System Identification – Practical Assignment 9
Pseudo-random binary sequences

- This practical assignment is a compulsory part of the course "System identification".
- The assignment is preferably carried out in groups of two, but can also be done alone if necessary. Note that groups of three or more students are strongly discouraged.
- 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 (either Lucian Busoniu at `lucian@busoniu.net` or Zoltán Nagy at `zoltan.nagy@aut.utcluj.ro`), using the following filename template:
  `sysid_labN_indexINDEX_STUDENTNAME1_STUDENTNAME2`
  where N is the lab number, INDEX stands for your dataset index, see below; STUDENTNAME1 and 2 stand for the last names of the two students in the group. Please *include this file name also in the subject line of your email* (for automatic email filing purposes).
- 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 the creation and properties of pseudo-random binary sequences, PRBS. See the course material, Parts VII: *Input Signals*.

Each student group is assigned an index number by the lecturer. Then, the group downloads from the course webpage:
`http://busoniu.net/teaching/sysid2016`
the data file that forms the basis of the assignment, as well as the `simulateid` function, which obtains identification data in simulation given an input signal. For the purposes of this lab, since we do not have access to the real system, this simulation takes the place of the real identification experiment. The function is given in obfuscated, so-called p-code, so that you can treat the simulator as an unknown system, as would be the case in a real experiment.

The data file now only contains the validation data in variable `val`, while the identification data will be generated during the assignment using the simulator of the system, stored in variable `plant`. From prior knowledge, it is known that the system to be identified has order not larger than 4, and that the disturbance is colored (not white noise). We choose to identify ARX models, and due to the colored disturbance take large values for the orders: $na = nb = 15$. Note that in order to identify an ARX model with $n = na + nb$ parameters, the input signal should be persistently exciting of at least order n.

Requirements:

- Write a function that generates an input signal of length N using a maximum-length PRBS with a register of a given length m, and which switches between given values a and b. Parameters

N, m, a, b are given as inputs to this function, and m is limited to the range $3, 4, \ldots, 10$. Note that if $N > P$, the period of the PRBS, then the input signal will consist of several repetitions of the maximum-length PRBS. Test this function for some values of N, m, a and b. Hints: Using the state space representation of the linear shift feedback register, it is easy to create a vectorized implementation in Matlab. You can use function `mod` to implement the modulo-2 summation.

- Generate an input signal of sufficient length (say around 300 samples) with $m = 3$, taking values $a = 0.5$ and $b = 1$. Apply this signal to the system simulator using `simulateid`; the signature of this function is `id = simulateid(plant, u)` where `plant` is the system simulator, `u` is the input sequence, and `id` is the returned identification data as a standard object of type `iddata`.
- Identify an ARX model with this data; does the data have a sufficient order of persistent excitation? Verify the quality of the ARX model.
- Repeat the above but now with $m = 10$. Does this new data have a sufficient order of persistent excitation? Verify the quality of the ARX model.
- Repeat the experiment, but now using the default input signal generator in the System Identification toolbox, `idinput`. Study the interface of this function and generate a PRBS taking values $a = 0.5$ and $b = 1$, without limiting its frequency content; note that the order m of the register is chosen automatically. Study also the possibilities for generating other types of input (white noise, sum of sines).

Relevant functions from the System Identification toolbox: `arx`, `idinput`, `plot`, `compare`.