Boston University
Department of Electrical and Computer Engineering

EC505 STOCHASTIC PROCESSES

Problem Set No. 6 Solutions

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.

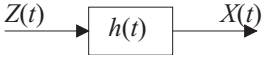
(c) Use your model from (a) and the statistics you found in part (b) together with results on processes through linear systems to find the mean $m_X(t)$, power spectral density $S_{XX}(f)$, autocorrelation $R_{XX}(\tau)$ function and variance of $X(t)$.

Solution:

(a) We can represent $X(t)$ by the model in the figure, so that $X(t) = h(t) * Z(t)$ where

$$Z(t) = \sum_i \delta(t - \tau_i)$$

where $Z(t) = dN(t)/dt$ is the derivative of the counting process $N(t)$. The impulses occur at the steps of the counting process. Note that $H(f) = \frac{1}{\alpha + j2\pi f}$.



(b) First recall that $m_N(t) = \lambda t$ and $R_{NN}(t, s) = \lambda^2 ts + \lambda \min(t, s)$. Now from the relationships between a process and its derivative we have:

$$m_Z(t) = E[dN(t)/dt] = \frac{dm_N(t)}{dt} = \lambda$$

$$R_{ZZ}(t, s) = \frac{\partial^2 R_{NN}(t, s)}{\partial t \partial s} = \lambda^2 + \lambda\delta(s - t)$$

Note that indeed the mean is constant and the autocorrelation function depends on time difference only, hence $Z(t)$ is wide-sense stationary. It is easy to see that:

$$S_{ZZ}(f) = \lambda^2\delta(f) + \lambda$$

(c) From our results on wide-sense stationary stochastic processes through linear time-invariant systems we immediately have that:

$$m_X(t) = m_Z H(0) = \frac{\lambda}{\alpha}$$

$$S_{XX}(f) = |H(f)|^2 S_{ZZ}(f) = \frac{1}{\alpha^2 + 4\pi^2 f^2}\left(\lambda^2\delta(f) + \lambda\right) = \frac{\lambda}{\alpha^2 + 4\pi^2 f^2} + \frac{\lambda^2}{\alpha^2}\delta(f)$$

$$R_{XX}(\tau) = \frac{\lambda}{2\alpha}e^{-\alpha|\tau|} + \frac{\lambda^2}{\alpha^2}$$

$$K_{XX}(\tau) = R_{XX}(\tau) - m_X^2 = \frac{\lambda}{2\alpha}e^{-\alpha|\tau|} + \frac{\lambda^2}{\alpha^2} - \frac{\lambda^2}{\alpha^2} = \frac{\lambda}{2\alpha}e^{-\alpha|\tau|}$$

Thus the variance is given by:

$$\mathrm{Var}[X(t)] = K_{XX}(0) = R_{XX}(0) - m_X^2 = \frac{\lambda}{2\alpha}$$

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:

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

Solution:

The purpose of this problem is to show that any p-th order scalar AR model can always be represented as an equivalent *first*-order state vector AR model. This is also true for MA systems, so this form (known as "state space" form) is quite general. Writing out the terms of the AR model we have:

$$X(n) \quad = \quad h_1 X(n-1) + h_2 X(n-2) + \cdots + h_p X(n-p) + E(n)$$

Now we need to define what we will use as the vector state $\mathbf{X}(n)$ and the vector noise $\mathbf{E}(n)$. The form of the above equation suggests we use the following definitions:

$$\mathbf{X}(n) = \begin{bmatrix} X(n-p+1) \\ \vdots \\ X(n-1) \\ X(n) \end{bmatrix} \qquad \mathbf{E}(n) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ E(n) \end{bmatrix}$$

Thus we see that with this notation we obtain:

$$\mathbf{X}(n) \quad = \quad \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & & & & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ h_p & h_{p-1} & h_{p-2} & h_{p-3} & \cdots & h_1 \end{bmatrix} \mathbf{X}(n-1) + \mathbf{E}(n)$$

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 $r_{XX}(m)$).

Solution:

(a) Using the Yule-Walker equations, one has the following matrix expression:

$$\begin{bmatrix} .5 \\ .1 \end{bmatrix} = \begin{bmatrix} 1 & .5 \\ .5 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

Solving this one finds that:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 3/5 \\ -1/5 \end{bmatrix}$$

(b) To get a general formula for $r_{XX}(m)$, recall that $R_{XX}(m) = E[X(n)X(n-m)] = E\left[(a_1 X(n-1) + a_2 X(n-2) + W(n))\right.$ $a_1 R_{XX}(m-1) + a_2 R_{XX}(m-2)$. Thus, the normalized autocorrelation coefficients also satisfy the same recursion:

$$r_{XX}(m) = a_1 r_{XX}(m-1) + a_2 r_{XX}(m-2)$$

This is a constant coefficient, linear difference equation. Its solutions are given as

$$r_{XX}(m) = C_1 \lambda_1^m + C_2 \lambda_2^m$$

for some constants C_1, C_2, where λ_1 and λ_2 are the roots of the characteristic equation

$$\lambda^2 - a_1 \lambda - a_2 = 0$$

The solutions are:

$$\lambda = \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} = \frac{3/5 \pm \sqrt{9/25 - 4/5}}{2} = \frac{3 \pm \sqrt{-11}}{10} = \frac{3 \pm j\sqrt{11}}{10} = .3 \pm .3317j$$

