# EECE5666 (DSP) : Homework-5

**Due on March 25, 2022 by 11:59 pm via submission portal.**

**NAME**: McKean, Tyler

**Table of Contents**

## Instructions

1. You are required to complete this assignment using Live Editor.
2. Enter your MATLAB script in the spaces provided. If it contains a plot, the plot will be displayed after the script.
3. All your plots must be properly labeled and should have appropriate titles to get full credit.
4. Use the equation editor to typeset mathematical material such as variables, equations, etc.
5. After completeing this assignment, export this Live script to PDF and submit the PDF file through the provided submission portal.
6. You will have only one attempt to submit your assignment. Make every effort to submit the correct and completed PDF file the first time.
7. Please submit your homework before the due date/time. A late submission after midnight of the due date will result in loss of points at a rate of 10% per hour until 8 am the following day, at which time the solutions will be published.

## 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:

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

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

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

**Proof**:

Using the following property of complex numbers:

$c = a + jb$

$jc^* = b + ja$

$j(jc^*)^* = a + jb = c$

This property demonstrates that the original complex number is equivalent to taking the conjugation of the complex number multiplied by a factor of $j$ and then again taking the conjugation after multiplying by another factor of $j$

This property can be subsituted into $(5.1.1)$ yielding:

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

$$x[n] = \frac{1}{N}j\left\{\sum_{k=0}^{N-1} jX[k]^*W_N^{kn}\right\}^*$$

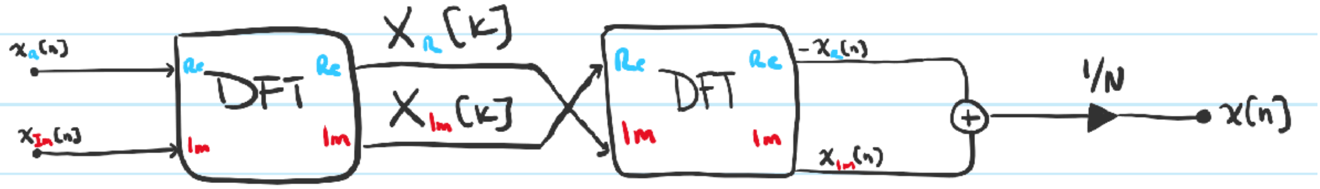
We notice the inside argument containing the DFT of $x[n]$ is now being multiplied by a factor of $j$ with it's conjugate pair and the twiddle factor's sign change comes from the result of taking the conjugate of the inner argument. The normalizing factor of $\frac{1}{N}$ can be pulled outside the curly brackets resulting int the IDFT being calculated by multiplying the inner summation by another factor of $j$.

Thus, our resulting formula mirrors that of $(5.1.2)$

$$x[n] = \frac{1}{N} j \left\{ \sum_{k=0}^{N-1} jX^*[k] W_N^{kn} \right\}^* \quad n = 0, 1, ..., N-1$$

**(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**:



**(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 (5.1.2)
%
% Enter your code below
X = 1i*conj(X); % Innner arguement takes the conjugate of X multiplied by j
x = (1/N)*1i*conj(fft(X,N)); % Take conjugate of fft and multiply by 1/N and factor of j
end
```

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

```
clc; close all; clear;
x = [1 2 3 4 5 6 7 8]; N = length(x); X = fft(x), xn = ifft(X)
```

```
X = 1×8 complex
  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i ···
xn = 1×8
     1     2     3     4     5     6     7     8
```

