Boat Lab Report

TTK4115 - Linear System Theory

Group 31
Øystein Brox - 741461
Morten Jansrud - 742003
Trym Nordal - 749720

Department of Engineering Cybernetics
NTNU
Norway
November 2016

Contents

Preliminary

We are to find a model and simulate a cargo ship influenced by wave and current disturbance. The ship will be modeled as a continuous system influenced by stochastic signals based on given information and be simulated in order to identify unknown parameters. A simple autopilot will be designed, and a discrete Kalman filter implemented for filtering noise based on estimations of disturbances. MATLAB and Simulink are used for simulations.

Summary of the complete system
The model which will be used is given by

$$\dot{\xi_w} = \psi_w \tag{0.1}$$

$$\dot{\psi_w} = -\omega_0^2 \xi_w - 2\lambda\omega_0\psi_w + K_w w_w \tag{0.2}$$

$$\dot{\psi} = r \tag{0.3}$$

$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - b) \tag{0.4}$$

$$\dot{b} = w_b \tag{0.5}$$

$$y = \psi + \psi_w + v \tag{0.6}$$

With state vector $[\xi_w \ \ \psi_w \ \ \psi \ \ r \ \ b]^T$, the system can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w}, \quad y = \mathbf{C}\mathbf{x} + v \tag{0.7}$$

ψ is the average heading, ψ_w is a high frequency wave disturbance component, $\dot{\xi_w} = \psi_w$, r is the rotation velocity about the z-axis, b is bias to the rudder angle, $u = \delta$ is the rudder angle relative to the body frame, $\mathbf{w}$ is the wave and current disturbance and v is the measurement noise.

1 Part 1 - Identification of the boat parameters

1.1 Task A - Calculation of the transfer function

This bias to the rudder angle is modelled as $\dot{b} = w_b$, where w_b is Gaussian white noise.

Without any disturbance, $b = 0$, the transfer function from δ to ψ can be derived based on equation 0.4 as

$$\mathcal{L}\{\ddot{\psi} = \frac{-1}{T}\dot{\psi} + \frac{K}{T}\delta\}$$

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

The initial heading of the cargo ship is modeled $\psi(0) = 0$.

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

$$H(s) = \frac{\psi}{\delta} = \frac{K}{s(sT+1)} \tag{1.1}$$

1.2 Task B - Parameters in smooth weather conditions

The boat parameters K and T describes the characteristics of the the cargo ship regarding rotation velocity around the z-axis, and are needed to calculate an optimal regulator.

The amplitudes of the sine waves on the output equals $|H(j\omega)|$, and must be determined to calculate the parameters. Substituting $s = j\omega$ into 1.1 gives

$$|H(j\omega)| = \left|\frac{K}{T(j\omega)^2 + j\omega}\right| = \frac{K}{\sqrt{(-T\omega^2)^2 + \omega^2}} = \frac{K}{\omega\sqrt{T\omega^2 + 1}} \tag{1.2}$$

Figure 1 and 2 displays the heading response with sine wave inputs on the rudder of frequency $\omega_1 = 0.005$ and $\omega_1 = 0.05$ (rad/s) .

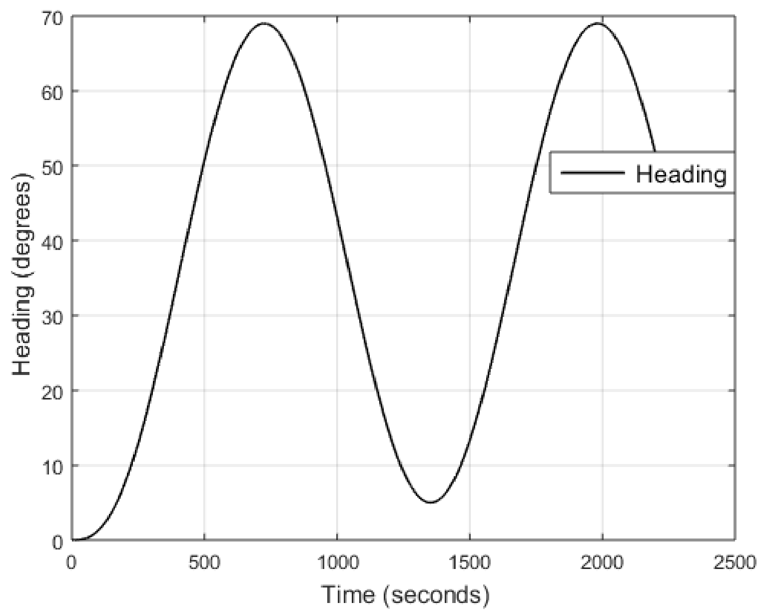

Figure 1: Heading of ship in smooth weather and $\omega_1 = 0.005$

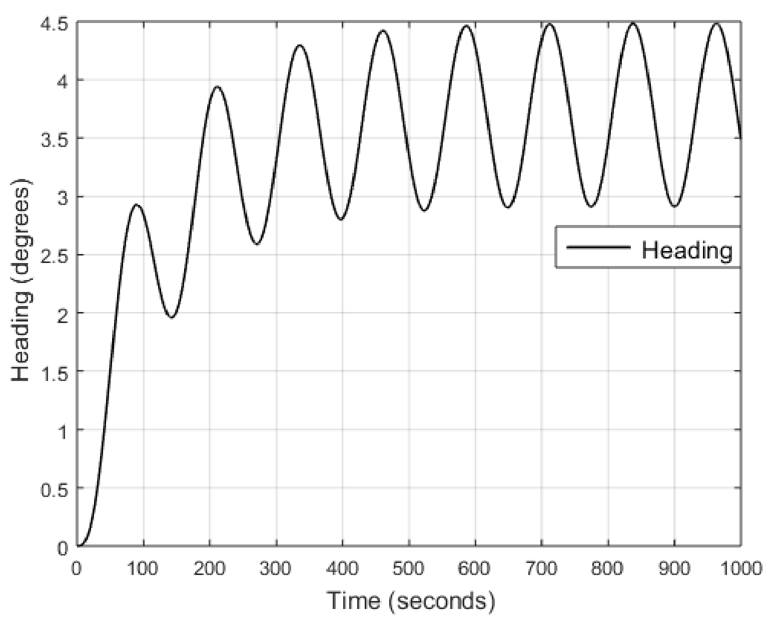

Figure 2: Heading of ship in smooth weather and $\omega_2 = 0.05$

The amplitudes was found by measuring the distance from equilibrium to the highest points on figure 1 and 2.

$$
\begin{aligned}
|H(j\omega_1)| &= A_1 = 31.9 \\
|H(j\omega_2)| &= A_2 = 0.785
\end{aligned}
$$

Using these values in equation 1.2 we obtain two new equations and solve for K and T.

$$K = A_1\omega_1\sqrt{T^2\omega_1^2 + 1} = 0.174 \tag{1.3}$$

$$T = \frac{\sqrt{K^2 - A_2^2\omega_2^2}}{A_2\omega_2^2} = 86.3 \tag{1.4}$$

The corresponding Simulink file is found in Appendix A, figure 18.

1.3 Task C - Parameters in rough weather conditions

Good estimates of the boat parameters are possible in good weather conditions as shown in section 1.2. When the cargo ship is affected by measurement noise or severe weather, the result changes.

