Delft University of Technology

Robust Control

Control Design for a Floating Wind Turbine

Author:
David Chen *(5705762)*
Floris van Leeuwen *(4676092)*

April 9, 2023

Contents

1 SISO Analysis and Control Design **2**
- 1.1 Open loop analysis . . . 3
- 1.2 Requirements transform . . . 5
- 1.3 SISO controller design . . . 7
- 1.4 Disturbance rejection design . . . 10

2 Multi variable Mixed-Sensitivity **14**
- 2.1 Relative Gain Array . . . 14
- 2.2 MIMO poles and zeros . . . 15
- 2.3 Performance weights design . . . 16
- 2.4 Block diagram . . . 17
- 2.5 Generalized plant . . . 18
- 2.6 Interpretation performance weights . . . 18
- 2.7 mixed-sensitivity generalized controller . . . 19
- 2.8 Time-domain simulation . . . 19

3 MIMO Weighting Design **22**
- 3.1 Generalized plant . . . 23
- 3.2 Output controller weight w_p design . . . 24
- 3.3 Input controller weight w_u design . . . 27
- 3.4 Synthesized system visualizing . . . 29
- 3.5 System simulation . . . 31

1 SISO Analysis and Control Design

In this chapter, we are asked to design a SISO controller for the floating wind turbine. For every control task, the most important thing is to figure out what is the output and the input of the system. From the name SISO, we know that the system should have a single input and a single output. Further reading of the question indicates that the goal of the controller is to pitch the blades to increase the rotational velocity of the turbine, which indicates that the input of the system should be blade pitch angle β and the output of the system should be the rotational velocity ω_r.

Based on the above analysis, we are able to construct the system in a state space representation way, which is shown below:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

where A is a 5×5 matrix that is given, B is a 5×1 matrix since we only select the first column for input β, C is a 1×5 matrix since we only select the first row for output ω_r, and D equals to zero. Noted that in order to make the design easier, the C matrix is multiplied by -1 for system construction. The definition of state variables, input, and output is shown below:

$$x = \begin{bmatrix} \omega_r \\ \dot{z_1} \\ z_1 \\ \dot{z_2} \\ z_2 \end{bmatrix} u = \begin{bmatrix} \beta \end{bmatrix} y = \begin{bmatrix} \omega_r \end{bmatrix}$$

1.1 Open loop analysis

Based on the above state space model, we obtained the open loop system. Using Matlab function **ss2tf** and **zpk**, we can easily obtain the transfer function of the open loop system.

$$G(s) = \frac{0.079878(s^2 - 0.007693s + 0.04)(s^2 + 0.0492s + 10.82)}{(s + 0.4104)(s^2 + 0.02113s + 0.04101)(s^2 + 0.1664s + 10.85)} \quad (1)$$

According to equation (1), zeros and poles of the system can be calculated from which we can tell that the system is stable since all poles are located on the left-hand side. This can be shown by the pole-zero map.

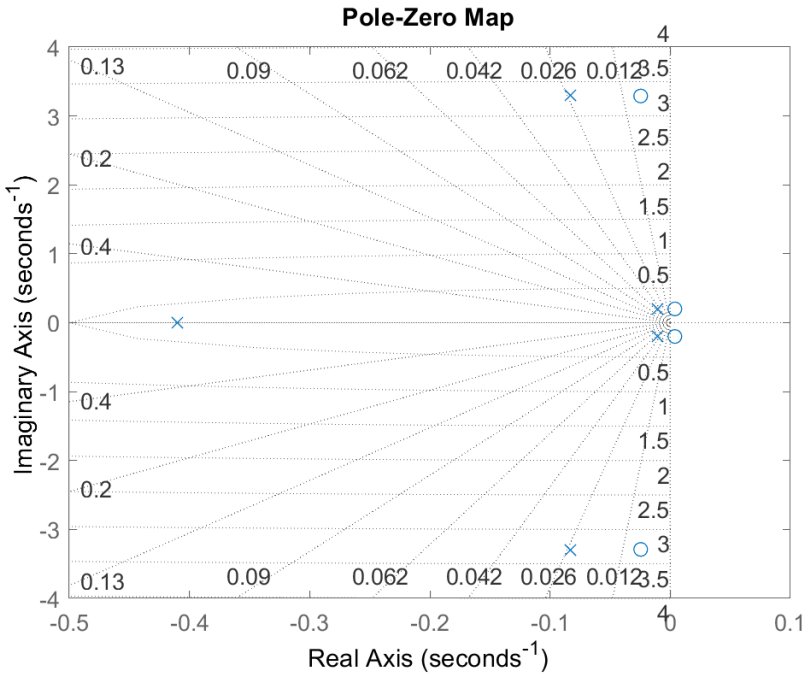


Figure 1: Pole zero map of the open loop system from which we can tell that all poles locate in the left-hand side, indicating stability

The stability of the open loop system can be proved or shown either by the step response of the corresponding close loop system which can be easily constructed or by the bode diagram.

In order to gain more understanding of the system, the bode diagram is created. The gain margin of the open loop system is $24.2dB$ at a frequency

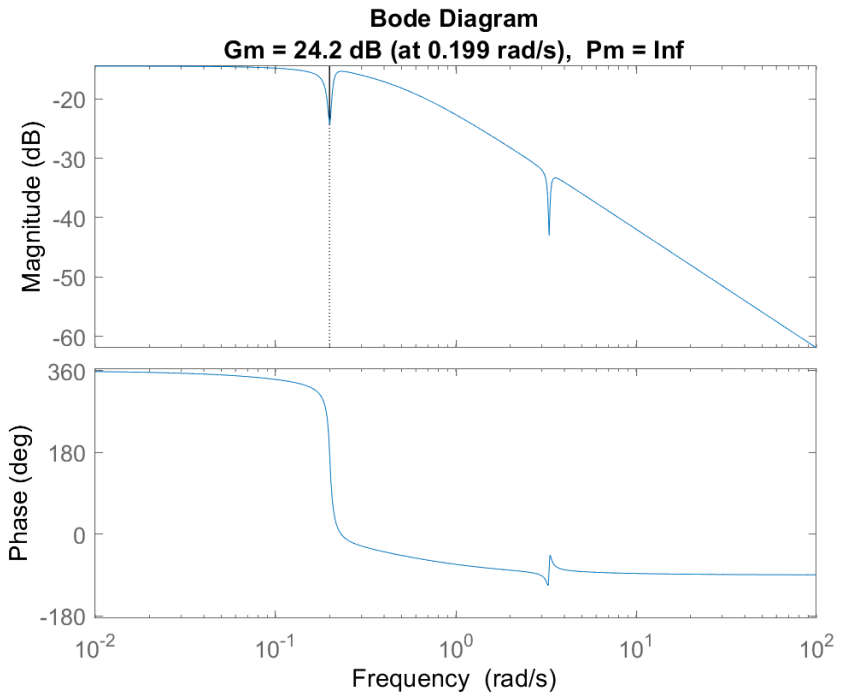


Figure 2: Bode diagram of the open loop system

of $0.199rad/s$. The phase margin of the system is ∞. Noted that since we multiplied the matrix C by -1 in the beginning, the phase plot starts at -360^o. Combined with the pole-zero map, we are able to spot the feature that hinders us from achieving high bandwidth — the right-hand side zeros. Zeros in the RHS prevent us from achieving high bandwidth by introducing -90^o lag to the phase plot which causes the phase to fall. In the following chapter, we will design a controller to compensate for the influence of the existence of left-hand side zeros.

