CHAPTER 14

Random Signal Processing

Tutorial Problems

1. (a) See plot below.

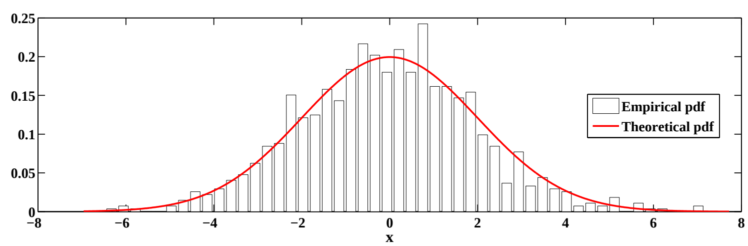

FIGURE 14.1: Empirical and theoretical pdfs of random variable $x \sim N(0, \sigma^2)$ for $\sigma = 2$.

   (b) See script below.
   (c) See plot below.
   (d) Comments:
   See script output.
   (e) See plot below.
   (f) See plots below.

   MATLAB script:

```
% P1401: Simulation for the quality of unbiased mean estimator
close all; clc
% sigma = 2; N = 40; K = 1000;
sigma = sqrt(2); N = 20; K = 1000; % part f
%% Part a
```

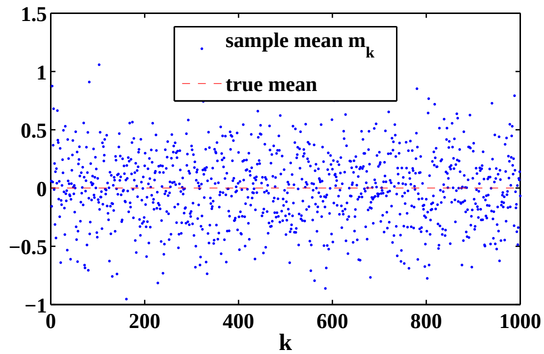

FIGURE 14.2: Plot of the true mean μ and the K sample means.

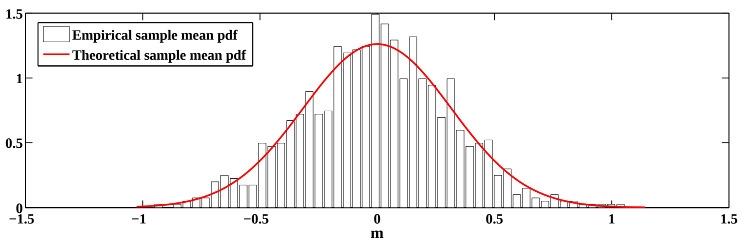

FIGURE 14.3: Empirical and theoretical pdfs of the sample mean.

```
x = randn(1,1000)*sigma;
[xo px] = epdf(x,50);
xx = linspace(1.1*min(xo),1.1*max(xo),1000);
fxx = normpdf(xx,0,sigma);
hfa = figconfg('P1401a','long');
bar(xo,px,'facecolor','w'); hold on
plot(xx,fxx,'r','linewidth',2)
xlabel('x','fontsize',LFS)
legend('Empirical pdf','Theoretical pdf','location','best')
%% Part b
X = randn(K,N)*sigma;
%% Part c
mhatk = mean(X,2);
hfb = figconfg('P1401b','small');
plot(1:K,mhatk,'.b',1:K,zeros(1,K),'--r')
xlabel('k','fontsize',LFS)
legend('sample mean m_k','true mean','location','best')
```

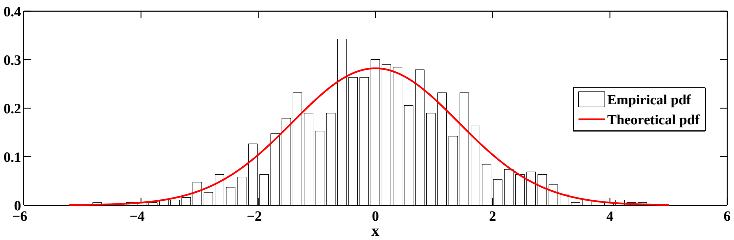

FIGURE 14.4: Empirical and theoretical pdfs of random variable $x \sim N(0, \sigma^2)$ for $\sigma = \sqrt{2}$.

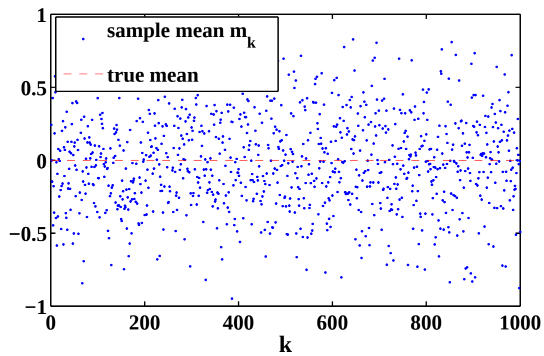

FIGURE 14.5: Plot of the true mean μ and the K sample means.

```
%% Part d
mu_m = mean(mhatk), mu_true = 0,
var_m = var(mhatk), var_true = sigma^2/N,
%% Part e
[mo pm] = epdf(mhatk,50);
mm = linspace(1.1*min(mo),1.1*max(mo),1000);
fmm = normpdf(mm,0,sigma/sqrt(N));
hfc = figconfg('P1401c','long');
bar(mo,pm,'facecolor','w'); hold on
plot(mm,fmm,'r','linewidth',2)
xlabel('m','fontsize',LFS)
legend('Empirical sample mean pdf',...
    'Theoretical sample mean pdf','location','best')
```

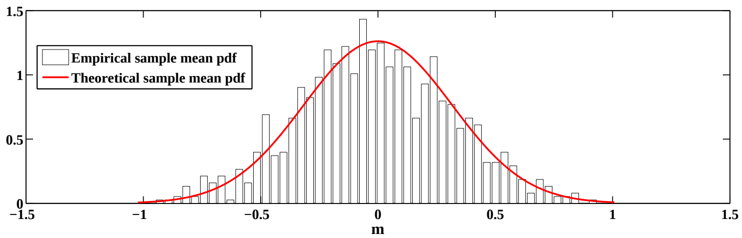

FIGURE 14.6: Empirical and theoretical pdfs of the sample mean.

2. tba.

3. (a) Solution:
We first determine the marginal pdfs of x and y.
If $-1 \leq x \leq 0$, the pdf of x is

$$f(x) = \int_{-2}^{0} \frac{1}{4} dy = \frac{1}{2}$$

If $0 \leq x \leq 2$, the pdf of x is

$$f(x) = \int_{0}^{1} \frac{1}{4} dy = \frac{1}{4}$$

Hence, the pdf of x is

$$f(x) = \begin{cases} 1/2, & -1 \leq x \leq 0 \\ 1/4, & 0 \leq x \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

If $0 \leq y \leq 1$, the pdf of y is

$$f(y) = \int_{0}^{2} \frac{1}{4} dx = \frac{1}{2}$$

If $-2 \leq y \leq 0$, the pdf of y is

$$f(y) = \int_{-1}^{0} \frac{1}{4} dx = \frac{1}{4}$$

Hence, the pdf of y is

$$f(y) = \begin{cases} 1/4, & -2 \leq y \leq 0 \\ 1/2, & 0 \leq y \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

We can compute the mean of x and y as

$$\mu_x = \int_{-\infty}^{\infty} xf(x)dx = \int_{-1}^{0} \frac{x}{2}dx + \int_{0}^{2} \frac{x}{4}dx = \frac{1}{4}$$

$$\mu_y = \int_{-\infty}^{\infty} yf(y)dy = \int_{-2}^{0} \frac{y}{4}dy + \int_{0}^{1} \frac{y}{2}dy = -\frac{1}{4}$$

(b) Solution:
Given the pdfs of x and y obtained in last step, we have

$$E[x^2] = \int_{-1}^{0} \frac{x^2}{2}dx + \int_{0}^{2} \frac{x^2}{4}dx = \frac{5}{6}$$

$$\sigma_x^2 = E[x^2] - E^2[x] = \frac{5}{6} - \left(\frac{1}{4}\right)^2 = \frac{37}{48}$$

$$\sigma_x = \sqrt{\frac{37}{48}} \approx 0.8780$$

$$E[y^2] = \int_{-2}^{0} \frac{y^2}{4}dy + \int_{0}^{1} \frac{y^2}{2}dy = \frac{5}{6}$$

$$\sigma_y^2 = E[y^2] - E^2[y] = \frac{5}{6} - \left(-\frac{1}{4}\right)^2 = \frac{37}{48}$$

$$\sigma_y = \sqrt{\frac{37}{48}} \approx 0.8780$$

(c) Solution:
The correlation r_{xy} is

$$\begin{aligned} r_{xy} = E[xy] &= \iint xyf(x,y)dxdy \\ &= \frac{1}{4}\left(\int_0^2 xdx\right)\left(\int_0^1 ydy\right) + \frac{1}{4}\left(\int_{-1}^0 xdx\right)\left(\int_{-2}^0 ydy\right) = \frac{1}{2} \end{aligned}$$

$$c_{xy} = r_{xy} - \mu_x\mu_y = \frac{1}{2} - \frac{1}{4}\left(-\frac{1}{4}\right) = \frac{9}{16} = 0.5625$$

The correlation coefficient ρ_{xy} is

$$\rho_{xy} = \frac{c_{xy}}{\sigma_x\sigma_y} = \frac{9/16}{37/48} = \frac{27}{37} \approx 0.7297$$

(d) MATLAB script:

```
% P1403: Investigation of correlation between two random variables
close all; clc
% N = 1e4;
% N = 1e5;
N = 1e6;
x1 = rand(N/2,1)*2; y1 = rand(N/2,1);
x2 = rand(N/2,1)-1; y2 = (rand(N/2,1)-1)*2;
x = [x1;x2]; y = [y1;y2];
mu_x = mean(x), mu_y = mean(y),
std_x = std(x), std_y = std(y),
rxy = corr(x,y)*std_x*std_y + mu_x*mu_y,
rhoxy = corrcoef(x,y),
```

4. (a) Solution:
The mean of $x_2[n]$ is:

$$E[x_2[n]] = E[x_1[n]] + E[0.01n] = 0 + 0.01 \times \sum_{i=1}^{1000} i/1000 = 5.005$$

$$E[x_2^2[n]] = E[(x_1[n] + 0.01n)^2] = E[x_1^2[n] + 0.02nx_1[n] + 0.0001n^2]$$
$$= 1 + 0 + 10^{-4} \times \sum_{n=1}^{1000} n^2/1000 = 34.3834$$

The variance of $x_2[n]$ is:

$$\sigma_{x_2}^2 = E[x_2^2[n]] - E^2[x_2[n]] = 9.3784$$

The mean of $x_3[n]$ is:

$$E[x_3[n]] = \frac{1}{2}$$
$$E[x_3^2[n]] = \frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2}$$

The variance of $x_3[n]$ is:

$$\sigma_{x_3}^2 = \frac{3}{2} - \left(\frac{1}{2}\right)^2 = \frac{5}{4}$$

The mean of $x_4[n]$ is

$$E[x_4[n]] = 0$$

The variance of $x_4[n]$ is

$$\sigma_{x_4}^2 = E[x_4^2[n]] = \frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2}$$

(b) See plot below.

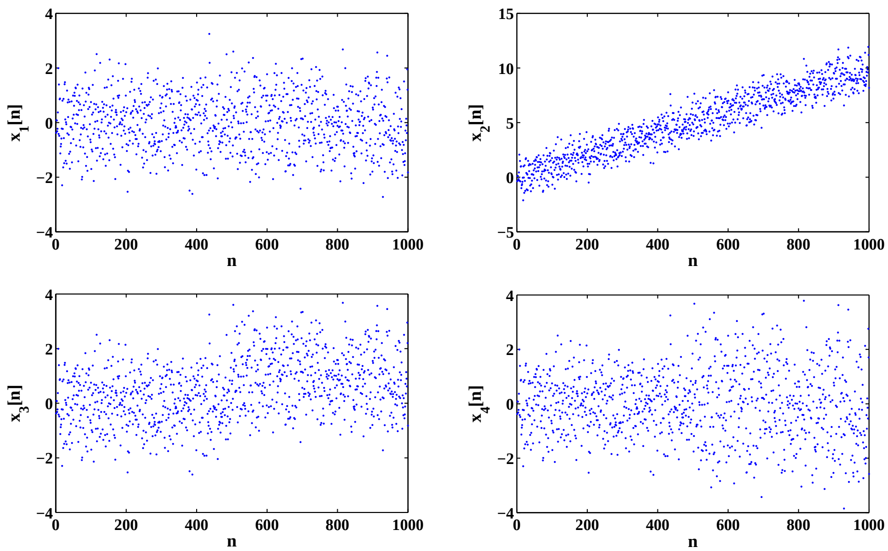

FIGURE 14.7: Plot of the data sets.

MATLAB script:

```
% P1404: Mean and Variance estimate verification
close all; clc
N = 1e3; n = (1:N)';
x1 = randn(N,1);
x2 = x1 + 0.01*n;
x3 = x1 + (n > 500);
x4 = x1; x4(501:end) = x4(501:end)*sqrt(2);
%% Plot
hfa = figconfg('P1404a','small');
plot(n,x1,'.')
xlabel('n','fontsize',LFS)
```

```
ylabel('x_1[n]','fontsize',LFS)
hfb = figconfg('P1404b','small');
plot(n,x2,'.')
xlabel('n','fontsize',LFS)
ylabel('x_2[n]','fontsize',LFS)
hfc = figconfg('P1404c','small');
plot(n,x3,'.')
xlabel('n','fontsize',LFS)
ylabel('x_3[n]','fontsize',LFS)
hfd = figconfg('P1404d','small');
plot(n,x4,'.')
xlabel('n','fontsize',LFS)
ylabel('x_4[n]','fontsize',LFS)
```

5. (a) Solution:
   The mean μ_x is:

$$\mu_x = aE[x[n-1]] + E[w[n]] = 0$$

Define $\ell = m - n$, and $\ell \geq 0$ we have

$$\begin{aligned} r_{xx}[\ell] &= E[x[m]x[n]] = E[(ax[m-1]+w[m])(ax[n-1]+w[n])] \\ &= a^2 r_{xx}[\ell] + aE[x[m-1]w[n]] + aE[x[n-1]w[m]] + \sigma_w^2\delta[\ell] \\ &= a^2 r_{xx}[\ell] + aE[x[m-1]w[n]] + \sigma_w^2\delta[\ell] \end{aligned}$$

If $\ell = 1$,

$$r_{xx}[1] = a^2 r_{xx}[1] + a\sigma_w^2 \implies r_{xx}[1] = \frac{a\sigma_w^2}{1-a^2}$$

If $\ell = 0$,

$$r_{xx}[0] = a^2 r_{xx}[0] + \sigma_w^2 \implies r_{xx}[0] = \frac{\sigma_w^2}{1-a^2}$$

Otherwise

$$r_{xx}[\ell] = 0$$

Hence, we can conclude that

$$r_{xx}[\ell] = \frac{\sigma_w^2}{1-a^2}\delta[\ell] + \frac{a\sigma_w^2}{1-a^2}\delta[|\ell|-1]$$

Let $\ell = 0$, the variance σ_x^2 is

$$\sigma_x^2 = r_{xx}[0] = \frac{\sigma_w^2}{1-a^2}$$

The correlation coefficient $\rho_x[\ell]$ is

$$\rho_x[\ell] = \delta[\ell] + a\,\delta[|\ell| - 1]$$

(b) See plot below.

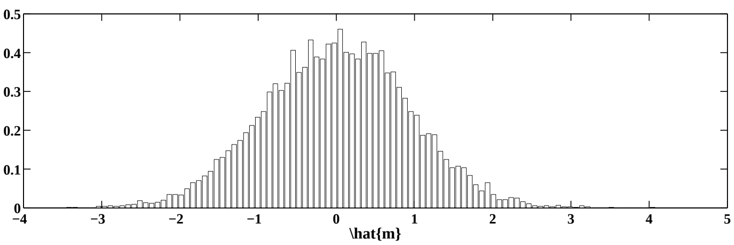

FIGURE 14.8: Plot of the histogram of the estimated mean when $a = 0.9$.

(c) See plot below.

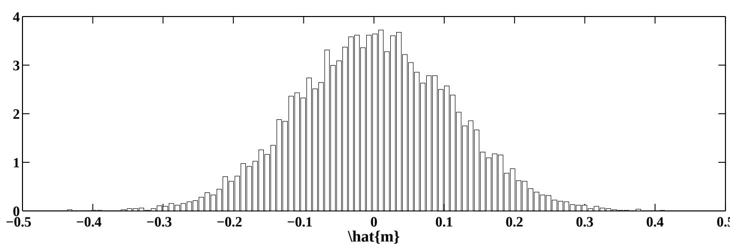

FIGURE 14.9: Plot of the histogram of the estimated mean when $a = 0.1$.

(d) tba.

MATLAB script:

```
% P1405: Sample Mean Investigation
close all; clc
N = 100; K = 1e4;
a = 0.9;
% a = 0.1; % Part c
wn = randn(N,K);
```

```
xn = filter(1,[1 -a],wn);
mhatk = mean(xn,1);
[mo pm] = epdf(mhatk,100);
%% Plot
hfa = figconfg('P1405a','long');
bar(mo,pm,'facecolor','w');
xlabel('\hat{m}','fontsize',LFS)
```

6. (a) See plot below.

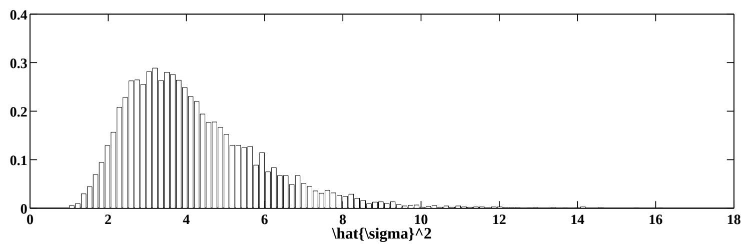

FIGURE 14.10: Plot of the histogram of the estimated mean when $a = 0.9$.

(b) See plot below.

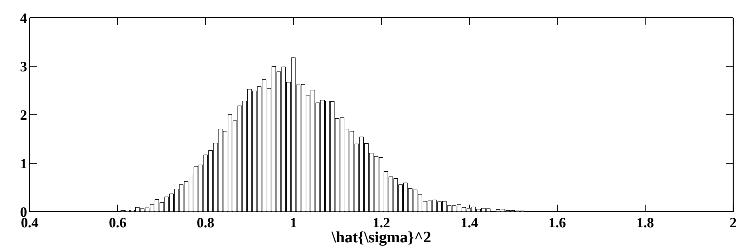

FIGURE 14.11: Plot of the histogram of the estimated mean when $a = 0.1$.

(c) tba.

MATLAB script:

```
% P1406: Sample Variance Investigation
close all; clc
N = 100; K = 1e4;
% a = 0.9;
a = 0.1; % Part c
wn = randn(N,K);
```

```
xn = filter(1,[1 -a],wn);
varhatk = var(xn,1,1);
[mo pm] = epdf(varhatk,100);
%% Plot
hfa = figconfg('P1406a','long');
bar(mo,pm,'facecolor','w');
xlabel('\hat{\sigma}^2','fontsize',LFS)
```

7. tba

8. (a) Proof:

$$E[\hat{S}_D(\omega_k)] = E\left[\frac{1}{2Q+1}\sum_{m=-Q}^{Q} I(\omega_{k-m})\right] = \frac{1}{2Q+1}\sum_{m=-Q}^{Q} E\left[I(\omega_{k-m})\right]$$

Since $I(\omega)$ is essentially an unbiased estimate, we have

$$E[\hat{S}_D(\omega_k)] = \frac{1}{2Q+1}\sum_{m=-Q}^{Q} S(\omega_{k-m}) \approx \frac{1}{2Q+1}\sum_{m=-Q}^{Q} S(\omega_k) = S(\omega_k)$$

(b) Proof:

$$\begin{aligned}
E[\hat{S}_D^2(\omega_k)] &= \frac{1}{(2Q+1)^2} E\left[\sum_{m=-Q}^{Q} I(\omega_{k-m}) \sum_{n=-Q}^{Q} I(\omega_{k-n})\right] \\
&= \frac{1}{(2Q+1)^2}\left(E\left[\sum_{m=-Q}^{Q} I(\omega_{k-m})\right] E\left[\sum_{n=-Q}^{Q} I(\omega_{k-n})\right] + \sum_{m=-Q}^{Q} \mathrm{Var}[I(\omega_{k-m})]\right) \\
&= \frac{1}{(2Q+1)^2}\left(\sum_{m=-Q}^{Q} S(\omega_{k-m}) \sum_{n=-Q}^{Q} S(\omega_{k-n}) + \sum_{m=-Q}^{Q} \mathrm{Var}[I(\omega_{k-m})]\right) \\
&\approx \frac{1}{(2Q+1)^2}\left[(2Q+1)^2 S^2(\omega_k) + (2Q+1)\mathrm{Var}[I(\omega_k)]\right] \\
&= S^2(\omega_k) + \frac{\mathrm{Var}[I(\omega_k)]}{2Q+1}
\end{aligned}$$

$$\mathrm{Var}[\hat{S}_D(\omega_k)] = E[\hat{S}_D^2(\omega_k)] - E^2[\hat{S}_D(\omega_k)] = \frac{\mathrm{Var}[I(\omega_k)]}{2Q+1} \approx \frac{S^2(\omega_k)}{2Q+1}$$

(c) Comments:
The variance reduction factor is $2Q+1$.

(d) See plots below.

