University of Liège

Study of an active mass damper

Linear control systems

Bastien Hoffmann (20161283)
Maxime Meurisse (20161278)
Valentin Vermeylen (20162864)

Master in Civil Engineering

Academic year 2019-2020

Contents

1 Control problem

1.1 Choice of the topic

The chosen topic is : **Active mass damper**.

1.2 Context

The current engineering prowesses allow us to construct buildings higher and higher. These constructions are subject to various disturbances (mainly wind, but also earthquakes) that make them oscillate. They turn into giant pendulum and swing from left to right, sometimes moving several meters at the top ![1]

To reduce these oscillations, we use a passive system, called *tuned mass damper*, which consists of concealing a tuned and harmonic oscillator at the top of the tower. It is coupled to its movement and oscillates in phase opposition to recover the kinetic energy of the tower and thus reduces the oscillations.[2]

An active version of this system exists : the *active mass damper*. It consists of the same principle as the tuned mass damper but it is equipped with sensors and actuators to measure the oscillations of its environment and, via an algorithm, generate a movement for the mass that reduce, or totally remove, these oscillations.[3]

Our study field focuses on the active mass damper systems used to reduce the oscillations caused by the **wind** on **tall** buildings.

1.3 Control problem diagram

The diagram of our control problem is shown in figure 1.

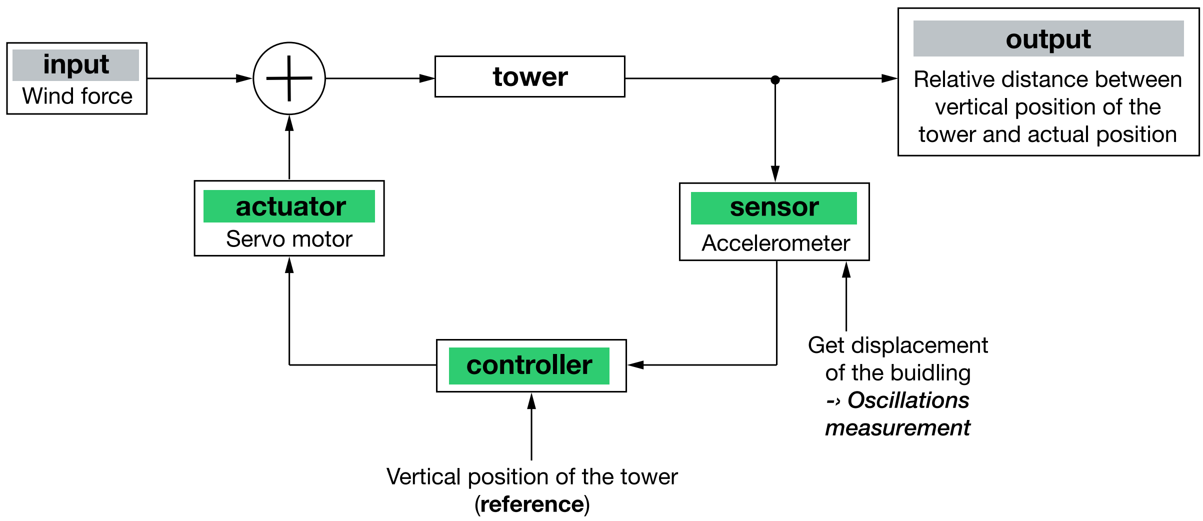


Figure 1 – Control problem diagram of the active mass damper for tall buildings

1.4 Control problem description

- **Utility of the controller** : the controller (the algorithm) allows the system (the tower) to be active, *i.e.* to measure the oscillations to which it is subjected and to cancel it. Thanks to a servo-motor connected to the controller, the mass can move and reduce, or even eliminate totally, the oscillations.
- **System to be controlled** : the tower (and the position of the tower is the signal)
- **Inputs of the system** : wind force acting on the tower (uncontrollable) and force acting on the mass damper (controllable).
- **Outputs of the system** : the relative distance between the vertical position and the displacement of the tower.
- **Reference** : the vertical position of the tower.
- **Actuators** : servo-motor to move the mass that reduces the oscillations.
- **Constraints and limitations** : to simplify our system, we consider a tower 200 m high, perfectly vertical when it undergoes no disturbance. The only disturbance on this tower is the strength of the wind. The wind, ranging from a few tens of km/h to a hundred km/h, can swing the tower from a few centimetres to several meters.

1.5 Open loop system diagram

The detailed schematic of the open loop studied system is shown in figure 2.

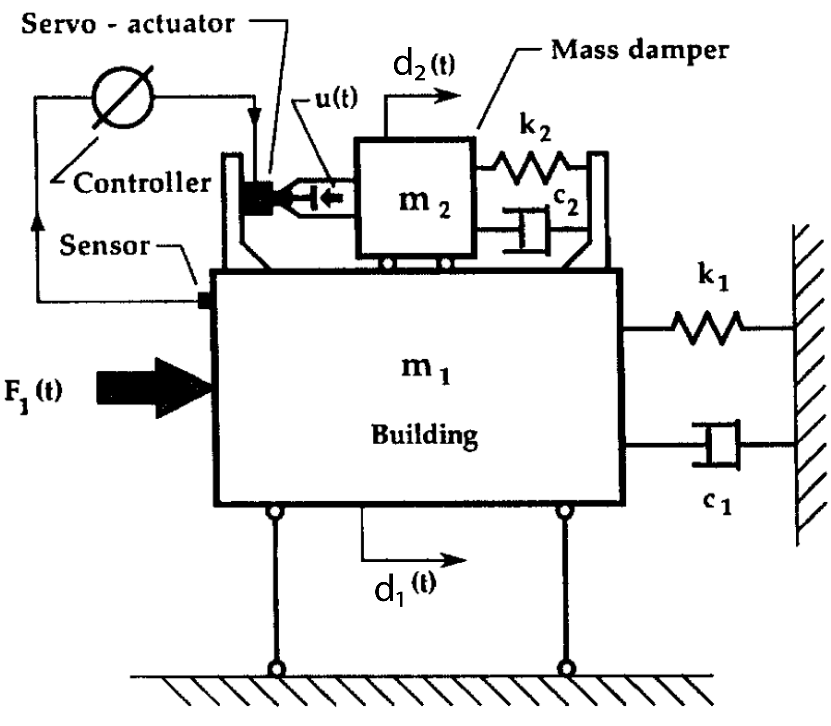


Figure 2 – Detailed schematic of the open loop studied system [3]

The building is represented by the mass m_1 and its oscillation motion is simulated by the spring k_1 and the damper c_1.

The mass damper is represented by the mass m_2 and its movement is simulated by the spring k_2 and the damper c_2.

The force $F_1(t)$ represents the wind force (uncontrollable) on the building.

The force $u(t)$ represents the force applied on the mass damper by the controller (controllable).

We are studying, at first, our system without a control mechanism. Our controllable input $u(t)$ will therefore be 0 for all our simulations in this section.

2 State-space representation

2.1 Open loop model description

Input vector U and state vector X are given by :

$$U = \begin{pmatrix} F_1 \\ u \end{pmatrix} \qquad\qquad X = \begin{pmatrix} d_1 \\ \dot{d}_1 \\ d_2 \\ \dot{d}_2 \end{pmatrix}$$

2.1.1 Inputs

- $F_1(t)$, the force of the wind (uncontrollable), approximately between 1000 and 2000 kN.
- $u(t)$, the force applied on the mass damper (controllable), approximately between 1000 and 2000 kN.

Our sensor is a measurement of the horizontal position of the top of the building relatively to the vertical position $d_1 = 0$.

Our actuator provides a force on the mass of the dampener, sets it in motion.

2.1.2 Outputs

$y = d_1(t)$ the relative position of the building with respect to the vertical position.

2.1.3 States

- $x_1 = d_1$, as described above, ranging from a few millimeters to a few meters.
- $x_2 = \dot{d}_1$, the speed of the building, ranging from about 0.1 to $5\,\mathrm{m\,s^{-1}}$.
- $x_3 = d_2$, the relative displacement of the mass damper, ranging from a few millimeters to a few meters.
- $x_4 = \dot{d}_2$, the speed of the mass damper, ranging from about 0.1 to $5\,\mathrm{m\,s^{-1}}$.

2.1.4 Output law

The output is one of the states : $y = x_1$.

2.1.5 Input law

The input law is given by [3] :

$$\begin{cases} m_1\ddot{d}_1 + c_1\dot{d}_1 + k_1 d_1 = c_2\dot{z} + k_2 z + F_1(t) - u(t) \\ m_2\ddot{z} + c_2\dot{z} + k_2 z = -m_2\ddot{d}_1 + u(t) \end{cases}$$

with $z = d_2 - d_1$.

2.2 State-space model

The system is **linear**. We can easily derive the ABCD matrices.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_1-k_2}{m_1} & \frac{-c_2-c_1}{m_1} & \frac{k_2}{m_1} & \frac{c_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} & \frac{-k_2}{m_2} & \frac{-c_2}{m_2} \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ \frac{1}{m_1} & -\frac{1}{m_1} \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \end{pmatrix}$$

2.3 Constraints, limitations and numerical choice of parameter values

To model and study the system, we defined a series of constraints, assumptions and limitations, presented in table 1.

Building	height of 200 m, width of 30 m movement along a single axis (horizontal)
Mass	no friction between m_1 and m_2

Table 1 – Constraints, assumptions and limitations of the system.

To simulate the system (without control mechanism), we choose a series of numerical values, presented in table 2[1].