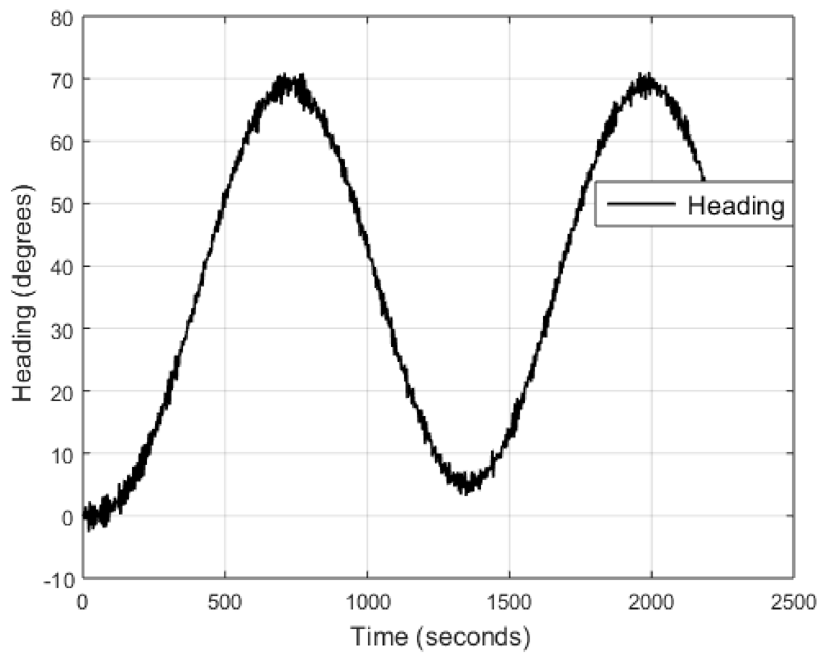

Figure 3: Heading of ship in rough weather and $\omega_1 = 0.005$

It is possible to approximate the parameters with measurement noise activated, as the noise only cause small deviations from the correct heading. It will though, not be possible to obtain equally good estimations as in task B. The measurement noise makes it harder to observe the correct amplitude of the oscillations, which will, in some degree, affect the values of the parameters.

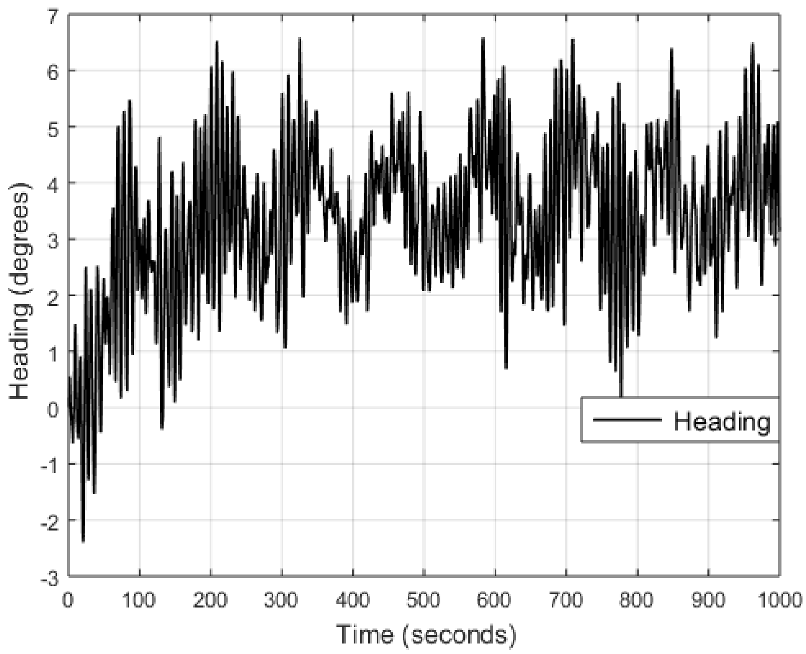

Figure 4: Heading of ship in rough weather and $\omega_2 = 0.05$

With wave noise activated it will be nearly impossible to approximate good parameters for the cargo ship. As shown in figure 4, it is difficult to separate the noise from the heading. A solution is to decrease the frequency on the input, making the cargo ship turn in slower oscillations and as a result, the amplitudes becomes easier to read.

1.4 Task D - Step input of 1 degree

As shown in figure 5 the ship and the model are close to each other the first 600 seconds, but after this point the difference between the ship and the model increases.

The model is used for designing an optimized regulator and to calculate state estimations of the system. A bigger deviation will result in a non optimal regulator, as the model does not correspond to the actual ship.

An approximation will never be exactly like the original system. The resulting deviation is acceptable.

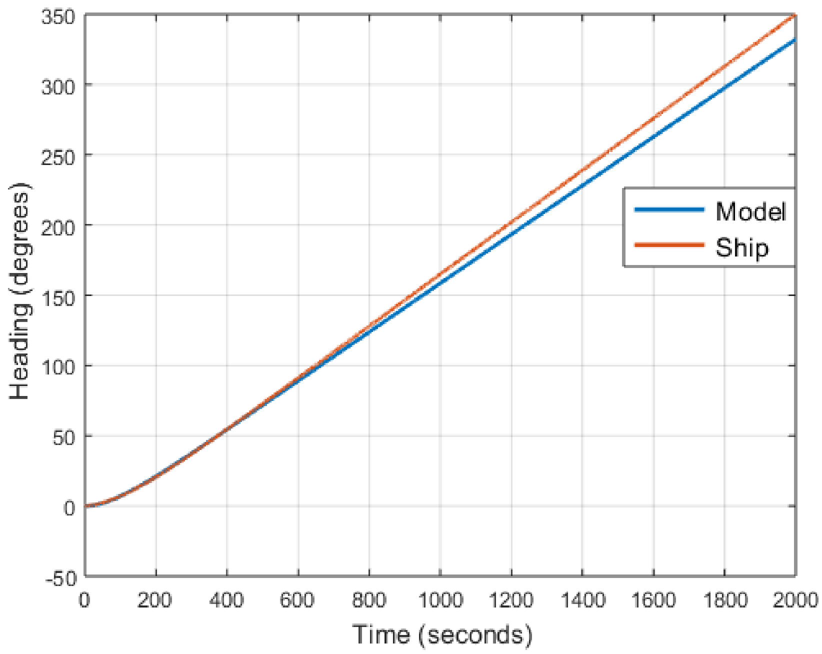

Figure 5: Step response of the model compared to the step response of the ship

2 Part 2 - Identification of wave spectrum model

2.1 Task A - Estimate of the PSD function

The estimate for Power Spectral Density (PSD) was computed using MATLAB function *pwelch*, witch uses Welch's Method. This function, given input arguments as seen in Appendix B, divides the input data into 4096 parts. These parts overlap by 50% since *noverlap* is set to a vector. The PSD is calculated using discrete Fourier transform. Welch's Method is based on Bartlett's method, but gives a significantly lower variance.

The plot in figure 6 displays how waves with given frequencies affects the cargo ship with different magnitudes.

