Boston University

Department of Electrical and Computer Engineering

EC505 STOCHASTIC PROCESSES

Problem Set No. 5 Solutions

Fall 2016

Problem 5.1 (Shanmugan and Breipohl 4.6)

Suppose the process $X(n)$ is a stationary, zero mean, discrete-time white Gaussian noise with $R_{XX}(k) = \delta(k)$. The process $Y(n)$ is obtained as the output of a discrete-time linear time invariant moving average system driven by $X(n)$:

$$Y(n) = h(0)X(n) + h(1)X(n-1) + \cdots + h(k)X(n-k).$$

(a) Find $p_{Y(n)}(y)$, the first-order density of the output.

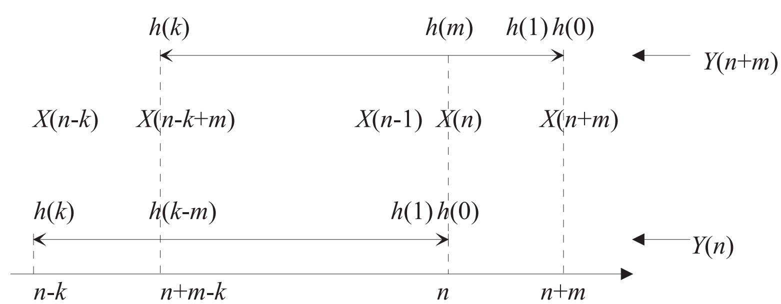
(b) Find $R_{YY}(k)$.

(c) Is the output wide-sense and/or strict-sense stationary?

Solution:

(a) This is a moving-average process with a Gaussian white-noise input. Thus the output is a linear combination of the input points, which are a sequence of uncorrelated (and therefore independent) Gaussian random variables. So the output process $Y(n)$ is also a Gaussian random variable, so all we need to characterize the pdf $p_{Y(n)}(y)$ of $Y(n)$ is the mean and variance. Now it's easy to see that $E[Y(n)] = 0$ from its definition, since the mean of $X(n)$ is zero. The variance is $R_{YY}(0) = \sum_{i=0}^{k} h^2(i)$, which we compute in part (b).

(b) In the figure below we show a "timing diagram" for the problem, showing the relationship between the values of $h(i)$ and $X(n)$ involved in the definition of $Y(n)$ and $Y(n+m)$. We can see that $R_{YY}(m)$ will be zero when the two bars do not overlap and only the overlap part contributes to the solution.



Direct calculation establishes:

$$\begin{aligned}
R_{YY}(m) &= E[Y(n)Y(n+m)] = \\
&= E\left[\left(\sum_{i=0}^{k} h(i)X(n-i)\right)\left(\sum_{j=0}^{k} h(j)X(n+m-j)\right)\right] \\
&= \sum_{i=0}^{k}\sum_{j=0}^{k} h(i)h(j)E\left[X(n-i)X(n+m-j)\right] \\
&= \begin{cases} \sum_{j=m}^{k} h(j-m)h(j) & \text{for } |m| \le k \\ 0 & \text{otherwise} \end{cases} = \begin{cases} \sum_{i=0}^{k-m} h(i)h(m+i) & \text{for } |m| \le k \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

The simple way to get this result is to use our results for processes through linear systems:

$$\begin{aligned} R_{YY}(m) &= R_{XX}(m) * h(m) * h(-m) = h(m) * h(-m) \\ &= \sum_{i=-\infty}^{\infty} h(m-i)h(-i) = \sum_{i=-\infty}^{\infty} h(m+i)h(i) = \begin{cases} \sum_{i=0}^{k-m} h(i)h(m+i) & \text{for } |m| \le k \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The power spectral density can be found by taking the transform of the $R_{YY}(m)$ we just found.

$$\begin{aligned} S_{YY}(f) &= \mathcal{F}\{R_{YY}(m)\} = \mathcal{F}\left\{\sum_{i=0}^{k-m} h(i)h(m+i)\right\} \qquad \text{for } m \le k \\ &= \sum_{m=-k}^{k}\left[\sum_{i=0}^{k-m} h(i)h(m+i)\right] e^{-j2\pi mf} \qquad |f| < 1/2 \end{aligned}$$

Of course, we can also get this result by observing that $S_{XX}(f) = 1$ since the input is white, and the transfer function of the system is $H(f) = \sum_{i=0}^{k} h(i)e^{-j2\pi fi}$. Thus, $S_{YY}(f) = |H(f)|^2$.

(c) Yes the output is stationary. First each output point is a linear combination of jointly Gaussian random variables, thus the output points are also jointly Gaussian, so it is a Gaussian random process. Second, the output mean is constant and autocorrelation only a function of time difference. So it is WSS, and since Gaussian, also SSS.

Problem 5.2 (Shanmugan and Breipohl 4.7)

Consider the following difference equation with initial condition $X(0) = 1$ and $U(n)$, $n \ge 0$ a sequence of zero mean, uncorrelated Gaussian random variables with variance σ_U^2.

$$X(n+1) = \sqrt{n}X(n) + U(n), \quad n = 0, 1, 2, \ldots$$

(a) Find $m_X(n)$.

(b) Find $R_{XX}(0,k)$, $R_{XX}(1,1)$, $R_{XX}(1,2)$, and $R_{XX}(3,1)$.

(c) Is the output process $X(n)$ wide-sense and/or strict-sense stationary?

Solution:

(a) First note that $X(0) = 1$ so that $m_X(0) = 1$. Now taking expectations of the difference equation we find that:

$$\begin{aligned} & E[X(n+1)] = E\left[\sqrt{n}X(n) + U(n)\right] \\ \Longrightarrow \quad & m_X(n+1) = \sqrt{n}\, m_X(n) \\ \Longrightarrow \quad & m_X(0) = 1 \\ & m_X(1) = \sqrt{0}\, m_X(0) = 0 \cdot 1 = 0 \\ & m_X(2) = \sqrt{1}\, m_X(1) = 1 \cdot 0 = 0 \\ & \vdots \end{aligned}$$

Therefore,

$$m_X(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{else} \end{cases}$$

(b) First note that:

$$\begin{aligned}
X(0) &= 1 \\
X(1) &= U(0) \\
X(2) &= U(0) + U(1) \\
X(3) &= \sqrt{2}\,(U(0) + U(1)) + U(2)
\end{aligned}$$

Thus,

$$\begin{aligned}
R_{XX}(0,n) &= E\left[X(0)X(n)\right] = E\left[1 \cdot X(n)\right] = E[X(n)] = \begin{cases} 1 & n = 0 \\ 0 & \text{else} \end{cases} \\
R_{XX}(1,1) &= E\left[X(1)X(1)\right] = E\left[U(0)U(0)\right] = \sigma^2_{U(0)} \\
R_{XX}(1,2) &= E\left[X(1)X(2)\right] = E\left[U(0)\left(\sqrt{1}\,X(1) + U(1)\right)\right] = E\left[U(0)\left(U(0) + U(1)\right)\right] = \sigma^2_{U(0)} \\
R_{XX}(3,1) &= E\left[X(3)X(1)\right] = E\left[\left(\sqrt{2}\,X(2) + U(2)\right)U(0)\right] = E\left[\left(\sqrt{2}\,U(0) + \sqrt{2}\,U(1) + U(2)\right)U(0)\right] = \sqrt{2}\sigma^2_{U(0)}
\end{aligned}$$

(c) The process is not WSS and thus not SSS. This can be traced to two “problems” – there is a nonzero initial condition and also the system is not shift-invariant.

