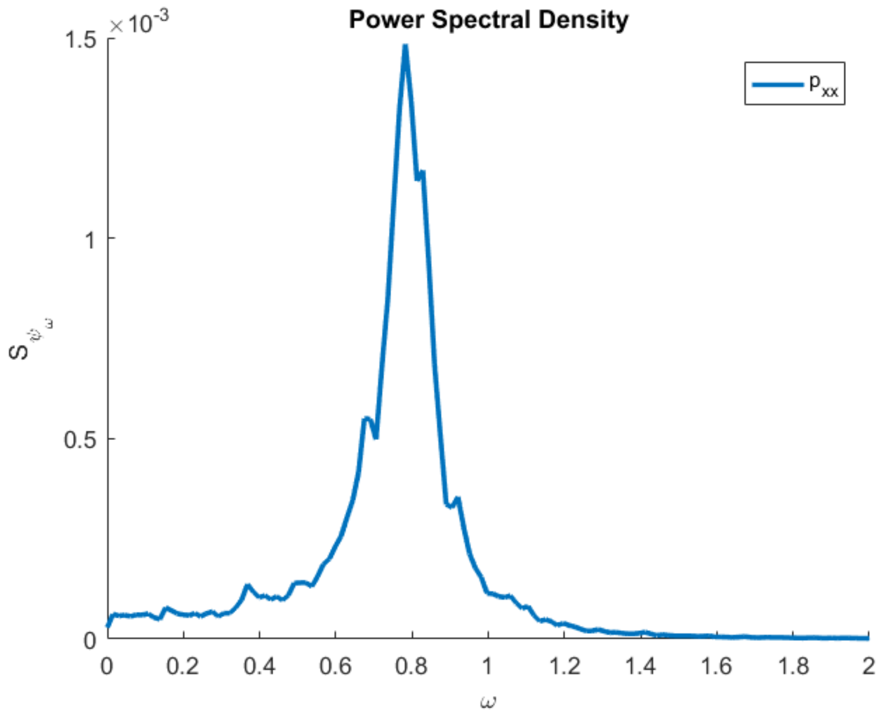

Figure 4: PSD estimate, S_{ψ_ω}

4.2 Transferfunction from ω_ω to ψ_ω

The system equations 1a and 1b are used to find the transfer function from ω_ω to ψ_ω. Taking the Laplace of these equations gives the following:

$$\xi_\omega = \frac{\psi_\omega}{s} \tag{8}$$

$$s\psi_\omega = -\omega_0^2\xi_\omega - 2\lambda\omega_0\psi_\omega + K_\omega\omega_\omega \tag{9}$$

Combining these equations gives the desired transfer function:

$$G(s) = \frac{\psi_\omega}{\omega_\omega} = \frac{K_\omega s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \tag{10}$$

In order to find the power density function of the output, ψ_ω, the following relation is used [5]

$$P_{\psi_\omega}(jw) = |G(jw)|^2 P_{\omega_\omega}(jw) \tag{11}$$

The input is white noise (ω_ω), and the PSD of white noise is constant and equal to the variance. In the lab assigment it is given that the variance of the white noise is equal to 1. $P_{\omega_\omega} = 1$ due to this.

$$P_{\psi_\omega} = |G(jw)|^2 = G(jw)G^*(jw) = \frac{\omega^2 K_\omega^2}{\omega^4 + 2\omega^2\omega_0^2(2\lambda^2 - 1) + \omega_0^4} \tag{12}$$

4.3 Resonance frequency, ω_0

As mentioned earlier, the resonance frequency ω_0 is the frequency where the system oscillates with the highest amplitude. ω_0 must therefore bew the vertex of the power density function found in section 4.1. By zooming in on the plot in figure 4, it is found that:

$$\omega_0 = 0.78 rad/s = 0.124 Hz \tag{13}$$

4.4 Damping factor λ

The damping factor λ in equation 10 is still unknown. The value is determined by adjusting the analytic model of the PSD to fit the estimate, and defining $K_\omega = 2\lambda\omega_0\sigma$. Here $\sigma^2 = S_{\psi_\omega}(\omega_0) = 1.49 \cdot 10^{-3}$. The best value for λ is found by testing many different values, as shown in figure 5. $\lambda = 0.08$ gives the curve closest to the estimate.

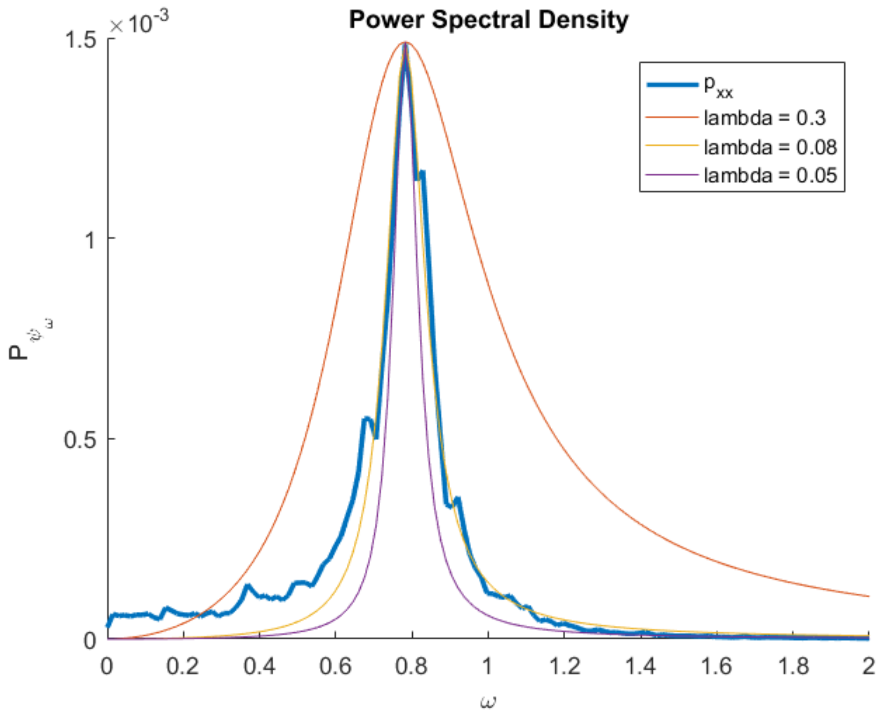

Figure 5

5 Part III - Control System Design

In this section the goal is designing an autopilot for the ship. This means that the autopilot should compute the input to the rudder such that the ship follows the desired course. The reference ψ_r is set to be 30 degrees throughout this section.

