Robust Control

Part 1: SISO Analysis and Control Design

Ilse Jansen - 6195733
Daan van Haasteren - 4665465

November 2024

The system of a floating wind-turbine has been linearised with respect to wind speed $V_{lin} = 14$ [km/h]. The following state-space model with inputs β and τ_e and input disturbance V now describes this system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{15} \\ \vdots & \ddots & \vdots \\ A_{51} & \dots & A_{55} \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ \vdots & \vdots & \vdots \\ B_{51} & B_{52} & B_{53} \end{bmatrix} \begin{bmatrix} \beta \\ \tau_e \\ V \end{bmatrix}$$

$$\begin{bmatrix} \omega_r \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix}$$

SISO Analysis and Control Design

For this part, a SISO system is considered with input blade pitch angle β and output generator speed ω_r. The state-space model may be reduced to:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{15} \\ \vdots & \ddots & \vdots \\ A_{51} & \dots & A_{55} \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} B_{11} \\ \vdots \\ B_{51} \end{bmatrix} \begin{bmatrix} \beta \end{bmatrix}$$

$$\begin{bmatrix} \omega_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix}$$

1. Open-Loop Bode plot and Pole-Zero Map

In order to design a controller, the system to be controlled can be analysed in the frequency domain to effectively determine the controller specifications. The Bode diagram and pole-zero map of the plant can be seen in Figure 1 below.

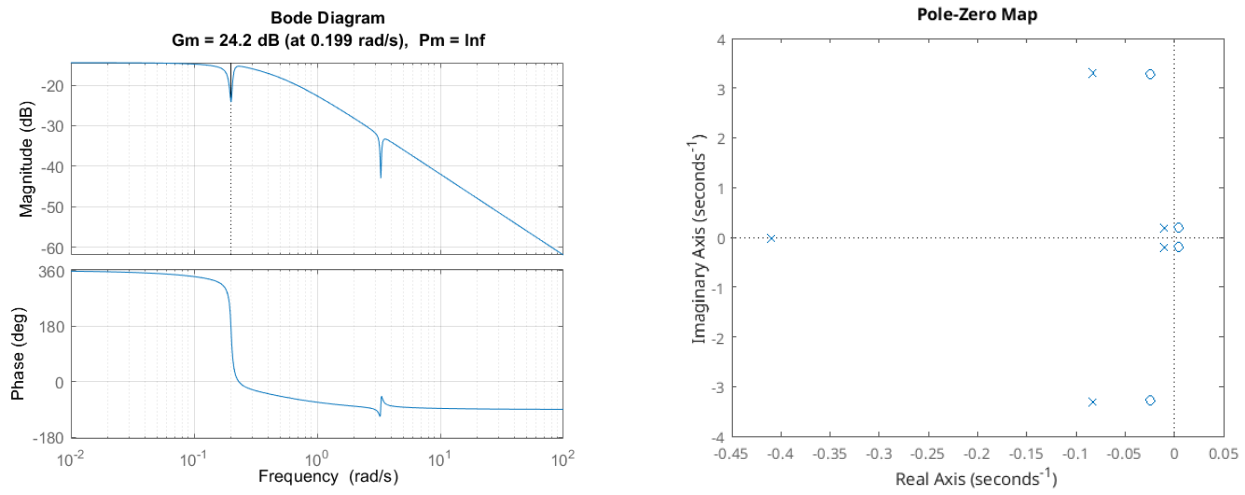


Figure 1: Bode plot and pole-zero map of the uncontrolled wind turbine plant

From the magnitude plot in Fig 1(a), it should be observed that the gain is high for low frequencies, meaning that the system effectively tracks these. A sharp notch occurs at low frequencies, around the same frequency as the gain margin (0.199 rad/s). This notch corresponds to the pole-zero pair around the origin in Fig 1(b), and may affect the ability to control the system around these frequencies. A second notch at slightly higher frequencies (≈ 3 rad/s) does not have as strong as an effect, since the frequencies are already more attenuated by the plant. An increased negative slope indicates more poles at higher frequencies. Indeed, in the pole zero map this can be observed to be located around -0.4 on the real axis. The gain margin of the system is 24.2 dB, as this is where the phase crosses 180°. While the phase of the system initially stays close to 360°, the pair of complex conjugate poles near the origin cause a significant phase drop.

As can be seen in Fig 1(b), the system has a total of four zeros, of which two are in the right half-plane (RHP) at $0.004 \pm 0.200i$. These zeros ensure that the system is not minimum-phase. In the time-domain, these zeros introduce a lag in the response. This makes the system more vulnerable to disturbances, as the reaction to these disturbances will be delayed. In addition, the system must respond quickly to high-frequency inputs in order to achieve a high-bandwidth. However, the time-delay in the response complicates this, compromising stability at these high frequencies.

There are a total of 5 poles in the system, of which two poles located are located close to the Im-axis and origin, namely at $-0.010 \pm 0.202i$. The real part of these poles are small, relative to their imaginary part. This indicates a low damping ratio and a slow response, both of which are further detrimental to a high bandwidth. In the time domain, the low damping of the poles enable sustained oscillations in response to a step input or disturbances. In addition, the system will take longer to settle after an input or disturbance relative to a pole further from the origin, due to the slow response. These observations mean that a trade-off must be made between stability and performance, balancing the need for higher bandwidth against the risk of instability and excessive control effort.

2. Controller requirements in Frequency Domain

The goal is to design a reference tracking controller that achieves the highest possible bandwidth whilst remaining within the design requirements:

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

In addition to these requirements, the controller should address the issues described in the previous section. To begin with, the controller should enhance damping of the system. This may be done by shifting poles farther into the LHP to speed up the response in the time domain and reduce oscillations. Furthermore, the zeros in the RHP

must be addressed in order to counteract the lag in the time-response and increase the phase margin, as well as the bandwidth.

These requirements can be achieved by manipulating the equivalent metrics in the frequency domain. To begin with, requirement (1) asks for minimal settling time. This can be achieved by implementing a large gain in a proportional controller. Furthermore, the percentage overshoot can be calculated from the phase margin as:

$$\% \; OS = 100 \; exp(-\frac{\pi}{\sqrt{4Q^2 - 1}}) \tag{1}$$

with

$$Q = \sqrt{tan(PM - 90°) + tan(PM - 90°)^2} \tag{2}$$

where PM is the phase margin in degrees. Using these equations, this results in the desired phase margin to be at least 118.67° in order to ensure an overshoot of $< 1\%$ and fulfil requirement (2).

Lastly, design requirement (3) requires the absence of a steady-state error. The final value theorem (FVT) may be used to obtain the controller design requirement. This states that:

$$\lim_{t \to \infty} e_{ss} = \lim_{s \to 0} \frac{sR(s)}{1 + K(s)G(s)} \tag{3}$$

where $R(s)$ is the Laplace transform of the reference signal, which is $R(s) = \frac{1}{s}$ for a step input, while $K(s)$ is the Laplace transform of the controller and $G(s)$ is the plant. This may be rewritten to:

$$\lim_{t \to \infty} e_{ss} = \frac{1}{1 + \lim_{s \to 0} K(s) \lim_{s \to 0} G(s)} \tag{4}$$

Substituting G(s) and taking the limit:

$$\lim_{t \to \infty} e_{ss} = \frac{1}{1 + \lim_{s \to 0} K(s) \frac{0.0346}{0.1827}}$$

Clearly, in order for the steady-state error to reduce to zero, the limit of K(s) must be infinite. From this, it may be concluded that an integral term is necessary in order to fulfil design requirement (3).