Problem 5.3 (Shanmugan and Breipohl 4.14)

Assume that the input to a linear time-invariant system with impulse response $h(t) = e^{-t}u(t)$ is a zero-mean Gaussian random process with power spectral density $S_{XX}(f) = \eta/2$.

(a) Find the power spectral density $S_{YY}(f)$ of the output $Y(t)$.

(b) Find the average power in the output $E[Y^2(t)]$.

Solution:

(a) The transfer function is given by:

$$H(f) = \frac{1}{1 + j2\pi f}$$

The input is white noise, so the output spectrum is given by:

$$S_{YY}(f) = |H(f)|^2 S_{XX}(f) = \frac{\eta}{2}|H(f)|^2 = \frac{\eta/2}{1 + (2\pi f)^2}$$

(b) If we do this in the frequency domain we get:

$$\begin{aligned}
E\left[Y^2(t)\right] &= R_{YY}(0) = \int_{-\infty}^{\infty} S_{YY}(f)\,df = \int_{-\infty}^{\infty} \frac{\eta/2}{1 + (2\pi f)^2}\,df = \frac{\eta}{2}\left[\frac{1}{2\pi}\tan^{-1}(2\pi f)\right]_{f=-\infty}^{\infty} = \frac{\eta}{4\pi}\left[\frac{\pi}{2} + \frac{\pi}{2}\right] \\
&= \frac{\eta}{4}
\end{aligned}$$

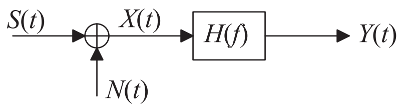
It is actually easier in the time domain. Taking the inverse transform of $S_{YY}(f)$ we find for $R_{YY}(\tau)$:

$$R_{YY}(\tau) = \frac{\eta}{4}e^{-|\tau|}$$

So $R_{YY}(0) = \frac{\eta}{4}$ again.

Problem 5.4 (Shanmugan and Breipohl 4.20)

In this problem we study filter design for a stochastic signal in noise. Consider the system shown in the figure below, where the input is given by $S(t) = a\sin(2\pi f_c t + \Theta)$ with a, f_c real constants and Θ uniformly distributed in the interval $[-\pi, \pi]$. The process $N(t)$ is white Gaussian noise with $S_{NN}(f) = \eta/2$ which is independent of Θ.



$$H(f) = \frac{1}{1 + j\left(\frac{f}{f_0}\right)}$$

(a) Let $Y(t)$ be the total output due to $X(t)$, $Y_1(t)$ be the output due to the signal $S(t)$ alone, and $Y_2(t)$ be the output due to the noise $N(t)$ alone. Since the system is linear $Y(t) = Y_1(t) + Y_2(t)$. Find $S_{Y_1Y_1}(f)$, $S_{Y_2Y_2}(f)$, and $S_{YY}(f)$.

(b) Find the ratio of total average power in the signal component of the output $Y_1(t)$ to the total average power in the noise component of the output $Y_2(t)$. This is the output signal-to-noise ratio (SNR).

(c) What value of the filter bandwidth f_0 will maximize the output SNR you found in part (b).

Solution:

(a) First note that:

$$\begin{aligned}
E[S(t)S(s)] &= E\left[A^2\sin(2\pi f_c t + \Theta)\sin(2\pi f_c s + \Theta)\right] && (1)\\
&= E\left[\frac{A^2}{2}\left(\cos(2\pi f_c(s-t)) - \cos(2\pi f_c(s+t) + 2\Theta)\right)\right] && (2)\\
&= \frac{A^2}{2}\cos(2\pi f_c(s-t)) && (3)
\end{aligned}$$

Thus we find:

$$\begin{aligned}
S_{SS}(f) &= \frac{A^2}{4}[\delta(f-f_c) + \delta(f+f_c)]\\
S_{NN}(f) &= \eta/2
\end{aligned}$$

Thus,

$$\begin{aligned}
S_{Y_1Y_1}(f) &= |H(f)|^2 S_{SS}(f) = \frac{1}{1+(f_c/f_0)^2}\frac{A^2}{4}[\delta(f-f_c)+\delta(f+f_c)]\\
S_{Y_2Y_2}(f) &= |H(f)|^2 S_{NN}(f) = \frac{\eta/2}{1+(f/f_0)^2}\\
S_{YY}(f) &= |H(f)|^2[S_{SS}(f) + S_{NN}(f)] = \frac{1}{1+(f/f_0)^2}\left[\frac{A^2}{4}\{\delta(f-f_c)+\delta(f+f_c)\} + \frac{\eta}{2}\right]
\end{aligned}$$

In the answer for $S_{Y_1Y_1}(f)$ we have used the fact that the impulses will sift out the value of the system function at $f = f_c$.

(b) The average power in $Y_1(t)$ is given by $E[Y_1(t)^2] = R_{Y_1Y_1}(0)$ while the average power in $Y_2(t)$ is given by $E[Y_2(t)^2] = R_{Y_2Y_2}(0)$. Now:

$$\begin{aligned}
R_{Y_1Y_1}(\tau) &= 2\frac{A^2}{4}\left[\frac{1}{1+(f_c/f_0)^2}\right]\cos(2\pi f_c\tau) = \frac{A^2}{2}\left[\frac{1}{1+(f_c/f_0)^2}\right]\cos(2\pi f_c\tau)\\
R_{Y_2Y_2}(\tau) &= \frac{\eta\pi f_0}{2}e^{-2\pi f_0|\tau|}
\end{aligned}$$

Thus

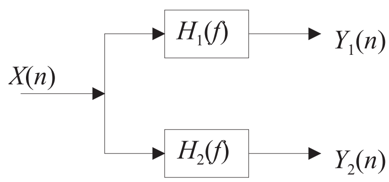
$$\text{SNR} = \frac{R_{Y_1Y_1}(0)}{R_{Y_2Y_2}(0)} = \frac{2}{\eta\pi f_0}\frac{A^2}{2}\left[\frac{1}{1+(f_c/f_0)^2}\right] = \frac{A^2}{\eta\pi}\left[\frac{f_0}{f_0^2+f_c^2}\right]$$

(c) The SNR is maximized if $f_0 = f_c$. This is intuitively obvious, since to filter as much noise out as possible, you must set the break frequency in the low-pass filter to as low a value as possible, without throwing out any of the signal. Of course, this is the answer you get by differentiating the ratio and setting the derivative equal to zero.