5.1 PD controller

A PD-controller is designed using the transfer function (4). T_d is chosen to be equal to T in (6) such that the transfer function time constant is cancelled. This gives the following transfer function for the open loop system.

$$H_0(s) = H_{pd}(s)H_{ship}(s) = \frac{K_{pd}(1+T_d s)}{1+T_f s}\frac{K}{(1+Ts)s} = \frac{KK_{pd}}{(1+T_f s)s} \tag{14}$$

T_f and K_{pd} are obtained by using the desired phase margin $\phi = 50$ degrees, the cross frequency $\omega_c = 0.10$ rad/s and K in (6).

$$
\begin{aligned}
|H_0(j\omega_0)| &= 1 \\
\left|\frac{K_{pd}K}{j\omega_c - T_f\omega_c^2}\right| &= 1 \\
K_{pd} &= \frac{\sqrt{\omega_c^2 + T_f^2\omega_c^4}}{K}
\end{aligned} \tag{15a}
$$

$$
\begin{aligned}
\phi\frac{\pi}{180} &= \pi + \angle H_0 = \pi + \angle\frac{K_{pd}K}{(1 + T_f j\omega_c)j\omega_c} \\
&= \pi + \angle(K_{pd}K) - \angle(\omega_c j - T_f\omega_c^2) \\
&= \pi - \tan^{-1}(\frac{\omega_c}{-\omega_c^2 T_f})
\end{aligned} \tag{15b}
$$

Solving equation set 15 for T_f and K_{pd} gives the following values:

$$
T_f = 8.65 \qquad K_{pd} = 0.816 \tag{16}
$$

From the bode plot in figure 6 it is verified that $\omega_c = 0.1$ rad/s and $\phi = 50$ degrees. It can also be seen that increasing K_{pd} would increase the band with and the cross frequency. However it could also lead to overshoot in the system. A larger K_{pd} would in addition result in a smaller phase margin. It is generally found that phase margins between 30 and 60 degrees result in reasonable stability [3], and too low phase margins result in unstable systems. In any way, the obtained constants - based on the ω_c and phase margin given in the assignment - will be used throughout the rest of the report.

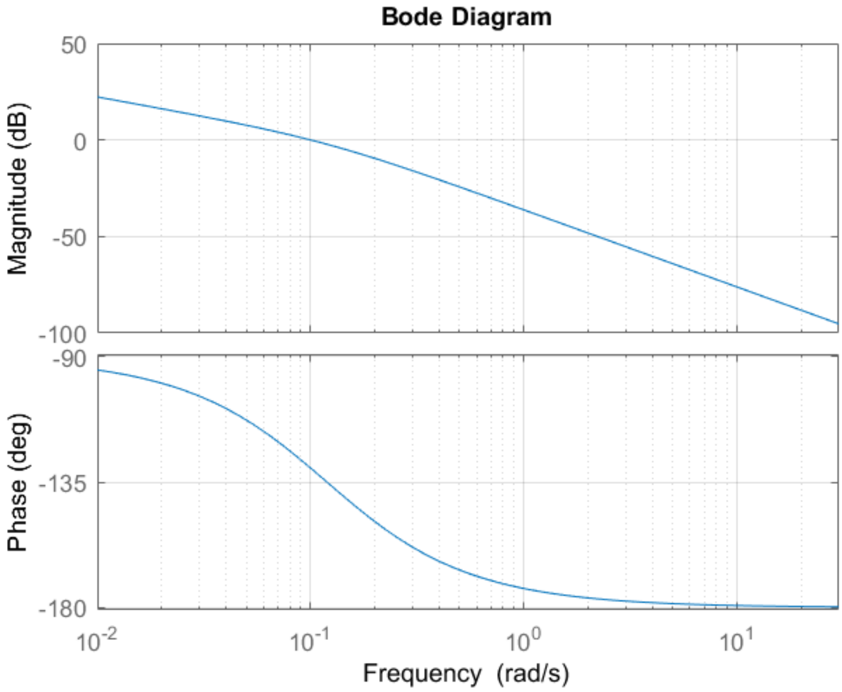

Figure 6

5.2 Simulation without disturbance

In the upper part of figure 7 the compass course ψ is plotted together with the reference ψ_r, and in the lower part the rudder input is plotted. The plot shows the step response of the system with PD-regulation. The response reaches the reference efficiently and the autopilot seems to be working. This plot shows how the direction quickly reaches a state where it is easy to control with only small movements of the rudder around zero.

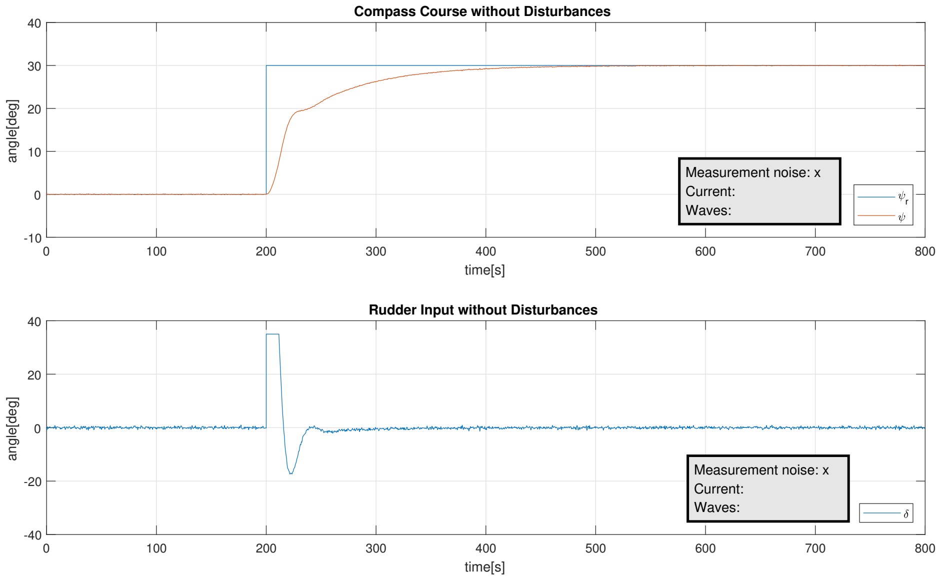

Figure 7: Impulse response of the compass course when simulated without disturbance(upper) and rudder angle(lower)

5.3 Simulation with current disturbance

When the system is affected by a current disturbance, a stationary deviation appears as shown in the upper part of figure 8. The derivative effect does not have the capability to compensate for the stationary deviation caused by the current. To remove this error, integral effect or Kalman filter must be applied. The rudder input, showed in the lower part of figure 8, is this time varying around a higher value. This represents an attempt to compensate for the current disturbance. However, this error is hard to remove with this type of controller.

