Solution Manual for Digital Signal Processing Using MATLAB A Problem Solving Companion 4th Edition by Ingle Proakis ISBN 1305635124 9781305635128
Full link download:
Solution Manual:
https://testbankpack.com/p/solution-manual-for-digital-signal-processing-using-matlab-a-problem-solving-companion-4th-edition-by-ingle-proakis-isbn-1305635124-9781305635128/

Chapter 2

Discrete-Time Signals and Systems

P2.1 Generate the following sequences using the basic MATLAB signal functions and the basic MATLAB signal operations discussed in this chapter. Plot signal samples using the stem function.

1. $x_1(n) = 3\delta(n+2) + 2\delta(n) - \delta(n-3) + 5\delta(n-7), \; -5 \le n \le 15$

```
% P0201a: x1(n) = 3*delta(n + 2) + 2*delta(n) - delta(n - 3) +
%                           5*delta(n - 7), -5 <= n <= 15.
clc; close all;
x1 = 3*impseq(-2,-5,15) + 2*impseq(0,-5,15) - impseq(3,-5,15) + 5*impseq(7,-5,15); Hf_1 =
figure; set(Hf_1,'NumberTitle','off','Name','P0201a'); n1 = [-5:15];
Hs = stem(n1,x1,'filled'); set(Hs,'markersize',2); axis([min(n1)-
1,max(n1)+1,min(x1)-1,max(x1)+1]); xlabel('n','FontSize',LFS);
ylabel('x_1(n)','FontSize',LFS); title('Sequence x_1(n)','FontSize',TFS);
set(gca,'XTickMode','manual','XTick',n1,'FontSize',8); print -deps2
../EPSFILES/P0201a;
```

The plots of $x_1(n)$ is shown in Figure 2.1.

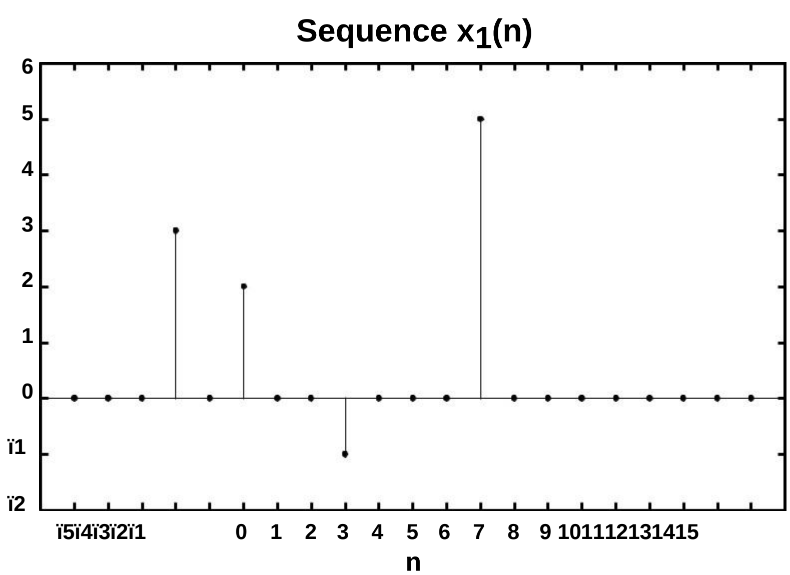

Figure 2.1: Problem P2.1.1 sequence plot

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 2

2. $x_2(n) = \sum_{k=-5}^{5} e^{-|k|}\delta(n-2k), -10 \le n \le 10.$

```
% P0201b: x2(n) = sum_{k = -5}^{5} e^{-|k|}*delta(n - 2k), -10 <= n <= 10 clc; close all;

n2 = [-10:10]; x2 = zeros(1,length(n2));
for k = -5:5
    x2 = x2 + exp(-abs(k))*impseq(2*k ,-10,10);
end
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201b'); Hs =
stem(n2,x2,'filled'); set(Hs,'markersize',2); axis([min(n2)-
1,max(n2)+1,min(x2)-1,max(x2)+1]); xlabel('n','FontSize',LFS);
ylabel('x_2(n)','FontSize',LFS); title('Sequence x_2(n)','FontSize',TFS);
set(gca,'XTickMode','manual','XTick',n2); print -deps2
../EPSFILES/P0201b;
```

The plots of $x_2(n)$ is shown in Figure 2.2.

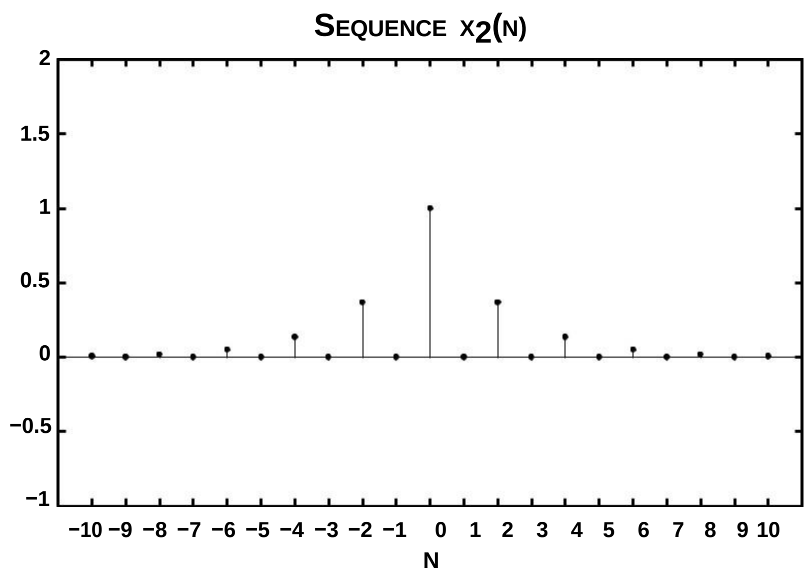

Figure 2.2: Problem P2.1.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 3

3. $x_3(n) = 10u(n) - 5u(n-5) - 10u(n-10) + 5u(n-15)$.

```
% P0201c: x3(n) = 10u(n) - 5u(n - 5) + 10u(n - 10) + 5u(n - 15). clc; close
all;

x3 = 10*stepseq(0,0,20) - 5*stepseq(5,0,20) - 10*stepseq(10,0,20) ...
        + 5*stepseq(15,0,20);
n3 = [0:20];
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201c'); Hs =
stem(n3,x3,'filled'); set(Hs,'markersize',2); axis([min(n3)-
1,max(n3)+1,min(x3)-1,max(x3)+2]);
ytick = [-6:2:12];
xlabel('n','FontSize',LFS); ylabel('x_3(n)','FontSize',LFS);
title('Sequence x_3(n)','FontSize',TFS);
set(gca,'XTickMode','manual','XTick',n3);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0201c;
```

The plots of $x_3(n)$ is shown in Figure 2.3.

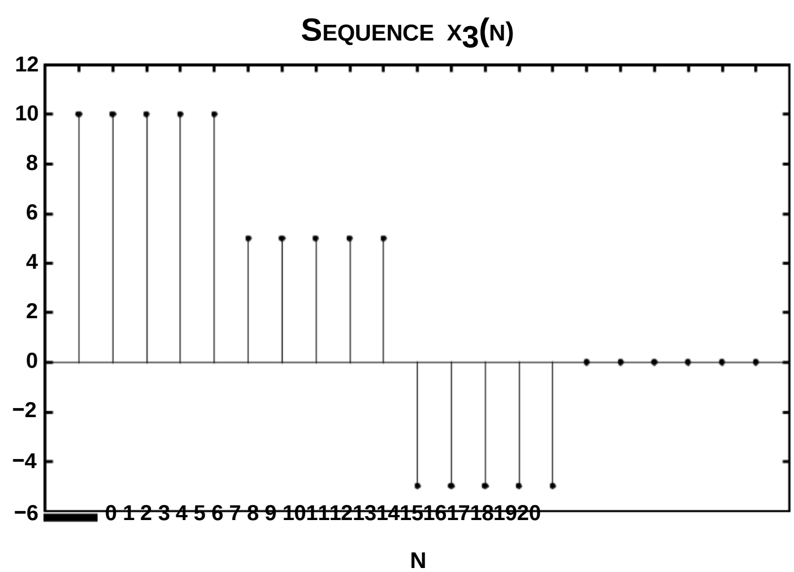

Figure 2.3: Problem P2.1.3 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

 SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 4

4. $x_4.n/ \ D \ e^{0:1n}$ u.n C 20/ u.n 10/ .

```
% P0201d: x4(n) = e ^ {0.1n} [u(n + 20) - u(n - 10)]. clc; close
all;

n4 = [-25:15];
x4 = exp(0.1*n4).*(stepseq(-20,-25,15) - stepseq(10,-25,15));

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201d'); Hs =
stem(n4,x4,'filled'); set(Hs,'markersize',2); axis([min(n4)-2,max(n4)+2,min(x4)-
1,max(x4)+1]); xlabel('n','FontSize',LFS); ylabel('x_4(n)','FontSize',LFS);
title('Sequence x_4(n)','FontSize',TFS); ntick = [n4(1):5:n4(end)];
set(gca,'XTickMode','manual','XTick',ntick);

print -deps2 ../CHAP2_EPSFILES/P0201d; print -deps2 ../../Latex/P0201d;
```

The plots of $x_4.n/$ is shown in Figure 2.4.

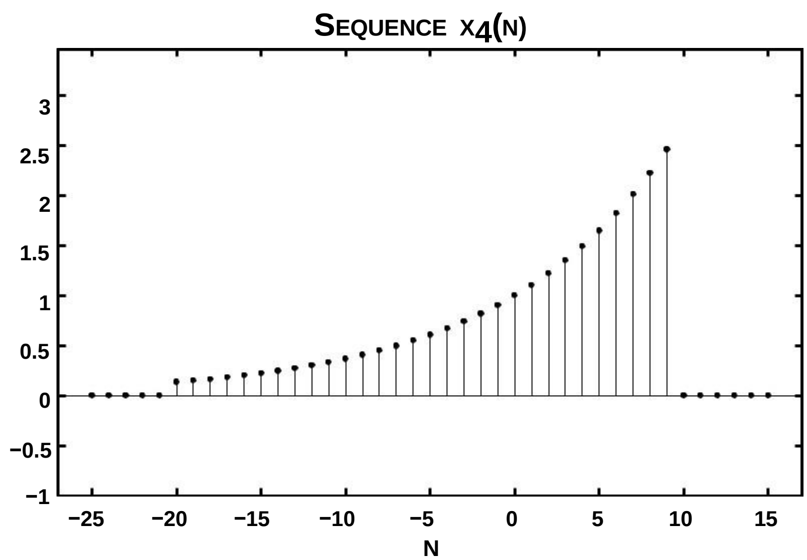

Figure 2.4: Problem P2.1.4 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Solutions Manual for DSP using Matlab (4th Edition) 5

5. $x_5(n) = 5\left[\cos(0.49\pi n) + \cos(0.51\pi n)\right],\ -200 \le n \le 200$. Comment on the waveform shape.

```
% P0201e: x5(n) = 5[cos(0.49*pi*n) + cos(0.51*pi*n)], -200 <= n <= 200. clc; close all;

n5 = [-200:200]; x5 = 5*(cos(0.49*pi*n5) + cos(0.51*pi*n5));

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201e'); Hs = stem(n5,x5,'filled'); set(Hs,'markersize',2); axis([min(n5)-10,max(n5)+10,min(x5)-2,max(x5)+2]); xlabel('n','FontSize',LFS); ylabel('x_5(n)','FOntSize',LFS); title('Sequence x_5(n)','FontSize',TFS);

ntick = [n5(1): 40:n5(end)]; ytick = [-12 -10:5:10 12];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../CHAP2_EPSFILES/P0201e; print -deps2 ../../Latex/P0201e;
```

The plots of $x_5(n)$ is shown in Figure 2.5.

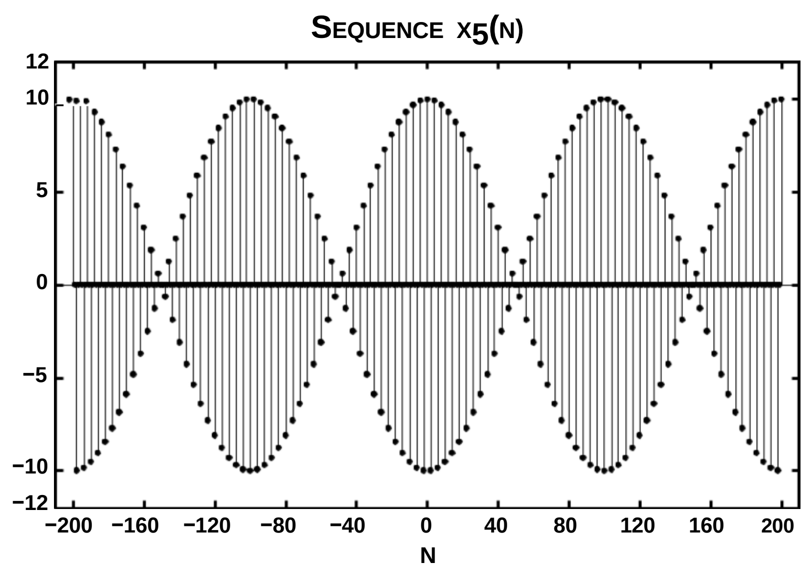

Figure 2.5: Problem P2.1.5 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 6

6. $x_6(n) = 2\sin(0.01\pi n)\cos(0.5\pi n), \; -200 \le n \le 200.$

```
%P0201f: x6(n) = 2 sin(0.01*pi*n) cos(0.5*pi*n), -200 <= n <= 200.
clc; close all;

n6 = [-200:200]; x6 = 2*sin(0.01*pi*n6).*cos(0.5*pi*n6);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201f'); Hs =
stem(n6,x6,'filled'); set(Hs,'markersize',2); axis([min(n6)-
10,max(n6)+10,min(x6)-1,max(x6)+1]); xlabel('n','FontSize',LFS);
ylabel('x_6(n)','FontSize',LFS); title('Sequence x_6(n)','FontSize',TFS);
ntick = [n6(1): 40:n6(end)];

set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0201f; print -deps2 ../../Latex/P0201f;
```

The plots of $x_6(n)$ is shown in Figure 2.6.

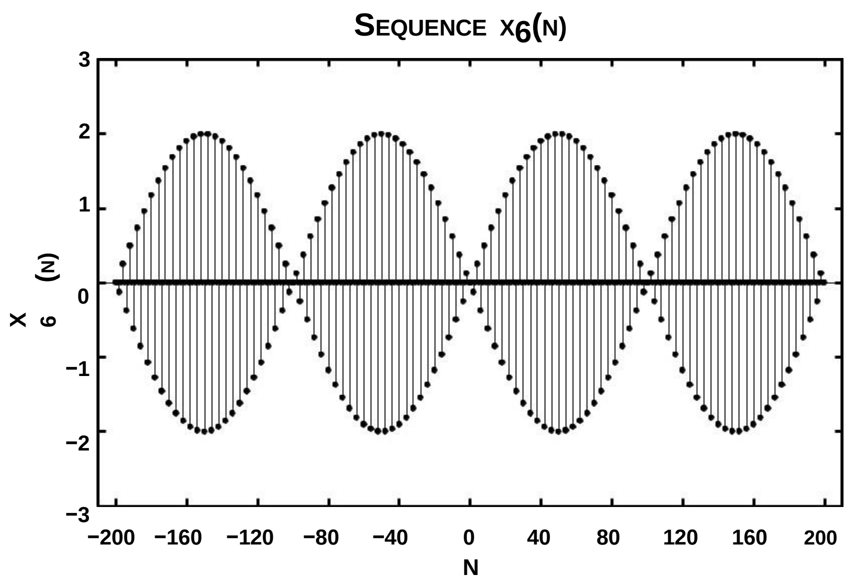

Figure 2.6: Problem P2.1.6 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 7

7. $x_7(n) = e^{-0.05n}\sin(0.1\pi n + \pi/3)$; $0 \le n \le 100$.

```
% P0201g: x7(n) = e ^ {-0.05*n}*sin(0.1*pi*n + pi/3), 0 <= n <=100. clc; close
all;

n7 = [0:100]; x7       = exp(-0.05*n7).*sin(0.1*pi*n7 + pi/3);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201g'); Hs =
stem(n7,x7,'filled'); set(Hs,'markersize',2); axis([min(n7)-
5,max(n7)+5,min(x7)-1,max(x7)+1]); xlabel('n','FontSize',LFS);
ylabel('x_7(n)','FontSize',LFS); title('Sequence x_7(n)','FontSize',TFS);

ntick = [n7(1): 10:n7(end)]; set(gca,'XTickMode','manual','XTick',ntick); print -deps2
../CHAP2_EPSFILES/P0201g;
```

The plots of $x_7(n)$ is shown in Figure 2.7.

