EECE5666 (DSP) : Homework-5 Solutions

Default Plot Parameters

```
set(0,'defaultfigurepaperunits','points','defaultfigureunits','points');
set(0,'defaultaxesfontsize',10); set(0,'defaultaxeslinewidth',1.5);
set(0,'defaultaxestitlefontsize',1.4,'defaultaxeslabelfontsize',1.2);
```

Problem 5.1

Text Problem 8.16 (Page 477)

Consider the inverse DFT given in the textbook (8.2) and repeated below:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1 \qquad (5.1.1)$$

(a) Show that $(5.1.1)$ can also be written as

$$x[n] = \frac{1}{N}\mathrm{j}\left\{\sum_{k=0}^{N-1} (\mathrm{j}X^*[k])W_N^{kn}\right\}^*, \quad n = 0, 1, \ldots, N-1 \qquad (5.1.2)$$

Proof: The IDFT in $(5.1.1)$ can be expressed as

$$\begin{aligned} x[n] &= \frac{1}{N}(1)\sum_{n=0}^{N-1} X[k]W_N^{-kn} = \frac{1}{N}(\mathrm{j}(-\mathrm{j}))\sum_{n=0}^{N-1} X[k]W_N^{-kn} = \frac{1}{N}\mathrm{j}\left(\sum_{n=0}^{N-1}(-\mathrm{j}X[k])W_N^{-kn}\right) \\ &= \frac{1}{N}\mathrm{j}\left\{\sum_{k=0}^{N-1}(\mathrm{j}X^*[k])W_N^{kn}\right\}^*. \quad n = 1, 2, \ldots, N-1. \end{aligned}$$

which completes the proof.

(b) The quantity inside the curly brackets in $(5.1.2)$ is the DFT $y[n]$ of the sequence $\mathrm{j}X^*[k]$; thus, the inverse DFT of $X[k]$ is $x[n] = (1/N)(\mathrm{j}y^*[n])$. Note that if $c = a + \mathrm{j}b$ then $\mathrm{j}c^* = b + \mathrm{j}a$. Using this interpretation, draw a block diagram that computes IDFT using a DFT block that has separate real and imaginary input/output ports.

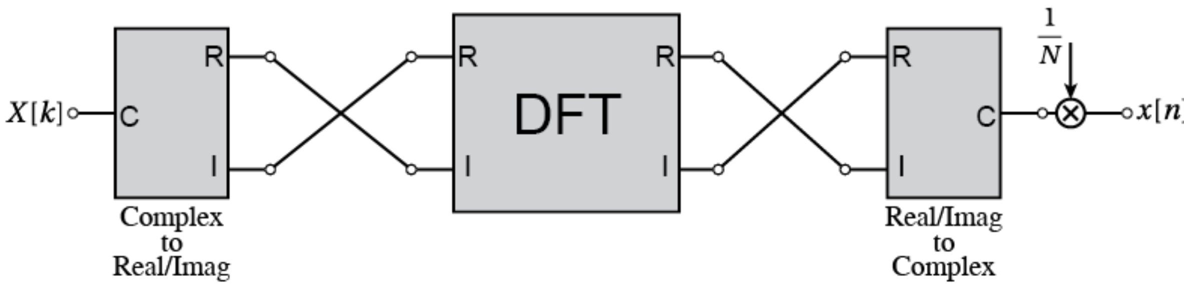
Solution: Let $X[k] = X_R[k] + \mathrm{j}X_I[k]$. Then

$$\mathrm{j}X^*[k] = X_I[k] + \mathrm{j}X_R[k] \qquad (5.1.3)$$

which forms the input to the DFT block. The output of the DFT box is $y[n] = y_R[n] + \mathrm{j}y_I[n]$. Finally,

$$x[n] = \frac{1}{N} \mathrm{j} y^*[n] = \frac{1}{N}\left(y_{\mathrm{I}}[n] + \mathrm{j} y_{\mathrm{R}}[n]\right). \qquad (5.1.4)$$

Thus using 5.1.3) and (5.1.4) we can draw the following block diagram.



(c) Develop a MATLAB function **x = idft(X,N)** using the **fft** function. Verify your function on signal $x[n] = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

MATLAB function: Enter your **idft** function code below in the code example area for the TA to analyze and grade it. Create your **idft** function at the end of this file.

```
function x = idft(X,N)
    % Compute an N-point idft x[n] of X[k] using the fft function according
    % to block diagram
    %

    X = X(:).'; Nx = length(X);
    if nargin == 1
        N = Nx;
    elseif N <= Nx
        X = X(1:N);
    else
        X = [X zeros(1,N-Nx)];
    end
    XR = real(X); XI = imag(X);
    Y = XI+1j*XR; y = fft(Y,N);
    yR = real(y); yI = imag(y);
    x = (1/N)*(yI+1j*yR);
end
```

MATLAB script for verification: $x[n] = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

```
clc; close all; clear;
x = 1:8; X = fft(x);
xidft = idft(X); display(xidft)
```

```
xidft = 1×8
     1     2     3     4     5     6     7     8
```

