CHAPTER 9

Structure for Discrete-Time Systems

Tutorial Problems

1. (a) Solution:

$$v[n] = x[n] + \frac{1}{3}v[n-1] \qquad \text{(A)}$$
$$y[n] = 6v[n-1] + 3(2x[n] + v[n]) \qquad \text{(B)}$$

From difference equation (A), we have

$$\frac{V(z)}{X(z)} = \frac{1}{1 - \frac{1}{3}z^{-1}}$$

From difference equation (B) (plug (A) in), we have

$$\frac{Y(z)}{V(z)} = 6 + 4z^{-1}$$

Hence, the system function is:

$$\frac{Y(z)}{X(z)} = \frac{Y(z)}{V(z)} \cdot \frac{V(z)}{X(z)} = \frac{6 + 4z^{-1}}{1 - \frac{1}{3}z^{-1}}$$

The difference equation is:

$$y[n] = 6x[n] + 4x[n-1] + \frac{1}{3}y[n-1]$$

(b) Solution:
The system function is:

$$H(z) = \frac{6 + 4z^{-1}}{1 - \frac{1}{3}z^{-1}} = -12 + \frac{18}{1 - \frac{1}{3}z^{-1}}$$

Applying the inverse z-transform, the impulse response is:

$$h[n] = -12\delta[n] + 18 \cdot \left(\frac{1}{3}\right)^n u[n]$$

2. Solution:
The difference equation of system (a) is:

$$y[n] = x[n] + 2r\cos\theta y[n-1] - r^2 y[n-2]$$

For system (b), we have

$$v[n] = x[n] + r\cos\theta v[n-1] + r\sin\theta y[n-2] \quad \text{(A)}$$
$$y[n] = r\sin\theta v[n] + r\cos\theta y[n-1] \quad \text{(B)}$$

Solving equation (B), we have

$$v[n] = \frac{y[n] - r\cos\theta y[n-1]}{r\sin\theta}$$

Plug $v[n]$ into equation (A), after simple algebraic manipulations, we can conclude the difference equation of system (b) as:

$$y[n] = x[n] + 2r\cos\theta y[n-1] - r^2\cos^2\theta y[n-2] + r^2\sin^2\theta y[n-2]$$

Comparing the two difference equations, we can tell the two system is not the same.

3. (a) See graph below.
   (b) See graph below.
   (c) See graph below.
   (d) See graph below.