Problem 5.5

Let $X[n]$ be a real-valued, discrete-time, zero-mean wide-sense stationary random process with correlation function $R_{XX}(m)$ and spectrum $S_{XX}(f)$.

(a) Suppose $X(n)$ is the input to two real-valued linear time-invariant systems as depicted below, producing two new processes, $Y_1(n)$ and $Y_2(n)$. Find $R_{Y_1Y_2}(n)$ and $S_{Y_1Y_2}(f)$ in terms of $h_1(n)$, $h_2(n)$, $R_{XX}(n)$, $H_1(f)$, $H_2(f)$, and $S_{XX}(f)$.



(b) Suppose next that $H_1(f)$ and $H_2(f)$ are non-overlapping frequency responses, i.e.,

$$|H_1(f)| \cdot |H_2(f)| = 0 \ , \qquad \text{all } f \ .$$

Show that in this case $Y_1(n)$ and $Y_2(m)$ are uncorrelated for all n and m. Are $Y_1(n)$ and $Y_2(m)$ statistically independent (for all n and m) in the case that $x(n)$ is a Gaussian random process? Explain.

Solution:

(a) First we find $S_{Y_1Y_2}(f)$ using the properties of random processes through linear systems:

$$S_{Y_1Y_2}(f) = H_1(-f)S_{XY_2}(f) = H_1(-f)H_2(f)S_{XX}(f)$$

Now we can similarly find $R_{Y_1Y_2}(n)$:

$$R_{Y_1Y_2}(n) = h_1(-n) * R_{XY_2}(n) = h_1(-n) * h_2(n) * R_{XX}(n)$$

(b) Note from part (a) we have that:

$$|S_{Y_1Y_2}(f)| = |H_1(-f)| \cdot |H_2(f)| \cdot |S_{XX}(f)| = |H_1^*(f)| \cdot |H_2(f)| \cdot |S_{XX}(f)| = |H_1(f)| \cdot |H_2(f)| \cdot |S_{XX}(f)|$$

Where the second to the last equality follows from the fact that $h_1(n)$ is real. Thus, since $|H_1(f)| \cdot |H_2(f)| = 0$ for all f, we have that $S_{Y_1Y_2}(f) = 0$ for all f. This implies that the inverse Fourier transform of $S_{Y_1Y_2}(f)$ is zero and thus that $R_{Y_1Y_2}(n)$ is zero for all n. Since $R_{Y_1Y_2}(n) = K_{Y_1Y_2}(n)$ we see that $Y_1(n)$ and $Y_2(n)$ are uncorrelated for all n.

Now if $X(n)$ is a Gaussian random process, then $Y_1(n)$ and $Y_2(n)$ are jointly Gaussian random variables since they are obtained by a linear transformation (i.e. the convolution with the impulse responses) of a set of jointly Gaussian random variables (i.e. the set of inputs $X(n)$). Since $Y_1(n)$ and $Y_2(n)$ are jointly Gaussian, if they are uncorrelated they are also independent.

Problem 5.6

Assume that a shaping filter $H(f)$, when driven by white noise $W(t)$ with autocorrelation $R_{WW}(\tau) = \delta(\tau)$, produces a wide-sense stationary output with power spectral density

$$S_{YY}(f) = \frac{1 + (2\pi f)^2}{(2\pi f)^4 + 12(2\pi f)^2 + 36}.$$

(a) Find the variance of the output $E[Y^2(t)]$.

(b) Find the transfer function of the shaping filter $H(f)$.

Solution:

(a) We know that $E[Y^2(t)] = R_{YY}(0)$, so let's find $R_{YY}(\tau)$.

$$R_{YY}(\tau) = \int_{-\infty}^{\infty} \frac{(2\pi f)^2 + 1}{(2\pi f)^4 + 12(2\pi f)^2 + 36} e^{j2\pi f\tau}\, df$$

To do the integral, expand the integrand in a partial fraction expansion, with the substitution $r = (2\pi f)^2$, to get

$$\frac{r+1}{r^2 + 12r + 36} = \frac{1}{r+6} + \frac{-5}{(r+6)^2} = \frac{1}{(2\pi f)^2 + 6} + \frac{-5}{((2\pi f)^2 + 6)^2}$$

The inverse transform of the first term is given by:

$$\frac{1}{(2\pi f)^2 + 6} \iff \frac{1}{2\sqrt{6}} e^{-\sqrt{6}|\tau|}$$

The inverse transform of the second term is given by

$$\begin{aligned} \frac{-5}{((2\pi f)^2 + 6)^2} &= \frac{-5}{12}\left[\frac{(6 - (2\pi f)^2) + (6 + (2\pi f)^2)}{((2\pi f)^2 + 6)^2}\right] = \frac{-5}{12}\left[\frac{(6 - (2\pi f)^2)}{((2\pi f)^2 + 6)^2} + \frac{1}{(2\pi f)^2 + 6}\right] \\ &\iff \frac{-5}{24} e^{-\sqrt{6}|\tau|}\left(|\tau| + \frac{1}{\sqrt{6}}\right) = \frac{-5}{24\sqrt{6}} e^{-\sqrt{6}|\tau|}(1 + \sqrt{6}|\tau|) \end{aligned}$$

Combining yields:

$$R_{YY}(0) = \frac{1}{2\sqrt{6}} + \frac{-5}{24\sqrt{6}} = \frac{7}{24\sqrt{6}}$$

(b) The input power spectral density is $S_{WW}(f) = 1$. The output power spectral density is $S_{YY}(f) = |H(f)|^2 = H(f)H^*(f) = H(f)H(-f)$ for a real impulse response. We also know that the filter must be stable (or the resulting output will not be stationary), so that all of the poles must be in the left-half complex plane. It's a matter of taste, but I think its easier to convert to the Laplace transform first, to obtain:

$$S_{YY}(s) = \frac{1 - s^2}{s^4 - 12s^2 + 36} = \frac{1 - s^2}{(s^2 - 6)^2} = \frac{(1+s)(1-s)}{(s+\sqrt{6})^2(s-\sqrt{6})^2} = \frac{(1+s)}{(s+\sqrt{6})^2}\frac{(1-s)}{(s-\sqrt{6})^2}$$

Equivalently, using the Fourier transform we get:

$$S_{YY}(f) = \frac{(1 + j2\pi f)}{(j2\pi f + \sqrt{6})^2}\frac{(1 - j2\pi f)}{(j2\pi f - \sqrt{6})^2}$$

Now we need to factor these into the stable and unstable parts. I've already done this above. So now we choose the stable factor for $H(f)$:

$$H(j\omega) = \frac{1 + j\omega}{(j\omega + \sqrt{6})^2}$$

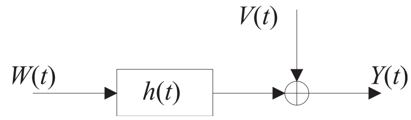
Note that the choice of numerator is a bit arbitrary, since we could also have chosen:

$$H(j\omega) = \frac{1 - j\omega}{(j\omega + \sqrt{6})^2}$$

But this would have yielded a system with an unstable inverse, so usually we choose both the poles and the zeros to be “stable” (i.e. LHP or inside the unit circle).

