Boston University
Department of Electrical and Computer Engineering

EC505 STOCHASTIC PROCESSES

Problem Set No. 6

Fall 2016

Issued: Wednesday, Oct. 26, 2016 **Due:** Wednesday, Nov. 2, 2016

Problem 6.1

In this problem we study the statistics of "shot noise." This point process is a model for the noise arising in many physical situations. We take a "systems" perspective here. Let the points $\{\tau_i\}$ correspond to the random arrivals of a Poisson counting process $N(t)$ with parameter λ. We define the shot noise process $X(t)$ as follows:

$$X(t) = \sum_{i=-\infty}^{\infty} h(t - \tau_i)$$

where $h(t) = e^{-\alpha t}u(t)$

(a) Show that we can model $X(t)$ as the output of a linear time-invariant system with input $Z(t)$, where $Z(t)$ is obtained as the derivative of the underlying Poisson counting process $N(t)$.

(b) Using the results for the statistics of derivatives of processes, find the mean $m_Z(t)$ and autocorrelation $R_{ZZ}(t, s)$ functions for the Poisson point process $Z(t)$. Note: $N(t)$ is not MS differentiable, so generalized functions will be required. You should find that $Z(t)$ is wide-sense stationary.

(c) Use your model from (a) and the statistics you found in part (b) together with results on processes through linear systems to find the mean $m_X(t)$, power spectral density $S_{XX}(f)$, autocorrelation $R_{XX}(\tau)$ function and variance of $X(t)$.

Problem 6.2 (Shanmugan and Breipohl 5.1)

This problem shows that the first-order, state-space form actually captures higher order systems. As a consequence, results presented for such first-order state-space systems are applicable to a broad range of practical problems. Consider the autoregressive process represented by the following p-th order scalar difference equation:

$$X(n) = \sum_{i=1}^{p} h_i X(n - i) + E(n)$$

Show that this p-th order difference equation can be written in the following first-order state-space form where $\mathbf{X}(n)$ is a vector and $\mathbf{\Phi}$ is a matrix:

$$\mathbf{X}(n) = \mathbf{\Phi}\mathbf{X}(n - 1) + \mathbf{E}(n)$$

Problem 6.3 (Shanmugan and Breipohl 5.13)

Suppose that $X(t)$ is a second-order autoregressive process

$$X(n) = a_1 X(n - 1) + a_2 X(n - 2) + W(n)$$

with $R_{XX}(1)/R_{XX}(0) = 0.5$ and $R_{XX}(2)/R_{XX}(0) = 0.1$, and $W(n)$ white noise with autocorrelation $R_{WW}(m) = \sigma_W^2 \delta(m)$.

(a) Find the two AR parameters a_1 and a_2.

(b) Give a *closed form* expression for $r_{XX}(m) \equiv R_{XX}(m)/R_{XX}(0)$ for $m \geq 2$ (i.e. solve the difference equation for $r_{XX}(m)$).

Problem 6.4

An EC505 student is trying to decide whether he should undergo a risky operation. He consults three independent experts who each give him a binary (i.e. yes or no) answer as to whether he should have the operation. The experts have different degrees of reliability. Expert 1 is correct with probability p_1, expert 2 is correct with probability p_2, and expert 3 is correct with probability p_3. The student desires an optimal way of combining the answers of the experts to maximize the probability of making a correct overall decision whether to operate. The a priori probability of being a good candidate for the operation is $\Pr(\text{yes}) = 1/2$. You can assume that the experts make decisions independent of one another.

(a) Formulate the problem as a detection problem by identifying the observation space and the probability associated with each observation, conditioned on the hypotheses $H_0 = \text{no}$ and $H_1 = \text{yes}$. (Hint: There are 8 possible observation values, so just make a table)

(b) Find the optimal decision rule that minimizes the probability of making a mistake for the following two cases: i) $p_1 = 0.8$, $p_2 = 0.6$, $p_3 = 0.7$ and ii) $p_1 = 0.75$, $p_2 = 0.6$, $p_3 = 0.7$. Try to simplify your rules as much as possible.

Problem 6.5

In the binary communication system shown in Figure 1, messages $X = 0$ and $X = 1$ occur with a priori probabilities $\frac{1}{4}$ and $\frac{3}{4}$ respectively. Suppose that we observe R,

$$R = X + N,$$

where N is a continuous valued random variable with the probability density function shown in Fig. 2.

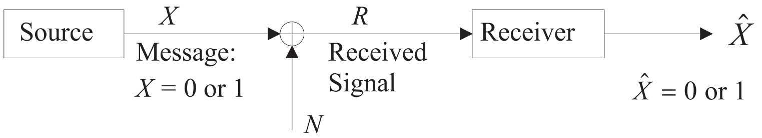

Figure 1

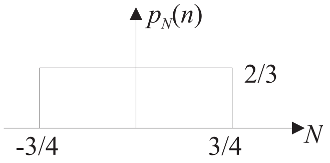

Figure 2

(a) Find the minimum probability of error detector, and compute the associated probability of error, $\Pr(\hat{X} \neq X)$.