MATLAB script:

```
% P1408: Daniell Method
close all; clc
N = 2048;
vn = randn(1,N);
xn = filter(1,[1 -0.75 0.5],vn);
Q = 4;
% Q = 8;
% Q = 16;
% Q = 32;
I = psdper(xn,N);
I_zp = [zeros(2*Q,1);I;zeros(2*Q,1)];
Im = zeros(2*Q+1,N);
for ii = 1:N
    Im(:,ii) = I_zp(ii:ii+2*Q);
end
SD = mean(Im,1);
w = linspace(0,2,N)*pi;
%% Plot
hfa = figconfg('P1408a','long');
plot(w/pi,I,'g'); hold on
plot(w/pi,SD,'linewidth',1,'color','r')
ylim([0 20])
xlabel('\omega/\pi','fontsize',LFS)
title(['Q = ',num2str(Q)],'fontsize',LFS)
legend('I(\omega_k)','\hat{S}_D(\omega_k)','location','best')
```

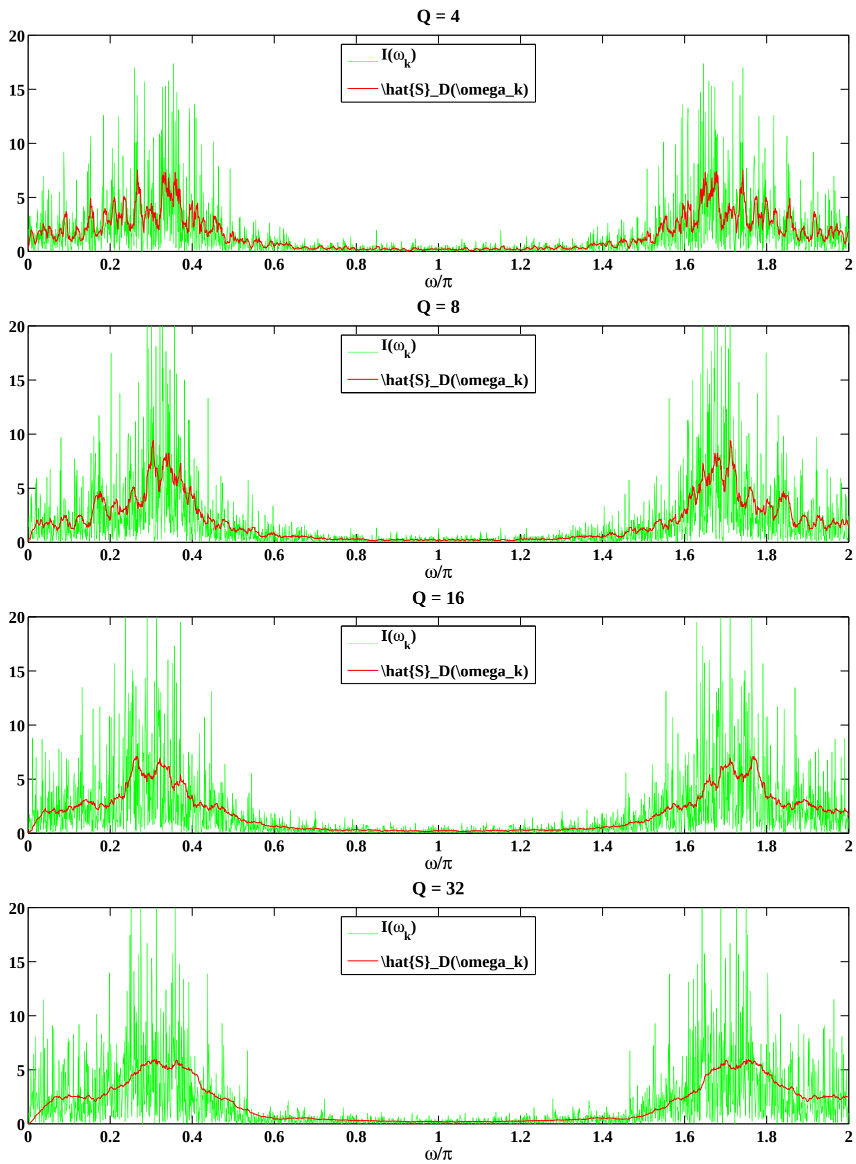

FIGURE 14.12: Plot of $I(\omega_k)$ and $\hat{S}_D(\omega_k)$ for $Q = 4$, $Q = 8$, $Q = 16$, and $Q = 32$.

9. (a) See plot below.

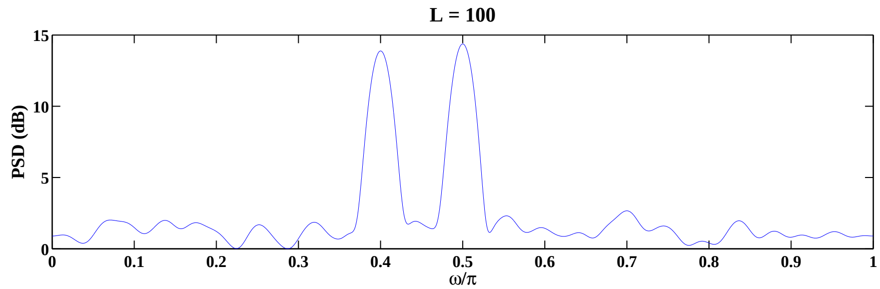

FIGURE 14.13: Plot of PSD when $L = 100$.

(b) See plot below.

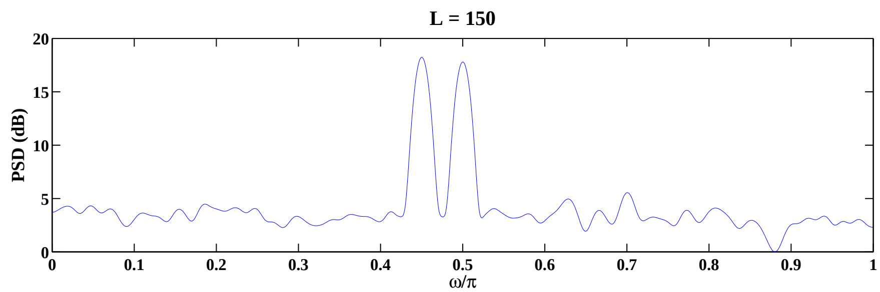

FIGURE 14.14: Plot of PSD when $L = 150$.

(c) tba.

MATLAB script:

```
% P1409: Investigate closing window aspect of Blackman-Tukey PSD estimate
close all; clc
A1 = 1; A2 = 1; A3 = 0.2; om2 = 0.5*pi; om3 = 0.7*pi; N = 5e3;
om1 = 0.4*pi; % L = 100;
% om1 = 0.45*pi; % L = 150
L = 20;
vn = randn(1,N); phi = rand(1,3)*2*pi-pi;
n = 1:N;
xn = A1*cos(om1*n+phi(1)) + A2*cos(om2*n+phi(2))...
    + A3*cos(om3*n+phi(3)) + vn;
w = linspace(0,1,2^(ceil(log2(N))-1))*pi;
```

```
hfa = figconfg('P1409a','long');
while L < N
    S = psdbt(xn',L,2^(ceil(log2(N))));
    figure(hfa)
    Sdb = 10*log10(S./min(S));
    plot(w/pi,Sdb)
    xlabel('\omega/\pi','fontsize',LFS)
    ylabel('PSD (dB)','fontsize',LFS)
    title(['L = ',num2str(L)],'fontsize',TFS)
    pause
    L = L + 10;
end
```

10. (a) See plot below.

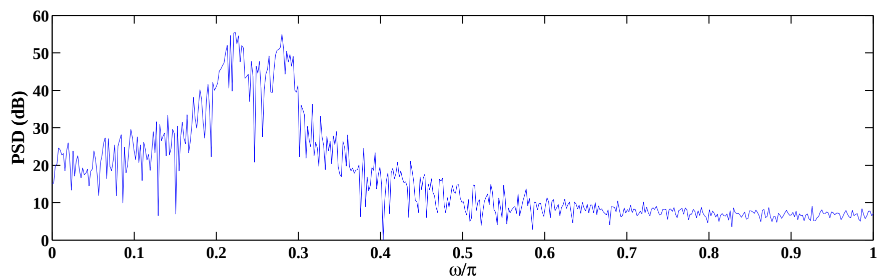

FIGURE 14.15: Periodogram of $x[n]$ using a 1024-point FFT.

(b) See plot below.

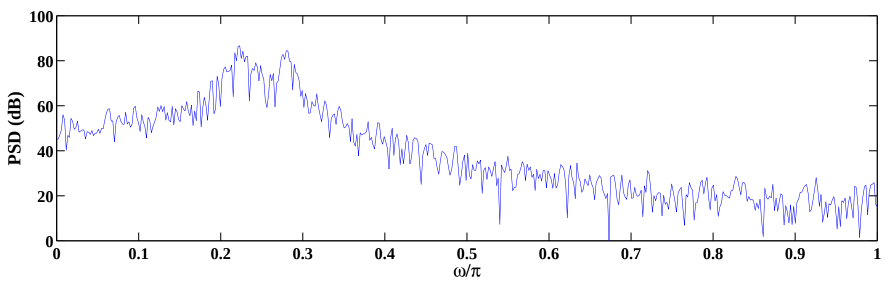

FIGURE 14.16: Modified periodogram of $x[n]$ using a 1024-point FFT based on Bartlett data window.

(c) tba.

MATLAB script:

```
% P1410: Periodogram and Modified Periodogram
close all; clc
N = 1024;
vn = randn(N,1);
xn = filter(1,[1 -2.7607 3.8106 -2.6535 0.9238],vn);
w = linspace(0,1,N/2)*pi;
%% Part a: Periodogram
I1 = psdper(xn,N);
hfa = figconfg('P1410a','long');
I1db = 10*log10(I1(1:N/2)./min(I1(1:N/2)));
plot(w/pi,I1db)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
%% Part b: Modified Periodogram
I2 = psdmodper2(xn,N,bartlett(N));
hfb = figconfg('P1410b','long');
I2db = 10*log10(I2(1:N/2)./min(I2(1:N/2)));
plot(w/pi,I2db)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
```

11. (a) See plot below.

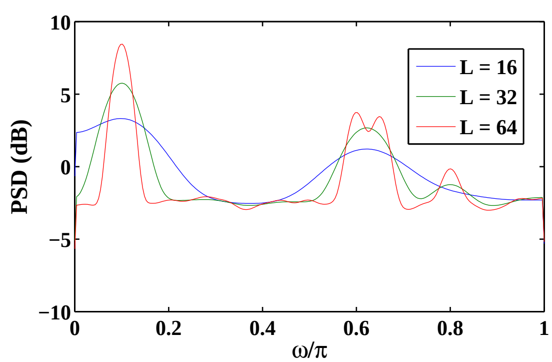

FIGURE 14.17: Plot when using 75 percent overlap, Hamming window, and $L = 16$, 32, and 64.

(b) See plot below.

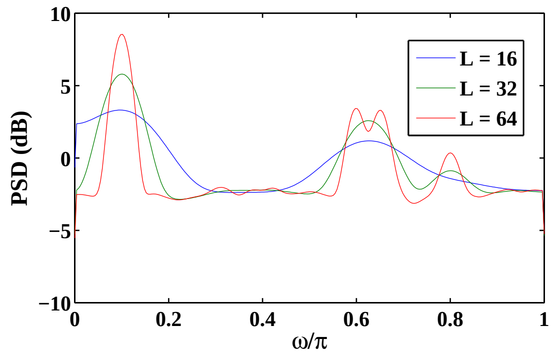

FIGURE 14.18: Plot when using 50 percent overlap, Hamming window, and $L = 16$, 32, and 64.

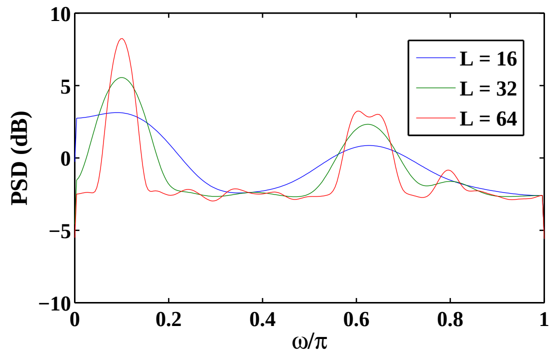

FIGURE 14.19: Plot when using 50 percent overlap, Hann window, and $L = 16$, 32, and 64.

(c) See plot below.

(d) tba.

MATLAB script:

```
% P1411: Investigate Welsh PSD estimate
close all; clc
A = [1 0.5 0.5 0.25]; om = [0.1 0.6 0.65 0.8]*pi;
phi = rand(1,4)*2*pi-pi;
N = 256; K = 50; n = (0:N-1)'; Nk = repmat(n,1,K);
V = randn(N,K);
X = V;
```

```
for jj = 1:4
    X = X + A(jj)*sin(om(jj)*Nk+phi(jj));
end
L = [16 32 64]; Nfft = 512;
Pxx = zeros(K,Nfft/2+1,length(L));
for jj = 1:length(L)
    for ii = 1:K
        [Pxx(ii,:,jj), w] = ...
            pwelch(X(:,ii),hamming(L(jj)),fix(0.75*L(jj)),Nfft); % part a
%          [Pxx(ii,:,jj), w] = ...
%              pwelch(X(:,ii),hamming(L(jj)),fix(0.5*L(jj)),Nfft); % part b
%          [Pxx(ii,:,jj), w] = ...
%              pwelch(X(:,ii),hann(L(jj)),fix(0.5*L(jj)),Nfft); % part c
    end
end
P = squeeze(mean(Pxx,1));
Pdb = bsxfun(@rdivide,P,mean(P,1));
Pdb = 10*log10(Pdb);
%% Part a: Periodogram
hfa = figconfg('P1411a','small');
plot(w/pi,Pdb)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
legend(['L = ',num2str(L(1))],['L = ',num2str(L(2))],...
    ['L = ',num2str(L(3))],'location','best')
```

12. Solution:

$$H_1(z) = \frac{1}{1-az^{-1}} \Leftrightarrow h_1[n] = a^n u[n] \Leftrightarrow H_1(\mathrm{e}^{\mathrm{j}\omega}) = \frac{1}{1-a\mathrm{e}^{-\mathrm{j}\omega}}$$

$$H_2(z) = \frac{1}{1-z^{-1}} \Leftrightarrow h_1[n] = u[n] \Leftrightarrow H_2(\mathrm{e}^{\mathrm{j}\omega}) = \frac{1}{1-\mathrm{e}^{-\mathrm{j}\omega}}$$

$$|H_1(\mathrm{e}^{\mathrm{j}\omega})|^2 = \frac{1}{(1-a\cos\omega)^2 + a^2\sin^2\omega}, \quad |H_2(\mathrm{e}^{\mathrm{j}\omega})|^2 = \frac{1}{2-2\cos\omega}$$

The variance of $x[n]$ is

$$\sigma_x^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_1(\mathrm{e}^{\mathrm{j}\omega})|^2 |H_2(\mathrm{e}^{\mathrm{j}\omega})|^2 \sigma_w^2 \delta(\omega) d\omega = \infty$$

Hence, the process $x[n]$ is NOT stationary.

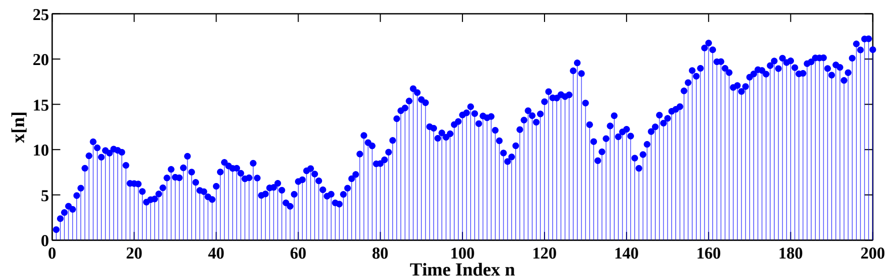

FIGURE 14.20: Plot $N = 200$ samples of $x[n]$.

(a) See plot below.

(b) Solution:

$$S(\omega) = \sigma_w^2 \times \frac{1}{(1 - a\cos\omega)^2 + a^2\sin^2\omega} \times \frac{1}{2 - 2\cos\omega}$$

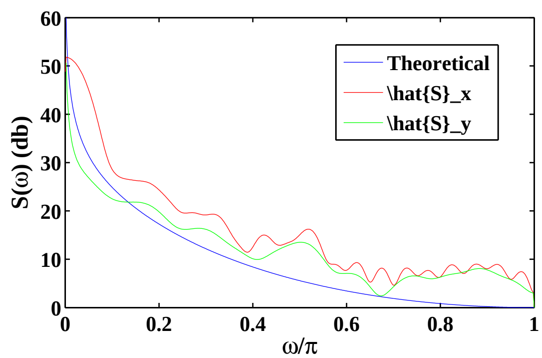

FIGURE 14.21: Plots of theoretical and empirical $S(\omega)$ and Welch PSD estimate $\tilde{S}_y(\omega)$ when $N = 200$.

(c) See plot below.

(d) See script for details.

(e) See script for details.

MATLAB script:

```
% P1412: PSD estimate
close all; clc
```

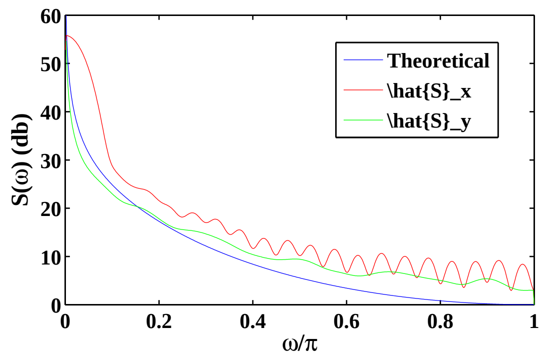

FIGURE 14.22: Plots of theoretical and empirical $S(\omega)$ and Welch PSD estimate $\tilde{S}_y(\omega)$ when $N = 1000$.

```
% N = 200;
N = 1e3;
sigmaw = 1;
a = 0.5;
randn('seed',0)
wn = randn(1,N)*sigmaw;
vn = filter(1,[1 -a],wn);
xn = filter(1,[1 -1],vn);
%% Part a:
hfa = figconfg('P1412a','long');
stem(1:N,xn,'filled')
xlabel('Time Index n','fontsize',LFS)
ylabel('x[n]','fontsize',LFS)
%% Part b:
w = linspace(0,1,1001)*pi;
Sw = sigmaw^2./(1+a^2-2*a*cos(w))./(2-2*cos(w));
Sw(1) = nan;
Swdb = 10*log10(Sw./min(Sw));
hfb = figconfg('P1412b','small');
plot(w/pi,Swdb)
ylim([0 60])
xlabel('\omega/\pi','fontsize',LFS)
ylabel('S(\omega) (db)','fontsize',LFS)
%% Part c:
L = 40; Nfft = 1024;
```

```
[Swhat, w2] = pwelch(xn,hamming(L),fix(0.5*L),Nfft);
figure(hfb)
hold on
Swhatdb = 10*log10(Swhat./min(Swhat));
plot(w2/pi,Swhatdb,'r')
%% Part d:
yn = filter([1 -1],1,xn);
[Swhat2, w3] = pwelch(yn,hamming(L),fix(0.5*L),Nfft);
Swhat2 = Swhat2./abs(1-exp(-w3)).^2;
Swhat2(1) = nan;
Swhat2db = 10*log10(Swhat2./min(Swhat2));
figure(hfb)
plot(w3/pi,Swhat2db,'g')
legend('Theoretical','\hat{S}_x','\hat{S}_y','location','best')
```

13. Solution:

$$\begin{aligned} J &= E[(y-\hat{y})^2] = E[(y-b)^2] = E[y^2 - 2by + b^2] \\ &= E[y^2] - 2bE[y] + b^2 \end{aligned}$$

Take the first derivative and assign it to zero, we have

$$\frac{dJ}{db} = -2E[y] + 2b = 0 \implies b = E[y]$$

Hence, the optimum constant linear estimator is the mean

14. (a) See script for details.
    (b) See plot below.
    (c) See plot below.
    (d) tba.

    MATLAB script:

```
% P1414: Matched filtering
close all; clc
N = 200; n = 0:N-1;
sn = zeros(size(n));
%% Part b
p = 10;
ind = (n >= 0 & n <= p-1);
```

```
sn(ind) = cos(2*pi*n(ind)/10);
%% Part c
% p = 100;
% ind = (n >= 0 & n <= p-1);
% sn(ind) = cos(2*pi*n(ind)/10)/sqrt(10);

randn('seed',0)
xn = sn + randn(1,N);
yn = filter(sn(p:-1:1),1,xn);

%% Part a:
hfa = figconfg('P1414a','small');
plot(1:N,sn)
title('Signal','fontsize',LFS)

hfb = figconfg('P1414b','small');
plot(1:N,xn)
title('Signal+Noise','fontsize',LFS)

hfc = figconfg('P1414c','small');
plot(1:N,yn)
title('Matched Filter Output','fontsize',LFS)
xlabel('n','fontsize',LFS)
```

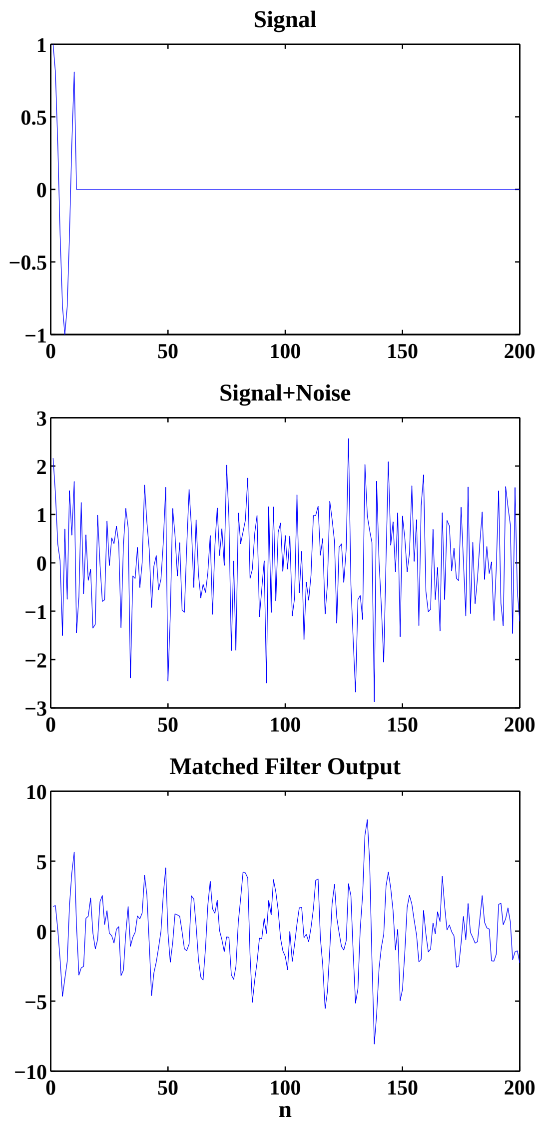

FIGURE 14.23: Plots of $s_i[n] = \cos(2\pi n/10)$, noisy signal $x[n]$ and matched filter output.

