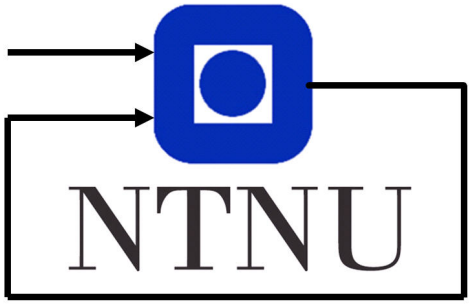
Boat Lab Report

TTK4115 – Linear System Theory

Group 69
Aksel Danielsen
Student 478464
Bernhard Paus Græsdal
Student 478404

November 25, 2018



Department of Engineering Cybernetics

Contents

1 Introduction

In this project, a cargo ship is to be modelled and controlled. The first part of the assignment will be about partly modelling and simulating a continuous system influenced by stochastic signals. This will require using basic identification techniques to find parameters which are not explicitly given by the assignment. Furthermore, basic control theory will be applied to design a simple autopilot. In the end a discrete Kalman filter will be implemented in `MATLAB` and `Simulink` for wave filtering and to estimate disturbances on the system.

The actual cargo ship is simulated in `Simulink`, while the ship will be controlled using a first order Nomoto-approximation as the cargo ship model, provided by the assigment text.

2 Part 5.1 – Identification of the boat parameters

The Nomoto-approximated model for the cargo ship, given by the assignment, is as follows:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\mathbf{w} \quad , \quad y = \mathbf{C}\mathbf{x} + v \\ \mathbf{x} &= \begin{bmatrix} \varepsilon_w & \psi_w & \psi & r & b \end{bmatrix}^T \end{aligned} \tag{1}$$

With the following relationships:

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

Here, w_b, w_w and v are white noise processes. Further, ψ is the average heading, ψ_w is a high frequency component due to the wave disturbance w_w, r is the rate of change for the average heading (without the wave disturbance), ε_w is the integral of the wave influence on the average heading ψ_w, and finally, b is the bias to the rudder angle caused by the current disturbance.

2.1 Task a) Transfer function from δ to ψ

Taking the Laplace transform of eq. (2c) in the assignment text with zero initial conditions gives

$$s\psi = r \tag{3}$$

Taking the Laplace transform of eq. (2d) with zero initial conditions and inserting eq. (3) gives

$$sr = -\frac{1}{T}r + \frac{K}{T}\delta \tag{4}$$
$$\implies H(s) = \frac{\psi}{\delta}(s) = \frac{K}{Ts^2 + s} \tag{5}$$

2.2 Task b) Identifying the boat parameters

The unknown constants K and T can be approximated numerically by observing the response of the system. The compass measurements from the

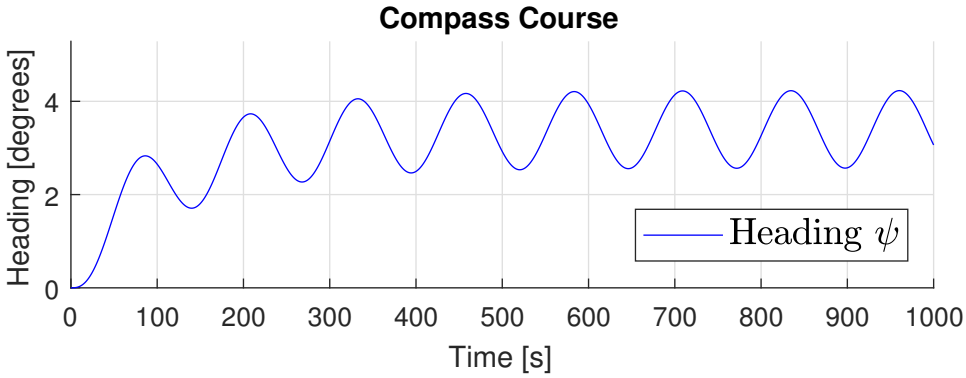


Figure 1: Compass course with input $u = sin(0.05t)$

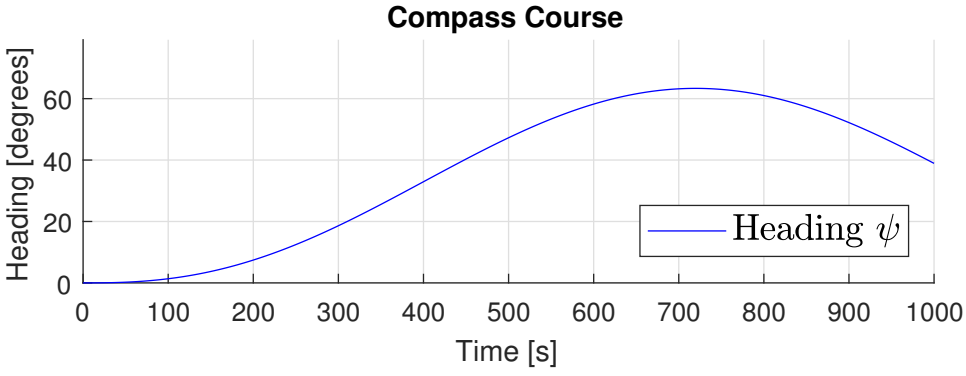


Figure 2: Compass course with input $u = sin(0.005t)$

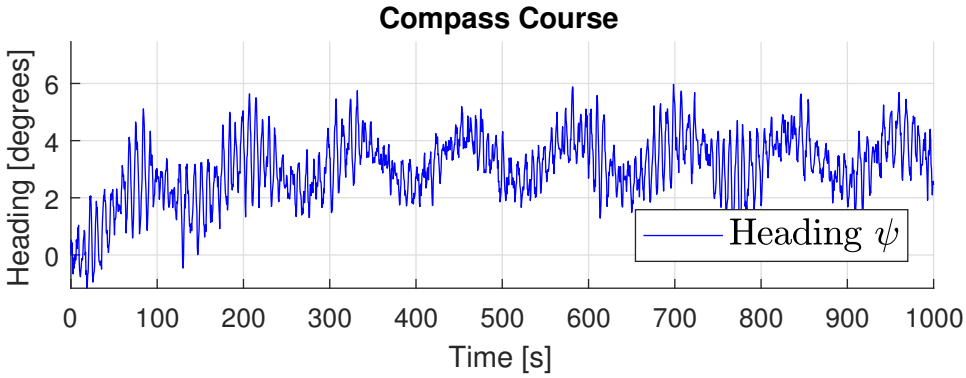


Figure 3: Compass course with input $u = sin(0.05t)$ with wave disturbance added to the simulation

inputs $\sin \omega_1 t$ and $\sin \omega_2 t$ with $w_1 = 0.005$ and $w_2 = 0.05$ are given in figs. 1 and 2. From this plot one can define the following constants

$$h_1 = |H(j\omega_1)| = 29.3575 \tag{6a}$$
$$h_2 = |H(j\omega_2)| = 0.8315 \tag{6b}$$

Inserting $s = j\omega$ in eq. (5) and calculating the magnitude of $H(j\omega)$ gives

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

Combining eqs. (6) and (7) gives two equations for T and K. Solving this system of equations gives the values

$$T = 72.3835 \quad \text{and} \quad K = 0.1561 \tag{8}$$

2.3 Task c) Waves and measurement noise

From fig. 3 one can clearly see the problem with estimating the values for T and K when disturbance is introduced in the model. The disturbances are adding noise to the pure sine waves of the output, making correct estimation of the parameters more difficult. The results from section 2.2 will therefore be used for the rest of the assignment.

2.4 Task d) Evaluating the model

To evaluate the chosen model, both the actual system and the model of the system is simulated while applying a step response in the input u. The results can be seen in fig. 4.