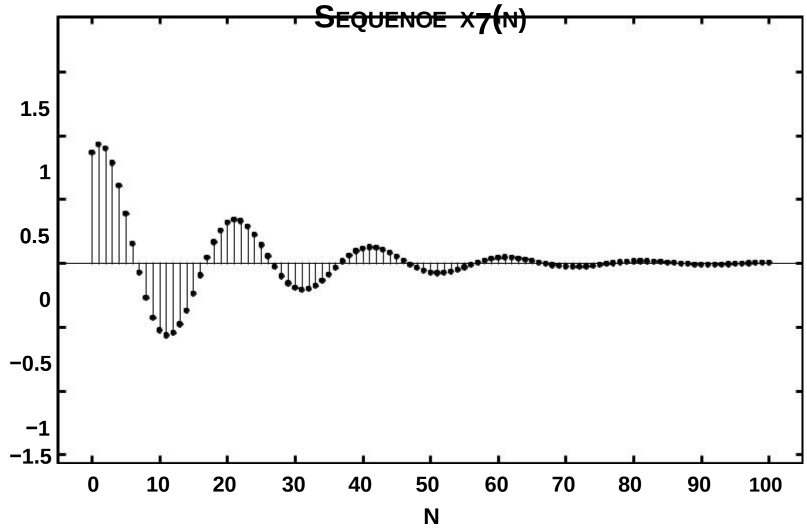

Figure 2.7: Problem P2.1.7 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 8

8. $x_8(n) = e^{0.01n} \sin(0.1\pi n),\ 0 \le n \le 100$.

```
% P0201h: x8(n) = e ^ {0.01*n}*sin(0.1*pi*n), 0 <= n <=100. clc; close
all;

n8 = [0:100]; x8      = exp(0.01*n8).*sin(0.1*pi*n8);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0201h'); Hs =
stem(n8,x8,'filled'); set(Hs,'markersize',2); axis([min(n8)-
5,max(n8)+5,min(x8)-1,max(x8)+1]); xlabel('n','FontSize',LFS);
ylabel('x_8(n)','FontSize',TFS); title('Sequence x_8(n)','FontSize',TFS);

ntick = [n8(1): 10:n8(end)]; set(gca,'XTickMode','manual','XTick',ntick); print -deps2
../CHAP2_EPSFILES/P0201h
```

The plots of $x_8(n)$ is shown in Figure 2.8.

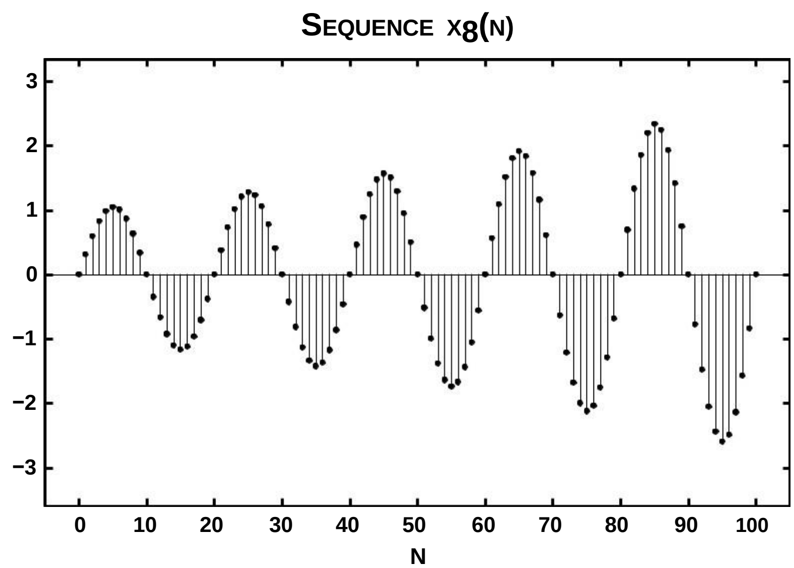

Figure 2.8: Problem P2.1.8 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 9

P2.2 Generate the following random sequences and obtain their histogram using the hist function with 100 bins. Use the bar function to plot each histogram.

1. x_1.n/ is a random sequence whose samples are independent and uniformly distributed over 0; 2 interval. Generate 100,000 samples.

```
% P0202a: x1(n) = uniform[0,2] clc;
close all;

n1 = [0:100000-1]; x1 = 2*rand(1,100000);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0202a'); [h1,x1out] =
hist(x1,100); bar(x1out, h1); axis([-0.1 2.1 0 1200]);

xlabel('interval','FontSize',LFS);
ylabel('number of elements','FontSize',LFS);
title('Histogram of sequence x_1(n) in 100 bins','FontSize',TFS);
print -deps2 ../CHAP2_EPSFILES/P0202a;
```

The plots of x_1.n/ is shown in Figure 2.9.

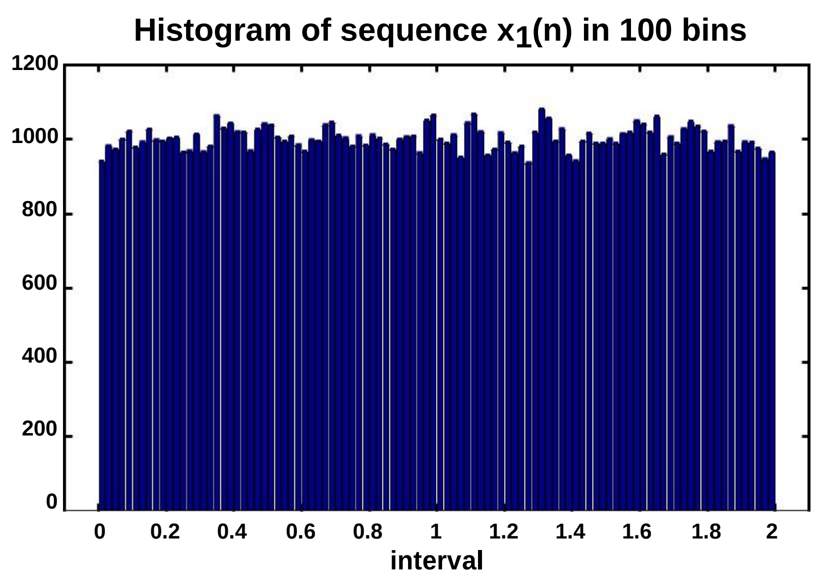

Figure 2.9: Problem P2.2.1 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 10

2. $x_2.n/$ is a Gaussian random sequence whose samples are independent with mean 10 and variance 10. Generate 10,000 samples.

```
% P0202b: x2(n) = gaussian{10,10}
clc; close all;

n2 = [1:10000]; x2 = 10 + sqrt(10)*randn(1,10000);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0202b'); [h2,x2out] =
hist(x2,100); bar(x2out,h2); xlabel('interval','FontSize',LFS);
ylabel('number of elements','FontSize',LFS);

title('Histogram of sequence x_2(n) in 100 bins','FontSize',TFS); print -deps2
../CHAP2_EPSFILES/P0202b;
```

The plots of $x_2.n/$ is shown in Figure 2.10.

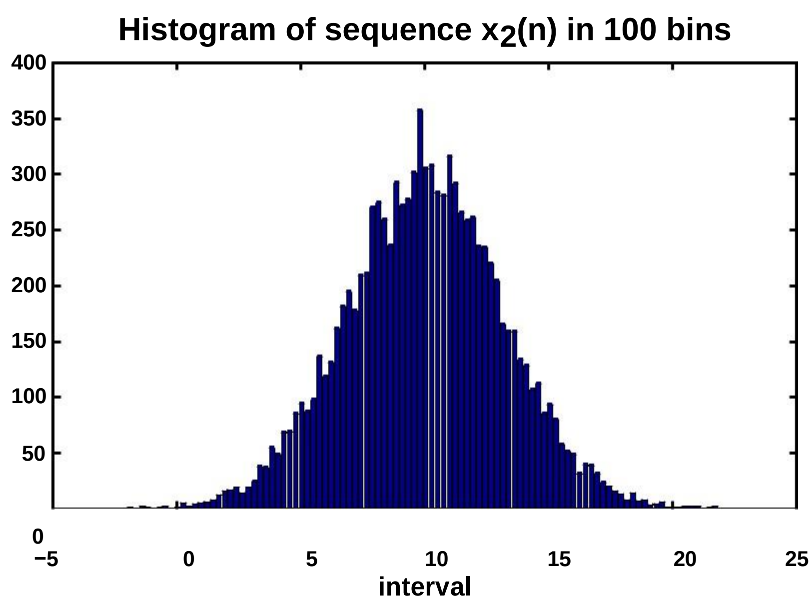

Figure 2.10: Problem P2.2.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 11

3. x_3.n/ D x_1.n/ C x_1.n 1/ where x_1.n/ is the random sequence given in part 1 above. Comment on the shape of this histogram and explain the shape.

```
% P0202c: x3(n) = x1(n) + x1(n - 1) where x1(n) = uniform[0,2] clc; close
all;

n1 = [0:100000-1]; x1 = 2*rand(1,100000);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0202c');
[x11,n11] = sigshift(x1,n1,1); [x3,n3] =
sigadd(x1,n1,x11,n11); [h3,x3out] =
hist(x3,100); bar(x3out,h3); axis([-0.5
4.5 0 2500]);
xlabel('interval','FontSize',LFS);
ylabel('number of elements','FontSize',LFS);
title('Histogram of sequence x_3(n) in 100 bins','FontSize',TFS); print -deps2
../CHAP2_EPSFILES/P0202c;
```

The plots of x_3.n/ is shown in Figure 2.11.

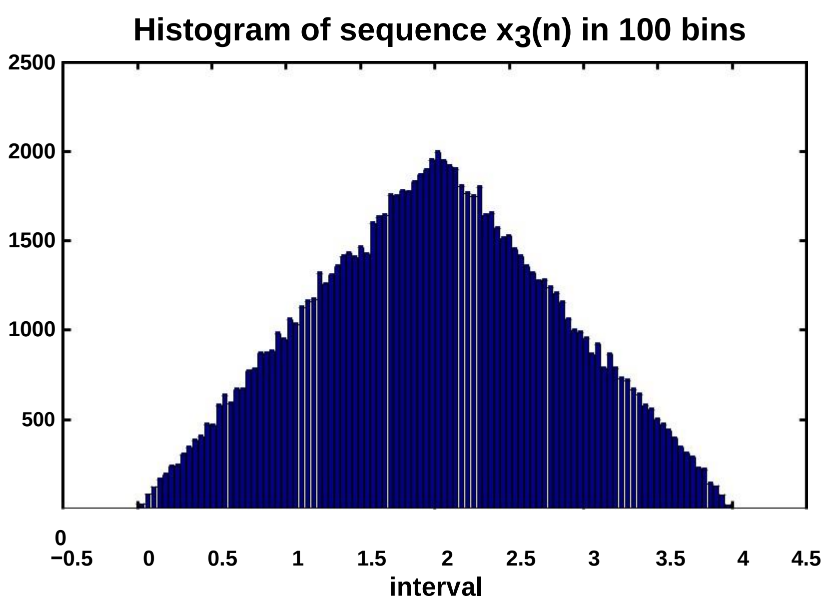

Figure 2.11: Problem P2.2.3 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 12

4. $x_4(n) = \sum_{k=1}^{4} y_k(n)$ where each random sequence $y_k(n)$ is independent of others with samples uniformly dis-tributed over $[-0.5, 0.5]$. Comment on the shape of this histogram.

```
%P0202d: x4(n) = sum_{k=1} ^ {4} y_k(n), where each independent of others
%           with samples uniformly distributed over [-0.5,0.5]; clc; close
all;

y1 = rand(1,100000) - 0.5; y2 = rand(1,100000) - 0.5;
y3 = rand(1,100000) - 0.5; y4 = rand(1,100000) - 0.5;
x4 = y1 + y2 + y3 + y4;

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0202d'); [h4,x4out] =
hist(x4,100); bar(x4out,h4); xlabel('interval','FontSize',LFS);
ylabel('number of elements','FontSize',LFS);

title('Histogram of sequence x_4(n) in 100 bins','FontSize',TFS); print -deps2
../CHAP2_EPSFILES/P0202d;
```

The plots of $x_4(n)$ is shown in Figure 2.12.

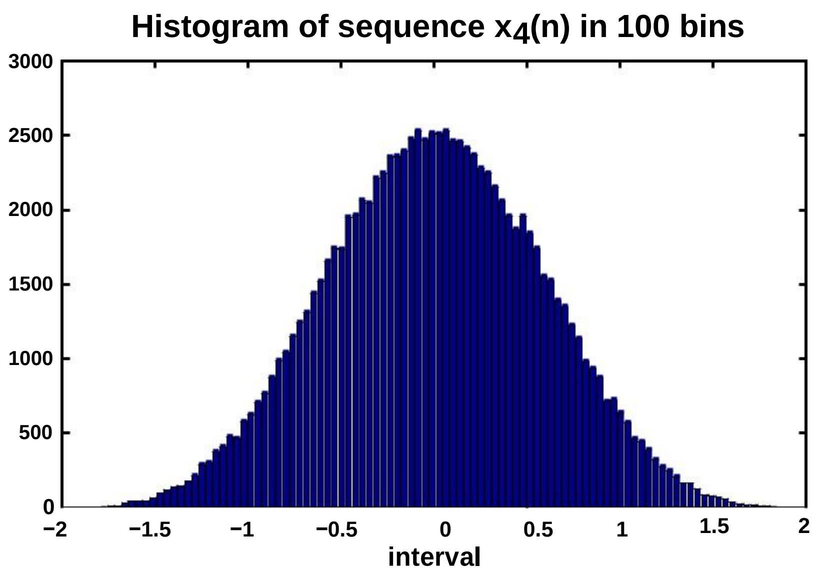

Figure 2.12: Problem P2.2.4 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

 SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 13

P2.3 Generate the following periodic sequences and plot their samples (using the stem function) over the indicated number of periods.

1. xQ1.n/ D f: : : ; 2;1; 0; 1; 2; : : :gperiodic. Plot 5 periods.

```
% P0203a: x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...} periodic. 5 periods clc; close all;

n1 = [-12:12]; x1 = [-2,-1,0,1,2];
x1 = x1'*ones(1,5); x1 = (x1(:))';

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0203a'); Hs =
stem(n1,x1,'filled'); set(Hs,'markersize',2); axis([min(n1)-
1,max(n1)+1,min(x1)-1,max(x1)+1]); xlabel('n','FontSize',LFS);
ylabel('x_1(n)','FontSize',LFS); title('Sequence x_1(n)','FontSize',TFS);

ntick = [n1(1):2:n1(end)]; ytick = [min(x1) - 1:max(x1) + 1];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick); print -deps2
../CHAP2_EPSFILES/P0203a
```

The plots of xQ1.n/ is shown in Figure 2.13.

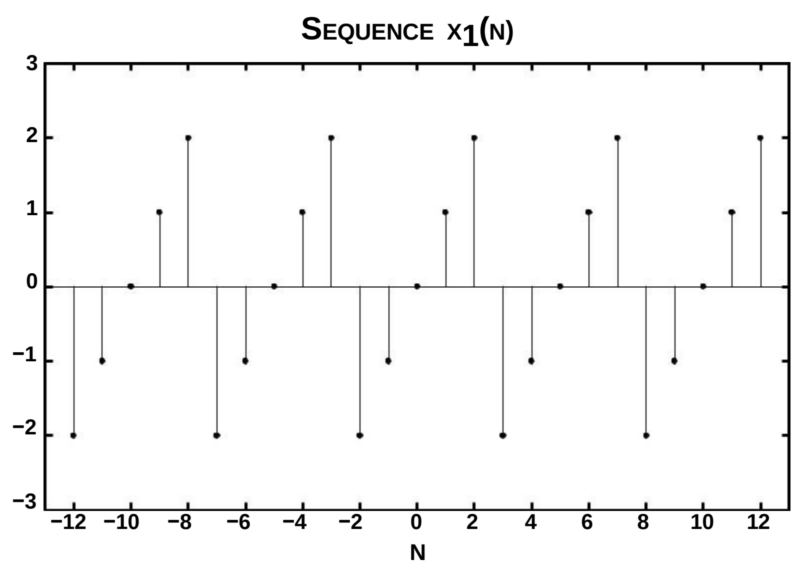

Figure 2.13: Problem P2.3.1 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 14

2. $xQ_2.n/ \; D \; e^{0:1n} \; u.n/ \;\; u.n \;\; 20_{periodic}$. Plot 3 periods.

```
% P0203b: x2 = e ^ {0.1n} [u(n) - u(n-20)] periodic. 3 periods clc; close
all;

n2 = [0:21]; x2 = exp(0.1*n2).*(stepseq(0,0,21)-stepseq(20,0,21));
x2 = x2'*ones(1,3); x2 = (x2(:))'; n2 = [-22:43];

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0203b'); Hs =
stem(n2,x2,'filled'); set(Hs,'markersize',2); axis([min(n2)-
2,max(n2)+4,min(x2)-1,max(x2)+1]); xlabel('n','FontSize',LFS);
ylabel('x_2(n)','FontSize',LFS); title('Sequence x_2(n)','FontSize',TFS);
ntick = [n2(1):4:n2(end)-5 n2(end)];

set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../Chap2_EPSFILES/P0203b;
```

The plots of $xQ_2.n/$ is shown in Figure 2.14.

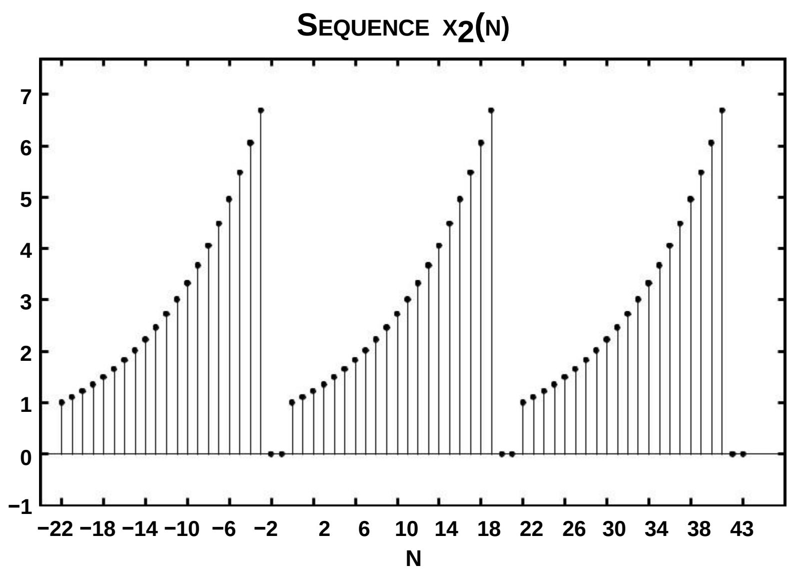

Figure 2.14: Problem P2.3.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 15

3. xQ3.n/ D si n.0:1 n/ u.n/ u.n 10/ . Plot 4 periods.

```
% P0203c: x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...} periodic. 5 periods clc; close all;

n3 = [0:11]; x3 = sin(0.1*pi*n3).*(stepseq(0,0,11)-stepseq(10,0,11));
x3 = x3'*ones(1,4); x3 = (x3(:))'; n3 = [-12:35];

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0203c'); Hs =
stem(n3,x3,'filled'); set(Hs,'markersize',2); axis([min(n3)-
1,max(n3)+1,min(x3)-0.5,max(x3)+0.5]); xlabel('n','FontSize',LFS);
ylabel('x_3(n)','FontSize',LFS); title('Sequence x_3(n)','FontSize',TFS);
ntick = [n3(1):4:n3(end)-3 n3(end)];

set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0203c;
```

The plots of xQ3.n/ is shown in Figure 2.15.

