Identification of linear systems : Characterization of linear systems, parameters estimation using Ordinary least squares method and Recursive least squares (online) method

1. Characterization using a random signal: quotient of instrumental determinants (QID) test :

Many experiments have been carried out and they consist on applying a random input signal to different systems and on measuring the output responses. The obtained results from these experiments are stored respectively within the following files Qu1_1.sod, Qu1_2.sod and Qu1_3.sod.

These latter can be imported on scilab software by using for instance the command load("Qu1_1.sod"). This command enables to import the line vectors « u » and « y » which encompass the experimental inputs and outputs .

1. Write a Scilab script which make it possible to implement the QID test. Then, apply the developed script in order to determine each time the order of the systems involved on the experiments which are described by the set of data Qu1_1.sod, Qu1_2.sod and Qu1_3.sod.

2. parameters estimation using Ordinary least squares method and Recursive least squares (online) method

The characterization of a discrete time sampled system led to the following parametric model :

$$y(k) = -a_1 y(k-1) - a_0 y(k-2) + b_1 u(k-1) + b_0 u(k-2)$$

2.1. Stationary Case

An experiment was carried out consisting of exciting the system with a Pseudo-Random Binary Sequence (PRBS) and measuring its response. During this experiment, the system is stationary (the value of its parameters does not vary) and its parameters are :

a1 = −1.2, a0 = 0.6, b1 = 0.1 and b0 = 0.12.

The result of this experiment is stored in the Qu2_1.sod file which can be imported into Scilab via the load("Qu2_1.sod") command. At the end of this command the Scilab variables 'u' and 'y' contain the samples of the experimental input and output.

1. Write a Scilab script that allows you to construct Y and Φ, and deduce $\hat{\theta}$ using the ordinary least squares method.

2. Write a Scilab script that implements the recursive least squares method. Note that the advantage of this last method is that it can be implemented “online” (allowing the updating of the estimates over the course of the experiment). To simulate an “on-line” implementation, we will focus on implementing formulas (2.6) and (2.8) in our support for each sample of the vectors 'u' and 'y' (i.e. as an iterative procedure). Use $P0 = \alpha I$ with $\alpha = 200000$ and $\lambda = 1$. Compare the obtained result with that of the ordinary least squares method. Experiment your script with different values of P0 (typically $\alpha = 0.5$, $\alpha = 100$ and $\alpha = 200000$). Conclude on the influence of α on the speed of convergence of the method.

2.2 Non-stationary case

A new experiment was carried out on the same system (sending a PRBS and measuring its response), but this time by varying the value of the system parameters. More precisely, the values of the system parameters were during the experiment as follows:

— for the first 25 samples: $a1 = -1.2$, $a0 = 0.6$, $b1 = 0.1$ and $b0 = 0.12$;

— for the last 25 samples: $a1 = -1.15$, $a0 = 0.55$, $b1 = 0.15$ and $b0 = 0.1$.

The result of this experiment is stored in the Qu2_2.sod file, which can be imported into Scilab via the load("Qu2_2.sod") command. At the end of this command the Scilab variables 'u' and 'y' contain the samples of the experimental input and output.

1. Use the previous script which implements the ordinary least squares method to estimate the system parameters. What do you think about the accuracy of the method in this case?

2. Now use your script which implements the recursive least squares method. Use as forgetting coefficient $\lambda = 1$, then $\lambda = 0.5$. Conclude on the accuracy of the m method, and the influence of λ on its ability to follow the non stationarities.