# Chapter 2

# Discrete-Time Signals and Systems

1. Problem **P2.1**:

   (a) $x_1(n) = \sum_{m=0}^{10}(m+1)[\delta(n-2m-1)-\delta(n-2m)]$, $0 \le n \le 25$.

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P2.1ac');
%
% x1(n) = sum_{m=0}^{10} (m+1)*[delta(n-2*m)-delta(n-2*m-1)]
n1 = [0:25]; x1 = zeros(1,length(n1));
for m = 0:10
    x1 = x1 + (m+1)*(impseq(2*m,0,25) - impseq(2*m+1,0,25));
end
subplot(2,1,1); stem(n1,x1);
axis([min(n1)-1,max(n1)+1,min(x1)-2,max(x1)+2]);
xlabel('n'); ylabel('x1(n)'); title('Sequence x1(n)');
ntick = [n1(1):n1(length(n1))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

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

   (b) $x_2(n) = n^2[u(n+5)-u(n-6)]+10\delta(n)+20(0.5)^n[u(n-4)-u(n-10)]$.

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P2.1be');
%
% (b) x2(n) = (n^2)*[u(n+5)-u(n-6)]+10*delta(n)+20*(0.5)^n*[u(n-4)-u(n-10)]
n2 = -5:10; % Overall support of x2(n)
x2 = (n2.^2).*(stepseq(-5,-5,10)-stepseq(6,-5,10))+10*impseq(0,-5,10)+...
       20*((0.5).^n2).*(stepseq(4,-5,10)-stepseq(10,-5,10));
subplot(2,1,1); stem(n2,x2);
axis([min(n2)-1,max(n2)+1,min(x2)-2,max(x2)+2]);
xlabel('n'); ylabel('x1(n)'); title('Sequence x2(n)');
ntick = [n2(1):n2(length(n2))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

   The plots of $x_2(n)$ is shown in Figure 2.3.

   (c) $x_3(n) = (0.9)^n \cos(0.2\pi n + \pi/3)$, $\quad 0 \le n \le 20$.

```
% x3(n) = (0.9)^n*cos(0.2*pi*n+pi/3); 0<=n<=20
n3 = [0:20];
```

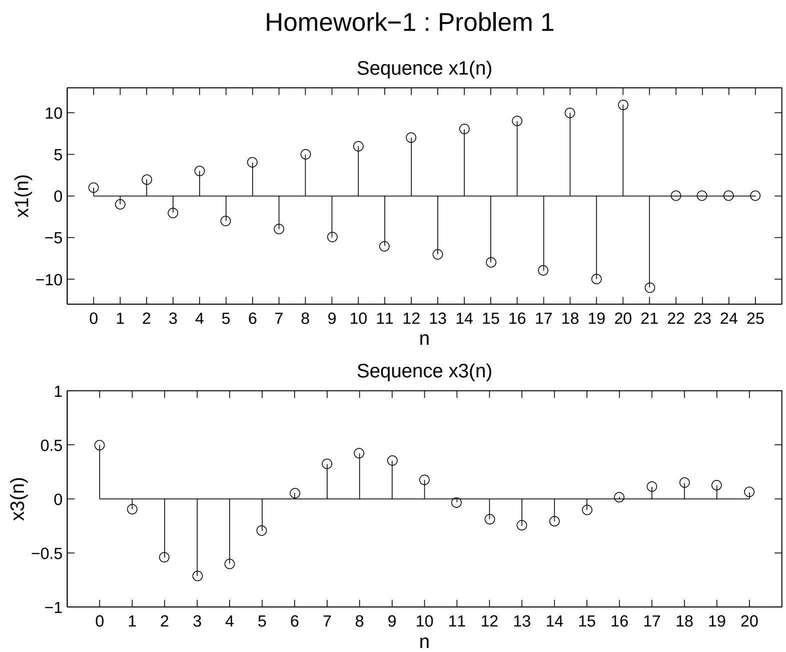


Figure 2.1: Problem P2.1 Sequence Plots

```
x3 = ((0.9).^n3).*cos(0.2*pi*n3+pi/3);
subplot(2,1,2); stem(n3,x3);
axis([min(n3)-1,max(n3)+1,-1,1]);
xlabel('n'); ylabel('x3(n)'); title('Sequence x3(n)');
ntick = [n3(1):n3(length(n3))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

The plots of $x_3(n)$ is shown in Figure 2.1.

(d) $x_4(n) = 10\cos\left(0.0008\pi n^2\right) + w(n)$, $0 \le n \le 100$ where $w(n)$ is a random sequence uniformly distributed between $[-1,1]$.

```
clear; close all;
% (d) x4(n) = 10*cos(0.0008*pi*n.^2)+w(n); 0 <= n <= 100; w(n)~uniform[-1,1]
w = 2*(rand(1,101)-0.5);
n4 = [0:100]; x4 = 10*cos(0.0008*pi*n4.^2)+w;
subplot(2,1,2); stem(n4,x4); axis([min(n4)-1,max(n4)+1,min(x4)-2,max(x4)+2]);
xlabel('n'); ylabel('x4(n)'); title('Sequence x4(n)');
ntick = [n4(1):10:n4(length(n4))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

The plot of $x_4(n)$ is shown in Figure 2.2 from which we observe that it is a noisy sinusoid with increasing frequency (or a noisy *chirp* signal).

(e) $\tilde{x}_5(n) = \left\{\ldots,1,2,3,\underset{\uparrow}{2},1,2,3,2,1,\ldots\right\}$. Plot 5 periods.

```
% (e) x5(n) = {...,1,2,3,2,1,2,3,2,1,...}periodic. 5 periods
n5 = [-8:11]; x5 = [2,1,2,3];
x5 = x5'*ones(1,5); x5 = (x5(:))';
subplot(2,1,2); stem(n5,x5);
axis([min(n5)-1,max(n5)+1,0,4]);
```

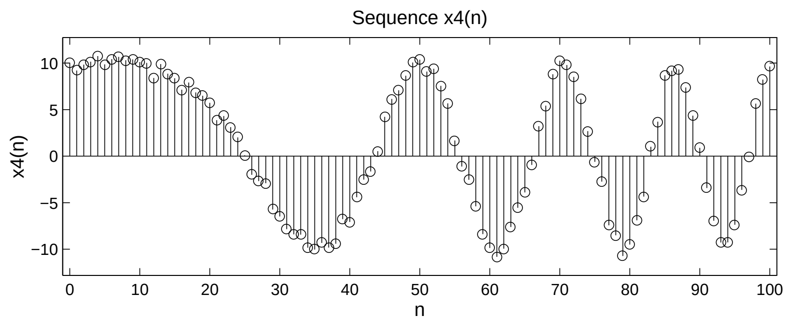


Figure 2.2: Plot of the sequence $x_4(n)$ in Problem P2.1d.

```
xlabel('n'); ylabel('x5(n)'); title('Sequence x5(n)');
ntick = [n5(1):n5(length(n5))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

The plots of $x_5(n)$ is shown in Figure 2.3.

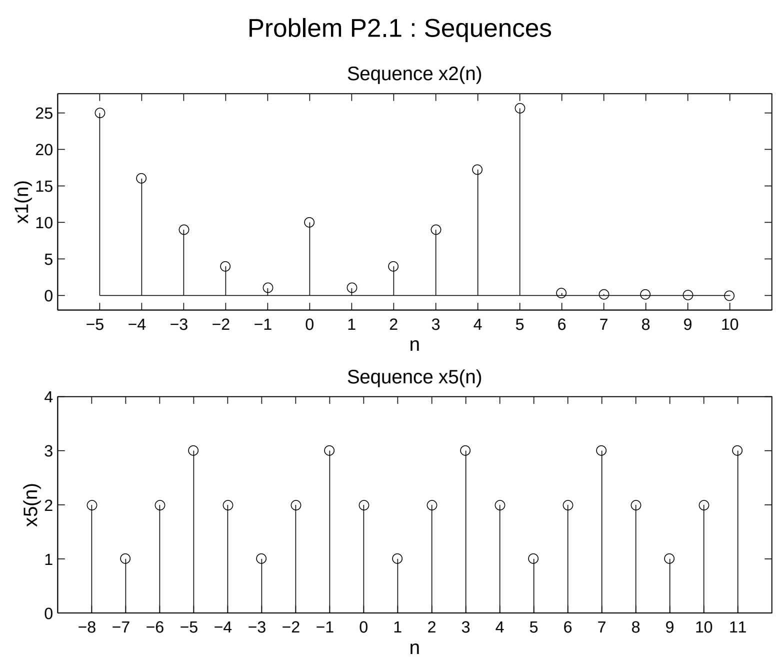


Figure 2.3: Problem P2.1 Sequence Plots

2. Problem **P2.2**: The sequence $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$ is given.

   (a) $x_1(n) = 3x(n+2) + x(n-4) - 2x(n)$.

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P2.2ab');
```

```
n = [-4:2]; x = [1,-2,4,6,-5,8,10]; % given seq x(n)
%
% (a) x1(n) = 3*x(n+2) + x(n-4) - 2*x(n)
[x11,n11] = sigshift(3*x,n,-2);        % shift by -2 and scale by 3
[x12,n12] = sigshift(x,n,4);           % shift x(n) by 4
[x13,n13] = sigadd(x11,n11,x12,n12);   % add two sequences at time
[x1,n1] = sigadd(x13,n13,2*x,n);       % add two sequences
subplot(2,1,1); stem(n1,x1);
axis([min(n1)-1,max(n1)+1,min(x1)-2,max(x1)+2]);
xlabel('n'); ylabel('x1(n)'); title('Sequence x1(n)');
ntick = [n1(1):1:n1(length(n1))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10);
```

The plot of $x_1(n)$ is shown in Figure 2.4.

(b) $x_2(n) = 5x(5+n) + 4x(n+4) + 3x(n)$.

```
% (b) x2(n) = 5*x(5+n) + 4*x(n+4) +3*x(n)
[x21,n21] = sigshift(5*x,n,-5);
[x22,n22] = sigshift(4*x,n,-4);
[x23,n23] = sigadd(x21,n21,x22,n22);
[x2,n2] = sigadd(x23,n23,3*x,n);
subplot(2,1,2); stem(n2,x2);
axis([min(n2)-1,max(n2)+1,min(x2)-0.5,max(x2)+0.5]);
xlabel('n'); ylabel('x2(n)'); title('Sequence x2(n)');
ntick = [n2(1):1:n2(length(n2))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

The plot of $x_2(n)$ is shown in Figure 2.4.

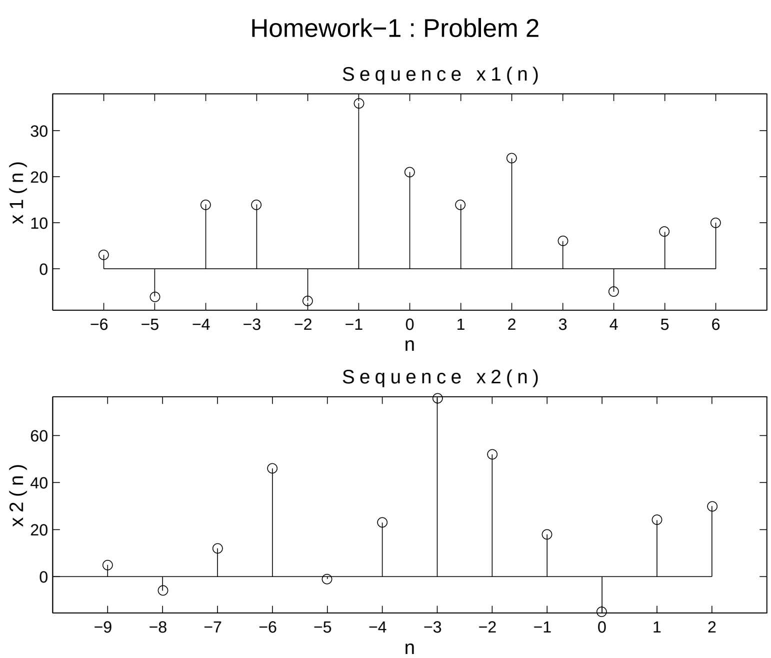


Figure 2.4: Problem P2.2 Sequence Plots

(c) $x_3(n) = x(n+4)x(n-1) + x(2-n)x(n)$.

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P2.2cd');
n = [-4:2]; x = [1,-2,4,6,-5,8,10]; % given seq x(n)
%
% (c) x3(n) = x(n+4)*x(n-1) + x(2-n)*x(n)
[x31,n31] = sigshift(x,n,-4);          % shift x(n) by -4
[x32,n32] = sigshift(x,n,1);           % shift x(n) by 1
[x33,n33] = sigmult(x31,n31,x32,n32);  % multiply two sequences
[x34,n34] = sigfold(x,n);              % fold x(n)
[x34,n34] = sigshift(x34,n34,2);       % shift x(-n) ny 2
[x34,n34] = sigmult(x34,n34,x,n);      % shift x(-n) ny 2
  [x3,n3] = sigadd(x33,n33,x34,n34);   % add two sequences
subplot(2,1,1); stem(n3,x3);
axis([min(n3)-1,max(n3)+1,min(x3)-2,max(x3)+2]);
xlabel('n'); ylabel('x3(n)'); title('Sequence x3(n)');
ntick = [n3(1):1:n3(length(n3))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10);
```

The plots of $x_3(n)$ is shown in Figure 2.5.

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

```
% (d) x4(n) = 2*exp(0.5*n)*x(n)+cos(0.1*pi*n)*x(n+2); -10 <= n <= 10
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,n4,x43,n43);
  [x4,n4] = sigadd(x42,n42,x44,n44);
subplot(2,1,2); stem(n4,x4);
axis([min(n4)-1,max(n4)+1,min(x4)-0.5,max(x4)+0.5]);
xlabel('n'); ylabel('x4(n)'); title('Sequence x4(n)');
ntick = [n4(1):1:n4(length(n4))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
```

The plots of $x_4(n)$ is shown in Figure 2.5.

(e) $x_5(n) = \sum_{k=1}^{5} nx(n-k)$ where $x(n) = \{1, -2, 4, 6, -5, 8, 10\}$.

```
clear; close all;

n = [-4:2]; x = [1,-2,4,6,-5,8,10]; % given seq x(n)
% (e) x5(n) = sum_{k=1}^{5}n*x(n-k);
[x51,n51] = sigshift(x,n,1); [x52,n52] = sigshift(x,n,2);
[x5,n5] = sigadd(x51,n51,x52,n52);
[x53,n53] = sigshift(x,n,3); [x5,n5] = sigadd(x5,n5,x53,n53);
[x54,n54] = sigshift(x,n,4); [x5,n5] = sigadd(x5,n5,x54,n54);
[x55,n55] = sigshift(x,n,5); [x5,n5] = sigadd(x5,n5,x55,n55);
[x5,n5] = sigmult(x5,n5,n5,n5);
subplot(2,1,2); stem(n5,x5); axis([min(n5)-1,max(n5)+1,min(x5)-2,max(x5)+2]);
xlabel('n'); ylabel('x5(n)'); title('Sequence x5(n)');
ntick = [n5(1):1:n5(length(n5))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10);
```

The plot of $x_5(n)$ is shown in Figure 2.6.

3. Problem **P2.3**: A sequence $x(n)$ is periodic if $x(n+N) = x(n)$ for all $n$. Consider a complex exponential sequence $e^{j\omega_0 n} = e^{j2\pi f_0 n}$.

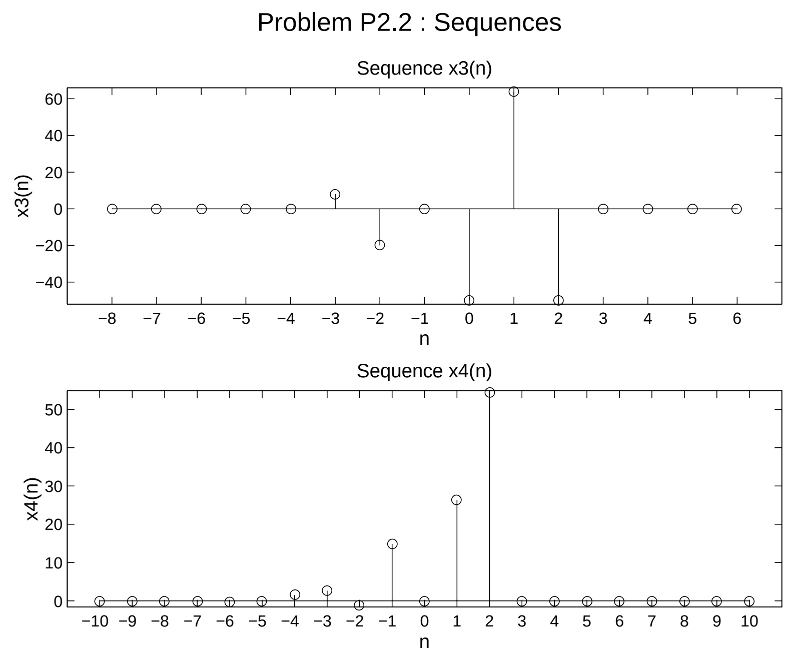


Figure 2.5: Problem P2.2 Sequence Plots

(a) Analytical proof: The above sequence is periodic if

$$e^{j2\pi f_0(n+N)} = e^{j2\pi f_0 n}$$

or

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

which proves the result.

(b) $x_1(n) = \cos(0.3\pi n),\ -20 \le n \le 20.$

```
% (b) x1(n) = cos(0.3*pi*n)
x1 = cos(0.3*pi*n);
subplot(2,1,1); stem(n,x1);
axis([min(n)-1,max(n)+1,-1.1,1.1]);
```

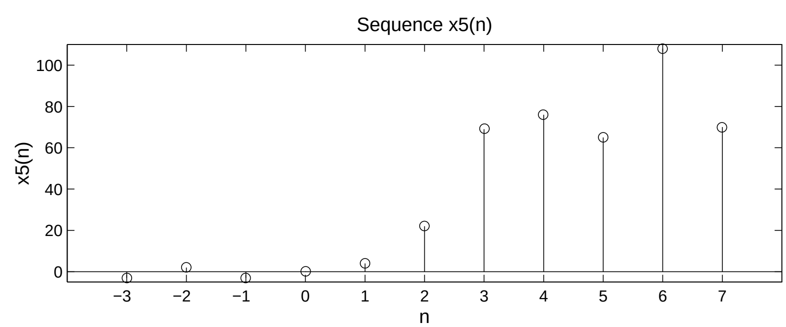


Figure 2.6: Plot of the sequence $x_5(n)$ in Problem P2.5e.

```
ylabel('x1(n)'); title('Sequence cos(0.3*pi*n)');
ntick = [n(1):5:n(length(n))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10);
```

Since $f_0 = 0.3/2 = 3/20$ the sequence is periodic. From the plot in Figure 2.7 we see that in one period of 20 samples $x_1(n)$ exhibits three cycles. This is true whenever $K$ and $N$ are relatively prime.

(c) $x_2(n) = \cos(0.3n),\ -20 \le n \le 20.$

```
% (b) x2(n) = cos(0.3*n)
x2 = cos(0.3*n);
subplot(2,1,2); stem(n,x2);
axis([min(n)-1,max(n)+1,-1.1,1.1]);
ylabel('x2(n)'); title('Sequence cos(0.3*n)');
ntick = [n(1):5:n(length(n))];
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10);
```

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 2.7.

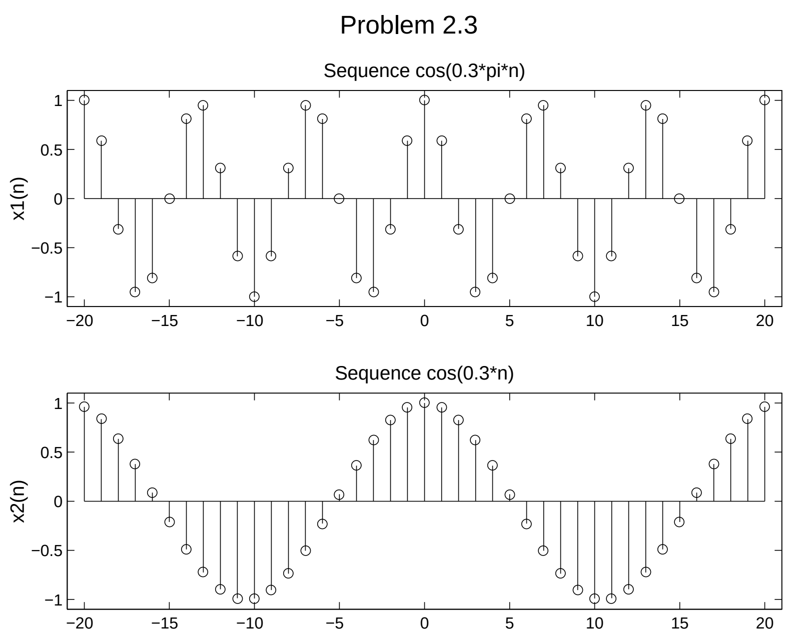


Figure 2.7: Problem P2.3 Sequence Plots

4. Problem **P2.5**: Even-odd decomposition of complex-valued sequences.

(a) MATLAB function `evenodd`:

```
function [xe, xo, m] = evenodd(x,n)
% Complex-valued signal decomposition into even and odd parts
% -----------------------------------------------------------
% [xe, xo, m] = evenodd(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);
```

(b) Even-odd decomposition of $x(n) = 10e^{-(0.4\pi n)}$, $0 \le n \le 10$.

```
n = 0:10; x = 10*exp(-0.4*pi*n);
[xe,xo,neo] = evenodd(x,n);
Re_xe = real(xe); Im_xe = imag(xe);
Re_xo = real(xo); Im_xo = imag(xo);
% Plots of the sequences

subplot(2,2,1); stem(neo,Re_xe);
ylabel('Re{xe(n)}'); title('Real part of Even Seq.');
subplot(2,2,3); stem(neo,Im_xe);
xlabel('n'); ylabel('Im{xe(n)}'); title('Imag part of Even Seq.');
subplot(2,2,2); stem(neo,Re_xo);
ylabel('Re{xo(n)}'); title('Real part of Odd Seq.');
subplot(2,2,4); stem(neo,Im_xo);
xlabel('n'); ylabel('Im{xo(n)}'); title('Imag part of Odd Seq.');
```

The MATLAB verification plots are shown in Figure 2.8.

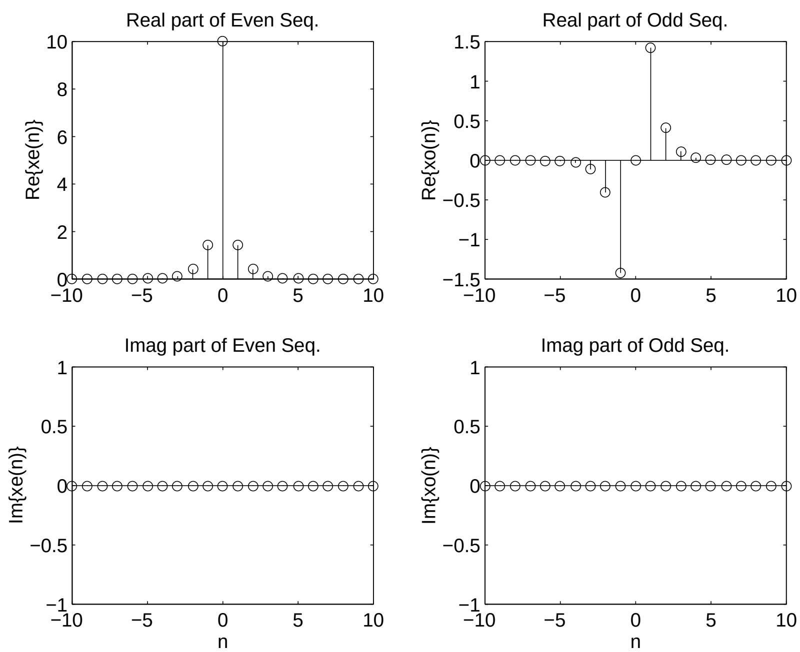


Figure 2.8: Plots in Problem P2.5

5. Problem **P2.12:** Properties of linear convolution.

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

(a) <u>Commutation</u>:

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

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

Association:

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

Distribution:

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

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.

(b) Verification using MATLAB:

```
n1 = -10:20; x1 = n1;
n2 = 0:30; x2 = cos(0.1*pi*n2);
n3 = -5:10; x3 = (1.2).^n3;

% Commutative Property
[y1,ny1] = conv_m(x1,n1,x2,n2);
[y2,ny2] = conv_m(x2,n2,x1,n1);
ydiff = max(abs(y1-y2))
ydiff =
  4.2633e-014
ndiff = max(abs(ny1-ny2))
ndiff =
     0

% Associative 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))
```

```
ydiff =
  6.8212e-013
ndiff = max(abs(ny1-ny2))
ndiff =
     0

% Distributive Property
[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);
[y2,ny2] = sigadd(y2,ny2,y3,ny3);
ydiff = max(abs(y1-y2))
ydiff =
  1.7053e-013
ndiff = max(abs(ny1-ny2))
ndiff =
     0

% Identity Property
n0 = fix(100*(rand(1,1)-0.5));
[dl,ndl] = impseq(n0,n0,n0);
[y1,ny1] = conv_m(x1,n1,dl,ndl);
[y2,ny2] = sigshift(x1,n1,n0);
ydiff = max(abs(y1-y2))
ydiff =
     0
ndiff = max(abs(ny1-ny2))
ndiff =
     0
```

6. Problem **P2.13**: Linear convolution as a matrix-vector multiplication. Consider the sequences

$$x(n) = \{1,2,3,4\} \text{ and } h(n) = \{3,2,1\}$$

(a) The linear convolution of the above two sequences is

$$y(n) = \{3,8,14,20,11,4\}$$

(b) The vector representation of the above operation is:

$$\underbrace{\begin{bmatrix} 3 \\ 8 \\ 14 \\ 20 \\ 11 \\ 4 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}}_{\mathbf{x}}$$

(c) 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

$$[\mathbf{H}]_{i,j} = [\mathbf{H}]_{i-j}$$

which is similar to the definition of time-invariance.

(d) 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.

7. Problem **P2.14**:

(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]'; h = [3,2,1]';
[y,H] = conv_tp(h,x); y = y', H
y =
     3     8    14    20    11     4
H =
     3     0     0     0
     2     3     0     0
     1     2     3     0
     0     1     2     3
     0     0     1     2
     0     0     0     1
```

8. Problem **P2.15:** Let $x(n) = (0.8)^n u(n)$.

(a) Convolution $y(n) = x(n) * x(n)$:

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} x(k)\,x(n-k) = \sum_{k=0}^{\infty} (0.8)^k (0.8)^{n-k} u(n-k) \\ &= \left[\sum_{k=0}^{n} (0.8)^k (0.8)^n (0.8)^{-k}\right] u(n) = (0.8)^n \left[\sum_{k=0}^{n} (8/8)^k\right] u(n) \\ &= (0.8)^n (n+1)\, u(n) = (n+1)(0.8)^n u(n) \end{aligned}$$

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P2.15');
% (a) analytical solution: y(n) = (n+1)*(0.8)^(n+1)*u(n)
na = [0:50]; ya = (na+1).*(0.8).^(na);
subplot(2,1,1); stem(na,ya); axis([-1,51,-1,3]);
xlabel('n'); ylabel('ya(n)'); title('Analytical computation');
```

(b) To use the MATLAB's `filter` funtion we have to represent one of the $x(n)$ sequence by coefficients an equivalent difference equation. See Example 2.10 on page 32 for this procedure. MATLAB solution using the `filter` function:

```
% (b) use of the filter function
nb = [0:50]; x = (0.8).^nb;
yb = filter(1,[1, -0.8],x);
subplot(2,1,2); stem(nb,yb); axis([-1,51,-1,3])
xlabel('n'); ylabel('yb(n)'); title('Filter output');
%
error = max(abs(ya-yb))
error =
  4.4409e-016
%
% Super Title
suptitle('Problem P2.15');
```

The analytical solution to the convolution in (8a) is the exact answer. In the `filter` function approach of (8b), 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 plots of this solution are shown in Figure 2.9.

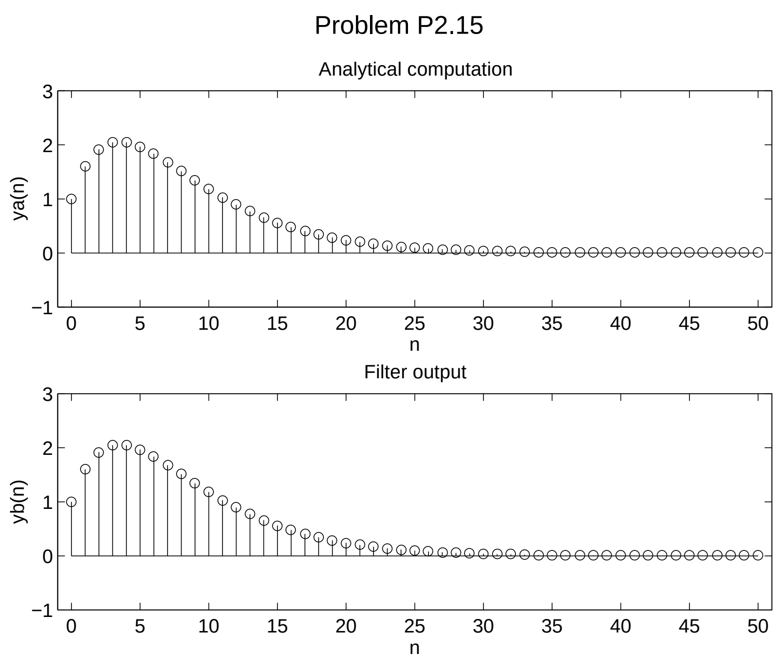


Figure 2.9: Problem P2.15 Plots

# Chapter 3

# Discrete-Time Fourier Transform

1. Problem **P3.1:**

   A MATLAB function to compute DTFT:

   ```
   function [X] = dtft(x,n,w)
   % Computes Discrete-time Fourier Transform
   % [X] = dtft(x,n,w)
   %
   % X = DTFT values computed at w frequencies
   % x = finite duration sequence over n (row vector)
   % n = sample position row vector
   % W = frequency row vector

   X = x*exp(-j*n'*w);
   ```

2. Problem **P3.2**

   (a) Part (c)**:** $x(n) = \left\{ \underset{\uparrow}{4}, 3, 2, 1, 2, 3, 4 \right\}$

   ```
   clear; close all;
   Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
   set(Hf_1,'NumberTitle','off','Name','P3.2c');
   %
   n = 0:6; x = [4,3,2,1,2,3,4];
   w = [0:1:500]*pi/500;
   X = dtft(x,n,w); magX = abs(X); phaX = angle(X);
   %
   % Magnitude Response Plot
   subplot(2,1,1); plot(w/pi,magX);grid;
   xlabel('frequency in pi units'); ylabel('|X|');
   title('Magnitude Response');
   wtick = [0:0.2:1]; magtick = [0;10;20];
   set(gca,'XTickMode','manual','XTick',wtick)
   set(gca,'YTickMode','manual','YTick',magtick)
   %
   % Phase response plot
   subplot(2,1,2); plot(w/pi,phaX*180/pi);grid;
   xlabel('frequency in pi units'); ylabel('Degrees');
   title('Phase Response'); axis([0,1,-180,180])
   wtick = [0:0.2:1]; phatick = [-180;0;180];
   ```

```
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',phatick)
```

The plots are shown in Figure 3.1. The angle plot for this signal is a linear function of ω.

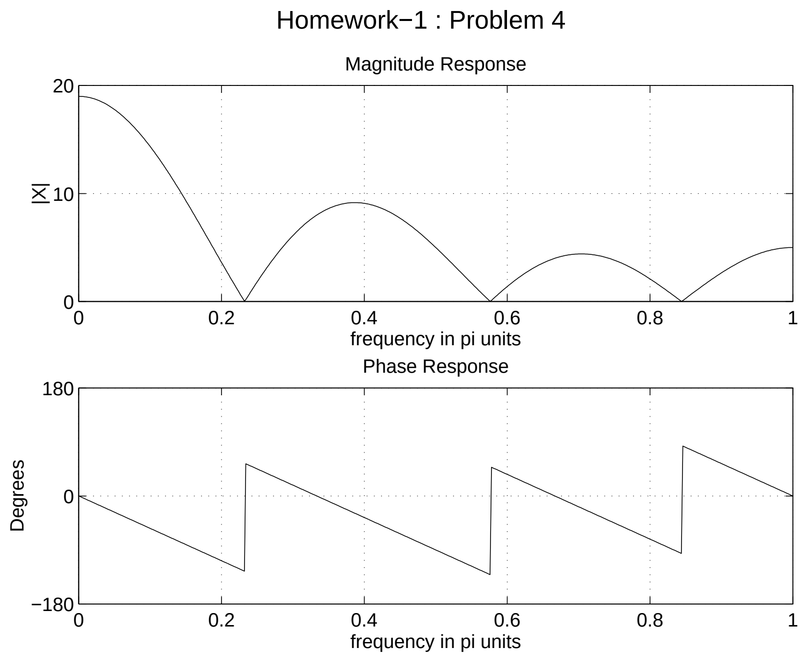


Figure 3.1: Frequency Response Plots in Problem P3.2c

(b) Part (d): $x(n) = \left\{\underset{\uparrow}{4}, 3, 2, 1, 1, 2, 3, 4\right\}$

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P3.2d');
%
% (d) DTFT of x(n) = {4,3,2,1,1,2,3,4}
n = 0:7; x = [4,3,2,1,1,2,3,4];
w = [0:1:500]*pi/500;
X = dtft(x,n,w); magX = abs(X); phaX = angle(X);
%
% Magnitude Response Plot
subplot(2,1,1); plot(w/pi,magX);grid;
xlabel('frequency in pi units'); ylabel('|X|');
title('Magnitude Response');
wtick = [0:0.2:1]; magtick = [0;10;20];
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',magtick)
%
% Phase response plot
subplot(2,1,2); plot(w/pi,phaX*180/pi);grid;
xlabel('frequency in pi units'); ylabel('Degrees');
title('Phase Response'); axis([0,1,-180,180])
wtick = [0:0.2:1]; phatick = [-180;0;180];
```

```
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',phatick)
```

The plots are shown in Figure 3.2. The angle plot for this signal is a linear function of $\omega$.

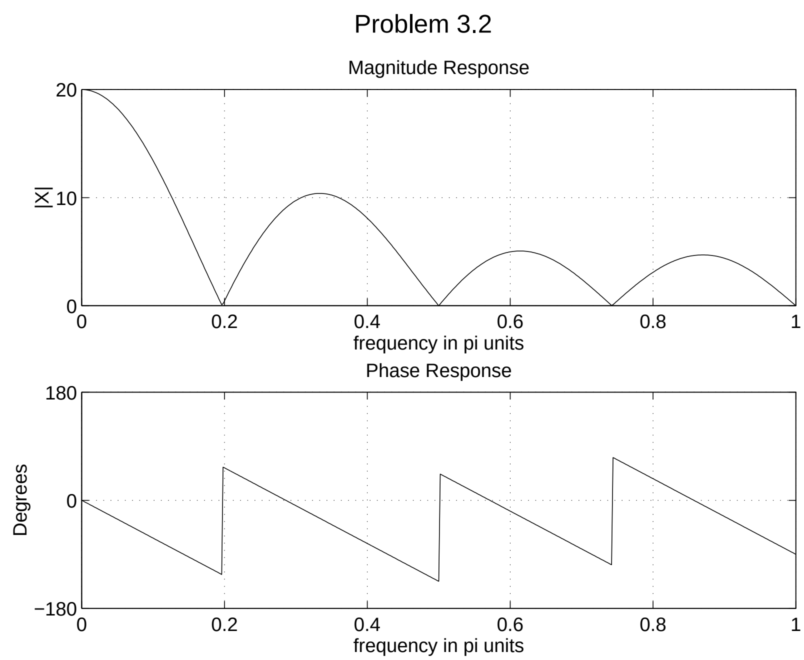


Figure 3.2: Frequency Response Plots in Problem P3.2d

3. Problem **P3.3:**

(a) $x(n) = 3(0.9)^n u(n)$. The DTFT is given by

$$\begin{aligned} X\left(e^{j\omega}\right) &= 3\sum_{n=0}^{\infty}(0.9)^n e^{-j\omega n} = 3\sum_{n=0}^{\infty}\left(0.9e^{-j\omega}\right)^n \\ &= \frac{3}{1-0.9e^{-j\omega}} \end{aligned}$$

MATLAB script for magnitude and angle plot:

```
% Problem P3.3 : Magnitude and Angle Plot of DTFT
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P3.3');
w = [0:200]*pi/200; Z = exp(-j*w); ZZ = Z.^2;

% (a) x(n) = 3*(0.9)^n*u(n)
X = 3*(1-0.9*Z).^(-1); X_mag = abs(X); X_pha = angle(X)*180/pi;
subplot(5,2,1); plot(w/pi,X_mag); axis([0,1,0,max(X_mag)]);
title('Magnitude Plots','fontweight','bold'); ylabel('a.');
set(gca,'YTickMode','manual','YTick',[0,max(X_mag)],'FontSize',10);
subplot(5,2,2); plot(w/pi,X_pha); axis([0,1,-180,180]);
title('Angle Plots','fontweight','bold')
set(gca,'YTickMode','manual','YTick',[-180,0,180],'FontSize',10);
```

The plots are given in Figure 3.3.

(b) $x(n) = 2(0.8)^{n+2} u(n-2)$. The sequence can be written as

$$\begin{aligned} x(n) &= 2(0.8)^{n-2+4} u(n-2) = 2(0.8)^4 (0.8)^{n-2} u(n-2) \\ &= 0.8192 (0.8)^{n-2} u(n-2) \end{aligned}$$

Now using properties of DTFT, we obtain

$$X\left(e^{j\omega}\right) = 0.8192 \frac{e^{-j2\omega}}{1-0.8e^{-j\omega}}$$

MATLAB script for magnitude and angle plot:

```
% (b) x(n) = 2*(0.8)^(n+2)*u(n-2)
X = 0.8192*ZZ./(1-0.8*Z); X_mag = abs(X); X_pha = angle(X)*180/pi;
subplot(5,2,3); plot(w/pi,X_mag); axis([0,1,0,max(X_mag)]);  ylabel('b.');
set(gca,'YTickMode','manual','YTick',[0,max(X_mag)],'FontSize',10);
subplot(5,2,4); plot(w/pi,X_pha); axis([0,1,-180,180]);
set(gca,'YTickMode','manual','YTick',[-180,0,180],'FontSize',10);
```

The plots are given in Figure 3.3.

(c) $x(n) = n(0.5)^n u(n)$. The DTFT of $(0.5)^n u(n)$ is given by $1/\left(1-0.5e^{-j\omega}\right)$, that is,

$$\sum_{n=0}^{\infty} (0.5)^n e^{-j\omega n} = \frac{1}{1-0.5e^{-j\omega}}$$

Differentiating both sides with respect to $\omega$, we obtain

$$-j\sum_{n=0}^{\infty} n(0.5)^n e^{-j\omega n} = (-1)\frac{1}{(1-0.5e^{-j\omega})^2}\left[-0.5e^{-j\omega}(-j)\right]$$

Hence

$$\begin{aligned} X\left(e^{j\omega}\right) &= \mathfrak{F}\{n(0.5)^n u(n)\} = \sum_{n=0}^{\infty} n(0.5)^n e^{-j\omega n} \\ &= \frac{0.5e^{-j\omega}}{(1-0.5e^{-j\omega})^2} \end{aligned}$$

MATLAB script for magnitude and angle plot:

```
% (c) x(n) = n*(0.5)^n*u(n)
X = (0.5)*Z./((1-0.5*Z).^(2)); X_mag = abs(X); X_pha = angle(X)*180/pi;
subplot(5,2,5); plot(w/pi,X_mag); axis([0,1,0,max(X_mag)]);  ylabel('c.');
set(gca,'YTickMode','manual','YTick',[0,max(X_mag)],'FontSize',10);
subplot(5,2,6); plot(w/pi,X_pha); axis([0,1,-180,180]);
set(gca,'YTickMode','manual','YTick',[-180,0,180],'FontSize',10);
```

The plots are given in Figure 3.3.

(d) $x(n) = (n+2)(-0.7)^{n-1} u(n-2)$. The sequence $x(n)$ can be arranged as

$$\begin{aligned} x(n) &= (n-2+4)(-0.7)^{n-2+1} u(n-2) \\ &= (-0.7)(n-2+4)(-0.7)^{n-2} u(n-2) \\ &= (-0.7)(n-2)(-0.7)^{n-2} u(n-2) + 4(-0.7)(-0.7)^{n-2} u(n-2) \\ &= (-0.7)(n-2)(-0.7)^{n-2} u(n-2) - 2.8(-0.7)^{n-2} u(n-2) \end{aligned}$$

Using properties and the result from part (c), we obtain,

$$
\begin{aligned}
X\left(e^{j\omega}\right) &= (-0.7)\,e^{-j2\omega}\frac{(-0.7)\,e^{-j\omega}}{[1+0.7e^{-j\omega}]^2} - 2.8\frac{e^{-j\omega}}{1+0.7e^{-j\omega}} \\
&= \frac{0.49e^{-j3\omega}}{[1+0.7e^{-j\omega}]^2} - 2.8\frac{e^{-j\omega}}{1+0.7e^{-j\omega}}
\end{aligned}
$$

MATLAB script for magnitude and angle plot:

```
% (d) x(n) = (n+2)*(-0.7).^(n-1)*u(n-1)
X = (0.49)*(Z.*ZZ)./((1+0.7*Z).^2) - (2.8*ZZ)./(1+0.7*Z);
X_mag = abs(X); X_pha = angle(X)*180/pi;
subplot(5,2,7); plot(w/pi,X_mag); axis([0,1,0,max(X_mag)]);  ylabel('d.');
set(gca,'YTickMode','manual','YTick',[0,max(X_mag)],'FontSize',10);
subplot(5,2,8); plot(w/pi,X_pha); axis([0,1,-180,180]);
set(gca,'YTickMode','manual','YTick',[-180,0,180],'FontSize',10);
```

The plots are given in Figure 3.3.

(e) $x(n) = 5\,(-0.9)^n \cos(0.1\pi n)\,u(n)$. The sequence can be written as

$$
\begin{aligned}
x(n) &= 5\,(-0.9)^n\,\frac{e^{j0.1\pi n}+e^{-j0.1\pi n}}{2}u(n) \\
&= \frac{5}{2}\left(-0.9e^{j0.1\pi}\right)^n u(n) + \frac{5}{2}\left(-0.9e^{-j0.1\pi}\right)^n u(n)
\end{aligned}
$$

Hence the DTFT is given by

$$
\begin{aligned}
X\left(e^{j\omega}\right) &= \frac{5/2}{1+0.9e^{j0.1\pi}e^{-j\omega}} + \frac{5/2}{1+0.9e^{-j0.1\pi}e^{-j\omega}} \\
&= \frac{1+0.9\cos(0.1\pi)\,e^{-j\omega}}{1+1.8\cos(0.1\pi)\,e^{-j\omega}+0.81e^{-j2\omega}}
\end{aligned}
$$

MATLAB script for magnitude and angle plot:

```
% (e) x(n) = 5*(-0.9).^n*cos(0.1*pi*n)*u(n)
X = (1+0.9*cos(0.1*pi)*Z)./(1+1.8*cos(0.1*pi)*Z+(0.81)*ZZ);
X_mag = abs(X); X_pha = angle(X)*180/pi;
subplot(5,2,9); plot(w/pi,X_mag); axis([0,1,0,max(X_mag)]);  ylabel('e.');
set(gca,'XTickMode','manual','XTick',[0,0.5,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[0,max(X_mag)],'FontSize',10);
subplot(5,2,10); plot(w/pi,X_pha); axis([0,1,-180,180]);
set(gca,'XTickMode','manual','XTick',[0,0.5,1],'FontSize',10);
```

The plots are given in Figure 3.3.

4. Problem **P3.4:** This problem is solved using MATLAB.

