System Identification – Practical Assignment 2
Transient Analysis of Step Responses

Logistics

- Development: This practical assignment should be carried out by each student separately, on the Matlab Grader platform. You should have received an invite to the Grader platform on the email address communicated to the teachers. The assignment solution consists of Matlab code. Develop this code in a single Matlab function. You can pretest your solution as many times as you want, to see if it works.
- Submission: Only after you are satisfied with your solution, it must be submitted in Matlab Grader. Do this only once; in case of mistakes, you have a second chance to submit, but this should be a last resort.
- Verification: The code will be checked both by Grader automatically, and by the teacher, including a plagiarism check against the code of all your colleagues. Your attendance to the lab will only be registered if you have a working, original solution. Validated attendances for all the labs are necessary for eligibility to the exam. Moreover, at most two labs can be recovered at the end of the semester, which means accumulating three or more missing labs leads to ineligibility.
- Other remarks: Discussing ideas amongst the students is encouraged; however, directly sharing and borrowing pieces of code is forbidden, and any violation of this rule will lead to consequences as described in the discipline rules.

Assignment description

In this assignment we will perform transient analysis based on nonparametric, step response models – see the course material, Part II: *Step and Impulse Response Graphical Models*. We will do this for both first-order and second-order systems.

You will develop a function with the exact signature:

```
[index, K1, T, MSE1, K2, xi, omega, MSE2] = transient_step(index)
```

Each student is assigned an index number in the set 1-8, which needs to be saved to variable `index` in the first line of the function. The index dictates which data files the student should load. For instance, if you have index 4, you load file `lab2_order1_4.mat` for the first-order exercise, and `lab2_order2_4.mat` for the second-order exercise. All these datafiles are already accessible from your function code, they have been uploaded to the Grader problem (even though they are not visible explicitly).

The first file contains several step inputs signals and the response of a first-order system, and the second contains similar data for a second-order system. The data is provided as an object called `data` of type `iddata` from the system identification toolbox, see `help iddata`. For convenience, a separate variable `t` holds the time vector of the experiment. Every dataset contains *five* consecutive step signals, each corresponding to 100 time steps; see the figure. The initial conditions are zero. The first step should be used for identification; the second step simply brings the system back to zero initial conditions, so that you do not have trouble with initial output mismatches due to nonzero initial conditions; and the last three steps should be used for validation.

Work on the following requirements:

- Develop a transfer function model of the first-order system with the method described in Lecture 2, using the *first* step signal and response from the data. Compute the gain and the time constant in variables `K1` and `T`, respectively.

- Validate your model using steps three to five (this is the validation data). The validation should consist of: (a) a plot where the system output is compared with the model output on the same graph; (b) and the computation of the MSE, returned in variable `MSE1`. Use Matlab function `lsim` to simulate the system response to the validation input.

- Next, develop a transfer function model of the second-order system, again using the first step signal and response from the data. Compute the gain, damping factor, and natural frequency in variables `K2`, `xi`, and `omega`, respectively.

- Validate your second-order model using steps three to five (this is the validation data). As before, the validation should consist of: (a) a plot where the system output is compared with the model output on the same graph; (b) and the computation of the MSE, returned in variable `MSE2`.