EECE5666 (DSP) : Homework-5 Solutions

Table of Contents

Default Plot Parameters:

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

Problem-1: Numerical computation of CTFS using FFT

Consider a periodic analog signal $x_c(t)$ with fundamental period $T_0 = 2\,\mathrm{s}$ given by $x_c(t) = te^{-0.5t}$ over one period $0 \le t \le 2$.

(a) Show that the CTFS coefficients c_k of $x_c(t)$ are given by

$$c_k = \frac{2\left(1-(2+\mathrm{j}2\pi k)e^{-(1+\mathrm{j}2\pi k)}\right)}{(1+\mathrm{j}2\pi k)^2} \quad \text{for all } k.$$

Solution: The CTFS coefficients are given by

$$c_k = \frac{1}{T_0}\int_0^{T_0} x_c(t)e^{-\mathrm{j}(2\pi/T_0)kt}\mathrm{d}t = \frac{1}{T_0}\int_0^{T_0} (te^{-0.5t})e^{-\mathrm{j}(2\pi/T_0)kt}\mathrm{d}t = \frac{1}{T_0}\int_0^{T_0} te^{-\left(0.5+\mathrm{j}\frac{2\pi k}{T_0}\right)t}\mathrm{d}t.$$

Let $\alpha = 0.5 + \mathrm{j}\frac{2\pi k}{T_0}$. Then we have

$$c_k = \frac{1}{T_0}\int_0^{T_0} te^{-\alpha t}\mathrm{d}t = \frac{1}{T_0}\left[\frac{te^{-\alpha t}}{-\alpha}\Big|_0^{T_0} - \int_0^{T_0}\frac{e^{-\alpha t}}{-\alpha}\mathrm{d}t\right]$$

$$= \frac{1}{T_0}\left[-\frac{T_0e^{-\alpha t}}{\alpha} - \frac{1}{\alpha^2}\left(e^{-\alpha T_0}-1\right)\right] = \frac{1}{T_0\alpha^2}\left[1-e^{-\alpha T_0}(1+\alpha T_0)\right].$$

Substituting $\alpha = 0.5 + \jmath\frac{2\pi k}{T_0}$ and simplifying, we obtain

$$c_k = \frac{T_o}{(0.5T_0 + \jmath 2\pi k)^2}\left(1 - (1 + 0.5T_0 + \jmath 2\pi k)\mathrm{e}^{-(0.5T_0+\jmath 2\pi k)}\right).$$

Finally, substituting $T_0 = 2\mathrm{s}$ and simplifying, we obtain the final result

$$c_k = \frac{2\left(1 - (2 + \jmath 2\pi k)\mathrm{e}^{-(1+\jmath 2\pi k)}\right)}{(1 + \jmath 2\pi k)^2} \quad \text{for all } k.$$

(b) Let T be the sampling interval in seconds such that $T_0 = NT$. Thus, there are N samples in one T_0 period. Let $x(n) = x_\mathrm{c}(t)\big|_{t=nT}$, $0 \le n \le N-1$. Show that the CTFS coefficients c_k of $x_\mathrm{c}(t)$ can be numerically approximated by

$$\widehat{c}_k = \frac{1}{N}\underset{N\text{-point}}{\mathrm{DFT}}\{x(n)\}.$$

Solution: To numerically compute $\widehat{c}_k$, the integral in the computation of c_k is replaced by a summation, t by nT, and $\mathrm{d}t$ by T to obtain

$$\begin{aligned}
\widehat{c}_k &= \frac{1}{T_0}\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\jmath(2\pi/T_0)knT}T = \frac{T}{T_0}\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\jmath(2\pi/T_0)knT} \\
&= \frac{1}{T_0/T}\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\jmath(2\pi/(T_0/T))kn} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\jmath(2\pi/(N))kn} \quad (\text{since } T_0 = NT) \\
&= \frac{1}{N}\mathrm{DFT}\big(x(n)\big).
\end{aligned}$$

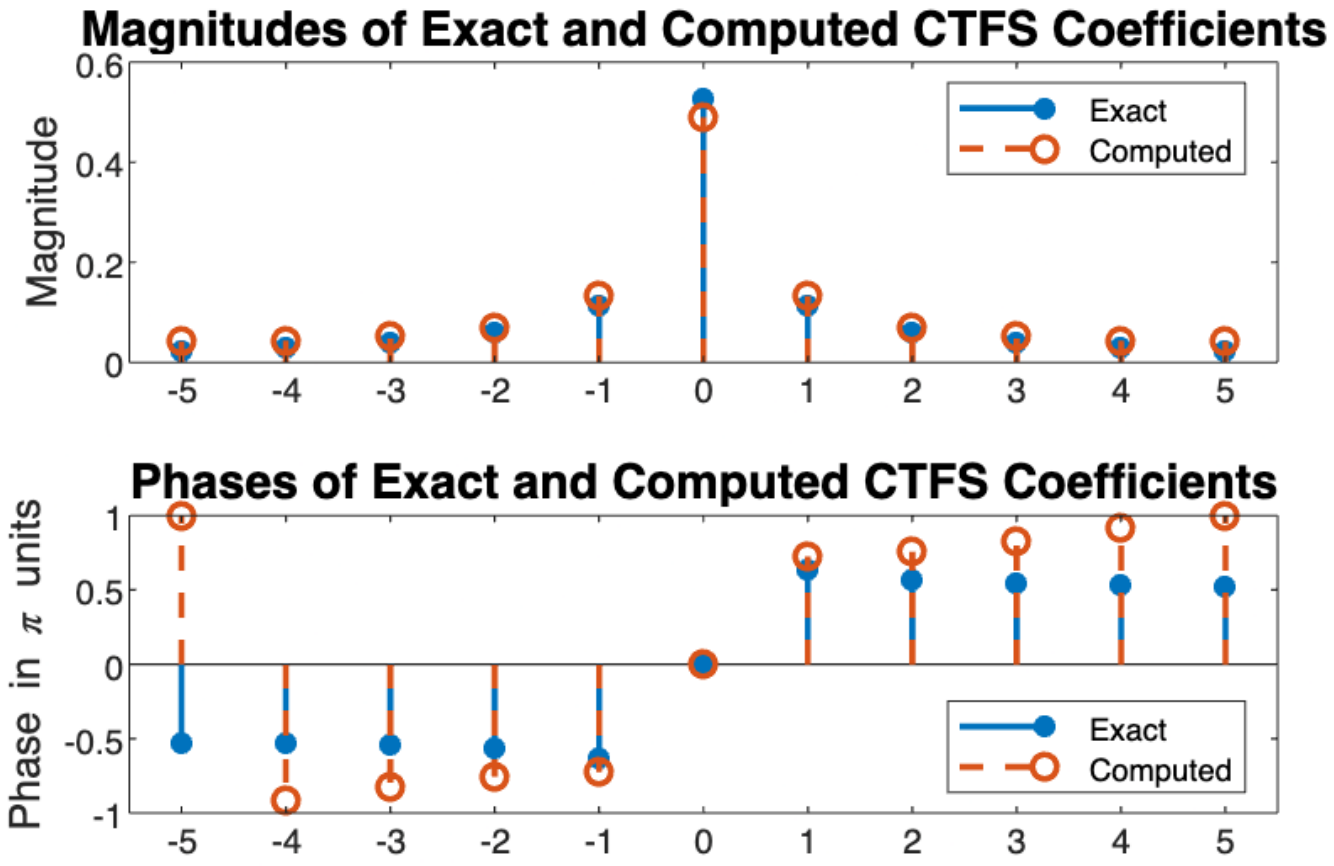
We will evaluate the above DFT using the **`fft`** algorithm.

(c) Choose sampling interval $T = 0.2\,\mathrm{s}$. Sample one period of $\widetilde{x}_\mathrm{c}(t)$ to obtain $N = 10$ samples and compute the approximate CTFS $\widehat{c}_k$ using the **`fft`** function. Graph magnitude **`stem`** plots of c_k and $\widehat{c}_k$ in the first sub-plot over $-5 \le k \le 5$. Similarly graph phase **`stem`** plots in the second sub-plot. Comment on your results.