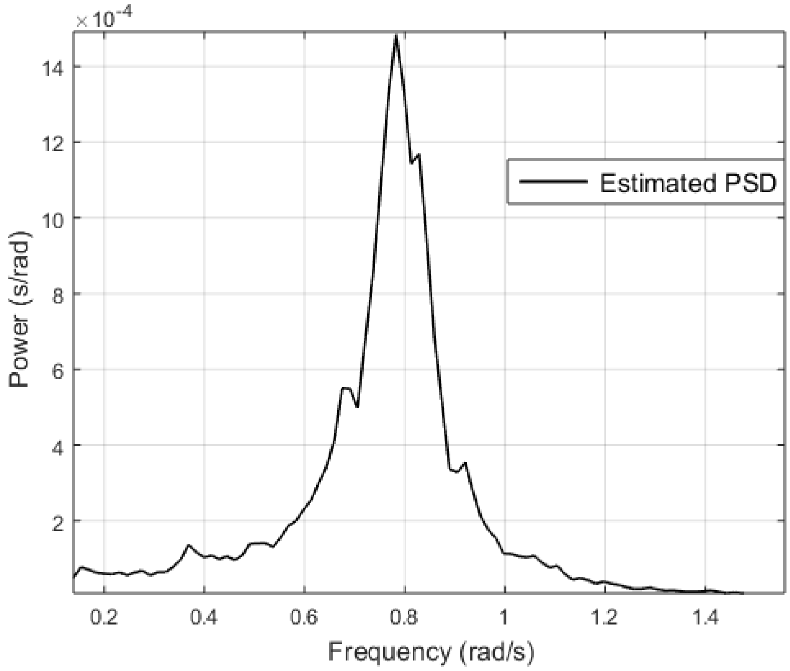

Figure 6: Estimated Power Spectral Density from waves

2.2 Task B - Analytic expression for the transfer function

The transfer function of the wave response model(from w_w to ψ_w) is calculated as follows using equations 0.1 and 0.2

$$\mathcal{L}\{\dot{\psi_w} = -\omega_0^2\xi_w - 2\lambda\omega_0\psi_w + K_w w_w\}$$

$$\Rightarrow \quad s\psi_w(s) + \psi_w(0) = -\frac{\omega_0^2\psi_w(s)}{s} - 2\lambda\omega_0\psi_w(s) + K_w w_w$$

The initial heading of the cargo ship is modeled $\psi_w(0) = 0$.

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

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

Since $H(s)$ is a stable transfer function, the PSD of ψ_w can be found through stationary analysis. The formula is given in *Introduction to random signals and* *applied Kalman filtering*[2] and its equation 3.2.2, hereby referenced as equation 2.2.

$$S_x(j\omega) = G(j\omega)G(-j\omega)S_f(j\omega) = |G(j\omega)|^2 S_f(j\omega) \tag{2.2}$$

In this report, equation 2.2 is written as

$$P_{\psi_w}(j\omega) = G(j\omega)G(-j\omega)P_{w_w}(j\omega)$$

where

$$G(j\omega) = \frac{\psi_w}{w_w} = \frac{K_w j\omega}{((j\omega)^2 + 2j\omega\lambda\omega_0 + w_0^2)}$$

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

Since $S_f(j\omega) = P_{w_w}(j\omega)$ is the amplitude of w_w, which is white noise, it will be constant and consequently equal to one.

Thus the PSD becomes

$$\frac{(K_w j\omega)(-K_w j\omega)}{((j\omega)^2 + 2j\omega\lambda\omega_0 + w_0^2)((-j\omega)^2 - 2j\omega\lambda\omega_0 + w_0^2)}$$

$$\Rightarrow \frac{(K_w\omega)^2}{\omega_0^4 + \omega^4 + (4\omega^2\lambda^2\omega_0^2) - 2\omega_0^2\omega^2}$$

$$P_{\psi_w}(j\omega) = \frac{K_w^2\omega^2}{\omega_0^4 + \omega^4 + (4\lambda^2 - 2)\omega_0^2\omega^2} \tag{2.3}$$

Based on the properties of spectral functions, the PSD becomes a real function of ω. So $P_{\psi_w}(j\omega)$ can be written as $P_{\psi_w}(\omega)$.

2.3 Task C - Finding ω_0 from the estimated PSD

MATLAB has a built in function, *max*, which was utilized in order to find the largest value of the estimated $S_{\psi_w}(\omega)$ from task 2A. This resulted in $\omega_0 = 0.7823$.

The ω_0 is the resonance frequency, and describes the frequency of waves that have the greatest impact on the heading of the ship.

2.4 Task D - Identifying the damping factor λ

By fitting the analytic Power Spectral Density Function (equation 2.3) to the Estimated Power Spectral Density Function (task 2A), the dampening factor, λ can be found through trial and error. A dampening factor of $\lambda = 0.08$ gave the most accurate representation.

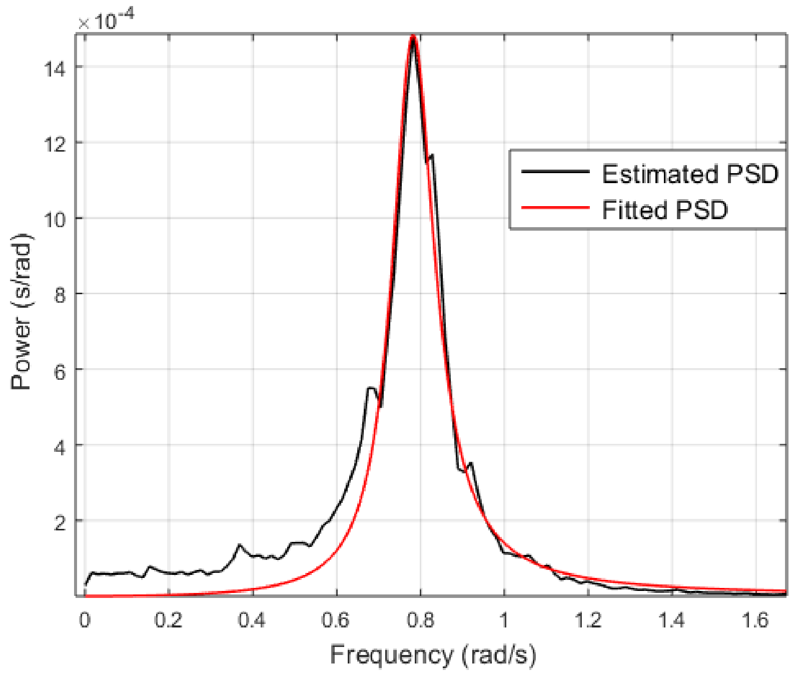

Figure 7: The fitted PSD function compared to the estimate

3 Part 3 - Control system design

3.1 Task A - PD controller

Given the PD controller $H_{pd}(s) = K_{pd}\frac{1+T_d s}{1+T_f s}$, the transfer function of the total system $H(s) = H_{pd}(s) \cdot H_{ship}(s)$ becomes

$$H(s) = K_{pd}\frac{1+T_d s}{1+T_f s} \cdot \frac{K}{Ts^2+s} = K_{pd}\frac{\frac{K}{T}+\frac{KT_d}{T}s}{T_f s^3 + (1+\frac{T_f}{T})s^2 + \frac{1}{T}s} \tag{3.1}$$

Based on observation of the transfer function 3.1, T must equal T_d to cancel the derivative time constant for an optimal implementation of the controller. According to the task, ω_c and the phase margin of the open loop system are desired to be approximately 0.10 (rad/s) and 50 degrees. These values can be achieved by changing K_{pd} and T_f and studying the Bode Diagram, with the result shown in Figure 8. T_f affects both the magnitude and phase of the system. The desired phase, $50° - 180° = -130°$ was reached by setting $T_f = 8.5$. As K_{pd} only affects the magnitude, the crossover frequency was adjusted to $0.10\frac{rad}{s}$. $K_{pd} = 0.7612$ gave an accurate approach.

