Boat Lab Report

Group 36
Cale, Rendell (768663)
Ke, William (768698)

November 16, 2017

Department of Engineering Cybernetics

Abstract

In this lab we implemented a PD Controller to control the heading on a ship. We also observed how current and wave disturbances can influence the behaviour on a ship. A Kalman filter was implemented to compensate for some of the problems that occurs with current and wave disturbances

Contents

0 System

The mathematical model for the system as described in the assignment document is written out below for future referencing.

$$\dot{\xi_w} = \psi_w \tag{1a}$$

$$\dot{\psi_w} = -\omega_0^2 \xi_w - 2\lambda\omega_0\psi_w + K_w w_w \tag{1b}$$

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

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

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

$$\dot{y} = \psi + \psi_w + v \tag{1f}$$

1 Identification of boat parameters

1.a Transfer function from δ to ψ, $H(s)$

The transfer function from δ to ψ is calculated. Using the equations for the ship assuming no disturbances, meaning rudder bias $b = 0$, we have the following:

$$\dot{\psi} = r \tag{2}$$
$$\ddot{\psi} = \dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - b), \qquad b = 0 \tag{3}$$
$$\ddot{\psi} = -\frac{1}{T}\dot{\psi} + \frac{K}{T}\delta \tag{4}$$

taking the Laplace transform gives us:

$$\psi(s)s^2 = -\frac{1}{T}\psi(s)s + \frac{K}{T}\delta(s) \tag{5}$$
$$\psi(s)(s^2 + \frac{1}{T}s) = \frac{K}{T}\delta(s) \tag{6}$$
$$H(s) = \frac{\psi(s)}{\delta(s)} = \frac{K}{s(Ts+1)} \tag{7}$$

equation (7) is the transfer function, $H(s)$, parameterized by T and K. The transfer function tells us how the system average heading ψ will respond to changes in the rudder angle δ.

1.b Identify boat parameters T and K

The disturbances for the model was turned off in order to identify the boat parameters T and K in calm weather conditions. We ran the simulation two times with different inputs. Input 1 was a sine signal with amplitude 1 and frequency $\omega_1 = 0.005$ (rad/s). Input 2 was a sine signal with amplitude 1 and frequency $\omega_1 = 0.05$ (rad/s). The amplitude of the sine waves on the output equals $|H(j\omega_1|$ and $|H(j\omega_2|$ respectively. We then solve for T and K in the resulting two equations.

The amplitude was found using the signal statistics functionality found in the Simulink scope block. This gives us the following values:

$$A_1 = \frac{63.358 - 4.638}{2} = 29.360 \quad \text{for} \quad \omega_1 = 0.005 \tag{8}$$
$$A_2 = \frac{4.232 - 2.569}{2} = 0.8315 \quad \text{for} \quad \omega_2 = 0.05 \tag{9}$$

$$A_1 = |H(j\omega_1)| = \left|\frac{K}{j\omega_1(Tj\omega_1 + 1)}\right| \tag{10}$$
$$A_2 = |H(j\omega_2)| = \left|\frac{K}{j\omega_2(Tj\omega_2 + 1)}\right| \tag{11}$$

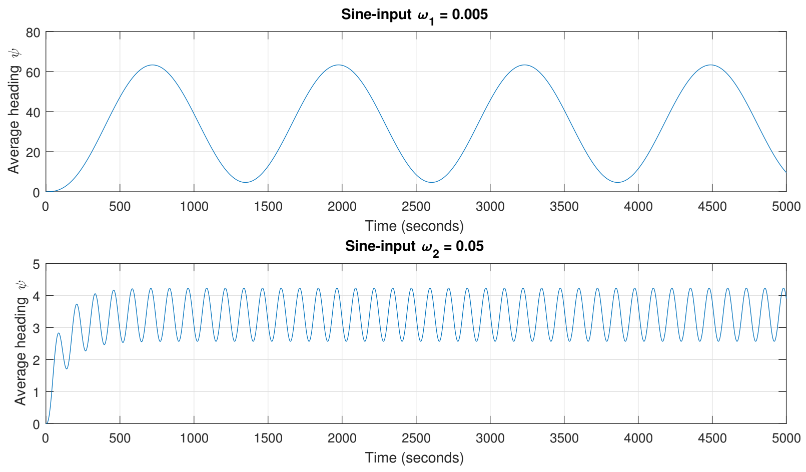

Figure 1: Simulation for finding T and K, no noise.

Solving for T and K gives us:

$$T = \sqrt{\frac{A_2^2\omega_2^2 - A_1^2\omega_1^2}{A_1^2\omega_1^4 - A_2^2\omega_2^4}} \tag{12}$$

$$K = A_1\omega_1\sqrt{T^2\omega_1^2 + 1} \tag{13}$$

Inserting values for A_1 and A_2 results in the following values for T and K:

$$T = 0.1561 \tag{14}$$

$$K = 72.3910 \tag{15}$$

1.c Identify T and K in rough weather.

Now we try to identify the boat parameters T and K with waves and measurement noise turned on, which simulates rough weather. Using the same equations as in section 1.b, we can calculate the new values for T and K.

As we can see in fig. 2, it is difficult to find good values for T and K due to the random noise on the resulting signal. Therefore we can not trust the equations for finding the boat parameters T and K to be accurate in rough weather conditions, compared to in calm weather.

1.d Step response

We want to compare the step response of the transfer function versus the step response of the ship. We applied a step input of 1 degree on the rudder at $t = 0$. fig. 3 shows us that the response between the transfer function and ship diverges, i.e. the difference grows over time. The model is a good

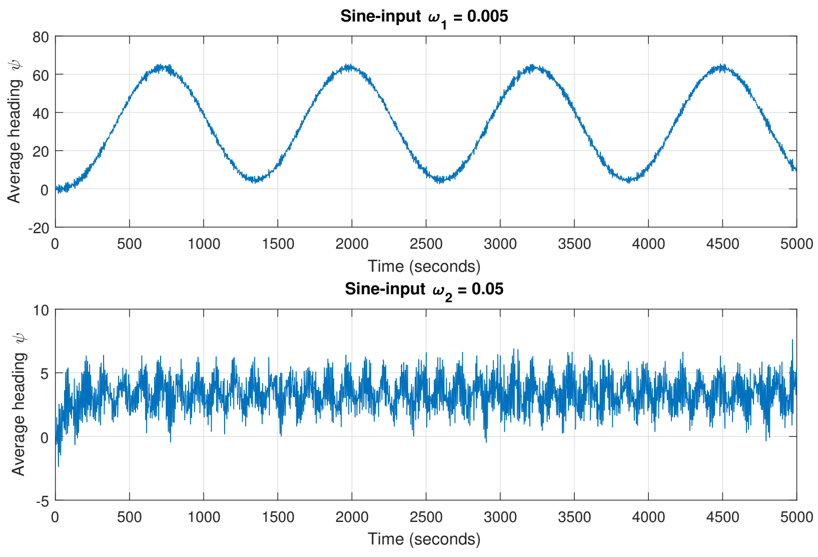

Figure 2: Simulation for finding T and K, noise.

