

Reading Assignment: Read Sections 9.5 and 10.1-10.5 of the EE 3025 Matlab Notes.

Part I(25 min): Matlab

Part II(25 min): Worked Problems on Chap 10

1 Matlab

1.1 Estimating Process Means and Autocorrelations

Let $X(t)$ be a random process. Suppose we sample at time $t = t_0$, getting the sample $X(t_0)$. If we look at many different realizations and average the $X(t_0)$ values, then we obtain an estimate of $E[X(t_0)]$. Suppose we sample at another time $t = t_1$, and compute the product $X(t_0)X(t_1)$. Looking at many different realizations and averaging the $X(t_0)X(t_1)$ values, we obtain an estimate of the autocorrelation $E[X(t_0)X(t_1)]$ at these two times. This method of estimating process means and autocorrelations by averaging across realizations is called “space averaging”.

Example. We do the preceding estimation procedure for a Poisson arrival process $X(t)$ with arrival rate of 2 arrivals per unit time. At each time $t > 0$, $X(t)$ is the number of arrivals that have occurred in the time interval $[0, t]$. If we sample at time $t = t_0$, then we have

$$E[X(t_0)] = 2t_0.$$

In our Matlab script, we take $t_0 = 4$. Then theoretically we have

$$E[X(4)] = 2(4) = 8.$$

The following script estimates $E[X(4)]$ across 1000 realizations.

```
arr_rate=2;
n=1000; %number of realizations
for i=1:1000
t=0;
arrivaltimes=[];
while t<=4
t=t-log(rand(1,1))/arr_rate; %time of next arrival
arrivaltimes=[arrivaltimes t];
end
x4(i)=sum(arrivaltimes<=4); %X(4) on i-th realization
end
EX4_estimate=mean(x4)

EX4_estimate =

8.0370
```

Note that the estimate is about right. In the script, the “while loop” counts the number of arrivals occurring at times up to and including time 4, which is the random value $X(4)$. The length of the time interval between arrivals is modeled as

$$-\log(\text{rand}(1,1))/\text{arr_rate}$$

which is an exponentially distributed random variable whose mean is the reciprocal of the arrival rate.

Example. With the same arrival rate 2 for our Poisson process $X(t)$, we now estimate the autocorrelation $E[X(2)X(4)]$ across 1000 realizations. From Chap 10 of your textbook, the theoretical value of $E[X(t_0)X(t_1)]$ for $t_0 < t_1$ is:

$$E[X(t_0)X(t_1)] = \lambda t_0 + \lambda^2 t_0 t_1,$$

where λ is the arrival rate. One proves this using the facts that

- $Cov(X(t_0), X(t_1)) = \lambda t_0$.
- $E[X(t_0)] = \lambda t_0$.
- $E[X(t_1)] = \lambda t_1$.

Plugging in $\lambda = 2$, $t_0 = 2$, and $t_1 = 4$, we obtain

$$E[X(2)X(4)] = 36.$$

The following Matlab script estimates this quantity across 1000 realizations of the Poisson process.

```
arr_rate=2;
n=1000; %number of realizations
for i=1:1000
t=0;
arrivaltimes=[];
while t<=4
t=t-log(rand(1,1))/arr_rate; %time of next arrival
arrivaltimes=[arrivaltimes t];
end
x2(i)=sum(arrivaltimes<=2); %X(2) on i-th realization
x4(i)=sum(arrivaltimes<=4); %X(4) on i-th realization
end
aucorr_estimate=mean(x2.*x4)
aucorr_estimate =
```

36.6800

1.2 Plotting Poisson Process Realizations

The Poisson process (also called Poisson arrival process) is used to model arrivals in a queueing system. In this experiment, we show how to plot a realization of a Poisson process.

Consider a Poisson process for which there is one arrival per second, on average. We can simulate the first six arrival times via the Matlab code:

```
t=cumsum(-log(rand(1,6)));
```

Let these six arrival times be $t_1, t_2, t_3, t_4, t_5, t_6$; these are the entries of the vector \mathbf{t} . Consider the step function defined $s(t)$ over the time interval $0 \leq t \leq t_6$ defined as follows: $s(t)$ is equal to 0 in the time interval $0 \leq t < t_1$, is equal to 1 in the time interval $t_1 \leq t < t_2$, 2 in the time interval $t_2 \leq t < t_3$, etc., ending up equal to 5 in the time interval $t_5 \leq t \leq t_6$. If we were to consider more and more arrivals until we had infinitely many, then the step function $s(t)$ would keep getting extended until the result would be a realization of the Poisson process. We plot $s(t)$ using the following Matlab script:

```
t=cumsum(-log(rand(1,6)));
t=round(10^3*t)/10^3; %rounds arrival time to 3 decimal places
delta=.001;
x1=0:delta:t(1)-delta; y1=0*ones(1,length(x1));
x2=t(1):delta:t(2)-delta; y2=1*ones(1,length(x2));
x3=t(2):delta:t(3)-delta; y3=2*ones(1,length(x3));
x4=t(3):delta:t(4)-delta; y4=3*ones(1,length(x4));
x5=t(4):delta:t(5)-delta; y5=4*ones(1,length(x5));
x6=t(5):delta:t(6)-delta; y6=5*ones(1,length(x6));
plot([x1 x2 x3 x4 x5 x6],[y1 y2 y3 y4 y5 y6])
axis([0 t(6) 0 6])
xlabel('time axis t')
ylabel('number of arrivals s(t) up through time t')
title('plot of realization of Poisson process')
```