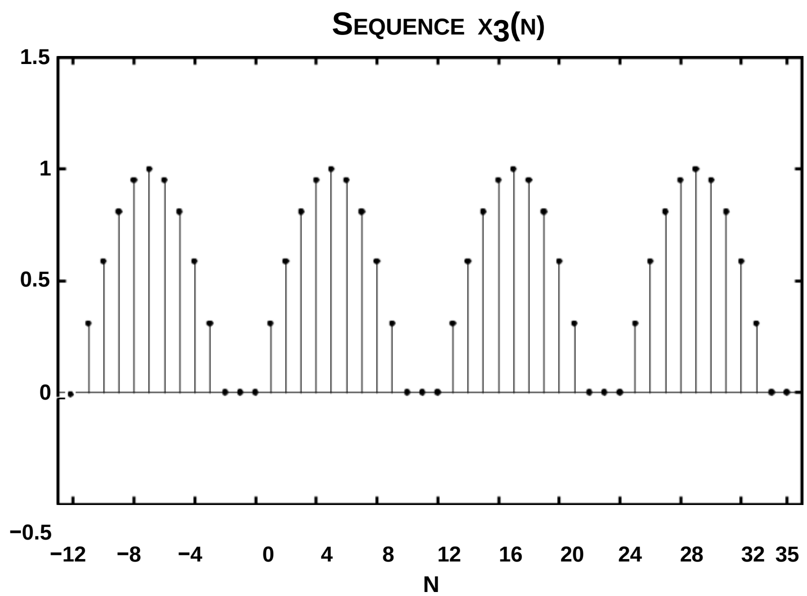

Figure 2.15: Problem P2.3.3 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 16

4. $\tilde{x}_4(n) = \{\dots,1,2,3,\dots\}_{\text{periodic}} + \{\dots,1,2,3,4,\dots\}_{\text{periodic}}$; $0 \le n \le 24$. What is the period of $\tilde{x}_4(n)$?

```
% P0203d x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...} periodic. 5 periods clc; close all;

n4 = [0:24]; x4a = [1 2 3]; x4a = x4a'*ones(1,9); x4a = (x4a(:))'; x4b = [1 2 3
4]; x4b = x4b'*ones(1,7); x4b = (x4b(:))';
x4 = x4a(1:25) + x4b(1:25);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0203d');
Hs = stem(n4,x4,'filled'); set(Hs,'markersize',2);
axis([min(n4)-1,max(n4)+1,min(x4)-1,max(x4)+1]);
xlabel('n', 'FontSize', LFS); ylabel('x_4(n)','FontSize',LFS);
title('Sequence x_4(n):Period = 12','FontSize',TFS);
ntick = [n4(1) :2:n4(end)]; set(gca,'XTickMode','manual','XTick',ntick); print -deps2
../CHAP2_EPSFILES/P0203d;
```

The plots of $\tilde{x}_4(n)$ is shown in Figure 2.16. From the figure, the fundamental period of $\tilde{x}_4(n)$ is 12.

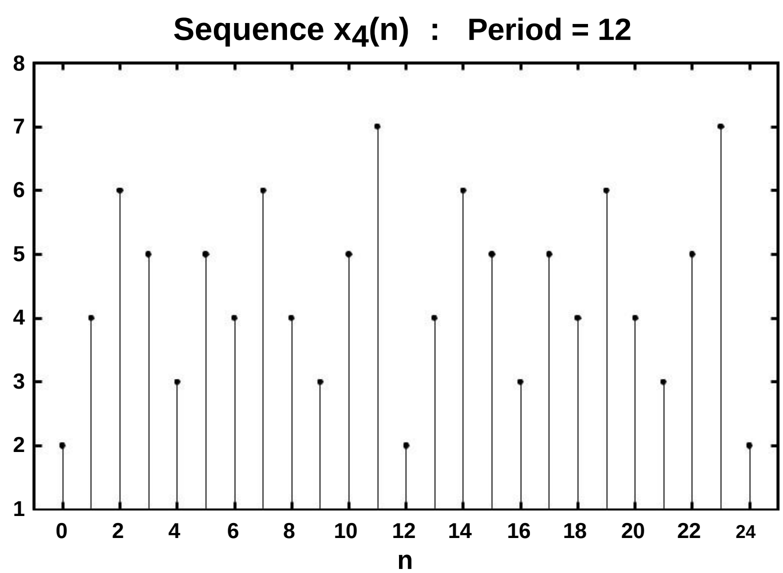

Figure 2.16: Problem P2.3.4 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

P2.4 Let x.n/ D f2; 4; 3; 1; 5; 4; 7g. Generate and plot the samples (use the stem function) of the following sequences.

1. x_1 .n/ D 2x.n 3/ C 3x.n C 4/ x.n/

```
% P0204a: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x1(n) = 2x(n - 3) + 3x(n + 4) - x(n)
clc; close all;

n = [-3:3];       x = [2,4,-3,1,-5,4,7];
[x11,n11] =    sigshift(x,n,3);                         % shift by 3
[x12,n12] =    sigshift(x,n,-4);                        % shift by -4
[x13,n13] = sigadd(2*x11,n11,3*x12,n12); % add two      sequences
[x1,n1] = sigadd(x13,n13,-x,n);                     % add two sequences

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0204a');
Hs = stem(n1,x1,'filled'); set(Hs,'markersize',2);
axis([min(n1)-1,max(n1)+1,min(x1)-3,max(x1)+1]);
xlabel('n','FontSize',LFS);
ylabel('x_1(n)','FontSize',LFS);
title('Sequence x_1(n)','FontSize',TFS); ntick = n1; ytick =
[min(x1)-3:5:max(x1)+1];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick); print -deps2
../CHAP2_EPSFILES/P0204a;
```

The plots of x_1.n/ is shown in Figure 2.17.

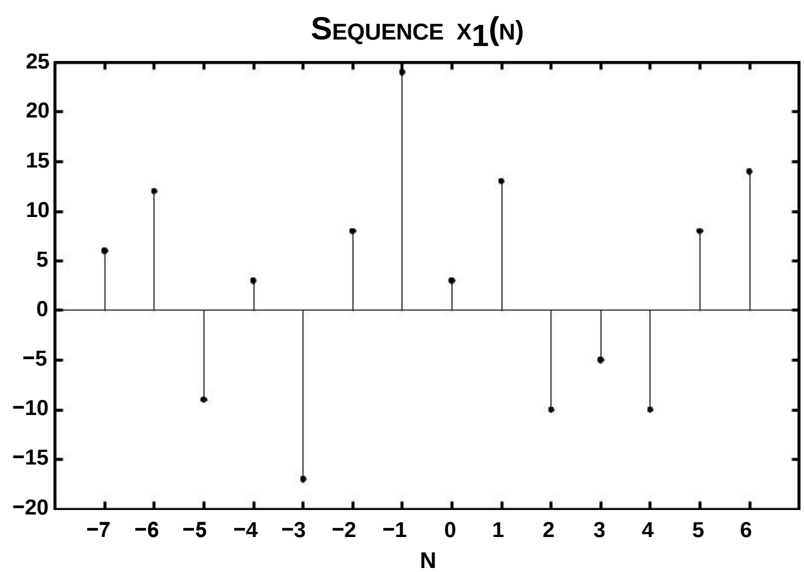

Figure 2.17: Problem P2.4.1 sequence plot

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 18

2. x_2 .n/ D 4x.4 C n/ C 5x.n C 5/ C 2x.n/

```
% P0204b: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x2(n) = 4x(4 + n) + 5x(n + 5) + 2x(n)
clc; close all;
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0204b'); n = [-3:3]; x
= [2,4,-3,1,-5,4,7];

[x21,n21] = sigshift(x,n,-4);                          % shift by -4
[x22,n22] = sigshift(x,n,-5);                          % shift by -5
[x23,n23] = sigadd(4*x21,n21,5*x22,n22); % add two sequences
[x2,n2] = sigadd(x23,n23,2*x,n);                 % add two sequences

Hs = stem(n2,x2,'filled'); set(Hs,'markersize',2); axis([min(n2)-
1,max(n2)+1,min(x2)-4,max(x2)+6]); xlabel('n','FontSize',LFS);
ylabel('x_2(n)','FontSize',LFS); title('Sequence x_2(n)','FontSize',TFS);
ntick = n2; ytick = [-25 -20:10:60 65];

set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../CHAP2_EPSFILES/P0204b;
```

The plots of x_2.n/ is shown in Figure 2.18.

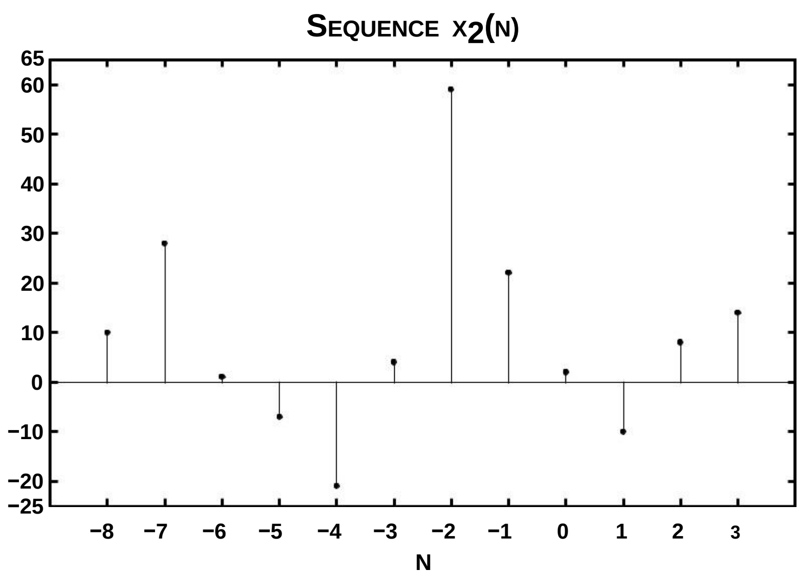

Figure 2.18: Problem P2.4.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 19

3. $x_3(n) = x(n+3)x(n-2) + x(1-n)x(n+1)$

```
% P0204c: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
%         x3(n) = x(n + 3)x(n - 2) + x(1 - n)x(n + 1) clc; close
all;

n = [-3:3]; x = [2,4,-3,1,-5,4,7];             % given sequence x(n)
[x31,n31] = sigshift(x,n,-3);                  % shift sequence by -3
[x32,n32] = sigshift(x,n,2);                   % shift sequence by 2
[x33,n33] = sigmult(x31,n31,x32,n32);          % multiply 2 sequences
[x34,n34] = sigfold(x,n);                      % fold x(n)
[x34,n34] = sigshift(x34,n34,1);               % shift x(-n) by 1
[x35,n35] = sigshift(x,n,-1);                  % shift x(n) by -1
[x36,n36] = sigmult(x34,n34,x35,n35);          % multiply 2 sequences
[x3,n3] = sigadd(x33,n33,x36,n36);             % add 2 sequences

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0204c'); Hs =
stem(n3,x3,'filled'); set(Hs,'markersize',2); axis([min(n3)-
1,max(n3)+1,min(x3)-10,max(x3)+10]); xlabel('n','FontSize',LFS);
ylabel('x_3(n)','FontSize',LFS); title('Sequence x_3(n)','FontSize',TFS);
ntick = n3; ytick = [-30:10:60];

set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../CHAP2_EPSFILES/P0204c;
```

The plots of $x_3(n)$ is shown in Figure 2.19.

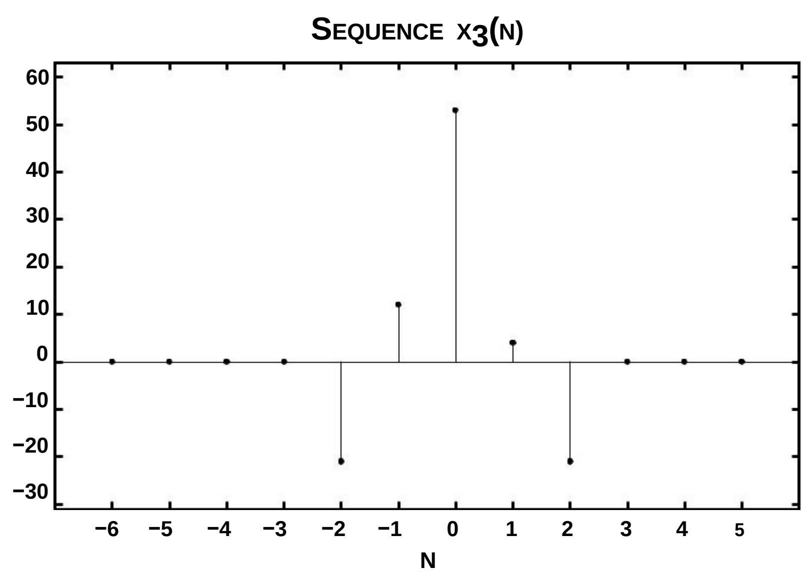

Figure 2.19: Problem P2.4.3 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 20

4. $x_4(n) = 2e^{0.5n}x(n) + \cos(0.1\pi n)\, x(n+2);\ -10 \le n \le 10$

```
% P0204d: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
%             x4(n) = 2*e^{0.5n}*x(n)+cos(0.1*pi*n)*x(n+2), -10 <=n< =10 clc;
close all;

n = [-3:3]; x = [2,4,-3,1,-5,4,7];                    % given sequence x(n)
n4 = [-10:10]; x41 = 2*exp(0.5*n4); x412 = cos(0.1*pi*n4);
[x42,n42] = sigmult(x41,n4,x,n);
[x43,n43] = sigshift(x,n,-2);
[x44,n44] = sigmult(x412,n42,x43,n43);
  [x4,n4] = sigadd(x42,n42,x44,n44);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0204d'); Hs =
stem(n4,x4,'filled'); set(Hs,'markersize',2); axis([min(n4)-
1,max(n4)+1,min(x4)-11,max(x4)+10]); xlabel('n','FontSize',LFS);
ylabel('x_4(n)','FontSize',LFS); title('Sequence x_4(n)','FontSize',TFS);
ntick = n4; ytick = [ -20:10:70];

set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../CHAP2_EPSFILES/P0204d;
```

The plot of $x_4(n)$ is shown in Figure 2.20.

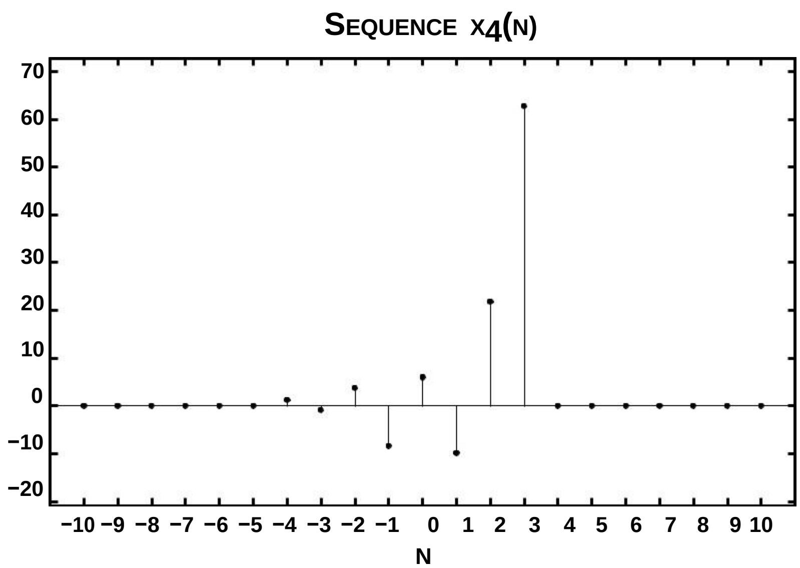

Figure 2.20: Problem P2.4.4 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

 SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 21

P2.5 The complex exponential sequence $e^{j\omega_0 n}$ or the sinusoidal sequence $\cos(\omega_0 n)$ are periodic if the normalized frequency $f_0 \triangleq \frac{\omega_0}{2\pi}$ is a rational number; that is, $f_0 = \frac{K}{N}$, where K and N are integers.

1. Analytical proof: The exponential sequence is periodic if

$$e^{j2\pi f_0(n+N)} = e^{j2\pi f_0 n} \text{ or } e^{j2\pi f_0 N} = 1 \Rightarrow f_0 N = K \text{ (an integer)}$$

which proves the result.