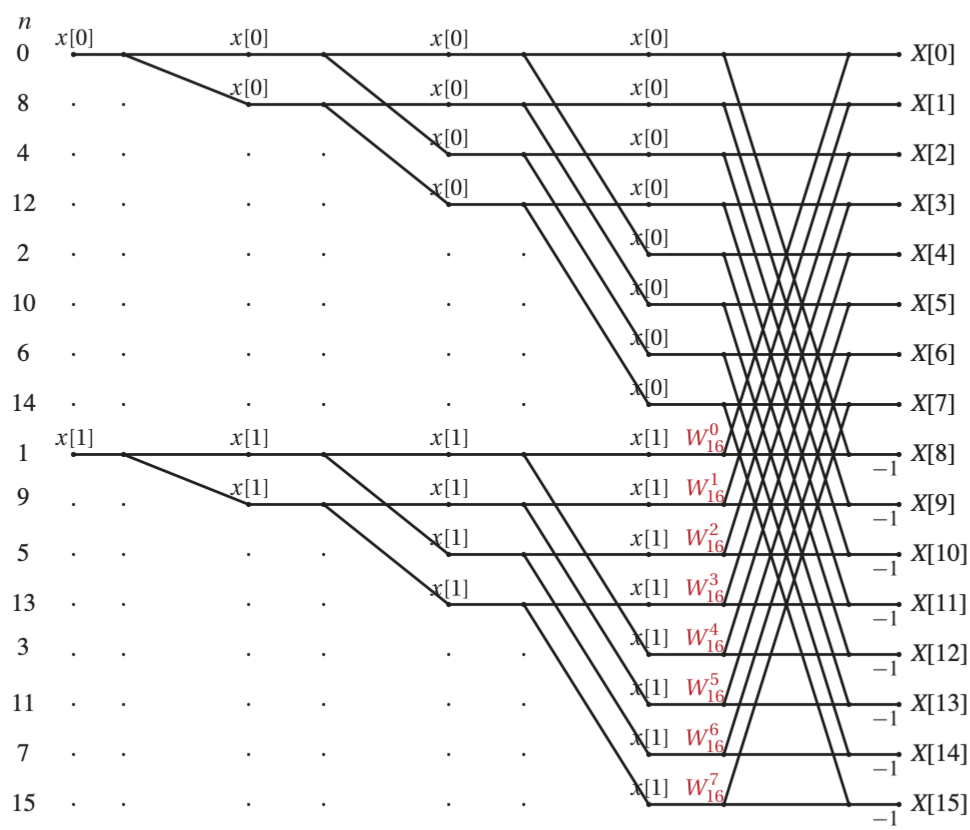
Problem 5.2

Text Problem 8.29 (Page 479)

Let the sequence $x[n]$ be of length L and we wish to compute an N-point DFT of $x[n]$ where $L \ll N$. Assume that the first $L = 2$ signal values $x[0]$ and $x[1]$ are non-zero.

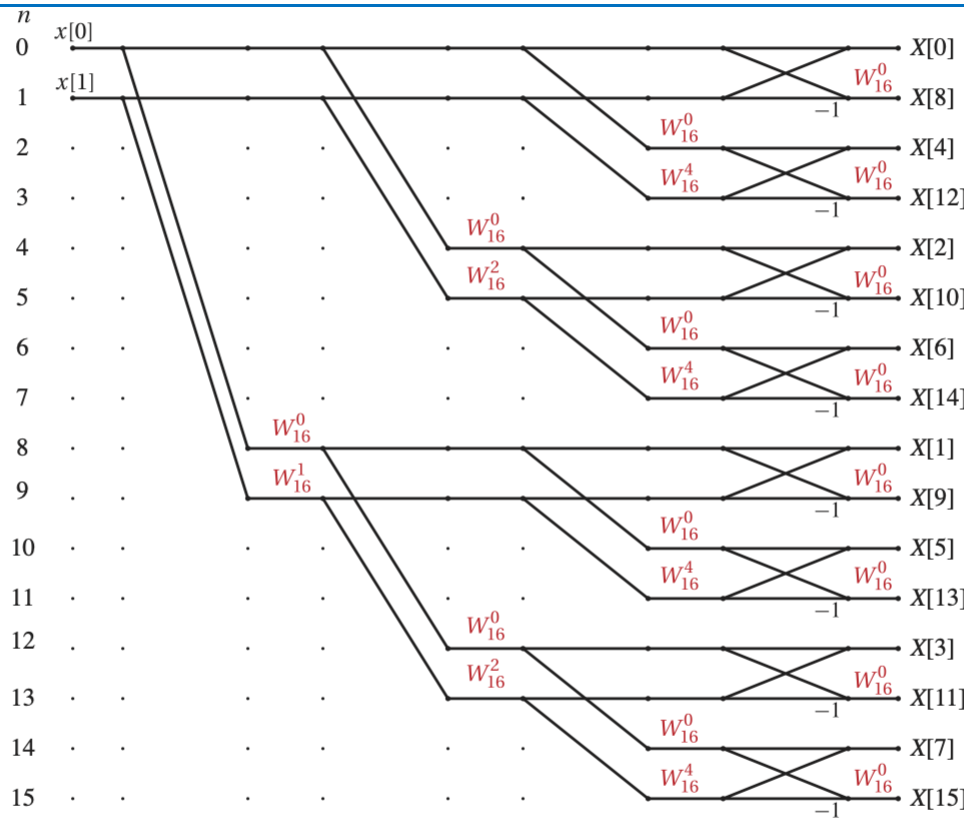
(a) Draw a radix-2 $N = 16$-point DIT-FFT flow-chart in which only those paths originating from the non-zero signal values are retained.

Solution: The DIT-FFT flowchart showing the requisite butterflies is shown below.



(b) Draw a radix-2 $N = 16$-point DIF-FFT flow-chart in which only those paths originating from the non-zero signal values are retained.

Solution: The DIF-FFT flowchart containing the requisite butterflies is shown below.



(c) Determine the total number of complex multiplications in each of the above flow-graphs. Which algorithm gives the fewer number of multiplications? Assume that $W_{16}^0 = 1 + \mathrm{j}0$ is stored as a complex number.

Solution: In part (a), the total number of complex multiplications is 8. For part (b), the total number of complex multiplications is 22. This calculation assumes that $W_{16}^0 = 1 + \jmath 0$ is stored as a complex number.

(d) Develop a general rule in terms of $L = 2^\ell$ and $N = 2^\nu$ for selecting DIT- or DIF-FFT algorithm in FFT. This approach is called ***input pruning***.

Solution: We assume that $N = 2^\nu$, $L = 2^\ell$ and that the first L samples are non-zero.

- **DIT-FFT**: From the figure in part (a) we note that, in the case of DIT-FFT flow graph, only $\ell = 1$ stage (the last one) requires the complete set of butterflies. If $\ell = 2$ or $L = 4$, it is easy to see that the last two stages would require a complete set. Furthermore, the remaining $\nu - \ell$ stages need only copying the input sample (data) values up to the stage with full set. Thus in the DIT-FFT input pruning, the number of complex multiplications are $\ell N/2$, since each full stage requires $N/2$ complex multiplications.
- **DIF-FFT**: The general situation in DIF-FFT is more complicated. From the figure in part (b), we note that $\ell = 1$ stage (the last one) is full but has only unity multipliers. In the remaining $\nu - \ell$ stages, we have geometrically increasing (two, four , eight, etc.) complex multiplications. Thus the total complex multiplications in the DIF-FFT input pruning is always higher that those in DIT-FFT. The DIF-FFT has advantages in output pruning since this flow graph is similar to the DIT-FFT input pruning.

