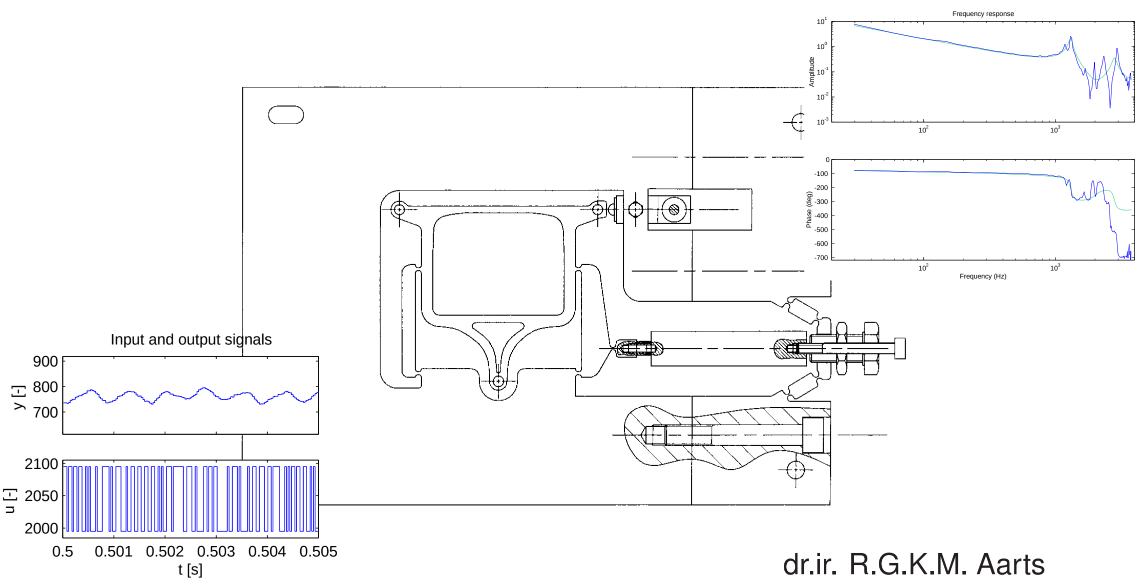
UNIVERSITY OF TWENTE.

Faculty of
Engineering Technology

Applied Mechanics and Data Analysis

System Identification with Parameter Estimation and Machine Learning

exercises



dr.ir. R.G.K.M. Aarts
prof.dr.ing. B. Rosic

Edition: 2022/2023

Course: System Identification with Parameter Estimation
and Machine Learning (202200111, ROB / ME / S&C)

AMDA

General remarks for all exercises

Exercises 1–5 are for the first part of the course. You are asked to prepare these exercises prior to a lecture as announced on CANVAS. At the start of this lecture the exercise will be discussed. In this way you should be optimally prepared for the individual exam of this course.

The remaining exercises can be made as assignments in the second part of the course. Additional information about the procedure is given on page 10.

Some of the exercises start with a list of MATLAB commands that may be helpful to complete the exercise. In these overviews only the short descriptions of the commands are given. For all commands extensive on-line `help` and/or `doc`umentation are available in MATLAB. Note also that usually commands are introduced only once, so the lists are cumulative. Furthermore, the exact syntax of commands and terms used in (graphical) user interfaces are sometimes updated in newer MATLAB versions. Hence, the command lists and hints given in the exercises may not apply to older (or the most recent) MATLAB versions.

Please save all relevant output generated during MATLAB sessions. In general this may help you to re-examine results obtained previously. Furthermore, some exercises continue with the results from one or more previous exercises.

Exercise 1 Time and frequency domain, impulse and Bode response

The following MATLAB commands *may* be helpful to complete this exercise:

`abs`	Absolute value and complex magnitude.
`angle`	Phase angle.
`bode`	Bode frequency response of LTI models.
`bodeplot`	Plot Bode frequency response with additional plot customization options.
`c2d`	Conversion of continuous-time models to discrete-time.
`fft`	Discrete Fourier transform.
`freqresp`	Frequency response of dynamic systems.
`gensig`	Periodic signal generator for time response simulations with `lsim`.
`impulse`	Impulse response of LTI models.
`loglog`	Log-log scale plot.
`lsim`	Simulate time response of LTI models to arbitrary inputs.
`plot`	Linear plot.
`semilogx`	Semi-log scale plot.
`sin`	Sine of argument in radians.
`stairs`	Stairstep plot.
`tf`	Creation of transfer functions or conversion to transfer function.
`unwrap`	Unwrap phase angle.
`zpk`	Create zero-pole-gain models or convert to zero-pole-gain format.