Hence, a controller was designed with a large proportional term to ensure high gain, and an integrator term. No derivative term was necessary achieve design goals, hence was omitted to keep the control inputs low. A PI controller with $P = 0.8$ and $I = 0.28$ controls the plant adequately while fulfilling the design requirements and ensuring the desired phase margin.

3. Controller Performance

Figure 2 shows the step response of the open-loop, uncontrolled plant, as well as that of the closed-loop system with the PI controller. The specifications of the system performances are summarised in Table 1.

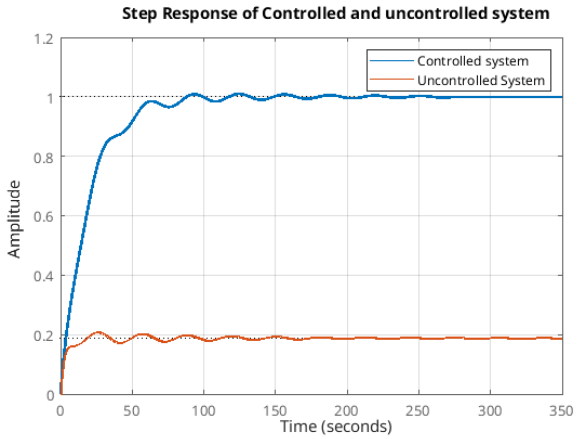


Figure 2: Step response of uncontrolled plant and controlled system

Parameter	Plant	Controlled System
RiseTime	13.7	45.8
SettlingTime	168.7	82.6
Overshoot (%)	10.2	0.97
Peak	0.21	1.01
PeakTime	26.5	124.5
Steady-State Value	0.19	1.00

Table 1: System Performance Metrics

Clearly, the open-loop uncontrolled system has a steady-state offset from the step value of 1, namely it settles at 0.19. This is corrected in the controlled case, which settles at 1.00. While the response of the controlled system is significantly slower than that of the plant itself, the settling time of the controlled system after a step input is much faster, with a reduction of $> 50\%$. In addition, the overshoot has significantly reduced from $> 10\%$ in the uncontrolled plant to $< 1\%$ in the controlled system. It may therefore safely be concluded that the PI controller effectively controls the plant and reaches all design specifications.

4. Controller Performance for Disturbance Input

When applying the SISO controller to reject disturbance input from wind V, the following state-space model must be considered;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{15} \\ \vdots & \ddots & \vdots \\ A_{51} & \dots & A_{55} \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} B_{13} \\ \vdots \\ B_{53} \end{bmatrix} \begin{bmatrix} V \end{bmatrix}$$

$$\begin{bmatrix} \omega_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_r \\ \dot{z}_1 \\ z_1 \\ \dot{z}_2 \\ z_2 \end{bmatrix}$$

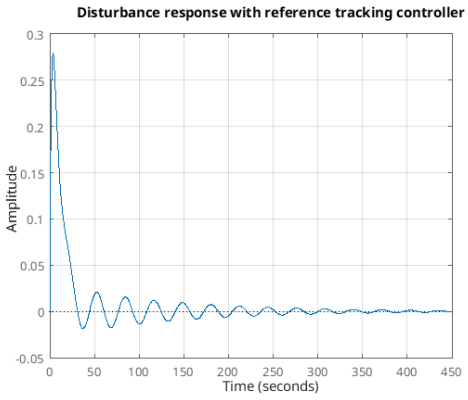


Figure 3: Disturbance Rejection response of PI controller

Parameter	Controlled system
TransientTime	213.5
Peak	0.28
PeakTime	3.67

Table 2: System Performance Metrics - Disturbance Input

Figure 3 shows the response of the SISO system to a step disturbance on the input as in the state-space model above. The step disturbance response reaches its peak of 0.28 at 3.7s. After this peak, there is a sharp decline in amplitude, after which oscillations slowly reduce in amplitude. In order to adapt the controller for better performance in disturbance rejection, a higher integrator term could be used in order to reduce the effect of the oscillations. Caution should be taken as to not significantly increase the peak amplitude.

Multi-variable Mixed-Sensitivity

For this section, the MIMO system with inputs pitch blade angle β and torque τ_e is considered. This results in the following state-space model:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix} = \begin{bmatrix}A_{11} & \dots & A_{15}\\ \vdots & \ddots & \vdots\\ A_{51} & \dots & A_{55}\end{bmatrix}\begin{bmatrix}w_r\\ \dot{z}_1\\ z_1\\ \dot{z}_2\\ z_2\end{bmatrix} + \begin{bmatrix}B_{11} & B_{12}\\ \vdots & \vdots\\ B_{51} & B_{52}\end{bmatrix}\begin{bmatrix}\beta\\ \tau_e\end{bmatrix}$$

$$\begin{bmatrix}\omega_r\\ z\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}w_r\\ \dot{z}_1\\ z_1\\ \dot{z}_2\\ z_2\end{bmatrix}$$

The transfer functions corresponding to this state-space are obtained by defining relations for each input to each output individually. The resulting 2x2 matrix is

$$G(s) = \begin{bmatrix}\frac{-0.07988s^4-0.003315s^3-0.8677s^2+0.006493s-0.03458}{s^5+0.5979s^4+10.98s^3+4.709s^2+0.5421s+0.1827} & \frac{-0.04874s^3-0.03481s^2-0.07497s-0.052}{s^5+0.5979s^4+10.98s^3+4.709s^2+0.5421s+0.1827}\\ \frac{-0.009564s^4-0.001683s^3-0.1039s^2-0.001187s-0.00414}{s^5+0.5979s^4+10.98s^3+4.709s^2+0.5421s+0.1827} & \frac{-0.001614s^2-2.522\times10^{-5}s-0.002465}{s^5+0.5979s^4+10.98s^3+4.709s^2+0.5421s+0.1827}\end{bmatrix}$$

1. RGA

The RGA gives a measure for how much impact each control input has on each of the outputs. Since the MIMO system considered has 2 inputs and 2 outputs, the RGA will be a 2x2 matrix, in which each column and row sums to 1. Ideally, the diagonals are close to zero and one, regardless of which diagonal is which. If they are exactly zero and one, this would mean that the system is completely decoupled, where one input fully determines one output, and the other input the other. This significantly simplifies control.
The RGA is found by:

$$RGA = G(G^T)^{-1}$$

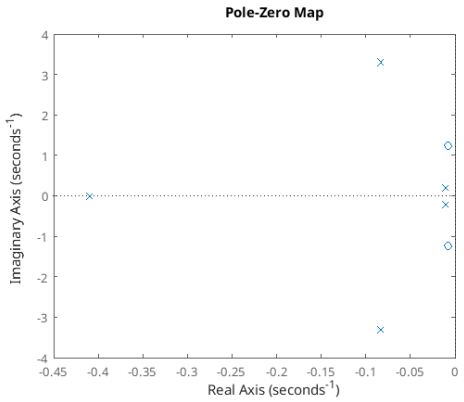
The RGA was evaluated at the frequencies $\omega_1 = 0$ and $\omega_2 = 0.3 \times 2 \times \pi$. This resulted in

$$RGA(\omega_1) = \begin{bmatrix}-0.6554 & 1.6554\\ 1.6554 & -0.6554\end{bmatrix} \quad \text{and} \quad RGA(\omega_2) = \begin{bmatrix}-0.0283+0.1398i & 1.0283-0.1398i\\ 1.0283-0.1398i & -0.0283+0.1398i\end{bmatrix} \quad \text{respectively.}$$

For both evaluations of the RGA, the diagonal terms are closer to zero, whereas the off-diagonal terms are closer to one. At $w_2 = 0.6\pi$, the values are closest to an ideal RGA. Therefore, the system is more decoupled at this frequency, which makes it easier to control. For both $w_1 = 0$ and $w_2 = 0.6\pi$, the RGA indicates that it is most efficient to use u_1 to steer y_2 and u_2 to steer y_1 for both frequencies.