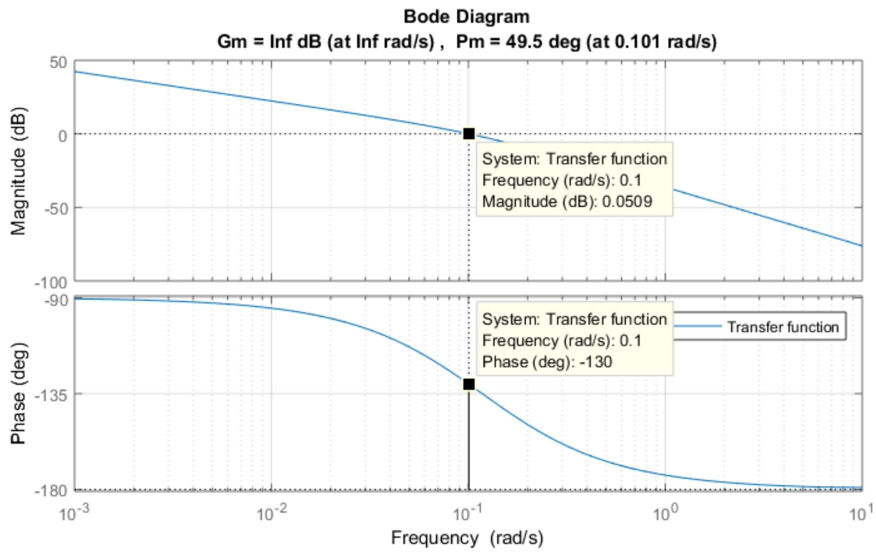

Figure 8: Bode diagram of the transfer function

3.2 Task B - Simulation with measurement noice

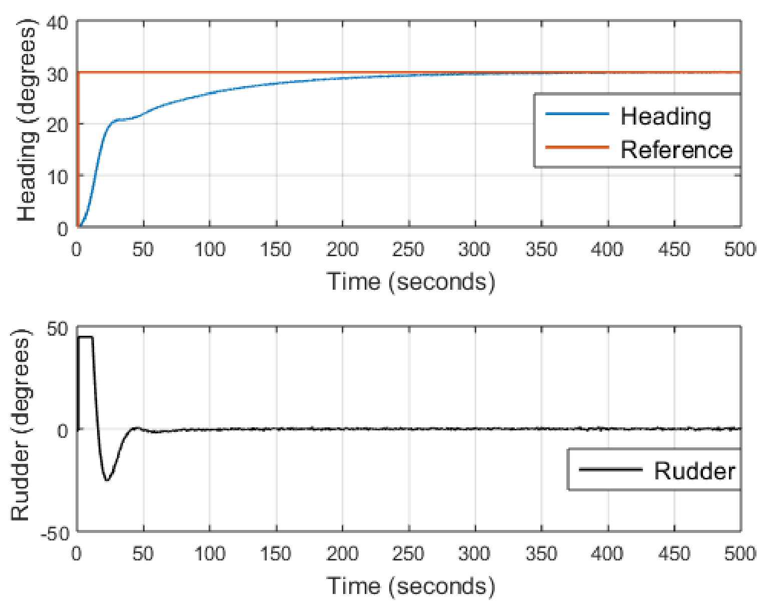

Figure 9: Comparison of heading heading and reference with measurement noise

The autopilot works satisfactory with activated measurement noise and the reference heading $\psi_r = 30°$. If the PD controller was unable to to control the heading with such minor noise, a retuning of the controller would be needed.

It takes 300 seconds to turn the ship heading from 0 to 30 degrees. It is possible to generate a faster response by increasing K_{pd}. However, as a result the phase margin would decrease and make the system less stable.

3.3 Task C - Simulation with current

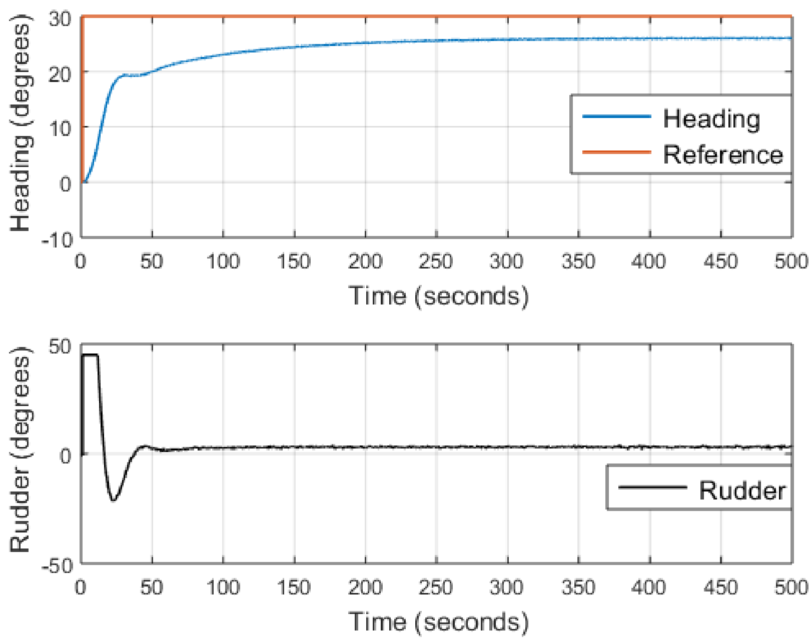

Figure 10: Comparison of heading heading and reference with current

The cargo ship does not reach the reference angle with current activated and get a stationary deviation from the desired course. An issue that would make the PD controller nearly useless on ships affected by real ocean. If the ship deviated 5° on the course from Trondheim to New York, we would end up in Florida, as seen in figure 11. This problem can be solved by upgrading to a PID controller. The new controller would integrate up the stationary deviation and reach the desired heading.

w is given in the NED reference frame. The greater the change in heading, the more inaccurate the model will become, as ψ_w changes. In other words, it's a difference between having the wind and current normal or parallel to the cargo ship. The ship heading was limited to $\psi_w = \pm 35°$ to plot accurate simulations.