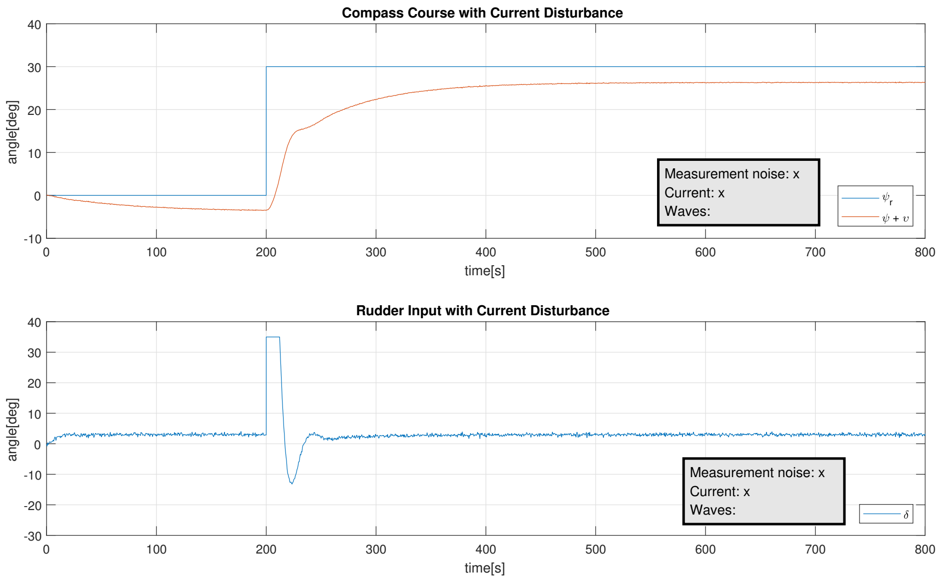

Figure 8: Impulse response of the compass course and rudder input angle when simulated with current disturbance

5.4 Simulation with wave disturbance

This time the disturbance is not constant, but rather variating as it is caused by waves. In this case, there is no longer a stationary deviation. The rudder input recognizes the disturbances and tries to actively compensate for the deviations that appear. This leads to the behaviour plotted in figure 9. It is seen that it varies from around -20 to 20 degrees very rapidly. This behaviour would probably cause damage to the ship if it is even able to change the rudder so quickly. Because the waves influence the ship approximately the same from both sides, it would be better to ignore them to some degree. Later in the assignment a Kalman filter will be used to filter the wave disturbance and give a smoother input to the rudder.

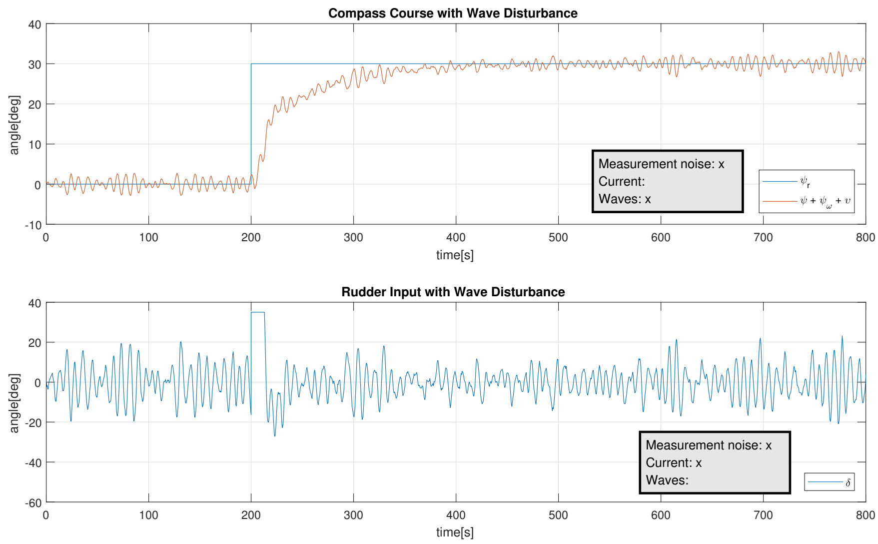

Figure 9: Impulse response of the compass course(upper) and rudder input angle(lower) when simulated with wave disturbance

6 Part IV - Observability

In this part of the assignment the observability of the system is determined considering the system in the four following cases: No disturbance, only current disturbance, only wave disturbance, both current and wave disturbance.

The observability matrices are computed using MATLAB. See appendix A.

6.1 Computation of the state space model

The matrices for the model are computed from the differential equations in (1) and the system equtions in 2. The following system matrices are obtained.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\omega_0\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1/T & K/T \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ K/T \\ 0 \end{bmatrix}$$
$$\boldsymbol{C} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T \qquad \boldsymbol{E} = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

Where $K_\omega = 2\lambda\omega_0\sigma$.

6.2 No disturbance

Without any disturbances we have that $\boldsymbol{w} = \begin{bmatrix}\omega_w \ \omega_b\end{bmatrix} = \begin{bmatrix}0 \ 0\end{bmatrix}$. This reduces the number of states in the state vector $\boldsymbol{x}$. The new state vector is a fragment of the original state vector as seen in (18). The corresponding fragments of A,B and C gives the system shown in (19). The last part of the differential state equation containing $\boldsymbol{w}$ is removed due to no disturbance.

$$\boldsymbol{x} = \begin{bmatrix} \psi \\ r \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \tag{18}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 \ 0 \end{bmatrix} \boldsymbol{x} + v \tag{19}$$

The observability of this system is determined by computing the observability matrix. The result is given in (20). Full rank in the observability matrix implies an observable system when all disturbances are removed. This is expected as the unmeasured parameter r is easily derived from the measured ψ using (1c).

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} C \\ AC \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

6.3 Current disturbance

In this case the current disturbance is included in the system such that $\boldsymbol{w} = \begin{bmatrix} 0 \; \omega_b \end{bmatrix}$. The new state vector and corresponding system is given in (21) and (22) respectively. The observability matrix in (23) has full rank. The system is therefore observable also in this case. This is expected despite having only one measurement, as the A matrix in the system in (22) shows a relation between r and b. Thus, b can be estimated using the estimated r.

$$\boldsymbol{x} = \begin{bmatrix} \psi \\ r \\ b \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} \tag{21}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/T & -K/T \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{x} + v \tag{22}$$

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} C \\ AC \\ AC^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.014 & -0.002 \end{bmatrix} \tag{23}$$

6.4 Wave disturbance

In this case only the wave disturbance is applied to the system. The state space model is as follows

$$\boldsymbol{x} = \begin{bmatrix} \varepsilon_\omega \\ \psi_\omega \\ \psi \\ r \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{24}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \end{bmatrix} u, \qquad y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}^T \boldsymbol{x} + v \tag{25}$$

This results in the following obsevability matrix

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} C \\ AC \\ AC^2 \\ AC^3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -0.608 & -0.125 & 0 & 1 \\ 0.076 & -0.593 & 0 & -0.014 \\ 0.361 & 0.150 & 0 & 0 \end{bmatrix} \tag{26}$$

The observability matrix has full column rank, and the system is therefore observable in this case as well.