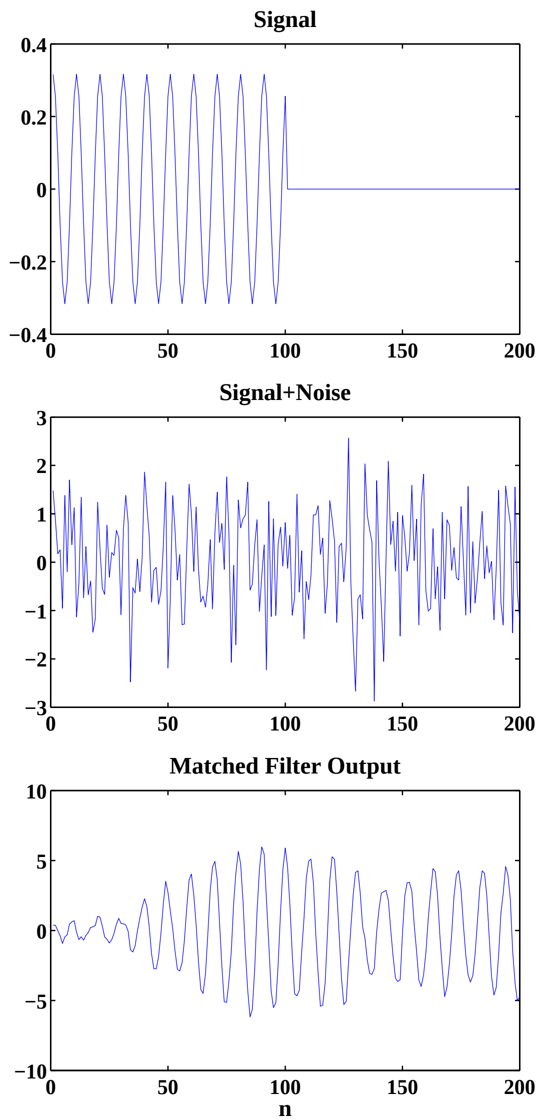

FIGURE 14.24: Plots of $s_i[n] = (1/\sqrt{10})\cos(2\pi n/10)$, noisy signal $x[n]$ and matched filter output.

15. (a) Solution:
The theoretical PSD $S_y(\omega)$ is:

$$S_y(\omega) = \left|\frac{1}{1-0.95\mathrm{e}^{-\mathrm{j}\omega}}\right|^2 \cdot 4 = \frac{4}{1+0.95^2-1.9\cos\omega}$$

(b) Solution:

$$H_o(\omega) = \frac{S_y(\omega)}{S_y(\omega)+1}$$

(c) See script below.

(d) See plot below.

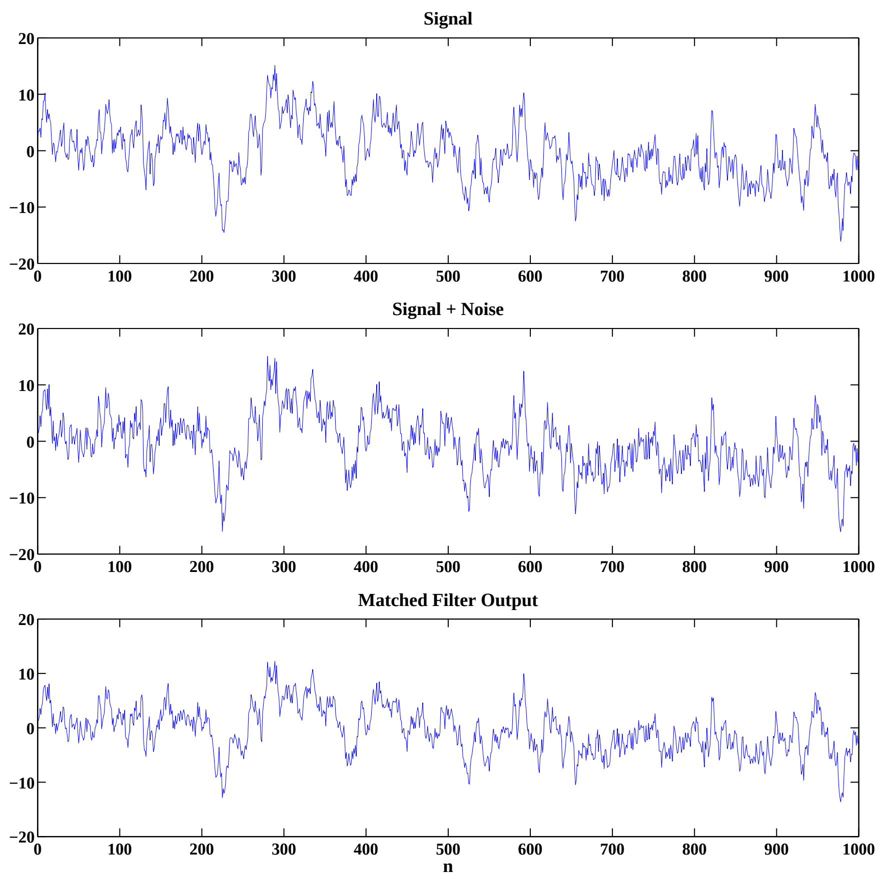

FIGURE 14.25: Plots of clean, noisy and filtered signals.

MATLAB script:

```
% P1415: Wiener Filter
close all; clc
N = 1000; n = 0:N-1;
randn('seed',0)
wn = randn(1,N)*2;
yn = filter(1,[1 -0.95],wn);
vn = randn(1,N);
xn = yn + vn;
w = linspace(0,2,1001)*pi;
Sy = 4./(1+0.95^2-1.9*cos(w));
Hw = Sy./(Sy+1);
hn = real(ifft(Hw));
ynhat = filter(hn,1,xn);
%% Plot
hfa = figconfg('P1415a','long');
plot(n,yn)
title('Signal','fontsize',LFS)

hfb = figconfg('P1415b','long');
plot(n,xn)
title('Signal + Noise','fontsize',LFS)

hfc = figconfg('P1415c','long');
plot(n,ynhat)
title('Matched Filter Output','fontsize',LFS)
xlabel('n','fontsize',LFS)
```

16. Proof:

$$J_o = r_y[0] - \sum_{k=-\infty}^{\infty} h_o[k] r_{yx}[k]$$

$$r_y[0] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_y(\omega) d\omega$$

Applying the Parseval's theorem, we have

$$\sum_{k=-\infty}^{\infty} h_o[k] r_{yx}[k] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{S_{yx}(\omega)}{S_x(\omega)} S'_x(\omega) \frac{S'_{yx}(\omega)}{S'_x(\omega)} d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|S_{yx}(\omega)|^2}{S_x(\omega)} d\omega$$

Hence, we have

$$J_o = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_y(\omega)d\omega - \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|S_{yx}(\omega)|^2}{S_x(\omega)}d\omega$$
$$= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[1 - \frac{|S_{yx}(\omega)|^2}{S_x(\omega)S_y(\omega)}\right]S_y(\omega)d\omega$$

17. (a) Proof:
Suppose we have

$$\begin{cases} c_1 = r\cos\alpha \\ c_2 = r\sin\alpha \end{cases} \qquad \begin{cases} x_1 = r\cos(\alpha+\theta) \\ x_2 = r\sin(\alpha+\theta) \end{cases}$$

Applying the trigonometric identities, we have

$$\begin{cases} x_1 = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta \\ x_2 = r\sin\alpha\cos\theta + r\cos\alpha\sin\theta \end{cases}$$

which equals

$$\begin{cases} x_1 = c_1\cos\theta - c_2\sin\theta \\ x_2 = c_2\cos\theta + c_1\sin\theta \end{cases}$$

(b) Proof:
Suppose we have

$$\begin{cases} x_1 = r\cos\beta \\ x_2 = r\sin\beta \end{cases} \qquad \begin{cases} c_1 = r\cos(\beta-\theta) \\ c_2 = r\sin(\beta-\theta) \end{cases}$$

Applying the trigonometric identities, we have

$$\begin{cases} c_1 = r\cos\beta\cos\theta + r\sin\beta\sin\theta \\ c_2 = r\sin\beta\cos\theta - r\cos\beta\sin\theta \end{cases}$$

which equals

$$\begin{cases} c_1 = x_1\cos\theta + x_2\sin\theta \\ c_2 = x_2\cos\theta - x_1\sin\theta \end{cases}$$

(c) Proof:

$$\boldsymbol{Q}^T\boldsymbol{Q} = \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}$$

18. (a) Proof:

$$\boldsymbol{a}^T \boldsymbol{\Gamma} \boldsymbol{a} = \sum_{i=1}^{3} \sum_{j=1}^{3} a_i a_j \gamma_{ij}$$

Hence,

$$\begin{aligned}
\frac{\partial \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\Gamma} \boldsymbol{a}}{\partial \boldsymbol{a}} &= \left[\begin{array}{ccc} \frac{\sum_{i=1}^{3} \sum_{j=1}^{3} a_i a_j \gamma_{ij}}{\partial a_1} & \frac{\sum_{i=1}^{3} \sum_{j=1}^{3} a_i a_j \gamma_{ij}}{\partial a_2} & \frac{\sum_{i=1}^{3} \sum_{j=1}^{3} a_i a_j \gamma_{ij}}{\partial a_3} \end{array} \right]^T \\
&= \left[\begin{array}{c} 2a_1\gamma_{11} + a_2\gamma_{12} + a_3\gamma_{13} + a_2\gamma_{21} + a_3\gamma_{31} \\ 2a_2\gamma_{22} + a_1\gamma_{12} + a_3\gamma_{32} + a_3\gamma_{23} + a_1\gamma_{21} \\ 2a_3\gamma_{33} + a_1\gamma_{13} + a_2\gamma_{23} + a_1\gamma_{31} + a_2\gamma_{32} \end{array} \right] \\
&= \left[\begin{array}{c} 2a_1\gamma_{11} + 2a_2\gamma_{12} + 2a_3\gamma_{13} \\ 2a_2\gamma_{22} + 2a_1\gamma_{21} + a_3\gamma_{23} \\ 2a_3\gamma_{33} + 3a_1\gamma_{31} + 2a_2\gamma_{32} \end{array} \right] \\
&= 2 \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \right] = 2\boldsymbol{\Gamma}\boldsymbol{a}
\end{aligned}$$

(b) Proof:

$$\boldsymbol{a}^{\mathrm{T}} \boldsymbol{a} = \sum_{i=1}^{3} a_i^2$$

$$\begin{aligned}
\frac{\partial \boldsymbol{a}^{\mathrm{T}} \boldsymbol{a}}{\partial \boldsymbol{a}} &= \frac{\partial \sum_{i=1}^{3} a_i^2}{\partial \boldsymbol{a}} \\
&= \left[\begin{array}{ccc} \frac{\partial \sum_{i=1}^{3} a_i^2}{\partial a_1} & \frac{\partial \sum_{i=1}^{3} a_i^2}{\partial a_2} & \frac{\partial \sum_{i=1}^{3} a_i^2}{\partial a_3} \end{array} \right]^{\mathrm{T}} \\
&= \left[\begin{array}{ccc} 2a_1 & 2a_2 & 2a_3 \end{array} \right]^{\mathrm{T}} = 2\boldsymbol{a}
\end{aligned}$$

19. Proof:

Suppose x_k is zero mean, that is $E[\boldsymbol{x}] = \mathbf{0}$, hence we have

$$E[\boldsymbol{c}] = E[\boldsymbol{A}^T \boldsymbol{x}] = \boldsymbol{A}^T E[\boldsymbol{x}] = \mathbf{0}$$

The variance of x_k is

$$\text{var}(x_k) = E[x_k^2]$$

The left hand of the given equation is

$$\begin{aligned}\sum_{k=1}^{p}\text{var}(x_k) &= \sum_{k=1}^{p}E[x_k^2] = E\left(\sum_{k=1}^{p}x_k^2\right) = E[\boldsymbol{x}^T\boldsymbol{x}]\\ &= E[\boldsymbol{c}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{c}] = E[\boldsymbol{c}^T\boldsymbol{c}] = E\left(\sum_{k=1}^{p}c_k^2\right)\\ &= \sum_{k=1}^{p}\text{var}(c_k)\end{aligned}$$

20. (a) Proof:
The mean of random vector $\boldsymbol{y}$ is

$$E[\boldsymbol{y}] = E[\boldsymbol{\Lambda}^{1/2}\boldsymbol{z}] = \boldsymbol{\Lambda}^{1/2}E[\boldsymbol{z}] = \boldsymbol{0}$$

The covariance matrix of random vector $\boldsymbol{y}$ is

$$\begin{aligned}E[\boldsymbol{y}\boldsymbol{y}^T] &= E[\boldsymbol{\Lambda}^{1/2}\boldsymbol{z}(\boldsymbol{\Lambda}^{1/2}\boldsymbol{z})^T]\\ &= \boldsymbol{\Lambda}^{1/2}E[\boldsymbol{z}\boldsymbol{z}^T]\boldsymbol{\Lambda}^{1/2} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{I}\boldsymbol{\Lambda}^{1/2}\\ &= \boldsymbol{\Lambda}\end{aligned}$$

(b) Proof:
The mean of random vector $\boldsymbol{x}$ is

$$E[\boldsymbol{x}] = E[\boldsymbol{A}\boldsymbol{y}+\boldsymbol{\mu}] = \boldsymbol{A}E[\boldsymbol{y}]+\boldsymbol{\mu} = \boldsymbol{\mu}$$

The covariance matrix of random vector $\boldsymbol{x}$ is

$$\begin{aligned}E[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T] &= E[\boldsymbol{A}\boldsymbol{y}(\boldsymbol{A}\boldsymbol{y})^T] = \boldsymbol{A}E[\boldsymbol{y}\boldsymbol{y}^T]\boldsymbol{A}^T\\ &= \boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{A}^T = \boldsymbol{\Gamma}\end{aligned}$$

(c) MATLAB function:

```
function X = Normal_ND(N,Mu,Gamma)
% P1420: Gaussian random sample vector generator
[U S A] = svd(Gamma);
Z = randn(N,length(Mu));
X = Z*sqrt(S)*A';
X = bsxfun(@plus,X,Mu(:)');
```

(d) MATLAB script:

```
% P1420: Test Gaussian Random Sample Vector Generator
close all; clc
p = 4; N = 1e6;
mu = 1:p; sig = 1:p; rho = abs(bsxfun(@minus,1:p,[1:p]'));
rho = 1-0.1*rho;
Gamma = bsxfun(@times,rho,sig);
Gamma = bsxfun(@times,Gamma,sig');
X = Normal_ND(N,mu,Gamma);
mean(X,1)
mu
cov(X,0)
Gamma
```

21. (a) See plot below.

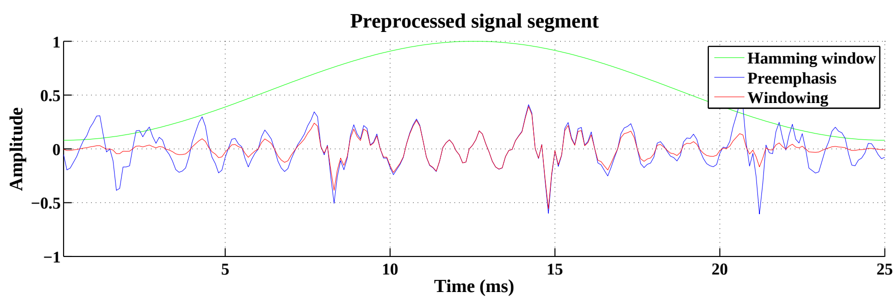

FIGURE 14.26: Replot of Figure 14.22(b).

(b) See plot below.

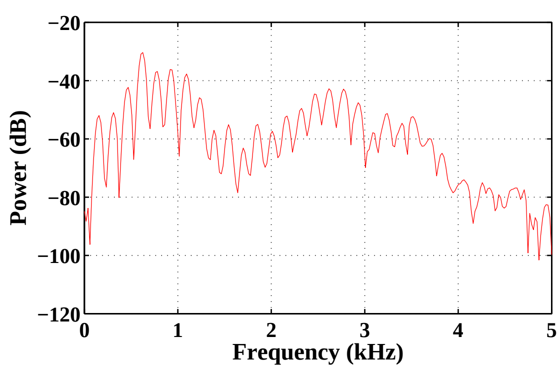

FIGURE 14.27: Plot of the periodogram estimate of the speech spectrum.

(c) See plot below.

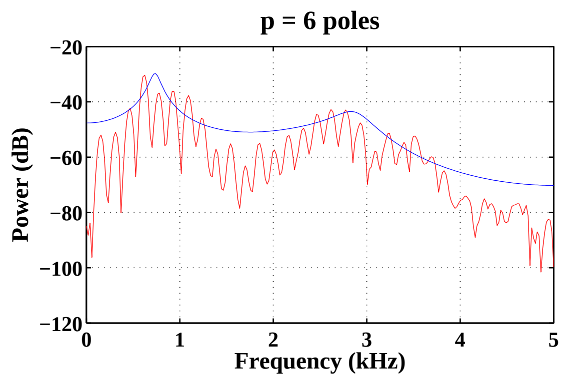

FIGURE 14.28: Plot of the spectrum using all-pole model when $p = 6$.

(d) See plot below.

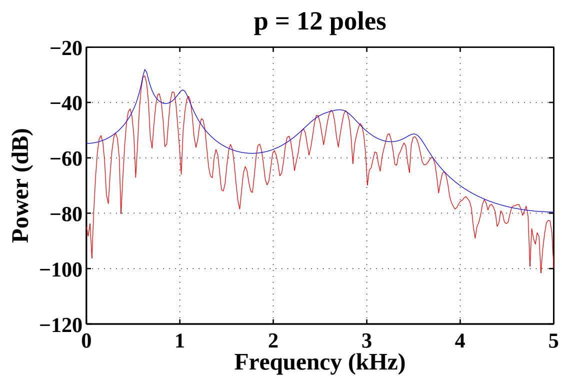

FIGURE 14.29: Plot of the spectrum using all-pole model when $p = 12$.

22. (a) See plot below.
    (b) See plot below.
    (c) Yes.
    (d) Comments:
    See script output for details.
    (e) Solution:
    Data set with $\rho = 0.9$ can be described more accurately if we retain only the first Karhunen-Loève Transform coefficient.

MATLAB script:

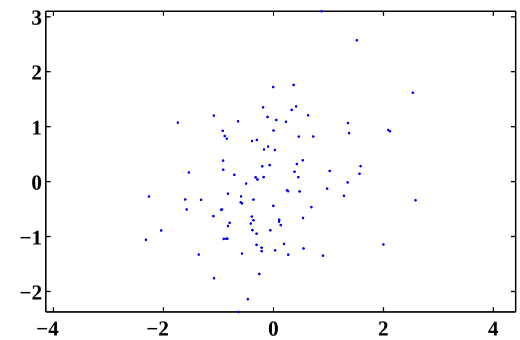

FIGURE 14.30: Scatter diagram of data set with $\sigma_1 = \sigma_2 = 1$, and $\rho = 0.4$.

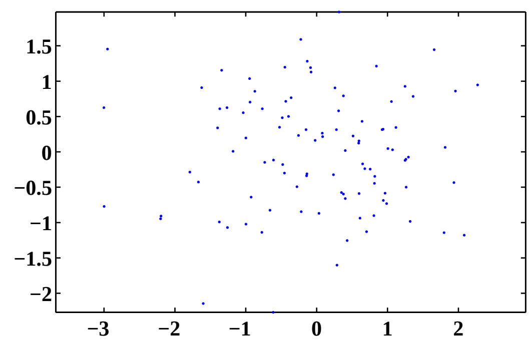

FIGURE 14.31: Scatter diagram of data set Karhunen-Loève Transform $\boldsymbol{c}[n]$.

```
% P1422: Geometrical Interpretations of KLT
close all; clc
N = 100; mu = zeros(1,2);
sig1 = 1; sig2 = 1;
rho = 0.4;
% rho = 0.9;
Gamma = [sig1^2 sig1*sig2*rho;sig1*sig2*rho sig2^2];
randn('seed',0)
X = Normal_ND(N,mu,Gamma);
%% Part b
[N,p]=size(X);
G=cov(X,1);
[U,L,A]=svd(G);
```

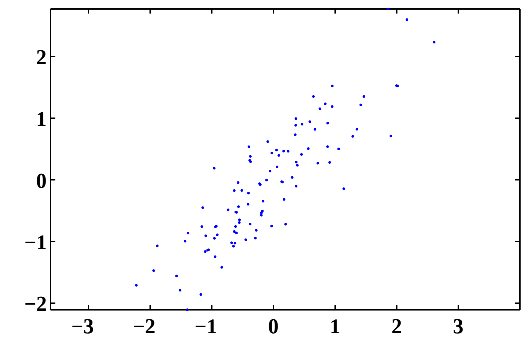

FIGURE 14.32: Scatter diagram of data set with $\sigma_1 = \sigma_2 = 1$, and $\rho = 0.9$.

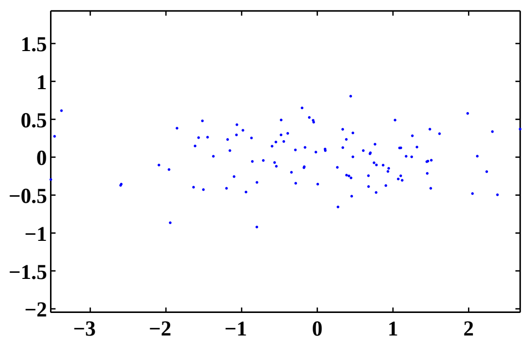

FIGURE 14.33: Scatter diagram of data set Karhunen-Loève Transform $\boldsymbol{c}[n]$.

```
lambda=diag(L);
Mx=repmat(mean(X),N,1);
C=(X-Mx)*A;
m = 1;
Xhat=C(:,1:m)*A(:,1:m)'+Mx;
E=X(:)-Xhat(:);
MSE1=sum(E.^2)/N
MSE2=sum(lambda(m+1:p))
%% Plot
hfa = figconfg('P1420a','small');
plot(X(:,1),X(:,2),'.'); axis equal
hfb = figconfg('P1420b','small');
plot(C(:,1),C(:,2),'.'); axis equal
```

Basic Problems

23. (a) Proof:

$$
\begin{aligned}
E[\hat{\sigma}^2] &= E\left[\frac{1}{N}\sum_{k=1}^{N}(x_k-\hat{\mu})^2\right] = \frac{1}{N}\sum_{k=1}^{N}E[(x_k-\hat{\mu})^2] \\
&= \frac{1}{N}\sum_{k=1}^{N}\left(E[x_k^2]-2E[x_k\hat{\mu}]+E[\hat{\mu}^2]\right) \\
&= \frac{1}{N}\sum_{k=1}^{N}\left(\sigma^2+\mu^2-2\times\frac{1}{N}(N\mu^2+\sigma^2)+\frac{1}{N^2}(N^2\mu^2+N\sigma^2)\right) \\
&= \frac{1}{N}\sum_{k=1}^{N}\left(\frac{N-1}{N}\sigma^2\right) = \frac{N-1}{N}\sigma^2
\end{aligned}
$$

(b) Proof:

$$
\begin{aligned}
E[\hat{\sigma}^2] &= \frac{1}{N}\sum_{k=1}^{N}E[(x_k-\mu)^2] = \frac{1}{N}\sum_{k=1}^{N}\left(E[x_k^2]-2\mu E[x_k]+\mu^2\right) \\
&= \frac{1}{N}\sum_{k=1}^{N}\left(\mu^2+\sigma^2-2\mu^2+\mu^2\right) \\
&= \sigma^2
\end{aligned}
$$

24. (a) Comments:
See script output for details.