Problem 5.3

Text Problem 8.35 (Page 480)

Suppose we need any $K \leq N$ DFT values of the N-point DFT. We have two choices: the direct approach or the radix-2 DIT-FFT algorithm. At what minimum value of K, the FFT algorithm will become more efficient than the direct approach? Determine these minimum values for $N = 128$, 1024, and 8192.

Note: Compare the computation complexity using number of complex multiplications.

Solution: To compute K DFT values using direct form would require KN complex multiplications while the radix-2 DIT-FFT algorithm would require $\frac{1}{2}N\log_2 N$ complex multiplications. Note that any FFT algorithm computes all DFT values in one block calculation. Hence, the minimum values K is computed as follow:

$$KN \underset{\text{DIR}}{\overset{\text{DFT}}{\gtrless}} \tfrac{1}{2}N\log_2 N \Rightarrow K_{\min} = \left\lfloor \tfrac{1}{2}\log_2 N \right\rfloor.$$

Thus, for $N = 128$, 1024, and 8192, $K_{\min}$ is 3, 5, and 6, respectively.

Problem 5.4

Text Problem 8.38 (Page 480)

Consider a 6-point DIF-FFT that uses a mixed-radix implementation. There are two approaches.

(a) In the first approach, combine two inputs in three sequences and take 3-point DFTs to obtain the 6-point DFT. Draw a flow-graph of this approach and properly label all relevant path gains as well as input/output nodes. How many real multiplications and additions are needed? Assume that signals in general are complex-valued and hence multiplication and addition operations are also complex valued.

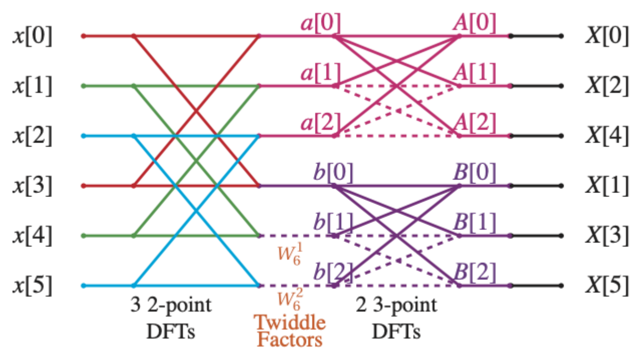
Solution: In this case we first compute three 2-point DFTs and then combine these into two 3-point DFTs to obtain the requisit 6-point DFT. Towards this we consider even- and odd-ordered DFT coefficients as follows:

$$\begin{aligned}
X[2k] &= \sum_{n=0}^{5} x[n]W_6^{2nk} = \sum_{n=0}^{5} x[n]W_3^{nk}, \quad k = 0,1,2 \\
&= \sum_{n=0}^{2} x[n]W_3^{nk} + \sum_{k=3}^{5} x[n]W_3^{nk} = \sum_{n=0}^{2} x[n]W_3^{nk} + \sum_{k=0}^{2} x[n+3]W_3^{(n+3)k}, k = 0,1,2 \\
A[k] &\triangleq \sum_{n=0}^{2} \Big(\underbrace{x[n] + x[n+3]}_{a[n],n=0,1,2}\Big) W_3^{nk}, \quad k = 0,1,2
\end{aligned} \tag{5.4.1}$$

Similarly, we have

$$
\begin{aligned}
X[2k+1] &= \sum_{n=0}^{5} x[n]W_6^{(2k+1)n} = \sum_{n=0}^{5} x[n]W_6^{n}W_3^{nk}, \quad k=0,1,2 \\
&= \sum_{n=0}^{2} x[n]W_6^{n}W_3^{nk} + \sum_{k=3}^{5} x[n]W_6^{n}W_3^{nk}, k=0,1,2 \\
&= \sum_{n=0}^{2} x[n]W_6^{n}W_3^{nk} + \sum_{k=0}^{2} x[n+3]W_6^{n+3}W_3^{(n+3)k}, k=0,1,2 \\
B[k] &\triangleq \sum_{n=0}^{2} \Big\{ \underbrace{\big(x[n]-x[n+3]\big)W_6^{n}}_{b[n],n=0,1,2} \Big\} W_3^{nk}, \quad k=0,1,2 \qquad (5.4.2)
\end{aligned}
$$

Thus the 3-point DFTs of sequences $a[n]$ and $b[n]$ in $(5.4.1)$ and $(5.4.2)$ result in a decimated set of DFT coefficients. The resulting flow graph is shown below.



To determine the computational complexity we will assume that signals in general may be complex-valued and hence the multiplication and addition operations are also complex valued. Now one complex multiplication requires 4 real multiplications and 2 real additions while a complex addition requires 2 real additions. From the flow graph above, we have three 2-point DFTs, followed by two twiddle factors, and finally two 3-point DFTs.

- Three 2-point DFTs: Each 2-point DFT has two complex-additions or four real additions for a total of 12 real additions.
- Two twiddle factors: These require two complex multiplications or 8 real multiplications and 4 real additions.
- Two 3-point DFTs: Each 3-point DFT requires four complex multiplications and six complex additions or 16 real multiplications and $8+12=20$ real additions for a total of 32 real multiplications and 40 real additions.

Hence the total computational complexity is

$$\text{Real-multiplications } = 8+32 = 40, \qquad \text{Real-additions } = 12+4+40 = 56.$$

(b) In the second approach combine three inputs in two sequences and take 2-point DFTs to obtain the 6-point DFT. Draw a flow-graph of this approach and properly label all relevant path gains as well as input/output nodes. How many real multiplications and additions are needed? Again, assume that signals in general are complex-valued and hence multiplication and addition operations are also complex valued.