```
x_n = idft(X,length(x))
```

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

Thus, comparing the results of the ifft() function and my custom function, idft(), the results are identical and prove the N-point IDFT can be computed from $X[k]$ using formula $(5.1.2)$ and the fft() function.

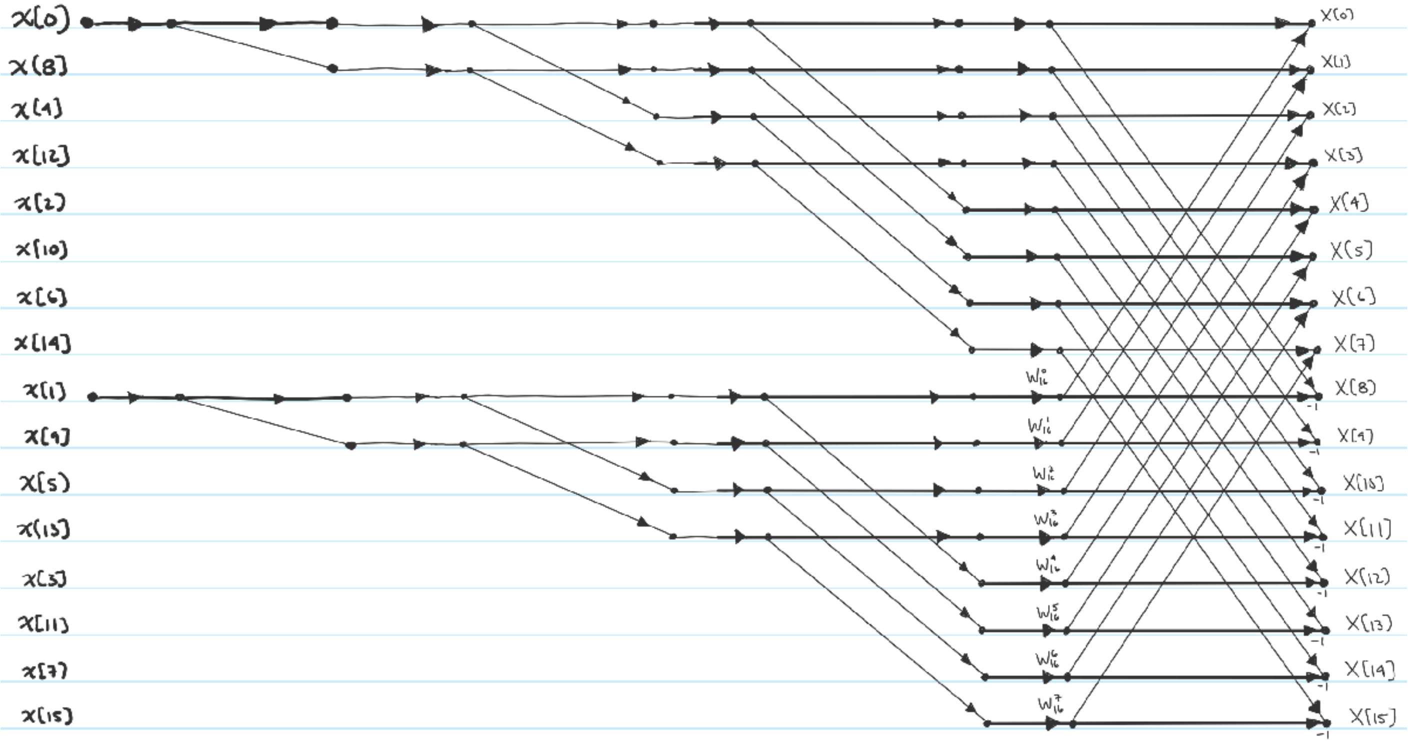
## Problem 5.2

### 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.

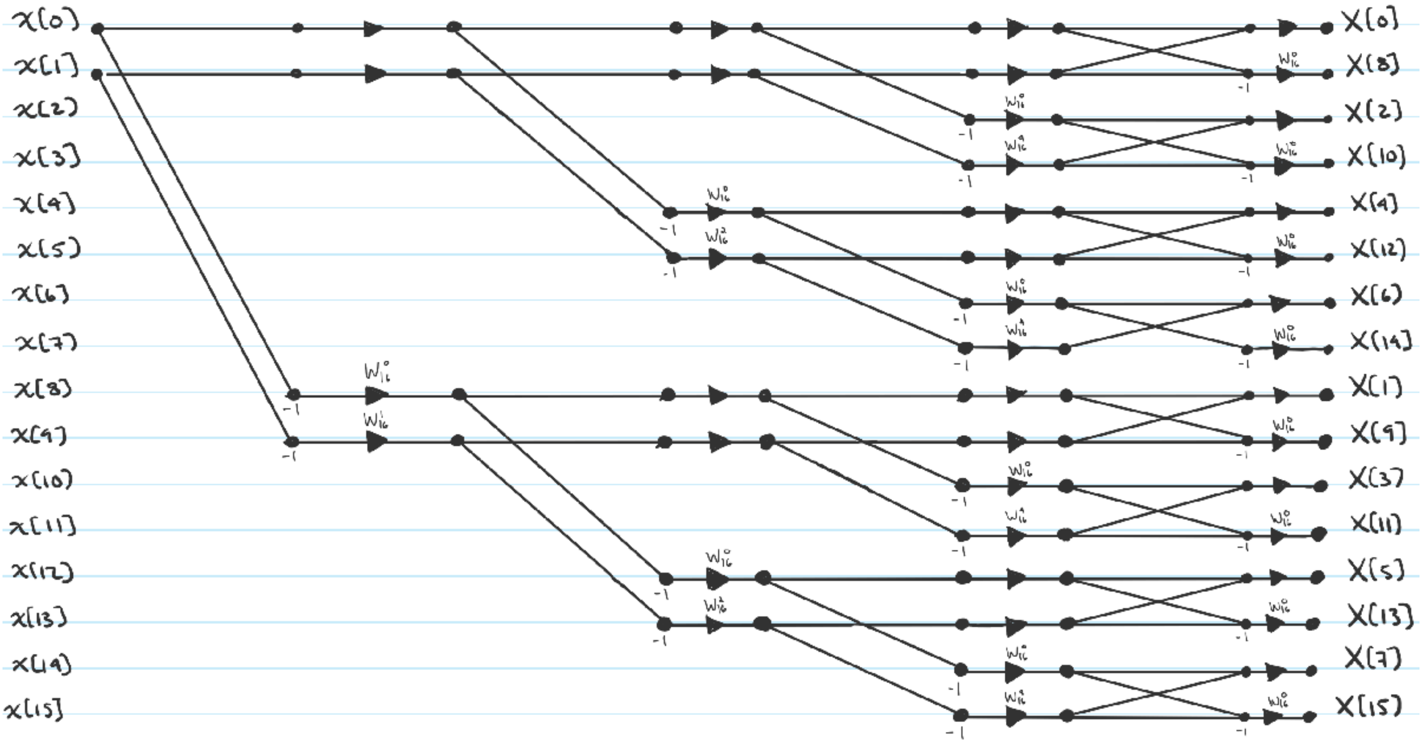
**(a)** Draw a radix-2 $N = 16$-point DIT-FFT flow-chart in which only those paths originating from the non-zero signal values are retained.

**Solution**:



**(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**:



**(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**:

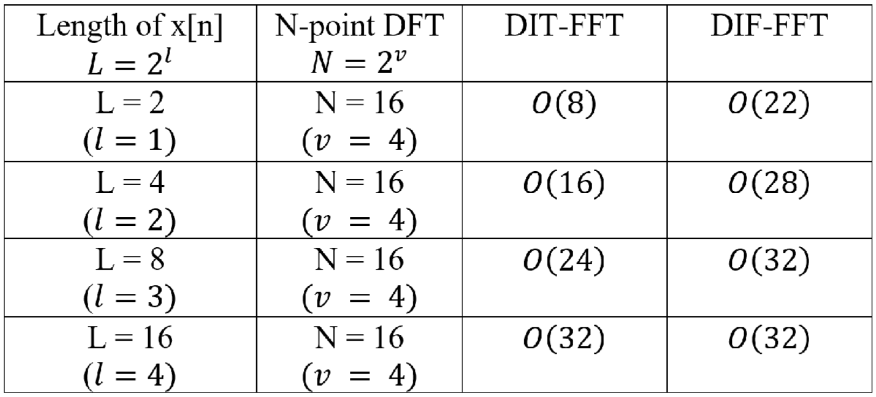
From my flowcharts above, it takes about 22 complex multiplications for DIF and 8 complex multiplications for DIT.

Thus, DIT has fewer number of multiplications.

**(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**:

Below is a table that reveals the number of complex multiplications for both the DIT-FFT and DIF-FFT when the length of the sequence is varied from 2 to 16.



| Length of x[n]<br>$L = 2^l$ | N-point DFT<br>$N = 2^v$ | DIT-FFT | DIF-FFT |
|---|---|---|---|
| L = 2<br>$(l = 1)$ | N = 16<br>$(v = 4)$ | $O(8)$ | $O(22)$ |
| L = 4<br>$(l = 2)$ | N = 16<br>$(v = 4)$ | $O(16)$ | $O(28)$ |
| L = 8<br>$(l = 3)$ | N = 16<br>$(v = 4)$ | $O(24)$ | $O(32)$ |
| L = 16<br>$(l = 4)$ | N = 16<br>$(v = 4)$ | $O(32)$ | $O(32)$ |

We see that when $(l = v)$ the number of complex multiplications for both algorithms will be equivalent. However, when the sequence is much less the $N$- point DFT, the DIT-FFT algorithm has almost $1/3$ of the complex computations.

Thus,

if the length of the sequence $(L = 2^l)$ is much less than the $N$-point DFT value $(N = 2^v)$ than the DIT-FFT algorithms will result in much less complex computations. That is,

$l << v \rightarrow$ DIT-FFT is faster

$l = v \rightarrow$DIT-FFT and DIF-FFT will have equivalent number of complex multiplications

In the case when $l > v$, the $N$-point DFT will only take$N$values from the sequence's length which will result in equivalent complex multiplications for both the DIT and DIF, respectively.

## 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**:

The number of complex multiplications for Direct Approach: $O(KN)$

The number of complex multiplications for Radix-2 DIT-FFT algorithm: $\frac{N}{2} log_2 N$

In order for $K$to be at the minimum value where the DIT-FFT approach becomes more efficient:

$$KN > \frac{N}{2} log_2 N \rightarrow K > \frac{1}{2} log_2 N$$

For N = 128:

$$K > \frac{1}{2}log_2 128 \rightarrow K > 3.5 \rightarrow K > 3$$

$$K > 3$$

```
N = 128; K = floor((1/2)*log2(N))
```

```
K = 3
```

For N = 1024:

$$K > \frac{1}{2}log_2 1024 \rightarrow K > 5$$

$$K > 5$$