MATLAB script and plots:

```
clear; T0 = 2; F0 = 1/T0; % fundamental frequency
N = 10; t = linspace(0,T0,N+1); t = t(1:N); % sampling instances
xn = t.*exp(-0.5*t); % one period of sampled periodic sequence
k = -5:5; % CTFS index
ck = (T0./(0.5*T0+2j*pi*k).^2).*(1-(1+0.5*T0+2j*pi*k).*exp(-(0.5*T0+2j*pi*k))); % exac
ckhat = (1/N)*fft(xn); ckhat = fftshift(ckhat); ckhat = [ckhat,ckhat(1)]; % computed
figure('PaperPosition',[0,0,7,4.5]*72,'position',[0,0,7,4.5]*72);
subplot(2,1,1); % stem plots of magnitudes
stem(k,abs(ck),'filled','markersize',4,'LineWidth',1.5); hold on;
```

```
stem(-N/2:N/2,abs(ckhat),'LineWidth',1.5,'LineStyle','--'); axis([-5.5,5.5,0,0.6]);
ylabel('Magnitude'); title('Magnitudes of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','northeast'); hold off;
subplot(2,1,2); % stem plots of phases in units of pi
stem(k,angle(ck)/pi,'filled','markersize',4,'linewidth',1.5); hold on;
stem(-N/2:N/2,angle(ckhat)/pi,'LineStyle','--','LineWidth',1.5);
ylabel('Phase in \pi units'); grid('minor'); grid; axis([-5.5,5.5,-1,1]);
title('Phases of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','southeast'); grid('minor'); grid; hold off;
```



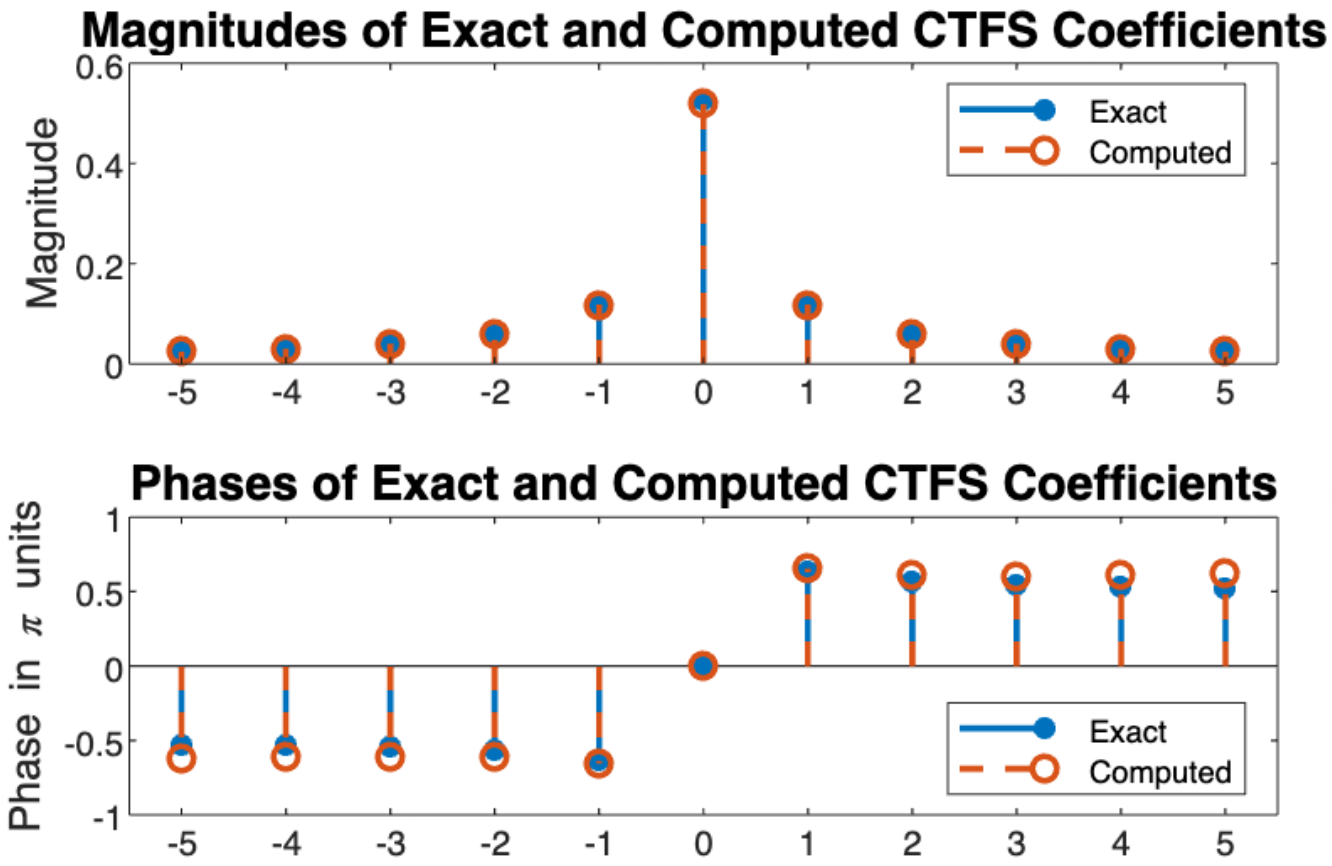
Comment: From the above plots it is obvious that the numerically computed CTFS coefficients are different from the exact CTFS coefficients, specifically in the phase values. Clearly, the sampling frequency is not sufficient.

(d) Repeat part (c) for $T = 0.04\,\mathrm{s}$ and $N = 50$.

MATLAB script and plots:

```
clear; T0 = 2; F0 = 1/T0; % fundamental frequency
N = 50; t = linspace(0,T0,N+1); t = t(1:N); % sampling instances
xn = t.*exp(-0.5*t); % one period of sampled periodic sequence
k = -5:5; % CTFS index
ck = (T0./(0.5*T0+2j*pi*k).^2).*(1-(1+0.5*T0+2j*pi*k).*exp(-(0.5*T0+2j*pi*k))); % exac
```

```
ckhat = (1/N)*fft(xn); ckhat = fftshift(ckhat); ckhat = [ckhat,ckhat(1)]; % computed
figure('PaperPosition',[0,0,7,4.5]*72,'position',[0,0,7,4.5]*72);
subplot(2,1,1); % stem plots of magnitudes
stem(k,abs(ck),'filled','markersize',4,'LineWidth',1.5); hold on;
stem(-N/2:N/2,abs(ckhat),'LineWidth',1.5,'LineStyle','--'); axis([-5.5,5.5,0,0.6]);
ylabel('Magnitude'); title('Magnitudes of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','northeast'); hold off;
subplot(2,1,2); % stem plots of phases in units of pi
stem(k,angle(ck)/pi,'filled','markersize',4,'linewidth',1.5); hold on;
stem(-N/2:N/2,angle(ckhat)/pi,'LineStyle','--','LineWidth',1.5);
ylabel('Phase in \pi units'); grid('minor'); grid; axis([-5.5,5.5,-1,1]);
title('Phases of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','southeast'); grid('minor'); grid; hold off;
```



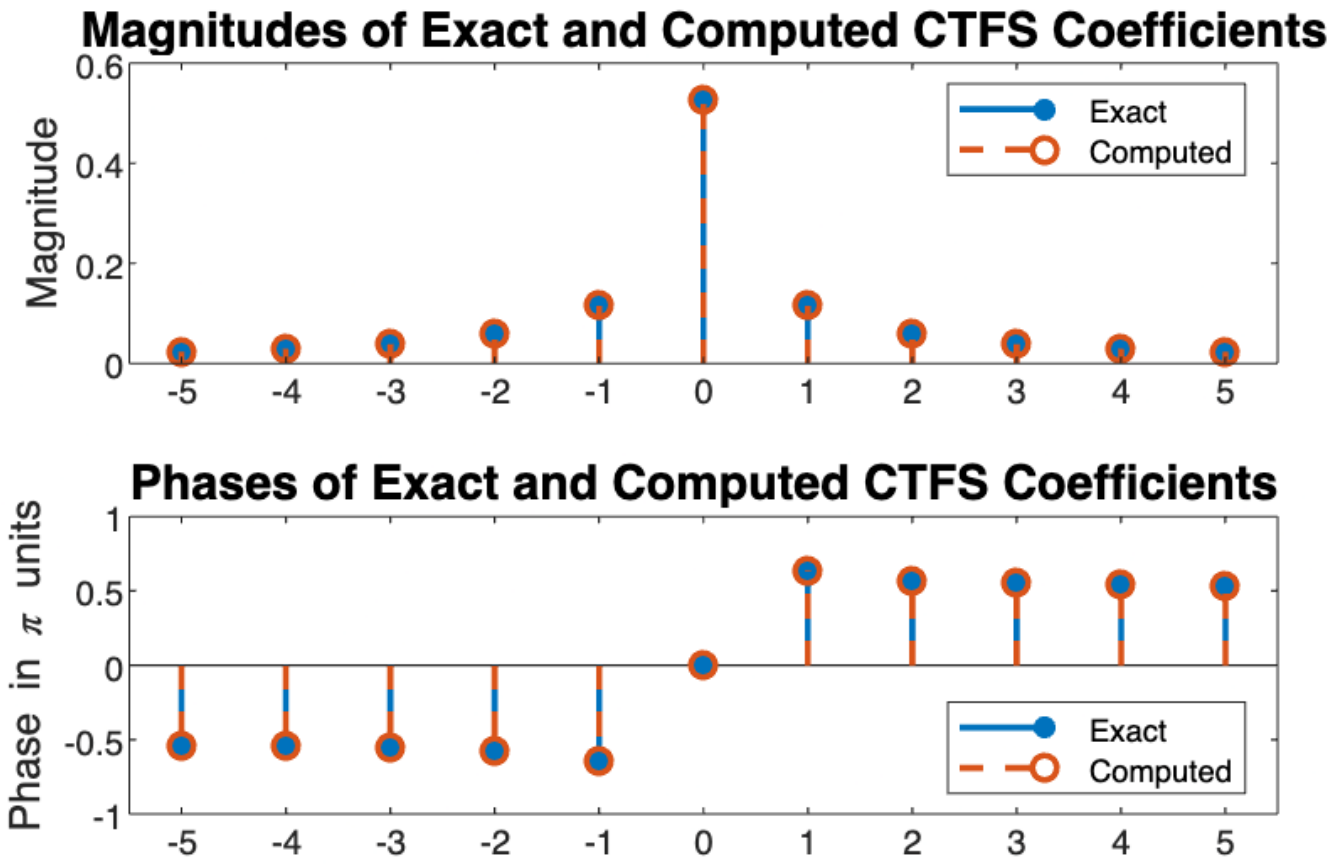
Comment: From the above plots we observe that the numerically computed CTFS coefficients are approaching the exact CTFS coefficients. The magnitude values are almost correct but still phase values are not. So, we will need a higher sampling frequency.