2. MIMO poles and zeros

The MIMO poles and zeros are the union of the zeros and poles of all entries of $G(s)$ above. These are depicted in Fig. 4 and are listed below.



$$p = \begin{bmatrix}-0.4104\\ -0.0832 \pm 3.2936i\\ -0.0106 \pm 0.2022i\end{bmatrix}$$

$$z = \begin{bmatrix}-0.0078 \pm 1.2358i\end{bmatrix}$$

Figure 4: Pole-zero map of MIMO plant

The first thing to note is that all poles and zeros fall in the left half-plane, and therefore the system is stable. The strictly real pole, relatively far from the origin, corresponds to a fast mode that will exponentially die out rapidly. The complex conjugate poles around -0.08 result in in a slightly more damped and rapid dynamic. There is still a pole-zero pair very close to the imaginary axis. These will result in slow dynamics, and dominate the system more in the longer term, affecting the steady-state dynamics.

3. Design of $W_{p_{11}}$

The goal is to design $W_{p_{11}}$ for a cut-off frequency of 0.3 Hz, to get an attenuation of low-frequency disturbances of 10^{-4}, and H_∞ norm of the sensitivity function of 3. With these performance criteria, $W_{p_{11}}$ can be found using:

$$W_{p_i} = \frac{\frac{s}{M_i} + \omega_{Bi}}{s + \omega_{Bi} A_i}$$

where M_i is the desired $\mathcal{H}_\infty$ norm, ω_{Bi} is the desired closed loop bandwidth, and A_i is the desired disturbance attenuation within the bandwidth. The weight becomes:

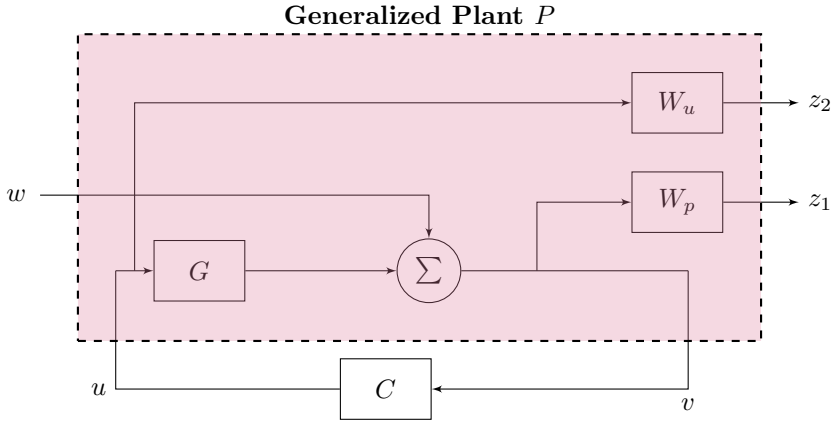
$$W_{p_{11}} = \frac{\frac{s}{3} + 0.6\pi}{s + 6\pi \times 10^{-5}}$$

This results in weighting matrices W_p and W_u of

$$W_p = \begin{bmatrix} \frac{\frac{s}{3}+0.6\pi}{s+6\pi\times 10^{-5}} & 0 \\ 0 & 0.5 \end{bmatrix} \qquad W_u = \begin{bmatrix} 0.005 & 0 \\ 0 & \frac{5\cdot 10^{-3}s^2+7\cdot 10^{-4}s+5\cdot 10^{-5}}{s^2+14\cdot 10^{-4}s+10^{-6}} \end{bmatrix}$$

4. Block diagram and deriving generalised plant

The block diagram model of the floating wind turbine and controller which includes the performance weights W_p and W_u is shown below. Input w is the wind disturbance, v is the disturbed state and u is the control input. G is the MIMO plant of the wind turbine and C is the controller, while z_1 and z_2 are the performance measures calculated by weights W_p and W_u respectively.



5. Generalized plant

The generalised plant P is now given by

$$P = \begin{bmatrix} W_p & W_p G \\ 0 & W_u \\ I & G \end{bmatrix} \tag{5}$$

where G, W_p and W_u are as previously defined. P has 6 outputs, 4 inputs, and 8 states, as expected. Outputs v, z_1 and z_2 are all of size 2x1, as well as inputs w and v, totalling 6 outputs and 4 inputs. The plant G(s) has 5 states as defined in the beginning of this section, and W_p and W_u are of order 1 and 2 respectively. This leads to the total of 8 states.

6. Interpretation of Performance Weights

Matrices W_p and W_u are weighting functions which represent performance requirements and constraints on the control effort. The W_p matrix gives a weight to the outputs of the plant, hence reflects the desired reference tracking, and penalises deviations from the desired output. Each entry is specific to an output. Namely, W_{p11} measures how much the rotational speed ω_r is allowed to deviate from the set-point when a disturbance of a certain frequency is applied to the system. For low frequencies, the denominator is small and the weight is relatively high, hence this is stricter on performance. On the other hand, the weight is less strict on higher frequencies. This fulfils the desired function of W_{p11} which was that the disturbance attenuation for low frequencies should be 10^{-4} up to a bandwidth of 0.6π, after which the maximum value of the sensitivity function is allowed to be 3. In a practical sense, W_{p11} regulates the rotor speed ω_r to ensure constant power output according to $P_o = \tau_e \cdot \omega_r$, as well as rejects disturbance due to changes in wind speed, which mostly occurs at low frequency.

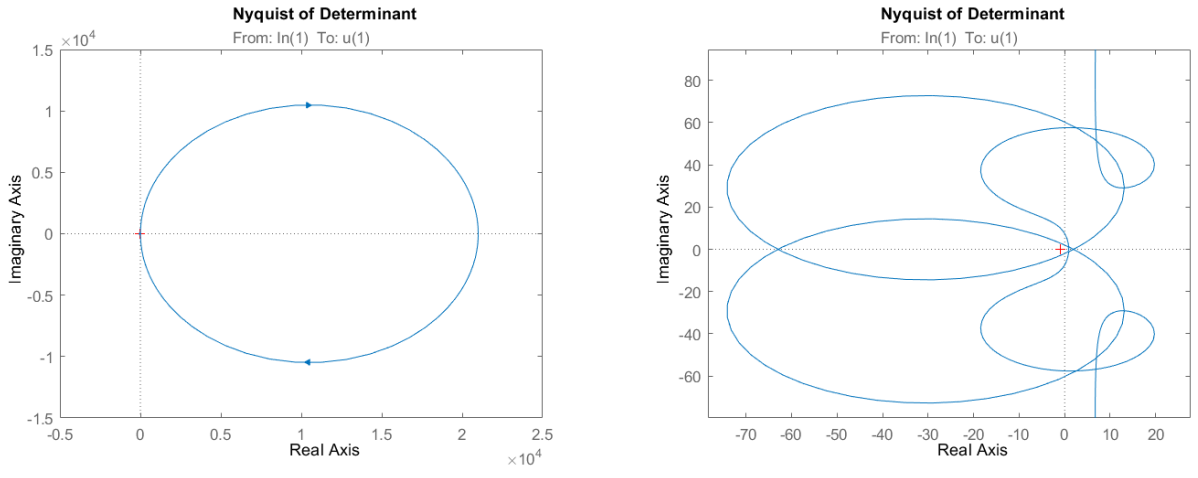
The W_{p22} gives a penalty on the second output, that is, the z-displacement of the nacelle. This output needs to stay below $\frac{1}{W_{P22}}$. As $W_p = 0.05$, this means the z displacement may not exceed 20 metres, a quite a loose constraint. This weight is constant, and thus does not vary with frequency.