```
N = 1024; K = floor((1/2)*log2(N))
```

```
K = 5
```

For N = 8192:

$$K > \frac{1}{2}log_2 8192 \rightarrow K > 6.5$$

$$K > 6$$

```
N = 8192; K = floor((1/2)*log2(N))
```

```
K = 6
```

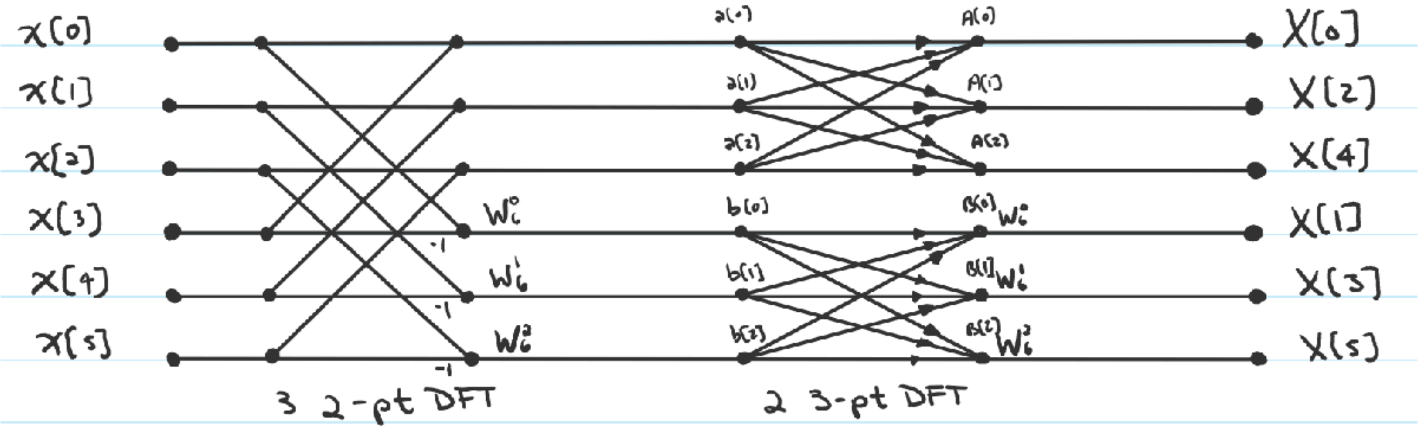
## Problem 5.4

### 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**:



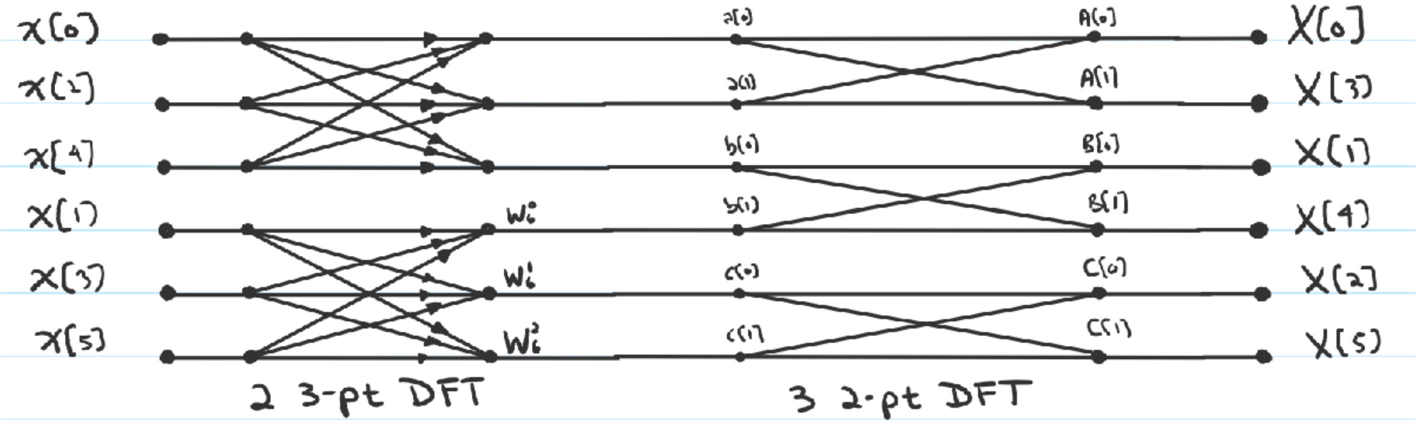
The 3 2-pt DFTs have a total of 12 real additions and then 8 multiplications and 4 real additions from the Twiddle Factors

Then, the 2 3-pt DFTs have a total of 40 real additions and 32 multiplications for a total of:

40 Real Multiplications and 56 Real Additions

**(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**:



Since the composite N can be computed from either order of values of $N_1$ and $N_2$, this FFT will have the same number of real multiplciations and additions as before. Thus,

40 Real Multiplications and 56 Real Additions.

## 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.

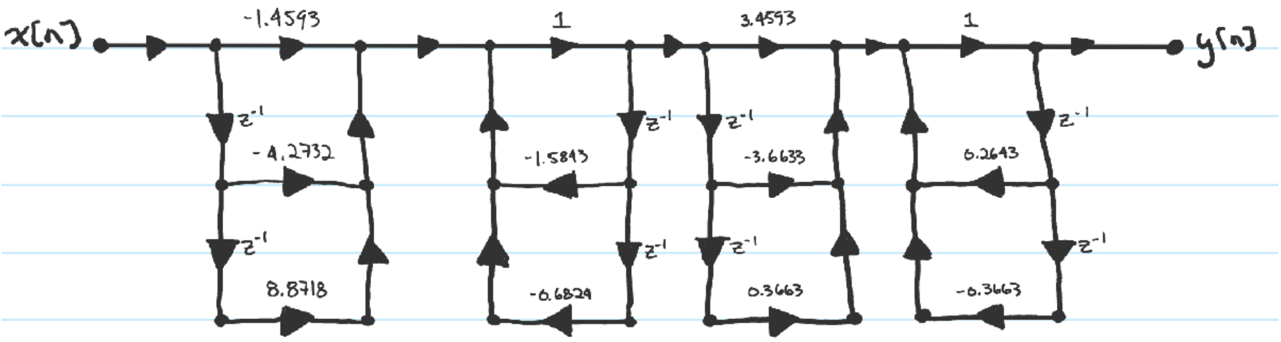
### (a) A cascade form with second-order sections in normal direct form I

**Solution**:

```
b = [1 -3.39 5.76 -6.23 3.25]; a = [1 1.32 0.63 0.4 0.25];
[R,p,C] = residuez(b,a)
```