2. $x_1 = \exp(0.1\pi n)$, $-100 \le n \le 100$.

```
% P0205b: x1(n) = e^{0.1*j*pi*n} -100 <=n <=100 clc;
close all;
n1 = [-100:100]; x1 = exp(0.1*j*pi*n1);
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0205b'); subplot(2,1,1); Hs1 =
stem(n1,real(x1),'filled'); set(Hs1,'markersize',2); axis([min(n1)-
5,max(n1)+5,min(real(x1))-1,max(real(x1))+1]); xlabel('n','FontSize',LFS);
ylabel('Real(x_1(n))','FontSize',LFS); title(['Real part of sequence x_1(n) = ' ...

        'exp(0.1 \times j \times pi \times n) ' char(10) ...
        ' Period = 20, K = 1, N = 20'],'FontSize',TFS);
ntick = [n1(1):20:n1(end)]; set(gca,'XTickMode','manual','XTick',ntick); subplot(2,1,2);
Hs2 = stem(n1,imag(x1),'filled'); set(Hs2,'markersize',2); axis([min(n1)-
5,max(n1)+5,min(real(x1))-1,max(real(x1))+1]); xlabel('n','FontSize',LFS);
ylabel('Imag(x_1(n))','FontSize',LFS); title(['Imaginary part of sequence x_1(n) = ' ...

        'exp(0.1 \times j \times pi \times n) ' char(10) ...
        ' Period = 20, K = 1, N = 20'],'FontSize',TFS);
ntick = [n1(1):20:n1(end)]; set(gca,'XTickMode','manual','XTick',ntick); print -deps2
../CHAP2_EPSFILES/P0205b; print -deps2 ../../Latex/P0205b;
```

The plots of $x_1(n)$ is shown in Figure 2.21. Since $f_0 = 0.1/2 = 1/20$ the sequence is periodic. From the plot in Figure 2.21 we see that in one period of 20 samples $x_1(n)$ exhibits cycle. This is true whenever K and N are relatively prime.

3. $x_2 = \cos(0.1n)$, $-20 \le n \le 20$.

```
% P0205c: x2(n) = cos(0.1n), -20 <= n <= 20 clc;
close all;
n2 = [-20:20]; x2 = cos(0.1*n2);
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0205c'); Hs =
stem(n2,x2,'filled'); set(Hs,'markersize',2); axis([min(n2)-
1,max(n2)+1,min(x2)-1,max(x2)+1]); xlabel('n','FontSize',LFS);
ylabel('x_2(n)','FontSize',LFS); title(['Sequence x_2(n) = cos(0.1 \times
n)' char(10) ...
        'Not periodic since f_0 = 0.1 / (2 \times \pi)' ...
        ' is not a rational number'], 'FontSize',TFS);
ntick = [n2(1):4:n2(end)]; set(gca,'XTickMode','manual','XTick',ntick); print -deps2
../CHAP2_EPSFILES/P0205c;
```

The plots of $x_1(n)$ is shown in Figure 2.22. In this case f_0 is not a rational number and hence the sequence $x_2(n)$ is not periodic. This can be clearly seen from the plot of $x_2(n)$ in Figure 2.22.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Solutions Manual for DSP using Matlab (4th Edition) 22

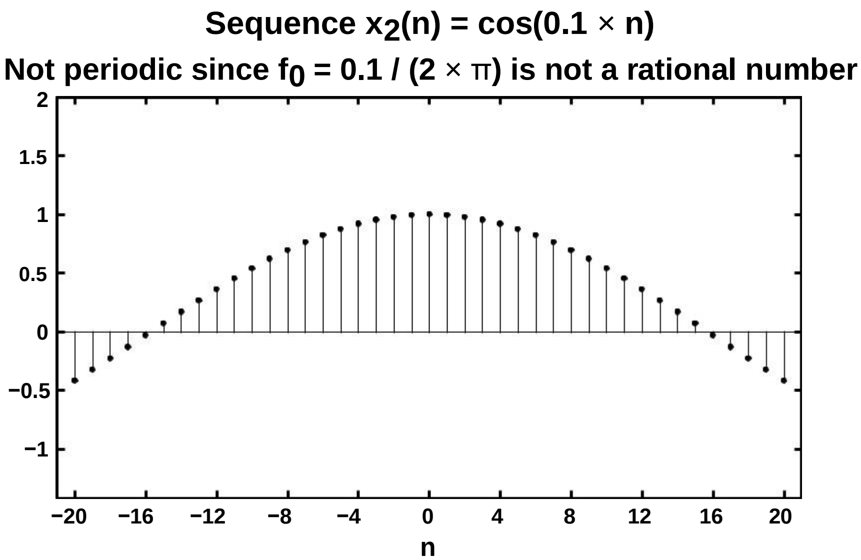

Figure 2.21: Problem P2.5.2 sequence plots

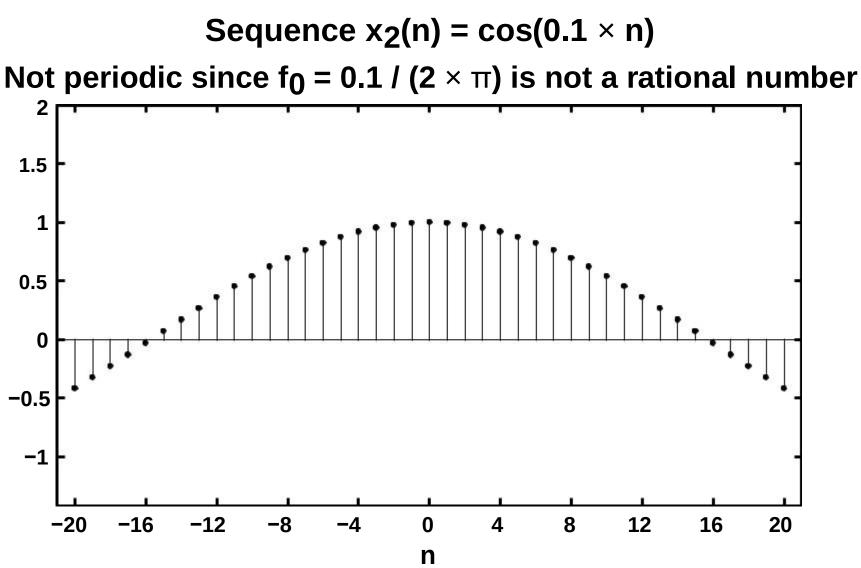

Figure 2.22: Problem P2.5.3 sequence plots

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 23

P2.6 Using the evenodd function decompose the following sequences into their even and odd components. Plot these compo-nents using the stem function.

1. $x_1(n) = \{\underset{\uparrow}{0}, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

```
% P0206a: % Even odd decomposition of x1(n) = [0 1 2 3 4 5 6 7 8 9];
% n = 0:9; clc; close all;

x1 = [0 1 2 3 4 5 6 7 8 9]; n1          = [0:9]; [xe1,xo1,m1] = evenodd(x1,n1);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0206a'); subplot(2,1,1); Hs
= stem(m1,xe1,'filled'); set(Hs,'markersize',2); axis([min(m1)-
1,max(m1)+1,min(xe1)-1,max(xe1)+1]); xlabel('n','FontSize',LFS);
ylabel('x_e(n)','FontSize',LFS); title('Even part of x_1(n)','FontSize',TFS); ntick
= [m1(1):m1(end)]; ytick = [-1:5];

set(gca,'XTick',ntick);set(gca,'YTick',ytick);
subplot(2,1,2); Hs = stem(m1,xo1,'filled'); set(Hs,'markersize',2);
axis([min(m1)-1,max(m1)+1,min(xo1)-2,max(xo1)+2]);
xlabel('n','FontSize',LFS); ylabel('x_o(n)','FontSize',LFS); title('Odd part of
x_1(n)','FontSize',TFS); ntick = [m1(1):m1(end)]; ytick = [-6:2:6];

set(gca,'XTick',ntick);set(gca,'YTick',ytick);
print -deps2 ../CHAP2_EPSFILES/P0206a; print -deps2 ../../Latex/P0206a;
```

The plots of $x_1(n)$ is shown in Figure 2.23.

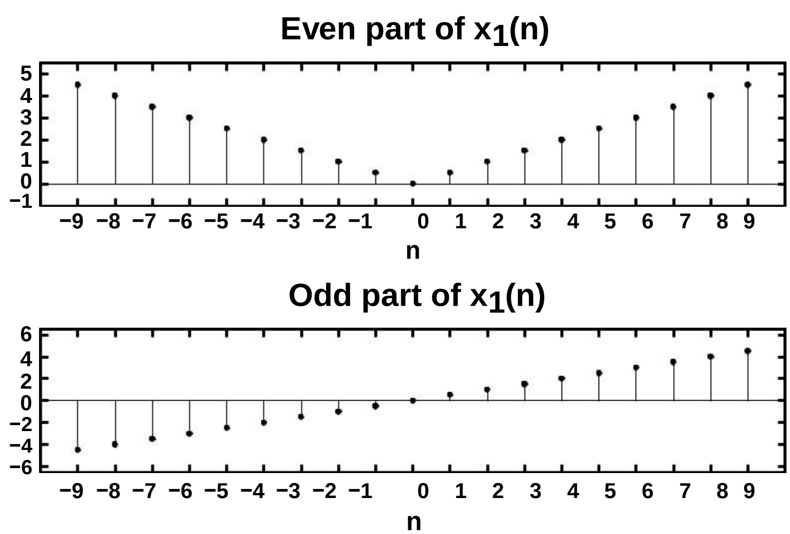

Figure 2.23: Problem P2.6.1 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 24

2. $x_2(n) = e^{0.1n}\left[u(n+5) - u(n-10)\right]$.

```
% P0206b: Even odd decomposition of x2(n) = e ^ {0.1n} [u(n + 5) - u(n - 10)]; clc; close all;

n2 = [-8:12]; x2 = exp(0.1*n2).*(stepseq(-5,-8,12) - stepseq(10,-8,12)); [xe2,xo2,m2]
= evenodd(x2,n2);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0206b'); subplot(2,1,1); Hs
= stem(m2,xe2,'filled'); set(Hs,'markersize',2); axis([min(m2)-
1,max(m2)+1,min(xe2)-1,max(xe2)+1]); xlabel('n','FontSize',LFS);
ylabel('x_e(n)','FontSize',LFS); title('Even part of x_2(n) = exp(0.1n) [u(n + 5) -
u(n - 10)]',...
        'FontSize',TFS);
ntick = [m2(1):2:m2(end)]; set(gca,'XTick',ntick);
subplot(2,1,2); Hs = stem(m2,xo2,'filled'); set(Hs,'markersize',2);
axis([min(m2)-1,max(m2)+1,min(xo2)-1,max(xo2)+1]);
xlabel('n','FontSize',LFS); ylabel('x_o(n)','FontSize',LFS); title('Odd part of
x_2(n) = exp(0.1n) [u(n + 5) - u(n - 10)]',...
        'FontSize',TFS);
ntick = [m2(1) :2:m2(end)]; set(gca,'XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0206b; print -deps2 ../../Latex/P0206b;
```

The plots of $x_2(n)$ is shown in Figure 2.24.

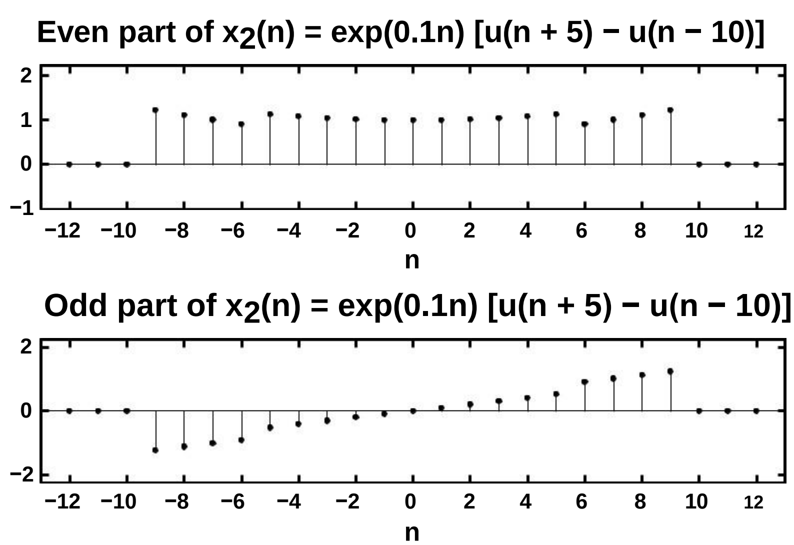

Figure 2.24: Problem P2.6.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 25

3. $x_3(n) = \cos(0.2\pi n + \pi/4)$; $-20 \le n \le 20$.

```
% P0206c: Even odd decomposition of x2(n) = cos(0.2*pi*n + pi/4);
% -20 <= n <= 20; clc; close all;

n3 = [-20:20]; x3 = cos(0.2*pi*n3 + pi/4);
[xe3,xo3,m3] = evenodd(x3,n3);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0206c'); subplot(2,1,1); Hs =
stem(m3,xe3,'filled'); set(Hs,'markersize',2); axis([min(m3)-2,max(m3)+2,min(xe3)-
1,max(xe3)+1]); xlabel('n','FontSize',LFS); ylabel('x_e(n)','FontSize',LFS); title('Even
part of x_3(n) = cos(0.2 \times \pi \times n + \pi/4)',...

        'FontSize',TFS);
ntick = [m3(1):4:m3(end)]; set(gca,'XTick',ntick);
subplot(2,1,2); Hs = stem(m3,xo3,'filled'); set(Hs,'markersize',2); axis([min(m3)-
2,max(m3)+2,min(xo3)-1,max(xo3)+1]); xlabel('n','FontSize',LFS);
ylabel('x_o(n)','FontSize',LFS); title('Odd part of x_3(n) = cos(0.2 \times \pi \times n
+ \pi/4)',...
        'FontSize',TFS);
ntick = [m3(1):4 :m3(end)]; set(gca,'XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0206c; print -deps2 ../../Latex/P0206c;
```

The plots of $x_3(n)$ is shown in Figure 2.25.

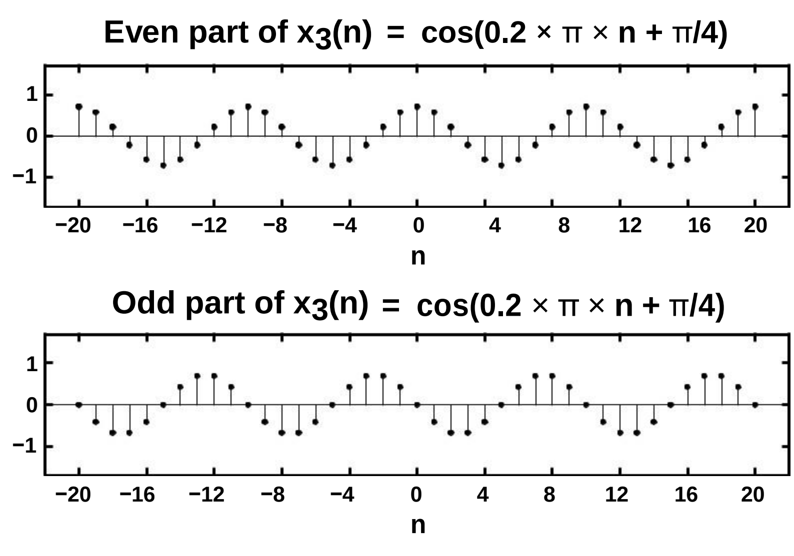

Figure 2.25: Problem P2.6.3 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 26

4. $x_4(n) = e^{-0.05n} \sin(0.1\pi n + \pi/3),\ 0 \le n \le 100.$

```
% P0206d: x4(n) = e ^ {-0.05*n}*sin(0.1*pi*n + pi/3), 0 <= n <= 100 clc; close
all;

n4 = [0:100]; x4 = exp(-0.05*n4).*sin(0.1*pi*n4 + pi/3);
[xe4,xo4,m4] = evenodd(x4,n4);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0206d'); subplot(2,1,1); Hs
= stem(m4,xe4,'filled'); set(Hs,'markersize',2); axis([min(m4)-
10,max(m4)+10,min(xe4)-1,max(xe4)+1]); xlabel('n','FontSize',LFS);
ylabel('x_e(n)','FontSize',LFS); title(['Even part of x_4(n) = ' ...

        'exp(-0.05 \times n) \times sin(0.1 \times \pi \times n + ' ...
        '\pi/3)'],'FontSize',TFS);
ntick = [m4(1):20:m4(end)]; set(gca,'XTick',ntick);

subplot(2,1,2); Hs = stem(m4,xo4,'filled'); set(Hs,'markersize',2);
axis([min(m4)-10,max(m4)+10,min(xo4)-1,max(xo4)+1]);
xlabel('n','FontSize',LFS); ylabel('x_o(n)','FontSize',LFS); title(['Odd part of
x_4(n) = ' ...
        'exp(-0.05 \times n) \times sin(0.1 \times \pi \times n + ' ...
        '\pi/3)'],'FontSize',TFS);
ntick = [m4(1):20 :m4(end)]; set(gca,'XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0206d; print -deps2 ../../Latex/P0206d;
```

The plots of $x_1(n)$ are shown in Figure 2.26.

