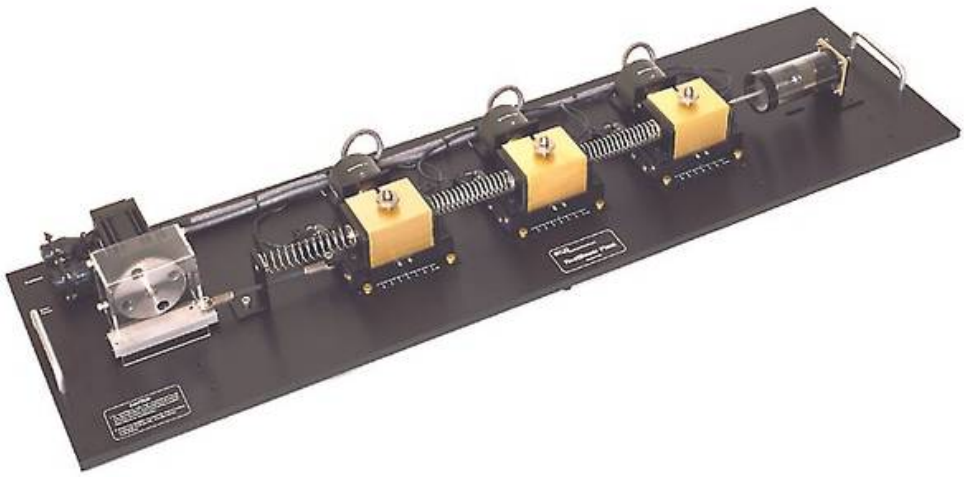
TUHH
Technische Universität Hamburg

ICS
Institute of Control Systems

Control Lab
ORC1

Robust Control of a Spring-Mass-Damper System



14th April, 2021

Room: Online | N-1.077

Summer Semester 2021

Contents

About this Document

This document is intended to help you with the design task *ORC1—Robust Control of a Spring-Mass-Damper System* of the *Control Lab* practical course. It is written mainly in the style of a tutorial and should provide you with all the necessary tools and MATLAB commands to solve the task.

This document is accompanied by MATLAB files that you need to modify and execute in order to develop your own design.

`orc1_model.m` This file deals with uncertainty modeling of the spring-mass-damper system. It is meant to familiarize you with the MATLAB functions and tools needed to construct an uncertain model which can be used for controller synthesis.

- You need to complete the code on your own and submit a published version of this file (see *Publishing*) no later than one week before the scheduled date for the experiment.

`orc1_design.m` This file deals with nominal and robust controller synthesis for the spring-mass-damper system. Robust controller design is based on the uncertain model constructed in the former file. It is meant to familiarize you with the MATLAB functions and tools needed to design a robust $\mathcal{H}_\infty$ controller by applying the Small Gain Theorem.

- You need to complete the code on your own and submit a published version of this file (see *Publishing*) no later than one week before the scheduled date for the experiment.

`orc1_simulation.slx` This file contains the uncertain simulation model and is used for controller evaluation.

The code is guaranteed to work with MATLAB 2019a 64bit, other versions might not be supported. You can get the latest MATLAB version from `https://www.tuhh.de/rzt/usc/Matlab/index.html` or use the pool computers.

In this document, you will encounter blocks that indicate MATLAB code:

```
[MATLAB COMMANDS]=USEFUL(TOOLS)
```

These are meant to get you started. You can (and should) use the `help` command within MATLAB to find out more about a particular command.

Another thing that you will encounter are preparation tasks:

Preparation: Sketch the Nyquist plot of $G(s) = \dfrac{1}{s^2 + 0.01s + 1}$ and check if unity feedback stabilizes the standard closed-loop interconnection.

These are meant to prepare you for the question session that will take place prior to conducting the experiment.

Task

Design a nominal controller and a robust controller for the spring-mass-damper system that achieve the requirements specified in Section 1 and evaluate them in simulation.

Attach all necessary MATLAB and SIMULINK files that were provided to you no later than one week before the experiment via email to the responsible Tutor and Supervisor. You will get an email when your preparation is not sufficient to pass the Lab and will get time to revise your design.

Checklist

- ☐ I read this whole document carefully.
- ☐ I did all preparation tasks and can explain them.
- ☐ I completed `orc1_model.m`.
- ☐ I completed `orc1_design.m`.
- ☐ I submitted all MATLAB files.

1 Plant

In this laboratory project the design of a nominal and a robust control system for a single-input-single-output (SISO) *two-degrees-of-freedom* (2-DOF) spring-mass-damper system is considered. The spring-mass-damper system itself is a common control experimental device which however is closely related to industrial control applications. The system to be controlled consists of two rail mounted masses interconnected with two springs. The springs connected to the masses can be changed. Furthermore, the rightmost cart can be connected to a viscous damper. A free body diagram of the plant is shown in Fig. 1.

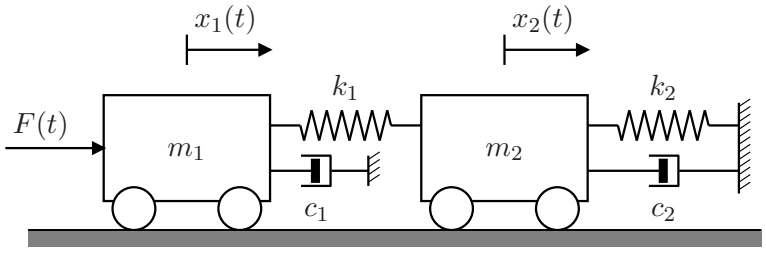


Figure 1: Free body diagram of 2-DOF spring-mass-damper plant

The springs connected to the carts are assumed to have spring stiffness constants k_1 and k_2, whereas the masses of the carts are m_1 and m_2, respectively. The equations of motion of the system can be expressed as:

$$m_1\ddot{x}_1(t) + c_1\dot{x}_1(t) + k_1x_1(t) - k_1x_2(t) = F(t), \tag{1a}$$

$$m_2\ddot{x}_2(t) + c_2\dot{x}_2(t) + (k_1 + k_2)x_2(t) - k_1x_1(t) = 0, \tag{1b}$$

where $y(t) = x_2(t)$ is the measured position of cart 2 in centimetres and $F(t)$ is a force input in Newton. Here, $F(t)$ represents the control input to control $y(t)$. Viscous friction for each cart is taken into account by the damping coefficients c_1 and c_2.

During the experiment spring 1 represented by k_1 and spring 2 represented by k_2 can be changed such that $k_1 \in \mathbb{K}_1$ and $k_2 \in \mathbb{K}_2$, where $\mathbb{K}_1 = \{200\,\text{N/m},\ 390,\text{N/m},\ 830\,\text{N/m}\}$ and $\mathbb{K}_2 = \{200\,\text{N/m},\ 390\,\text{N/m}\}$. Here, (1) with $k_1 = 390\,\text{N/m}$ and $k_2 = 390\text{N/m}$ is considered as the nominal plant model. The masses of cart 1 and 2 are assumed to be equal and given by $m_1 = m_2 = 1.55\,\text{kg}$. To illustrate the effect of changing spring stiffness constants for springs 1 and 2, Fig. 2 shows the Bode plots for all possible combinations of springs connected to cart 1 and 2, where the damping coefficients c_1 and c_2 have been set to $c_1 = c_2 = 0.01\,\text{Ns/m}$.

Prep. 1.1: Construct a state space model based on (1) for the given physical parameters, where $x(t) = [x_1(t)\ \dot{x}_1(t)\ x_2(t)\ \dot{x}_2(t)]^T$.

As can be seen two resonance frequencies $\omega_{\text{n}1}$, $\omega_{\text{n}2}$ appear in the Bode plot which change significantly with changing spring stiffness such that $\omega_{\text{n}1}$ varies between $\omega_{\text{n}1} \approx 7\text{rad/s}$ and $\omega_{\text{n}1} \approx 10.5\,\text{rad/s}$ while $\omega_{\text{n}2}$ varies between $\omega_{\text{n}2} \approx 18.4\,\text{rad/s}$ and $\omega_{\text{n}2} \approx 34.8\,\text{rad/s}$.