(e) Repeat part (c) for $T = 0.004\,\text{s}$ and $N = 500$.

MATLAB script and plots:

```
clear; T0 = 2; F0 = 1/T0; % fundamental frequency
```

```
N = 500; t = linspace(0,T0,N+1); t = t(1:N); % sampling instances
xn = t.*exp(-0.5*t); % one period of sampled periodic sequence
k = -5:5; % CTFS index
ck = (T0./(0.5*T0+2j*pi*k).^2).*(1-(1+0.5*T0+2j*pi*k).*exp(-(0.5*T0+2j*pi*k))); % exact
ckhat = (1/N)*fft(xn); ckhat = fftshift(ckhat); ckhat = [ckhat,ckhat(1)]; % computed
figure('PaperPosition',[0,0,7,4.5]*72,'position',[0,0,7,4.5]*72);
subplot(2,1,1); % stem plots of magnitudes
stem(k,abs(ck),'filled','markersize',4,'LineWidth',1.5); hold on;
stem(-N/2:N/2,abs(ckhat),'LineWidth',1.5,'LineStyle','--'); axis([-5.5,5.5,0,0.6]);
ylabel('Magnitude'); title('Magnitudes of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','northeast'); hold off;
subplot(2,1,2); % stem plots of phases in units of pi
stem(k,angle(ck)/pi,'filled','markersize',4,'linewidth',1.5); hold on;
stem(-N/2:N/2,angle(ckhat)/pi,'LineStyle','--','LineWidth',1.5);
ylabel('Phase in \pi units'); grid('minor'); grid; axis([-5.5,5.5,-1,1]);
title('Phases of Exact and Computed CTFS Coefficients');
legend('Exact','Computed','location','southeast'); grid('minor'); grid; hold off;
```



Comment: The above plots show that the numerically computed CTFS coefficients and the exact CTFS coefficients are same in both the magnitude and phase values. Clearly, the sampling frequency is sufficient.

Problem-2: Text Problem 8.3 (page 566)

Let $x(n)$ be a real-valued *N*-point $(N = 2^{\nu})$ sequence. Develop a method to compute an N-point DFT $X'(k)$, which contains only the odd harmonics [i.e., $X'(k) = 0$ if k is even] by using only a real $N/2$-point DFT.

Solution: To achieve sequence $X'(k)$ for $k \in (0, N-1)$, we first ignore the elements when k is even and foucs on those elements with k as odd.

Therefore, we define $X_1(k) = X'(2k+1)$ for $k \in (0, N/2-1)$. Since we have to use the real N/2-point DFT, we have:

$$\begin{aligned} X_1(k) &= X'(2k+1) = \sum_{n=0}^{N-1} x(n) W_N^{(2k+1)n} \\ &= \sum_{n=0}^{N/2-1} x(n) W_N^{(2k+1)n} + \sum_{n=N/2}^{N-1} x(n) W_N^{(2k+1)n} \\ &= \sum_{n=0}^{N/2-1} x(n) W_N^{(2k+1)n} + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{(2k+1)(n+N/2)} \end{aligned}$$

Based one the property of W_N^{kn}, we can further derive the equation above as:

$$\begin{aligned} X_1(k) &= \sum_{n=0}^{N/2-1} x(n) W_N^{(2k+1)n} + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{(2k+1)(n+N/2)} \\ &= \sum_{n=0}^{N/2-1} x(n) W_N^{2kn} W_N^n + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{2k(n+N/2)} W_N^{n+N/2} \\ &= \sum_{n=0}^{N/2-1} x(n) W_N^{2kn} W_N^n + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{2kn} W_N^{2kN/2} W_N^n W_N^{N/2} \\ &= \sum_{n=0}^{N/2-1} x(n) W_{N/2}^{kn} W_N^n + \sum_{n=0}^{N/2-1} x(n+N/2) W_{N/2}^{kn} W_N^{kN} W_N^n W_N^{N/2} \\ &= \sum_{n=0}^{N/2-1} x(n) W_{N/2}^{kn} W_N^n - \sum_{n=0}^{N/2-1} x(n+N/2) W_{N/2}^{kn} W_N^n \qquad \left(W_N^{2kN/2} = 1, W_N^{N/2} = -1\right) \\ &= \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_{N/2}^{kn} W_N^n \end{aligned}$$

Therefore, we have $X'(k)$ as:

$$X'(k) = \begin{cases} 0, & \text{for } k \text{ is even} \\ X_1(k/2-1), & \text{for } k \text{ is odd} \end{cases}$$

where $X_1(k/2-1) = \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_{N/2}^{(k/2-1)n} W_N^n$

Problem-3: Text Problems 8.10 and 8.12 (page 567)

Note: This problem requires drawings of flow-graphs. MATLAB drawing would be tedious so draw flow-graph neatly on a paper, scan it as an image file (**`.jpeg`** or **`.png`**) and insert the image file.

(a) Derive the signal flow graph for the $N = 16$-point, radix-4 decimation-in-frequency FFT algorithm in which the input sequence is in digit-reversed order and the output DFT is in normal order.

Solution: First provide a mathematical basis for your SFG and then provide the drawing. Do not simply insert the drawing.

Since we have $N = 16$, we break the DFT formula into 4 smaller DFTs. Hence

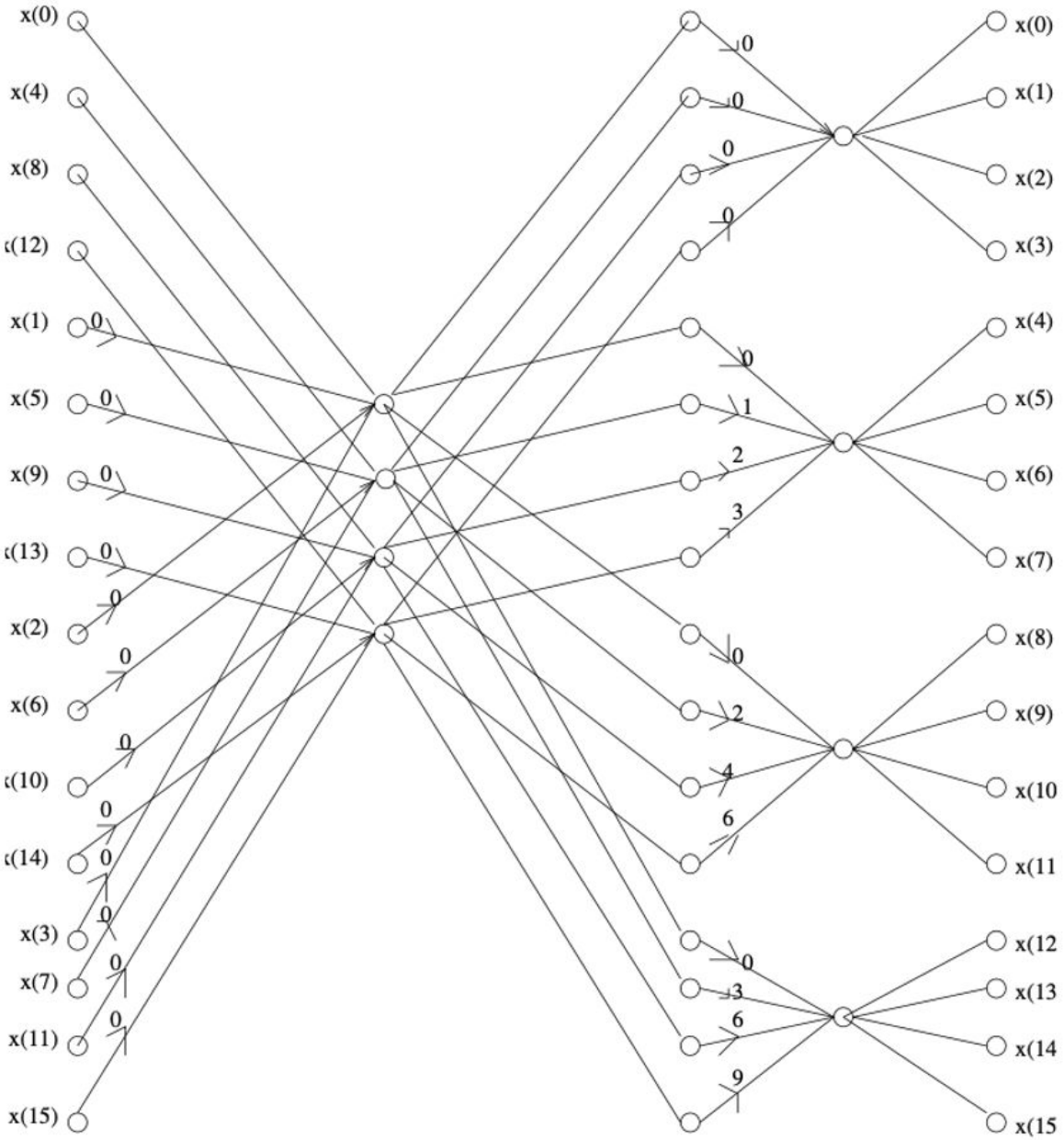
$$
\begin{aligned}
&X[k] \\
&= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
&= \sum_{n=0}^{N/4-1} x[n] W_N^{kn} + \sum_{n=N/4}^{N/2-1} x[n] W_N^{kn} + \sum_{n=N/2}^{3N/4-1} x[n] W_N^{kn} + \sum_{n=3N/4-1}^{N-1} x[n] W_N^{kn} \\
&= \sum_{n=0}^{N/4-1} x[n] W_N^{kn} + W_N^{Nk/4} \sum_{n=0}^{N/4-1} x\left[n+\frac{N}{4}\right] W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{N/4-1} x\left[n+\frac{N}{2}\right] W_N^{kn} + W_N^{3Nk/4} \sum_{n=0}^{N/4-1} x\left[n+\frac{3N}{4}\right] W_N^{kn} \\
&= \sum_{n=0}^{N/4-1} \left[x[n] + W_4^k x\left[n+\frac{N}{4}\right] + W_2^k x\left[n+\frac{N}{2}\right] + W_4^{3k} x\left[n+\frac{3N}{4}\right]\right] W_N^{kn}
\end{aligned}
$$