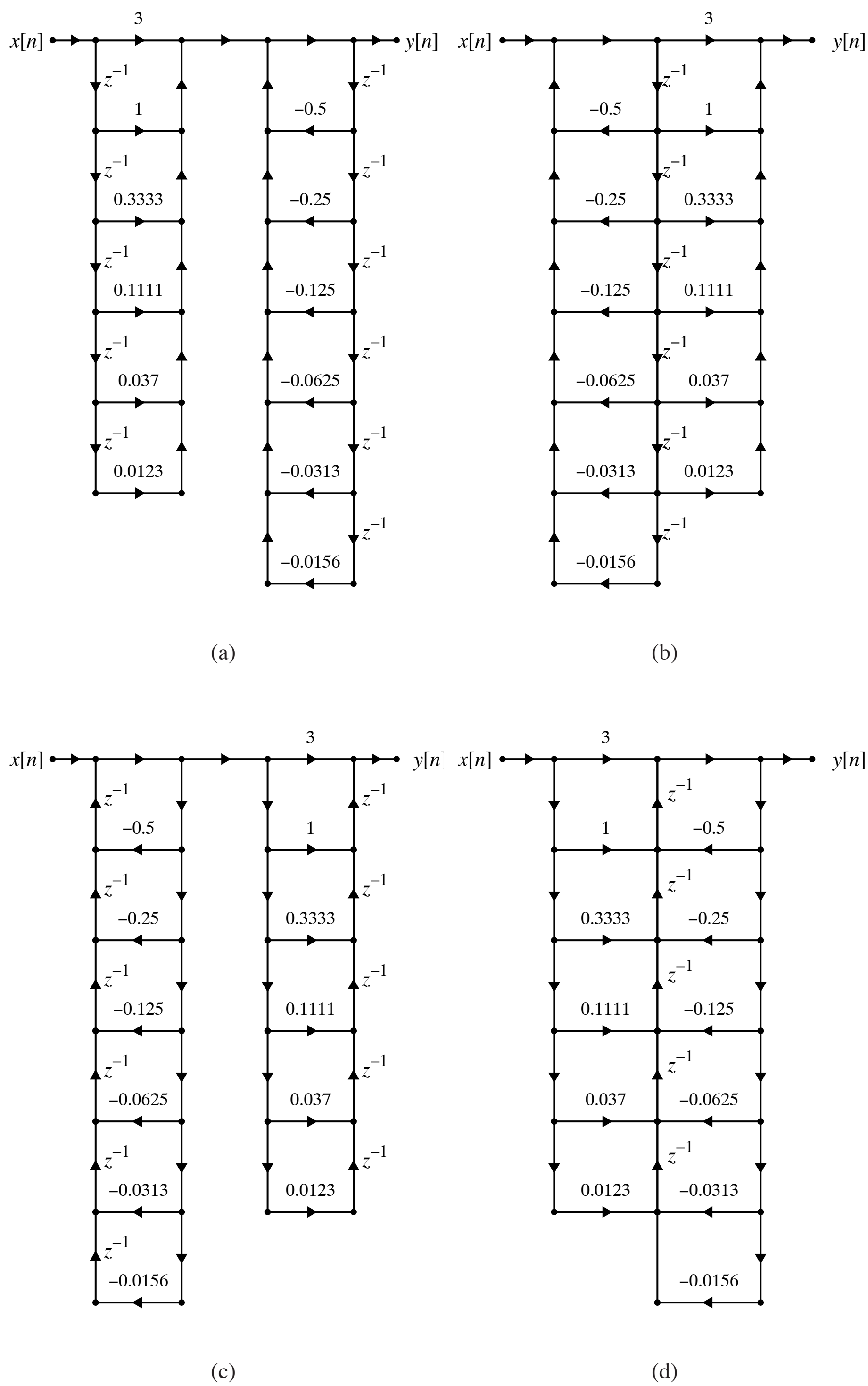


FIGURE 9.1: (a) Normal direct I form. (b) Normal direct II form. (c) Transposed direct I form. (d) Transposed direct II form.

4. (a) MATLAB function:

```
function [y] = filterdf1(b,a,x,yi,xi)
% Implementation of Direct Form I structure (Normal Form)
%  with initial conditions
% [y] = filterdf1(b,a,x,yi,xi)
if nargin < 5
    xi = zeros(length(b)-1,1);
end
if nargin < 4
    yi = zeros(1,length(a)-1);
end
M = length(b)-1; N = length(a)-1;
a0 = a(1); a = reshape(a,1,N+1)/a0;
b = reshape(b,1,M+1)/a0; a = a(2:end);
Lx = length(x); x = [flipud(xi(:));x(:)];
y = [fliplr(yi) zeros(1,Lx)];
for n = 1:Lx
    sn = b*x(n+M:-1:n);
    y(n+N) = sn - y(n+N-1:-1:n)*a';
end
y = y(N+1:end);
```

(b) Solution:
Taking the one-sided z-transform, we have

$$\begin{aligned}
Y^+(z) &= \frac{1}{1-\frac{1}{4}z^{-1}} + \frac{3}{2}(y[-1] + z^{-1}Y^+(z)) - \frac{1}{2}(y[-2] + y[-1]z^{-1} + z^{-2}Y^+(z)) \\
&= \frac{1}{1-\frac{1}{4}z^{-1}} + \frac{3}{2}(4 + z^{-1}Y^+(z)) - \frac{1}{2}(10 + 4z^{-1} + z^{-2}Y^+(z)) \\
&= \frac{2-\frac{9}{4}z^{-1}+\frac{1}{2}z^{-2}}{(1-\frac{1}{2}z^{-1})(1-z^{-1})(1-\frac{1}{4}z^{-1})} \\
&= \frac{\frac{2}{3}}{1-z^{-1}} + \frac{1}{1-\frac{1}{2}z^{-1}} + \frac{\frac{1}{3}}{1-\frac{1}{4}z^{-1}}
\end{aligned}$$

Hence, the impulse response is:

$$h[n] = \left[\frac{2}{3} + \left(\frac{1}{2}\right)^n + \frac{1}{3}\left(\frac{1}{4}\right)^n\right] \cdot u[n]$$

(c) See plot below.
MATLAB script:

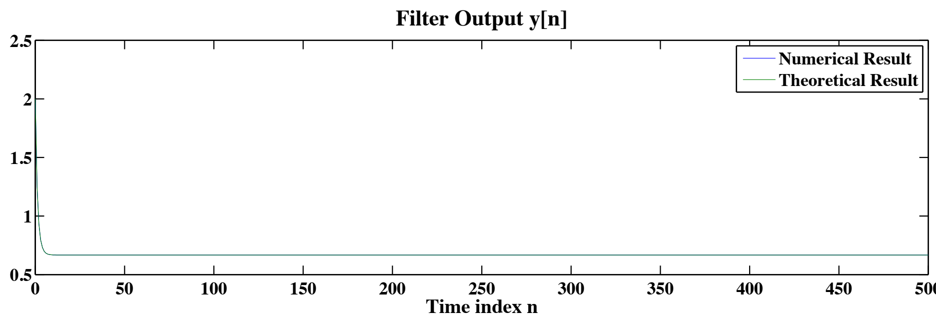


FIGURE 9.2: Numerical filter output $y[n]$ computed by `y=filterdf1(b,a,x,yi,xi)` compared to the theoretical output.

```
% P0904: Testing function y = filterdf1(b,a,x,yi,xi)
close all; clc
b = 1;
a = [1 -3/2 1/2];
n = 0:500;
%% Theoretical Result:
bt = [2 -9/4 1/2];
at = conv(a,[1 -1/4]);
[r p k] = residuez(bt,at);
ynt = r(1)*p(1).^n + r(2)*p(2).^n + r(3)*p(3).^n;
%% Numerical Result:
xn = (1/4).^n;
yi = [4 10];
yn = filterdf1(b,a,xn,yi);
%% plot:
hfa = figconfg('P0904a','long');
plot(n,yn,n,ynt)
xlabel('Time index n','fontsize',LFS)
title('Filter Output y[n]','fontsize',TFS)
legend('Numerical Result','Theoretical Result',...
'location','northeast')
colordef white;
```