```
% Problem P3.4 : DTFT of Rectangular pulse
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P3.4');
w = [-100:100]*pi/100;

% x(n) = -N:N;
% (a) N = 5
N = 5; n = -N:N; x = ones(1,length(n)); X = dtft(x,n,w); X = real(X); X = X/max(X);
```

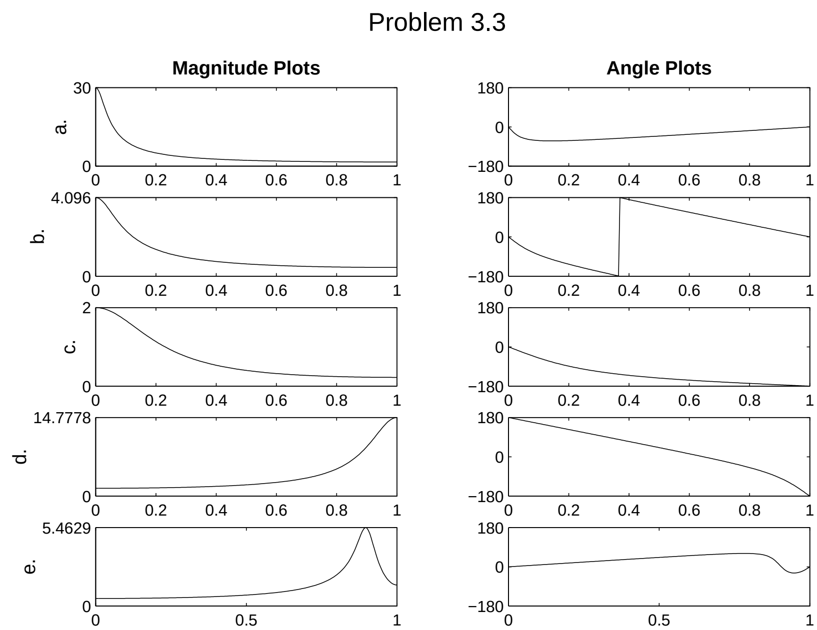


Figure 3.3: Problem P3.3 DTFT Plots

```
subplot(2,2,1); plot(w/pi,X); axis([-1,1,min(X),1]);
title('DTFT for N = 5','fontweight','bold'); ylabel('X');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (b) N = 15
N = 15; n = -N:N; x = ones(1,length(n)); X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,2); plot(w/pi,X); axis([-1,1,min(X),1]);
title('DTFT for N = 15','fontweight','bold'); ylabel('X');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (c) N = 25
N = 25; n = -N:N; x = ones(1,length(n)); X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,3); plot(w/pi,X); axis([-1,1,min(X),1]);
title('DTFT for N = 25','fontweight','bold'); ylabel('X');
xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (d) N = 100
N = 100; n = -N:N; x = ones(1,length(n)); X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,4); plot(w/pi,X); axis([-1,1,min(X),1]);
title('DTFT for N = 100','fontweight','bold'); ylabel('X');
xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;
```

```
% Super Title
suptitle('Problem 3.4');
```

The plots are shown in Figure 3.4. These plots show that the DTFT of a rectangular pulse is similar to a sinc function and as $N$ increases the function becomes narrowrer and narrower.

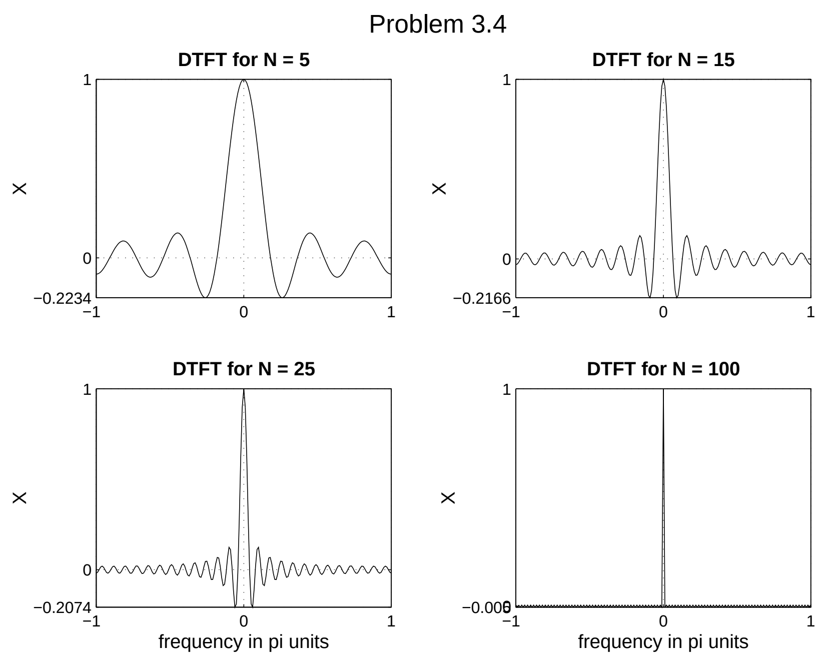


Figure 3.4: Problem P3.4 DTFT Plots

5. Problem **P3.5:** DTFT of a symmetric traingular pulse

$$\mathcal{T}_N = \left[1 - \frac{|n|}{N}\right] \mathcal{R}_N(n)$$

This problem is solved using MATLAB. It uses the function dtft described in P3.1 ans is given below.

```
function [X] = dtft(x,n,w)
% Computes Discrete-time Fourier Transform
% [X] = dtft(x,n,w)
%
% X = DTFT values computed at w frequencies
% x = finite duration sequence over n (row vector)
% n = sample position row vector
% W = frequency row vector
X = x*exp(-j*n'*w);
```

The MATLAB script for the problem is:

```
clear; close all;

w = [-100:100]*pi/100;
```

```
% x(n) = (1-abs(n)/N)*R_N(n);
% (a) N = 5
N = 5; n = -N:N; x = 1-abs(n)/N; X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,1); plot(w/pi,X); axis([-1,1,0,1]);
title('DTFT for N = 5','fontweight','bold'); ylabel('X');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (b) N = 15
N = 15; n = -N:N; x = 1-abs(n)/N; X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,2); plot(w/pi,X); axis([-1,1,0,1]);
title('DTFT for N = 15','fontweight','bold'); ylabel('X');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (c) N = 25
N = 25; n = -N:N; x = 1-abs(n)/N; X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,3); plot(w/pi,X); axis([-1,1,0,1]);
title('DTFT for N = 25','fontweight','bold'); ylabel('X');
xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% (d) N = 100
N = 100; n = -N:N; x = 1-abs(n)/N; X = dtft(x,n,w); X = real(X); X = X/max(X);
subplot(2,2,4); plot(w/pi,X); axis([-1,1,0,1]);
title('DTFT for N = 100','fontweight','bold'); ylabel('X');
xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[-1,0,1],'FontSize',10);
set(gca,'YTickMode','manual','YTick',[min(X),0,1],'FontSize',10); grid;

% Super Title
suptitle('Problem 3.5');
```

The DTFT plots are shown in Figure 3.5. These plots show that the DTFT of a triangular pulse is similar to a $(\text{sinc})^2$ function and as $N$ increases the function becomes narrowrer and narrower.

6. Problem **P3.7:** Consider $x_e(n) = \frac{1}{2}[x(n) + x^*(-n)]$. Using $\mathcal{F}[x^*(-n)] = X^*\left(e^{j\omega}\right)$ (see page 48, properties 4 and 5) we obtain

$$\begin{aligned}
\mathcal{F}[x_e(n)] &= \mathcal{F}\left[\frac{1}{2}\{x(n)+x^*(-n)\}\right] = \frac{1}{2}\{\mathcal{F}[x(n)] + \mathcal{F}[x^*(-n)]\} \\
&= \frac{1}{2}\left\{X\left(e^{j\omega}\right) + X^*\left(e^{j\omega}\right)\right\} \triangleq X_R\left(e^{j\omega}\right).
\end{aligned}$$

Similarly

$$\begin{aligned}
\mathcal{F}[x_o(n)] &= \mathcal{F}\left[\frac{1}{2}\{x(n)-x^*(-n)\}\right] = \frac{1}{2}\{\mathcal{F}[x(n)] - \mathcal{F}[x^*(-n)]\} \\
&= \frac{1}{2}\left\{X\left(e^{j\omega}\right) - X^*\left(e^{j\omega}\right)\right\} \triangleq jX_I\left(e^{j\omega}\right).
\end{aligned}$$

MATLAB verification using $x(n) = e^{j0.1\pi n}[u(n) - u(n-20)]$:

```
clear; close all;
%
```

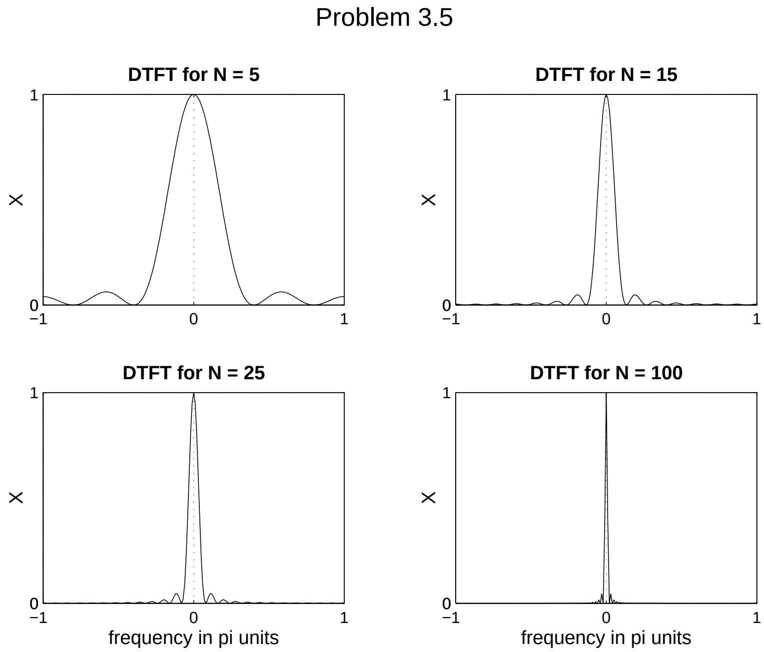


Figure 3.5: DTFT plots in Problem P3.5.

```
n = 0:20; x = exp(0.1*pi*n);
w = [-100:100]*pi/100; X = dtft(x,n,w);
XR = real(X); XI = imag(X);
[xe,xo,neo] = evenodd(x,n);
Xe = dtft(xe,neo,w); Xo = dtft(xo,neo,w);
diff_e = max(abs(XR-Xe))
diff_e =
  5.5511e-017
diff_o = max(abs(j*XI-Xo))
diff_o =
  6.9389e-017
```

7. Problem **P3.16**

   (a) Part (b)**:** A digital filter is described by the difference equation

$$y(n) = x(n) + 2x(n-1) + x(n-2) - 0.5y(n-1) + 0.25y(n-2)$$

The frequency response $H(e^{j\omega})$ of the filter: Substituting $x(n) = e^{j\omega n}$ and $y(n) = H\left(e^{j\omega n}\right)e^{j\omega n}$ in the above difference equation and simplifying

$$H\left(e^{j\omega}\right) = \frac{1 + 2e^{-j\omega} + e^{-j2\omega}}{1 + 0.5e^{-j\omega} - 0.25e^{-j2\omega}}$$

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P3.16b');
%
% Analytical calculations of Frequency Response using diff eqn
b = [1,2,1]; a = [1,0.5,0.25];
w = [0:1:500]*pi/500; kb = 0:length(b)-1; ka = 0:length(a)-1;
```

```
H = (b*exp(-j*kb'*w)) ./ (a*exp(-j*ka'*w));
magH = abs(H); phaH = angle(H);
%
% Magnitude Responses Plot
subplot(2,1,1); plot(w/pi,magH);grid; axis([0,1,0,3]);
xlabel('frequency in pi units'); ylabel('|H|');
title('Magnitude Response');
wtick = [0:0.2:1]; magtick = [0:3];
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',magtick)
%
% Phase response plot
subplot(2,1,2); plot(w/pi,phaH*180/pi);grid;
xlabel('frequency in pi units'); ylabel('Degrees');
title('Phase Response'); axis([0,1,-180,180])
wtick = [0:0.2:1]; phatick = [-180;0;180];
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',phatick)
```

The magnitude and phase response plots are shown in Figure 3.6.

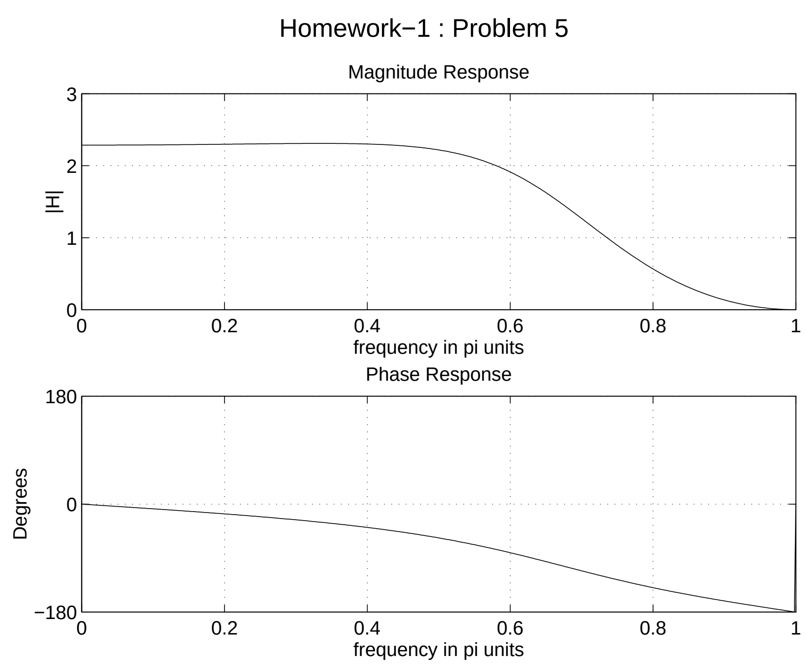


Figure 3.6: Problem P3.16b Plots

(b) Part (c)**:** A digital filter is described by the difference equation

$$y(n) = 2x(n) + x(n-1) - 0.25y(n-1) + 0.25y(n-2)$$

The frequency response $H(e^{j\omega})$ of the filter: Substituting $x(n) = e^{j\omega n}$ and $y(n) = H\left(e^{j\omega n}\right)e^{j\omega n}$ in the above difference equation and simplifying

$$H\left(e^{j\omega}\right) = \frac{2 + e^{-j\omega}}{1 + 0.25e^{-j\omega} - 0.25e^{-j2\omega}}$$

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P3.16c');
%
% Analytical calculations of Frequency Response using diff eqn
b = [2,1]; a = [1,0.25,-0.25];
w = [0:1:500]*pi/500; kb = 0:length(b)-1; ka = 0:length(a)-1;
H = (b*exp(-j*kb'*w)) ./ (a*exp(-j*ka'*w));
magH = abs(H); phaH = angle(H);
%
% Magnitude Responses Plot
subplot(2,1,1); plot(w/pi,magH);grid; axis([0,1,0,3]);
xlabel('frequency in pi units'); ylabel('|H|');
title('Magnitude Response');
wtick = [0:0.2:1]; magtick = [0:3];
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',magtick)
%
% Phase response plot
subplot(2,1,2); plot(w/pi,phaH*180/pi);grid;
xlabel('frequency in pi units'); ylabel('Degrees');
title('Phase Response'); axis([0,1,-180,180])
wtick = [0:0.2:1]; phatick = [-180;0;180];
set(gca,'XTickMode','manual','XTick',wtick)
set(gca,'YTickMode','manual','YTick',phatick)
```

The magnitude and phase response plots are shown in Figure 3.7.

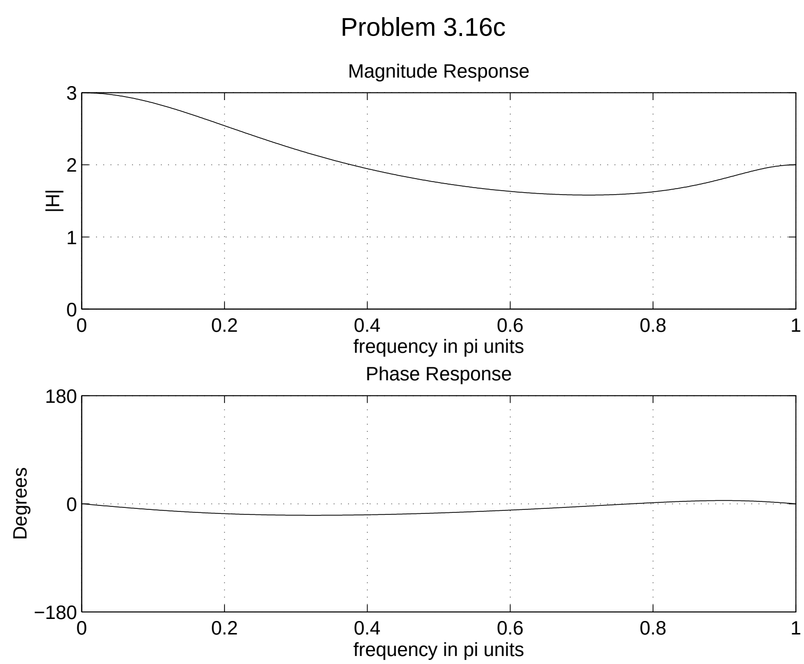


Figure 3.7: Problem P3.16c Plots

8. Problem **P3.19**: Sampling frequency $F_s = 100$ sam/sec (or sampling interval $T_s = 0.01$ sec/sam) and impulse response $h(n) = (0.5)^n u(n)$.

(a) $x_a(t) = 3\cos(20\pi t)$. Hence $x(n) = x_a(nT_s) = 3\cos(0.2\pi n)$. Therefore, the digital frequency is $0.2\pi$ rad/sam.

(b) The steady-state response when $x(n) = 3\cos(0.2\pi n)$: The frequency response is

$$H\left(e^{j\omega}\right) = F[h(n)] = F\left[(0.5)^n u(n)\right] = \frac{1}{1-0.5e^{j\omega}}.$$

At $\omega = 0.2\pi$, the response is

$$H\left(e^{j0.2\pi}\right) = \frac{1}{1-0.5e^{j0.2\pi}} = 0.6969\left(\angle -0.2063^c\right).$$

Hence

$$y_{ss}(n) = 3(0.6969)\cos(0.2\pi n - 0.2363)$$

which after D/A conversion gives $y_{ss}(t)$ as

$$y_{ss,a}(t) = 2.0907\cos(20\pi t - 0.2363).$$

(c) The steady-state DC gain is obtained by setting $\omega = 0$ which is equal to $H\left(e^{j0}\right) = 2$. Hence $y_{ss}(n) = 3(2) = y_{ss,a}(t) = 6$.

(d) Aliased frequencies of $F_0$ for the given sampling rate $F_s$ are $F_0 + kF_s$. Now for $F_0 = 10$ Hz and $F_s = 100$, the aliased frequencies are $10 + 100k = \{110, 210, \ldots\}$. Therefore, two other $x_a(t)$'s are

$$3\cos(220\pi t) \text{ and } 3\cos(420\pi t).$$

(e) The prefilter should be a lowpass filter with the cutoff frequency of 50 Hz.

9. Problem **P3.20**: An analog signal $x_a(t) = \cos(20\pi t)$, $0 \le t \le 1$ is sampled at $T_s = 0.01$, 0.05, and 0.1 sec intervals.

(a) Plots of $x(n)$ for each $T_s$. MATLAB script:

```
clear; close all;
%
t = 0:0.001:1; xa = cos(20*pi*t);
% (a) Plots of sequences

Ts = 0.01; N1 = round(1/Ts); n1 = 0:N1; x1 = cos(20*pi*n1*Ts);
subplot(3,1,1); plot(t,xa,n1*Ts,x1,'o'); axis([0,1,-1.1,1.1]);
ylabel('x1(n)'); title('Sampling of xa(t) using Ts=0.01');
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 0.05; N2 = round(1/Ts); n2 = 0:N2; x2 = cos(20*pi*n2*Ts);
subplot(3,1,2); plot(t,xa,n2*Ts,x2,'o'); axis([0,1,-1.1,1.1]);
ylabel('x2(n)'); title('Sampling of xa(t) using Ts=0.05');
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 00.1; N3 = round(1/Ts); n3 = 0:N3; x3 = cos(20*pi*n3*Ts);
subplot(3,1,3); plot(t,xa,n3*Ts,x3,'o'); axis([0,1,-1.1,1.1]);
ylabel('x3(n)'); title('Sampling of xa(t) using Ts=0.1');
set(gca,'xtickmode','manual','xtick',[0:1]);xlabel('t,sec');
```

The plots are shown in Figure 3.8.

(b) Reconstruction from $x(n)$ using the sinc interpolation. MATLAB script:

```
% (b) Reconstruction using sinc function

Ts = 0.01; Fs = 1/Ts;
xa1 = x1*sinc(Fs*(ones(length(n1),1)*t-(n1*Ts)'*ones(1,length(t))));
subplot(3,1,1);plot(t,xa1); axis([0,1,-1.1,1.1]);
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.01');
```

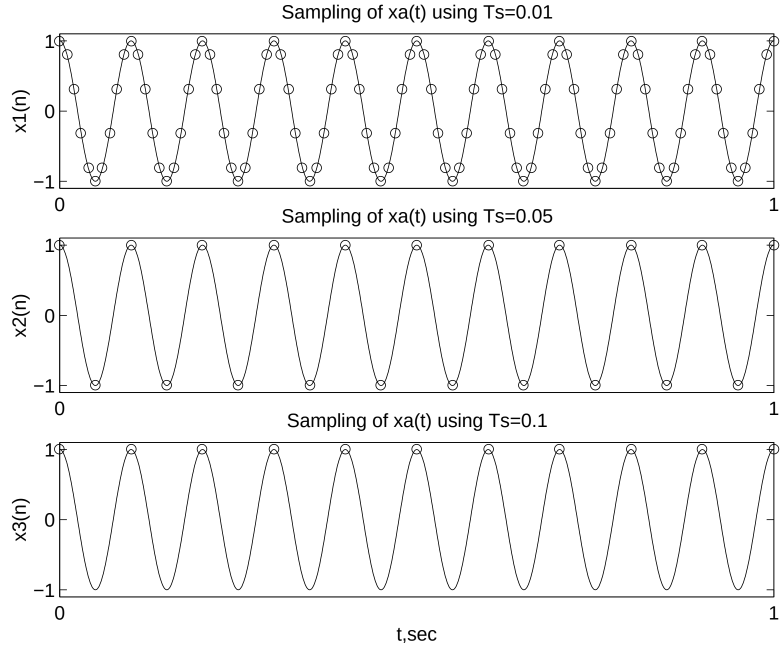


Figure 3.8: Plots of $x(n)$ for various $T_s$ in Problem P3.20a.

```
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 0.05; Fs = 1/Ts;
xa2 = x2*sinc(Fs*(ones(length(n2),1)*t-(n2*Ts)'*ones(1,length(t))));
subplot(3,1,2);plot(t,xa2); axis([0,1,-1.1,1.1]);
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.05');
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 0.1; Fs = 1/Ts;
xa3 = x3*sinc(Fs*(ones(length(n3),1)*t-(n3*Ts)'*ones(1,length(t))));
subplot(3,1,3);plot(t,xa3); axis([0,1,-1.1,1.1]);
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.1');
set(gca,'xtickmode','manual','xtick',[0:1]);xlabel('t,sec');
```

The reconstruction is shown in Figure 3.9.

(c) Reconstruction from $x(n)$ using the cubic spline interpolation. MATLAB script:

```
% (c) Reconstruction using cubic spline interpolation

Ts = 0.01; Fs = 1/Ts;
xa1 = spline(Ts*n1,x1,t);
subplot(3,1,1);plot(t,xa1); axis([0,1,-1.1,1.1]);
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.01');
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 0.05; Fs = 1/Ts;
xa2 = spline(Ts*n2,x2,t);
subplot(3,1,2);plot(t,xa2); axis([0,1,-1.1,1.1]);
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.05');
set(gca,'xtickmode','manual','xtick',[0:1]);
Ts = 0.1; Fs = 1/Ts;
xa3 = spline(Ts*n3,x3,t);
subplot(3,1,3);plot(t,xa3); axis([0,1,-1.1,1.1]);
```

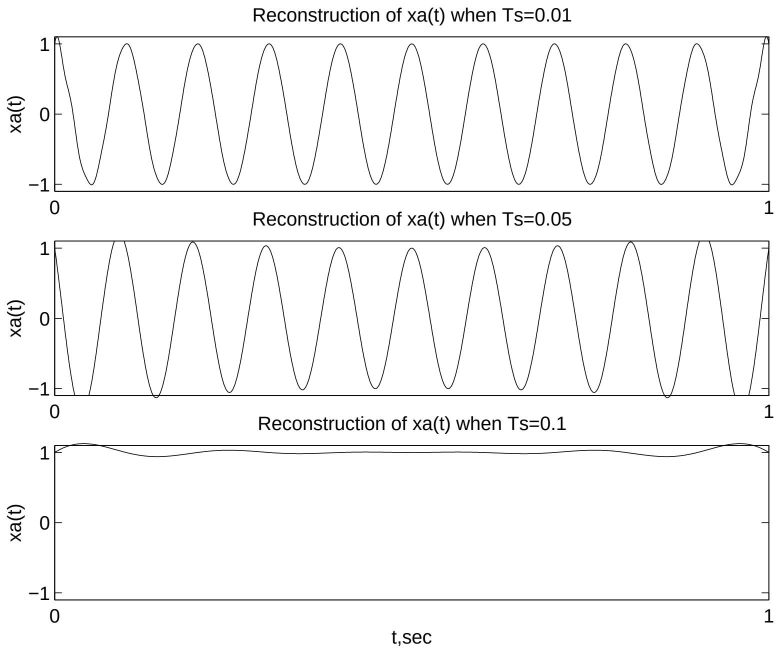


Figure 3.9: The sinc interpolation in Problem P3.20b.

```
ylabel('xa(t)'); title('Reconstruction of xa(t) when Ts=0.1');
set(gca,'xtickmode','manual','xtick',[0:1]);xlabel('t,sec');
```

The reconstruction is shown in Figure 3.10.

(d) Comments: From the plots in Figures it is clear that reconstructions from samples at $T_s = 0.01$ and 0.05 depict the original freuwncy (excluding end effects) but reconstructions for $T_s = 0.1$ show the original frequency aliased to zero. Furthermore, the cubic spline interpolation is a better recosntruction than the sinc interpolation, that is, the sinc interpolation is more susceptible to boundary effect.

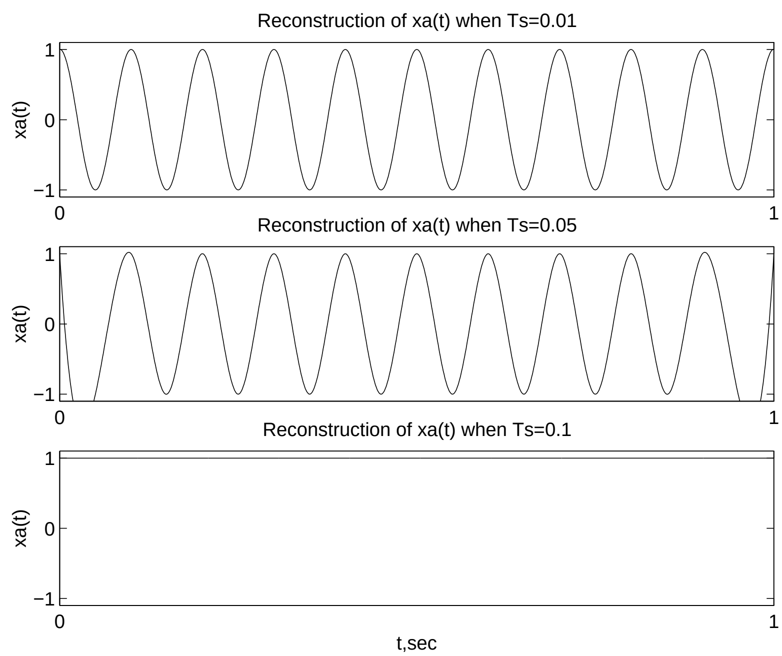


Figure 3.10: The cubic spline interpolation in Problem P3.20c.

# Chapter 4

# The $z$-transform

1. Problem **P4.1**

   (a) Part (c): The given sequence is $x(n) = (4/3)^n u(1-n)$. Hence the $z$-transform is

$$
\begin{aligned}
X(z) &= \sum_{-\infty}^{1} \left(\frac{4}{3}\right)^n z^{-n} = \sum_{-\infty}^{1} \left(\frac{4}{3z}\right)^n \\
&= \sum_{-1}^{\infty} \left(\frac{3z}{4}\right)^n = \frac{4}{3z} \sum_{0}^{\infty} \left(\frac{3z}{4}\right)^n \\
&= \frac{4}{3z} \frac{1}{1-(3z/4)} = \frac{-16/9}{z(z-4/3)}, \ |z| < \frac{4}{3}
\end{aligned}
$$

MATLAB verification: Since the sequence is a left-sided sequence, the difference equation (and hence the `filter` function) should be run backward in time. This means that $X(z)$ should be a rational function in $z$. Furthermore, since $x(n)$ begins at $n = 1$ and continues (backwards in time) to $-\infty$, we will advance $x(n)$ by one sample and simulate $zX(z)$ for verification purposes where

$$zX(z) = \frac{4/3}{1-(3/4)z}$$

from the first term on the right-hand side of $X(z)$ above.

```
%(c) x(n) = (4/3)^n*u(1-n)
b = [4/3]; a = [1, -3/4];            % Difference equation
delta = [1, zeros(1,7)];              % Input sequence

% filter solution
x = filter(b,a,delta)
x =
  Columns 1 through 7
    1.3333    1.0000    0.7500    0.5625    0.4219    0.3164    0.2373
  Column 8
    0.1780

% simulation of x(n)
n = [1:-1:-6]; x = (4/3).^n
x =
  Columns 1 through 7
    1.3333    1.0000    0.7500    0.5625    0.4219    0.3164    0.2373
  Column 8
    0.1780
```

(b) Part (d): The given sequence $x(n) = 2^{-|n|} + 3^{-|n|}$ can be rearranged as

$$x(n) = 2^{-n}u(n) - [-2^{n}u(-n-1)] + 3^{-n}u(n) - [-3^{n}u(-n-1)]$$

i. The $z$-transform is

$$\begin{aligned} X(z) &= \underbrace{\frac{1}{1-2^{-1}z^{-1}}}_{|z|>2^{-1}} - \underbrace{\frac{1}{1-2z^{-1}}}_{|z|<2} + \underbrace{\frac{1}{1-3^{-1}z^{-1}}}_{|z|>3^{-1}} - \underbrace{\frac{1}{1-3z^{-1}}}_{|z|<3} \\ &= \underbrace{\frac{-2+5z^{-1}}{1-5z^{-1}+6z^{-2}}}_{|z|<2} + \underbrace{\frac{2-(5/6)z^{-1}}{1-(5/6)z^{-1}+(1/6)z^{-2}}}_{|z|>2^{-1}} \end{aligned} \tag{4.1}$$

which after simplification becomes

$$X(z) = \frac{-4.1667z^{-1} + 11.6667z^{-2} - 4.1667z^{-3} + 0.5z^{-4}}{1 - 5.8333z^{-1} + 10.3333z^{-2} - 5.8333z^{-3} + 1.0000z^{-4}}, \; 0.5 < |z| < 2$$

ii. MATLAB verification: Since the sequence is two-sided, it is not possible to drive the difference equation in both directions using the `filter` function. However we can verify the step in (4.1) by generating positive-time and negative-time sequences as shown below.

```
%(d) x(n) = (2)^(-|n|)+(3)^(-|n|)
R = [1;-1;1;-1];                    % residues
p = [1/2;2;1/3;3];                  % poles
[b,a] = residuez(R,p,[])            % Difference equation coefficients
b =
         0   -4.1667   11.6667   -4.1667
a =
    1.0000   -5.8333   10.3333   -5.8333    1.0000

% Forward difference equation
Rf = [1;1]; pf = [1/2;1/3];
[bf,af] = residuez(Rf,pf,[])
bf =
    2.0000   -0.8333
af =
    1.0000   -0.8333    0.1667
[delta,nf]= impseq(0,0,30);
xf = filter(bf,af,delta);

% Backward difference equation
Rb = [-1;-1]; pb = [2;3];
[bb,ab] = residuez(Rb,pb,[])
bb =
    -2     5
ab =
     1    -5     6
[delta,nb]= impseq(0,0,29);
xb = filter(fliplr(bb),fliplr(ab),delta);

% Total solution
x1 = [fliplr(xb),xf];

% simulation of x(n)
n = [-fliplr(nb+1),nf];
```

```
x2 = 2.^(-abs(n)) + 3.^(-abs(n));

% difference
diff = max(abs(x1-x2))
diff =
  1.1102e-016
```

2. Problem **P4.2**

   (a) Part (b)**:**The given sequence $x(n)$ can be rearranged as

$$\begin{aligned} x(n) &= \left(\frac{1}{3}\right)^n u(n-2) + (0.9)^{n-3} u(n) \\ &= \left(\frac{1}{3}\right)^2 \left(\frac{1}{3}\right)^{n-2} u(n-2) + (0.9)^{-3} (0.9)^n u(n) \\ &= \frac{1}{9} \left(\frac{1}{3}\right)^{n-2} u(n-2) + \frac{1000}{729} (0.9)^n u(n) \end{aligned}$$

The $z$-transform is

$$\begin{aligned} X(z) &= \frac{1}{9} z^{-2} \mathcal{Z}\left[\left(\frac{1}{3}\right)^n u(n)\right] + \frac{1000}{729} \mathcal{Z}[(0.9)^n u(n)] \\ &= \frac{1}{9} z^{-2} \left(\frac{1}{1-\frac{1}{3}z^{-1}}\right) + \frac{1000}{729} \left(\frac{1}{1-0.9z^{-1}}\right) \end{aligned}$$

which after simplification becomes

$$X(z) = \frac{\frac{1000}{729} - \frac{1000}{2187}z^{-1} + \frac{1}{9}z^{-2} - 0.1z^{-3}}{1 - \frac{37}{30}z^{-1} + 0.3z^{-2}}$$

MATLAB verification:

```
% Sequence:
% x(n) = (1/3)^n*u(n-2) + (0.9)^(n-3)*u(n)
%
% Analytical Expression of X(z)
% X(z) = ((1000/729) - (1000/2187)*z^(-1) + (1/9)*z^(-2) -0.1*z^(-3)
% X(z) = ---------------------------------------------------
%                  1 - (37/30)*z^(-1) + 0.3*z^(-2)

% Matlab verification
b = [1000/729, -1000/2187, 1/9, -0.1]; a = [1, -37/30, 0.3];
delta = impseq(0,0,7); format long
xb1 = filter(b,a,delta)
xb1 =
  Columns 1 through 4
   1.37174211248285   1.23456790123457   1.22222222222222   1.03703703703704
  Columns 5 through 8
   0.91234567901235   0.81411522633745   0.73037174211248   0.65655724737083
%
% check
n = 0:7;
xb2 = ((1/3).^n).*stepseq(2,0,7) + ((0.9).^(n-3)).*stepseq(0,0,7)
xb2 =
  Columns 1 through 4
   1.37174211248285   1.23456790123457   1.22222222222222   1.03703703703704
```

```
  Columns 5 through 8
   0.91234567901235   0.81411522633745   0.73037174211248   0.65655724737083
%
error = abs(max(xb1-xb2)), format short;
error =
    4.440892098500626e-016
```

(b) Part (d)**:**The given sequence $x(n)$ can be rearranged as

$$\begin{aligned} x(n) &= \left(\frac{1}{2}\right)^n \cos\left(\frac{\pi n}{4} - 45^\circ\right) u(n-1) \\ &= \tfrac{1}{2}\left(\frac{1}{2}\right)^{n-1} \cos\left(\frac{\pi n}{4} - \frac{\pi}{4}\right) u(n-1) \\ &= \tfrac{1}{2}\left(\frac{1}{2}\right)^{n-1} \cos\left\{\frac{\pi}{4}(n-1)\right\} u(n-1) \end{aligned}$$

The $z$-transform is

$$\begin{aligned} X(z) &= \frac{1}{2} z^{-1} \mathcal{Z}\left[\left(\frac{1}{2}\right)^n \cos\left(\frac{\pi}{4} n\right) u(n)\right] \\ &= \frac{1}{2} z^{-1} \left(\frac{1 - \frac{1}{2} z^{-1} \cos(\pi/4)}{1 - z^{-1}\cos(\pi/4) + \frac{1}{4} z^{-2}}\right) \end{aligned}$$

which after simplification becomes

$$X(z) = \frac{0.5 z^{-1} - \frac{1}{4\sqrt{2}} z^{-2}}{1 - \frac{1}{\sqrt{2}} z^{-1} + 0.25 z^{-2}}, \ |z| > 0.5$$

MATLAB verification:

```
% Sequence:
% x(n) = (1/2)^n*cos(pi*n/4-pi/4)*u(n-1)
%
% Analytical Expression of X(z)
%         0.5*z^(-1) - 1/(4*sq(2))*z^(-2)
% X(z) = --------------------------------
%        1 - 1/sq(2)*z^(-1) + 0.25*z^(-2)

% Matlab verification
b = [0, 0.5, -1/(4*sqrt(2))]; a = [1, -1/sqrt(2), 0.25];
delta = impseq(0,0,7); format long
xb1 = filter(b,a,delta)
xb1 =
  Columns 1 through 4
                  0   0.50000000000000   0.17677669529664   0.00000000000000
  Columns 5 through 8
  -0.04419417382416  -0.03125000000000  -0.01104854345604   0.00000000000000
%
% check
n = 0:7;
xb2 = ((1/2).^n).*cos(pi*n/4-pi/4).*stepseq(1,0,7)
xb2 =
  Columns 1 through 4
                  0   0.50000000000000   0.17677669529664   0.00000000000000
  Columns 5 through 8
  -0.04419417382416  -0.03125000000000  -0.01104854345604   0.00000000000000
```

```
%
error = abs(max(xb1-xb2)), format short;
error =
    6.938893903907228e-018
```

3. Problem **P4.3**

   (a) Part (b): The $z$-transform of $x(n)$ is $X(z) = \left(1+2z^{-1}\right), z \neq 0$. Consider

   $$x_2(n) = \left(1+n+n^2\right)x(n) = x(n) + n[x(n)] + n[nx(n)].$$

   Then

   $$\begin{aligned} X_2(z) &= X(z) + \left\{-z\frac{d}{dz}X(z)\right\} + \left[-z\frac{d}{dz}\left\{-z\frac{d}{dz}X(z)\right\}\right] \\ &= X(z) - z\frac{d}{dz}X(z) + z\frac{d}{dz}\left\{z\frac{d}{dz}X(z)\right\} \\ &= \left(1+2z^{-1}\right) - z\left\{-2z^{-2}\right\} + 2z^{-1} = 1+6z^{-1},\ z \neq 0. \end{aligned}$$

   (b) Part (c): The $z$-transform of a sequence $x(n)$ is $X(z) = \left(1+2z^{-1}\right), z \neq 0$. A new sequence is $x_3(n) = \left(\frac{1}{2}\right)^n x(n-2)$. Using the time-shift property

   $$\mathcal{Z}[x(n-2)] = z^{-2}X(z) = z^{-2} + 2z^{-3}$$

   with no change in ROC. Now using the frequency-shift property

   $$\begin{aligned} \mathcal{Z}\left[\left(\frac{1}{2}\right)^n x(n-2)\right] &= X_3(z) = \left[z^{-2} + 2z^{-3}\right]\Big|_{z \to z/0.5} \\ &= (2z)^{-2} + 2(2z)^{-3} = 0.25z^{-2} + 0.25z^{-3} \end{aligned}$$

   with ROC scaled by $(1/2)$. Since the old ROC is $|z| > 0$, the new ROC is also $|z| > 0$.

4. Problem **P4.5** (b)**:** The inverse $z$-transform of $X(z)$ is $x(n) = 2^{-n}u(n)$. Then

   $$\begin{aligned} \mathcal{Z}^{-1}\left[zX\left(z^{-1}\right)\right] &= x(-(n+1)) = x(-n-1) \\ &= 2^{-(-n-1)}u(-n-1) = 2^{n+1}u(-n-1) \end{aligned}$$

5. Problem **P4.9**

   (a) Part (b)**:** The $z$-transform $X(z)$ of a sequence is given as follows:

   $$X(z) = \frac{1 - z^{-1} - 4z^{-2} + 4z^{-3}}{1 - \frac{11}{4}z^{-1} + \frac{13}{8}z^{-2} - 0.25z^{-3}}, \text{ absolutely summable sequence}$$

   The partial fractions are computed using the `residuez` function.

```
b = [1,-1,-4,4]; a = [1,-11/4,13/8,-1/4];
[R,p,k] = residuez(b,a)
R =
    0.0000
  -10.0000
   27.0000
p =
    2.0000
    0.5000
    0.2500
k =
   -16
```

Therefore,

$$X(z) = -16 + \frac{0}{1-2z^{-1}} - \frac{10}{1-0.5z^{-1}} + \frac{27}{1-0.25z^{-1}}, \; 0.5 < |z| < 2$$

Hence from the $\mathcal{Z}$-transform table:

$$x(n) = -16\delta(n) - 10(0.5)^n u(n) + 27(0.25)^n u(n)$$

MATLAB verification:

```
[delta,n] = impseq(0,0,7);
xb1 = filter(b,a,delta)
xb1 =
  Columns 1 through 4
   1.00000000000000   1.75000000000000  -0.81250000000000  -0.82812500000000
  Columns 5 through 8
  -0.51953125000000  -0.28613281250000  -0.14965820312500  -0.07647705078125
xb2 = -16*delta - 10*(0.5).^n + 27*(0.25).^n
xb2 =
  Columns 1 through 4
   1.00000000000000   1.75000000000000  -0.81250000000000  -0.82812500000000
  Columns 5 through 8
  -0.51953125000000  -0.28613281250000  -0.14965820312500  -0.07647705078125
error = abs(max(xb1-xb2))
error =
     0
```

(b) Part (d)**:** The $z$-transform $X(z)$ of a sequence is given as follows:

$$X(z) = \frac{z}{z^3 + 2z^2 + 1.25z + 0.25}, \quad |z| > 1$$

The partial fractions are computed using the `residuez` function.