Mass	$m_1 = 1 \times 10^7$ kg	$m_2 = 3 \times 10^3$ kg
Spring	$k_1 \approx 4 \times 10^8$ N/m	$k_2 = 10^5$ N/m
Damper	$c_1 \approx 1.3 \times 10^6$ Ns/m	$c_2 = 10^4$ Ns/m
Wind	$F_{max} = 810\,000$ N	

Table 2 – Numerical values of the system

[1]We would like to thank Professor Denoël for discussing these values with us.

For the strength of the wind, we considered 2 cases (in newton) :

$$F_1 = F_{max} \quad \forall t \qquad\qquad \text{Constant wind force}$$
$$F_1(t) = F_{max}\sin(2\pi t) \qquad\qquad \text{Sinusoidal wind force}$$

The stiffness and viscosity values for the building were obtained using the formulas :

$$k_1 = (2\pi f)^2 m_1$$
$$c_1 = 2m_1(2\pi f)0.01$$

where $f = 1\,\text{Hz}$ is the natural frequency associated with the mass of the building.

The maximum wind force, on the other hand, was approximated by

$$F_{max} = \frac{1}{2}\rho v^2 A$$

with

- $\rho \approx 1.2\,\text{kg/m}^3$, the air density;
- $v = 15\,\text{m/s}$, the wind speed;
- $A = 200 \times 30 = 6000\,\text{m}^2$, the area of one side of the building.

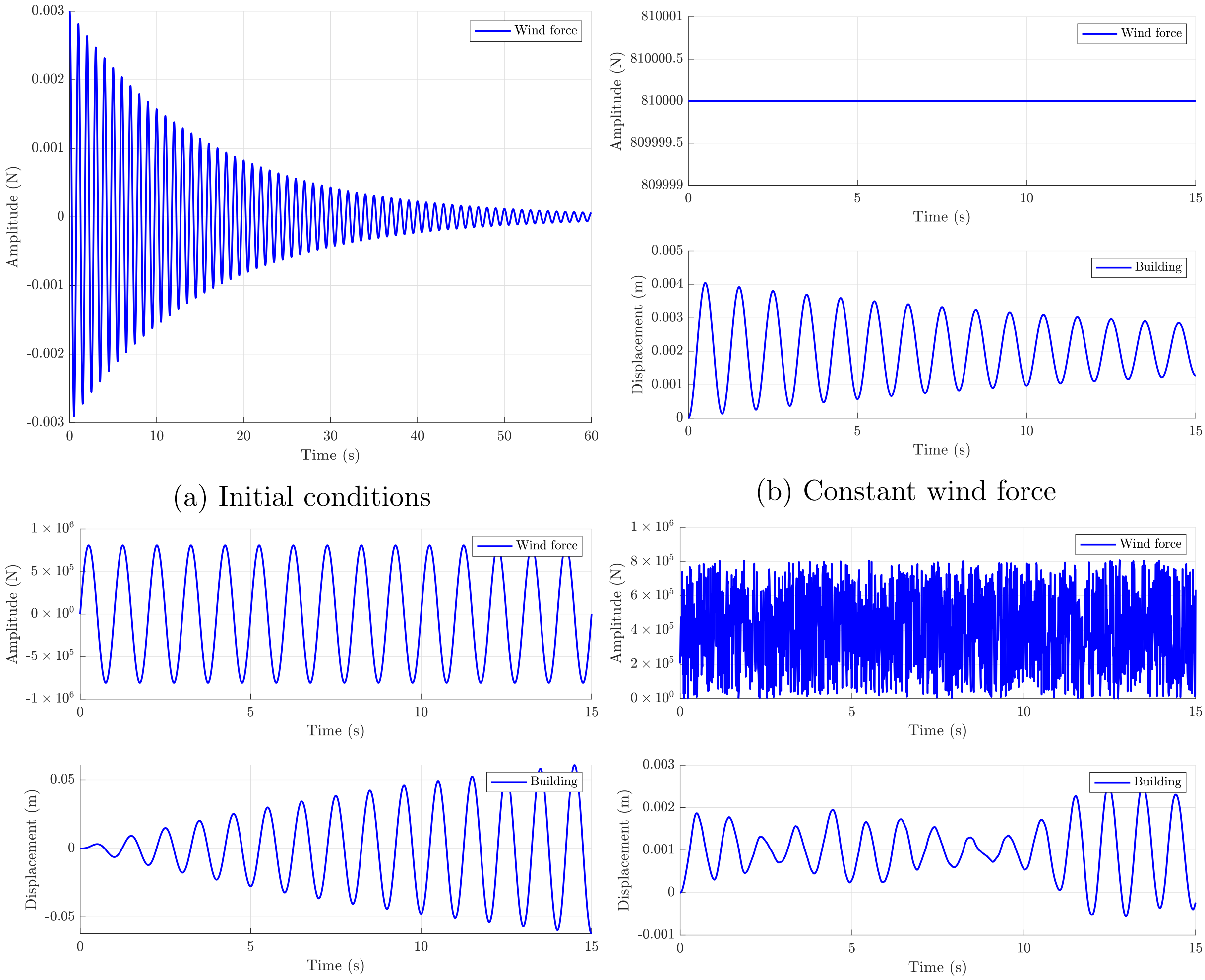
2.4 Stability and eigenvalues

To study the stability of the system, we compute the eigenvalues of the dynamic matrix A thanks to Matlab function (`eig`) :

$$\lambda_1 = -0.0645 + 6.2824\text{i}$$
$$\lambda_2 = -0.0645 - 6.2824\text{i}$$
$$\lambda_3 = -1.6655 + 5.5285\text{i}$$
$$\lambda_4 = -1.6655 - 5.5285\text{i}$$

The system is stable if the real parts of the eigenvalue are all negative. In our case, the system is stable.

2.5 Open loop system simulations



(a) Initial conditions

(b) Constant wind force

(c) Sinusoidal wind force

(d) Random wind force

Figure 3 – Simulation results

The first simulation (figure 3a) is a response of our system to initial conditions : the initial displacement of the building is defined at 0.5 m. We observe that the building oscillates and tends to regain its reference position.

The other simulations are responses of our system to an input (the wind).

In the case of a constant force (figure 3b), the building oscillates at the beginning and then tends to stabilize (at a position different from its reference).

In cases of sinusoidal and random forces, the building oscillates and follows approximately the wind movement.

2.6 Observability

To determine whether or not the system is observable, we compute the observability matrix thanks to Matlab function (`obsv`).

The matrix is full rank (verified with Matlab), the system is thus fully observable.

As seen on the matrix C, we need one sensor. According to the place of the non zero value, this sensor has to measure the x_1 state, namely the horizontal position of the top of the building d_1. This state is indeed the objective of the active mass damper and has thus to be observed.

2.7 Controllability

To determine whether or not the system is controllable, we compute the controllable matrix thanks to Matlab function (`ctrb`). In order not to take into account the uncontrollable input (wind), only the second column of the B matrix was kept for the calculation.

The matrix is full rank (verified with Matlab), the system is thus fully controllable.

As seen on matrix B, we need only one actuator. The first column of the B matrix represents the wind, while the second one concerns the damper. This latter is indeed the only controllable input and contains two non-zero elements. As a result, only one actuator is needed, and acts on two states, the speed of the building and the speed of the damper, as they take place on x_2 and x_4.

3 Controller in time domain

3.1 State feedback controller

As the reference is 0, we need not care about k_r, so we can fix it to 0.

However, if the reference was to change, we could compute k_r, it would be nice. Some tests of a change in reference will be performed in this report.

In a first time, we only need to compute the gain matrix K.

In order not to apply a gain on the wind force, our matrix K is as follows :

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 \\ g_1 & g_2 & g_3 & g_4 \end{pmatrix}$$

The new dynamic matrix of the closed-loop system is $A_{CL} = A - BK$. Let's determine the eigenvalues of that matrix.

As we have a matrix of dimension 4, we will make the approximation of the dominant poles. Indeed, we have, from the previous matrix A, the eigenvalues :

$$\begin{aligned} \lambda_1 &= -0.0645 + 6.2824\mathrm{i} \\ \lambda_2 &= -0.0645 - 6.2824\mathrm{i} \\ \lambda_3 &= -1.6655 + 5.5285\mathrm{i} \\ \lambda_4 &= -1.6655 - 5.5285\mathrm{i} \end{aligned}$$

We can see that λ_3 and λ_4 are about 100 times bigger than the last two, and so we do not need to work on them. Those two will therefore remain in A_{CL}.

Imposing that $(s-\lambda_3)(s-\lambda_4)$ is part of the decomposition, we get that the determinant of A_{CL} is equal to :

$$(s-\lambda_3)(s-\lambda_4)(s^2+2\xi\omega_c s+\omega_c^2)=0$$

Since λ_3 and λ_4 are fixed, we only need to solve the equation of the second degree in s in order to find the expressions of λ_1 and λ_2 as a function of ξ and ω_c.

The solutions of the equation are given by :

$$\begin{cases} \lambda_1 = -\xi\omega_c - \omega_c\sqrt{\xi^2-1} \\ \lambda_2 = -\xi\omega_c + \omega_c\sqrt{\xi^2-1} \end{cases}$$

The values of ξ and ω_c will be determined by simulations in the following sections. When these have been fixed, we will obtain the values of the 4 poles of A_{CL}. Then we will just have to use the `place` function of Matlab to obtain the values g_i of matrix K associated with the eigenvalues.

3.2 Observer

We need to compute the gain matrix L :

$$L = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{pmatrix}$$

The new dynamic matrix is given by $A_{obs} = A - LC$.

As previously, we will keep the same two dominant eigenvalues and determine the two other via the same method we have used for K.

Imposing that $(s-\lambda_3)(s-\lambda_4)$ is part of the decomposition, we get that the determinant of A_{obs} is equal to :

$$(s-\lambda_3)(s-\lambda_4)(s^2+2\xi\omega_c s+\omega_c^2)=0$$

Since λ_3 and λ_4 are fixed, we only need to solve the equation of the second degree in s in order to find the expressions of λ_1 and λ_2 as a function of ξ and ω_c.