```
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
```

```
[b1,a1] = residuez(R(1:2),p(1:2),C)
[b2,a2] = residuez(R(3:4),p(3:4),1)
```

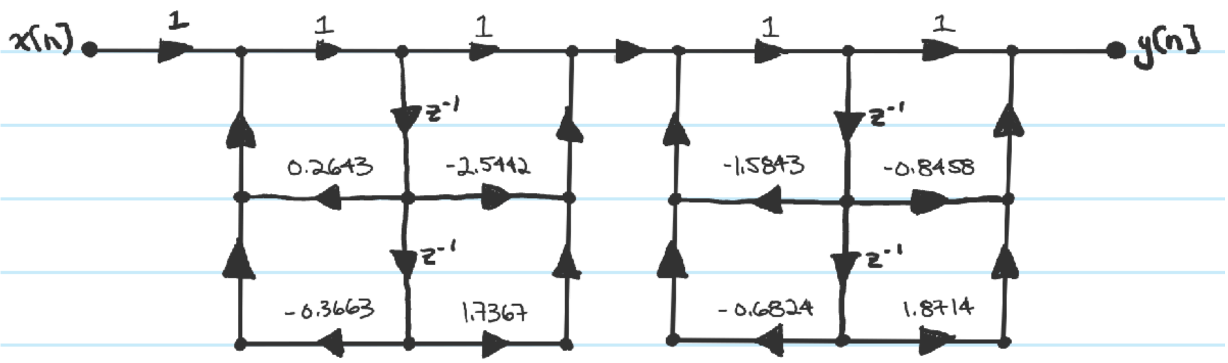


### (b) A cascade form with second-order sections in normal direct form II

**Solution**:

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

```
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
```



## 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**:

To solve for the Direct Forms of this system function, we must return it to a single rational system function from its current partial fraction expansion form

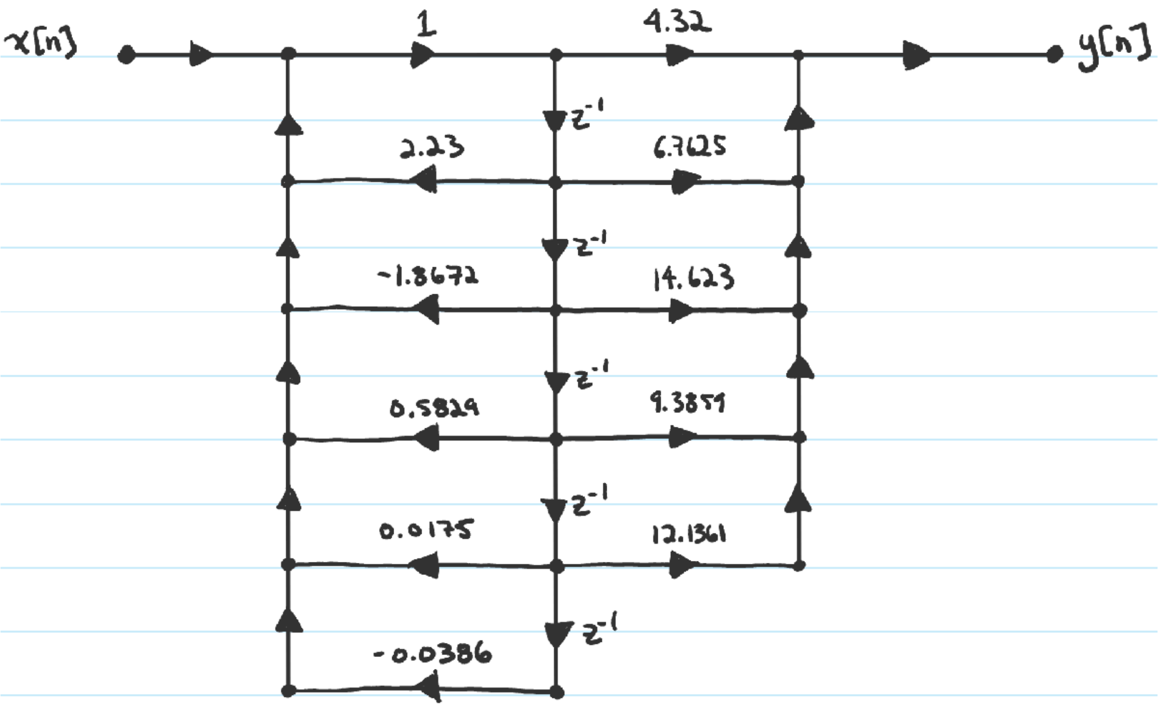
```
b1 = [376.63 -89.05]; a1 = [1 -0.91 0.28];
b2 = [-393.11 364.4]; a2 = [1 -1.52 0.69];
b3 = 20.8; a3 = [1 0.2];
b = conv(b1,a2) + conv(b2,a1);
a = conv(a1,a2); b = conv(b,a3) + conv(a,b3), a = conv(a,a3)
```

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

Thus, our single rational system function can be described as:

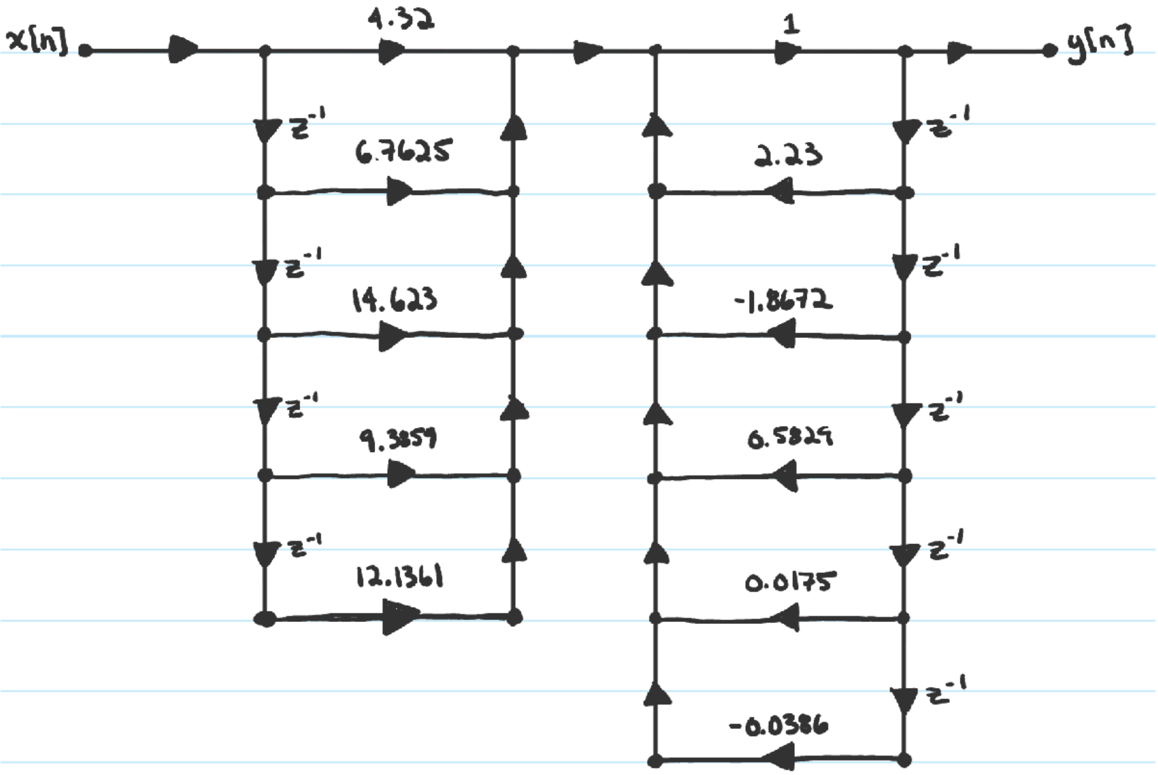
$$H(z) = \frac{4.32 + 6.7625z^{-1} + 14.623z^{-2} + 9.3859z^{-3} + 12.1361z^{-4}}{1 - 2.23z^{-1} + 1.8672z^{-2} - 0.5829z^{-3} - 0.0175z^{-4} + 0.0386z^{-5}}$$

Our Direct Forms can then described from the coefficients of our rational function to be:



**(b) Direct form I (normal)**

**Solution**:

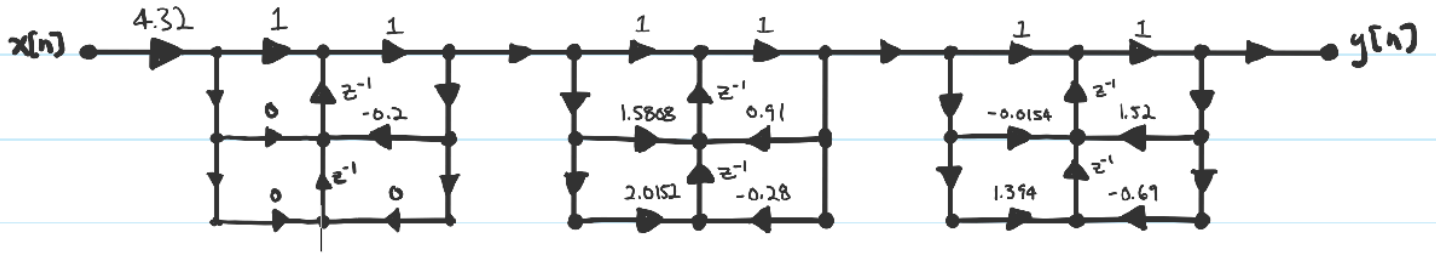


### (c) Cascade form with transposed second-order sections

**Solution**:

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