5. (a) MATLAB function:

```
function y = filterdf1t(b,a,x)
% Implementation of Direct Form I structure (Transposed Form)
```

```
%  with initial conditions
% y = filterdf1t(b,a,x)
M = length(b)-1; N = length(a)-1; K = max(M,N);
a0 = a(1); a = reshape(a,1,N+1)/a0;
b = reshape(b,1,M+1)/a0; a = a(2:end);
Lx = length(x);
wn = zeros(K-1+Lx,1);
y = zeros(1,Lx);
for n = 1:Lx
    wn(K+n) = -a*wn(K+n-1:-1:K+n-N) + x(n);
    y(n) = b*wn(K+n:-1:K+n-M);
end
```

(b) See plot below.

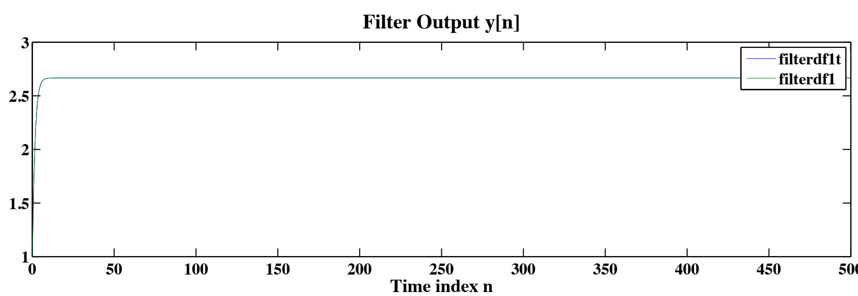


FIGURE 9.3: Numerical filter output $y[n]$ computed by `y=filterdf1(b,a,x,yi,xi)` compared to the output of `filterdf1` function.

MATLAB script:

```
% P0905: Testing function y = filterdf1t(b,a,x)
close all; clc
b = 1;
a = [1 -3/2 1/2];
n = 0:500;
%% Numerical Result:
xn = (1/4).^n;
yn = filterdf1t(b,a,xn);
yn_ref = filterdf1(b,a,xn); % reference
%% plot:
hfa = figconfg('P0905a','long');
```

```
colordef white;
plot(n,yn,n,yn_ref)
xlabel('Time index n','fontsize',LFS)
title('Filter Output y[n]','fontsize',TFS)
legend('filterdf1t','filterdf1','location','northeast')
```

6. (a) Solution:
   Repeat the scalar form equation as:

$$v_k[n] = v_{k+1}[n-1] - a_k y[n] + b_k x[n], \quad k = 1, \ldots, N-1. \tag{9.23b}$$

$$v_N[n] = b_N x[n] - a_N y[n] \tag{9.23c}$$

   By aligning the scalar equations into matrix form, it is trivial to prove the matrix equaiton.

   (b) Solution:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

   (c) MATLAB function:

```
function [v] = filteric(b,a,yic,xic)
% Computes direct form II initial conditions
% using initial conditions of direct form I
if nargin==4
    N = max([length(b)-1,length(a)-1,length(yic),length(xic)]);
    xic = [xic,zeros(N-length(yic))];
end
if nargin == 3
    N = max([length(b)-1,length(a)-1,length(yic)]);
    xic = zeros(1,N);
end
b = [b,zeros(N-length(b))];
a = [a,zeros(N-length(a))];
yic = [yic,zeros(N-length(yic))];
v = zeros(N,1);
A = diag(ones(1,N-1),1);
```

```
B = b(2:end)'; C = a(2:end)';
for n = 1:N
    v = A*v + B*xic(N-n+1) - C*yic(N-n+1);
end
```

(d) MATLAB script:

```
% P0906: Testing function v = filteric(b,a,yic,xic)
close all; clc
b = 1;
a = [1 -3/2 1/2];
yi = [4 10];
v = filteric(b,a,yi);
v_ref = filtic(b,a,yi);
```