6.5 Current and wave disturbance

Finally adding the wave disturbance together with the disturbance from current gives the disturbance matrix $\boldsymbol{w} = \begin{bmatrix} \omega_w & \omega_b \end{bmatrix}$. $\boldsymbol{b}$ remains left out

Checking the observability for both current and wave disturbance is done by using the matrices in 17. This results in

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} C \\ AC \\ AC^2 \\ AC^3 \\ AC^4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -0.608 & -0.125 & 0 & 1 & 0 \\ 0.076 & -0.593 & 0 & -0.014 & -0.002 \\ 0.361 & 0.150 & 0 & 0 & 0 \\ -0.091 & 0.342 & 0 & 0 & 0 \end{bmatrix} \tag{27}$$

Which has full column rank and the system is therefore observable.
The fact that the system is observable in all cases means that it is possible to estimate the different states, which is really useful for implementing the Kalman filter.

7 Part V - Discrete Kalman Filter

In this section a discrete Kalman filter will be implemented to estimate the bias b, the heading ψ and the high-frequency wave induced motion on the heading ψ_ω.

7.1 Discretization

The sample frequency $f_s = 10$ Hz, which means that the sample time $T_s = 1/f_s = 0.1$ s. Discretization [4] of the state space model found in section 6.1 using exact discretization results in

$$\begin{aligned} \dot{\boldsymbol{x}} = \frac{\boldsymbol{x_{k+1}} - \boldsymbol{x_k}}{T_s} &= \boldsymbol{A}\boldsymbol{x_k} + \boldsymbol{B}u_k + \boldsymbol{E}\boldsymbol{w_k} \\ \boldsymbol{x_{k+1}} &= \boldsymbol{x_k}(\boldsymbol{A}T_s + 1) + \boldsymbol{B}T_s u_k + \boldsymbol{E}T_s\boldsymbol{w_k} \end{aligned} \tag{28a}$$

$$y_k = \boldsymbol{C}\boldsymbol{x_k} + v \tag{28b}$$

$$\boldsymbol{x_{k+1}} = \boldsymbol{A_d}\boldsymbol{x_k} + \boldsymbol{B_d}u_k + \boldsymbol{E_d}\boldsymbol{w_k} \tag{29a}$$

$$y_k = \boldsymbol{C_d}\boldsymbol{x_k} + \boldsymbol{D_d}\boldsymbol{u_k} + v \tag{29b}$$

From this it follows that $\boldsymbol{A_d} = \boldsymbol{A}T_s + 1$, $\boldsymbol{B_d} = \boldsymbol{B}T_s$, $\boldsymbol{E_d} = \boldsymbol{E}T_s$, $\boldsymbol{C_d} = \boldsymbol{C}$ and $\boldsymbol{D_d} = \boldsymbol{D} = 0$.

$$\begin{aligned} \boldsymbol{A_d} &= \begin{bmatrix} 1 & T_s & 0 & 0 & 0 \\ -\omega_0^2 T_s & -2\lambda\omega_0 T_s + 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & 0 & -T_s/T & -KT_s/T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ -0.0608 & 0.9875 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.1 & 0 \\ 0 & 0 & 0 & 0.999 & 0.0002 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{30a}$$

$$\boldsymbol{B_d} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ KT_s/T \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.0002 \\ 0 \end{bmatrix} \tag{30b}$$

$$\boldsymbol{E_d} = \begin{bmatrix} 0 & 0 \\ K_\omega T_s & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & T_s \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0.00048 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{30c}$$

$$C_d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{30d}$$

7.2 Variance of measurement noise

The measurement noise is measured by sending zero input as the rudder input to the ship and removing the wave- and current disturbance. It is then imported to MATLAB as a vector. An estimate of the variance of the measurement noise is found using MATLAB function `var`. The measurement noise vector is given in degrees and is multiplied by $\pi/180$ to obtain radians. The result is

$$\sigma^2 = 6.0657 \cdot 10^{-7} \tag{31}$$

The MATLAB code for this part can be found in section 9.5.

7.3 Implementation of discrete Kamlan filter

The discrete Kalman filter is implemented using MATLAB and Simulink.

In the rest of the assignment the following detonations are used:
$\hat{x}^-[k]$: A priori state estimate
$\hat{x}[k]$: A posteriori state estimate
The same standard is used for the error covariace P.

The first step in implementing the Kalman filter is to initialize the filter using equation 33 and 32 given below.

$$\boldsymbol{P}^-[0] = E[(\mathbb{x}[0] - \boldsymbol{m}_{\boldsymbol{x}_0})(\mathbb{x}[0] - \boldsymbol{m}_{\boldsymbol{x}_0})^T] = \boldsymbol{\zeta}_{\boldsymbol{x}_0} \tag{32}$$

$$\hat{\boldsymbol{x}}^-[0] = E[\mathbb{x}(0)] = \boldsymbol{m}_{\boldsymbol{x}_0} \tag{33}$$

The initial a priori state estimate and error covariance is given in the lab assignment, as shown below.

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

$$\boldsymbol{P}_{\boldsymbol{0}}^{-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-3} \end{bmatrix}, \qquad \hat{\boldsymbol{x}}_{\boldsymbol{0}}^{-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

$E\{v^2\} = R$ is the measurment noise variance from section 7.2 divided by the sample interval.