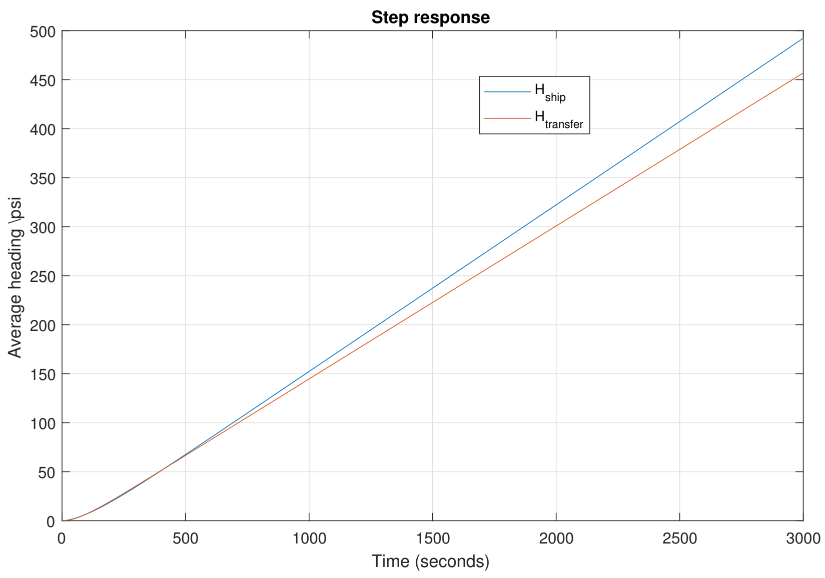

Figure 3: Comparison of step response between Ship model and transfer function.

approximation for shorts periods of time, but the accuracy gets worse over time. Better values for T and K, that diverge less, might be found by trial-and-error. But we let the boat parameter be defined as found in section 1.b for the rest of the problems.

2 Identification of the wave spectrum model

We wish to analyze how the waves affect the ship heading as this will provide useful data later when it's time to implement an estimator. First we will use wave measurements to estimate the power spectral density, then we will use this estimate to compute the resonance frequency ω_0 and the dampening factor λ.

2.a Power Spectral Density estimate

In order to estimate the power spectral density (PSD) of the wave disturbance, we use Welch's method in MATLAB. This is done simply by loading the wave data, running the command "pwelch" with a window size of 4096, and then converting the results the units to radians instead of Hz.

Listing 1: Welch's method used to estimate PSD in MATLAB

```
load('wave.mat');
t = psi_w(1,:);
[pxx,f] = pwelch(psi_w(2,:), 4096, [],[]  ,10);
pxx = pxx*(1/(2*pi));
w = 2*pi*f;
```

A plot of this is provided later in fig. 4, but before we get there we will find an analytical model of the system with which we can compare the obtained results.

2.b Analytical wave response and PSD

The waves will cause affect the ship heading as a relatively high frequency disturbance. We compute the transfer function from the wave disturbance to the disturbance componenet of the ship heading using eq. (1b) by Laplace transforming and rearranging. Note that we've used that ξ_w is the integral of ψ_w or equivalently in the s-domain $\xi_w = \psi_w/s$.

$$\begin{aligned}
\dot{\psi_w} &= -\omega_0^2\xi_w - 2\lambda\omega_0\psi_w + K_w w_w \\
s\psi_w &= -\omega_0^2\frac{\psi_w}{s} - 2\lambda\omega_0\psi_w + K_w w_w \\
\psi_w\left(s + \frac{\omega_0^2}{s} + 2\lambda\omega_0\right) &= K_w w_w \\
\frac{\psi_w}{w_w} &= \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \qquad (16)
\end{aligned}$$

From the transfer function calculated above, we can find an expression for the PSD ${}_{\psi_w}(\omega)$ of ψ_w. In the model, w_w is assumed to be a white noise process with unity variance. In general a stationary white noise process W with variance σ^2 will have autocorrelation given by

$$R_W(t) = \sigma^2 \delta(t). \tag{17}$$

By the Wiener-Khinchine relation, this same process will have a PSD $S_W(\omega)$ given by the Fourier transform of the autocorrelation [1]. It can easily be verified that this gives

$$S_W(\omega) = \sigma^2. \tag{18}$$

This means that the PSD of w_w is simply $S_{w_w}(\omega) = 1$. Writing the PSD of ψ_w using the PSD of w_w, this gives

$$\begin{aligned}
S_{\psi_w}(w) &= S_{w_w}(\omega) \left| \frac{\psi_w}{w_w} \right|^2 \\
&= \left| \frac{\psi_w}{w_w} \right|^2 \\
&= \frac{(K_w \omega)^2}{|-\omega^2 + j2\lambda\omega_0\omega + \omega_0|^2} \\
S_{\psi_w}(w) &= \frac{(K_w \omega)^2}{(\omega_0^2 - \omega^2)^2 + (2\lambda\omega_0\omega)^2}.
\end{aligned} \tag{19}$$

2.c Obtaining the resonance frequency

ω_0 is the resonance frequency of the system, so it will be estimated by the peak frequency of the PSD. Using MATLAB on the PSD computed above, we then get

Listing 2: Estimating ω_0

```
    [pmax, imax] = max(pxx);
    w0_est = w(imax);
```

Which after running the code gives an answer $\hat{\omega}_0 = 0.78233$, where the hat denotes an estimate.

2.d Obtaining the dampening factor

The only parameter missing in our current description is the constant K_w. We define $K_w = 2\lambda\omega_0\sigma$, where σ^2 is the peak value of the PSD. Since the peak value already was obtained in listing 2, all that remains is to estimate λ and we will have a full description of our system. Lambda is estimated by least-square fitting the analytical PSD computed in eq. (19) to the empirical PSD computed in listing 5. This is done in listing 3. Note that in the script, H a function of lambda and w, and it is the PSD in eq. (19) with $K_w = 2\lambda\omega_0\sigma$ and $w_0 = \hat{w}_0$.

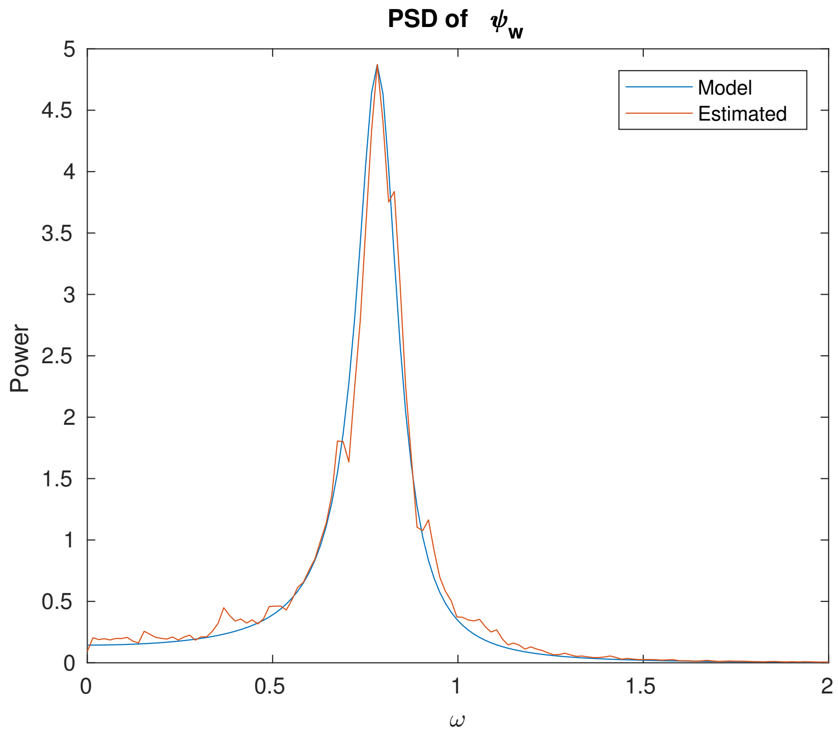

Figure 4: Power spectral density estimated using Welch's method and plotted against the model