```
b = [0,0,1]; a = [1,2,1.25,0.25];
[R,p,k] = residuez(b,a), echo on;
R =
   4.0000
   0.0000 - 0.0000i
  -4.0000 + 0.0000i
p =
  -1.0000
  -0.5000 + 0.0000i
  -0.5000 - 0.0000i
k =
     []
```

Therefore,

$$X(z) = \frac{4}{1+z^{-1}} + \frac{0}{1+0.5z^{-1}} + \frac{-4}{(1+0.5z^{-1})^2}$$

Hence from the $\mathcal{Z}$-transform table:

$$\begin{aligned} x(n) &= 4(-1)^n u(n) + (-4)\left(-\tfrac{1}{0.5}\right)(n+1)(-0.5)^{n+1} u(n+1) \\ &= 4(-1)^n u(n) + 8(n+1)(-0.5)^{n+1} u(n) \end{aligned}$$

MATLAB verification:

```
[delta,n] = impseq(0,0,7);
xd1 = filter(b,a,delta)
xd1 =
         0         0    1.0000   -2.0000    2.7500   -3.2500    3.5625   -3.7500
xd2 = 4*(-1).^n + (8*(n+1)).*((-0.5).^(n+1))
xd2 =
         0         0    1.0000   -2.0000    2.7500   -3.2500    3.5625   -3.7500
error = abs(max(xd1-xd2))
error =
     0
```

6. Problem **P4.10:** The $z$-transform $X(z)$ is given as follows:

$$X(z) = \frac{2+3z^{-1}}{1-z^{-1}+0.81z^{-2}}, \quad |z| > 0.9$$

(a) Sequence $x(n)$ in a form that contains no complex numbers: Compare the denominator $1-z^{-1}+0.81z^{-2}$ with the denominator $1-2az^{-1}\cos\omega_0+a^2z^{-2}$ of the sin(cos) transform pairs (see page 174 of the text).

$$\begin{aligned} 1-z^{-1}+0.81z^{-2} &= 1-2az^{-1}\cos\omega_0+a^2z^{-2} \\ &\Rightarrow \quad a^2 = 0.81, \quad \cos\omega_0 = \frac{1}{2a} \end{aligned}$$

or

$$a = 0.9, \; \cos\omega_0 = \frac{1}{1.8} \Rightarrow \omega_0 = 0.3125\pi, \text{ and } \sin\omega_0 = 0.8315$$

Now $X(z)$ can be put in the form

$$\begin{aligned} X(z) &= \frac{2-z^{-1}+4z^{-1}}{1-2(0.9)z^{-1}\frac{1}{1.8}+(0.9)^2z^{-2}}, \quad |z|>0.9 \\ &= \frac{2\left(1-0.9z^{-1}\frac{1}{1.8}\right)}{1-2(0.9)z^{-1}\frac{1}{1.8}+(0.9)^2z^{-2}} + 5.3452\frac{0.9(0.8315)z^{-1}}{1-2(0.9)z^{-1}\frac{1}{1.8}+(0.9)^2z^{-2}} \end{aligned}$$

Finally, after table lookup

$$x(n) = \left[2(0.9)^n\cos(0.3125\pi) + (5.3452)(0.9)^n\sin(0.3125\pi)\right]u(n)$$

For another approach using PFE and residues, see the MATLAB script below.

```
clear all
% (a) Use of the residue and transform table
b=[2, 3]; a=[1,-1,0.81];
[R,p,k] = residuez(b,a);
R_real = (real(R))
R_real =
     1
     1
R_imag = (imag(R))
R_imag =
   -2.6726
    2.6726
p_magn = (abs(p))
p_magn =
    0.9000
    0.9000
p_angl = (angle(p))/pi
```

```
p_angl =
    0.3125
   -0.3125
[delta,n] = impseq(0,0,19);
xa = (p_magn(1).^n).*(2*R_real(1)*cos(p_angl(1)*pi*n)-2*R_imag(1)*sin(p_angl(1)*pi*
%
% Print Response
fprintf(1,'\n Hence the sequence x(n) is \n')

 Hence the sequence x(n) is
fprintf(1,'\n\tx(n) = (%1.1f)^n * (%1.0f*cos(%1.4f*pi*n) - (%2.4f)*sin(%1.4f*pi*n)\
    p_magn(1),2*R_real(1),p_angl(1),2*R_imag(1),p_angl(1));

x(n) = (0.9)^n * (2*cos(0.3125*pi*n) - (-5.3452)*sin(0.3125*pi*n)
```

(b) MATLAB verification:

```
xb = filter(b,a,delta);
error = abs(max(xa-xb))
error =
  1.5543e-015
```

7. Problem **P4.11**

(a) Part (a): The impulse response is $h(n) = 2(0.5)^n u(n)$. The system function representation is

$$H(z) = Z[h(n)] = \frac{2}{1-0.5z^{-1}},\ |z| > 0.5.$$

The difference equation representation is

$$\frac{Y(z)}{X(z)} = \frac{2}{1-0.5z^{-1}} \Rightarrow Y(z) - 0.5z^{-1}Y(z) = 2X(z)$$

or

$$y(n) = 2x(n) + 0.5y(n-1).$$

The pole-zero description is given by a zero at $z = 0$ and a pole at $z = 0.5$. Finally, to compute output $y(n)$ when $x(n) = (1/4)^n u(n)$ we use the $z$-transform approach (since the ROCs overlap):

$$\begin{aligned} Y(z) &= H(z)X(z) = \frac{2}{1-0.5z^{-1}} \times \frac{1}{1-0.25z^{-1}} \\ &= \frac{4}{1-0.5z^{-1}} - \frac{2}{1-0.25z^{-1}},\ |z| > 0.5 \end{aligned}$$

Hence

$$y(n) = 4(0.5)^n u(n) - 2(0.25)^n u(n).$$

(b) Part (d): Impulse response of an LTI system:

$$h(n) = n[u(n) - u(n-10)] = \left[\underset{\uparrow}{0}, 1, 2, \ldots, 9\right]$$

**i)** System function representation:

$$H(z) = z^{-1} + 2z^{-2} + \ldots + 9z^{-9} = \sum_{k=1}^{9} kz^{-k}$$

**ii)** Difference equation representation:

$$y(n) = \sum_{k=1}^{9} kx(n-k).$$

**iii)** Pole-zero plot – MATLAB script:

```
clear, close all;
hb = [0:9]; ha = [1,0]; zplane(hb,ha);
```

The pole-zero plot is shown in Figure 4.1.

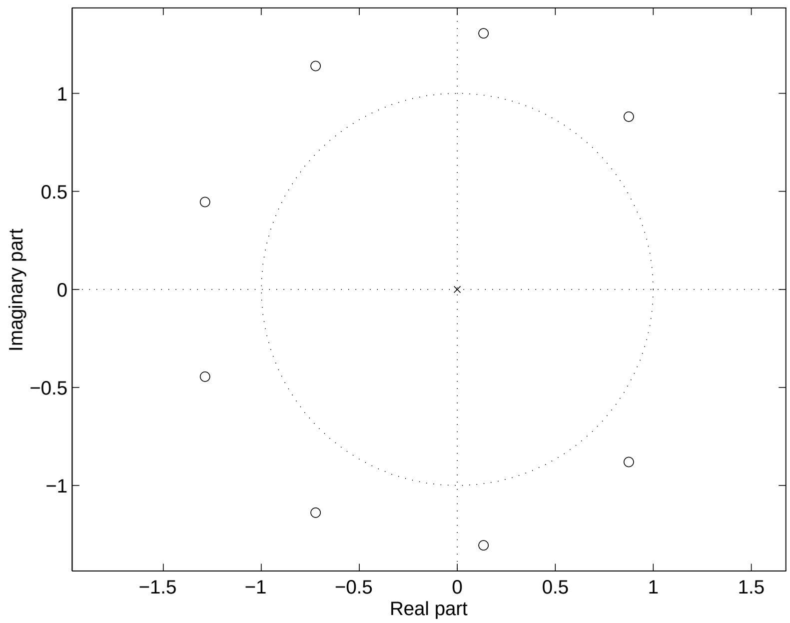


Figure 4.1: Pole-zero plot in Problem P4.11d.

**iv)** The output $y(n)$ when the input is $x(n) = (1/4)^n u(n)$:

$$\begin{aligned} Y(z) &= H(z)X(z) = \left(z^{-1} + 2z^{-2} + \ldots + 9z^{-9}\right) \frac{1}{1-0.25z^{-1}}, \ |z| > 0.25 \\ &= \frac{z^{-1} + 2z^{-2} + \ldots + 9z^{-9}}{1-0.25z^{-1}} \\ &= \frac{3029220}{1-0.25z^{-1}} - 3029220 - 757304z^{-1} - 189324z^{-2} - 47328z^{-3} - 11828z^{-4} \\ &\quad -2952z^{-5} - 732z^{-6} - 176z^{-7} - 36z^{-8} \end{aligned}$$

where the PFE was performed using MATLAB.

```
clear, close all;
hb = [0:9]; ha = [1,0]; xb = [1]; xa = [1,-0.25];
yb = hb; ya = xa;
[R,p,k] = residuez(yb,ya)
R =
     3029220
p =
    0.2500
k =
```

```
Columns 1 through 6
  -3029220     -757304     -189324      -47328      -11828       -2952
Columns 7 through 9
      -732        -176         -36
```

Hence

$$\begin{aligned} y(n) &= 3029220\,(0.25)^n\,u(n) - 3029220\delta(n) - 757304\delta(n-1) - \cdots \\ &\quad -176\delta(n-7) - 36\delta(n-8). \end{aligned}$$

8. Problem **P4.15:** A stable system has the following pole-zero locations:

$$z_1 = j, \quad z_2 = -j, \quad p_1 = -\frac{1}{2} + j\frac{1}{2}, \quad p_2 = -\frac{1}{2} - j\frac{1}{2}$$

It ia also known that the frequency response function $H\left(e^{j\omega}\right)$ evaluated at $\omega = 0$ is equal to 0.8, i.e.,

$$H\left(e^{j0}\right) = 0.8$$

**a)** System function $H(z)$ and region of convergence.

$$H(z) = K\frac{(z-j)(z+j)}{(z+0.5-j0.5)(z+0.5+j0.5)} = K\frac{z^2+1}{z^2+z+0.5}, |z| > \frac{1}{\sqrt{2}}$$

Now $@z = e^{j0} = 1$, $H(1) = 0.8$ is given, hence

$$0.8 = K\frac{1+1}{1+1+0.5} = K\frac{2}{2.5} \Rightarrow K = 1$$

or

$$\boxed{H(z) = \frac{z^2+1}{z^2+z+0.5}, |z| > \frac{1}{\sqrt{2}}}$$

**b)** Difference equation representation.:

$$H(z) = \frac{z^2+1}{z^2+z+0.5} = \frac{1+z^{-2}}{1+z^{-1}+0.5z^{-2}} = \frac{Y(z)}{X(z)}$$

After cross multiplying and inverse transforming

$$\boxed{y(n) + y(n-1) + 0.5y(n-2) = x(n) + x(n-1)}$$

**c)** Steady-state response $y_{ss}(n)$ if the input is

$$x(n) = \frac{1}{\sqrt{2}}\sin\left(\frac{\pi n}{2}\right)u(n).$$

From the $z$-transform Table

$$X(z) = \left(\frac{1}{\sqrt{2}}\right)\frac{z}{z^2+1}, \ |z| > 1$$

Hence

$$\begin{aligned} Y(z) = H(z)\,X(z) &= \frac{z^2+1}{z^2+z+0.5}\left(\frac{1}{\sqrt{2}}\right)\frac{z}{z^2+1} \\ &= \left(\frac{1}{\sqrt{2}}\right)\frac{z}{z^2+z+0.5}, \ |z| > \tfrac{1}{\sqrt{2}} \end{aligned}$$

Thus the poles of $Y(z)$ are the poles of $H(z)$ which are inside the unit circle. Therefore, there is **NO** steady-state response or $y_{ss}(n) = 0$.

**d)** Transient response $y_{tr}(n)$: Since $y_{ss}(n) = 0$, the total response $y(n) = y_{tr}(n)$. From the $Y(z)$ expression from part c) above

$$Y(z) = \left(\frac{1}{\sqrt{2}}\right)\frac{z}{z^2+z+0.5} = \sqrt{2}\frac{\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}z}{1+z^{-1}+\left(\frac{1}{\sqrt{2}}\right)^2 z^{-2}}, \ |z| > \frac{1}{\sqrt{2}}$$

Hence from table lookup we have

$$y(n) = y_{tr}(n) = \sqrt{2}\left(\frac{1}{\sqrt{2}}\right)^n \sin(0.75\pi n)\, u(n)$$

9. Problem **P4.16:** Digital filter is described by the difference equation:

$$y(n) = x(n) + x(n-1) + 0.9y(n-1) - 0.81y(n-2)$$

**a)** Magnitude and phase of the frequency response: MATLAB script

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P4.16a');
% (a) Magnitude and Phase Plots
b = [1, 1]; a = [1,-0.9,0.81];
w = [0:1:500]*pi/500; H = freqz(b,a,w);
magH = abs(H); phaH = angle(H)*180/pi;
subplot(2,1,1); plot(w/pi,magH); axis([0,1,0,12]); grid
title('Magnitude Response'); xlabel('frequency in pi units'); ylabel('|H|');
subplot(2,1,2); plot(w/pi,phaH); axis([0,1,-180,180]); grid
title('Phase Response'); xlabel('frequency in pi units'); ylabel('Degrees');
%
w = [pi/3,pi]; H = freqz(b,a,w); magH = abs(H); phaH = angle(H)*180/pi;
fprintf(1,'\n At w = pi/3 the magnitude is %1.4f and the phase is %3.4f degrees \n',...
        magH(1), phaH(1));

 At w = pi/3 the magnitude is 10.5215 and the phase is -58.2595 degrees
fprintf(1,'\n At w =  pi  the magnitude is %1.4f and the phase is %3.4f degrees \n',...
        magH(2), phaH(2));

 At w =  pi  the magnitude is 0.0000 and the phase is -90.0000 degrees
```

The frequency response plots are shown in Figure 4.2.

**b)** Steady-state response: MATLAB script

```
n = [0:200]; x = sin(pi*n/3) + 5*cos(pi*n);
y = filter(b,a,x);
n = n(101:201); x = x(101:201); y = y(101:201);  % Steady-state section
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P4.16b');
subplot(2,1,1); plot(n,x); title('Input sequence'); axis([100,200,-6,6])
xlabel('n'); ylabel('x(n)');
subplot(2,1,2); plot(n,y); title('Output sequence'); axis([100,200,-12,12])
xlabel('n'); ylabel('y(n)');
```

The steady-state response is shown in Figure 4.3. It shows that the $\omega = \pi$ frquency is suppressed while the only component in the output is due to the $\omega = \pi/3$ frequency which is amplified by about 10.

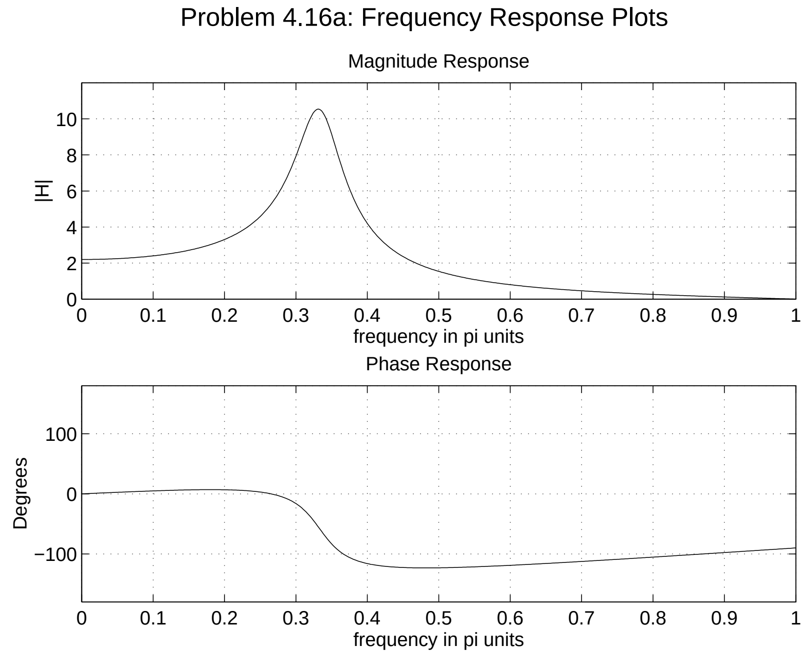


Figure 4.2: Frequency Response Plots in Problem P4.16a

10. Problem **P4.17:** Difference equation solution using one-sided $z$-transform approach:

$$\begin{aligned}&y(n) = 0.5y(n-1) + 0.25y(n-2) + x(n),\ n \geq 0\\ &x(n) = (0.8)^n u(n)\\ &y(-1) = 1,\ y(-2) = 2\end{aligned}$$

Taking one-sided $z$-transform:

$$Y^+(z) = 0.5\left[z^{-1}Y^+(z) + y(-1)\right] + 0.25\left[z^{-2}Y^+(z) + y(-2) + z^{-1}y(-1)\right] + X^+(z)$$

or

$$\begin{aligned}Y^+(z)\left(1 - 0.5z^{-1} - 0.25z^{-2}\right) &= 0.5 + 0.5 + 0.25z^{-1} + \frac{1}{1-0.8z^{-1}}\\ &= \underbrace{1 + 0.25z^{-1}}_{\text{Equi. I.C. Input}} + \frac{1}{1-0.8z^{-1}} = \frac{1 - 0.55z^{-1} - 0.2z^{-2} + 1}{1-0.8z^{-1}}\end{aligned}$$

or

$$\begin{aligned}Y^+(z) &= \frac{2 - 0.55z^{-1} - 0.2z^{-2}}{(1-0.8z^{-1})(1 - 0.5z^{-1} - 0.25z^{-2})}\\ &= \frac{1 - 0.55z^{-1} - 0.2z^{-2}}{1 - 1.3z^{-1} + 0.15z^{-2} + 0.2z^{-3}}\\ &= \frac{65.8702}{1 - 0.8090z^{-1}} + \frac{-64}{1-0.8z^{-1}} + \frac{0.1298}{1 + 0.3090z^{-1}}\end{aligned}$$

Hence

$$y(n) = 65.8702\,(0.8090)^n - 64\,(0.8)^n + 0.1298\,(-0.3090)^n,\ n \geq 0$$

MATLAB verification:

```
b = [1]; a = [1, -0.5, -0.25];        % Difference equation
yic = [1,2];                          % Initial conditions
```

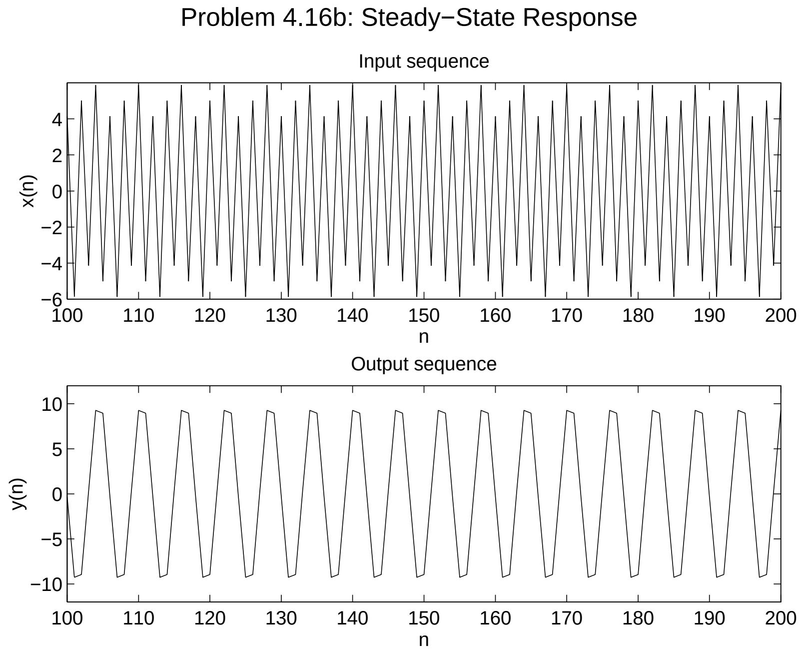


Figure 4.3: Steady-state Response in problem P4.16b

```
n = [0:20]; x = (0.8).^n;               % Input sequence

% Numerical Solution
V = filtic(b,a,yic), echo on;           % equivalent initial condition input
V =
    1.0000    0.2500
y1 = filter(b,a,x,V);                   % Output sequence

% Analytical solution
b1 = conv(V,[1,-0.8])+[1,0,0]           % Num of Y(z)
b1 =
    2.0000   -0.5500   -0.2000
a1 = conv([1,-0.8],[1,-0.5,-0.25])      % Denom of Y(z)
a1 =
    1.0000   -1.3000    0.1500    0.2000
[R,p,k] = residuez(b1,a1), echo on;     % PFE
R =
   65.8702
  -64.0000
    0.1298
p =
    0.8090
    0.8000
   -0.3090
k =
     []
y2 = zeros(1,21); L = length(R);        %
for l = 1:L                             % Assemble
```

```
    y2 = y2 + R(l)*(p(l).^n);          %          Output
end                                    %               Sequence
%
error = abs(max(y1-y2))                % Difference
error =
  1.1169e-013
```

11. Problem **P4.19:** A causal, linear, and time-invariant system is given by

$$y(n) = y(n-1) + y(n-2) + x(n-1)$$

**a)** System Function: Taking $z$-transform of both sides, we obtain

$$Y(z) = z^{-1}Y(z) + z^{-2}Y(z) + z^{-1}X(z) \Rightarrow H(z) \stackrel{\text{def}}{=} \frac{Y(z)}{X(z)} = \frac{z^{-1}}{1 - z^{-1} - z^{-2}}, |z| > r_0$$

where $r_0$ is the magnitude of the largest pole since the system is causal.

**b)** Pole-zero pot and the ROC:

$$H(z) = \frac{z}{z^2 - z - 1} = \frac{z}{\left(z - \frac{1+\sqrt{5}}{2}\right)\left(z - \frac{1-\sqrt{5}}{2}\right)}$$

The zero is @ $z = 0$ and poles are @ $z = \frac{1 \pm \sqrt{5}}{2}$. Hence the ROC is $|z| > \frac{1+\sqrt{5}}{2} = 1.618$.

**c)** Impulse response: Using PFE

$$H(z) = \frac{z}{\left(z - \frac{1+\sqrt{5}}{2}\right)\left(z - \frac{1-\sqrt{5}}{2}\right)} = \frac{1}{\sqrt{5}} \frac{z}{\left(z - \frac{1+\sqrt{5}}{2}\right)} - \frac{1}{\sqrt{5}} \frac{z}{\left(z - \frac{1-\sqrt{5}}{2}\right)}, |z| > \frac{1+\sqrt{5}}{2}$$

Hence

$$h(n) = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n u(n) - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n u(n)$$

**d)** Clearly the system is not stable since $h(n) \nearrow \infty$ as $n \nearrow \infty$. For a stable unit sample response the ROC should be

$$\frac{1-\sqrt{5}}{2} < |z| < \frac{1+\sqrt{5}}{2}$$

Then

$$h(n) = -\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n u(-n-1) - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n u(n)$$

12. Problem **P4.20**: The difference eqution is

$$y(n) = \frac{1}{4}y(n-1) + x(n) + 3x(n-1), n \geq 0; \quad y(-1) = 2$$

with the input $x(n) = e^{j\pi n/4}u(n)$. Taking one-sided $z$-transform of the difference equation, we obtain,

$$Y^+(z) = \frac{1}{4}\left[z^{-1}Y^+(z) + y(-1)\right] + \frac{1}{1 - e^{j\pi/4}z^{-1}} + 3\frac{z^{-1}}{1 - e^{j\pi/4}z^{-1}}.$$

Substituting the initial condition and rearranging, we obtain

$$Y^+(z)\left[1 - \frac{1}{4}z^{-1}\right] = \frac{1}{2} + \frac{1 + 3z^{-1}}{1 - e^{j\pi/4}z^{-1}}. \tag{4.2}$$

The second term on the right-hand side provides the zero-state response of the system. Thus

$$\begin{aligned} Y_{zs}^{+}(z) &= \frac{1+3z^{-1}}{\left(1-\frac{1}{4}z^{-1}\right)\left(1-e^{j\pi/4}z^{-1}\right)} \\ &= \frac{4.4822e^{-j0.8084}}{1-\frac{1}{4}z^{-1}} + \frac{3.8599e^{j2.1447}}{1-e^{j\pi/4}z^{-1}} \end{aligned}$$

Hence the zero-state response is

$$y_{zs}(n) = \left[4.4822e^{-j0.8084}\left(\frac{1}{4}\right)^{n} + 3.8599e^{j2.1447}e^{j\pi n/4}\right]u(n).$$

The steady-state part of the total response is due to simple poles on the unit circle. The pole on the unit circle is at $z = e^{j\pi/4}$ which is due to the input sequence. From (4.2), the total response is

$$\begin{aligned} Y^{+}(z) &= \left(\frac{1}{1-\frac{1}{4}z^{-1}}\right)\left(\frac{1}{2} + \frac{1+3z^{-1}}{1-e^{j\pi/4}z^{-1}}\right) \\ &= \frac{3/2 + (2.6464 - j0.3536)z^{-1}}{\left(1-\frac{1}{4}z^{-1}\right)\left(1-e^{j\pi/4}z^{-1}\right)} \\ &= \frac{4.4822e^{-j0.8084}}{1-\frac{1}{4}z^{-1}} + \frac{3.6129e^{j2.0282}}{1-e^{j\pi/4}z^{-1}}. \end{aligned}$$

The steady-state response is given by the second-term on the right-hand side. Thus

$$y_{ss}(n) = 3.6129e^{j2.0282}\,e^{j\pi n/4}u(n) = 3.6129e^{j(\pi n/4 - 2.0282)}u(n).$$

# Chapter 5

# The Discrete Fourier Transform

1. Problem **P 5.1**

   (a) Periodic sequence: $\tilde{x}_1(n) = \left\{\ldots, \underset{\uparrow}{2}, 0, 2, 0, 2, 0, 2, 0, 2, 0, \ldots\right\}$. Now,

   $$\tilde{X}_1(k) = \sum_{n=0}^{N-1} \tilde{x}_1(n) W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

   Hence,

   $$\begin{aligned}
   \tilde{X}_1(0) &= 2(1) + 0(1) + 2(1) + 0(1) &&= 4 \\
   \tilde{X}_1(1) &= 2(1) + 0(-j) + 2(-1) + 0(j) &&= 0 \\
   \tilde{X}_1(2) &= 2(1) + 0(-1) + 2(1) + 0(-1) &&= 4 \\
   \tilde{X}_1(3) &= 2(1) + 0(j) + 2(-1) + 0(-j) &&= 0
   \end{aligned}$$

   MATLAB verification:

   ```
   xtilde1 = [2,0,2,0]; N = length(xtilde1);
   [Xtilde1] = dft(xtilde1,N)
   Xtilde1 =
      4.0000                  0 - 0.0000i   4.0000 + 0.0000i        0 - 0.0000i
   ```

   (b) Periodic sequence: $\tilde{x}_2(n) = \left\{\ldots, \underset{\uparrow}{0}, 0, 1, 0, 0, 0, 0, 1, 0, 0, \ldots\right\}$. Now,

   $$\tilde{X}_2(k) = \sum_{n=0}^{N-1} \tilde{x}_2(n) W_N^{nk};\ N = 5;\ W_5 = e^{-j2\pi/5} = 0.3090 - j0.9511$$

   Hence,

   $$\begin{aligned}
   \tilde{X}_2(0) &= 0 + 0 + 1(1) + 0 + 0 &&= 1 \\
   \tilde{X}_2(1) &= 0 + 0 + 1(0.3090 - j0.9511)^2 + 0 + 0 &&= -0.8090 - j0.5878 \\
   \tilde{X}_2(2) &= 0 + 0 + 1(0.3090 - j0.9511)^4 + 0 + 0 &&= 0.3090 + j0.9511 \\
   \tilde{X}_2(3) &= 0 + 0 + 1(0.3090 - j0.9511)^6 + 0 + 0 &&= 0.3090 - j0.9511 \\
   \tilde{X}_2(4) &= 0 + 0 + 1(0.3090 - j0.9511)^8 + 0 + 0 &&= -0.8090 + j0.5878
   \end{aligned}$$

   MATLAB verification:

   ```
   xtilde2 = [0,0,1,0,0]; N = length(xtilde2);
   [Xtilde2] = dft(xtilde2,N)
   Xtilde2 =
     Columns 1 through 4
      1.0000            -0.8090 - 0.5878i   0.3090 + 0.9511i   0.3090 - 0.9511i
     Column 5
     -0.8090 + 0.5878i
   ```

(c) Periodic sequence: $\tilde{x}_3(n) = \left\{\ldots, \underset{\uparrow}{3}, -3, 3, -3, 3, -3, 3, -3, \ldots\right\}$. Now,

$$\tilde{X}_3(k) = \sum_{n=0}^{N-1} \tilde{x}_3(n) W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

Hence,

$$\begin{aligned}
\tilde{X}_3(0) &= 3(1) - 3(1) + 3(1) - 3(1) &= 0\\
\tilde{X}_3(1) &= 3(1) - 3(-j) + 3(-1) - 3(j) &= 0\\
\tilde{X}_3(2) &= 3(1) - 3(-1) + 3(1) - 3(-1) &= 12\\
\tilde{X}_3(3) &= 3(1) - 3(j) + 3(-1) - 3(-j) &= 0
\end{aligned}$$

MATLAB verification:

```
xtilde = [3,-3,3,-3]; N = length(xtilde);
[Xtilde] = dft(xtilde,N)
Xtilde =
        0             0.0000 - 0.0000i  12.0000 + 0.0000i   0.0000 - 0.0000i
```

(d) Periodic sequence: $\tilde{x}_4(n) = \left\{\ldots, \underset{\uparrow}{j}, j, -j, -j, j, j, -j, -j, \ldots\right\}$. Now,

$$\tilde{X}_4(k) = \sum_{n=0}^{N-1} \tilde{x}_4(n) W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

Hence,

$$\begin{aligned}
\tilde{X}_4(0) &= j(1) + j(1) - j(1) - j(1) &= 0\\
\tilde{X}_4(1) &= j(1) + j(-j) - j(-1) - j(j) &= 2 + j2\\
\tilde{X}_4(2) &= j(1) + j(-1) - j(1) - j(-1) &= 0\\
\tilde{X}_4(3) &= j(1) + j(j) - j(-1) - j(-j) &= -2 + j2
\end{aligned}$$

MATLAB verification:

```
xtilde = [j,j,-j,-j]; N = length(xtilde4);
[Xtilde4] = dft(xtilde4,N)
Xtilde4 =
        0             2.0000 + 2.0000i        0             -2.0000 + 2.0000i
```

(e) Periodic sequence: $\tilde{x}_5(n) = \left\{\ldots, \underset{\uparrow}{1}, j, j, 1, 1, j, j, 1, j, j, \ldots\right\}$. Now,

$$\tilde{X}_5(k) = \sum_{n=0}^{N-1} \tilde{x}_5(n) W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

Hence,

$$\begin{aligned}
\tilde{X}_5(0) &= 1(1) + j(1) + j(1) + 1(1) &= 2 + 2j\\
\tilde{X}_5(1) &= 1(1) + j(-j) + j(-1) + 1(j) &= 2\\
\tilde{X}_5(2) &= 1(1) + j(-1) + j(1) + 1(-1) &= 0\\
\tilde{X}_5(3) &= 1(1) + j(j) + j(-1) + 1(-j) &= -2j
\end{aligned}$$

MATLAB verification:

```
xtilde5 = [1,j,j,1]; N5 = length(xtilde5);
[Xtilde5] = dft(xtilde5,N5)
Xtilde5 =
   2.0000 + 2.0000i   2.0000 + 0.0000i   0.0000 - 0.0000i   0.0000 - 2.0000i
```

2. Problem **P 5.2**

(a) Periodic DFS sequence: $\tilde{X}_1(k) = \{5, -2j, 3, 2j\}$, $N = 4$. Now,

$$\tilde{x}_1(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_1(k)\,W_N^{-nk};\ N = 4;\ W_4^{-1} = e^{j2\pi/4} = j$$

Hence,

$$\begin{aligned}
\tilde{x}_1(0) &= [5(1) - 2j(1) + 3(1) + 2j(1)]/4 &&= 2\\
\tilde{x}_1(1) &= [5(1) - 2j(j) + 3(-1) + 2j(-j)]/4 &&= 1.5\\
\tilde{x}_1(2) &= [5(1) - 2j(-1) + 3(1) + 2j(-1)]/4 &&= 2\\
\tilde{x}_1(3) &= [5(1) - 2j(-j) + 3(-1) + 2j(j)]/4 &&= -0.5
\end{aligned}$$

MATLAB verification:

```
Xtilde1 = [5,-2*j,3,2*j]; N1 = length(Xtilde1);
[xtilde1] = real(idfs(Xtilde1,N1))
xtilde1 =
    2.0000    1.5000    2.0000   -0.5000
```

(b) Periodic DFS sequence: $\tilde{X}_2(k) = \{4, -5, 3, -5\}$. Now,

$$\tilde{x}_2(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_2(k)\,W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

Hence,

$$\begin{aligned}
\tilde{x}_2(0) &= [4(1) - 5(1) + 3(1) - 5(1)]/4 &&= -0.75\\
\tilde{x}_2(1) &= [4(1) - 5(-j) + 3(-1) - 5(j)]/4 &&= 0.25\\
\tilde{x}_2(2) &= [4(1) - 5(-1) + 3(1) - 5(-1)]/4 &&= 4.25\\
\tilde{x}_2(3) &= [4(1) - 5(j) + 3(-1) - 5(-j)]/4 &&= 0.25
\end{aligned}$$

MATLAB verification:

```
Xtilde2 = [4,-5,3,-5]; N = length(Xtilde2);
[xtilde2] = real(idfs(Xtilde2,N))
xtilde2 =
   -0.7500    0.2500    4.2500    0.2500
```

(c) Periodic DFS sequence: $\tilde{X}_3(k) = \{1, 2, 3, 4, 5\}$. Now,

$$\tilde{x}_3(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_3(k)\,W_N^{nk};\ N = 5;\ W_5 = e^{-j2\pi/5} = 0.3090 - j0.9511$$

Hence,

$$\begin{aligned}
\tilde{x}_3(0) &= \left[1W_5^0 + 2W_5^0 + 3W_5^0 + 4W_5^0 + 5W_5^0\right]/5 &&= 3\\
\tilde{x}_3(1) &= \left[1W_5^0 + 2W_5^1 + 3W_5^2 + 4W_5^3 + 5W_5^4\right]/5 &&= -0.5000 - j0.6882\\
\tilde{x}_3(2) &= \left[1W_5^0 + 2W_5^2 + 3W_5^4 + 4W_5^6 + 5W_5^8\right]/5 &&= -0.5000 - j0.1625\\
\tilde{x}_3(3) &= \left[1W_5^0 + 2W_5^3 + 3W_5^6 + 4W_5^9 + 5W_5^{12}\right]/5 &&= -0.5000 + j0.1625\\
\tilde{x}_3(4) &= \left[1W_5^0 + 2W_5^4 + 3W_5^8 + 4W_5^{12} + 5W_5^{16}\right]/5 &&= -0.5000 + j0.6882
\end{aligned}$$

MATLAB verification:

```
Xtilde3 = [1,2,3,4,5]; N = length(Xtilde3);
[xtilde3] = idfs(Xtilde3,N)
xtilde3 =
  Columns 1 through 4
   3.0000            -0.5000 - 0.6882i  -0.5000 - 0.1625i  -0.5000 + 0.1625i
  Column 5
  -0.5000 + 0.6882i
```

(e) Periodic DFS sequence: $\tilde{X}_5(k) = \{0, j, -2j, -j\}$, $N = 4$. Now,

$$\tilde{x}_5(n) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}_5(k) W_N^{-nk};\ N = 4;\ W_4^{-1} = e^{j2\pi/4} = j$$

Hence,

$$\begin{aligned}
\tilde{x}_5(0) &= [0(1) + j(1) - 2j(1) - j(1)]/4 &&= -j0.5 \\
\tilde{x}_5(1) &= [0(1) + j(j) - 2j(-1) - j(-j)]/4 &&= -0.5 + j0.5 \\
\tilde{x}_5(2) &= [0(1) + j(-1) - 2j(1) - 2j(-1)]/4 &&= -j0.25 \\
\tilde{x}_5(3) &= [0(1) + j(-j) - 2j(-1) - j(j)]/4 &&= 0.5 + j0.5
\end{aligned}$$

MATLAB verification:

```
Xtilde5 = [0,j,-2*j,-j]; N5 = length(Xtilde5);
[xtilde5] = real(idfs(Xtilde5,N1))
xtilde1 =
     0 - 0.5000i  -0.5000 + 0.5000i   0.0000 - 0.2500i   0.5000 + 0.5000i
```

(d) Periodic DFS sequence: $\tilde{X}(k) = \{0, 0, 2, 0\}$. Now,

$$\tilde{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}(k) W_N^{nk};\ N = 4;\ W_4 = e^{-j2\pi/4} = -j$$

Hence,

$$\begin{aligned}
\tilde{x}(0) &= [0(1) + 0(1) + 2(1) + 0(1)]/4 &&= 0.5 \\
\tilde{x}(1) &= [0(1) + 0(-j) + 2(-1) + 0(j)]/4 &&= -0.5 \\
\tilde{x}(2) &= [0(1) + 0(-1) + 2(1) + 0(-1)]/4 &&= 0.5 \\
\tilde{x}(3) &= [0(1) + 0(j) + 2(-1) + 0(-j)]/4 &&= -0.5
\end{aligned}$$

MATLAB verification:

```
Xtilde4 = [00,0,2,0]; N = length(Xtilde4);
[xtilde4] = real(idfs(Xtilde4,N))
xtilde4 =
    0.5000   -0.5000    0.5000   -0.5000
```

3. Problem **P 5.3**

Periodic $\tilde{x}_1(n)$ with fundamental period $N = 50$

$$\tilde{x}_1(n) = \left\{ \begin{array}{cc} ne^{-0.3n}, & 0 \le n \le 25 \\ 0, & 26 \le n \le 49 \end{array} \right|_{\text{PERIODIC}}$$

Periodic $\tilde{x}_2(n)$ with fundamental period $N = 100$

$$\tilde{x}_2(n) = \left\{ \begin{array}{cc} ne^{-0.3n}, & 0 \le n \le 25 \\ 0, & 26 \le n \le 99 \end{array} \right\}_{\text{PERIODIC}}$$

(a) Computation of $\tilde{X}_1(k)$ using MATLAB:

```
clear; close all;
% (a) DFS Xtilde1(k)
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.3a');
n1 = [0:49]; xtilde1 = [n1(1:26).*exp(-0.3*n1(1:26)),zeros(1,24)]; N1 = length(n1);
[Xtilde1] = dft(xtilde1,N1); k1 = n1;
mag_Xtilde1 = abs(Xtilde1); pha_Xtilde1 = angle(Xtilde1)*180/pi;
zei = find(mag_Xtilde1 < 1000*eps);
pha_Xtilde1(zei) = zeros(1,length(zei));
```

```
subplot(3,1,1); stem(n1,xtilde1); axis([-1,N1,min(xtilde1),max(xtilde1)]);
title('One period of the periodic sequence xtilde1(n)'); ylabel('xtilde1');
ntick = [n1(1):2:n1(N1),N1]';
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',10)
subplot(3,1,2); stem(k1,mag_Xtilde1); axis([-1,N1,min(mag_Xtilde1),max(mag_Xtilde1)
title('Magnitude of Xtilde1(k)'); ylabel('|Xtilde1|')
ktick = [k1(1):2:k1(N1),N1]';
set(gca,'XTickMode','manual','XTick',ktick,'FontSize',10)
subplot(3,1,3); stem(k1,pha_Xtilde1); axis([-1,N1,-180,180]);
title('Phase of Xtilde1(k)'); xlabel('k'); ylabel('Angle in Deg')
ktick = [k1(1):2:k1(N1),N1]';
set(gca,'XTickMode','manual','XTick',ktick,'FontSize',10)
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180])
```

Plots of $\tilde{x}_1(n)$ and $\tilde{X}_1(k)$ are shown in Figure 5.1.

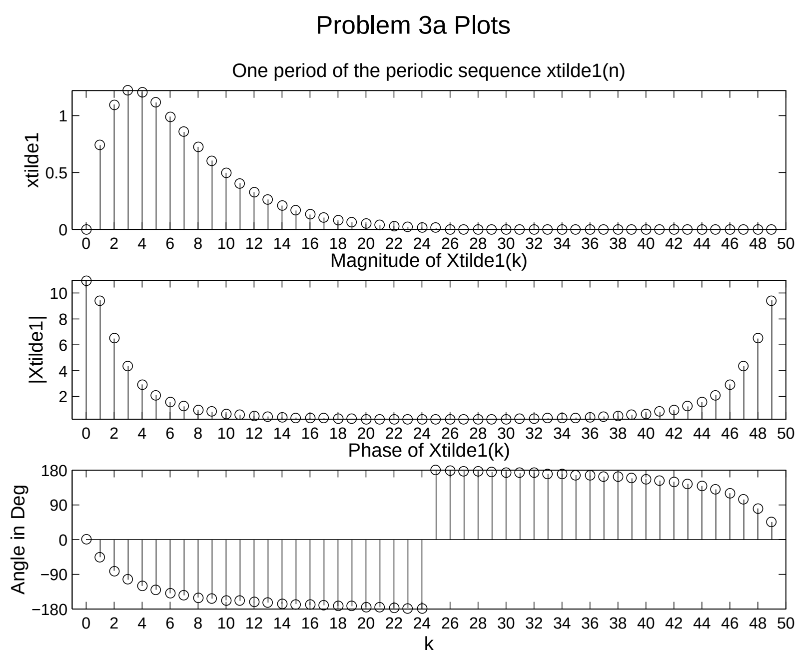


Figure 5.1: Plots of $\tilde{x}_1(n)$ and $\tilde{X}_1(k)$ in Problem 5.3a

(b) Computation of $\tilde{X}_2(k)$ using MATLAB:

```
% (b) DFS Xtilde2(k)
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P5.3b');
n2 = [0:99]; xtilde2 = [xtilde1, zeros(1,50)]; N2 = length(n2);
[Xtilde2] = dft(xtilde2,N2); k2 = n2;
mag_Xtilde2 = abs(Xtilde2); pha_Xtilde2 = angle(Xtilde2)*180/pi;
zei = find(mag_Xtilde2 < 1000*eps);
pha_Xtilde2(zei) = zeros(1,length(zei));
subplot(3,1,1); stem(n2,xtilde2); axis([-1,N2,min(xtilde2),max(xtilde2)]);
title('One period of the periodic sequence xtilde2(n)'); ylabel('xtilde2');
ntick = [n2(1):5:n2(N2),N2]';
set(gca,'XTickMode','manual','XTick',ntick)
```

```
subplot(3,1,2); stem(k2,mag_Xtilde2); axis([-1,N2,min(mag_Xtilde2),max(mag_Xtilde2)]);
title('Magnitude of Xtilde2(k)'); ylabel('|Xtilde2|')
ktick = [k2(1):5:k2(N2),N2]';
set(gca,'XTickMode','manual','XTick',ktick)
subplot(3,1,3); stem(k2,pha_Xtilde2); axis([-1,N2,-180,180]);
title('Phase of Xtilde2(k)'); xlabel('k'); ylabel('Degrees')
ktick = [k2(1):5:k2(N2),N2]';
set(gca,'XTickMode','manual','XTick',ktick)
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180])
```

Plots of $\tilde{x}_2(n)$ and $\tilde{X}_2(k)$ are shown in Figure 5.2.

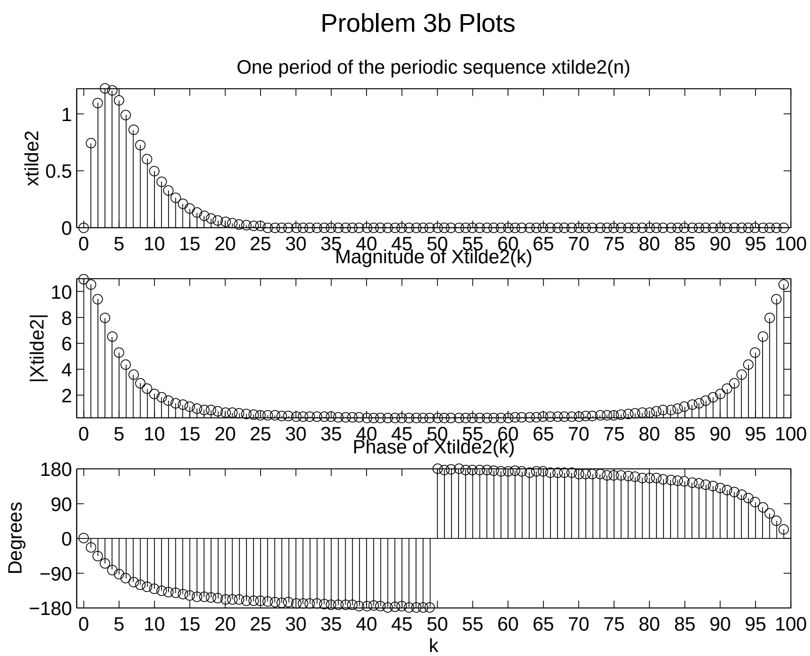


Figure 5.2: Plots of Magnitude and Phase of $\tilde{X}_2(k)$ in Problem 5.3b

(c) Changing the period from $N = 50$ to $N = 100$ resulted in a lower frequency sampling interval (higher frequency resolution) $\omega_1$, i.e., in (3a) $\omega_1 = \pi/25$ and in (3b) $\omega_2 = \pi/50$. Hence there are more terms in the DFS expansion of $\tilde{x}_2(n)$. The shape of the DTFT begins to fill in with $N = 100$.