7. (a) MATLAB function:

```
function y = filterfirlp(h,x)
% Implements the FIR linear-phase form given
% the impulse response
h = h(:)';
nh = length(h);
M = nh-1;
nx = length(x);
x = [zeros(1,M) x(:)'];
y = zeros(1,nx);
eo = mod(M,2) ~= 0;
if max(abs(h + fliplr(h))) == 0
    syasy = 1;
elseif max(abs(h - fliplr(h))) == 0
    syasy = 0;
else
    error('Impulse Response is not symmetric')
end
caseind = 2*syasy + eo;
switch caseind
    case 0
        MM = M/2;
        for n = 1:nx
            y(n) = (x(n+M:-1:n+M-MM+1)+x(n:1:n+MM-1))*h(1:MM)'...
                + h(MM+1)*x(n+M-MM);
        end
```

```
        case 1
            MM = (M-1)/2+1;
            for n = 1:nx
                y(n) = (x(n+M:-1:n+M-MM+1)+x(n:1:n+MM-1))*h(1:MM)';
            end
        case 2
            MM = M/2;
            for n = 1:nx
                y(n) = (x(n+M:-1:n+M-MM+1)-x(n:1:n+MM-1))*h(1:MM)';
            end
        case 3
            MM = (M-1)/2+1;
            for n = 1:nx
                y(n) = (x(n+M:-1:n+M-MM+1)-x(n:1:n+MM-1))*h(1:MM)';
            end
    end
```

(b) MATLAB script:

```
% P0907: Testing function y = filterfirlp(h,x)
close all; clc
n = 0:10;
xn = ones(size(n));
%% Part (a):
h = [1 2 3 2 1];
y = filterfirlp(h,xn);
y_ref = filter(h,1,xn);
max(abs(y-y_ref))

%% Part (b):
h = [1 -2 3 3 -2 1];
y = filterfirlp(h,xn);
y_ref = filter(h,1,xn);
max(abs(y-y_ref))

%% Part (c):
h = [1 -2 0 2 -1];
y = filterfirlp(h,xn);
y_ref = filter(h,1,xn);
max(abs(y-y_ref))
```

```
%% Part (d):
h = [1 -2 3 -3 2 -1];
y = filterfirlp(h,xn);
y_ref = filter(h,1,xn);
max(abs(y-y_ref))

%% Part (e):
h = [1 2 3 -2 -1];
y = filterfirlp(h,xn);
y_ref = filter(h,1,xn);
max(abs(y-y_ref))
```

8. (a) See graph below.
   (b) See graph below.
   (c) See graph below.
   (d) See graph below.

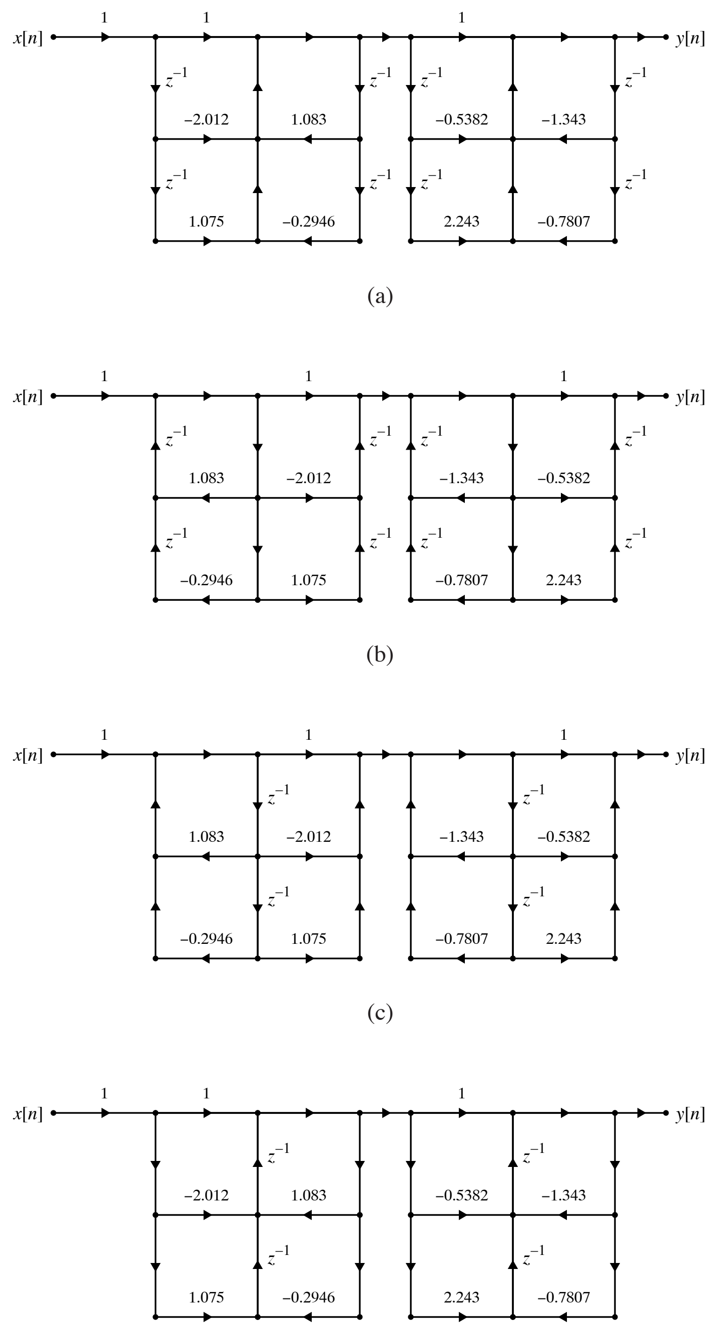
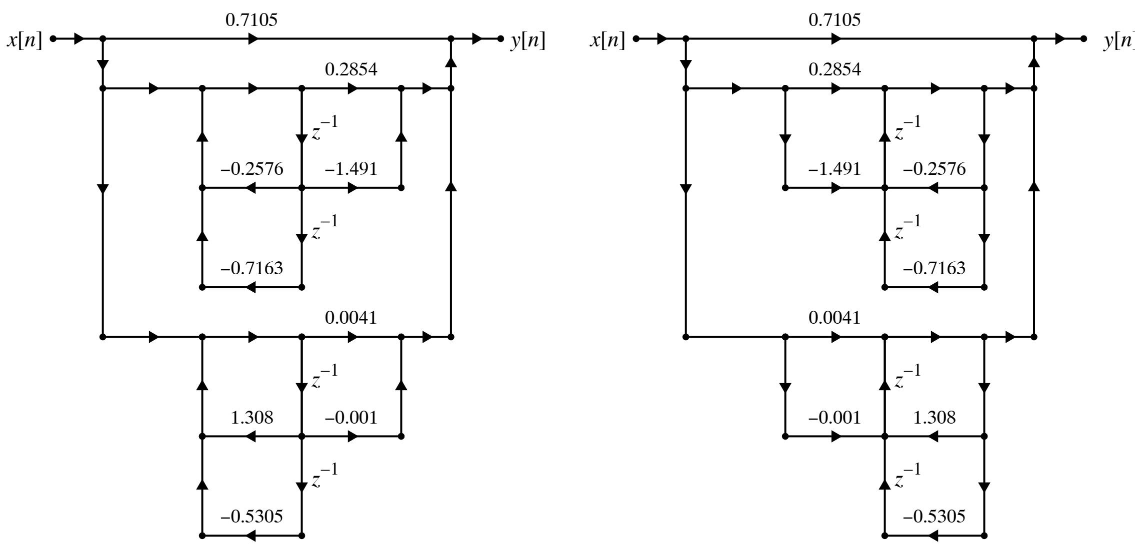