Problem 5.7 (Old Exam Question)

Let $W(t)$ and $V(t)$ be independent stationary zero-mean white Gaussian random processes with correlation functions $R_{WW}(\tau) = \sigma_W^2 \delta(\tau)$ and $R_{VV}(\tau) = \sigma_V^2 \delta(\tau)$. A new signal $Y(t)$ is generated from $W(t)$ and $V(t)$ via the processing shown in the figure, with $h(t)$ stable and causal and $H(f) = \dfrac{1}{\alpha + j2\pi f}$:



(a) Find $R_{YY}(\tau)$ and $S_{YY}(f)$ (you need not simplify $S_{YY}(f)$).

(b) We want to find a simpler system whose output random process has the same statistical properties as $Y(t)$. Let $E(t)$ be a stationary zero-mean white Gaussian random process with covariance function $R_{EE}(\tau) = \delta(\tau)$. Find the impulse response $h'(t)$ of a linear time invariant system such that $Z(t) = h'(t) * E(t)$ has the same covariance function as $Y(t)$. Use $\alpha = 2$, $\sigma_W^2 = 5$, and $\sigma_V^2 = 1$. Hint: Consider factoring $S_{YY}(f)$.

(c) Is the answer to part (b) unique? Explain.

(d) Do the random processes $Y(t)$ and $Z(t)$ have the same pdfs, i.e. for all $t_0, \dots, t_n$ and all $a_0, \dots, a_n$ does: $p_{Y(t_0),Y(t_1),\dots,Y(t_n)}(a_0, a_1, \dots a_n) = p_{Z(t_0),Z(t_1),\dots,Z(t_n)}(a_0, a_1, \dots a_n)$?

Solution:

(a) Note that $W(t)$ and $V(t)$ are independent, thus cross terms cancel and:

$$\begin{aligned} S_{YY}(f) &= |H(f)|^2 S_{WW} + S_{VV}(f) = \frac{1}{\alpha^2 + (2\pi f)^2}\sigma_W^2 + \sigma_V^2 = \frac{\sigma_W^2}{\alpha}\frac{1/\alpha}{1 + \left(\frac{2\pi f}{\alpha}\right)^2} + \sigma_V^2 \\ \implies \quad R_{YY}(\tau) &= \frac{\sigma_W^2}{2\alpha} e^{-\alpha|\tau|} + \sigma_V^2 \delta(\tau) \end{aligned}$$

(b) We want to find an $H'(f)$ such that:

$$S_{YY}(f) = |H'(f)|^2 S_{EE}(f) = H'(f)H'(-f)$$

Clearly this requires spectral factorization, so lets start by factoring $S_{YY}(f)$:

$$\begin{aligned} S_{YY}(f) &= \frac{1}{\alpha^2 + (2\pi f)^2}\sigma_W^2 + \sigma_V^2 = \frac{5}{4 + (2\pi f)^2} + 1 = \frac{9 + (2\pi f)^2}{4 + (2\pi f)^2} = \frac{(3 + j2\pi f)(3 - j2\pi f)}{(2 + j2\pi f)(2 - j2\pi f)} \\ &= \left[\frac{3 + j2\pi f}{2 + j2\pi f}\right]\left[\frac{3 - j2\pi f}{2 - j2\pi f}\right] = H'(f)H'(-f) \end{aligned}$$

Thus one possible choice of the system function is $H'(f) = \dfrac{3 + j2\pi f}{2 + j2\pi f} = 1 + \dfrac{1}{2 + j2\pi f}$

The corresponding impulse response is $h'(t) = \delta(t) + e^{-2t}u(t)$.

(c) No the answer is not unique. For example, note that multiplying the $H'(f)$ given in (b) by any term of the form $(k + j2\pi f)/(k - j2\pi f)$ will also work.

(d) Yes, they have the same pdfs. Their second order statistics are the same by design and they are Gaussian, thus their pdfs must be the same. Note that in general this will *not* be the case – i.e. we have only assured that their second-order characteristics are the same, not their complete pdfs. Gaussians are the exception.

Problem 5.8 A lab test is carried out to determine the model for a spring. The experiment include a position, $X[n]$, and force measuring transducer that measures the force $F[n]$. Data is collected for compression and expansion of the spring. We model the system using a second order ARMA model:

$$X[n] + a_1 X[n-1] + a_2 X[n-2] = F[n]$$

Setup a linear system of equations to identify the coefficients, a_1, a_2.
Solution: Assuming that we have joint wide-sense stationarity we obtain

$$R_{XF}[0] + a_1 R_{XF}[1] + a_2 R_{XF}[2] = R_{FF}[0] \tag{4}$$
$$R_{XF}[1] + a_1 R_{XF}[0] + a_2 R_{XF}[1] = R_{FF}[1] \tag{5}$$

We can solve for a_1 and a_2 by solving this linear set of equations.

Computer Problems

Problem 5.9 Linear Systems Driven by White Noise

In this project we examine the output of a number of systems driven by white noise. The systems will be discrete-time rational systems specified in terms of the vectors of their numerator and denominator coefficients, `b` and `a`. We will use the MATLAB definition of such systems, which differs from that used in the notes – in particular, the non-unity coefficients differ by a sign. MATLAB uses the following system definition:

$$H(e^{j\omega}) = \frac{B(z)}{A(z)} = \frac{b_1 + b_2 z^{-1} + \cdots + b_{N_z+1} z^{-N_z}}{1 + a_2 z^{-1} + \cdots + a_{N_p+1} z^{-N_p}} \tag{6}$$

Thus the a_i coefficients define the poles and the b_j coefficients define the zeros and gain. The numerator coefficients are placed in the vector `b` and the denominator coefficients in the vector `a`. Let us define 5 systems, specified by the following sets of coefficients:

System #0:	`b0 = [5, 0]`	`a0 = [1, -1.3, 0.845]`
System #1:	`b1 = [1, 1.3, 0.845]`	`a1 = [1, -1.3, 0.845]`
System #2:	`b2 = [0.3, 0]`	`a2 = [1, -0.8]`
System #3:	`b3 = [0.06, 0.12, 0.06]`	`a3 = [1, -1.3, 0.845]`
System #4:	`b4 = [0.845, -1.3, 1]`	`a4 = [1, -1.3, 0.845]`

System #0 is second-order AR, System #2 is first-order AR, while Systems #1 and #3 are 2nd order ARMA, and System #4 is an all pass. We will drive each of these systems by a white noise process $U(n)$ to obtain an output $Y(n)$, as depicted in the figure below.

$$U(n) \longrightarrow \boxed{h(n)} \longrightarrow Y(n)$$

(a) We will need to estimate autocorrelation functions from data in our study of linear systems and random processes. As we learned in class, if a process is ergodic, we can estimate its autocorrelation function from a single sample path by forming the following time average:

$$\widehat{R}_{XX}(k) = \frac{1}{L} \sum_{n=0}^{L-1} x(n)x(n+k) \tag{7}$$

If the signal is finite, of length N, then we must have $L \le N - k$. Note in particular, that this means that we must average fewer points in estimating large values of lag k. Thus, given a finite signal, our estimate $\widehat{R}_{XX}(k)$ is more reliable for small values of k. The estimate of $R_{XX}(k)$ in (7) based on the data in the vector `x` can be computed in MATLAB as follows:

```
[Rest,lags] = xcorr(x,maxlag,'unbiased');
```

where `maxlag` is the maximum lag to compute (i.e. maximum value of k in (7)) and the parameter `'unbiased'` tells matlab to divide the sum by L as defined in (7) (by default MATLAB only performs the summing operation). The resulting correlation function can be viewed using `plot(lags,Rest)`.