4. Problem **P 5.4**

New periodic sequence $\tilde{x}_3(n)$ with period $N = 100$

$$\tilde{x}_3(n) = [\tilde{x}_1(n), \tilde{x}_1(n)]_{\text{PERIODIC}}$$

(a) Computation of $\tilde{X}_3(k)$ using MATLAB:

```
clear; close all;
% (a) DFS Xtilde3(k)
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.4a');
n1 = [0:49]; xtilde1 = [n1(1:26).*exp(-0.3*n1(1:26)),zeros(1,24)]; N1 = length(n1);
n3 = [0:99]; xtilde3 = [xtilde1, xtilde1]; N3 = length(n3);
[Xtilde3] = dft(xtilde3,N3); k3 = n3;
mag_Xtilde3 = abs(Xtilde3); pha_Xtilde3 = angle(Xtilde3)*180/pi;
```

```
zei = find(mag_Xtilde3 < 0.00001);
pha_Xtilde3(zei) = zeros(1,length(zei));
subplot(3,1,1); stem(n3,xtilde3); axis([-1,N3,min(xtilde3),max(xtilde3)]);
title('One period of the periodic sequence xtilde3(n)'); ylabel('xtilde3');
ntick = [n3(1):5:n3(N3),N3]';
set(gca,'XTickMode','manual','XTick',ntick)
subplot(3,1,2); stem(k3,mag_Xtilde3); axis([-1,N3,min(mag_Xtilde3),max(mag_Xtilde3)
title('Magnitude of Xtilde3(k)'); ylabel('|Xtilde3|')
ktick = [k3(1):5:k3(N3),N3]';
set(gca,'XTickMode','manual','XTick',ktick)
subplot(3,1,3); stem(k3,pha_Xtilde3); axis([-1,N3,-180,180]);
title('Phase of Xtilde3(k)')
xlabel('k'); ylabel('Degrees')
ktick = [k3(1):5:k3(N3),N3]';
set(gca,'XTickMode','manual','XTick',ktick)
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180])
```

Plots of $\tilde{x}_3(n)$ and $\tilde{X}_3(k)$ are shown in Figure 5.3.

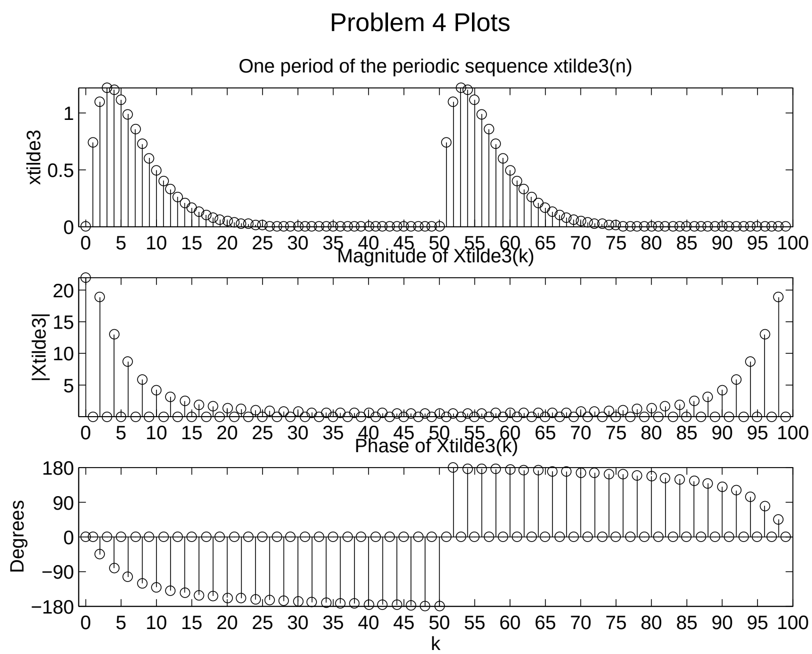


Figure 5.3: Plots of $\tilde{x}_3(n)$ and $\tilde{X}_3(k)$ in Problem 5.4a

(b) Comparing the magnitude plot above with that of $\tilde{X}_1(k)$ in Problem (3a), we observe that these plots are essentially similar. Plots of $\tilde{X}_3(k)$ have one zero between every sample of $\tilde{X}_1(k)$. (In general, for phase plots, we do get non-zero phase values when the magnitudes are zero. Clearly these phase values have no meaning and should be ignored. This happens because of a particular algorithm used by MATLAB. I avoided this problem by using the `find` function.) This makes sense because sequences $\tilde{x}_1(n)$ and $\tilde{x}_3(n)$, when viewed over $-\infty < n < \infty$ interval, look exactly same. The effect of periodicity doubling is in the doubling of magnitude of each sample.

(c) We can now generalize this argument. If

$$\tilde{x}_M(n) = \left\{ \underbrace{\tilde{x}_1(n), \tilde{x}_1(n), \ldots, \tilde{x}_1(n)}_{M \text{ times}} \right\}_{\text{PERIODIC}}$$

then there will be $(M-1)$ zeros between samples of $\tilde{X}_M(k)$. The magnitudes of non-zero samples of $\tilde{X}_M(k)$ will be $M$ times the magnitudes of the samples of $\tilde{X}_1(k)$, i.e.,

$$\begin{aligned} \tilde{X}_M(Mk) &= M\tilde{X}_1(k) &&, k = 0,1,\ldots,N-1 \\ \tilde{X}_M(k) &= 0 &&, k \neq 0,1,\ldots,MN \end{aligned}$$

5. Problem **P 5.5**

   $X\left(e^{j\omega}\right)$ is a DTFT of a 10-point sequence $x(n) = \{2,5,3,-4,-2,6,0,-3,-3,2\}$.

   (a) Since $y_1(n)$ is a 3-point IDFS of three samples of $X\left(e^{j\omega}\right)$ on the unit circle, it can be obtained as a 3-point aliasing operation on $x(n)$. Thus

   $$y_1(n) = \{2+(-4)+0+2, 5+(-2)+(-3), 3+6+(-3)\} = \{0,0,6\}_{\text{periodic}}$$

   MATLAB verification:

```
clear; close all;
x = [2,5,3,-4,-2,6,0,-3,-3,2]; n = 0:9;
% (a) y1(n) = 3-point IDFS{X(0),X(2*pi/3),X(4*pi/3)}
N = 3; k = 0:N-1; w = 2*pi*k/N;
[Y1] = dtft(x,n,w); y1 = real(idfs(Y1,N))
y1 =
    0.0000    0.0000    6.0000
```

   (b) Since $y_2(n)$ is a 20-point IDFS of twenty samples of $X\left(e^{j\omega}\right)$ on the unit circle, then from the frequency sampling theorem there will not be any aliasing of $x(n)$ and $y_2(n)$ will be a zero-padded version of $x(n)$. Thus

   $$y_2(n) = \{2,5,3,-4,-2,6,0,-3,-3,2,0,0,0,0,0,0,0,0,0,0\}_{\text{periodic}}$$

   MATLAB verification:

```
% (b) y1(n) = 20-point IDFS{X(0),X(2*pi/20),...,X(28*pi/20)}
N = 20; k = 0:N-1; w = 2*pi*k/N;
[Y2] = dtft(x,n,w); y2 = real(idfs(Y2,N))
y2 =
  Columns 1 through 7
    2.0000    5.0000    3.0000   -4.0000   -2.0000    6.0000    0.0000
  Columns 8 through 14
   -3.0000   -3.0000    2.0000    0.0000    0.0000    0.0000    0.0000
  Columns 15 through 20
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

6. Problem **P 5.6**

   A 12-point sequence $x(n) = \{1,2,3,4,5,6,6,5,4,3,2,1\}$.

   (a) DFT $X(k)$:

```
clear; close all;
%
xn = [1,2,3,4,5,6,6,5,4,3,2,1]; N = length(xn); % given signal x(n)
Xk = dft(xn,N); k = 0:N-1;                      % DFT of x(n)
mag_Xk = abs(Xk); pha_Xk = angle(Xk)*180/pi;    % Mag and Phase of X(k)
zei = find(mag_Xk < 0.00001);                   % Set phase values to
pha_Xk(zei) = zeros(1,length(zei));             %  zero when mag is zero
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','HS020500');
subplot(2,1,1); stem(k,mag_Xk); axis([0,N,0,40])
```

```
set(gca,'XTickMode','manual','XTick',[0:1:N]);
set(gca,'YTickMode','manual','YTick',[0;20;40]);
xlabel('k'); ylabel('magnitude'); title('Magnitude plots of DFT and DTFT')
hold on
subplot(2,1,2); stem(k,pha_Xk); axis([0,N,-180,180])
set(gca,'XTickMode','manual','XTick',[0:1:N]);
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180]);
xlabel('k'); ylabel('Degrees'); title('Phase plots of DFT and DTFT')
hold on
```

The `stem` plot of $X(k)$ is shown in 5.4.

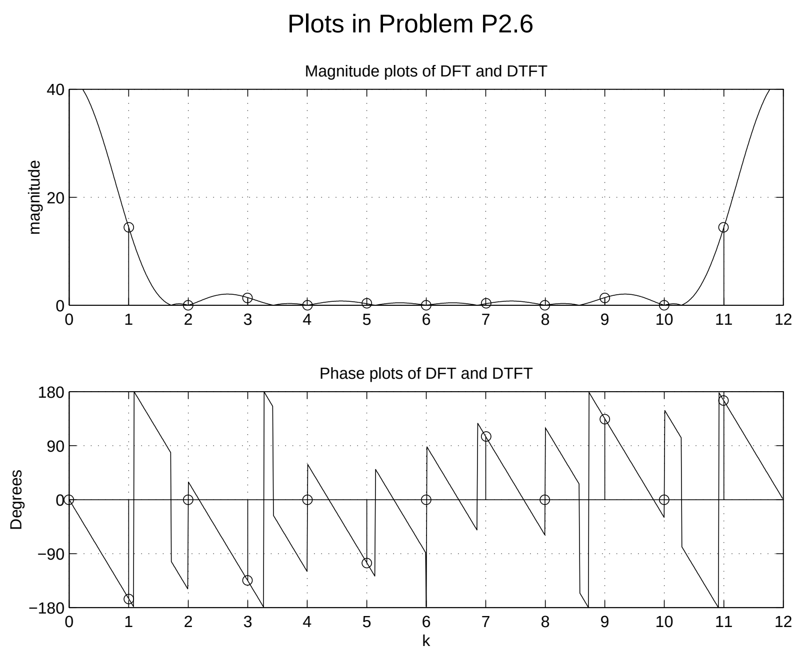


Figure 5.4: Plots of DTFT and DFT of signal in Problem 5.6

(b) DTFT $X\left(e^{j\omega}\right)$:

```
[X,w] = freqz(xn,1,1000,'whole');          % DTFT of xn
mag_X = abs(X); pha_X = angle(X)*180/pi;   % mag and phase of DTFT
Dw = (2*pi)/N;                             % frequency resolution
subplot(2,1,1); plot(w/Dw,mag_X); grid
hold off
subplot(2,1,2); plot(w/Dw,pha_X); grid
hold off
```

The continuous plot of $X\left(e^{j\omega}\right)$ is also shown in Figure 5.4.

(c) Clearly, the DFT in part (6a) is the sampled version of $X\left(e^{j\omega}\right)$.

(d) It is possible to reconstruct the DTFT from the DFT if length of the DFT is larger than or equal to the length of sequence $x(n)$. We can reconstruct using the complex interpolation formula

$$X\left(e^{j\omega}\right) = \sum_{k=0}^{N-1} X(k)\,\phi\left(\omega - \frac{2\pi k}{N}\right), \quad \text{where} \quad \phi(\omega) = e^{-j\omega(N-1)/2}\frac{\sin(\omega N/2)}{N\sin(\omega/2)}$$

For $N = 12$, we have

$$X\left(e^{j\omega}\right) = \sum_{k=0}^{11} X(k)\, e^{-j(5.5)\omega} \frac{\sin(6\omega)}{12\sin(\omega/2)}$$

7. Problem **P 5.7**

This problem is done using MATLAB.

(a) $x_1(n) = 2\cos(0.2\pi n)[u(n) - u(n-10)]$.

```
% (a) x1(n) = 2*cos(0.2*pi*n)*[u(n)-u(n-10)]
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.7a');
n1 = [0:9]; x1 = 2*cos(0.2*pi*n1); N1 = length(n1); N = 100;   % Length of DFT
[X1] = dft([x1, zeros(1,N-N1)],N);
mag_X1 = abs(X1(1:N/2+1)); w = (0:N/2)*2*pi/N;
subplot(2,1,1); stem(n1,x1); axis([-1,N1,min(x1),max(x1)]);
title('Sequence x1(n)'); ylabel('x1'); ntick = [n1(1):5:n1(N1),N1]';
set(gca,'XTickMode','manual','XTick',ntick); xlabel('n');
subplot(2,1,2); plot(w/pi,mag_X1); axis([0,1,0,max(mag_X1)]);
title('Magnitude of DTFT X1(omega)'); ylabel('|X1|');
xlabel('frequency in pi units'); suptitle('Problem P 5.7a Plots');
```

The plots are shown in Figure 5.5.

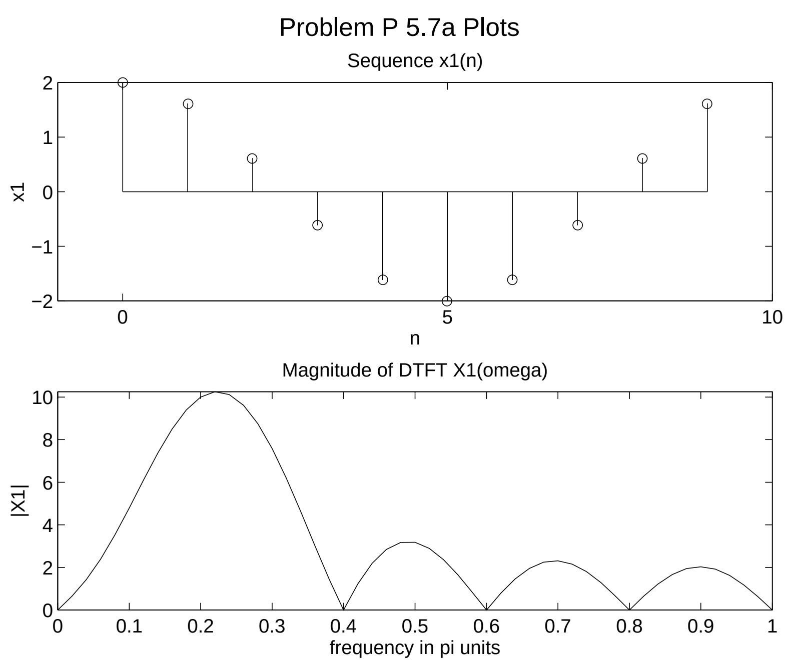


Figure 5.5: Sequence and its DTFT magnitude plots in Problem P5.7a

(b) $x_2(n) = \sin(0.45\pi n)\sin(0.55\pi n)$, $0 \le n \le 50$.

```
% (b) x2(n) = sin(0.45*pi*n)*sin(0.55*pi*n), 0 <= n <= 50
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P5.7b');
n2 = [0:50]; x2 = sin(0.45*pi*n2).*sin(0.55*pi*n2); N2 = length(n2);  N = 300;
[X2] = dft([x2, zeros(1,N-N2)],N);
```

```
mag_X2 = abs(X2(1:N/2+1)); w = (0:N/2)*2*pi/N;
subplot(2,1,1); stem(n2,x2); axis([-1,N2,min(x2),max(x2)]);
title('Sequence x2(n)'); ylabel('x2'); ntick = [n2(1):10:n2(N2)]';
set(gca,'XTickMode','manual','XTick',ntick); xlabel('n');
subplot(2,1,2); plot(w/pi,mag_X2); axis([0,1,0,max(mag_X2)]);
title('Magnitude of DTFT X2(omega)'); ylabel('|X2|');
xlabel('frequency in pi units'); suptitle('Problem P 5.7b Plots');
```

The plots are shown in Figure 5.6.

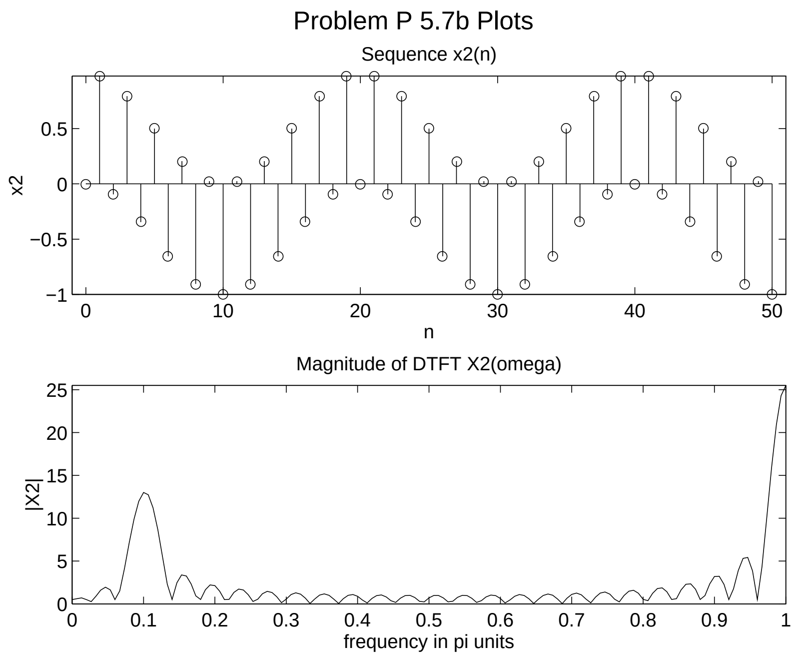


Figure 5.6: Sequence and its DTFT magnitude plots in Problem P5.7b

(c) $x_3(n) = 3(2)^n$, $-10 \le n \le 10$. This problem is done using MATLAB.

```
% (c) x3(n) = 3*(2)^n*[u(n+10)-u(n-11)]
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.7c');
n3 = [-10:10]; x3 = 3*(2).^n3; N3 = length(n3); N = 100;   % Length of DFT
[X3] = dft([x3, zeros(1,N-N3)],N);
mag_X3 = abs(X3(1:N/2+1)); w = (0:N/2)*2*pi/N;
subplot(2,1,1); stem(n3,x3); axis([-11,11,min(x3),max(x3)]);
title('Sequence x3(n)'); ylabel('x3'); ntick = [n3(1):5:n3(N3),N3]';
set(gca,'XTickMode','manual','XTick',ntick); xlabel('n');
subplot(2,1,2); plot(w/pi,mag_X3); axis([0,1,0,max(mag_X3)]);
title('Magnitude of DTFT X3(omega)'); ylabel('|X3|');
xlabel('frequency in pi units'); suptitle('Problem P 5.7c Plots');
```

The plots are shown in Figure 5.7.

(e) $x_5(n) = 5\left(0.9e^{j\pi/4}\right)^n u(n)$.

```
% (e) x5(n) = 5*(0.9*exp(j*pi/4)).^n*u(n)
Hf_1 = figure('Units','normalized','position',[.1,.1,.8,.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.7e');
```

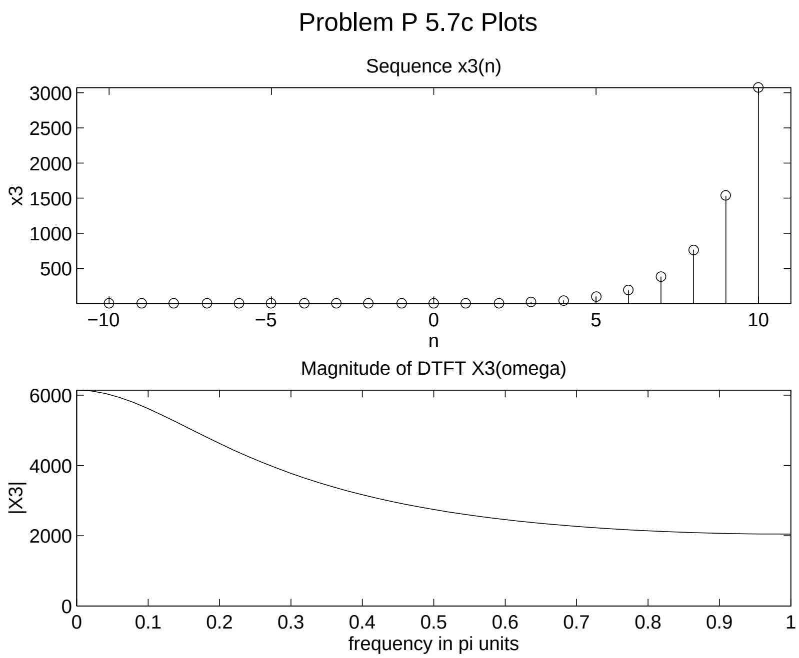


Figure 5.7: Plots in Problem P5.7c

```
n5 = [0:99]; x5 = 5*(0.9*exp(j*pi/4)).^n5;
N5 = length(n5);   % Length of DFT
[X5] = dft(x5,N5); x5 = real(x5);
mag_X5 = abs(X5(1:N5/2+1)); w = (0:N5/2)*2*pi/N5;
subplot(2,1,1); stem(n5,x5); %axis([-11,11,min(x5),max(x5)]);
title('Sequence real(x5(n))'); ylabel('real(x5)'); ntick = [n5(1):10:n5(N5),N5]';
set(gca,'XTickMode','manual','XTick',ntick); xlabel('n');
subplot(2,1,2); plot(w/pi,mag_X5); axis([0,1,0,max(mag_X5)]);
title('Magnitude of DTFT X5(omega)'); ylabel('|X5|');
xlabel('frequency in pi units'); suptitle('Problem P 5.7e Plots');
```

The plots are shown in Figure 5.8.

8. Problem **P 5.8**

The impulse response $h(n)$ is causal and real-valued.

(a) It is known that $\mathrm{Re}\left\{H\left(e^{j\omega}\right)\right\} = \sum_{k=0}^{5}(0.5)^k \cos\omega k$. Consider

$$\mathrm{Re}\left\{H\left(e^{j\omega}\right)\right\} = \mathrm{Re}\left\{\sum_{k=0}^{\infty} h(k)\,e^{-j\omega k}\right\} = \sum_{k=0}^{\infty} h(k)\,\mathrm{Re}\left\{e^{-j\omega k}\right\} = \sum_{k=0}^{\infty} h(k)\cos\omega k$$

Comparing with the given expression, we obtain

$$h(n) = \begin{cases} (0.5)^k, & 0 \le k \le 5 \\ 0, & \text{else} \end{cases}$$

(b) It is known that $\mathrm{Im}\left\{H\left(e^{j\omega}\right)\right\} = \sum_{\ell=0}^{5} 2\ell \sin\omega\ell$ and $\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) d\omega = 0$. From the second condition

$$\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) d\omega = h(0) = 0$$

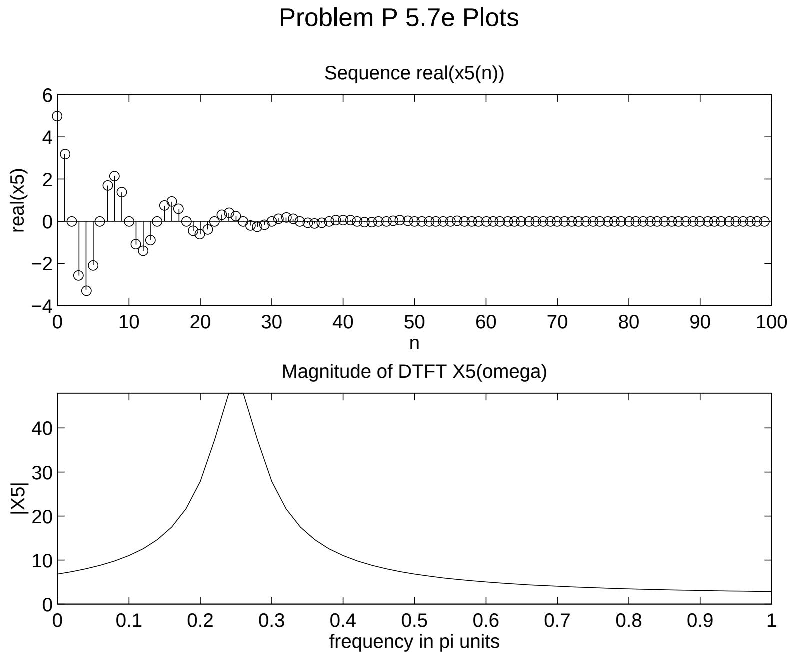


Figure 5.8: DTFT plot in Problem P5.7e.

Consider

$$\mathrm{Im}\left\{H\left(e^{j\omega}\right)\right\} = \mathrm{Im}\left\{\sum_{\ell=0}^{\infty} h(\ell)\, e^{-j\omega\ell}\right\} = \sum_{\ell=0}^{\infty} h(\ell)\,\mathrm{Im}\left\{e^{-j\omega\ell}\right\} = -\sum_{\ell=0}^{\infty} h(\ell)\sin\omega\ell$$

Comparing with the given expression, we obtain

$$h(n) = \begin{cases} -2\ell, & 0 \le \ell \le 5 \\ 0, & \text{else} \end{cases}$$

9. Problem **P 5.9**

An $N$-point sequence $x(n)$:

(a) The $N$-point DFT of $x(n)$: $X(k) = \sum_{m=0}^{N-1} x(m)\, W_N^{mk}$. The $N$-point DFT of $X(k)$: $y(n) = \sum_{k=0}^{N-1} X(k)\, W_N^{kn}$. Hence,

$$\begin{aligned} y(n) &= \sum_{k=0}^{N-1}\left\{\sum_{m=0}^{N-1} x(m)\, W_N^{mk}\right\} W_N^{kn} = \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} W_N^{mk} W_N^{kn},\ 0 \le n \le N-1 \\ &= \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} W_N^{(m+n)k} = \sum_{m=0}^{N-1} x(m) \sum_{r=-\infty}^{\infty} N\delta(m+n-rN),\, 0 \le n \le N-1 \\ &= N \sum_{r=-\infty}^{\infty} x(-n+rN) = N x((-n))_N,\, 0 \le n \le N-1 \end{aligned}$$

This means that $y(n)$ is a “circularly folded and amplified (by $N$)” version of $x(n)$. Continuing further, if we take two more DFTs of $x(n)$ then

$$x(n) \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow N^2 x(n)$$

Therefore, if a given DFT function is working correctly then four successive applications of this function on any arbitrary signal will produce the same signal (multiplied by $N^2$). This approach can be used to verify a DFT function.

(b) MATLAB function for circular folding:

```
function x2 = circfold(x1,N)
% Circular folding with respect to N
% ---------------------------------
% function x2 = circfold(x1,N)
%          x2(n) = x1((-n) mod N)
%
x2 = real(dft(dft(x1,N),N))/N;
```

(c) MATLAB verification:

```
x = [1,2,3,4,5,6,6,5,4,3,2,1], N = length(x);
x =
     1     2     3     4     5     6     6     5     4     3     2     1
y = circfold(x,N)
y =
  Columns 1 through 7
    1.0000    1.0000    2.0000    3.0000    4.0000    5.0000    6.0000
  Columns 8 through 12
    6.0000    5.0000    4.0000    3.0000    2.0000
```

Clearly, the circular folding of the sequence is evident.

10. Problem P 5.10. Circular even-odd decomposition of a complex sequence.

$$x_{ec}(n) \triangleq \frac{1}{2}\left[x(n) + x^*((-n))_N\right]$$
$$x_{oc}(n) \triangleq \frac{1}{2}\left[x(n) + x^*((-n))_N\right]$$

(a) Using the DFT properties of conjugation and circular folding, we obtain

$$\begin{aligned}\mathrm{DFT}\left[x_{ec}(n)\right] &= \frac{1}{2}\left\{\mathrm{DFT}\left[x(n)\right] + \mathrm{DFT}\left[x^*((-n))_N\right]\right\} \\ &= \frac{1}{2}\left\{X(k) + \tilde{X}^*((-k))_N\right\}, \text{ where } \tilde{X}(k) = \mathrm{DFT}\left[x((-n))_N\right] \\ &= \frac{1}{2}\left\{X(k) + X^*(k)\right\} = \mathrm{Re}\left[X(k)\right] = \mathrm{Re}\left[X((-k))_N\right]\end{aligned}$$

similarly, we can show that

$$\mathrm{DFT}\left[x_{oc}(n)\right] = j\,\mathrm{Im}\left[X(k)\right] = j\,\mathrm{Im}\left[X((-k))_N\right]$$

(b) The modified `circevod` function:

```
function [xec, xoc] = circevod(x)
% Complex-valued signal decomposition into circular-even and circular-odd parts
% ----------------------------------------------------------------------------
--
% [xec, xoc] = circecod(x)
%
N = length(x); n = 0:(N-1);
xec = 0.5*(x + conj(x(mod(-n,N)+1)));
xoc = 0.5*(x - conj(x(mod(-n,N)+1)));
```

(c) MATLAB verification:

```
% (c) Matlab Verification
n = 0:19; x = (0.9*exp(j*pi/3)).^n; N = length(x);
[xec, xoc] = circevod(x);
```

```
X = dft(x,N); Xec = dft(xec,N); Xoc = dft(xoc,N);
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.10');
subplot(2,2,1); stem(n,real(X)); axis([-0.5,20.5,-1,7]);
title('Real{DFT[x(n)]}'); ylabel('Re{X(k)}');
subplot(2,2,3); stem(n,real(Xec)); axis([-0.5,20.5,-1,7]);
title('DFT[xec(n)]'); ylabel('Xec(k)'); xlabel('k');
subplot(2,2,2); stem(n,imag(X)); axis([-0.5,20.5,-5,5]);
title('Imag{DFT[x(n)]}'); ylabel('Im{X(k)}');
subplot(2,2,4); stem(n,imag(Xoc)); axis([-0.5,20.5,-5,5]);
title('DFT[xoc(n)]'); ylabel('Xoc(k)'); xlabel('k');
suptitle('Problem P 5.10 Plots');
```

The plots are shown in Figure 5.9.

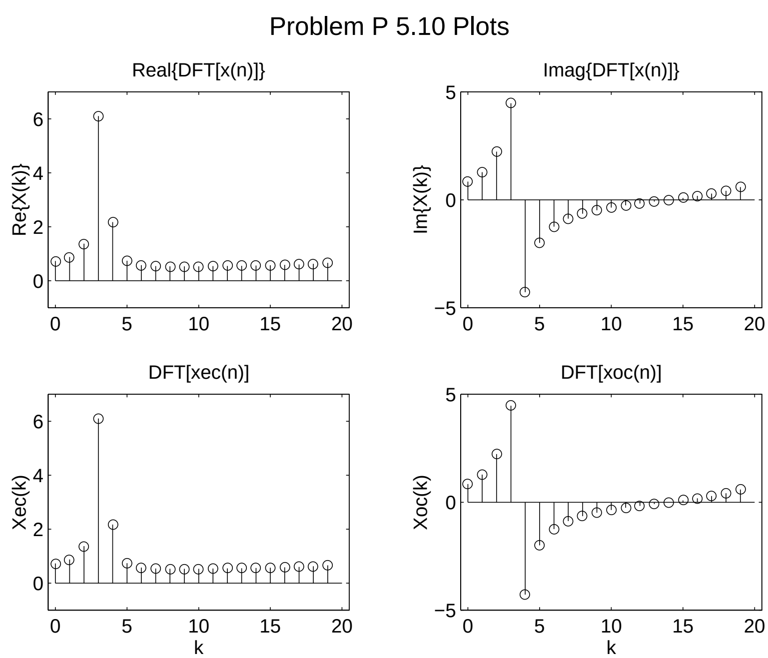


Figure 5.9: Plots in Problem P5.10

11. Problem **P 5.11**

Since $x(n)$ is a real-valued sequence, its DFT is conjugate symmetric. Thus

$$X(k) = \{0.25, 0.125 - j0.3, 0, 0.125 - j0.06, 0.5, 0.125 + j0.06, 0, 0.125 + j0.3\}.$$

From this we can compute the 8-point sequence $x(n)$. MATLAB script:

```
clear, close all;
N = 8; X = [0.25,0.125-j*0.3,0,0.125-j*0.06,0.5];
X = [X,conj(X(4:-1:2))]            % Reconstruct the entire X(k)
X =
  Columns 1 through 4
   0.2500            0.1250 - 0.3000i        0             0.1250 - 0.0600i
  Columns 5 through 8
```

```
   0.5000             0.1250 + 0.0600i        0             0.1250 + 0.3000i
x = real(idft(X,N))
x =
  Columns 1 through 7
    0.1562    0.0324    0.1538    0.0324    0.0312   -0.0949    0.0337
  Column 8
   -0.0949
```

(a) Consider $x_1(n) = x((2-n))_8 = x((-[n-2]))_8$. It is obtained by first circular folding and then circular shifting by 2 of $x(n)$. Hence using properties of the DFT

$$X_1(k) = \text{DFT}[x((2-n))_8] = W_8^{-2k} X_{8}((-k))_8 .$$

MATLAB script:

```
% (a) x1(n) = x((2-n))_8; Circ folding followed by circ shifting by 2
N = 8; WN = exp(-j*2*pi/N); k = 0:N-1; m = 2;
X1 = circfold(X,N)
X1 =
  Columns 1 through 4
   0.2500 + 0.0000i   0.1250 + 0.3000i   0.0000 - 0.0000i   0.1250 + 0.0600i
  Columns 5 through 8
   0.5000 + 0.0000i   0.1250 - 0.0600i   0.0000 - 0.0000i   0.1250 - 0.3000i
X1 = (WN.^(m*k)).*X1
X1 =
  Columns 1 through 4
   0.2500 + 0.0000i   0.3000 - 0.1250i   0.0000 + 0.0000i  -0.0600 + 0.1250i
  Columns 5 through 8
   0.5000 + 0.0000i  -0.0600 - 0.1250i   0.0000 + 0.0000i   0.3000 + 0.1250i
% Matlab verification
x1 = circfold(x,N); x1 = cirshftt(x1,m,N);
X12 = dft(x1,N)
X12 =
  Columns 1 through 4
   0.2500 + 0.0000i   0.3000 - 0.1250i   0.0000 + 0.0000i  -0.0600 + 0.1250i
  Columns 5 through 8
   0.5000 + 0.0000i  -0.0600 - 0.1250i   0.0000 - 0.0000i   0.3000 + 0.1250i
difference = max(abs(X1-X12))
difference =
  4.7692e-016
```

(b) Let $X_{10}(k)$ be the 10-point DFT of $x(n)$. Then

$$\text{DFT}[x((n+5))_{10}] = W_{10}^{5k} X_{10}(k)$$

Note that $x((n+5))_{10}$ is 10-point circularly shifted sequence of $x(n)$, shifted to the left by 5 samples. MATLAB script:

```
% (b) x2(n) = x((n+5))_{10}
N = 10; WN = exp(-j*2*pi/N); k = 0:N-1; m = -5;
X2 = (WN.^(m*k)).*dft([x,0,0],N)
X2 =
  Columns 1 through 4
   0.2500            -0.2916 + 0.2848i   0.0176 - 0.0487i  -0.0863 - 0.1368i
  Columns 5 through 8
```

```
   0.0108 + 0.0047i  -0.5000 + 0.0000i   0.0108 - 0.0047i  -0.0863 + 0.1368i
  Columns 9 through 10
   0.0176 + 0.0487i  -0.2916 - 0.2848i
% Matlab verification
x2 = cirshftt(x,m,N); X22 = dft(x2,N)
X22 =
  Columns 1 through 4
   0.2500            -0.2916 + 0.2848i   0.0176 - 0.0487i  -0.0863 - 0.1368i
  Columns 5 through 8
   0.0108 + 0.0047i  -0.5000 - 0.0000i   0.0108 - 0.0047i  -0.0863 + 0.1368i
  Columns 9 through 10
   0.0176 + 0.0487i  -0.2916 - 0.2848i
difference = max(abs(X2-X22))
difference =
  8.2523e-016
```

(c) Consider $x_3(n) = x^2(n) = x(n)x(n)$. Hence the DFT of $x_3(n)$ is given by

$$X_3(k) = \frac{1}{N} X(k) \, (N) \, X(k), \quad N = 8.$$

MATLAB script:

```
% (c) x3(n) = x(n)*x(n);
N = 8;
X3 = circonvf(X,X,N)/N;
% Matlab verification
x3 = x.*x;
X32 = dft(x3,N);
difference = max(abs(X3-X32))
difference =
  4.5103e-017
```

(d) Let $X_{(10)}(k)$ be the 10-point DFT of $x(n)$. Then

$$\text{DFT}\left[ x(n) \, (8) \, x((-n))_8 \right] = X(k) X((-k))_8$$

Note that $x((-n))_8$ and $X((-k))_8$ are the 8-point circular foldings of their respective sequences. MATLAB script:

```
clear, close all;
N = 8; X = [0.25,0.125-j*0.3,0,0.125-j*0.06,0.5]; k = 0:N-1; n = k;
X = [X,conj(X(4:-1:2))]            % Reconstruct the entire X(k)
X =
  Columns 1 through 4
   0.2500             0.1250 - 0.3000i        0              0.1250 - 0.0600i
  Columns 5 through 8
   0.5000             0.1250 + 0.0600i        0              0.1250 + 0.3000i
x = real(idft(X,N))
x =
  Columns 1 through 7
    0.1562    0.0324    0.1538    0.0324    0.0312   -0.0949    0.0337
  Column 8
   -0.0949
%
% (d) x4(n) = x(n)(8)x((-n))_8
X0 = X(mod(-k,N)+1);               % DFT of x((-n))_8
```

```
X4 = X .* X0                        % DFT of x(n) (8) x((-n))_8
X4 =
  Columns 1 through 7
    0.0625    0.1056         0    0.0192    0.2500    0.0192         0
  Column 8
    0.1056
% Verification
x4 = circonvt(x,x(mod(-n,N)+1),N); X44 = dft(x4,N);
difference = max(abs(X4-X44))
difference =
  2.1330e-016
```

12. Problem **P 5.12** (Two real DFTs using one complex DFT)

$$x(n) = x_R(n) + jx_I(n);\ x_R(n) \text{ and } x_I(n) \text{ are real sequences}$$

Then

$$\begin{aligned} x_R(n) &= \frac{1}{2}\{x(n) + x^*(n)\}, \quad \text{and} \\ jx_I(n) &= \frac{1}{2}\{x(n) - x^*(n)\} \end{aligned}$$

(a) Consider

$$\begin{aligned} X_R(k) &\stackrel{\text{def}}{=} \text{DFT}[x_R(n)] = \frac{1}{2}\{\text{DFT}[x(n)] + \text{DFT}[x^*(n)]\} \\ &= \frac{1}{2}\{X(k) + X^*((-k))_N\} \stackrel{\text{def}}{=} X_{ec} \end{aligned}$$

Similarly

$$\left.\begin{aligned} jX_I(k) &\stackrel{\text{def}}{=} \text{DFT}[jx_I(n)] = \frac{1}{2}\{\text{DFT}[x(n)] - \text{DFT}[x^*(n)]\} \\ &= \frac{1}{2}\{X(k) - X^*((-k))_N\} \stackrel{\text{def}}{=} X_{oc} \end{aligned}\right\} \Rightarrow X_I = \frac{X_{oc}}{j} = -jX_{oc}$$

(b) MATLAB function `real2dft`:

```
function [X1,X2] = real2dft(x1,x2,N)
% DFTs of two real sequences
% [X1,X2] = real2dft(x1,x2,N)
%  X1 = N-point DFT of x1
%  X2 = N-point DFT of x2
%  x1 = real-valued sequence of length <= N
%  x2 = real-valued sequence of length <= N
%   N = length of DFT
%
% Check for length of x1 and x2
if length(x1) > N
        error('*** N must be >= the length of x1 ***')
end
if length(x2) > N
        error('*** N must be >= the length of x2 ***')
end
N1 = length(x1); x1 = [x1 zeros(1,N-N1)];
N2 = length(x2); x2 = [x2 zeros(1,N-N2)];
x = x1 + j*x2;
X = dft(x,N);
[X1, X2] = circevod(X); X2 = X2/j;
```

We will also need the `circevod` function for complex sequences (see Problem P5.10). This can be obtained from the one given in the text by two simple changes.

```
function [xec, xoc] = circevod(x)
% Complex signal decomposition into circular-even and circular-odd parts
% -----------------------------------------------------------------------
% [xec, xoc] = circecod(x)
%
N = length(x); n = 0:(N-1);
xec = 0.5*(x + conj(x(mod(-n,N)+1)));
xoc = 0.5*(x - conj(x(mod(-n,N)+1)));
```

(c) MATLAB verification:

```
clear; close all;
N = 64; n = 0:N-1; x1 = cos(0.25*pi*n); x2 = sin(0.75*pi*n);
[X1,X2] = real2dft(x1,x2,N);
X11 = dft(x1,N); X21 = dft(x2,N);
difference = max(abs(X1-X11))
difference =
  1.4918e-013
difference = max(abs(X2-X21))
difference =
  1.4914e-013
```

13. Problem **P 5.13**

Circular shifting:

(a) The MATLAB routine `cirshftf.m` to implement circular shift is written using the frequency-domain property

$$y(n) \stackrel{\triangle}{=} x((n-m))_N = \text{IDFT}\left[X(k)\,W_N^{mk}\right]$$

This routine will be used in the next problem to generate a circulant matrix and has the following features. If $m$ is a scaler then $y(n)$ is circularly shifted sequence (or array). If $m$ is a vector then $y(n)$ is a matrix, each row of which is a circular shift in $x(n)$ corresponding to entries in the vector $m$.

