Communication Theory
Lab session 1: Stochastic Processes
Academic Year 2023/2024

Objectives

In this lab session the student will learn to

- implement stochastic processes using Matlab.
- calculate the mean and the autocorrelation of a stochastic process.
- extract conclusions concerning the stationarity and ergodicity of a stochastic process.

Rules and submission of the lab report

- The lab report must be done in pairs.
- It will contain the matlab code (including explaining comments) ,as well as the corresponding figures, generated by solving the exercises (2) and (3).
- The deadline is two weeks after the lab session. Late submissions will imply a penalty.
- The report and the code will be submitted via aulaglobal.
- Please, generate a .zip file including report and code. The file name must contain the name of both authors. (Example: lancho_alejandro_and_lancho_alejandro.zip)

1 Example of a Stochastic Process

In this section, it is described the implementation of a stochastic process using Matlab. It is also shown how to calculate the corresponding mean and autocorrelation of the process. The matlab file `example_stochastic_process.m` contains the commands that will help you to define a stochastic process, as well as how to represent the mean, the autocorrelation and how to simulate several realizations of it.

A stochastic process can be defined as follows:

$$\begin{aligned} X(t,\omega_1) &= 1, \quad t \geq 0 \\ X(t,\omega_2) &= 2, \quad t \geq 0 \\ X(t,\omega_3) &= e^{-t}, \quad t \geq 0 \\ X(t,\omega_4) &= \sin(t), \quad t \geq 0 \end{aligned} \tag{1}$$

with probabilities $p(\omega_1) = p(\omega_2) = p(\omega_3) = p(\omega_4) = \frac{1}{4}$.

The following sections will explain in detail how to implement this stochastic process.

1.1 Process Definition

First, the temporal signal are defined according to (1):

```
x_1 = 1;
x_2 = 2;
x_3 = exp(-t);
x_4 = sin(t);
```

where the temporal vector `t` is defined as follows:

```
t=-2:0.01:10;
```

In the proposed stochastic process (1), the temporal signals are defined for $t \geq 0$, but, as we can see in the definition of the vector t, this starts at -2. To match this up, let's multiply each signal by a pulse $p(t)$, defined as

$$p(t) = \begin{cases} 0 & -2 \leq t < 0 \\ 1 & 0 \leq t \leq 10 \end{cases} \tag{2}$$

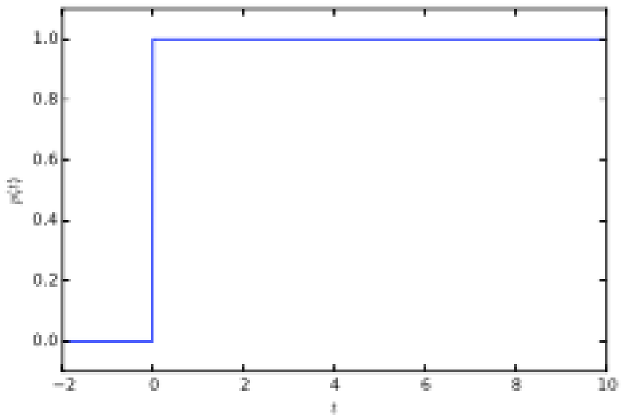

Figure 1: Pulse $p(t)$.

In Matlab, the pulse shape can be generated as follows:

```
pulse = [zeros(1,200) ones(1,1001)];
```

Since the temporal step is 0.01, the number of samples in the range $[-2, 0)$ are 200, and the number of samples in $[0, 10]$ are 1001. In general, the number of samples between [a, b], with a step d can be calculated as $\frac{b-a}{d}$. If b is included in the interval, an extra sample must be added.

By Multiplying the signal x_i by the pulse $p(t)$, we can ensure that all signals are defined in the range $t \geq 0$.

In order to simulate one realization of the stochastic process, we generate a random number between 1 and 4 (variable `w` in the code). The corresponding signal is drawn with the command `plot` according to the value of `w`.

```
w = ceil(4*rand);
if(w == 1)
   H = plot(t,x_1*pulse,'linewidth',2);
elseif(w == 2)
   H = plot(t,x_2*pulse,'linewidth',2);
elseif(w == 3)
   H = plot(t,x_3.*pulse,'linewidth',2);
elseif(w == 4)
   H = plot(t,x_4.*pulse,'linewidth',2);
end
```

1.2 Calculating the mean

The mean of the process can be obtained as follows:

$$m_X(t) = E\{X(t)\} = \int x \; f_{X(t)}(x) \; dx = \sum_i Pr(\omega_i)X(t,\omega_i) \tag{3}$$

Since our cases occur with the same probability, $Pr(\omega_i) = \frac{1}{4}$, $\forall i$, we can write the equation (3) as follows:

```
media = x_1.*pulse/4 + x_2.*pulse/4 + x_3.*pulse/4 + x_4.*pulse/4;
```

Figure 2 shows the representation of the mean, as well as the signals defined in (1). Note that according to the result shown in Figure 2, we can conclude that the process is not wide-sense stationary (WSS) because the mean varies with the time.