1.2 Requirements transform

There are three design requirements we should meet in the time domain, those requirements are:

1. Small settling time
2. Overshoot < 1 %
3. No steady state error

In this section, we translated time domain requirements into frequency domain requirements based on numerical analysis or bode diagrams.

For the first requirement, a small settling time usually indicates a small rising time. Regardless of settling time or rising time, it is used to reflect the speed of the response. This leads us to consider the bandwidth of the system. To achieve **a smaller settling time**, a **large close-loop bandwidth** is adopted so that high-frequency signals are more easily passed on to the outputs. However, increasing the bandwidth also makes the system more sensitive to noise and disturbance. Thus, we need to find a balance and increase close-loop bandwidth at a certain point where the system has a fast response while maintaining stability and robustness.

By looking at the second-order system which is presented in the below equation, we are able to find the relation between overshoot and phase margin.

$$Y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} X(s) \tag{2}$$

For a second-order system like in equation (2), the damping ratio is ζ, by numerical analysis, we are able to obtain the overshoot of the system which is defined below.

$$Overhoot = 100e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{3}$$

Since we also know the approximate relation between the phase margin and the damping ratio presented in equation (4). The relation between phase margin and overshoot can be drawn.

$$PhaseMargin \approx 100\zeta \tag{4}$$

Noted that even though we are looking at a system whose order is more than 2, higher-order systems can be reduced to lower-order systems, for example,

second-order systems, when analyzing and designing controllers. Thus, we are able to conclude that **a large phase margin corresponds to a small overshoot**.

Lastly, the transform of the steady-state error to the frequency domain can be done by using the final value theorem of the type zero system which is shown below:

$$\lim_{x \to \infty} f(x) = \lim_{s \to 0} F(s) \tag{5}$$

According to the open-loop transfer function of the system, we used the Matlab built-in function to acquire the steady-state error and got the result of 15.92%. This shows that the system is far from reaching zero steady-state error. In the frequency domain, **the steady-state error corresponds to a high gain for low frequency and a low gain for high frequency for the open loop bode diagram** since it prevents high-frequency noise from entering the system.

1.3 SISO controller design

The step response of the original closed-loop system is shown below.

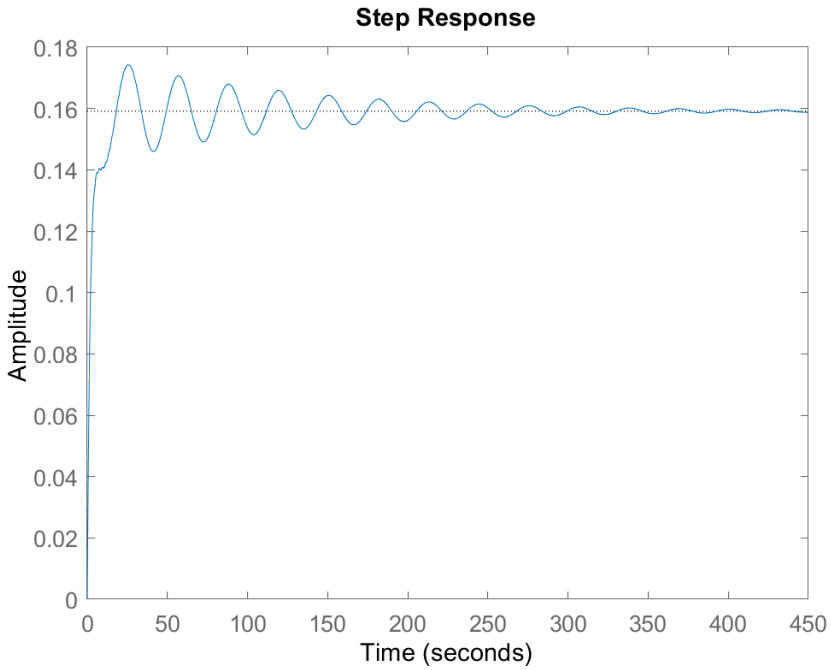


Figure 3: Step response of the original closed-loop system. Even though the system is able to reach the stable value after 300s, we noticed a series of several oscillations in the step response along with the cumbersome settling time. Thus, the system should be optimized by using either a PI controller or a PID controller.

By using the Matlab function **stepinfo**, we are able to acquire the step response information about the uncontrolled closed-loop system: the original system has an overshoot of 9.45%, a settling time of 199.3618s, and steady-state error of 15.94%. Combined with the aforementioned analysis, in order to achieve all the requirements, we have to design a controller to acquire higher bandwidth, adjust phase margin and acquire desired open-loop low-frequency gain. In order to achieve this, a PI controller is designed for our system.

The P in the PI controller contributes to the shift of phase margin, there-

fore, moving the crossover point of the system, which implies the change of system bandwidth. As a result, the settling time can be decreased by adopting a P controller. However, as the aforementioned analysis implied, increasing the gain of the system to a certain extent results in unstable oscillations. Thus, it is important to find the threshold. In addition, the I in the PI controller contributes to the zero steady-state error by creating a very high gain for the low-frequency. Thus, we used a PI controller. It is true that by adopting more complicated controllers, we might be able to achieve better performance. Nevertheless, in the real scenario, we have to take the cost of implementation of those controllers into consideration. Thus, we chose to design by using a simple PI controller.

After a long period of tuning, we created a PI controller using Matlab **PID** function. The definition of our PI controller is shown below in equation (6).

$$PID = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \tag{6}$$

with $K_p = 1$, $K_i = 0.25$, and $K_d = 0$. The bode diagram of the controller is presented.

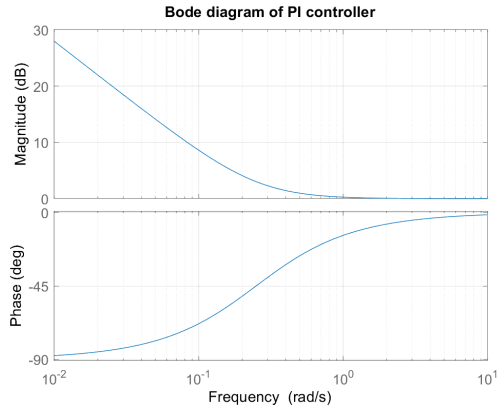


Figure 4: Bode diagram of the designed PI controller, we can tell that this controller introduces high gain and proper phase lag for low-frequency as expected.

After emerging the controller with the original system, the step response of the controlled system is shown below. Noted that in order to gain more information about the system, we created a visualization function in our code.