Fig.1 gives the realization plot when we got the one time we ran the above script. Of course, the realization plot will change each time you run the script.

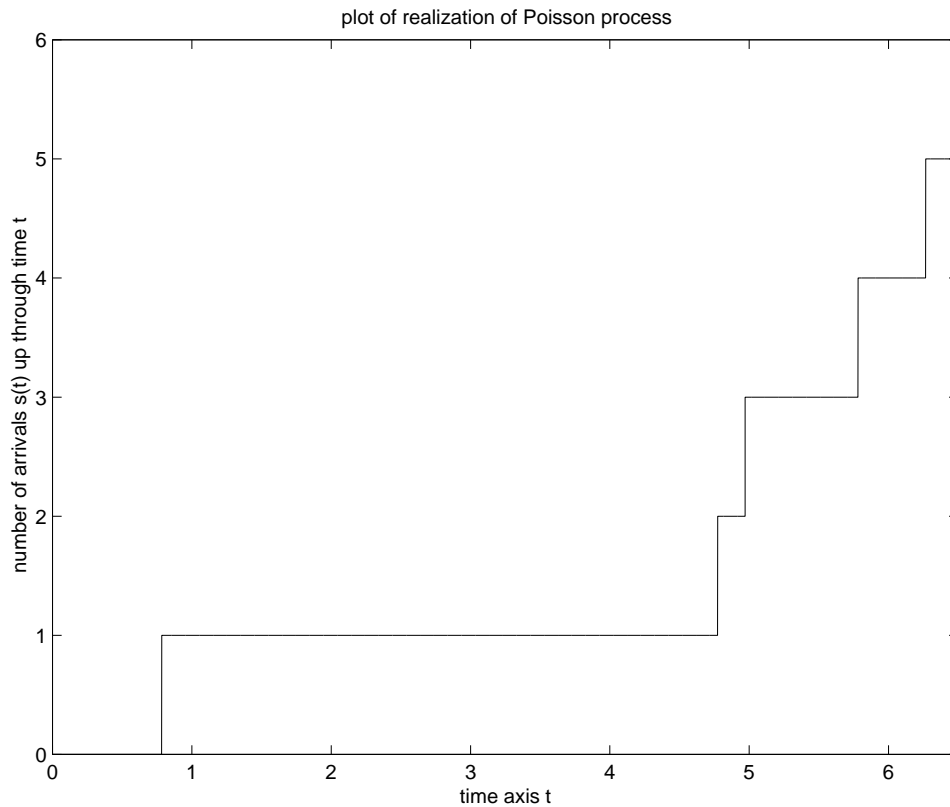


Figure 1: Plot of Poisson Process Realization (Arrival Rate = 1)

Look at your realization $s(t)$, and answer the following questions:

- At what times do each of the first 6 arrivals occur?
- What is the length of time between the first arrival and the second arrival, or between the second arrival and the fourth arrival?
- What does the arrival rate seem to be in number of arrivals per second (approximately)?

Re-run the preceding lines of code repeatedly to get other realizations. They should all give you roughly 6 arrivals in the first 6 seconds. But of course no two realizations will be the same.

Exercise. Suppose you now want the arrival rate of the Poisson process to be two arrivals per second. Then, you replace the first line of the Matlab script with

```
t=cumsum(-log(rand(1,6))/2);
```

Run the preceding Matlab script after making this change and then look at the realization of the Poisson process that you see plotted on your Matlab screen. It should look somewhat like Figure 2.