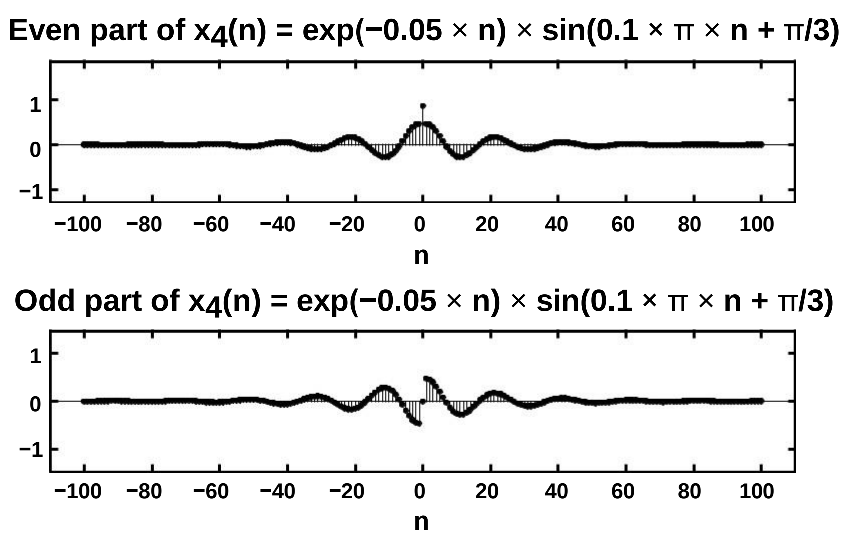

Figure 2.26: Problem P2.6.1 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 27

P2.7 A complex-valued sequence $x_e(n)$ is called conjugate-symmetric if $x_e(n) = x_e^*(-n)$ and a complex-valued sequence $x_o(n)$ is called conjugate-antisymmetric if $x_o(n) = -x_o^*(-n)$. Then any arbitrary complex-valued sequence $x(n)$ can be decomposed into $x(n) = x_e(n) + x_o(n)$ where $x_e(n)$ and $x_o(n)$ are given by

$$x_e(n) = \frac{1}{2}\left[x(n) + x^*(-n)\right] \quad \text{and} \quad x_o(n) = \frac{1}{2}\left[x(n) - x^*(-n)\right] \tag{2.1}$$

respectively.

1. Modify the evenodd function discussed in the text so that it accepts an arbitrary sequence and decomposes it into its conjugate-symmetric and conjugate-antisymmetric components by implementing (2.1).

```
function [xe , xo , m] = evenodd_c(x , n)
% Complex-valued signal decomposition into even and odd parts (version-2)
% ----------------------------------------------------------------------
%[xe , xo , m] = evenodd_c(x , n);
%
[xc , nc] = sigfold(conj(x) , n);
[xe , m] = sigadd(0.5 * x , n , 0.5 * xc , nc);
[xo , m] = sigadd(0.5 * x , n , -0.5 * xc , nc);
```

2. $x(n) = 10\exp[(-0.1 + j0.2\pi)n], \quad 0 \le n \le 10$

```
% P0207b: Decomposition of x(n) = 10*e ^ {(-0.1 + j*0.2*pi)*n},
%                                                0 < = n <= 10
% into its conjugate symmetric and conjugate antisymmetric parts. clc; close
all;

n = [0:10]; x = 10*exp((-0.1+j*0.2*pi)*n); [xe,xo,neo] = evenodd(x,n); Re_xe =
real(xe); Im_xe = imag(xe); Re_xo = real(xo); Im_xo = imag(xo);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0207b'); subplot(2,2,1);
Hs = stem(neo,Re_xe); set(Hs,'markersize',2);
ylabel('Re[x_e(n)]','FontSize',LFS); xlabel('n','FontSize',LFS);
axis([min(neo)-1,max(neo)+1,-5,12]); ytick = [-5:5:15]; set(gca,'YTick',ytick);

title(['Real part of' char(10) 'even sequence x_e(n)'],'FontSize',TFS);

subplot(2,2,3); Hs = stem(neo,Im_xe); set(Hs,'markersize',2);
ylabel('Im[x_e(n)]','FontSize',LFS); xlabel('n','FontSize',LFS);
axis([min(neo)-1,max(neo)+1,-5,5]); ytick = [-5:1:5]; set(gca,'YTick',ytick);

title(['Imaginary part of' char(10) 'even sequence x_e(n)'],'FontSize',TFS);

subplot(2,2,2); Hs = stem(neo,Re_xo); set(Hs,'markersize',2);
ylabel('Re[x_o(n)]','FontSize',LFS); xlabel('n','FontSize',LFS);
axis([min(neo)-1,max(neo)+1,-5,+5]); ytick = [-5:1:5]; set(gca,'YTick',ytick);

title(['Real part of' char(10) 'odd sequence x_o(n)'],'FontSize',TFS);

subplot(2,2,4); Hs = stem(neo,Im_xo); set(Hs,'markersize',2);
ylabel('Im[x_o(n)]','FontSize',LFS); xlabel('n','FontSize',LFS);
axis([min(neo)-1,max(neo)+1,-5,5]); ytick = [-5:1:5]; set(gca,'YTick',ytick);

title(['Imaginary part of' char(10) 'odd sequence x_o(n)'],'FontSize',TFS); print -deps2
../EPSFILES/P0207b;%print -deps2 ../../Latex/P0207b;
```

The plots of $x(n)$ are shown in Figure 2.27.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 28

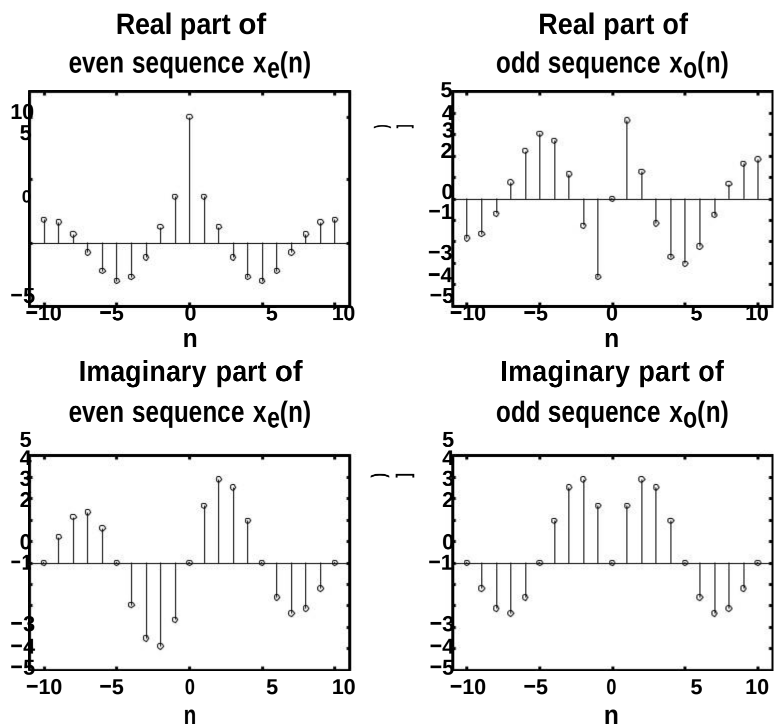

Figure 2.27: Problem P2.7.2 sequence plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

 SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 29

P2.8 The operation of signal dilation (or decimation or down-sampling) is defined by y.n/ D x.nM / in which the sequence x.n/ is down-sampled by an integer factor M .

1. MATLAB function:

```
function [y,m] = dnsample(x,n,M)
% [y,m] = dnsample(x,n,M)
% Downsample sequence x(n) by a factor M to obtain y(m) mb =
ceil(n(1)/M)*M; me = floor(n(end)/M)*M;
nb = find(n==mb); ne = find(n==me);
y = x(nb:M:ne); m = fix((mb:M:me)/M);
```

2. x_1.n/ D sin.0:125 n/; 50 n 50. Decimation by a factor of 4.

```
% P0208b: x1(n) = sin(0.125*pi*n),-50 <= n <= 50
%              Decimate x(n) by a factor of 4 to obtain y(n) clc;
close all;
n1 = [-50:50]; x1 = sin(0.125*pi*n1); [y1,m1] = dnsample(x1,n1,4); Hf_1 =
figure; set(Hf_1,'NumberTitle','off','Name','P0208b'); subplot(2,1,1); Hs =
stem(n1,x1); set(Hs,'markersize',2); xlabel('n','FontSize',LFS);
ylabel('x(n)','FontSize',LFS); title('Original sequence x_1(n)','FontSize',TFS);
axis([min(n1)-5,max(n1)+5,min(x1)-0.5,max(x1)+0.5]);

ytick = [-1.5:0.5:1.5]; ntick = [n1(1):10:n1(end)]; set(gca,'XTick',ntick);
set(gca,'YTick',ytick); subplot(2,1,2); Hs = stem(m1,y1);
set(Hs,'markersize',2); xlabel('n','FontSize',LFS); ylabel('y(n) =
x(4n)','FontSize',LFS);
title('y_1(n)  =  Original sequence x_1(n) decimated by a factor of 4',...
          'FontSize',TFS); axis([min(m1)-2,max(m1)+2,min(y1)-
0.5,max(y1)+0.5]); ytick = [-1.5:0.5:1.5]; ntick =
[m1(1):2:m1(end)]; set(gca,'XTick',ntick);
set(gca,'YTick',ytick); print -deps2
../CHAP2_EPSFILES/P0208b;
```

The plots of x_1.n/ and y_1.n/ are shown in Figure 2.28. Observe that the original signal x_1.n/ can be recovered.

3. x.n/ D sin.0:5 n/; 50 n 50. Decimation by a factor of 4.

```
% P0208c: x2(n) = sin(0.5*pi*n),-50 <= n <= 50
%              Decimate x2(n) by a factor of 4 to obtain y2(n) clc;
close all;
n2 = [-50:50]; x2 = sin(0.5*pi*n2); [y2,m2] = dnsample(x2,n2,4); Hf_1 =
figure; set(Hf_1,'NumberTitle','off','Name','P0208c'); subplot(2,1,1); Hs =
stem(n2,x2); set(Hs,'markersize',2); xlabel('n','FontSize',LFS);
ylabel('x(n)','FontSize',LFS); axis([min(n2)-5,max(n2)+5,min(x2)-
0.5,max(x2)+0.5]); title('Original sequence x_2(n)','FontSize',TFS);

ytick = [-1.5:0.5:1.5]; ntick = [n2(1):10:n2(end)]; set(gca,'XTick',ntick);
set(gca,'YTick',ytick); subplot(2,1,2); Hs = stem(m2,y2);
set(Hs,'markersize',2); xlabel('n','FontSize',LFS); ylabel('y(n) =
x(4n)','FontSize',LFS); axis([min(m2)-1,max(m2)+1,min(y2)-1,max(y2)+1]);

title('y_2(n) = Original sequence x_2(n) decimated by a factor of 4',...
          'FontSize',TFS);
ntick = [m2(1):2:m2(end)]; set(gca,'XTick',ntick);
print -deps2 ../CHAP2_EPSFILES/P0208c; print -deps2 ../../Latex/P0208c;
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 30

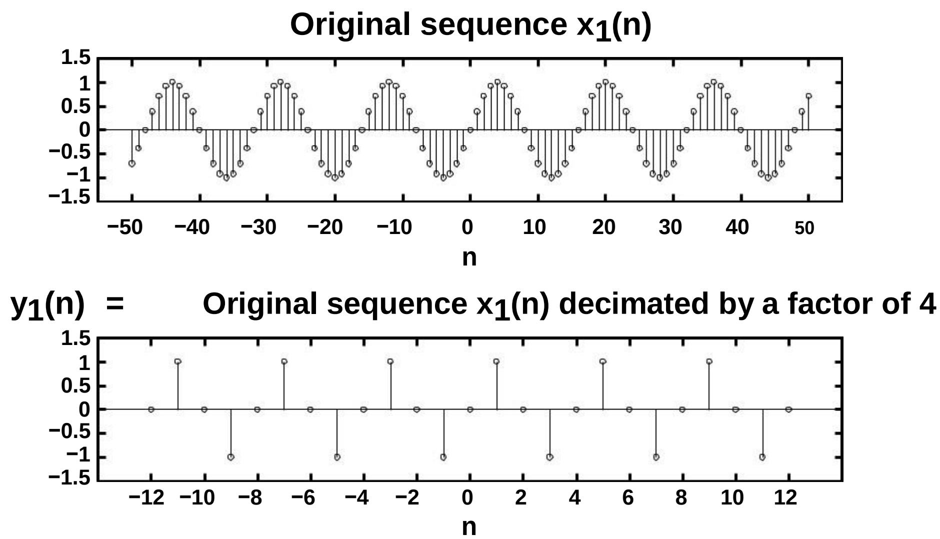

Figure 2.28: Problem P2.8.2 sequence plot

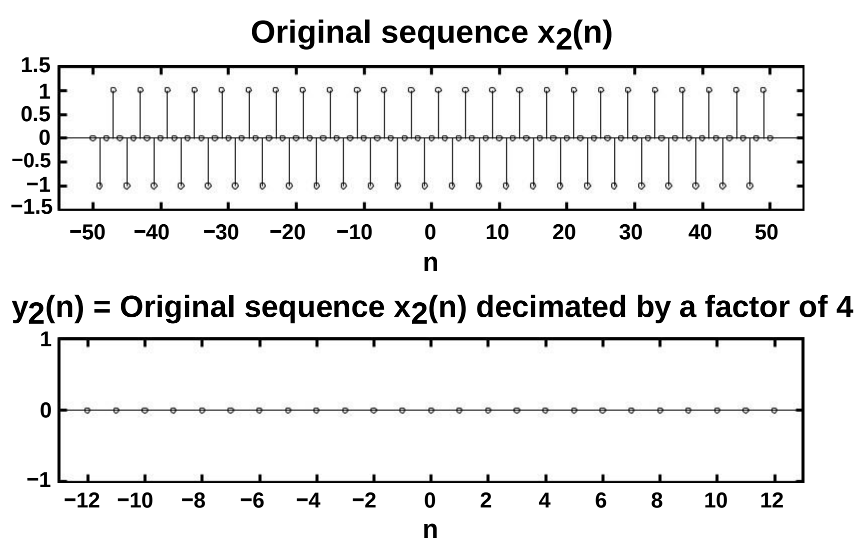

Figure 2.29: Problem P2.8.3 sequence plot

The plots of x_2.n/ and y_2.n/ are shown in Figure 2.29. Observe that the downsampled signal is a signal with zero frequency. Thus the original signal x_2.n/ is lost.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 31

P2.9 The autocorrelation sequence r_{xx}.‘/ and the crosscorrelation sequence r_{xy} .‘/ for the sequences:

x.n/ D .0:9/^{n} ; 0 n 20I y.n/ D .0:8/^{n} ; 20 n 0

```
% P0209a: autocorrelation of sequence x(n) = 0.9 ^ n,  0 <= n <= 20
% using the conv_m function
clc; close all;

nx = [0:20]; x = 0.9 .^ nx; [xf,nxf] = sigfold(x,nx); [rxx,nrxx] =
conv_m(x,nx,xf,nxf);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0209a'); Hs =
stem(nrxx,rxx); set(Hs,'markersize',2); xlabel('n','FontSize',LFS);
ylabel('r_x_x(n)','FontSize',LFS); title('Autocorrelation of
x(n)','FontSize',TFS); axis([min(nrxx)-
1,max(nrxx)+1,min(rxx),max(rxx)+1]);
ntick = [nrxx(1):4:nrxx(end)]; set(gca,'XTick',ntick); %print -
deps2 ../CHAP2_EPSFILES/P0209a;
```

The plot of the autocorrelation is shown in Figure 2.30.

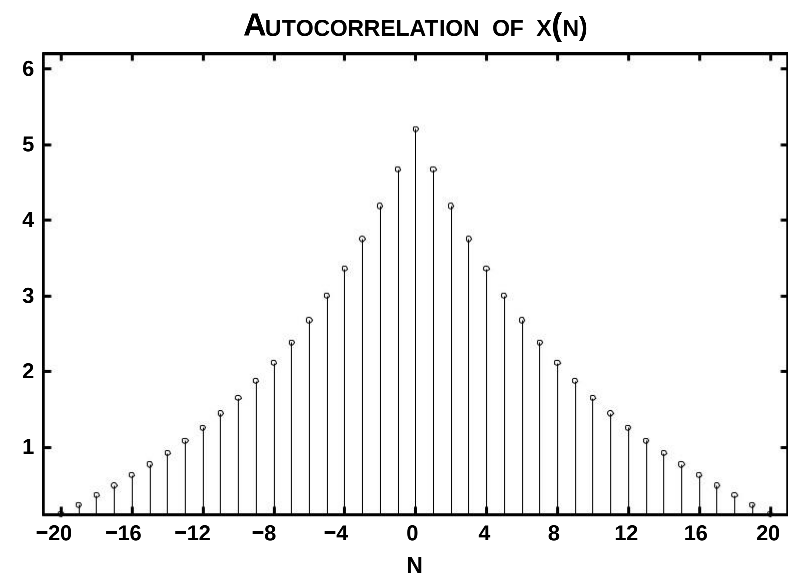

Figure 2.30: Problem P2.9 autocorrelation plot

```
% P0209b: crosscorrelation of sequence x(n) = 0.9 ^ n,  0 <= n <= 20
%            with sequence y = 0.8.^n, -20 <=n <= 0 using the conv_m function clc; close
all;

nx = [0:20]; x = 0.9 .^ nx; ny = [-20:0]; y = 0.8 .^ (-ny); [yf,nyf] =
sigfold(y,ny); [rxy,nrxy] = conv_m(x,nx,yf,nyf);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0209b'); Hs =
stem(nrxy,rxy); set(Hs,'markersize',3); xlabel('n','FontSize',LFS);
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 32

```
ylabel('r_x_y(n)','FontSize',LFS); title('Crosscorrelation of x(n) and
y(n)','FontSize',TFS);
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 33