A stepwise solution of the Kalman is given by

1. Compute the Kalman gain:

$$\boldsymbol{K}[k] = \boldsymbol{P}^{-}[k]\boldsymbol{C}_d[k]^T(\boldsymbol{C}_d[k]\boldsymbol{P}^{-}[k]\boldsymbol{C}_d[k]^T + \boldsymbol{R})^{-1} \tag{35}$$

2. Update estimate with measurement $u(2) = b$:

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}^{-}[k] + \boldsymbol{K}[k]\boldsymbol{u}(2)[k] - \boldsymbol{C}_d[k]\hat{\boldsymbol{x}}^{-}[k]) \tag{36}$$

3. Compute error covariance for updated estimate:

$$\boldsymbol{P}[k] = (\mathbb{I} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])\boldsymbol{P}^{-}(\mathbb{I} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^T + \boldsymbol{K}[k]\boldsymbol{R}[k]\boldsymbol{K}[k]^T \tag{37}$$

4. Project ahead:

$$\hat{\boldsymbol{x}}^{-}[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{x}}[k] + \boldsymbol{B}_d[k]\boldsymbol{u}(1)[k] \tag{38}$$

$$\boldsymbol{P}^{-}[k+1] = \boldsymbol{A}_d[k]\boldsymbol{P}[k]\boldsymbol{A}_d[k]^T + \boldsymbol{E}_d\boldsymbol{Q}[k]\boldsymbol{E}_d^T \tag{39}$$

This is implemented using MATLAB and an s-function block in Simulink. The measured rudder angle (u(1)) and compass course (u(2)) are used as inputs. This is shown in figure 19. The outputs are estimated values of the rudder bias b and the heading ψ. These estimations can now be used for regulation with feed forward ad feedback respectively. The wave disturbance is also an output in the s-function block. The only purpose of this output variable is to measure and observe the disturbance. The implementation in Simulink is shown in figure 19. The code for the s-function block is shown in section 9.5.

The s-function-block in Simulink is connected to an s-function implementation in the MATLAB script. This script executes the Kalman filtration by iteratig frequently through the s-function followig the steps metioned above. The system is initialized once at start (see appendix A).

7.4 Feed forward estimation bias

A feed forward is made from the estimated a posteriori bias found in part 7.3. The reference course is set to 30°, and the system is simulated with current disturbance.
The measured compass course is i this case reaching the reference angle wich implies that the stationary deviation from section 7.3 is removed. Looking at the rudder in figure 14 it is seen that the estimated bias in figure 10 is equal to the stationary deviation which is added to the signal from the PD-controller. This corresponds to regulating the system using feed forward for the estimated bias. This is shown in figure 11. This way the discrete Kalman filter improved the autopilot compared to only having the PD-controller in section 7.3.

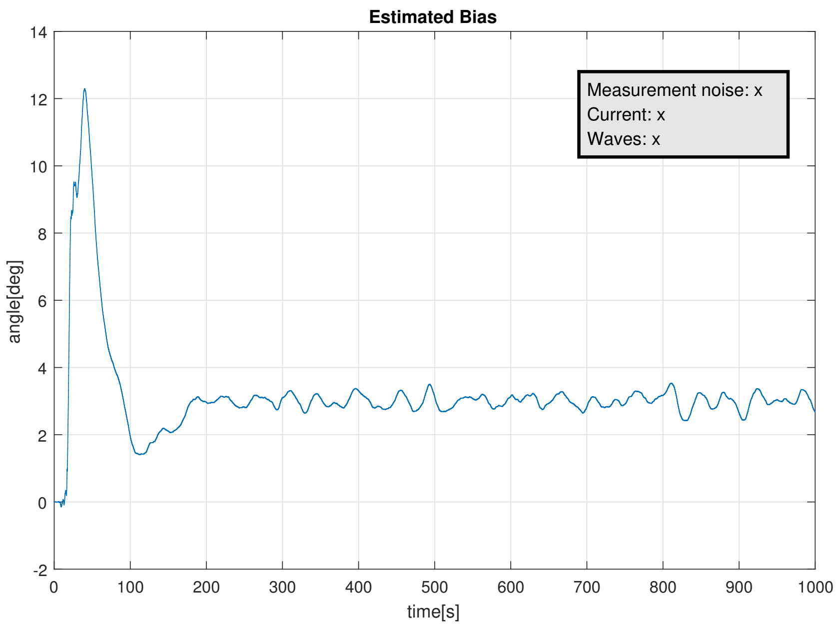

Figure 10: Estimated bias

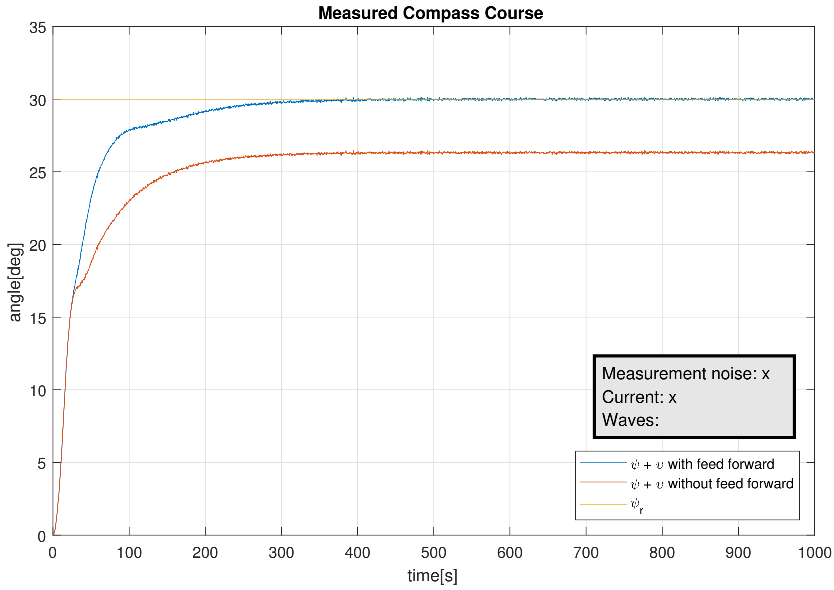

Figure 11: Measured compass course

7.5 Wave filtering

In this part the wave filtered ψ is used instead of the measured heading in the autopilot, and the system is simulated with wave and current disturbance.

From section 5.4 the rudder constantly tries to compensate for the wave disturbance. The controller uses the measured compass course affected by with wave disturbance as feedback, which causes unnecessary fluctuation on the rudder. In this part the measured compass course is filtered before it is used in the feedback. The behaviours in both cases (filtered and o-filtered feedback) is shown in figure 12. The filtered feedback avoid rapid changes in the rudder input, shown in figure 14. Although the filtered compass course in the lower part of 12 has some oscillations, it is much less than the oscillations in the upper plot i the figure.