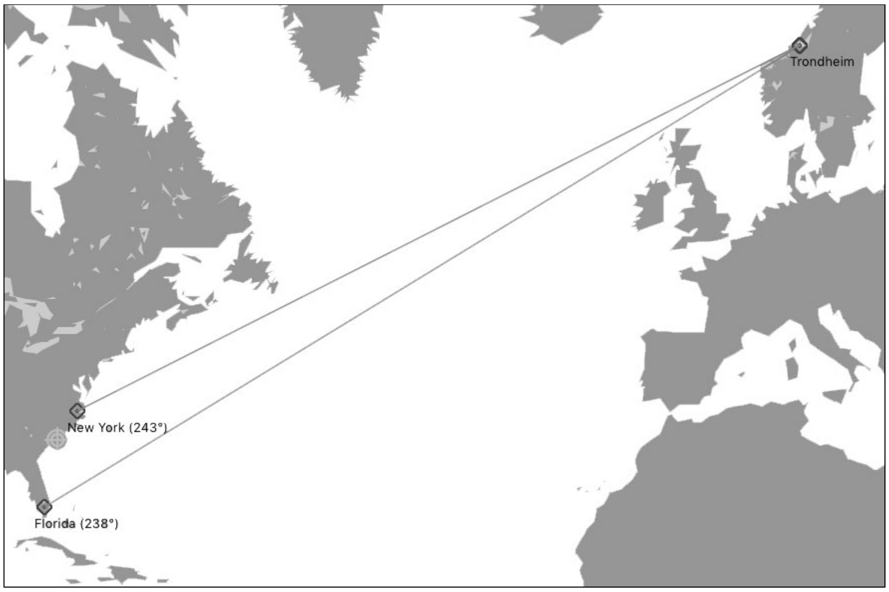

Figure 11: Course from Norway to USA with 5° deviation

3.4 Task D - Simulation with waves

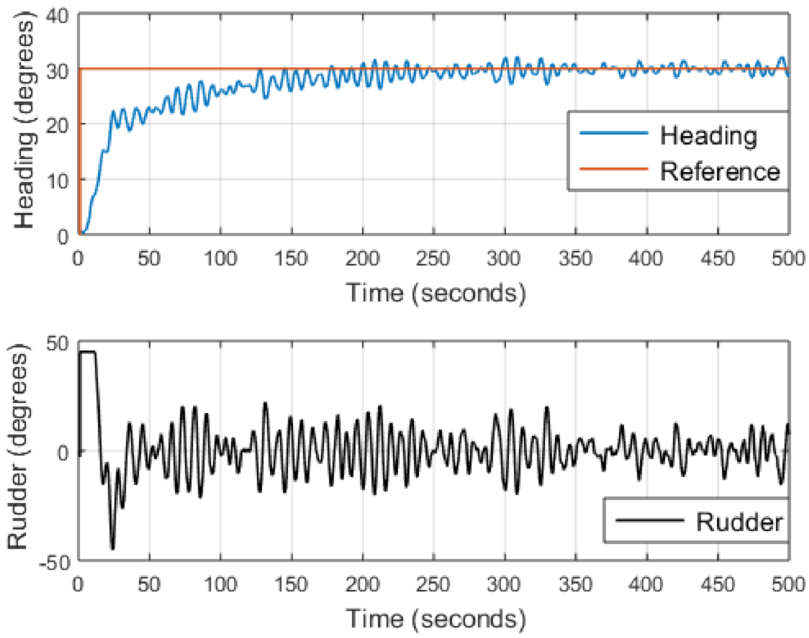

Figure 12: Comparison of heading heading and reference with waves

When waves are the only noise, the ship is able to reach the desired heading with some oscillations. A problem however, is the unstable fluctuations on the rudder. Over time, such fluctuations will probably cause damage. With the rudder bias oscillating the controller will unnecessarily compensate for wave noise. To solve these issues it is possible to implement state estimators, which will be introduced later in the report.

4 Part 4 - Observability

In this part of the report the observability of the system with different states is to be determined.

4.1 Task A - Finding the matrices A, B, C and E

Given the system 0.7, **A**, **B**, **C** and **E** matrices are obtained using equations 0.1 - 0.6

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

$$
\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}
$$

$$
\mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}
$$

4.2 Task B - Observability without disturbances

When studying the system without disturbances the states are reduced to $\mathbf{x} = [\psi \;\; r]^T$. $\mathbf{A}$ and $\mathbf{C}$ are now

$$
\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.1}
$$

The observability matrix is found to be

$$
\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.2}
$$

where $rank(\mathcal{O}) = 2$. Since the number of states equal the rank of the observability matrix, the system without disturbances is observable.

4.3 Task C - Observability with ocean current

The system with current disturbance has the following states: $\mathbf{x} = [\psi \;\; r \;\; b]^T$. $\mathbf{A}$, $\mathbf{C}$ and $\mathcal{O}$ matrices are now given by

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

$$
\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \end{bmatrix} \tag{4.4}
$$

$rank(\mathcal{O}) = 3$ which equals the number of states. The system is observable with current disturbance.

4.4 Task D - Observability with waves

The system with wave disturbance has the following states: $\mathbf{x} = [\xi_w \ \ \psi_w \ \ \psi \ \ r]^T$. The $\mathbf{A}$, $\mathbf{C}$ and $\mathcal{O}$ matrices are now given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \tag{4.5}$$

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 1 \\ 2\lambda\omega_0^3 & (4\lambda^2-1)\omega_0^2 & 0 & -\frac{1}{T} \\ (4\lambda^2-1)\omega_0^4 & (4\lambda^2-1)2\lambda\omega_0^3 & 0 & \frac{1}{T^2} \end{bmatrix} \tag{4.6}$$

This $\mathcal{O}$ matrix was found in MATLAB and the rank was found using the function

```
rank(O)
```

which gave $rank(\mathcal{O}) = 4$. The rank of $\mathcal{O}$ equals the number of states and makes the system observable with wave disturbance.

4.5 Task E - Observability with current and wave disturbance

With all the states and matrices from Task A, the $\mathcal{O}$ matrix is obtained from MATLAB, and found to be

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \mathbf{CA}^4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 1 & 0 \\ 2\lambda\omega_0^3 & (4\lambda^2-1)\omega_0^2 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ (4\lambda^2-1)\omega_0^4 & (4\lambda^2-1)2\lambda\omega_0^3 & 0 & \frac{1}{T^2} & \frac{K}{T^2} \\ (2\lambda^2-1)4\lambda\omega_0^5 & (4\lambda^4-12\lambda^2+1)\omega_0^4 & 0 & -\frac{1}{T^3} & -\frac{K}{T^3} \end{bmatrix} \tag{4.7}$$

Using the same function as in Task D, $rank(\mathcal{O}) = 5$. The rank of $\mathcal{O}$ is the same as the number of states, which concludes that this system, with both current and wave disturbance, is observable.

5 Part 5 - Discrete Kalman filter

A Kalman filter is to be implemented in order to make estimators for the heading and rudder bias. A precondition for implementing the Kalman filter is an observable system, which was confirmed in part 4.

5.1 Task A - Exact discretization of the model

The model from Task 4A was discretized using the MATLAB function $c2d(sys, Ts)$, where Ts is the sample time. The sampling frequency was 10Hz, resulting in $Ts = 0.10$s. Since the method of discretization was not specified in the function, MATLAB used the default *Zero Order Hold*, witch assumes the value to be piecewise constant over each period of Ts. This discretization gave the following matricies.

$$\boldsymbol{A_d} = \begin{bmatrix} 0.9970 & 0.0993 & 0 & 0 & 0 \\ -0.0608 & 0.9845 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0999 & 0 \\ 0 & 0 & 0 & 0.9988 & -0.0002 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B_d} = \begin{bmatrix} 0 \\ 0 \\ 0.0000101 \\ 0.0002013 \\ 0 \end{bmatrix}, \quad \boldsymbol{E_d} = \begin{bmatrix} 0 & 0 \\ 0.0005 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad \boldsymbol{C_d} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

All of the discretized matricies are denoted with subscript d.

5.2 Task B - Variance of the measurement noise

In order to find the variance of the measurement noise the reference heading, wave disturbance and current disturbance was set to zero. The measurement was sent to the workspace of MATLAB. In order to calculate the variance the built in MATLAB function, *var*, was used. This turned out to be $\sigma^2 = 6.0226 \cdot 10^{-7}$. The variance was used in the calculation of R

$$R = \frac{\sigma^2}{T_s} = \frac{6.0226 \cdot 10^{-7}}{0.10} = 6.0226 \cdot 10^{-6} \tag{5.1}$$

5.3 Task C - Implementing the Kalman filter

As stated in the task the following matricies were set

$$\boldsymbol{P_0^-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-4} \end{bmatrix}, \quad \boldsymbol{x_o^-} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_w & w_b \end{bmatrix}^T, \quad E\{\mathbf{w}\mathbf{w}^T\} = \mathbf{Q} = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix}$$

R was found by sampling the measurement noise variance, $R = \sigma^2/0.1 = 6.023 \cdot 10^-6$. Although only one of the methods are necessary, the Kalman filter was implemented with a S-function and a MATLAB-function. The remaining part of the report rely on the MATLAB-function implementation, while the S-function solution is shown in figure 22 and section 7.3. The filter was modeled in Simulink as shown in figure 21, with *rudder* and *compass* as inputs. The variables had to be taken through a *Zero Order Hold* block in order to make them discrete, as the Kalman filter works in discrete time. A *Memory* block was added after the MATLAB-function block in order to avoid an algebraic loop. An event where Simulink attempts to set the states based on the initial values, but is unable because of the Kalman filter and the controller are dependent on each other.