We consider the continuous-time system $G_c(s)$ with zeros $z_i = 3 \pm 100i, -141 \pm 141i$ and poles $p_i = -80, -10 \pm 60i, -45 \pm 120i$. The gain $k = 25$ is defined as in MATLAB's `zpk` command. Furthermore, we consider two discrete-time systems $G_{d1}(z)$ and $G_{d2}(z)$ that are found by applying zero-order hold (`zoh`) discretization on the inputs of system G_c and with sample times 0.01 s and 0.04 s, respectively.

Initialization: To define the continuous-time system $G_c(s)$ in MATLAB we start by defining the system's zeros, poles and gain. For this purpose, a small initialization script `ex1.m` can be downloaded from the CANVAS page of this course. Alternatively, the commands can be types manually at the MAT-

LAB prompt as:

```
%% Initialize variables of system Gc(s):

z = [       3+100i;  3-100i; -141+141i; -141-141i ];
p = [ -80; -10+60i; -10-60i;  -45+120i;  -45-120i ];
k = 25;
```

Edit and extend this script with MATLAB commands like `tf` and `zpk` to create MATLAB variables e.g. to represent the transfer function $G_c(s)$ of this system. Next the `c2d` command can convert the continuous-time system into the two discrete-time systems.

Simulations: Consult MATLAB's `help` or `doc` of the `lsim` command to compute the responses of all three systems G_c, G_{d1} and G_{d2} to an harmonic input, i.e. `sin`e function. Consider three frequencies of 5 Hz, 10 Hz and 15 Hz. Do not confuse frequencies (in Hz) and angular frequencies (in rad/s).

For the continuous-time simulation approximate the input with about 100 time steps per period. For each frequency compute the response for at least 4 periods. More specifically, from each simulation it should be possible to determine the frequency response function at the frequency of the input signal.

For both discrete-time systems choose the time steps in agreement with the documentation of the `lsim` command.

a. Present the results of your simulations. Compare the simulations of both discrete-time systems with the continuous-time system and comment on notable differences.

   Hint: The `stairs` command can be used to plot discrete-time signals.

b. Plot the combined Bode frequency response of all three transfer functions. Check the largest frequency shown in the graphs of the discrete-time transfer functions. What are other notable differences between the graphs?

   Hint: Use a single `bode` or `bodeplot` command to combine Bode plots in one graph for easy comparison.

c. Determine the frequency response functions for all three systems (G_c, G_{d1}, G_{d2}) and for all three frequencies of 5 Hz, 10 Hz and 15 Hz in two ways:

   1. Analyze the input and output signals in the simulations (item a). For the discrete-time systems you only need to investigate the gain and you don't have to consider the phase.

      This analysis does not have to be exact with high precision, but you should explain clearly how you analyze the data.
   2. Determine the amplitude ratios and phase shifts according to the Bode frequency response (item b).

      You can read this information from the Bode plots, but it may be more convenient to compute the requested values directly with MATLAB's `bode` or `freqresp` commands at the specified frequencies.

   Do the results agree? If not, can you explain striking differences?

d. Use MATLAB's `impulse` command to compute impulse responses $g_{d1}(k)$ of the discrete-time system G_{d1}. Consider at least two different lengths of these responses. Apply the `fft` command to compute the Fourier transforms of these responses. Do these Fourier transforms agree with the Bode response plot of G_{d1} in item b?

   Hint: The length of the impulse response computed by the `impulse` command can be modified with the optional parameter `TFINAL`.

The discrete-time transfer function G_{d1} will be used again in the next exercise.

Exercise 2 Non-parametric system identification

Will be published later ....