Thus the solution is of the form:

$$r_{XX}(m) = C_1 \left(\frac{3 + \sqrt{-11}}{10}\right)^m + C_2 \left(\frac{3 - \sqrt{-11}}{10}\right)^m$$

The constants C_1 and C_2 can be found by using the initial conditions $r_{XX}(1) = .5$ and $r_{XX}(2) = .1$.

$$\begin{bmatrix} .5 \\ .1 \end{bmatrix} = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_1^2 & \lambda_2^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

This yields:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} .5 - .3015j \\ .5 + .3015j \end{bmatrix}$$

Thus:

$$\begin{aligned} r_{XX}(m) &= 2\mathrm{Re}\left[(.5 - .3015j)(0.3 + .3317j)^m\right] \\ &= 1.1677(.4472)^m \cos\left(m0.8355 - .5426\right) = 1.1677(.4472)^m \cos\left(m47.8696^\circ - 31.0909^\circ\right) \end{aligned}$$

Note that since we are not told the value of $R_{XX}(0)$ but rather only the *normalized* the values of the autocorrelation coefficients, we cannot determine σ_W^2 the process noise variance. Note that this did not stop us from finding the coefficients a_i of the AR model. This value serves only to scale the value of the output process and thus determines the DC gain of the system, not the pole locations. Another way of looking at it is that we have been given a trajectory $r_{XX}(m)$ of the system that has been scaled by the first value $R_{XX}(0)$ of the trajectory. Thus we know the shape of the trajectory, and so can determine the relative dynamics of the system, but, being scaled, we cannot determine the DC Gain of the system. That σ_W^2 determines this gain for the system can be seen by an examination of its place in the system transfer function.

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.

Solution:

(a) The problem is to decide if the operation should be done, given the doctors opinions. Let H_0 be the hypothesis that an operation should not be performed and H_1 be the hypothesis that an operation should be performed. Now expert 1 is correct with probability p_1, expert 2 is correct with probability p_2, expert 3 is correct with probability p_3. This means e.g. that if you really should operate (i.e., if H_1 is true), expert 1 says "operate" with probability p_1 and says "don't operate" with probability $1 - p_1$, etc. Let the statements of the 3 experts be denoted by the random variables Y_1, Y_2, Y_3, respectively, and let the values of the Y_i be 0 if an expert says "don't operate" and 1 if he says "operate". We will denote the overall observation by the triple $Y = [Y_1, Y_2, Y_3]$. Then the following observations are possible along with the associated conditional probabilities:

$[y_1, y_2, y_3]$	$\Pr(Y = y\|H_1)$	$\Pr(Y = y\|H_0)$
000	$(1-p_1)(1-p_2)(1-p_3)$	$p_1p_2p_3$
001	$(1-p_1)(1-p_2)p_3$	$p_1p_2(1-p_3)$
010	$(1-p_1)p_2(1-p_3)$	$p_1(1-p_2)p_3$
011	$(1-p_1)p_2p_3$	$p_1(1-p_2)(1-p_3)$
100	$p_1(1-p_2)(1-p_3)$	$(1-p_1)p_2p_3$
101	$p_1(1-p_2)p_3$	$(1-p_1)p_2(1-p_3)$
110	$p_1p_2(1-p_3)$	$(1-p_1)(1-p_2)p_3$
111	$p_1p_2p_3$	$(1-p_1)(1-p_2)(1-p_3)$

where $\Pr(Y = y|H_i) = p_{Y_1,Y_2,Y_3|H_i}(y_1, y_2, y_3|H_i) = \prod_j p_{Y_j|H_i}(y_j|H_i)$, since the y_i are either "operate" (1) or "don't operate" (0) independently.

In addition, from the statement of the problem we also know that the prior probabilities of operating and not operating are equal, thus $\Pr_H(H_0) = \Pr_H(H_1) = 1/2$. At this point, the costs are not

determined. Thus, the optimal Bayes decision rule would be given by the likelihood ratio test:

$$\mathcal{L}(y) = \frac{P(y|H_1)}{P(y|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})} = \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})}$$

where the data likelihoods $P(y|H_1)$ and $P(y|H_0)$ are specified in the table above.

(b) Since in part b) we are asked to find the decision rule that minimizes the probability of error, we want the MPE cost assignment: $C_{ij} = 1 - \delta_{ij}$. The resulting minimum probability of error decision rule is given by:

$$\mathcal{L}(y) = \frac{P(y|H_1)}{P(y|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})} = 1$$

where, again, the data likelihoods $P(y|H_1)$ and $P(y|H_0)$ are specified in the table above.