we then conclude that the larger N-point DFT values, for $k = 0, 1, \ldots, (N/4) - 1$, are given by the merging formulas

$$
\begin{aligned}
X[4k] &= \sum_{n=0}^{N/4-1} \left[x[n] + x\left[n+\frac{N}{4}\right] + x\left[n+\frac{N}{2}\right] + x\left[n+\frac{3N}{4}\right]\right] W_N^0 W_{N/4}^{kn}, \\
X[4k+1] &= \sum_{n=0}^{N/4-1} \left[x[n] + W_4^1 x\left[n+\frac{N}{4}\right] + W_2^1 x\left[n+\frac{N}{2}\right] + W_4^3 x\left[n+\frac{3N}{4}\right]\right] W_N^n W_{N/4}^{kn} \\
X[4k+2] &= \sum_{n=0}^{N/4-1} \left[x[n] + W_4^2 x\left[n+\frac{N}{4}\right] + W_2^2 x\left[n+\frac{N}{2}\right] + W_4^2 x\left[n+\frac{3N}{4}\right]\right] W_N^{2n} W_{N/4}^{kn} \\
X[4k+3] &= \sum_{n=0}^{N/4-1} \left[x[n] + W_4^3 x\left[n+\frac{N}{4}\right] + W_2^1 x\left[n+\frac{N}{2}\right] + W_4^1 x\left[n+\frac{3N}{4}\right]\right] W_N^{3n} W_{N/4}^{kn}
\end{aligned}
$$

The resulting the Radix-4 DIT-FFT butterfly flow graph is shown below.

Radix-4 DIF-FFT flow graph: Draw the diagram neatly on a paper, scan it as an image file (**`.jpeg`** or **`.png`**) and insert the image file.



(b) Compute the 16-point DFT of the sequence $x(n) = \cos(\pi n/2),\ 0 \le n \le 15$ using the radix-4 SFG developed in part (a) above.

Solution: First compute and then provide a drawing of a SFG that has node values filled in with your computed values.

With $k = 0$,

$$X[0] = \sum_{n=0}^{3} \left[x[n] + x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] + x\left[n + \frac{3N}{4}\right] \right] W_N^0 W_{N/4}^0 = 0,$$

$$X[1] = \sum_{n=0}^{3} \left[\left[x[n] - jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] + jx\left[n + \frac{3N}{4}\right] \right] W_N^n W_{N/4}^0 = 0,$$

$$X[2] = \sum_{n=0}^{3} \left[\left[x[n] - x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] - x\left[n + \frac{3N}{4}\right] \right] W_N^{2n} W_{N/4}^{0} = 0,$$

$$X[3] = \sum_{n=0}^{3} \left[\left[x[n] + jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] - jx\left[n + \frac{3N}{4}\right] \right] W_N^{3n} W_{N/4}^{0} = 0;$$

With $k = 1$,

$$X[4] = \sum_{n=0}^{3} \left[x[n] + x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] + x\left[n + \frac{3N}{4}\right] \right] W_N^{0} W_{N/4}^{n} = 8,$$

$$X[5] = \sum_{n=0}^{3} \left[\left[x[n] - jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] + jx\left[n + \frac{3N}{4}\right] \right] W_N^{n} W_{N/4}^{n} = 0,$$

$$X[6] = \sum_{n=0}^{3} \left[\left[x[n] - x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] - x\left[n + \frac{3N}{4}\right] \right] W_N^{2n} W_{N/4}^{n} = 0,$$

$$X[7] = \sum_{n=0}^{3} \left[\left[x[n] + jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] - jx\left[n + \frac{3N}{4}\right] \right] W_N^{3n} W_{N/4}^{n} = 0;$$

With $k = 2$,

$$X[8] = \sum_{n=0}^{3} \left[x[n] + x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] + x\left[n + \frac{3N}{4}\right] \right] W_N^{0} W_{N/4}^{2n} = 0,$$

$$X[9] = \sum_{n=0}^{3} \left[\left[x[n] - jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] + jx\left[n + \frac{3N}{4}\right] \right] W_N^{n} W_{N/4}^{2n} = 0,$$

$$X[10] = \sum_{n=0}^{3} \left[\left[x[n] - x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] - x\left[n + \frac{3N}{4}\right] \right] W_N^{2n} W_{N/4}^{2n} = 0,$$

$$X[11] = \sum_{n=0}^{3} \left[\left[x[n] + jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] - jx\left[n + \frac{3N}{4}\right] \right] W_N^{3n} W_{N/4}^{2n} = 0;$$

With $k = 3$,

$$X[12] = \sum_{n=0}^{3} \left[x[n] + x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] + x\left[n + \frac{3N}{4}\right] \right] W_N^{0} W_{N/4}^{3n} = 8,$$

$$X[13] = \sum_{n=0}^{3} \left[\left[x[n] - jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] + jx\left[n + \frac{3N}{4}\right] \right] W_N^{n} W_{N/4}^{3n} = 0,$$

$$X[14] = \sum_{n=0}^{3} \left[\left[x[n] - x\left[n + \frac{N}{4}\right] + x\left[n + \frac{N}{2}\right] - x\left[n + \frac{3N}{4}\right] \right] W_N^{2n} W_{N/4}^{3n} = 0,$$

$$X[15] = \sum_{n=0}^{3} \left[\left[x[n] + jx\left[n + \frac{N}{4}\right] - x\left[n + \frac{N}{2}\right] - jx\left[n + \frac{3N}{4}\right] \right] W_N^{3n} W_{N/4}^{3n} = 0.$$

Problem-4: Text Problem 8.19 (page-568)

Let $X(k)$ be the N-point DFT of the sequence $x(n)$, $0 \le n \le N-1$.

(a) What is the N-point DFT $S(k)$ of the sequence $s(n) = X(n)$, $0 \le n \le N-1$?

Solution:

$$
\begin{aligned}
S(k) &= \sum_{n=0}^{N-1} s(n) W_N^{kn} = \sum_{n=0}^{N-1} X(n) W_N^{kn} = \sum_{n=0}^{N-1} \left(\sum_{t=0}^{N-1} x(t) W_N^{tn} \right) W_N^{kn} \\
&= \sum_{t=0}^{N-1} x(t) \sum_{n=0}^{N-1} W_N^{tn} W_N^{kn} = \sum_{t=0}^{N-1} x(t) \sum_{n=0}^{N-1} W_N^{(t+k)n} \\
&= \sum_{t=0}^{N-1} x(t) N\delta(N-t-k) = N\{\underset{\uparrow}{x(0)}, x(N-1), \cdots, x(1)\} \quad \text{for} \quad k \in \{0, 1, \cdots, N-1\} \\
&= Nx(\langle -k \rangle_N)
\end{aligned}
$$

which is a scaled and circularly-folded version of $x(n)$.

(b) Let $Y(k)$ be the N-point DFT of the sequence $y(n) = S(n)$, $0 \le n \le N-1$. What is the N-point DFT $V(k)$ of the sequence $v(n) = Y(n)$, $0 \le n \le N-1$?