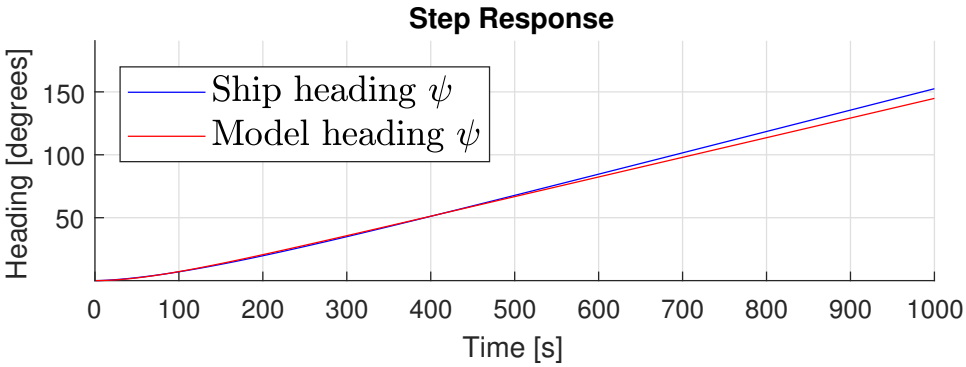


Figure 4: Step response of the model and simulated ship heading

As expected, the model deviates from the actual system. The difference between the model and the actual system seems to be increasing with time. Despite this it does seem that the model is a quite good approximation of the physical system. This is essential for designing a controller and observer for the ship, as these are based on the mathematical model of the system. A perfect model is never realistic, so this imperfection is a discrepancy that must be accepted.

3 Part 5.2 – Identification of wave spectrum model

3.1 Task a) Power Spectral Density (PSD) function

The Power Spectral Density (PSD) function describes the distribution of power into frequency components. In this problem, the PSD function is to be found for the wave disturbance, ψ_w, shown in figure fig. 5. Finding this function is done using the `MATLAB` function `pwelch` which, given the measurement and its associated sampling frequency, returns the estimated PSD function. The input is given in Hz, and in order to get the PSD function in terms of radians, the scaling factors 2π and $\frac{1}{2\pi}$ are applied to the input and the PSD function, respectively. The estimated PSD function, S_{ψ_w} can be seen in fig. 6.

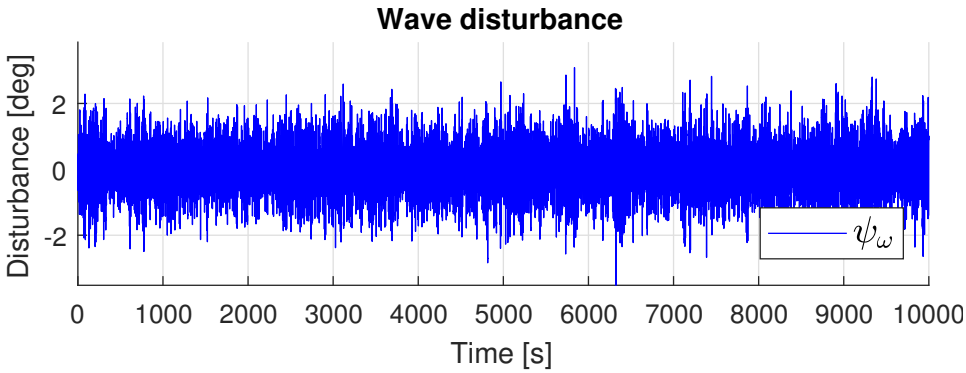


Figure 5: ψ_ω – Influence from the wave disturbance

3.2 Task b) Transfer function and analytical PSD

The transfer function from w_w to ψ_w is found using eqs. (9) and (10).

$$\dot{\psi}_w = -\omega_0^2 \varepsilon - 2\lambda\omega_0\psi_w + K_w w_w \tag{9}$$

$$\dot{\varepsilon}_w = \psi_w \tag{10}$$

taking the laplace transform of eq. (10) gives

$$s\varepsilon_w = \psi_w \implies \varepsilon_w = \frac{\psi_w}{s} \tag{11}$$

Inserting this into the laplace transform of eq. (9) gives

$$s\psi_w = -\omega_0^2 \frac{\psi_w}{s} - 2\lambda\omega_0\psi_w + K_w w_w \tag{12}$$

Which in turns gives the desired transfer function:

$$\frac{\psi_w}{w_w}(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \tag{13}$$

Finding the analytical expression for the PSD function of ψ_w is found using the following relationship [1, Eq 3.2.2]:

$$P_{\psi_w}(j\omega) = |H(j\omega)|^2 P_{w_w}(j\omega) \tag{14}$$

It is stated in the task that w_w is a zero mean white noise with unity variance. According to [1, Chapter 2.8] this gives

$$P_{\psi_w}(j\omega) = A = 1 \tag{15}$$

Using eqs. (14) and (15) the analytical PSD function for ψ_w is found:

$$P_{\psi_w}(j\omega) = |H(j\omega)|^2 = \frac{K_w^2 w^2}{(\omega_0^2 - \omega^2)^2 + (2\lambda\omega_0\omega)^2} \tag{16}$$

3.3 Task c) Modal peak frequency

The modal peak frequency w_0 is found by finding frequency corresponding to the peak value for the PSD function. This frequency contains the most power. The `max` function in `MATLAB` returns the maximum value for the estimated PSD function S_{ψ_w}, in fig. 6. The modal peak frequency is found to be $\omega_0 = 0.7823$. The corresponding intensity σ^2 given by the peak value is $\sigma^2 = 2.5997$.

3.4 Task d) Damping factor

Lastly, the damping factor λ is to be identified in order to complete the model for the wave response. To do so K is defined such that $K_w = 2\lambda\omega_0\sigma$. Then λ is found using the `MATLAB` function `lsqcurvefit`, which, by minimizing the sum of square errors, makes an attempt to fit a given function to some given values. This gives $\lambda = 0.0827$. The final result, both for the estimated PSD function S_{ψ_w} and the analytical PSD function P_{ψ_w}. can be seen in fig. 6. The analytic solution seems to be a good approximation to the true measured noise.

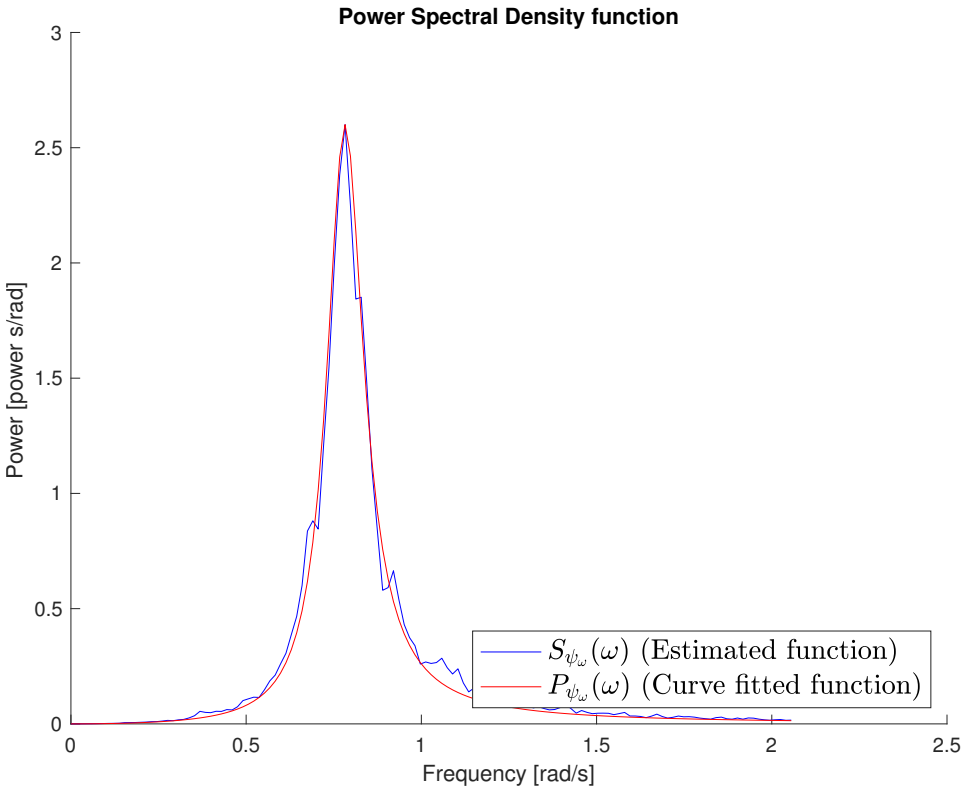


Figure 6: Comparison between the estimated and analytic PSD functions.