Solution: In this case we first compute two 3-point DFTs and then combine these into three 2-point DFTs to obtain the 6-point DFT. Towards this we consider every third DFT coefficient as follows:

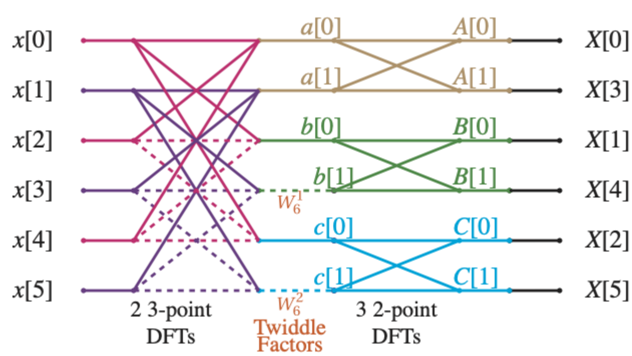
$$\begin{aligned} X[3k] &= \sum_{n=0}^{5} x[n]W_6^{n(3k)} = \sum_{n=0}^{5} x[n]W_2^{nk}, \quad k=0,1 \\ A[k] &\triangleq \sum_{n=0}^{1}\Big(\underbrace{x[n]+x[n+2]+x[n+4]}_{a[n],n=0,1}\Big)W_2^{nk}, \quad k=0,1 \quad (5.4.3) \end{aligned}$$

Similarly we obtain

$$B[k] \triangleq X[3k+1] = \sum_{n=0}^{1}\Big\{\underbrace{\big(x[n]+x[n+2]W_3+x[n+4]W_3^2\big)W_6^n}_{b[n],n=0,1}\Big\}W_2^{nk}, \quad k=0,1 \qquad (5.4.4)$$

$$C[k] \triangleq X[3k+2] = \sum_{n=0}^{1}\Big\{\underbrace{\big(x[n]+x[n+2]W_3^2+x[n+4]W_3\big)W_6^n}_{c[n],n=0,1}\Big\}W_2^{nk}, \quad k=0,1 \qquad (5.4.5)$$

Once again the 2-point DFTs of sequences $a[n]$, $b[n]$, and $c[n]$ in $(5.4.3) - (5.4.5)$ result in a decimated set of DFT coefficients. The resulting flow graph is shown below from which we note that it requires two 3-point DFTs, followed by two twiddle factors, and finally three 2-point DFTs.



- <u>Two 3-point DFTs</u>: Each 3-point DFT requires four complex multiplications and six complex additions or 16 real multiplications and $8+12=20$ real additions for a total of 32 real multiplications and 40 real additions.
- <u>Two twiddle factors</u>: These require two complex multiplications or 8 real multiplications and 4 real additions.
- <u>Three 2-point DFTs</u>: Each 2-point DFT has two complex-additions or four real additions for a total of 12 real additions.

Hence the total computational complexity is

$$\text{Real-multiplications } = 32+8=40, \qquad \text{Real-additions } = 40+4+12=56.$$

Problem 5.5

Text Problem 9.19 parts (a) and (c) only (Page 531)

A discrete-time system is given by

$$H(z) = \frac{1 - 3.39z^{-1} + 5.76z^{-2} - 6.23z^{-3} + 3.25z^{-4}}{1 + 1.32z^{-1} + 0.63z^{-2} + 0.4z^{-3} + 0.25z^{-4}}.$$

Determine and draw each of the following structures.

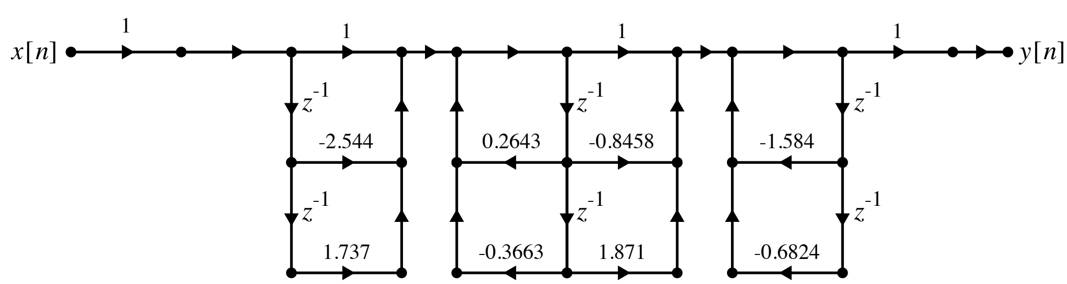
(a) A cascade form with second-order sections in normal direct form I

Solution: The cascade form coefficients are computed using the following script.

```
clc; close all; clear;
b = [1 -3.39 5.76 -6.23 3.25];
a = [1 1.32 0.63 0.4 0.25];
[sos,G] = tf2sos(b,a); sos, G
```

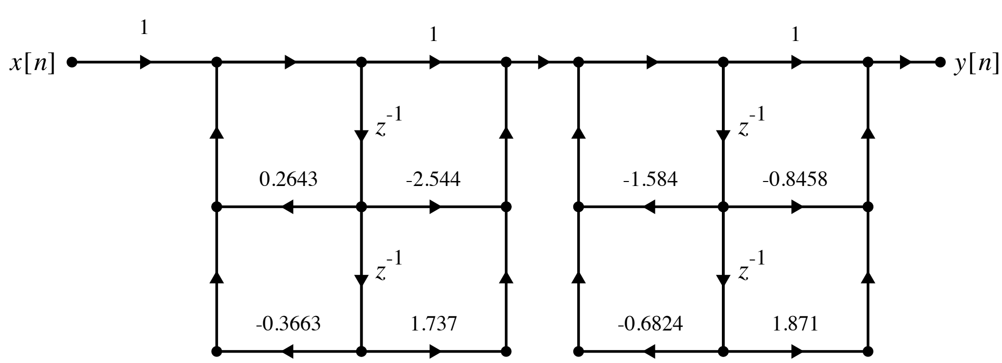
```
sos = 2×6
     1.0000   -2.5442    1.7367    1.0000   -0.2643    0.3663
     1.0000   -0.8458    1.8714    1.0000    1.5843    0.6824
G = 1
```

The signal flow graph is shown below.



(b) A cascade form with second-order sections in normal direct form II

Solution: The above computed cascade form coefficients are still applicable in this part. The signal flow graph is shown below.