Solution: From part (a) we observe that $\underset{N\text{-point}}{\text{DFT}}\left[\underset{N\text{-point}}{\text{DFT}}[x(n)]\right] = Nx(\langle -k \rangle_N)$, that is, twice DFT operation on a sequence results in a scaled (by N) and circularly folded version of that sequence. Now $V(k)$ is obtained by taking two consecutive DFTs on the sequence $S(n)$. Therefore,

$$
\begin{aligned}
V(k) &= \underset{N\text{-point}}{\text{DFT}}\left[\underset{N\text{-point}}{\text{DFT}}[S(n)]\right] = N\,S(\langle -k \rangle_N) = N\left\{Nx(\langle -(-k) \rangle_N)\right\} \\
&= N^2 x(k)
\end{aligned}
$$

Thus, four consecutive operations of DFT on an arbitrary sequence $x(n)$ results in the same sequence, scaled by N^2. This technique is used to verify if a new FFT code is correct (or valid). Just run an arbitrary sequence through the new code four times, divide the result by N^2 and check if you get the original sequence back.

(c) Let $x(n) = [1:8]$. Using MATLAB verify your results in (a) and (b).

MATLAB script:

```
clc; close all; clear;
N= 8; xn = 1:8;
Xk = fft(xn, N); Sk = fft(Xk, N); Yk = fft(Sk,N); Vk = fft(Yk,N);
x_check = Vk/N^2
```

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

Problem-5: Radix-3 FFT Algorithm

Let $N = 3^{\nu}$ where ν is a positive integer.

(a) Derive a radix-3 DIT-FFT algorithm for this case. Draw a radix-3 DIT-FFT butterfly flow graph using merging formulas.

Solution: Since we have $N = 3^{\nu}$, we decimate the time-domain sequence by choosing every-third sample. Hence

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} = \sum_{m=0}^{N/3-1} x[3m]W_N^{k(3m)} + \sum_{m=0}^{N/3-1} x[3m+1]W_N^{k(3m+1)} + \sum_{m=0}^{N/3-1} x[3m+2]W_N^{k(3m+2)}$$

$$= \left(\sum_{m=0}^{N/3-1} x[3m]W_{N/3}^{km}\right) + \left(\sum_{m=0}^{N/3-1} x[3m+1]W_{N/3}^{km}\right)W_N^k + \left(\sum_{m=0}^{N/3-1} x[3m+2]W_{N/3}^{km}\right)W_N^{2k}$$

If we define the following: $a[n] = x[3n]$, $b[n] = x[3n+1]$, $c[n] = x[3n+2]$, $n = 0, 1, \ldots, \frac{N}{3} - 1$, and

$$A[k] = \sum_{n=0}^{N/3-1} a[n]W_{N/3}^{kn}, \quad B[k] = \sum_{n=0}^{N/3-1} b[n]W_{N/3}^{kn}, \quad C[k] = \sum_{n=0}^{N/3-1} c[n]W_{N/3}^{kn}, k = 0, 1, \ldots, N/3 - 1$$

we then conclude that the larger N-point DFT values, for $k = 0, 1, \ldots, (N/3) - 1$, are given by the merging formulas

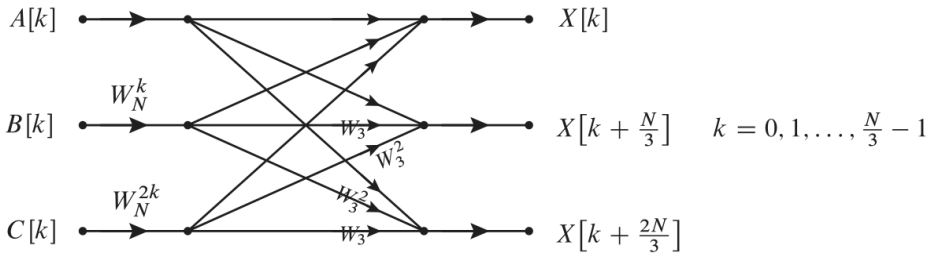
$$X[k] = A[k] + \left(B[k]W_N^k\right) + \left(C[k]W_N^{2k}\right),$$

$$X\left[k + \frac{N}{3}\right] = A[k] + B[k]W_N^k W_3^{\frac{N}{3}} + C[k]W_N^{2k}W_3^{\frac{2N}{3}} = A[k] + \left(B[k]W_N^k\right)W_3 + \left(C[k]W_N^{2k}\right)W_3^2$$

$$X\left[k + \frac{2N}{3}\right] = A[k] + B[k]W_N^k W_3^{\frac{2N}{3}} + C[k]W_N^{2k}W_3^{\frac{N}{3}} = A[k] + \left(B[k]W_N^k\right)W_3^2 + \left(C[k]W_N^{2k}\right)W_3$$

The resulting the Radix-3 DIT-FFT butterfly flow graph is shown below.

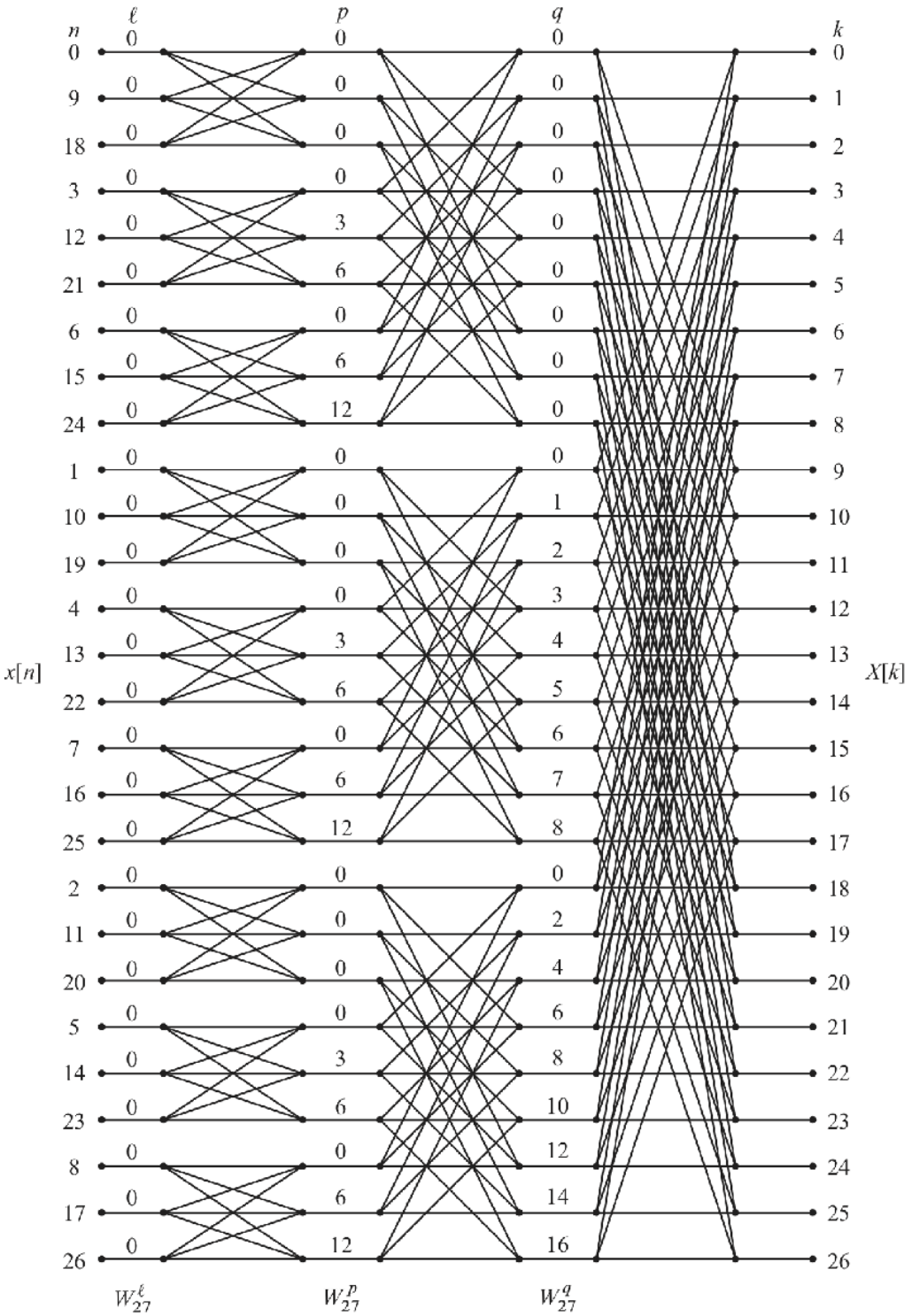
Radix-3 DIT-FFT butterfly flow graph: Draw the diagram neatly on a paper, scan it as an image file (**`.jpeg`** or **`.png`**) and insert the image file.



(b) Draw the complete flow-chart for $N = 27$. MATLAB drawing would be tedious so draw it neatly on a paper, scan it as an image file (**`.jpeg`** or **`.png`**) and insert the image file.

Solution: Since $N = 27 = 9 \times 3 = 3 \times 3 \times 3$, we will need three-stages of Radix-3 butterflies. The resulting flow graph is shown below.