The solutions of the equation are given by :

$$\begin{cases} \lambda_1 = -\xi\omega_c - \omega_c\sqrt{\xi^2-1} \\ \lambda_2 = -\xi\omega_c + \omega_c\sqrt{\xi^2-1} \end{cases}$$

The poles of the observer are determined by taking the poles of the controller and moving them. To do this, the real parts of each pole are multiplied by a constant α. In the case of poles λ_1 and λ_2, this amounts to multiplying w_c by α.

We finally have :

$$\begin{cases} \lambda_1 = -\xi\omega_c\alpha - \omega_c\alpha\sqrt{\xi^2-1} \\ \lambda_2 = -\xi\omega_c\alpha + \omega_c\alpha\sqrt{\xi^2-1} \\ \lambda_3 = \mathbb{R}(\lambda_3)\alpha + \mathbb{I}(\lambda_3)i \\ \lambda_4 = \mathbb{R}(\lambda_4)\alpha + \mathbb{I}(\lambda_4)i \end{cases}$$

We then obtain the values l_i of the matrix L by using the `place` function of Matlab.

3.3 Simulations and discussion

Through several tests, we have determined the following values to obtain acceptable results :

$$\begin{cases} \xi = 0.8 \\ \omega_c = 5 \\ \alpha = 5 \end{cases}$$

3.3.1 Response to a reference variation

The new reference has been set at 0.002 m for simulations.

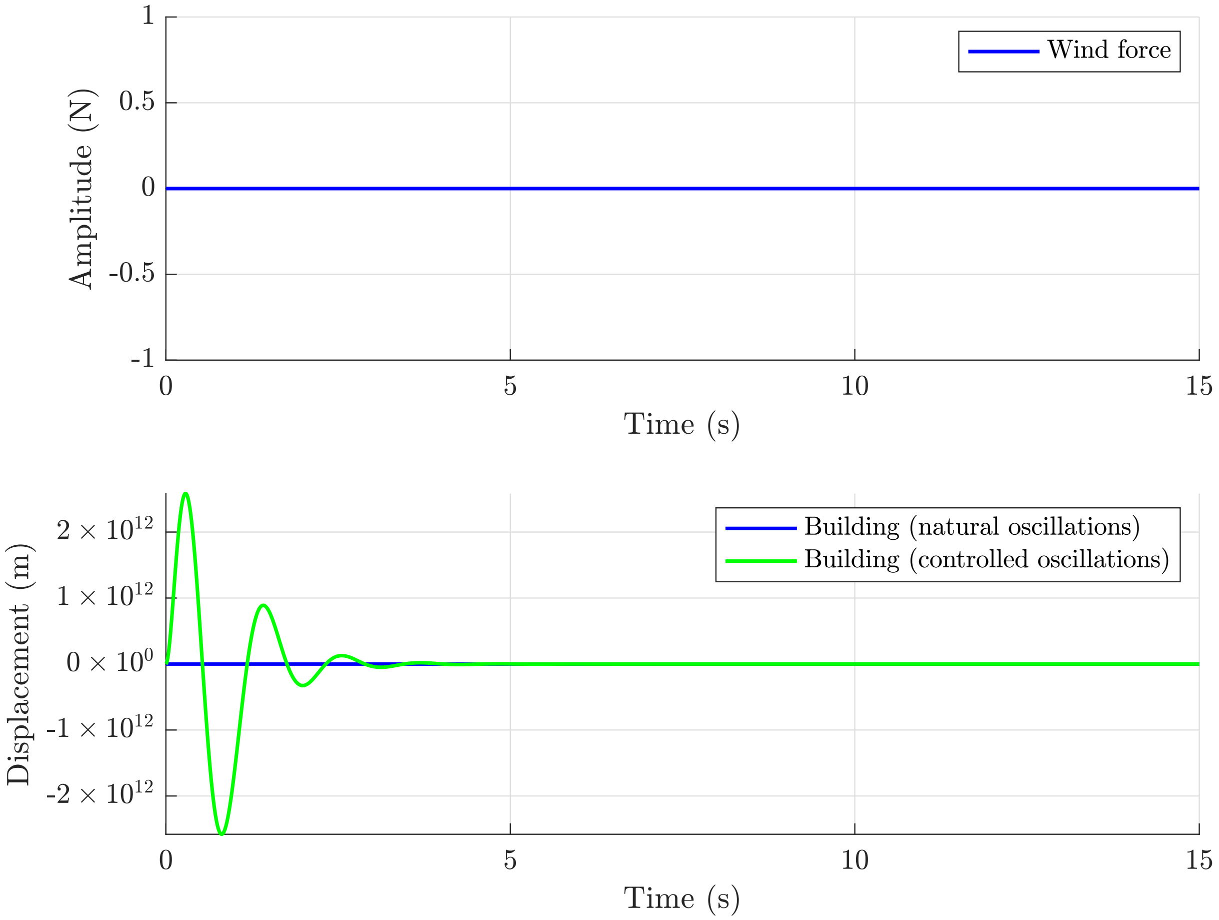


Figure 4 – Response to a reference variation - controlled output

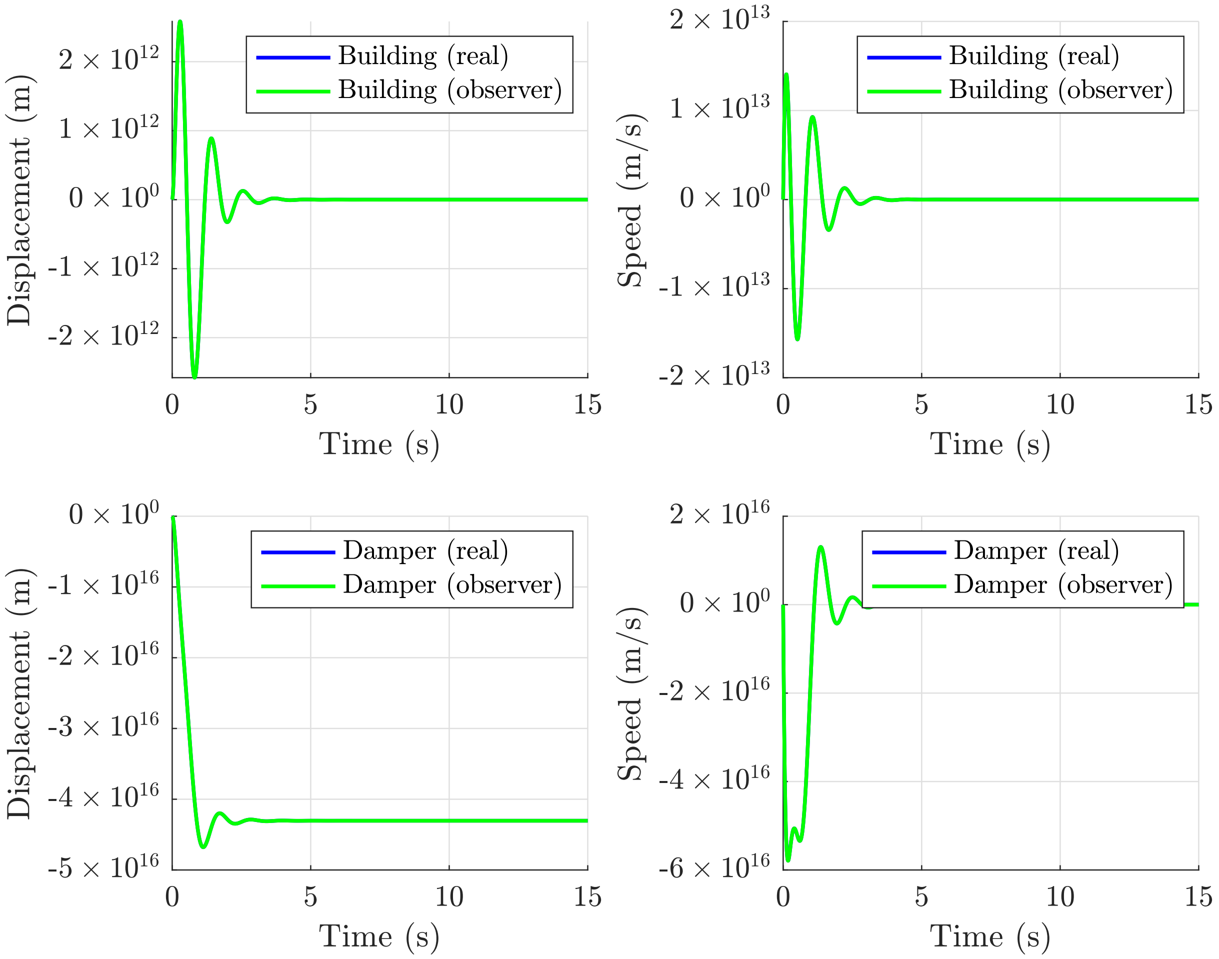


Figure 5 – Response to a reference variation - states

3.3.2 Response to a perturbation (disturbance)

For a constant wind force, we get :