(b) See plot below.

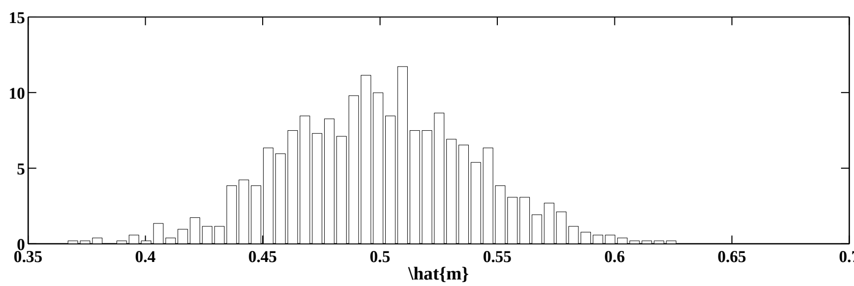

FIGURE 14.34: Plot of empirical pdf of the sample mean.

MATLAB script:

```
% P1424: Uniform Distribution: Mean and Variance of sample mean
close all; clc
K = 1e3; N = 50;
X = rand(N,K);
mhatk = mean(X,1);
mu_m = mean(mhatk), var_m = var(mhatk),
mu_true = 0.5, var_true = 1/12/N,
[mo pm] = epdf(mhatk,50);
%% Plot
hfa = figconfg('P1424a','long');
bar(mo,pm,'facecolor','w');
xlabel('\hat{m}','fontsize',LFS)
```

25. Proof:

$$
\begin{aligned}
\sum_{k=1}^{N}(x_k-\hat{\mu})^2 &= \sum_{k=1}^{N}(x_k-\mu+\mu-\hat{\mu})^2 \\
&= \sum_{k=1}^{N}\left[(x_k-\mu)^2+2(x_k-\mu)(\mu-\hat{\mu})+(\mu-\hat{\mu})^2\right] \\
&= \sum_{k=1}^{N}(x_k-\mu)^2+N(\mu-\hat{\mu})^2+2(\mu-\hat{\mu})\sum_{k=1}^{N}(x_k-\mu) \\
&= \sum_{k=1}^{N}(x_k-\mu)^2+N(\mu-\hat{\mu})^2-2N(\mu-\hat{\mu})^2 \\
&= \sum_{k=1}^{N}(x_k-\mu)^2-N(\mu-\hat{\mu})^2
\end{aligned}
$$

26. (a) Solution:
    See script output.

    (b) Solution:
    See script output.

    (c) See plot below.

    (d) tba.

MATLAB script:

```
% P1426: Sample covariance performance
```

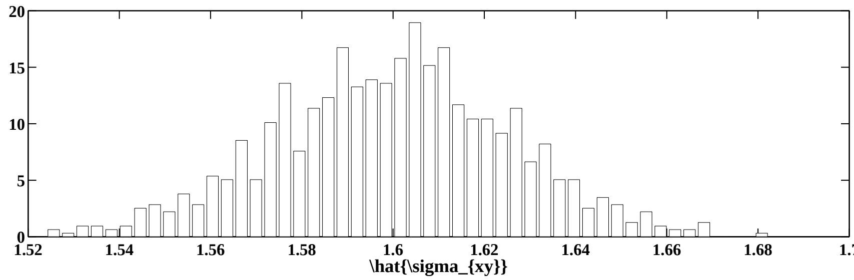

FIGURE 14.35: Plot of histogram of the estimate $\hat{\sigma}_{xy}$.

```
close all; clc
N = 1e4; K = 1e3;
X = randn(N,2,K);
mx = 1; my = 2; sigx2 = 4; sigy2 = 1; rho = 0.8;
C = [sigx2 sqrt(sigx2*sigy2)*rho;sqrt(sigx2*sigy2)*rho sigy2];
R = chol(C);
for ii = 1:K
X(:,:,ii) = X(:,:,ii)*R;
end
X(:,1,:) = X(:,1,:) + mx; X(:,2,:) = X(:,2,:) + my;
sighatxy = zeros(1,K);
for ii = 1:K
sighatxy(ii) = mean((X(:,1,ii)-mean(X(:,1,ii))).*(X(:,2,ii)-mean(X(:,2,ii))));
end
mu_sig2 = mean(sighatxy), var_sig2 = var(sighatxy),
[mo pm] = epdf(sighatxy,50);
%% Plot
hfa = figconfg('P1426a','long');
bar(mo,pm,'facecolor','w');
xlabel('\hat{\sigma_{xy}}','fontsize',LFS)
```

27. (a) See plot below.

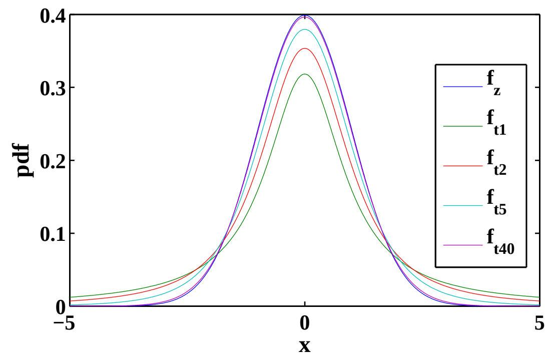

FIGURE 14.36: Plot of pdf of t_ν for $\nu = 1, 2, 5, 40$ and the pdf of z.

(b) See plot below.

MATLAB script:

```
% P1427: Student t distribution
close all; clc
%% Part a
nu = [1 2 5 40];
x = linspace(-5,5,1001);
fz = normpdf(x,0,1);
ft = zeros(length(nu),length(x));
for ii = 1:length(nu)
    ft(ii,:) = tpdf(x,nu(ii));
end
hfa = figconfg('P1427a','small');
plot(x,fz,x,ft)
xlabel('x','fontsize',LFS)
ylabel('pdf','fontsize',LFS)
legend('f_z','f_{t1}','f_{t2}','f_{t5}','f_{t40}','location','best')
%% Part b
N = 1000;
z = randn(1,N);
[zo,pz] = epdf(z,50);
hfb = figconfg('P1427b','long');
bar(zo,pz,'facecolor','w'); hold on
plot(x,fz,'linewidth',2)
```

```
legend('Empirical pdf','Theoretical pdf','location','best')
FL = 'cdef';
for ii = 1:length(nu)
    t = trnd(nu(ii),1,N);
    [to,pt] = epdf(t,50);
    figconfg(['P1427',FL(ii)],'long')
    bar(to,pt,'facecolor','w'); hold on
    plot(x,ft(ii,:),'linewidth',2)
    title(['\nu = ',num2str(nu(ii))],'fontsize',TFS)
    legend('Empirical pdf','Theoretical pdf','location','best')
end
```

28. See plots below.

MATLAB script:

```
% P1428: Simulation for the quality of unbiased mean estimator
close all; clc
nu = 100; N = 40; K = 1000;
% nu = 1; N = 40; K = 1000; % part ii
%% Part a
x = trnd(nu,1,K);
[xo px] = epdf(x,50);
xx = linspace(1.1*min(xo),1.1*max(xo),1000);
fxx = tpdf(xx,nu);
hfa = figconfg('P1428a','long');
bar(xo,px,'facecolor','w'); hold on
plot(xx,fxx,'r','linewidth',2)
xlabel('x','fontsize',LFS)
legend('Empirical pdf','Theoretical pdf','location','best')
%% Part b
X = trnd(nu,N,K);
%% Part c
mhatk = mean(X,1);
hfb = figconfg('P1428b','small');
plot(1:K,mhatk,'.b',1:K,zeros(1,K),'--r')
xlabel('k','fontsize',LFS)
legend('sample mean m_k','true mean','location','best')
%% Part d
mu_m = mean(mhatk), mu_true = 0,
var_m = var(mhatk), if nu > 2, var_true = nu/(nu-2), end
```

```
%% Part e
[mo pm] = epdf(mhatk,50);
mm = linspace(1.1*min(mo),1.1*max(mo),1000);
fmm = normpdf(mm,0,1/sqrt(N));
hfc = figconfg('P1428c','long');
bar(mo,pm,'facecolor','w'); hold on
plot(mm,fmm,'r','linewidth',2)
xlabel('m','fontsize',LFS)
legend('Empirical sample mean pdf',...
    'Theoretical sample mean pdf','location','best')
```

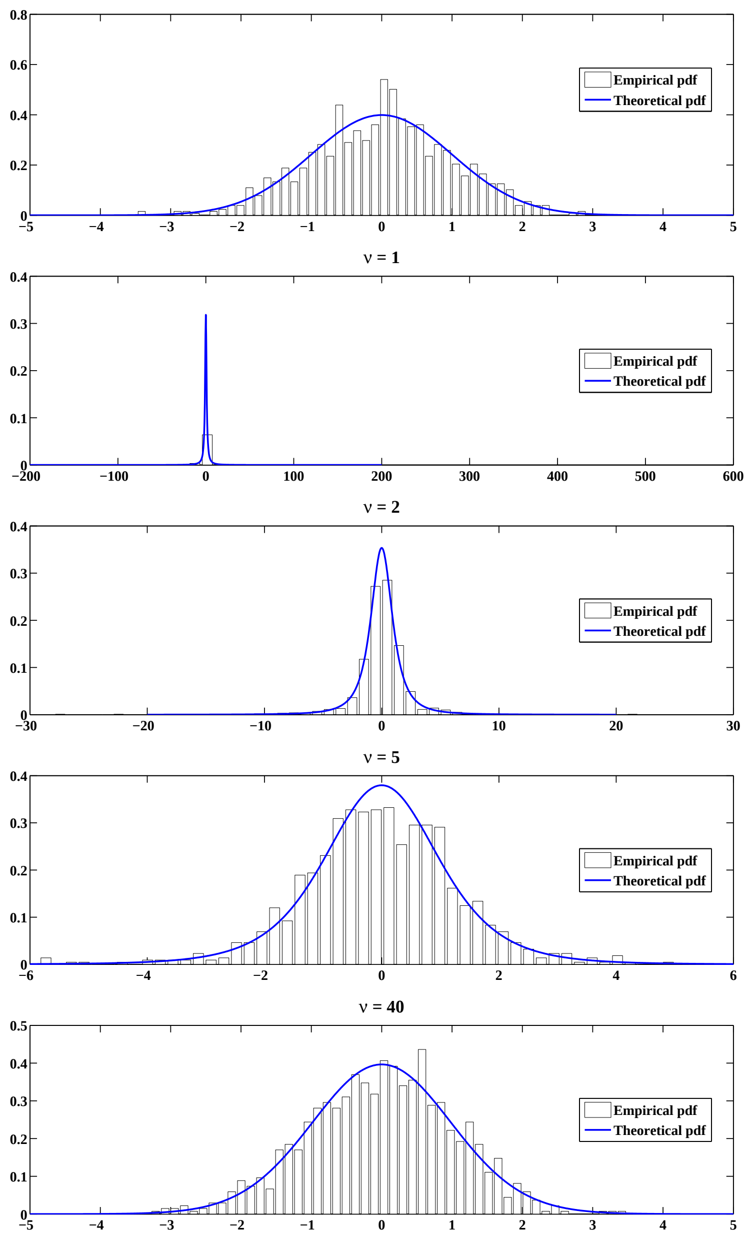

FIGURE 14.37: Comparisons between empirical pdf and the theoretical pdf.

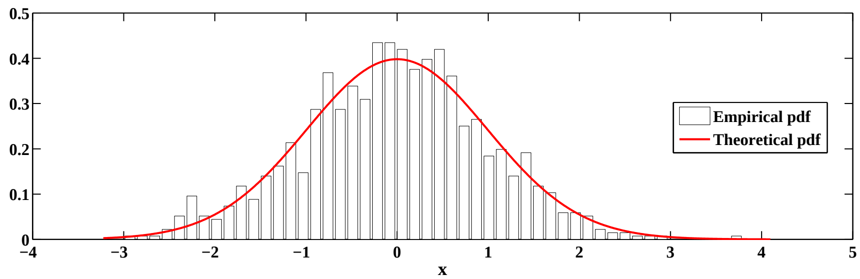

FIGURE 14.38: Empirical and theoretical pdfs of random variable $x_i \sim t_\nu$ for $\nu = 100$.

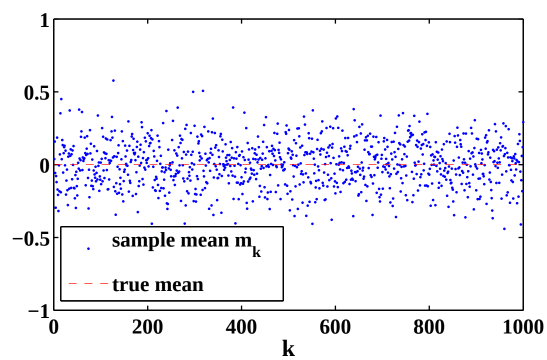

FIGURE 14.39: Plot of the true mean μ and the K sample means.

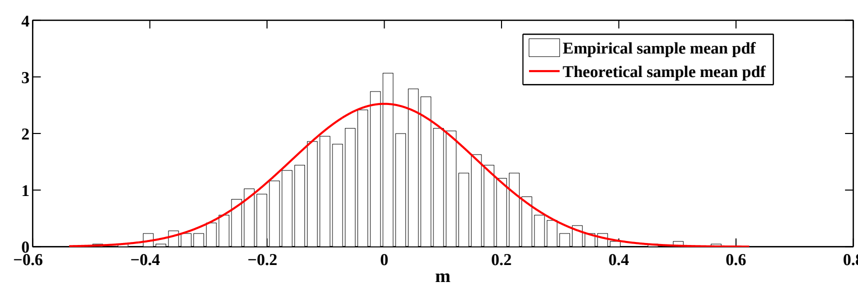

FIGURE 14.40: Empirical and theoretical pdfs of the sample mean.

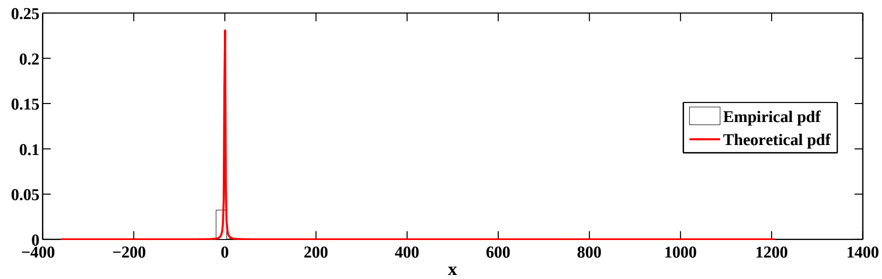

FIGURE 14.41: Empirical and theoretical pdfs of random variable $x_i \sim t_\nu$ for $\nu = 1$.

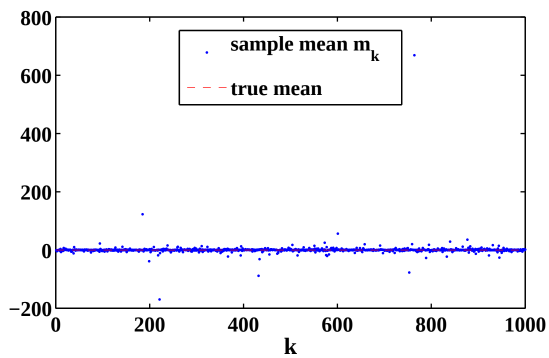

FIGURE 14.42: Plot of the true mean μ and the K sample means.

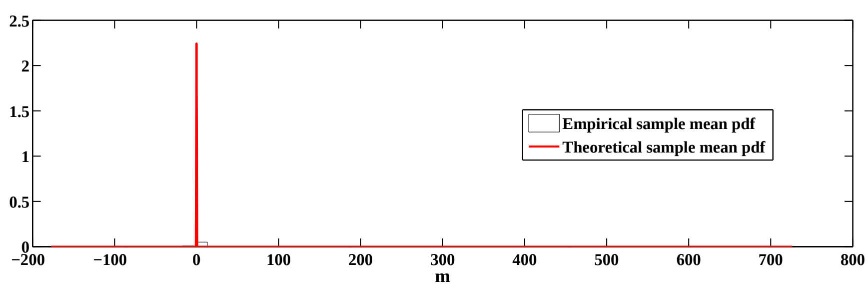

FIGURE 14.43: Empirical and theoretical pdfs of the sample mean.

29. (a) See plot below.

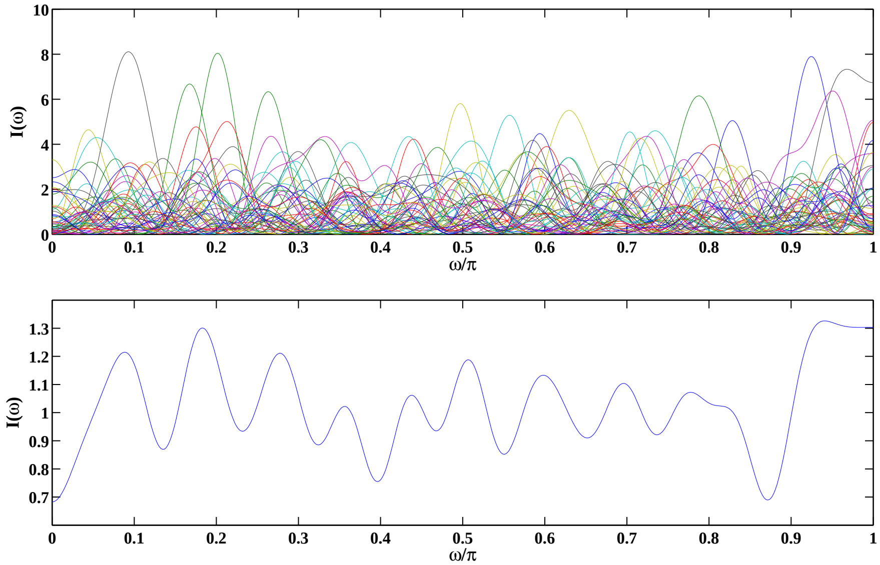

FIGURE 14.44: Plot of 50 periodogram estimates and the average of these overlays for $N = 32$.

(b) See plot below.

(c) See plot below.

(d) tba.

MATLAB script:

```
% P1429: Investigate PSD estimate
close all; clc
K = 50;
N = 32;
% N = 128;
% N = 256;
xn = randn(N,K);
I = zeros(1024,K);
for ii = 1:K
    I(:,ii) = psdper(xn(:,ii),1024);
end
I = I(1:513,:);
```

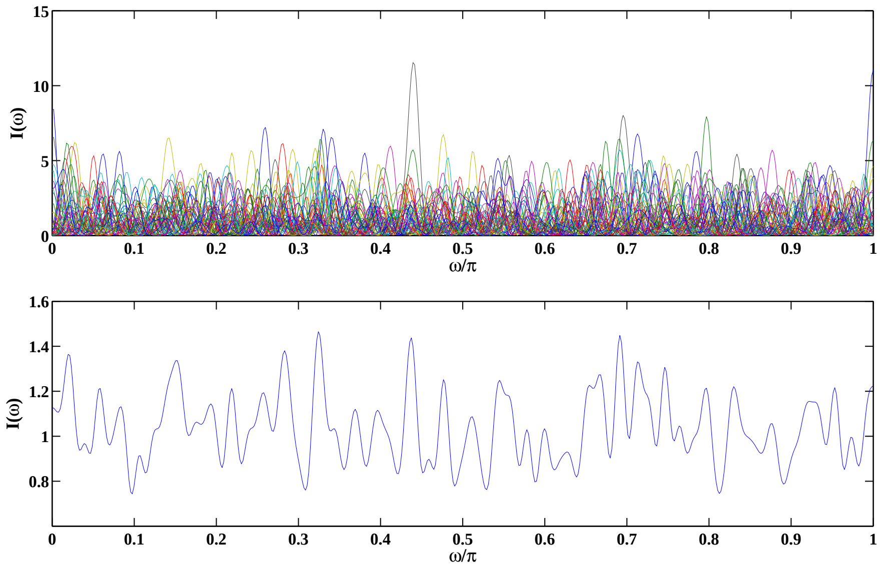

FIGURE 14.45: Plot of 50 periodogram estimates and the average of these overlays for $N = 128$.

```
Im = mean(I,2);
w = linspace(0,1,513)*pi;
%% Part a: Periodogram
hfa = figconfg('P1429a','long');
plot(w/pi,I)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('I(\omega)','fontsize',LFS)
hfb = figconfg('P1429a','long');
plot(w/pi,Im)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('I(\omega)','fontsize',LFS)
```

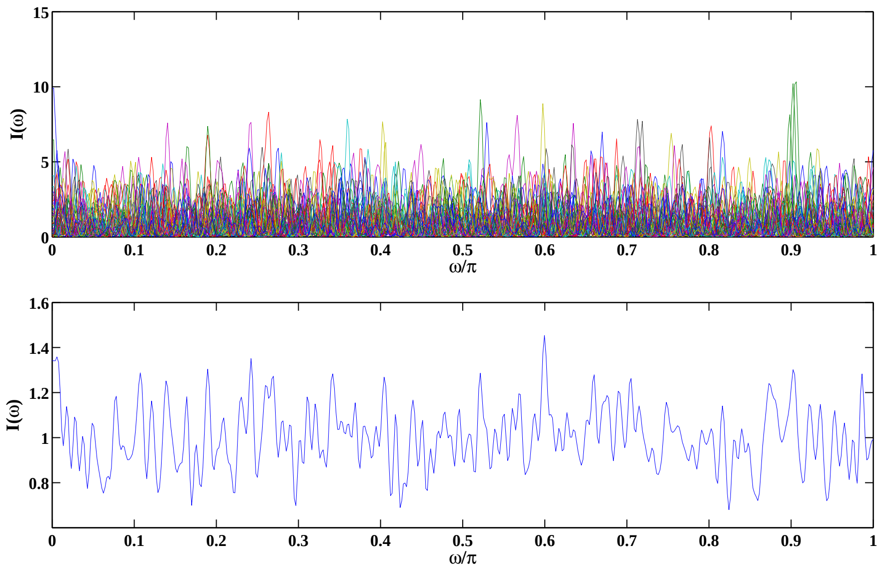

FIGURE 14.46: Plot of 50 periodogram estimates and the average of these overlays for $N = 256$.

