System Identification – Practical Assignment 9
Instrumental variable methods

Logistics

This practical assignment should very preferably be carried out by each student separately. Otherwise, if there are more students than computers, *with the explicit agreement of the lab teacher for each group*, students may team up in groups of 2.

The assignment solution consists of Matlab code. Develop this code in a single Matlab script. If you need to create functions, they can be local to the script, see local functions in scripts.

The overall rules for labs are described on the website. For each particular lab, your attendance will only be registered if you have a working, original solution. The teacher will check that your code **works** during the lab class. Only after this has been done, for the **originality** check, upload your solution here:

`https://www.dropbox.com/request/66xY7nVF3BFdVc6tztA3`

Upload *only once, a single m-file, named exactly according to the following pattern*:

`L9_ENgh_LastnameFirstname.m`

where `g` is the group, `h` the halfgroup, and the last and first names follow. E.g., `L9_EN32_PopAlex.m`. *If you worked in a group per the above, only upload one file with both student names included.* Any files that are duplicate, nonstandard, inappropriately named, or corresponding to unchecked solutions will not be considered. Files will be automatically tested for plagiarism, and any solution that fails this test will be marked copied; only solutions that pass both the working and the originality test are definitively validated. Therefore, while you are encouraged to discuss ideas and algorithms amongst colleagues, sharing and borrowing pieces of code is forbidden.

Assignment description

In this assignment we will study instrumental variable methods. 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. 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 is of the order given in variable `n` in the data file; and that the disturbance is not white noise, but colored. Thus, for all models we will use $na = nb =$`n`, the value from the datafile.

Your task is to implement the IV algorithm using instruments based on ARX-outputs. 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 the equation 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 $(na + nb) \times (na + nb)$ matrix $\tilde{\Phi} = \frac{1}{N}\sum_{k=1}^{N} Z(k)\varphi^T(k)$ and the $(na + nb) \times 1$ 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 regressors and of the original system outputs, "modified" by the IVs.

In the equation above, the instrument vector is:

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

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!

Note that the regressor vector $\varphi(x)$ is the usual one from ARX:

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

Requirements:

- Identify an ARX model with orders chosen as above and inspect its quality. For this step you may use either the Matlab function `arx`, or your code developed for the ARX lab.
- Identify a model with the IV method, with the same orders, using the method described above.
- Compare the quality of the IV model with that of the original ARX model, in prediction and simulation.

Hints: (i) For simplicity, fill in the vectors Z “manually’ with the $\hat{y}$ and u values, rather than defining polynomials C and D. (ii) Construct $\tilde{\Phi}$, $\tilde{Y}$ efficiently by summing up terms computed using matrix operations in Matlab.