4 Part 5.3 – Control system design

In this section, an autopilot is to be designed for the ship. The purpose of this is to be able to control the course angle ψ to follow a given course angle ψ_r. A PD controller will therefore be implemented, and will be tested under different conditions with the different disturbances on the system. It will become clear during this part that a PD controller is not sufficient on its own to achieve the desired performance.

The actual ship model is only valid inside a boundary of $\psi = \pm 35$ degrees. This presents no problem however, as none of the simulations in this task is outside of this interval. In addition, the rudder input is saturated to the same range to make the simulation more realistic.

4.1 Task a) PD controller

The PD controller to be implemented is given by

$$H_{pd}(s) = K_{pd}\frac{1 + T_d s}{1 + T_f s} \tag{17}$$

Then the open loop system is given in is entirety by

$$\begin{aligned} H(s) &= H_{pd}(s) \cdot H_{ship}(s) && (18)\\ &= K_{pd}\frac{1 + T_d s}{1 + T_f s} \cdot \frac{K}{s(1 + Ts)} && (19)\\ &= K_{pd}\frac{K}{s(1 + T_f s)} && (20) \end{aligned}$$

for $T_d = T$ such that the derivative time constant cancels the transfer function time constant. The desired stability margins of the system is $w_c = 0.1$ rad/s and PM = 50 degrees. For $w = w_c$ the gain is 0 db. The optimal K_{pd} is therefore given by

$$\begin{aligned} |H(j\omega_c)| &= K_{pd}\frac{K}{\omega_c\sqrt{1 + T_f^2 w_c^2}} = 1 && (21)\\ K_{pd} &= \frac{\omega_c\sqrt{1 + T_f^2 w_c^2}}{K} && (22) \end{aligned}$$

The Bode diagram for the open loop system in eq. (20) can be seen in fig. 7. By tuning T_f and observing the Bode diagram, the value $T_f = 8.4$ gave the desired phase margin.

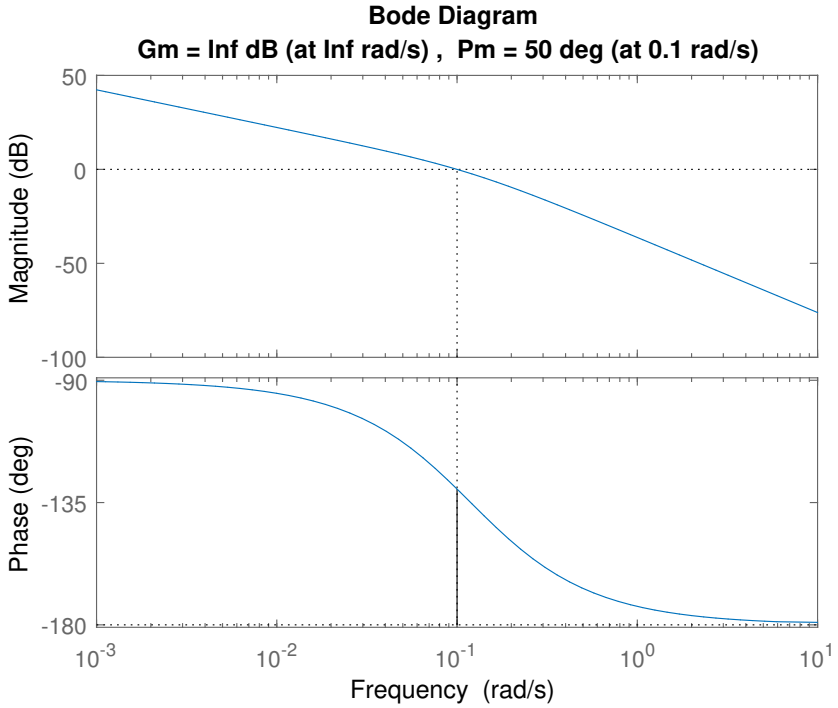


Figure 7: Bode diagram for the open loop system in Task a)

4.2 Task b) Simulation without disturbances

The resulting compass measurments from the simulation can be seen in fig. 8, and the rudder input can be seen in fig. 9. The autopilot seems very satisfactory, converging to the desired reference and without oscillations in the input δ. The controller uses approximately 250 seconds to change the heading by 30 degrees, which is acceptable for a large ship.

The differentiation effect in the PD controller raises the phase margin allowing for a more aggressive controller, while maintaining stability. This means quicker control and better response to changes in the reference heading ψ_r,

4.3 Task c) Simulation with current disturbance

The current disturbance introduces a stationary error in the system, as seen in fig. 10. This is caused by the rudder bias b which the PD controller alone is not able to correct. The plot of the rudder input in fig. 11 illustrates the issue. The system reaches the steady state when the controller output δ is equal to the rudder bias. Introducing integral effect in the controller would help in solving this problem.

In part 5 in section 6 it is shown how a Kalman filter can be used to

estimate, and correct for, the rudder bias b using feed forward.

4.4 Task d) Simulation with wave disturbance

The plot of the compass fig. 12 shows that the ship is able to maintain the desired reference course, but oscillates with approximately 2 degrees around the reference. This is causing the controller to aggressively attempt to compensate for the disturbance ψ_w as seen in fig. 13. Such oscillations will cause tear and damage to the rudder over time, and is therefore not desirable.

Smoothing the signal using a low pass filter could reduce the oscillations, but this problem can best be solved by introducing a state observer that can estimate the average heading ψ. The optimal way of applying this technique is through a Kalman filter, and part 5 in section 6 shows how this can be implemented.