Problem 5.6

Text Problem 9.23 (Page 532)

An IIR system is given by

$$H(z) = \frac{376.63 - 89.05z^{-1}}{1 - 0.91z^{-1} + 0.28z^{-2}} + \frac{-393.11 + 364.4z^{-1}}{1 - 1.52z^{-1} + 0.69z^{-2}} + \frac{20.8}{1 + 0.2z^{-1}}.$$

Determine and draw the following structures.

(a) Direct form II (normal)

Solution: We have to first convert the above parallel form into a rational function form for to obtain direct form coefficients. This calculation is done using the following script.

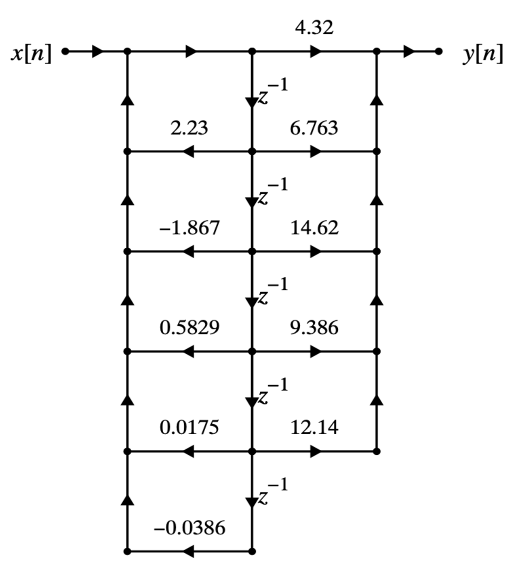
```
clc; close all; clear;
b1 = [376.63 -89.05]; b2 = [-393.11 364.4]; b3 = 20.8;
a1 = [1 -0.91 0.28]; a2 = [1 -1.52 0.69]; a3 = [1 0.2];
[r1,p1,k1] = residuez(b1,a1); [r2,p2,k2] = residuez(b2,a2);
[r3,p3,k3] = residuez(b3,a3);
r = [r1;r2;r3]; p = [p1;p2;p3]; k = [k1;k2;k3];
[b,a] = residuez(r,p,k); real(b), real(a)
```

```
ans = 1×5
    4.3200    6.7625   14.6230    9.3859   12.1361
ans = 1×6
    1.0000   -2.2300    1.8672   -0.5829   -0.0175    0.0386
```

Hence, the rational function form is

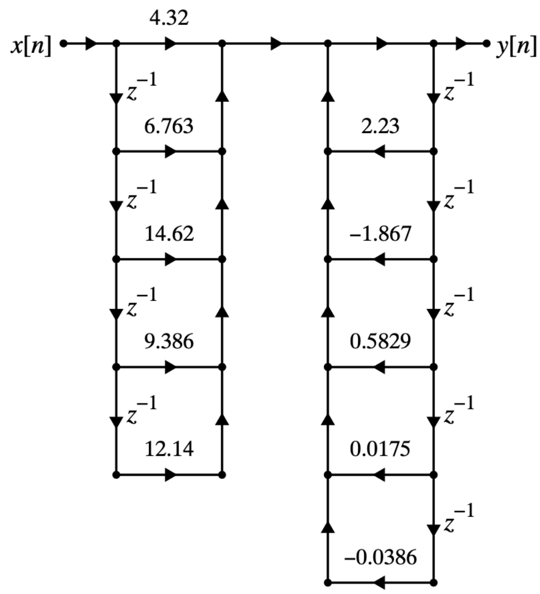
$$H(z) = \frac{4.32 + 6.7625z^{-1} + 14.623z^{-2} + 9.3859z^{-3} + 12.1361z^{-4}}{1 - 2.23z^{-1} + 1.8672z^{-2} - 0.5829z^{-3} - 0.0175z^{-4} + 0.0386z^{-5}}.$$

The resulting signal flow graph is shown below.



(b) Direct form I (normal)

Solution: The above computed direct form coefficients are still applicable in this part. The signal flow graph is shown below.



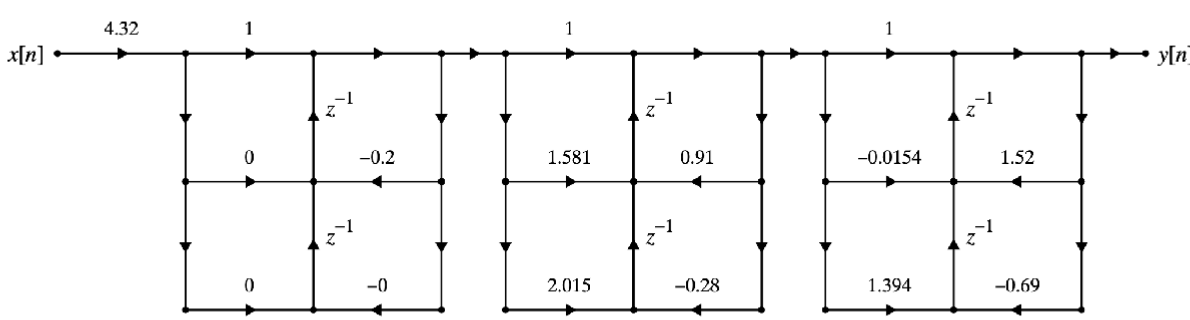
(c) Cascade form with transposed second-order sections

Solution: We convert the direct form into cascade form containing second-order sections using the following script.

```
[sos,G] = tf2sos(b,a); sos, G
```

```
sos = 3×6
    1.0000         0         0    1.0000    0.2000         0
    1.0000    1.5808    2.0152    1.0000   -0.9100    0.2800
    1.0000   -0.0154    1.3940    1.0000   -1.5200    0.6900
G = 4.3200
```

The signal flow graph is shown below.