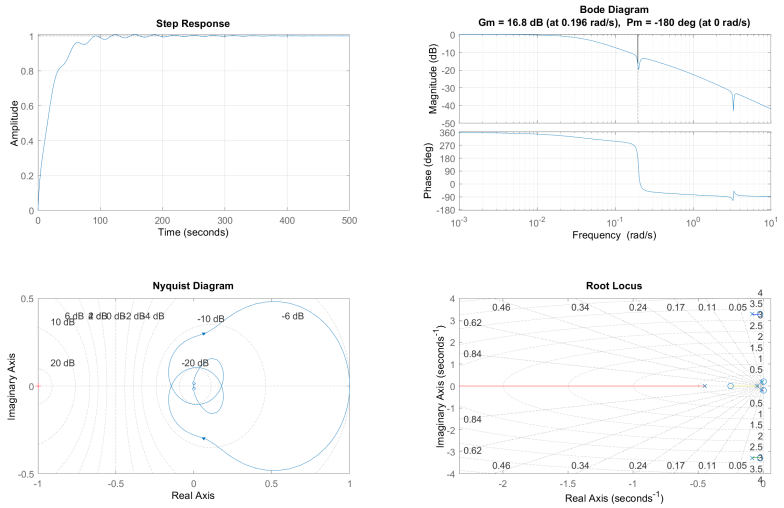


Figure 5: Visualization of the controlled system, the four plots are the step response, the bode diagram, the Nyquist plot, and the root locus. From those plots, we are able to notice that the performance of the SISO system improves dramatically while maintaining stability.

By using the **stepinfo** function, we are able to acquire some system parameters. The controlled system has a settling time of $84.6080s$, an overshoot rate of 0.7157%, a phase margin of -180^o, and a bandwidth of $0.0460rad/s$.

1.4 Disturbance rejection design

When designing a system to reject output disturbance, we take the disturbance step input on the third input channel of the state space. Consequently, the diagram of the system changed from the original one to the one presented below.

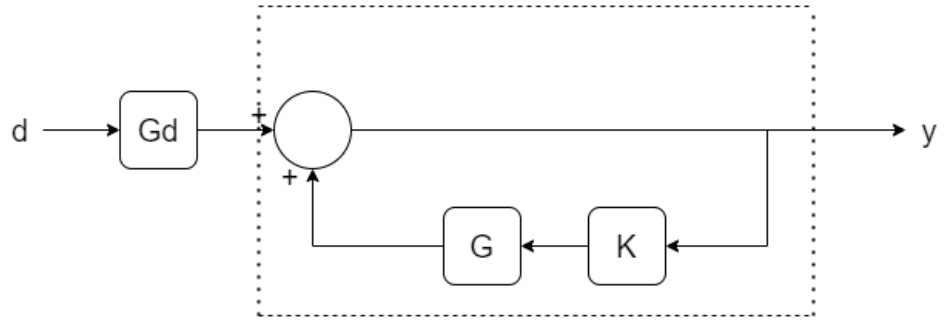


Figure 6: Diagram of the disturbance rejection system. As we can see, the original plant G and controller K appear in the feedback line this time, which we treated as a general subsystem. As a result, the transfer function of the system is recreated as a series connection of two subsystems.

In the above diagram, the G and K are already known from previous questions. The definition of G_d is below:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

where A is a 5×5 matrix that is given, B is a 5×1 matrix since we only select the third column for input V, C is a 1×5 matrix since we only select the first row for output ω_r, and D equals to zero. Noted that the C matrix does not need to multiply by -1 for system construction. The definitions of state variables, input, and output are shown:

$$x = \begin{bmatrix} \omega_r \\ \dot{z_1} \\ z_1 \\ \dot{z_2} \\ z_2 \end{bmatrix} u = \begin{bmatrix} V \end{bmatrix} y = \begin{bmatrix} \omega_r \end{bmatrix}$$

Using the **ss2tf** function, we acquired the transfer function of the new system:

$$tf = \frac{G_d}{1 - GK} \tag{7}$$

where definitions of Gd, G, and K can be found above.

We used the original PI controller to perform the same time-domain simulation as in the previous question. The visualization of the result is shown in figure 9.

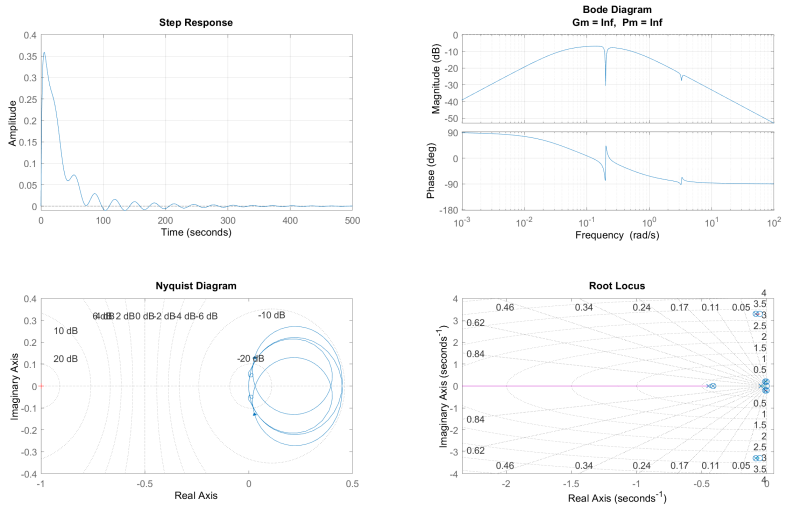


Figure 7: Time domain simulation result of the new disturbance rejection with unchanged PI controller.

From the plot, we can tell that the system rejects disturbance as required. This can be proved by the fact that the steady-state error reaches zero for the time-domain response. However, oscillations of the system indicate that the controller needs certain optimization for optimal performance.**Thus, we have to change the controller for the disturbance rejection scenario.**

To do that, we changed the P and I parameters of the system and plotted all the time-domain responses. From figure 8, one can easily conclude that with the increasing value of K_p, the height of the peak of the system decrease

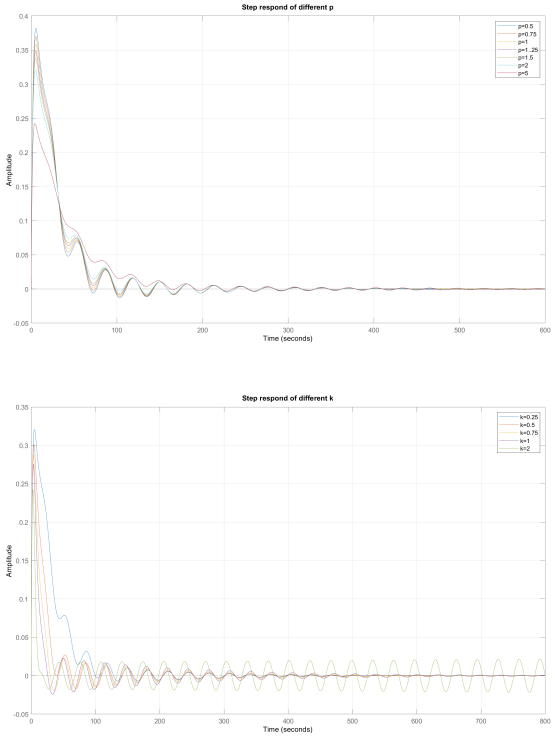


Figure 8: Time domain simulation result of the new disturbance rejection with PI controller. We changed the parameter P and parameter I to qualitatively analyze the influence of those parameters on the system's performance.

whereas the settling time increases as oscillation becomes more and more. In addition, increasing the K_i value results in more oscillation, which at a certain point breaks stability. After spending some time fine-tuning the PI controller, we acquired a relatively good controller for the system with $K_p = 1.5$ and $K_i = 0.25$. The visualization of the system can be found below as well. Noted that it is certain that even though using the word "good", the controller is far from being perfect.