30. (a) Comments: tba.

(b) See plot below.

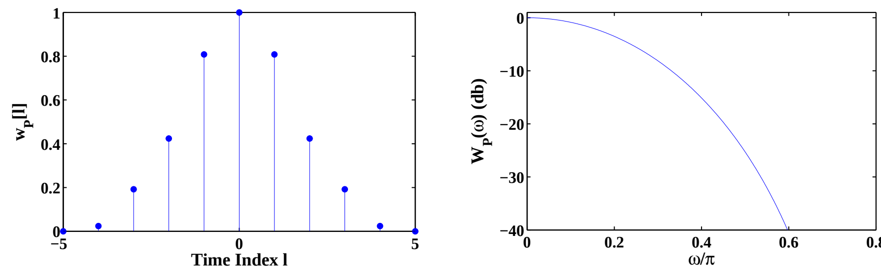

FIGURE 14.47: Plot of the data window $w_P[\ell]$ and its frequency-domain response $W_P(\omega)$ for $L = 5$.

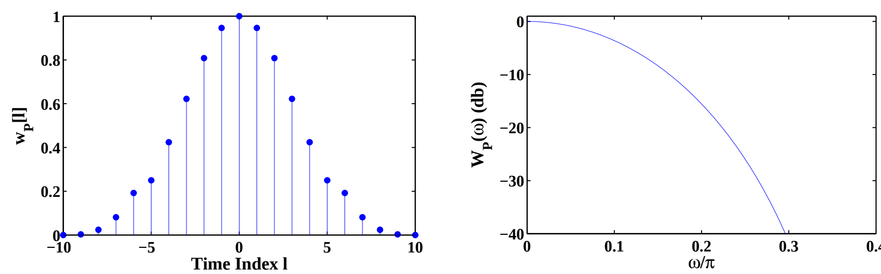

FIGURE 14.48: Plot of the data window $w_P[\ell]$ and its frequency-domain response $W_P(\omega)$ for $L = 10$.

(c) Solution:
See script output for detail.

MATLAB script:

```
% P1430: Parzen window investigation
close all; clc
L = 5;
% L = 10;
% L = 20;
l = -L:L;
wp = zeros(size(l));
ind = (abs(l)<=L/2);
wp(ind) = 1 - 6*(abs(l(ind))/L).^2 + 6*(abs(l(ind))/L).^3;
```

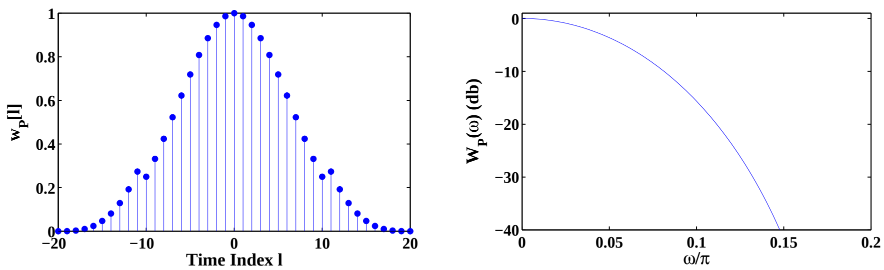

FIGURE 14.49: Plot of the data window $w_P[\ell]$ and its frequency-domain response $W_P(\omega)$ for $L = 20$.

```
ind = (abs(l)<=L & abs(l)>L/2);
wp(ind) = 3*(1-abs(l(ind))/L).^3;
w = linspace(0,1,501)*pi;
Wp = 3*L/4*(sin(w*L/4)./sin(w/4)).^4;
Wp(1) = 3*L^5/4;
Wpdb = 20*log10(Wp./max(Wp));
ind = find(Wpdb<-3,1,'first');
Deltaw = 2*w(ind-1)/pi;
%% Plot
hfa = figconfg('P1430a','small');
stem(l,wp,'filled')
xlabel('Time Index l','fontsize',LFS)
ylabel('w_P[l]','fontsize',LFS)
hfb = figconfg('P1430b','small');
plot(w/pi,Wpdb); ylim([-40 1])
xlabel('\omega/\pi','fontsize',LFS)
ylabel('W_P(\omega) (db)','fontsize',LFS)
```

31. Solution:
The mean value of $\hat{\theta}$ is:

$$E[\hat{\theta}] = E[a_1\hat{\theta}_1 + a_2\hat{\theta}_2] = a_1E[\hat{\theta}_1] + a_2E[\hat{\theta}_2] = (a_1 + a_2)\theta$$

In order to constrain that $\hat{\theta}$ is unbiased, we need

$$a_1 + a_2 = 1$$

Hence, the estimate $\hat{\theta}$ can be written as

$$\hat{\theta} = a_1\hat{\theta}_1 + a_2\hat{\theta}_2$$

The variance of $\hat{\theta}$ is

$$
\begin{aligned}
E[(\hat{\theta}-\theta)^2] &= E[(a_1\hat{\theta}_1 - a_1\theta + (1-a_1)\hat{\theta}_2 - (1-a_1)\theta)^2] \\
&= a_1^2 E[(\hat{\theta}_1-\theta)^2] + (1-a_1)^2 E[(\hat{\theta}_2-\theta)^2] + 2a_1(1-a_1)E[\hat{\theta}_1-\theta]E[\hat{\theta}_2-\theta] \\
&= a_1^2\sigma_1^2 + (1-a_1^2)\sigma_2^2
\end{aligned}
$$

In order to achieve the minimum variance goal, we conclude the minimum variance J is

$$
J = \begin{cases} \sigma_2^2, & \text{if } \sigma_1^2 > \sigma_2^2, \quad a_1 = 0,\ a_2 = 1 \\ \sigma_1^2, & \text{if } \sigma_1^2 < \sigma_2^2, \quad a_1 = 1,\ a_2 = 0 \\ \sigma_1^2, & \text{if } \sigma_1^2 = \sigma_2^2, \quad\ \ a_1 + a_2 = 1 \end{cases}
$$

32. Solution:
The mean of the linear estimator can be written as

$$
E[\hat{\theta}] = E[\boldsymbol{a}^T\boldsymbol{x}] = \boldsymbol{a}^T E[\boldsymbol{x}] = \theta\sum_{k=1}^{N} a_k
$$

Since the estimator is unbiased, we constrain that

$$
\theta\sum_{k=1}^{N} a_k = \theta
$$

which requires that

$$
\sum_{k=1}^{N} a_k = 1
$$

Using the constraint above and formulate the Lagrange equation as

$$
J = E[(\sum_{k=1}^{N} a_k x_k)^2] - \theta^2 + \lambda(\sum_{k=1}^{N} a_k x_k - 1) = 0
$$

Take the partial derivative with respect x_k, we have

$$
\frac{\partial J}{\partial a_k} = E[2(a_k x_k)a_k] + \lambda = 2\theta a_k^2 + \lambda = 0
$$

Solve for the above equations, we conclude that

$$
a_k = \frac{\lambda}{2\theta}, \quad \lambda = \frac{2\theta}{N}, \quad a_k = \frac{1}{N}
$$

The minimum variance is

$$
J_{\min} = \frac{\sum_{k=1}^{N}\sigma_k^2}{N^2}
$$

33. Solution:
The variance of the estimator is

$$J = E[(X - \hat{X})^2] = E[(X - aY - b)^2] = E[X^2] + E[(aY + b)^2] - 2E[X(aY + b)]$$
$$= (\mu_x^2 + \sigma_x^2) + a^2(\mu_y^2 + \sigma_y^2) + 2ab\mu_y + b^2 - 2a(\sigma_{xy} + \mu_x\mu_y) - 2b\mu_x$$

Take the partial derivatives with respect to a and b and constrain the results equal to zero, we have

$$\frac{\partial J}{\partial a} = 2a(\mu_y^2 + \sigma_y^2) + 2b\mu_y - 2\sigma_{xy} - 2\mu_x\mu_y = 0 \qquad \text{(P33a)}$$

$$\frac{\partial J}{\partial b} = 2a\mu_y + 2b - 2\mu_x = 0 \qquad \text{(P33b)}$$

Solve Eq. (P33b), we have

$$b = \mu_x - a\mu_y$$

Substitute b back to Eq.(P33a) and solve for a, we have

$$a = \frac{\sigma_{xy}}{\sigma_y^2}$$

34. See plots below.

MATLAB script:

```
% P1434: Matched filtering
close all; clc
N = 200; n = 0:N-1;
sn = zeros(size(n));
%% Part b
p = 9;
ind = (n >= 0 & n <= p-1);
sn(ind) = 1/3;
%% Part c
% p = 100;
% ind = (n >= 0 & n <= p-1);
% sn(ind) = 1/10;

randn('seed',0)
xn = sn + randn(1,N);
```

```
yn = filter(sn(p:-1:1),1,xn);
%% Part a:
hfa = figconfg('P1434a','small');
plot(n,sn)
title('Signal','fontsize',LFS)
hfb = figconfg('P1434b','small');
plot(n,xn)
title('Signal+Noise','fontsize',LFS)
hfc = figconfg('P1434c','small');
plot(n,yn)
title('Matched Filter Output','fontsize',LFS)
xlabel('n','fontsize',LFS)
```

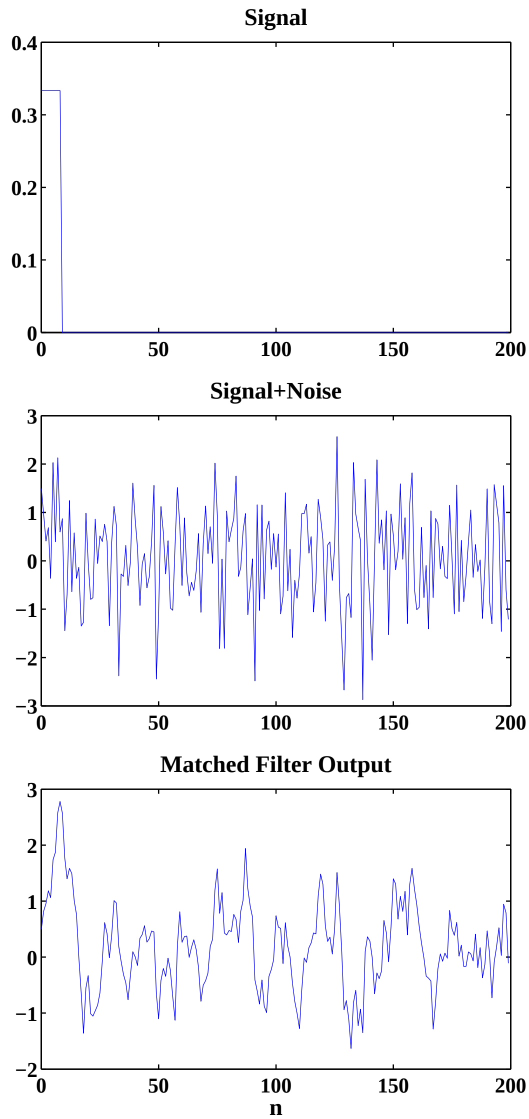

FIGURE 14.50: Plots of $s_i[n] = 1/3$, $p = 9$, noisy signal $x[n]$ and matched filter output.

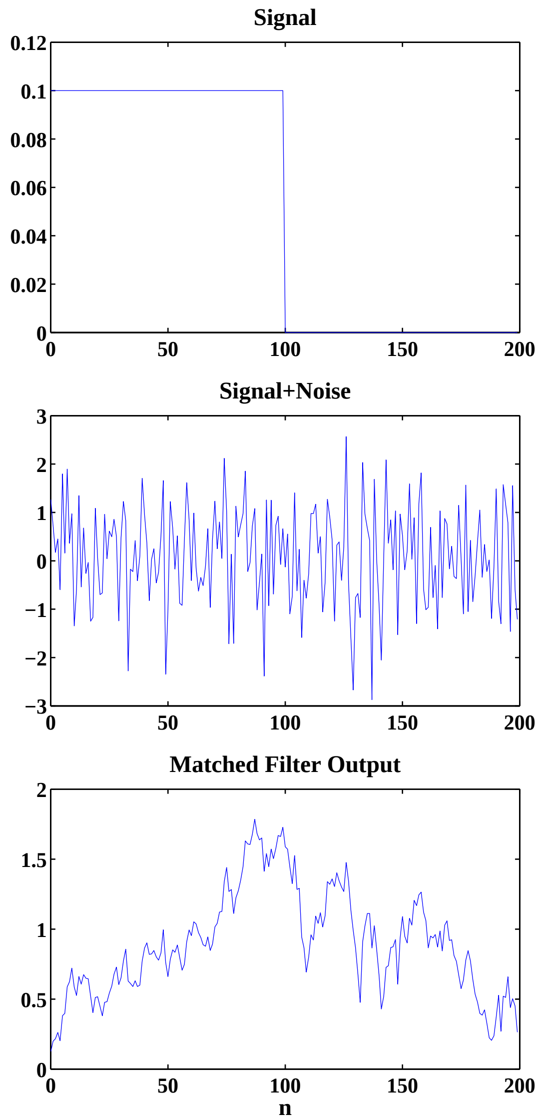

FIGURE 14.51: Plots of $s_i[n] = 1/3$, $p = 100$, noisy signal $x[n]$ and matched filter output.

35. (a) See plot below.

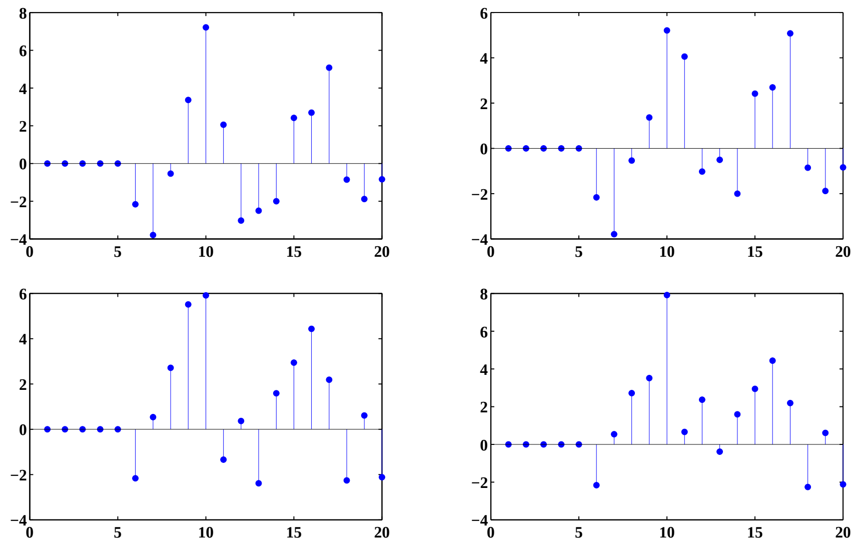

FIGURE 14.52: Plots of responses of $h_0[n]$ to $x_0[n]$ and $x_1[n]$ and $h_1[n]$ to $x_0[n]$ and $x_1[n]$.

(b) See plot below.

(c) Comments:

$$x_0[n] = \{-1, 0, 1, -1, -1, 0, 0, 0, 0, 0\}$$

MATLAB script:

```
% P1435:
close all; clc
N = 20; n = 1:N;
x0 = [1 1 1 -1 -1 0 0 0 0 0];
x1 = [1 1 1 1 -1 0 0 0 0 0];
z0 = zeros(1,N); z0(1:10) = x0;
z1 = zeros(1,N); z1(1:10) = x1;
randn('seed',0)
vn = randn(1,N);
%% Part a:
y00 = filter(x0(end:-1:1),1,z0+vn);
y01 = filter(x1(end:-1:1),1,z0+vn);
```

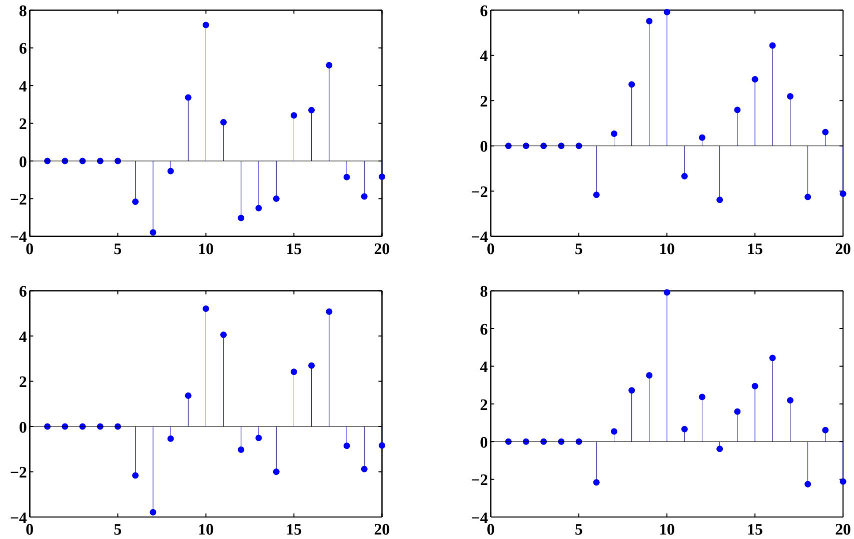

FIGURE 14.53: Plots of correlation sequences of impulse $h_0[n]$ to inputs $x_0[n]$ and $x_1[n]$ and impulse $h_1[n]$ to inputs $x_0[n]$ and $x_1[n]$.

```
y10 = filter(x0(end:-1:1),1,z1+vn);
y11 = filter(x1(end:-1:1),1,z1+vn);
%% Part b:
% y00 = conv(x0(end:-1:1),z0+vn);
% y00 = y00(1:N);
% y01 = conv(x1(end:-1:1),z0+vn);
% y01 = y01(1:N);
% y10 = conv(x0(end:-1:1),z1+vn);
% y10 = y10(1:N);
% y11 = conv(x1(end:-1:1),z1+vn);
% y11 = y11(1:N);
%% Plot
hfa = figconfg('P1435a','small');
stem(n,y00,'filled');
hfb = figconfg('P1435b','small');
stem(n,y01,'filled');
hfc = figconfg('P1435c','small');
stem(n,y10,'filled');
hfd = figconfg('P1435d','small');
stem(n,y11,'filled');
```

36. (a) See plot below.

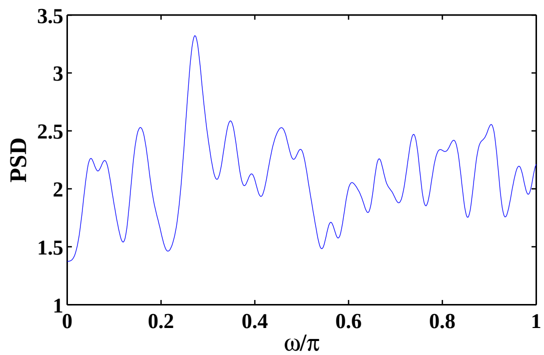

FIGURE 14.54: Plot of Welch PSD estimate of $v[n]$ at 512 frequency values over $[0, \pi]$.

(b) See plot below.

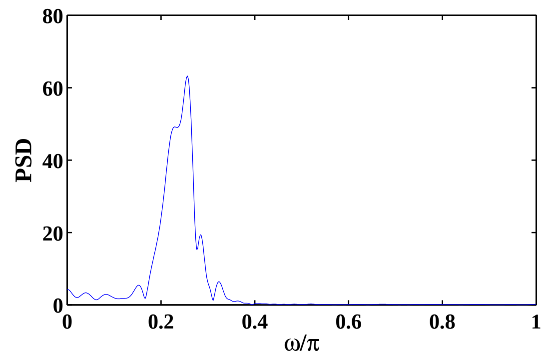

FIGURE 14.55: Plot of Welch cross-PSD estimate between $v[n]$ and $x[n]$ at 512 frequency values over $[0, \pi]$.

(c) See plot below.

MATLAB script:

```
% P1436: Welch PSD estimate and Welch cross PSD estimate
close all; clc
N = 2048;
vn = randn(1,N);
xn = filter(1,[1 -2.7607 3.8106 -2.6535 0.9238],vn);
```

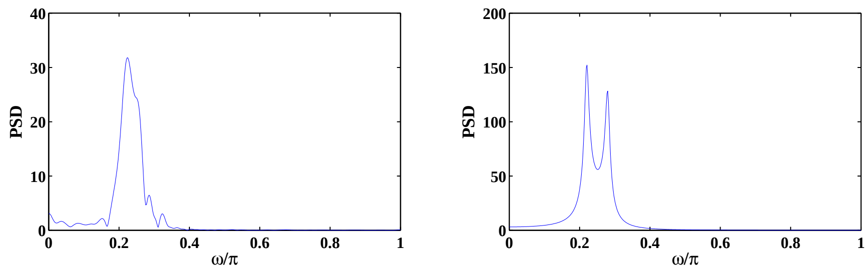

FIGURE 14.56: Plot of estimate and theoretical frequency response $H(e^{j\omega})$.

```
%% Part a
L = 100;
Pxx = psdwelch(vn',L,1024);
Pxy = psdwelchxy(vn',xn',L,1024);
w = linspace(0,1,512)*pi;
H = Pxy./Pxx;
H_ref = freqz(1,[1 -2.7607 3.8106 -2.6535 0.9238],w);
%% Plot
hfa = figconfg('P1436a','small');
plot(w/pi,Pxx)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD','fontsize',LFS)
hfb = figconfg('P1436b','small');
plot(w/pi,abs(Pxy))
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD','fontsize',LFS)
hfc = figconfg('P1436c','small');
plot(w/pi,abs(H))
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD','fontsize',LFS)
hfd = figconfg('P1436d','small');
plot(w/pi,abs(H_ref))
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD','fontsize',LFS)
```

37. (a) See plot below.
    (b) See plot below.
    (c) tba.