FIGURE 9.4: (a) Cascade form with second-order sections in normal direct form I. (b) Cascade form with second-order sections in transposed direct form I. (c) Cascade form with second-order sections in normal direct form II. (d) Cascade form with second-order sections in transposed direct form II.

9. MATLAB script:

```
% P0909: Draw the following parallel form
%        with second-order section in direct form II
close all; clc
b = [1 -2.61 2.75 -1.36 0.27];
a = [1 -1.05 0.91 -0.8 0.38];
[r p k] = residuez(b,a);
[B1 A1] = residuez(r(1:2),p(1:2),[]);
B1 = real(B1)
A1 = real(A1)
[B2 A2] = residuez(r(3:4),p(3:4),[]);
B2 = real(B2)
A2 = real(A2)
```



(a) (b)

FIGURE 9.5: (a) Parallel form structure with second-order section in direct form II normal. (b) Parallel form structure with second-order section in direct form II transposed.

10. (a) See graph below.
    (b) See graph below.
    (c) See graph below.
    MATLAB script:

```
% P0910: Draw the following structures
close all; clc
g = 4.32;
sos = [1 2.39 2.17 1 -0.91 0.28;
    1 -0.33 1.32 1 -1.5 0.69;
    1 0 0 1 0.2 0];
[b a] = sos2tf(sos,g);
%% Parllel with transposed second-order sections
[r p k] = residuez(b,a);
[B1 A1] = residuez(r(1:2),p(1:2),[]);
B1 = real(B1)
A1 = real(A1)
[B2 A2] = residuez(r(3:4),p(3:4),[]);
B2 = real(B2)
A2 = real(A2)
B3 = [r(end) 0]
A3 = [1 -p(end) 0]
```