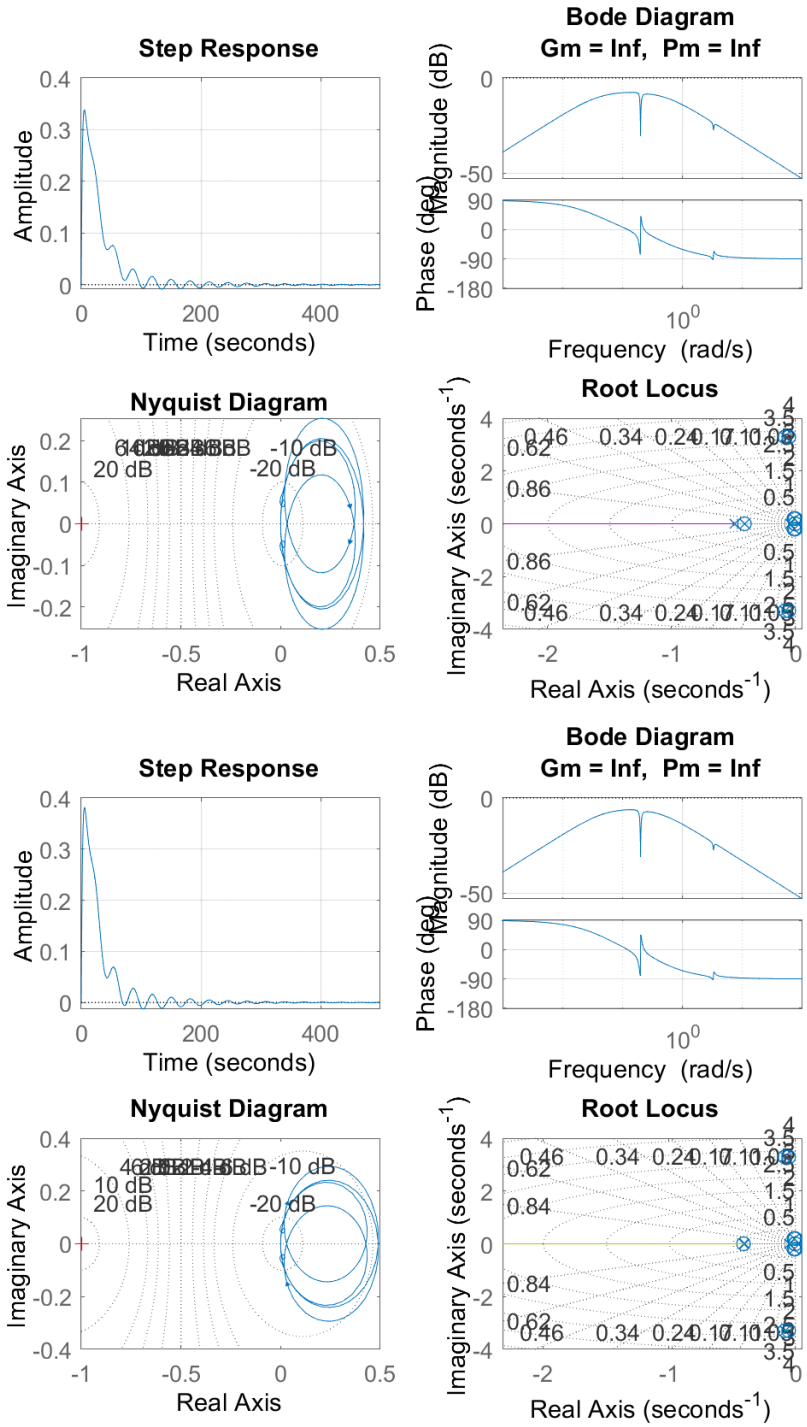


Figure 9: The final controller that we created. The above plot is the tuned one, and the below one shows the process of fine-tuning.

2 Multi variable Mixed-Sensitivity

The objective of this question is to do a mixed-sensitivity design of a Multi-variable controller to achieve the required performance and robustness of the system. Contrary to the first section, now both channels can be used as control inputs (β and τ_e) where β is the blade pitch angle $[rad]$ and τ_e is the generator torque $[Nm]$. The generator torque τ_e is constrained by the fact that the power production must remain unchanged. This results in the following state space representation:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

$$x = \begin{bmatrix} \omega_r \\ \dot{z_1} \\ z_1 \\ \dot{z_2} \\ z_2 \end{bmatrix} u = \begin{bmatrix} \beta \\ \tau_e \end{bmatrix} y = \begin{bmatrix} \omega_r \\ z \end{bmatrix}$$

2.1 Relative Gain Array

The Relative Gain Array (RGA) is a commonly used method for determining the best channel pairing. As a general rule for RGA-values, the closer the values on the diagonal are to identity, the higher the decoupling between the channels. Otherwise the control on one channel will highly effect the other channel. To compute the RGA for the given values, we use the following equation:

$$RGA(G) = \Lambda(G) \triangleq G \times (G^{-1})^T \tag{8}$$

Here $\times$ represents the Hadamard product. We decided to opt for the following equation:

$$\Lambda(G) = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix} \tag{9}$$

Where λ_{11} can be calculated with:

$$\lambda(G) = \frac{1}{1 - \frac{g12 * g21}{g11 * g22}} \tag{10}$$

for $\omega_1 = 0$ and $\omega_2 = 0.3 * 2 * \pi$ this results in the following values:

$$RGA(\omega_1) = \begin{bmatrix} -0.6554 & 1.6554 \\ 1.6554 & -0.6554 \end{bmatrix}$$

$$RGA(\omega_2) = \begin{bmatrix} -0.1166 & 1.1166 \\ 1.1166 & -0.1166 \end{bmatrix}$$

From this we can conclude that $\omega_2 = 0.3 * 2 * \pi$ resulted in the best values since the values on the diagonal were the closest to identity and thus had the highest decoupling.

2.2 MIMO poles and zeros

The requested MIMO poles and zeros were calculated in the following way. First, we constructed the transfer function matrix so we could find all the poles and zeros without any zero-pole cancellation. This resulted in 7 zeros and 10 poles. After this, we used zero-pole cancellation to find the final zeros and poles. The resulting zeros and poles are [-0.0078 ± 1.2358i] and [-0.0832 ± 3.2936i, -0.4104, -0.0106 ± 0.2022i]. We can conclude from this that the system is asymptotically stable since all zeros have negative real parts.

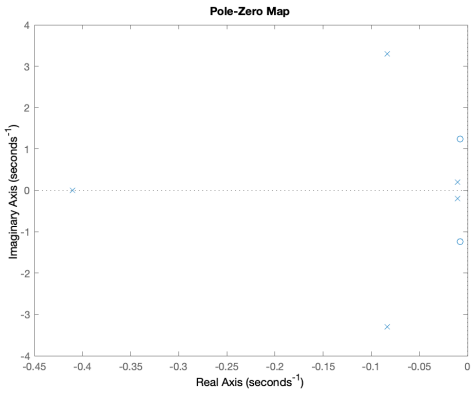


Figure 10: Pole-Zero map

2.3 Performance weights design

In this section, we are asked to design an appropriate W_{p11} for the perfomance weights (see table 1), given a cut-off frequency of 0.3 Hz, a low-frequency disturbance attenuation of 10^{-4} and H_∞ norm of the sensitivity function of 3.