(b) Suppose that the receiver does not know the true a priori probabilities, so it decides to use a maximum likelihood (ML) detector (i.e. the detector will *assume* something about the prior probabilities, which may in fact be different from these assumptions). Find the ML detector and the associated probability of error if this decision rule is applied when the true prior probabilities are in fact different. Is the ML detector unique? Justify your answer. If your answer is no, find a different ML receiver and the associated probability of error.

Problem 6.6

You are a lawyer for the EPA. In a community near Boston, there is the suspicion that the ground-water has been polluted by a large chemical company. Two hypotheses exist: under hypothesis H_1 the ground-water is polluted and will cause cancer, and under hypothesis H_0 the ground-water is fine and drinkable. Assume that the results of tests on the ground-water obtained from a series of pilot wells in the area are summarized by the value of a random variable X. If the ground-water is polluted the random variable X has the following pdf

$$p_{X|H_1}(x \mid H_1) = xe^{-x}u(x)$$

where $u(x)$ is the unit step function. In the case that the ground-water is fine the random variable X obeys the following pdf

$$p_{X|H_0}(x \mid H_0) = e^{-x}u(x).$$

You must decide, based on the value of X that is obtained from the well tests, whether or not to initiate legal action against the chemical company. Design a hypothesis testing rule (find decision regions) such that, P_D, the probability that you decide to sue given that the ground-water is polluted is present is maximized subject to the constraint that the probability of suing given the ground-water is fine is 0.1. Find the resulting value of P_D.

Problem 6.7

In a radar system, a decision about the presence or absence of a target is made on the basis of an observation Y. If the target is present, $Y = A + X$, where A is a known constant. If the target is not present, $Y = X$. X has normal distribution: $X \sim N(0, N_0)$. The *a priori* probability that the target is not present is 0.999.

(a) Derive the MAP decision rule for detecting the target and the associated probability of error.

(b) Now, assume that missing the target is ten times worse than falsely detecting a target: $C_{00} = C_{11} = 0$, $C_{01} = 10$ and $C_{10} = 1$, where H_0 is "target absent" and H_1 is "target present". What is the decision rule which minimizes the conditional risk for y?

(c) Sketch a hypothetical ROC curve for this problem, illustrating the two points corresponding to the two decision rules. Make a sketch to illustrate qualitatively how the ROC curve would change if you were to have 5 independent observations to base your decision on. Make a sketch to illustrate qualitatively how the ROC curve would change if the noise level increased (i.e. A/N_0 decreased).

Computer Problems

Problem 6.8 Power Spectral Density Estimation via AR Modeling

The FFT-based spectrum estimator you created in Problem Set No. 5 is sometimes termed a "model-free" or "nonparametric" approach, since minimal assumptions are made about the underlying random process. In particular, a separate spectral estimate is generated at each frequency. This means that many parameters (one for each frequency) are being estimated from the data. Further, the resolution of the estimate is limited by the resolution of the FFT, which is determined by the length of the estimated autocorrelation sequence, which in turn is limited by the amount of data we have. In this project we will study an alternative approach, based on identification of an AR model. Such a so-called "modern" spectral estimation method is based on a model for the underlying data, and hence, is sometimes termed "model-based." Data is used more efficiently, by focusing the given information into a few unknowns and the resolution limit imposed by FFT-based techniques is circumvented since an analytic model is found. This topic is discussed in more detail in the reserve texts Shanmugan and Breipohl, Section 9.4 and Therrien, Section 10.2.

The specific method we will study in this problem is sometimes called the "autocorrelation method" for reasons that will become obvious. It models the observed process as the output of an all-pole, autoregressive model of the form:

$$Y(n) + a_2 Y(n-1) + a_3 Y(n-2) + \cdots a_{N_p+1} Y(n-N_p) = b_1 W(n) \tag{1}$$

where $W(n)$ is unit variance white noise and N_p is the number of poles in the model. Note that the coefficient of the $Y(n)$ term would be a_1 and we have assumed that $a_1 = 1$. In the above model the sign of the coefficients a_i is the opposite of that found in the notes, in keeping with MATLAB's convention. The final spectral estimate in then based on this model and is thus of the form:

$$S_{YY}(\omega) = \frac{b_1^2}{\left|A(e^{j\omega})\right|^2} = \frac{b_1^2}{\left|1 + a_2 e^{-j\omega} + a_3 e^{-j2\omega} + \cdots + a_{N_p+1} e^{-jN_p\omega}\right|^2} \tag{2}$$

Thus the problem of spectrum estimation is reduced to the problem of estimating the coefficients of this AR model. Since the spectrum is defined by an analytic expression, we can find its value for any choice of ω once the model is found.