Also the bandwidth ω_{b} defined as the first frequency where the gain drops 3dB below its static gain value varies and changes between $\omega_{\text{b}} \approx 12.5\text{rad/s}$ and $\omega_{\text{b}} \approx 28\text{rad/s}$. Furthermore, it can be observed that the static gain K_0 solely depends on k_2 and attains the values $K_0 = \{0.2564, 0.5\}$. Fig. 3 (a) shows the Nyquist plot of the plant for the different springs k_1 and k_2, where the critical point marked in red is visible in the enlargement shown in Fig. 3 (b).

Prep. 1.2: Looking at the Bode plot, what kind of controller properties (roll-off, poles, zeros) can achieve robust stability of the closed loop? Which closed loop bandwidth is reasonable for robust controller design?

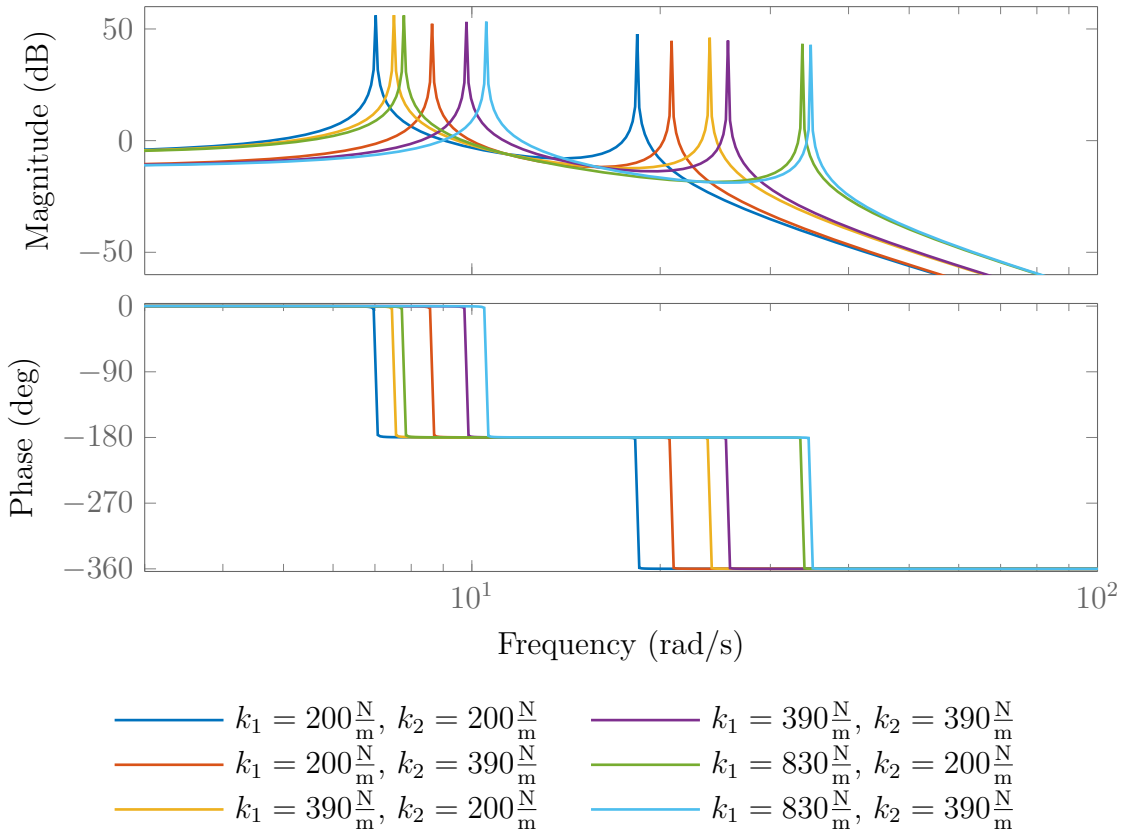


Figure 2: Bode plot of the plant for different spring stiffness

Prep. 1.3: What can be concluded regarding stability of the closed-loop system under unity feedback from the Nyquist plot? What can be said about gain margin and phase margin?

1.1 Including The Actuator

As mentioned before the force $F(t)$ shown in Fig. 1 denotes the control force. This force is generated by an actuator which is a DC brushless motor (permanent magnet synchronous motor). Consequently, the input to the system which consists of actuator and mechanical plant is not a force but a voltage $V(t)$ applied to the DC brushless motor that generates the force $F(t)$ acting on mass 1. This is illustrated in Fig. 4, where the actuator dynamics are represented by $M(s)$ and the mechanical plant by $G(s)$.

However, for controller design it is assumed that the output of the controller is the force $F(t)$ acting directly on the first mass. This is realized by inserting an approximated

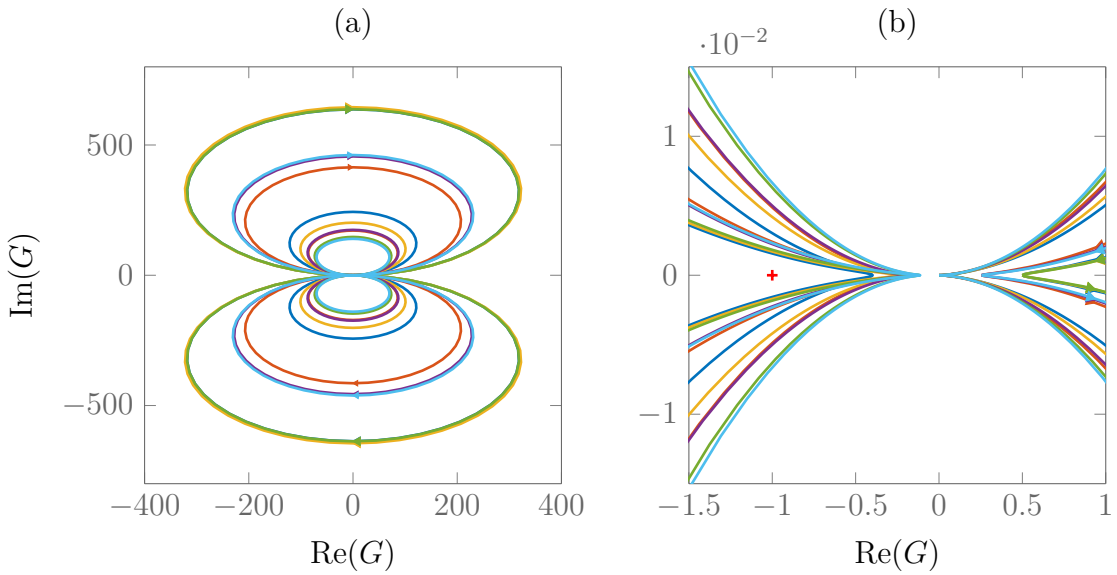


Figure 3: Nyquist plot of the plant for different spring stiffness

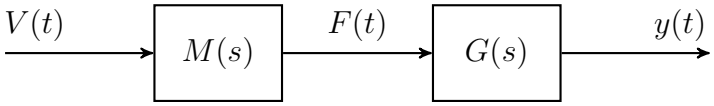


Figure 4: Actuator and mechanical plant

inverse motor model given by a constant gain U in front of the actuator as shown in Fig. 5. The gain is chosen such that $F_c(t) \approx F(t)$ within the actuator bandwidth ω_{ab} , i.e., $U = M(0)^{-1}$ such that $UM(j\omega) \approx 1$ for $\omega \leq \omega_{\text{ab}}$.

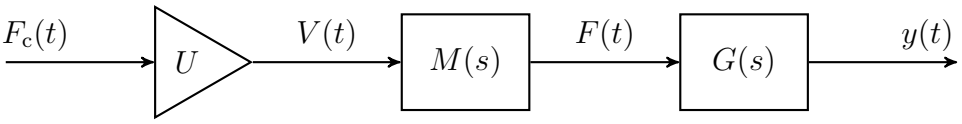


Figure 5: Inverse actuator model, actuator and mechanical plant

1.2 Limits of The Plant

Since the range of motion for the carts is limited, there are two limit switches such that the maximal displacement is about 30mm in each direction. If the displacement exceeds this limit an emergency shut down will be triggered to avoid damage. Therefore, and to

leave room for overshoot when exciting the plant the reference signal $r(t)$ is limited to $|r(t)| \leq 20\text{mm}$. Regarding the actuator, input voltages of $V(t) \leq 10\text{V}$ can be applied.

