Experiments X

Experiments on the motion setup

Make sure to read the SPERTE documentation on Canvas first!

X.1 Introduction

An essential part of the Control Engineering course involves gaining hands-on experience by working with the experimental setup depicted in Figure X.1. This setup will also be used for the final assignment. This document will introduce the hardware, and will describe what is expected from you during the experiment sessions.

X.1.1 The hardware

The setup consists of two rotating masses, with a flexible joint in between; it is often referred to as the *4th-order motion setup*. One of the masses is connected to a DC-motor, but each mass is connected to an incremental encoder (with 2000 counts per revolution) to measure the actual rotation. As such, one sensor is co-located with the motor, the other is on the

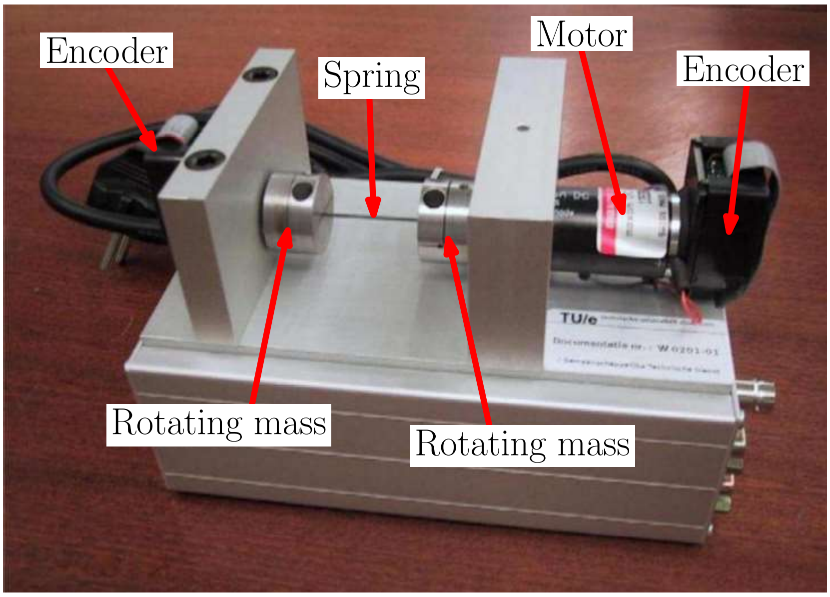

Figure X.1: Motion setup used for the experiments.

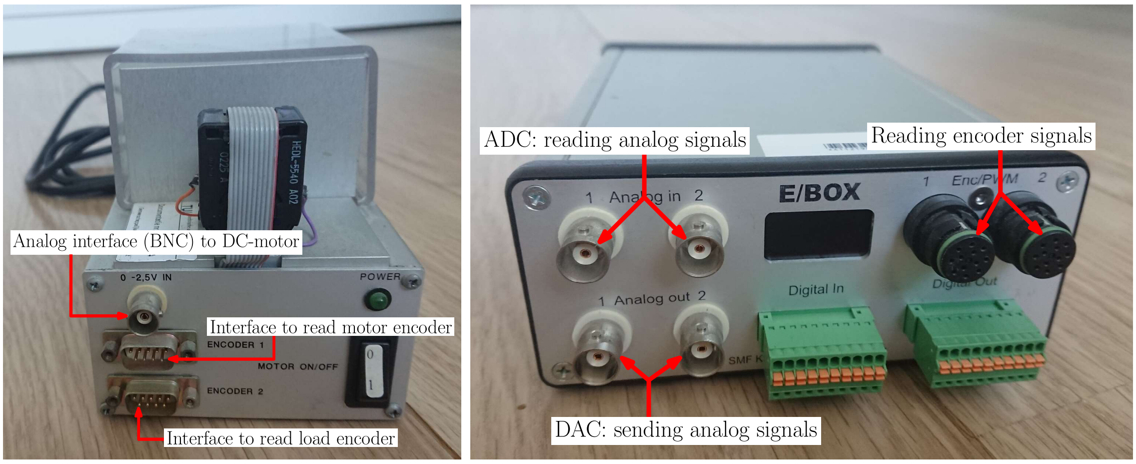

Figure X.2: Interfaces on the 4th-order motion setup (left) and the e-box data acquisition system (right).

load-side (i.e. non-collocated); feedback control using the first sensor will thus be referred to as *motor feedback*, whereas feedback control using the non-collocated sensor will be referred to as *load feedback*.

The whole assembly is mounted onto a current amplifier; the voltage sent to this amplifier is scaled linearly into a current sent to the motor, which scales linearly with the torque that is then applied by the motor. The electrical interfaces to the setup are provided on the side of the setup, as shown on the left of Figure X.2, by means of a BNC connector for the voltage towards the current amplifier (and thus the motor), and two 9-pin D-sub connectors to read the encoders.

Note: for some setups the numbering of the encoders is a bit messed up; make sure that for your specific setup you convince yourself (e.g. with a simple test) which connector represents the sensor on the motor-side, and which one the load-side!

To communicate with this setup the e-box data acquisition system will be used, shown on the right of Figure X.2. This system provides 2 ADCs (Analog to Digital Converters) and 2 DACs (Digital to Analog Converters); the latter can be used to actuate the motor, the former could be used to verify this actuation signal (e.g. by directly connecting the DAC to an ADC). Moreover, it provides two interfaces to read the encoder outputs, and some additional digital I/O which are not used in this course. At the back of the e-box there is an EtherCAT port to connect it to the SPERTE Raspberry Pi.

