System Identification – Practical Assignment 8
Instrumental variable methods

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 compare instrumental variable and prediction error methods, by identifying ARX, IV, and ARMAX models. See the course material, Part VI: *Instrumental Variable Methods* and Part V: *Prediction Error Methods*.

Each student group is assigned an index number by the lecturer. 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`. From prior knowledge, it is known that the system has order n, given in variable `n` in the data file; and that the disturbance is not white noise, but colored. All polynomial orders in the models below should be set in accordance with this value of n.

Requirements:

- Plot and examine the data supplied.
- Identify an ARX model of order $na = nb = n$ and inspect its quality.
- Apply IV identification with the simple instruments:

$$Z(k) = [u(k - nb - 1), \dots u(k - na - nb), u(k - 1), \dots, u(k - nb)]^T$$

by using function `iv` with appropriately chosen polynomials, $C(q^{-1}) = 1$ and $D(q^{-1}) = -q^{-nb}$. (Keep in mind that `iv` expects C and D to be provided as vectors of coefficients in *increasing* order of the power of q^{-1}.) Compare the results with those obtained using ARX, in terms of model fit.

- Apply IV identification the with instruments:

$$Z(k) = [-\hat{y}(k-1), \ldots, -\hat{y}(k-na), u(k-1), \ldots, u(k-nb)]^T$$

  where the predicted outputs $\hat{y}$ are obtained by simulating the ARX model found earlier. Use again function `iv` but now setting C and D from the ARX model. Compare the results with those obtained using ARX and using the simple instruments.

- Apply IV identification with automatically chosen instruments, using `iv4`. Compare the results with those obtained before.

- Finally, identify also an ARMAX model and compare it to the best IV model.

- So far you have been comparing models only in terms of fit. Now we will investigate the computational cost to obtain each model. Use functions `tic`, `toc` to measure the execution time of each algorithm; to keep results accurate, only measure the call to the system identification function, and not your other code such as for plotting graphs. What is the ranking? Can you explain the relationship between the execution times of the algorithms using what you know from the lectures about their implementation? If you were to choose the best compromise between accuracy and computational complexity, which algorithm would win? Keep in mind that running times may be small on your desktop or laptop, but in practice the algorithms may need to be implemented on embedded, low-performance hardware.

Relevant functions from the System Identification toolbox: `arx`, `armax`, `iv`, `iv4`, `compare`, and generic Matlab functions: `tic`, `toc`.