Flow-graph for $N = 27$:

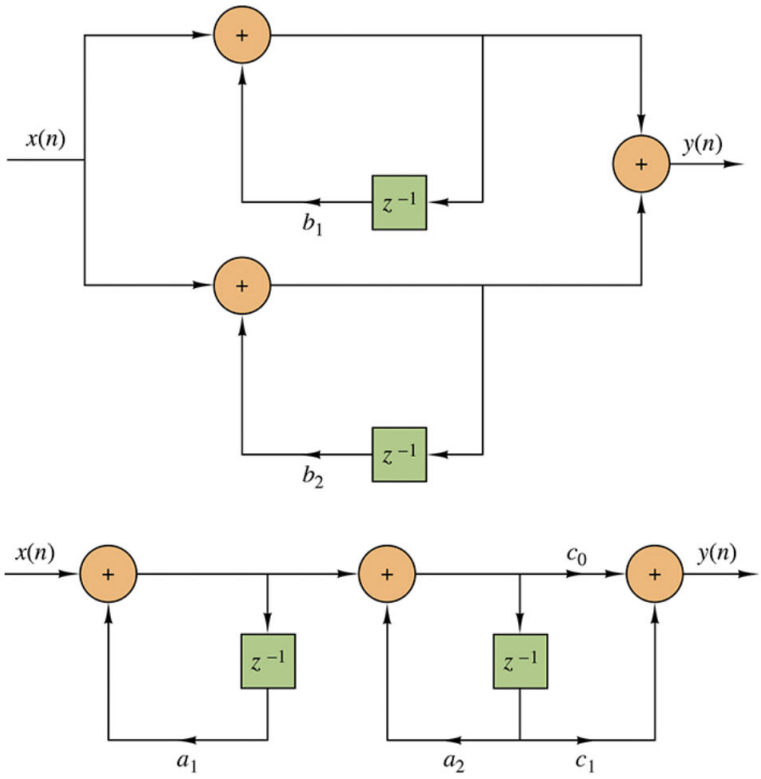


(c) Determine the total number of complex multiplications needed to implement the algorithm in part (b) above.

Answer: There are 6 complex multiplications per butterfly, 9 butterflies per stage, and 3 stages in the implementation of the above flow-graph. Hence the total number of complex multiplications needed to implement 27-point DFT is $6 \times 9 \times 3 = 162$.

Problem-6: Text Problem 9.6 (page-652)

Determine a_1, a_2 and c_1, and c_0 in terms of b_1 and b_2 so that the two systems in Fig. P9.6 (shown below) are equivalent.



Solution:

For the first system, we have the transfer function as:

$$H(z) = \frac{1}{1 - b_1 z^{-1}} + \frac{1}{1 - b_2 z^{-1}} = \frac{2 - (b_1 + b_2) z^{-1}}{\left(1 - b_1 z^{-1}\right)\left(1 - b_2 z^{-1}\right)}$$

For the second system, we have the transfer function as:

$$H(z) = \frac{c_0 + c_1 z^{-1}}{\left(1 - a_1 z^{-1}\right)\left(1 - a_2 z^{-1}\right)}$$

Thus, by comparing those two transfer function, we can see that the two system are equivalent when:

$$c_0 = 2, c_1 = -(b_1 + b_2), a_1 = b_1, a_2 = b_2$$

or

$$c_0 = 2, c_1 = -(b_1 + b_2), a_1 = b_2, a_2 = b_1$$

Problem-7:

The FIR system is given by

$$H(z) = 5.84 - 20.99z^{-1} + 35.3z^{-2} - 35.3z^{-3} + 20.99z^{-4} - 5.84z^{-5}.$$

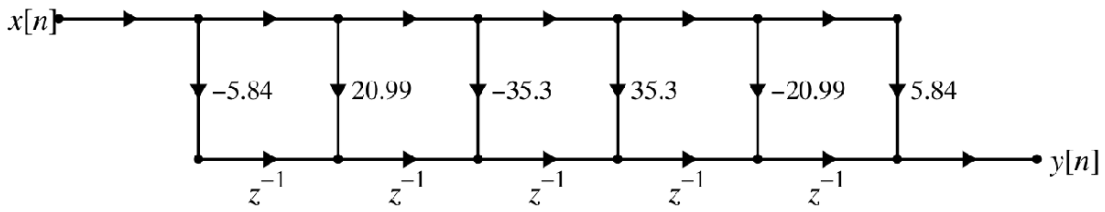
Determine and draw the following structures.

Notes:

1. Neatly draw signal flow graphs for the following parts on a paper and then scan and import them as **`.png`** or **`.jpeg`** file.
2. If a parenthetical description is not given then assume normal form.

(a) Direct form II (transposed)

Signal flow graph:



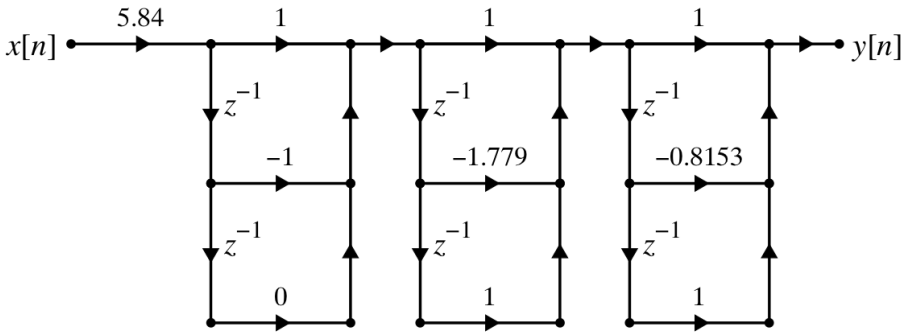
(b) Cascade form

MATLAB script:

```
clear; close all; clc;
h = [5.84,-20.99,35.3,-35.3,20.99,-5.84];
[sos,G] = tf2sos(h,1)
```

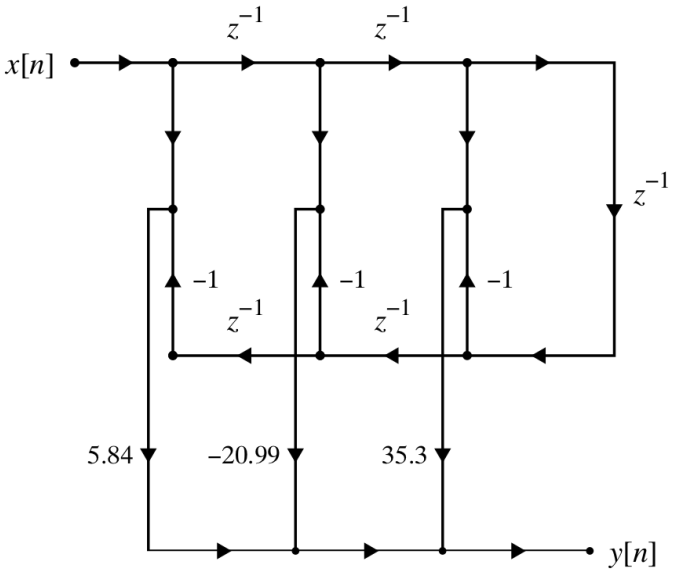
```
sos = 3×6
    1.0000   -1.0000         0    1.0000         0         0
    1.0000   -0.8153    1.0000    1.0000         0         0
    1.0000   -1.7789    1.0000    1.0000         0         0
G = 5.8400
```

Signal flow graph:



(c) Linear-phase form

Signal flow graph:



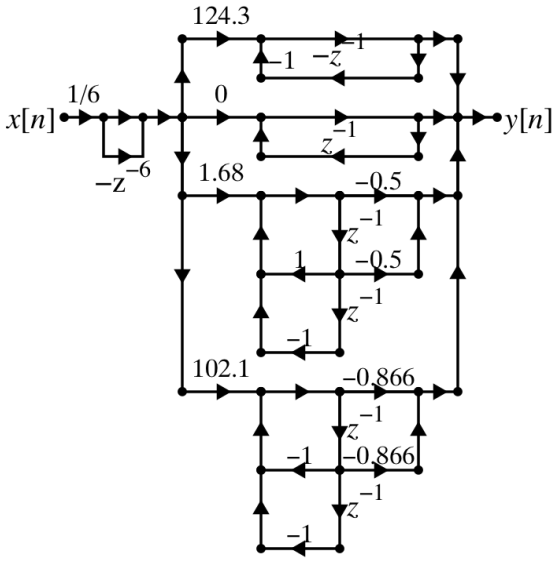
(d) Frequency-sampling form

MATLAB script:

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

```
C = 4×1
    1.6800
  102.0524
         0
  124.2600
B = 2×2
   -0.5000   -0.5000
   -0.8660   -0.8660
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
```

Signal flow graph:



Problem-8: Text Problem 9.9(d)

Obtain the direct form I, direct form II, cascade, and parallel structures for the following system:

$$H(z) = \frac{2(1 - z^{-1})(1 + \sqrt{2}\,z^{-1} + z^{-2})}{(1 + 0.5z^{-1})(1 - 0.9z^{-1} + 0.81z^{-2})}$$

Note: Neatly draw signal flow graphs for the following parts on a paper and then scan and import them as **`.png`** or **`.jpeg`** file.

```
% clc; clear;
```