Listing 3: Estimating λ

```
    sigma = sqrt(pmax);
    lambda_est = lsqcurvefit(H, 1, w', pxx');
```

Running this gives $\hat{\sigma} = 2.2073$ and $\hat{\lambda} = 0.03093$. To verify these results, we plot the analytical PSD with these estimated values togheter with the PSD computed using Welch's method. Doing this gives the plot in fig. 4.

The plot in fig. 4 shows only the relevant section of the frequency spectrum, and we can see that there is a high correspondance between the model and estimated spectrum. This indicates that the values found for ω_0, σ, and λ are reasonable, and that the model is a good approximation.

3 Control system design

3.a Design PD controller

we designed a PD controller, $H_{pd}(S) = K_{pd}\frac{1+T_d s}{1+T_f s}$, based on the transfer function from δ to ψ without disturbances. The open loop system, $H_{pd}(s)H_{ship}(s)$ becomes the following:

$$H_{ol} = H_{pd}(s)H_{ship}(s) = \frac{K_{pd}K(T_d s + 1)}{(T_f s + 1)s(Ts + 1)} \tag{20}$$

when we set the derivative time constant, t_d, such that it cancel the transfer function time constant we end up with:

$$H_{ol} = \frac{K_{pd}K}{s(T_f s + 1)} \tag{21}$$

we want to have a phase margin $\varphi = 50$ and cross frequency $\omega_c = 0.1$. We can solve for T_f by looking at the angle of the transfer function:

$$\varphi = \angle H_{ol}(j\omega_c) - (-180^\circ) \tag{22}$$

$$50^\circ - 180^\circ = \angle H_{ol}(j\omega_c) \tag{23}$$

$$\angle \frac{K_{pd}K}{s(T_f s + 1)} = -130^\circ \tag{24}$$

$$0 - \angle - w_c^2 Tf + j\omega_c = -130^\circ \tag{25}$$

$$tan^{-1}(\frac{\omega_c}{-\omega_c^2 T_f}) = 130^\circ \tag{26}$$

$$T_f = -\frac{1}{\omega_c tan(130^\circ)} \tag{27}$$

$$T_f = 8.39s \tag{28}$$

fig. 5 shows us the bode plot of the open loop system with $\omega_c = 0.1$ and $\varphi = 50^\circ$ as desired.

The T_f constant acts as the low-pass component of the transfer function. A higher T_f would would allow more frequencies to pass, which might induce oscillations or let the response overshoot. While lowering T_f would make the response slower. T_d is the derivative time constant. lowering T_d would reduce the damping effect of the PD-controller, which would make the response faster. while increasing T_d would increase the damping of the response. Tweaking T_f and and T_d could give a better and more optimal response for the system. But T_f and T_d will remain as defined in the assignment to have the desired phase margin φ and cross frequency w_c.

The rudder on our model has a boundary of $\pm 35^\circ$. The relation between heading change and rudder input is probably not linear in real life. For example a rudder that is at 90°, or some other high angle, would probably

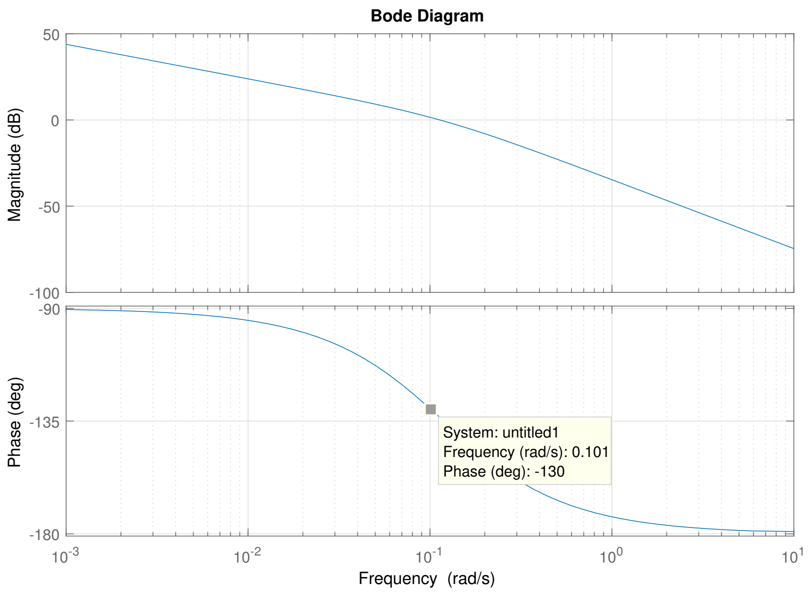

Figure 5: Bode plot of the transfer function with $\omega_c = 0.1$ and $\varphi = 50°$

just slow the ship down, since it would act as flat obstacle in the water. A rudder needs to push the water in some specific direction, which would the push the ship in the opposite direction because of Newton's third law.

3.b Simulate with no disturbances

A saturation block was added to the implementation of the Simulink model to limit the rudder input inside the boundary of $\pm 35°$. the rudder angle went way over 90 degrees if we did not have the saturation block there, that makes no sense since a physical rudder does not work over angles of $\pm 90°$. No blocks to convert degrees to radians and vice versa were needed because the PD controller is linear. As long as the sum block have equal units on the inputs it should be fine. fig. 6 shows the the rudder input and the compass reading. The ship reached the desired reference angle after about 300 seconds, which seems reasonable for large ships. The autopilot works for smooth weather conditions.

We can see that the derivative effect on the PD controller works, because the the rudder angle changes before the reference angle has been reached. The large inertia of a big ship necessitates this to prevent overshoot.