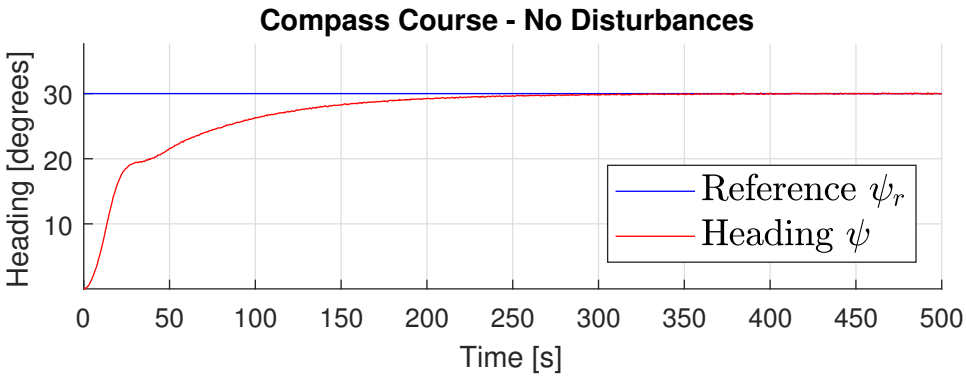


Figure 8: The compass heading from simulating the model without disturbances

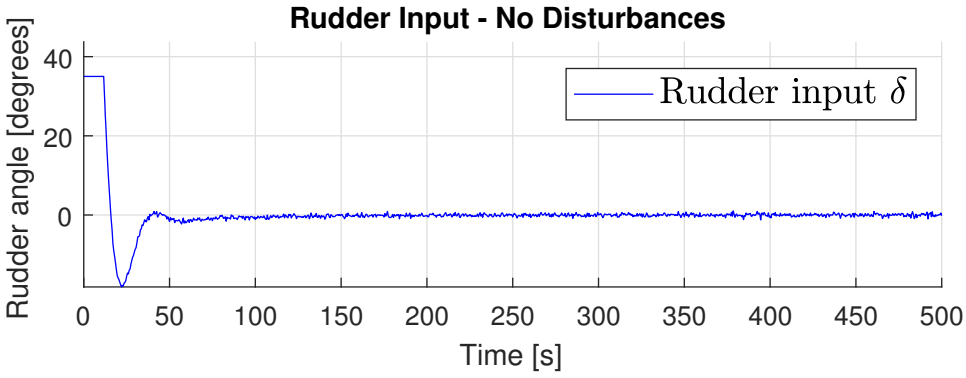


Figure 9: The rudder input from simulating the model without disturbances

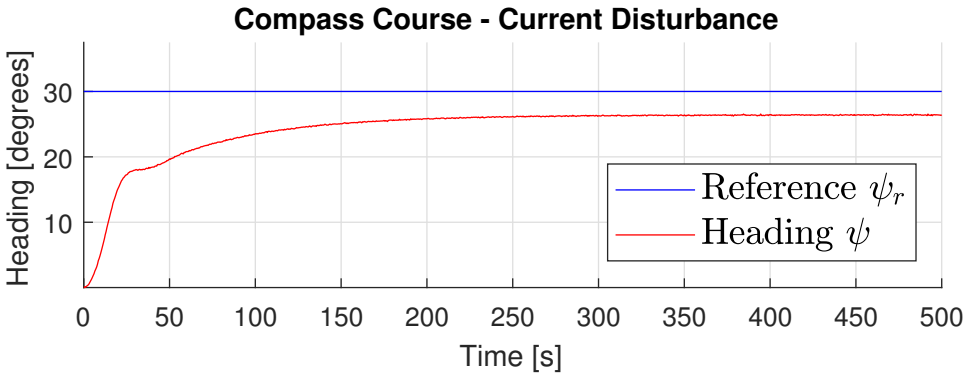


Figure 10: The compass heading from simulating the model with a current disturbance

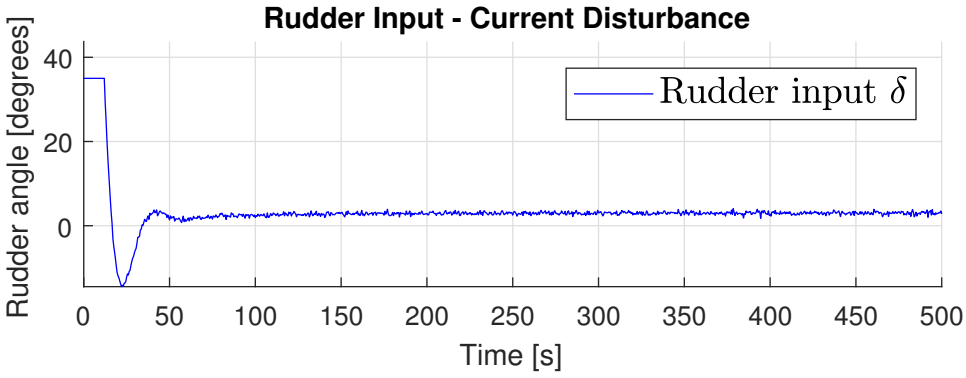


Figure 11: The rudder input from simulating the model with a current disturbance

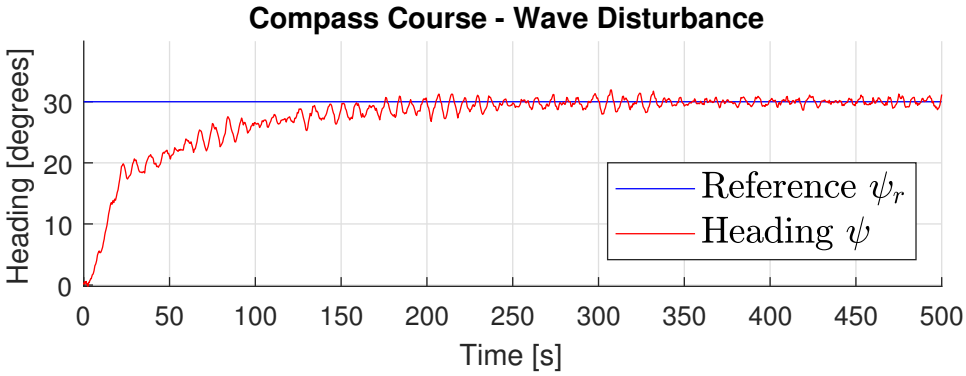


Figure 12: The compass heading from simulating the model with a wave disturbance

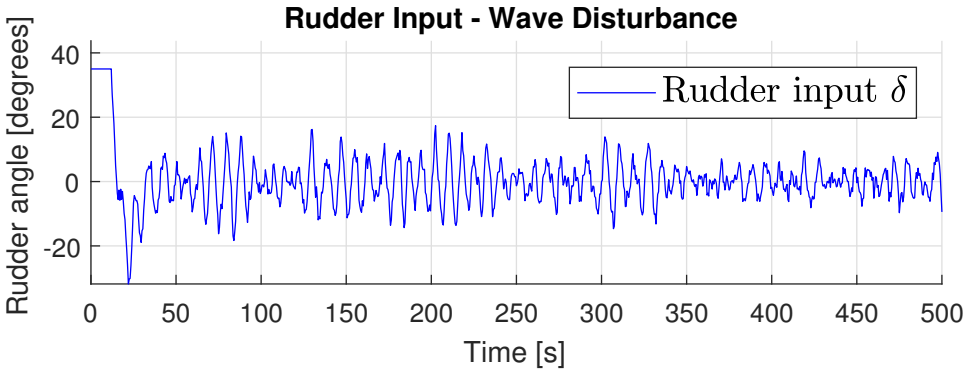


Figure 13: The rudder input from simulating the model with a wave disturbance

5 Part 5.4 – Observability

In this part the observability of the system will be determined. The observability of the system will be determined without disturbances, with the different disturbances by themselves and with all disturbances at the same time.

In order to implement a Kalman filter, which is done in Part 5 in section 6, it is a precondition that the system is observable. It is therefore no surprise that the results of this part will show that the system is observable with any one of the different disturbances.

5.1 Task a) State space model

By using eq. (2) the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{E}$ where found to be

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{-1}{T} & \frac{-K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{E} = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{23}$$

5.2 Task b) Observability without disturbances

Without disturbances, the states ξ_w, ψ_w and b are not included in the model reducing the system to

$$\mathbf{x} = \begin{bmatrix} \psi \\ r \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{24}$$

The `MATLAB` function `obsv` can be used to calculate the observability matrix

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

If the observability matrix is at full rank the system is observable by [2, Theorem 6.01]. The `rank` function in `MATLAB` gives $rank(\mathcal{O}) = 2$ and therefore the system in eq. (24) is observable.

5.3 Task c) Observability with current disturbance

With current disturbance the rudder bias b is reintroduced to the system giving

$$\mathbf{x} = \begin{bmatrix} \psi \\ r \\ b \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{25}$$

The `obsv` command can then be used to calculate the observability matrix

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

By the `rank` function in `MATLAB` this matrix has rank 3, which is full rank. Therefore the system given in eq. (25) is observable by [2, Theorem 6.01].

5.4 Task d) Observability with wave disturbance

With wave disturbance the states ξ_w and ψ_w are reintroduced to the system from eq. (24) giving

$$\mathbf{x} = \begin{bmatrix} \xi_w \\ \psi_w \\ \psi \\ r \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{K}{T} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \tag{26}$$

The `obsv` command can be used to calculate the observability matrix

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

By the `rank` function in `MATLAB` this matrix has rank 4, which is full rank. Therefore the system given in eq. (26) is observable by [2, Theorem 6.01].

5.5 Task e) Observability with all disturbances

Using the full system matrices from eq. (23) in the calculation of the observability matrix gives

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

Using the `rank` function in `MATLAB` this matrix is also at full rank, which, by [2, Theorem 6.01], means that the complete system is also observable.

6 Part 5.5 – Discrete Kalman filter

In order to improve the performance of the system, a Kalman filter is to be implemented. Doing so will make it possible to attain estimates for the bias b, the heading ψ, and the high-frequency wave induced notion on the heading ψ_w, which in turn will make it possible to reduce wear and tear on the system by implementing wave filtering.