1.3 Design Specifications

1. **Nominal design (all model parameters are fixed to nominal vales)**

- Command tracking: The rise time (10%-90% of the final value) for the reference input $r(t) = \sigma(t)$ is less than 0.75 seconds.
- Command tracking: The overshoot is less than 10%.
- Command tracking: The steady state error is less than 2%.
- Input Constraints: The control input $u(t)$ satisfies $|u(t)| < 10$ for $r(t) = \sigma(t)$.
- Stability margins: At least 6dB gain margin and 60deg phase margin.
- All controller poles satisfy: $|p_i| \leq 2000$

2. **Robust design**

- Robust stability is guaranteed for all admissible operating conditions.
- Command tracking: The rise time is less than 3 seconds for all admissible operating conditions.
- Stability margins: At least 6dB gain margin and 20deg phase margin for all admissible operating conditions.
- All other requirements of the nominal design are met.

2 Modeling Uncertainty

To model rational parametric uncertainties *linear fractional representations* or LFRs are very useful. In this framework uncertainties are modeled by including an additional uncertainty channel in a nominal plant description. To illustrate this, consider the state space model you constructed as preparation in the last section. Now represent the uncertain spring constants as

$$k_i(\delta) = k_{i0} + \tilde{k}_i\,\delta, \tag{2}$$

where k_{i0} is the nominal spring stiffness of spring i and $\tilde{k}_i$ is determined such that $k_i(\delta)$ covers all admissible values of k_i when $|\delta| \leq 1$ for $i = 1, 2$.

Prep. 2.1: How should k_{i0} and $\tilde{k}_i$ be selected to cover the admissible values specified in Section 1?

Let

$$\dot{x}(t) = A_0 x(t) + B_0 u(t),$$
$$y(t) = C_0 x(t),$$

be the nominal plant model of the spring-mass-damper system. By adding a fictitious input channel $w(t) \in \mathbb{R}^{n_\mathrm{w}}$ and output channel $z(t) \in \mathbb{R}^{n_\mathrm{z}}$ connected by a block Δ_u, we can construct an uncertain LFR

$$\dot{x}(t) = A_0 x(t) + B_0 u(t) + B_\mathrm{w} w(t), \tag{3a}$$
$$z(t) = C_\mathrm{z} x(t), \tag{3b}$$
$$y(t) = C_0 x(t), \tag{3c}$$
$$w(t) = \Delta_\mathrm{u} z(t), \tag{3d}$$

where the model uncertainty is represented by the $\Delta_\mathrm{u} \in \mathbb{R}^{n_\mathrm{w} \times n_\mathrm{z}}$ block and Δ_u is scaled such that the constraint $\|\Delta_\mathrm{u}\| \leq 1$ is satisfied.

Prep. 2.2: Review Exercise 21.1 ORC, and construct an LFR model as in (3) of the spring-mass-damper system with uncertain spring constants k_1 and k_2. How should Δ_u be selected for (3) to cover all admissible spring constants?

One convenient way to model parametric or structured uncertainties in Matlab is to use uncertain state space models. In order to do that we can define real parametric uncertainties with the Matlab command `ureal` (see help and Exercise 21.1 ORC lecture notes). This allows us to define the uncertain state space matrices symbolically and extract the LFR later on from an uncertain state space model created with the Matlab command `uss`. As a simple example we can define the uncertain state space matrices of a 1-DOF spring-mass-damper system with the following code.

```
k_range = [0.75 1.25]; k0 = mean(k_range);
m_range = [0.75 1.25]; m0 = mean(m_range);
c_range = [0.05 0.1]; c0 = mean(c_range);

k = ureal('k', k0, 'range', k_range);
m = ureal('m', m0, 'range', m_range);
c = ureal('c', c0, 'range', c_range);

A  = [ 0            1;...
       -k/m     -c/m];

B = [ 0;...
    1/m];

C = [1 0];

D = 0;

sys = uss(A,B,C,D);
```

2.1 How to proceed

Take a look at the provided Matlab File `orc1_model.m` and complete it to familiarize yourself with modeling parametric uncertainties using Matlab. Create an uncertain state space model of the plant that can be used as a synthesis model for controller design later on.

3 Controller Design

For designing an LTI output feedback controller, which stabilizes the plant robustly and achieves the performance objectives for all admissible springs, MATLABS $\mathcal{H}_\infty$ synthesis tools for controller design will be used. Before the synthesis is discussed two weighting schemes known from the lecture are briefly reviewed.

3.1 S/KS Design

As it is known from the lecture *Optimal and Robust Control* (ORC lecture notes, Chapter 17), a powerful method for controller design is to shape certain transfer functions of the closed-loop system shown in Fig. 6. A standard approach is to shape the sensitivity S and the control sensitivity KS.

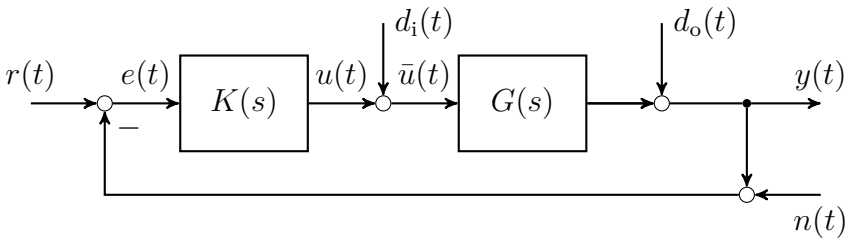


Figure 6: Closed-loop system with input and output disturbance, noise and reference signal.

To actually shape S and KS weighting filters W_S and W_K are introduced to weight the outputs specified by the performance objective.

3.2 SG/T_K Design

A design based on shaping S and KS can lead to undesirable pole-zero cancelations between plant and controller, which lead to a poor response to input disturbances. This can be avoided by shaping the transfer functions from d_i to y and $\bar{u}$, respectively (compare Exercise 17.2.c ORC). In this way we consider input disturbances instead of references changes or output disturbances as the external input. The transfer function $T_{\mathrm{yd_i}}(s)$ is denoted by SG and $-T_{\mathrm{ud_i}}(s)$ by T_K respectively. Since we consider a SISO plant in this laboratory experiment it holds that $T = T_\mathrm{K}$. By shaping SG we can design a controller which rejects input disturbances to a satisfactory degree, whereas shaping T_K means that we shape the transfer function from input disturbance to controller output. However, to achieve high frequency controller roll-off, shaping T_K is more involved than shaping KS.

Prep. 3.1: Review the concept of closed-loop shaping (Chapter 16 and 17 of the ORC lecture notes). Make sure you understand how the tuning works qualitatively.

Prep. 3.2: What kind of weighting filter is needed for T_K to achieve high frequency controller roll-off?

3.3 Generalized Plant

In the lecture *Optimal and Robust Control* the generalized plant concept was introduced, where the generalized plant P is of the form

$$\dot{x}(t) = Ax(t) + B_{\mathrm{w}}w(t) + B_{\mathrm{u}}u(t), \tag{4a}$$
$$z(t) = C_{\mathrm{z}}x(t) + D_{\mathrm{zw}}w(t) + D_{\mathrm{zu}}u(t), \tag{4b}$$
$$v(t) = C_{\mathrm{v}}x(t) + D_{\mathrm{vw}}w(t), \tag{4c}$$

with external input $w(t)$, control input $u(t)$, performance output $z(t)$ and measured output $v(t)$. For our synthesis tools to work, we need to first construct a generalized plant that includes the desired performance inputs and outputs. The generalized plant for the S/KS formulation is depicted in Fig. 7.

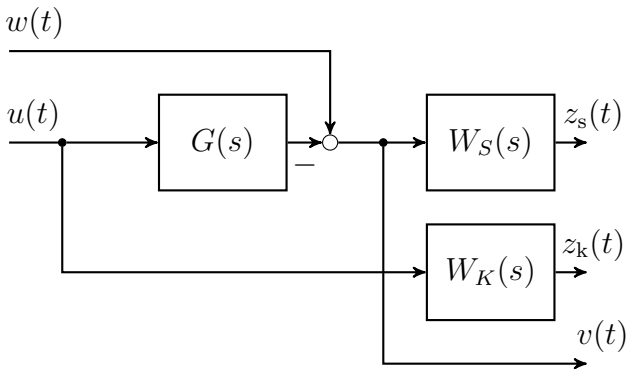


Figure 7: Open-loop generalized Plant P for the S/KS mixed sensitivity design.