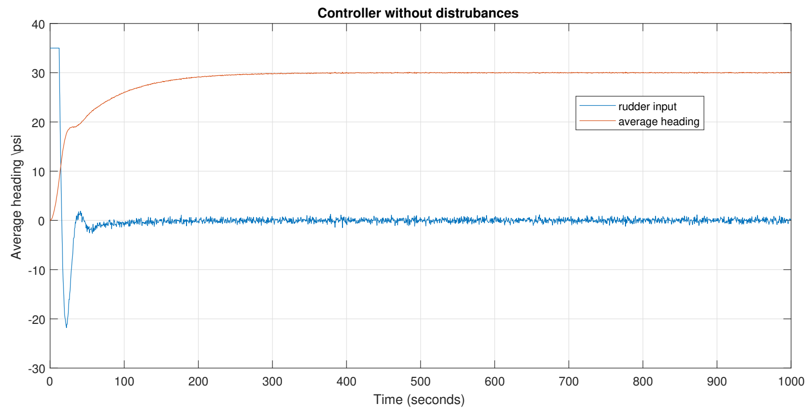

Figure 6: Controller without disturbances

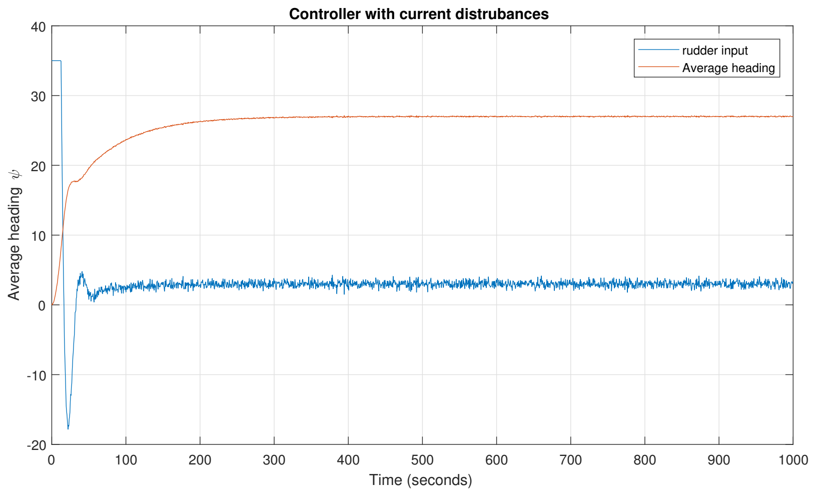

Figure 7: Controller with current disturbances

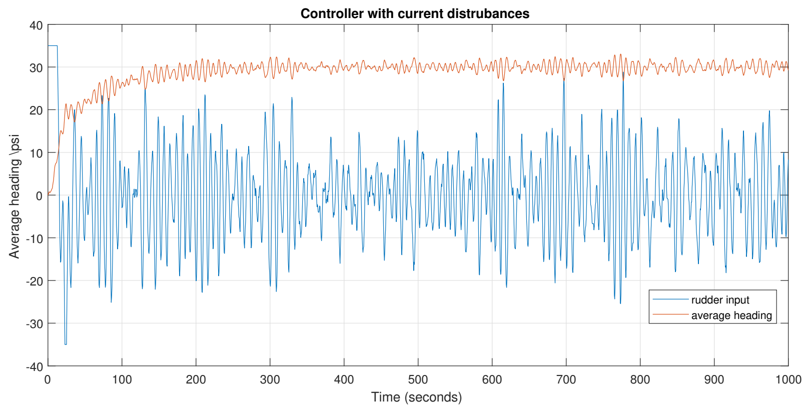

Figure 8: Controller with wave disturbances but no current

3.c Simulate with current disturbances

As we can see from fig. 7, the addition of a current disturbance results in a steady state error on the compass reading. The time to reach steady state seems to be equal as with no disturbances. It's assumed the current only have an effect on the rudder angle bias. The rudder angle input we see on the figure is not the same as the actual rudder angle on the ship. The bias results in that the actual rudder angle on the ship becomes zero before we reach the desired compass heading. The steady state error is about $3°$ off the desired $30°$ reference course angle. The steady state error could have been compensated for with the addition of an integral effect on the controller. Another solution would be to use Kalman filter to estimate the bias, then use feedforward on the rudder input. In real life conditions the current disturbance would most probably affect the position of the ship hull also.

3.d Simulate with wave disturbances

Figure 8 shows that the inclusion of wave disturbance results in oscillations on both the rudder angle input and the compass reading. The rudder tries to compensate for the effect the waves has on the heading of the ship. The waves are modelled as zero mean white noise, this means that there is no steady state error since the waves has a mean of zero. The rapid changing of the rudder could lead to unnecessary wear. The rudder angle changes by about $45°$ in about 4 seconds in some cases. A rudder on big ship in real life have probably a rate limit in real life. The controller would not be very good in a real life situation. These problems can be solved by using a Kalman filter to estimate the compass heading without the high frequency components.

4 Observability analysis

The main goal of this lab, is to apply the discrete Kalman filter as a state observer for a ship system. But before we do so, we should investigate whether or not the system is observable. If the system is unobservable, then the application of the Kalman filter will be futile.

4.a State space description

We start by finding the state space model for the system given by

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu + E\mathbf{w} \tag{29}$$

$$\mathbf{y} = C\mathbf{x} + v \tag{30}$$

where $\mathbf{x} = [\xi_w, \psi_w, \psi, r, b]^T$, $u = \delta$, and $\mathbf{w} = [w_w, w_b]^T$.

By taking the derivative of the system in eq. (1) with respect to all the different states, we get the A matrix to be

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1/T & -K/T \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{31}$$

By taking the derivative with respect to the input δ, we get the vector B to be

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ K/T \\ 0 \end{bmatrix}. \tag{32}$$

The same is done with w_w and w_b to get the E matrix

$$E = \begin{bmatrix} 0 & 0 \\ K_w & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{33}$$

Since $y = \psi_w + \psi + v$, we have that C is given by

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{34}$$

We will now gradually investigate the observability of this system first without any disturbances, then with wave and current disturbances induvidually, and then with both disturbances at the same time.

4.b Observability without disturbances

Without disturbences, the system is reduced considerably since now.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1/T \end{bmatrix} \tag{35}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{x} = \begin{bmatrix} \psi \\ r \end{bmatrix} \tag{36}$$

The observability matrix is then

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{37}$$

Note that the rank of $\mathcal{O}$ is 2, which is the same as the number of states, so the system is observable when all disturbances are ignored.

4.c Observability with current disturbance

When current disturbance is introduced to the model, a third term b will appear in the state, since the current will cause give a bias to the rudder angler. Taking the relevant parts of the A and C (from section 4.a) matrices we now have

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/T & -K/T \\ 0 & 0 & 0 \end{bmatrix}, \tag{38}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{x} = \begin{bmatrix} \psi \\ r \\ b \end{bmatrix} \tag{39}$$

With this setup, the observability matrix is

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \tag{40}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/T & -K/T \end{bmatrix}. \tag{41}$$

We see that $\mathcal{O}$ has rank 3, which is the same as the number of states in consideration. The system is thus observable if there is current disturbance.