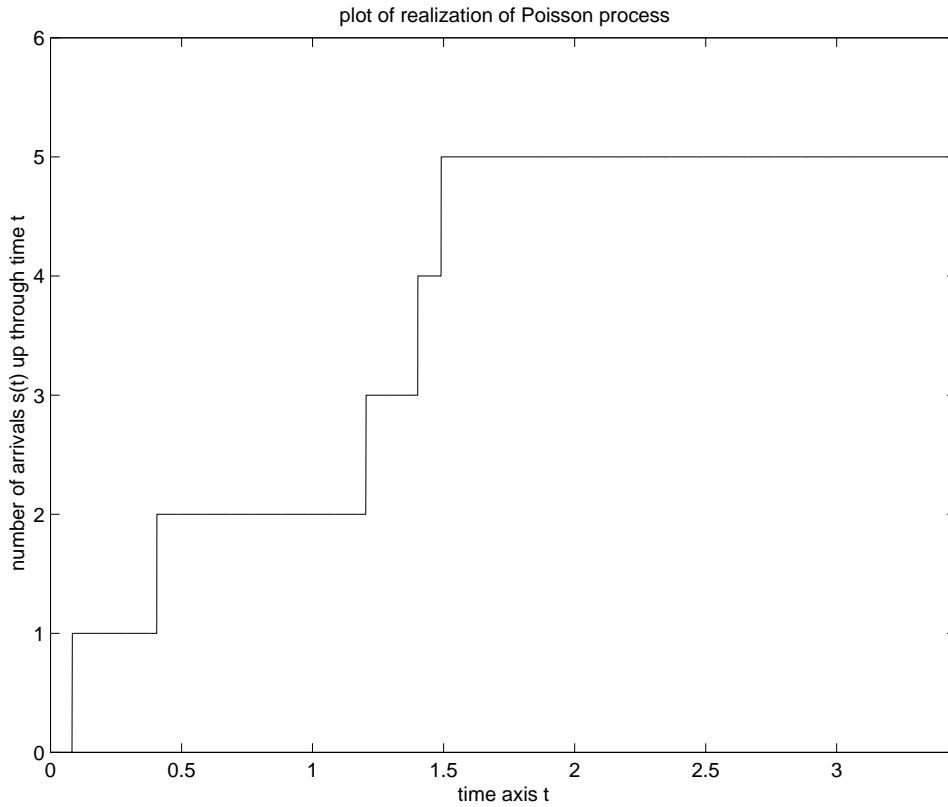


Figure 2: Plot of Poisson Process Realization (Arrival Rate = 2)

1.3 Predictor Design for WSS Process

Let (X_n) be a WSS process. Letting $R_X(\tau)$ be the autocorrelation function, we can compute autocorrelations as

$$E[X_m X_n] = R_X(|m - n|).$$

The minimum mean-square first, second, and third order linear predictors for X_n take the form:

$$\begin{aligned}\hat{X}_n &= AX_{n-1} \\ \hat{X}_n &= BX_{n-1} + CX_{n-2} \\ \hat{X}_n &= DX_{n-1} + EX_{n-2} + FX_{n-3}\end{aligned}$$

In each case, the predictor coefficients are selected so that the mean-square prediction error

$$E[(X_n - \hat{X}_n)^2]$$

is a minimum. The orthogonality principle can be used to find the predictor coefficients, and the solutions are:

- The first order predictor coefficient A is given by the equation

$$A = R_X(1)/R_X(0). \quad (1)$$

- The second order predictor coefficients B, C are obtained by solving the matrix equation:

$$\begin{bmatrix} R_X(0) & R_X(1) \\ R_X(1) & R_X(0) \end{bmatrix} \begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} R_X(1) \\ R_X(2) \end{bmatrix} \quad (2)$$

- The third order predictor coefficients D, E, F are obtained by solving the matrix equation:

$$\begin{bmatrix} R_X(0) & R_X(1) & R_X(2) \\ R_X(1) & R_X(0) & R_X(1) \\ R_X(2) & R_X(1) & R_X(0) \end{bmatrix} \begin{bmatrix} D \\ E \\ F \end{bmatrix} = \begin{bmatrix} R_X(1) \\ R_X(2) \\ R_X(3) \end{bmatrix} \quad (3)$$

Example. Let the WSS process X_n have autocorrelation function

$$R_X(\tau) = \begin{cases} 5, & \tau = 0 \\ -1, & \tau = \pm 1 \\ 2, & \tau = \pm 2 \\ 0, & \text{elsewhere} \end{cases}$$

Run the following Matlab script, which separately computes the predictor coefficients for the first, second, and third order predictor:

```
R=[5 -1 2 0];
%find first order predictor coefficient A
A = R(2)/R(1);
%find second order predictor coefficients B,C
Q2=inv(toeplitz(R(1:2)))*R(2:3)';
B=Q2(1), C=Q2(2);
%find third order predictor coefficients D,E,F
Q3=inv(toeplitz(R(1:3)))*R(2:4)';
D=Q3(1), E=Q3(2), F=Q3(3);
[A 0 0; B C 0; D E F]
```

Observe what you get on your computer screen from running the preceding script. You should see a 3×3 matrix. The first entry in the first row gives the first order predictor coefficient A . The first two entries in the second row give the second order predictor coefficients B, C . The third row gives the third order predictor coefficients D, E, F . This matrix gives a convenient way to express all three of these predictors.