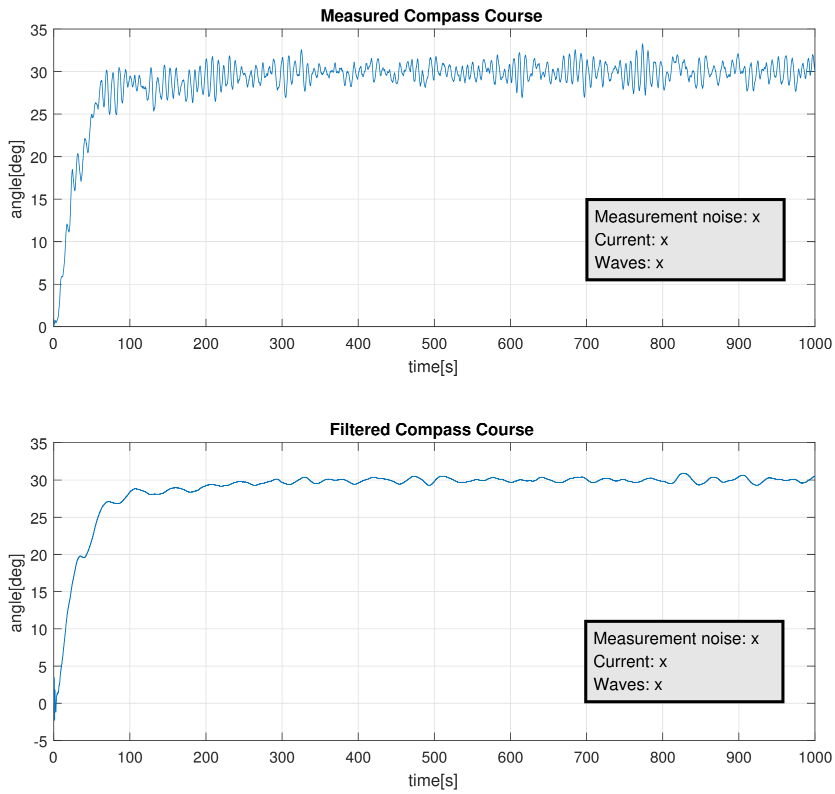

Figure 12: Measured versus filtered compass course

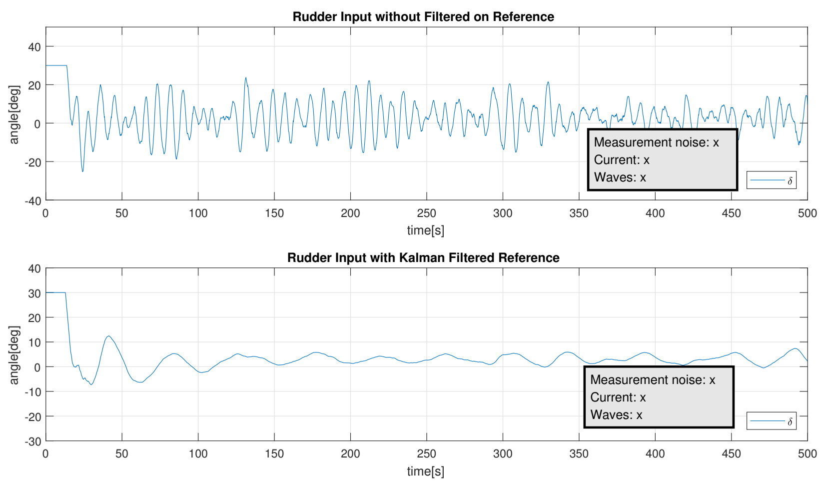

Figure 14: Rudder input

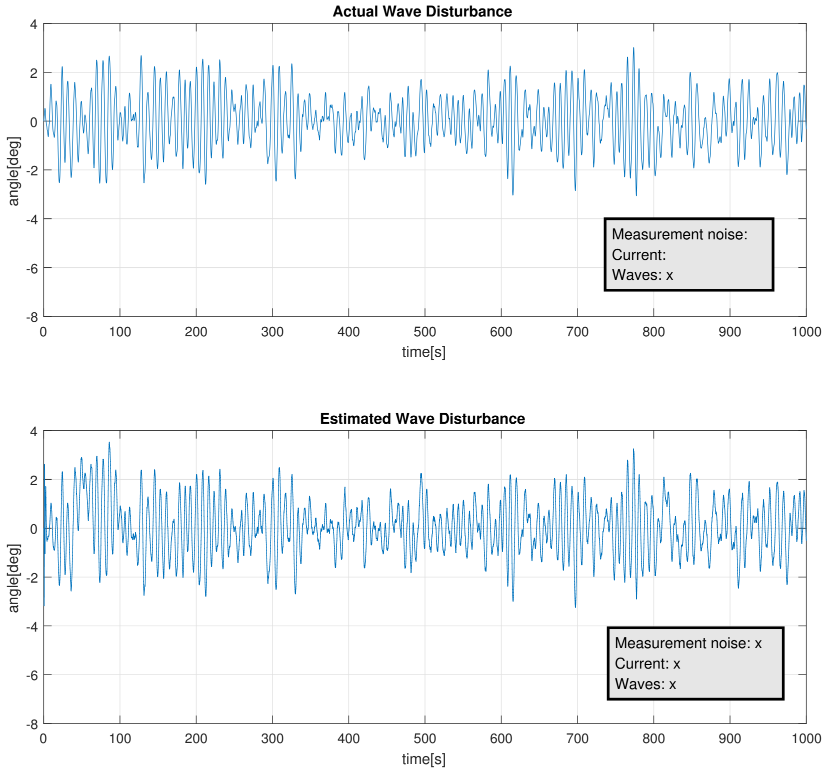

Figure 13: Actual versus estimated waves

To measure the actual wave disturbance, the measurement noise and current disturbances are turned off. The rudder input is also set to zero. The actual wave disturbance and the estimated wave disturbance from the Kalman filter is plotted in figure 13. Comparing the plots shows that the estimated wave disturbance is very similar to the actual one. The estimator is thus concluded as well-working.

8 Conclusion

In this report it is observed how disturbances influence the behaviour of a ship. PD-regulation is a good solution when there are no disturbances. Affection from current disturbance leads to stationary deviation, and affection of wave disturbance leads to a rapidly oscillating rudder in attempt to compensate for the wave disturbances.

Since the system is observable in all considered cases, these problems can be solved using a discrete Kalman filter. The feedforward from the estimated bias cancel out the current disturbance, so that the course angle follows the reference angle. The filtered compass course in the feedback avoids unnecessary change in the rudder angle, and creates a smooth course input.
The discrete Kalman filter handles both the wave and current disturbance, and gives the ship a smooth and accurate autopilot.

