System Identification – Practical Assignment 9
Instrumental variable methods

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 study instrumental variable methods, see Part 8 of the course material, *Instrumental Variable Methods. Closed-Loop Identification.*

Each student is assigned an index number by the lecturer. 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`. 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.

We will work with two particular types of instruments: simple, input-based or ARX-output-based (see below for details). Your task is to implement the algorithm in a way that takes at the input the identification dataset, the model orders na and nb, and the instrument type; and that produces at the output the IV model found, in the `idpoly` format. For simplicity, fill in the vectors Z “manually’, rather than defining polynomials C and D.

Requirements:

- Identify an ARX model of orders $na = nb = n$ and inspect its quality. For this step you may use either the Matlab function `arx`, or your code developed for the ARX lab.
- Use your IV function to identify a model with the simple instruments:

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

- Use your IV function to identify a model with the instruments:

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

where the outputs $\hat{y}$ are **simulated** with the ARX model found earlier. Do not use predicted outputs, as those are correlated with the disturbance and will likely break the IV method!

- Compare the quality of the IV model with the simple instruments, with that of the IV model with the ARX instruments, and with that of the original ARX model.

To solve the identification problem efficiently in Matlab, it will be useful to rewrite the IV system of equations in a form amenable to matrix left division. To that end, let us take equation (8.3) from the lecture slides and rewrite it as:

$$\left[\frac{1}{N}\sum_{k=1}^{N} Z(k)\varphi^T(k)\right]\theta = \frac{1}{N}\sum_{k=1}^{N} Z(k)y(k)$$

$$\text{or equivalently: } \tilde{\Phi}\theta = \tilde{Y}$$

where the $n \times n$ matrix $\tilde{\Phi} = \frac{1}{N}\sum_{k=1}^{N} Z(k)\varphi^T(k)$ and the $n \times 1$ vector $\tilde{Y} = \frac{1}{N}\sum_{k=1}^{N} Z(k)y(k)$. Note the tildes, which signify that these quantities are variants of the original ARX regressors and of the original system outputs, "modified" by the IVs.

Hints: (i) Construct $\tilde{\Phi}$, $\tilde{Y}$ efficiently by summing up terms computed using matrix operations in Matlab. (ii) Once you have your polynomials A and B as vectors of coefficients in increasing powers of q^{-1}, use `idpoly(A, B, [], [], [], 0, Ts)` to generate the IV model, where `Ts` is the sampling period. (iii) Do not forget that all vectors of polynomial coefficients must always contain the leading constant coefficients (i.e. for power 0 of the argument q^{-1}), which must be 1 in A, and 0 in B.

Relevant functions from the System Identification toolbox: `arx`, `iv`, `compare`.