4.d Observability with wave disturbance

When considering only the wave disturbance, we see that the system will be given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1/T \end{bmatrix}, \tag{42}$$

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{x} = \begin{bmatrix} \xi \\ \psi_w \\ \psi \\ r \end{bmatrix}. \tag{43}$$

The observability matrix is then

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} \tag{44}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 1 \\ 2\lambda\omega_0^3 & (4\lambda^2-1)\omega_0^2 & 0 & -1/T \\ -(4\lambda^2-1)\omega_0^4 & (4-8\lambda^2)\lambda\omega_0^3 & 0 & 1/T^2 \end{bmatrix}. \tag{45}$$

To check if this has full rank, we take the determinant of $\mathcal{O}$ and check when the determinant is zero.

$$det(\mathcal{O}) = \frac{\omega_0^4(T^2\omega_0^2 - 2T\lambda\omega_0 + 1)}{T^2} = 0 \tag{46}$$

$$\rightarrow \omega_0 = \frac{\lambda \pm \sqrt{\lambda^2 - 1}}{T}. \tag{47}$$

Note that eq. (46) is non-zero for all real $\omega_0 \neq 0$ since the dampening factor $\lambda < 1$, so the observability matrix $\mathcal{O}$ has a non-zero determinant, which is sufficient to show that it has full rank. Thus the system is observable with only wave disturbance.

4.e Observability with both disturbances

With both disturbences modelled, our system A and C matrices are the same as in eq. (31) and eq. (34), respectively. This gives an observability matrix

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

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 1 & 0 \\ 2\lambda\omega_0^3 & \omega_0^2(4\lambda-1) & 0 & -1/T & -K/T \\ -(4\lambda^2-1)\omega_0^4 & 4\omega_0^3\lambda(1-2\lambda^2) & 0 & 1/T^2 & K/T^2 \\ -4\lambda\omega_0^5(\lambda^2-1) & -4\lambda^2\omega_0^4+(4\lambda^2\omega_0^2-\omega_0^2)^2 & 0 & -1/T^3 & -K/T^3 \end{bmatrix}. \tag{48}$$

Computing the determinant, we get

$$det(\mathcal{O}) = \frac{-\omega_0^6 K(\omega_0^2 T^2 - 2T\lambda\omega_0 + 1)}{T^3} \tag{49}$$

which is non-zero for all real $\omega_0 \neq 0$ for the same reason as in section 4.d. Since the determinant of the observability matrix is non-zero, it has full rank and it follows that the system is observable.

Remarks We have now seen that the ship system is observable with both disturbance sources. This is good news as it allows us to apply a Kalman filter as a state estimator and get good results even when measurement noise is added.

5 Discrete Kalman filter

In the final part of this report, we will implement a discrete Kalman state observer to filter out the high frequency noise components of the heading. There are several reasons one might wish to do this, and one of them is that if our regulator were to use the high frequency signal, then the actuator signal would also have some high frequency component to it. Over time this causes the motor and other mechanical parts to wear out significantly faster than they would otherwise, so from a robustness and economical perspective it makes sense to filter out the high frequency components and using a more stable estimated heading as input to the control system. But before we can implement a discrete Kalman filter, we have to discretize our system and characterise the inherent measurement noise.

5.a Discretization

The CT LTI version of the system was computed in section 4.a. We will now discretize the system in MATLAB using a combination of Van Loan's method and the built in function c2d. First we will discretize A and B, and also compute the discrete disturbance covariance numerically. The stochastic state space model that is under study is

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu + E\mathbf{w} \tag{50}$$

$$y = C\mathbf{x} + v \tag{51}$$

$$E\left\{\mathbf{w}\mathbf{w}^T\right\} = Q_w = \begin{bmatrix} 30 & 0 \\ 0 & 10^{-6} \end{bmatrix} \tag{52}$$

Van Loan's method allows us to compute the matrices in the discrete state space given by

$$\mathbf{x}[n+1] = A_d\mathbf{x}[n] + B_d u[n] + E_d\mathbf{w}[n] \tag{53}$$

$$y[n] = C_d\mathbf{x}[n] + v[n]. \tag{54}$$

We compute A_d, B_d and the discrete disturbance variance Q_d by computing two matrix exponetials. We let T_s denote the sampling rate, and solve the following expression for M_{11}, M_{12}, M_{22}, and N_{12}.

$$\exp\left(\begin{bmatrix} A & EQ_wE^T \\ \mathbf{0} & -A^T \end{bmatrix} T_s\right) = \begin{bmatrix} M_{11} & M_{12} \\ \mathbf{0} & M_{22} \end{bmatrix} \tag{55}$$

$$\exp\left(\begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{0} \end{bmatrix} T_s\right) = \begin{bmatrix} N_{11} & N_{12} \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \tag{56}$$

After doing so, we can compute parts of the discrete state space as follows:

$$A_d = M_{11} \tag{57}$$

$$B_d = N_{12} \tag{58}$$

$$Q_d = M_{12}M_{11}^T. \tag{59}$$

To compute E_d, we could repeat the N-matrix part but with A and E instead of A and B, but simplify we will just use the MATLAB function c2d.

Listing 4: Exact discretization with c2d(..)

```matlab
[A_d, E_d] = c2d(A,E,Ts);
```

Since $C_d = C$, we have now calculated all matrices in the discretized system.

After writing these methods in a MATLAB script and using the paramaters we've computed so far, we got the following results

$$A_d = \begin{bmatrix} 0.9969 & 0.0997 & 0 & 0 & 0 \\ -0.0610 & 0.9921 & 0 & 0 & 0 \\ 0 & 0 & 1.000 & 0.0999 & 0 \\ 0 & 0 & 0 & 0.0986 & -0.002 \\ 0 & 0 & 0 & 0 & 1.000 \end{bmatrix} \tag{60}$$

$$B_d = \begin{bmatrix} 0 \\ 0 \\ 0.0099 \cdot 10^{-3} \\ 0.1987 \cdot 10^{-3} \\ 0 \end{bmatrix} \tag{61}$$

$$E_d = \begin{bmatrix} 0.0005 & 0 \\ 0.0106 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{62}$$

$$C_d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{63}$$

$$Q_d = \begin{bmatrix} 0.0001 & 0.0017 & 0 & 0 & 0 \\ 0.00170 & 0.0340 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{64}$$

5.b Estimation of measurement noise

To estimate the measurement noise, the system was simulated without any disturbance or input for about 20000 seconds. This gave a timeseries where the only factor that affected the signal was the measurement noise. A plot of this signal is given in fig. 9.

The variance of this signal was computed with the function var(...) and to discretize this variance, we took the average over a sample.

Listing 5: Estimating variance of measurement noise

```matlab
d = load('../data/p5_var_measurement.mat'); d = d.ans;
R = var(d.Data(:));
Rd = R/Ts;
```

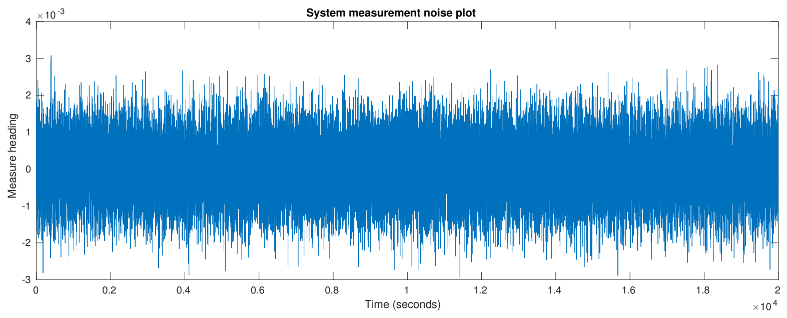

Figure 9: Recoring of measurement noise

Based on this we estimate that the variance of the noise is $R = 6.0929 \cdot 10^{-7}$ and that the discrete noise variance is $R_d = 6.0929 \cdot 10^{-6}$.

