Laboratory Work #6

EVALUATION OF NUMERICAL CHARACTERISTICS OF RANDOM PROCESSES IN LINEAR SYSTEM

ESTIMATING CONTINUOUS-TIME MODELS USING SIMULINK DATA

1. Goal of the lab:

To demonstrate the use of MATLAB/SIMULINK for calculation and visualization of some basic numerical characteristics of random processes.

The laboratory work illustrates how models simulated in Simulink® can be identified using System Identification Toolbox™. It describes how to deal with continuous-time systems and delays, as well as the comparison of different types estimated models.

2. Introduction

Random processes, converted through a linear dynamic system usually change its characteristics. They become bearers of some new information, which is derived in the process of identification.

Correlation function between the random variables or samples of random signals is calculated using the following equations:

- Autocorrelation function:

$$R_x(k) = \frac{1}{N-|k|} \sum_{i=0}^{N-1-|k|} x(i)\, x(i+k)$$

where $k \in [-N; N]$ and $R_x(k) = R_x(-k)$ and $R_x(k) \leq R_x(0)$.

- Correlation function:

$$R_{xy}(k) = \frac{1}{N-|k|} \sum_{i=0}^{N-1-|k|} x(i)\, y(i+k)$$

$$R_{yx}(k) = \frac{1}{N-|k|} \sum_{i=0}^{N-1-|k|} y(i)\, x(i+k)$$

MATLAB gives exceptional opportunities for calculation and visualization of the estimates of the characteristics of random processes. The following function is used to measure the correlation between the random variables:

- ***xcorr*** – the correlation functions of the processes are calculated the equation mentioned above (Autocorrelation and Correlation functions).

- ***xcov* -** estimates autocovariance and cross-covariance sequences. This function has the same options and evaluates the same sum as ***xcorr***, but first removes the means of x and y. Cross covariance between x and y signals is calculated using the equation below:

$$C_{xy}(m) = E\{(x(n+m) - \mu_x)\,(y(n) - \mu_y)\} = R_{xy}(m) - \mu_x \mu_y{}^{*}.$$

- ***crossxorr*** **-** Computes the sample cross-correlation function (XCF) between univariate, stochastic time series. When called with no output arguments, ***crosscorr*** plots the XCF sequence with confidence bounds.

Tasks

3. Evaluation of numerical characteristics of signals

1. Create a time vector (samples) with the values 0 to 200.
2. Generate a sine wave with the frequency *f = 0.05 Hz.*

$$u[k] = \cos(2\pi f k)$$

3. Plot the first 50 samples of the data.
4. Compute the correlation between the samples of the signal (***xcov*** & ***xcorr***) and plot it (**stem**). *MaxLag = 50.*
   Examine the option *ScaleOpt = 'biased'* case and observe the difference from the previous result.
5. Generate another signal – Exponentially decaying signal for the same samples:

$$u[k] = e^{-0.05k}$$

6. Plot the first 100 samples of the data.
7. Compute the correlation between the samples of the signals (***xcov*** & ***xcorr***) and plot it (**stem**). (Show the second half of the correlation function in the plot).
   MaxLag = 100, ScaleOpt = 'coeff'.

8. Add noise on the signal initiated above and create 2 random signals as described below:

$$u_1[k] = \cos(2\pi f k_1 + \theta_1)$$

$$u_2[k] = \cos(2\pi f k_2 + \theta_2)$$

$$\theta_1 = 2\pi N_1, \quad k_1 = 0{:}200,$$

$$\theta_2 = 2\pi N_2, \quad k_2 = 0{:}300,$$

$$N_1/N_2 - Normally\ distributed\ random\ signal$$

9. Compute the cross-correlation function between the samples of the random signals $u_1[k]$ and $u_2[k]$ (***crosscorr***) and plot it (**stem**). *MaxLag = 50*
10. Find the lag where the CCF gets its maximum value.
11. Find the delay between two signals ($u_1[k]$ and $u_2[k]$).

Note: *Provide the titles for the plot results for better representation/understanding*.

4. Estimating Continuous-Time Models Using Simulink Data

1. Create a new Simulink model in MATLAB.
2. Add a source signal ***Chirp signal*** from Simulink Library with the properties below:

   Initial frequency = 0.001 Hz
   Target time = 500 secs
   Frequency at target time = 0.2 Hz

3. Increase the amplitude of the input signal using ***Gain*** block. (Gain = 10)
4. Pre-process the input using the filter shown below:
$$\frac{0.05s + 0.1}{s}$$
   Use ***Transfer Fcn*** block in Simulink.
5. The signal should pass through the ***Zero-Order Hold*** block. *Sample time = 1 s*
6. Use input delay of 1.001 seconds (***Transport Delay***)
7. The input will be pass through the system with the transfer function of $\frac{0.1}{s+0.5}$.
   Use ***Transfer Fcn*** block in Simulink.
8. Add Noise to the output of the process using ***Random Number*** block (*Hint*: Use a ***Sum*** block to add the noise to the output).
   Mean = 0
   Variance = 0.01
   Seed = 0
   Sample time = 0.5

9. Noise-corrupted output should be fed back to the input to create a closed loop industrial process. (*Hint*: Use a ***Sum*** block to achieve the feedback loop).
10. Display the plot of the input, noise-free output and noisy output in the same window of ***Scope***.
11. Change the Simulation stop time to *500 s* and run the simulation to obtain output data accordingly.
12. Write the input (after Zero-Order Hold) and output (noise corrupted) values to MATLAB Workspace for further use. Use ***To Workspace*** block.
    Sample time = 0.5 s
    Save format: Array (Inherit from input)

13. Open a new script file in MATLAB and run the simulation (Use ***set_param*** and ***sim*** commands).
14. Save the input/output data in an IDDATA object (***iddata***). *Sample time = 1 sec*.
    This data will be used for estimation purposes.
15. Change the ***Seed*** value of Random Number to ***13*** in Simulink file and run the simulation again. (Use ***set_param*** and ***sim*** commands).

16. Save the input/output data in an IDDATA object (***iddata***). This data will be used for validation purposes.
17. Represent the system built in Simulink file above using ***idpoly*** structure (with the properties below) in a new MATLAB Script file*:*
    Ts = 0
    Input Delay = 1
    Noise Variance = 0.01

18. Firstly, estimate a state-space model with the **best** order (automatically defined). Use ***n4sid*** command in MATLAB. Which order was selected for the model?
19. Estimate the Impulse Response model of the process using estimation data. Select the negative delays as well for analyzing the system. (Use ***impulseest*** and ***impulseplot*** commands).
20. Show the 3 standard deviations of confidence region for the plotted impulse response (***showConfidence***). What does the plot result mean? Is it possible to determine the time delay based on the plot?
21. Build a ARX model (***arxstruc***) with the estimated structure of:
    na = 1:2
    nb = 1:2
    nk = 1:10

    Select the best structure of the model that fits the validation data. Use ***selstruc*** command in MATLAB.
    What is the delay value selected?
22. Build an ARX model based on the coefficients estimated above (***arx***).
23. Compare the IDPOLY, state-space and ARX models built above with the validation dataset (***compare***).
24. Demonstrate the residual analysis of each model and explain the plot results (***resid***).

5. Questions

1. Give a definition of random variable and random process.

2. Give a definition of correlation function. What is the meaning of correlation? Analyze the plots with autocorrelation functions. What conclusion can be done by looking at the graphs of autocorrelation functions?

3. What are the main statistical properties of random processes?