

# Control Systems Lab (SC4070)

## Laboratory Sessions

### Overview

The laboratory sessions are an essential part of this course. The purpose is to gain hands-on experience with the control of a real physical process. You will be working in groups of two students. After you have chosen a partner to work with and a laboratory setup, you should proceed along the following lines:

1. Implement a nonlinear, physical model of the setup in Simulink as your home preparation. Prepare an identification experiment to identify a black-box model of the system (or its part) and test it with the physical model.
2. In the lab: calibrate the physical model to match the process. Carry out the identification experiment, collect the data.
3. Analyze the results from the previous step, design a controller for the simulation model as your home preparation.
4. In the lab: test and fine-tune the controller on the process.
5. Present the results – one presentation per group, one report per group.

More details about the above steps are given in the subsequent section of this text. Week 1 is the first week of the quarter. One lab session takes three to four hours. If the schedule permits, you may, of course, spend more time in the lab. Let us stress that it makes little sense to come to the lab sessions unprepared. For this course it holds perhaps more than for other ones that what you get out of it is directly proportional to what you yourself put in.

### Experimental Setups

Each experimental setup consists of a hardware system (inverted pendulum, helicopter, inverted wedge or rotational pendulum), a computer running MATLAB and Simulink, and the interfacing hardware (amplifiers, A/D and D/A converters), see Figure 1.

Simulink interacts with the hardware in real time by reading the measured system's output  $y$  and commanding the control input  $u$  that drives the corresponding actuator (motor, pump, etc.). Both  $u$  and  $y$  can be multi-dimensional signals (vectors). The corresponding real-time input and output Simulink blocks are provided, you only need to define

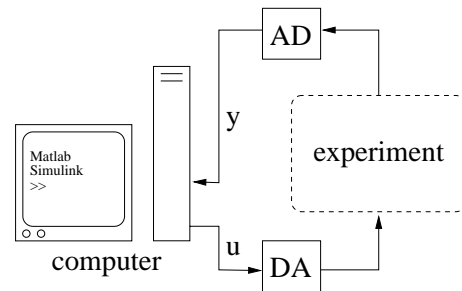


Figure 1: General scheme of an experimental setup.

a proper sampling period  $h$  in the MATLAB workspace. Table 1 shows the signals  $u$  and  $y$  for the individual experimental setups.

Table 1: Inputs and outputs for the individual setups.

Experiment	$u$	$y$
Pendulum	motor voltage	cart pos., pend. angle
Balance	motor voltage	cart pos., balance angle
Helicopter	motor voltage	arm angle, prop. speed
Rot. pend.	motor voltage	arm and pendulum angle

In Simulink, these signals can be displayed in graphical scopes and can also be exported to the Matlab environment (using the scopes themselves or the “To Workspace” block) and saved. A digital controller will be implemented also in Simulink, typically using blocks from the libraries “Discrete” and “Control System Toolbox”.

The sensor readings  $y$  are expressed in their respective physical units (e.g., meters, radians, etc.). The actuator command  $u$ , however, is always normalized between -1 and +1 (or 0 and +1), where 1 corresponds to 100% of the available range.

### Design Procedure

Each experimental setup comes with a certain control task, e.g., ‘balance the pendulum while making the cart follow a specified reference trajectory’. You may choose to design a pole-placement controller (output feedback) controller or a PID controller. It is recommended to use output-feedback control, as this approach is more systematic and works for all setups without using special schemes (e.g., cascade control). For either choice of the controller, the design procedure

cedure is roughly the following:

