System Identification – Practical Assignment 7
Prediction error 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.

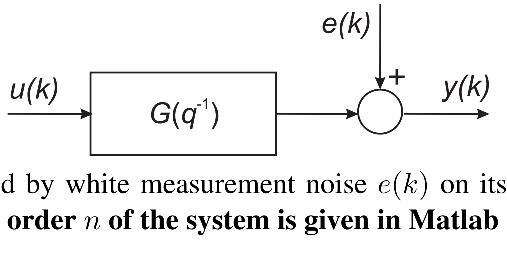
Assignment description

In this assignment we will identify and compare ARX, ARMAX, and OE models, using prediction error methods. See the course material, 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`. Both these variables are objects of type `iddata` from the system identification toolbox of Matlab, see `doc iddata`. From prior knowledge, it is known that the system has the structure in the figure below, where G is a discrete-time transfer function:

$$G(q^{-1}) = \frac{\beta_1 q^{-1} + \beta_2 q^{-2} + \dots \beta_n q^{-n}}{1 + \alpha_1 q^{-1} + \alpha_2 q^{-2} + \dots \alpha_n q^{-n}}$$



So, the system is only affected by white measurement noise $e(k)$ on its output. It is known that the system has no time delay. **The order n of the system is given in Matlab variable** `n` in your datafile.

Requirements:

- Before performing any identification experiment, try to write (on paper) the model in ARX, ARMAX, and OE form. Which of these forms does it/does it not obey? If it obeys several, which is the most efficient (in the sense of representing the correct structure with the smallest number of parameters)?

- Plot and examine the data supplied.

- Given your value of n, establish the orders and delays that must be set in the System Identification toolbox to achieve the appropriate forms for the three model types. Do not use `arxstruc` or similar automatic model selection functions, but select the order models based on your insight and experience. If the math at the first bullet point above tells you that the system does not obey the model structure, think about what would be an appropriate course of action, using also the experience from previous labs.

- Using these orders and delays, identify an ARX model, an ARMAX model, and an OE model. Compare their fit scores and discuss the results. What is the best model, when both fit quality and model complexity are considered? Does this confirm your conclusions at the point above?

Since insight is very important in this lab (rather than coding), 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`, `armax`, `oe`, `plot`, `compare`. If you have a model called e.g. `mod`, you can just type `mod` at the command line to investigate its structure, and its component polynomials are accessible as named fields, e.g. `mod.A`, `mod.B`. See also `doc ident` for the full documentation of the toolbox.