    MATLAB script:

```
% P1437: Matched filtering
close all; clc
N = 200; n = 0:N-1;
sn = zeros(size(n));
%% Part b
p = 9;
ind = (n >= 0 & n <= p-1);
sn(ind) = 1/3;
%% Part c
% p = 100;
% ind = (n >= 0 & n <= p-1);
% sn(ind) = 1/10;

randn('seed',0)
xn = sn + randn(1,N);
yn = filter(sn(p:-1:1),1,xn);
%% Part a:
hfa = figconfg('P1434a','small');
plot(n,sn)
ylim([-0.01 max(sn)*1.1])
title('Signal','fontsize',LFS)
hfb = figconfg('P1434b','small');
plot(n,xn)
title('Signal+Noise','fontsize',LFS)
hfc = figconfg('P1434c','small');
plot(n,yn)
title('Matched Filter Output','fontsize',LFS)
xlabel('n','fontsize',LFS)
```

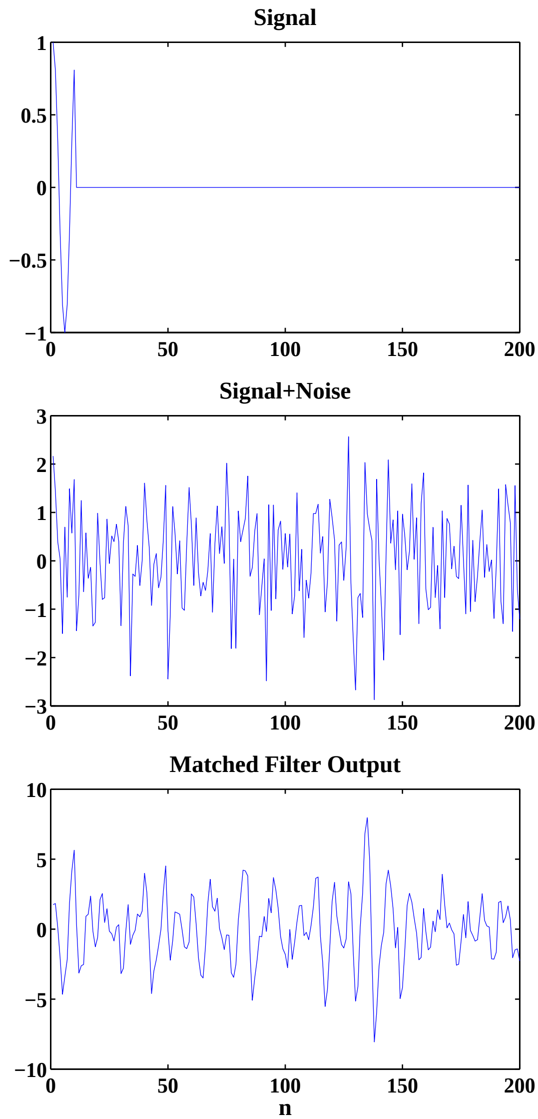

FIGURE 14.57: Plots of original signal $s_1[n]$, noisy signal $x_1[n]$ and matched filter output.

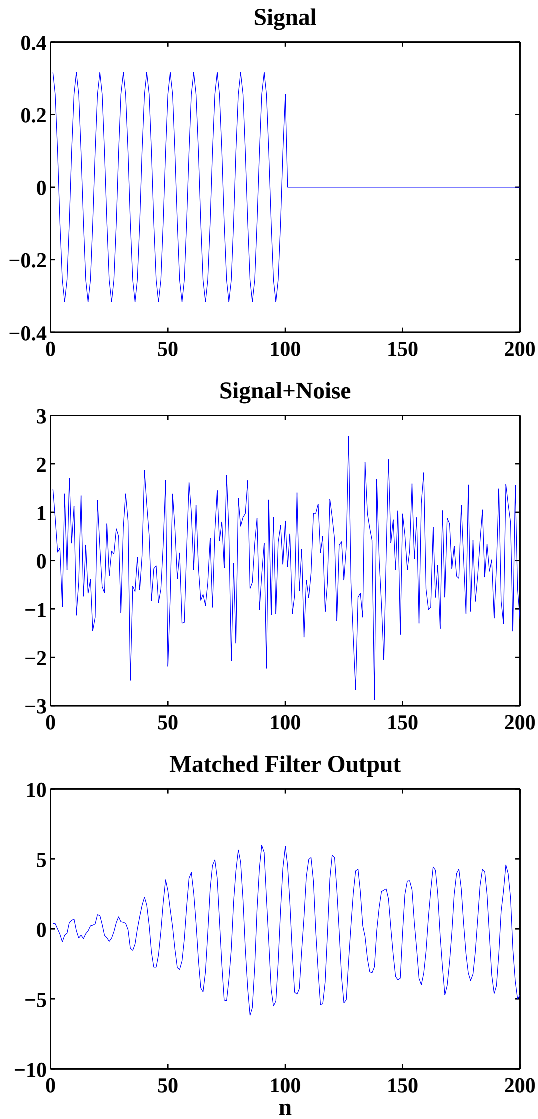

FIGURE 14.58: Plots of original signal $s_2[n]$, noisy signal $x_2[n]$ and matched filter output.

38. (a) Proof:

Since we have

$$e_m^b[n-1] \triangleq \boldsymbol{b}_m^T \boldsymbol{x}_m[n-1] + x[n-1-m]$$

By applying the orthogonality principle, we have

$$\begin{aligned} E[e_m^b[n-1]\boldsymbol{x}_m^T[n-1]] &= E[\boldsymbol{b}_m^T \boldsymbol{x}_m[n-1]\boldsymbol{x}_m^T[n-1]] + E[x[n-1-m]\boldsymbol{x}_m^T[n-1]] \\ &= \boldsymbol{b}_m^T \boldsymbol{R}_m + \boldsymbol{r}_m^T \boldsymbol{J}_m = \boldsymbol{0} \end{aligned}$$

which equals that

$$\boldsymbol{b}_m^T \boldsymbol{R}_m = -\boldsymbol{J}_m \boldsymbol{r}_m$$

(b) Proof:

$$\begin{aligned} J_m^b &= E[e_m^b[n-1]e_m^b[n-1]] \\ &= E[(\boldsymbol{b}_m^T \boldsymbol{x}_m[n-1] + x[n-1-m])(\boldsymbol{x}_m^T[n-1]\boldsymbol{b}_m + x[n-1-m])] \\ &= \boldsymbol{b}_m^T \boldsymbol{R}_m \boldsymbol{b}_m + r[0] + 2\boldsymbol{b}_m^T \boldsymbol{J}_m \boldsymbol{r}_m \\ &= \boldsymbol{b}_m^T(-\boldsymbol{J}_m \boldsymbol{r}_m) + r[0] + 2\boldsymbol{b}_m^T \boldsymbol{J}_m \boldsymbol{r}_m \\ &= r[0] + \boldsymbol{b}_m^T \boldsymbol{J}_m \boldsymbol{r}_m \end{aligned}$$

(c) Proof:

$$\begin{aligned} J_m^b &= r[0] + \boldsymbol{b}_m^T \boldsymbol{J}_m \boldsymbol{r}_m = r[0] - \boldsymbol{r}_m^T \boldsymbol{J}_m \boldsymbol{R}_m^{-1} \boldsymbol{J}_m \boldsymbol{r}_m \\ &= r[0] - \boldsymbol{r}_m^T \boldsymbol{R}_m^{-1} \boldsymbol{r}_m = J_m \end{aligned}$$

39. Proof:

$$J_{m+1} = r[0] + \boldsymbol{a}_{m+1}^T \boldsymbol{r}_{m+1}$$

where

$$\boldsymbol{a}_{m+1} = \begin{bmatrix} \boldsymbol{a}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_m \boldsymbol{a}_m \\ 1 \end{bmatrix} k_{m+1}$$

$$\boldsymbol{r}_{m+1} = \begin{bmatrix} \boldsymbol{r}_m \\ r(m+1) \end{bmatrix}$$

Hence, we have

$$J_{m+1} = (r[0] + \boldsymbol{a}_m^T \boldsymbol{r}_m) + [\boldsymbol{a}_m^T \boldsymbol{J}_m^T \boldsymbol{r}_m + r(m+1)]k_{m+1}$$

Since

$$J_m = r[0] + \boldsymbol{a}_m^T \boldsymbol{r}_m$$

$$\beta_{m+1} = \boldsymbol{r}_m^T \boldsymbol{J}_m \boldsymbol{a}_m + r[m+1] = \boldsymbol{a}_m^T \boldsymbol{J}_m^T \boldsymbol{r}_m + r[m+1]$$

Thus, we proved that

$$J_{m+1} = J_m + \beta_{m+1} k_{m+1}$$

Since

$$k_{m+1} = \frac{\beta_{m+1}}{J_m}$$

we have

$$J_{m+1} = (1 - k_{m+1}^2) J_m$$

Assessment Problems

40. (a) Solution:
The mean of the output voltage random variable y is

$$\mu_y = E[y] = E[5x^2] = 2E[x^2] = 5(0+9) = 45$$

$$E[y^2] = E[25x^4] = 25E[x^4] = 25 \times 3 \times 9^2 = 6075$$

The variance of the output voltage random variable y is

$$\sigma_y^2 = E[y^2] - E^2[y] = 6075 - 45^2 = 4050$$

(b) MATLAB script:

```
% P1440: Square-Law full-wave diode
close all; clc
% N = 1e4;
% N = 1e5;
N = 1e6;
x = randn(1,N)*3;
y = 5*x.^2;
mu_y = mean(y),
var_y = var(y),
```

41. See plot below.

(a) Comments:
See script for details.

(b) See plot below.

(c) Comments:
See script output for comparison.

(d) See plot below.

MATLAB script:

```
% P1441: Simulation for the quality of unbiased mean estimator
close all; clc
sigma = 2; N = 40; K = 1000;
% sigma = sqrt(2); N = 20; K = 1000; % part d
%% Part a
X = randn(K,N)*sigma;
```

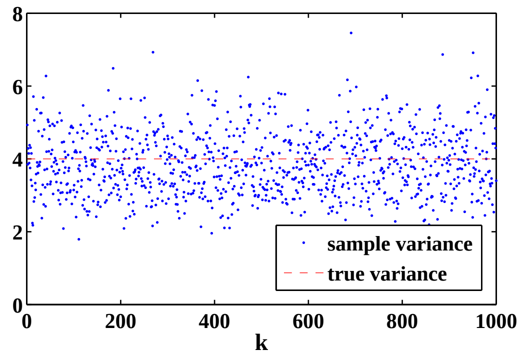

FIGURE 14.59: Plot of the true variance σ^2 and the K sample variances.

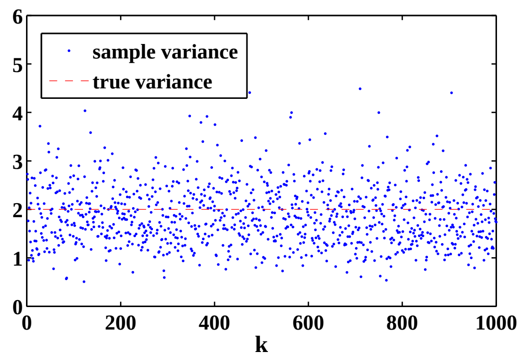

FIGURE 14.60: Plot of the true variance σ^2 and the K sample variances.

```
%% Part b
varhatk = var(X,1,2);
hfb = figconfg('P1441b','small');
plot(1:K,varhatk,'.b',1:K,ones(1,K)*sigma^2,'--r')
xlabel('k','fontsize',LFS)
legend('sample variance','true variance','location','best')
%% Part c
mu_var = mean(varhatk), mu_true = sigma^2,
var_var = var(varhatk), var_true = 2*(N-1)/N^2*sigma^4,
```

42. (a) See plot below.
    (b) See plot below.

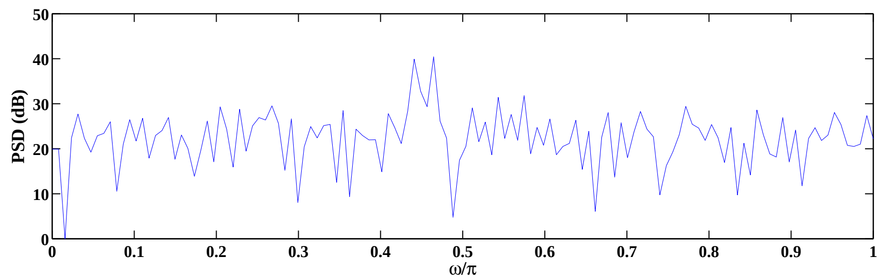

FIGURE 14.61: Plot of the estimated PSD using the periodogram.

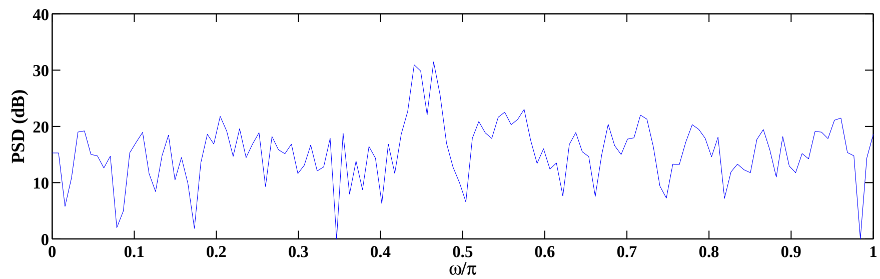

FIGURE 14.62: Plot of the estimated PSD using the modified periodogram with Bartlett window.

(c) See plot below.

(d) Comments:
The method in part (a) performs best in terms of signal resolution.

MATLAB script:

```
% P1442: Periodogram, Modified Periodogram and Blackman-Tukey method
close all; clc
N = 256; n = 0:N;
vn = randn(1,N+1)*sqrt(2); phi = rand(1,2)*2*pi-pi;
xn = cos(0.44*pi*n+phi(1)) + cos(0.46*pi*n+phi(2)) + vn;
w = linspace(0,1,N/2)*pi;
%% Part a: Periodogram
I1 = psdper(xn,N);
hfa = figconfg('P1442a','long');
I1db = 10*log10(I1(1:N/2)./min(I1(1:N/2)));
plot(w/pi,I1db)
xlabel('\omega/\pi','fontsize',LFS)
```

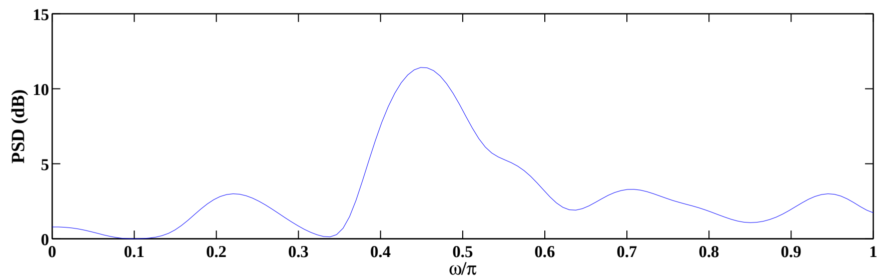

FIGURE 14.63: Plot of the estimated PSD using the Blackman-Tukey method with Parzen window.

```
ylabel('PSD (dB)','fontsize',LFS)
%% Part b: Modified Periodogram
I2 = psdmodper2(xn,N+1,bartlett(N+1));
hfb = figconfg('P1442b','long');
I2db = 10*log10(I2(1:N/2)./min(I2(1:N/2)));
plot(w/pi,I2db)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
%% Part c: Blackman-Tukey method with Parzen window
I3 = abs(psdbt(xn(:),32,N));
hfc = figconfg('P1442c','long');
I3db = 10*log10(I3(1:N/2)./min(I3(1:N/2)));
plot(w/pi,I3db)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
```

43. (a) See plot below.
    (b) See plot below.
    (c) tba.

    MATLAB script:

```
% P1443: Investigate PSD estimate using Blackman-Tukey method
close all; clc
A = [1 0.5 0.5 0.25]; om = [0.1 0.6 0.65 0.8]*pi;
phi = rand(1,4)*2*pi-pi;
N = 256; K = 50; n = (0:N-1)'; Nk = repmat(n,1,K);
V = randn(N,K);
```

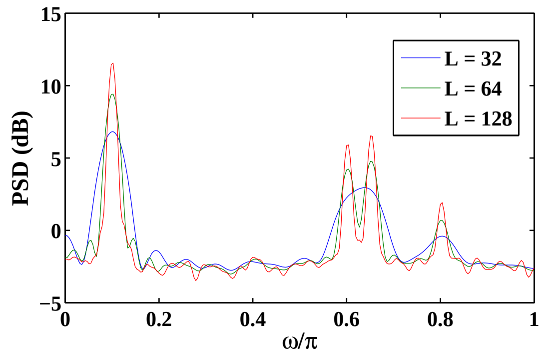

FIGURE 14.64: Plot of the Blackman-Tukey estimates for $L = 32$, 64, and 128 with Bartlett lag window.

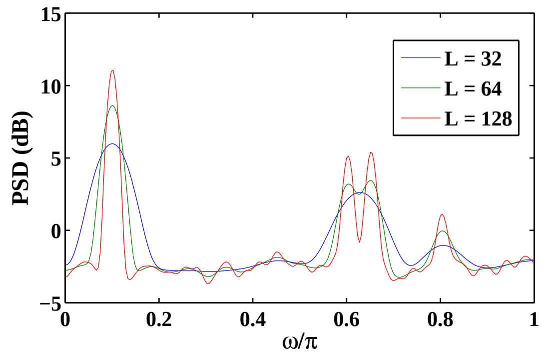

FIGURE 14.65: Plot of the Blackman-Tukey estimates for $L = 32$, 64, and 128 with Parzen window.

```
X = V;
for jj = 1:4
    X = X + A(jj)*sin(om(jj)*Nk+phi(jj));
end
L = [32 64 128]; Nfft = 512;
Pxx = zeros(K,Nfft/2,length(L));
for jj = 1:length(L)
    for ii = 1:K
        Pxx(ii,:,jj) = ...
            psdbt_bartlett(X(:,ii),L(jj),Nfft); % part a
%         Pxx(ii,:,jj) = ...
```

```
%               psdbt(X(:,ii),L(jj),Nfft); % part b
    end
end
P = squeeze(mean(Pxx,1));
Pdb = bsxfun(@rdivide,P,mean(P,1));
Pdb = 10*log10(Pdb);
w = linspace(0,1,Nfft/2)*pi;
%% Part a: Periodogram
hfa = figconfg('P1443a','small');
plot(w/pi,Pdb)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
legend(['L = ',num2str(L(1))],['L = ',num2str(L(2))],...
    ['L = ',num2str(L(3))],'location','best')
```

44. (a) See plot below.

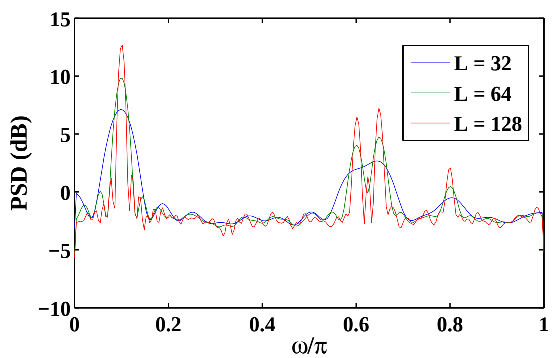

FIGURE 14.66: Plot of the Bartlett estimates for $L = 16$, 32, and 64.

(b) See plot below.

(c) tba.

MATLAB script:

```
% P1444: Investigate Bartlett and Welsh PSD estimate
close all; clc
A = [1 0.5 0.5 0.25]; om = [0.1 0.6 0.65 0.8]*pi;
phi = rand(1,4)*2*pi-pi;
N = 256; K = 50; n = (0:N-1)'; Nk = repmat(n,1,K);
V = randn(N,K);
```

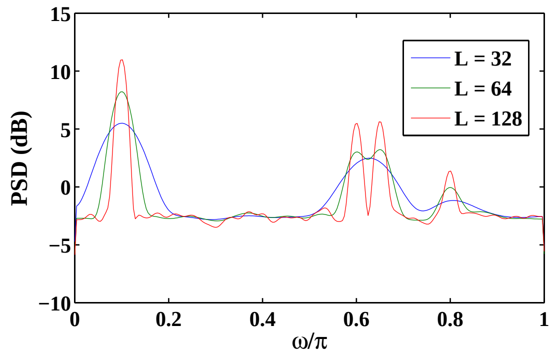

FIGURE 14.67: Plot of the Welch estimates, using 50 percent overlap, Hamming window for $L = 16$, 32, and 64.

```
X = V;
for jj = 1:4
    X = X + A(jj)*sin(om(jj)*Nk+phi(jj));
end
L = [32 64 128]; Nfft = 512;
Pxx = zeros(K,Nfft/2+1,length(L));
for jj = 1:length(L)
    for ii = 1:K
        [Pxx(ii,:,jj), w] = ...
            pwelch(X(:,ii),rectwin(L(jj)),0,Nfft); % part a
%           [Pxx(ii,:,jj), w] = ...
%               pwelch(X(:,ii),hann(L(jj)),fix(0.5*L(jj)),Nfft); % part b
    end
end
P = squeeze(mean(Pxx,1));
Pdb = bsxfun(@rdivide,P,mean(P,1));
Pdb = 10*log10(Pdb);
%% Part a: Periodogram
hfa = figconfg('P1444a','small');
plot(w/pi,Pdb)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
legend(['L = ',num2str(L(1))],['L = ',num2str(L(2))],...
    ['L = ',num2str(L(3))],'location','best')
```

45. (a) Proof:

$$E[(X-\hat{X}(Y))^2] = \iint (x-\hat{x}(y))^2 f_{X,Y}(x,y)dxdy$$
$$= \int f_Y(y)\left(\int (x-\hat{x}(y))^2 f_{X|Y}(x|y)dx\right)dy$$

Since $f_Y(y) \geq 0$, minimizing $E[(X-\hat{X})^2]$ equals minimizing the inner integral of the above equation, that is

$$\int (x-\hat{x}(y))^2 f_{X|Y}(x|y)dx$$

for every y. For a fixed y, the integral means minimizing the estimated error by a constant estimate, that is $\hat{x}(y) = c$. The optimal constant minimizer is the mean value, hence we can conclude that

$$\hat{X} = \mathrm{E}[X|Y]$$

(b) Proof:
Since for two jointly Gaussian random variables, uncorrelatedness equals independence. Thus, we can conclude that X is independent with any higher order of Y except linear relationship.
Suppose $\hat{X} = aY + b$, the estimation error variance is

$$J = E[(X-aY-b)^2] = E[X^2] - 2E[X(aY+b)] + E[(aY+b)^2]$$

Take the partial derivatives with respect to a and b and set the results equal to zero, we have

$$\frac{\partial J}{\partial a} = -2E[XY] + E[2Y(aY+b)] = 0$$
$$\frac{\partial J}{\partial b} = -2E[X] + E[2(aY+b)] = 0$$

Solve the equations above, we have

$$\begin{pmatrix} a = \frac{E[XY]-E[X]E[Y]}{\sigma_Y^2} \\ b = E[X] - aE[Y] \end{pmatrix}$$

(c) Proof:
Using the results from the previous question, if $E[X] = E[Y] = 0$, we have

$$b = E[X] - aE[Y] = 0$$

Hence, the minimum mse estimate is a linear function of Y.

46. tba

47. (a) See plot below.