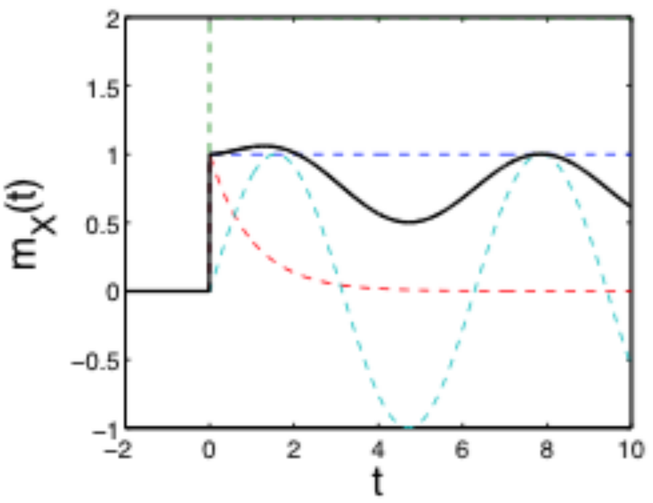

Figure 2: Mean of the process $X(t, \omega)$.

1.3 Calculating the autocorrelation

The autocorrelation can be calculated as

$$R_X(t_1, t_2) = E\{X(t_1)X(t_2)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 \cdot x_2 \cdot f_{X(t_1),X(t_2)}(x_1, x_2)\ dx_1 dx_2 \qquad (4)$$
$$= \sum_i Pr(\omega_i) X(t_1, \omega_i) \cdot X(t_2, \omega_i)$$

Let's redefine the vector t as follows:

```
t=-2:0.1:10;
```

Note that the time step has increased to $d = 0.1$, i.e., there are fewer samples in our interval $[-2, 10]$ (in order to reduce the computation time required to calculate $R_X(t_1, t_2)$). The new pulse vector $p(t)$ can be defined as follows:

```
pulse = [zeros(1,20) ones(1,101)];
```

and the autocorrelation can be calculated using the following commands

```
Rx = zeros(length(t));
for t1=1:length(t)
   for t2=1:length(t)
       Rx(t1,t2) = (x1(t1)*x1(t2) + x2(t1)*x2(t2) + x3(t1)*x3(t2) + x4(t1)*x4(t2))/4;
   end
end
```

Figure 3 shows the autocorrelation obtained using the former code.

When the autocorrelation only depends on the temporal difference between the time instants t_1 and t_2, we can redefine the autocorrelation as $R_X(t_1, t_2) = R_X(t_1 - t_2) = R_X(\tau)$, where $\tau = t_1 - t_2$. Therefore, in this case, we can represent the figure τ Vs $R_X(\tau)$ as a 2-d graph.
Example: $R_X(t_1, t_2) = \frac{1}{2}\cos(t_1 - t_2)$. This autocorrelation can be written as $R_X(\tau) = \frac{1}{2}\cos(\tau)$, where $\tau = t_1 - t_2$, so we can obtain it as follows

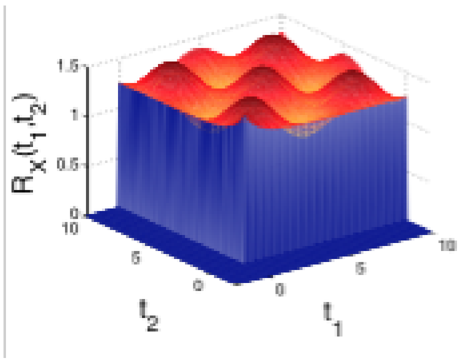

Figure 3: Autocorrelation of the process $X(t, \omega)$.

```
tau = -10:0.01:10;
Rx = 1/2*cos(tau);
plot(tau,Rx)
grid on
```

Figure 4 shows the autocorrelation obtained using the former code.

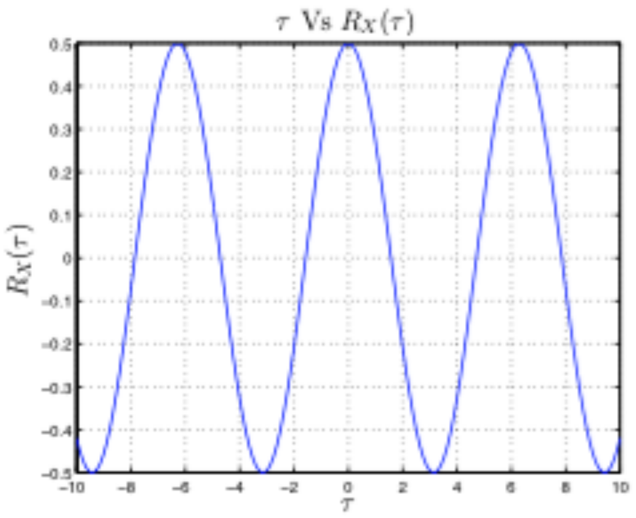

Figure 4: Autocorrelation $R_X(t_1, t_2) = R_X(t_1 - t_2) = R_X(\tau)$.