```
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
```



## Problem 5.7

### Text Problem 9.26 (Page 532)

A discrete time system is described by the difference equation

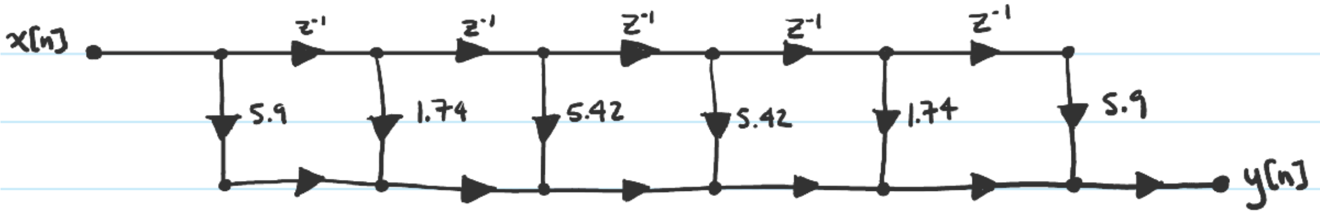
$$y[n] = 5.9x[n] + 1.74x[n-1] + 5.42x[n-2] + 5.42x[n-3] + 1.74x[n-4] + 5.9x[n-5]. \qquad (5.7.1)$$

Determine and draw the following structures.

### (a) Direct form

**Solution**:

$$y[n] = 5.9x[n] + 1.74x[n-1] + 5.42x[n-2] + 5.42x[n-3] + 1.74x[n-4] + 5.9x[n-5]$$



### (b) Cascade form

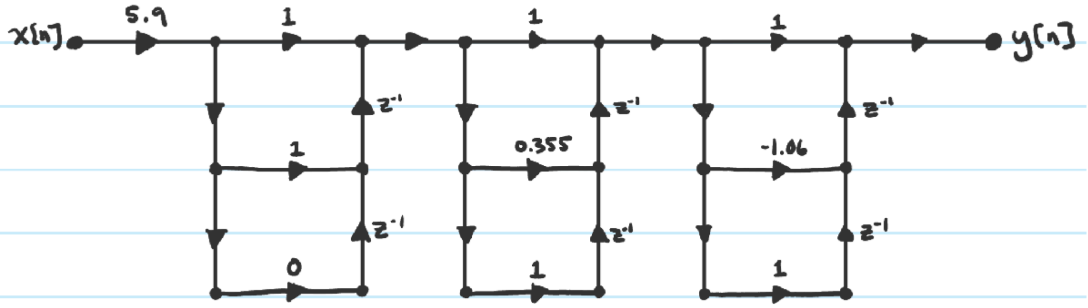
**Solution**:

Converting the difference equation into the system function, we get:

$$H(z)=\frac{Y(z)}{X(z)}=5.9+1.74z^{-1}+5.42z^{-2}+5.42z^{-3}+1.74z^{-4}+5.9z^{-5}$$

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

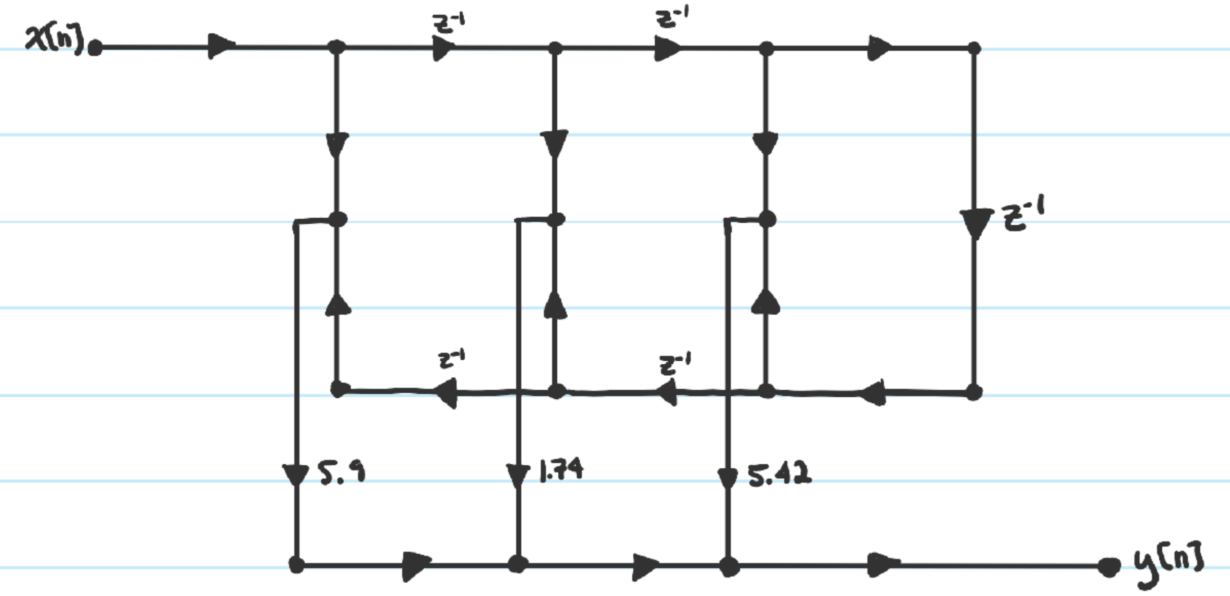
```
sos = 3×6
    1.0000    1.0000         0    1.0000         0         0
    1.0000    0.3550    1.0000    1.0000         0         0
    1.0000   -1.0600    1.0000    1.0000         0         0
G = 5.9000
```



### (c) Linear-phase form

**Solution**:

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



### (d) Frequency sampling form

**Solution**:

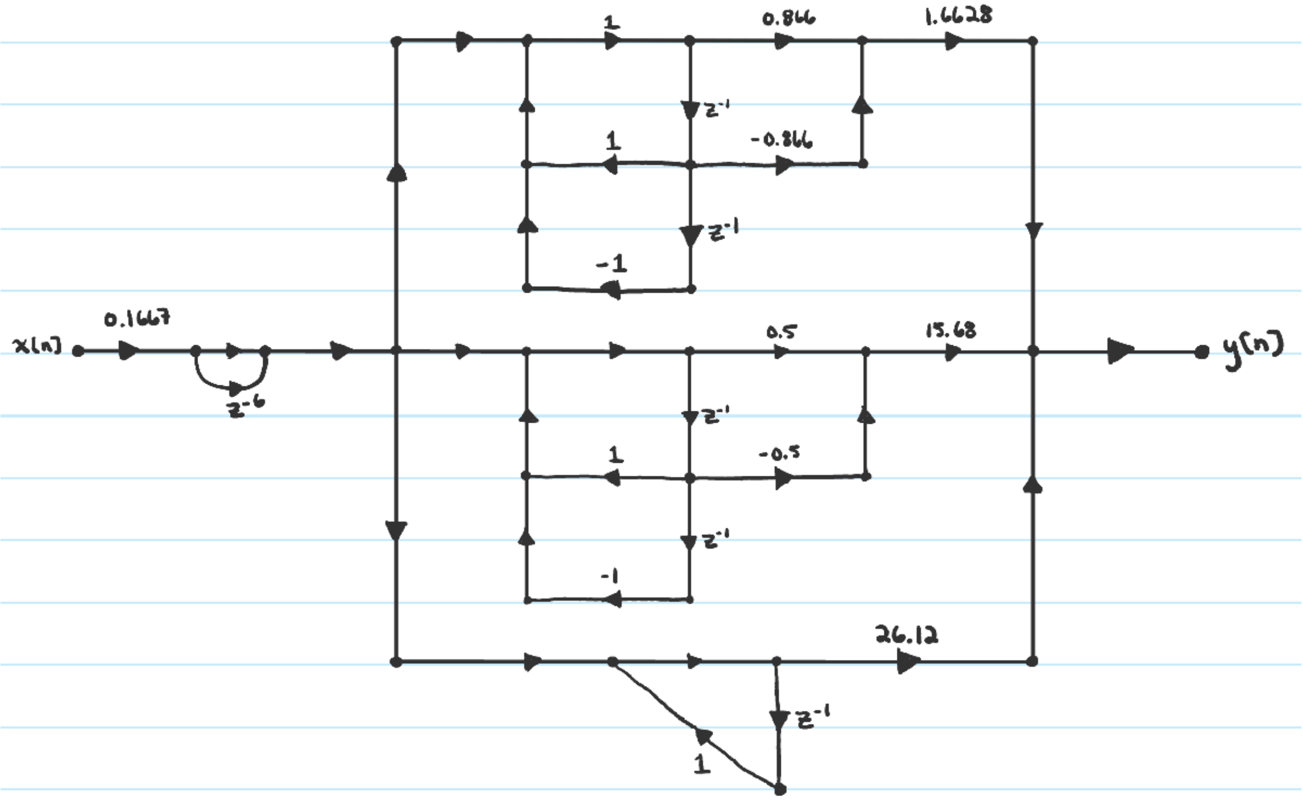
We can calculate the Frequency sampling form coefficients from the direct form of the impulse response:

```
h = [5.9 1.74 5.42 5.42 1.74 5.9]; [C,B,A] = dir2fs(h)
```

```
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 Frequency sampling form can be represented by:

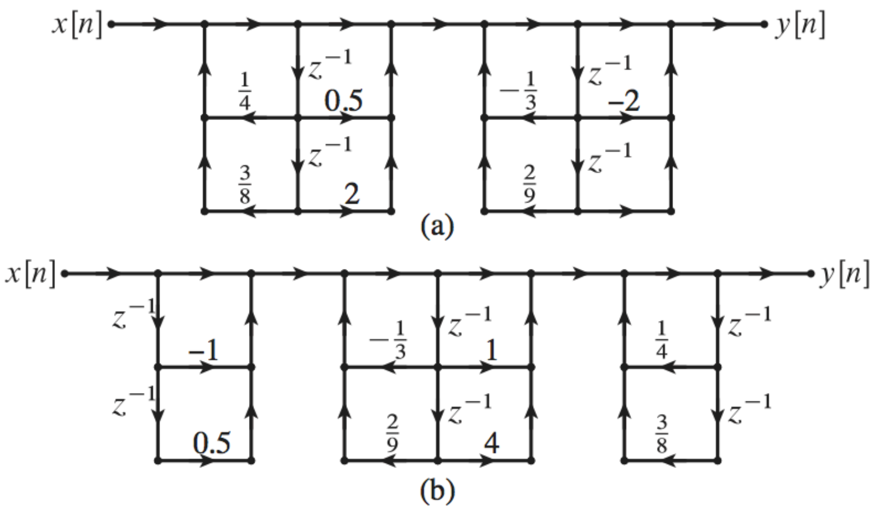
$$H(z) = \frac{1 - z^{-6}}{6}\left[1.6628\,\frac{0.866 - 0.866z^{-1}}{1 - z^{-1} + z^{-2}} + 15.68\,\frac{0.5 - 0.5z^{-1}}{1 + z^{-1} + z^{-2}} + 26.12\,\frac{1}{1 - z^{-1}}\right]$$



## Problem 5.8

### Text Problem 9.29 (Page 533)

Two signal flow graphs are shown below.



(a)

(b)

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

**Solution:**

The structure in signal flow graph (a) appears to be a cascaded structure in direct form II. If we derive its B and A coefficients along with the Gain, we can use the sos2tf() function to derive the original system function coefficients.

```
sos = [1, 0.5, 2, 1, -1/4, -3/8; 1, -2, 1, 1, 1/3, -2/9];
[b,a] = sos2tf(sos)
```

```
b = 1×5
    1.0000   -1.5000    2.0000   -3.5000    2.0000
a = 1×5
    1.0000    0.0833   -0.6806   -0.0694    0.0833
R = 4×1
    0.3318
   37.2549
  -51.8400
   -8.7467
p = 4×1
    0.7500
   -0.6667
   -0.5000
    0.3333
C = 24
b1 = 1×3
   61.5867  -29.7200  -12.0000
a1 = 1×3
    1.0000   -0.0833   -0.5000
b2 = 1×3
  -59.5867   13.0733   -0.1667
a2 = 1×3
    1.0000    0.1667   -0.1667
```

Thus, the system function that is defined by this signal flow graph is:

$H(z) = \dfrac{1 - \frac{3}{2}z^{-1} + 2z^{-2} - \frac{7}{2}z^{-3} + 2z^{-4}}{1 + 0.0833z^{-1} - 0.6806z^{-2} - 0.0694z^{-3} + 0.0833z^{-4}}$ , which now that we know the system function, the difference equation can be derived as:

$$y[n] = -0.0833y[n-1] + 0.6806y[n-2] + 0.0694y[n-3] - 0.0833y[n-4] + x[n] - \frac{3}{2}x[n-1]$$

$$+2x[n-2] - \frac{7}{2}x[n-3] + 2x[n-4]$$

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

**Solution**:

Signal flow graph (b) appears to be a cascaded structure that includes the direct form I of an all-pole and all-zero response multiplied by the another structure in a direct form II. We can define the SOS of this structure that includes zeros in the appropriate locations for the all-pole and all-zero structures:

```
sos = [1, -1, 0.5, 1, 0, 0; 1, 1, 4, 1, 1/3, -2/9; 1, 0, 0, 1, -1/4, -3/8];
[b,a] = sos2tf(sos)
```

```
b = 1×6
    1.0000         0    3.5000   -3.5000    2.0000         0
a = 1×6
    1.0000    0.0833   -0.6806   -0.0694    0.0833         0
```

Now that we have the coefficients for the numerator and denominator of the system function, we see that

$$H(z) = \frac{1 + \frac{7}{2}z^{-2} - \frac{7}{2}z^{-3} + 2z^{-4}}{1 + 0.0833z^{-1} - 0.6806z^{-2} - 0.0694z^{-3} + 0.0833z^{-4}}$$

We can derive the difference equation to be:

$$y[n] = -0.0833y[n-1] + 0.6806y[n-2] + 0.0694y[n-3] - 0.0833y[n-4] + x[n] + \frac{7}{2}x[n-2] - \frac{7}{2}x[n-3] + 2x[n-4]$$

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

**Solution**:

After comparing the two difference equations, we see that both the denominator coefficients and delays match for each structure's system function, however, the numerator's contain different delays, and almost all the same coefficients. Thus, the two signal graphs are do not represent the same discrete-time system.

## Problem 5.9

### Text Problem 9.32 (Page 534)

The system function of an IIR system is given by

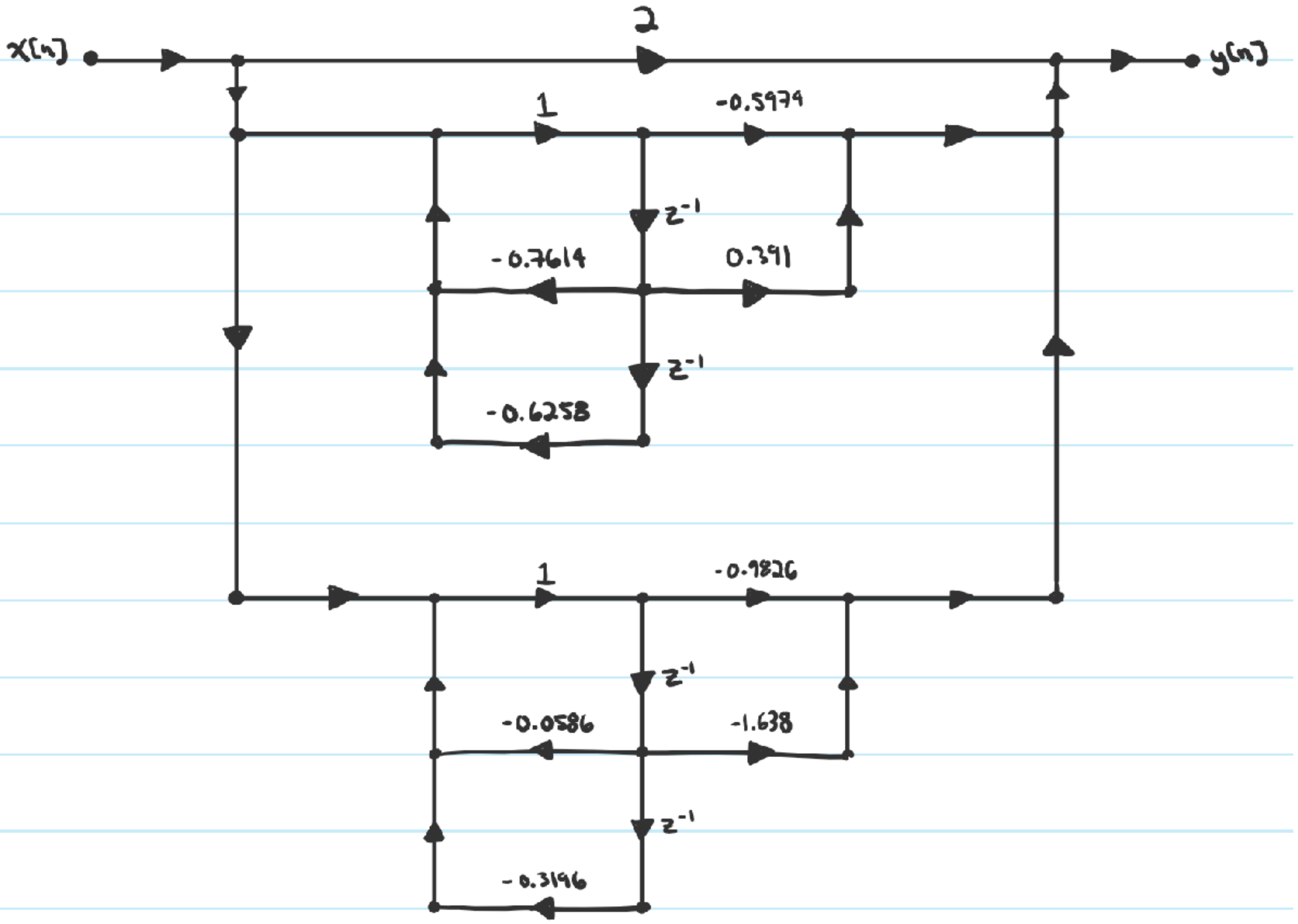
$$H(z) = \frac{0.42 - 0.39z^{-1} - 0.05z^{-2} - 0.34z^{-3} + 0.4z^{-4}}{1 + 0.82z^{-1} + 0.99z^{-2} + 0.28z^{-3} + 0.2z^{-4}}.$$