The W_u block penalises control inputs, hence represents constraints on the magnitude of control effort. This takes into account the actuator saturation, energy limitations or robustness. The penalty on the first input is constant and independent of frequency, but penalizes excessive use of blade pitch angle β. This should stay below $\frac{1}{W_{u11}} = 200°$. The weight on the second input, generator torque τ_e, acts as a band-pass filter. This means it applies more weight to specific frequency ranges, which are specified by the dynamics of the numerator and denominator. It penalises high-frequency control actions to avoid for example the amplification of noise, while also penalising low-frequency control actions in order to avoid too slow responsiveness of the system. Overall, W_u penalises high control efforts, and especially so at frequencies specified by the bandpass filter.

7. $\mathcal{H}_\infty$ Controller Synthesis and Internal Stability

By defining the inputs and outputs of every element as in the generalised plant diagram, the generalised plant was built in Matlab using the 'connect' command. The 'hinfsyn' function was then used to synthesise a mixed-sensitivity generalised controller. For exact methods and code, refer to Appendix A. The output of this function consists of the general controller, the resulting closed-loop system and performance measure γ. The performance measure, which is the $\mathcal{H}_\infty$ norm of the system, should remain below 1. The obtained controller, has a norm of 0.5132, which implies that the worst-case amplification of input disturbances to outputs is significantly less than 1. Hence, this controller satisfies the desired performance criteria with a margin. As specified before, the generalised plant consists of a state-space model with 6 outputs, 4 inputs, and 8 states. The generalised controller on the other hand, has 2 outputs, 2 inputs, and 8 states. This too is as expected, as both input v and output u of the controller as in the diagram, have dimension 2x1. The eigenvalues of the synthesised controller are listed below. Clearly, the controller is not stable, as it has a positive complex conjugate pair in its eigenvalues. In order to evaluate internal stability of the closed-loop, the generalised Nyquist should be evaluated, plotted in Figure 5.

$$eig(H_\infty) = \begin{bmatrix} -0.0002 \\ -26.4187 \\ 0.0007 \pm 3.3316i \\ -0.0133 \pm 0.2018i \\ -0.0688 \pm 0.0708i \end{bmatrix}$$



(a) (b)

Figure 5: Nyquist of determinant and a close-up around the origin and -1 (+)

While it is difficult to draw any conclusions from Figure 5(a), Figure 5(b) illustrates how the generalised Nyquist of the system encircles the point -1, indicated by + in the plot, exactly twice in counter-clockwise direction. This indicates that while the synthesised controller is not stable, the closed loop is internally stable.

The H_∞ is an ideal controller, difficult to impossible to implement in real life. Nevertheless, it is useful to analyse the behaviour of the perfect controller, in order to design a controller which exhibits similar behaviour, but has specified constraints. The Bode-magnitude plot is an indicator of controller dynamics, and has been plotted below.

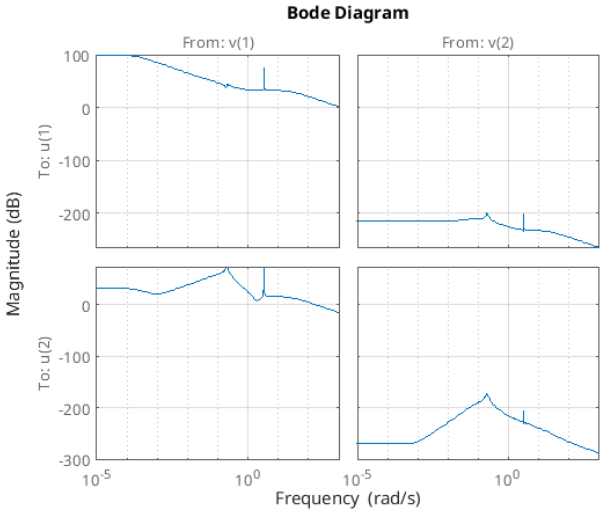


Figure 6: Bode plot of MIMO $\mathcal{H}_\infty$-controller

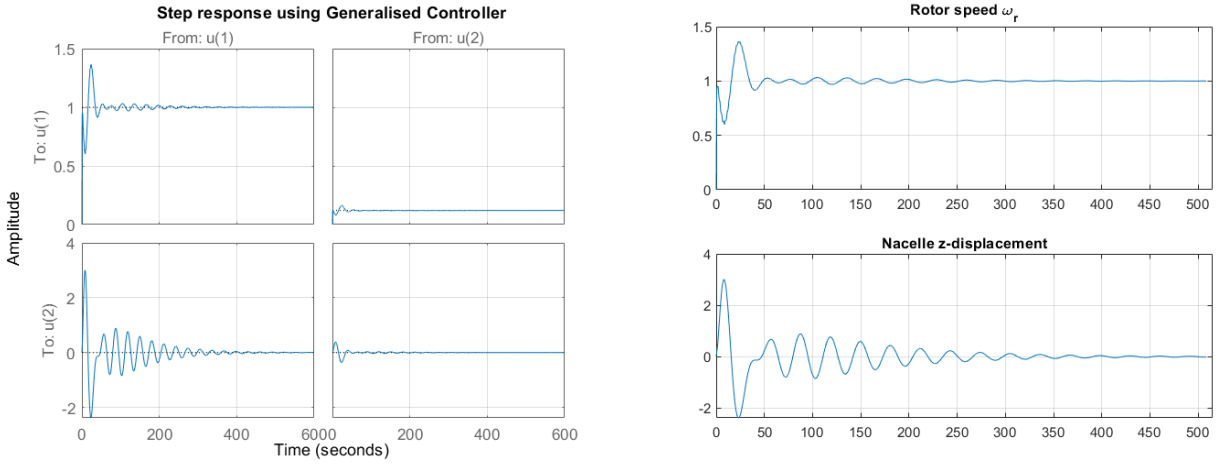
The first thing to note is that the second input, generator torque τ_e, has very high attenuation for all frequencies, namely in the order of magnitude of -200dB. Therefore, the ideal controller only uses the first input blade pitch angle β to steer the system.

Next, it is relevant to analyse the specific dynamics of this first input. The shape of the bode-magnitude diagram of the control on the first output, rotor speed ω_r, is that of a PID-controller. It has high gain at low frequencies, high attenuation at high frequencies, with relatively constant attenuation at frequencies of order 10^0. Interesting is the sharp peak occurring between 10^0 and 10^1 meaning control inputs of exactly those frequencies have a large response in the corresponding output channel. The influence of blade pitch angle β on the second output, the z-displacement of the nacelle, is controlled by an inverse notch filter, which has high gain for frequencies between 10^{-2} and 10^0. These properties of the H_∞-controller should be taken into account when later designing a constricted controller.

8. Performance in Time-Domain

Reference Tracking

The step response of the closed-loop using the H_∞-controller has been plotted below. Since the second input will minimally affect the system, only the response of the system to a step on blade pitch angle β is relevant.



(a) (b)

Figure 7: Step-response of the CL-system and a close-up on input blade pitch angle β

The controller can be seen to act rather aggressively in response to the input step on the pitch blade angle. This has as a result that oscillations in the rotor speed decay much faster than in the previously designed PI-controller. However, the overshoot of this response is nearly 40%, and the sharp initial peak followed by a sharp decay is not a desired response in a physical context. This would lead to strong and sudden loads on the rotor, which might cause material fatigue. The z-displacement initially peaks around ± 2 metres, which is in line with the previously described constraint of 20 metres. While oscillations are relatively strong initially, they are reduced significantly, though not completely within this time frame. Since the SISO PI-controller has been designed for the input β and output ω_r, this controller can be compared to its respective channel of the H_∞-controller. The comparison has been plotted below, while the performance specifications can be found in Table 3.