Problem 5.7

Text Problem 9.26 (Page 532)

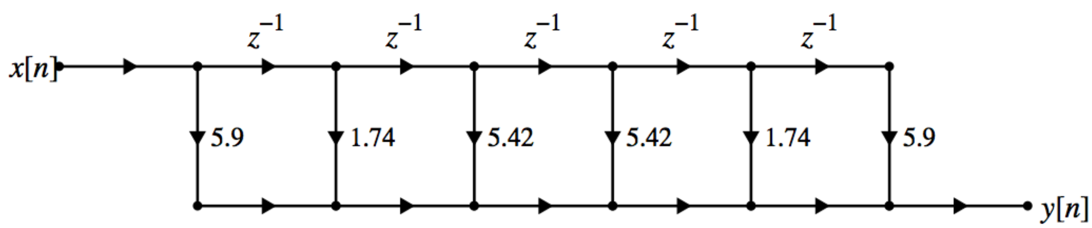
A discrete time system is described by the difference equation

$$y[n] = 5.9x[n] + 1.74x[n-1] + 5.42x[n-2] + 5.42x[n-3] + 1.74x[n-4] + 5.9x[n-5]. \qquad (5.7.1)$$

Determine and draw the following structures.

(a) Direct form

Solution: In this structure we implement the difference equation (1) as given using signal flow graph elements which is shown below.



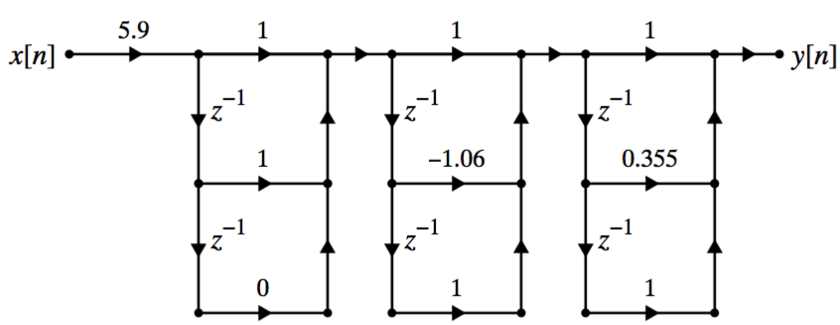
(b) Cascade form

Solution: Using the following script we compute second-order section coefficients.

```
clc; close all; clear;
b = [5.9,1.74,5.42,5.42,1.74,5.9];
[sos,G] = tf2sos(b,1)
```

```
sos = 3×6
     1.0000    1.0000         0    1.0000         0         0
     1.0000   -1.0600    1.0000    1.0000         0         0
     1.0000    0.3550    1.0000    1.0000         0         0
G = 5.9000
```

The resulting signal flow graph is shown below.

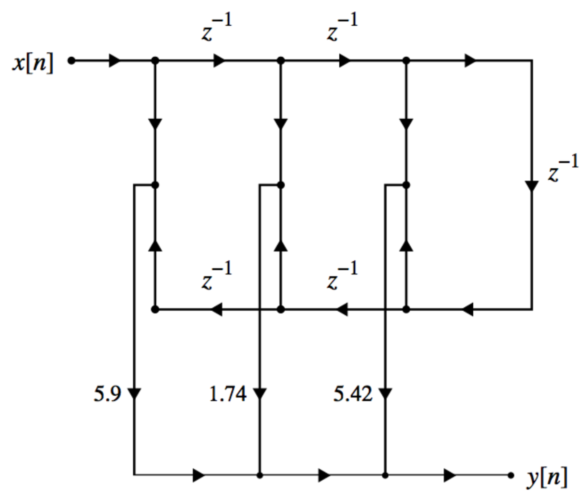


(c) Linear-Phase Form

Solution: From the given difference equation $(5.7.1)$ the impulse response is

$$h[n] = \{\underset{\uparrow}{5.9}, 1.74, 5.42, 5.42, 1.74, 5.9\}$$

which has an even symmetry with respect to $n = 2.5$. The resulting signal flow graph is shown below.



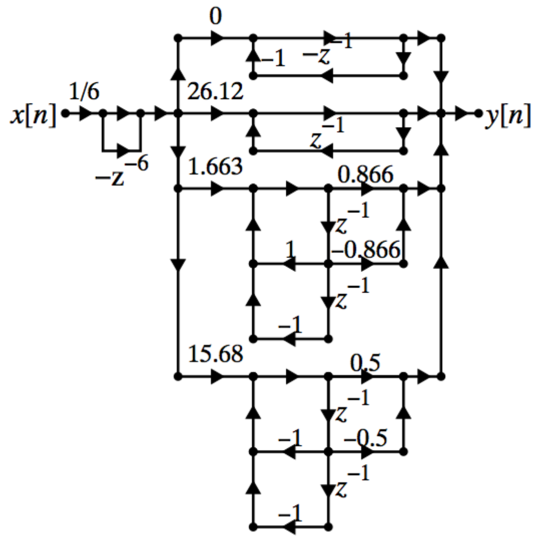
(c) Frequency sampling form

Solution: We determine this structure using the **`dir2fs`** function.

```
[C,B,A] = dir2fs(b)
```

```
C = 4×1
     1.6628
    15.6800
    26.1200
          0
B = 2×2
     0.8660   -0.8660
     0.5000   -0.5000
A = 4×3
     1.0000   -1.0000    1.0000
     1.0000    1.0000    1.0000
     1.0000   -1.0000         0
     1.0000    1.0000         0
```

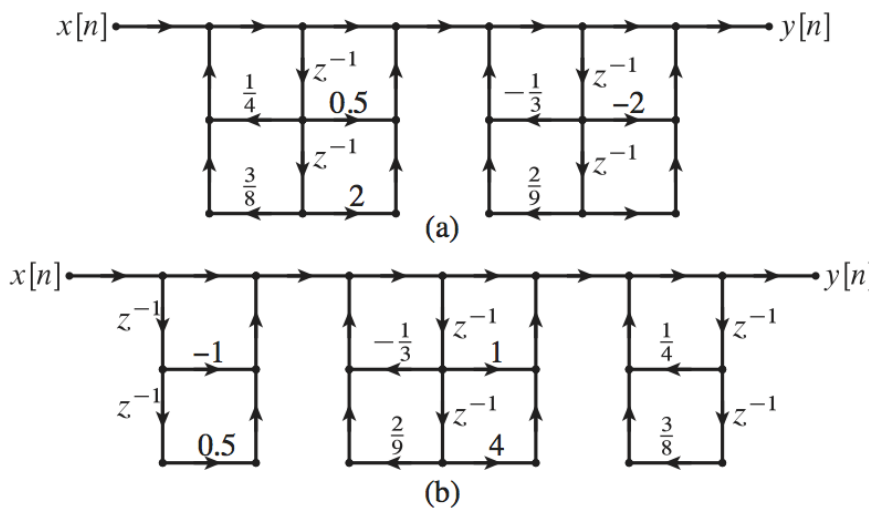
The resulting signal flow graph is shown below.



