ME 4022 Homework 4

In this homework you will perform transient analysis based on nonparametric, impulse response models – see the course material, Part 7: Step and Impulse Response Graphical Models. You will do this for both first-order and second-order systems.

There are two files for each student: the first contains several impulse 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. Each experiment begins with 30 initial steps where the system is in its initial, steady-state condition, after which three consecutive impulse-response experiments are performed, each corresponding to 100 time steps. Keep in mind that the initial conditions are nonzero.

Your homework solution must consist of a Matlab code! Develop this code in a single Matlab script. Please write your code in a self-explanatory fashion (adding comments where necessary), so as to make it understandable on its own.

Requirements (apply this procedure first for the first-order system, and then for the second-order one):

- Develop a transfer function model of the system using the method described in the lecture Part 7, using the first impulse signal and response from the data. Include instructions that print out the transfer function, as well as relevant intermediate values (e.g. overshoot M, oscillation period T_0 in the case of second order systems) at the console when your script is run.

- Validate your model using the data for the second and third impulse responses (this is the validation data). Simulate the system from the correct non-zero initial condition; to this end, create a state space model using Matlab function `ss`. 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. Both of these results should be automatically produced by the Matlab code you provide. See function `lsim` to simulate the system response to the validation input, and investigate how you can provide the initial condition to this function.

- Compare the value of damping ratio that you have obtained for the second order system with the damping ratio that you can obtain by *logarithmic decrement method*. Comment on the differences, if any.

A programmatic, code-based solution for the identification of the intermediate values (e.g. M, K, T_0) is more general and elegant. However, it is not strictly necessary, and you can also perform this step "by eye" on the graph. Especially if you see that you are running out of time, it is recommended that you apply this second, simpler solution.

One step that has to be performed programmatically in any case is the numerical computation of the areas for the second-order impulse response as shown in the lecture. *Hint*: always keep in mind the difference between continuous time and the corresponding indices of discrete-time steps; and watch the signs of the integrals.

Some relevant Matlab functions: `ss, lsim, find, sum.`