Recall from class that the AR coefficients in (1) and the corresponding autocorrelation function values $R_{YY}(k)$ for such an AR process are related by the Yule-Walker equations:

$$\begin{bmatrix} R_{YY}(0) & R_{YY}(1) & R_{YY}(2) & \cdots & R_{YY}(N_p-1) \\ R_{YY}(1) & R_{YY}(0) & R_{YY}(1) & R_{YY}(2) & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ R_{YY}(N_p-1) & \cdots & R_{YY}(2) & R_{YY}(1) & R_{YY}(0) \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \\ \vdots \\ a_{N_p+1} \end{bmatrix} = - \begin{bmatrix} R_{YY}(1) \\ R_{YY}(2) \\ R_{YY}(3) \\ \vdots \\ R_{YY}(N_p) \end{bmatrix} \tag{3}$$

where the minus sign is due to our use of MATLAB's coefficient sign convention. In addition, the DC gain term b_1 is given by:

$$b_1^2 = R_{YY}(0) + \sum_{k=1}^{N_p} R_{YY}(k) a_{k+1} \tag{4}$$

We will use these equations as the basis for our estimation procedure.

(a) Write a function `psdest2.m` to estimate the power spectral density of a process given sample path data in the vector `y`. The function will first estimate the autocorrelation function of the process and then use these values to estimate an AR model. The function call for your program will be the following: `[Syy,w,a,b] = psdest2(y,Np,Nf)`, where `y` is the input data vector, `Np` will be the number of poles in the AR model, `Nf` is the number of frequency samples to generate, `Syy` will be the estimated power spectral density, `w` will be the corresponding vector of frequency values, and `a` and `b` will be the vectors of denominator and numerator coefficients defining the estimated AR model. Each step below will form a part of the program:

(i) The first step is to estimate the values of the autocorrelation function for lags between `-Np` and `Np` and only keep the positive lag values:

```
[Ryy,lag] = xcorr(y,Np,’unbiased’);
Ryy = Ryy(Np+1:2*Np+1);
Ryy = Ryy(:);
```

The second line just picks out the positive lag values, which is all we need to setup and solve the Yule-Walker equations (3). The last line simply transforms these values into a column vector. Note that compared to the FFT-based method of Problem Set No. 5, we are only estimating a few, low order autocorrelation values.

(ii) Now we form and solve the linear Yule-Walker equations for the AR coefficients:

```
a = toeplitz(Ryy(1:Np)) \ (-Ryy(2:Np+1));
a = [1 a’];
```

The first line uses the MATLAB function `toeplitz.m` to create the Toeplitz matrix defined in the Yule-Walker equations, i.e. the left-hand side of (3). The MATLAB “backslash” operator is then used to perform the “matrix division” into the right hand side of (3) thus estimating the non-unitary AR coefficients. The second line creates a vector of system denominator coefficients following MATLAB's standard system definition, in particular, we prepend a “1” to the estimated coefficients since we have assumed that $a_1 = 1$.

(iii) Now we need to find the DC gain term b_1^2 using (4):

```
b2 = a*Ryy;
b = [sqrt(b2) 0];
```

The first line performs the sum operation in (4), while the second line creates the vector of the system numerator coefficients following MATLAB's standard system definition. We append a “0” to the coefficient vector due to a quirk in MATLAB's function `zplane.m`. Note that this does not change the system definition.

(iv) Finally, we use the AR system we have identified, and represented in the vectors `b` and `a` to generate an associated estimated power spectrum using the function `syspsd.m`:

```
[Syy,w] = syspsd(b,a,Nf);
```

Add these steps together to create your program. Test your function by estimating and plotting the power spectral density of the Gaussian white noise process `u` of Problem Set No. 5 vs frequency. Use `Np = 1` (i.e. a one-pole AR model) and `Nf = 100`;. You may wish to use `semilogy.m` for your plots. Also plot the pole/zero locations of your estimated AR model using `zplane.m`. Explain your estimated pole/zero locations for the white noise spectral model.

(b) Use your function `psdest2.m` to estimate the power spectral densities of the outputs of the 5 given systems for the white Gaussian noise input `u`. Plot your results on the same axes as the theoretical power spectral density obatined using `syspsd.m`. Use `Np = 2` and `Nf = 100`. Again, you may wish to use `semilogy.m` for your plots so you can see more detail. Compare to the power spectral density estimates you obtained using the FFT-based technique in Problem Set No. 5.

(c) In the model-based approach the number of poles (i.e. the model order) must be selected a priori. While it is an advantage to know the true value of `Np`, the procedure is relatively insensitive to overestimation of the model order. Use the output of the second-order AR System #0 and investigate the effect of using some higher model orders (`Np` = 2,3,5,10,...) by comparing the estimated and true power spectral density. Does the quality of the estimate depend strongly on model order? Compare the true and estimated pole/zero locations for these higher order models using `zplane.m`.

While the model based method appears robust to overestimation of model order, what happens if you *underestimate* the model order? Compare the true and estimated power spectrum for output of the second-order System #0 using `Np=1`. Comments on the estimate.

(d) Another issue of interest is what happens if the spectrum we are trying to estimate is not all-pole – i.e. if the true system has both poles *and* zeros. Use the output of the second-order ARMA (pole/zero) System #1 and investigate the effect of using a variety of model orders `Np` = 1,2,3,6,12,... in estimating the power spectrum. Comment on the differences and whether you think the AR-based technique is still reasonable in estimating the power spectra of general processes. Again, compare both the true and estimated pole/zero locations for these models using `zplane.m`.