System Identification – Practical Assignment 6
ARX model identification

Logistics

- This practical assignment is a compulsory part of the course “System identification”.
- The assignment is preferably carried out in groups of two, but can also be done alone if necessary. Note that groups of three or more students are strongly discouraged.
- 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 in order to validate your attendance to the lab; the teacher will strive to do this as far as possible during the lab, together with you. Nevertheless, please write your code in a self-explanatory fashion (adding comments where necessary), so as to make it understandable on its own. At the end of the lab, please email the code as an m-file or ZIP file to the teacher (either Lucian Busoniu at `lucian@busoniu.net` or Zoltán Nagy at `zoltan.nagy@aut.utcluj.ro`), using the following filename template:
  `sysid_labN_indexINDEX_STUDENTNAME1_STUDENTNAME2`
  where N is the lab number, INDEX stands for your dataset index, see below; STUDENTNAME1 and 2 stand for the last names of the two students in the group. Please *include this file name also in the subject line of your email* (for automatic email filing purposes).
- 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: *Prediction Error Methods*.

Each student group is assigned an index number by the lab teacher. Then, the group downloads the data file that forms the basis of the assignment from the course webpage:
`http://busoniu.net/teaching/sysid2016`

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`.

Requirements:

- Plot and examine the data supplied.
- Try to guess a system structure from the step response shapes in the validation data. Set the na, nb, nk parameters of the ARX model accordingly, and identify a model with `arx`. Verify the fit on the validation data.
- Next, create a set of model structures and use the `***struc` functions to find the structure with the best it on the validation data. Plot the fit on the validation data and compare it to the fit of the model found at the first bullet point.

- Consider the results. Does the system have the structure you were guessing initially? Does the system have a time delay or not? If the orders automatically identified are much larger than what the step response indicates, can you find a reason for that (recalling the discussion in the lecture)?

- Use correlation analysis to identify an FIR model of the system (either with `impulseest`, or with `cra`). Compare the fit quality of the ARX and FIR models, as well as their numbers of parameters (if you have a model in the format of the System Identification toolbox, called e.g. `mod`, you can just type `mod` at the command line to investigate its structure). Consider the results. E.g., which model has a better fit and why? What is the influence of the number of parameters?

Since insight is much more important than the actual coding needed to solve this lab, try to summarize the choices that you make in comments at the relevant places in your script file, with supporting data such as numerical fit or MSE, number of parameters, so that your train of thought can be followed by looking at the script.

Relevant functions from the System Identification toolbox: `arx`, `struc`, `arxstruc`, `selstruc`, `plot`, `compare`, `impulseest`, `cra`. 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.