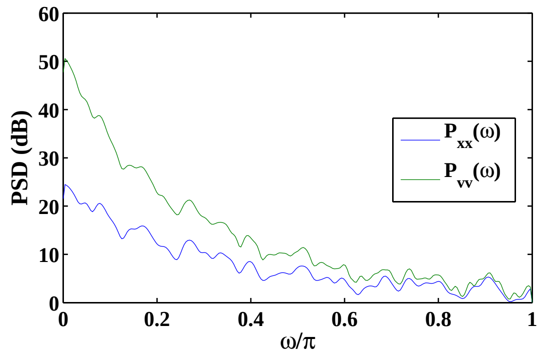

FIGURE 14.68: Plot of PSD estimates of $x[n]$ and $v[n]$ using Welch's method.

(b) See plot below.

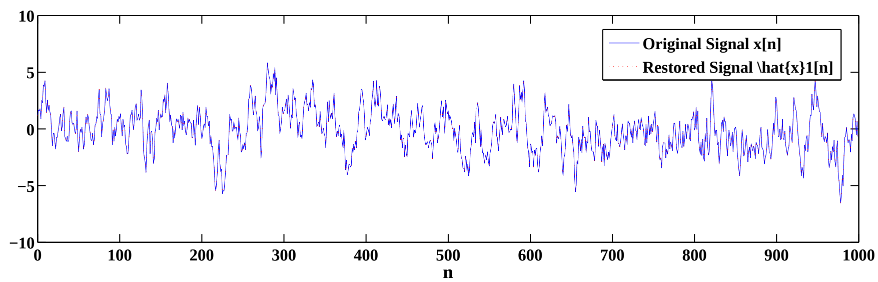

FIGURE 14.69: Plot of the original signal $x[n]$ and restored signal $\hat{x}_1[n]$.

(c) See plot below.

(d) See plot below.

MATLAB script:

```
% P1447:
close all; clc
N = 1e3; n = 0:N-1;
randn('seed',0)
wn = randn(1,N);
xn = filter(1,[1,-0.9],wn);
```

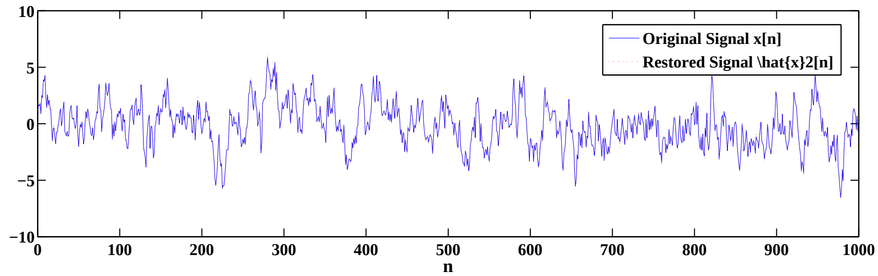

FIGURE 14.70: Plot of the original signal $x[n]$ and restored signal $\hat{x}_2[n]$.

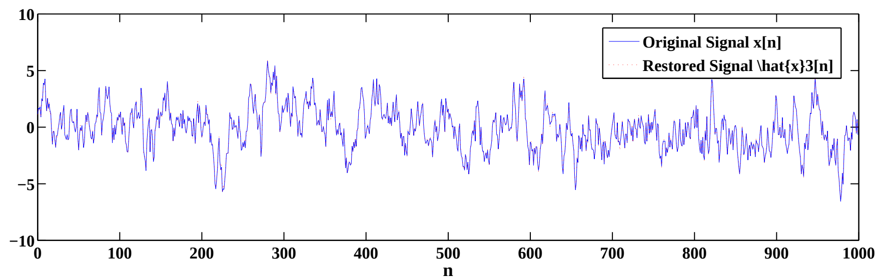

FIGURE 14.71: Plot of the original signal $x[n]$ and restored signal $\hat{x}_3[n]$.

```
vn = filter(1,[1,-0.9],xn);
L = 128; Nfft = 512;
[Pxx,w] = pwelch(xn,L,fix(0.5*L),Nfft);
[Pvv,w] = pwelch(vn,L,fix(0.5*L),Nfft);
Pxxdb = 10*log10(Pxx./min(Pxx));
Pvvdb = 10*log10(Pvv./min(Pvv));
%% Part b
x1hat = filter([1,-0.9],1,vn);
%% Part c
sigz = 0.1;
yn = vn + randn(1,N)*sigz;
x2hat = filter([1,-0.9],1,yn);
%% Part d
w2 = linspace(0,2,1001)*pi;
Sx = abs(1./(1-0.9*exp(-1j*w2))).^4;
H = Sx./(Sx + sigz^2);
hn = ifft(H);
vhat = filter(hn,1,yn);
```

```
x3hat = filter([1,-0.9],1,vhat);
%% Part a: Periodogram
hfa = figconfg('P1447a','small');
plot(w/pi,Pxxdb,w/pi,Pvvdb)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD (dB)','fontsize',LFS)
legend('P_{xx}(\omega)','P_{vv}(\omega)','location','best')

hfb = figconfg('P1447b','long');
plot(n,xn,n,x1hat,':r')
xlabel('n','fontsize',LFS)
legend('Original Signal x[n]','Restored Signal \hat{x}1[n]',...
    'location','best')

hfc = figconfg('P1447c','long');
plot(n,xn,n,x2hat,':r')
xlabel('n','fontsize',LFS)
legend('Original Signal x[n]','Restored Signal \hat{x}2[n]',...
    'location','best')

hfd = figconfg('P1447d','long');
plot(n,xn,n,x3hat,':r')
xlabel('n','fontsize',LFS)
legend('Original Signal x[n]','Restored Signal \hat{x}3[n]',...
    'location','best')
```

48. (a) See script output.
    (b) See script output.
    (c) See plot below.

    MATLAB script:

```
% P1448:
close all; clc
N = 1e5; p = 4;
mu = zeros(1,p);
R = bsxfun(@minus,0:3,(0:3)');
R = 0.95.^abs(R);
randn('seed',0)
X = Normal_ND(N,mu,R);
```

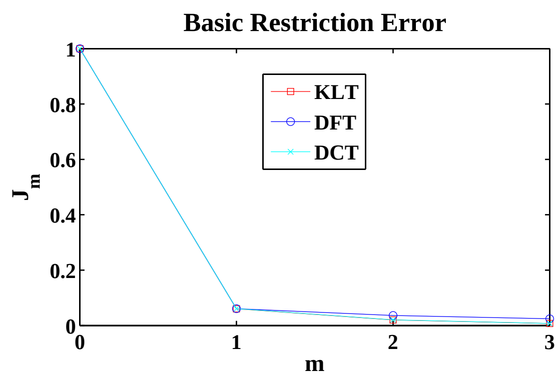

FIGURE 14.72: Plot of the basis restriction error J_m as a function of m of KLT, DFT, and DCT.

```
%% KLT
[U S A] = svd(R);
C1 = X*A;
Jm1 = var(C1,1); Jm1 = fliplr(cumsum(Jm1(end:-1:1))/sum(Jm1)), A
%% DFT
W = dftmtx(4);
C2 = X*W';
Jm2 = var(C2,1); Jm2 = fliplr(cumsum(Jm2(end:-1:1))/sum(Jm2)), W'
%% DCT
B = dctmtx(4);
C3 = X*B';
Jm3 = var(C3,1); Jm3 = fliplr(cumsum(Jm3(end:-1:1))/sum(Jm3)), B'
%% Plot:
hfa = figconfg('P1448a','small');
plot(0:3,Jm1,'-sr',0:3,Jm2,'-ob',0:3,Jm3,'-xc')
set(gca,'Xtick',0:3)
xlabel('m','fontsize',LFS)
ylabel('J_m','fontsize',LFS)
title('Basic Restriction Error','fontsize',TFS)
legend('KLT','DFT','DCT','location','best')
```

49. (a) Proof:

Suppose $m - n = \ell$, the cross-correlation $r_{yx}[\ell]$ is

$$\begin{aligned} r_{yx}[\ell] &= E[y[m]x[n]] = E[\alpha x[m-k]x[n]] = \alpha E[x[m-k]x[n]] \\ &= \alpha r_{xx}[m-k-n] = \alpha r_{xx}[\ell-k] \end{aligned}$$

$$\max_{\ell} r_{xx}[\ell - k] = r_{xx}[k-k] = r_{xx}[0]$$

Thus, we proved that $r_{yx}[k] = \max_{\ell} r_{yx}[\ell]$.

(b) Solution:
Construct $h[\ell] = x[n-\ell]$, $\ell = 0, 1, 2, \ldots$ and search for the largest output from $h[\ell] * \alpha x[n-k]$ to find k.

50. (a) Comments:
Yes, $y[n]$ is wide-sense stationary.

(b) Solution:
The mean of the output process is

$$E[y[n]] = E[x[n]] \sum_{n=-\infty}^{\infty} h[n] = 0$$

The variance of the output process is

$$E[y^2[n]] = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\mathrm{e}^{\mathrm{j}\omega})|^2 d\omega$$

The ACRS of the output process is

$$r_y[\ell] = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\mathrm{e}^{\mathrm{j}\omega})|^2 \mathrm{e}^{\mathrm{j}\omega\ell} d\omega$$

(c) Comments:
See script output for details.

(d) Comments:
See script output for details.

(e) Comments:
See script output for details.

(f) See plot below.

MATLAB script:

```
% P1450:
% close all; clc
b = [1 -0.1 -0.72]; a = [1 -0.9 0.81];
N = 5e3;
randn('seed',0)
xn = randn(1,N);
```

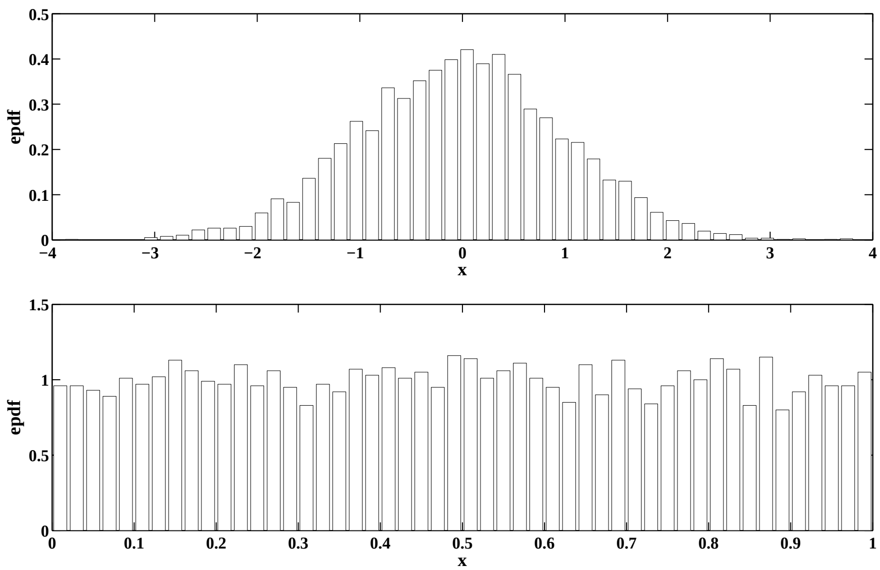

FIGURE 14.73: Plots of the histograms of the two output sequences.

```
% rand('seed',0)
% xn = rand(1,N); % part e
yn = filter(b,a,xn);
yn(1:1e3) = [];
mu_y = mean(yn), var_y = var(yn),
ryy = acrs(yn,10),
[xo,px]=epdf(xn,50);
%% Plot
hfa = figconfg('P1450a','long');
bar(xo,px,'facecolor','w')
xlabel('x','fontsize',LFS)
ylabel('epdf','fontsize',LFS)
```

51. (a) See plot below.
    (b) See plot below.
    (c) See plot below.
    (d) See plot below.

    MATLAB script:

```
% P1451:
```

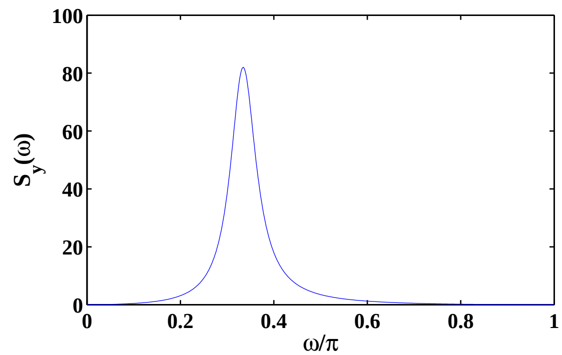

FIGURE 14.74: Plot of the true PSD.

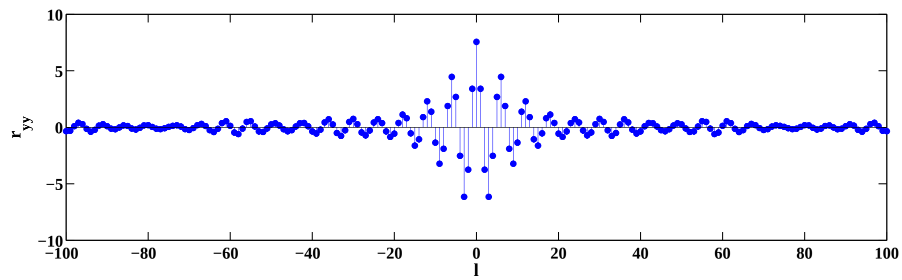

FIGURE 14.75: Plot of the acrs $r_y[\ell]$, for $-100 \le \ell \le 100$.

```
close all; clc
b = [1 -0.1 -0.72]; a = [1 -0.9 0.81];
N = 5e3;
randn('seed',0)
xn = randn(1,N);
yn = filter(b,a,xn);
yn(1:1e3) = [];
L = 100;
ryy = acrs(yn,L+1);
w = linspace(0,1,512)*pi;
Sy = freqz(b,a,w);
Sy = Sy.*conj(Sy);
N = 32;
% N = 64;
% N = 128;
```

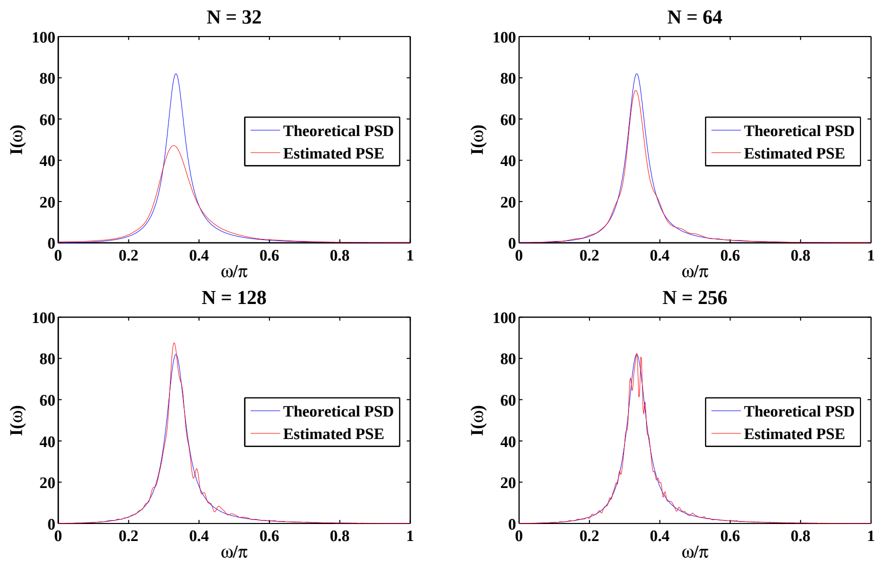

FIGURE 14.76: Plot of the true PSD.

```
% N = 256;
K = 100;
randn('seed',0)
X = randn(N,K);
Y = filter(b,a,X);
I = zeros(K,1024);
for ii = 1:K
     I(ii,:) = psdmodper2(Y(:,ii),1024,rectwin(length(X(:,ii))));
end
I = mean(I,1);
%% Plot
hfa = figconfg('P1451a','small');
plot(w/pi,Sy)
xlabel('\omega/\pi','fontsize',LFS)
ylabel('S_y(\omega)','fontsize',LFS)
hfb = figconfg('P1451b','long');
stem(-L:L,[flipud(ryy(2:end));ryy],'filled')
xlabel('l','fontsize',LFS)
ylabel('r_{yy}','fontsize',LFS)
hfc = figconfg('P1451c','small');
plot(w/pi,Sy,w/pi,I(1:512),'r')
```

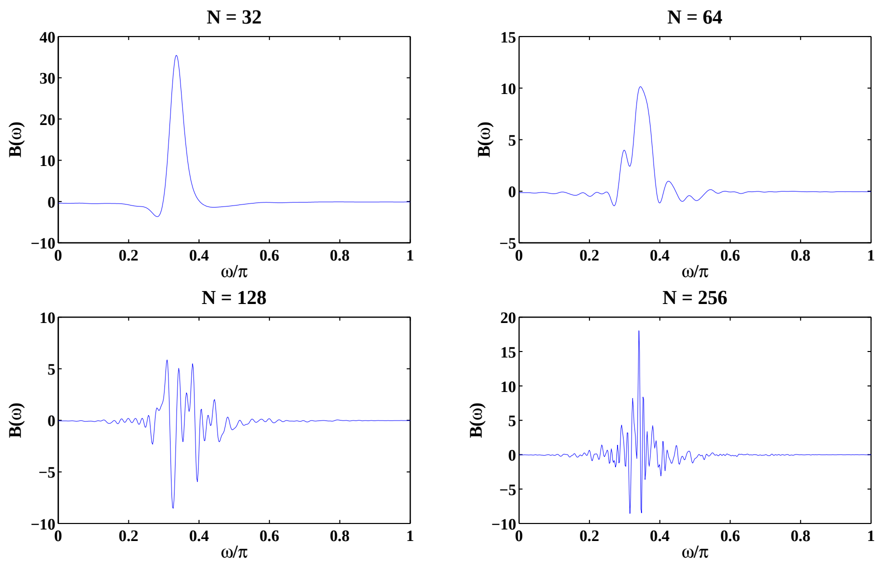

FIGURE 14.77: Plot of the true PSD.

```
xlabel('\omega/\pi','fontsize',LFS)
ylabel('I(\omega)','fontsize',LFS)
title(['N = ',num2str(N)],'fontsize',TFS)
legend('Theoretical PSD','Estimated PSE','location','best')
hfd = figconfg('P1451d','small');
plot(w/pi,Sy-I(1:512))
xlabel('\omega/\pi','fontsize',LFS)
ylabel('B(\omega)','fontsize',LFS)
title(['N = ',num2str(N)],'fontsize',TFS)
```

52. tba

53. Proof:
Using the equation (14.156), we have

$$e^b_{m+1}[n] = \boldsymbol{b}^T_{m+1}\boldsymbol{x}_{m+1}[n] + x[n-m]$$

Substitute equations (14.153) and (14.159) into the above equation, we have

$$\begin{aligned} e^b_{m+1}[n] &= \left(\begin{bmatrix} 0 \\ \boldsymbol{b}_m \end{bmatrix} + \begin{bmatrix} 1 \\ \boldsymbol{a}_m \end{bmatrix} k_{m+1}\right)^T \begin{bmatrix} x[n] \\ \boldsymbol{x}_m[n-1] \end{bmatrix} + x[n-m] \\ &= \left(\boldsymbol{b}^T_m\boldsymbol{x}_m[n-1] + x[n-m]\right) + k_{m+1}\left(x[n] + \boldsymbol{a}^T_m\boldsymbol{x}_m[n-1]\right) \end{aligned}$$

Given equations (14.156) and (14.152), we have

$$e_m^b[n-1] = \boldsymbol{b}_m^T \boldsymbol{x}_m[n-1] + x[n-m]$$

$$e_m^f[n] = x[n] + \boldsymbol{a}_m^T \boldsymbol{x}_m[n-1]$$

Hence, we proved that

$$e_{m+1}^b[n] = e_m^b[n-1] + k_{m+1} e_m^f[n]$$

54. Proof:
For $i = 1, 2, \ldots, p$, we have

$$\begin{aligned}
\frac{\partial S}{\partial a_i} &= \frac{\partial}{\partial a_i}\left(\gamma_{00} + 2\sum_{k=1}^{p} a_k \gamma_{k0} + \sum_{k=1}^{p}\sum_{m=1}^{p} a_k a_m \gamma_{km}\right) \\
&= 0 + 2\gamma_{i0} + \left(\sum_{k=1}^{p} a_k \gamma_{ki} + a_i \gamma_{ii}\right) + \left(\sum_{m=1}^{p} a_m \gamma_{mi} + a_i \gamma_{ii}\right) - 2a_i \gamma_{ii} \\
&= 0
\end{aligned}$$

Since we have $\gamma_{ij} = \gamma_{ji}$, the above equation can be simplified as

$$2\gamma_{i0} + 2\sum_{m=1}^{p} a_m \gamma_{im} = 0$$

which can be written as

$$-\gamma_{i0} = \sum_{m=1}^{p} \hat{a}_m \gamma_{im} = \begin{bmatrix} \gamma_{i1} & \cdots & \gamma_{ip} \end{bmatrix} \hat{\boldsymbol{a}}_m$$

Hence, we proved that

$$\boldsymbol{\Gamma}\hat{\boldsymbol{a}} = -\boldsymbol{\gamma}$$

We can also prove that

$$\begin{aligned}
S &= \gamma_{00} + 2\boldsymbol{\gamma}^T \hat{\boldsymbol{a}} + \hat{\boldsymbol{a}}^T \boldsymbol{\Gamma} \hat{\boldsymbol{a}} = \gamma_{00} + 2\boldsymbol{\gamma}^T \hat{\boldsymbol{a}} + \hat{\boldsymbol{a}}^T(-\boldsymbol{\gamma}) \\
&= \gamma_{00} + \boldsymbol{\gamma}^T \hat{\boldsymbol{a}} = \gamma_{00} + \boldsymbol{\gamma}^T(-\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}) \\
&= \gamma_{00} - \boldsymbol{\gamma}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}
\end{aligned}$$

Review Problems

55. (a) See plot below.