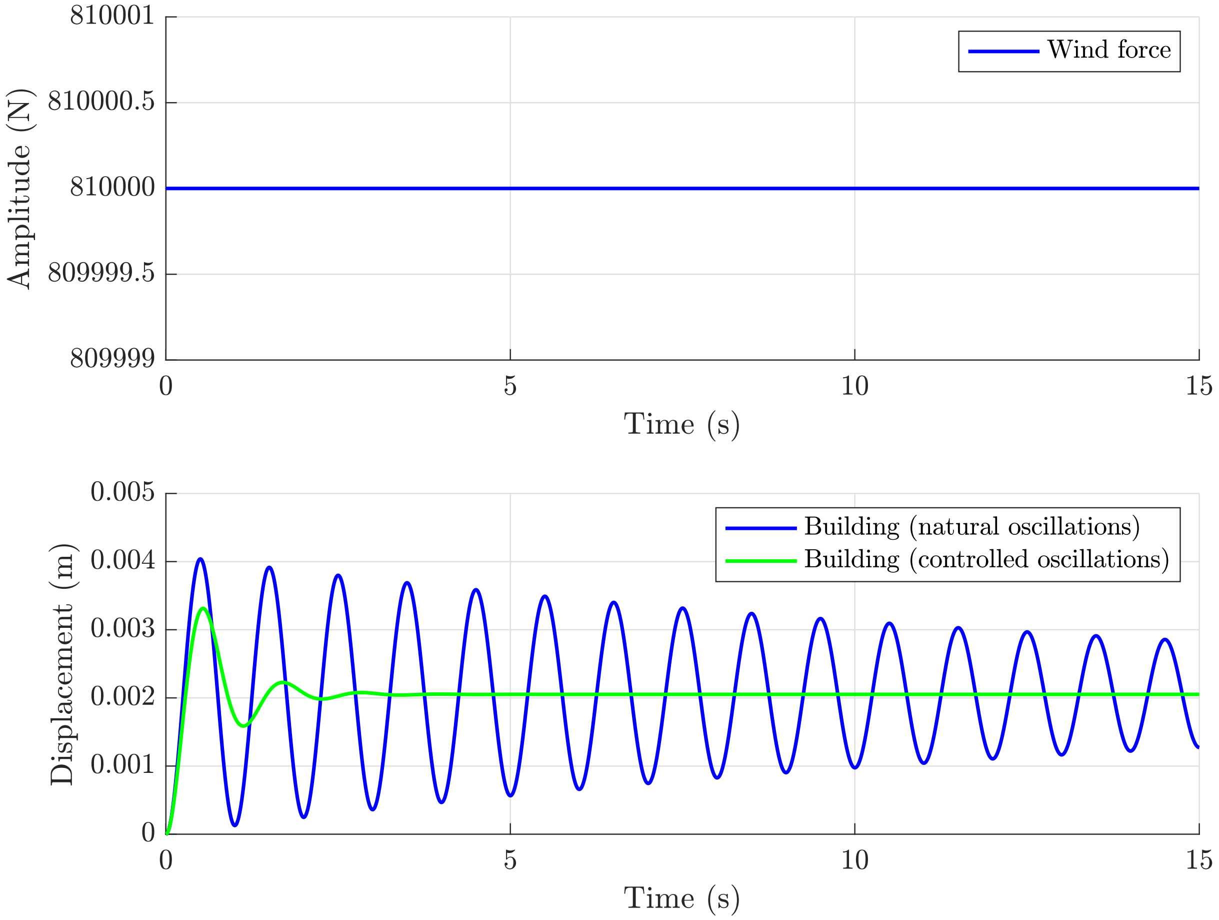


Figure 6 – Response to a constant wind force - controlled output

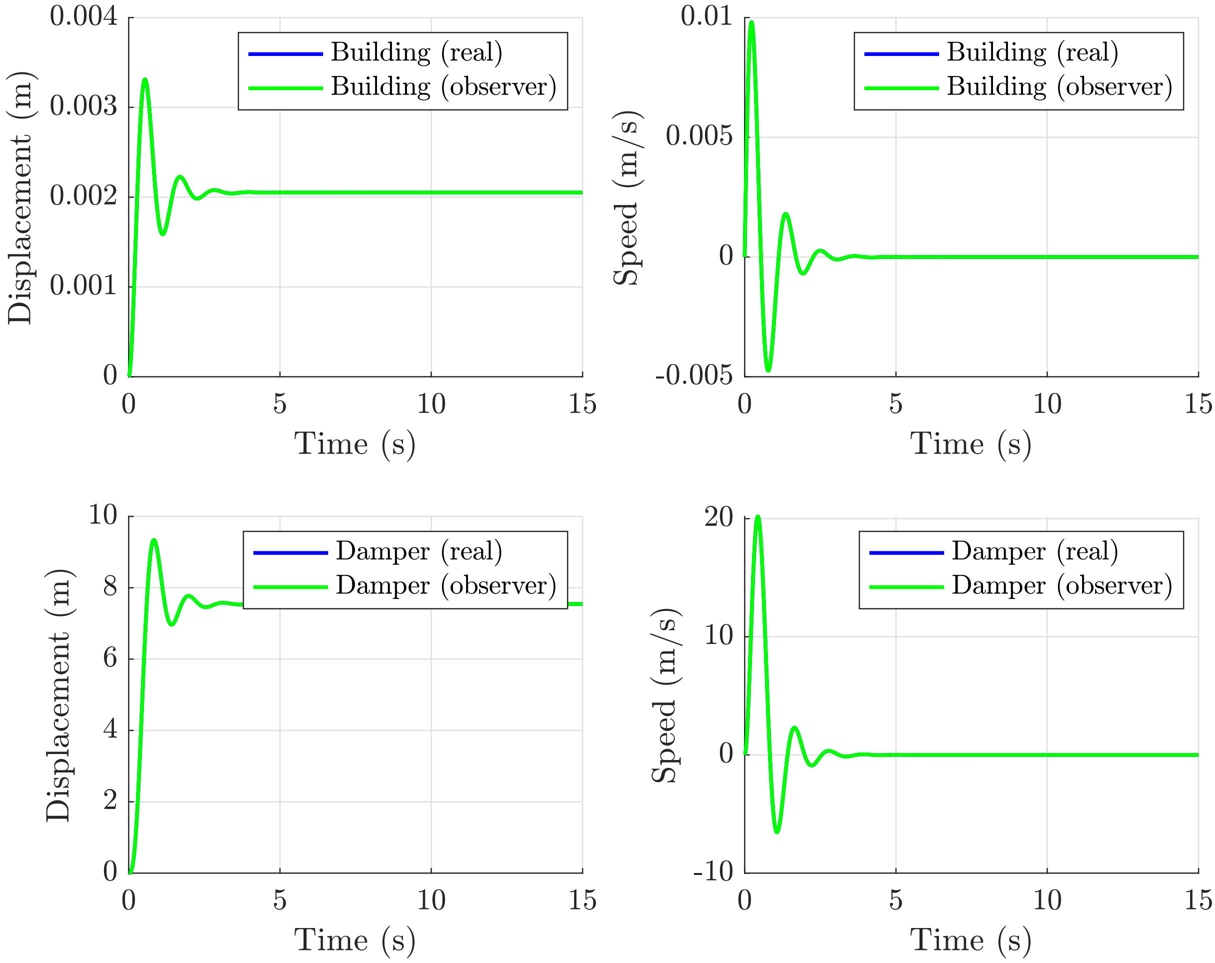


Figure 7 – Response to a constant wind force - states

For a sinusoidal wind force, we get :