A Kalman filter requires a system to be observable, which was confirmed in Part 5.4 in section 5. This is due to the nature of the Kalman filter; a Kalman filter estimates all the states, constantly comparing them to the measured output. An unobservable state means that no information about the state may be obtained through the output, which will make the Kalman filter estimate for the state not converge on a meaningful solution.

As in Part 5.3 in section 5, the actual ship model is only valid inside a boundary of $\psi = \pm 35$ degrees, and the rudder input is saturated. This presents no problem however, as none of the simulations in this task is outside of this interval.

6.1 Task a) Discretization

In order to use a Kalman filter, the system has to be discretized. This is done using the `Matlab` function `c2d`, which uses zero-order-hold in order to discretize a system. The function needs to be called twice, once for the pair $\mathbf{A}, \mathbf{B}$ and once for the pair $\mathbf{A}, \mathbf{E}$, in order to get the discretized matrices $\mathbf{A_d}$,$\mathbf{B_d}$, and $\mathbf{E_d}$ ($\mathbf{C_d}$ remains the same as $\mathbf{C}$ in the continuous system):

$$\begin{aligned}
\mathbf{A_d} &= \begin{bmatrix} 0.9970 & 0.0993 & 0 & 0 & 0 \\ -0.0607 & 0.9841 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0999 & 0 \\ 0 & 0 & 0 & 0.9986 & -0.0002 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{B_d} &= \begin{bmatrix} 0 \\ 0 \\ 0.0000108 \\ 0.0002155 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{E_d} = \begin{bmatrix} 0.0010 & 0 \\ 0.0207 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1000 \end{bmatrix}
\end{aligned} \tag{27}$$

6.2 Task b) Variance of the measurement noise

In order to apply a Kalman filter, $\mathbf{R}$ and $\mathbf{Q}$ have to be found. $\mathbf{R}$ and $\mathbf{Q}$ are the covariance matrices for $\mathbf{w}$ and $\mathbf{v}$, respectively, but since v is a scalar, $\mathbf{R}$ reduces to R. $\mathbf{Q}$ is given in the task c), but R remains to be found. Since R is a scalar, it is equal to the variance of v.

First v is measured while simulating the system, applying only the measurement noise and setting $\delta = 0$. Then the variance of this measurement vector is calculated using the `Matlab` function `var`, which gives $\sigma^2 = 0.0020$. R is then found by dividing this result by the sampling period $T_s = 0.1$:

$$R = \frac{\sigma^2}{T_s} = 0.0202 \tag{28}$$

We have now obtained a complete model of the system and can begin implementing the Kalman filter.

6.3 Task c) Implementing the Kalman filter

The matrix $\mathbf{Q}$, as well as the initial values for $\mathbf{P_k^-} = \mathbf{P_0^-}$ and $\hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{0}$, are given in the assignment:

$$\begin{aligned} \mathbf{P_0^-} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-6} \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{x}}_\mathbf{0}^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{29}$$

The algorithm for the discrete Kalman filter implemented on the system can be described mathematically as repeating four steps every kth iteration. In the mathematical description, $\hat{\mathbf{x}}_\mathbf{k}$ and $\mathbf{P_k}$ *a priori* are denoted as $\hat{\mathbf{x}}_\mathbf{k}^-$ and $\mathbf{P_k^-}$, and $\hat{\mathbf{x}}_\mathbf{k}$ and $\mathbf{P_k}$ *a posteriori* is denoted as $\hat{\mathbf{x}}_\mathbf{k}$ and $\mathbf{P_k}$, respectively. The steps presented here are the same as the steps presented in [1, Chapter 4.2], with some modifications to the variable names. The steps are as follows:

1. Compute the Kalman gain $\mathbf{K_k}$:

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{C_k}^T(\mathbf{C_k}\mathbf{P_k^-}\mathbf{C_k}^T + \mathbf{R})^{-1} \tag{30}$$

This is the optimal gain matrix for minimizing the variance in the estimated state matrix $\hat{x}$.

2. Update the estimate $\hat{\mathbf{x}}_k^-$ with measurement $\mathbf{y}$:

$$\hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\mathbf{y_k} - \mathbf{C_k}\hat{\mathbf{x}}_\mathbf{k}^-) \tag{31}$$

This is the core step of the Kalman filter. The predicted *a priori* estimate is adjusted by the measurement from the sensors using the optimal gain matrix K_k.

3. Compute error covariance for the updated estimate:

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k C_k})\mathbf{P_k^-}(\mathbf{I} - \mathbf{K_k C_k})^T + \mathbf{K_k R_k K_k}^T \tag{32}$$

This Matrix contains the error covariances to be minimized by the Kalman Filter gain.

4. Project ahead (for the next iteration, $k+1$):

$$\mathbf{\hat{x}_{k+1}^-} = \mathbf{A_k \hat{x}_k} + \mathbf{B_k} u_k \tag{33a}$$

$$\mathbf{P_{k+1}^-} = \mathbf{A_k P_k A_k}^T + \mathbf{EQE}^T \tag{33b}$$

The last step is to calculate *a priori* estimates for the next iteration.

The algorithm for the discrete Kalman filter is implemented as `MATLAB` code in `Simulink`, using the `MATLAB function` block. $\mathbf{P_k}$ and $\mathbf{\hat{x}_k}$ are initialized to $\mathbf{P_0^-}$ and $\mathbf{\hat{x}_0^-}$ the first time the algorithm runs.

The system in itself is continuous, but in order to utilize the discrete Kalman filter, the values given as parameters to the Kalman filter has to be discretized. A `zero-order-hold Simulink` block is therefore added to the compass measurement ψ and the rudder command δ before they are given as parameters to the Kalman filter. The same sampling period as before, $T_s = 0.1$, is used.

An algebraic loop is a problem which can arise in `Simulink` when it tries to start a simulation if there are circular dependencies in the model. This is the case with the Kalman filter, as the Kalman filter depends on the control input, while at the same time the input depends on the output from the Kalman filter due to feedback. To avoid this problem, `memory` blocks are added after all the outputs from the Kalman filter in the `Simulink` model. This completes the implementation of the Kalman filter.

6.4 Task d) Feed forward for rudder bias

The constant error seen in section 4.3 is a result from the rudder bias b. In order to compensate for this, a feed forward is to be implemented on the system using the estimated rudder bias $\hat{b}$ from the Kalman filter. The result can be seen in fig. 14. The corresponding rudder input δ and the estimated bias $\hat{b}$ can be seen in fig. 15.

The Kalman filter estimates the bias, which is then feed forwarded. Given a perfect bias estimate, this is equivalent to not having a bias at all. The

estimated bias $\hat{b}$ can be seen to be approaching 4 degrees as time passes, which in turn gives a constant rudder input $u = 4$ degrees. It is clear from the plots that the constant error is now fully compensated for, and hence the overall performance of the autopilot is greatly increased compared to Part 3 in section 4.

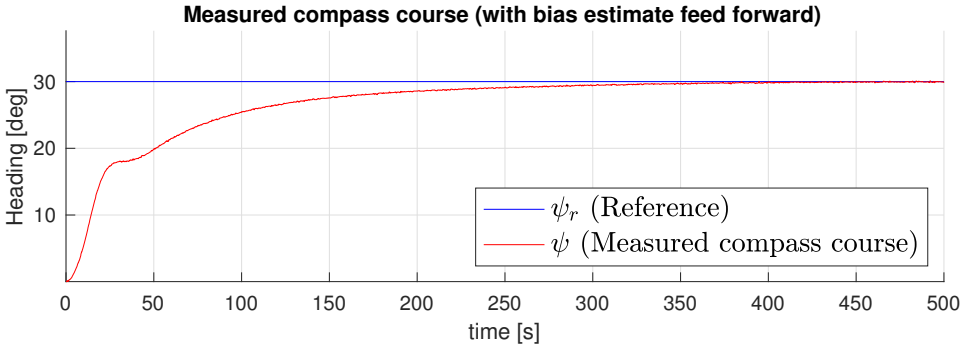


Figure 14: Measured compass course compared to reference, when the estimated rudder bias is being feed forwarded.