In order construct it, it is convenient to use MATLABS `sysic` command (see help). To get you started, consider the following code:

```
systemnames = 'G Ws Wk';
inputvar = '[w{1}; u{1}]';
input_to_G = '[u]';
input_to_Ws = '[w-G]';
input_to_Wk = '[u]';
outputvar = '[Ws; Wk; w-G]';
P = sysic
```

However, to familiarize yourself with the generalized plant concept we want to construct it by hand. Consider the physical plant described in state space by

$$\dot{x}_\mathrm{p}(t) = A_\mathrm{p} x_\mathrm{p}(t) + B_\mathrm{p} u(t), \tag{5a}$$
$$y(t) = C_\mathrm{p} x_\mathrm{p}(t) \tag{5b}$$

and let sensitivity and control sensitivity weighting filter W_S and W_K be given by

$$\dot{x}_\mathrm{s}(t) = A_\mathrm{s} x_\mathrm{s}(t) + B_\mathrm{s} v(t), \tag{6a}$$
$$z_\mathrm{s}(t) = C_\mathrm{s} x_\mathrm{s}(t) + D_\mathrm{s} v(t), \tag{6b}$$

where $v(t) = w(t) - y(t)$ and

$$\dot{x}_\mathrm{k}(t) = A_\mathrm{k} x_\mathrm{k}(t) + B_\mathrm{k} u(t), \tag{7a}$$
$$z_\mathrm{k}(t) = C_\mathrm{k} x_\mathrm{k}(t) + D_\mathrm{k} u(t). \tag{7b}$$

Prep. 3.3: Construct the generalized plant matrices given in (4) based on (5) to (7) for $x^T(t) = [x_\mathrm{p}^T(t)\ x_\mathrm{s}^T(t)\ x_\mathrm{k}^T(t)]^T$. Why do we need $D_\mathrm{k} \neq 0$?

A typical choice of weighting filters W_S and W_K is shown in Fig. 8.

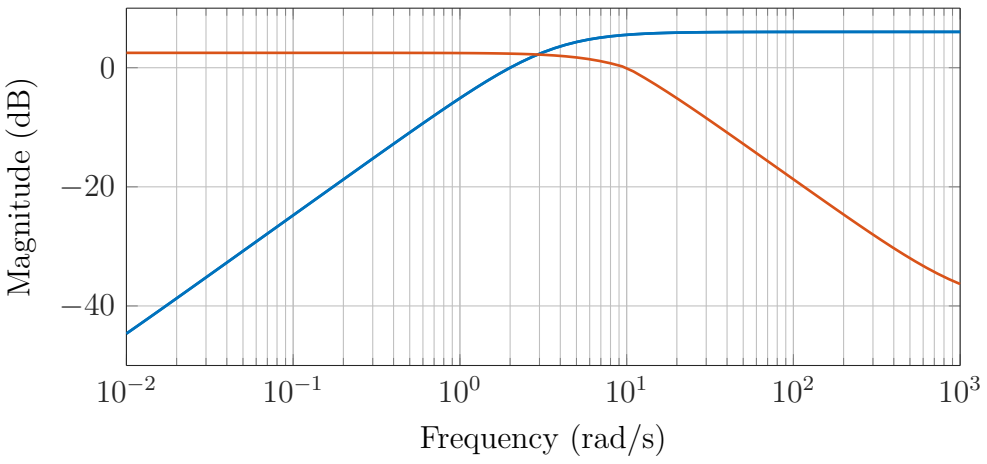


Figure 8: Typical choice of weighting filters. Plotted are the inverse weights W_S^{-1} (blue) and W_K^{-1} (red)

3.4 Synthesis & Robust Stability

Stability and $\mathcal{H}_\infty$ constraints can be expressed as linear matrix inequalities (LMIs). If there exists a symmetric positive definite Lyapunov matrix P, which satisfies

$$P > 0 \,, \tag{8a}$$

$$\begin{bmatrix} A^TP + PA & PB & C^T \\ B^TP & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \,, \quad (\mathcal{H}_\infty \text{ constraint, Theorem 18.3 ORC lecutre notes}) \tag{8b}$$

the system is stable and $\|T_{zw}\|_\infty < \gamma$, where T_{zw} denotes the closed loop transfer function from external input $w(t)$ to performance output $z(t)$ and some performance index γ. The output feedback controller synthesis is based on solving LMIs (8) for the closed-loop system, which is shown in Fig. 9.

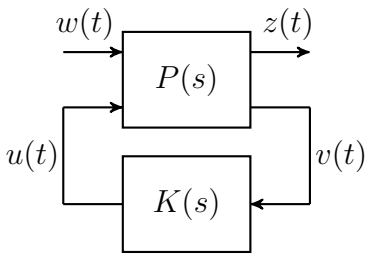


Figure 9: Closed-loop interconnection of generalized plant and controller

Since we are not only interested in designing a nominal controller but also in designing a robust controller the external input and the performance output are augmented by the uncertainty channels of the LFR. Consequently, the closed-loop interconnection of generalized plant and controller becomes as shown in Fig. 10, where P_0 represents the nominal plant with $w_\mathrm{u}(t) = 0$ and $w_\mathrm{p}(t)$ equals $w(t)$ from Fig. 7 in case of S/KS design.

Comparing the block diagrams shown in Fig. 9 and in Fig. 10, we see that for the uncertain generalized plant P we have $w(t) = [w_\mathrm{u}^T(t)\ w_\mathrm{p}^T(t)]^T$ and $z(t) = [z_\mathrm{u}^T(t)\ z_\mathrm{p}^T(t)]^T$. To include the uncertainty of the plant we close the upper loop by inserting a Δ_u block and obtain the block diagram shown in Fig. 11 for which we want to find a controller K that guarantees robust stability.

To achieve our goal we can invoke the Small Gain Theorem (Theorem 21.1 ORC). To further include performance objectives we can formally close the performance loop by

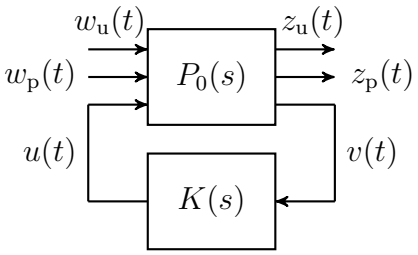


Figure 10: Uncertain closed-loop interconnection of generalized plant and controller

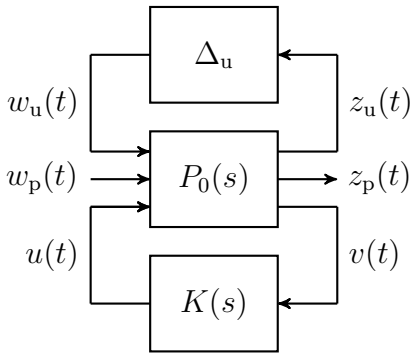


Figure 11: Closed-loop interconnection of generalized plant and controller with uncertainty block Δ_u

adding a second, fictitious uncertainty block Δ_p, yielding the block diagram shown in Fig. 12. In contrast to Δ_u which is diagonal by construction of the uncertain plant model (3), we assume Δ_p to be an arbitrary, unstructured complex matrix that is subject to the constraint $\|\Delta_\mathrm{p}\| \leq \frac{1}{\gamma}$, where $\gamma > 0$ will be used as a performance measure.

Theorem 1 (Small Gain Theorem)**.** *Suppose that $M(s)$ is a proper real rational stable transfer function. Then the interconnected system shown in Fig. 13 is internally stable for all proper real rational stable transfer functions $\Delta(s)$ with*

a) $\|\Delta(s)\|_\infty \leq 1/\gamma$ *if and only if* $\|M(s)\|_\infty < \gamma$.

b) $\|\Delta(s)\|_\infty < 1/\gamma$ *if and only if* $\|M(s)\|_\infty \leq \gamma$.

Prep. 3.4: Express the closed-loop interconnection in Fig. 12 in terms of M and Δ as shown in Fig. 13. What is M and what is Δ? (Hint: consider a block diagonal matrix Δ).