W_p	$\begin{bmatrix} W_{p11} & 0 \\ 0 & 0.2 \end{bmatrix}$
W_u	$\begin{bmatrix} 0.011 & 0 \\ 0 & \frac{5*10^{-3}*s^2+7*10^{-4}*s+5*10^{-5}}{s^2+14*10^{-4}*s+10^{-6}} \end{bmatrix}$

Table 1: Given performance Weights

We can express W_{p11} as:

$$W_{p11} = \frac{\frac{s}{M_i} + \omega_{Bi}}{s + \omega_{Bi} * A_i} \tag{11}$$

Where A_i is the attenuation of low frequency-disturbances of 10^{-4}, ω_{Bi} is the bandwidth with a value of $2 * \pi * 0.3$ Hz and M_i is the H_∞ norm with a value of 3.

$$W_{p11} = \frac{\frac{s}{3} + 2 * \pi * 0.3}{s + 2 * \pi * 0.3 * 10^{-4}} \tag{12}$$

2.4 Block diagram

The figure below is the block diagram for the generalized plant, where W are the inputs signals β and τ_e, G the plant, K the controller, u the controller output, v the error, W_p and W_u the performance weights and z_1 and z_2 the performance signals.

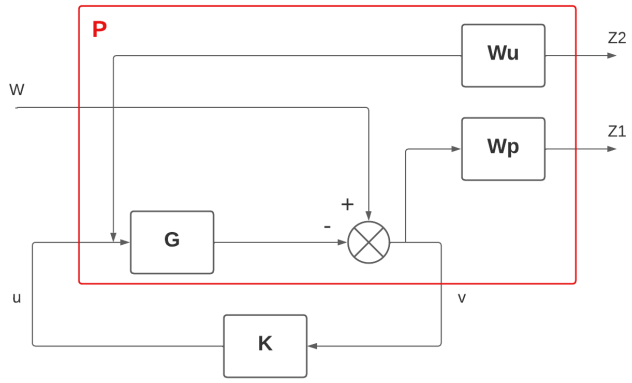


Figure 11: Block diagram

The equations for the performance signals are:

$$z_1 = W_p * W - W_p * G * u \tag{13}$$

$$z_2 = W_u * u \tag{14}$$

The equations for the performance weights can be found in section 2.3

2.5 Generalized plant

Based on the block diagram and equations from the previous section, the generalized plant can be mathematically described as:

$$\begin{bmatrix} z_1 \\ z_2 \\ v \end{bmatrix} = \begin{bmatrix} W_p & -W_p * G \\ 0 & W_u \\ I & -G \end{bmatrix} \begin{bmatrix} W & u \end{bmatrix}$$

This system has 4 inputs, 2 inputs at w and 2 inputs at u, and produces 6 outputs, 2 outputs for z_1, z_2 and v. Based on the 5 inherent states of G, the 2 poles of W_u and the single pole of W_u, we are expecting the system to have 8 states. This matches with the result from the generalized plant we built in Matlab.

2.6 Interpretation performance weights

The weights we found in section 2.3 can be interpreted in the following way. W_u puts a weight on the control inputs while W_p puts a weight on the errors.

$$W_u = \begin{bmatrix} 0.011 & 0 \\ 0 & \frac{5*10^{-3}*s^2+7*10^{-4}*s+5*10^{-5}}{s^2+14*10^{-4}*s+10^{-6}} \end{bmatrix} \tag{15}$$

W_u puts a constant penalty of a factor of 0.011 on the blade pitch angle β and a low-pass filter on the torque τ_e. The low pass filter (W_{u22}) provides a smoother form of signal and removes the effect of the high fluctuations of the generator torque and leaves the long-term trend.

$$W_p = \begin{bmatrix} \frac{\frac{s}{3}+2*\pi*0.3}{s+2*\pi*0.3*10^{-4}} & 0 \\ 0 & 0.2 \end{bmatrix} \tag{16}$$

W_p puts a constant penalty of a factor of 0.2 on z and a lag compensator on W_r. The lag compensator designed in section 2.3 allows a high gain at low frequencies and a small gain at high frequencies.

2.7 mixed-sensitivity generalized controller

Through the Matlab function **hinfsyn**, we obtained the generalized controller K. The Nyquist stability theorem states that for a system to be stable the Nyquist contour has to encircle the origin Z-P times where Z represents the number of unstable closed-loop zeros and P represents the number of unstable closed-loop Poles. The closed loop is defined as $L = G * K + I$ and has no unstable zeros and 2 unstable poles. According to theorem, for the system to be stable we should have 2 counterclockwise encirclement's around the origin.

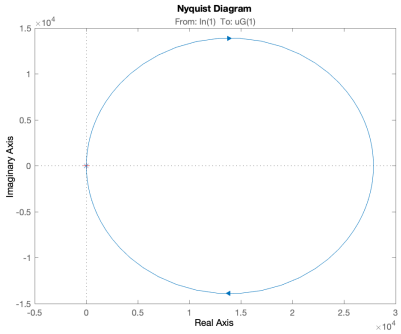


Figure 12: Nyquist plot

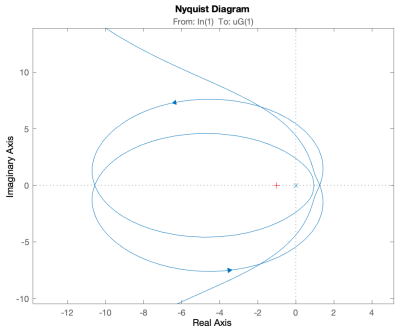


Figure 13: Nyquist plot close-up

From figure 12 and 13 we can see two counterclockwise encirclement's and thus conclude that the system is stable.
Moreover, the controller K has 8 states which is the same amount as the generalized plant.

2.8 Time-domain simulation

In this section, we explain the performed time-domain simulation and the fundamental limitations of the design. For the first simulation we look at the step response of the close loop of the generalized plant, where z_1 is the effect on the errors and z_2 is the effect on the input. We notice an average settling time of around 45 seconds for the blade pitch angle β input, which is an improvement over the SISO-controller. For the generator torque τ_e input we notice no steady state contribution which matches with the set constraints in the beginning of the chapter.

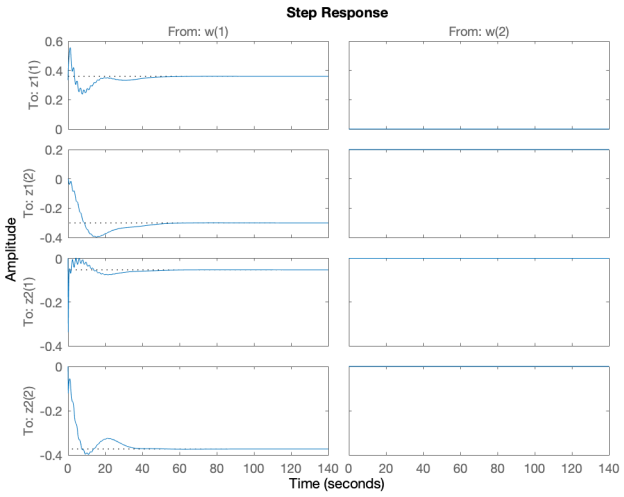


Figure 14: Simulation results z_1 and z_2

From figure 15, we can tell that the system rejects the disturbance as required. This again can be concluded by the fact that the steady state error goes to zero. The oscillations slowly damps out but it takes a long time for the system to settle.