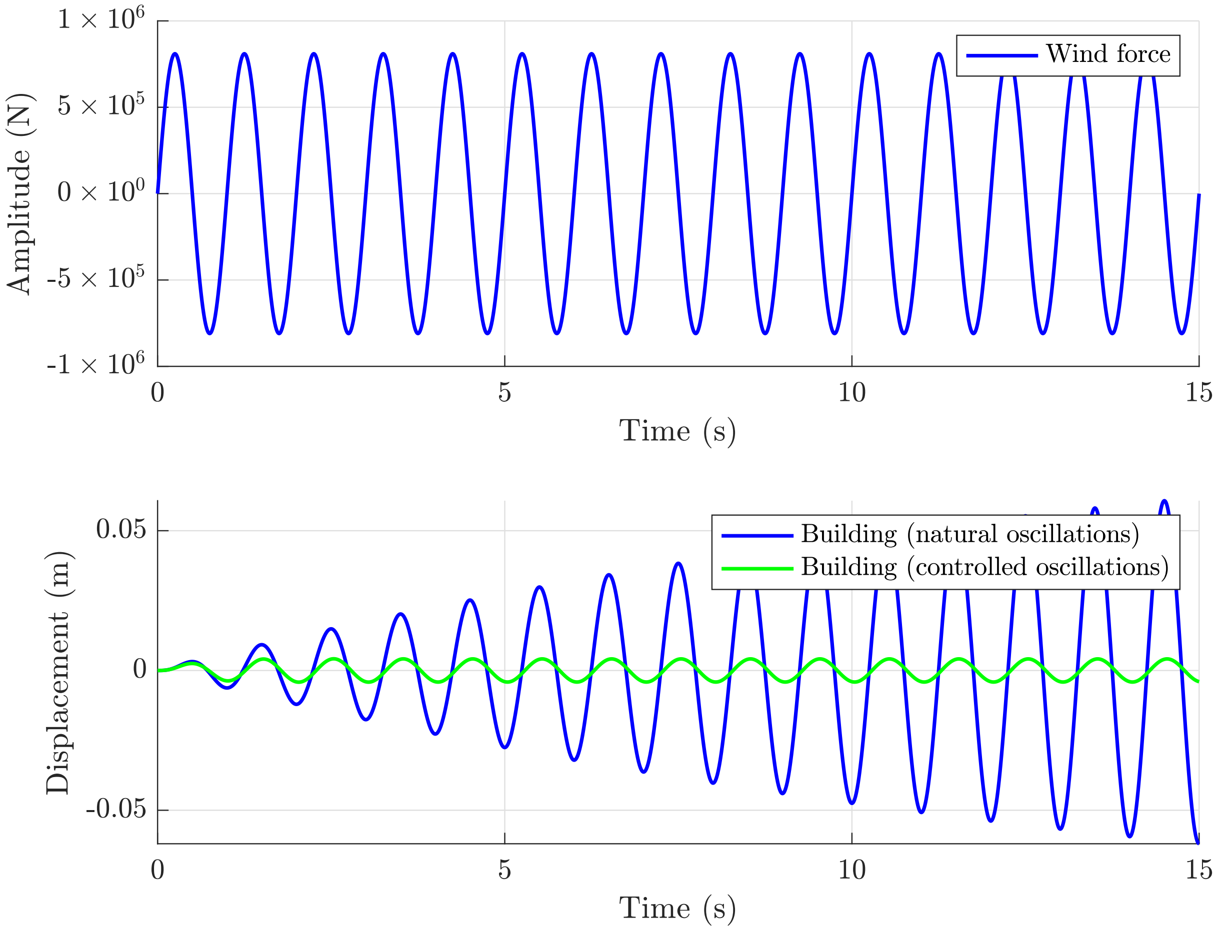


Figure 8 – Response to a sinusoidal wind force - controlled output

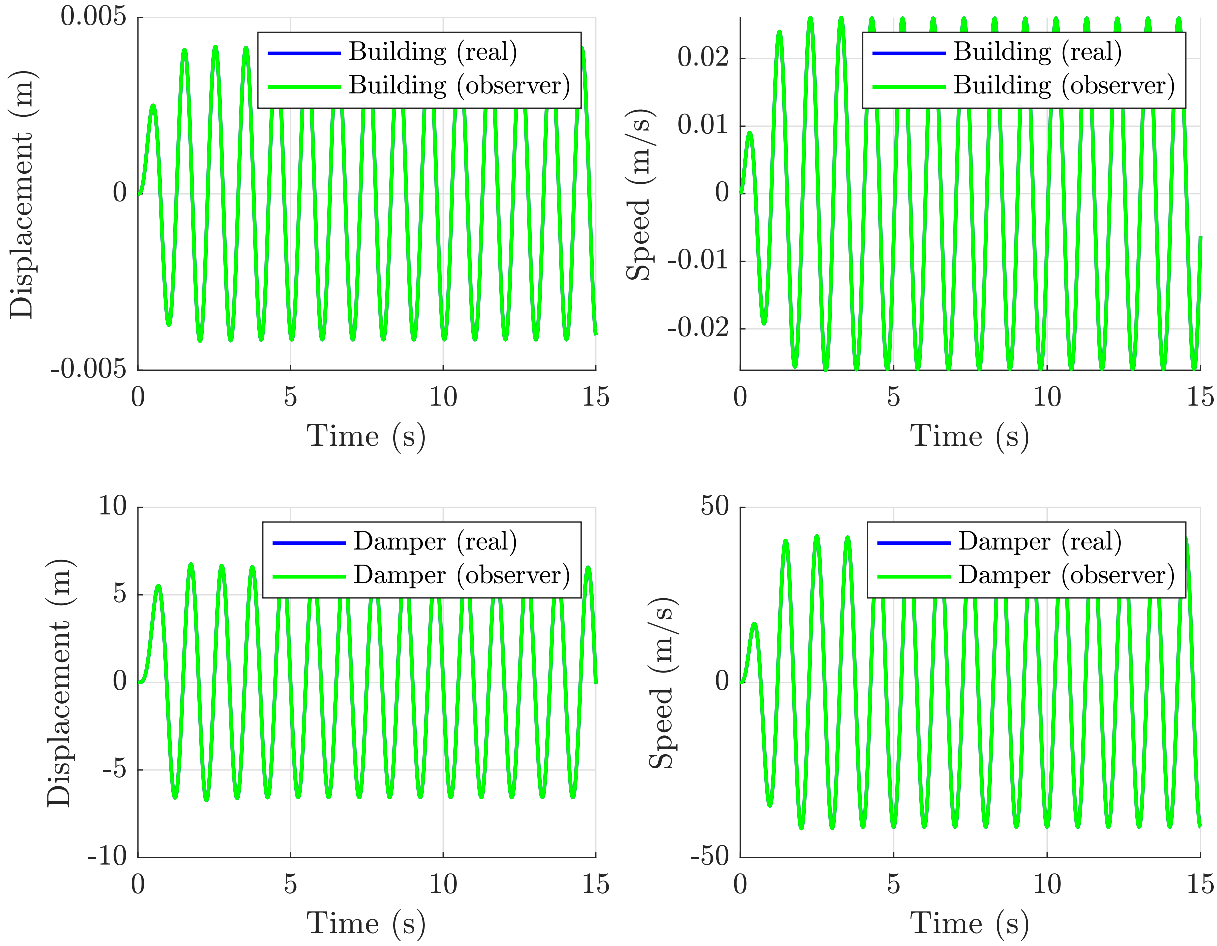


Figure 9 – Response to a sinusoidal wind force - states

3.3.3 Presence of noise

to do

3.3.4 Impact of delays

to do

4 Controller in frequency domain

4.1 Framework

For this part of the work, we have decided to simplify our system and use 2 states instead of 4. The position and speed of the damper are therefore hidden in the force of the actuator, which is still the controllable input of our system.

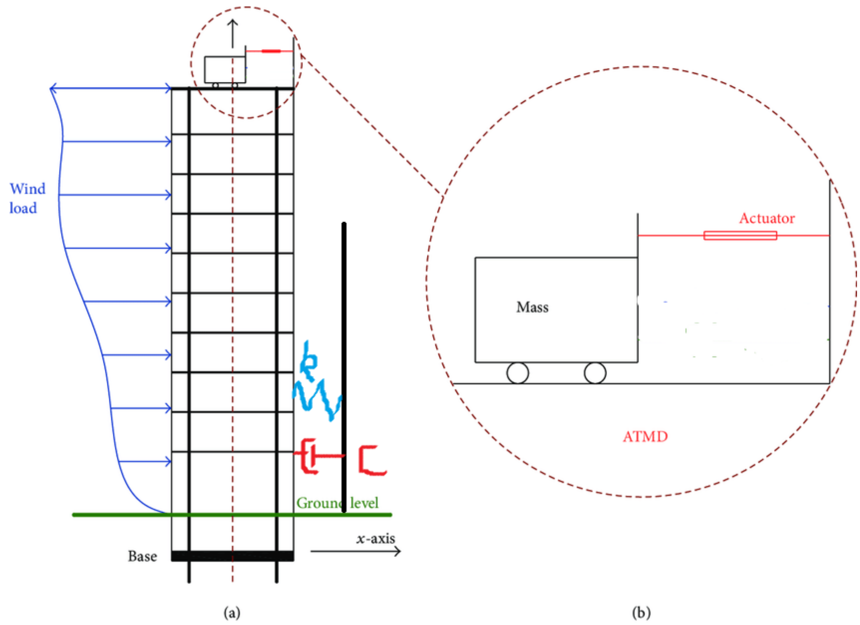


Figure 10 – Simplified system of an active mass damper

The law that governs that system is the following :

$$m_{tot}\ddot{x} + c\dot{x} + kx = F_{wind} + F_{damper}$$

where

- $F_{damper} = m_{damper} a_{damper}$
- $m_{tot} = m_{building} + m_{damper}$
- x is the position of the building relative to its rest position ($x = 0$)

Let's now define the input, output and states :

- $u_1 = F_{wind}$
- $u_2 = F_{damper}$
- $x_1 = x$
- $x_2 = \dot{x}$
- $y = x_1$

By doing so, our ABCD matrices are the following :

$$A = \begin{pmatrix} 0 & 1 \\ \dfrac{-k}{m_{tot}} & \dfrac{-c}{m_{tot}} \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ \dfrac{1}{m_{tot}} & \dfrac{1}{m_{tot}} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \end{pmatrix}$$

4.1.1 Constraints and simulation specifications

We have the following constraints :

- Acceleration of the mass damper between 0.3 and $0.6g$, as advised by Prof. Denoël.
- Power injected in the mass of below 10 kW so as to not have too much electrical consumption.
- Lateral movement of the top of the building not above 1 m.

The two scenario we look at are the following : a turbulent wind of maximum 810 kN, that we represented as a sine function and a constant wind of the same intensity.

Here are the values of the different parameters we have chosen :

Mass	$m_{building} = 1 \times 10^7\,\mathrm{kg}$	$m_{damper} = 3 \times 10^3\,\mathrm{kg}$
Spring	$k \approx 4 \times 10^8\,\mathrm{N\,m^{-1}}$	
Damper	$c \approx 1.3 \times 10^6\,\mathrm{N\,s\,m^{-1}}$	
Wind	$F_{max} = 810\,\mathrm{kN}$	

Table 3 – Numerical values of the system

4.1.2 Choice of cross-over frequency

The frequency of our building is of about 1 Hz, as advised by Pr. Denoël, and the frequency of the sinusoidal wind we have decided to study is also of 1 Hz, so we have decided to use a cross-over frequency of 5 Hz. All frequencies above that, probably coming from noise and unwanted phenomena, will be attenuated, while the amplitudes of the frequencies below that, which correspond to the internals of our system, will be amplified.

$$w_{co} = 2\pi f_{co} \approx 30\,\mathrm{rad\,s^{-1}}$$

4.2 Transfer function of the open-loop system

The Bode plots of our open-loop system are given at figure 11.