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

**Solution:**

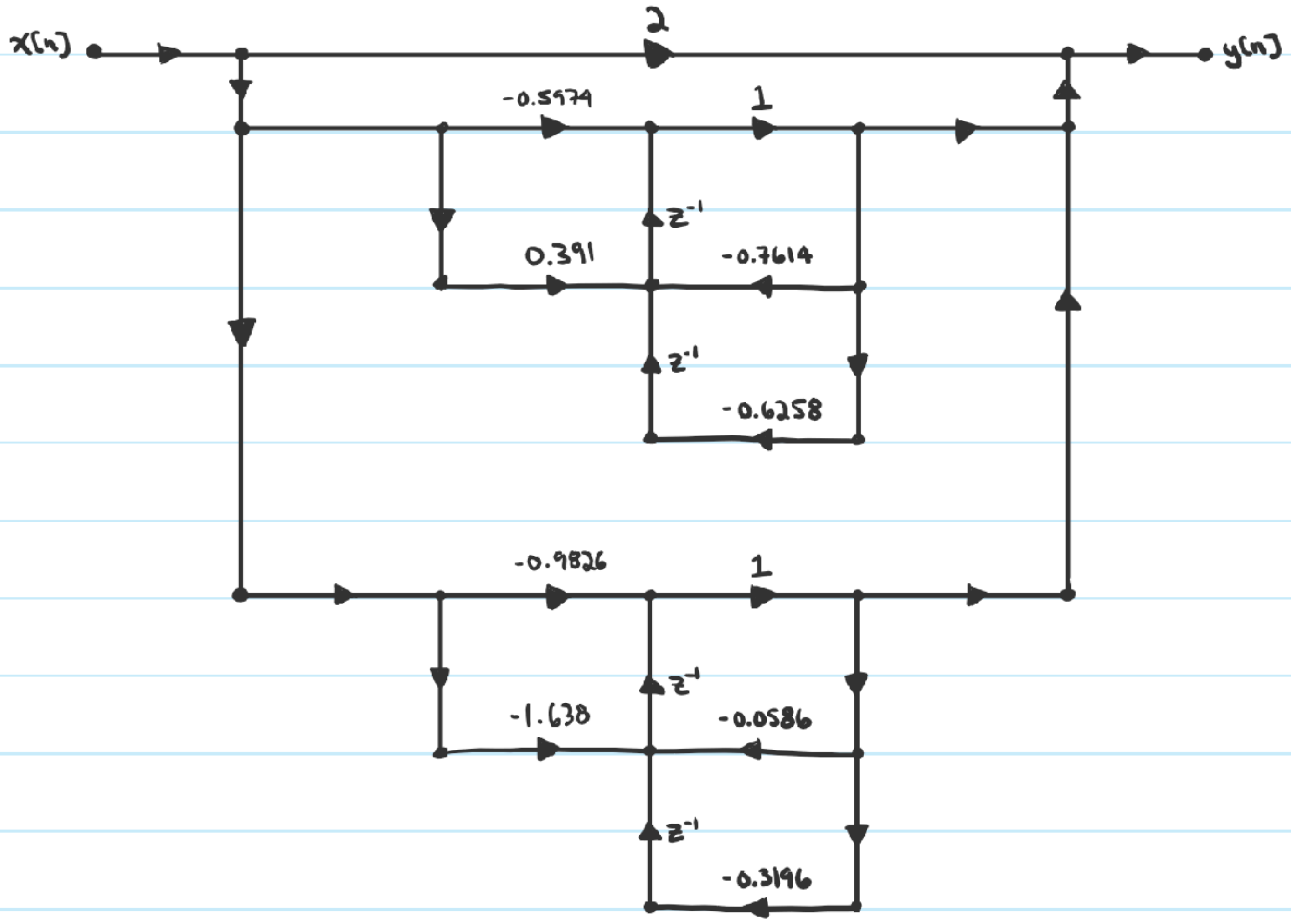
```
b = [0.42 -0.39 -0.05 -0.34 0.4]; a = [1 0.82 0.99 0.28 0.2];
[C,B,A] = dir2par(b,a)
```

```
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
```



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

**Solution:**



## Problem 5.10

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

Consider the FIR system function

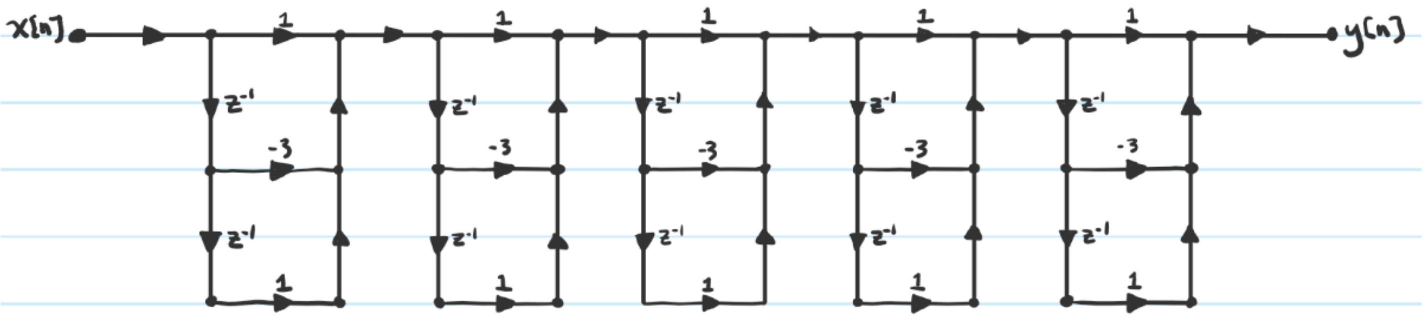
$$H(z) = (1 - 3z^{-1} + z^{-2})^5.$$

Determine and draw the following structures.

**(c) Cascade of second-order sections**

**Solution**:

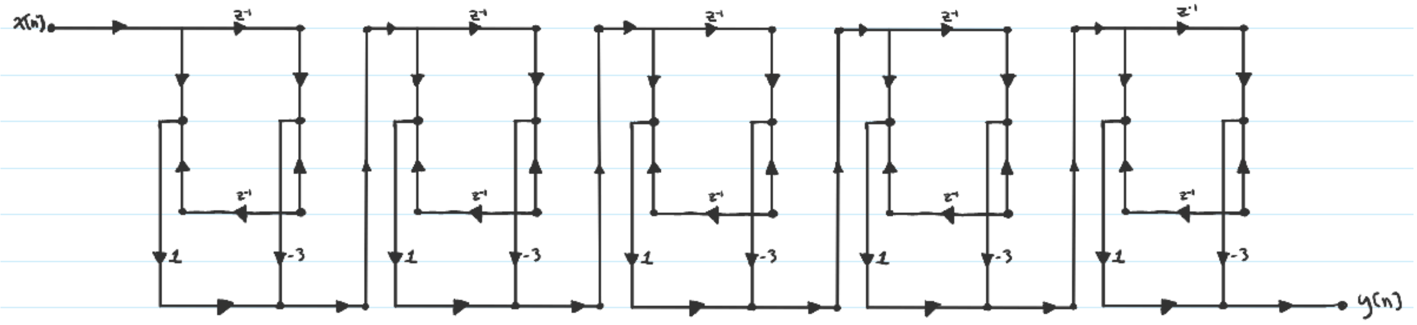
Since the FIR system function is already represented in a product form, we derive the cascade form from the equation above:



(f) **Cascade of five linear-phase form**

**Solution**:

We can determine the Linear-Phase Form after we expand the FIR system function and examine all of its polynomial coefficients



Create your MATLAB function below.

```
function x = idft(X,N)
% Compute an N-point idft x[n] of X[k] using the fft function according to (5.1.2)
%
% Enter your code below
X = 1i*conj(X); % Innner arguement takes the conjugate of X multiplied by j
x = (1/N)*1i*conj(fft(X,N)); % Take conjugate of fft and multiply by 1/N and factor of j
end
```