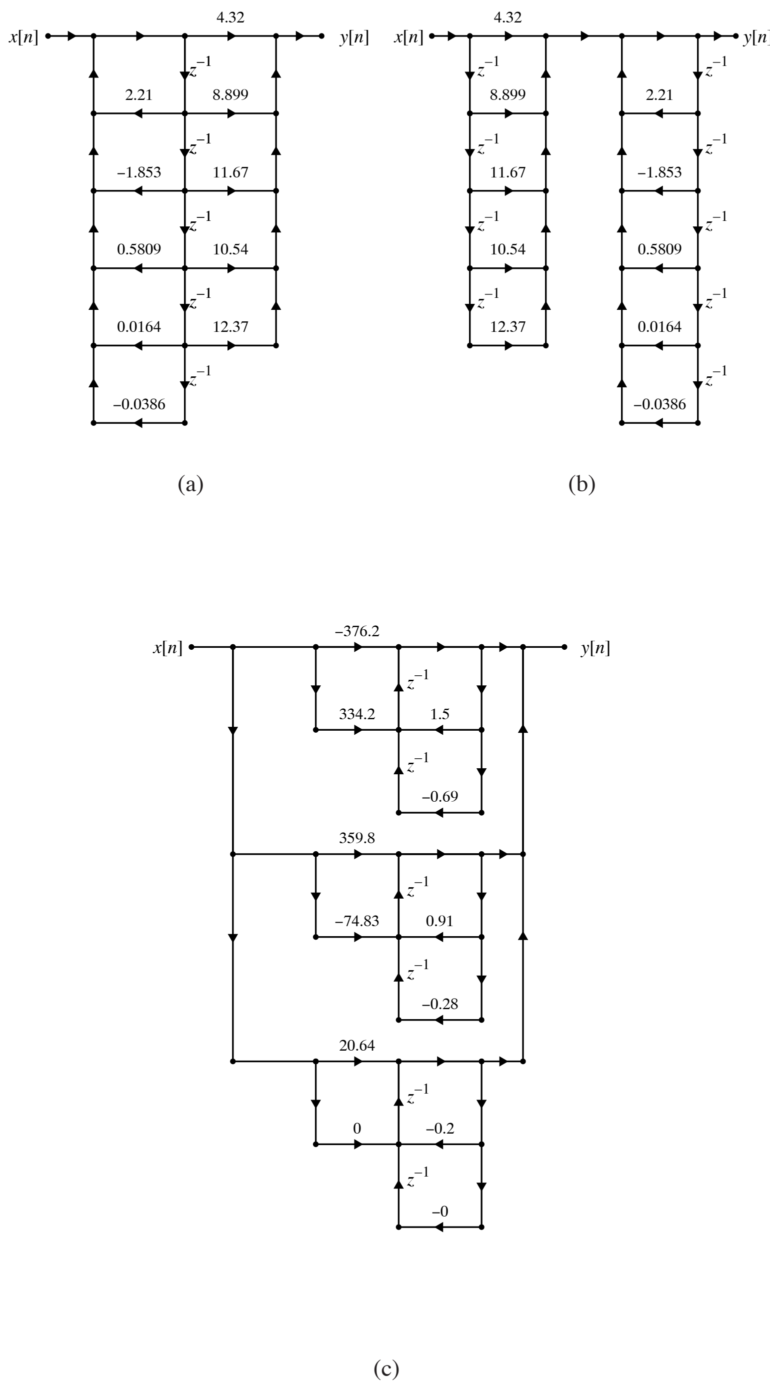


FIGURE 9.6: (a) Direct form II (normal). (b) Direct form I (normal). (c) Parallel form with transposed second-order sections

11. (a) See graph below.
    (b) See graph below.
    (c) See graph below.
    (d) tba.

    MATLAB script:

```
% P0911: Draw FIR structures
close all; clc
b = [1.9 3.94 4.72 2.71 0.61];
%% Cascade form:
[sos g] = tf2sos(b,1);
Draw_FIR_CF_Normal(g,sos(:,1:3))
%% Frequency-sampling form:
[C,B,A] = dir2fs(b);
%% Lattice form:
```