While it is not necessary, by examining the likelihood ratios, the general decision rule given the observations can be written in the following form:

$$\mathcal{L}(y) = \prod_{i=1}^{3} \left(\frac{p_i}{1 - p_i} \right)^{(2y_i - 1)} \underset{H_0}{\overset{H_1}{\gtrless}} 1$$

The associated probability of error is given by:

$$\Pr(\text{err}) = \Pr(\text{yes}|H_0)P(H_0) + \Pr(\text{no}|H_1)P(H_1) = \frac{1}{2}\left[\Pr(\text{yes}|H_0) + \Pr(\text{no}|H_1)\right]$$

Now let us examine the two given cases. We can find the decision rule for each case by creating a table similar to that for the data likelihoods for each case.

i) When $p_1 = 0.8$, $p_2 = 0.6$, $p_3 = 0.7$ we have:

$[y_1, y_2, y_3]$	$\Pr(Y = y\|H_1)$	$\Pr(Y = y\|H_0)$	Likelihood ratio	Decision
000	0.024	0.336	1/14	No
001	0.056	0.144	7/18	No
010	0.036	0.224	9/56	No
011	0.084	0.096	7/8	No
100	0.096	0.084	8/7	Yes
101	0.224	0.036	56/9	Yes
110	0.144	0.056	18/7	Yes
111	0.336	0.024	14/1	Yes

We can see that the overall decision rule can be reduced to the rule "Use expert #1". The associated probability of error is:

$$\Pr(\text{err}) = \frac{1}{2}\left[(0.084 + 0.036 + 0.056 + 0.024) + (0.024 + 0.056 + 0.036 + 0.084)\right] = 0.2 = 1 - p_1$$

Which we could have guessed intuitively – because if we're only relying on expert number 1 now – the only error is the probability that that expert will err.

ii) When $p_1 = 0.75$, $p_2 = 0.6$, $p_3 = 0.7$ we have:

$[y_1, y_2, y_3]$	$\Pr(Y = y\|H_1)$	$\Pr(Y = y\|H_0)$	Likelihood ratio	Decision
000	0.03	0.315	2/21	No
001	0.07	0.135	14/27	No
010	0.045	0.21	3/14	No
011	0.105	0.09	7/6	Yes
100	0.09	0.105	6/7	No
101	0.21	0.045	14/3	Yes
110	0.135	0.07	27/14	Yes
111	0.315	0.03	21/2	Yes

We can see that the overall decision rule can be reduced to the rule "Do what the majority say". The associated probability of error is:

$$\Pr(\text{err}) = \frac{1}{2}\left[(0.09 + 0.045 + 0.07 + 0.03) + (0.03 + 0.07 + 0.045 + 0.09)\right] = 0.235$$

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.

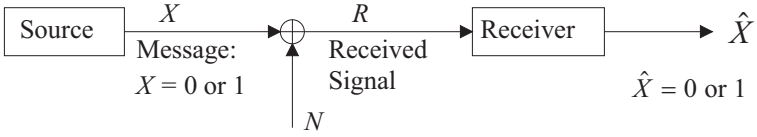


Figure 1

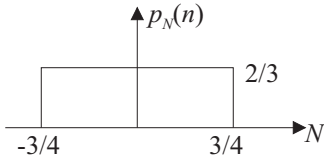


Figure 2

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

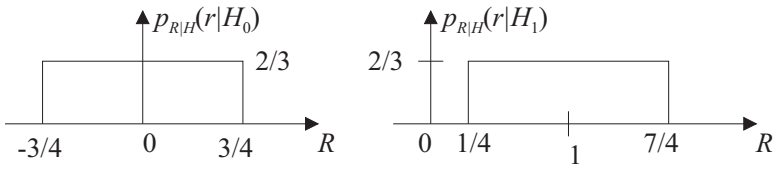
(b) Suppose that the receiver does not know the true a priori probabilities, so it decides to use a maximum likelihood (ML) detector (i.e. the detector will *assume* something about the prior probabilities, which may in fact be different from these assumptions). Find the ML detector and the associated 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.

Solution:

(a) For this problem we have:

$$\begin{aligned} H_0 :\ R = N &\implies P_{R|H}(r|H_0) = p_N(r) \\ H_1 :\ R = N + 1 &\implies P_{R|H}(r|H_1) = p_N(r-1) \end{aligned}$$

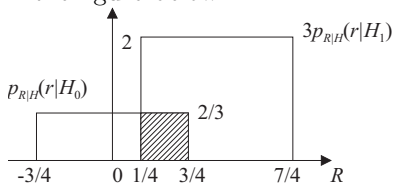
Graphically this gives us:



The minimum probability of error decision rule is given by the zero-one cost structure $C_{ij} = 1 - \delta_{ij}$:

$$\mathcal{L}(R) = \frac{P(r|H_1)}{P(r|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0}{P_1} = \frac{1}{3} \quad \Longrightarrow \quad 3P(r|H_1) \underset{H_0}{\overset{H_1}{\gtrless}} P(r|H_0)$$

These two densities are shown in the figure below.



By inspection of the conditional densities we sketched we can write down the MPE decision rule:

$$\text{MPE Decision Rule:} \quad r \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1}{4}$$

The probability of error is given by:

$$\Pr(\varepsilon) = P_0 P_F + P_1 P_M = \frac{1}{4} P_F + \frac{3}{4} P_M = \frac{1}{4} \int_{1/4}^{\infty} P(r|H_0)\, dr + \frac{3}{4} \int_{-\infty}^{1/4} P(r|H_1)\, dr = \frac{1}{4}\left(\frac{1}{3}\right) + \frac{3}{4}(0) = \frac{1}{12}$$

Note that $P_M = 0$ for this case, and P_F is the shaded area in the figure.