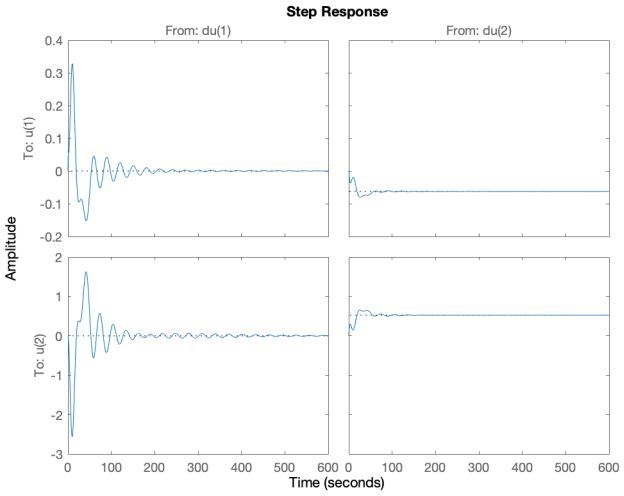


Figure 15: Disturbance rejection simulation

For Multi variable Mixed-Sensitivity, we can conclude on multiple fundamental limitations of the design:

- The Mixed sensitivity controller is of the 8^{th} order which is hard to implement in real life and requires a lot of computation power.
- From the plots, we can clearly see multiple steady state errors, undesired settling behaviour and a slow settling time.
- The mixed-sensitivity controller itself is unstable with two unstable poles.

3 MIMO Weighting Design

In this section, we explored an alternative way to design controllers using the robust control toolbox. From the given wind turbine data figure, we recognized a low-frequency part and some high-frequency parts. Our goal is to control those two parts separately. In addition, based on the given information, we were aware that torque τ_e is best used to counteract the high-frequency changes in wind speed, and blade pitch β is used for low-frequency control. In other words, for this MIMO system, the ideal control result will be:

- For high-frequency control (τ_e), a small gain is given to low-frequency inputs whereas a large gain is given to high-frequency inputs to counteract.
- For low-frequency control (β), a large gain is given to low-frequency inputs to counteract whereas a small gain is given to high-frequency inputs.

The structure of the following content is as follows. A generalized plant is given in subsection 1 for the MIMO system. In subsections 2 and 3, we design weight controllers W_u and W_p. The synthesized system is created in subsection 4, and figures are illustrated to validate the design. In the last subsection, we simulate the system by using the given wind data and a reference sine input with a frequency of $2 \times 10^{-3}\pi(rad/s)$ and amplitude of 2.

3.1 Generalized plant

The generalized plant of the system is shown below.

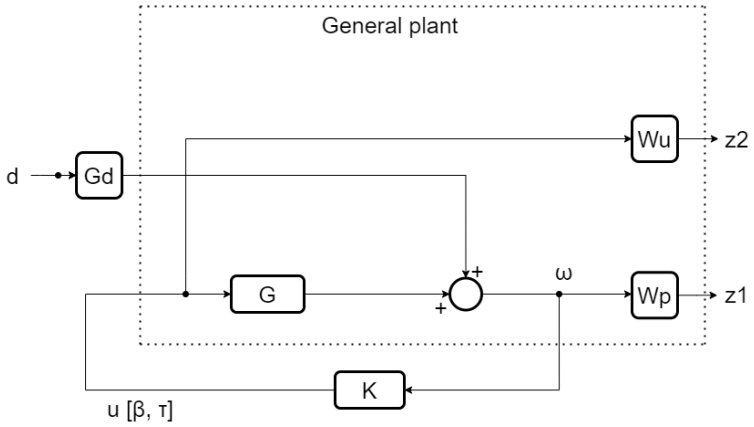


Figure 16: Generalized plant of the system

The generalized plant has 2 inputs and 3 outputs.

The two inputs are u which stands for control input and the disturbance d. Additionally, the control input u is composed of blade pitch β and torque τ_e. Noted that in the designing process, we see the control input as two separate inputs β and τ_e instead of u. But theoretically speaking, there should be two specific inputs in total.

The three outputs are z_1, z_2, and ω_r. According to figure 17, the relation between weight controller outputs z_1 and z_2 can be represented as below.

$$z_1 = W_p\omega + W_pG_dd \tag{17}$$

$$z_2 = W_uu \tag{18}$$

Noted that the disturbance transfer function G_d is not included in the generalized plant for robustness purposes.

3.2 Output controller weight w_p design

The general diagram of the MIMO system can be represented below. From

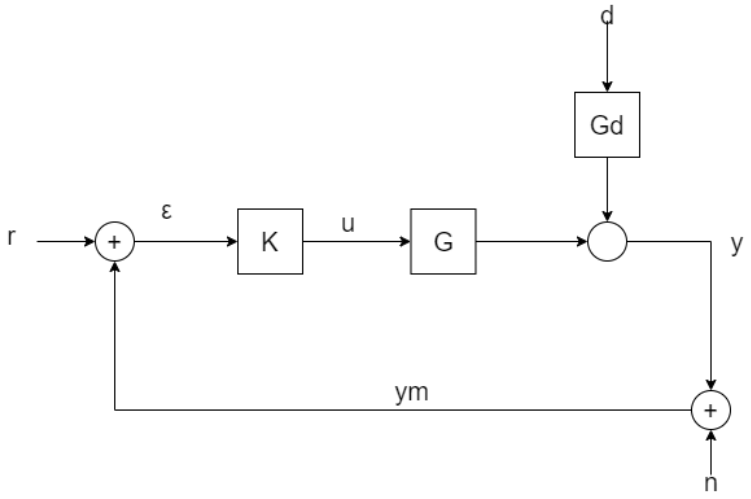


Figure 17: Generalized diagram of the system

this plot, we are able to derive the relation between reference input r, disturbance input d, and normal input n. The relation can be illustrated by equations (19) in which we define the term $(I+GK)^{-1}$ as sensitivity function S.

$$y = (I + GK)^{-1}GKr + (I + GK)^{-1}G_d d - (I + GK)^{-1}GKn \tag{19}$$

The goal of the design is to minimize the H_∞ norm of

$$N = \begin{bmatrix} W_P S \\ W_u K S \end{bmatrix} \tag{20}$$

A common choice of W_p is

$$Wp = \frac{\frac{s}{M} + \omega_B}{s + \omega_B A} \tag{21}$$

with $A \ll 1$. We first chose A to be 10^{-3} to ensure approximate integral action with $S(0) \approx 0$. Furthermore, we selected Mi about 2 for all outputs. The choosing of ω_B requires us to think about the property feature of both

weight functions. For a good weight controller, we have to fulfill the requirements of W_p bounded sensitivity function S and W_u bounded function KS. In terms of mathematical representation, those two requirements are shown below.