```
function y = cirshftf(x,m,N)
%  Circular shift of m samples wrt size N in sequence x: (freq domain)
%  -------------------------------------------------------------------
% function y=cirshift(x,m,N)
%       y : output sequence containing the circular shift
%       x : input sequence of length <= N
%       m : sample shift
%       N : size of circular buffer
%
%  Method: y(n) = idft(dft(x(n))*WN^(mk))
%
% If m is a scalar then y is a sequence (row vector)
% If m is a vector then y is a matrix, each row is a circular shift
% in x corresponding to entries in vecor m
% M and x should not be matrices
%
% Check whether m is scalar, vector, or matrix
[Rm,Cm] = size(m);
if Rm > Cm
m = m'; % make sure that m is a row vector
```

```
end
[Rm,Cm] = size(m);
if Rm > 1
error('*** m must be a vector ***') % stop if m is a matrix
end
% Check whether x is scalar, vector, or matrix
[Rx,Cx] = size(x);
if Rx > Cx
x = x'; % make sure that x is a row vector
end
[Rx,Cx] = size(x);
if Rx > 1
error('*** x must be a vector ***') % stop if x is a matrix
end
% Check for length of x
if length(x) > N
        error('N must be >= the length of x')
end
x=[x zeros(1,N-length(x))];
X=dft(x,N);
X=ones(Cm,1)*X;
WN=exp(-2*j*pi/N);
k=[0:1:N-1];
Y=(WN.^(m' * k)).*X;
y=real(conj(dfs(conj(Y),N)))/N;
```

(b) MATLAB verification:

```
n = [0:1:10]; x = 11*ones(1,length(n))-n;
y = cirshftf(x,10,15)
y =
  Columns 1 through 7
    6.0000    5.0000    4.0000    3.0000    2.0000    1.0000    0.0000
  Columns 8 through 14
    0.0000    0.0000    0.0000   11.0000   10.0000    9.0000    8.0000
  Column 15
    7.0000
```

14. Problem **P 5.14**

Parseval's relation for the DFT:

$$\begin{aligned}\sum_{n=0}^{N-1}|x(n)|^2 &= \sum_{n=0}^{N-1}x(n)x^*(n) = \sum_{n=0}^{N-1}\left\{\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk}\right\}x^*(n) \\ &= \frac{1}{N}\sum_{k=0}^{N-1}X(k)\left\{\sum_{n=0}^{N-1}x^*(n)W_N^{-nk}\right\} = \frac{1}{N}\sum_{k=0}^{N-1}X(k)\left\{\sum_{n=0}^{N-1}x(n)W_N^{nk}\right\}^*\end{aligned}$$

Therefore,

$$\sum_{n=0}^{N-1}|x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1}X(k)X^*(k) = \frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2$$

MATLAB verification:

```
x = [1,2,3,4,5,6,6,5,4,3,2,1]; N = length(x);
% power of x(n) in the time-domain
power_x = sum(x.*conj(x))
```

```
power_x =
   182
X = dft(x,N);
power_X = (1/N)*sum(X.*conj(X))
power_X =
  182.0000
```

15. Problem **P 5.15**

MATLAB function `circonvf`:

```
function y = circonvf(x1,x2,N)
%
%function y=circonvf(x1,x2,N)
%
%  N-point circular convolution between x1 and x2: (freq domain)
%  ---------------------------------------------------------------
%       y  : output sequence containing the circular convolution
%       x1 : input sequence of length N1 <= N
%       x2 : input sequence of length N2 <= N
%       N  : size of circular buffer
%
%  Method: y(n) = idft(dft(x1)*dft(x2))

% Check for length of x1
if length(x1) > N
        error('N must be >= the length of x1')
end

% Check for length of x2
if length(x2) > N
        error('N must be >= the length of x2')
end

x1=[x1 zeros(1,N-length(x1))];
x2=[x2 zeros(1,N-length(x2))];

X1=fft(x1); X2=fft(x2);
y=real(ifft(X1.*X2));
```

16. Problem P 5.16. Circular convolution using circulant matrix operation.

$$x_1(n) = \{1,2,2\},\ x_2(n) = \{1,2,3,4\},\ x_3(n) \triangleq x_1(n) \textcircled{4} x_2(n)$$

(a) Using the results from Example 5.13, we can express the above signals as

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

The matrix $\mathbf{X}_2$ has the property that its every row or column can be obtained from the previous row or column using circular shift. Such a matrix is called a *circulant* matrix. It is completely described by the first column or the row. The following MATLAB function `circulnt` uses the `mod` function to generate a circulant matrix.

```
function C = circulnt(h,N)
%  Circulant matrix generation using vector data values
%  -----------------------------------------------------
%  function C = circulnt(h,N)
%
%   C : Circulant matrix
%   h : input sequence of length <= N
%   N : size of the circular buffer
%   Method: C = h((n-m) mod N); n : col vec, m : row vec

  Mh = length(h); h = reshape(h,Mh,1);   % reshape h into column vector
   h = [h; zeros(N-Mh,1)];               % zero-pad h
   C = zeros(N,N);                       % establish size of C
   m = 0:N-1; n=m';                      % indices n and m
  nm = mod((n*ones(1,N)-ones(N,1)*m),N); % (n-m) mod N in matrix form
C(:) = h(nm+1);                          % h((n-m) mod N)
```

(b) Circular convolution:

$$\mathbf{x}_3 = \mathbf{X}_2\mathbf{x}_1 = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 15 \\ 12 \\ 9 \\ 14 \end{bmatrix}$$

MATLAB script:

```
% Chapter 5: Problem P5.16: Circular convolution using Circulant matrix
clear, close all;
N = 4; x1 = [1,2,2,0]; x2 = [1,2,3,4];

% (a) Circulant matrix
X2 = circulnt(x2,N)
X2 =
     1     4     3     2
     2     1     4     3
     3     2     1     4
     4     3     2     1

% (b) Circular Convolution
x3 = X2*x1'; x3 = x3'
x3 =
    15    12     9    14
```

17. Problem **P 5.17**.

MATLAB function `circulnt`:

```
function C = circulnt(x,N)
%  Circulant matrix generation using vector data values
%  -----------------------------------------------------
%  function C = circulnt(h,N)
%
%   C : Circulant matrix
%   x : input sequence of length <= N
%   N : size of the circular buffer
%   Method: C = h((n-m) mod N);

  Mx = length(x);                        % length of x
```

```
x = [x, zeros(N-Mx,1)];              % zero-pad x
C = zeros(N,N);                      % establish size of C
m = 0:N-1;                           % indices n and m
x = circfold(x,N);                   % Circular folding
C = cirshift(x,m,N);                 % Circular shifting
```

MATLAB verification on sequences in Problem 5.16 (or Example 5.13):

```
clear, close all;
N = 4; x1 = [1,2,2,0]; x2 = [1,2,3,4];
% (a) Circulant matrix
X2 = circulnt(x2,N)
X2 =
    1.0000    4.0000    3.0000    2.0000
    2.0000    1.0000    4.0000    3.0000
    3.0000    2.0000    1.0000    4.0000
    4.0000    3.0000    2.0000    1.0000

% (b) Circular Convolution
x3 = X2*x1'; x3 = x3'
x3 =
   15.0000   12.0000    9.0000   14.0000
```

18. Problem **P 5.18**

(a) Circular convolution:

$$x_1(n) = \{1,1,1,1\},\ x_2(n) = \cos(\pi n/4)\,R_N(n),\ N = 8; \quad x_3(n) = x_1(n) \textcircled{N} x_2(n)$$

Circular convolutions using MATLAB:

```
clear, close all;
%(a) x1(n) = [1,1,1,1]; x2(n) = cos(pi*n/4); N = 8;
N = 8; n = 0:N-1;
x1 = [1,1,1,1,0,0,0,0]; x2 = cos(pi*n/4);
x3 = circonvt(x1,x2,N)
x3 =
  Columns 1 through 7
    1.0000    2.4142    2.4142    1.0000   -1.0000   -2.4142   -2.4142
  Column 8
   -1.0000
```

(d) Circular convolution:

$$x_1(n) = nR_N,\ x_2(n) = (N-n)\,R_N(n),\ N = 10; \quad x_3(n) = x_1(n) \textcircled{N} x_2(n)$$

Circular convolutions using MATLAB:

```
N = 10; n = [0:N-1]; x1 = n; x2 = (N-n);
x3 = real(idft(dft(x1,N).*dft(x2,N),N))
x3 =
  Columns 1 through 7
  285.0000  250.0000  225.0000  210.0000  205.0000  210.0000  225.0000
  Columns 8 through 10
  250.0000  285.0000  330.0000
```

19. Problem **P 5.19**

(b) Error between Linear and Circular Convolutions:

$$x_1(n) = \cos(2\pi n/N)\mathcal{R}_{16}(n), \; x_2(n) = \sin(2\pi n/N)\mathcal{R}_{16}(n); \quad N = 32$$

MATLAB script:

```
% (b) x1(n) = cos(2*pi*n/N)*R16, x2(n) = cos(2*pi*n/N)*R16; N = 32;
N = 32; n = 0:15;
x1 = cos(2*pi*n/N); x2 = sin(2*pi*n/N);
x3 = circonvf(x1,x2,N)
x3 =
  Columns 1 through 7
    0.0000    0.1951    0.5740    1.1111    1.7678    2.4944    3.2336
  Columns 8 through 14
    3.9231    4.5000    4.9039    5.0813    4.9888    4.5962    3.8890
  Columns 15 through 21
    2.8701    1.5607    0.0000   -1.3656   -2.4874   -3.3334   -3.8891
  Columns 22 through 28
   -4.1573   -4.1575   -3.9231   -3.5000   -2.9424   -2.3097   -1.6629
  Columns 29 through 32
   -1.0607   -0.5556   -0.1913    0.0000
x4 = conv(x1,x2), x4 = [x4, zeros(1,33)]; N4 = length(x4);
x4 =
  Columns 1 through 7
         0    0.1951    0.5740    1.1111    1.7678    2.4944    3.2336
  Columns 8 through 14
    3.9231    4.5000    4.9039    5.0813    4.9888    4.5962    3.8890
  Columns 15 through 21
    2.8701    1.5607    0.0000   -1.3656   -2.4874   -3.3334   -3.8891
  Columns 22 through 28
   -4.1573   -4.1575   -3.9231   -3.5000   -2.9424   -2.3097   -1.6629
  Columns 29 through 31
   -1.0607   -0.5556   -0.1913
 e = round(x3 - x4(1:N))
e =
  Columns 1 through 12
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 13 through 24
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 25 through 32
     0     0     0     0     0     0     0     0
x4(N+1:N4)
ans =
  Columns 1 through 12
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 13 through 24
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 25 through 32
     0     0     0     0     0     0     0     0
```

(e) Error between Linear and Circular Convolutions:

$$x_1(n) = \{1, -1, 1, -1\}, \; x_2(n) = \{1, 0, -1, 0\}; \quad N = 5$$

MATLAB script:

```
% (e) x1(n) = {1,-1,1,-1}, x2(n) = {1,0,-1,0}; N = 5
x1 = [1,-1,1,-1]; x2 = [1,0,-1,0]; N = 5;
x3 = circonvf(x1,x2,N)
x3 =
    2.0000   -1.0000    0.0000    0.0000   -1.0000
x4 = conv(x1,x2), N4 = length(x4);
x4 =
     1    -1     0     0    -1     1     0
 e = x3 - x4(1:N)
e =
    1.0000    0.0000    0.0000    0.0000    0.0000
x4(N+1:N4)
ans =
     1     0
```

20. Problem **P 5.21**.

$$x_1(n) = \{2,1,1,2\},\ x_2(n) = \{1,-1,-1,1\}$$

(a) Circular convolution $x_1(n) \textcircled{N} x_2(n)$ for $N = 4,7,8$. MATLAB script:

```
clear, close all;
x1 = [2,1,1,2]; x2 = [1,-1,1,-1];
% (a) Circular Convolution for N = 4,7,8
y_4 = round(circonv(x1,x2,4))
y_4 =
     0     0     0     0
y_7 = round(circonv(x1,x2,7))
y_7 =
     2    -1     2     0    -2     1    -2
y_8 = round(circonv(x1,x2,8))
y_8 =
     2    -1     2     0    -2     1    -2     0
```

(b) Linear convolution $x_1(n) * x_2(n)$. MATLAB script:

```
% (b) linear Convolution
y_l = conv(x1,x2)
y_l =
     2    -1     2     0    -2     1    -2
```

(c) The minimum value of $N$ should be 7.

(d) Since $x_1(n)$ and $x_2(n)$ are 4-point sequences, the minimum length should be $4+4-1=7$.

# Chapter 6

# Digital Filter Structures

1. Problem **P 6.1**

   A causal LTI system is described by

   $$y(n) = \sum_{k=0}^{5} \left(\frac{1}{2}\right)^k x(n-k) + \sum_{\ell=1}^{5} \left(\frac{1}{3}\right)^k y(n-\ell)$$

   and is driven by the input

   $$x(n) = u(n),\ 0 \le n \le 100$$

   The MATLAB script to determine various realizations is:

```
clear; close all;
%% Direct Forms %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Direct Form Coefficients ***')
*** Direct Form Coefficients ***
k = 0:5; b = (0.5).^k
b =
    1.0000    0.5000    0.2500    0.1250    0.0625    0.0312
l = 1:5; a = [1, -(1/3).^l]
a =
    1.0000   -0.3333   -0.1111   -0.0370   -0.0123   -0.0041
% Output Response
[x,n] = stepseq(0,0,100);
y_DF = filter(b,a,x);

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
     1
B =
    1.0000    0.5000    0.2500
    1.0000   -0.5000    0.2500
    1.0000    0.5000         0
A =
    1.0000    0.4522    0.0745
    1.0000   -0.1303    0.0843
    1.0000   -0.6553         0
% Output Response
```

```
y_CF = casfiltr(b0,B,A,x);

%% Parallel Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel Form Coefficients ***')
*** Parallel Form Coefficients ***
[C,B,A] = dir2par(b,a)
C =
   -7.5938
B =
    3.4526    0.8052
    3.3193   -0.0367
    1.8219         0
A =
    1.0000    0.4522    0.0745
    1.0000   -0.1303    0.0843
    1.0000   -0.6553         0
% Output Response
y_PF = parfiltr(C,B,A,x);

%% Filter response plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P6.1');

subplot(3,1,1); stem(n,y_DF); axis([-1,101,0,5])
set(gca,'XTickMode','manual','XTick',[0:25:100],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[0;4]);
ylabel('y_DF'); title('Direct Form Filter Response')

subplot(3,1,2); stem(n,y_CF); axis([-1,101,0,5])
set(gca,'XTickMode','manual','XTick',[0:25:100],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[0;4]);
ylabel('y_DF'); title('Cascade Form Filter Response')

subplot(3,1,3); stem(n,y_PF); axis([-1,101,0,5])
set(gca,'XTickMode','manual','XTick',[0:25:100],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[0;4]);
xlabel('n'); ylabel('y_DF'); title('Parallel Form Filter Response')
```

The block diagrams are shown in Figure 6.1 and the output responses are shown in Figure 6.2.

2. Problem **P 6.2**

The IIR filter is decribed by the system function

$$H(z) = 2\left(\frac{1+0z^{-1}+z^{-2}}{1-0.8z^{-1}+0.64z^{-2}}\right)\left(\frac{2-z^{-1}}{1-0.75z^{-1}}\right)\left(\frac{1+2z^{-1}+z^{-2}}{1+0.81z^{-2}}\right)$$

The given structure is a parallel form. The MATLAB script to determine various realizations is:

```
clear; close all;
%% Cascade Form Coefficients %%%%%%%%%%%%%%%%%%%%%%%%%%
b0 = [2]; B = [1,0,1; 2,-1,0; 1,2,1];
A = [1,-0.8,0.64; 1,-0.75,0; 1,0,0.81];

%% Direct Forms %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Direct Form Coefficients ***')
```

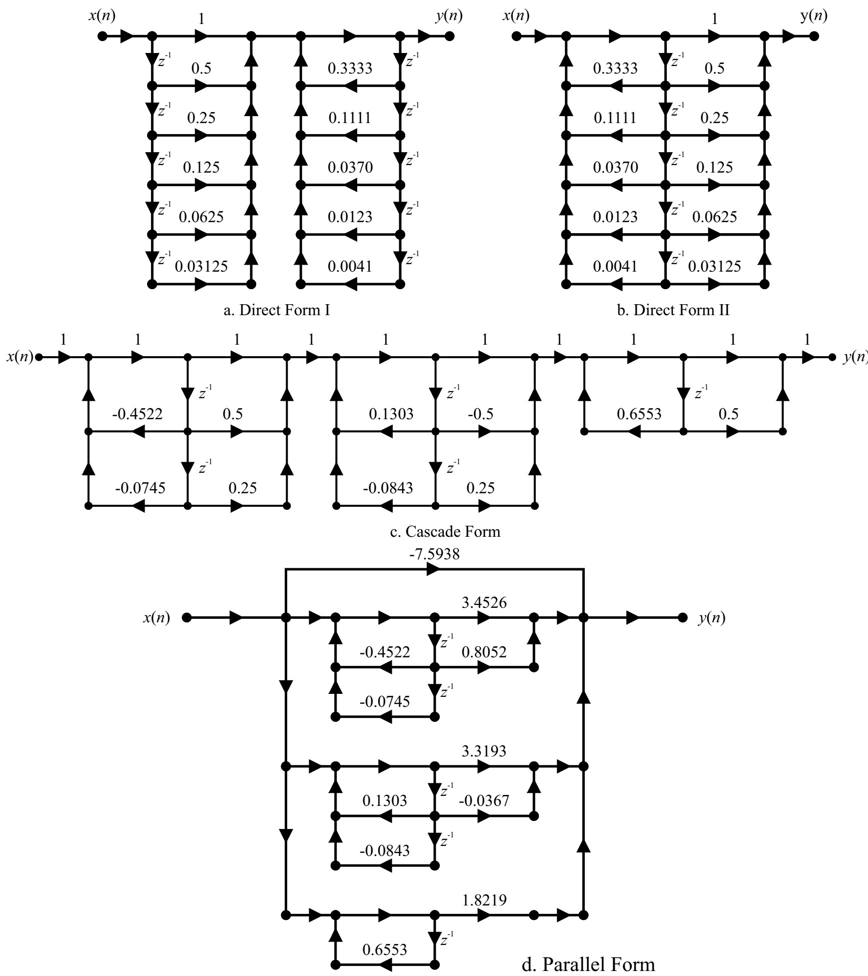


Figure 6.1: Block Diagrams in Problem P 6.1

```
*** Direct Form Coefficients ***
[b,a] = cas2dir(b0,B,A); b = b(1:length(b)-1), a = a(1:length(a)-1)
b =
     4     6     4     4     0    -2
a =
    1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
     4
B =
    1.0000    0.0000    1.0000
    1.0000    2.0000    1.0000
    1.0000   -0.5000         0
A =
    1.0000    0.0000    0.8100
    1.0000   -0.8000    0.6400
```

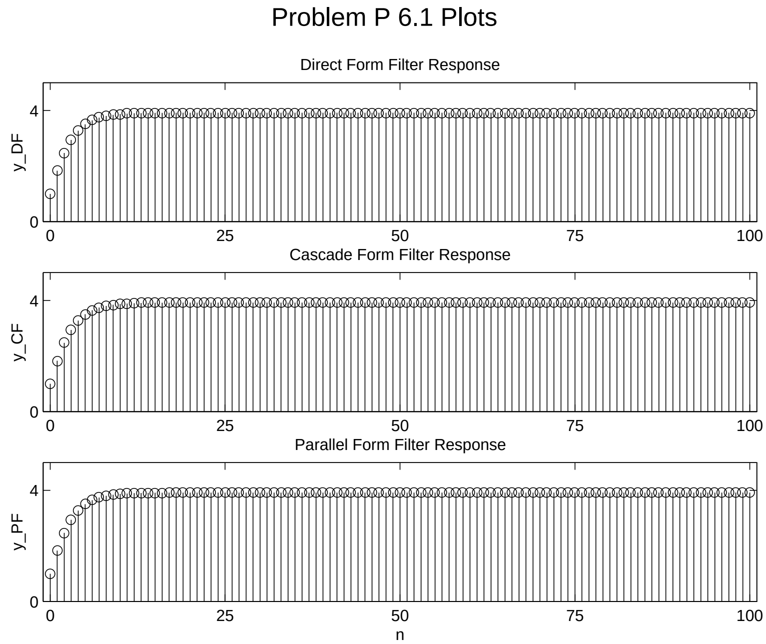


Figure 6.2: Output Responses in Problem P 6.1

```
    1.0000   -0.7500         0

%% Parallel Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel Form Coefficients ***')
*** Parallel Form Coefficients ***
[C,B,A] = dir2par(b,a)
C =
    5.1440
B =
    2.0137    0.1106
  -10.8732   15.0013
    7.7155         0
A =
    1.0000    0.0000    0.8100
    1.0000   -0.8000    0.6400
    1.0000   -0.7500         0
```

The block diagrams are shown in Figure 6.3.

3. Problem **P 6.3**

The IIR filter is decribed by the system function

$$H(z) = \left(\frac{-14.75 - 12.9z^{-1}}{1 - \frac{7}{6}z^{-1} + \frac{3}{32}z^{-2}}\right) + \left(\frac{24.5 + 26.82z^{-1}}{1 - z^{-1} + 0.5z^{-2}}\right)$$

The given structure is a parallel form. The MATLAB script to determine various realizations is:

```
clear; close all;
%% Parallel Form Coefficients %%%%%%%%%%%%%%%%%%%%%%%%%%%
```

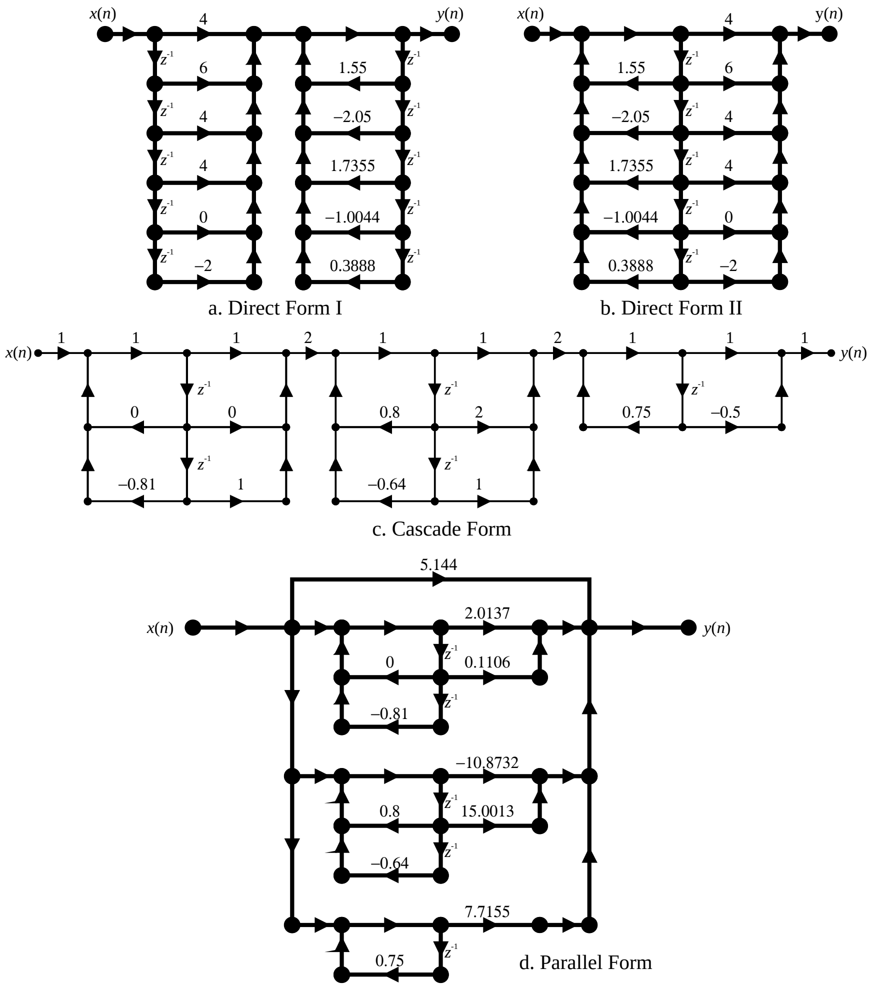


Figure 6.3: Block Diagrams in Problem P 6.2

```
C = []; B = [-14.75, -12.9; 24.5, 26.82];
A = [1, -7/8, 3/32; 1, -1, 1/2];

%% Direct Forms %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Direct Form Coefficients ***')
*** Direct Form Coefficients ***
[b,a] = par2dir(C,B,A)
b =
    9.7500    7.2325  -15.6456   -3.9356
a =
    1.0000   -1.8750    1.4688   -0.5312    0.0469

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
    9.7500
B =
```

```
    1.0000    1.8251    0.3726
    1.0000   -1.0834         0
A =
    1.0000   -1.0000    0.5000
    1.0000   -0.8750    0.0938

%% Parallel Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel Form Coefficients ***')
*** Parallel Form Coefficients ***
[C,B,A] = dir2par(b,a)
C =
     []
B =
   24.5000   26.8200
  -14.7500  -12.9000
A =
    1.0000   -1.0000    0.5000
    1.0000   -0.8750    0.0938
```

The block diagrams are shown in Figure 6.4.

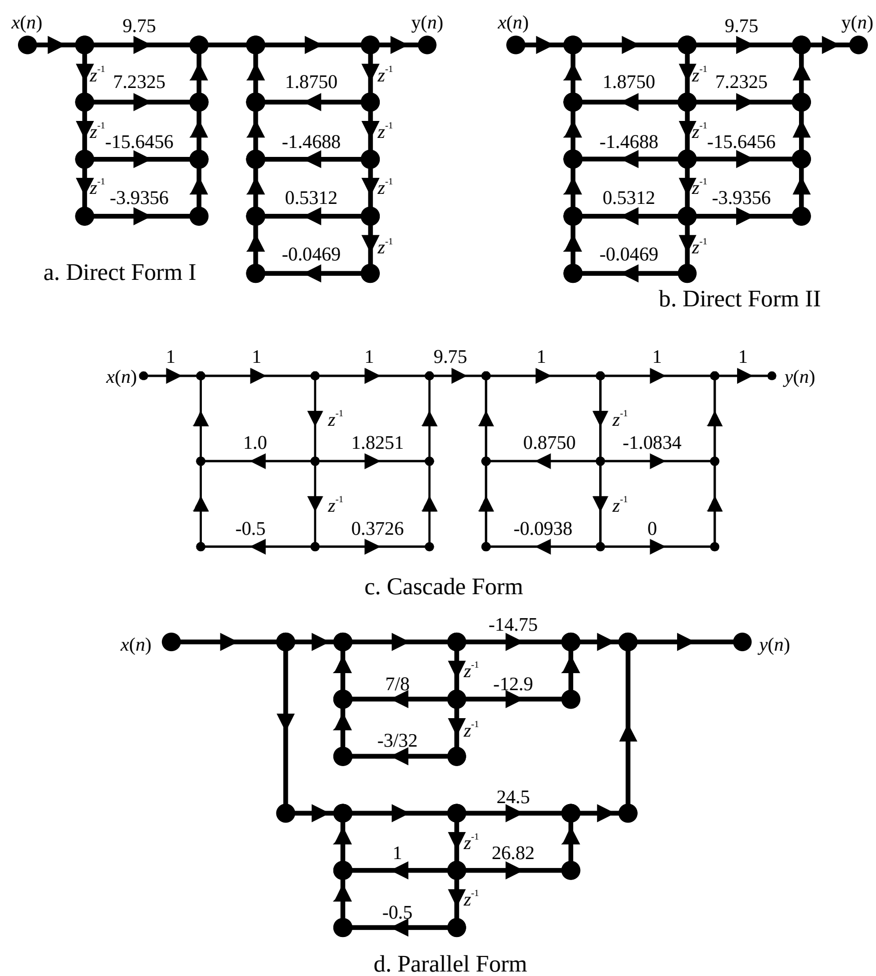


Figure 6.4: Block Diagrams in Problem P 6.3

4. Problem **P 6.5**

   A linear shift-invariant system has the system function

$$H(z) = \frac{0.5\left(1+z^{-1}\right)^{6}}{1-\frac{3}{2}z^{-1}+\frac{7}{8}z^{-2}-\frac{13}{16}z^{-3}-\frac{1}{6}z^{-4}-\frac{11}{32}z^{-5}+\frac{7}{16}z^{-6}}$$

   The given block diagram structure is a cascade form.

   (a) Conversion to the cascade form: The MATLAB script is:

```
clear; close all;
%% Direct Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b0 = 0.5; broots = -ones(1,6); b = b0*poly(broots);
a = [1,-3/2,7/8,-13/16,-1/8,-11/32,7/16];

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
    0.5000
B =
    1.0000    2.0067    1.0067
    1.0000    2.0000    1.0000
    1.0000    1.9933    0.9934
A =
    1.0000    1.0000    0.5000
    1.0000   -0.5000    1.0000
    1.0000   -2.0000    0.8750
```

   The block diagram structure is shown in Figure 6.5.

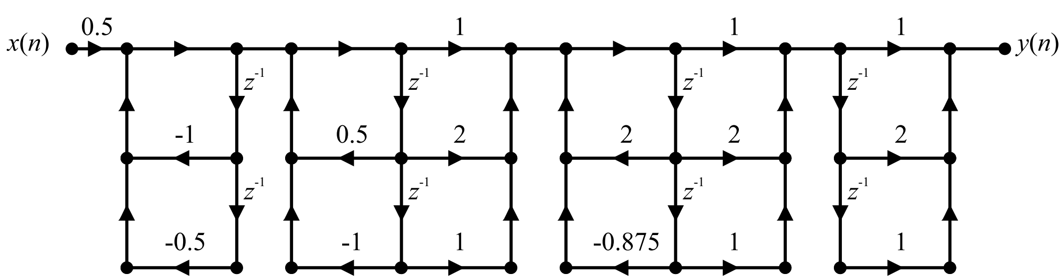


Figure 6.5: Cascade Form Structure in Problem P 6.5

   (b) The above solution is not unique because the numerator and the denominator factors can be grouped differently.

5. Problem **P 6.6**

   A linear shift-invariant system has the system function

$$H(z) = \frac{5+11.2z^{-1}+5.44z^{-2}-0.384z^{-3}-2.3552z^{-4}-1.2288z^{-5}}{1+0.8z^{-1}-0.512z^{-3}-0.4096z^{-4}}$$

   The given structure is a parallel form.

   (a) Conversion to the parallel form: The MATLAB script is

```
clear; close all;
%% Direct Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b = [5, 11.2, 5.44, -0.384, -2.3552, -1.2288];
a = [1, 0.8, 0, -0.512, -0.4096];

%% Parallel Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel Form Coefficients ***')
*** Parallel Form Coefficients ***
[C,B,A] = dir2par(b,a)
C =
    2.0000    3.0000
B =
    1.0000    2.0000
    2.0000    3.0000
A =
    1.0000    0.8000    0.6400
    1.0000    0.0000   -0.6400
```

The block diagram structure is shown in Figure 6.6.

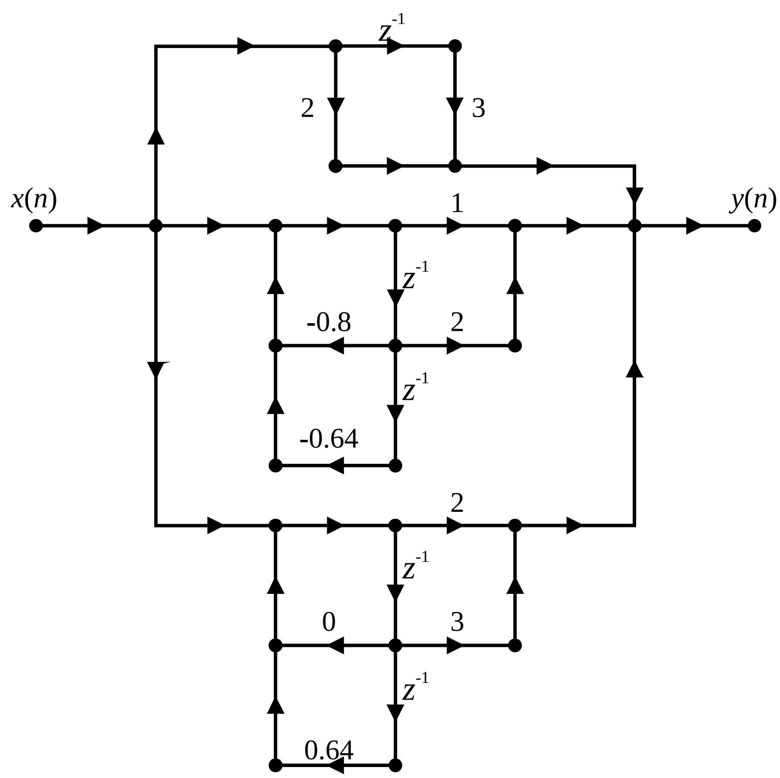


Figure 6.6: Parallel Form Structure in Problem P6.6

(b) The above solution is unique because the numerator residues and the corresponding denominator poles must be grouped into a second order section.

6. Problem **P 6.7**

A linear shift-invariant system has the system function

$$H(z) = \frac{0.5\left(1+z^{-1}\right)^6}{1-\frac{3}{2}z^{-1}+\frac{7}{8}z^{-2}-\frac{13}{16}z^{-3}-\frac{1}{8}z^{-4}-\frac{11}{32}z^{-5}+\frac{7}{16}z^{-6}}$$

The new structure is a parallel-of-cascades form.

(a) Conversion to the parallel-of-cascades form: The MATLAB script is

```
clear; close all;
%% Direct Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b = 0.5*poly([-1,-1,-1,-1,-1,-1]);
a = [1, -3/2, 7/8, -13/16, -1/8, -11/32, 7/16];

%% Parallel Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel Form Coefficients ***')
*** Parallel Form Coefficients ***
[C,B,A] = dir2par(b,a)
C =
    1.1429
B =
   -0.0210   -0.0335
    0.9252   -3.6034
   -1.5470    9.9973
A =
    1.0000    1.0000    0.5000
    1.0000   -0.5000    1.0000
    1.0000   -2.0000    0.8750

%% Parallel of Cascade Forms: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Parallel of Cascade Forms Coefficients ***')
*** Parallel of Cascade Forms Coefficients ***
B = B(1:2,:), A = A(1:2,:)
B =
   -0.0210   -0.0335
    0.9252   -3.6034
A =
    1.0000    1.0000    0.5000
    1.0000   -0.5000    1.0000
[b1,a1] = par2dir(C,B,A)
b1 =
  Columns 1 through 4
   2.0470            -2.1298            -2.0023            -0.9780
  Column 5
   0.5714 + 0.0000i
a1 =
  Columns 1 through 4
   1.0000             0.5000             1.0000             0.7500
  Column 5
   0.5000 + 0.0000i
[b0,B1,A1] = dir2cas(b1,a1)
b0 =
    2.0470
B1 =
    1.0000    0.9985    0.5050
    1.0000   -2.0389    0.5527
A1 =
    1.0000    1.0000    0.5000
    1.0000   -0.5000    1.0000
```

The block diagram structure is shown in Figure 6.7.

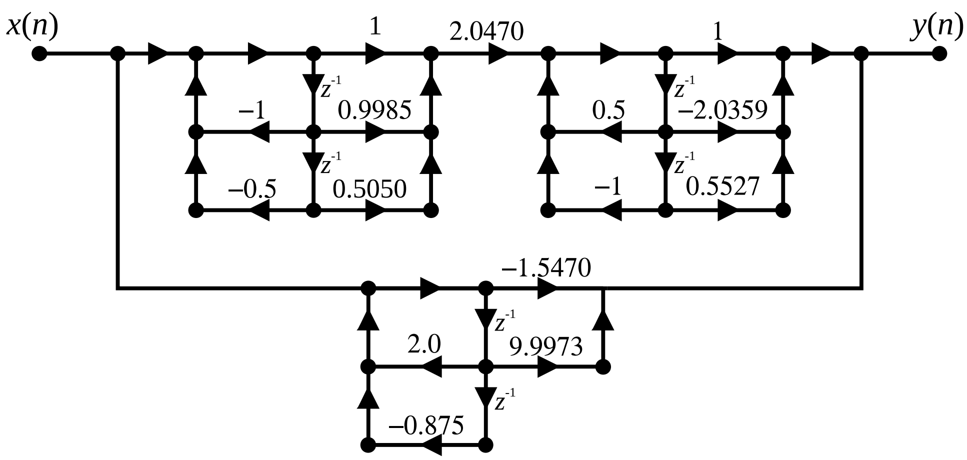


Figure 6.7: Parallel of Cascades Structure in Problem P6.6

(b) The above solution is unique because the numerator residues and the corresponding denominator poles must be grouped into a second order section.

7. Problem **P 6.8**

A causal FIR filter is described by

$$y(n) = \sum_{k=0}^{10} \left(\frac{1}{2}\right)^{|5-k|} x(n-k)$$

The MATLAB script to determine various realizations is:

```
clear; close all;
%% Direct Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Direct Form Coefficients ***')
*** Direct Form Coefficients ***
k = 0:10; b = (0.5).^(abs(5-k)), a = 1;
b =
  Columns 1 through 7
    0.0312    0.0625    0.1250    0.2500    0.5000    1.0000    0.5000
  Columns 8 through 11
    0.2500    0.1250    0.0625    0.0312

%% Linear-Phase Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Linear-Phase Form Coefficients ***')
*** Linear-Phase Form Coefficients ***
M = length(b); L = (M-1)/2;
B = b(1:1:L+1)
B =
    0.0312    0.0625    0.1250    0.2500    0.5000    1.0000

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
    0.0312
B =
```

```
    1.0000    1.7495    3.4671
    1.0000    0.5046    0.2884
    1.0000   -0.5506    0.2646
    1.0000   -2.0806    3.7789
    1.0000    2.3770    1.0000
A =
     1     0     0
     1     0     0
     1     0     0
     1     0     0
     1     0     0
```

The block diagrams are shown in Figure 6.8.

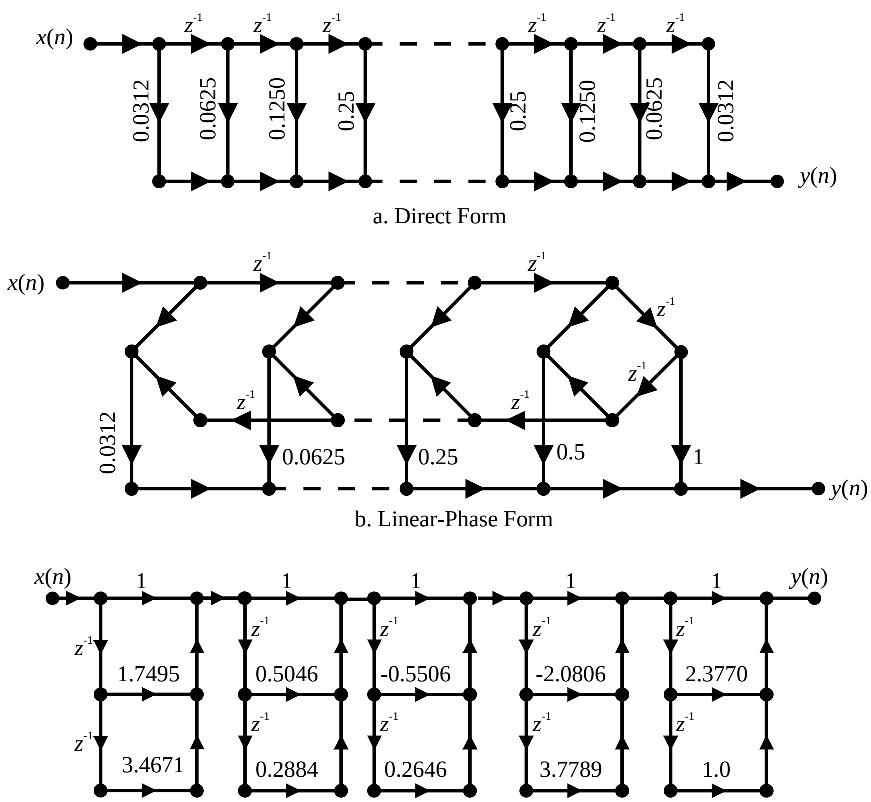


Figure 6.8: Block Diagrams in Problem P 6.8

8. Problem **P 6.9**

   The FIR filter is decribed by the system function

$$H(z) = \sum_{k=0}^{10} (2z)^{-k}$$

   The MATLAB script to determine various realizations is:

```
clear; close all;
%% Direct Form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Direct Form Coefficients ***')
```

```
*** Direct Form Coefficients ***
k = 0:10; b = (0.5).^k, a = 1;
b =
  Columns 1 through 7
    1.0000    0.5000    0.2500    0.1250    0.0625    0.0312    0.0156
  Columns 8 through 11
    0.0078    0.0039    0.0020    0.0010

%% Cascade Form: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('*** Cascade Form Coefficients ***')
*** Cascade Form Coefficients ***
[b0,B,A] = dir2cas(b,a)
b0 =
     1
B =
    1.0000    0.9595    0.2500
    1.0000    0.6549    0.2500
    1.0000    0.1423    0.2500
    1.0000   -0.4154    0.2500
    1.0000   -0.8413    0.2500
A =
     1     0     0
     1     0     0
     1     0     0
     1     0     0
     1     0     0
```

The block diagrams are shown in Figure 6.9.

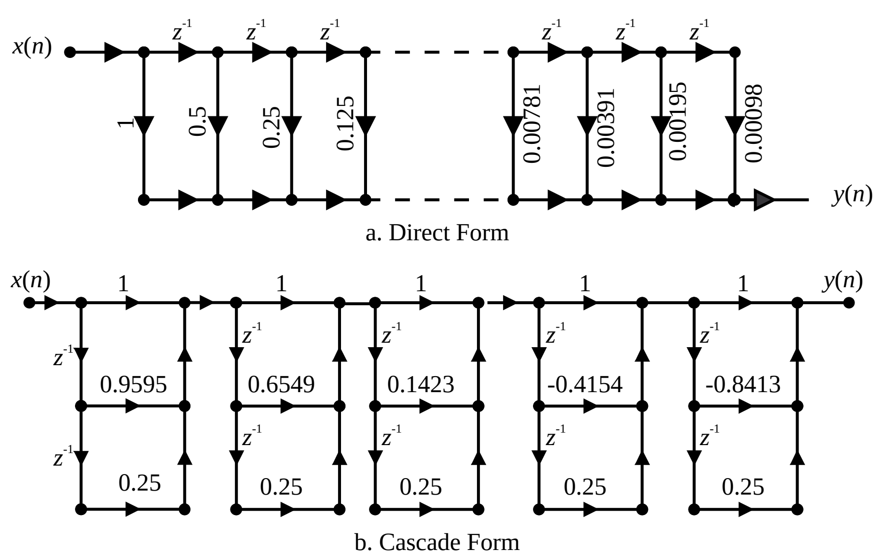


Figure 6.9: Block diagrams in Problem P 6.9

# Chapter 7

# FIR Filter Design

1. Problem **P 7.2**

   Amplitude response function for **Type-II** LP FIR filter

   (a) Type-II $\Rightarrow$ symmetric $h(n)$ and$M$–even, i.e.,

   $$h(n) = h(M-1-n),\, 0 \le n \le M-1; \quad \alpha = \frac{M-1}{2} \text{ is not an integer}$$

   Consider,

   $$H\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} h(n)\, e^{-j\omega n} = \sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega n} + \sum_{\frac{M}{2}}^{M-1} h(n)\, e^{-j\omega n}$$

   Change of variable in the second sum above:

   $$n \to M-1-n \Rightarrow \frac{M}{2} \to \frac{M}{2}-1,\; M-1 \to 0, \text{ and } h(n) \to h(n)$$

   Hence,

   $$\begin{aligned}
   H\left(e^{j\omega}\right) &= \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega n} + \sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega(M-1-n)} \\
   &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n) \left\{ e^{-j\omega n + j\omega\left(\frac{M-1}{2}\right)} + e^{-j\omega(M-1-n)+j\omega\left(\frac{M-1}{2}\right)} \right\} \\
   &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n) \left\{ e^{+j\omega\left(\frac{M-1}{2}-n\right)} + e^{-j\omega\left(\frac{M-1}{2}-n\right)} \right\} \\
   &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} 2h(n) \cos\left[\left(\frac{M-1}{2}-n\right)\omega\right]
   \end{aligned}$$

   Change of variable:

   $$\frac{M}{2}-n \to n \Rightarrow n = 0 \to n = \frac{M}{2},\; n = \frac{M}{2}-1 \to n = 1$$

   and

   $$\cos\left[\left(\frac{M-1}{2}-n\right)\omega\right] \to \cos\left[\omega\left(n-\frac{1}{2}\right)\right]$$

   Hence,

   $$H\left(e^{j\omega}\right) = e^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M}{2}} 2h\left(\frac{M}{2}-n\right) \cos\left[\omega\left(n-\frac{1}{2}\right)\right]$$

   Define $b(n) = 2h\left(\frac{M}{2}-n\right)$. Then,

   $$H\left(e^{j\omega}\right) = e^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M}{2}} b(n) \cos\left[\omega\left(n-\frac{1}{2}\right)\right] \Rightarrow H_r(\omega) = \sum_{n=1}^{\frac{M}{2}} b(n) \cos\left[\omega\left(n-\frac{1}{2}\right)\right]$$

(b) Now $\cos\left[\omega\left(n-\frac{1}{2}\right)\right]$ can be written as a linear combination of higher harmonics in $\cos\omega$ multiplied by $\cos\left(\frac{\omega}{2}\right)$, i.e.,

$$\begin{aligned}\cos\left(\frac{1\omega}{2}\right) &= \cos\frac{\omega}{2}\{\cos 0\omega\}\\ \cos\left(\frac{3\omega}{2}\right) &= \cos\frac{\omega}{2}\{2\cos\omega-1\}\\ \cos\left(\frac{5\omega}{2}\right) &= \cos\frac{\omega}{2}\{\cos 0\omega-\cos\omega+\cos 2\omega\}\end{aligned}$$

etc. Note that the lowest harmonic frequency is zero and the highest harmonic frequency is $(n-1)\omega$ in the $\cos\omega\left(n-\frac{1}{2}\right)$ expansion. Hence,

$$H_r(\omega)=\sum_{n=1}^{\frac{M}{2}} b(n)\cos\left[\omega\left(n-\frac{1}{2}\right)\right]=\cos\frac{\omega}{2}\sum_{0}^{\frac{M}{2}-1}\tilde{b}(n)\cos(\omega n)$$

where $\tilde{b}(n)$ are related to $b(n)$ through the above trignometric identities.

2. (Problem P 7.3) Type-III $\Rightarrow$ anti-symmetric $h(n)$ and $M$–odd, i.e.,

$$h(n)=-h(M-1-n),\ 0\le n\le M-1;\ h\left(\frac{M-1}{2}\right)=0;\quad \alpha=\frac{M-1}{2}\ \text{is an integer}$$

(a) Consider,

$$H\left(e^{j\omega}\right)=\sum_{n=0}^{M-1}h(n)e^{-j\omega n}=\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega n}+\sum_{\frac{M+1}{2}}^{M-1}h(n)e^{-j\omega n}$$

Change of variable in the second sum above:

$$n\to M-1-n\Rightarrow\frac{M+1}{2}\to\frac{M-3}{2},\ M-1\to 0,\ \text{and}\ h(n)\to -h(n)$$

Hence,

$$\begin{aligned}H\left(e^{j\omega}\right) &= \sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega n}-\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega(M-1-n)}\\ &= e^{-j\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M-3}{2}}h(n)\left\{e^{-j\omega n+j\omega\left(\frac{M-1}{2}\right)}-e^{-j\omega(M-1-n)+j\omega\left(\frac{M-1}{2}\right)}\right\}\\ &= e^{-j\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M-3}{2}}h(n)\left\{e^{+j\omega\left(\frac{M-1}{2}-n\right)}-e^{-j\omega\left(\frac{M-1}{2}-n\right)}\right\}\\ &= je^{-j\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M-3}{2}}2h(n)\sin\left[\left(\frac{M-1}{2}-n\right)\omega\right]\end{aligned}$$

Change of variable:

$$\frac{M-1}{2}-n\to n\Rightarrow n=0\to n=\frac{M-1}{2},\ n=\frac{M-3}{2}\to n=1$$

and

$$\sin\left[\left(\frac{M-1}{2}-n\right)\omega\right]\to\sin(\omega n)$$

Hence,

$$H\left(e^{j\omega}\right)=je^{-j\omega\left(\frac{M-1}{2}\right)}\sum_{n=1}^{\frac{M-1}{2}}2h\left(\frac{M-1}{2}-n\right)\sin(\omega n)$$

Define $c(n) = 2h\left(\frac{M-1}{2} - n\right)$. Then,

$$H\left(e^{j\omega}\right) = je^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M-1}{2}} c(n)\sin(\omega n) \Rightarrow H_r(\omega) = \sum_{n=1}^{\frac{M-1}{2}} c(n)\sin(\omega n)$$

(b) Now $\sin(\omega n)$ can be written as a linear combination of higher harmonics in $\cos\omega$ multiplied by $\sin\omega$, i.e.,

$$\begin{aligned} \sin(\omega) &= \sin\omega\{\cos 0\omega\} \\ \sin(2\omega) &= \sin\omega\{2\cos\omega\} \\ \sin(3\omega) &= \sin\omega\{\cos 0\omega + 2\cos 2\omega\} \end{aligned}$$

etc. Note that the lowest harmonic frequency is zero and the highest harmonic frequency is $(n-1)\omega$ in the $\sin\omega n$ expansion. Hence,

$$H_r(\omega) = \sum_{n=1}^{\frac{M-1}{2}} c(n)\sin(\omega n) = \sin\omega \sum_{0}^{\frac{M-3}{2}} \tilde{c}(n)\cos(\omega n)$$

where $\tilde{c}(n)$ are related to $c(n)$ through the above trignometric identities.

Problem **P 7.5**

MATLAB function for amplitude response:

```
function [Hr,w,P,L] = ampl_res(h);
%
% function [Hr,w,P,L] = ampl_res(h)
%
% Computes Amplitude response Hr(w) and its polynomial P of order L,
%   given a linear-phase FIR filter impulse response h.
% The type of filter is determined automatically by the subroutine.
%
% Hr = Amplitude Response
%  w = frequencies between [0 pi] over which Hr is computed
%  P = Polynomial coefficients
%  L = Order of P
%  h = Linear Phase filter impulse response
%

 M = length(h);
 L = floor(M/2);
if fix(abs(h(1:1:L))*10^10) ~= fix(abs(h(M:-1:M-L+1))*10^10)
error('Not a linear-phase impulse response')
end

if 2*L ~= M
if fix(h(1:1:L)*10^10) == fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-1 Linear-Phase Filter ***')
[Hr,w,P,L] = hr_type1(h);
elseif fix(h(1:1:L)*10^10) == -fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-3 Linear-Phase Filter ***')
h(L+1) = 0;
[Hr,w,P,L] = hr_type3(h);
end
else
if fix(h(1:1:L)*10^10) == fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-2 Linear-Phase Filter ***')
```

```
[Hr,w,P,L] = hr_type2(h);
elseif fix(h(1:1:L)*10^10) == -fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-4 Linear-Phase Filter ***')
[Hr,w,P,L] = hr_type4(h);
end
end
```

MATLAB verification:

```
clear; close all;
%% Matlab verification
h1 = [1 2 3 2 1]; [Hr1,w,P1,L1] = ampl_res(h1);
*** Type-1 Linear-Phase Filter ***
P1, L1,
P1 =
     3     4     2
L1 =
     2
%
h2 = [1 2 2 1]; [Hr2,w,P2,L2] = ampl_res(h2);
*** Type-2 Linear-Phase Filter ***
P2, L2,
P2 =
     4     2
L2 =
     2
%
h3 = [1 2 0 -2 -1]; [Hr3,w,P3,L3] = ampl_res(h3);
*** Type-3 Linear-Phase Filter ***
P3, L3,
P3 =
     0     4     2
L3 =
     2
%
h4 = [1 2 -2 -1]; [Hr4,w,P4,L4] = ampl_res(h4);
*** Type-4 Linear-Phase Filter ***
P4, L4,
P4 =
     4     2
L4 =
     2
%
%% Amplitude response plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.5');

subplot(2,2,1);plot(w/pi,Hr1);title('Type-1 FIR Filter');
ylabel('Hr(w)');
set(gca,'XTickMode','manual','XTick',[0:0.2:1],'fontsize',10);

subplot(2,2,2);plot(w/pi,Hr2);title('Type-2 FIR Filter');
ylabel('Hr(w)');
set(gca,'XTickMode','manual','XTick',[0:0.2:1],'fontsize',10);
```

```
subplot(2,2,3);plot(w/pi,Hr3);title('Type-3 FIR Filter');
ylabel('Hr(w)'); xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[0:0.2:1],'fontsize',10);

subplot(2,2,4);plot(w/pi,Hr4);title('Type-4 FIR Filter');
ylabel('Hr(w)'); xlabel('frequency in pi units');
set(gca,'XTickMode','manual','XTick',[0:0.2:1],'fontsize',10);

% Super Title
suptitle('Problem P7.5');
```

The plots are shown in Figure 7.1.

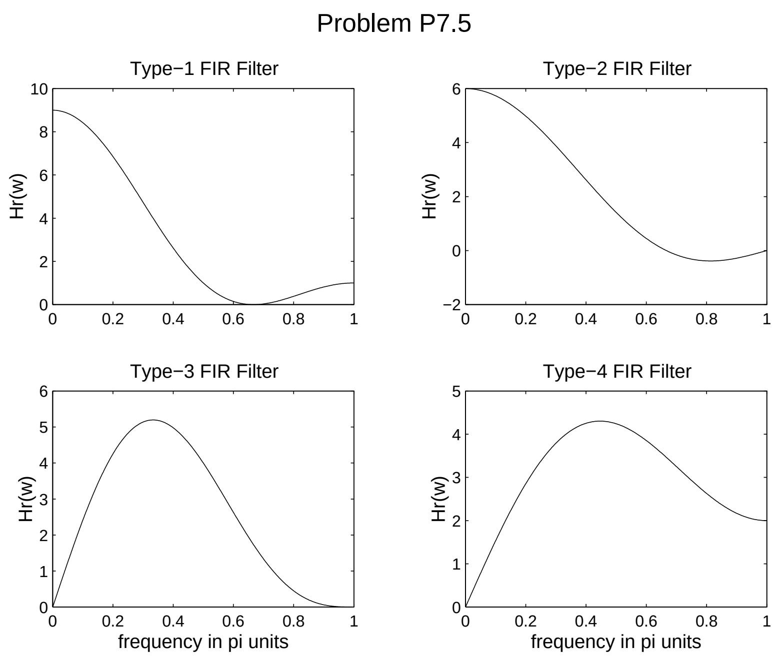


Figure 7.1: Amplitude Response Plots in Problem P 7.5

Problem **P 7.6**

The filter $H(z)$ has the following four zeros

$$z_1 = re^{j\theta}, \quad z_2 = \frac{1}{r}e^{j\theta}, \quad z_2 = re^{-j\theta}, \quad z_2 = \frac{1}{r}e^{-j\theta}$$

The system function can be written as

$$\begin{aligned}
H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right)\left(1 - z_3 z^{-1}\right)\left(1 - z_4 z^{-1}\right) \\
&= \left(1 - re^{j\theta}z^{-1}\right)\left(1 - \tfrac{1}{r}e^{j\theta}z^{-1}\right)\left(1 - re^{-j\theta}z^{-1}\right)\left(1 - \tfrac{1}{r}e^{-j\theta}z^{-1}\right) \\
&= \left\{1 - (2r\cos\theta)z^{-1} + r^2 z^{-2}\right\}\left\{1 - \left(2r^{-1}\cos\theta\right)z^{-1} + r^{-2}z^{-2}\right\} \\
&= 1 - 2\cos\theta\left(r + r^{-1}\right)z^{-1} + \left(r^2 + r^{-2} + 4\cos^2\theta\right)z^{-2} - 2\cos\theta\left(r + r^{-1}\right)z^{-3} + z^{-4}
\end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -2\cos\theta\left(r + r^{-1}\right), \left(r^2 + r^{-2} + 4\cos^2\theta\right), -2\cos\theta\left(r + r^{-1}\right), 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

Problem **P 7.7**

The bandstop filter specifications are:

| | |
|---|---|
| lower passband edge: | $0.3\pi$ |
| lower stopband edge: | $0.4\pi$ |
| upper stopband edge: | $0.6\pi$ |
| upper passband edge: | $0.7\pi$ |
| passband ripple: | 0.5 dB |
| stopband attenuation: | 40 dB |

Hanning window design using MATLAB:

```
clear; close all;
%% Specifications:
wp1 = 0.3*pi; % lower passband edge
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.5;     % passband ripple
As = 40;      % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
   delta2 = delta1; disp('Delta1 is smaller than delta2')
   [Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = min((ws1-wp1),(wp2-ws2));
M = ceil(6.2*pi/tr_width); M = 2*floor(M/2)+1, % choose odd M
M =
    63
n = 0:M-1;
w_han = (hanning(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M);
h = hd .* w_han;
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db((ws1/delta_w)+1:(ws2/delta_w)+1))),        % Actual Attn
Asd =
    43
Rpd = -min(db(1:(wp1/delta_w)+1)),              % Actual passband ripple
Rpd =
    0.0884
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.7');

subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('hd(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
```

```
subplot(2,2,2); stem(n,w_han); title('Hanning Window');
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_ham(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;1],'fontsize',10)

subplot(2,2,3); stem(n,h); title('Actual Impulse Response');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.3;0.4;0.6;0.7;1])
set(gca,'XTickLabelMode','manual','XTickLabels',[' 0 ';'0.3';'0.4';'0.6';'0.7';' 1 '],...
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-40;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 40';' 0 ']);grid

% Super Title
suptitle('Problem P7.7: Bandstop Filter');
```

The filter design plots are shown in Figure 7.2.

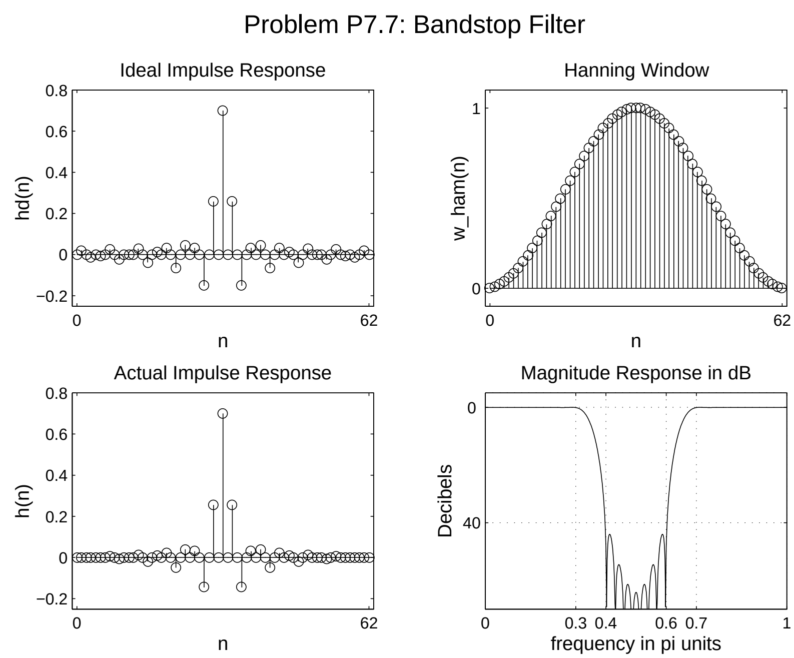


Figure 7.2: Bandstop filter design using Hanning Window in Problem P 7.7

Problem **P 7.8**

The bandpass filter specifications are:

| | |
|---|---|
| lower stopband edge: | $0.3\pi$ |
| lower passband edge: | $0.4\pi$ |
| upper passband edge: | $0.5\pi$ |
| upper stopband edge: | $0.6\pi$ |
| passband ripple: | 0.5 dB |
| stopband attenuation: | 50 dB |

Hamming window design using MATLAB:

```
clear; close all;
%% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.5*pi; % upper passband edge
ws2 = 0.6*pi; % upper stopband edge
Rp = 0.5;     % passband ripple
As = 50;      % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
   delta2 = delta1; disp('Delta1 is smaller than delta2')
   [Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = min((wp1-ws1),(ws2-wp2));
M = ceil(6.6*pi/tr_width); M = 2*floor(M/2)+1, % choose odd M
M =
    67
n = 0:M-1;
w_ham = (hamming(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
h = hd .* w_ham;
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)+1]))),          % Actual Attn
Asd =
    51
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), % Actual passband ripple
Rpd =
    0.0488
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.8');

subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response: Bandpass');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('hd(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,2); stem(n,w_ham); title('Hamming Window');
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_ham(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;1],'fontsize',10)

subplot(2,2,3); stem(n,h); title('Actual Impulse Response: Bandpass');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');
```

```
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.3;0.4;0.5;0.6;1])
set(gca,'XTickLabelMode','manual','XTickLabels',[' 0 ';'0.3';'0.4';'0.5';'0.6';' 1 '],...
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-50;0])
set(gca,'YTickLabelMode','manual','YTickLabels',['-50';' 0 ']);grid
```

The filter design plots are shown in Figure 7.3.

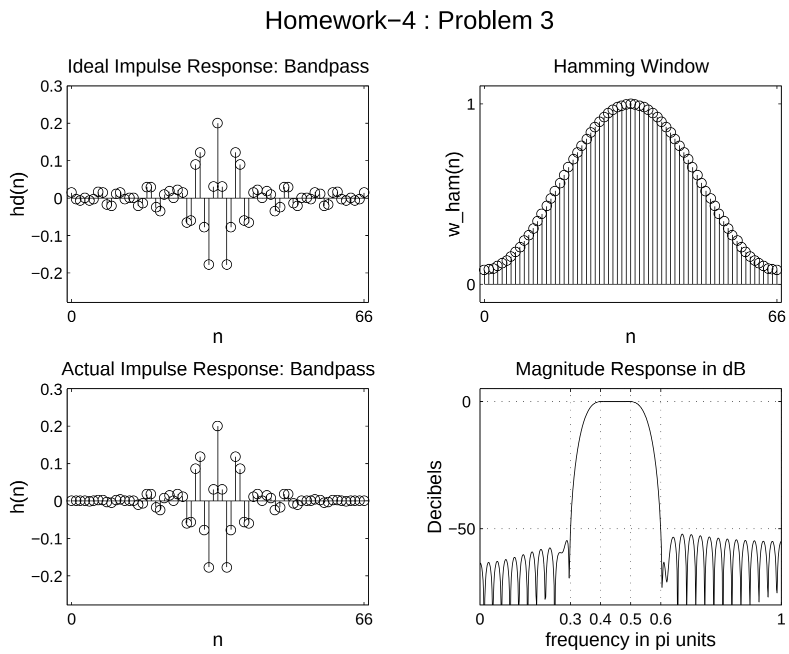


Figure 7.3: Bandpass filter design using Hamming Window in Problem P 7.8

3. (Problem P 7.9) The highpass filter specifications are:

| | |
|---|---|
| stopband edge: | $0.4\pi$ |
| passband edge: | $0.6\pi$ |
| passband ripple: | 0.5 dB |
| stopband attenuation: | 60 dB |

Kaiser window design using MATLAB:

```
clear; close all;
%% Specifications:
ws = 0.4*pi;  % stopband edge
wp = 0.6*pi;  % passband edge
Rp = 0.5;     % passband ripple
As = 60;      % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
   delta2 = delta1; disp('Delta1 is smaller than delta2')
```

```
   [Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = wp-ws; M = ceil((As-7.95)/(14.36*tr_width/(2*pi))+1)+1;
M = 2*floor(M/2)+3, % choose odd M, Increased order by 2 to get Asd>60
M =
    41
n = [0:1:M-1]; beta = 0.1102*(As-8.7);
w_kai = (kaiser(M,beta))';                          % Kaiser Window
wc = (ws+wp)/2; hd = ideal_lp(pi,M)-ideal_lp(wc,M); % Ideal HP Filter
h = hd .* w_kai;                                    % Window design
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = -floor(max(db(1:1:(ws/delta_w)+1))),          % Actual Attn
Asd =
    61
Rpd = -min(db((wp/delta_w)+1:1:501)),               % Actual passband ripple
Rpd =
    0.0148
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.9');

subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('hd(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,2); stem(n,w_kai); title('Kaiser Window');
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_kai(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;1],'fontsize',10)

subplot(2,2,3); stem(n,h); title('Actual Impulse Response');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.4;0.6;1])
set(gca,'XTickLabelMode','manual','XTickLabels',[' 0 ';'0.4';'0.6';' 1 '],...
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-60;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 60';' 0 ']);grid

% Super Title
suptitle('Problem P7.9: Highpass Filter Design');
```

The filter design plots are shown in Figure 7.4.

Problem **P 7.10**

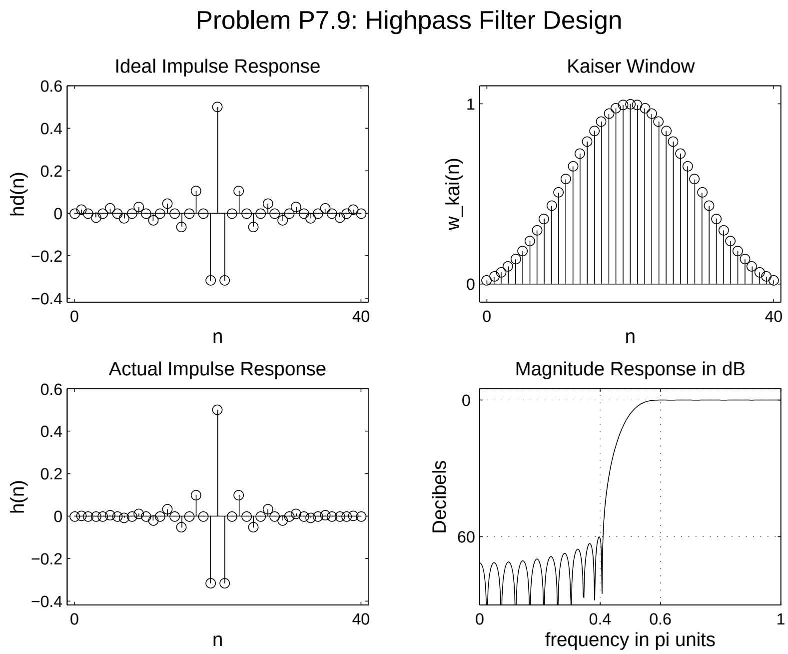


Figure 7.4: Highpass Filter design using Kaiser window in Problem P 7.9.

The bandpass filter specifications are:

| | |
|---|---|
| lower stopband edge: | $0.25\pi$ |
| lower passband edge: | $0.35\pi$ |
| upper passband edge: | $0.65\pi$ |
| upper stopband edge: | $0.75\pi$ |
| passband tolerance: | 0.05 |
| stopband tolerance: | 0.01 |

Kaiser window design using MATLAB:

```
clear; close all;
%% Specifications:
ws1 = 0.25*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.65*pi; % upper passband edge
ws2 = 0.75*pi; % upper stopband edge
delta1 = 0.05; % passband ripple magnitude
delta2 = 0.01; % stopband attenuation magnitude
%
%% Determination of Rp and As in dB
[Rp,As] = delta2db(delta1,delta2)
Rp =
    0.8693
As =
   40.4238
%
%% Determination of Window Parameters
tr_width = min((wp1-ws1),(ws2-wp2));
M = ceil((As-7.95)/(14.36*tr_width/(2*pi))+1)+1;
```

```
M = 2*floor(M/2)+1, % Odd filter length
M =
    49
n=[0:1:M-1];
if As >= 50
        beta = 0.1102*(As-8.7);
elseif (As < 50) & (As > 21)
        beta = 0.5842*(As-21)^(0.4) + 0.07886*(As-21);
else
        error('As must be greater than 21')
end
w_kai = (kaiser(M,beta))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
h = hd .* w_kai;
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)+1]))),          % Actual Attn
Asd =
    42
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), % Actual passband ripple
Rpd =
    0.1097
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.10');

subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response: Bandpass');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('hd(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,2); stem(n,w_kai); title('Kaiser Window');
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_ham(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;1],'fontsize',10)

subplot(2,2,3); stem(n,h); title('Actual Impulse Response: Bandpass');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.25;0.35;0.65;0.75;1])
set(gca,'XTickLabelMode','manual','XTickLabels',...
['  0 ';'0.25';'0.35';'0.65';'0.75';'  1 '], 'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-40;0])
set(gca,'YTickLabelMode','manual','YTickLabels',['-40';' 0 ']);grid
```

The filter design plots are shown in Figure 7.5.

4. (Problem P 7.11b) The `kai_hpf` function:

```
function [h,M] = kai_hpf(ws,wp,As);
% [h,M] = kai_hpf(ws,wp,As);
```

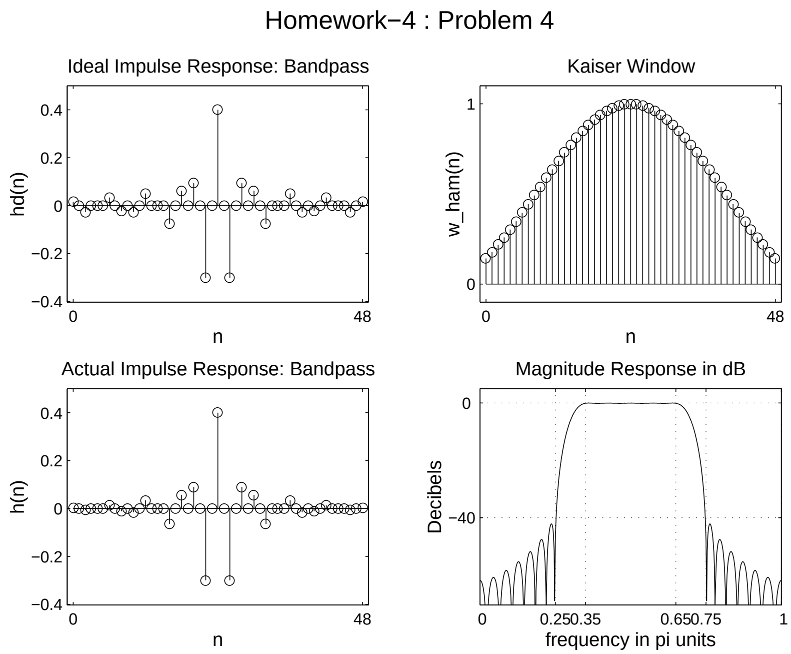


Figure 7.5: Bandpass filter design using Kaiser Window in Problem P 7.10

```
% HighPass FIR filter design using Kaiser window
%
%  h = Impulse response of the designed filter
%  M = Length of h which is an odd number
% ws = Stopband edge in radians (0 < wp < ws < pi)
% wp = Passband edge in radians (0 < wp < ws < pi)
% As = Stopband attenuation in dB (As > 0)

if ws <= 0
error('Stopband edge must be larger than 0')
end
if wp >= pi
error('Passband edge must be smaller than pi')
end
if wp <= ws
error('Passband edge must be larger than Stopband edge')
end

% Select the min(delta1,delta2) since delta1=delta2 in windodow design
tr_width = wp-ws; M = ceil((As-7.95)/(14.36*tr_width/(2*pi))+1)+1;
M = 2*floor(M/2)+1;                                    % choose odd M
if M > 255
error('M is larger than 255')
end
n = [0:1:M-1]; beta = 0.1102*(As-8.7);
w_kai = (kaiser(M,beta))';                             % Kaiser Window
wc = (ws+wp)/2; hd = ideal_lp(pi,M)-ideal_lp(wc,M); % Ideal HP Filter
h = hd .* w_kai;                                       % Window design
```

MATLAB verification:

```
clear; close all;
%% Specifications:
ws = 0.4*pi;  % stopband edge
wp = 0.6*pi;  % passband edge
As = 60;      % stopband attenuation
%
[h,M] = kai_hpf(ws,wp,As); n = 0:M-1;
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = -floor(max(db(1:1:(ws/delta_w)+1))),          % Actual Attn
Asd =
    60
Rpd = -min(db((wp/delta_w)+1:1:501)),               % Actual passband ripple
Rpd =
    0.0147
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.11b');

subplot(2,1,1); stem(n,h); title('Actual Impulse Response');
axis([-1,M,min(h)-0.1,max(h)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,1,2); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.4;0.6;1])
set(gca,'XTickLabelMode','manual','XTickLabels',[' 0 ';'0.4';'0.6';' 1 '],...
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-60;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 60';' 0 ']);grid

% Super Title
suptitle('Problem P7.11b: Highpass Filter Design');
```

The filter design plots are shown in Figure 7.6.

Problem **P 7.12**

The staircase filter specifications are:

$$\begin{array}{lrcccllcllcl}
\text{Band-1:} & 0 & \leq & \omega & \leq & 0.3\pi & \text{Ideal Gain} & = & 1 & \delta_1 & = & 0.010 \\
\text{Band-2:} & 0.4\pi & \leq & \omega & \leq & 0.7\pi & \text{Ideal Gain} & = & 0.5 & \delta_2 & = & 0.005 \\
\text{Band-3:} & 0.8\pi & \leq & \omega & \leq & \pi & \text{Ideal Gain} & = & 0 & \delta_3 & = & 0.001
\end{array}$$

Blackman window design using MATLAB:

```
clear; close all;
%% Specifications:
w1L = 0.0*pi; w1U = 0.3*pi; delta1 = 0.010; % Band-1 Specs
w2L = 0.4*pi; w2U = 0.7*pi; delta2 = 0.005; % Band-2 Specs
w3L = 0.8*pi; w3U = 1.0*pi; delta3 = 0.001; % Band-3 Specs
%
%% Determination of Rp and As in dB
```

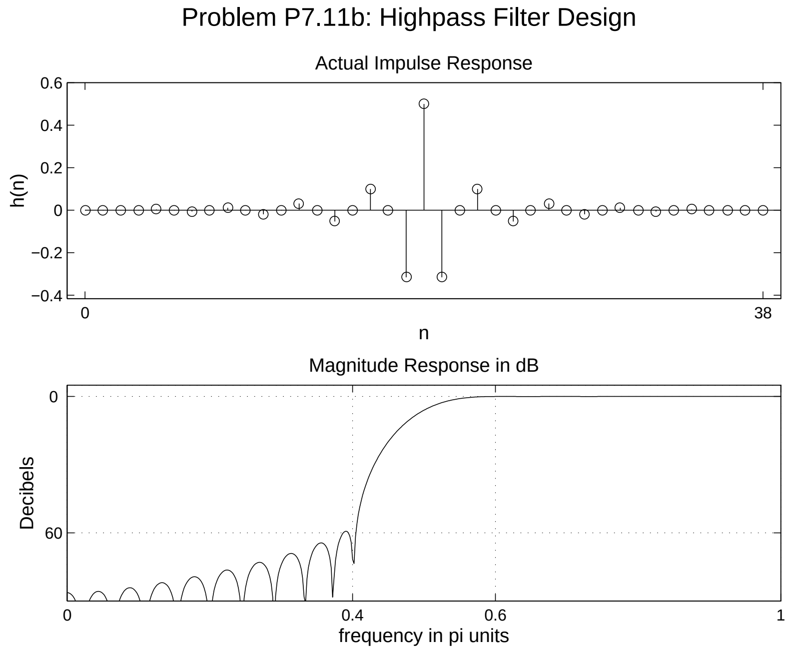


Figure 7.6: Highpass filter design using the \texttt {kai_hpf} function in Problem 7.11b.

```
As = -20*log10(delta3)
As =
   60.0000
%
%% Determination of Window Parameters
tr_width = min((w2L-w1U),(w3L-w2U));
M = ceil(11*pi/tr_width); M = 2*floor(M/2)+1, % choose odd M
M =
   111
n=[0:1:M-1];
w_bla = (blackman(M))';
wc1 = (w1U+w2L)/2; wc2 = (w2U+w3L)/2;
hd = 0.5*ideal_lp(wc1,M) + 0.5*ideal_lp(wc2,M);
h = hd .* w_bla;
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([floor(w3L/delta_w)+1:501]))),          % Actual Attn
Asd =
    79
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.12');

subplot(2,2,1); stem(n,w_bla); title('Blackman Window');
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_ham(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;1],'fontsize',10)
```

```
subplot(2,2,2); stem(n,h); title('Actual Impulse Response');
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;M-1],'fontsize',10)

subplot(2,2,3); plot(w/pi,mag); title('Magnitude Response');
axis([0,1,0,1]); xlabel('frequency in pi units'); ylabel('|H|')
set(gca,'XTickMode','manual','XTick',[0;0.3;0.4;0.7;0.8;1])
set(gca,'XTickLabelMode','manual','XTickLabels',['  0';'0.3';'0.4';'0.7';'0.8';'  1'],
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[0;0.5;1])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 0 ';'0.5';' 1 ']);grid

subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',[0;0.3;0.4;0.7;0.8;1])
set(gca,'XTickLabelMode','manual','XTickLabels',['  0';'0.3';'0.4';'0.7';'0.8';'  1'],
'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-60;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 60';' 0 ']);grid

% Super Title
suptitle('Problem P7.12: Staircase Filter');
```

The filter design plots are shown in Figure 7.7.

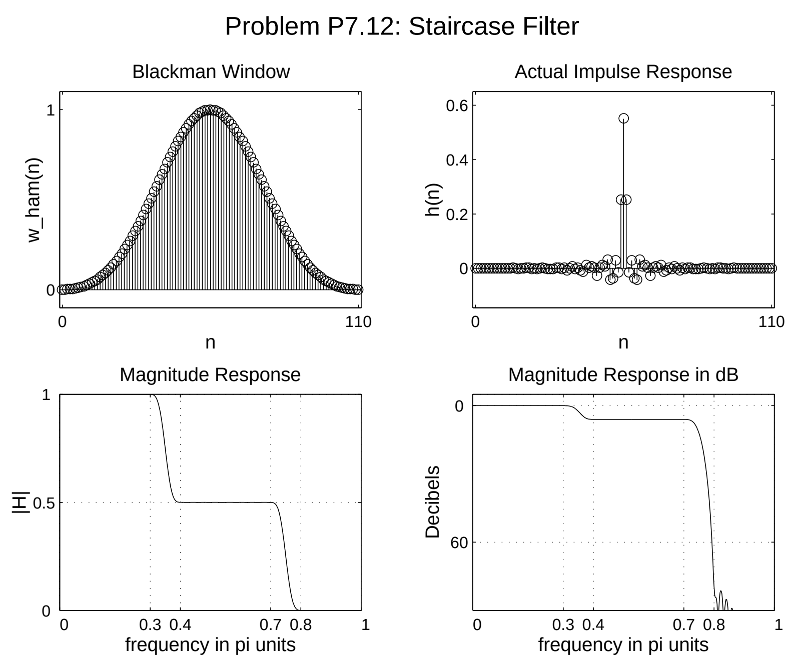


Figure 7.7: Staircase filter design using Blackman Window in Problem P 7.12

Problem **P 7.18**

Bandpass filter design using Parks-McClellan Algorithm

(a) Since we are given $M = 25$, $A_s = 50$ dB, and the ideal cutoff frequencies, $\omega_{c_1} = \pi/3$ and $\omega_{c_2} = 2\pi/3$, we will have to determine the band edge frequencies, $\omega_{s_1}$, $\omega_{p_1}$, $\omega_{p_2}$, and $\omega_{S_2}$, to implement the Parks-McClellan algorithm. Using the formula due to Kaiser we can determine the transition width as follows

$$M - 1 \simeq \frac{A_s - 13}{14.36\,(\Delta\omega/2\pi)} \Rightarrow \Delta\omega \simeq 2\pi \frac{A_s - 13}{14.36\,(M-1)}$$

Since no additional information is given, we will assume that the transition bandwidths are equal and that the tolerances are also equal in each band, i.e.,

$$\omega_{s_1} = \omega_{c_1} - \frac{\Delta\omega}{2},\ \omega_{p_1} = \omega_{c_1} + \frac{\Delta\omega}{2},\ \omega_{p_2} = \omega_{c_2} - \frac{\Delta\omega}{2},\ \omega_{s_2} = \omega_{c_2} + \frac{\Delta\omega}{2}$$

and

$$\delta_1 = \delta_2 = \delta_3$$

Using this we will run the `remez` algorithm and check for the stopband attenuation of 50 dB. If the actual attenuation is less than (more than) 50 then we will increase (decrease) $\Delta\omega$ until the attenuation condition is met. In the following MATLAB script, the conditionw were met at the above computed value.

```
clear; close all;
%% Specifications
wc1 = pi/3;   % lower cutoff frequency
wc2 = 2*pi/3; % upper cuoff frequency
As = 50;      % stopband attenuation
M = 25;       % filter length
%
% (a) Design
tr_width = 2*pi*(As-13)/(14.36*(M-1)), % transition width in radians
tr_width =
    0.6746
ws1 = wc1-tr_width/2; wp1 = wc1+tr_width/2;
wp2 = wc2-tr_width/2; ws2 = wc2+tr_width/2;
f = [0,ws1/pi,wp1/pi,wp2/pi,ws2/pi,1];
m = [0,0,1,1,0,0];
n = 0:M-1;
h = remez(M-1,f,m);
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)+1]))), % Actual Attn
Asd =
    50
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), % Actial ripple
Rpd =
    0.0518
%
%% Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.18');

subplot(2,1,1); stem(n,h); title('Impulse Response: Bandpass');
axis([-1,M,min(h)-0.1,max(h)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;12;24],'fontsize',10)

subplot(2,1,2); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-80,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',f,'fontsize',10)
```

```
set(gca,'YTickMode','manual','YTick',[-50;0])
set(gca,'YTickLabelMode','manual','YTickLabels',['-50';' 0 '],'fontsize',10);grid
```

The impulse response plot is shown in Figure 7.8.

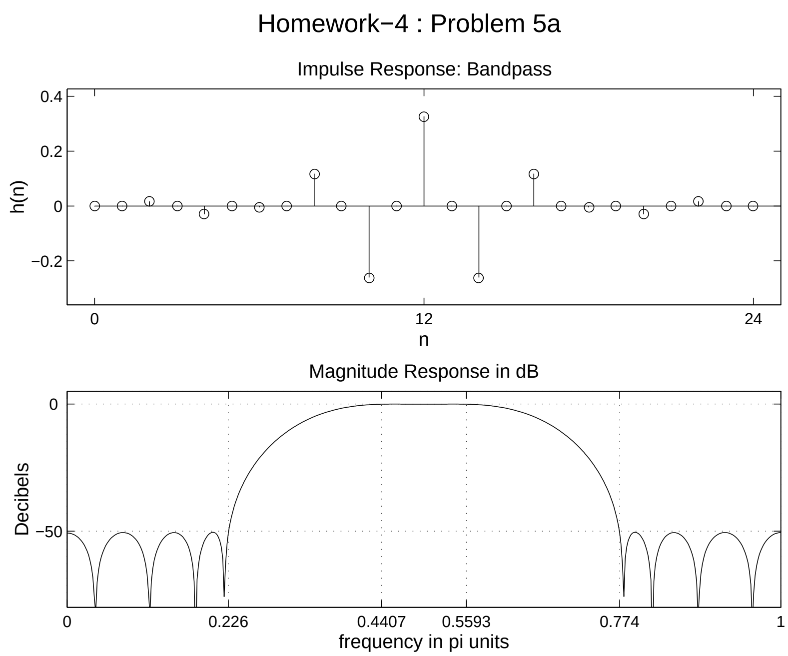


Figure 7.8: Impulse Response Plot in Problem P 7.18

(b) Amplitude Response:

```
% (b) Amplitude Response Plot
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P7.18b');

[Hr,w,a,L] = Hr_type1(h);
plot(w/pi,Hr); title('Amplitude Response in Problem 5b')
xlabel('frequency in pi units'); ylabel('Hr(w)')
axis([0,1,-0.1,1.1])
set(gca,'XTickMode','manual','XTick',f)
```

The amplitude response plot is shown in Figure 7.9.

Problem **P 7.19**

Bandstop filter design of Problem P7.7 using Parks-McClellan Algorithm

(a) MATLAB script:

```
clear; close all;
%% Specifications
wp1 = 0.3*pi; % lower passband edge
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.5;     % passband ripple
```

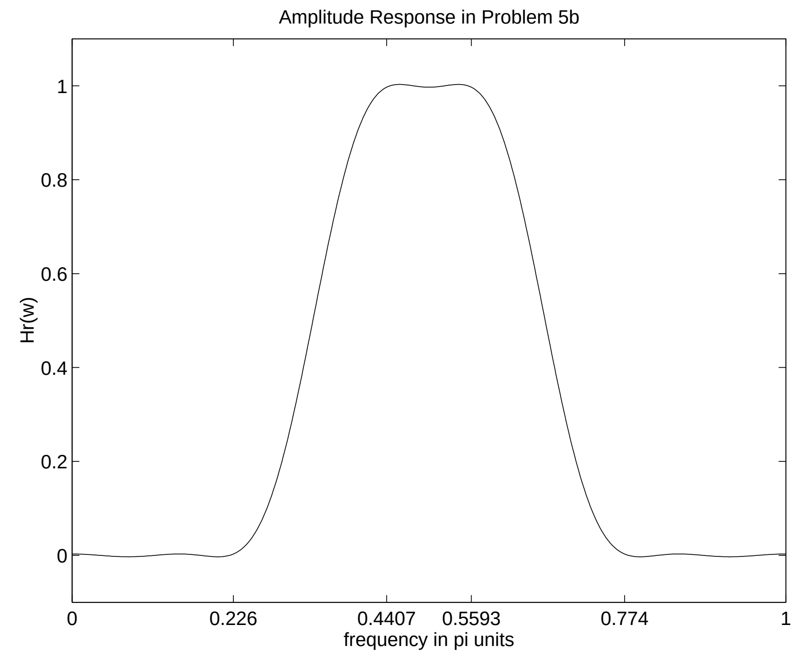


Figure 7.9: Amplitude Response Plot in Problem P 7.18

```
As = 40;      % stopband attenuation
%
% (a) Design
delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);
delta2 = (1+delta1)*(10^(-As/20));
weights = [delta2/delta1, 1, delta2/delta1];
delta_f =min((wp2-ws2)/(2*pi), (ws1-wp1)/(2*pi));
M = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*delta_f)+1);
M = 2*floor(M/2)+1
M =
    33
f = [0, wp1/pi, ws1/pi, ws2/pi, wp2/pi, 1];
m = [1 1 0 0 1 1];
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,[1]);
delta_w = pi/500;
Asd = floor(-max(db([floor(ws1/delta_w)+1:floor(ws2/delta_w)]))), % Actual Attn
Asd =
    40
M = M+2
M =
    35
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,[1]);
delta_w = pi/500;
Asd = floor(-max(db([floor(ws1/delta_w)+1:floor(ws2/delta_w)]))), % Actual Attn
Asd =
    40
```

```
n = 0:M-1;
%
%%Filter Response Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.19a');

subplot(2,1,1); stem(n,h); title('Impulse Response: Bandpass');
axis([-1,M,min(h)-0.1,max(h)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;17;34],'fontsize',10)

subplot(2,1,2); plot(w/pi,db); title('Magnitude Response in dB');
axis([0,1,-60,5]); xlabel('frequency in pi units'); ylabel('Decibels')
set(gca,'XTickMode','manual','XTick',f,'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-40;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 40';' 0 '],'fontsize',10);grid

% Super Title
suptitle('Problem P7.19a: Bandstop Filter');
```

The filter response plots are shown in Figure 7.10.

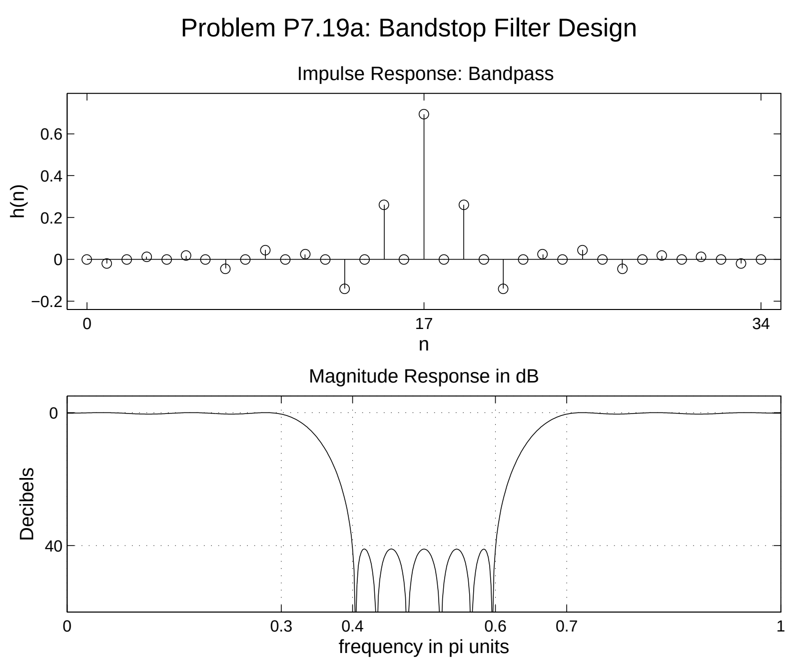


Figure 7.10: Bandstop Filter Response Plots in Problem P7.19a

(b) Amplitude Response:

```
% (b) Amplitude Response Plot
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P7.19b');

[Hr,w,a,L] = Hr_type1(h);
plot(w/pi,Hr); title('Amplitude Response in Problem P7.19b')
xlabel('frequency in pi units'); ylabel('Hr(w)')
```