To test this procedure generate a long vector of unit variance discrete-time white Gaussian noise `u` as follows:

```
N = 5000;
u = randn(1,N);
```

Now use the approach described above to estimate the correlation function for the Gaussian white noise vector `u`. Use a value of `maxlag=50`. What is the theoretical autocorrelation function $R_{UU}(k)$ of `u`? Plot both the theory and experiment together. Do they agree? We now have a way of estimating the autocorrelation function of a sequence.

(b) For each of the given systems find and plot together both the impulse response `h` and the pole/zero plot. This is easily done given the coefficient vectors `b` and `a` by using the MATLAB functions `impz.m` and `zplane.m`. Make your impulse responses 50 steps long (i.e. use $N = 50$ in `impz.m`). Save your impulse responses for part (d).

(c) Find the output `y` of each of the systems when driven by the Gaussian white noise process `u`. This can be simply done using the coefficient vectors for the systems together with the MATLAB function `filter.m`. For each of the systems, compute the pdf of the output sequence using `pdf1d.m` when the input is the Gaussian white noise sequence `u`.

(d) Using the output `y` you found in part (c), estimate the output autocorrelation function $R_{YY}(k)$ for each system, in the manner described in part (a) above. Use `maxlag=50`.

On the web site you will find the MATLAB function `syscor.m`, which calculates the *theoretical* autocorrelation function of the output of a discrete-time system when driven by white noise. Recall from class that for a system driven by white noise, this output is given by $R_{YY}(n) = h(n) * h(-n)$. Using this supplied function, calculate the theoretical autocorrelation functions of the output of each system with `maxlag = 50`.

For each system, plot your autocorrelation estimates together with the corresponding theoretical values. How sensitive are the autocorrelation estimates to data length (i.e. what happens if you use only part of the output data in `y`). What happens if you use uniformly distributed white noise instead of Gaussian white noise? Can you explain the results obtained for System #4? What happens to the estimate for larger lag values?

(e) One way to find the impulse response of an linear time-invariant (LTI) system is to drive it with an impulse and record the output. In this part, we will use a different approach based on the cross-correlation function. Recall that the cross-correlation between the input $U(n)$ and output $Y(n)$ of an LTI system with impulse response $h(n)$ is given by:

$$R_{UY}(k) = R_{UU}(k) * h(k) \tag{8}$$

If the input sequence $U(n)$ is unit variance white noise such as we are using, so that $R_{UU}(k) = \delta(k)$, what is the cross-correlation function $R_{UY}(k)$? Use this insight to estimate the impulse response of each of the systems based on the cross-correlation between the white Gaussian noise input and the corresponding output. The cross-correlation between an input `u` and output `y` can be estimated in MATLAB as follows:

```
[Ruy,lag] = xcorr(u,y,maxlag,'unbiased');
```

Use `maxlag = 50`. Plot your estimated impulse responses together with the true impulse responses from part (a) and compare. Note that driving a real system with an impulse can be difficult, while driving it with white noise is often much easier.

Solution:

(a) Since the noise is white and of unit variance the theoretical autocorrelation function is given by :

$$R_{UU}(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

Figure 1 shows the theoretical and empirical values of the autocorrelation function for white Gaussian noise.

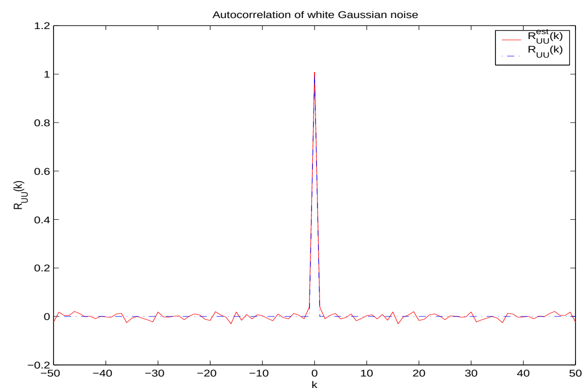


Figure 1: Theoretical and empirical autocorrelation function for white (unit variance) Gaussian noise.

(b) Figure 2 displays the required impulse responses and pole-zero plots for the given five systems.

(c) When the input to the systems is white Gaussian noise, the output is Gaussian (because it is sum of Gaussians), stationary and ergodic. Hence we can calculate the pdf of the output by using the output samples at different times (instead of having to generate several sample paths of the output). These pdfs are shown in Figure 3 for the given systems driven by white Gaussian noise.

(d) Figure 4 shows the theoretical and estimated values of the autocorrelation function of the output of the systems driven by white noise. As seen from the figures, as expected, the estimated value of the autocorrelation function deviates from the theoretical values to a greater extent for larger lag values than for small lag values. Also the estimate worsens if we use fewer number of data samples to estimate the autocorrelation function. However if we replace the white Gaussian noise by white uniform noise (with unit variance) the estimate of the autocorrelation function does not change at all, since we are dealing with second order statistics only.

System #4 is an allpass system (as can be seen by plotting the FFT of it's impulse response) and hence it does not change the autocorrelation function of the input random signal. Therefore the autocorrelation function of the output of System # 4 is the same as the one given in Equation(7).

(e) From Equation (6) we see that if the input sequence is unit variance white noise, so that $R_{UU}(k) = \delta(k)$ then :

$$R_{UY}(k) = h(k) \tag{9}$$

Hence we can estimate the impulse response of the system by estimating the cross-correlation between the input and output of the system. Figure 5 displays the estimated cross correlation between the input and output of the systems along with the true impulse response of the systems.