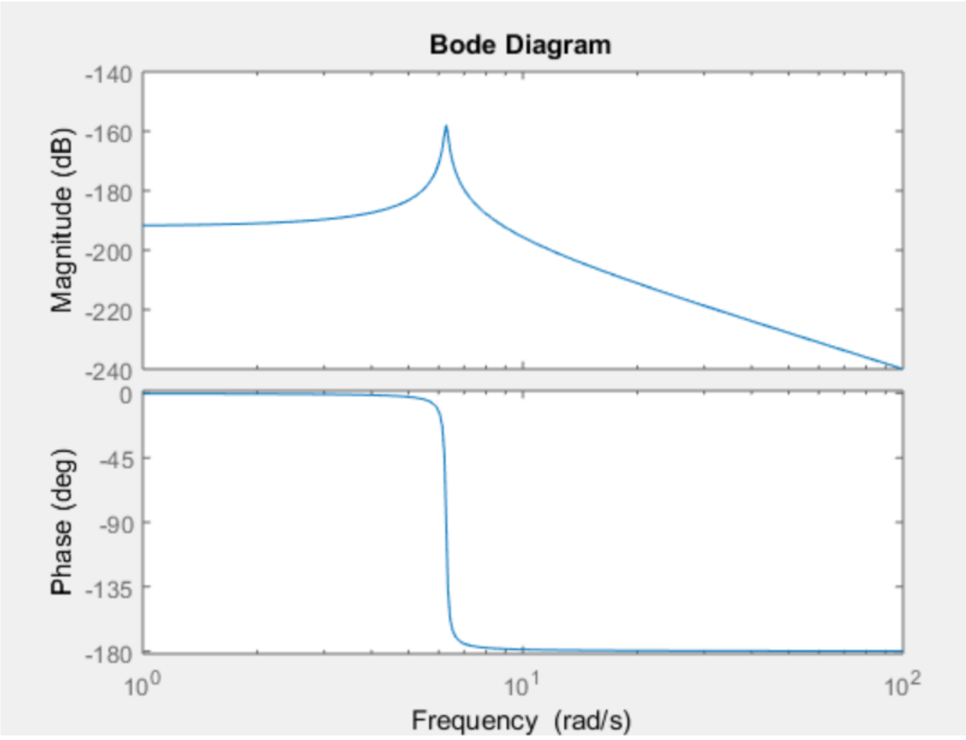


Figure 11 – Bode plots for 2D system.

As can be seen, every frequency is well attenuated, the high ones as well as the low ones.

At the cross-over frequency, the gain of the system is of about -215 dB. This is not what we want. We would like low frequencies to have a positive gain, high frequencies to have a negative one and the gain at the crossover frequency to be of 0 dB.

Furthermore, we need a big enough phase margin at the crossover frequency to be resistant to the delays we will have in our system. In order to do that, we have decided to use a lead compensator as well as a gain. We will not need a low-pass filter as high frequencies will be well attenuated without it.

4.3 Loop shaping

4.3.1 Lead compensator

Let's first start with the desired phase margin. Delays are discussed after, but we want to be able to respond at least to 0.02 s delays, which correspond to the 50 Hz of the actuator's piston [4].

We have decided to have a phase margin of 70°. In order to increase the phase margin at the crossover frequency, we have decided to use a lead compensator.

Its transfer function is given by :

$$G(s) = \frac{\frac{s}{w_z} + 1}{\frac{s}{w_p} + 1}$$

We now have to determine the values of the parameters G_{LC}, w_z and w_p. For a given crossover frequency ω_{co} and a phase margin ϕ_m, we can determine the two w in the following way :

$$\begin{cases} w_z = \tan(\alpha) w_{\text{co}} \\ w_p = \frac{w_{\text{co}}}{\tan(\alpha)} \end{cases}$$

with $\alpha = \frac{\pi}{4} - \frac{\phi_m}{2}$.

For our crossover frequency and our desired value of ϕ_m, we get that :

$$w_z = 5.2898 \qquad w_p = 170.1385$$

4.3.2 Gain

After that, we need to add a gain to our system in order to increase the amplitude gains for all frequencies and make it so that the amplitude is at 0 dB at the crossover frequency. That is done by using a constant gain of 1.5178×10^9. This does not affect the phase but increases the amplitudes of about 183.6 dB, which positions our Bode plot to where we wanted it to be.

4.3.3 Trade-offs

The Bode and Nyquist plots of the controlled system are given at figures 12 and 13. As can be seen, the desired results are obtained, and we have a phase margin of 70° on the Nyquist plot.

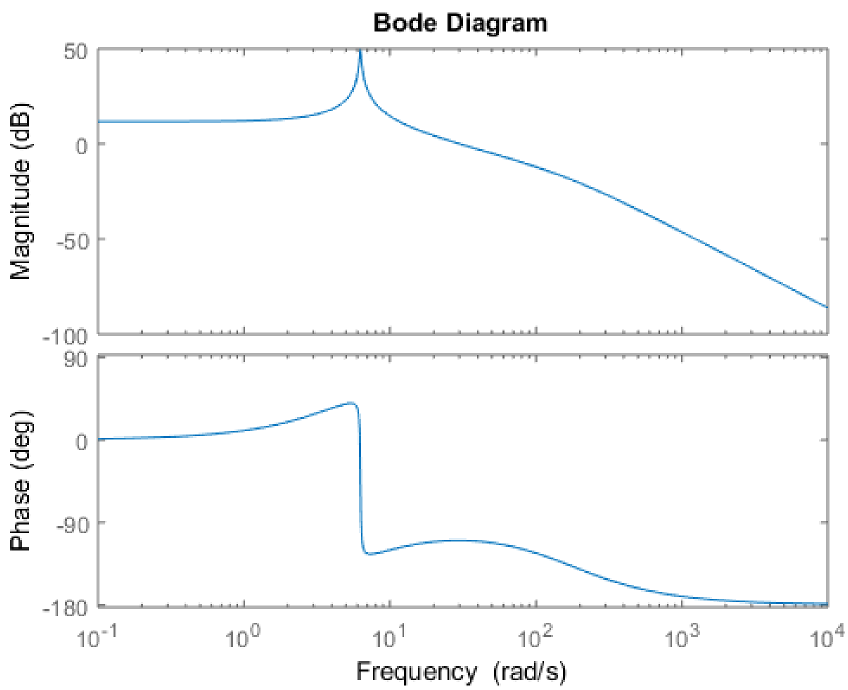


Figure 12 – Bode plots of the controlled system

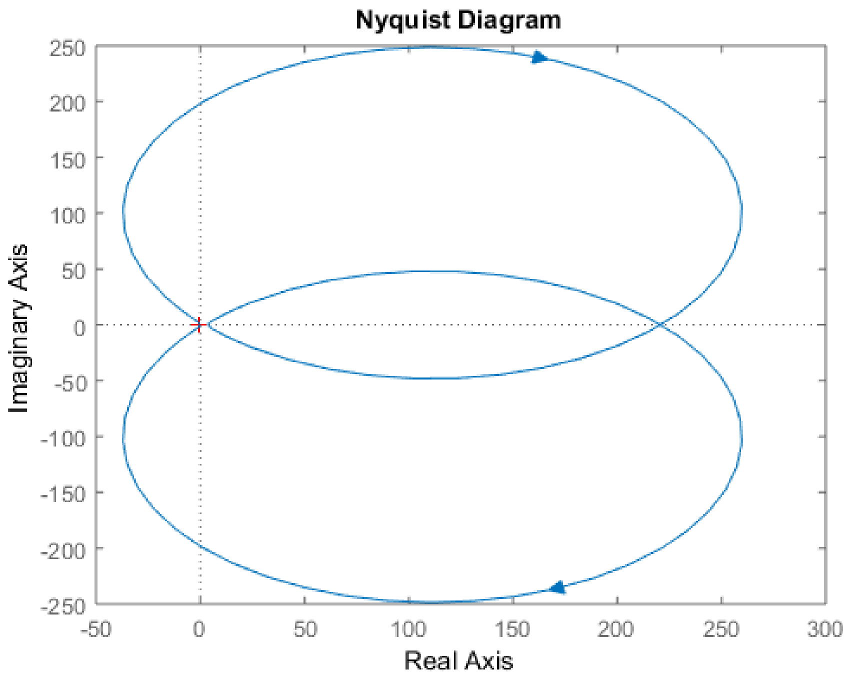


Figure 13 – Nyquist plot of the controlled system

Concerning the impacts on the output signal and control input signal, they are in an acceptable range of values with the parameters we have chosen, as can be seen in figures 14 and 15.