1. Implement a Simulink model and find the unknown system parameters by using simple (but well thought of) experiments. The response of your simulation model should reasonably fit the measured data. Store some typical results for your presentation.
2. Linearize this model around an operating point. Even though our simple lab processes can easily be linearized by hand, it is recommended to implement the linearization procedure as a MATLAB script (using functions `trim` and `linmod`), such that the controller can easily be re-designed for different operating points.
3. Choose design parameters (closed-loop natural frequency, damping, etc.) for the controller and observer (if used). Design the controller (using, e.g., `place` or `acker`) and implement it in Simulink.
4. Check the controller with your models (nonlinear and linearized). If the response of the system is not satisfactory, change the controller parameters and try a new controller. Store some typical results for your presentation.
5. Only after the simulated closed-loop response meets the requirements, test the controller on the actual system. Because of some model-plant mismatch, you may need to fine-tune the controller by changing the design parameters. Therefore, the design procedure should be implemented as an easy-to-use MATLAB script or function. Store some typical results for your presentation.

Compare the results with your expectations after each step, and if the results are not satisfactory, go back a step (or more). Since all the computations are done by MATLAB, it is easy to try a different setting and quickly compute the corresponding controller to check what happens ('rapid prototyping').

## Simulation Model

For mechanical systems, the model equations are typically given as (pairs of) second-order nonlinear differential equations of the form

$$\ddot{x} = f(x, \dot{x}, u)$$

where  $x$  is the generalized position (angle, horizontal position, etc.),  $u$  is the input, and  $f(\cdot)$  is some static function parameterized by several (physical) parameters. Equations of this type can easily be translated into a Simulink model such as the one shown in Figure 2.

As an example, consider a mass sliding across a surface with viscous friction (coefficient  $d$ ), pushed by a certain (known) input force  $F$  (Figure 3).

The model equation for this system is

$$\ddot{x} = \frac{1}{m}(F - d\dot{x}) \quad (1)$$

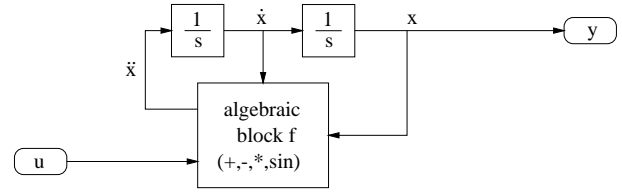


Figure 2: General model structure in Simulink.

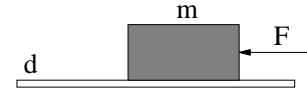


Figure 3: Sliding mass with viscous friction.

where  $x$  is the position of the mass. This equation is in the required form and can directly be implemented in a Simulink model shown in Figure 4, where the position of the mass is the output of the system.

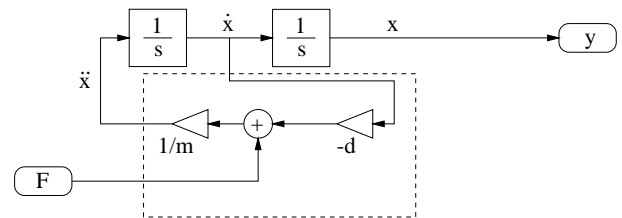


Figure 4: Simulink model for the sliding mass.

## Parameter Identification

Each experimental setup has certain parameters that need to be identified by means of experiments. These experiments are not prescribed, you should figure them out yourself.

One possible way to come up with a useful experiment is the following. Analyze the model equations and try to eliminate most of the terms in the equation (e.g., by assuming derivatives equal to zero). Then, devise an experiment that satisfies these assumptions and collect data. Finally, compute the value of the parameter using the reduced equations that apply to that particular experiment.

As an example, let us identify the friction coefficient for our sliding mass. By assuming  $\ddot{x} = 0$  (or  $\dot{x} = \text{const}$ ), equation (1) reduces to

$$F = d\dot{x}.$$

Hence, when we apply a constant force, plot the resulting position as a function of time, and look at the part of the plot where  $\ddot{x} = 0$ , the friction coefficient follows easily as the ratio of the force and the resulting velocity.

Each experiment should be repeated several times to see whether it is reproducible and to obtain a reliable estimate of the parameter value.

**Note:** Always start experiments with small actuator values (e.g., 0.2) in order not to damage the setup.