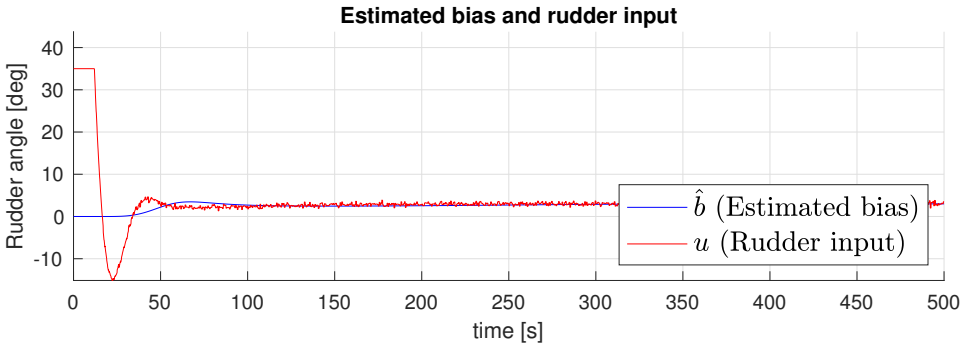


Figure 15: Comparison of estimated rudder bias and rudder input with rudder bias feed forward

6.5 Task e) Wave filtering

In this task, wave filtering is used to separate the wave disturbance ψ_w from the compass heading ψ in the measured output y. This is implemented by using the estimated compass heading $\hat{\psi}$ from the Kalman filter as feedback to the controller. The system is then simulated with both wave disturbance w_w and current disturbance w_b.

The measured results for the compass heading can be seen in fig. 16. The corresponding rudder input δ and the estimated rudder bias $\hat{b}$ can be seen in in fig. 17. Compared to fig. 13 the rudder input oscillations is gone

after the transient response has died out, and the ship is still maintaining the desired heading. Figure fig. 18 shows how the wave influence ψ_w is accurately estimated after about 130 seconds. Overall this is much better performance than the naive model with only a PD controller, and displays the Kalman filters capabilities for noise suppression by state estimation.

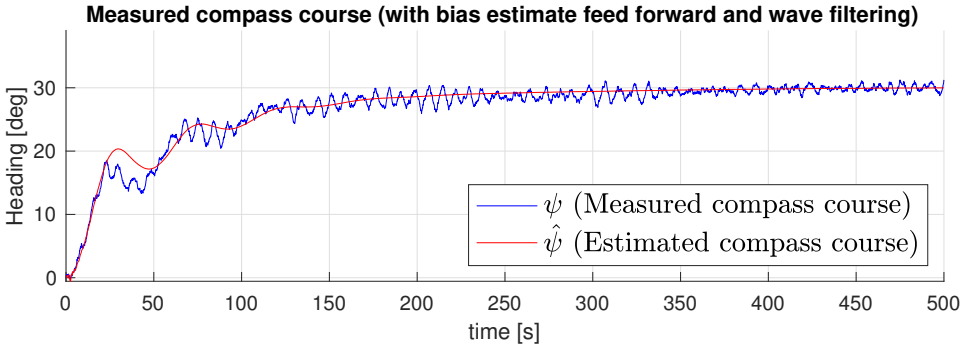


Figure 16: Measured compass course compared to the reference, when wave disturbances are enabled and wave filtering is utilized.

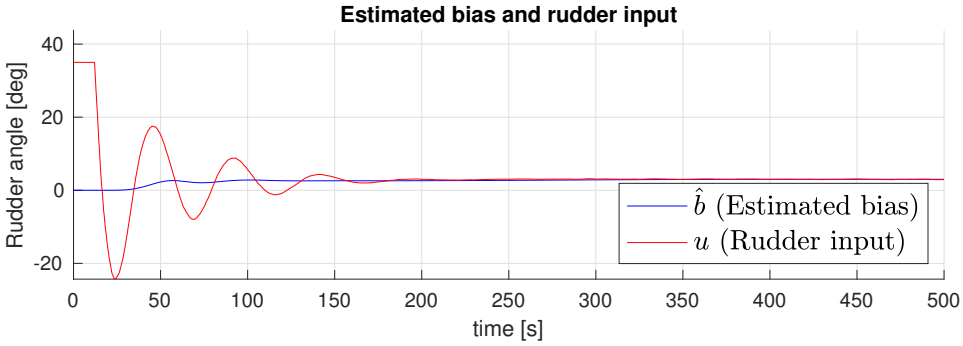


Figure 17: Comparison of estimated rudder bias and rudder input with rudder bias feedforward and wave filtering

6.6 Task f) Covariance of the disturbance

The process covariance matrix $\mathbf{Q}$ contains an estimate of the variance in ψ_w and b. It therefore serves as an indicator of how much the Kalman filter should adjust its own *a priori* state prediction with the measurement from the sensors. Normally, this matrix would have to be calculated, but here the matrix $\mathbf{Q}$ was given in the assignment. In this task, different values for $\mathbf{Q}$ will be tested and simulated.

Choosing too low values for $\mathbf{Q}$ will result in a filter which is underestimating the process disturbance. The Kalman filter will therefore underestimate

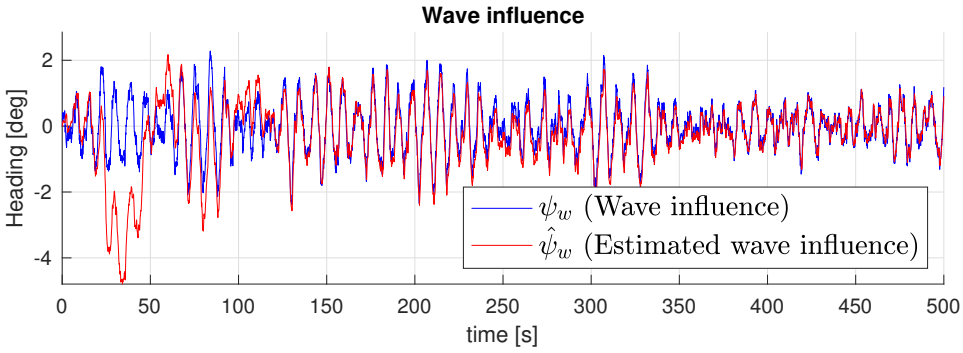


Figure 18: The actual wave influence shown in comparison to the wave influence estimated by the Kalman filter.

the variance in the process disturbance, and in turn rely too heavily on its own prediction. The corresponding results can be seen when the system is simulated with values of $\mathbf{Q}$ that are far smaller than before. Figure 21 shows how this is causing a small error in the compass heading, and fig. 19 shows how the wave disturbance is now underestimated.

On the other hand, choosing too high values for $\mathbf{Q}$ will result in a filter which is compensating too much for the process disturbance, due to an overestimate in the disturbances variance. This can be seen when the different entries of $\mathbf{Q}$ are greatly increased. Figure 20 shows the oscillating effect of increasing the entry in $\mathbf{Q}$ corresponding to the bias b. The resulting state estimates is seen to be overshooting as a response to the increased disturbance. Figure 22 shows how the initial wave disturbance is greatly overestimated, and the resulting error in the compass heading is shown in fig. 23.

The experiments in this task demonstrate the importance of choosing appropriate $\mathbf{Q}$ matrices for designing the Kalman filter.