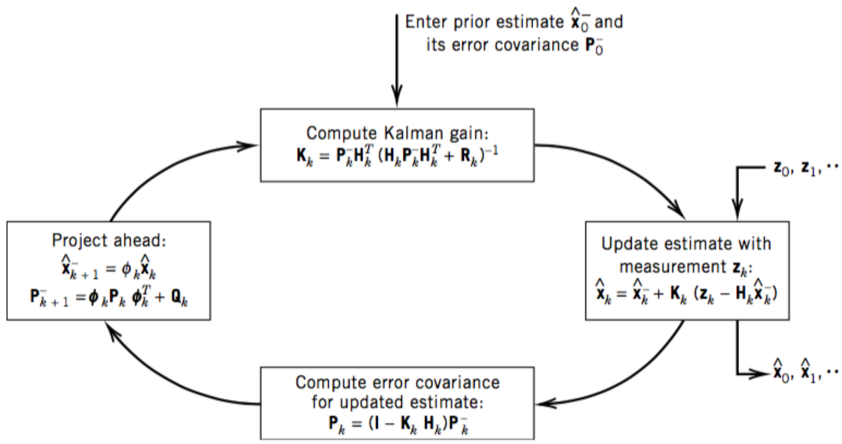

Figure 13: The filter loop, from page 147 in [2]

The inputs of the MATLAB-function block are stated at the top of the code, as seen in section 7.2, with rudder input and compass as parameters. The other variables are fetched from the workspace. *if isempty* together with the variable *init_flag* was used in order to initialize the system the first time. After initialization, the filter can start estimating heading and bias, remove noise, and feed the results back to Simulink.

The implementation of the Kalman filter was created based on the filter loop from figure 13 and equation and our Kalmanfiter is given by

Kalman gain

$$\boldsymbol{K_k} = \boldsymbol{P_k C_k^T}(\boldsymbol{C_k P_K^- C_k^T} + \boldsymbol{R}) \tag{5.2}$$

Update estimate

$$\hat{\boldsymbol{x_k}} = \boldsymbol{\hat{x}_k^-} + \boldsymbol{K_k}(z_k - \boldsymbol{C_k \hat{x}_k^-}) \tag{5.3}$$

Update error covariance

$$\boldsymbol{P_k} = (\boldsymbol{I} - \boldsymbol{K_k C_k})\boldsymbol{P_k^-}(\boldsymbol{I} - \boldsymbol{K_k C_k})^{\boldsymbol{T}} + \boldsymbol{K} R \boldsymbol{K_k^T} \tag{5.4}$$

Project ahead

$$\boldsymbol{\hat{x}_{k+1}} = \boldsymbol{A_k \hat{x}_k} + \boldsymbol{B_k} u_k \tag{5.5}$$

$$\boldsymbol{P_{k+1}} = \boldsymbol{A_k P_k A_k^T} + \boldsymbol{EQE^T} \tag{5.6}$$

Where z_k is compass measurement and u_k is rudder input. Through a two step, recursive method the Kalman filter will find an optimal estimate of the bias and heading with reduced disturbances. The first step calculate estimates based on current states and observations. Subsequently, these estimations will be corrected using covariance, which is calculated and updated at each iteration using the measurements being made. The Kalman Filter gathers as much information as possible out of uncertain measurements, and utilize this knowlegde to create the best estimates possible.

On the cargo ship, The Kalman filter estimations is used to prevent high frequency oscillations that can damage the rudder machinery over time and remove the stationary error for the PD controller from part 3, task C.

5.4 Task D - Simulation with current disturbance

In order to cancel the bias the estimated bias is added to the rudder input. This can be seen from eq. 0.5, assuming $\hat{b} = b$.

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

Plot 14 shows the response of the measured compass course, the rudder input and the estimated bias with current disturbance.

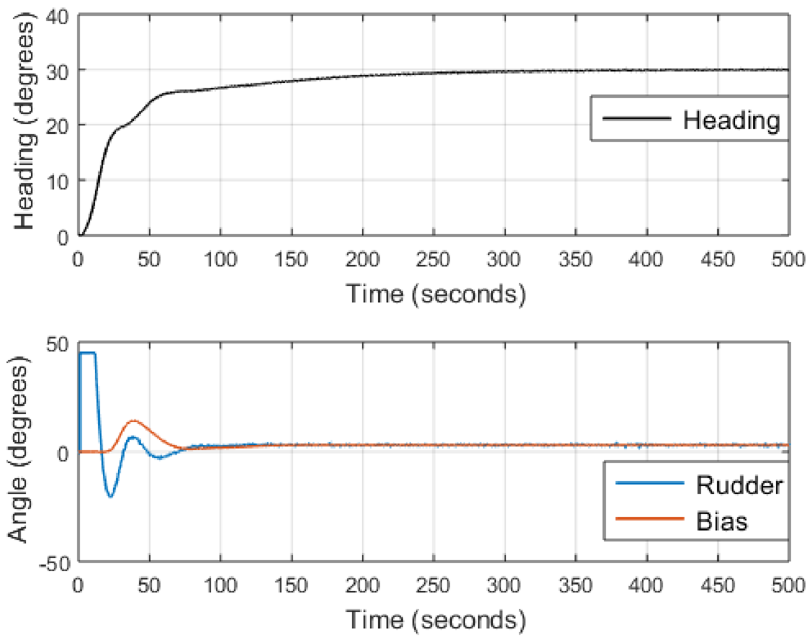

Figure 14: Comparison of rudder and bias together with the heading

In part 3, task C, the autopilot had a stationary error. As plot 14 shows, the ship reaches the reference heading value, resulting in a properly working autopilot. The reason being that the estimate of the rudder bias, based on our model of the system, is accurate enough to actually cancel out the real rudder bias. Hence, the Kalman filter gives a good estimate of the rudder bias.

5.5 Task E - Simulation with current and wave disturbance

The wave filtered ψ is now used as feedback to the regulator, opposed to the measured compass course. Plot 15 shows the response of the system with the measured compass course, while plot 16 displays the rudder input and the estimated bias. When comparing the response to part 4, task D, the greatest improvement is the rudder input. The rudder does not fluctuate like it did without the estimated heading as feedback, resulting in a system less likely to damage the rudder. However, the rudder has some initial oscillations, but seeing the overall improvement of the rudder movement, a configuration with an implemented Kalman filter gives a much better response than the system in part 3, task D. However, a question arises. Why not put a simple low-pass filter on the compass measurement and feed it back to the controller? A well designed low pass filter could probably remove much of ψ_w and give a good feedback to the controller. But as the initial wave disturbance is a Gaussian noise process, and not only a high frequency disturbance, using estimators from a Kalman filter will result in a more reliable and robust system.