$$||S||_\infty \leq |W_1^{-1}| \tag{22}$$

$$||KS||_\infty \leq |W_u^{-1}| \tag{23}$$

Thus, when choosing parameters, we first randomly chose one value of ω_B and plot the bode diagram of the sensitivity function S along with the bode diagram of function $\frac{1}{W_p}$. Then, we tuned the parameters so that function $\frac{1}{W_p}$ bounded sensitivity function S nicely. The result of this process will be shown in subsection 4. After a few tries, we found the optimal value for ω_B is 0.08. Therefore, the weight W_p that ensures output disturbance rejection is below:

$$Wp = \frac{\frac{s}{2} + 8 \times 10^{-2}}{s + 8 \times 10^{-5}} \tag{24}$$

The bode diagram of W_p is created as well to validate our design. The

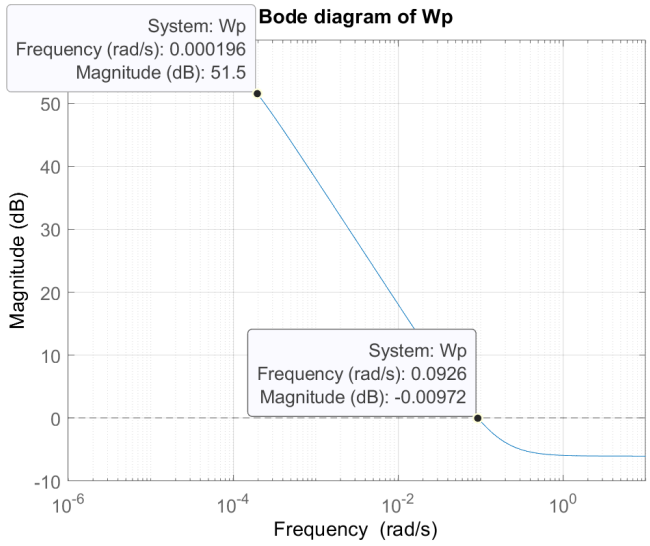


Figure 18: The bode diagram of W_p

bode diagram shows that W_p shares the feature of a low pass filter, which validates our design since most noises are high-frequent. Thus, a low-pass filter is able to stop the noise from influencing the system's behavior. And

that is the purpose of our design.

In order to scrutinize the wind data, we performed Fourier Transform on the given wind data and plotted the distribution of information of the given data. The explanation of the data can be found in the caption part.

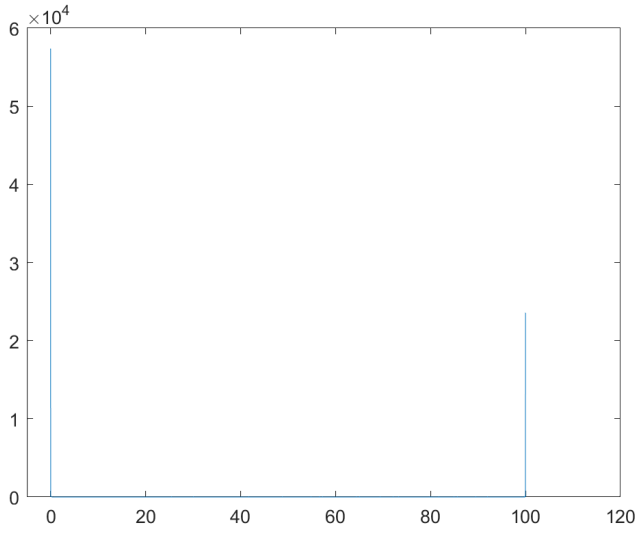


Figure 19: Wind data after fast Fourier transform, the two peaks in the data validate the existence of a low pass filter for output disturbance rejection. In detail, we maintain that the peak in the low-frequency part stands for useful information while the peak in the high-frequency part stands for noises. Consequently, a low-pass filter is able to filter out those noises and only leave useful information behind.

3.3 Input controller weight w_u design

Since the control of high-frequency signal and low-frequency signal is independent, so we know that the matrix W_u should be diagonal. And since we have only two control inputs, the dimension of the matrix should be 2×2 with the definition below.

$$Wu = \begin{bmatrix} W_u(\beta) & 0 \\ 0 & W_u(\tau_e) \end{bmatrix} \tag{25}$$

The feature of W_u is mentioned at the beginning of this chapter. Therefore, two entries of the diagonal matrix should offer gains for two inputs accordingly. For τ_e, the small gain on low frequency and large gain on high frequency meets the feature of a high pass filter, and for the case of β, it should be a low pass filter as well. According to the given text, we chose the cut-off frequency for both filters $0.5Hz$. Therefore, using the Matlab function **makeweight**, we design W_u for the system. The bode diagram of W_u is shown. Thus, the mathematical representation of W_u is shown in the

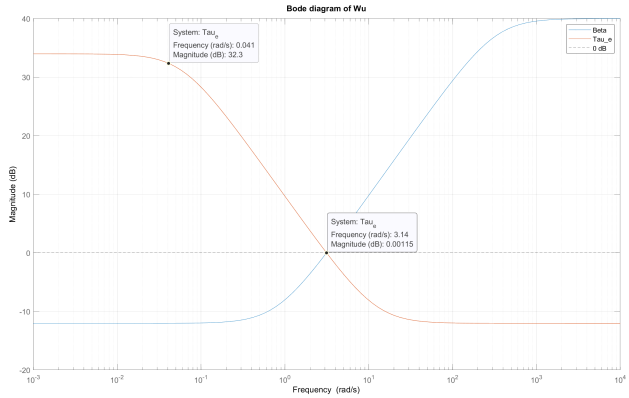


Figure 20: Bode diagram of W_u, noted that the yellow line and the blue line represent the τ part and the β part separately. For τ, a low pass filter is designed to counteract the high-frequency input. Additionally, for β, a high pass filter is designed to counteract the low-frequency input. Both filters have the cut-off frequency of $0.5Hz$, and both filters give $40dB$ and $-15dB$ for the part that it allows to pass and doesn't allow to pass.

below equation.

$$Wu = \begin{bmatrix} \frac{100s+81.07}{s+324.3} & 0 \\ 0 & \frac{0.25s+3.041}{s+0.06082} \end{bmatrix} \tag{26}$$

To scrutinize the data, we perform the Fourier Transform of the data again for some insights into the cut-off frequency of the filter. This time, we zoomed in for the frequency to gain more knowledge.

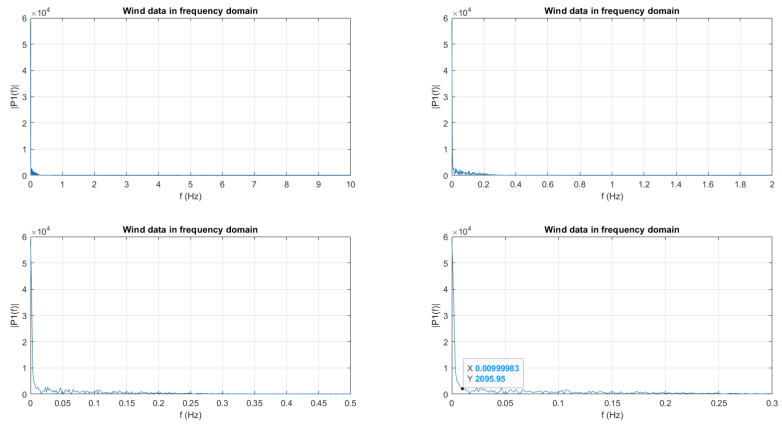


Figure 21: We scrutinized the wind data for some insights about the cut-off frequency. From the plot, we can see that at the frequency of 0.5Hz, there is not much information left. Thus, we think the system might perform better if the cut-off frequency is more accurate (For example, at the frequency of 0.001Hz). The reason is that we noticed that the amount of information in the data changes dramatically before 0.001Hz and after 0.001Hz.

