System Identification – Practical Assignment 10
Arbitrary instruments and closed-loop identification with IV methods

Logistics

Same as for previous labs.

Assignment description

This assignment continues the work from lab 9, on instrumental variable methods. It will be useful to keep at hand that lab's description and your solution for it.

Each student is assigned an index number by the lecturer. Then, the student downloads the data files that form the basis of the assignment from the course webpage:
`http://busoniu.net/teaching/sysid2019`

Part 1: IV methods with arbitrary instruments

For this part, use the same dataset as for Lab 9. 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:

- Extend your solution from the previous lab so that it works for *any* instrument vector. To this end, the instruments should be generated by a discrete-time transfer function:

  $$x(k) = \frac{D(q^{-1})}{C(q^{-1})}u(k)$$

  As an example, if $C(q^{-1}) = 1+0.5q^{-1}$ and $D = 0.1q^{-1}$, then the IVs are computed by simulating $x(k) = -0.5x(k-1) + 0.1u(k-1)$. Thus, your task is to extend the algorithm implementation so that it takes at the input the identification dataset, the model orders na and nb, **and the polynomials C, D**; and so that it produces at the output the IV model found, in the `idpoly` format. Hint: The vector of coefficients of any polynomial must always contain the leading constant coefficient (for power 0 of the argument q^{-1}), equal to 1 for C.

- Identify an ARX model of orders $na = nb = n$, either with the Matlab function `arx`, or with your code developed for the ARX lab.

- 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 your implementation with appropriately chosen polynomials: $C(q^{-1}) = 1$ and $D(q^{-1}) = -q^{-nb}$. Check that you obtain the same results as in Lab 9 with the simple code.

- Apply IV identification with the instruments:

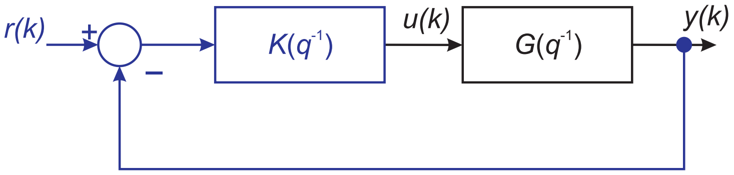
$$Z(k) = [-\hat{y}(k-1), \dots, -\hat{y}(k-na), u(k-1), \dots, u(k-nb)]^T$$

  where the outputs $\hat{y}$ are those of the ARX model found earlier. Use again your implementation but now setting C and D from the ARX model. Check that you obtain the same results as in Lab 9 with the simple code.

- Repeat the previous two points with the existing Matlab function `iv`, and verify that you obtain similar results to those of your function (due to algorithmic details they may not be exactly identical). Set $nk = 1$ in `iv`.

Part 2: Closed-loop identification

Here we must identify an unstable, second-order, mechanical system with unknown transfer function G. Since the system is unstable, it could be damaged or destroyed if run in open loop; instead, it is always controlled by a PID with transfer function K, see the figure below. This scheme can easily be placed in the general structure (with T, R, S, B, A) discussed in the lectures.



The data file for this part contains the identification and validation data in variables `id` and `val`, with the usual format. The additional, reference signals for the two experiments are given in vectors `rid` and `rval`, respectively. In the datasets, the output is additionally corrupted by zero-mean Gaussian noise (not shown in the figure).

Start by identifying a standard instrumental variable model using the input-output identification data, where the instruments are generated with an ARX model (since this scenario worked well before). It is preferable to use your own code from Lab 9, as it will be the easiest to extend. Study and interpret how this model works on the validation dataset, keeping in mind that this technique requires an open-loop identification experiment.

Next, building on your own code that you developed for Lab 9, change the IV method so that the instrument vector is constructed using the reference signal:

$$Z(k) = [r(k-1), r(k-2), \dots r(k-na-nb)]^\top$$

Run the new algorithm on the identification data, and study the performance of the model on the validation dataset.

Note that for this experiment, it is *not* informative to test the models on the identification dataset, as they will not work due to numerical issues.

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