Problem 5.8

Text Problem 9.29 (Page 533)

Two signal flow graphs are shown below.



(a) Determine the difference equation relating $y[n]$ to $x[n]$ corresponding to signal flow graph (a) above.

Solution: Let the signal in the middle branch for system (a) be $w[n]$. Then we have

$$w[n] = \frac{1}{4}w[n-1] + \frac{3}{8}w[n-2] + x[n] + 0.5x[n-1] + 2x[n-2], \tag{5.8.1}$$

$$y[n] = -\frac{1}{3}y[n-1] + \frac{2}{9}y[n-2] + w[n] - 2w[n-1] + w[n-2]. \tag{5.8.2}$$

From equation $(5.8.1)$, we have

$$\frac{W(z)}{X(z)} = \frac{1 + \frac{1}{2}z^{-1} + 2z^{-2}}{1 - \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2}}.$$

From equation $(5.8,2)$, we have

$$\frac{Y(z)}{W(z)} = \frac{1 - 2z^{-1} + z^{-2}}{1 + \frac{1}{3}z^{-1} - \frac{2}{9}z^{-2}}.$$

Hence, the system function of system (a) is:

$$\begin{aligned} H_{\mathrm{a}}(z) &= \frac{Y(z)}{W(z)}\frac{W(z)}{X(z)} = \frac{(1 + \frac{1}{2}z^{-1} + 2z^{-2})(1 - 2z^{-1} + z^{-2})}{(1 - \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2})(1 + \frac{1}{3}z^{-1} - \frac{2}{9}z^{-2})} \\ &= \frac{1 - 1.5z^{-1} + 2z^{-2} - 3.5z^{-3} + 2z^{-4}}{1 + 0.0833z^{-1} - 0.6806z^{-2} - 0.0694z^{-3} + 0.0833z^{-4}}. \qquad (5.8.3) \end{aligned}$$

(b) Determine the difference equation relating $y[n]$ to $x[n]$ corresponding to signal flow graph (b) above.

Solution: For system (b) let the signals in the left middle branch be $v[n]$ and in the right middle branch be $w[n]$. Then we have

$$\begin{aligned} v[n] &= x[n] - x[n-1] + 0.5x[n-2], && (5.8.4) \\ w[n] &= -\frac{1}{3}w[n-1] + \frac{2}{9}w[n-2] + v[n] + v[n-1] + 4v[n-2], && (5.8.5) \\ y[n] &= w[n] + \frac{1}{4}y[n-1] + \frac{3}{8}y[n-2]. && (5.8.6) \end{aligned}$$

From equation $(5.8.4)$, we have

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

From equation $(5.8.5)$, we have

$$\frac{W(z)}{V(z)} = \frac{1 + z^{-1} + 4z^{-2}}{1 + \frac{1}{3}z^{-1} - \frac{2}{9}z^{-2}}.$$

From equation $(5.8.6)$, we have

$$\frac{Y(z)}{W(z)} = \frac{1}{1 - \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2}}.$$

Hence, the system function of system (b) is:

$$H_b(z) = \frac{Y(z)}{W(z)}\frac{W(z)}{V(z)}\frac{V(z)}{X(z)} = \frac{(1-z^{-1}+0.5z^{-2})(1+z^{-1}+4z^{-2})}{(1-\frac{1}{4}z^{-1}-\frac{3}{8}z^{-2})(1+\frac{1}{3}z^{-1}-\frac{2}{9}z^{-2})}$$

$$= \frac{1+3.5z^{-2}-3.5z^{-3}+2z^{-4}}{1+0.0833z^{-1}-0.6806z^{-2}-0.0694z^{-3}+0.0833z^{-4}} \qquad (5.8.7)$$

(c) Determine if the above two signal flow graphs represent the same discrete-time system.

Solution: Comparing (5.8.3) and (5.8.7), we can conclude that the two systems have the same denominator but different numerators. Hence the systems are not identical.

Problem 5.9

Text Problem 9.32 (Page 534)

The system function of an IIR system is given by

$$H(z) = \frac{0.42-0.39z^{-1}-0.05z^{-2}-0.34z^{-3}+0.4z^{-4}}{1+0.82z^{-1}+0.99z^{-2}+0.28z^{-3}+0.2z^{-4}}.$$

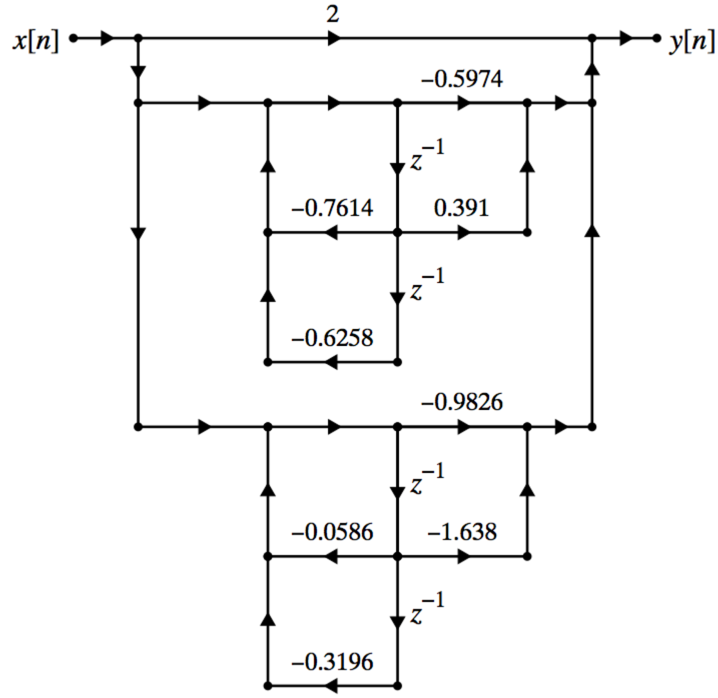
(a) Determine and draw a parallel form structure with second-order sections in direct form II (normal).