(b) The maximum likelihood rule is given by:

$$P(r|H_1) \underset{H_0}{\overset{H_1}{\gtrless}} P(r|H_0)$$

Note that:

$$\begin{array}{lcll} P(r|H_1) & < & P(r|H_0) & \text{for } r < \frac{1}{4} \\ P(r|H_1) & = & P(r|H_0) & \text{for } \frac{1}{4} \le r < \frac{3}{4} \\ P(r|H_1) & > & P(r|H_0) & \text{for } \frac{3}{4} \le r \end{array}$$

Which is the overlapping region in these two superimposed probability density functions.
Thus any decision rule of the form:

$$r \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \qquad \alpha \in [1/4, 3/4)$$

constitutes a valid ML rule for this problem. Clearly, since α is not unique, neither is the ML rule.

Now let us find the probability of error when we apply the ML decision rule and the prior probabilities are, in fact, not the same.

$$\begin{aligned}\Pr(\varepsilon) &= P_0P_F + P_1P_M = P_0\int_{\alpha}^{\infty} P(r|H_0)\,dr + P_1\int_{-\infty}^{\alpha} P(r|H_1)\,dr = P_0\int_{\alpha}^{3/4}\frac{2}{3}\,dr + P_1\int_{1/4}^{\alpha}\frac{2}{3}\,dr \\ &= \frac{2}{3}\left[P_0\left(\frac{3}{4}-\alpha\right) + P_1\left(\alpha-\frac{1}{4}\right)\right] = \frac{2}{3}\left[\frac{3P_0-P_1}{4} + \alpha(P_1-P_0)\right]\end{aligned}$$

Note that under the ML assumption, the decision rule is not unique, but the actual probability of error based on the true prior probabilities and different values of $\alpha \in [1/4, 3/4)$ will have different $\Pr(\varepsilon)$. In particular, note that $\Pr(\varepsilon)$ is a function of α unless the ML assumption is correct, i.e. unless $P_0 = P_1 = 1/2$. For other P_0, P_1 combinations $\Pr(\varepsilon)$ will be different. For example if $P_1 = 3/4$ and $P_0 = 1/4$ as in (a), then $\Pr(\varepsilon) = \alpha/3$ and, not surprisingly, the error is minimized for $\alpha = 1/4$ as in part (a).

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.

Solution: This problem calls for a Neyman-Pearson test. Recall that an NP test is also an LRT, so the only question is what threshold to pick to meet the P_F specification. The LRT is given by:

$$\frac{P(x \mid H_1)}{P(x \mid H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \implies \frac{xe^{-x}}{e^{-x}} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \implies x \underset{H_0}{\overset{H_1}{\gtrless}} \eta \geq 0$$

Note that this expression is only valid for $x \geq 0$, but our observed value of x will always satisfy this, so we don't need to worry about the $u(t)$ terms. Now given this test we can calculate P_F for any value of η:

$$P_F = \Pr(\text{Choose } H_1 \mid H_0 \text{ True}) = \Pr(x > \eta \mid H_0) = \int_{\eta}^{\infty} e^{-x}u(x)\,dx = e^{-\eta} \qquad \eta \geq 0$$

The specified performance requirement is that $P_F = .1$, thus we choose η so that $e^{-\eta} = .1$ or $\eta = -\ln(0.1) \approx 2.3$. Thus the NP test is given by:

$$x \underset{H_0}{\overset{H_1}{\gtrless}} 2.3$$

The corresponding value of P_D is given by:

$$P_D = \Pr(\text{Choose } H_1 \mid H_1 \text{ True}) = \Pr(x > \eta \mid H_1) = \int_{\eta}^{\infty} xe^{-x}u(x)\,dx = -xe^{-x}|_{\eta}^{\infty} - e^{-x}|_{\eta}^{\infty} = \eta e^{-\eta} + e^{-\eta} \approx 0.33$$

Since the test producing this P_D is an LRT we know that this is the highest P_D for the given P_F.

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).

Solution:

(a) Note that given the statements in the problem we have:

$$\begin{aligned} P(y \mid H_0) &= N(y; 0, N_0) \\ P(y \mid H_1) &= N(y; A, N_0) \end{aligned}$$

and $P_0 = 0.999$, $P_1 = 0.001$. Now the MAP rule is given by:

$$\begin{aligned} P(H_1 \mid y) &\underset{H_0}{\overset{H_1}{\gtrless}} P(H_0 \mid y) \\ \Longrightarrow \quad P(y \mid H_1)P_1 &\underset{H_0}{\overset{H_1}{\gtrless}} P(y \mid H_0)P_0 \\ \Longrightarrow \quad y &\underset{H_0}{\overset{H_1}{\gtrless}} \frac{2N_0 \ln\left(\frac{P_0}{P_1}\right) + A^2}{2A} = \Gamma_{\text{MAP}} \end{aligned}$$

where we have used the results for Gaussian problems given in class to simplify the last expression. Now the corresponding probability of error is given by:

$$\begin{aligned} \Pr(\varepsilon) &= P_F P_0 + P_M P_1 = \Pr(y > \Gamma_{\text{MAP}} \mid H_0)P_0 + \Pr(y < \Gamma_{\text{MAP}} \mid H_1)P_1 \\ &= P_0 \int_{\Gamma_{\text{MAP}}}^{\infty} P(y \mid H_0)\, dy + P_1 \int_{-\infty}^{\Gamma_{\text{MAP}}} P(y \mid H_1)\, dy \\ &= P_0 \int_{\Gamma_{\text{MAP}}}^{\infty} N(y; 0, N_0)\, dy + P_1 \int_{-\infty}^{\Gamma_{\text{MAP}}} N(y; A, N_0)\, dy \\ &= P_0 Q\left(\frac{\Gamma_{\text{MAP}}}{\sqrt{N_0}}\right) + P_1 Q\left(-\frac{\Gamma_{\text{MAP}} - A}{\sqrt{N_0}}\right) \end{aligned}$$

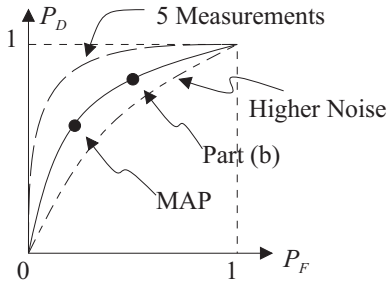
(b) In this case we have for the decision rule:

$$y \underset{H_0}{\overset{H_1}{\gtrless}} \frac{2N_0 \ln\left(\frac{P_0}{10P_1}\right) + A^2}{2A}$$

Note that this change to the cost structure serves to increase P_D. This makes sense, since we have raised the cost for missing the target, we would expect that the detection probability should rise.

(c) The ROC for the various cases is illustrated below. Note the following:

- The rule in part (b) will have a higher P_D than the rule in part (a), but they will be on the same ROC, since the rules differ only in their choice of threshold.
- If we have more measurements, P_D must be higher at any P_F than for the ROC obtained in parts (a),(b) – i.e. the ROC for this case will be above the ROC obtained for these parts.
- If the noise is increased, P_D must be lower at any P_F than the ROC obtained in parts (a),(b) – i.e. the ROC for this case will be below the ROC obtained for these parts.



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`.

Solution:

(a) The `MATLAB` code for the function `psdest2.m` is given below :

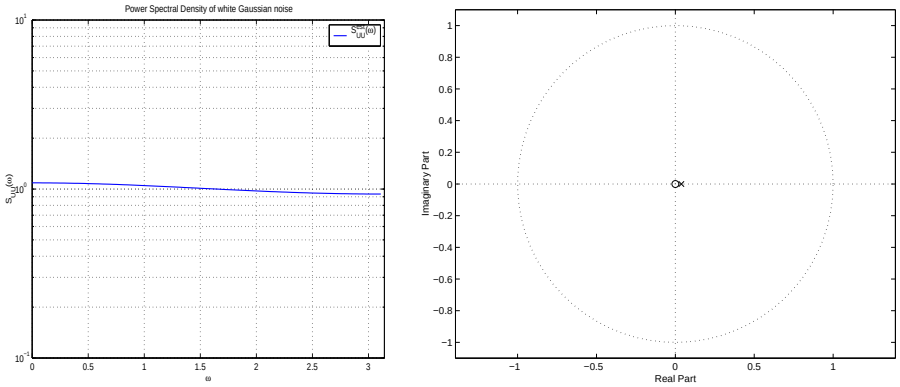
```
function [Syy,w,a,b] = psdest2(y,Np,Nf)

% [Syy,w,a,b] = psdest2(y,Np,Nf)
%
% y      : Vector containing sample path data.
% Np     : Number of poles in the AR model
% Nf     : Number of frequency samples to generate
%
% Syy    : Vector of estimeted power spectral density
% w      : Corresponding vector of frequency values
% a      : Vector of denominator coefficients
% b      : Vector of numerator coefficients.
%
% Estimates the power spectral density of a process given sample path data.
% The function first estimates the autocorrelation function of the process
% and then uses these values to estimate an AR model
```

```
% Estimating autocorrelation function values for lags between -Np and Np
[Ryy,lag] = xcorr(y,Np,'unbiased');

% Keeping the positive lag values in a column vector
Ryy = Ryy(Np+1:2*Np+1);
Ryy = Ryy(:);

% Forming and solving the Yule-Walker equations
a = toeplitz(Ryy(1:Np)) \ (-Ryy(2:Np+1));
a = [1 a'];

% Finding the DC gain term
b2 = a*Ryy;
b = [sqrt(b2) 0];

% Using estimated AR model to find the associated PSD
[Syy,w] = syspsd(b,a,Nf);
```

Fig 1(a) shows the power spectral density of white Gaussian noise calculated using the above code with `Np=1` and `Nf=100`. The estimated location of the poles is shown in Fig 1(b). It is seen that the only pole of the system is located close to `z=0` and the system model is :

$$Y(n) = U(n) \tag{5}$$

which is what one would expect.