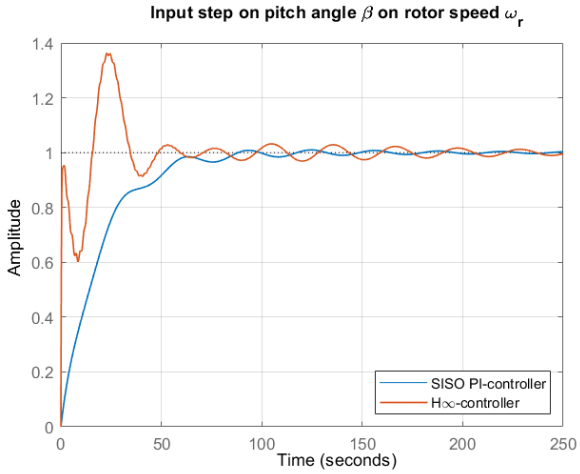


Figure 8: Step response on pitch angle β of both the PI- and H_∞-controller

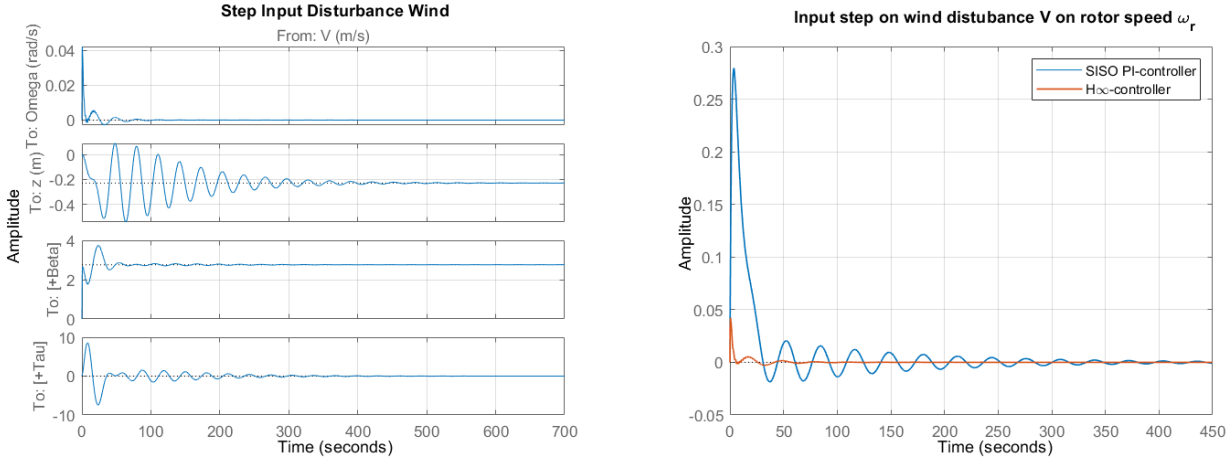
Parameter	PI-controller	H_∞-controller
RiseTime	45.8	0.51
SettlingTime	82.6	168.8
Overshoot (%)	0.97	36.34
Peak	1.01	1.36
PeakTime	124.5	22.9

Table 3: Performance of PI- and H_∞-controllers on step input on pitch angle β

As described, the H_∞-controller acts aggressively, hence has a rise-time reduced by 99% compared to the PI-controller, and a peak-time reduction of 80%. However, its oscillations decay much slower, and the settling time is actually two times slower than the PI-controller. In addition, it has an overshoot increase of $> 370\%$. Hence, this controller does not meet the design specifications on which the PI-controller was based. Lastly, the PI-controller does not present a sharp peak like the H_∞-controller, reducing the effects of fatigue in the physical application.

Disturbance Rejection

Since the H_∞-controller is designed to optimise control performance based on a given cost function, it would be expected that this controller behaves better than the previously designed PI-controller. The step response to input disturbance V, as well as the comparison to the mentioned PI-controller are plotted below and their performance specifications tabulated in Table 4.



(a) (b)

Figure 9: H_∞-response to step disturbance V and comparison of manual PI-controller vs H_∞-controller on output w_r

Parameter	PI-controller	H_∞-controller
TransientTime	213.5	65.1
Peak	0.28	0.04
PeakTime	3.67	0.53

Table 4: Performance of PI- and H_∞-controllers on disturbance rejection of wind V

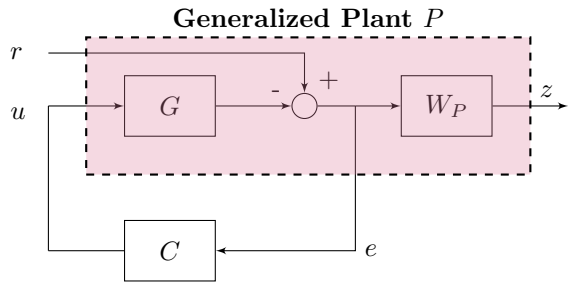
The first plot in Figure 9(a), representing the change in generator speed ω_r, shows minimal peak amplitude, and little oscillations. In addition, the second plot shows that the rejection of the wind disturbance causes the turbine to sway with an amplitude with 20cm for a long time. However, this is insignificant within the margin of 20m. This disturbance rejection is achieved with a combination of a new blade pitch angle, as well as a varying torque, to negate any oscillation in the output. When comparing the disturbance rejection of the PI- en H_∞-controller, it is obvious that the H_∞-controller has significantly better performance. Not only does it act faster, it also has significantly reduced oscillation time and amplitude. These performance specifications can also be seen in Table 4. The transient time has decreased by 70%, and the peak and peak time by 85%. Overall, the H_∞-controller has been designed for optimal performance in all situations, whereas the manual SISO PI-controller was specifically designed for reference tracking due to a step input. This explains the significant enhanced performance in disturbance rejection of the H_∞-controller.

Fixed structure control design

SISO

1. Block diagram

The block diagram for SISO fixed-structure controller design has can be found below. Here, r is the reference signal and z_1 is the performance measure, while u is the input to the plant and y the output. Controller C consists of one transfer function, as it is a SISO plant.



2. Fixed Structure Plant

Using the same plant set-up as described in Section 7 of the previous chapter, as well as the '`infstruct` command in Matlab, a plant was found of the form:

$$P = \begin{bmatrix} W_P & -W_P G \\ 1 & -G \end{bmatrix}$$

Note that the minuses in the second column originate from the fact that reference tracking is considered rather than disturbance rejection. Exact code of the plant synthesis can be found in Appendix C.

3. Controller Synthesis

From the analysis of the optimal $\mathcal{H}_\infty$-controller as in section 7 of the previous chapter, a PI- or PID-controller would be optimal for the SISO case with input β and output ω_r. In Figure 10, the sensitivity functions are plotted against the given performance measure. Neither controller fully stays below $\frac{1}{W_{PSISO}}$, with both controllers having a gain of slightly higher than one at a frequency of around $\omega_r = 0.9$ rad/s. The PID-controller reaches an $\mathcal{H}_\infty$-norm of 1.14, and the PI-controller 1.27. This means that the PID- controller performs better based on the given performance measure. Hence, when purely looking at the given performance measure, choosing a PID-controller would be preferable.