Note that the given system function is already in the cascade form.

```
sos = [1,-1,0,1,0.5,0;1,sqrt(2),1,1,-0.9,0.81], % second-order sections
```

```
sos = 2×6
    1.0000   -1.0000         0    1.0000    0.5000         0
    1.0000    1.4142    1.0000    1.0000   -0.9000    0.8100
```

```
G = 2, % overall gain
```

```
G = 2
```

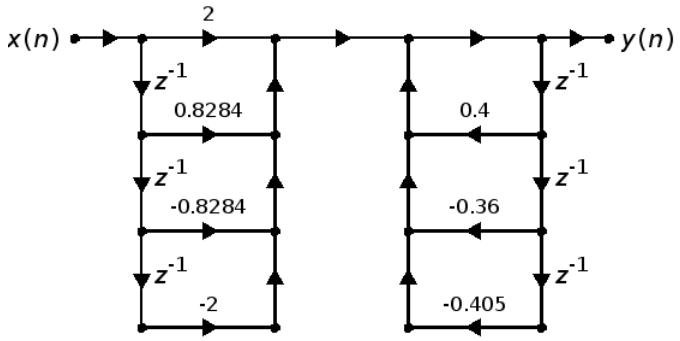
(i) Direct form I (normal):

```
[b,a] = sos2tf(sos,G)
```

```
b = 1×4
    2.0000    0.8284   -0.8284   -2.0000
a = 1×4
```

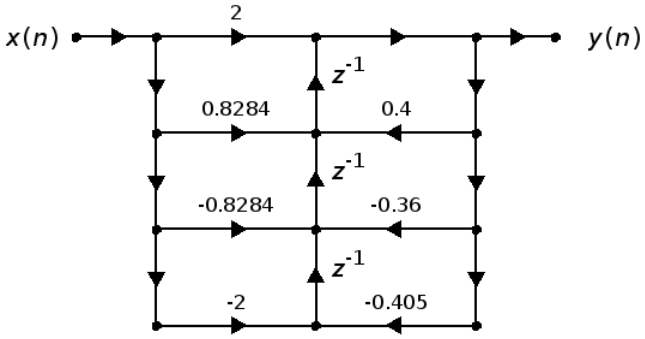
```
    1.0000   -0.4000    0.3600    0.4050
```

Signal flow graph:



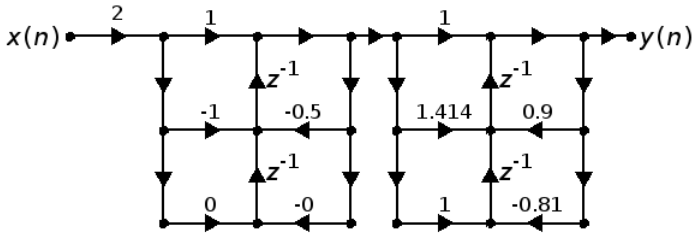
(il) Direct form II (transposed):

Signal flow graph:



(iii) Cascade form: Draw second-order sections using transposed direct form II structures

Signal flow graph:



(iv) Parallel form I: Draw second-order sections using normal direct form II structures

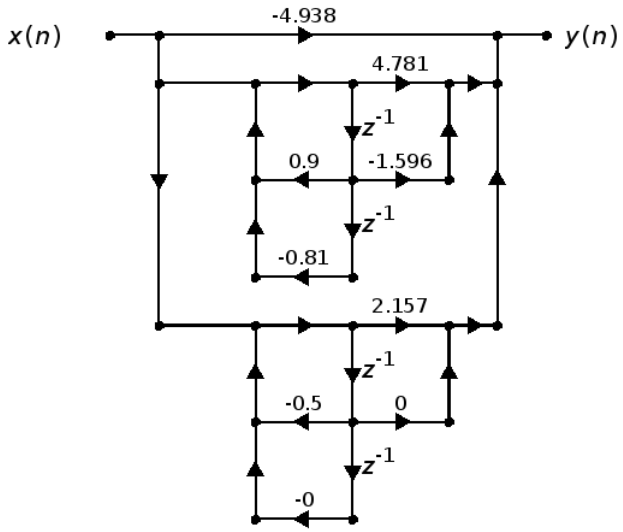
Solution:

```
[C,B,A] = dir2par(b,a)
```

```
C = -4.9383
B = 2×2
```

```
    4.7811   -1.5959
    2.1572         0
A = 2×3
    1.0000   -0.9000    0.8100
    1.0000    0.5000         0
```

Signal flow graph:



Problem-9:

Consider the IIR cascade form structure given in the Text **Figure 9.3.8 (a)** in which each second-order section is implemented using the transposed direct form II structure of Table 9.1. The basic **`filter`** function from MATLAB implements transposed direct form II structure.

(a) Develop a MATLAB function **`y = filtercf0(sos,G,x)`** that implements cascade form structure of **Figure 9.3.8(a)** in which each second order section is implemented by the **`filter`** function. Assume zero initial conditions.

Note: Provide your function code below in the `Code Example` area for the TA to grade it. Provide the same code at the end of this file for it to execute properly.

MATLAB function: The MATLAB function is given below as a code example.

```
function y = filtercf0(sos,G,x)
% Implements IIR cascade form given in Figure 9.11 using filter function
N = length(x); K = size(sos,1);
B = sos(:,1:3); A = sos(:,4:6); % numerators and denominatyors of each section
w = zeros(K+1,N); w(1,:) = (x(:))'; % first row of w is the input
for k = 1:K
    w(k+1,:) = filter(B(k,:),A(k,:),w(k,:));
end
y = G*w(K+1,:);
end
```

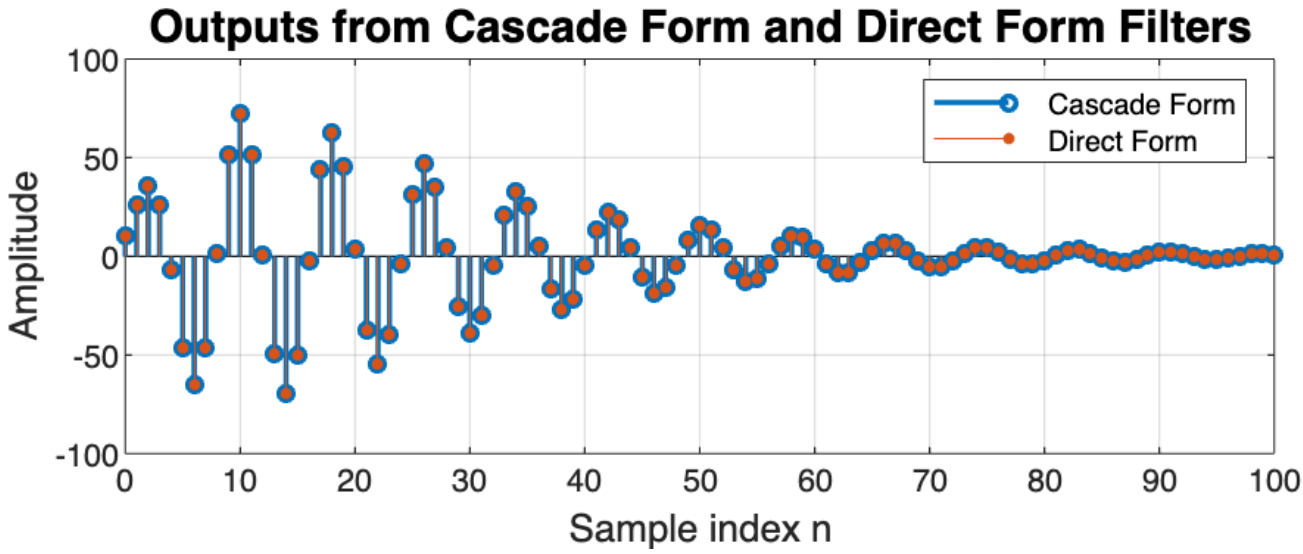
(b) Consider the system function given below.

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.81z^{-3} - 5.83z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}.$$

Convert the above $H(z)$ into a cascade form. Using your **`filtercf0`** function from part (a), compute the impulse response $h[n]$, $0 \le n \le 100$ of the system. Also compute the impulse response using the **`filter`** function and the direct form coefficients. To verify that your **`filtercf0`** function is correctly implemented, compare your two impulse response sequences by **`stem`** graphing them on the same plot. Use non-filled stem markers of size 6 for the **`filtercf0`** output samples and filled stem markers of size 3 for the **`filter`** output samples. Comment on your plot observation.

MATLAB script and plot:

```
% clear; close all; clc
b = [10 1 0.9 0.81 -5.83]; a = [1 -2.54 3.24 -2.06 0.66]; % direct form
[sos,G] = tf2sos(b,a); % cascade form
n = 0:100; xn = (delta(0,0,100))'; % impulse sequence
ycf = filtercf0(sos,G,xn); % output using cascade form filter
ydf = filter (b,a,xn); % output using direct form filter
% Plots of two output sequences
figure('PaperPosition',[0,0,8,3]*72,'position',[0,0,8,3]*72);
stem(n,ycf,'linewidth',1.5,'MarkerSize',4); hold on;
stem(n,ydf,'filled','linewidth',0.25,'MarkerSize',3);
xlabel('Sample index n'); ylabel('Amplitude'); grid;
title('Outputs from Cascade Form and Direct Form Filters');
legend('Cascade Form','Direct Form','location','northeast'); hold off;
```



Comment: Both filtering implementations give the same exact results.