(a) (b)

Figure 1: (a) Power Spectral Density for white (unit variance) Gaussian noise, (b) Pole-Zero plot for white-noise spectral model

(b) The estimates of the power spectral densities of the output of the five systems, calculated using the AR modeling approach (with `Np=2` and `Nf=100`) are shown in Figure 2. The plots also show the PSD of the outputs calculated using the FFT-based method described in Problem Set No.5 along with the theoretical PSDs. We note that the power spectral density estimate produced by the AR-modeling method is smoother than the one produced by the FFT-based method of the previous problem set.

(c) The power spectral density of the output of System #0 estimated using AR models of order `Np = 2,3,5,10` are shown in Figure 3 while the pole-zero plots of the model systems are shown in Figure 4.

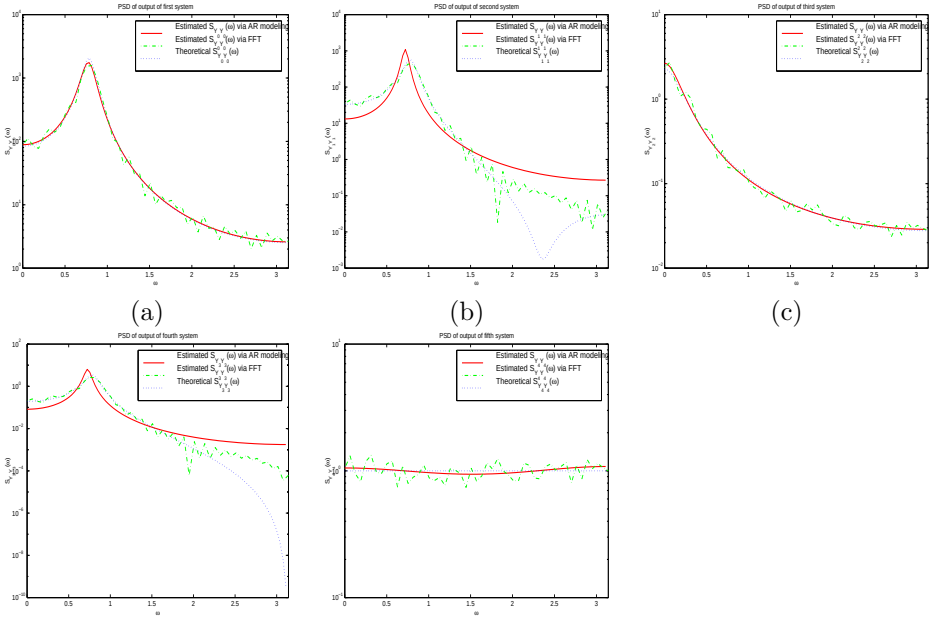


Figure 2: Estimated and theoretical power spectral densities of the output of the systems driven by white noise (a) System #0, (b) System #1, (c) System #2, (d) System #3, (e) System #4

It is seen that the estimates of the PSD are not sensitive to the use of a model order which is higher than that of the true system.

If we instead underestimate the model order of System #0 to be `Np=1` the estimated PSD of the output and the pole-zero plot of the corresponding model are shown in Figure 5. In this case the estimated PSD differs vastly from the one estimated using the true (or overestimated) model order. Therefore the method is sensitive to underestimation of the AR model order. The moral is to use a higher model order than you think you need.

(d) The power spectral density of the output of System #1 estimated using AR models of order `Np = 1,2,3,6,9,12` are shown in Figure 6 while the pole-zero plots of the model systems are shown in Figure 7. In this case System #1 is an second order ARMA system and we are modeling it by all-pole (AR) models of different orders. As seen in Figure 6 the estimate of the PSD is not close to the true theoretical PSD for model orders smaller than and even equal to the true system order. However as we increase the model order the estimated PSD begins to resemble the true PSD. Therefore the AR-based technique is reasonable for estimating the power spectra of AR and ARMA processes as long as we are careful to choose the model orders high enough. What is happening is that, while the AR form is not the true form of the model, as we add more poles to it we are adding more degrees of freedom, which are then used to try and approximate the corresponding frequency response.