```
axis([min(nrxy)-1,max(nrxy)+1,floor(min(rxy)),ceil(max(rxy))]); ytick =
[0:1:4]; ntick = [nrxy(1):2:nrxy(end)]; set(gca,'XTick',ntick);
set(gca,'YTick',ytick); print -deps2 ../epsfiles/P0209b;
```

The plot of the crosscorrelation is shown in Figure 2.31.

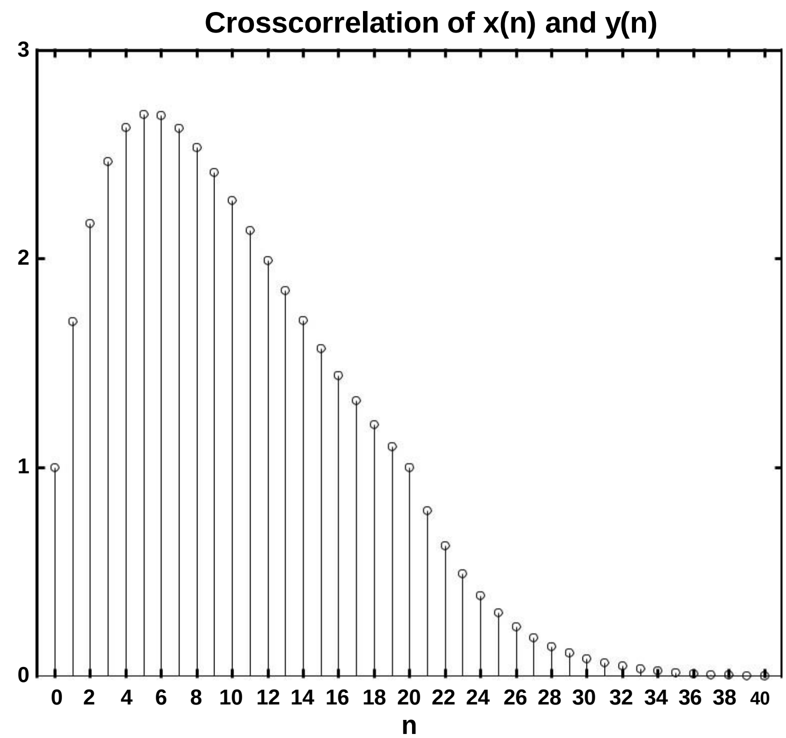

Figure 2.31: Problem P2.9 crosscorrelation plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 34

P2.10 In a certain concert hall, echoes of the original audio signal x.n/ are generated due to the reflections at the walls and ceiling. The audio signal experienced by the listener y.n/ is a combination of x.n/ and its echoes. Let y.n/ D x.n/ C x.n k/ where k is the amount of delay in samples and is its relative strength. We want to estimate the delay using the correlation analysis.

1. Determine analytically the autocorrelation r_{yy} .‘/ in terms of the autocorrelation r_{xx}.‘/. Consider the autocorrelation r_{yy} .‘/ of y.n/:

r_{yy} .‘/ D X‘ y.n/y.n ‘/ D X‘ x.n/ C x.n k/ x.n ‘/ C x.n k ‘/

D X‘ x.n/x.n ‘/C X‘ x.n/x.n k ‘/C X‘ x.n k/x.n ‘/

C ^{2}X‘ x.n k/x.n k ‘/

D r_{xx}.‘/ C r_{xx}.‘ C k/ C r_{xx}.‘ k/ C $^2 r_{xx}$.‘/

D 1 C ^{2} r_{xx}.‘/ C r_{xx}.‘ C k/ C r_{xx}.‘ k/

2. Let x.n/ D cos.0:2 n/ C 0:5 cos.0:6 n/, D 0:1, and k D 50. Generate 200 samples of y.n/ and determine its autocorrelation. Can you obtain and k by observing r_{yy} .‘/?

MATLAB script:

```
% P0210c: autocorrelation of sequence y(n) = x(n) + alpha*x(n - k)
%                                             alpha = 0.1,k = 50
% x(n) = cos(0.2*pi*n) + 0.5*cos(0.6*pi*n) clc;
close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0210c'); alpha =
.1;
n = [-100:100];
%x = 3*rand(1,201)-1.5;
x = cos(0.2*pi*n) + 0.5*cos(0.6*pi*n); [xf,nxf]
= sigfold(x,n);
[rxx,nrxx] = conv_m(x,n,xf,nxf);

y = filter([1 zeros(1,49) alpha], 1, x);
[yf,nyf] = sigfold(y,n);
[ryx,nryx] = conv_m(y,n,xf,nxf);
[ryy,nryy] = conv_m(y,n,yf,nyf);

subplot(2,1,1);
Hs = stem(nrxx,rxx,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',LFS);
ylabel('r_y_y(n)','FontSize',LFS);
title('autocorrelation of sequence x(n)','FontSize',TFS);
axis([-210, 210,-200,200]);
ntick = [-200:50:200]; ytick = [-200:50:200];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 35

```
subplot(2,1,2);
Hs = stem(nryx,ryx,'filled');
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 36

```
set(Hs,'markersize',2);
xlabel('n','FontSize',LFS);
ylabel('r_y_y(n)','FontSize',LFS);
title('autocorrelation of sequence y(n) = x(n) + 0.1 \times x(n - 50)' ,...
        'FontSize',TFS);
axis([-210,210,-200,200]);
ntick = [-200:50:200];
ytick = [-200:50:200];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);

print -deps2 ../CHAP2_EPSFILES/P0210b;
```

Stem plots of the autocorrelations are shown in Figure 2.32. We observe that it is not possible to visually separate components of r_{xx}.'/ In r_{yy} .'/ and hence it is not possible to estimate or k.

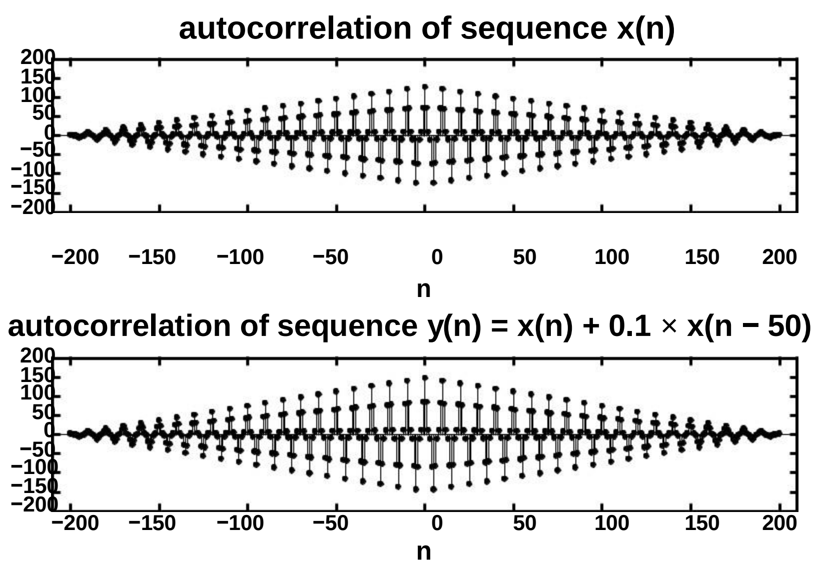

Figure 2.32: Problem P2.10 autocorrelation plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 37

P2.11 Linearity of discrete-time systems.

System-1: T_1 x.n/ D x.n/u.n/

1. Analytic determination of linearity:

$$T_1\ a_1x_1.n/\ C\ a_2x_2.n/\ D\ fa_1x_1.n/\ C\ a_2x_2.n/gu.n/\ D\ a_1x_1.n/u.n/\ C\ a_2x_2.n/u.n/$$
$$D\ a_1T_1\ x_1.n/\ C\ a_2T_1\ x_2.n/$$

   Hence the system T_1 x.n/ is linear.

2. MATLAB script:

```
% P0211a: To prove that the system T1[x(n)] = x(n)u(n) is linear clear; clc;
close all;

n = 0:100; x1 = rand(1,length(n));
x2 = sqrt(10)*randn(1,length(n)); u = stepseq(0,0,100);
y1 = x1.*u; y2 = x2.*u; y = (x1 + x2).*u;
diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
    disp(' *** System-1 is Linear *** ');
else
    disp(' *** System-1 is NonLinear *** ');
end
```

   MATLAB verification:

```
>> *** System-1 is Linear ***
```

System-2: T_2 x.n/ D x.n/ C n x.n C 1/

1. Analytic determination of linearity:

$$T_2\ a_1x_1.n/\ C\ a_2x_2.n/\ D\ fa_1x_1.n/\ C\ a_2x_2.n/g\ C\ n\ fa_1x_1.n\ C\ 1/\ C\ a_2x_2.n\ C\ 1/g$$
$$D\ \ a_1fx_1.n/\ C\ n\ x_1.n\ C\ 1/g\ C\ a_2fx_2.n/\ C\ n\ x_2.n\ C\ 1/g$$
$$D\ \ a_1T_2\ x_1.n/\ C\ a_2T_2\ x_2.n/$$

   Hence the system is T_2 x.n/ linear.

2. MATLAB script:

```
% P0211b: To prove that the system T2[x(n)] = x(n) + n*x(n+1) is linear clear; clc;
close all;

n = 0:100; x1 = rand(1,length(n)); x2 = sqrt(10)*randn(1,length(n));
z   = n; [x11,nx11] = sigshift(x1,n,-1);
[x111,nx111] = sigmult(z,n,x11,nx11); [y1,ny1] = sigadd(x1,n,x111,nx111);

[x21,nx21] = sigshift(x2,n,-1); [x211,nx211] = sigmult(z,n,x21,nx21); [y2,ny2] =
sigadd(x2,n,x211,nx211);
xs = x1 + x2; [xs1,nxs1] = sigshift(xs,n,-1);
[xs11,nxs11] = sigmult(z,n,xs1,nxs1); [y,ny] = sigadd(xs,n,xs11,nxs11);
diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
    disp(' *** System-2 is Linear *** ');
else
    disp(' *** System-2 is NonLinear *** ');
end
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 38

MATLAB verification:

```
>> *** System-2 is Linear ***
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 39

System-3: T_3 x.n/ D x.n/ C $\frac{1}{2}$x.n 2/ $\frac{1}{3}$ x.n 3/x.2n/

1. Analytic determination of linearity:

T_3 a_1x_1.n/ C a_2x_2.n/ D a_1x_1.n/ C a_2x_2.n/ C $\frac{1}{2}$ fa_1x_1.n 2/ C a_2x_2.n 2/g

C $\frac{1}{3}$ fa_1x_1.n 3/ C a_2x_2.n 3/gfa_1x_1.2n/ C a_2x_2.2n/g

D a_1 x_1.n/ C $\frac{1}{2}$ x_1.n 2/ $\frac{1}{3}$ a_1x_1.n 3/x_1.2n/ g

a x .n/ $\frac{1}{}$ x .n 2/ C $\frac{1}{}$ a x .n 3/x .2n/

$\frac{1}{}$ a x .n 3/a x .2n/ a x .n 3/a x .2n/

which clearly is not equal to a_1T_3 x_1.n/ C a_2T_3 x_2.n/ . The product term in the input-output equation makes the system T_3 x.n/ nonlinear.

2. MATLAB script:

```
% P0211c:   To prove that the system T3[x(n)] =          x(n) + 1/2*x(n - 2)
%                                                - 1/3*x(n - 3)*x(2n)
%            is linear
clear; clc; close all;

n = [0:100]; x1 = rand(1,length(n)); x2 = sqrt(10)*randn(1,length(n)); [x11,nx11] =
sigshift(x1,n,2); x11 = 1/2*x11; [x12,nx12] = sigshift(x1,n,3); x12 = -1/3*x12; [x13,nx13] =
dnsample(x1,n,2); [x14,nx14] = sigmult(x12,nx12,x13,nx13);

[x15,nx15] = sigadd(x11,nx11,x14,nx14);
[y1,ny1] = sigadd(x1,n,x15,nx15); [x21,nx21] = sigshift(x2,n,2); x21 =
1/2*x21; [x22,nx22] = sigshift(x2,n,3); x22 = -1/3*x22; [x23,nx23] =
dnsample(x2,n,2);
[x24,nx24] = sigmult(x22,nx22,x23,nx23);
[x25,nx25] = sigadd(x21,nx21,x24,nx24); [y2,ny2] = sigadd(x2,n,x25,nx25); xs = x1 +
x2; [xs1,nxs1] = sigshift(xs,n,2);
xs1 = 1/2*xs1; [xs2,nxs2] = sigshift(xs,n,3); xs2 = -1/3*xs2;
[xs3,nxs3] = dnsample(xs,n,2); [xs4,nxs4] = sigmult(xs2,nxs2,xs3,nxs3);
[xs5,nxs5] = sigadd(xs1,nxs1,xs4,nxs4);
[y,ny] = sigadd(xs,n,xs5,nxs5); diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
     disp(' *** System-3 is Linear *** ');
else
     disp(' *** System-3 is NonLinear *** ');
end
```

MATLAB verification:

>> *** System-3 is NonLinear ***

System-4: T_4 x.n/ D $\sum_{kD}^{nC5}$ 2x.k/

1. Analytic P

T_4 a_1x_1.n/ C a_2x_2.n/ D $\sum_{kD}^{nC5}$ 2fa_1x_1.k/ C a_2x_2.k/g D a_1 $\sum_{kD}^{nC5}$ 2x_1.k/ C a_2 $\sum_{kD}^{nC5}$ 2x_2.k/

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 40

$$= a_1 T_4\, x_1(n) + a_2 T_4\, x_2(n)$$

Hence the system $T_4\, x(n)$ is linear.

2. MATLAB script:

```
% P0211d: To prove that the system T4[x(n)] = sum_{k=-infinity}^{n+5}2*x(k)
%               is linear
clear; clc; close all;
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 41

```
n = [0:100]; x1 = rand(1,length(n)); x2 = sqrt(10)*randn(1,length(n)); y1 = cumsum(x1);
ny1 = n - 5; y2 = cumsum(x2); ny2 = n - 5; xs = x1 + x2; y = cumsum(xs); ny = n - 5;
diff = sum(abs(y - (y1 + y2))); if (diff < 1e-5)
    disp(' *** System-4 is Linear *** ');
else
    disp(' *** System-4 is NonLinear *** ');
end
```

MATLAB verification:

```
>> *** System-4 is Linear ***
```

System-5: T_5 x.n/ D x.2n/

1. Analytic determination of linearity:

$$T_5\ a_1x_1.n/\ C\ a_2x_2.n/\ D\ a_1x_1.2n/\ C\ a_2x_2.2n/\ D\ a_1T_5\ x_1.n/\ C\ a_2T_5\ x_2.n/$$

   Hence the system T_5 x.n/ is linear.

2. MATLAB script:

```
% P0211e: To prove that the system T5[x(n)] = x(2n) is linear clear; clc;
close all;

n = 0:100; x1 = rand(1,length(n)); x2 = sqrt(10)*randn(1,length(n)); [y1,ny1] =
dnsample(x1,n,2); [y2,ny2] = dnsample(x2,n,2); xs = x1 + x2; [y,ny] =
dnsample(xs,n,2); diff = sum(abs(y - (y1 + y2))); if (diff < 1e-5)
    disp(' *** System-5 is Linear *** ');
else
    disp(' *** System-5 is NonLinear *** ');
end
```

MATLAB verification:

```
>> *** System-5 is Linear ***
```

System-6: T_6 x.n/ D round x.n/

1. Analytic determination of linearity:

$$T_6\ a_1x_1.n/\ C\ a_2x_2.n/\ D\ \text{round}\ a_1x_1.n/\ C\ a_2x_2.n/\ \neq a_1\ \text{round}\ x_1.n/\ C\ a_2\ \text{round}\ x_2.n/$$

   Hence the system T_6 x.n/ is nonlinear.

2. MATLAB script:

```
% P0211f: To prove that the system T6[x(n)] = round(x(n)) is linear clear; clc;
close all;

n = 0:100; x1 = rand(1,length(n)); x2 = sqrt(10)*randn(1,length(n)); y1 =
round(x1); y2 = round(x2); xs = x1 + x2;
y    = round(xs); diff = sum(abs(y - (y1 + y2))); if (diff <
1e-5)
    disp(' *** System-6 is Linear *** ');
else
    disp(' *** System-6 is NonLinear *** ');
end
```

MATLAB verification:

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 42

```
>> *** System-6 is NonLinear ***
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 43

P2.12 Time-invariance of discrete-time systems.

System-1: T_1 x.n/ , y.n/ D x.n/u.n/

1. Analytic determination of time-invariance:

T_1 x.n k/ D x.n k/u.n/ ≠ x.n k/u.n k/ D y.n k/ Hence the system T_1 x.n/ is time-varying.

2. MATLAB script:

```
% P0212a: To determine whether T1[x(n)] = x(n)u(n) is time invariant clear; clc;
close all;

n = 0:100; x = sqrt(10)*randn(1,length(n)); u = stepseq(0,0,100);
y = x.*u; [y1,ny1] = sigshift(y,n,1); [x1,nx1] = sigshift(x,n,1);
[y2,ny2] = sigmult(x1,nx1,u,n); [diff,ndiff] = sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
    disp(' *** System-1 is Time-Invariant *** ');
else
    disp(' *** System-1 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-1 is Time-Varying ***
```

System-2: T_2 x.n/ , y.n/ D x.n/ C n x.n C 1/

1. Analytic determination of time-invariance:

T_2 x.n k/ D x.n k/ C n x.n k C 1/ ≠ x.n k/ C .n k/ x.n k C 1/ D y.n k/ Hence the system is T_2 x.n/ time-varying.

2. MATLAB script:

```
% P0212b: To determine whether the system T2[x(n)] = x(n) + n*x(n + 1) is
%               time-invariant
clear; clc; close all;

n = 0:100; x = sqrt(10)*randn(1,length(n));
z    = n; [x1,nx1] = sigshift(x,n,-1);
[x11,nx11] = sigmult(z,n,x1,nx1); [y,ny] = sigadd(x,n,x11,nx11);
[y1,ny1] = sigshift(y,ny ,1); [xs,nxs]                    = sigshift(x,n,1);
[xs1,nxs1] = sigshift(xs,nxs,-1); [xs11,nxs11] = sigmult(z,n,xs1,nxs1);
[y2,ny2] = sigadd(xs,nxs,xs11,nxs11); [diff,ndiff] = sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
    disp(' *** System-2 is Time-Invariant *** ');
else
    disp(' *** System-2 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-1 is Time-Varying ***
```

System-3: T_3 x.n/ , y.n/ D x.n/ C $\frac{1}{2}$x.n 2/ $\frac{1}{3}$x.n 3/x.2n/

1. Analytic determination of time-invariance:

1 1

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 44

$$
\begin{aligned}
T_3[x(n-k)] &= x(n-k) + \tfrac{1}{2}x(n-k-2) - \tfrac{1}{3}x(n-k-3)x(2n-k) \\
&\neq x(n-k) + \tfrac{1}{2}x(n-k-2) - \tfrac{1}{3}x(n-k-3)x(2n-2k) = y(n-k)
\end{aligned}
$$

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 45

Hence the system is T_3 x.n/ time-varying.

2. MATLAB script:

```
% P0212c: To find whether the system T3[x(n)] =              x(n) + 1/2*x(n - 2)
%                                                - 1/3*x(n - 3)*x(2n)
%                                                   is time invariant
clear; clc; close all;

n = 0:100; x = sqrt(10)*randn(1,length(n)); [x1,nx1] = sigshift(x,n,2);
  x1 = 1/2*x1; [x2,nx2] = sigshift(x,n,3);                 x2 = -1/3*x2;
[x3,nx3] = dnsample(x,n,2); [x4,nx4] = sigmult(x2,nx2,x3,nx3);
[x5,nx5] = sigadd(x1,nx1,x4,nx4); [y,ny]                 = sigadd(x,n,x5,nx5);
[y1,ny1] = sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1);
[xs1,nxs1] = sigshift(xs,nxs,2); xs1 = 1/2*xs1;
[xs2,nxs2] = sigshift(xs,nxs,3); xs2 = -1/3*xs2;
[xs3,nxs3] = dnsample(xs,nxs,2); [xs4,nxs4] = sigmult(xs2,nxs2,xs3,nxs3);
[xs5,nxs5] = sigadd(xs1,nxs1,xs4,nxs4); [y2,ny2] = sigadd(xs,nxs,xs5,nxs5);
[diff,ndiff] = sigadd(y1,ny1,-y2,ny2); diff = sum(abs(diff));
if (diff < 1e-5)
      disp(' *** System-3 is Time-Invariant *** ');
else
      disp(' *** System-3 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-3 is Time-Varying ***
```

System-4: T_4 x.n/ , y.n/ D $\sum_{kD}^{nC5}$ 2x.k/

1. Analytic determination of time-invariance:

$$T_4 \text{ x.n } ‘/ \text{ D } \sum_{kD}^{nC5}_{x} \text{ 2x.k } ‘/ \text{ D } \sum_{kD}^{n‘C5}_{x} \text{ 2x.k/ D y.n } ‘/$$

Hence the system T_4 x.n/ is time-invariant.

2. MATLAB script:

```
% P0212d: To find whether the system T4[x(n)]=sum_{k=-infinity}^{n+5}2*x(k)
%              is time-invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); y = cumsum(x); ny = n - 5; [y1,ny1] =
sigshift(y,ny,-1); [xs,nxs] = sigshift(x,n,-1); y2 = cumsum(xs); ny2 = nxs - 5; [diff,ndiff] =
sigadd(y1,ny1,-y2,ny2); diff = sum(abs(diff)); if (diff < 1e-5)
      disp(' *** System-4 is Time-Invariant *** ');
else
      disp(' *** System-4 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-4 is Time-Invariant ***
```

System-5: T_5 x.n/ , y.n/ D x.2n/

1. Analytic determination of time-invariance:

$$T_5 \text{ x.n k/ D x.2n k/} \not= \text{x 2.n k/ D y.n k/}$$

Hence the system T_5 x.n/ is time-varying.

2. MATLAB script:

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 46

```
% P0212e: To determine whether the system T5[x(n)] = x(2n) is time-invariant clear; clc;
close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); [y,ny] = dnsample(x,n,2); [y1,ny1] =
sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1);
[y2,ny2] = dnsample(xs,nxs,2); [diff,ndiff] = sigadd(y1,ny1,-y2,ny2); diff =
sum(abs(diff));
if (diff < 1e-5)
    disp(' *** System-5 is Time-Invariant *** ');
else
    disp(' *** System-5 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-5 is Time-Varying ***
```

System-6: $T_6[x(n)] \triangleq y(n) = \text{round}[x(n)]$

1. Analytic determination of time-invariance:

$$T_6[x(n-k)] = \text{round}[x(n-k)] = y(n-k) \text{ Hence}$$

   the system $T_6[x(n)]$ is time-invariant.

2. MATLAB script:

```
% P0212f: To determine if the system T6[x(n)]=round(x(n)) is time-invariant clear; clc;
close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); y = round(x); ny = n; [y1,ny1] =
sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1); y2 = round(xs); ny2 = nxs; [diff,ndiff] =
sigadd(y1,ny1,-y2,ny2); diff = sum(abs(diff)); if (diff < 1e-5)
    disp(' *** System-6 is Time-Invariant *** ');
else
    disp(' *** System-6 is Time-Varying *** ');
end
```

MATLAB verification:

```
>> *** System-6 is Time-Invariant ***
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 47

P2.13 Stability and Causality of Discrete-time Systems

System-1: T1 x.n/ , y.n/ D x.n/u.n/: This system is stable since jy.n/j D jx.n/j. It is also causal since the output depends only on the present value of the input.

System-2: T2 x.n/ , y.n/ D x.n/ C n x.n C 1/: This system is unstable since

jy.n/j jx.n/j C jnj jx.n C 1/j % 1 as n % 1 for jx.n/j < 1

It is also noncausal since the output y.n/ depends on the future input value x.n C 1/ for n > 0.

System-3: T3 x.n/ , y.n/ D x.n/ C 1/2 x.n 2/ 1/3 x.n 3/x.2n/: This system is stable since

jy.n/j jx.n/j C 1/2 jx.n 2/j C 1/3 jx.n 3/j jx.2n/j < 1 for jx.n/j < 1

It is however is noncausal since y.1/ needs x.2/ which is a future input value.

System-4: T4 x.n/ , y.n/ D P nC5 kD n C5 2x.k/ : This system is unstable since

jy.n/j 2 P kD jx.k/j % 1 as n % 1 for jx.n/j < 1

It is also noncausal since the output y.n/ depends on the future input value x.n C 5/ for n > 0.

System-5: T5 x.n/ , y.n/ D x.2n/: This system is stable since jy.n/j D jx.2n/j < 1 for jx.n/j < 1. It is however noncausal since y.1/ needs x.2/ which is a future input value.

System-6: T6 x.n/ , y.n/ D round x.n/ : This system is stable and causal.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 42

P2.14 Properties of linear convolution.

$$x_1(n) * x_2(n) = x_2(n) * x_1(n) \quad : \text{Commutation}$$
$$[x_1(n) * x_2(n)] * x_3(n) = x_1(n) * [x_2(n) * x_3(n)] \quad : \text{Association}$$
$$x_1(n) * [x_2(n) + x_3(n)] = x_1(n) * x_2(n) + x_1(n) * x_3(n) \quad : \text{Distribution}$$
$$x(n) * \delta(n - n_0) = x(n - n_0) \quad : \text{Identity}$$

1. Commutation:

$$x_1(n) * x_2(n) = \sum_{k=-\infty}^{\infty} x_1(k)\, x_2(\underbrace{n-k}_{=m}) = \sum_{m=-\infty}^{\infty} x_1(n-m)\, x_2(m)$$
$$= \sum_{m=-\infty}^{\infty} x_2(m)\, x_1(n-m) = x_2(n) * x_1(n)$$

Association:

$$[x_1(n) * x_2(n)] * x_3(n) = \left[\sum_{k=-\infty}^{\infty} x_1(k)\, x_2(n-k)\right] * x_3(n)$$
$$= \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x_1(k)\, x_2(m-k)\, x_3(n-m)$$
$$= \sum_{k=-\infty}^{\infty} x_1(k) \left[\sum_{m=-\infty}^{\infty} x_2(\underbrace{m-k}_{=\ell})\, x_3(n-m)\right]$$
$$= \sum_{k=-\infty}^{\infty} x_1(k) \left[\sum_{\ell=-\infty}^{\infty} x_2(\ell)\, x_3(n-k-\ell)\right]$$
$$= \sum_{k=-\infty}^{\infty} x_1(k)\, [x_2(n-k) * x_3(n-k)] = x_1(n) * [x_2(n) * x_3(n)]$$

Distribution:

$$x_1(n) * [x_2(n) + x_3(n)] = \sum_{k=-\infty}^{\infty} x_1(k)\, [x_2(n-k) + x_3(n-k)]$$
$$= \sum_{k=-\infty}^{\infty} x_1(k)\, x_2(n-k) + \sum_{k=-\infty}^{\infty} x_1(k)\, x_3(n-k)$$
$$= x_1(n) * x_2(n) + x_1(n) * x_3(n)$$

Identity:

$$x(n) * \delta(n - n_0) = \sum_{k=-\infty}^{\infty} x(k)\, \delta(n - n_0 - k) = x(n - n_0)$$

since $\delta(n - n_0 - k) = 1$ for $k = n - n_0$ and zero elsewhere.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 43

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 43

2. Verification using MATLAB:

Commutation MATLAB script:

```
% P0214a: To prove the Commutation property of convolution
%               i.e. conv(x1(n),x2(n)) = conv(x2(n), x1(n)) clear;
clc; close all;

n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 / 4);
n11 = n1; [x12,n12] = stepseq(-5,-10,30); [x13,n13] = stepseq(25,-10,30); [x14,n14] =
sigadd(x12,n12,-x13,n13); x1 = x11.*x14;
x21 = 0.9 .^ -n2; [x22,n22] = stepseq(0,0,25); [x23,n23] = stepseq(20,0,25);
x24 = x22 - x23; x2 = x21.*x24;
x3 = round((rand(1,21)*2 - 1)*5);

% Commutative property
[y1,ny1] = conv_m(x1,n1,x2,n2); [y2,ny2] = conv_m(x2,n2,x1,n1); ydiff =
max(abs(y1 - y2)), ndiff = max(abs(ny1 - ny2)), MATLAB verification:

  ydiff =
       0
  ndiff =
       0
```

Association MATLAB script:

```
% P0214b: To prove the Association property of convolution
%               i.e. conv(conv(x1(n),x2(n)),x3(n)) = conv(x1(n),conv(x2(n),x3(n))) clear; clc;
close all;

n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 / 4); n11 = n1;
[x12,n12] = stepseq(-5,-10,30); [x13,n13] = stepseq(25,-10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14;
x21 = 0.9 .^ -n2; [x22,n22] = stepseq(0,0,25); [x23,n23] = stepseq(20,0,25);
x24 = x22 - x23; x2 = x21.*x24; x3 = round((rand(1,21)*2 - 1)*5);

% Association property
[y1,ny1] = conv_m(x1,n1,x2,n2); [y1,ny1] = conv_m(y1,ny1,x3,n3);
[y2,ny2] = conv_m(x2,n2,x3,n3); [y2,ny2] = conv_m(x1,n1,y2,ny2);
ydiff = max(abs(y1 - y2)), ndiff = max(abs(ny1 - ny2)),
```

MATLAB verification:

```
  ydiff =
       0
  ndiff =
       0
```

Distribution MATLAB script:

```
% P0214c: To prove the Distribution property of convolution
%               i.e. conv(x1(n),(x2(n)+x3(n)))=conv(x1(n),x2(n))+conv(x1(n),x3(n)) clear; clc;
close all;

n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 / 4); n11 = n1; [x12,n12] =
stepseq(-5,-10,30); [x13,n13] = stepseq(25,-10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14; x21 = 0.9 .^ -n2;
[x22,n22] = stepseq(0,0,25); [x23,n23] = stepseq(20,0,25); x24 = x22 - x23;
x2 = x21.*x24; x3 = round((rand(1,21)*2 - 1)*5);
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Solutions Manual for DSP using Matlab (4th Edition) 44

% Distributive property

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 45

```
[y1,ny1] = sigadd(x2,n2,x3,n3); [y1,ny1] = conv_m(x1,n1,y1,ny1);
[y2,ny2] = conv_m(x1,n1,x2,n2); [y3,ny3] = conv_m(x1,n1,x3,n3);
[y4,ny4] = sigadd(y2,ny2,y3,ny3); ydiff = max(abs(y1 - y4)), ndiff =
max(abs(ny1 - ny4)),
```

MATLAB verification:

```
ydiff =
     0
ndiff =
     0
```

Identity MATLAB script:

```
% P0214d: To prove the Identity property of convolution
%          i.e. conv(x(n),delta(n - n0)) = x(n - n0) clear; clc;
close all;

n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 / 4); n11 = n1; [x12,n12] =
stepseq(-5,-10,30); [x13,n13] = stepseq(25,-10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14; x21 = 0.9 .^ -n2;
[x22,n22] = stepseq(0,0,25); [x23,n23] = stepseq(20,0,25); x24 = x22 - x23;
x2 = x21.*x24; x3 = round((rand(1,21)*2 - 1)*5);

% Identity property
n0 = fix(100*rand(1,1)-0.5); [dl,ndl] = impseq(n0,n0,n0); [y11,ny11] =
conv_m(x1,n1,dl,ndl); [y12,ny12] = sigshift(x1,n1,n0); y1diff = max(abs(y11 -
y12)), ny1diff = max(abs(ny11 - ny12)),

[y21,ny21] = conv_m(x2,n2,dl,ndl); [y22,ny22] = sigshift(x2,n2,n0); y2diff =
max(abs(y21 - y22)), ny2diff = max(abs(ny21 - ny22)),

[y31,ny31] = conv_m(x3,n3,dl,ndl); [y32,ny32] = sigshift(x3,n3,n0); y3diff =
max(abs(y31 - y32)), ny3diff = max(abs(ny31 - ny32)),
```

MATLAB verification:

```
ydiff =
     0
ndiff =
     0
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 46

P2.15 Convolutions using conv_m function.

1. x.n/ D f2; 4; 5; 3; 1; 2; 6g, h.n/ D f1; 1 ; 1; 1; 1g: MATLAB script:

```
n1 = -3:3; x = [2 -4 5 3 -1 -2 6]; n2 = -1:3; h = [1 -1 1 -1 1];
[y,n] = conv_m(x,n1,h,n2); y, n
y =
     2    -6    11    -8     7    -7     9    -4     7    -8     6
n =
    -4    -3    -2    -1     0     1     2     3     4     5     6
```

2. x.n/ D f1; 1; 0; 1; 1g, h.n/ D f1; 2; 3; 4g: MATLAB script:

```
n1 = -3:3; x = [1 1 0 1 1]; n2 = -3:0; h = [1 -2 -3 4];
[y,n] = conv_m(x,n1,h,n2); y, n,
y =
     1    -1    -5     2     3    -5     1     4
n =
    -6    -5    -4    -3    -2    -1     0     1
```

3. x.n/ D .1=4/^{n} u.n C 1/ u.n 4/ , h.n/ D u.n/ u.n 5/: MATLAB script:

```
n1 = -2:5; [x11,nx11] = stepseq(-1,-2,5); [x12,nx12] = stepseq(4,-2,5); [x13,n13] =
sigadd(x11,nx11,-x12,nx12); x14 = 0.25 .^ -n1; n14 = n1; x = x14 .* x13;

n2 = 0:6; [h11,nh11] = stepseq(0,0,6); [h12,nh12] = stepseq(5,0,6); h=h11-h12; [y,n] =
conv_m(x,n1,h,n2); y, n,
y = 0    0.2500    1.2500    5.2500    21.2500    85.2500    85.0000    84.0000
    80.0000    64.0000    0    0    0    0
n = -2    -1    0    1    2    3    4    5    6    7    8    9    10
    11
```

4. x.n/ D n=4 u.n/ u.n 6/ , h.n/ D 2 u.n C 2/ u.n 3/ : MATLAB script:

```
n1 = 0:7; [x11,nx11] = stepseq(0,0,7); [x12,nx12] = stepseq(6,0,7);
[x13,n13] = sigadd(x11,nx11,-x12,nx12); x14 = n1/4; n14 = n1; x = x14 .* x13;
n2 = -3:4; [h11,nh11] = stepseq(-2,-3,4); [h12,nh12] = stepseq(3,-3,4);
h = 2 * (h11 - h12); [y,n] = conv_m(x,n1,h,n2); y, n,
y = 0    0    0.5000    1.5000    3.0000    5.0000    7.5000    7.0000
    6.0000    4.5000    2.5000    0    0    0    0
n = -3    -2    -1    0    1    2    3    4    5    6    7    8    9
    10    11
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 47

P2.16 Let $x(n) = (0.8)^n u(n)$, $h(n) = (-0.9)^n u(n)$, and $y(n) = h(n) * x(n)$.

1. Convolution $y(n) = h(n) * x(n)$:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=-\infty}^{\infty} (-0.9)^k (0.8)^{n-k} u(n-k)$$

$$= \left[\sum_{k=0}^{n} (-0.9)^k (0.8)^n (0.8)^{-k}\right] u(n) = (0.8)^n \sum_{k=0}^{n} \left(-\frac{9}{8}\right)^k$$

$$= \frac{0.8^{n+1} - (-0.9)^{n+1}}{1.7}$$

MATLAB script:

```
clc; close all;run defaultsettings;
n = [0:50]; x = 0.8.^n; h = (-0.9).^n;
Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0216');

% (a) Plot of the analytical convolution
y1 = ((0.8).^(n+1) - (-0.9).^(n+1))/(0.8+0.9);
subplot(1,3,1); Hs1 = stem(n,y1,'filled'); set(Hs1,'markersize',2);
title('Analytical'); xlabel('n'); ylabel('y(n)');
```

2. Computation using convolution of truncated sequences: MATLAB script