5.c Discrete Kalman setup

We want to use a Kalman filter to estimate the average heading ψ. Since the majority of the system is simluated in Simulink as a dynamic system, we decided to go the route of the s-function. The s-function was setup so that the *sys* variable contained both a priori and a posteriori state estimates. We also added the covariance matrix of the a priori estimation error P_k^- since this is need in the "first" step of the Kalman filter loop.

After this initial setup, we implemented the Kalman filter loop prescribed in figure 4.1 of [1]. The a priore estimate and covariance we started with were

$$P_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2.5 \cdot 10^{-3} \end{bmatrix} \qquad \hat{\mathbf{x}}_0^- = \mathbf{0}. \tag{65}$$

The Kalman filter loop has four steps wich are implemented in the update part of the s-function. These part can described as follows: (1) Compute the optimal Kalman gain, (2) use measurements to update estimate, (3) compute the error covariance for the new estimate, (4) predict the next state.

(1): Computing the Kalman gain is done by evaluating the expression

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1}. \tag{66}$$

(2): Given some measurement y_k, we can update our estimate with the expression

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (y_k - C \hat{\mathbf{x}}_k^-) \tag{67}$$

This a posteriori estimate will be used to determine the output of the Kalman filter.

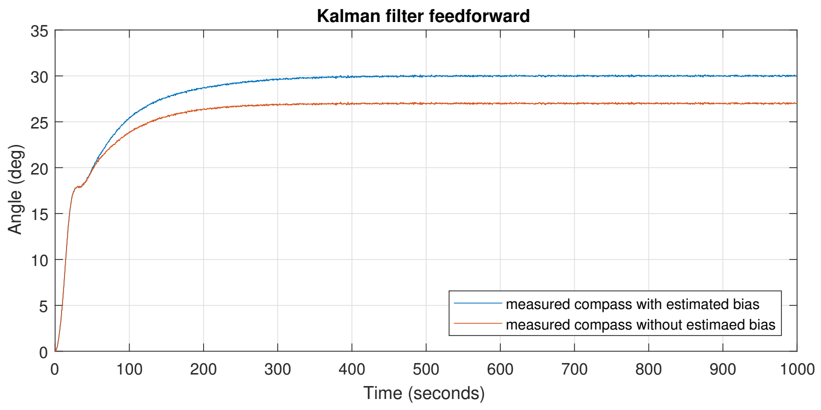

Figure 10: Comparison of measured compass course with and without rudder bias feedforward

(3): The error covariance for this updated estimate is given by

$$P_k = (\mathbb{I} - K_k C)P_k^-. \tag{68}$$

(4): Producing an a priori estimate for the next state is done with the following equation:

$$\hat{\mathbf{x}}_{k+1}^- = A\hat{\mathbf{x}}_k + Bu, \tag{69}$$

$$P_{k+1}^- = AP_kA^T + Q_d. \tag{70}$$

These four steps will estimate among other things the average heading ψ and rudder bias b. The a posteriori state estimate with an appropriate C matrix, is used to output the desired states from the Kalman filter.

Implementing this in Simulink, there is only one slight detail that has be added, and that is that since the process and and controller are continuous, but the filter isn't, we have to put a zero order hold on the input to the filter. This completes the Kalman implementation as s-function and sets the Simulink model up to use the Kalman filter in the loop.

5.d Feed forward

The a posteriori estimated bias is used a feedforward to the rudder input, such that the bias from the current is compensated. The reference course is 30°, and the sample frequency is 10 Hz. fig. 10 shows us the response of the ship with and without bias feedforward. The measured compass course became equal to the reference angle after about 330 seconds. The steady state error was compensated for because the estimated bias war being feed-forwarded to the rudder input.

The difference in rudder angle can be seen in fig. 11. There is a slight increase in the rudder input with feedforward compared to without. That

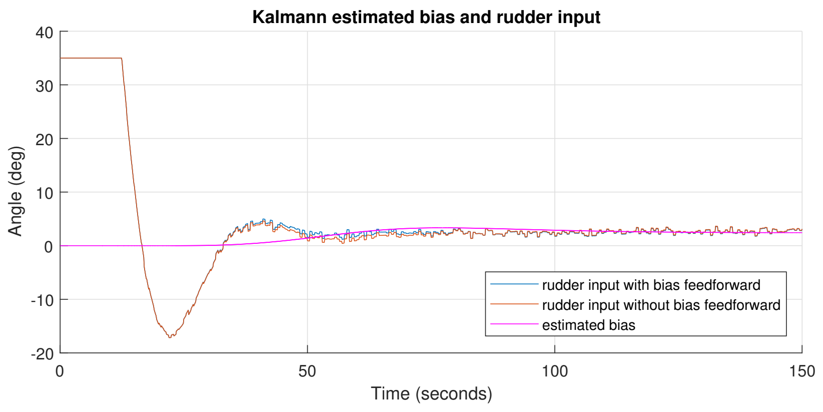

Figure 11: Comparison of rudder input with and without rudder bias feedforward

slight difference is enough to compensate for the 3° in steady state error. we see that both rudder inputs become equal after a while. In the case with feedforward. The error in the feedback loop becomes zero, and only the feedforward part contributes to the rudder input. In the case for without feedforward the feedback loop is nonzero, even after steady state, because of the steady state error. In essence the feedforward compensates for the current disturbance and the system is able to reach the desired reference angle with no steady state error.

5.e Putting the Kalman filter to use

The waves disturbances have a high frequency that induce high frequency changes on the heading of the ship. However the waves should have a mean value of zero. If we manage to filter out those high frequency noises, and use the ship heading without the high-frequency components in the feedback loop, we should still be able to reach the desired reference compass heading. This is known as wave filtering. Only the dynamics of the measured compass reading will influence the actuator, the rudder in our case.

We can see from fig. 12 that the high frequency components are eliminated by the Kalman filter. The frequency components of the wave however are still there. The estimated heading avoids excessive oscillation while following the dynamics of the measured course.

Figure 14 shows us that the rudder input oscillates way less with Kalman filtering than without. Thus we can avoid unnecessary wear and tear on the rudder. Figure 13 reveals that the measured compass heading does not change much even without the rudder trying to compensate for every wave.