9 Appendix A - MATLAB scripts

9.1 Script part I

```
K = 0.16;
T = 72.5;
```

9.2 Script part II

```
[pxx, f] = pwelch(pi/180*psi_w(2, :), 4096, [], [], 10);

lambda1 = 0.3;
lambda2 = 0.08;
lambda3 = 0.05;

w0 = 0.7823;
var = 1.49*10^(-3);
sigma = sqrt(var);

Kw1 = 2*lambda1*w0*sigma;
Kw2 = 2*lambda2*w0*sigma;
Kw3 = 2*lambda3*w0*sigma;

w = f*2*pi;

p1 = (w.^2*Kw1^2)./(w.^4+2*w.^2*w0^2*(2*lambda1^2-1)+w0^4);
```

```
p2 = (w.^2*Kw2^2)./(w.^4+2*w.^2*w0^2*(2*lambda2^2-1)+w0^4);
p3 = (w.^2*Kw3^2)./(w.^4+2*w.^2*w0^2*(2*lambda3^2-1)+w0^4);

figure;hold on;
plot1 = plot(w,pxx/(2*pi))
plot(w,p1)
plot(w,p2)
plot(w,p3)
legend('p_x_x','lambda = 0.3','lambda = 0.08', 'lambda = 0.05')
xlim([0 2])
set(plot1, 'Linewidth', 2)
title('Power Spectral Density')
```

9.3 Script part III

```
K = 0.16;
T = 72.5;
w_c = 0.10;
phi = 50*pi/180;
K_pd = 0.816;
T_f = 8.39;
T_d = T;

figure
H_sys = tf([0 K*K_pd],[T_f 1 0]);
bode(H_sys, {0.01,30})
grid on
```

9.4 Script part IV

```
T = 72.52;
K = 0.16;
lambda = 0.08;
w_0 = 0.78; %rad/s

%% 4

A = [0        1              0   0     0;
     -w_0^2   -2*lambda*w_0  0   0     0;
     0        0              0   1     0;
     0        0              0   -1/T  -K/T;
     0        0              0   0     0];

C = [0 1 1 0 0];

%% b

O_b = obsv(A(3:4,3:4),C(3:4))
```

```

%% c

O_c = obsv(A(3:5,3:5), C(3:5))

%% d

O_d = obsv(A(1:4,1:4), C(1:4))

%% e

O_e = obsv(A, C)
```

9.5 Script part V

Computing the measurment noise variance.

```
%measurmentnoise variable is from Simulink model
V = var(measurmentnoise*pi/180)
```

Initializing constants and matrices.

```
K = 0.16;
T = 72.5;

w_c = 0.10;
phi = 50*pi/180;

K_pd = 0.816;
T_f = 8.39;
T_d = T;
T_s = 0.1;

lambda = 0.08;
w_0 = 0.78; %rad/s

var = 1.484e-3;
sigma = sqrt(var);
K_w = 2*lambda*w_0*sigma;

%%

A = [0 1 0 0 0;
     -1*w_0^2 -2*lambda*w_0 0 0 0;
     0 0 0 1 0;
     0 0 0 -1/T -K/T;
     0 0 0 0 0];

A_d = A*T_s + eye(5);
```

```
B  = [0; 0; 0; K/T; 0];

B_d = B*T_s;

C = [0 1 1 0 0];

C_d = C;

D_d = 0;

E = [0 0; K_w 0; 0 0; 0 0; 0 1];

E_d = E*T_s;

Q = [30 0; 0 10^(-6)];

var = 6.0657e-07;
R = var/T_s;

P_0 = [1 0 0 0 0;
       0 0.013 0 0 0;
       0 0 pi^2 0 0;
       0 0 0 1 0;
       0 0 0 0 2.5e-3];

x_0 = [0; 0; 0; 0; 0];

%%

%struct data
data = struct('A', A_d,'B', B_d, 'C', C_d,'E', E_d,'Q', Q,'R', R,'P', P_0,'x', x_0);
```

S-function.

```
function [sys,x0,str,ts] = DiscKal(t,x,u,flag,data)

switch flag,

  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
   case 0
     [sys,x0,str,ts] = mdlInitializeSizes(data);

  %%%%%%%%%%%%%
  % Outputs   %
  %%%%%%%%%%%%%
```

```matlab

  case 3
    sys = mdlOutputs(t,x,u,data);
 %%%%%%%%%%%%%
 % Terminate %
 %%%%%%%%%%%%%

  case 2
    sys = mdlUpdate(t,x,u, data);

  case {1,4,}
    sys = [];

  case 9
      sys = mdlTerminate(t,x,u);
 %%%%%%%%%%%%%%%%%%%%
 % Unexpected flags %
 %%%%%%%%%%%%%%%%%%%%
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

function [sys,x0,str,ts]=mdlInitializeSizes(data)
% This is called only at the start of the simulation.

sizes = simsizes; % do not modify

sizes.NumContStates  = 0; % Number of continuous states in the system, do not modify
sizes.NumDiscStates  = 35; % Number of discrete states in the system, modify.
sizes.NumOutputs     = 3; % Number of outputs, the hint states 2
sizes.NumInputs      = 2; % Number of inputs, the hint states 2
sizes.DirFeedthrough = 1; % 1 if the input is needed directly in the
% update part
sizes.NumSampleTimes = 1; % Do not modify

sys = simsizes(sizes); % Do not modify

x0  = [reshape(data.x(:)', 1,5), 0 0 0 0 0, data.P(:)']; % Initial values for the discrete states, modify

str = []; % Do not modify

ts  = [-1 0]; % Sample time. [-1 0] means that sampling is
% inherited from the driving block and that it changes during
% minor steps.
```

```


function sys=mdlUpdate(t,x,u, data)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Update the filter covariance matrix and state etsimates here.
% example: sys=x+u(1), means that the state vector after
% the update equals the previous state vector + input nr one.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% A priori P
P_= reshape(x(11:35), sqrt(length(x(11:35))), sqrt(length(x(11:35))));

% Kalman gain
K = P_*data.C'*inv(data.C*P_*data.C'+data.R);

% A priori estimate of x
xh_= x(1:5);

% A posteriori estimate of x
xh = xh_ + K*(u(2)-data.C*xh_);

% A posteriori error covariance
P = (eye(5,5)-K*data.C)*P_*(eye(5,5)-K*data.C)'+K*data.R*K';

% A priori estimate of x
xh_ = data.A*xh + data.B*u(1);

% A priori error covariance
P_= data.A*P*data.A' + data.E*data.Q*data.E';

sys=[xh_', xh', P_(:)']';


function sys=mdlOutputs(t,x,u,data)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the outputs here
% example: sys=x(1)+u(2), means that the output is the first state+
% the second input.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
sys = [x(8) x(10) x(7)];

function sys=mdlTerminate(t,x,u)
sys = [];
```