```
axis([0,1,-0.1,1.1])
set(gca,'XTickMode','manual','XTick',f)
```

The amplitude response plot is shown in Figure 7.11.

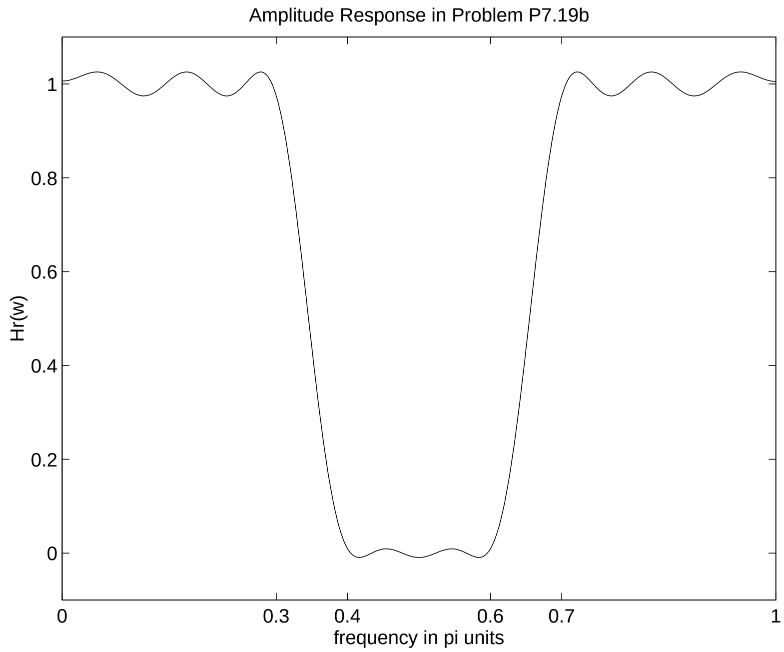


Figure 7.11: Amplitude Response Plot in Problem P 7.19b

Problem **P 7.21** Multiband filter design using Parks-McClellan Algorithm

The filter specifications are:

| | **Band-1** | **Band-2** | **Band-3** |
|---|---|---|---|
| Lower band-edge | 0 | 0.5 | 0.8 |
| Upper band-edge | 0.4 | 0.7 | 1 |
| Ideal value | 0.4 | 0.95 | 0.025 |
| Tolerance | 0.05 | 0.05 | 0.025 |

MATLAB Script:

```
clear; close all;
%% Specifications
f = [0,0.4,0.5,0.7,0.8,1];
m = [0.4,0.4,0.95,0.95,0.025,0.025];
delta1 = 0.05; delta2 = 0.05; delta3 = 0.025;
weights = [delta3/delta2, delta3/delta2, delta3/delta3];
As = -20*log10(0.05)
As =
   26.0206
%
%% Design
delta_f = 0.05; % Transition width in cycles per sample
M = ceil((-20*log10(sqrt(delta2*delta3))-13)/(14.6*delta_f)+1)
M =
```

```
    23
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual Attn
Asd =
    24
M = M+1
M =
    24
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual Attn
Asd =
    25
M = M+1
M =
    25
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual Attn
Asd =
    25
M = M+1
M =
    26
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual Attn
Asd =
    25
M = M+1
M =
    27
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual Attn
Asd =
    26
n = 0:M-1;
%
%% Impulse Response Plot
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.21a');

stem(n,h); title('Impulse Response Plot in Problem 6');
axis([-1,M,min(h)-0.1,max(h)+0.1]); xlabel('n'); ylabel('h(n)')
set(gca,'XTickMode','manual','XTick',[0;13;26],'fontsize',12)
%
%% Amplitude Response Plot
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P7.21b');

[Hr,w,a,L] = Hr_type1(h);
```

```
plot(w/pi,Hr); title('Amplitude Response in Problem 6')
xlabel('frequency in pi units'); ylabel('Hr(w)')
axis([0,1,0,1])
set(gca,'XTickMode','manual','XTick',f)
set(gca,'YTickMode','manual','YTick',[0;0.05;0.35;0.45;0.9;1]); grid
```

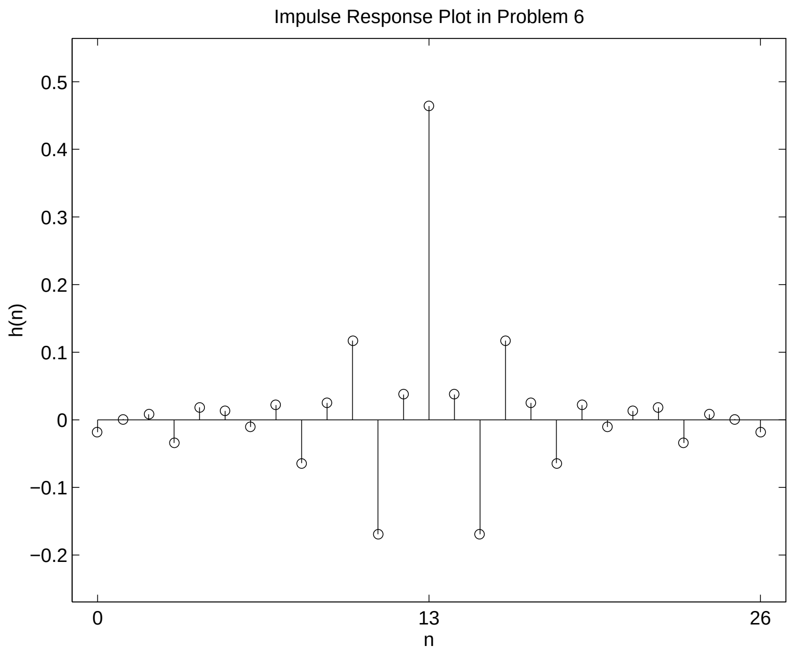


Figure 7.12: Impulse Response Plot in Problem P 7.21

The impulse response is shown in Figure 7.12 and the amplitude response plot is shown in Figure 7.13.

5. (Problem P 7.20) Design of a 25-tap differentiator with unit slope using Parks-McClellan algorithm

   (a) MATLAB script:

```
clear; close all;
% Specifications
M = 25; w1 = 0.1*pi; w2 = 0.9*pi; % slope = 1 sam/cycle
%
% (a) Design
f = [w1/pi w2/pi]; m = [w1/(2*pi) w2/(2*pi)];
h = remez(M-1,f,m,'differentiator');
[db,mag,pha,grd,w] = freqz_m(h,1);
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.20a');
subplot(2,1,1);stem([0:1:M-1],h);title('Impulse Response'); axis([-1,25,-0.2,.2]);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
set(gca,'XTickMode','manual','XTick',[0;12;24],'fontsize',10);
subplot(2,1,2);plot(w/(2*pi),mag);title('Magnitude Response');grid;
axis([0,0.5,0,0.5]);
xlabel('Normalized frequency in cycles/sam','fontsize',10)
set(gca,'XTickMode','manual','XTick',[0;w1/(2*pi);w2/(2*pi);0.5],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[0;0.05;0.45;0.5],'fontsize',10);
```

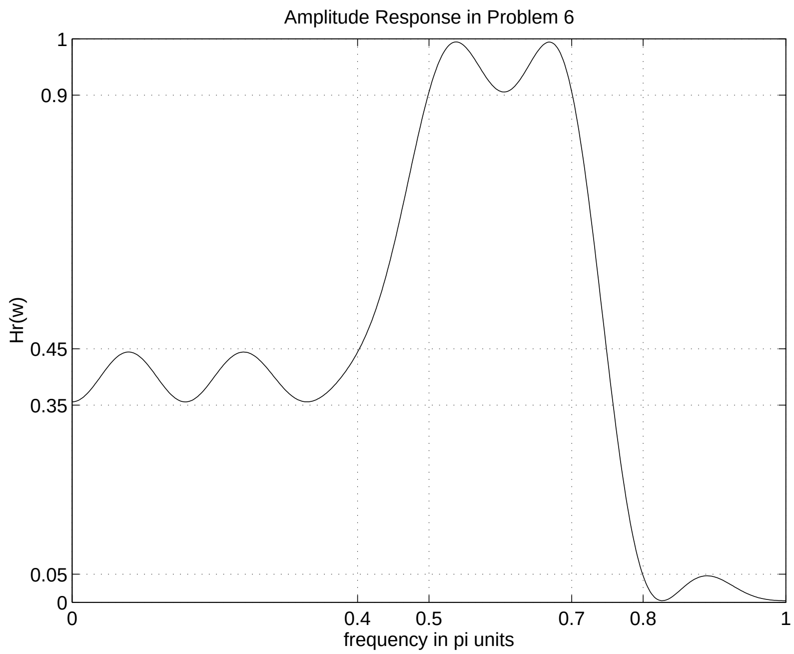


Figure 7.13: Amplitude Response Plot in Problem P 7.21

The differentiator design plots are shown in Figure 7.14 from which we observe that this filter provides a slope equal to 1 sam/cycle or $\pi/2$ sam/rad.

(b) Output when $x(n) = 3\sin(0.25\pi n)$, $0 \le n \le 100$: First determine the sign of $h(n)$ and then appropriately convolve with $x(n)$.

```
% (b) Differentiator verification
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P7.20b');
[Hr,w,P,L] = ampl_res(h); subplot; plot(w/(2*pi), Hr);
*** Type-3 Linear-Phase Filter ***
title('Amplitude Response'); grid; axis([0,0.5,-0.5,0]);
set(gca,'XTickMode','manual','XTick',[0;w1/(2*pi);w2/(2*pi);0.5],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[-0.5;-0.45;-0.05;0],'fontsize',10);
```

The amplitude response plot is shown in Figure 7.15. The sign of $h(n)$ from Figure 7.15 is negative, hence we will convolve $x(n)$ with $-h(n)$ and then compare the input and output in the steady-state (i.e. when $n > 25$) with output shifted by 12 to the left to account for the phase delay of the differentiator.

```
Hf_3 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_3,'NumberTitle','off','Name','P7.20c');
n=[0:1:100]; x = 3*sin(0.25*pi*n); y = conv(x,-h);
m = [41:1:81];
plot(m,x(41:1:81),m,y(41+12:1:81+12));grid % add 12 sample delay to y
xlabel('n'); title('Input-Output Sequences'); axis([40,82,-4,4]);
set(gca,'XTickMode','manual','XTick',[41;81],'fontsize',10);
set(gca,'YTickMode','manual','YTick',[-3;0;3],'fontsize',10);
```

The input-output plots are shown in Figure 7.16. Since the slope is $\pi/2$ sam/rad, the gain at $\omega = 0.25\pi$ is equal to 0.125. Therefore, the output (when properly shifted) should be

$$y(n) = 3(0.125)\cos(0.25\pi n) = 0.375\cos(0.25\pi n)$$

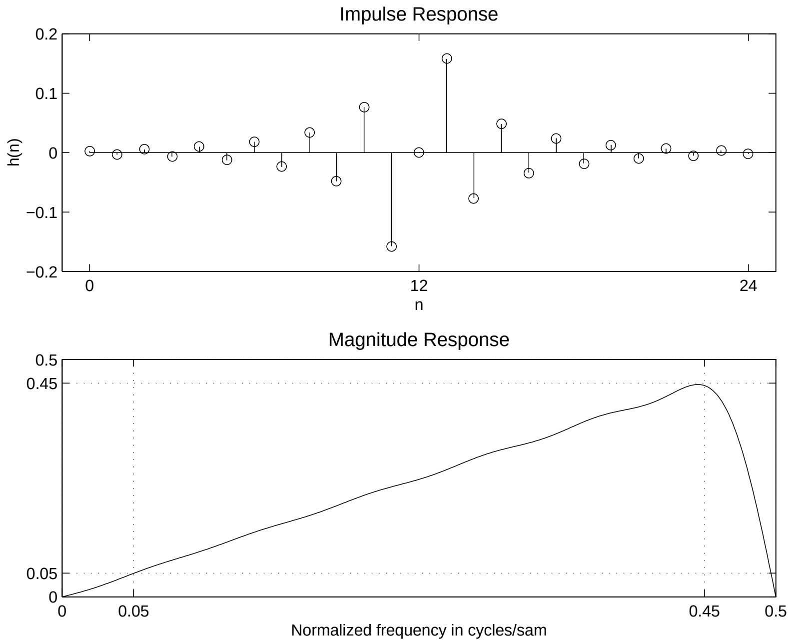


Figure 7.14: Digital differentiator design using PM algorithm in Problem P 7.20a.

From the above figure we can verify that $y(n)$(the lower curve) is indeed a cosine waveform with amplitude $\approx 0.4$.

Problem **P 7.22**

Narrow bandpass filter design using Parks-McClellan Algorithm

We want to design a 50th-order narrowband bandpass filter to filter out noise component with center frequency of $\omega_c = \pi/2$, bandwidth of $0.02\pi$, and stopband attenuation of 30 dB.

(a) In this design we already know the order of the filter. The only parameters that we don't know are the stopband cutoff frequencies $\omega_{s_1}$ and $\omega_{s_2}$. Let the transition bandwidth be $\Delta\omega$ and let the passband be symmetrical with respect to the center frequency $\omega_c$. Then

$$\omega_{p_1} = \omega_c - 0.01\pi,\ \omega_{p_2} = \omega_c + 0.01\pi,\ \omega_{s_1} = \omega_{p_1} - \Delta\omega, \text{ and } \omega_{s_2} = \omega_{p_2} + \Delta\omega$$

We will aslo assume that the tolerances in each band are equal. Now we will begin with initial value for $\Delta\omega = 0.2\pi$ and run the remez algorithm to obtain the actual stopband attenuation. If it is smaller (larger) than the given 30 dB then we will increase (decrease) $\Delta\omega$ then iterate to obtain the desired solution. The desired solution was found for $\Delta\omega = 0.5\pi$. MATLAB Script:

```
clear; close all;
%% Specifications
         N = 50;                                     % Order of the filter
        w0 = 0.5*pi;                                 % Center frequency
 Bandwidth = 0.02*pi;                                % Bandwidth
%
%   Deltaw = Transition bandwidth (iteration variable)
%
wp1 = w0-Bandwidth/2; wp2 = w0+Bandwidth/2;

% (a) Design
```

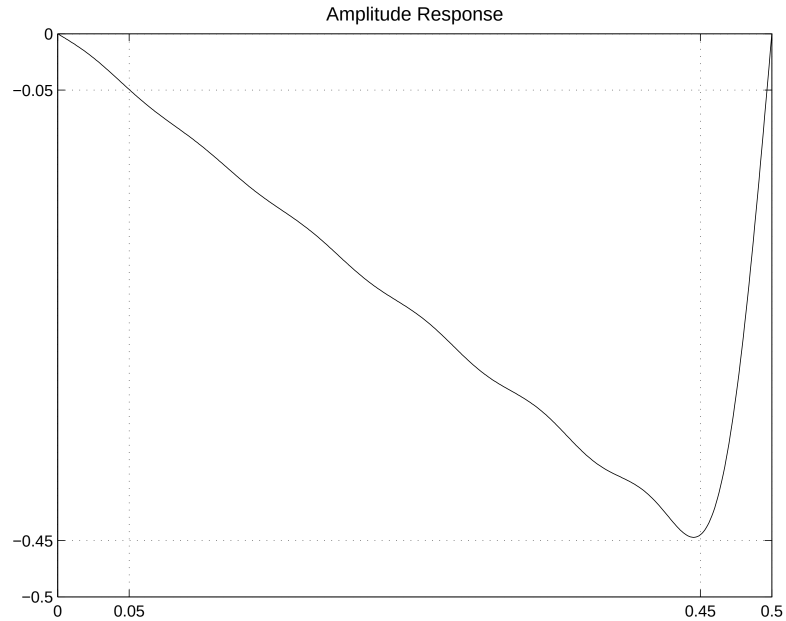


Figure 7.15: Amplitude response in Problem P 7.20b.

```
Deltaw = 0.02*pi;                                % Initial guess
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
m=[0,0,1,1,0,0];
h=remez(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual Attn
Asd =
    13

% Next iteration
Deltaw = Deltaw+0.01*pi;
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
h=remez(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual Attn
Asd =
    20

% Next iteration
Deltaw = Deltaw+0.01*pi;
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
h=remez(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
```

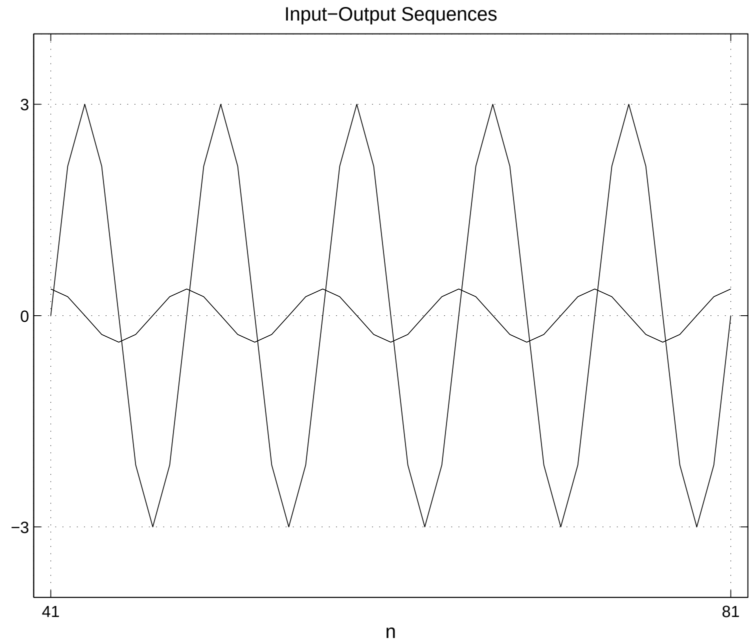


Figure 7.16: Digital differentiator operation in Problem P 7.20b.

```
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual Attn
Asd =
    26

% Next iteration
Deltaw = Deltaw+0.01*pi;
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
h=remez(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual Attn
Asd =
    30

Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P7.22a');
plot(w/pi,db); axis([0,1,-50,0]); title('Log-Magnitude Response in P7.22a');
xlabel('frequency in pi units'); ylabel('DECIBELS')
set(gca,'XTickMode','manual','XTick',[0;ws1/pi;ws2/pi;1],'fontsize',10)
set(gca,'YTickMode','manual','YTick',[-30;0])
set(gca,'YTickLabelMode','manual','YTickLabels',[' 30';' 0 '],'fontsize',10);grid
```

The log-magnitude response is shown in Figure 7.17.

(b) The time-domain response of the filter. MATLAB script:

```
% (b) Time-domain Response
n = [0:1:200]; x = 2*cos(pi*n/2)+randn(1,201); y = filter(h,1,x);
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
```

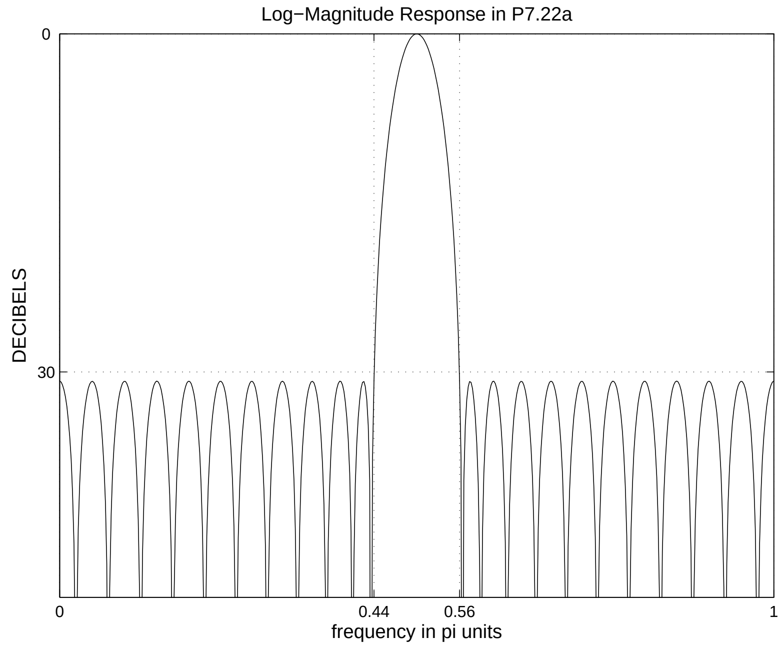


Figure 7.17: Log-Magnitude Response Plot in Problem P 7.22a

```
set(Hf_2,'NumberTitle','off','Name','P7.22b');
subplot(211);stem(n(101:201),x(101:201));title('Input sequence x(n)')
subplot(212);stem(n(101:201),y(101:201));title('Output sequence y(n)')
```

The time-domain response is shown in Figure 7.18.

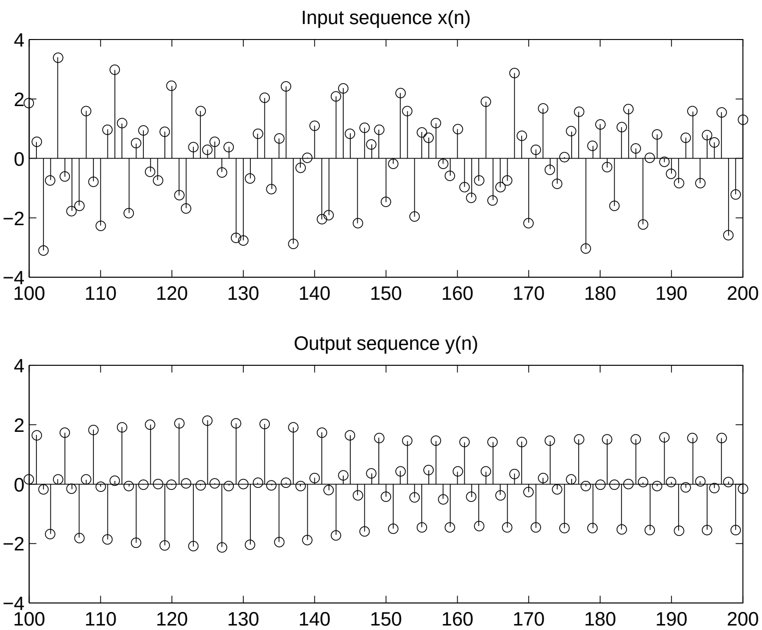


Figure 7.18: The Time-domain Response in Problem P7.22b

# Chapter 8

# IIR Filter Design

1. Problem **P 8.1**

   Analog Butterworth lowpass filter design: $\Omega_p = 30$ rad/s, $R_p = 1$ dB, $\Omega_s = 40$ rad/s, $A_s = 30$ dB.
   MATLAB Script:

```
clear, close all;
% Filter Specifications
Wp = 30; Ws = 40; Rp = 1; As = 30;
% Filter Design
[b,a] = afd_butt(Wp,Ws,Rp,As); format short e

*** Butterworth Filter Order = 15
% Cascade Structure
[C,B,A] = sdir2cas(b,a)
C =
  2.8199e+022
B =
     0     0     1
A =
  1.0000e+000  6.1393e+001  9.8484e+002
  1.0000e+000  5.7338e+001  9.8484e+002
  1.0000e+000  5.0777e+001  9.8484e+002
  1.0000e+000  4.1997e+001  9.8484e+002
  1.0000e+000  3.1382e+001  9.8484e+002
  1.0000e+000  1.9395e+001  9.8484e+002
  1.0000e+000  6.5606e+000  9.8484e+002
            0  1.0000e+000  3.1382e+001
% Frequency Response
Wmax = 100; [db,mag,pha,w] = freqs_m(b,a,Wmax); pha = unwrap(pha);
% Impulse Response
%   The impulse response of the designed filter when computed by Matlab is numerically
%   unstable due to large coefficient values.  Hence we compute the impulse response
%   of the filter with Wp/10 and Ws/10 band edges to keep coefficient values small.
%   The actual impulse response is time-scaled and amplitude scaled version of the
%   computed impulse response.
[b,a] = afd_butt(Wp/10,Ws/10,Rp,As); [ha,x,t] = impulse(b,a);

*** Butterworth Filter Order = 15
t = t/10; ha = ha/10;
%
```

```
% Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.1');
%
subplot(2,2,1);plot(w,mag); axis([0,Wmax,0,1.1]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('Magnitude','fontsize',10); title ('Magnitude Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
magRp = round(10^(-Rp/20)*100)/100;
set(gca,'YTickMode','manual','Ytick',[0;magRp;1],'fontsize',10);grid
%
subplot(2,2,2);plot(w,db); axis([0,Wmax,-100,0]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('Decibels','fontsize',10); title ('Log-Magnitude Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
set(gca,'YTickMode','manual','Ytick',[-100;-As;0],'fontsize',10);grid
AS = [' ',num2str(As)];
set(gca,'YTickLabelMode','manual','YTickLabels',['100';AS;'  0']);
%
minpha = floor(min(pha/pi)); maxpha = ceil(max(pha/pi));
subplot(2,2,3);plot(w,pha/pi); axis([0,Wmax,minpha,maxpha]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('Phase in pi units','fontsize',10); title ('Phase Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
phaWp = (round(pha(Wp/Wmax*500+1)/pi*100))/100;
phaWs = (round(pha(Ws/Wmax*500+1)/pi*100))/100;
set(gca,'YTickMode','manual','Ytick',[phaWs;phaWp;0],'fontsize',10); grid
%
subplot(2,2,4); plot(t,ha);  title ('Impulse Response','fontsize',10);
xlabel('t (sec)', 'fontsize',10); ylabel('ha(t)','fontsize',10);
%

suptitle('Analog Butterworth Lowpass Filter Design Plots in P 8.1')
```

The system function is given by

$$\begin{aligned} H_a(s) = \; & 2.8199\times 10^{22}\left(\frac{1}{s^2+61.393s+984.84}\right)\left(\frac{1}{s+31.382}\right) \\ & \left(\frac{1}{s^2+57.338s+984.84}\right)\left(\frac{1}{s^2+50.777s+984.84}\right)\times \\ & \left(\frac{1}{s^2+41.997s+984.84}\right)\left(\frac{1}{s^2+31.382s+984.84}\right)\times \\ & \left(\frac{1}{s^2+19.395s+984.84}\right)\left(\frac{1}{s^2+6.5606s+984.84}\right) \end{aligned}$$

The filter design plots are given in Figure 8.1.

2. Problem **P 8.2**

Analog Elliptic lowpass filter design: $\Omega_p = 10$ rad/s, $R_p = 1$ dB, $\Omega_s = 15$ rad/s, $A_s = 40$ dB.
MATLAB Script:

```
clear; close all;
% Filter Specifications
Wp = 10; Ws = 15; Rp = 1; As = 40;
% Filter Design
```

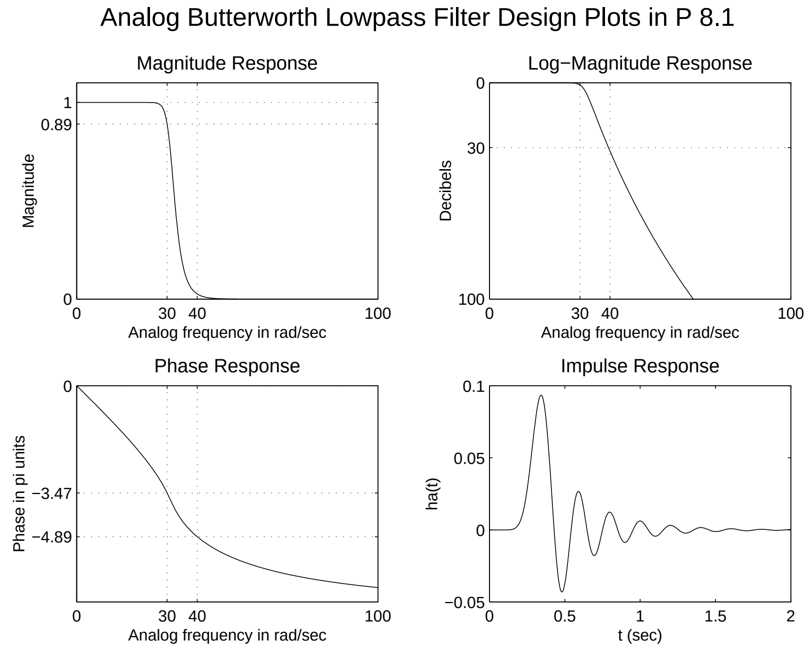


Figure 8.1: Analog Butterworth Lowpass Filter Plots in Problem P 8.1

```
[b,a] = afd_elip(Wp,Ws,Rp,As); format short e;

*** Elliptic Filter Order =  5
% Rational Function Form
a0 = a(1); b = b/a0, a = a/a0
b =
  4.6978e-001            0  2.2007e+002            0  2.2985e+004
a =
  1.0000e+000  9.2339e+000  1.8471e+002  1.1292e+003  7.8813e+003  2.2985e+004
% Frequency Response
Wmax = 30; [db,mag,pha,w] = freqs_m(b,a,Wmax); pha = unwrap(pha);
% Impulse Response
[ha,x,t] = impulse(b,a);
%
% Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.2');
%
subplot(2,2,1);plot(w,mag); axis([0,Wmax,0,1]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('Magnitude','fontsize',10); title ('Magnitude Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
magRp = round(10^(-Rp/20)*100)/100;
set(gca,'YTickMode','manual','Ytick',[0;magRp;1],'fontsize',10);grid
%
subplot(2,2,2);plot(w,db); axis([0,Wmax,-100,0]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('log-Magnitude in dB','fontsize',10);
```

```
title ('Log-Magnitude Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
set(gca,'YTickMode','manual','Ytick',[-100;-As;0],'fontsize',10);grid
AS = [' ',num2str(As)];
set(gca,'YTickLabelMode','manual','YTickLabels',['100';AS;'  0'],'fontsize',10);
%
minpha = floor(min(pha/pi)); maxpha = ceil(max(pha/pi));
subplot(2,2,3);plot(w,pha/pi); axis([0,Wmax,minpha,maxpha]);
xlabel('Analog frequency in rad/sec','fontsize',10);
ylabel('Phase in pi units','fontsize',10);
title ('Phase Response','fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;Wp;Ws;Wmax],'fontsize',10);
phaWp = (round(pha(Wp/Wmax*500+1)/pi*100))/100;
phaWs = (round(pha(Ws/Wmax*500+1)/pi*100))/100;
set(gca,'YTickMode','manual','Ytick',[phaWs;phaWp;0],'fontsize',10); grid
%
subplot(2,2,4); plot(t,ha);  title ('Impulse Response','fontsize',10);
xlabel('t (sec)', 'fontsize',10); ylabel('ha(t)','fontsize',10);
%
suptitle('Analog Elliptic Lowpass Filter Design Plots in P 8.2')
```

The system function is given by

$$H_a(s) = \frac{.46978s^4 + 220.07s^2 + 2298.5}{s^5 + 9.23s^4 + 184.71s^3 + 1129.2s^2 + 7881.3s + 22985}$$

The filter design plots are given in Figure 8.2.

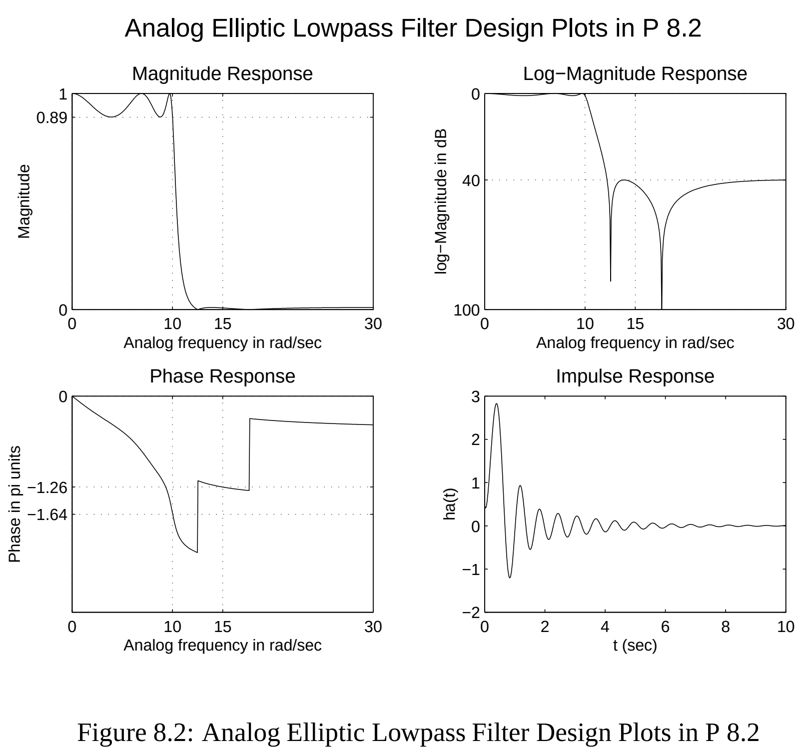


Figure 8.2: Analog Elliptic Lowpass Filter Design Plots in P 8.2

3. Problem **P 8.3**

The filter passband must include the 100 Hz component while the stopband must include the 130 Hz component. To obtain a minimum-order filter, the transition band must be as large as possible. This means that the passaband cutoff must be at 100 Hz while the stopband cutoff must be at 130 Hz. Hence the analog Chebyshev-I lowpass filter specifications are: $\Omega_p = 2\pi(100)$ rad/s, $R_p = 2$ dB, $\Omega_s = 2\pi(130)$ rad/s, $A_s = 50$ dB.
MATLAB Script:

```
clear, close all;
% Filter Specifications
Fp = 100; Fs = 130; Rp = 2; As = 50;
Wp = 2*pi*Fp; Ws = 2*pi*Fs;
% Filter Design
[b,a] = afd_chb1(Wp,Ws,Rp,As); format short e

*** Chebyshev-1 Filter Order =  9
% Cascade Structure
[C,B,A] = sdir2cas(b,a)
C =
  7.7954e+022
B =
     0     0     1
A =
  1.0000e+000  1.4245e+002  5.1926e+004
  1.0000e+000  1.1612e+002  1.6886e+005
  1.0000e+000  7.5794e+001  3.0183e+005
  1.0000e+000  2.6323e+001  3.8862e+005
            0  1.0000e+000  7.5794e+001
% Frequency Response
Fmax = 200; Wmax = 2*pi*Fmax; [db,mag,pha,w] = freqs_m(b,a,Wmax); pha = unwrap(pha);
%
% Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.3');
%
subplot(2,1,1);plot(w/(2*pi),mag); axis([0,Fmax,0,1.1]); set(gca,'fontsize',10);
xlabel('Analog frequency in Hz'); ylabel('Magnitude');
title ('Magnitude Response','fontsize',12);
set(gca,'XTickMode','manual','Xtick',[0;Fp;Fs;Fmax]);
magRp = round(10^(-Rp/20)*100)/100;
set(gca,'YTickMode','manual','Ytick',[0;magRp;1]);grid
%
subplot(2,1,2);plot(w/(2*pi),db); axis([0,Fmax,-100,0]); set(gca,'fontsize',10);
xlabel('Analog frequency in Hz'); ylabel('Decibels');
title ('Log-Magnitude Response','fontsize',12);
set(gca,'XTickMode','manual','Xtick',[0;Fp;Fs;Fmax]);
set(gca,'YTickMode','manual','Ytick',[-100;-As;0]);grid
AS = [' ',num2str(As)];
set(gca,'YTickLabelMode','manual','YTickLabels',['100';AS;'  0']);
%
suptitle('Analog Chebyshev-I Lowpass Filter Design Plots in P 8.3')
```

The system function is given by

$$H_a(s) = 7.7954 \times 10^{22} \left(\frac{1}{s^2 + 142.45s + 51926}\right) \left(\frac{1}{s^2 + 116.12s + 168860}\right) \left(\frac{1}{s^2 + 75.794s + 301830}\right) \left(\frac{1}{s^2 + 26.323s + 388620}\right) \left(\frac{1}{s + 75.794}\right)$$

The magnitude response plots are given in Figure 8.3.

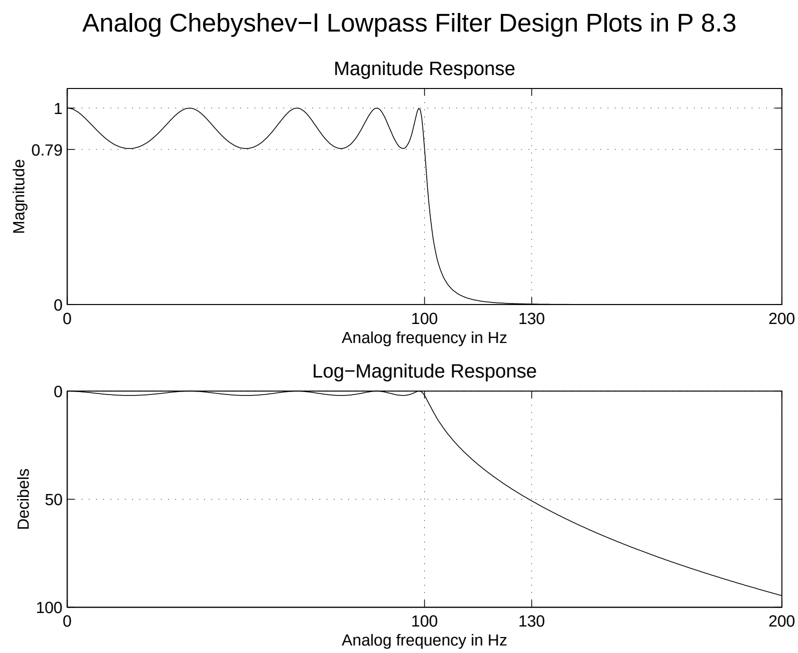


Figure 8.3: Analog Chebyshev-I Lowpass Filter Plots in Problem P 8.3

4. Problem **P 8.4**

Analog Chebyshev-II lowpass filter design: $\Omega_p = 2\pi(250)$ rad/s, $R_p = 0.5$ dB, $\Omega_s = 2\pi(300)$ rad/s, $A_s = 45$ dB.
MATLAB Script:

```
clear; close all;
% Filter Specifications
Fp = 250; Fs = 300; Rp = 0.5; As = 45;
Wp = 2*pi*Fp; Ws = 2*pi*Fs;
% Filter Design
[b,a] = afd_chb2(Wp,Ws,Rp,As); format short e;

*** Chebyshev-2 Filter Order = 12
% Rational Function Form
a0 = a(1); b = b/a0, a = a/a0
b =
  Columns 1 through 6
  5.6234e-003            0  1.4386e+006            0  5.9633e+013            0
  Columns 7 through 12
  9.0401e+020            0  6.1946e+027            0  1.9564e+034            0
  Column 13
```

```
  2.3171e+040
a =
  Columns 1 through 6
  1.0000e+000  1.5853e+004  1.2566e+008  6.5800e+011  2.5371e+015  7.5982e+018
  Columns 7 through 12
  1.8208e+022  3.5290e+025  5.5639e+028  6.9823e+031  6.9204e+034  4.7962e+037
  Column 13
  2.3171e+040
% Frequency Response
Fmax = 500; Wmax = 2*pi*Fmax; [db,mag,pha,w] = freqs_m(b,a,Wmax); pha = unwrap(pha);
% Impulse Response
%   The impulse response of the designed filter when computed by Matlab is numerically
%   unstable due to large coefficient values.  Hence we will compute the impulse respo
%   of the filter with Wp/1000 and Ws/1000 band edges to keep coefficient values small
%   The actual impulse response is time-scaled and amplitude scaled version of the
%   computed impulse response.
[b,a] = afd_chb2(Wp/1000,Ws/1000,Rp,As); [ha,x,t] = impulse(b,a);

*** Chebyshev-2 Filter Order = 12
t = t/1000; ha = ha/1000;
%
% Plots
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.4');
%
subplot(2,2,1);plot(w/(2*pi),mag); axis([0,Fmax,0,1]); set(gca,'fontsize',10);
xlabel('Analog frequency in Hz'); ylabel('Magnitude');
title ('Magnitude Response','fontsize',12);
set(gca,'XTickMode','manual','Xtick',[0;Fp;Fs;Fmax]);
magRp = round(10^(-Rp/20)*100)/100;
set(gca,'YTickMode','manual','Ytick',[0;magRp;1]);grid
%
subplot(2,2,2);plot(w/(2*pi),db); axis([0,Fmax,-100,0]); set(gca,'fontsize',10);
xlabel('Analog frequency in Hz'); ylabel('log-Magnitude in dB');
title ('Log-Magnitude Response','fontsize',12);
set(gca,'XTickMode','manual','Xtick',[0;Fp;Fs;Fmax]);
set(gca,'YTickMode','manual','Ytick',[-100;-As;0]);grid
AS = [' ',num2str(As)];
set(gca,'YTickLabelMode','manual','YTickLabels',['100';AS;'  0']);
%
minpha = floor(min(pha/pi)); maxpha = ceil(max(pha/pi));
subplot(2,2,3);plot(w/(2*pi),pha/pi); axis([0,Fmax,minpha,maxpha]); set(gca,'fontsize'
xlabel('Analog frequency in Hz'); ylabel('Phase in pi units');
title ('Phase Response','fontsize',12);
set(gca,'XTickMode','manual','Xtick',[0;Fp;Fs;Fmax]);
phaWp = (round(pha(Wp/Wmax*500+1)/pi*100))/100;
phaWs = (round(pha(Ws/Wmax*500+1)/pi*100))/100;
set(gca,'YTickMode','manual','Ytick',[phaWs;phaWp;0]); grid
%
subplot(2,2,4); plot(t,ha); set(gca,'fontsize',10);
title ('Impulse Response','fontsize',12); xlabel('t (sec)'); ylabel('ha(t)');
%
suptitle('Analog Chebyshev-II Lowpass Filter Design Plots in P 8.4')
```

The filter design plots are given in Figure 8.4.

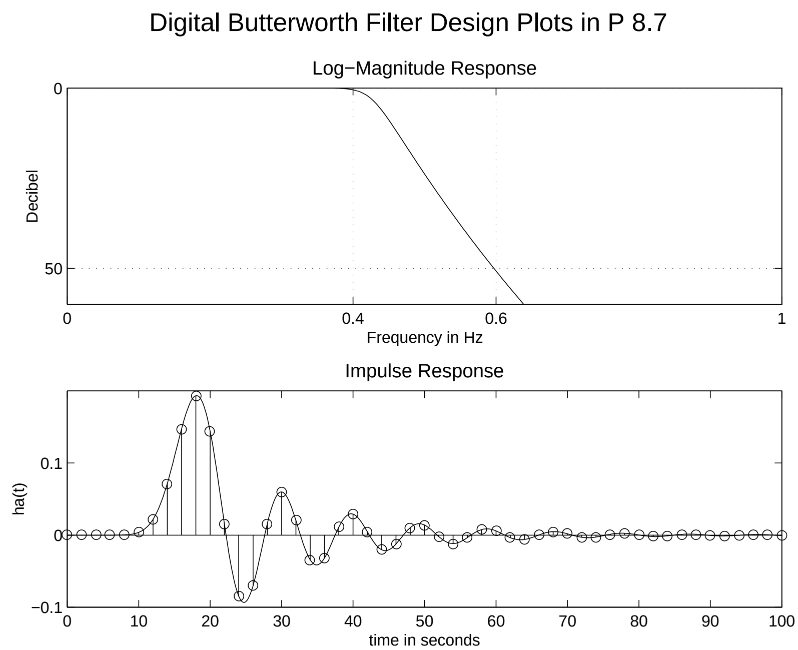


Figure 8.4: Analog Chebyshev-II Lowpass Filter Plots in Problem P 8.4

5. Problem **P 8.5**

   MATLAB function `afd.m`:

```
function [b,a] = afd(type,Fp,Fs,Rp,As)
%
% function [b,a] = afd(type,Fp,Fs,Rp,As)
%   Designs analog lowpass filters
% type = 'butter' or 'cheby1' or 'cheby2' or 'ellip'
%   Fp = passband cutoff in Hz
%   Fs = stopband cutoff in Hz
%   Rp = passband ripple in dB
%   As = stopband attenuation in dB
type = lower([type,' ']); type = type(1:6);
twopi = 2*pi;
if type == 'butter'
     [b,a] = afd_butt(twopi*Fp,twopi*Fs,Rp,As);
elseif type == 'cheby1'
     [b,a] = afd_chb1(twopi*Fp,twopi*Fs,Rp,As);
elseif type == 'cheby2'
     [b,a] = afd_chb2(twopi*Fp,twopi*Fs,Rp,As);
elseif type == 'ellip '
     [b,a] = afd_elip(twopi*Fp,twopi*Fs,Rp,As);
else
     error('Specify the correct type')
end
```

6. Problem **P 8.6**

Digital Chebyshev-1 Lowpass Filter Design using Impulse Invariance. MATLAB script:

```
clear; close all; Twopi = 2*pi;
%% Analog Filter Specifications
Fsam = 8000;                     % Sampling Rate in sam/sec
Fp = 1500;                       % Passband edge in Hz
Rp = 3;                          % Passband Ripple in dB
Fs = 2000;                       % Stopband edge in Hz
As = 40;                         % Stopband attenuation in dB
%
%% Digital Filter Specifications
wp = Twopi*Fp/Fsam;              % Passband edge in rad/sam
Rp = 3;                          % Passband Ripple in dB
ws = Twopi*Fs/Fsam;              % Stopband edge in rad/sam
As = 40;                         % Stopband attenuation in dB
```

(a) Part (a): $T = 1$. MATLAB script:

```
%% (a) Impulse Invariance Digital Design using T = 1
T = 1;
OmegaP = wp/T;                   % Analog Prototype Passband edge
OmegaS = ws/T;                   % Analog Prototype Stopband edge
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Chebyshev-1 Filter Order =  7
[b,a] = imp_invr(cs,ds,T);       % II Transformation
[C,B,A] = dir2par(b,a),          % Parallel form
C =
     []
B =
   -0.0561    0.0558
    0.1763   -0.1529
   -0.2787    0.2359
    0.1586         0
A =
    1.0000   -0.7767    0.9358
    1.0000   -1.0919    0.8304
    1.0000   -1.5217    0.7645
    1.0000   -0.8616         0
```

The filter design plots are shown in Figure 8.5.

(b) Part (b): $T = 1/8000$. MATLAB script:

```
%% (b) Impulse Invariance Digital Design using T = 1/8000
T = 1/8000;
OmegaP = wp/T;                   % Analog Prototype Passband edge
OmegaS = ws/T;                   % Analog Prototype Stopband edge
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Chebyshev-1 Filter Order =  7
[b,a] = imp_invr(cs,ds,T);       % II Transformation
[C,B,A] = dir2par(b,a),          % Parallel form
C =
     []
B =
  1.0e+003 *
```

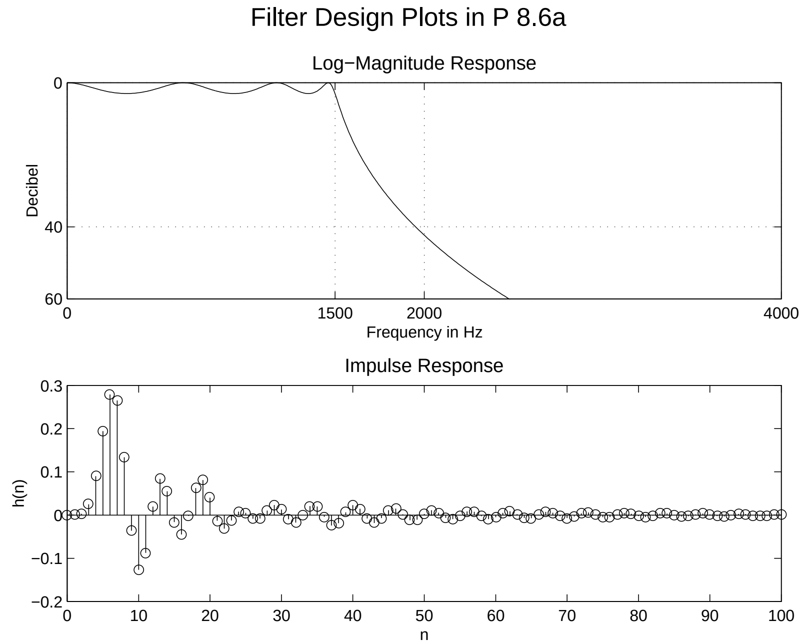


Figure 8.5: Impulse Invariance Design Method with $T = 1$ in Problem P 8.6a

```
   -0.4487    0.4460
    1.4102   -1.2232
   -2.2299    1.8869
    1.2684         0
A =
    1.0000   -0.7767    0.9358
    1.0000   -1.0919    0.8304
    1.0000   -1.5217    0.7645
    1.0000   -0.8616         0
```

The filter design plots are shown in Figure 8.6.

(c) Comparison: The designed system function as well as the impulse response in part 6b are *similar* to those in part 6a except for an overall gain due to $F_s = 1/T = 8000$. This problem can be avoided if in the impulse invariance design method we set

$$h(n) = T \cdot h_a(nT)$$

7. Problem **P 8.7**

Digital Butterworth Lowpass Filter Design using Impulse Invariance. MATLAB script:

```
clear; close all;
%% Digital Filter Specifications
wp = 0.4*pi;                      % Passband edge in rad/sam
Rp = 0.5;                         % Passband Ripple in dB
ws = 0.6*pi;                      % Stopband edge in rad/sam
As = 50;                          % Stopband attenuation in dB
%% Impulse Invariance Digital Design using T = 2
T = 2;
OmegaP = wp/T;                    % Analog Prototype Passband edge
OmegaS = ws/T;                    % Analog Prototype Stopband edge
```

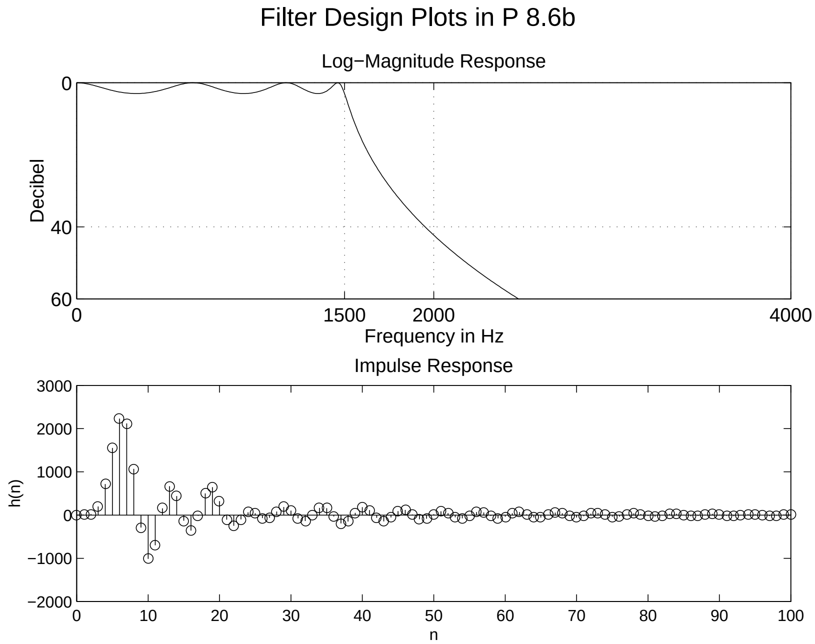


Figure 8.6: Impulse Invariance Design Method with $T = 1/8000$ in Problem P 8.6b

```
[cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Butterworth Filter Order = 17
[b,a] = imp_invr(cs,ds,T);       % II Transformation: rational form
% Plots of Log-magnitude Response and Impulse Response
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.7');
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-60,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
% Impulse response of the prototype analog filter
Nmax = 50; t = 0:T/10:T*Nmax; [ha,x,t] = impulse(cs,ds,t);
subplot(2,1,2); plot(t,ha); axis([0,T*Nmax,-0.1,0.2]); set(gca,'fontsize',10);
xlabel('time in seconds','fontsize',10); ylabel('ha(t)','fontsize',10);
title('Impulse Response','fontsize',12); hold on;
% Impulse response of the digital filter
[x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
stem(n*T,h); hold off;
suptitle('Digital Butterworth Filter Design Plots in P 8.7')
```

The filter design plots are shown in Figure 8.7.

Comparison: From Figure 8.7 we observe that the impulse response $h(n)$ of the digital filter is a sampled version of the impulse response $h_a(t)$ of the analog proptotype filter as expected.

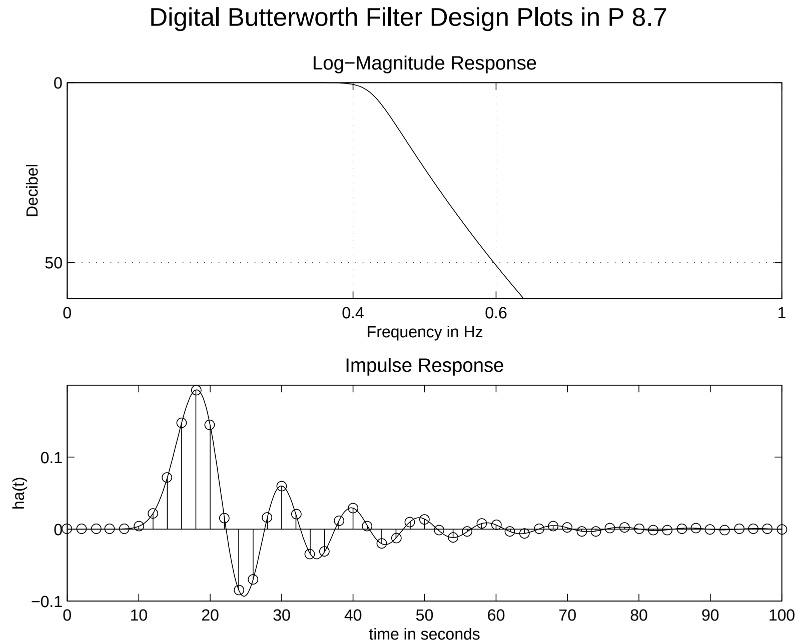


Figure 8.7: Impulse Invariance Design Method with $T = 2$ in Problem P 8.7

8. MATLAB function `dlpfd_ii.m`:

```
function [b,a] = dlpfd_ii(type,wp,ws,Rp,As,T)
%
% function [b,a] = dlpfd_ii(type,wp,ws,Rp,As,T)
%   Designs digital lowpass filters using impulse invariance mapping
% type = 'butter' or 'cheby1'
%   wp = passband cutoff in radians
%   ws = stopband cutoff in radians
%   Rp = passband ripple in dB
%   As = stopband attenuation in dB
%    T = sampling interval

if (type == 'cheby2')|(type == 'ellip ')
     error('Specify the correct type as butter or cheby1')
end
Fs = 1/T; twopi = 2*pi; K = Fs/twopi;
% Analog Prototype Specifications: Inverse mapping for frequencies
Fp = wp*K;                               % Prototype Passband freq in Hz
Fs = ws*K;                               % Prototype Stopband freq in Hz
ep = sqrt(10^(Rp/10)-1);                 % Passband Ripple parameter
Ripple = sqrt(1/(1+ep*ep));              % Passband Ripple
Attn = 1/(10^(As/20));                   % Stopband Attenuation
% Analog Butterworth Prototype Filter Calculation:
[cs,ds] = afd(type,Fp,Fs,Rp,As);
% Impulse Invariance transformation:
[b,a] = imp_invr(cs,ds,T);
[C,B,A] = dir2par(b,a)
```

MATLAB verification using Problem P8.7:

```
clear; close all; format short e
%% Problem P8.7 : Butterworth Design
%% Digital Filter Specifications
wp = 0.4*pi;                        % Passband edge in rad/sam
Rp = 0.5;                           % Passband Ripple in dB
ws = 0.6*pi;                        % Stopband edge in rad/sam
As = 50;                            % Stopband attenuation in dB
%
% Impulse Invariance Digital Design using T = 2
T = 2;
[b,a] = dlpfd_ii('butter',wp,ws,Rp,As,T);

*** Butterworth Filter Order = 17
C =
     []
B =
  3.3654e+000 -1.2937e+000
 -8.3434e+000 -6.9148e+000
  2.9916e-002  2.7095e-001
 -5.2499e+001  2.6839e+001
  1.2550e+002  4.3610e+000
  1.7953e+002 -1.0414e+002
 -5.1360e+002  1.0421e+002
 -1.3402e+002  8.1785e+001
  4.0004e+002            0
A =
  1.0000e+000 -3.9001e-001  4.8110e-001
  1.0000e+000 -4.0277e-001  3.0369e-001
  1.0000e+000 -4.1965e-001  7.8138e-001
  1.0000e+000 -4.3154e-001  1.9964e-001
  1.0000e+000 -4.6254e-001  1.3864e-001
  1.0000e+000 -4.8933e-001  1.0298e-001
  1.0000e+000 -5.0923e-001  8.2651e-002
  1.0000e+000 -5.2131e-001  7.2211e-002
  1.0000e+000 -2.6267e-001            0
%
% Plots of Log-magnitude Response and Impulse Response
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.8b');
% Log-Magnitude response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-60,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in pi units'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
% Magnitude response
subplot(2,1,2); plot(w/pi,mag); axis([0,1,0,0.55]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[0;0.5]);grid
```

```
set(gca,'YTickLabelMode','manual','YtickLabels',['0.0';'0.5']);
xlabel('Frequency in pi units'); ylabel('|H|');
title('Magnitude Response','fontsize',12); axis;
suptitle('Digital Butterworth Filter Design Plots in P 8.8')
```

The filter design plots are given in Figure 8.8.

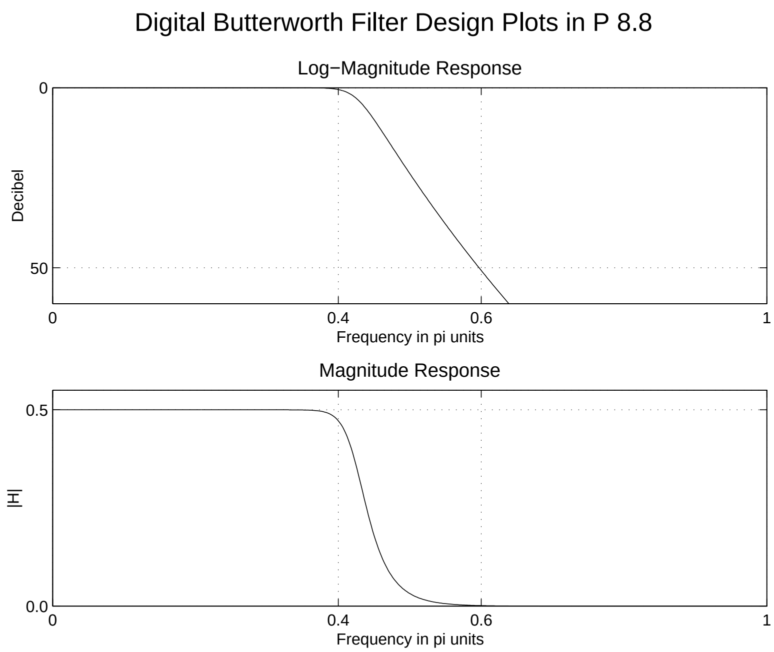


Figure 8.8: Digital filter design plots in Problem P8.8.

9. Problem **P 8.11**

   Digital Butterworth Lowpass Filter Design using Bilinear transformation. MATLAB script:

```
clear; close all;
%% Digital Filter Specifications
wp = 0.4*pi;                        % Passband edge in rad/sam
Rp = 0.5;                           % Passband Ripple in dB
ws = 0.6*pi;                        % Stopband edge in rad/sam
As = 50;                            % Stopband attenuation in dB
```

   (a) Part(a): $T = 2$. MATLAB script:

```
T = 2;
OmegaP = (2/T)*tan(wp/2);        % Analog Prototype Passband edge
OmegaS = (2/T)*tan(ws/2);        % Analog Prototype Stopband edge
[cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Butterworth Filter Order = 11
[b,a] = bilinear(cs,ds,1/T)      % Bilinear Transformation
b =
  Columns 1 through 7
    0.0004    0.0048    0.0238    0.0715    0.1429    0.2001    0.2001
```

```
  Columns 8 through 12
    0.1429    0.0715    0.0238    0.0048    0.0004
a =
  Columns 1 through 7
    1.0000   -1.5495    2.5107   -2.1798    1.7043   -0.8997    0.4005
  Columns 8 through 12
   -0.1258    0.0309   -0.0050    0.0005    0.0000
%
% Plots of Log-magnitude Response and Impulse Response
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.11a');
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-60,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
% Impulse response of the prototype analog filter
Nmax = 50; t = 0:T/10:T*Nmax; [ha,x,t] = impulse(cs,ds,t);
subplot(2,1,2); plot(t,ha); axis([0,T*Nmax,-0.3,0.4]); set(gca,'fontsize',10);
xlabel('time in seconds','fontsize',10); ylabel('ha(t)','fontsize',10);
title('Impulse Response','fontsize',12); hold on;
% Impulse response of the digital filter
[x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
stem(n*T,h); hold off;
suptitle('Digital Butterworth Filter Design Plots in P 8.11a');
```

The filter design plots are shown in Figure 8.9.

Comparison: If we compare filter orders from two methods then bilinear transformation gives the lower order than the impulse invariance method. This implies that the bilinear transformation design method is a better one in all aspects. If we compare the impulse responses then we observe from Figure 8.9 that the digital impulse response is *not* a sampled version of the analog impulse response as was the case in Figure 8.7.

(b) Part (b): Use of the `butter` function. MATLAB script:

```
[N,wn] = buttord(wp/pi,ws/pi,Rp,As);
[b,a] = butter(N,wn)
b =
  Columns 1 through 7
    0.0005    0.0054    0.0270    0.0810    0.1619    0.2267    0.2267
  Columns 8 through 12
    0.1619    0.0810    0.0270    0.0054    0.0005
a =
  Columns 1 through 7
    1.0000   -1.4131    2.3371   -1.9279    1.5223   -0.7770    0.3477
  Columns 8 through 12
   -0.1066    0.0262   -0.0042    0.0004    0.0000
%
% Plots of Log-magnitude Response and Impulse Response
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P8.11b');
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
```

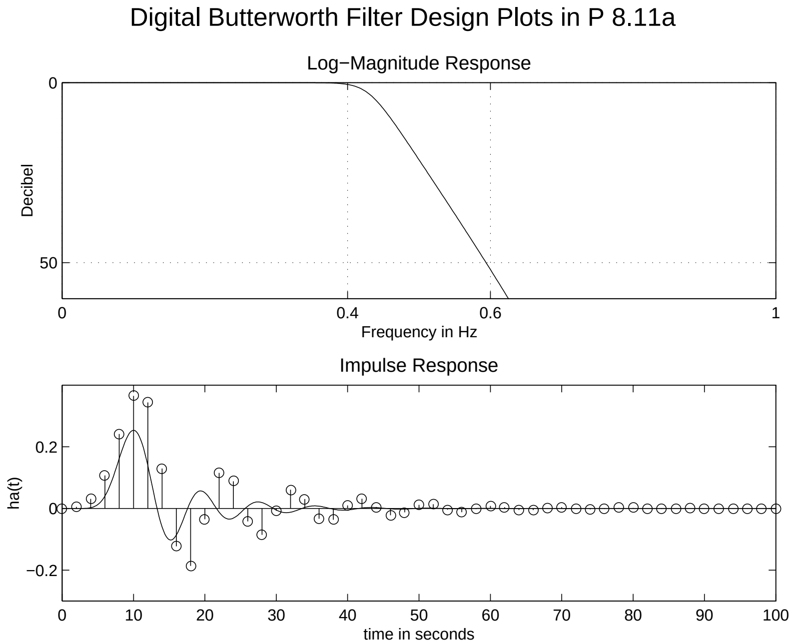


Figure 8.9: Bilinear Transformation Design Method with $T = 2$ in Problem P 8.11a

```
subplot(2,1,1); plot(w/pi,db); axis([0,1,-60,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
% Impulse response of the digital filter
Nmax = 50; [x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
subplot(2,1,2); stem(n,h); axis([0,Nmax,-0.3,0.4]); set(gca,'fontsize',10);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
title('Impulse Response','fontsize',12);
suptitle('Digital Butterworth Filter Design Plots in P 8.11b');
```

The filter design plots are shown in Figure 8.10.

Comparison: If we compare the plots of filter responses in part 9a with those in part 9b, then we observe that the `butter` function satisfies stopband specifications *exactly* at $\omega_s$. Otherwise the both designs are essentially *similar*.

10. Problem **P 8.13**

Digital Chebyshev-1 Lowpass Filter Design using Bilinear transformation. MATLAB script:

```
clear; close all; Twopi = 2*pi;
%% Analog Filter Specifications
Fsam = 8000;                         % Sampling Rate in sam/sec
Fp = 1500;                           % Passband edge in Hz
Rp = 3;                              % Passband Ripple in dB
Fs = 2000;                           % Stopband edge in Hz
As = 40;                             % Stopband attenuation in dB
%
```

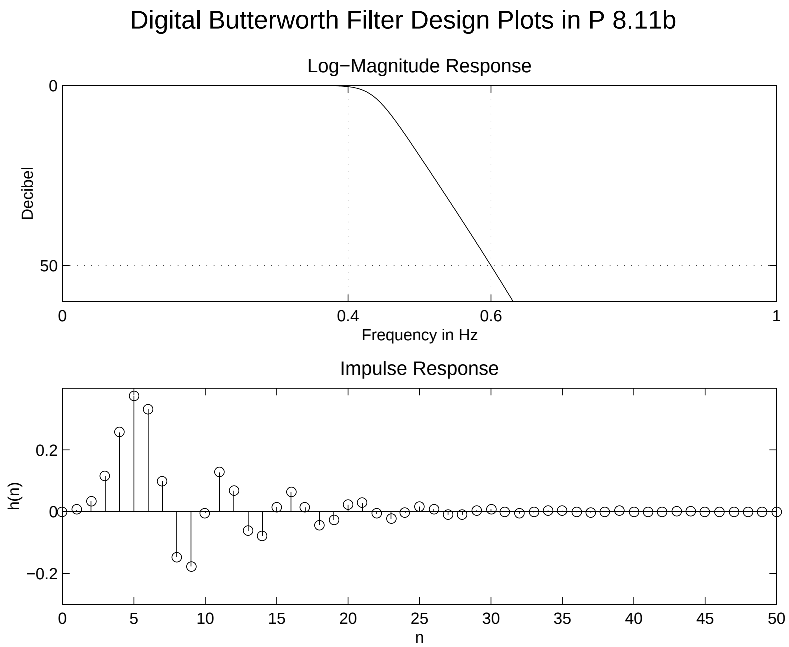


Figure 8.10: Butterworth filter design using the `butter` function in Problem P 8.11b

```
%% Digital Filter Specifications
wp = Twopi*Fp/Fsam;                 % Passband edge in rad/sam
Rp = 3;                             % Passband Ripple in dB
ws = Twopi*Fs/Fsam;                 % Stopband edge in rad/sam
As = 40;                            % Stopband attenuation in dB
```

(a) Part(a): $T = 1$. MATLAB script:

```
%% (a) Bilinear Transformation Digital Design using T = 1
T = 1;
OmegaP = (2/T)*tan(wp/2);         % Analog Prototype Passband edge
OmegaS = (2/T)*tan(ws/2);         % Analog Prototype Stopband edge
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Chebyshev-1 Filter Order =  6
[b,a] = bilinear(cs,ds,1/T);      % Bilinear Transformation
[C,B,A] = dir2cas(b,a),           % Cascade form
C =
    0.0011
B =
    1.0000    2.0126    1.0127
    1.0000    1.9874    0.9875
    1.0000    2.0001    0.9999
A =
    1.0000   -0.7766    0.9308
    1.0000   -1.1177    0.7966
    1.0000   -1.5612    0.6901
```

The filter design plots are shown in Figure 8.11.

(b) Part(b): $T = 1/8000$. MATLAB script:

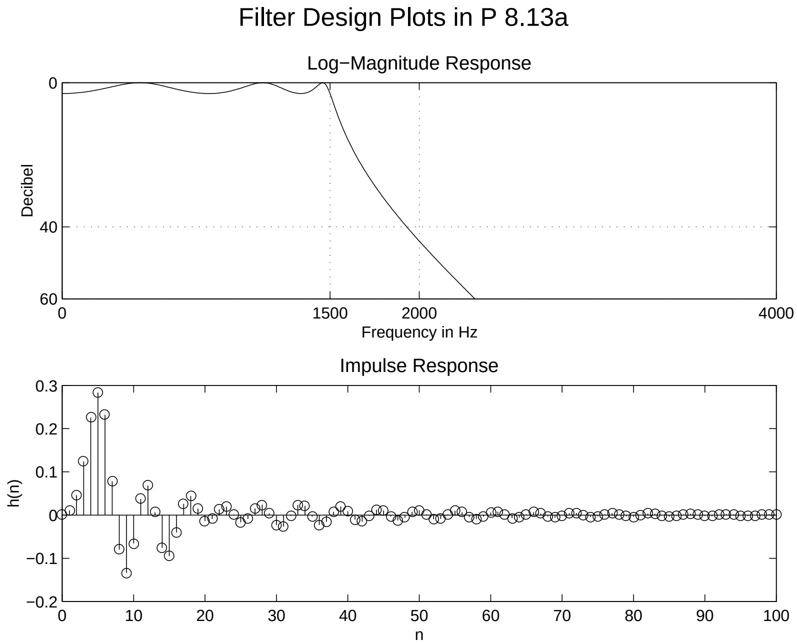


Figure 8.11: Bilinear Transformation Design Method with $T = 1$ in Problem P 8.13a

```
%% (b) Impulse Invariance Digital Design using T = 1/8000
T = 1/8000;
OmegaP = (2/T)*tan(wp/2);        % Analog Prototype Passband edge
OmegaS = (2/T)*tan(ws/2);        % Analog Prototype Stopband edge
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Chebyshev-1 Filter Order =  6
[b,a] = bilinear(cs,ds,1/T);     % II Transformation
[C,B,A] = dir2cas(b,a),          % Cascade form
C =
    0.0011
B =
    1.0000    2.0265    1.0267
    1.0000    1.9998    1.0001
    1.0000    1.9737    0.9739
A =
    1.0000   -0.7766    0.9308
    1.0000   -1.1177    0.7966
    1.0000   -1.5612    0.6901
```

The filter design plots are shown in Figure 8.12.

(c) Comparison: If we compare the designed system function as well as the plots of system responses in part 10a and in part 10a, then we observe that these are *exactly same*. If we compare the impulse invariance design in Problem 6 with this one then we note that the order of the impulse invariance designed filter is one higher. This implies that the bilinear transformation design method is a better one in all aspects.

11. Digital lowpass filter design using elliptic prototype.

    MATLAB script using the `bilinear` function:

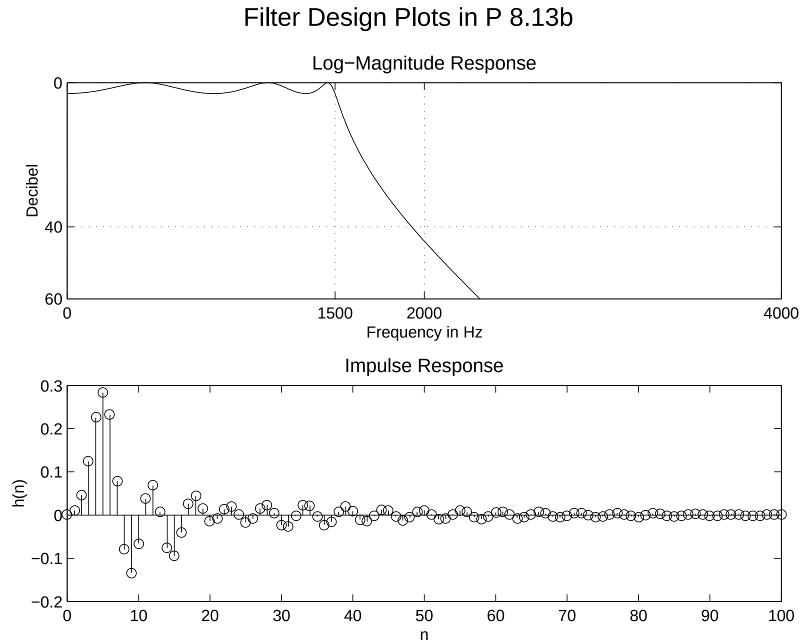


Figure 8.12: Bilinear Transformation Design Method with $T = 1/8000$ in Problem P 8.13b

```
clear; close all;
%% Digital Filter Specifications
wp = 0.4*pi;                     % Passband edge in rad/sam
Rp = 1;                          % Passband Ripple in dB
ws = 0.6*pi;                     % Stopband edge in rad/sam
As = 60;                         % Stopband attenuation in dB
%
%% (a) Bilinear Transformation Digital Design using T = 2
T = 2;
OmegaP = (2/T)*tan(wp/2);        % Analog Prototype Passband edge
OmegaS = (2/T)*tan(ws/2);        % Analog Prototype Stopband edge
[cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As); % Analog Prototype Design

*** Elliptic Filter Order =  5
[b,a] = bilinear(cs,ds,1/T);     % Bilinear Transformation
%
% Plots of Log-magnitude Response and Impulse Response
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.14a');
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-80,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
```

```
% Impulse response of the prototype analog filter
Nmax = 50; t = 0:T/10:T*Nmax; [ha,x,t] = impulse(cs,ds,t);
subplot(2,1,2); plot(t,ha); axis([0,T*Nmax,-0.3,0.4]); set(gca,'fontsize',10);
xlabel('time in seconds','fontsize',10); ylabel('ha(t)','fontsize',10);
title('Impulse Response','fontsize',12); hold on;
% Impulse response of the digital filter
[x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
stem(n*T,h); hold off;
suptitle('Digital Elliptic Filter Design Plots in P 8.14a');
```

The filter design plots are shown in Figure 8.13.

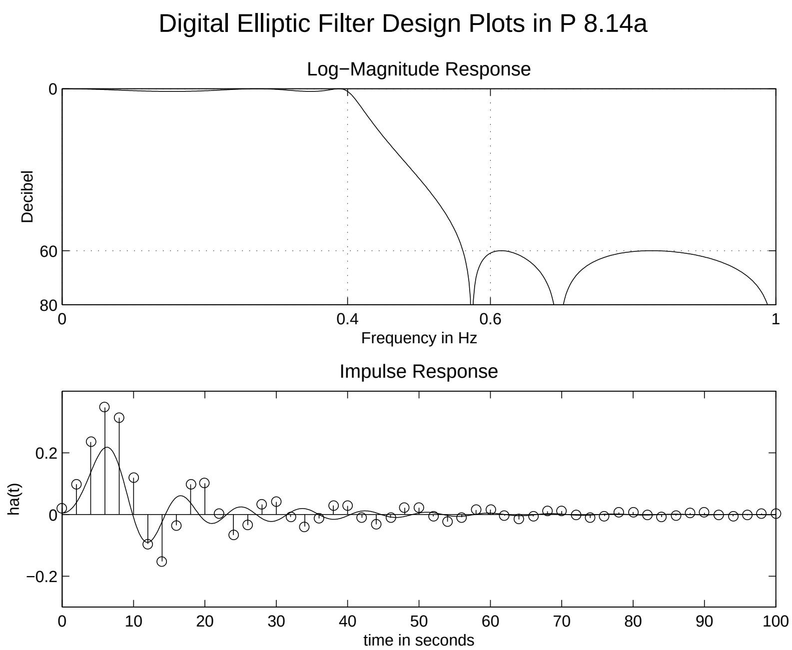


Figure 8.13: Digital elliptic lowpass filter design using the bilinear function in Problem P8.14a.

MATLAB script using the `ellip` function:

```
%% (b) Use of the 'Ellip' function
[N,wn] = ellipord(wp/pi,ws/pi,Rp,As);
[b,a] = ellip(N,Rp,As,wn);
%
% Plots of Log-magnitude Response and Impulse Response
Hf_2 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_2,'NumberTitle','off','Name','P8.14b');
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-80,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;wp/pi;ws/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
```

```
title('Log-Magnitude Response','fontsize',12); axis;
% Impulse response of the digital filter
Nmax = 50; [x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
subplot(2,1,2); stem(n,h); axis([0,Nmax,-0.3,0.4]); set(gca,'fontsize',10);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
title('Impulse Response','fontsize',12);
suptitle('Digital Elliptic Filter Design Plots in P 8.14b');
```

The filter design plots are shown in Figure 8.14. From these two figures we observe that both functions give the same design in which the digital filter impulse response is not a sampled version of the corresponding analog filter impulse response.

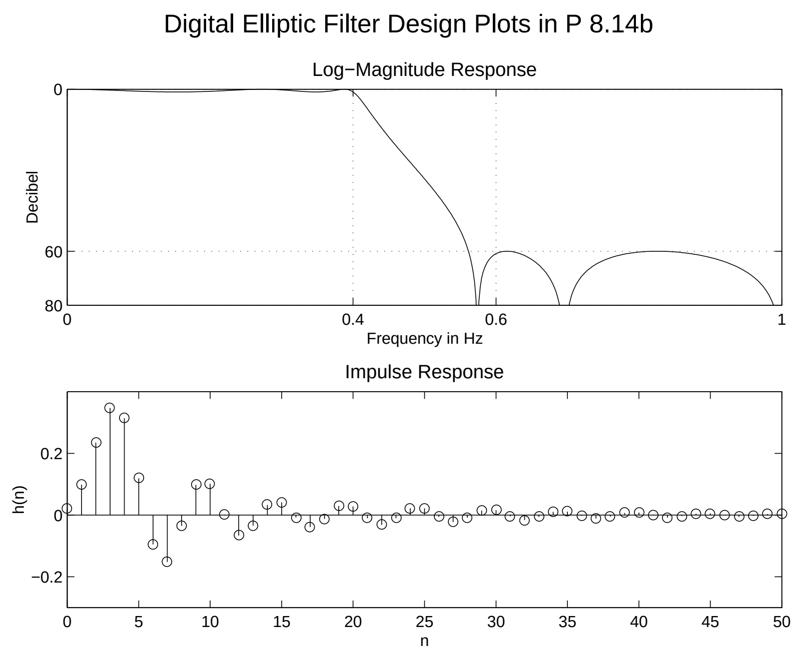


Figure 8.14: Digital elliptic lowpass filter design using the ellip function in Problem P8.14b.

12. Digital elliptic highpass filter design using bilinear mapping.

MATLAB function `dhpfd_bl.m`:

```
function [b,a] = dhpfd_bl(type,wp,ws,Rp,As)
%
% function [b,a] = dhpfd_bl(type,wp,ws,Rp,As,T)
%   Designs digital highpass filters using bilinear
%    b = Numerator polynomial of the highpass filter
%    a = Denominator polynomial of the highpass filter
% type = 'butter' or 'cheby1' or 'cheby2' or 'ellip'
%   wp = passband cutoff in radians
%   ws = stopband cutoff in radians (ws < wp)
%   Rp = passband ripple in dB
%   As = stopband attenuation in dB

% Determine the digital lowpass cutoff frequecies:
wplp = 0.2*pi;
```

```
alpha = -(cos((wplp+wp)/2))/(cos((wplp-wp)/2));
wslp = angle(-(exp(-j*ws)+alpha)/(1+alpha*exp(-j*ws)));
%
% Compute Analog lowpass Prototype Specifications:
T = 1; Fs = 1/T;
OmegaP = (2/T)*tan(wplp/2);
OmegaS = (2/T)*tan(wslp/2);

% Design Analog Chebyshev Prototype Lowpass Filter:
type = lower([type,' ']); type = type(1:6);
if type == 'butter'
     [cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As);
elseif type == 'cheby1'
     [cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
elseif type == 'cheby2'
     [cs,ds] = afd_chb2(OmegaP,OmegaS,Rp,As);
elseif type == 'ellip '
     [cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As);
else
     error('Specify the correct type')
end

% Perform Bilinear transformation to obtain digital lowpass
[blp,alp] = bilinear(cs,ds,Fs);

% Transform digital lowpass into highpass filter
Nz = -[alpha,1]; Dz = [1,alpha];
[b,a] = zmapping(blp,alp,Nz,Dz);
```

(a) Design using the `dhpfd_bl` function:

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.17')

%% Digital Filter Specifications
type = 'ellip';                 % Elliptic design
ws = 0.4*pi;                    % Passband edge in rad/sam
As = 60;                        % Stopband attenuation in dB
wp = 0.6*pi;                    % Stopband edge in rad/sam
Rp = 1;                         % Passband Ripple in dB
%
%% (a) Use of the dhpfd_bl function
[b,a] = dhpfd_bl(type,wp,ws,Rp,As)

*** Elliptic Filter Order =  5
b =
    0.0208   -0.0543    0.0862   -0.0862    0.0543   -0.0208
a =
    1.0000    2.1266    2.9241    2.3756    1.2130    0.3123
% Plot of Log-magnitude Response
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,1); plot(w/pi,db); axis([0,1,-80,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;ws/pi;wp/pi;1]);
```

```
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Design using the dhpfd_bl function','fontsize',12); axis;
```

The filter frequency response plot is shown in the top row of Figure 8.15.

(b) Design using the `ellip` function:

```
%% (b) Use of the 'Ellip' function
[N,wn] = ellipord(wp/pi,ws/pi,Rp,As);
[b,a] = ellip(N,Rp,As,wn,'high')
b =
    0.0208   -0.0543    0.0862   -0.0862    0.0543   -0.0208
a =
    1.0000    2.1266    2.9241    2.3756    1.2130    0.3123
%
% Plot of Log-magnitude Response
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,2); plot(w/pi,db); axis([0,1,-80,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0;ws/pi;wp/pi;1]);
set(gca,'YTickMode','manual','Ytick',[-80;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['80';AS;' 0']);
xlabel('Frequency in Hz'); ylabel('Decibel');
title('Design using the ellip function','fontsize',12); axis;
suptitle('Digital Elliptic Filter Design Plots in P 8.17');
```

The filter frequency response plot is shown in the bottom row of Figure 8.15. Both MATLAB scripts and the Figure 8.15 indicate that we designed essentially the same filter.

13. Digital Chebyshev-2 bandpass filter design using bilinear transformation. MATLAB script:

```
clear; close all;
Hf_1 = figure('Units','normalized','position',[0.1,0.1,0.8,0.8],'color',[0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P8.19')

%% Digital Filter Specifications
ws1 = 0.3*pi;                     % Lower passband edge in rad/sam
ws2 = 0.6*pi;                     % upper passband edge in rad/sam
 As = 50;                         % Stopband attenuation in dB
wp1 = 0.4*pi;                     % Lower passband edge in rad/sam
wp2 = 0.5 *pi;                    % Lower passband edge in rad/sam
 Rp = 0.5;                        % Passband Ripple in dB
%
%% Use of the 'cheby2' function
ws = [ws1,ws2]/pi; wp = [wp1,wp2]/pi;
[N,wn] = cheb2ord(wp,ws,Rp,As)
N =
     5
wn =
    0.3390    0.5661
[b,a] = cheby2(N,As,wn)
b =
  Columns 1 through 7
```

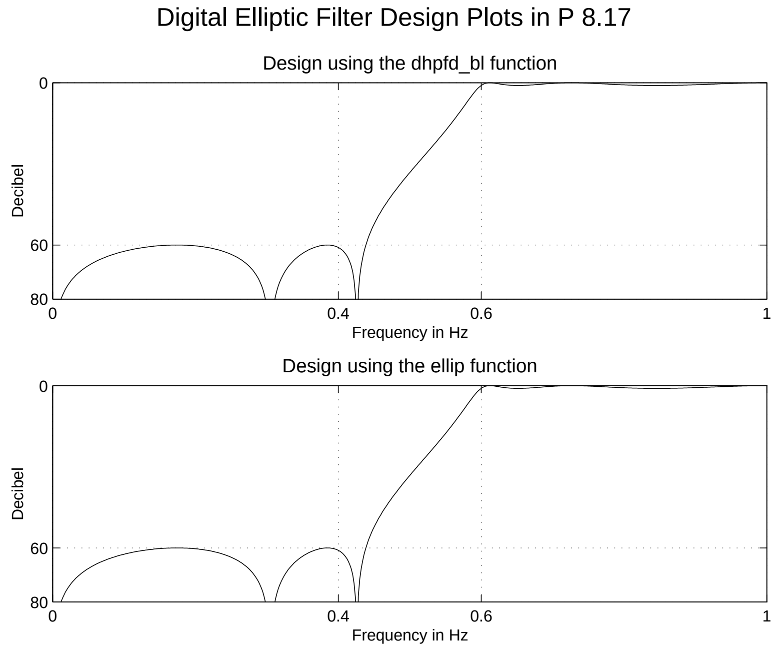


Figure 8.15: Digital elliptic highpass filter design plots in Problem 8.17.

```
    0.0050   -0.0050    0.0087   -0.0061    0.0060    0.0000   -0.0060
  Columns 8 through 11
    0.0061   -0.0087    0.0050   -0.0050
a =
  Columns 1 through 7
    1.0000   -1.3820    4.4930   -4.3737    7.4582   -5.1221    5.7817
  Columns 8 through 11
   -2.6221    2.0882   -0.4936    0.2764
%
% Plot of the filter Responses
% Impulse response
Nmax = 50; [x,n] = impseq(0,0,Nmax); h = filter(b,a,x);
subplot(2,1,1); stem(n,h); axis([0,Nmax,-0.15,0.15]); set(gca,'fontsize',10);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
title('Impulse Response','fontsize',12);
% Frequency response
[db,mag,pha,grd,w] = freqz_m(b,a);
subplot(2,1,2); plot(w/pi,db); axis([0,1,-70,0]); set(gca,'fontsize',10);
set(gca,'XTickMode','manual','Xtick',[0,ws,wp,1]);
set(gca,'YTickMode','manual','Ytick',[-70;-As;0]);grid
AS = [num2str(As)];
set(gca,'YTickLabelMode','manual','YtickLabels',['70';AS;' 0']);
xlabel('Frequency in pi units'); ylabel('Decibel');
title('Log-Magnitude Response','fontsize',12); axis;
suptitle('Digital Chebyshev-2 Bandpass Filter Design Plots in P 8.19');
```

The filter impulse and log-magnitude response plots are shown in Figure 8.16.

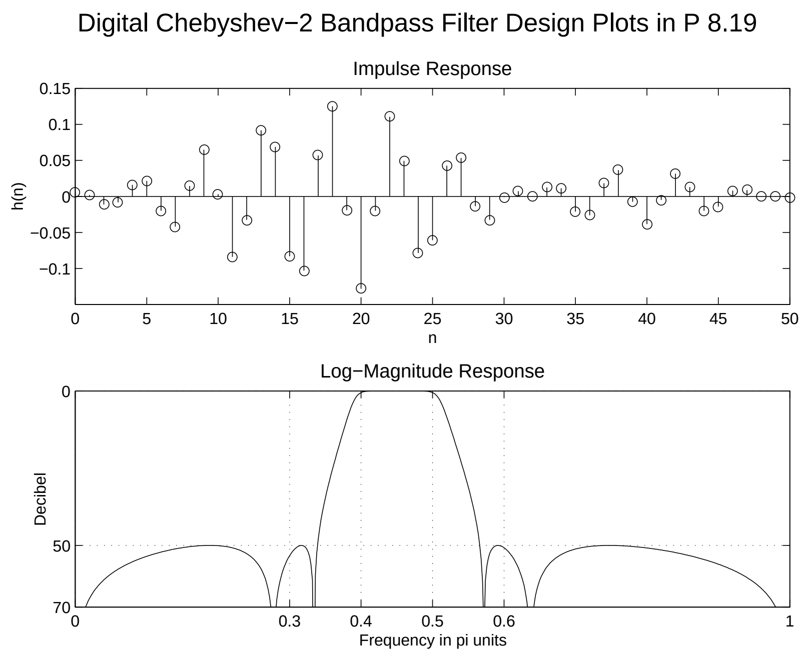


Figure 8.16: Digital Chebyshev-2 bandpass filter design plots in Problem P8.19.