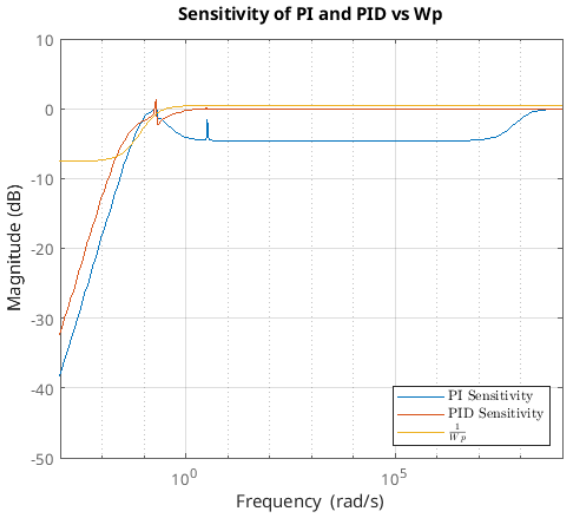
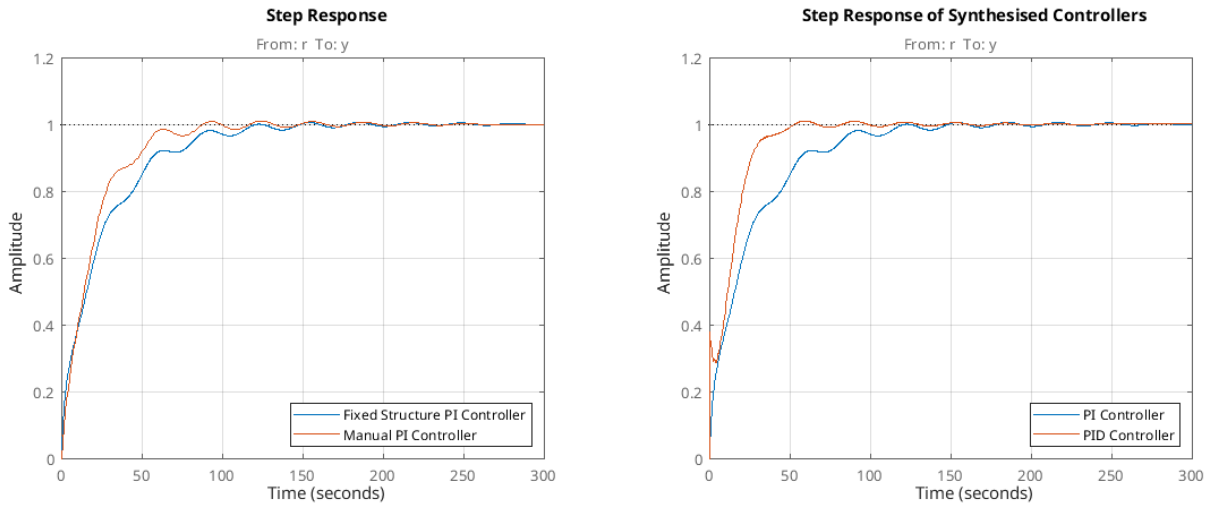


Figure 10: Sensitivity of PI and PID fixed structure SISO controller

4. Controller Performance

The reference tracking of the optimal fixed-structure controller and the manual PI-controller has been plotted below. As can be seen in Figure 11(a), the manually tuned PI-controller has a faster step response than the fixed structure PI-controller. The manually tuned controller has a rise time of 46 seconds and a settling time of 83 seconds, while remaining within 1% overshoot. In contrast, the fixed structure controller has a rise time of 55 seconds with a settling time of 113 seconds. However, the manually tuned PI-controller was designed specifically with specified criteria in mind, while the fixed structure controller is optimized to have a minimal $\mathcal{H}_\infty$-norm. Therefore, it is quite logical that the manually tuned controller performs slightly better in this specific scenario, while the fixed structure controller will likely perform better for other scenarios, such as disturbance rejection. Therefore, in the real world application, the fixed structure PI-controller would be preferable.

Figure 11(b) compares the step responses on the PI- and PID-controller as synthesised using the H_∞-norm. As described in the previous section, a PID-controller would be preferable when only considering the H_∞ as a performance measure. However, when observing the step response of the two controllers, the PID-controller shows a peak previously described in the synthesised SISO H_∞-controller. This makes that in a physical application, a PI-controller would be preferred in order to reduce material fatigue.



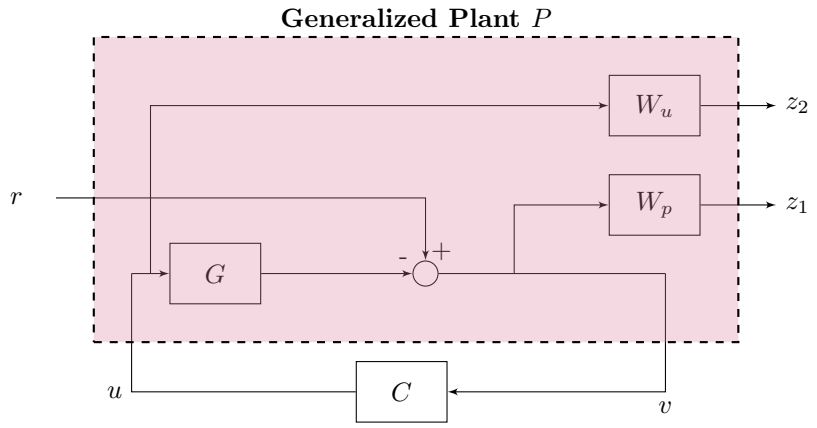
(a) PI, manual and fixed structure

(b) PID, manual and fixed structure

Figure 11: step response of manual and fixed structure PI and PID-controllers

MIMO

1. Fixed Structure Plant



This block diagram is very similar to the generalised plant found earlier for the $\mathcal{H}_\infty$-controller. However, since it is now a reference to follow, rather than a disturbance to reject, the summation between the external input is different. This results in the generalised plant:

$$P = \begin{bmatrix} W_p & -W_pG \\ 0 & W_u \\ I & G \end{bmatrix}$$

2. Choice of Controller

From the analysis of the bode-magnitude plot of the mixed-sensitivity generalised controller, only the first input will be used to control both outputs in this fixed-structure controller, since the $\mathcal{H}_\infty$-controller has close to 0 gain for all frequencies on the second input. Furthermore, a PI-controller optimally steers the first output, while an inverse notch filter ideally steers the second output. This results in the following shape for controller C:

$$C = \begin{bmatrix} K_p + \frac{K_i}{s} & 0 \\ \frac{s^2+2\zeta_z\omega_{n_z}s+\omega_{n_z}^2}{s^2+2\zeta_p\omega_{n_p}s+\omega_{n_p}^2} & 0 \end{bmatrix}$$

with K_p, K_i, ζ_z, ζ_p, ω_{n_z} and ω_{n_p} as tunable parameters.

3. Controller Synthesis

Using `matlab` and in a similar fashion as described before, a fixed structure controller was found with an $\mathcal{H}_\infty$ norm of 3.58. Exact code for the controller synthesis can be found in Appendix C. Figure 12 shows the Bode-magnitude plots of both the $\mathcal{H}_\infty$ PID controller and the fixed-structure PI-controller. The differences between the PID- and PI-controller on the $(1,1)$ channel can be noted first. While the PI-controller has an unbounded DC gain at low frequencies, the PID-controller is bounded as expected. In addition, the PID-controller has increased attenuation at high frequencies, while that of the PI-controller remains constant. The $\mathcal{H}_\infty$ PID-controller also has more complex features, such as a spike around 5 rad/s, which are not captured by the fixed structure controller, in order to reduce the complexity of the controller. The notch filter on the second output in the fixed structure controller has its peak at the same frequency as the $\mathcal{H}_\infty$ controller. While this is good, the optimal controller has more complex dynamics and better attenuation at high frequencies. As discussed before, the second input is not used for control.