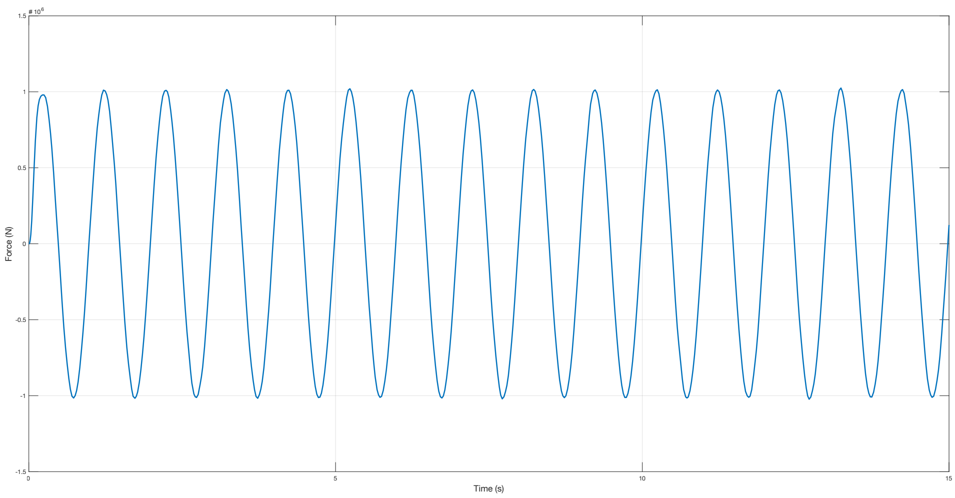


Figure 14 – Plot of the controllable input of the controlled system

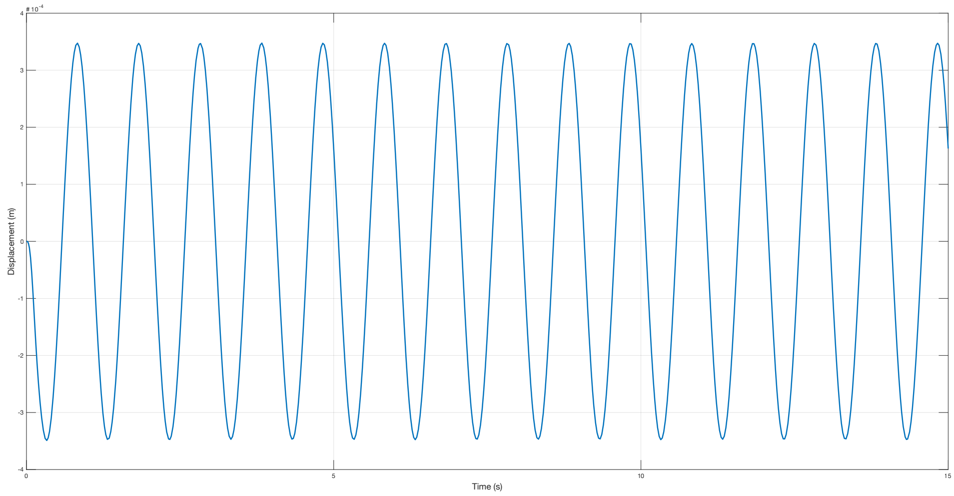


Figure 15 – Plot of the output of the controlled system

4.4 Gang of four

4.4.1 Sensitivity function

$$S(s) = \frac{1}{1 + PC}$$

The Bode plots of the sensitivity function are given at figure 16. That function tells us how the noise acts on the output. We do not want the system to react to the noise, as it is actually the measurement noise that must stay in the output.

That noise is a high-frequency phenomenon and, as can be seen on the Bode plots, there is no attenuation for high frequency, which is what we want.

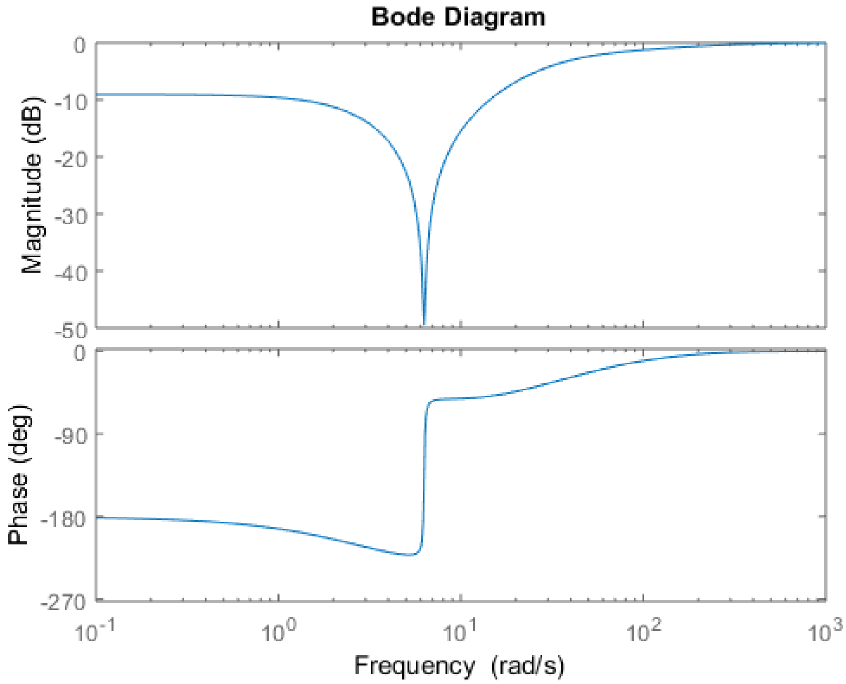


Figure 16 – Bode plots of the sensitivity function

4.4.2 Load sensitivity function

$$PS(s) = \frac{P}{1 + PC}$$

This function tells us how the disturbances act on the output and the Bode diagrams are given at figure 17. Our system needs to be robust against disturbances. In our case, these disturbances are low frequency phenomena (frequency of the wind, which we have either chosen constant or a sine function of frequency equal to 1). We can see that we have a very good reaction concerning the effect of the wind on the output of the system (attenuation of more than $-200\,\text{dB}$).

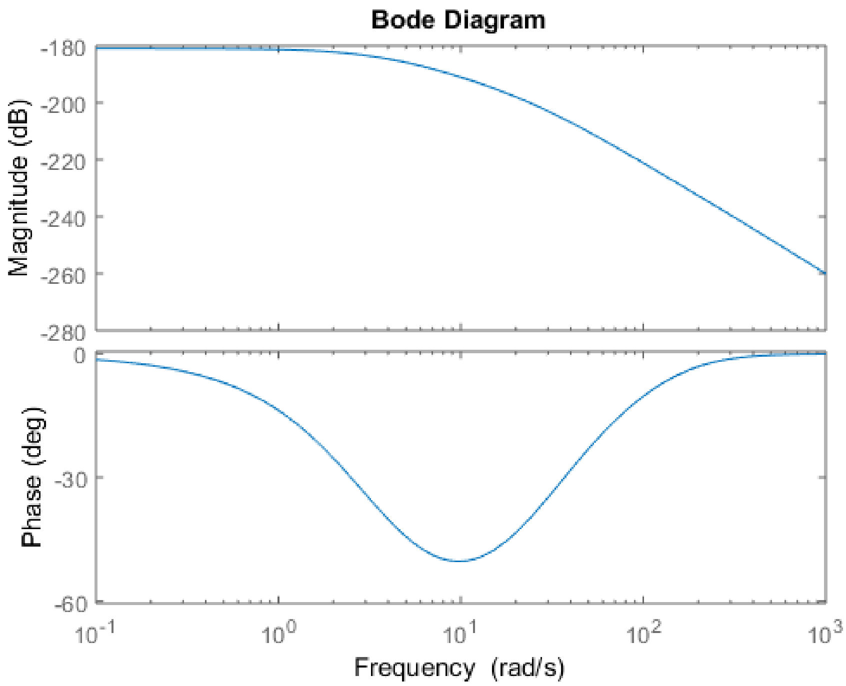


Figure 17 – Bode plots of the load sensitivity function

4.4.3 Complementary sensitivity function

$$PS(s) = \frac{PC}{1 + PC}$$

This function tells us how the disturbances act on the controllable input and the reference acts on the output, and the Bode diagrams are given at figure 18.

The control signal must be reactive to disturbance and the output should be able to track the reference. Amplitudes at low frequency should therefore not be dampened, and we see that they are not attenuated on the plots.

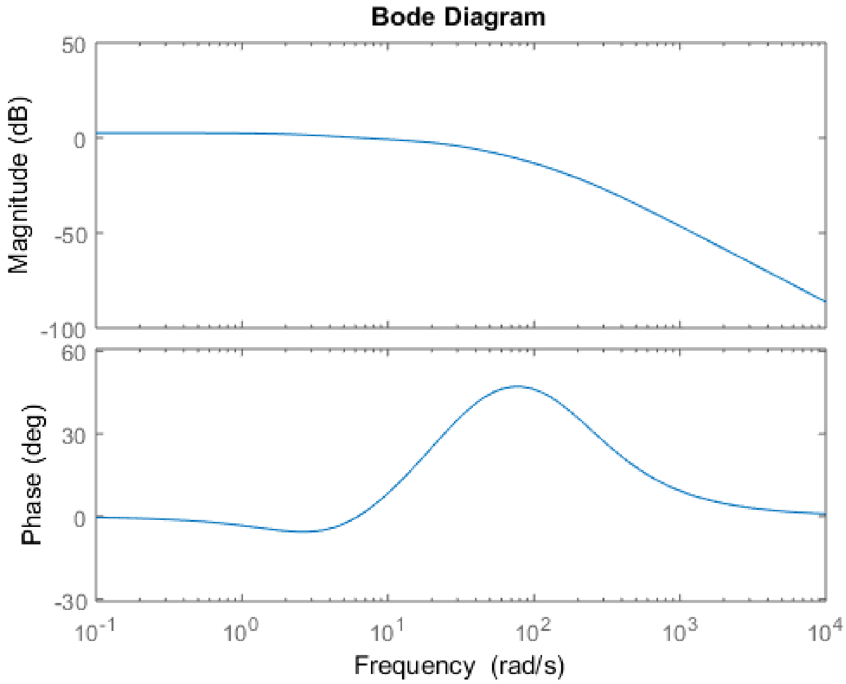


Figure 18 – Bode plots of the complementary sensitivity function

4.4.4 Noise sensitivity function

$$PS(s) = \frac{C}{1 + PC}$$

This function tells us how the noise and the reference act on the controllable input, and the Bode diagrams are given at figure 19.

That function should be reactive to reference changes, but not to noise, and so have a high magnitude at low frequency and low magnitude at high frequencies. We can see that it is not the case here. Indeed, we have high amplitudes for high frequencies. However, as our reference does not change in our system (we do not plan on dampening the oscillations in a Pisa Tower), it does not really matter.

It is also known that temporal domain controllers are better at reacting to reference changes than frequency domain ones.