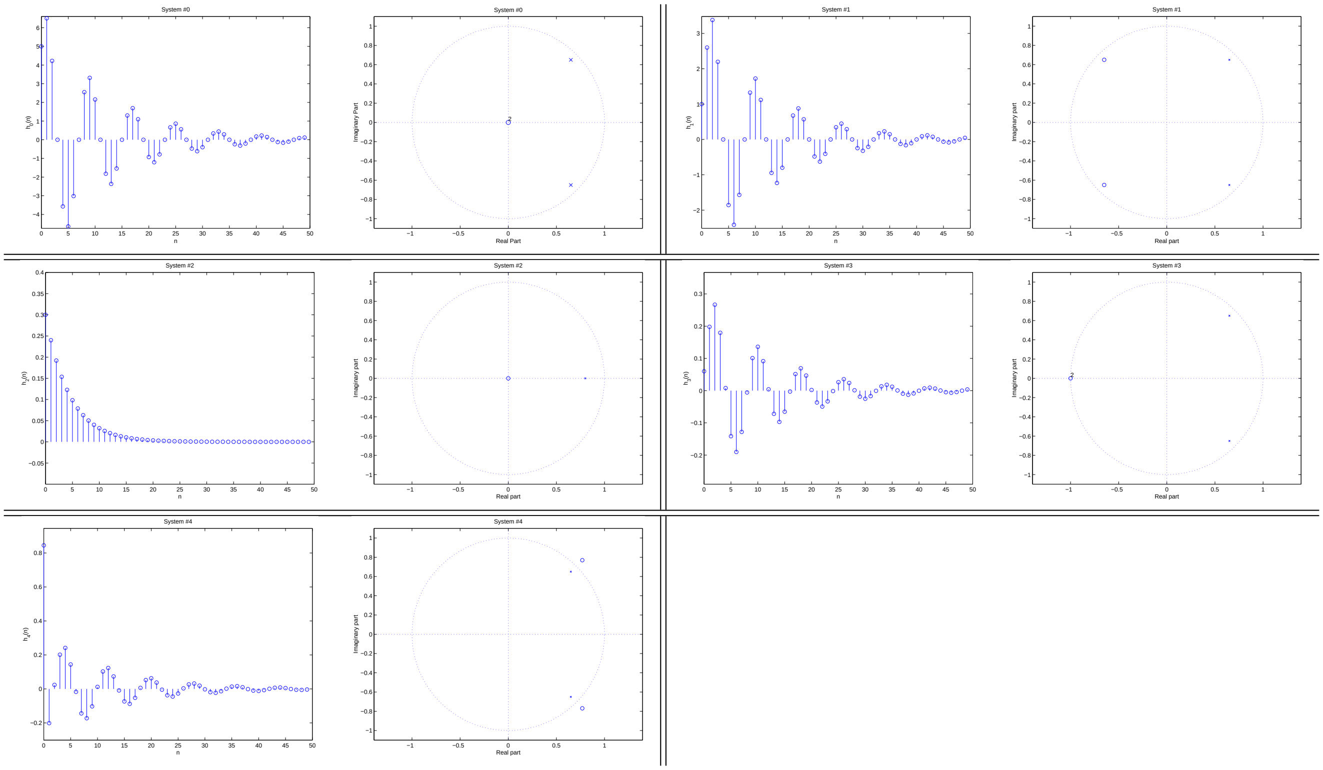


Figure 2: Impulse response and pole-zero plot for the given five systems

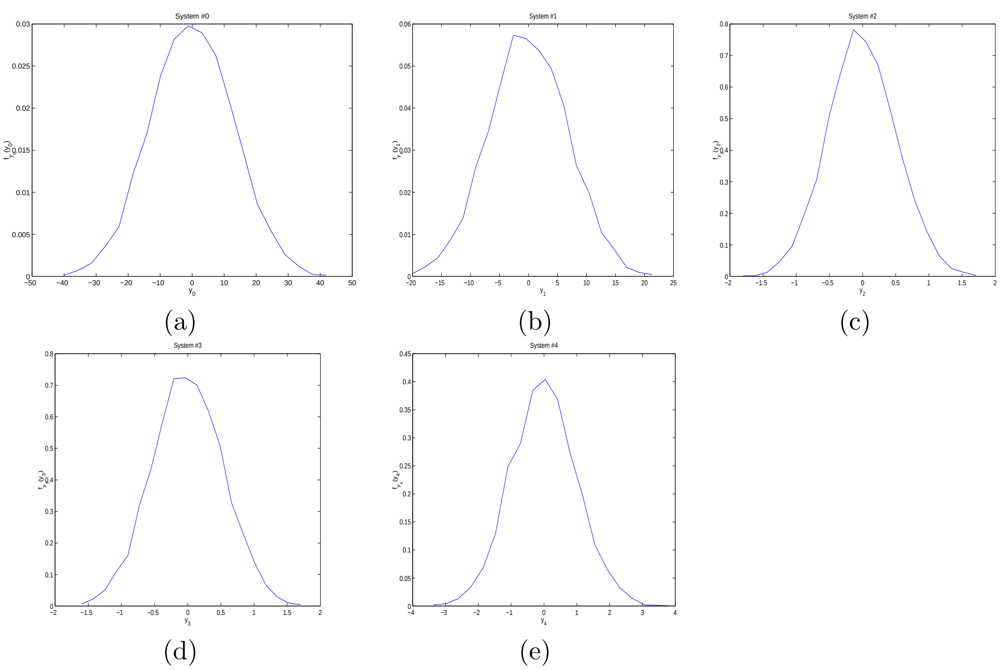


Figure 3: Probability density functions of the output of the systems driven by white Gaussian noise. (a) System #0, (b) System #1, (c) System #2, (d) System #3, (e) System #4

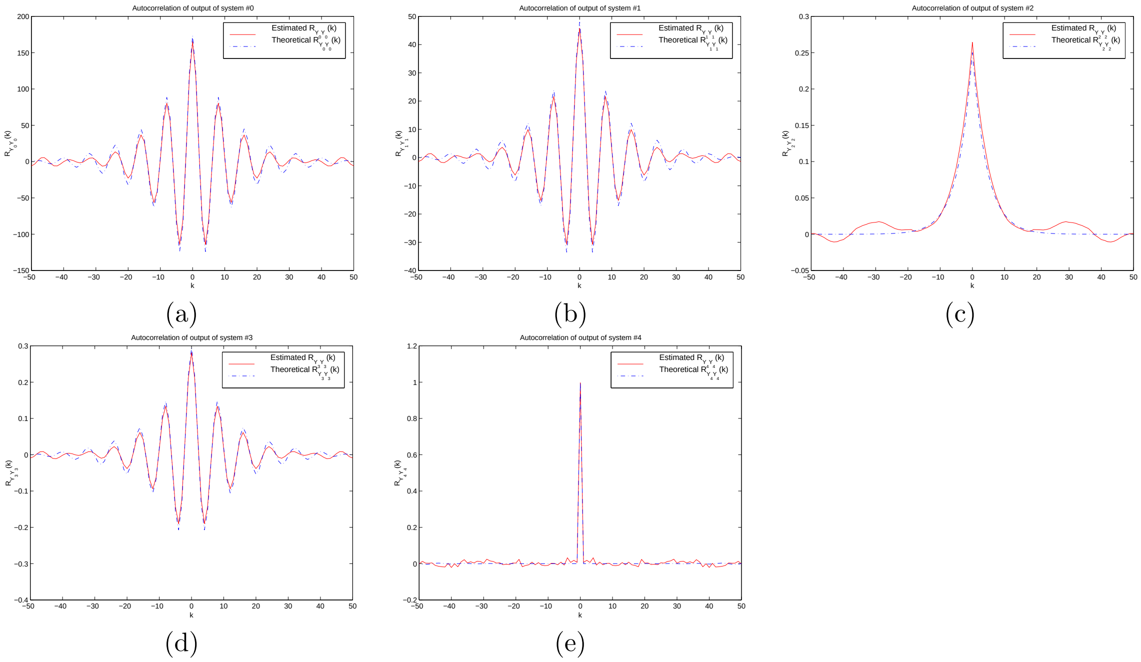


Figure 4: Autocorrelation functions of the output of the systems driven by white Gaussian noise. (a) System #0, (b) System #1, (c) System #2, (d) System #3, (e) System #4