10 Appendix B - Simulink models

10.1 Simulink part I

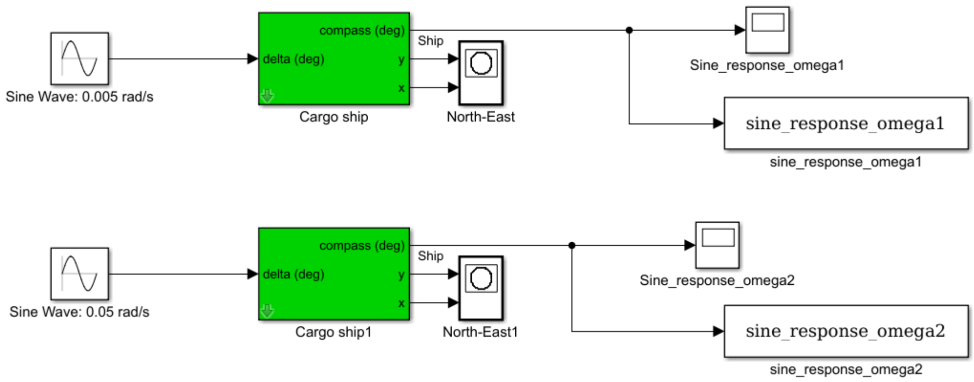

Figure 15: Sine response ship

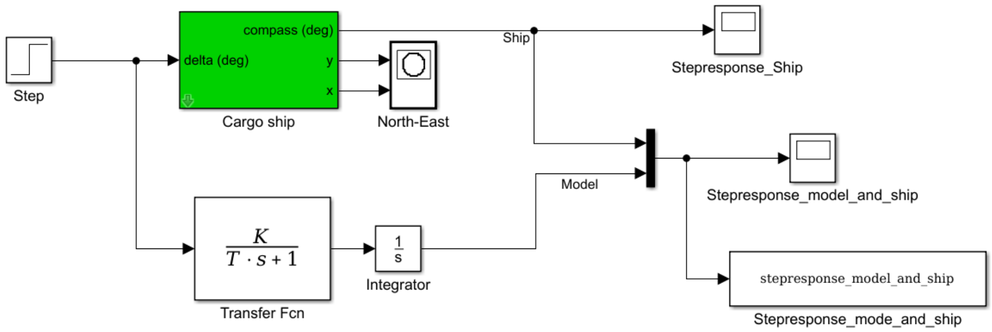

Figure 16: Step response model and ship

10.2 Simulink part III

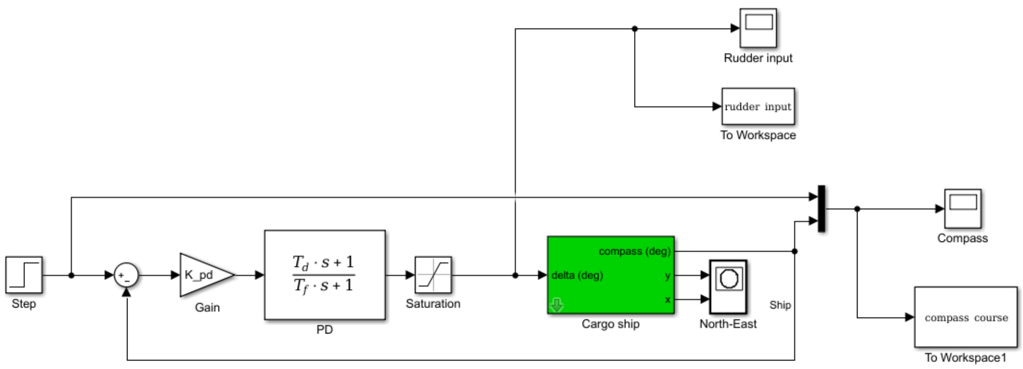

Figure 17: PD control system

10.3 Simulink part V

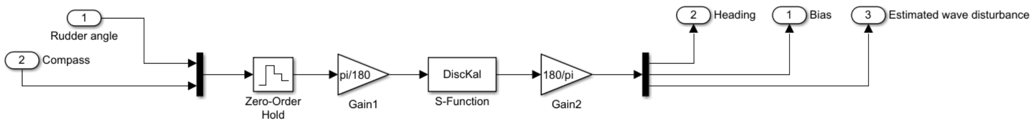

Figure 18: Kalman filter

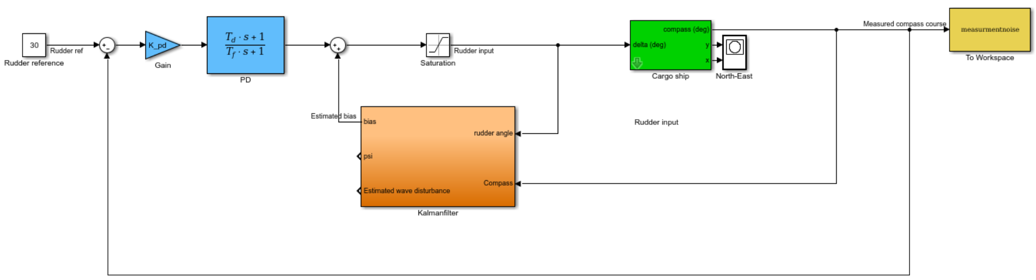

Figure 19: Estimated bias feed forward

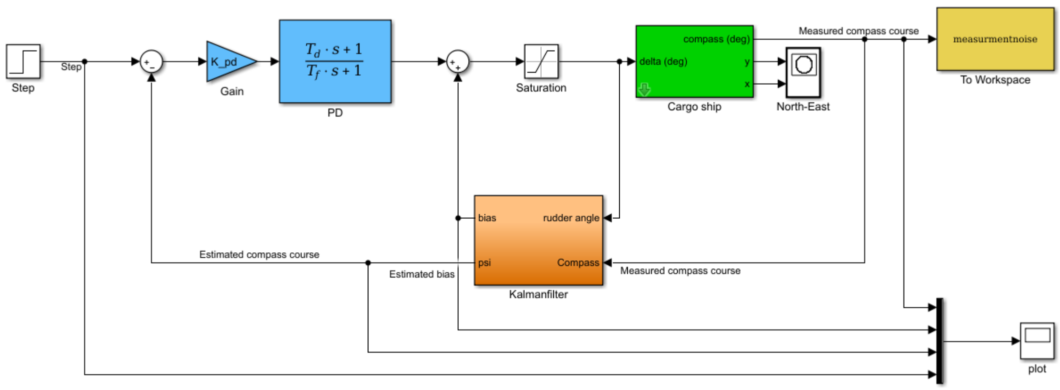

Figure 20: Estimated course angle feed back

References

[1] *Boat lab assignment*. Blackboard, 2017.

[2] Unknown Author. *Resonance frequency*. Wikipedia, 21.11.2017.

[3] Unknown Author. *Margin*. MathWorks, 22.11.2017.

[4] Faculty of technology. *Discretization*.

[5] Alan V. Oppenheim and George C. Verghese. *Power Spectral Density*. MIT, 2010.

[6] E. W. Weisstein. *Linear Systems Theory*. Princeton University Press, 2009.