In summary the Kalman filter is able to compensate for the steady state error from the current disturbance with bias feedforward. The Kalman filter is also able too reduce wear on the rudder by filtering out the high frequency

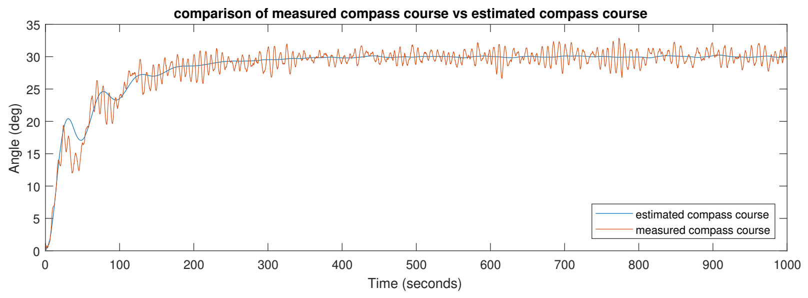

Figure 12: Comparison of measured vs estimated compass course

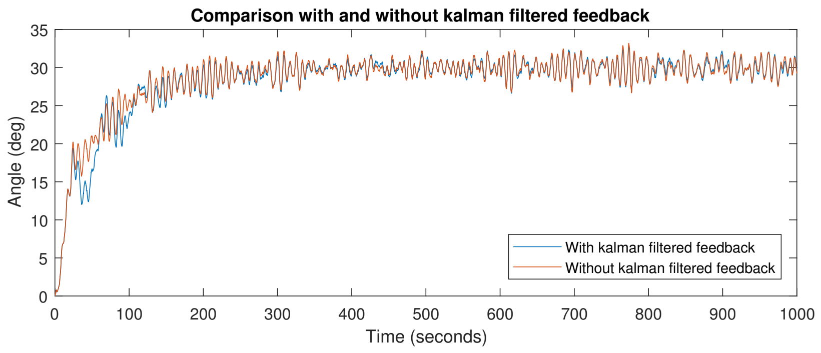

Figure 13: Comparison of measured compass heading with and without Kalman filtered feedback

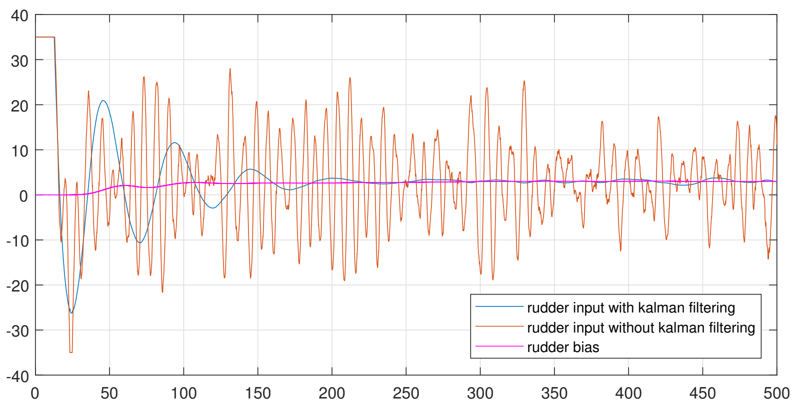

Figure 14: Comparison rudder input with and without Kalman filtering

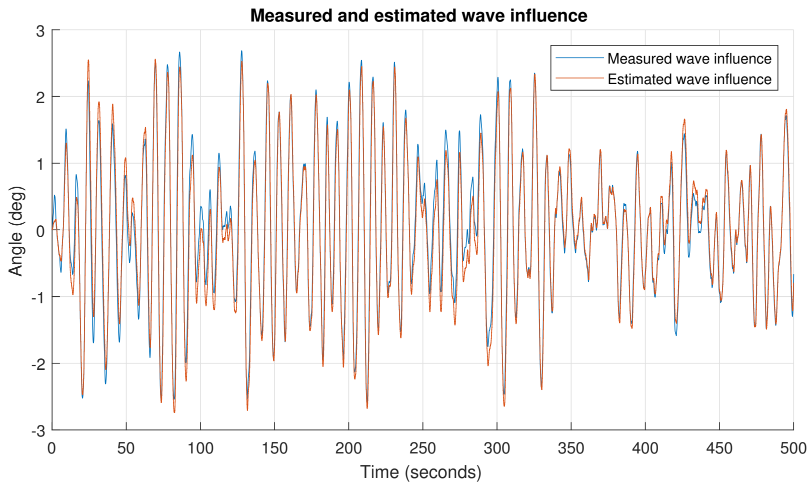

Figure 15: Comparison of measured wave influence and estimated wave influence

components of the waves, without having any significant worsening of measured compass heading. The Kalman filter solves two problems that occurs with current and wave disturbance. It is not however able to remove the effect the waves has on measured compass heading, only reduce the wear on the input rudder.

To measure the influence the waves disturbance had on the heading, we need to turn off current and measurement noise. The rudder input going to the model also needs to be zero.

Our S-function also had to be edited such that it also outputs the high frequency component ψ_w.

Figure 15 tells us that the measured and estimated wave influence largely corresponds to each other. This can be an indication that the estimation from the Kalman filter is quite good.

Conclusion

We observed how waves and current can influence the behaviour of a ship. A PD controller was able to control the heading of the ship with no disturbances, but turned out to be insufficient when dealing with waves and currents.

The current leads to a steady state error on the compass heading. Which we solved by estimating the rudder bias from the current, then feeding it forward to the rudder. The waves introduces a lot of oscillation on the measured compass heading. Which leads to excessive rudder actuation. We managed to minimise the amount of rudder actuation by using estimated compass heading,from the Kalman filter, without the high-frequency components of the waves, but we could not do anything with the waves leading to oscillations in the measured compass heading. The Kalman filter helped to deal with some of the problems that current and wave disturbances induces on the system.

A MATLAB Code

A.a common.m

```matlab
%% Parameters
K = 0.1561;          % from p1
T = 72.3910;         % from p1
w0 = 0.78233;        % from p2
lambda = 0.03093;    % from p2
sigma = 2.2073;      % from p2

Kw = 2*lambda*w0*sigma; % from p2
```

A.b Solve for T and K

```matlab
%% Solve for T and K
% see report for derivaion of this
T = sqrt(amp2^2*w2^2 - amp1^2*w1^2)/sqrt(amp1^2*w1^4 - amp2^2*w2^4);

% should be equal
K = amp1*w1*sqrt(T^2*w1^2 + 1);
K2 = amp2*w2*sqrt(T^2*w2^2 + 1); % for testing, gives same as above
```

A.c PD-controller

```matlab
run('common.m')

%% from problem 1
Hship = tf([K], [T, 1, 0]);

%% reference
psi_r = 30;


%% PD-reg
Td = T;
wc = 0.1;
phi = 50; % deg

Tf = -1/(tan(deg2rad(180-phi))*wc);

Kpd = 1;
Hpd = tf([Kpd*Td, Kpd], [Tf, 1]);
```

A.d Kalman filter

```
%% Parameters
run('common.m');
Fs = 10; % Sampling rate [Hz]
Ts = 1/Fs;

%% Regulator

% from problem 1
K = 0.1397;
T = 70.2393;
Hship = tf([K], [T, 1, 0]);

% reference
psi_r = 30; % [deg]


% PD-reg
Td = T;
wc = 0.1;
phi = 50; % deg
Tf = -1/(tan(deg2rad(180-phi))*wc);
Kpd = 1;
Hpd = tf([Kpd*Td, Kpd], [Tf, 1]);


%% Continous time LTI model
A = [0       1               0    0        0;
     -w0^2   -2*lambda*w0    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].';

E = [0       0;
     Kw      0;
     0       0;
     0       0;
     0       1];
C = [0   1   1   0   0];

Q = [30     0    ;
      0     1e-6];