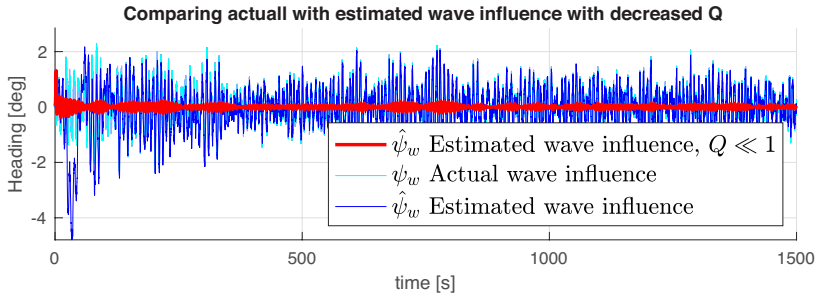


Figure 19: The estimated wave influence from part e) compared with $Q \ll 1$

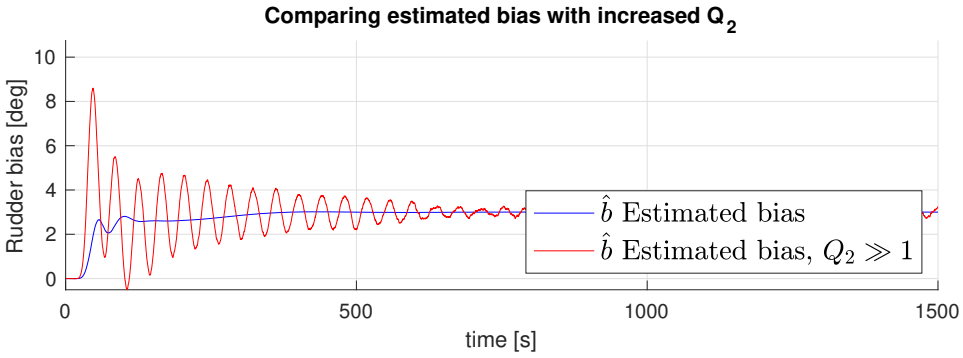


Figure 20: The estimated bias from part e) compared with $Q_2 \ll 1$

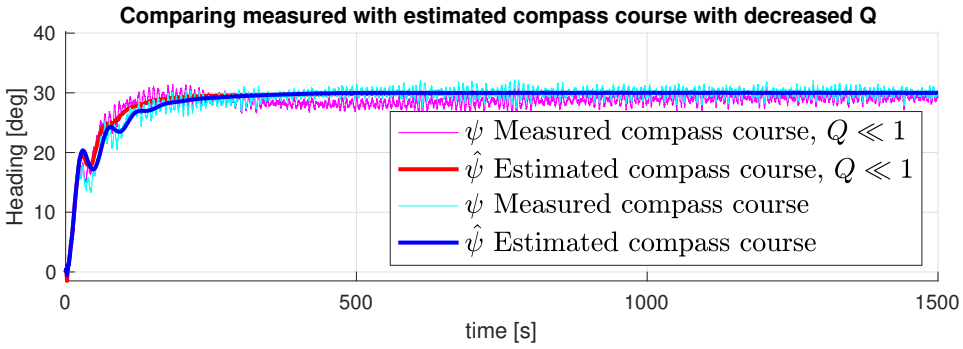


Figure 21: The compass course from part e) compared with $Q \ll 1$

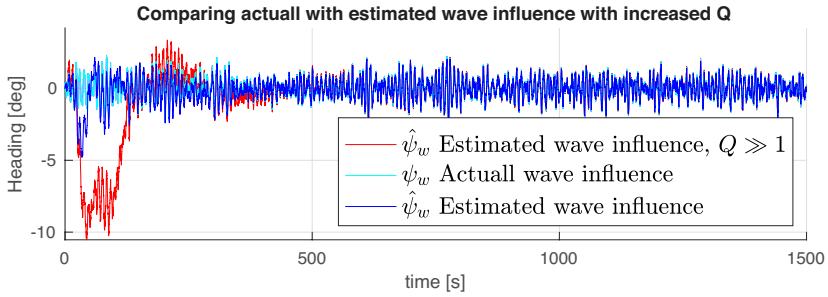


Figure 22: The estimated wave influence from part e) compared with $Q \gg 1$

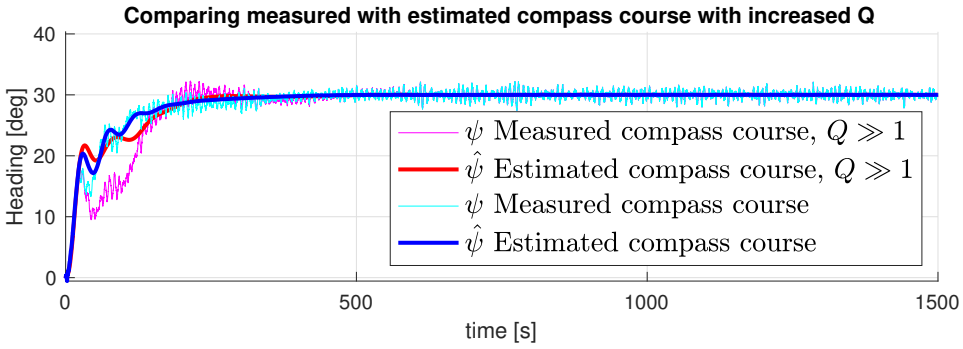


Figure 23: The compass course from part e) compared with $Q \gg 1$

7 Conclusion

The assignment has made clear some key aspects in the limitations of modelling and classical control theory.

Utilizing only classical control theory proved to be insufficient when working with a system influenced by stochastic signals. This could easily be seen with the simple PD controller implemented in Part 3 from section 4, as it gave rise to multiple problems once the stochastic disturbances were added to the model: constant deviation from the reference, as well as a rapidly varying input due to high frequency disturbances, which in time will tear on the equipment and the ship.

In Part 5 (section 6), a discrete Kalman filter was employed to separate the stochastic disturbances from the system states in the measured output. This made possible a feed forward of the disturbance in the rudder bias, which in turn solved the problem with the stationary deviation from the reference. Further, by separating the wave disturbance from the compass heading (wave filtering), the rapidly varying rudder angle from Part 3 (when using only a PD controller) was replaced by a steadier, more stable input, which in time will reduce overall wear and tear on the system.

To conclude, the assignment has given some valuable insights regarding modelling and controlling real-world systems, which, in reality, always will be influenced by stochastic signals.

A MATLAB Code

A.1 Part 5.1

```
%% 5.1 b)

h1_amp = 29.3575; % From plot measurement
h2_amp = 0.8315; % From plot measurement

w1 = 0.005;
w2 = 0.05;

syms K_ T_
eqs = [
    h1_amp == K_/sqrt(T_^2*w1^4 + w1^2),
    h2_amp == K_/sqrt(T_^2*w2^4 + w2^2)
    ];

[solK, solT] = solve(eqs, [K_ T_]);
solK = double(solK);
solT = double(solT);

K = solK(1);
T = solT(2);
```

```
%% 5.1 c)

h1_noise = (63.48 - 3.915)/2;
h2_noise = (5.371 - 1.979)/2;

sine_freq = 0.05;
```

```
%% 5.1 d)
run('p5p1b.m');

H_ship = tf(K, [T 1 0]);
```

A.2 Part 5.2

```
% 5.2 a)
load('Common/wave.mat');

fs = 10; % From assigment
window = 4096; % From assigment

t = psi_w(1,:);
x = psi_w(2,:);

[pxx,f] = pwelch(x, window, [], [], fs);

% Convert to radians
w = 2*pi*f;
pxx_radians = pxx / (2*pi);

%% 5.2 d)
w_0_index = find(pxx_radians==max(pxx_radians));
w_0 = w(w_0_index);

A = 1; % Unity variance white noise
sigma_squared = max(pxx_radians);
sigma = sqrt(sigma_squared);
lambda = 1; % Initial trial value for lsqcurvefit

fun = @(x, w)((2*x*w_0*sigma)^2 * w.^2) ./ ...
    ((w_0^2 - w.^2).^2 + (2*x(1)*w_0*w).^2);


param_vector = lsqcurvefit(...
        fun,...
        lambda,...
        w, pxx_radians);

lambda = param_vector(1); % Get value from least square fit
K_w = 2*lambda*w_0*sigma; % Update K_w with correct value

P_phi_w = (K_w^2 * w.^2) ./ ...
    ((w_0^2 - w.^2).^2 + (2*lambda*w_0*w).^2);
```

A.3 Part 5.3

```
%% Part 5.3 a)
run('../part_5_1/p5p1d.m');

T_d = T;
T_f = 8.4;
w_c = 0.1;
K_pd = sqrt(w_c ^2 + (T_f * w_c^2)^2) / K;
phase_margin = 50;
H_pd = K_pd * tf([T_d 1],...
                 [T_f, 1]);
H_ol = H_pd * H_ship;
```