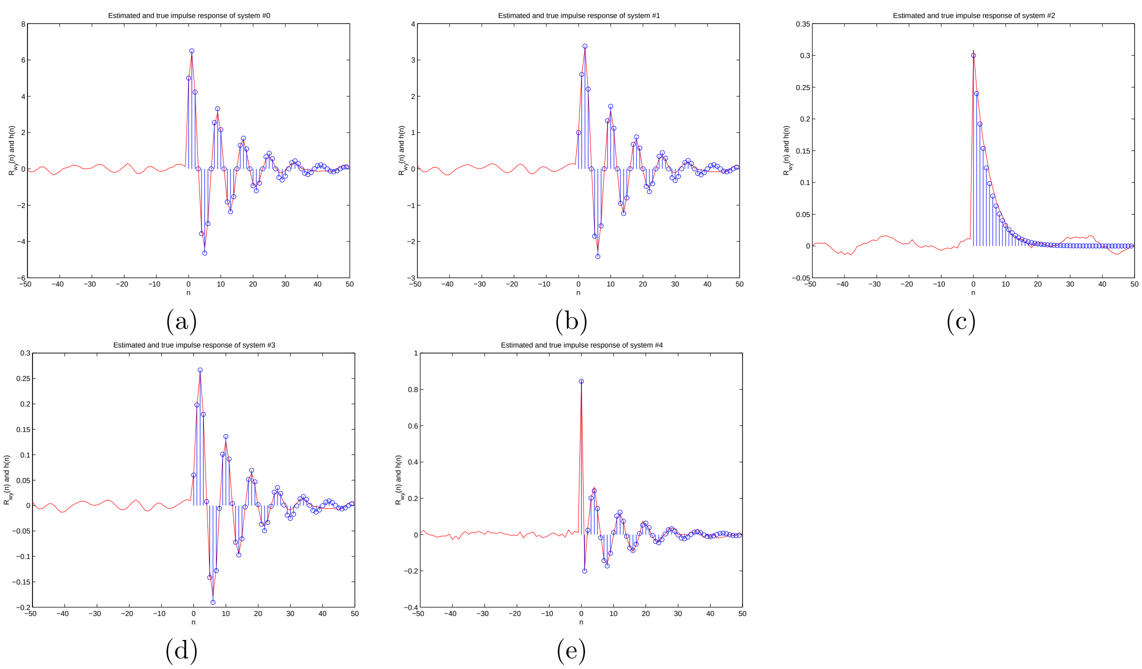


Figure 5: Estimated (*continuous curve*) and true (*stems*) impulse response of the systems (a) System #0, (b) System #1, (c) System #2, (d) System #3, (e) System #4

Problem 5.10 Power Spectral Density Estimation via FFT

An important precursor to many signal processing problems is the estimation of the power spectrum of a process. In this project we will construct and investigate an FFT-based method for power spectral density estimation. This topic is discussed in more detail in the reserve texts Shanmugan and Breiphol, Section 9.3 or Therrien Section 10.1.

(a) You will write a function `psdest1.m` to estimate the power spectral density of a process given the sample path data in the vector `y`. The function will first estimate the autocorrelation function of the process and then will take an FFT of these values to obtain an estimate of the power spectral density function. The function call for your program will be the following: `[Syy,w] = psdest1(y,maxlag)`, where `y` is the input data vector, `maxlag` will be the number of lags to use in the correlation estimate, `Syy` will be the estimated power spectral density, and `w` will be the corresponding vector of frequency values. Each step below will form a line of the program:

(i) The first step is estimate the values of the autocorrelation function for lags between `-maxlag` and `+maxlag` as described previously:

```
[Ryy,lag] = xcorr(y,maxlag,'unbiased');
```

(ii) Given the estimated autocorrelation function in `Ryy`, we now estimate the power spectral density by taking its FFT:

```
Syy = abs(fft(Ryy));
Syy = Syy(1:maxlag+1);
```

There are two slightly subtle implementational details in this part. First, because MATLAB assumes the first element of a vector representing a time series occurs at time $n = 0$ and not the "center" of the vector, an undesired linear phase will be added to the transform of `Ryy`. This is removed by taking the magnitude of the FFT via `abs.m`. Since we know the transform of `Ryy` is real, this will have no effect on the true `Syy`. Second, the FFT program calculates the values of the spectrum from $\omega = 0$ to $\omega = 2\pi$, but we know the spectrum is symmetric. Thus we only keep those values corresponding to frequency samples in the range $[0, \pi]$, which are the first `maxlag+1` values.

(iii) Finally, we need to generate the vector of corresponding frequency points `w`. The FFT samples we generate correspond to `maxlag+1` uniformly spaced frequency values in the range $[0, \pi]$, thus:

```
w = linspace(0,pi,maxlag+1);
```

Add these steps together to create your program. Test your function by estimating and plotting the power spectral densities of the white noise process `u` vs frequency. You may wish to use `semilogy.m` for your plots.

(b) Use your function `psdest1.m` to estimate the power spectral densities of the outputs of the 5 systems for the white Gaussian noise input `u`. On the web site you will find the function `syspsd.m` which finds the *theoretical* power spectral density for the output of a discrete-time system driven by white noise. Recall from class that for a system with frequency response $H(e^{j\omega})$ driven by white noise this output PSD is given by $|H(e^{j\omega})|^2$. For each of the 5 systems, use this function to find the corresponding theoretical power spectral density and plot it along with your experimentally determined power spectral density. Use `maxlag = 100` and `Nf = 100`. Again, you may wish to use `semilogy.m` for your plots so you can see more detail. How do the estimated power spectral densities change as `maxlag` is made smaller or larger?

The parameter `maxlag` truncates the correlation function $R_{YY}(m)$ to the range `[-maxlag, +maxlag]`, and thus essentially defines a rectangular window which we apply in the "time" domain to the estimated correlation function before taking the Fourier transform. As a result, in the frequency domain it can be shown that the expected value of the estimated spectrum is the convolution of the Fourier transform of this rectangular window (i.e. a sinc function) with the true spectrum. Thus, shorter rectangular windows (smaller values of `maxlag`) introduce blurring and bias into the estimate. There can also be problems with oscillations due to the sidelobes of the sinc function in the frequency domain.