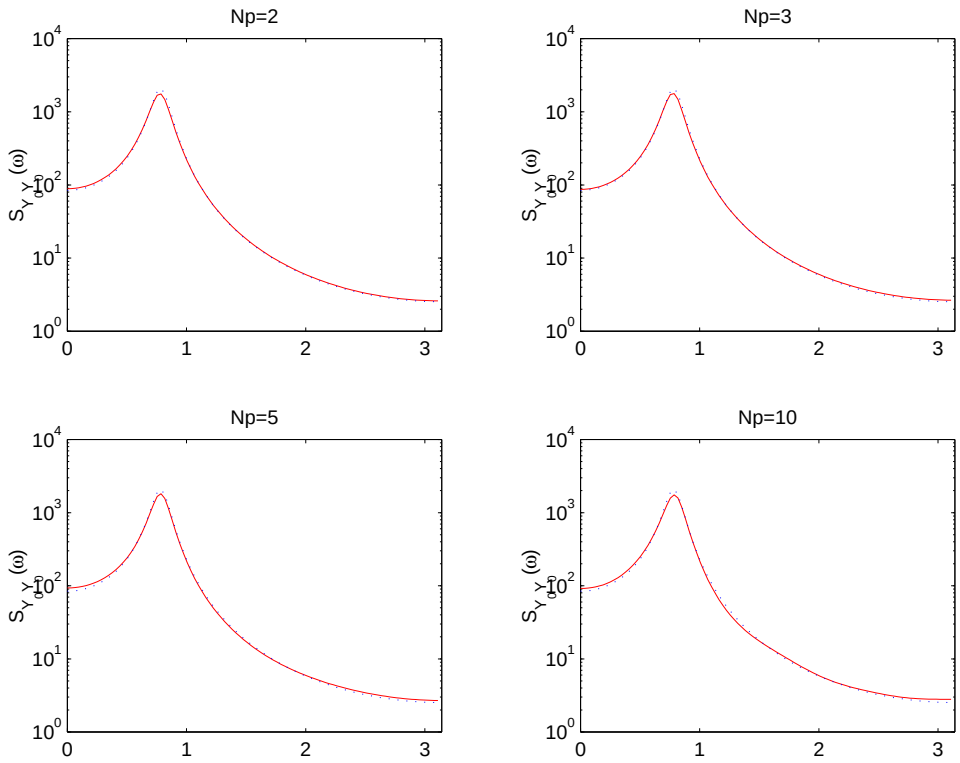


Figure 3: Estimated (*solid*) and true (*dashed*) power spectral density for output of System #0

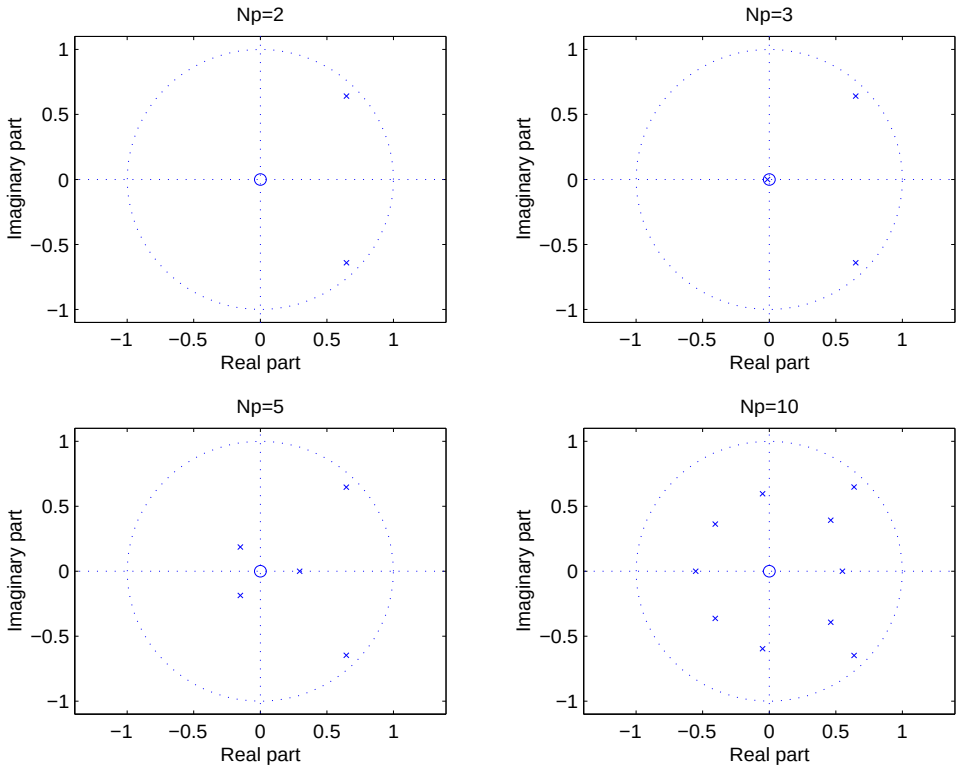
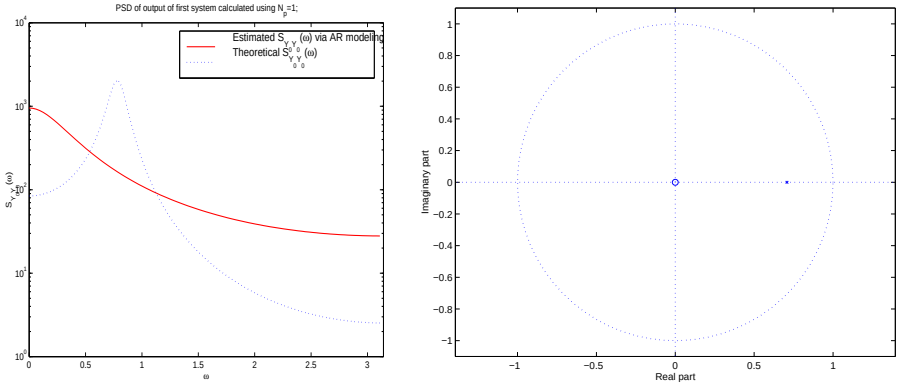


Figure 4: Pole-zero plots for spectral models of output of System #0 and different model orders



(a) (b)