```

```
%% Discretizing CT model using Van Loan's method

M = expm( [A                 E*Q*(E.');
              zeros(size(A)) -A.'      ] .*Ts);

N = expm( [A                  B;
              zeros(size(B')) zeros(size(B,2))] .*Ts);

M11 = M(1:size(A,1),1:size(A,2));
M12 = M(1:size(A,1),size(A,2) + (1:size( E*Q*(E.') ,2)));

N12 = N(1:size(B,1), size(A,2) + (1:size(B,2)));

Ad  = M11;
Qd = M12 * M11';
Bd  = N12;

%[Ad, Bd] = c2d(A,B,Ts);
[Ad, Ed] = c2d(A,E,Ts);

%% Variance of measurement noise
d = load('../data/p5_var_measurement.mat');
d = d.ans;

R = var(d.Data(:));
Rd = R/Ts;

%% Filter

% Ak = A;
% Bk = [0      0;
%        0      0;
%        1      0;
%        0      K/T;
%        0      0];
% Ck = [0  0   1   0   0;
%        0  0   0   0   1];

x0_minus = [0   0   0   0   0].';

P0_minus = [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];

%P0_minusvec = P0_minus(:);

%P0_minusmat = reshape(P0_minusvec, sqrt(length(P0_minusvec)),sqrt(leng

data = struct('Ad', Ad, 'Bd', Bd, 'C', C, 'Ed', Ed, 'Qd', Qd, 'Rd', Rd,
              'P0_minus', P0_minus, 'x0_minus', x0_minus);
```

A.e S-function

```
function [sys,x0,str,ts] = DiscKal(t,x,u,flag,data) %DiscKal(t,x,u,flag
% Shell for the discrete kalman filter assignment in
% TTK4115 Linear Systems.
%
% Author: J rgen Spj tvold
% 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
  %%%%%%%%%%%%%
  % Terminate %
  %%%%%%%%%%%%%

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

  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(d
% 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,
sizes.NumDiscStates  = 5+5+25; % Number of discrete states in the syste
sizes.NumOutputs     = 2; % Number of outputs, the hint states 2
sizes.NumInputs      = 2; % Number of inputs, the hint states 2
sizes.DirFeedthrough = 0; % 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.x0_minus; data.x0_minus; data.P0_minus(:)]; % Initial value
%Pk_minus = data.P0_minus;
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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute xk, xk_minus, zk
Pk_minus = reshape(x(11:35),sqrt(length(x(11:35))),sqrt(length(x(11:35)
xk_minus = x(1:5);
%xk = x(6:10);
zk = u(1);

```

```
% Kalman gain matrix
Kk = Pk_minus*(data.C') / (data.C*Pk_minus*(data.C') + data.Rd);

% Update estimate
xk = xk_minus + Kk * (zk - data.C*xk_minus);

% Compute error covariance
Pk = (eye(size(data.Ad)) - Kk*data.C)*Pk_minus*(((eye(size(data.Ad)) -

% Project ahead
xk_minus = data.Ad*xk + data.Bd*(u(2));
Pk_minus = data.Ad*Pk*data.Ad' + data.Qd; %data.Ed*data.Q*data.Ed';


sys=[xk_minus; xk; Pk_minus(:)];

function sys=mdlOutputs(t,x,u,data)% mdlOutputs(t,x,u,data) if mathod 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the outputs here
% example: sys=x(1)+u(2), means that the output is the first state+
% the second input.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%persistent C_;
C_ = [0 0 0 0 0 0 0 1 0 0 zeros(1, 25);
      0 0 0 0 0 0 0 0 0 1 zeros(1, 25)];
sys=C_*x;

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

B Simulink Diagrams

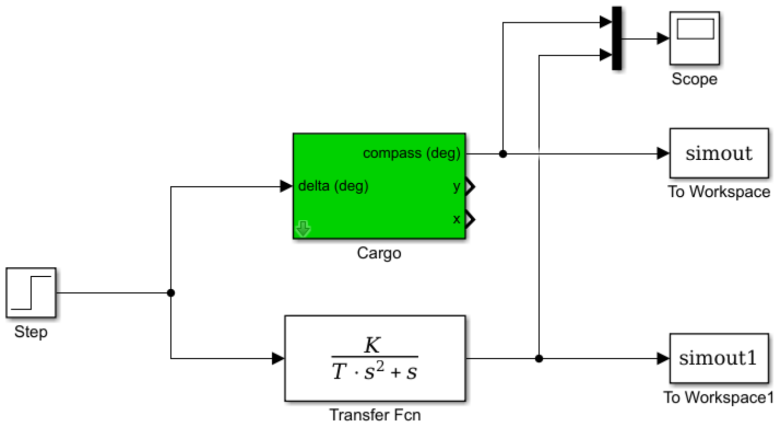

Figure 16: Transfer function and Ship model Step response

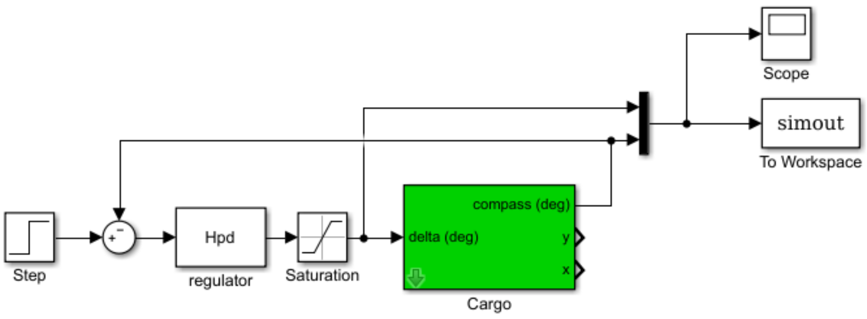

Figure 17: implementation of PD-Controller

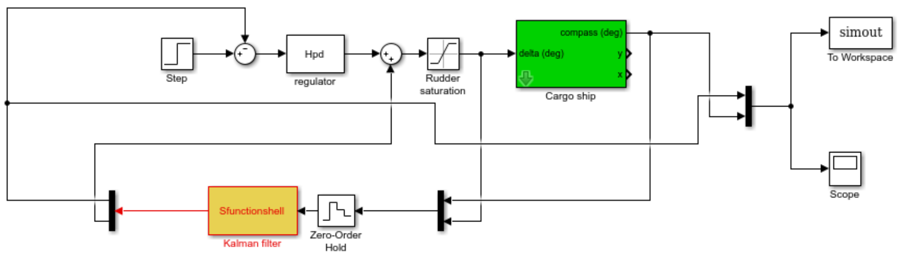

Figure 18: Implementation of Kalman filter

References

[1] Robert Grover Brown and Patrick Y.C Hwang. *Introduction to Random Signals and Applied Kalman Filtering - 4th ed.* John Wiley Sons, Inc, 2012.