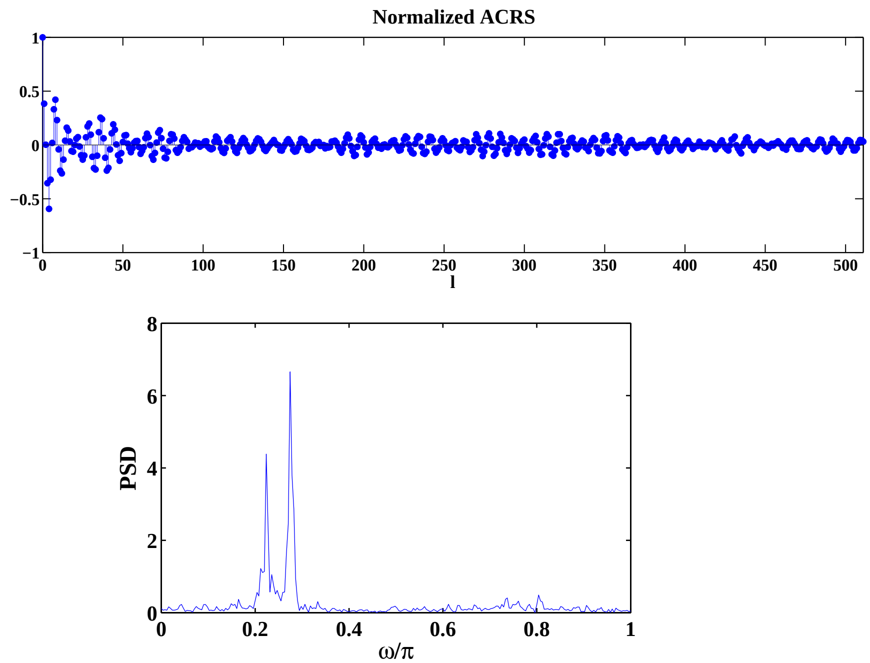

FIGURE 14.78: Plot of the ACRS and the PSD for a $B = 8$ bit quantizer use an $M = 512$ length window.

(b) See plot below.

(c) See plot below.

MATLAB script:

```
% P1455: Analyze the spectral characteristics of quantization noise
close all; clc
b = 1; a = [1 -2.7607 3.8106 -2.6535 0.9238];
N = 2048;
% N = 3;
randn('seed',0)
vn = randn(1,N);
xn = filter(b,a,vn);
B = 8;
% B = 3; % part c
```

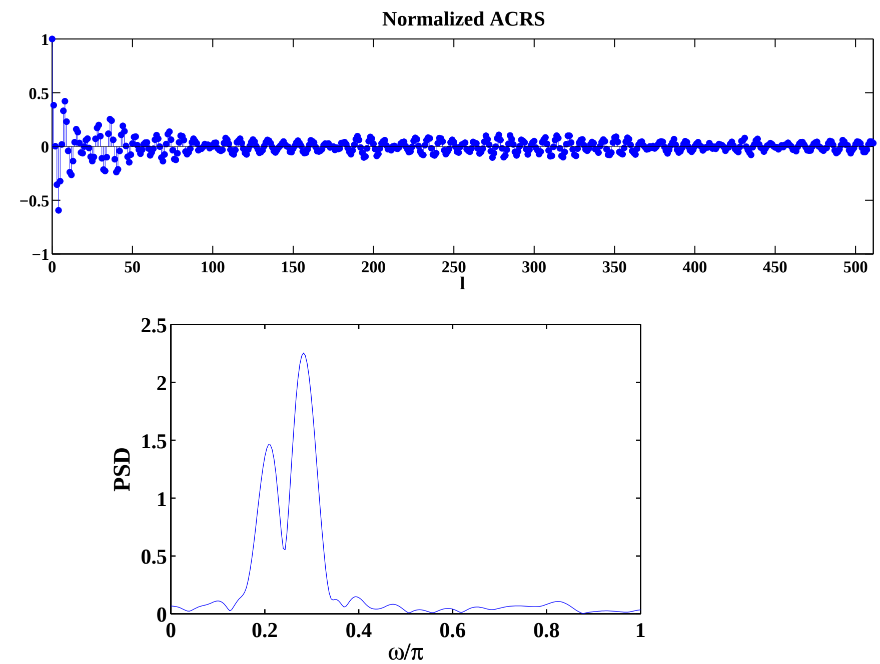

FIGURE 14.79: Plot of the ACRS and the PSD for a $B = 8$ bit quantizer use an $M = 50$ length window.

```
M = 512;
% M = 50;
%% Quantization
Bsign = sign(xn);
xq = dec2bin(abs(xn));
xq = xq(:,1:B-1);
xq = bin2dec(xq);
xq = xq.*Bsign';
en = xq(:) - xn(:);
%% Demean
en = en - mean(en);
%% Compute normalized acrs
re = acrs(en,512);
rho_en = re./re(1);
%% Compute PSD
wn = zeros(size(re));
wn(1:M) = bartlett(M);
```

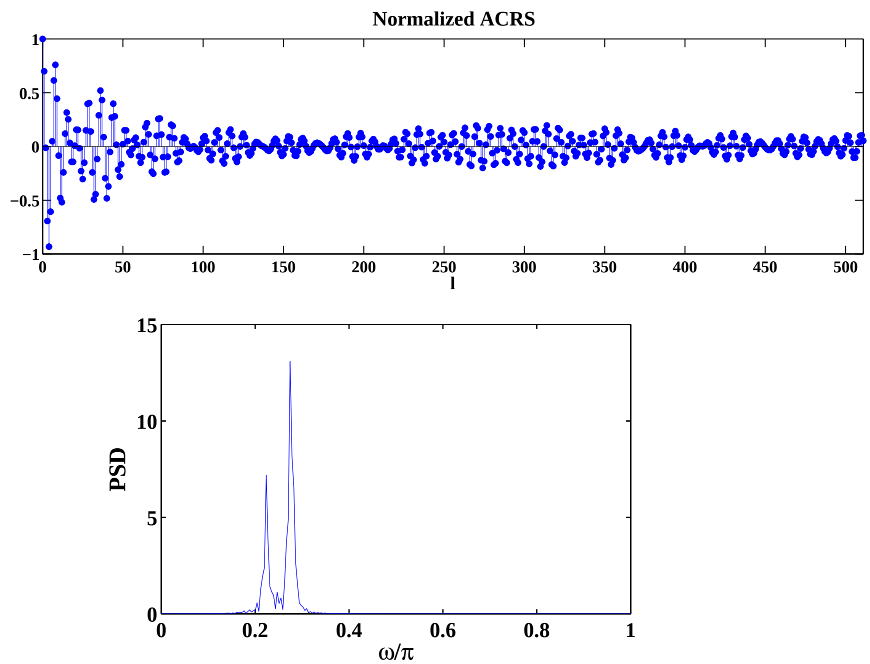

FIGURE 14.80: Plot of the ACRS and the PSD for a $B = 3$ bit quantizer use an $M = 512$ length window.

```
I = abs(fft(rho_en.*wn));
w = linspace(0,1,512/2)*pi;
%% Plot:
hfa = figconfg('P1455a','long');
stem(0:511,rho_en,'filled'); xlim([0 511])
xlabel('l','fontsize',LFS)
title('Normalized ACRS','fontsize',TFS)
hfb = figconfg('P1455b','small');
plot(w/pi,I(1:512/2));
xlabel('\omega/\pi','fontsize',LFS)
ylabel('PSD','fontsize',LFS)
```

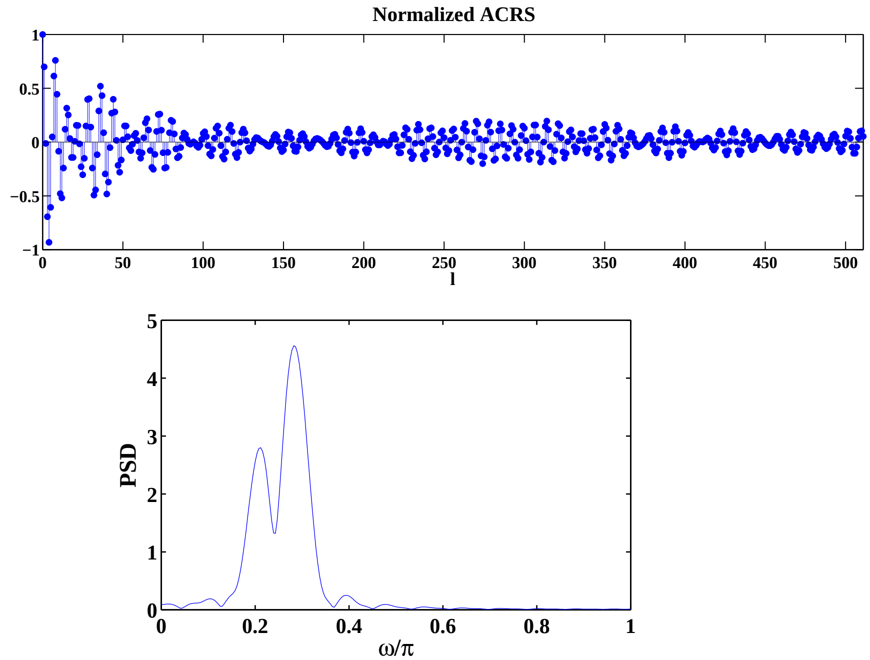

FIGURE 14.81: Plot of the ACRS and the PSD for a $B = 3$ bit quantizer use an $M = 50$ length window.

56. (a) MATLAB function:

```
function S=cpsd(x,y,L,Nfft,w)
% Welch cross-PSD estimator
M=fix((length(x)-L/2)/(L/2)); % 50% overlap
time=(1:L)';
I=zeros(Nfft,1);
if nargin < 4
    Nfft = max(2^L,512);
end
if nargin < 5
    w=hanning(L); % Choose default window
end
w=w/(norm(w)/sqrt(L)); % sum w^2[k]=L
for m=1:M
   %xw=w.*detrend(x(time)); % detrenting
   xw=w.*x(time); % data windowing
   yw=w.*y(time);
```

```
        X=fft(xw,Nfft);
        Y=fft(yw,Nfft);
        I=I+X.*conj(Y);
        time=time+L/2;
    end
    I=I/(M*L); S=2*I(1:Nfft/2); S(1)=S(2);
```

(b) See plot below.

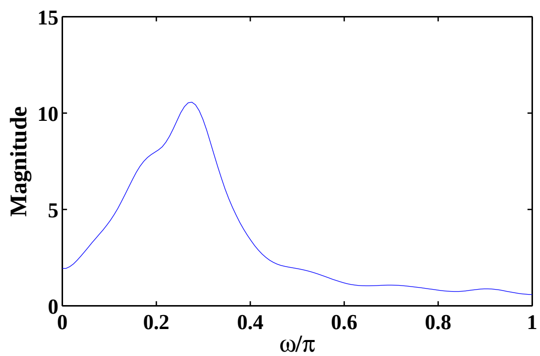

FIGURE 14.82: Plot of the magnitude of the cross-PSD using $L = 32$.

(c) See plot below.

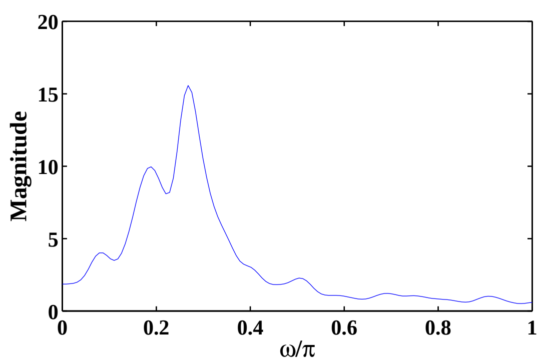

FIGURE 14.83: Plot of the magnitude of the cross-PSD using $L = 64$.

MATLAB script:

```
% P1456: Computation of cross-PSD
```

```
close all; clc
b = 1; a = [1 -1.2728 0.81];
N = 1024;
randn('seed',0)
xn = randn(1,N);
yn = filter(b,a,xn);
L = 32;
% L = 64;
Sxy = cpsd(xn(:),yn(:),L,256,hann(L));
Sxy_mag = abs(Sxy);
w = linspace(0,1,length(Sxy_mag))*pi;
%% Plot:
hfa = figconfg('P1456a','small');
plot(w/pi,Sxy_mag);
xlabel('\omega/\pi','fontsize',LFS)
ylabel('Magnitude','fontsize',LFS)
```

57. (a) See plot below.

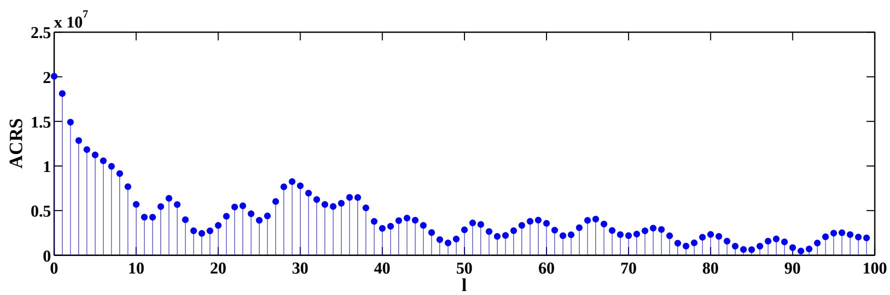

FIGURE 14.84: Plot of the ACRS estimate of the noise process.

(b) Solution:

$$p = 2, \qquad a = [1 - 1.26280.2976]$$
$$p = 4, \qquad a = [1 - 1.44720.9584 - 0.49420.0705]$$

(c) See plot below.

(d) See plot below.

(e) See plot below.

(f) See plot below.

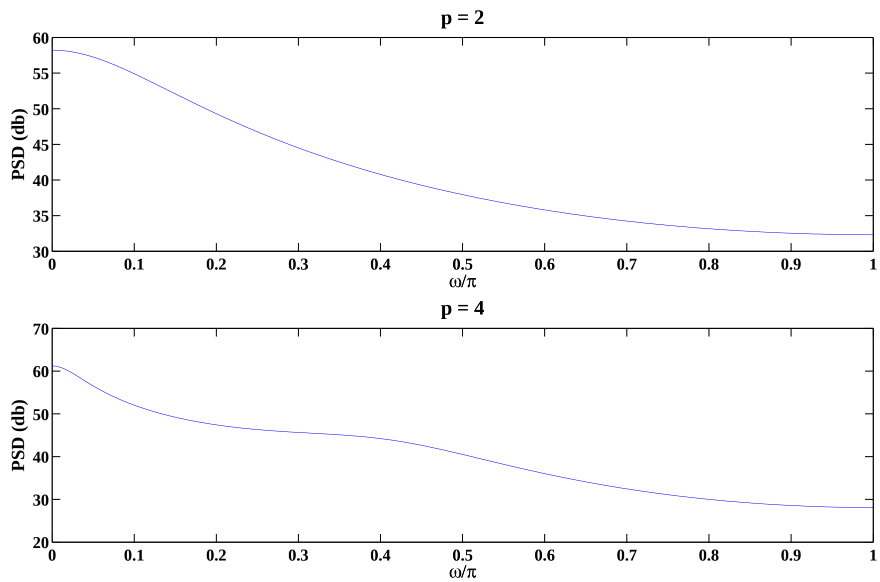

FIGURE 14.85: Plot of the PSD estimate using the AR(2) and AR(4) models.

(g) See plot below.

(h) Compare the plots in above four parts and comment on your observation.

MATLAB script:

```
% P1457: Signal ACRS and spectral characteristics analysis
close all; clc
load f16.mat
x = f16; Nx = length(x);
```

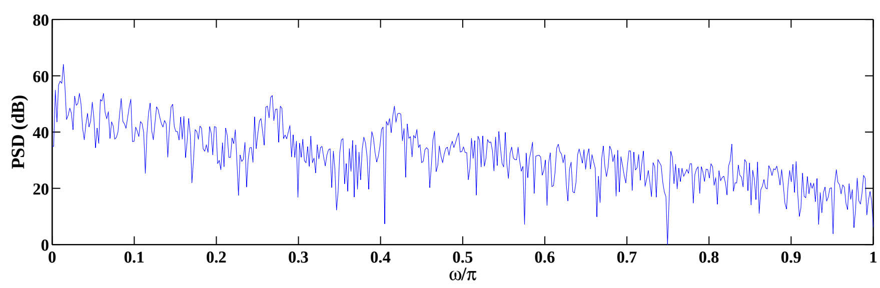

FIGURE 14.86: Plot of the periodogram PSD estimate of the noise process.

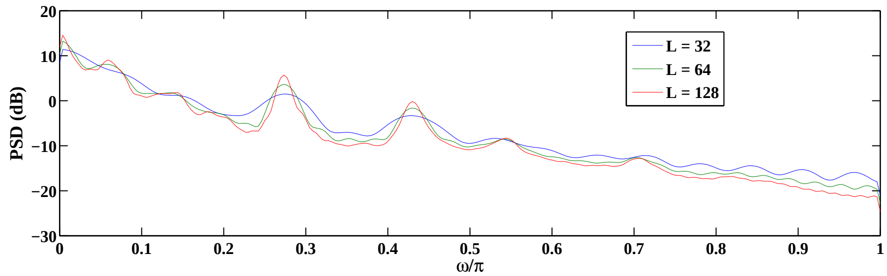

FIGURE 14.87: Plot of the Bartlett PSD estimate using $L = 32$, 64, and 128.

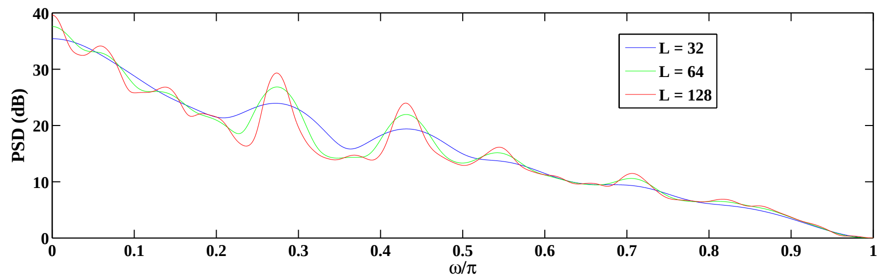

FIGURE 14.88: Plot of the Blackman-Tukey PSD estimate using $L = 32$, 64, and 128.

```
%% Part a
L = 100;
rx = acrs(x,L);
hfa = figconfg('P1457a','long');
stem(0:L-1,rx,'filled');
xlabel('l','fontsize',LFS)
ylabel('ACRS','fontsize',LFS)

%% Part b & c
% [a2,V2,FPE2]= arwin(x,2);
% [a4,V4,FPE4]= arwin(x,4);
% NFFT = 512;
% PSD2=20*log10(sqrt(V2)*abs(freqz(1,a2,NFFT/2+1)))-10*log10(NFFT/2+1);
% PSD4=20*log10(sqrt(V4)*abs(freqz(1,a4,NFFT/2+1)))-10*log10(NFFT/2+1);
% w = linspace(0,1,NFFT/2+1)*pi;
% hfb = figconfg('P1457b','long');
% plot(w/pi,PSD2);
```

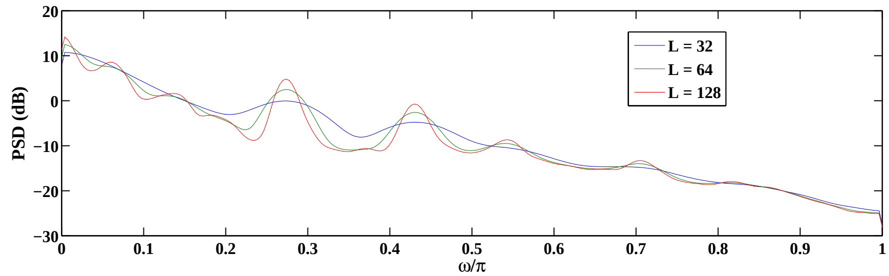

FIGURE 14.89: Plot of the Welch PSD estimate using 50% overlap, Hamming window, and $L = 32$, 64, and 128.

```
% xlabel('\omega/\pi','fontsize',LFS)
% ylabel('PSD (db)','fontsize',LFS)
% title('p = 2','fontsize',TFS)
% hfc = figconfg('P1457c','long');
% plot(w/pi,PSD4);
% xlabel('\omega/\pi','fontsize',LFS)
% ylabel('PSD (db)','fontsize',LFS)
% title('p = 4','fontsize',TFS)

%% Part d: Periodogram PSD
% N = 1024;
% % N = 2048;
% I1 = psdper(x,N);
% hfd = figconfg('P1410d','long');
% I1db = 10*log10(I1(1:N/2)./min(I1(1:N/2)));
% w = linspace(0,1,N/2)*pi;
% plot(w/pi,I1db)
% xlabel('\omega/\pi','fontsize',LFS)
% ylabel('PSD (dB)','fontsize',LFS)

%% Part e: Bartlett PSD
% N = 1024;
% K = floor(Nx/N);
% X = reshape(x(1:K*N),N,K);
% Nfft = 512;
% L = [32 64 128];
% Pxx = zeros(K,Nfft/2+1,length(L));
```

```
% for jj = 1:length(L)
%     for ii = 1:K
%         [Pxx(ii,:,jj), w] = ...
%             pwelch(X(:,ii),rectwin(L(jj)),0,Nfft);
%     end
% end
% P = squeeze(mean(Pxx,1));
% Pdb = bsxfun(@rdivide,P,mean(P,1));
% Pdb = 10*log10(Pdb);
% hfe = figconfg('P1457e','long');
% plot(w/pi,Pdb)
% xlabel('\omega/\pi','fontsize',LFS)
% ylabel('PSD (dB)','fontsize',LFS)
% legend(['L = ',num2str(L(1))],['L = ',num2str(L(2))],...
%     ['L = ',num2str(L(3))],'location','best')


%% Part f: Blackman-Tukey PSD
% N = length(x);
% w = linspace(0,1,2^(ceil(log2(N))-1))*pi;
% hff = figconfg('P1409a','long');
% L = [32 64 128]; PC = 'bgr';
% for ii = 1:3
%     S = psdbt(x(:),L(ii),2^(ceil(log2(N))));
%     figure(hff)
%     Sdb = 10*log10(S./min(S));
%     plot(w/pi,Sdb,PC(ii)); hold on
%     xlabel('\omega/\pi','fontsize',LFS)
%     ylabel('PSD (dB)','fontsize',LFS)
% end
% legend('L = 32','L = 64','L = 128','location','best')

%% Part g: Welch PSD
% N = 1024;
% K = floor(Nx/N);
% X = reshape(x(1:K*N),N,K);
% Nfft = 512;
% L = [32 64 128];
% Pxx = zeros(K,Nfft/2+1,length(L));
% for jj = 1:length(L)
```

```
%     for ii = 1:K
%         [Pxx(ii,:,jj), w] = ...
%             pwelch(X(:,ii),hamming(L(jj)),fix(0.5*L(jj)),Nfft);
%     end
% end
% P = squeeze(mean(Pxx,1));
% Pdb = bsxfun(@rdivide,P,mean(P,1));
% Pdb = 10*log10(Pdb);
% hfg = figconfg('P1457g','long');
% plot(w/pi,Pdb)
% xlabel('\omega/\pi','fontsize',LFS)
% ylabel('PSD (dB)','fontsize',LFS)
% legend(['L = ',num2str(L(1))],['L = ',num2str(L(2))],...
%     ['L = ',num2str(L(3))],'location','best')
```

58. tba