Prep. 3.5: How can the Small Gain Theorem be used to express the condition that the

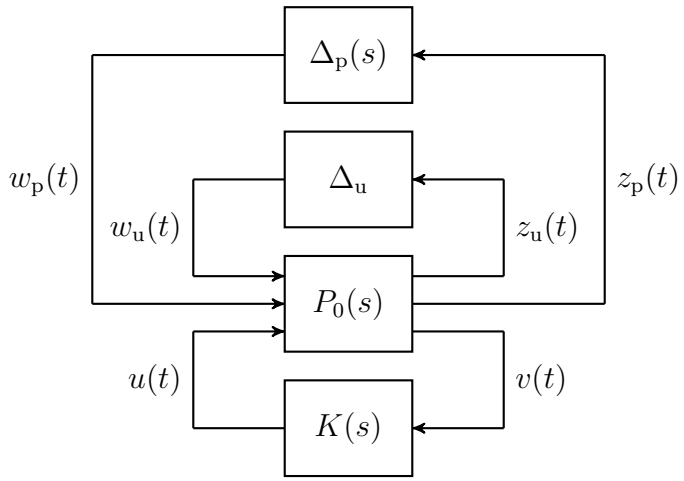


Figure 12: Closed-loop interconnection of generalized plant and controller with uncertainty block Δ_u and performance block Δ_p

closed-loop system is stable and the $\mathcal{H}_\infty$ performance from w_p to z_p is less than γ for all admissible springs, as a constraint on the $\mathcal{H}_\infty$ norm from w to z in Fig. 13?

Prep. 3.6: Why is this condition conservative?

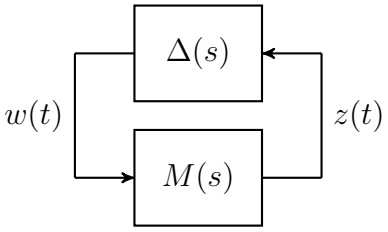


Figure 13: Small gain theorem

In this experiment we will use the tools for $\mathcal{H}_\infty$ optimal controller design implemented in MATLAB, where we just need to provide the generalized plant P and the number of available measurements `nmeas` and control inputs `ncont`:

```
[K,CL,gam,INFO] = hinfsyn(ssbal(P),nmeas,ncont,'method','lmi');
```

Note that we do not use the generalized plant "as it is" but that we do apply a balancing transformation by invoking `ssbal`. This transformation scales the states of the system in such a way, that the entries in the B and C matrices are "of the same size". The reasons for doing so are numerical issues related to representing the numbers

with finite precision. It is easy to find examples in which the synthesis code without this balancing does not produce any useful results at all, so balancing is one of the things that should always be performed to avoid unnecessary numerical issues.

A second thing to note is, that optimality with respect to the $\mathcal{H}_\infty$-norm can yield a numerically ill-conditioned controller. The reasons here are a bit more complex, and again are largely related to numerical issues within the solution of the LMI problem. So often a suboptimal, rather than a truly optimal controller is desired. We can get a suboptimal controller with a specified loss-of-performance by again calling the `hinfsyn` command, but this time tell the optimization to stop when a predefined γ is reached. We know from the first synthesis, that we can find a controller that achieves `gam`, so now we can just search for a controller that achieves `1.1*gam`, i.e., that is at most 10 % worse in the $\mathcal{H}_\infty$ sense.

```
[K,CL,gam,INFO] = hinfsyn(ssbal(P),nmeas,ncont,'method','lmi',
                'GMIN',1.1*gam);
```

Note that we need a first run to determine the `gam` that we use in this second synthesis step. In order to determine whether the designed controller K is feasible for implementation on a digital system we have to consider the sampling frequency of f_s =1000 Hz as well as the fastest controller dynamics. More precisely speaking the following inequality should be satisfied

$$f_\mathrm{s} > \frac{|p_\mathrm{max}|}{\pi}, \tag{9}$$

where p_max denotes the fastest controller pole. This can be checked in MATLAB using

```
pmax_abs = max(abs(eig(K.a)));
```

3.5 Robustness vs. Performance

As shown in the previous section the uncertain generalized plant description is given as shown in Fig. 14 and the Small Gain Theorem can be invoked to test for robust stability.

However, it is also known that this test is conservative and may lead to unsatisfactory results. To account for this conservatism, robustness can be traded against performance by rescaling the uncertainty as well as shifting the nominal plant model. In particular,

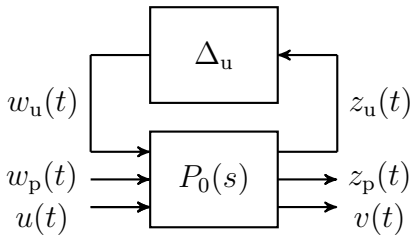


Figure 14: Generalized plant with uncertainty block Δ_u

the range of the uncertainty as well as the nominal plant model can be seen as additional tuning knobs in the design process. That is, instead of the generalized plant shown in Fig. 14 a modified generalized plant depicted in Fig. 15 is used for controller design.

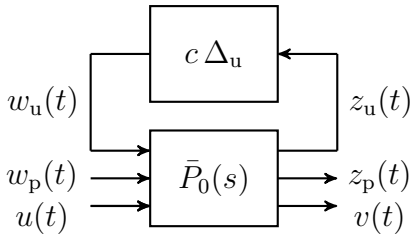


Figure 15: Generalized plant with rescaled Δ_u and shifted nominal plant model

Here, the scaling factor c can be chosen such that $c \in [0,\, 1]$ and for the shifted nominal plant model holds that $\bar{P}_0 \in \mathrm{Co}(\mathbb{K}_1 \times \mathbb{K}_2)$, where Co(.) denotes the convex hull.

Prep. 3.7: Which consequences has the choice $c = 0$ for controller design?

Prep. 3.8: Sketch the set $\mathrm{Co}(\mathbb{K}_1 \times \mathbb{K}_2)$.

4 Closed Loop Analysis

As a first step before entering a more detailed analysis of the closed-loop system, it has to be checked if the closed loop is stable for all possible springs k_1 and k_2. A sufficient test for stability is given by the Small Gain Theorem. However, since this test is conservative, even if it fails we may still have a stable closed-loop interconnection for all possible spring stiffness constants. To check *a posteriori* for robust stability the MATLAB command `isstable` can be used, which returns `true` for a stable closed-loop interconnection and `false` otherwise.

4.1 Frequency Domain

In order to check the stability margins of the uncertain closed-loop system the MATLAB command `allmargin` can be used. For a graphical interpretation it is also meaningful to look at the Nyquist plots generated for different combinations of springs. Such Nyquist plots are shown in Fig. 16, where an $\mathcal{H}_\infty$ suboptimal controller K has been designed as discussed in the previous section.

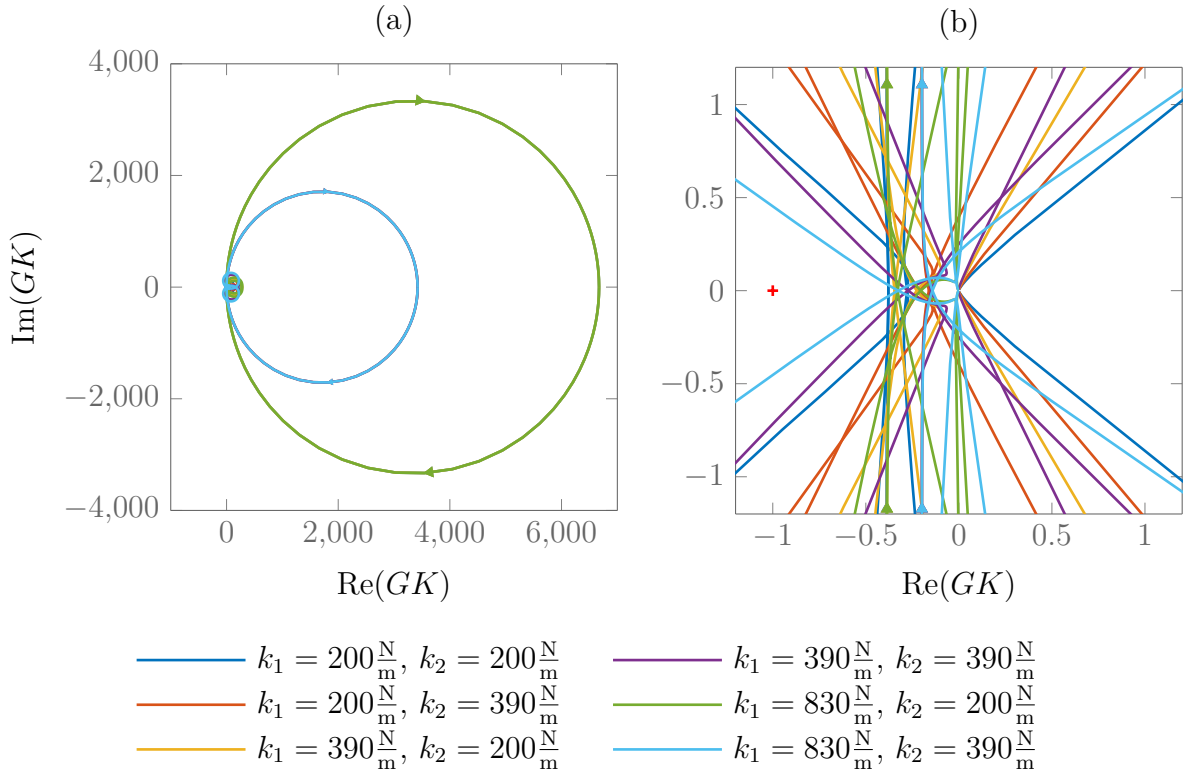


Figure 16: Nyquist plot of the closed-loop system for different spring stiffness

Comparing these results to Fig. 3, it becomes apparent that robustness margins as gain and phase margins have been improved and the controller K robustly stabilizes the plant G.