A.4 Part 5.4

```
%% Part 5.4 a)

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];

%% Part 5.4 b)
A_no_dist = [0 1;
             0 -1/T];

C_no_dist = [1 0];
ob_no_dist = obsv(A_no_dist,C_no_dist);
obs_no_dist = (rank(ob_no_dist) == size(A_no_dist,1));
fprintf('System is observable without disturbances: %d\n', ...
    obs_no_dist);

%% Part 5.4 c)
A_curr_dist = [0 1 0;
               0 -1/T -K/T;
               0 0 0];

C_curr_dist = [1 0 0];
ob_curr_dist = obsv(A_curr_dist,C_curr_dist);
obs_curr_dist = (rank(ob_curr_dist) == size(A_curr_dist,1));
fprintf('System is observable with current disturbance: %d\n', ...
    obs_curr_dist);

%% Part 5.4 d)
A_wave_dist = [0 1 0 0;
               -w_0^2 -2*lambda*w_0 0 0;
```

```
43                  0 0 0 1
44                  0 0 0 -1/T];
45
46  C_wave_dist = [0 1 1 0];
47  ob_wave_dist = obsv(A_wave_dist,C_wave_dist);
48  obs_wave_dist = (rank(ob_wave_dist) == size(A_wave_dist,1));
49  fprintf('System is observable with wave disturbance: %d\n', ...
50      obs_wave_dist);
51
52  %% Part 5.4 e)
53  ob = obsv(A, C);
54  obs = (rank(ob) == size(A, 1));
55  fprintf('System is observable with all disturbances: %d\n', ...
56      obs);
```

A.5 Part 5.5

```
%% Part 5.5 a)
% Continous state space model from 5.4 a)
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];


 % Only discretize A, B and E
C_d = C;

T_s = 0.1; % seconds (10 Hz)

[~, B_d] = c2d(A, B, T_s);
[A_d, E_d] = c2d(A, E, T_s);
```

```
%% Part 5.5 b)
run('p5p5a.m');
sim('p5p5bx.mdl');
R_sigma_squared = var(v_measurement.data); % [degrees^2]
R = R_sigma_squared / T_s;
```

```
%% Part 5.5 c)
run('p5p5b.m');
P_0_apriori = diag([1 0.013 pi^2 1 2.5*10^(-3)]);

x_hat_0_apriori = [0 0 0 0 0]';

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

% Create struct for Kalman function
```

```
10 system = struct('A_d', A_d, 'B_d', B_d,...
11     'C_d', C_d, 'E_d', E_d,...
12     'P_0_apriori', P_0_apriori,...
13     'x_hat_0_apriori', x_hat_0_apriori,...
14     'R', R, 'Q', Q);
```

B Simulink diagrams

B.1 Part 5.1

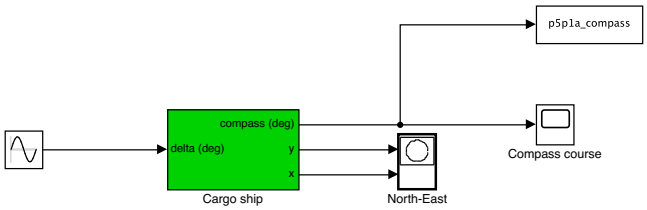


Figure 24: Simulink diagram used in Part 1 b), c) and d)

B.2 Part 5.3

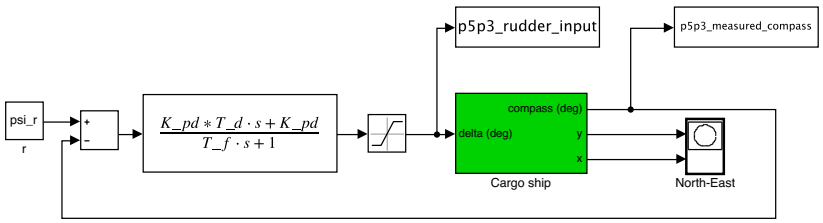


Figure 25: Simulink diagram used in Part 3 b), c) and d)

B.3 Part 5.5

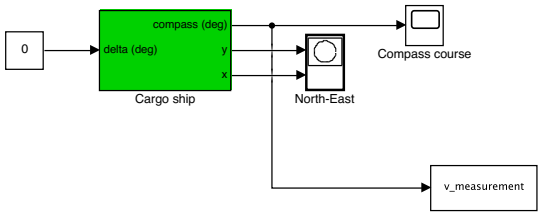


Figure 26: Simulink diagram used in Part 5 b)

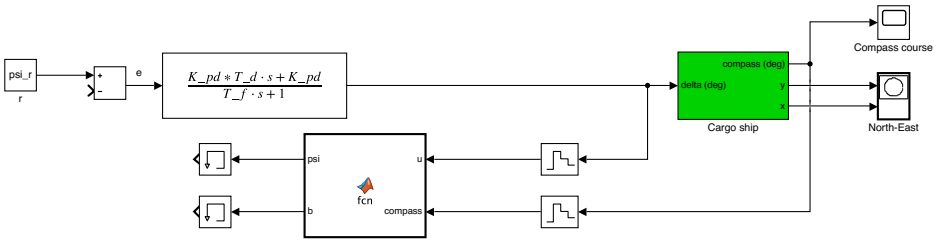


Figure 27: Simulink diagram used in Part 5 c)

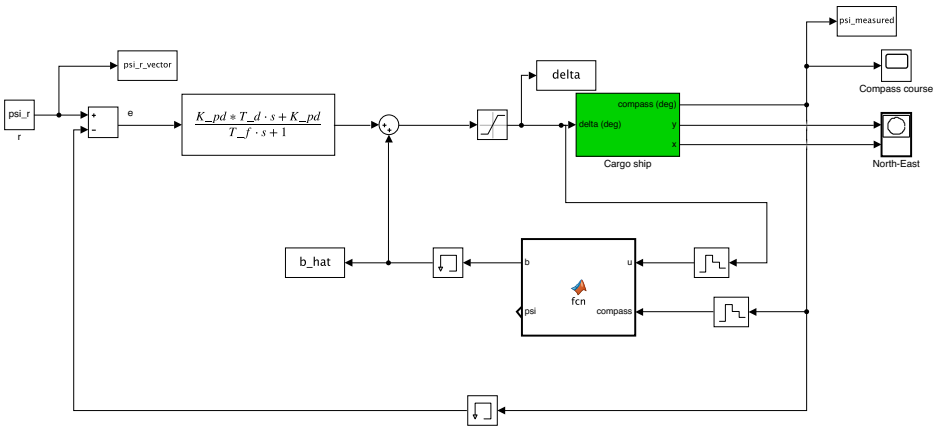


Figure 28: Simulink diagram used in Part 5 d)

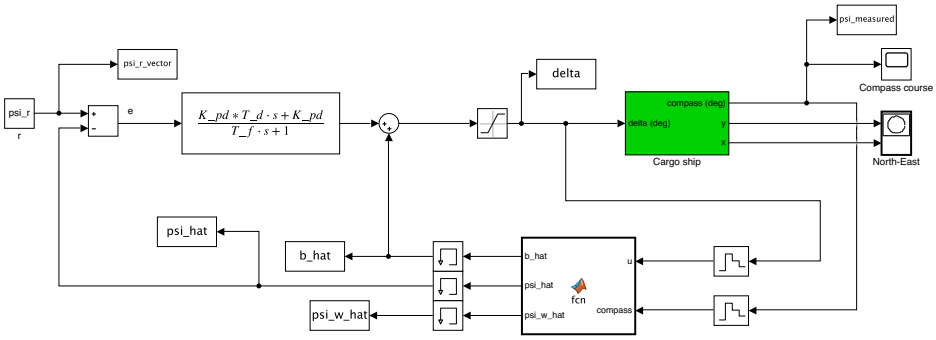


Figure 29: Simulink diagram used in Part 5 e)

References

[1] Hwang Brown. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, 2012.

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