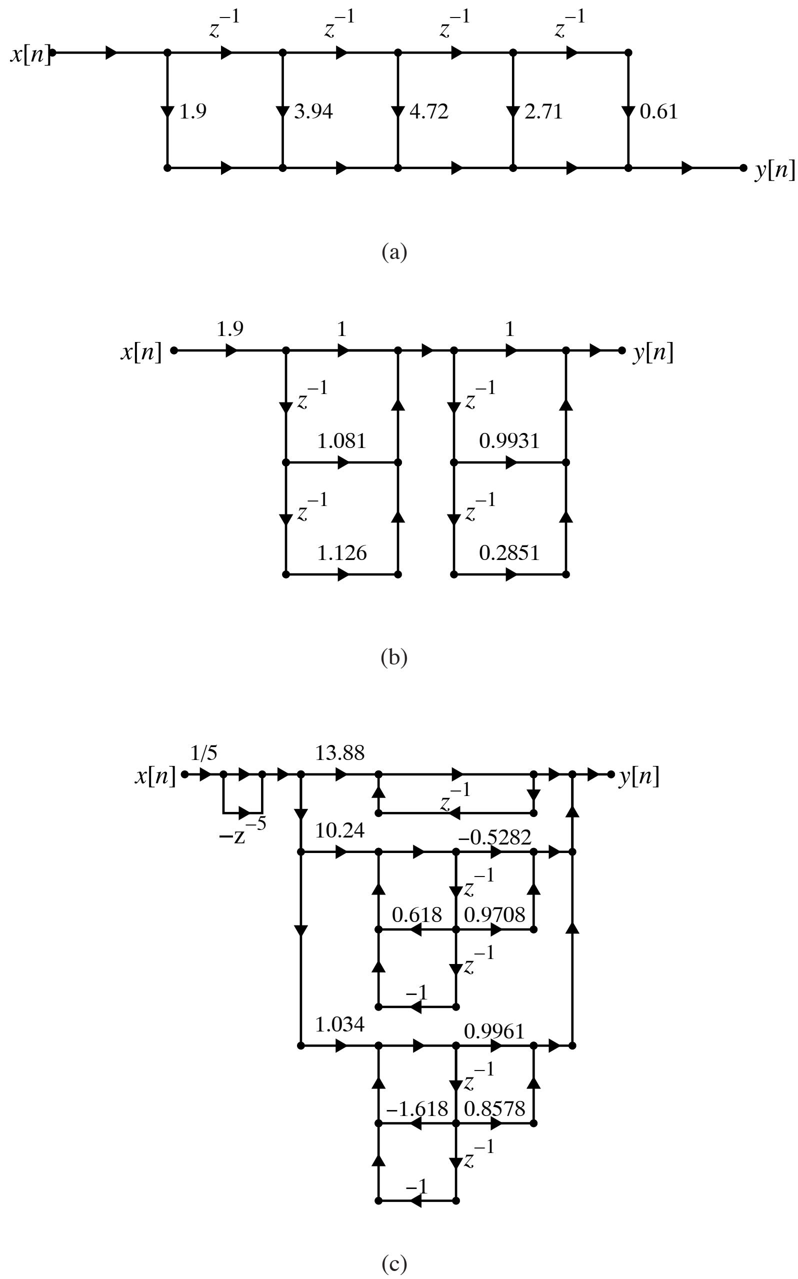
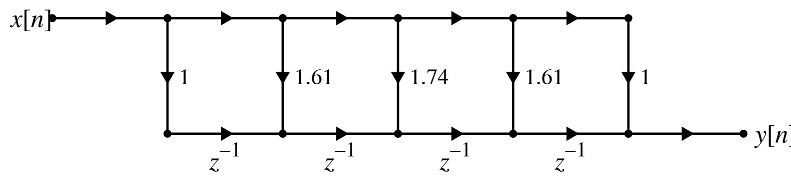


FIGURE 9.7: (a) Direct form (normal). (b) Cascade form. (c) Frequency-sampling form.

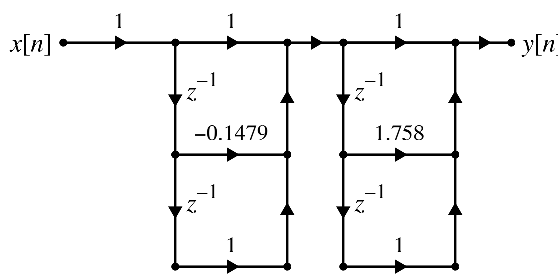
12. (a) See graph below.
    (b) See graph below.
    (c) See graph below.
    (d) See graph below.
    (e) tba.

    MATLAB script:

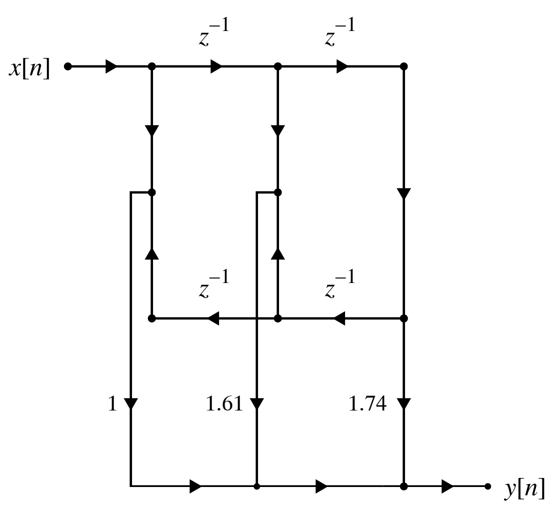
```
% P0912: Draw FIR structures
close all; clc
b = [1 1.61 1.74 1.61 1];
%% Cascade form:
[sos g] = tf2sos(b,1);
%% Frequency-sampling form:
[C,B,A] = dir2fs(b);
%% Lattice form:
```



(a)



(b)



(c)

FIGURE 9.8: (a) Direct form (normal). (b) Cascade form. (c) Linear-phase form.

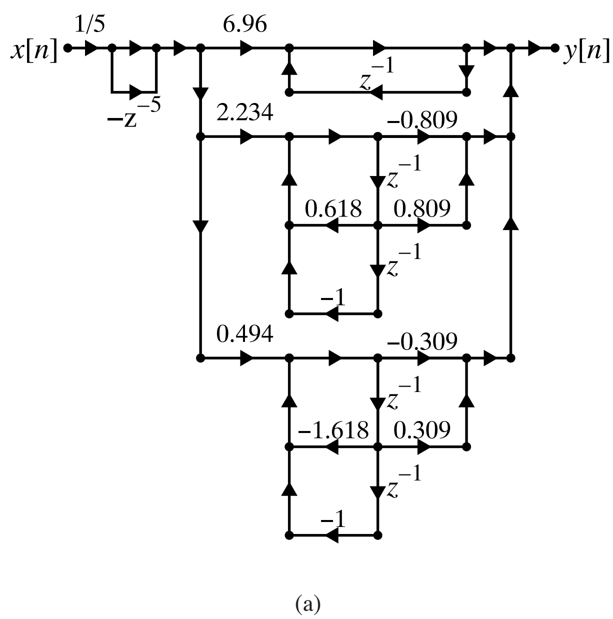


FIGURE 9.9: (a) Frequency-sampling form.