Prep. 4.1: For which springs k_1 and k_2 does the controller K yield the lowest phase margin?

Additionally, the sigma plots of the transfer functions S, KS, SG and T_K/T give valuable insights into the closed loop properties. One way to compute these transfer functions is to

use the MATLABs `sysic` command. For example to compute the closed-loop sensitivity transfer function S we can use the following code:

```
systemnames = 'G K';
inputvar    = '[r{1}]';
input_to_G = '[K]';
input_to_K  = '[r-G]';
outputvar   = '[r-G]';
S = sysic;
```

Prep. 4.2: Think about how to compute the remaining transfer functions KS, SG and T without using the `sysic` command.

Fig. 17 shows sigma plots of the four transfer functions mentioned above for all possible combinations of springs.

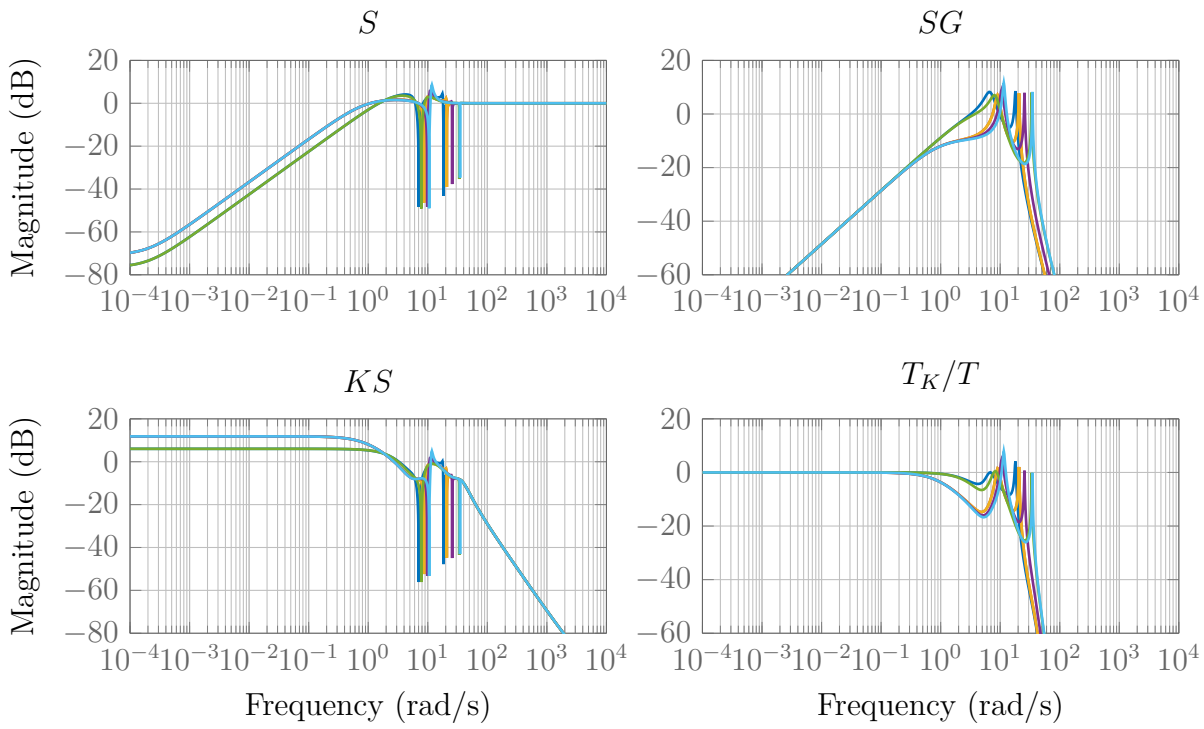


Figure 17: Typical sigma plots of the transfer functions S, SG, KS and T_K/T for S/KS design

In order to achieve good steady state accuracy we have to make sure that the sensitivity is small at low frequencies. Considering a low frequency reference input $r(t) = \sin(\omega_l\, t)$ with magnitude of 1 and some frequency $\omega_l \leq 10^{-4}$ rad/s we can see that the steady state error will be less than $-60\,\mathrm{dB}$. Furthermore, KS shows controller roll-off at high frequencies which is of dire importance since with higher frequencies model inaccuracy will inevitably increase.

Prep. 4.3: Think about which further information can be read off the four sigma plots.

4.2 Time Domain

We are interested in designing a controller that achieves good tracking performance. Looking at the frequency response of the open-loop plant shown in Fig. 2 it is intuitively clear that a robust controller of the order of the generalized plant can not cancel all poorly damped modes which can occur due to changing springs. Moreover, it is also unlikely that pole-zero cancelations occur at all which is a consequence of the uncertainty description. However, this is not the case for the nominal S/KS design. In this case the resonant frequencies are assumed to be known precisely such that controller zeros may cancel plant poles, yielding better tracking performance. Note that this price has to be paid for with reduced performance regarding input disturbance rejection. First, we look at the simulated step response for the nominal design shown in Fig. 18.

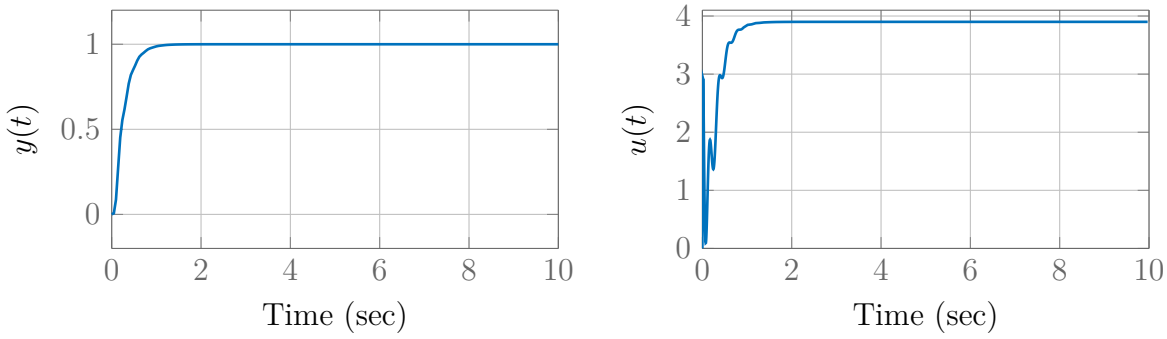


Figure 18: Step responses and control effort – Nominal design

It can be seen that resonances are not excited by the controller and quite a low rise time with no visible overshoot is achieved. Regarding the robust design and in order to really reduce the resonant peaks in S controller roll-off can be used. However, this will inevitably slow down the controller. Consequently, there will be a visible trade-off

between not exciting resonances and speed of response. Figures 19 and 20 show step response as well as the control effort $u(t)$ for $r(t) = \sigma(t)$ for two different controller designs illustrating the fact mentioned above.