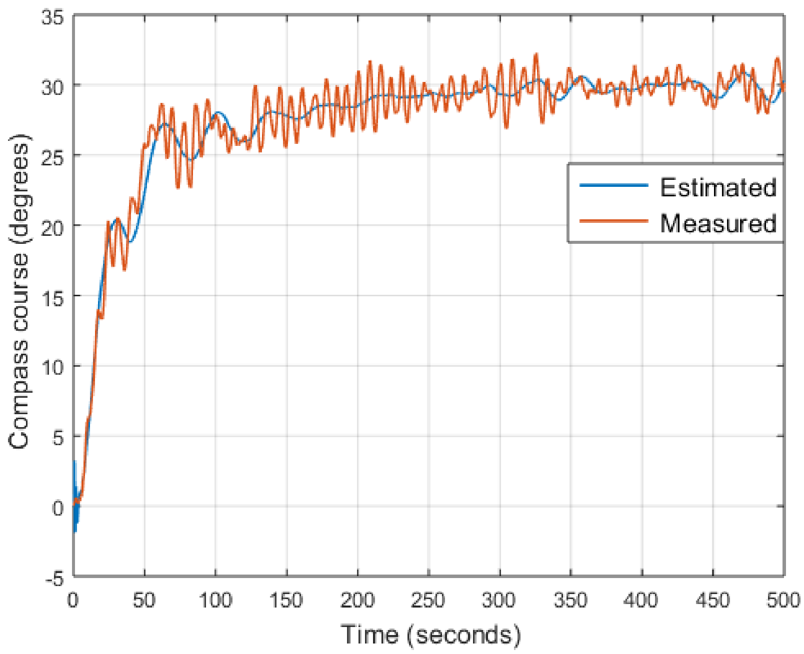

Figure 15: Comparison of the estimate and the measurement

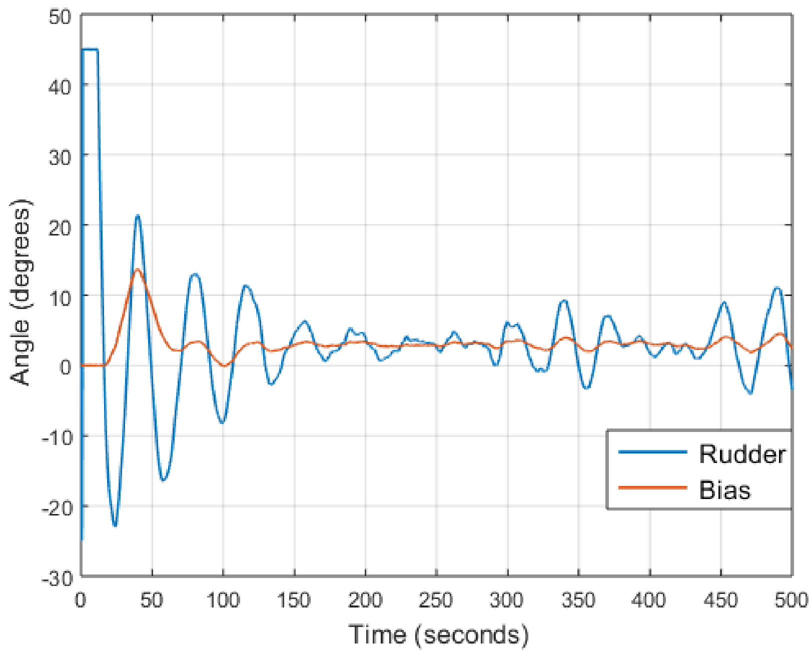

Figure 16: Comparison of rudder and bias

Plot 17 shows the estimated wave disturbance versus the actual wave disturbance. The wave disturbance estimator seem to calculate a smaller disturbance at some frequencies. This error can be traced back to part 2, task D, where a fitted PSD was found. From plot 7 it can be seen that the fitted PSD has a considerable error at lower frequencies, giving a lower value than the estimated PSD. As a result, the estimated wave disturbance has a lower value than the actual wave disturbance at small frequencies, which is seen in plot 17. A way to make the wave disturbance estimation better is to add more complexity to system 2.3. A thought experiment: by initially adding a new augmented state to the model, the integral of w_w, and replace it with w_w in eq. 0.2, the PSD denominator would be of a higher order and could be fitted more accurate. In turn, this would maybe give a better wave disturbance estimation. However, without testing, the response of the system remains unknown, and adding complexity is not always a good idea. As Einstein once said, *"Everything should be made as simple as possible, but not simpler"*, and as long as this system works, we are happy.