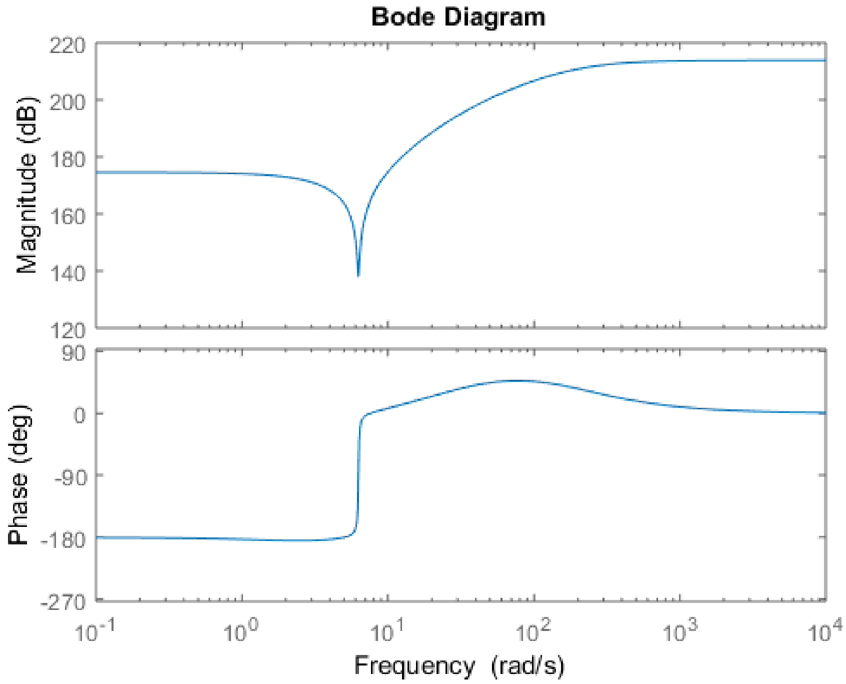


Figure 19 – Bode plots of the noise sensitivity function

4.5 Delays

We have delays in our system that are due to the sensor transmitting the information, the microcontroller processing that information and the actuator (piston) actually moving our mass damper. We have chosen that last source of delay as being the biggest one, with a frequency of 50 Hz, which induces delays of 0.02 s.

Some Bode and Nyquist plots showing delays are given at figures 20, 21 and 22.

As can be seen, for a phase margin of 70°, the delays in our system do not bring instabilities. However, something strange is that a bigger phase margin seems to make our system less robust to delays, as can be seen on the two Nyquist plots given below. That is not how it should be, theoretically speaking. However, we have compared with the way you have designed your components, and do not see a difference.

Our lead compensator and gain have the same "form" as the ones you have used. We have therefore included the Matlab script we have used to plot these figures in the mail we sent.

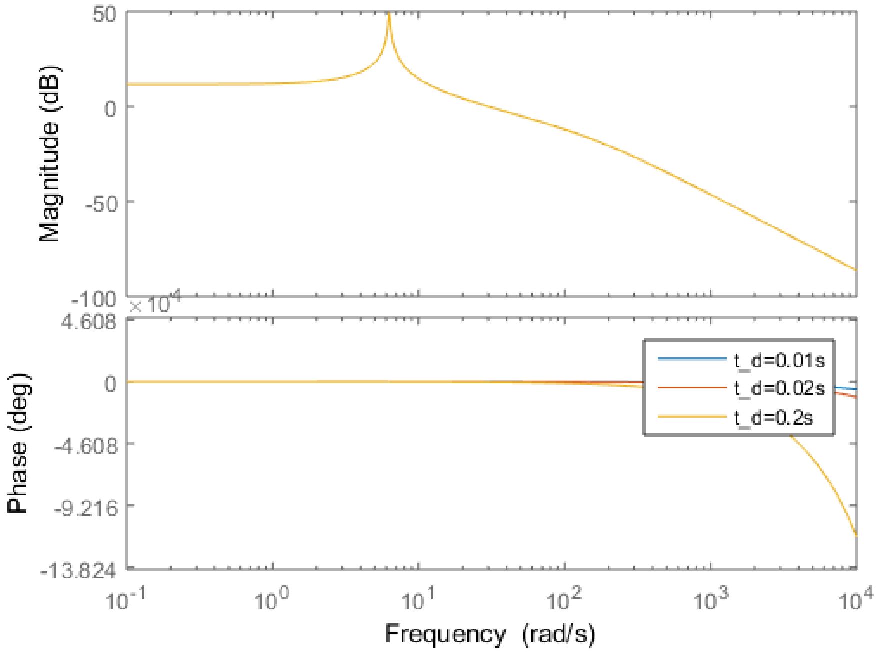


Figure 20 – Bode plots for various delays and a phase margin of 70 degrees.

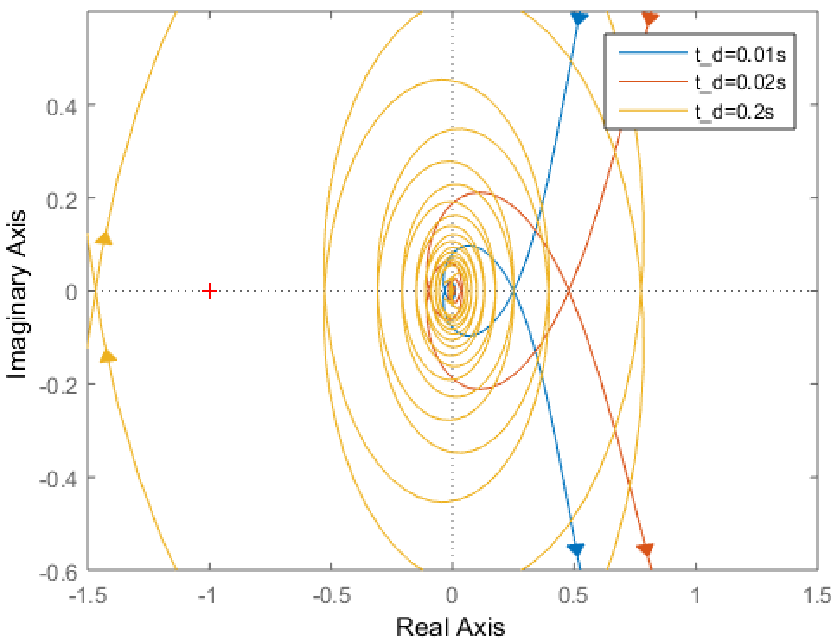


Figure 21 – Nyquist plots for various delays and a phase margin of 70 degrees.

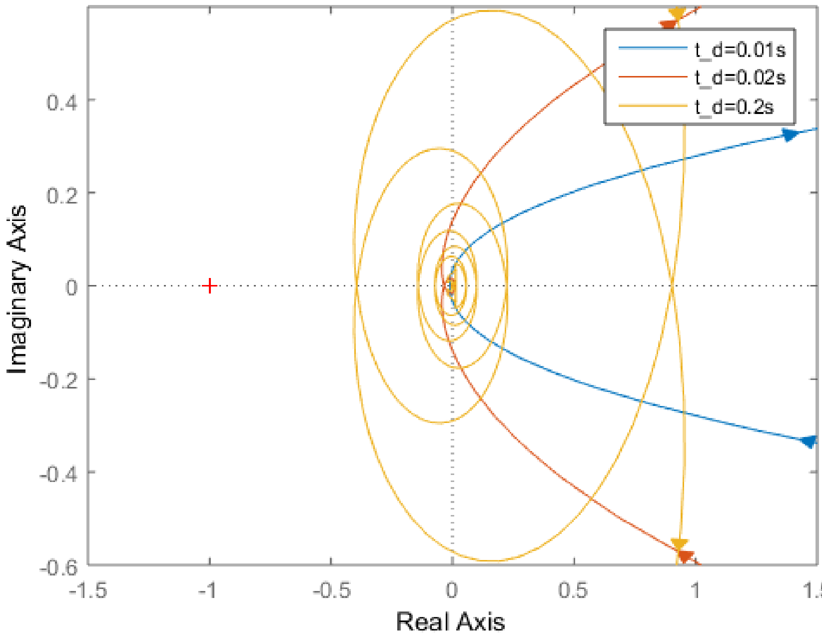


Figure 22 – Nyquist plots for various delays and a phase margin of 1 degree.

4.6 Noise

to do

4.7 Feedforward

to do

5 Conclusion

5.1 Time and frequency domains

to do

5.2 General conclusion

to do

5.3 References

[1] *How To Stop Structures from SHAKING: LEGO Saturn V Tuned Mass Damper*. https://www.youtube.com/watch?v=ft3vTaYbkdE. Accessed : 2019-09-29.

[2] *Tuned mass damper*. https://en.wikipedia.org/wiki/Tuned_mass_damper. Accessed : 2019-09-29.

[3] *Active vibration control of structure by Active Mass Damper and Multi-Modal Negative Acceleration Feedback control algorithm*. https://www.sciencedirect.com/science/article/pii/S0022460X16307957. Accessed : 2019-09-29.

[4] *The effect of time delay on control stability of an electromagnetic active tuned mass damper for vibration control*. https://iopscience.iop.org/article/10.1088/1742-6596/721/1/012007/pdf. Accessed : 2019-12-04.

[5] *Parametric study of active mass dampers for wind-excited tall buildings*. https://www.sciencedirect.com/science/article/pii/0141029695001088. Accessed : 2019-10-08.

[6] *Parameter identification for active mass damper controlled systems*. https://iopscience.iop.org/article/10.1088/1742-6596/744/1/012166/pdf. Accessed : 2019-09-29.

[7] *Active Mass Damper - One Floor (AMD-1)*. https://www.made-for-science.com/de/quanser/?df=made-for-science-quanser-active-mass-damper-coursewarestud-matlab.pdf. Accessed : 2019-09-29.