Figure 5: (a) Estimated Power Spectral Density for output of System #0 calculated using `Np=1` , (b) Pole-Zero plot for the corresponding system model

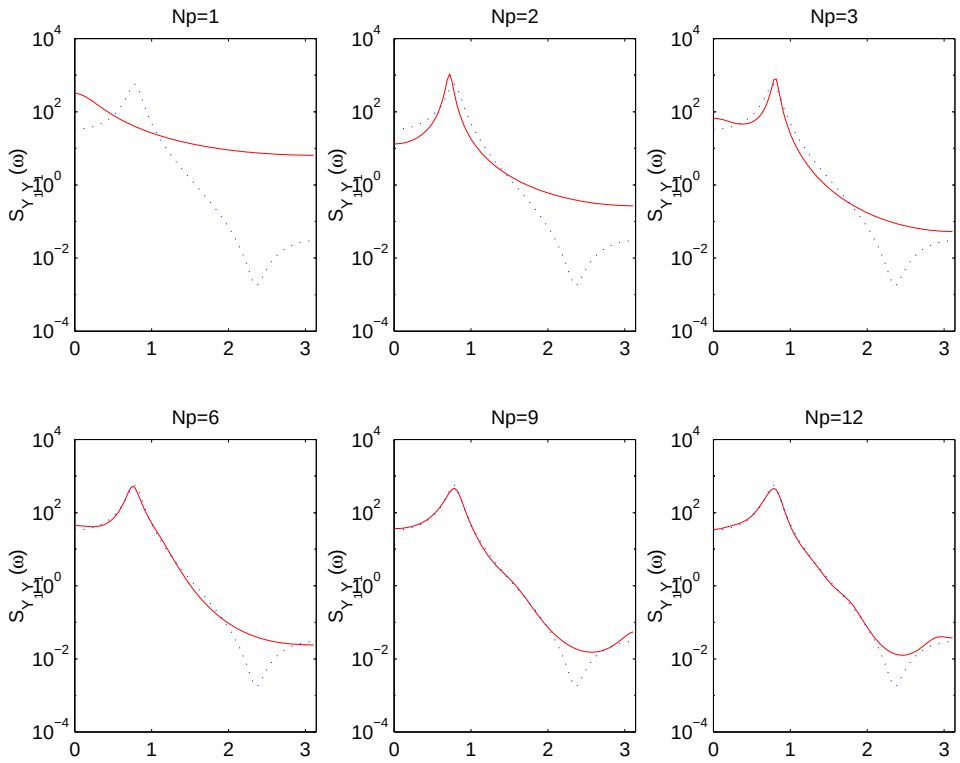


Figure 6: Estimated (*solid*) and true (*dashed*) power spectral density for output of System #1

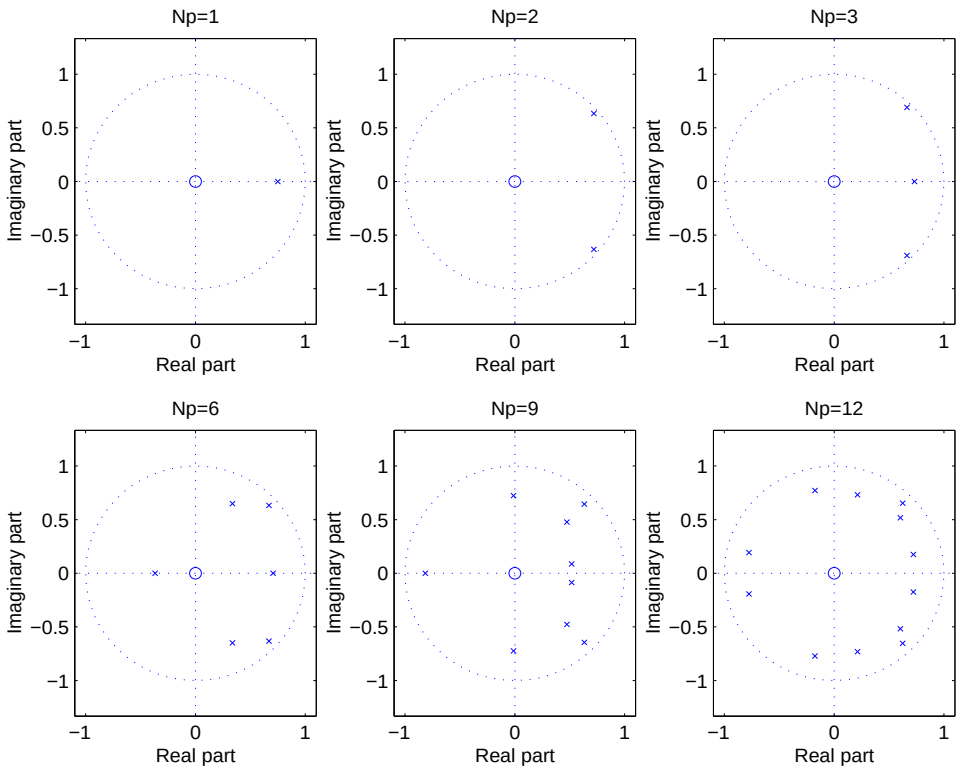


Figure 7: Pole-zero plots for spectral models of output of System #1 and different model orders