At the other extreme, as we let the rectangular window size `maxlag` become large and approach the length of the data N_y, it can be shown that the estimate produced by `psdest1.m` approaches $\widehat{S}_{YY}(\omega) = \frac{1}{N_y}|\mathcal{F}\{y(n)\}|^2$, where $\mathcal{F}[y(n)]$ is the discrete time Fourier transform of the data sequence. This estimate is known as the "periodogram" and it is unfortunately true that the variance of this estimate is as large as its mean! Thus, in choosing a value of `maxlag` we must trade off bias and resolution against variance. A general rule of thumb is to choose `maxlag` no larger than say 10% of N_y. Note that we have been using `maxlag` $= 1\%$ of N_y. Using the output of System #0, see what happens if you use a value of `maxlag` significantly larger than our rule. More on these issues can be found in the reserve texts Shanmugan and Breiphol Section 9.3 or Therrien Section 10.1.

Solution:

(a) The `MATLAB` code for the function `psdest1.m` is given below:

```
function [Syy,w] = psdest1(y,maxlag)
% [Syy,w] = psdest1(y,maxlag)
%
% y      : Vector of data
% maxlag : Max lag to use for psd estimation. Only the estimated ACF values
%          within this window around the zero lag are used. OPTIONAL. Default:
%          maxlag = 50.
%
% Syy    : Vector of PSD estimates, Syy(w)
% w      : Corresponding freqency vector (Radians)
%
% Finds psd estimate based on the "Periodogram Technique". Generates the
% estimate by first estimating the correlation function, then taking the FFT
% of the result.

% 2/98 W. C. Karl

if nargin < 2
  maxlag = 50;
end;

if min(size(y)) > 1
  error('Input must be a vector')
end;
y = y(:);

% Step 1: Estimate the autocorrelation function for lags:-maxlag< lag <maxlag.
[Ryy,lag] = xcorr(y,maxlag,'unbiased');

% Step 2: Estimate psd by taking fft of the estimated Ryy.
% We take the magnitude to obtain a real spectrum
% The result is shifted so that 0 frequency is at the center
Syy = abs(fft(Ryy));
Syy = Syy(1:maxlag+1);
```

```
% Generate corresponding frequency axis
% Frequency spacing dw is 2pi/(length of FFT + 1)
% Frequency samples run from -pi+dw to pi-dw in increments of dw
w = linspace(0,pi,maxlag+1);
```

The power spectral density of a white noise process, calculated using the above code, is plotted in Figure 6

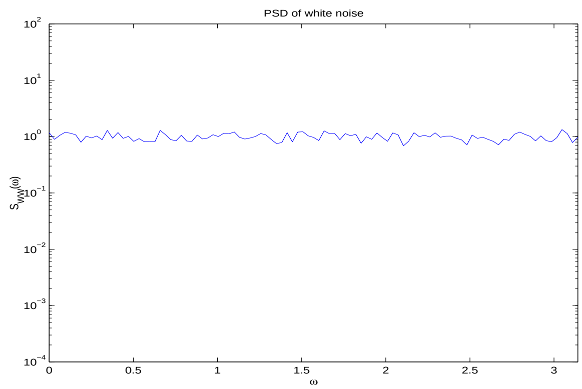


Figure 6: Power Spectral Density for white (unit variance) Gaussian noise.

(b) The estimates of the power spectral densities of the output of the five systems are shown in Figure 7 along with the plots of the theoretical value of the PSDs. These estimates were generated for a `maxlag` value of 100. If we use a much larger value of `maxlag`, the variance of the estimate increases as is seen in Figure 8 which shows the estimate of the PSD of the output of System #0 calculated using `maxlag` value of 1000.

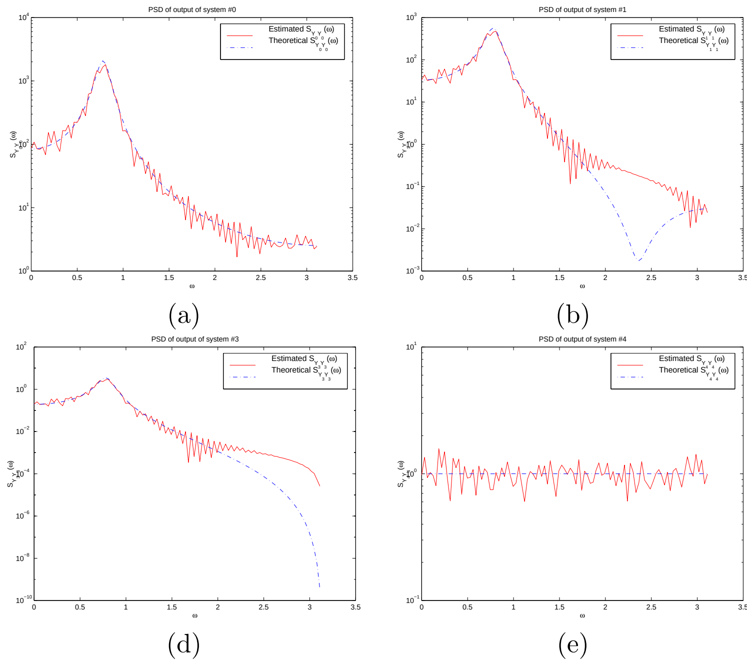


Figure 7: Estimated and theoretical power spectral densities of the output of the systems driven by white noise (a) System #0, (b) System #1, (c) System #2, (d) System #3, (e) System #4

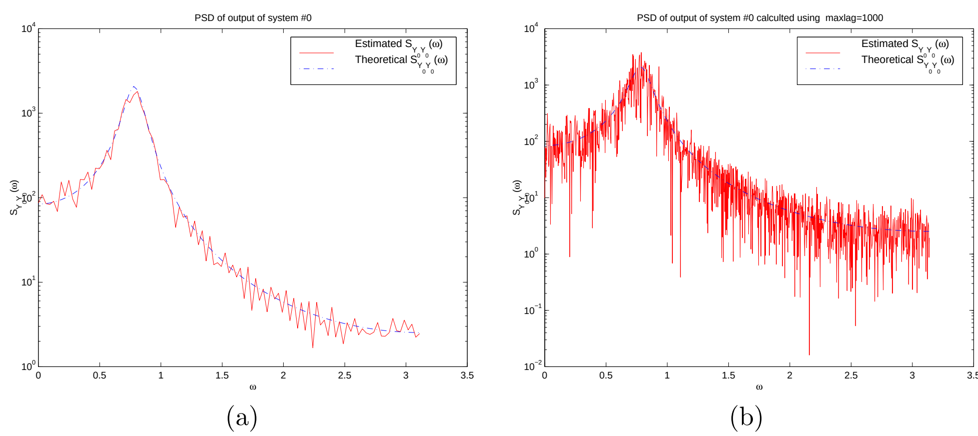


Figure 8: Power Spectral Density estimate for the output of System #0 calculated using (a) `maxlag` = 100, (b) `maxlag` = 1000