System Identification – Practical Assignment 6
ARX identification

Logistics

Please reread the logistics part of lab 2, the same rules will apply to this lab. The only thing that changes is the dropbox link, which for this lab is:

`https://www.dropbox.com/request/Lg1BQ6R4RmgPwTJdeG5q`

Assignment description

In this assignment we will identify ARX models (autoregressive with exogenous input), using least-squares, linear regression. See the course material, *ARX Identification*.

If you did not already use the real DC motor system, familiarize yourself with this real system as explained e.g. in Lab 5. The guide is at the link: `https://busoniu.net/teaching/sysid2023/dcguide.pdf`.

Each student will obtain a data set using the DC motor and will identify the system, as detailed in the following instructions.

1. To keep things simple, we will create a single, longer sequence of data containing both the identification and validation data. We will use a sampling rate of 0.01 s (10 ms). After a short range of zero inputs, apply a PRBS signal with amplitudes in the interval $[-0.7, 0.7]$ and a length of about 200 samples, followed by another range of zero inputs, and then a step signal of magnitude around 0.4 and around 70 samples in length. We will learn how to generate PRBS signals in the next lab; for the moment, you can use `idinput(N, 'prbs', [], [-0.7 0.7])`.

2. Apply the signal generated to the DC motor. The output is the rotational velocity. Isolate the data range corresponding to the random inputs and copy it to new input and output vectors; this will be our identification data. **Important note**: To minimize system wear, separate the code that generates the data from the code that performs the rest of the steps below (easiest using different script sections, see *Code Sections* in the Matlab documentation), and regenerate the data only when necessary.

3. Plot and examine the data obtained.

4. Implement ARX identification explicitly using linear regression, as described in the lecture. Recall that the regressors are $-y(k-1), \ldots, -y(k-na), u(k-1), \ldots, u(k-nb)$. Your code should work for any values of na and nb.

5. Implement the simulation of the computed model for the validation data (or, if you find it simpler, on the entire dataset, but in that case judge the model quality/compute MSEs only on the validation range of the signals). Keep in mind that for simulation, knowledge about the real outputs of the system is not available, so we can only use previous outputs of the model itself; in particular $y(k-i)$ in the model formula must be replaced by its previously simulated value $\tilde{y}(k-i)$, for $i = 1, \ldots, na$. Hint: signals at negative or zero time steps can be taken equal to zero.

6. Given the shape of the response (or the physics of the system), what is the system order? Set the na and nb orders of the ARX model accordingly, and identify a model with your code, on the identification data. Then, compare the output simulated with your model with the real output.

7. If the results are not good, increase na and nb in increments of 1 until you get a reasonable fit.

8. Optionally, if you still have time – or if you have bugs and want a known good solution – identify models with the same values of na, nb as above, but this time with the Matlab `arx` function (with $nk = 1$). Compare the results with those that you obtained using your code, and verify that the two results are similar. Keep in mind that to apply `arx`, you need to create an `iddata` object from your identification data.

If you run into problems with the DC motor system, as a backup solution, talk to your lab assistant so they assign you a simulation dataset index. Then replace steps 1–2 of the requirements above by the following. Downloads the data file corresponding to the assigned index from the course webpage. 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`. Apply all the other steps (3 and onward) of the requirements to this simulation data. Your assignment will be evaluated in the same way, regardless of whether you apply the method to the real or simulated data.