13. (a) Proof:

Repeat the equations as follows:

$$H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H[k]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi k}{N}}},\quad H[k]=H(z)|_{z=\mathrm{e}^{\mathrm{j}\frac{2\pi k}{N}}} \tag{9.50}$$

$$H(z)=\frac{1-z^{-N}}{N}\left\{\frac{H[0]}{1-z^{-1}}+\frac{H[\frac{N}{2}]}{1+z^{-1}}+\sum_{k=1}^{K}2|H[k]|H_k(z)\right\} \tag{9.51}$$

$$H_k(z)=\frac{\cos(\angle H[k])-z^{-1}\cos(\angle H[k]-\frac{2\pi k}{N})}{1-2\cos(\frac{2\pi k}{N})z^{-1}+z^{-2}} \tag{9.52}$$

where $K=N/2-1$ if N is even or $k=(N-1)/2$ if N odd.

From equation (9.50), we have

$$\begin{aligned}H(z)=\frac{1-z^{-N}}{N}&\left\{\frac{H[0]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}0}}+\frac{H[N/2]}{1+z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}\frac{N}{2}}}\right.\\&\left.+\sum_{k=1}^{K}\left(\frac{H[k]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k}}+\frac{H[N-k]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}(N-k)}}\right)\right\}\end{aligned}$$

Since we have

$$|H[N-k]|=|H[k]|,\quad \angle H[N-k]=\angle H[k]$$

$$\begin{aligned}&\frac{H[k]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k}}+\frac{H[N-k]}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}(N-k)}}=\frac{|H[k]|\mathrm{e}^{\mathrm{j}\angle H[k]}}{1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k}}+\frac{|H[k]|\mathrm{e}^{-\mathrm{j}\angle H[k]}}{1-z^{-1}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}k}}\\&=\frac{|H[k]|(\mathrm{e}^{\mathrm{j}\angle H[k]}+\mathrm{e}^{-\mathrm{j}\angle H[k]}-\mathrm{e}^{\mathrm{j}(\angle H[k]-\frac{2\pi k}{N})}-\mathrm{e}^{-\mathrm{j}(\angle H[k]-\frac{2\pi k}{N})})}{(1-z^{-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k})(1-z^{-1}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}k})}\\&=\frac{2|H[k]|(\cos(\angle H[k])-z^{-1}\cos(\angle H[k]-\frac{2\pi k}{N}))}{1-2\cos(\frac{2\pi k}{N})z^{-1}+z^{-2}}\end{aligned}$$

Thus, the system function can be proved as

$$H(z)=\frac{1-z^{-N}}{N}\left\{\frac{H[0]}{1-z^{-1}}+\frac{H[\frac{N}{2}]}{1+z^{-1}}+\sum_{k=1}^{K}2|H[k]|H_k(z)\right\}$$

where

$$H_k(z)=\frac{\cos(\angle H[k])-z^{-1}\cos(\angle H[k]-\frac{2\pi k}{N})}{1-2\cos(\frac{2\pi k}{N})z^{-1}+z^{-2}}$$

(b) MATLAB function:

```
function [G,sos] = firdf2fs(h)
% Convert FIR impulse response h into frequency-sampling
% implementation
N = length(h);
if mod(N,2) == 0
    K = N/2-1;
else
    K = (N-1)/2;
end
G = zeros(K+2,1);
H = fft(h);
Hmag = abs(H);
Hang = angle(H);
G(1) = H(1);
G(3:end) = 2*Hmag(2:1+K);
sos = zeros(K+2,6);
sos(1,:) = [1 0 0 1 -1 0];
sos(2,:) = [1 0 0 1 1 0];
for ii = 1:K
    sos(2+ii,:) = [cos(Hang(ii+1)) -cos(Hang(ii+1)-2*pi*ii/N) 0 ...
        1 -2*cos(2*pi*ii/N) 1];
end
if mod(N,2) == 0
    G(2) = H(N/2+1);
else
    G(2) = [];
end
```

(c) MATLAB script:

```
% P0913: Testing [G,sos] = firdf2fs(h)
close all; clc
N = 33; alpha = (N-1)/2; k = 0:N-1;
magHk = [1,1,1,0.5,zeros(1,26),0.5,1,1];
angHk = -32*pi*k/33;
H = magHk.*exp(1j*angHk);
h = real(ifft(H,N));
[G,sos] = firdf2fs(h);
```

14. tba