System Identification – Practical Assignment 5
Correlation Analysis

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. Develop this code in a single Matlab script. 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 Nagy at `zoltan.nagy@aut.utcluj.ro`, or Marius Costandin at `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

The last assignment dealt with linear regression. Here, we return to the system identification context and 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. Moreover, the 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/sysid2017`

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 (visually or programmatically) whether the input and output are zero-mean or not. If the signals are not zero-mean, remove the means using `detrend`. After making sure the signals are zero-mean, check whether the zero-mean input is white noise; the function `xcorr` will be useful to this end.
- Identify an FIR model of the system using `cra`, and compute the response of this model to the identification and the validation data. Apply an input whitening filter or not (see the help text or the Matlab example in the lecture), depending on wether the input is already white-noise or not, as

determined above. Plot the model output, comparing it with the system output, and verify the fit quality, e.g. by computing the MSE.

- Study the influence of the length M of the FIR model on the accuracy of the model. The 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.

- Identify an FIR model using `impulseest`. Use both the default filter length automatically computed by the function, as well as the best value of M found at the previous step. Compare the model output with the system output on the identification and the validation data.

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, compare the FIR models obtained with the true impulse response. Note: the model returned by `impulseest` is represented as a discrete-time transfer function, so the FIR model is stored as a vector (polynomial) in the numerator field `num` of the model.

This is the first lab where we start using in earnest the System Identification toolbox (`ident`). Relevant functions from this toolbox: `cra`, `impulseest`, `detrend`, `plot`, `compare`; and generic Matlab functions `conv`, `xcorr`. When the `ident` toolbox function has the same name as a function in another toolbox – like in the case of `impulse`, which overloads the Control Systems toolbox implementation – write e.g. `doc ident/impulse` to get the documentation of the `ident` variant. See also `doc ident` for the full documentation of the toolbox.