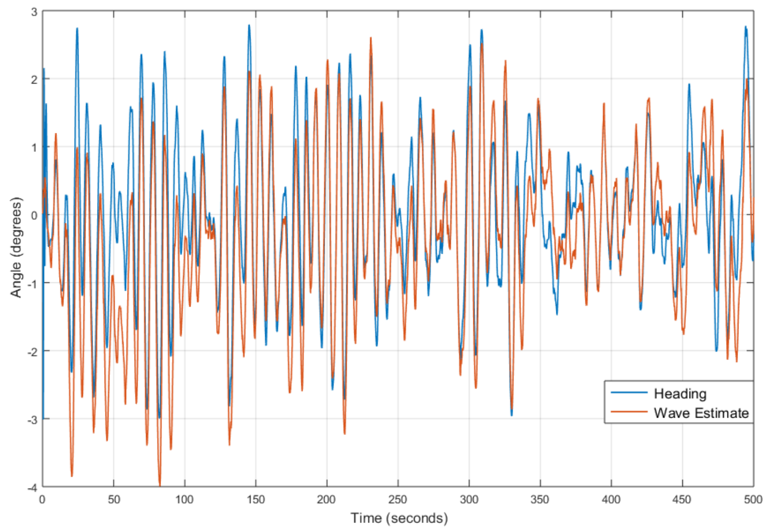

Figure 17: Comparison of heading and wave estimate

6 Appendix A - Simulink Models

Figure 18: Simulink model with sinus wave input

Figure 19: Simulink model with Step Response input

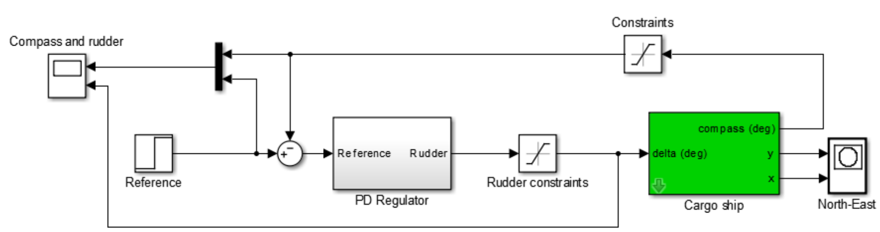

Figure 20: Simulink model with 30° as reference

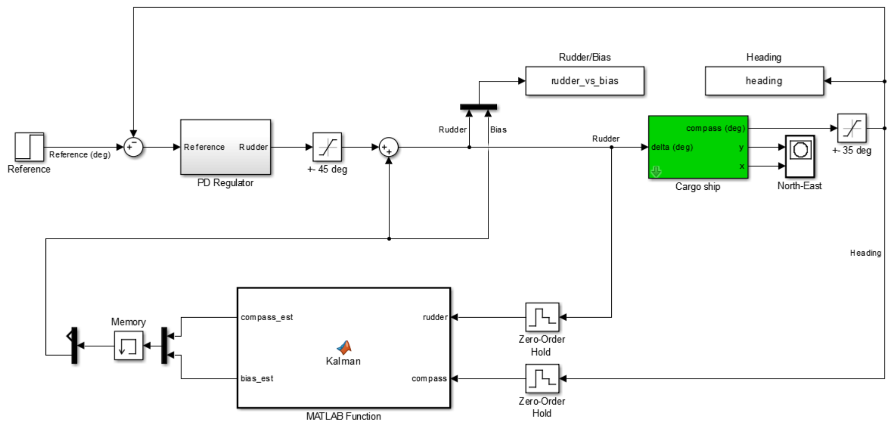

Figure 21: Kalman filter implemented as a MATLAB function

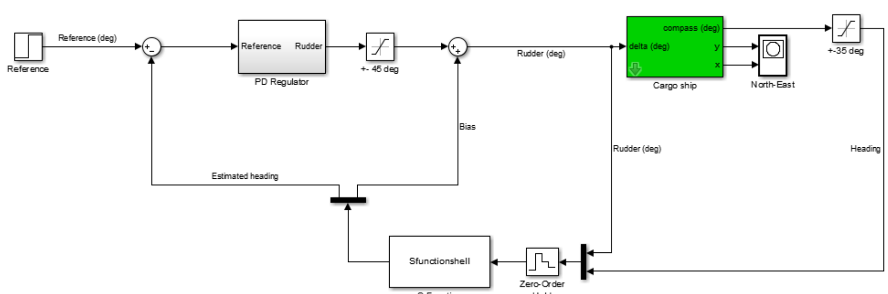

Figure 22: Kalman filter implemented as a SFunctionShell

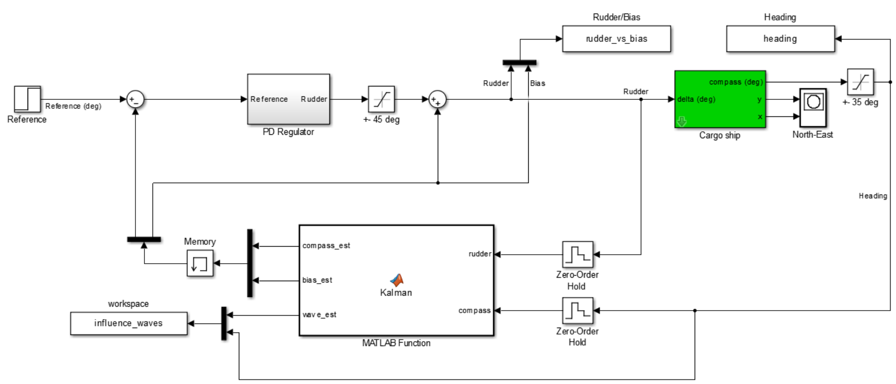

Figure 23: Estimated wave disturbance compared against real wave disturbance

7 Appendix B - MATLAB code

7.1 Main code

```
%{
 Authors:
 Trym Nordal
 Øystein Brox
 Morten Jansrud
 November 2016
%}

%%Task 1b
clear all;
clc;

load('data_w1_smooth');
load('data_w2_smooth');

w1_max = max(w1_smooth.signals.values);
w1_min = min(w1_smooth.signals.values);
w2_max = max(w2_smooth.signals.values);
w2_min = min(w2_smooth.signals.values);

w1 = 0.005;
w2 = 0.05;
A1 = (w1_max-w1_min)/2;
A2 = (w2_max-w2_min)/2;

T = sqrt((A1^2*w1^2 - A2^2*w2^2)/(A2^2*w2^4 - A1^2*w1^4));
K = A1*w1*sqrt(T^2*w1^2+1);

%%Task 2a
load('data_wave');
x = psi_w(2,:)*pi/180;
window = 4096;
noverlap = [];
nttf = [];
fs = 10;

% Power Spectral Density (PSD) function
[pxx,f] = pwelch(x, window, noverlap, nttf , fs);

%Scaling to s/rad & rad/s
pxx = pxx/(2*pi);
f = f*2*pi;
```

```
%%Task 2c
xmax = find(max(pxx) == pxx);
w_0 = f(xmax);

%%Task 2d
lambda = 0.080;
sigma = sqrt(max(pxx));
K_w = 2*lambda*w_0*sigma;
w = linspace(0,2,2000);
P_w = (K_w^2.*w.^2)./(w.^4+(4*lambda^2-2).*w.^2*w_0^2+w_0^4);

%%Task 3a
s=tf('s');
Kpd=0.7612;
Td=T;
Tf=8.5;
Hpd=Kpd*(1+Td*s)/(1+Tf*s);
Hs=K/(Td*s^2+s);
H=Hpd*Hs;

%%Task 4a
A = [0 1 0 0 0;-((w_0)^2) -(2*lambda*w_0) 0 0 0; 0 0 0 1 0; 0 0 0 -1/T -K/T; 0 0 0 0 0];
B = [0; 0; 0; K/T; 0];
C = [0 1 1 0 0];
E = [0 0; K_w 0; 0 0; 0 0; 0 1];

%%Task 5a
Ts= 0.1;
[~,B_d] = c2d(A,B,Ts);
[A_d, E_d] = c2d(A,E,Ts);
C_d = C;

%%Task 5b
load('data_measurement_noise.mat');
m_var = var(measurement_noise);
R = m_var/Ts;

%%Task 5c - Kalman filter
Q = [30 0; 0 10e-6];
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-4];
x_0 = [zeros(10,1); P_0(:)];
data = struct('A_d',A_d,'B_d',B_d,'C_d',C_d,'E_d',E_d,'Q',Q,'R',R,'P',P_0,'x_0',x_0);
```

7.2 Kalman Filter - Matlab Function

```
function [compass_est, bias_est] = Kalman(rudder,compass,A_d,B_d,C_d,E_d,R,Q,P_0)
%#codegen

persistent init_flag x_nextPri P_nextPri

if isempty(init_flag)
    init_flag = 1;
    x_nextPri = [0;0;0;0;0];
    P_nextPri = P_0;
end

I = eye(5);

P_pri = P_nextPri;

x_pri = x_nextPri;

K_k =  P_pri*C_d'*inv(C_d*P_pri*C_d'+R);

x_post = x_pri + K_k*(compass-C_d*x_pri);

P_k = (I-K_k*C_d)*P_pri*(I-K_k*C_d)'+K_k*R*K_k';

x_nextPri = A_d*x_post + B_d*rudder;

P_nextPri = A_d*P_k*A_d'+E_d*Q*E_d';

compass_est = x_post(3);

bias_est = x_post(5);
```

7.3 Kalman Filter - SFunctionShell

```
function [sys,x0,str,ts] = DiscKal(t,x,u,flag,data)
% Shell for the discrete kalman filter assignment in
% TTK4115 Linear Systems.
%
% Author: Jørgen Spjøtvold
% 19/10-2003
%

switch flag,
```

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

  %%%%%%%%%%%%%
  % Outputs   %
  %%%%%%%%%%%%%

  case 3,
    sys=mdlOutputs(t,x,u,data); % mdlOutputs(t,x,u,data) if mathod 2 is used
  %%%%%%%%%%%%%
  % Terminate %
  %%%%%%%%%%%%%

  case 2,
    sys=mdlUpdate(t,x,u,data); %mdlUpdate(t,x,u, data); if method 2 is used

  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) %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     = 2; % 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  = data.x_0; % 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)%mdlUpdate(t,x,u, data); if method 2 is used
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I = eye(5);

P_pri = reshape(x(11:35),5,5);

K_k = P_pri*(data.C_d)'*(data.C_d*P_pri*(data.C_d)'+data.R)^(-1);

x_pri = x(1:5);

x_post = x_pri + K_k*(u(2)-data.C_d*x_pri);

P_k = (I-K_k*data.C_d)*P_pri*(I-K_k*data.C_d)'+K_k*data.R*(K_k)';

x_nextPri = data.A_d*x_post + data.B_d*u(1);

P_nextPri = data.A_d*P_k*(data.A_d)'+data.E_d*data.Q*(data.E_d)';

out = [x_nextPri; x_post; P_nextPri(:)];

sys=out;

function sys=mdlOutputs(t,x,u,data)% mdlOutputs(t,x,u,data) if mathod 2 is used
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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)];

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

References

[1] *Discrete Kalman Filter Applied to a Ship Autopilot.* Itslearning, 2016.

[2] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley Sons, Inc, 2012.

[3] Wikipedia. Kalman-filter, . URL `https://en.wikipedia.org/wiki/Kalman_filter`.

[4] Wikipedia. Welchs method, . URL `https://en.wikipedia.org/wiki/Welch%27s_method`.