

# System Identification – Practical Assignment 11:

## Closed-loop identification.

### Correlation tests for model validation.

#### Logistics

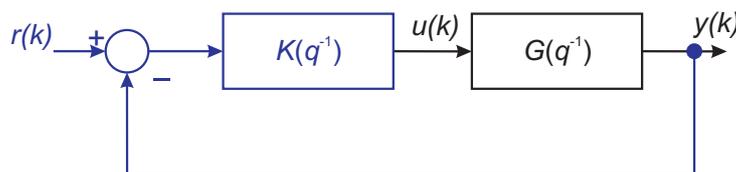
- This practical assignment is a compulsory part of the course “System identification”. It should be carried out by each student separately.
- The assignment solution consists of Matlab code. This code will be checked and run by the teacher in order to validate your attendance to the lab; the teacher 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 (Zoltán at [zoltan.nagy@aut.utcluj.ro](mailto:zoltan.nagy@aut.utcluj.ro), or Marius at [marius.costandin@aut.utcluj.ro](mailto:marius.costandin@aut.utcluj.ro)), using the following filename template:  
`sysid_labN_indexINDEX_NAME`  
where N is the lab number, INDEX stands for your dataset index, see below; and NAME is your last (family) name. Please *include this file name also in the subject line of your email*.
- 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 a closed-loop system identification technique, as well as model validation, see Parts VIII and X of the lectures. 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/sysid2017>

#### Part 1: 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 in lab 9). It is preferable to use your own code from that lab, as we will need it below anyway; but you may also use the Matlab function `iv` if you wish. 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:

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

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.

## Part 2: Model validation

The setting and datafiles are different from Part 1, as follows. The experiment is open-loop, and the data file contains the identification and validation data in variables `id` and `val`, as well as the true order of the system in variable `n`. From prior knowledge it is also known that the system is output-error type. We will use prediction error methods, and we will take the orders of the polynomials equal to the known system order  $n$ , accepting the risk of incorrect models – in fact, the goal of this part is exactly to learn how to check their correctness using correlation tests.

Requirements:

- Identify an ARX model with the orders  $na = nb = n$  equal to the system order available from prior knowledge. Use `resid` to perform correlation tests and determine whether the model is good. Consider separately the input-output and the disturbance models. Also validate the model in simulation to confirm the results.
- Identify an OE model with orders  $nb = nf = n$ . Use `resid` to perform correlation tests and determine whether the model is good. Also validate the model in simulation to confirm the results. Discuss the results with OE in comparison to the ARX model.

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