System Identification – Practical Assignment 7
Pseudo-random binary sequences

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`, or Marius 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

In this assignment we will study the creation and properties of pseudo-random binary sequences, PRBS. See the course material, Parts VI: *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/sysid2017`
the `system_simulator` function, which obtains experimental data from the system 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 experiment, for both identification and validation. 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 signature of the function is `data = system_simulator(index, u)` where `index` is your index number, `u` is the input sequence (in discrete time), and `data` is the returned experimental data as a standard object of type `iddata`.

From prior knowledge, it is known that the system to be identified has order not larger than 4, and that the disturbance does not satisfy the structural assumptions of the ARX model. Nevertheless, due to its simplicity we choose to identify an ARX model, and to compensate for the disturbance structure we take large values for the orders: $na = nb = 15$. Note that in order to identify an ARX model, the input signal should satisfy a certain persistence of excitation condition, see the lecture for the details.

Requirements:

- Generate first a validation dataset with `system_simulator`, using as input e.g. a sequence of steps with magnitudes on the order of $0.5 - 2$ and a length of about 50 steps each. You can use this dataset to validate all the models found below.
- 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 identification 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 using `system_simulator`.
- Identify an ARX model with the identification data obtained, using the Matlab `arx` function for simplicity. Compute the order of persistent excitation for the input. Is it sufficient to identify properly the ARX parameters? Verify the quality of the ARX model on the validation data, and see if it supports your conclusion.
- 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.
- Optionally, 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`.