```
% (b) Plot using the conv function and truncated sequences x2 =
x(1:26); h2 = h(1:26); y2 = conv(h2,x2);
subplot(1,3,2); Hs2 = stem(n,y2,'filled'); set(Hs2,'markersize',2); title('Using
conv function'); xlabel('n'); %ylabel('y(n)');
```

3. To use the MATLAB's filter function we have to represent the $h(n)$ sequence by coefficients an equivalent difference equation. MATLAB script:

```
% (c) Plot of the convolution using the filter function
y3 = filter([1],[1,0.9],x);
subplot(1,3,3); Hs3 = stem(n,y3,'filled'); set(Hs3,'markersize',2); title('Using
filter function'); xlabel('n'); %ylabel('y(n)');
```

The plots of this solution are shown in Figure 2.33. The analytical solution to the convolution in 1 is the exact answer. In the filter function approach of 2, the infinite-duration sequence $x(n)$ is exactly represented by coefficients of an equivalent filter. Therefore, the filter solution should be exact except that it is evaluated up to the length of the input sequence. The truncated-sequence computation in 3 is correct up to the first 26 samples and then it degrades rapidly.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 48

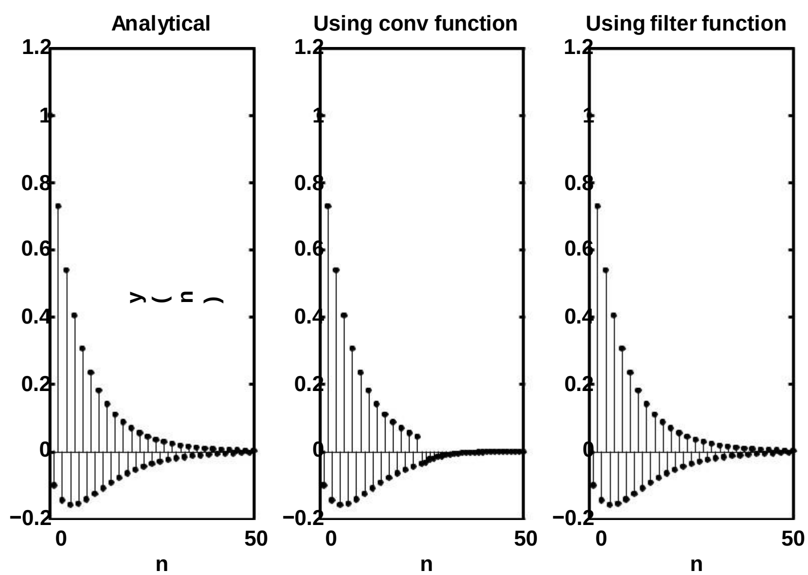

Figure 2.33: Problem P2.16 convolution plots

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 49

P2.17 Linear convolution as a matrix-vector multiplication. Consider the sequences

x .n/ D f1; 2; 3; 4; 5g and h .n/ D f6; 7; 8; 9g

1. The linear convolution of the above two sequences is

y .n/ D f6; 19; 40; 70; 100; 94; 76; 45g

2. The vector representation of the above operation is:

19 3 2 7 6 0 0 0 1
6 6 0 0 0 0 3
6 70 7 6 9 8 7 6 0 7 2 2
6 40 7 6 8 7 6 0 0 7 6 7
7 D 6 4
6 94 7 6 0 0 9 8 7 7 6 5 7
6 76 7 6 0 0 0 9 8 7 7 6 7
6 45 7 6 0 0 0 0 9 7 x
4 y 5 4 H 5

(a) Note that the matrix H has an interesting structure. Each diagonal of H contains the same number. Such a matrix is called a Toeplitz matrix. It is characterized by the following property

H i;j D H ij

which is similar to the definition of time-invariance.

(b) Note carefully that the first column of H contains the impulse response vector h .n/ followed by number of zeros equal to the number of x .n/ values minus one. The first row contains the first element of h .n/ followed by the same number of zeros as in the first column. Using this information and the above property we can generate the whole Topelitz matrix.

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 50

P2.18 MATLAB function conv_tp:

(a) The MATLAB function conv_tp:

```
function [y,H]=conv_tp(h,x)
% Linear Convolution using Toeplitz Matrix
% ----------------------------------------
% [y,H] = conv_tp(h,x)
% y = output sequence in column vector form
% H = Toeplitz matrix corresponding to sequence h so that y = Hx
% h = Impulse response sequence in column vector form
% x = input sequence in column vector form
%
Nx = length(x); Nh = length(h);
hc = [h; zeros(Nx-1, 1)]; hr = [h(1),zeros(1,Nx-1)]; H =
toeplitz(hc,hr); y = H*x;
```

(b) MATLAB verification:

```
x = [1,2,3,4,5]'; h = [6,7,8,9]'; [y,H] =
conv_tp(h,x); y = y', H
  y =
       6    19    40    70   100    94    76    45
  H =
       6     0     0     0     0
       7     6     0     0     0
       8     7     6     0     0
       9     8     7     6     0
       0     9     8     7     6
       0     0     9     8     7
       0     0     0     9     8
       0     0     0     0     9
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 51

P2.19 A linear and time-invariant system is described by the difference equation

y.n/ 0:5y.n 1/ C 0:25y.n 2/ D x.n/ C 2x.n 1/ C x.n 3/

(a) Impulse response using the Using the filter function.

```
% P0219a: System response using the filter function clc;
close all;

b = [1 2 0 1]; a = [1 -0.5 0.25]; [delta,n] = impseq(0,0,100); h =
filter(b,a,delta);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0219a'); Hs =
stem(n,h,'filled'); set(Hs,'markersize',2); axis([min(n)-5,max(n)+5,min(h)-
0.5,max(h)+0.5]); xlabel('n','FontSize',LFS);
ylabel('h(n)','FontSize',LFS); title('Impulse response','FontSize',TFS);
print -deps2 ../EPSFILES/P0219a.eps;
```

The plots of the impulse response h.n/ is shown in Figure 2.34.

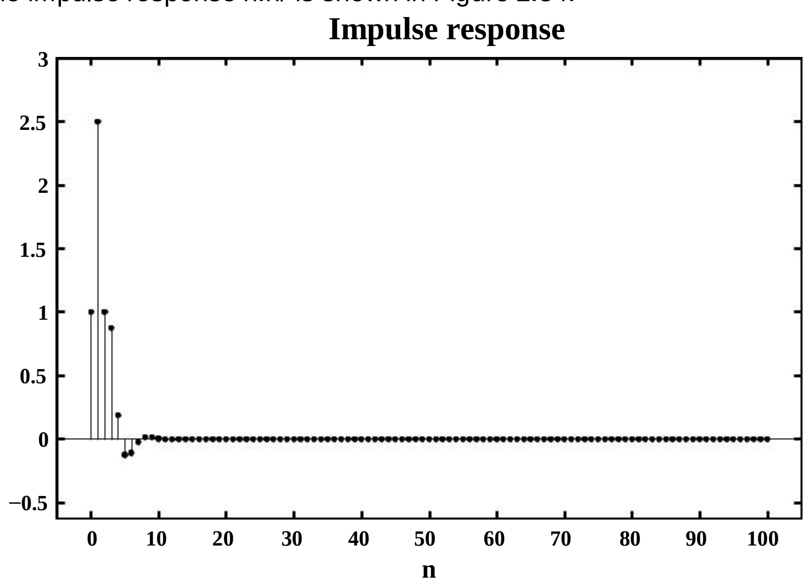

Figure 2.34: Problem P2.19.1 impulse response plot

(b) Clearly from Figure 2.34 the system is stable.

(c) Response y.n/ when the input is x.n/ D 5 C 3 cos.0:2 n/ C 4 sin.0:6 n/ u.n/:

```
% P0219c: Output response of a system using the filter function. clc; close
all;

b = [1 2 0 1]; a = [1 -0.5 0.25]; n = 0:200;
x = 5*ones(size(n))+3*cos(0.2*pi*n)+4*sin(0.6*pi*n); y = filter(b,a,x);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0219c');
Hs = stem(n,y,'filled'); set(Hs,'markersize',2); axis([-10,210,0,50]);
xlabel('n','FontSize',LFS); ylabel('y(n)','FontSize',LFS); title('Output
response','FontSize',TFS); print -deps2 ../EPSFILES/P0219c.eps;
```


© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 52

The plots of the response y.n/ is shown in Figure 2.35.

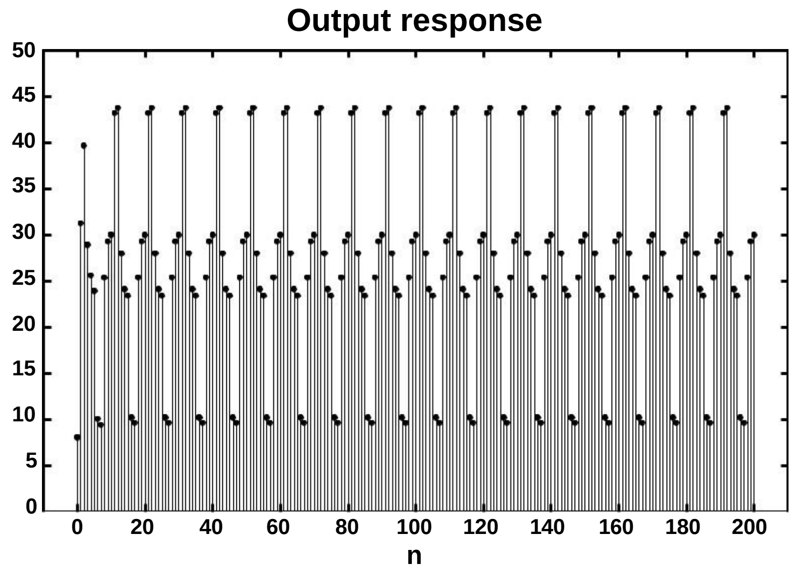

Figure 2.35: Problem P2.19.3 response plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 52

P2.20 A “simple” digital differentiator: y.n/ D x.n/ x.n 1/

(a) Response to a rectangular pulse x.n/ D 5 u.n/u.n 20/ :

```
% P0220a: Simple Differentiator response to a rectangular pulse clc; close
all;

a = 1; b = [1 -1]; n1 = 0:22;
[x11,nx11] = stepseq(0,0,22); [x12,nx12] = stepseq(20,0,22); x1 =
5*(x11 - x12); y1 = filter(b,a,x1);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0220a');
Hs = stem(n1,y1,'filled'); set(Hs,'markersize',2); axis([-1,23,-6,6]);
xlabel('n','FontSize',LFS); ylabel('y(n)','FontSize',LFS); ytick = [-6:6]; title('Output
response for rectangular pulse ','FontSize',TFS);
set(gca,'YTickMode','manual','YTick',ytick); print -deps2 ../EPSFILES/P0220a.eps;
```

The plots of the response y.n/ is shown in Figure 2.36.

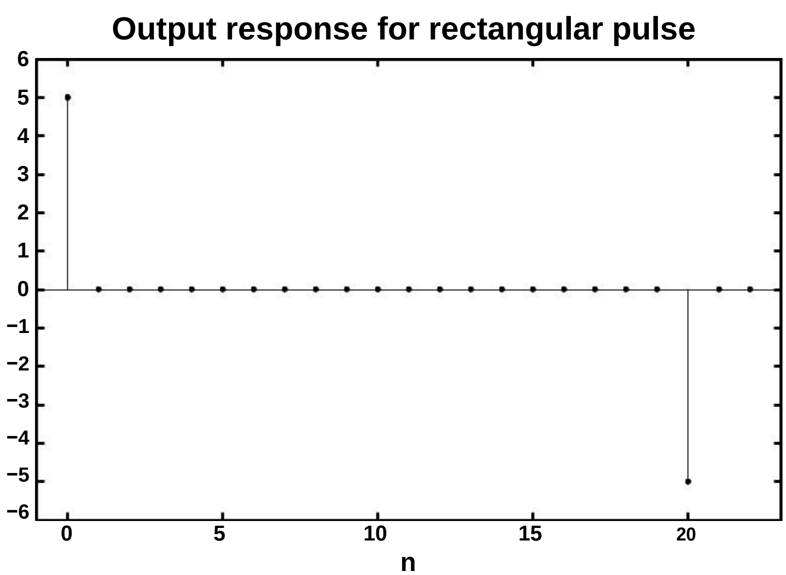

Figure 2.36: Problem P2.20.1 response plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 53

(b) Response to a triangular pulse $x(n) = n\,[u(n) - u(n-10)] + (20-n)\,[u(n-10) - u(n-20)]$:

```
% P0220b: Simple Differentiator response to a triangular pulse clc; close
all;

a = 1; b = [1 -1]; n2 = 0:21; [x11,nx11] = stepseq(0,0,21); [x12,nx12] =
stepseq(10,0,21); [x13,nx13] = stepseq(20,0,21);
x2   =   n2.*(x11 - x12) + (20 - n2).*(x12 - x13); y2 = filter(b,a,x2);

Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0220b'); Hs =
stem(n2,y2,'filled'); set(Hs,'markersize',2); axis([min(n2)-
1,max(n2)+1,min(y2)-0.5,max(y2) + 0.5]); xlabel('n','FontSize',LFS);
ylabel('y(n)','FontSize',LFS); title('Output response for triangular
pulse','FontSize',TFS); print -deps2 ../EPSFILES/P0220b.eps;
```

The plots of the response $y(n)$ is shown in Figure 2.37.

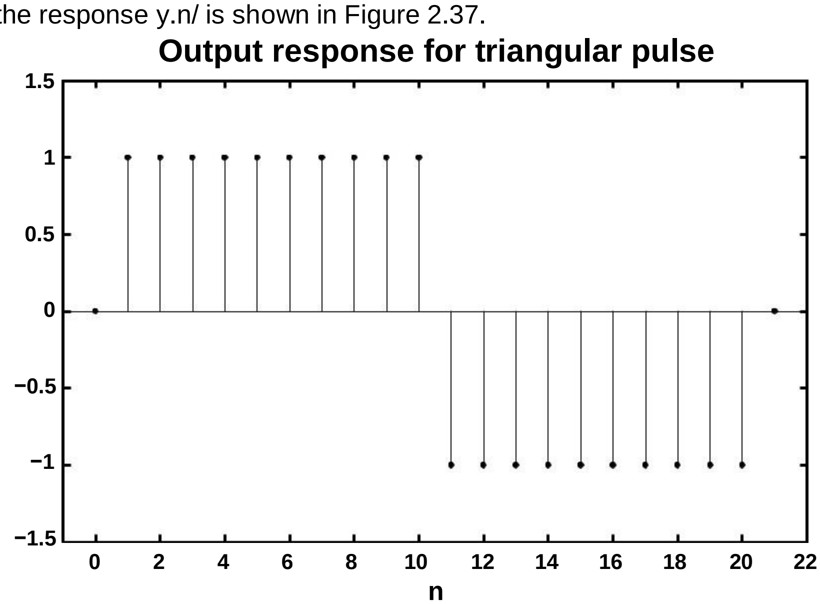

Figure 2.37: Problem P2.20.2 response plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

SOLUTIONS MANUAL FOR DSP USING MATLAB (4TH EDITION) 54

(c) Response to a sinusoidal pulse $x(n) = \sin\frac{n}{25}\, u(n)\ \ u(n - 100)$:

```
% P0220cSimple Differentiator response to a sinusoidal pulse clc;
close all;

a = 1; b = [1 -1]; n3 = 0:101; [x11,nx11] = stepseq(0,0,101); [x12,nx12] =
stepseq(100,0,101); x13 = x11-x12; x3 = sin(pi*n3/25).*x13; y3 = filter(b,a,x3);


Hf_1 = figure; set(Hf_1,'NumberTitle','off','Name','P0220c'); Hs =
stem(n3,y3,'filled'); set(Hs,'markersize',2); axis([-5,105,-0.15,0.15]);
ytick = [-0.15:0.05:0.15]; xlabel('n','FontSize',LFS);
ylabel('y(n)','FontSize',LFS); title('Output response for sinusoidal
pulse','FontSize',TFS); set(gca,'YTickMode','manual','YTick',ytick); print
-deps2 ../EPSFILES/P0220c.eps;
```

The plots of the response $y(n)$ is shown in Figure 2.38.

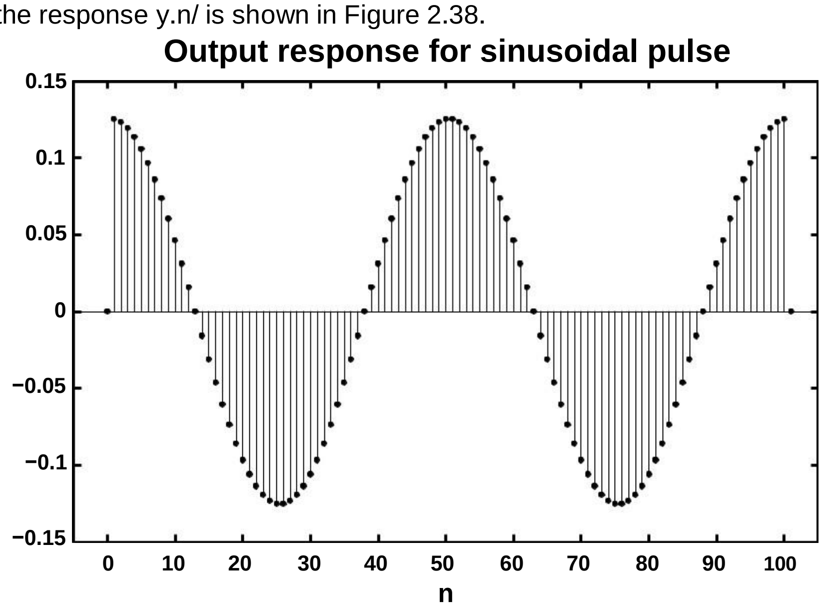

Figure 2.38: Problem P2.20.3 response plot

© 2017 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.