Problem-10:

An IIR system is given by

$$H(z) = (37.8 - 2.05z^{-1}) + \frac{-28.64 + 18.86z^{-1}}{1 - 0.32z^{-1} + 0.56z^{-2}} + \frac{-5 - 12.31z^{-1}}{1 - 0.93z^{-1} + 0.58z^{-2}}.$$

Determine and draw the following structures. Neatly draw signal flow graphs for the following parts on a paper and then scan and import them as **`.png`** files.

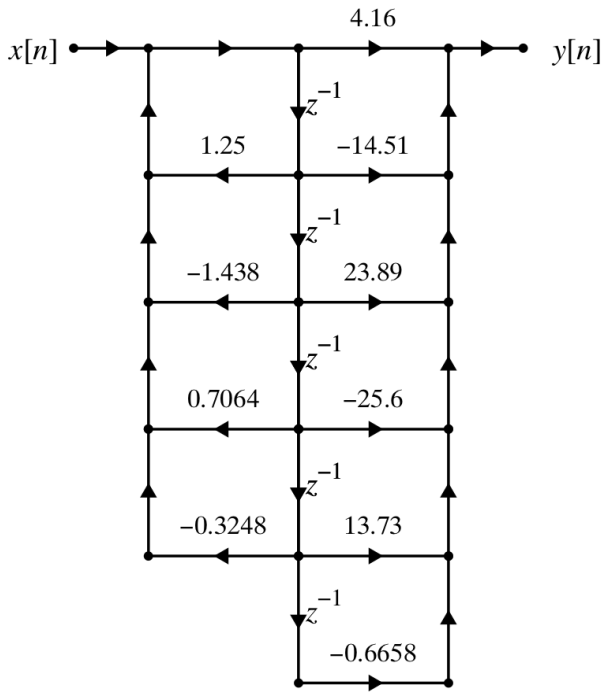
(a) Direct form II (normal)

MATLAB Script: We first need to determine the rational function representation for $H(z)$.

```
% clear; close all; clc;
C = [37.8,-2.05]; B = [-28.64,18.86;-5,-12.31];
A = [1,-0.32,0.56;1,-0.93,0.58];
[b,a] = par2dir(C,B,A); b = real(b), a
```

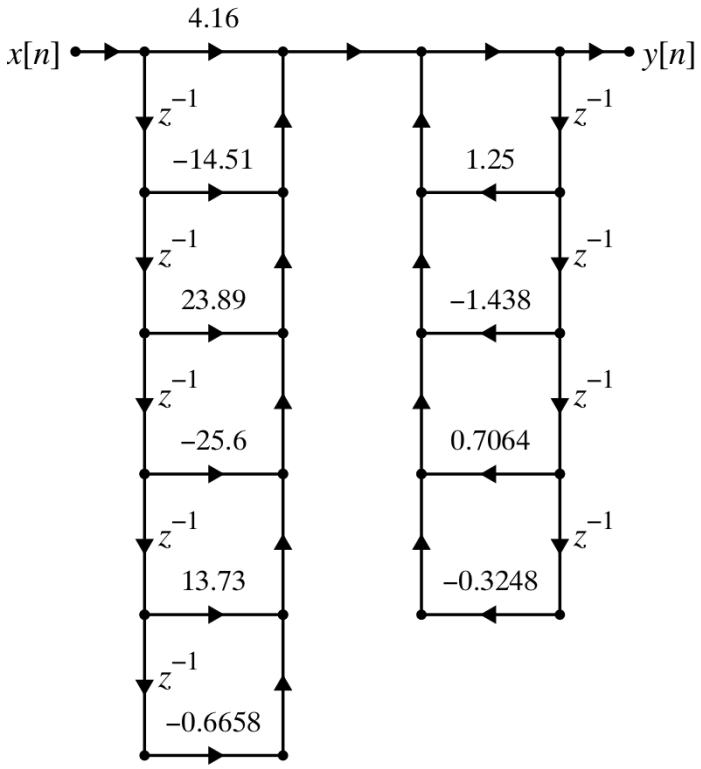
```
b = 1×6
    4.1600  -14.5148   23.8920  -25.6038   13.7256   -0.6658
a = 1×5
    1.0000   -1.2500    1.4376   -0.7064    0.3248
```

Signal Flow Graph:



(b) Direct form I (normal)

Signal Flow Graph:



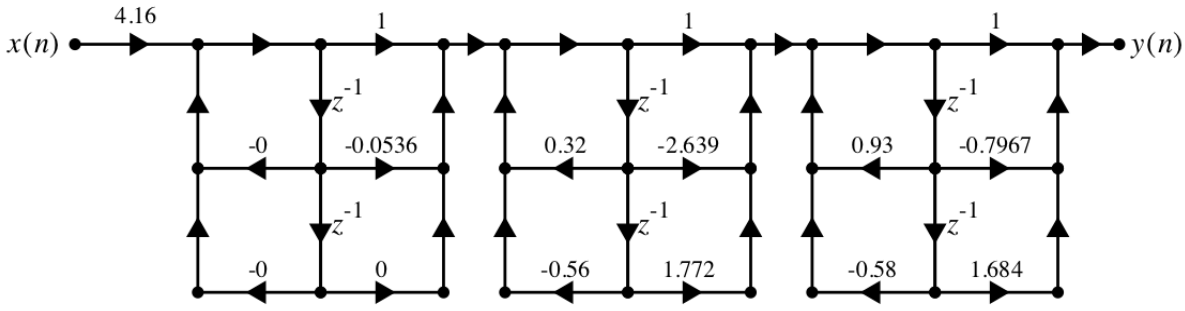
(c) Cascade form with normal second-order sections

MATLAB script:

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

```
sos = 3×6
    1.0000   -0.0536         0    1.0000         0         0
    1.0000   -2.6389    1.7725    1.0000   -0.3200    0.5600
    1.0000   -0.7967    1.6843    1.0000   -0.9300    0.5800
G = 4.1600
```

Signal Flow Graph:



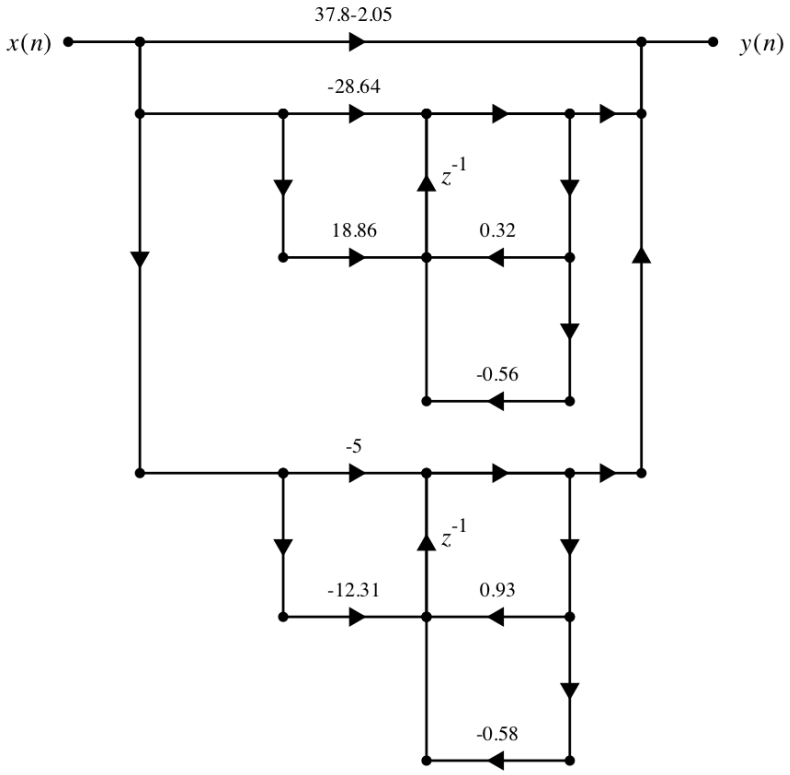
(d) Parallel form with transposed second-order sections

```
[C,B,A] = dir2par(b,a)
```

```
C = 1×2
   37.8000   -2.0500
B = 2×2
  -28.6400   18.8600
```

```
   -5.0000  -12.3100
A = 2×3
    1.0000   -0.3200    0.5600
    1.0000   -0.9300    0.5800
```

Signal Flow Graph:




```
function y = filtercf0(sos,G,x)
% Implements IIR cascade form given in Figure 9.11 using filter function
N = length(x); K = size(sos,1);
B = sos(:,1:3); A = sos(:,4:6); % numerators and denominatyors of each section
w = zeros(K+1,N); w(1,:) = (x(:))'; % first row of w is the input
for k = 1:K
    w(k+1,:) = filter(B(k,:),A(k,:),w(k,:));
end
y = G*w(K+1,:);
end
```