System Identification – Practical Assignment 6
ARX model identification

Logistics

- This practical assignment should be carried out by each student separately, if at all possible. Otherwise, if there are more students than computers, students may team up in groups of 2.
- The assignment solution consists of Matlab code. Develop this code in a single Matlab script. This code will be checked and run by the teacher during the lab class, and 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.
- 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 disqualification of the solution.

Assignment description

In this assignment we will identify ARX models (autoregressive with exogenous input), using least-squares, linear regression. See the course material, Part V: *ARX Identification*.

Each student is assigned an index number by the lab teacher. Then, the student downloads the data file that forms the basis of the assignment from the course webpage:
`http://busoniu.net/teaching/sysid2019`

The file contains the identification data in variable `id`, and separately the validation data in variable `val`. Both these variables are objects of type `iddata` from the system identification toolbox of Matlab, see `doc iddata`. It is known from prior knowledge that the system does not have any time delay.

Requirements:

- Plot and examine the data supplied.
- Implement ARX identification explicitly using linear regression, as described in the lecture. Recall that the regressors are $-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-nb)$. Your code should work for any values of na and nb.
- Moreover, implement the simulation of the computed model for the validation data. Keep in mind that for simulation, knowledge about the real outputs of the system is not available, so we can only use previous outputs of the model itself; in particular $y(k-i)$ in the model formula must be replaced by its previously simulated value $\hat{y}(k-i)$, for $i = 1, \dots, na$. Hint: signals at negative or zero time steps can be taken equal to zero.
- Try to guess a system order from the step response shapes in the validation data. Set the na and nb orders of the ARX model accordingly, and identify a model with your code, on the identification data. Then, on the validation data, compare the output simulated with your model with the real output.

- If the results are poor, increase na and nb (e.g., in increments of 1, or make them twice as big, etc.) until you get a good fit.
- Optionally, if you still have time – or if you have bugs and want a known good solution – identify models with the same values of na, nb as above, but this time with the Matlab `arx` function (with $nk = 1$ since the system is known to not have a time delay). Compare the results with those that you obtained using your code, and verify that the two results are similar.

Relevant functions from the System Identification toolbox: `arx`, `plot`, `compare`. When the `ident` toolbox function has the same name as a function in another toolbox – like in the case of `compare`, which overloads the MPC toolbox implementation – write e.g. `doc ident/compare` to get the documentation of the `ident` variant. See also `doc ident` for the full documentation of the toolbox.