Solution: We use the **dir2par** function to obtain the required parallel form structure.

```
clc; close all; clear;
b = [0.42 -0.39 -0.05 -0.34 0.4]; a = [1 0.82 0.99 0.28 0.2];
[C,B,A] = dir2par(b,a)
```

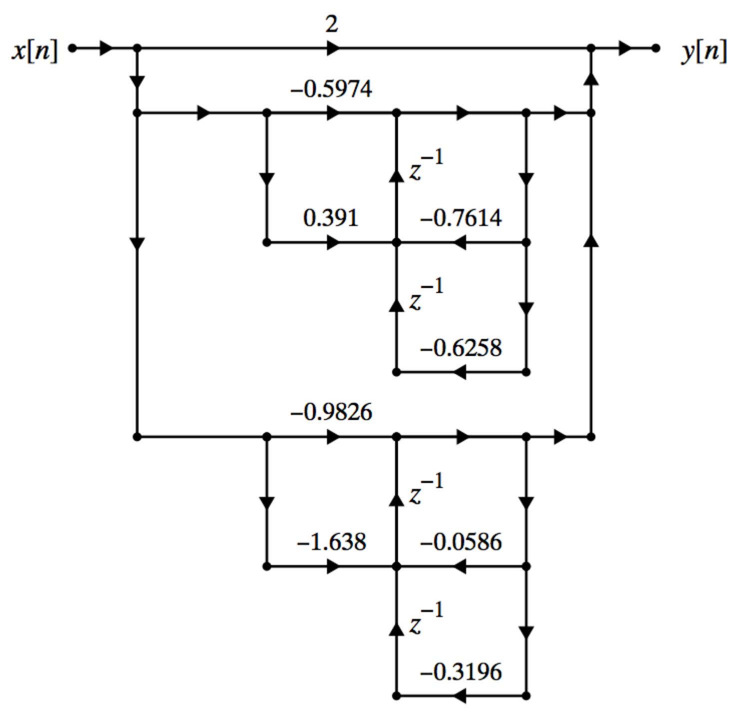
```
C = 2
B = 2×2
   -0.5974    0.3910
   -0.9826   -1.6378
A = 2×3
    1.0000    0.7614    0.6258
    1.0000    0.0586    0.3196
```

The parallel form structure with direct form II (normal) is shown below.



(b) Determine and draw a parallel form structure with second-order sections in direct form II (transposed).

Solution: The parallel form coefficients from part (a) are still applicable. The parallel form structure with direct form II (transposed) is shown below.



Problem 5.10

Text Problem 9.39, parts (c) and (f) only (Page 535)

Consider the FIR system function

$$H(z) = (1 - 3z^{-1} + z^{-2})^5.$$

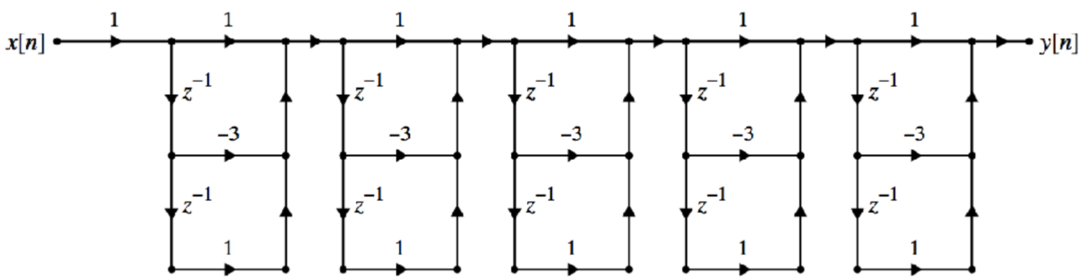
Determine and draw the following structures.

(c) Cascade of second-order sections

Solution: The given system function is in the proper cascade from

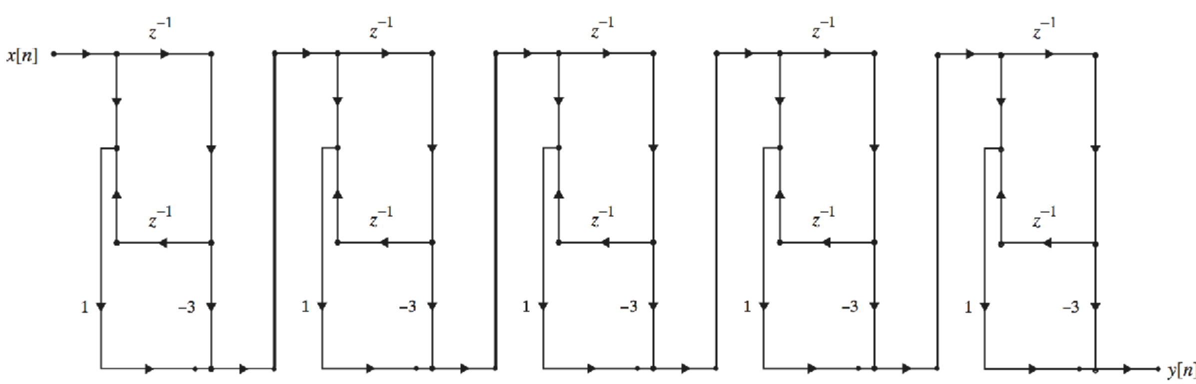
$$H(z) = (1 - 3z^{-1} + z^{-2})(1 - 3z^{-1} + z^{-2})(1 - 3z^{-1} + z^{-2})(1 - 3z^{-1} + z^{-2})(1 - 3z^{-1} + z^{-2}).$$

Hence the signal flow graph is



(f) Cascade of five linear-phase form

Solution: Each second-order section above is in even symmetry format which can be exploited as follows.



Create your MATLAB functions below.

```
function x = idft(X,N)
    % Compute an N-point idft x[n] of X[k] using the fft function according
    % to block diagram
    %

    X = X(:).'; Nx = length(X);
    if nargin == 1
        N = Nx;
    elseif N <= Nx
        X = X(1:N);
    else
        X = [X zeros(1,N-Nx)];
    end
    XR = real(X); XI = imag(X);
    Y = XI+1j*XR; y = fft(Y,N);
    yR = real(y); yI = imag(y);
    x = (1/N)*(yI+1j*yR);
end
```