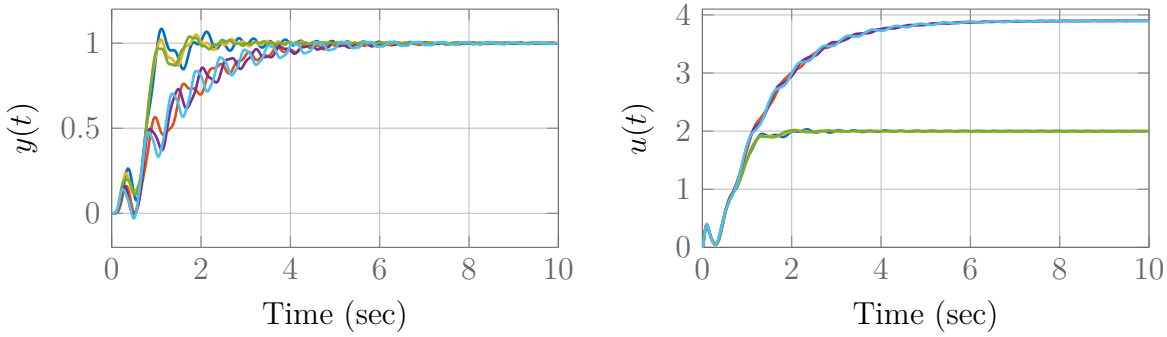


Figure 19: Step responses and control effort – Design 1

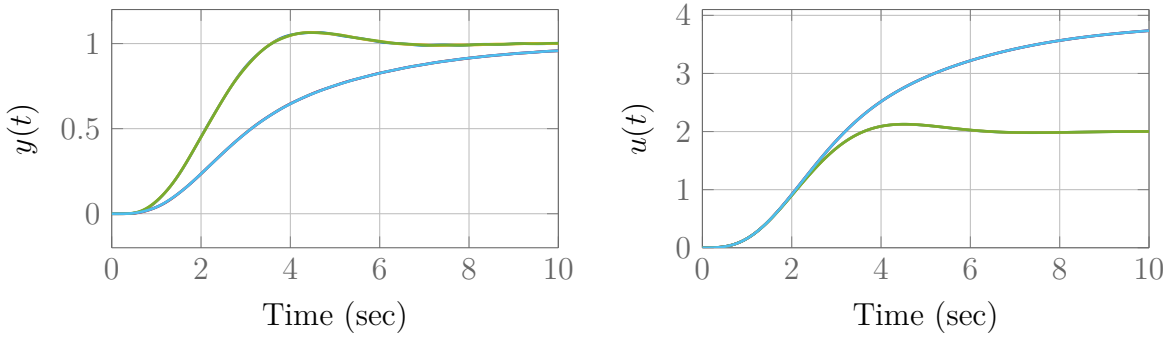


Figure 20: Step responses and control effort – Design 2

One problem which may occur using the S/KS controller design approach is that the controller $K(s)$ cancels lightly damped poles of $G(s)$. However, although the transfer functions S and KS may look fine, the transfer function SG can show undesirable peaks. This results in a poor response if input disturbances are applied. To assure us that undesirable pole zero cancellations do not occur we can plot the pole zero map of $G(s)$ and $K(s)$ using the MATLAB code

```
pzplot(G, 'b', K, 'r');
```

A plot of the pole zero map is shown in Fig 21 for the nominal system $G_0(s)$ and a controller $K(s)$.

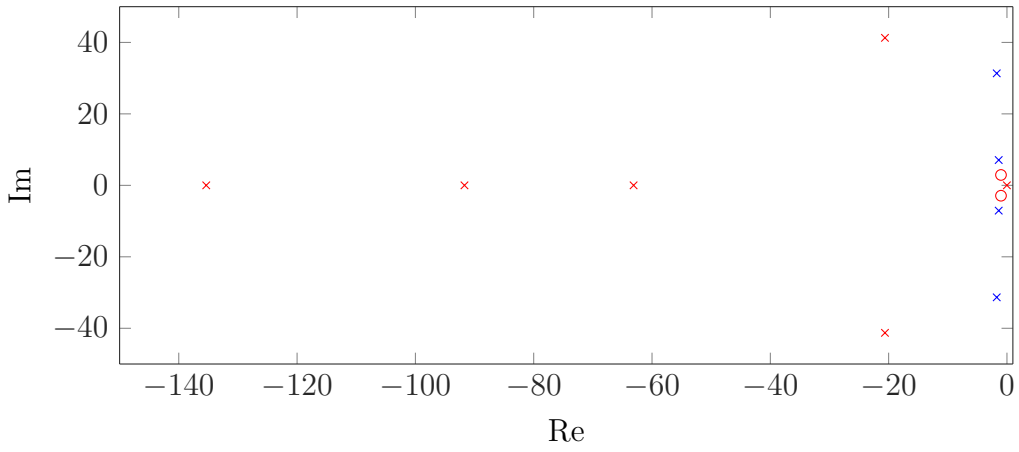


Figure 21: Pole zero map of $G_0(s)$ (blue) and $K(s)$ (red)

In fact due to the uncertain spring stiffness and therefore uncertain pole locations of $G(s)$ it is unlikely that the controller tends to cancel poles as becomes apparent here.

Furthermore, instead of looking at the closed-loop transfer functions in frequency domain we can analyze the closed-loop pole zero map by comparing it to the open-loop pole zero map of $G(s)$. As we want to improve the damping in closed-loop we would expect that the closed-loop poles in the complex plane are shifted to the left with respect to open-loop poles of $G(s)$. The relevant information of the pole zero map of the nominal plant $G_0(s)$ and the nominal complementary sensitivity are $T_0(s)$ is shown in Fig. 22.

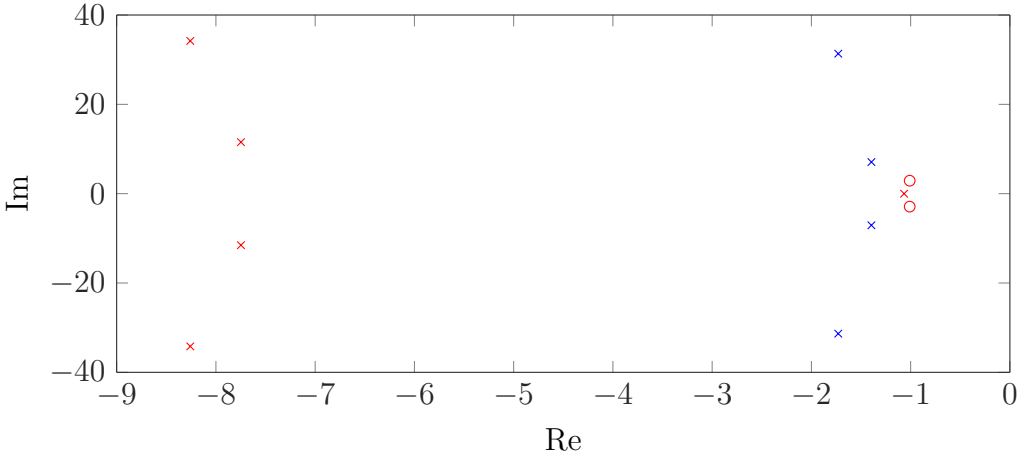


Figure 22: Pole zero map of $G_0(s)$ (blue) and $T_0(s)$ (red)

In fact damping is significantly improved by the designed controller and pole zero cancellations do not occur which is consistent with the shape of SG.

4.3 How to proceed

Take a look at the provided MATLAB file `orc1_design.m` and complete it in a way that allows you to design your own controller for nominal performance and for robust stability. Tune your controllers iteratively by adjusting the weights, scaling the uncertainty and shifting the nominal plant model to achieve the requirements given in the task description. For simulation in SIMULINK the file `orc1_simulation.slx` is provided.

5 Real-Time Experiments

During the lab, the file `orc1_interface.slx` will be provided to interface the spring-mass-damper plant. Its structure is depicted in Fig. 23. As you can see, it looks like the Simulink file used for simulation.