Please take into account that the previous mean and autocorrelation have been computed using their corresponding **analytical** expressions. We can also compute the mean and the autocorrelation **numerically**. To accomplish this, we generate N realizations of the stochastic process and store them in a matrix (each row contains one realization, therefore, this matrix will contain N rows and T columns, where T=*length*(t)). Let us call this matrix D. To compute the mean **numerically**, we average the rows of the matrix D. To compute the autocorrelation **numerically**, we compute the mean of the product $D(:, t_i)D(:, t_j)$, for all the possible values t_i and t_j ($D(:, t_i)$ denotes the column of the matrix D corresponding to the time instant t_i).

2 Analytical Description of a stochastic process

The stochastic process $Z(t)$ is defined as $Z(t) = X\cos(2\pi f_0 t) + Y\sin(2\pi f_0 t)$, where X and Y are independent Gaussian random variables with zero mean and variance $\sigma_X^2 = \sigma_Y^2 = \sigma^2$, $m_Z(t) = 0$ and $R_Z(\tau) = \sigma^2\cos(2\pi f_0 \tau)$.
Write a matlab program that simulates 100 realizations of $Z(t)$. Using such realizations, compute $\widehat{m}_Z(t)$ and $\widehat{R}_Z(t_1, t_2)$, **numerically**. Finally, the program should plot the following 3 graphs (all in the same window): The first one will show the simulated realizations (100), each one using a "random" color, as well

as $\widehat{m}_Z(t)$ (plotted in black and dotted line) and the $m_Z(t)$ (plotted in red and dotted line). The second plot will show $\widehat{R}_Z(t_1, t_2)$ (3-d plot), and finally, the third plot will show $R_Z(\tau)$ (2-d plot). All graphs must contain title and labeled axes.
Use the following parameters to simulate $Z(t)$

- $\sigma^2 = \frac{1}{2}$
- $f_0 = \frac{1}{3}$
- $t \in [0, 20]$ (Use a temporal step equal to 0.5 to generate this vector).

Note: plot $R_Z(\tau)$ using an appropriate symmetric range.

3 Statistical Description of a stochastic process

Let $X(t)$ be a Gaussian random process defined by

- $m_X(t) = [0] \in \mathbb{R}^{T\times 1}$
- $Cov(i,j) = Cov(X(t_i), X(t_j)) = \exp(-\frac{|t_i - t_j|^2}{2\sigma^2}) \in \mathbb{R}^{T\times T}$, $\sigma = \frac{1}{\sqrt{2\pi}}$
- `T=length(t)`

Write a matlab program that simulates 20 realizations of $X(t)$. The program should plot the following 3 graphs (all in the same window): The first one will show the simulated realizations (20), each one using a "random" color. The second plot will show $\widehat{R}_X(t_1, t_2)$ computed **numerically** (3-d plot), and finally, the last plot will show the **analytic** $R_X(\tau)$ (2-d plot). All graphs must contain title and labeled axes. Use a temporal vector $t_i, t_j \in t = [0, 0.1, 0.2, \ldots, 10]$ to simulate the realizations of $X(t)$.
Note: plot $R_X(\tau)$ using an appropriate symmetric range.

Bibliography

- *Comunicaciones Digitales.* A. ArtÃ©s, F. PÃ©rez GonzÃ¡lez, J. Cid Sueiro, R. LÃ³*pezValcarce, C.MosqueraNartalloy*
- *Communication Systems Engineering.* J.G. Proakis y M. Salehi. Prentice-Hall. 1994.

4 Appendix

	Matlab Functions
Display help text in the command window	`help command_name`
Similar to `help`, however, it shows the information in a more friendly format	`doc command_name`
Return the number of elements of the vector v	`length(v)`
Return the number of rows and columns of the matrix A	`[row, column]=size(A)`
Generate $m \times n$ samples according to a Gaussian random variable with mean μ and standard deviation σ	`random('norm',`μ`,`σ`,m,n)`
Break the Figure window into m rows and n columns and generate one axes at position p	`subplot(m,n,p)`
Plot the vector x vs y in a bidimensional graph with the colour *color* (Use `help plot` to find the available colours and line styles)	`plot(x,y,'color')`
Use this command to draw plots in the same axes preserving the former plots	`hold on`
Draw a grid on the plot area	`grid on`
Plot the vector x vs y as a discrete sequence (useful for impulse-like sequences)	`stem(x,y)`
Label the x-axis using the text *Message*	`xlabel('Message')`
Label the y-axis using the text *Message*	`ylabel('Message')`
Label the graph using the text *Message*	`title('Message')`
Create a new figure window	`figure`
Generate N samples according to a multidimensional Gaussian random variable with mean μ and covariance matrix Cov	`mvnrnd(`μ`,`Cov`,`N`)`

Table 1: Matlab Commands

$\sin(A)\sin(B) = \frac{\cos(A-B)-\cos(A+B)}{2}$
$\cos(A)\cos(B) = \frac{\cos(A-B)+\cos(A+B)}{2}$

Table 2: Trigonometric Relationships.