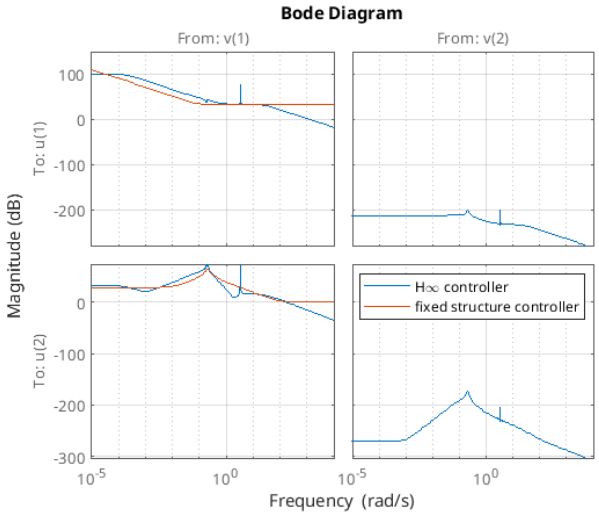
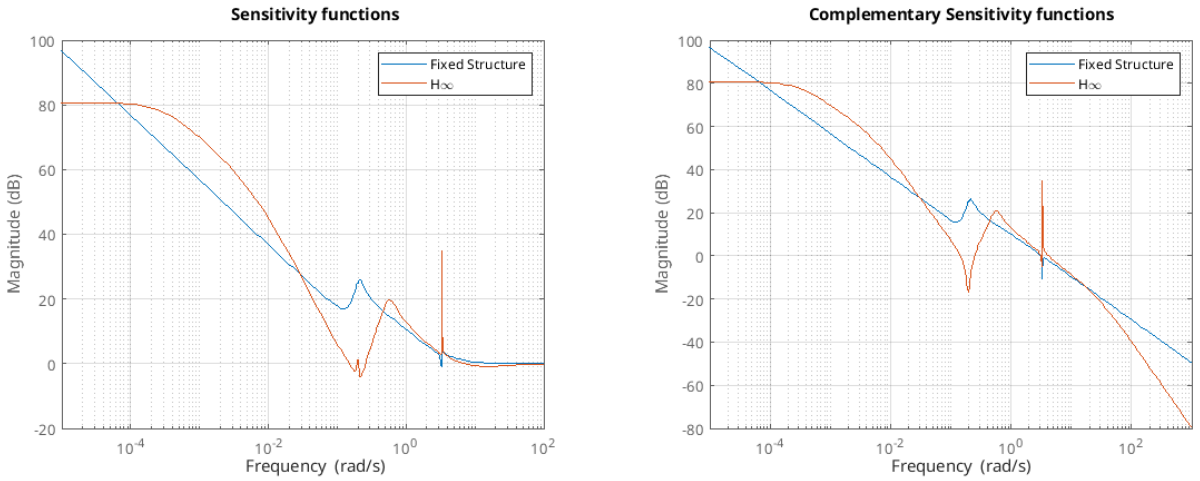


Figure 12: Bode magnitude plot of the $\mathcal{H}_\infty$ controller and fixed structure MIMO controller

4. Sensitivity and Complementary Sensitivity

The (complementary) sensitivity functions may be analysed to compare the sensitivity of the systems to disturbances, and the ability to track a reference. Figure 13a below shows the sensitivity function of the $\mathcal{H}_\infty$ optimal controller and the fixed structure controller. What can be seen from this plot is that both controllers are insensitive to disturbances at very high frequencies. However, the optimal H_∞ controller is also has a notch and thus attenuates disturbances of frequencies around 0.1 rad/s, while the fixed structure controller has increased sensitivity to disturbances at this frequency. Again, it should be noted that the fixed structure controller has an unbounded DC gain for low frequencies, whereas the $\mathcal{H}_\infty$ controller has a constant DC gain of 80 dB.

Figure 13b shows that the optimal controller has better control using inputs with frequencies of around 10^{-4} to 2×10^{-2} rad/s, as well as at higher frequencies, which are more attenuation. On the other hand, the fixed structure controller has a seemingly constant slope of 20dB per decade, with only a peak at around 2×10^{-4} rad/s. Again, the peak around 0.1 rad/s suggests poor control at this frequency relative to the H_∞ controller.



(a) Bode magnitude plot of Sensitivity

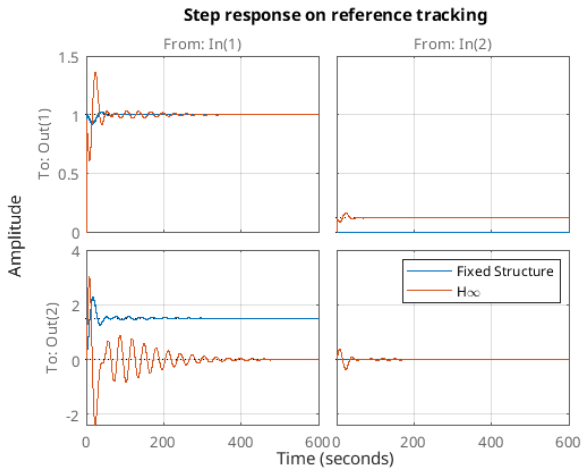
(b) Bode magnitude plot of Complementary Sensitivity

Figure 13: Bode magnitude plots of sensitivity and complementary sensitivity functions of $\mathcal{H}_\infty$ Controller and Fixed Structure Controller

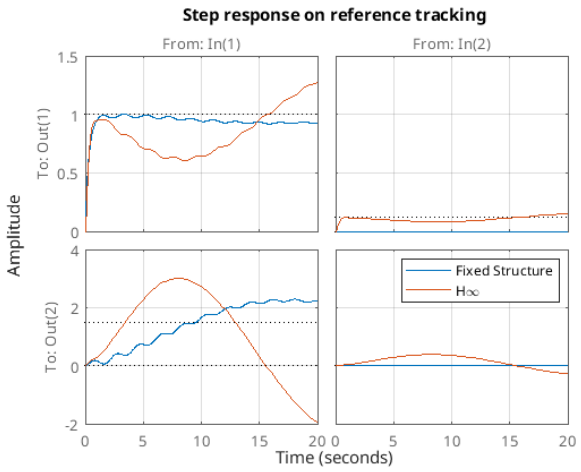
5. Controller Performance

Reference Tracking

Figure 14a below shows the step response of both controllers. As can be seen on the left, the fixed structure controller has a steady-state error in the second output channel when there is a step in the reference for the first channel, corresponding to slight lean in the tower. However, in the second channel the $\mathcal{H}_\infty$ controller has a larger and longer oscillation. In the performance measure these are constrained to 20 meters, since the oscillations are around 1 meter and the steady state displacement is around $1.8m$, this criterium is met. Looking at the first channel, it can be noted the $\mathcal{H}_\infty$ causes a larger initial oscillation which dies out sooner than the very slow oscillation caused by the fixed structure controller.



(a) step response of $\mathcal{H}_\infty$ controller and fixed structure



(b) transient of step response of $\mathcal{H}_\infty$ controller and fixed structure

Disturbance Rejection

Figure 15 shows the response of the $\mathcal{H}_\infty$ controller and fixed structure controller to a step in the wind input.The first plot shows the deviation of the rotational speed from the set point. As can be seen from the graph, the $\mathcal{H}_\infty$ rejects this disturbance very effectively, going back to very near the set point rapidly with high-frequency, low amplitude oscillations, while the fixed structure controller takes longer to respond, with a larger oscillation.

From the deviation in the second output it can be seen both controllers cause the turbine to sway for a long time after a change in wind speed. Although these oscillations are quite small, namely of only around 20 cm, the tower does find a new equilibrium and as a result leans 20 cm further. It can be noted that while the initial oscillations using the fixed structure controller are larger, they also decay faster.

The bottom two plots show the control inputs used to reject the disturbance. While the response is quite similar, it can be seen the $\mathcal{H}_\infty$ controller relies more on the blade pitch β while the fixed structure controller has a larger control input magnitude in the generator torque τ_e.