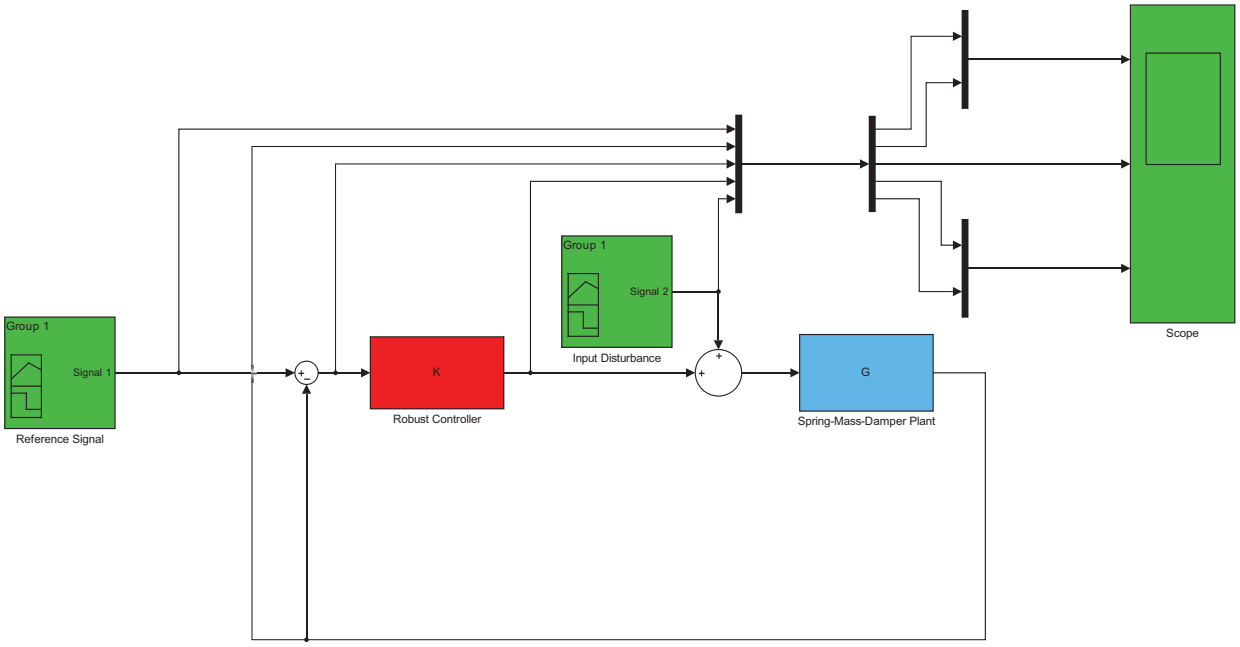


Figure 23: Simulink model: Plant, Controller and Interface

The blue block represents the spring-mass-damper plant and contains interfaces to the Digital/Analog and Analog/Digital converters of the physical plant. The inputs and outputs of the block are the same as in simulation.

The red block is identical to that used in the simulation.

The green blocks are identical to that used in the simulation except that the output structure is now named `expdata`. Code for plotting this data will also be provided.

5.1 The Experiment

Proceeding to the experiment and implementing the designed controller will reveal differences between simulation and experiment. This can be expected and is due to model errors and unmodeled dynamics. Experimental closed-loop step responses are shown in Fig. 24 for the nominal design as well as for the robust design evaluated on the nominal plant.

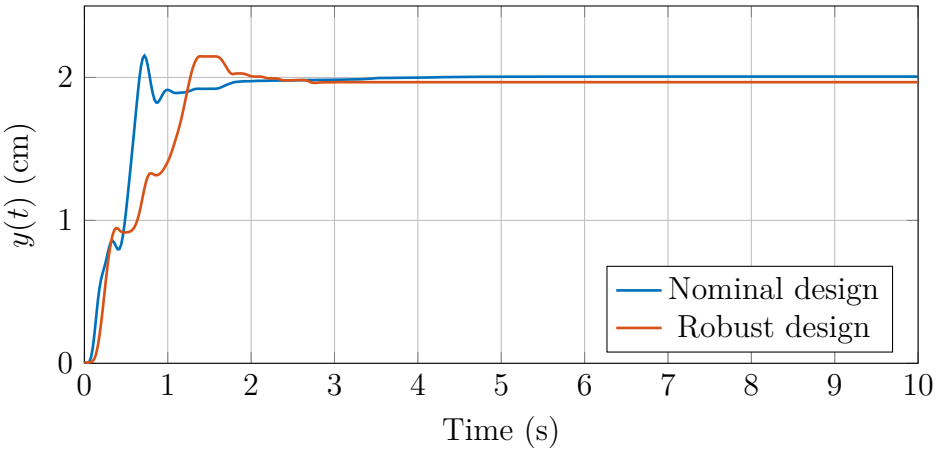


Figure 24: Nominal vs. robust design for $k_1 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$, $k_2 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$

It can be seen that the nominal design outperforms the robust design in terms of rise time and steady state error. Moreover, comparing these results to simulation results shown in the previous section it becomes apparent that viscous friction is underestimated by the values $c_1 = c_2 = 0.01\,\frac{\mathrm{N\,s}}{\mathrm{m}}$. In addition, the plant is subject to nonlinear stick slip friction resulting in flat plateaus visible in the responses which may yield limit cycles due to high loop gain at low frequencies. This is however not the case here. The Figures 25 and 26 depict experimental step responses for all combinations of springs and show that the robust design indeed stabilizes the plant robustly with worst performance for $k_1 = 200\,\frac{\mathrm{N}}{\mathrm{m}}$ and $k_2 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$.

5.2 How to proceed

- Adjust the damping coefficients c_1 and c_2 of the simulation model such that the simulated step response of the nominal plant model with $k_1 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$ and $k_2 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$ shows approximately the same damping when compared to the experimental step response.

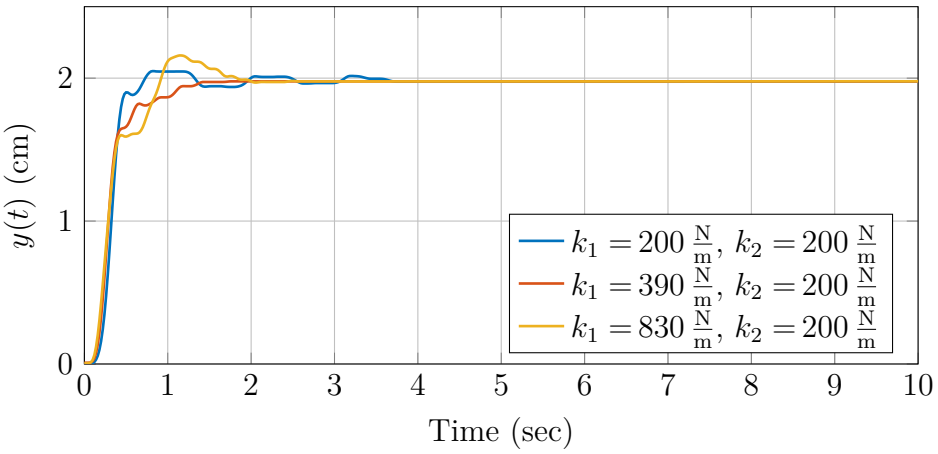


Figure 25: Robust design for $k_2 = 200\,\frac{\mathrm{N}}{\mathrm{m}}$

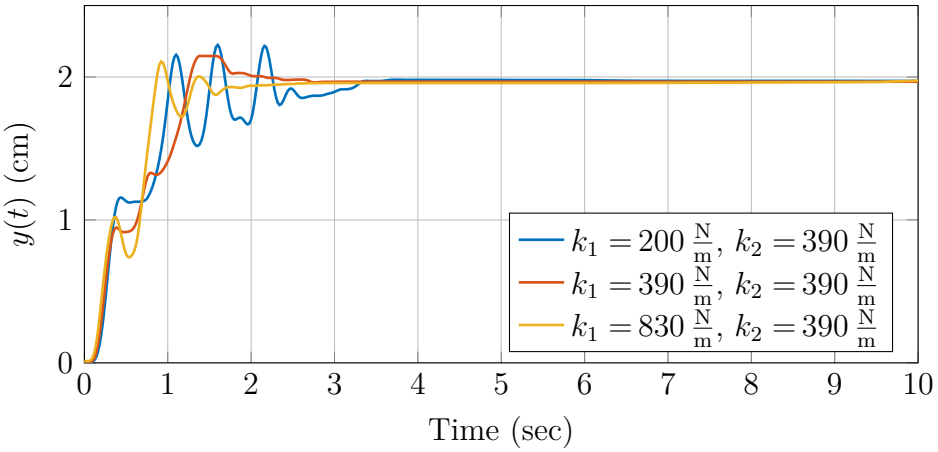


Figure 26: Robust design for $k_2 = 390\,\frac{\mathrm{N}}{\mathrm{m}}$

- Retune your controller for nominal performance tested in simulation such that all requirements are met in the experiment for the nominal plant.
- Test the same controller with combinations of springs different from the nominal plant model.
- Retune your robust controller tested in simulation such that all requirements are met in the experiment for all combinations of springs.