Note the Matlab function “`toeplitz`” in the script. This function allows one to easily generate a square matrix constant along the diagonal and subdiagonals. For example, the command

$$\text{toeplitz}([x \ y \ z])$$

generates the 3×3 matrix

x y z
y x y
z y x

What does the command `toeplitz([x,y,z,w])` generate?

Prediction Errors. The mean-square prediction errors $E[(X_n - \hat{X}_n)^2]$ for the first, second, and third order predictors are given by the formulas

$$\text{first order pred error} = R_X(0) - AR_X(1)$$

$$\text{second order pred error} = R_X(0) - BR_X(1) - CR_X(2)$$

$$\text{third order pred error} = R_X(0) - DR_X(1) - ER_X(2) - FR_X(3)$$

For example, for the 3rd order predictor, this is because

$$E[(X_n - \hat{X}_n)^2] = E[X_n - \hat{X}_n]X_n = E[X_n^2] - DE[X_n X_{n-1}] - EE[X_n X_{n-2}] - FE[X_n X_{n-3}],$$

the quantity $E[(X_n - \hat{X}_n)\hat{X}_n] = 0$ having dropped out due to the orthogonality principle.

Exercise. Use Matlab to compute the prediction errors for the first three predictors.

2 Worked Problems

2.1 Auto- and Cross-Correlation

$X(t)$ is a WSS process with mean 0 and autocorrelation function

$$R_X(\tau) = \exp(-|\tau|).$$

Θ is uniformly distributed in $[0, 2\pi]$ and independent of $X(t)$. $Y(t)$ and $Z(t)$ are the processes

$$Y(t) = X(t) \cos(2\pi f_0 t + \Theta).$$

$$Z(t) = X(t - 1) \cos(2\pi f_0 t + \Theta).$$

- (a) Are $Y(t)$ and $Z(t)$ WSS? If so, what are their mean and autocorrelation functions?

Solution.

$$E[Y(t)] = E[X(t)]E[\cos(2\pi f_0 t + \Theta)] = 0$$

$$E[Z(t)] = E[X(t - 1)]E[\cos(2\pi f_0 t + \Theta)] = 0$$

$$E[Y(t)Y(t + \tau)] = E[X(t)X(t + \tau)]E[\cos(2\pi f_0 t + \Theta) \cos(2\pi f_0(t + \tau) + \Theta)] = R_X(\tau)(1/2)\{E[\cos(2\pi f_0 \tau)] + E[\cos(2\pi f_0(2t + \tau) + 2\Theta)]\}.$$

The second term within the braces is of the form

$$\int_0^{2\pi} (1/2\pi) \cos(C + 2\theta) d\theta = 0.$$

Therefore,

$$E[Y(t)Y(t + \tau)] = (1/2)R_X(\tau) \cos(2\pi f_0 \tau).$$

Similarly,

$$E[Z(t)Z(t + \tau)] = (1/2)R_X(\tau) \cos(2\pi f_0\tau).$$

We conclude that $Y(t)$ and $Z(t)$ are both WSS and both have mean 0 and

$$R_Y(\tau) = R_Z(\tau) = (1/2) \exp(-|\tau|) \cos(2\pi f_0\tau).$$

(b) Determine the cross-correlation function

$$R_{ZY}(t, \tau) = E[Z(t)Y(t + \tau)].$$

Solution.

$$E[Z(t)Y(t + \tau)] = E[X(t - 1)X(t + \tau)]E[\cos(2\pi f_0t + \Theta) \cos(2\pi f_0(t + \tau) + \Theta)].$$

The first expected value on the right is

$$E[X(t - 1)X(t + \tau)] = R_X(\tau + 1).$$

The second expected value was already treated in (a) solution. Therefore,

$$R_{ZY}(t, \tau) = (1/2) \exp(-|\tau + 1|) \cos(2\pi f_0\tau).$$

2.2 Nonstationary $X(t)$ Stable as $t \rightarrow \infty$

Worked Problem 5.2.4.

2.3 Ergodic WSS Processes

Worked Problems 5.3.2-5.3.4 as time permits.

Background. A discrete-time ergodic WSS process (X_n) has the property that process parameters can be computed as time averages along realizations. For example, you'll have

$$\mu_X = \lim_{N \rightarrow \infty} (1/N) \sum_{n=1}^N X_n$$

with probability one, and for any fixed τ ,

$$R_X(\tau) = \lim_{N \rightarrow \infty} (1/N) \sum_{n=1}^N X_n X_{n+\tau}$$

with probability one. Also,

$$\mu_X^2 = \lim_{\tau \rightarrow \infty} R_X(\tau),$$

if the limit exists.