More details on connecting this hardware to your own pc, interfacing with it, and installing the required software can be found in the SPERTE documentation on Canvas. Make sure to carefully read this documentation before your very first experiment session!

X.1.2 Overview of experiments

During this course, you will have several opportunities to work with this hardware. In the first few weeks you will get acquainted with it, by carrying out numerous exercises (see section X.2) in which you learn to apply and understand the course material, supported by the help and assistance of our TAs. You are not obliged to make all exercises, but we strongly recommend you do, since it is the best preparation for the next phase.

During the last few experiment sessions you will namely use the same hardware for the final assignment of this course (see section X.3). Since your end grade will be based on the report you will write about this final assignment, supervision will be minimal during these sessions (limited and restricted to e.g. solving hardware issues); content-wise you are expected to be able to make your own decisions and value the quality of your own results by then.

X.2 Practical exercises

X.2.1 Time-domain tuning

Connect the 4th-order motion setup for the motor-feedback case, i.e. using the motor encoder as the plant output. Open the reference Simulink model (see the SPERTE documentation: `Reference_model_Control_Engineering`), and make sure to set the reference to zero (just accept any 'Ref3' trajectory, and then make sure that the red block on the left is turned 'off').

a. Set the controller to zero, i.e. start with $K_p = K_d = 0$. Then slowly and **carefully**[1] increase K_p; while you do so, touch the inertia on the motor side and carefully perturb it. Can you "feel the stiffness", and feel how K_p influences it?

b. Again start with $K_p = K_d = 0$, but now **carefully**[2] increase K_d, while touching and perturbing the motor inertia. Can you "feel the damping" and the influence of K_d?

c. Alternately increase K_p and K_d, aiming for a fast as possible response (settling time), without the system getting unstable. Feel the influence of your changes. Do some time-tuning this way to find a good as possible PD-controller (and feel what it does[3]), and write down your 'optimal' values for K_p and K_d.

d. Assuming your plant is a simple inertia, and your controller acts like a spring-damper, is it theoretically possible for the closed-loop system to become unstable for some values of K_p and K_d, or could they be arbitrarily large? Does this correspond to your observations? Explain why (not).

X.2.2 Frequency response measurements

Connect the 4th-order motion setup for the motor-feedback case, i.e. using the motor encoder as the plant output. Choose reasonable settings for K_p and K_d, stabilizing the closed loop, but not too aggressive. Set the reference to zero.

a. Determine the plant FRF in a closed-loop setting, using a sampling frequency of 4000 Hz. Choose either a 2-point or 3-point approach, or do both and compare the results. Play around with various settings and conditions (measurement time, frame length, etc) to get a satisfying measurement quality.

[1] In some extreme and exceptional unstable cases the axle between the inertias could break, shatter or even be swung away, which is why every setup has a protective cover. When doing this exercise *carefully*, always put on this cover first, change a parameter, observe whether the system remains stable, and if so, remove the cover and perturb the motor inertia with your hand. Then put the cover back on before you make another parameter change.

[2] See the previous remark: keep the protective cover on between parameter changes, and don't touch the inertias when things start to look unstable.

[3] Again, make sure to keep the protective cover in place while tuning!

b. Decrease the sampling frequency to 2000 Hz and 1000 Hz respectively and again determine the plant FRF with a 2-point and/or 3-point method. Compare the results of the different sampling frequencies. Explain what you observe.

c. Repeat your best measurement while bringing the system into motion, in order to investigate the effect of dry friction and possible other small non-linearities. Therefore choose a ramp-like setpoint (select 'jogmode' with an appropriate velocity in the 'Ref3' block, and turn the red block 'on') and collect new data. If desired, repeat for a few different velocities. Compare the obtained plant FRFs, do you observe any differences?

d. Repeat your best measurement for a few different controller settings, e.g. your initial reasonable K_p, K_d, your 'optimal' PD-controller (see exercise X.2.1) and a very sloppy low-stiffness K_p, K_d. Determine the actual open-loop FRFs $H \cdot C$, analyze the actual bandwidth and margins, and compare the quality of the obtained plant FRFs.

e. Set the reference and the controller to zero, and measure the plant FRF directly, in open loop. Compare the result with the closed-loop measurements.

X.2.3 Feedback and feedforward tuning

Connect the 4th-order motion setup for the motor-feedback case, i.e. using the motor encoder as the plant output.

a. Use your best FRF result (see exercise X.2.2) to tune an appropriate feedback controller in the frequency domain, e.g. using ShapeIt. Constrain yourself to using only a lead filter and a low-pass filter, and aim for a bandwidth of around 10 Hz, while satisfying $\max_\omega ||S(j\omega)|| < 6$ dB. Compare its Bode and Nyquist plots, bandwidth and margins with the PD-controller from exercise X.2.1.

b. Replace the default controller in the Simulink file by your new design (you can use the 'Save to Simulink' button in ShapeIt if you like). Do a quick 2-point FRF measurement to prove that your implementation has the same open-loop response $H \cdot C$ as designed.

c. Define an appropriate third-order setpoint trajectory and monitor the tracking error. Make sure you can clearly distinguish acceleration, constant-velocity and deceleration blocks in the error.

d. Minimize the closed-loop error by adjusting the feedforward gains K_{fc}, K_{fv} and K_{fa}. Determine the level of improvement that feedforward gives for this specific setpoint.

e. Choose a (significantly) different controller and/or setpoint, and show that the previously determined feedforward gains K_{fc}, K_{fv} and K_{fa} still minimize the error.