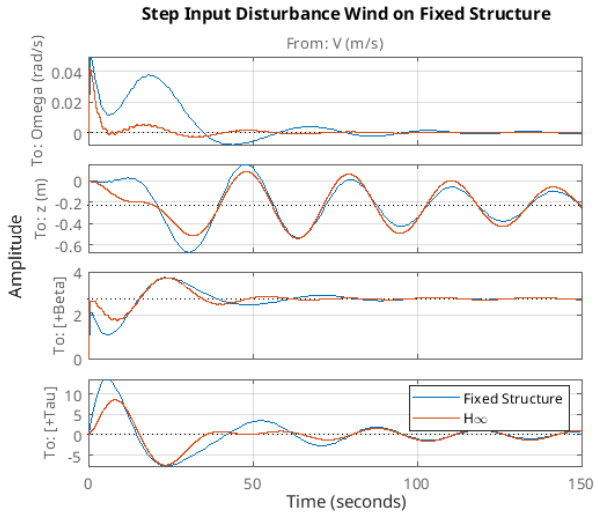


Figure 15: Wind disturbance rejection of $\mathcal{H}_\infty$ and Fixed structure controllers

6. $\mathcal{H}_\infty$ vs. Fixed-structure Controller

The $\mathcal{H}_\infty$ controller obviously has a lower $\mathcal{H}_\infty$ norm. However, it achieves this by making a very complicated controller, which might be difficult to realise in real life, where the control inputs need to be realised by real life actuators rather than a simulated value which is not constrained by inertia for instance. Furthermore, in a step response situation, while the $\mathcal{H}_\infty$ has a response with a lower settling time, the fixed structure controller oscillations of the blade speed are much smaller, again meaning less acceleration and thus wear and tear on the turbine blade. It is desirable to have the blade follow the reference well, so that it delivers a constant amount of power to the grid. In real life, a turbine has more components to ensure the delivered power is compatible with the grid, providing slightly more flexibility in the delivered output.

In conclusion, the realisability of the fixed structure controller and the abruptness of the $\mathcal{H}_\infty$ step response due to aggressive control make it preferable to use a fixed-structure controller over the $\mathcal{H}_\infty$ controller for a real-world application.

A Mixed-Sensitivity Controller Synthesis

```
% Define plant and in and outputs
MIMO_G = FWT(1:2, 1:2);
MIMO_G.u = 'u';
MIMO_G.y = 'y';

wp = [(s/3+0.3*2*pi)/(s+pi*6e-5) 0;
        0 0.05];
wp.u = 'v';
wp.y = 'z1';

wu = [0.005 0;
 0 (0.005*s^2 + 0.0007*s+0.00005)/(s^2 + 0.0014*s + 10^(-6))];
wu.u = 'u';
wu.y = 'z2';

% Build system
sumblock = sumblk("v = w + y", 2);

P = connect(MIMO_G, wu, wp, sumblock, {'w', 'u'}, {'z1', 'z2', 'v'});

% H infinity synthesis
nmeas = 2; %number of outputs plant
ncont = 2; %number of inputs controller
[MS_C,CLms,gamma] = hinfsyn(P,nmeas,ncont);
```

B Fixed-structure SISO controller

```
Gsiso = FWT(1, 1);

% Fixed Structure SISO controller
s = tf('s');
Kp = realp('Kp',1);
Ki = realp('Ki',1);
Kd = realp('Kd',1);
Tf = realp('Tf',1);

Wp_simple = 0.95*(s+0.02*2*pi)/(0.016*pi+s);
C_struct = Kp+Ki/s+(Kd*s)/(Tf*s+1);

% SISO hinfstruct
Wp_siso = Wp_simple;
Wp_siso.u = 'e'; % On output or on error???
Wp_siso.y = 'z1';

G_siso = -Gsiso;
G_siso.u = 'u';
G_siso.y = 'y';

C_siso = C_struct;
C_siso.u = 'e';
C_siso.y = 'u';

Sum1 = sumblk('e = r - y');
Siso_Con = connect(G_siso, Wp_siso, C_siso, Sum1, 'r', 'z1');

opt = hinfstructOptions ('Display', 'final', 'RandomStart', 5);
[N_siso, GAM] = hinfstruct(Siso_Con, opt);

% Extract controller gains :
Kp_opt = N_siso.Blocks.Kp.Value;

Ki_opt = N_siso.Blocks.Ki.Value;
Kd_opt = N_siso.Blocks.Kd.Value;
Tf_opt = N_siso.Blocks.Tf.Value;

% Constructing controllers with optimal parameters
Kfb_opt_pi = Kp_opt + Ki_opt / s;
Kfb_opt_pid = Kp_opt + Ki_opt / s +( Kd_opt * s ) /( Tf_opt * s +1);

Kfb_opt_pi.u = 'e';
Kfb_opt_pi.y = 'u';

Kfb_opt_pid.u = 'e';
Kfb_opt_pid.y = 'u';

S_pi = 1/ (1 + series(G_siso, Kfb_opt_pi));
S_pid = 1/ (1 + series(G_siso, Kfb_opt_pid));
```

C Fixed-structure MIMO controller

```
Gmimo = FWT(1:2, 1:2);

Kp1 = realp('Kp1',1);
Ki1 = realp('Ki1',1);

% Define notch filter parameters
wn_z = realp('wn_z', 6);
zeta_z = realp('zeta_z', 0.3);
wn_p = realp('wn_p', 2);
zeta_p = realp('zeta_p', 0.1);

wp = [(s/3+0.3*2*pi)/(s+pi*6e-5) 0;
    0 0.05];
wu = [0.005 0;
      0 (0.005*s^2 + 0.0007*s+0.00005)/(s^2 + 0.0014*s + 10e-6)];

C1 = Kp1+Ki1/s;
Notch2 = (s^2 + 2*zeta_z*wn_z*s + wn_z^2) / (s^2 + 2*zeta_p*wn_p*s + wn_p^2);

Wp_mimo = wp;
Wp_mimo.u = 'e';
Wp_mimo.y = 'z1';

Wu_mimo = wu;
Wu_mimo.u = 'u';
Wu_mimo.y = 'z2';

G_mimo = -Gmimo;
G_mimo.u = 'u';
G_mimo.y = 'y';

C_mimo = [C1 0; Notch2 0];
C_mimo.u = 'e';
C_mimo.y = 'u';

Sum1 = sumblk('e = r - y',2);
Mimo_Con = connect(G_mimo, Wp_mimo, Wu_mimo, C_mimo, Sum1, 'r', {'z1', 'z2', 'y'});

opt = hinfstructOptions ('Display', 'final', 'RandomStart', 50);
N_mimo = hinfstruct(Mimo_Con, opt);

% Extract controller gains :
Kp1_opt = N_mimo.Blocks.Kp1.Value;
Ki1_opt = N_mimo.Blocks.Ki1.Value;

wn_z_opt = N_mimo.Blocks.wn_z.Value;
zeta_z_opt = N_mimo.Blocks.zeta_z.Value;
wn_p_opt = N_mimo.Blocks.wn_p.Value;
zeta_p_opt = N_mimo.Blocks.zeta_p.Value;

Kfb1_opt = Kp1_opt + Ki1_opt / s;
notch_opt = (s^2 + 2*zeta_z_opt*wn_z_opt*s + wn_z_opt^2) / (s^2 + 2*zeta_p_opt*wn_p_opt*s + wn_p_opt^

Kfb_opt = [Kfb1_opt 0; notch_opt 0];

Kfb_opt.u = 'e';
Kfb_opt.y = 'u';

CLsysr = connect(G_mimo, Kfb_opt, Sum1, 'r', 'y');
```