System Identification – Practical Assignment 5
Correlation Analysis

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

We apply the linear regression method to obtain finite impulse response (FIR) models from input-output data – see the course material, Part IV: *Correlation Analysis*. This data is more general than just the step or impulse responses we have been seeing so far. Note that the linear regression itself will be only a tool, which is already implemented as a part of the system identification algorithms.

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`. Both these variables are objects of type `iddata` from the system identification toolbox of Matlab, see `doc iddata`. The corresponding time vectors are `tid`, `tval`.

Requirements:

- Plot and examine the data supplied. Determine whether the input and output are zero-mean or not. If the signals are not zero-mean, remove the means using e.g. `detrend`.
- After making sure the signals are zero-mean, compute the correlation functions r_u and r_{yu} from the identification data, using the formulas from the lecture. Check whether the input signal is white noise.
- Implement the system of linear equations to obtain the FIR model for any value of M supported by the data, and check that it works well for a few values of M. To that end, compute the convolutions to simulate the model response to the identification and validation inputs, and compare with the identification and validation outputs.
- Study the influence of the length M of the FIR model on the accuracy of the model. A rule of thumb for selecting a good M is as follows: preferably the entire transient regime of the impulse

response must be modeled (until it reaches steady state), but without estimating too many parameters because this would introduce overfitting. These requirements may be conflicting so you may need to deviate one way or the other to get a good fit.

- For comparison, identify an FIR model of the system using `cra` with the best value of M that you found, and compute the response of this model to the validation inputs. Compare with your model. Note: the responses may not be exactly the same due to different algorithmic details, but they should look similar.

To allow for more insight, the true impulse response of the system is provided as a vector `imp` in the datafile (note that this true response would not be available in a real identification experiment). Solve each part of the assignment without using the true response, but once you have solved it, you may optionally compare the FIR models obtained with the true impulse response.

This is the first lab where we start using in earnest the System Identification toolbox (`ident`). Relevant functions from this toolbox: `cra`, `detrend`, `plot`, `compare`; and generic Matlab function `conv`. See also `doc ident` for the full documentation of the toolbox.