3.4 Synthesized system visualizing

Using Matlab function **hinfsyn**, we synthesized controllers K with the designed weight. The sensitivity function S and complementary sensitivity function T are presented. To examine our design, we plot the controllers'

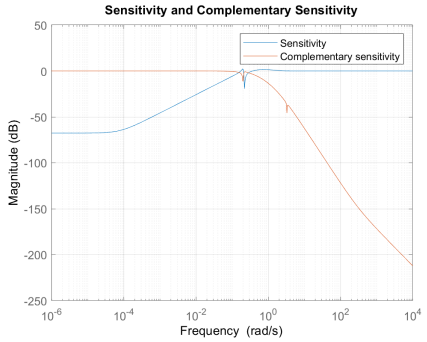
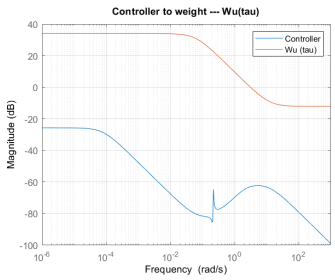
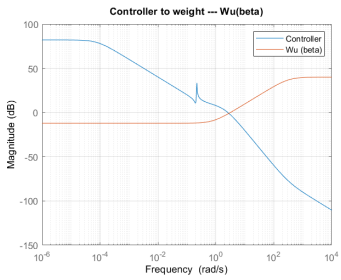


Figure 22: Sensitivity function S and complementary sensitivity function T.

bodes versus their weight first.



(a) High-frequency controller versus $W_u(\tau_e)$.



(b) High-frequency controller versus $W_u(\beta)$.

For high-frequency control τ_e, when it is low-frequency there is a low gain and when it is high-frequency there is a large gain, which functions as expected. The expected features can be observed in low-frequency control β as well. Thus, the designed controllers reach our expectations.

Then, to check the bounding of W regarding sensitivity functions, we visualized the sensitivity function with respect to its weights. According to

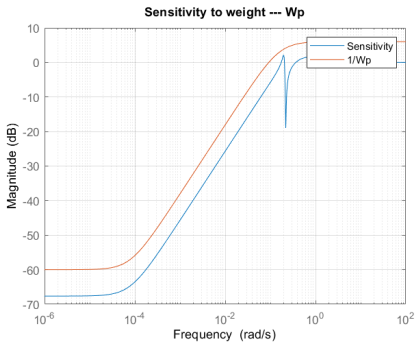
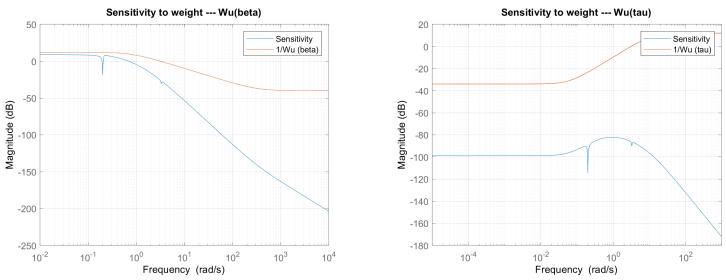


Figure 24: Sensitivity function S and output disturbance rejection weight W_p.

the above-mentioned analysis equation (22) and equation (23), W_p should create a bound for the sensitivity function S, which can be shown by figure 24. Then, bode diagrams of w_u and sensitivity function S are presented.



(a) Sensitivity function versus $W_u(\beta)$.

(b) Sensitivity function versus $W_u(\tau_e)$.

From the above figure, we can see that the high-frequency part weight ($W_u(\tau_e)$) created a nice bound for the sensitivity function and the low-frequency part weight ($W_u(\beta)$) functions as expected as well. Thus, we can conclude that the designed weights meet all the requirements. Noted that when creating the sensitivity function, we took the disturbance transfer function G_d into consideration.

3.5 System simulation

We simulated the system using two input signals. The first one is the given wind data. The simulation result for this input signal is given in figure 26. We can tell from the simulation result that β and τ_e counteract the low- and

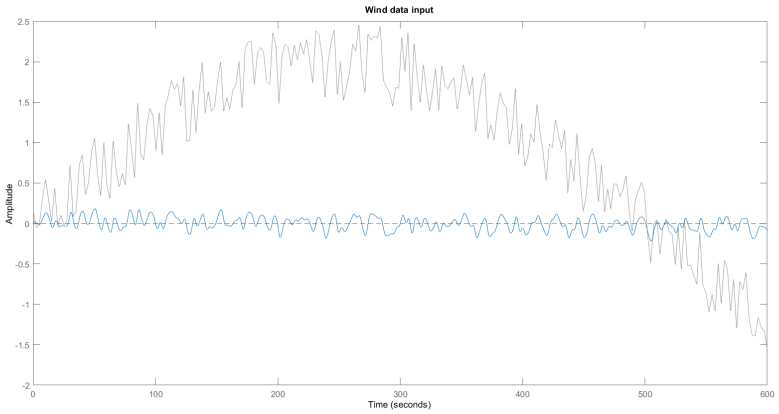


Figure 26: Given wind data simulation. The grey line stands for the input and the blue line stands for the output of the general plant.

high-frequency input according to our will. The reference $0dB$ line and the sin shape output justify this observation further.

To check this, we change the input signal to a sin wave with a stipulated frequency and amplitude. Since the cut-off frequency of W_u is $0.5Hz$ which is more than the frequency of the sin input, the ideal output of the system will be β tracks the sin input while τ_e remains still. The simulation result for this input is presented in figure 27. As is shown, the output of the system goes as we expected. Therefore, the designed MIMO system achieves all the requirements.

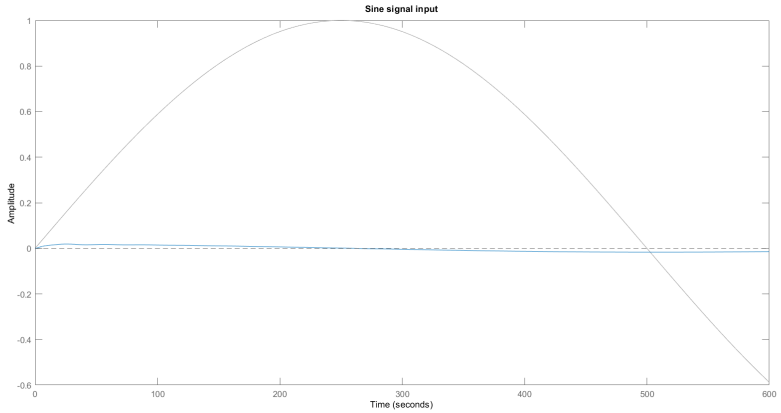


Figure 27: Sine data simulation. Given wind data simulation. The grey line stands for the input and the blue line stands for the output of the general plant. From the plot, β successfully tracks the sin input signal while the output of τ_e almost remains zeros compared to the reference 0dB line.