Name: Do Trung Hau

Section:

Laboratory Exercise 2

DISCRETE-TIME SYSTEMS: TIME-DOMAIN REPRESENTATION

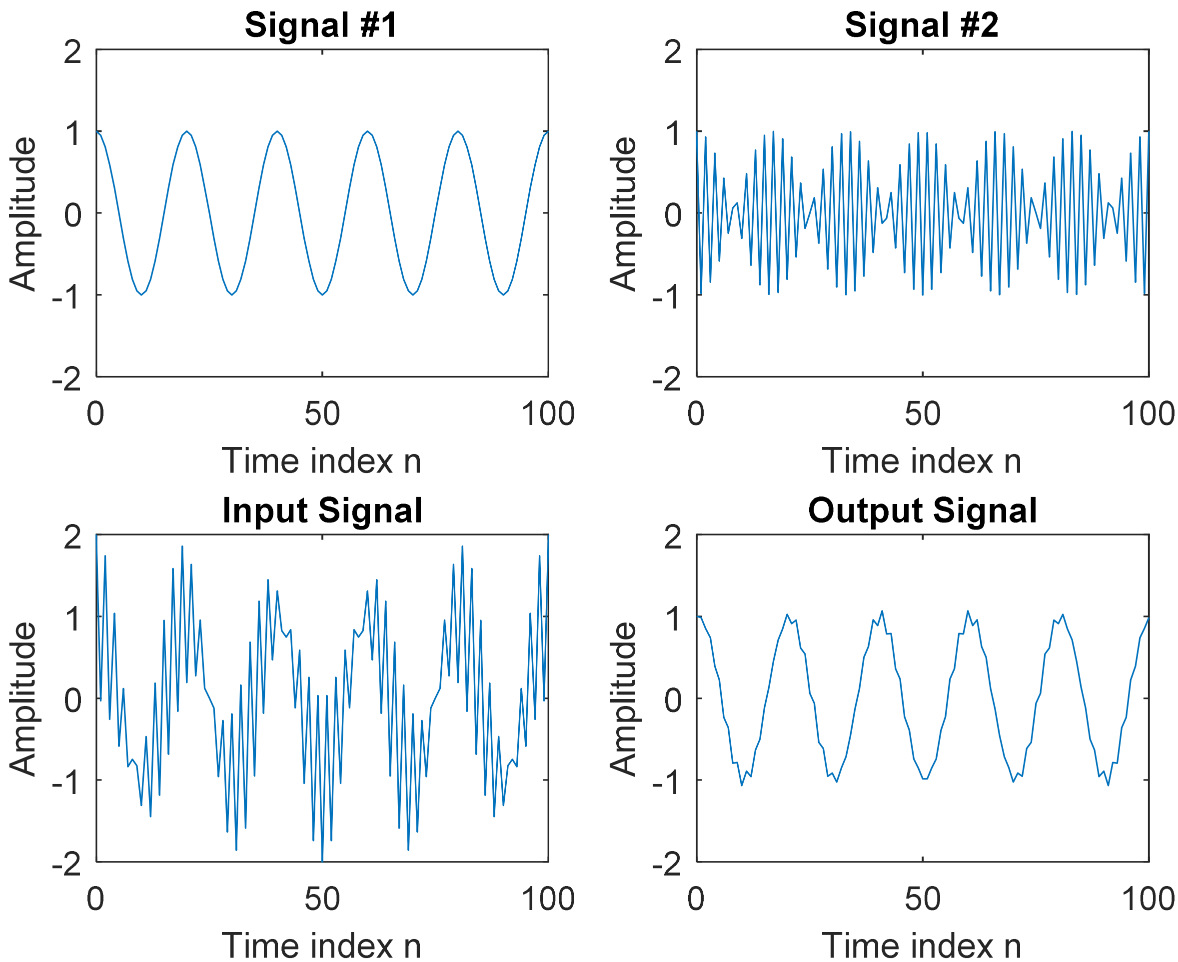
2.1 SIMULATION OF DISCRETE-TIME SYSTEMS

Project 2.1 The Moving Average System

A copy of Program P2_1 is given below:

```
% Program P2_1
% Simulation of an M-point Moving Average Filter
% Generate the input signal
clc; clear all; close all;
n = 0:100;
s1 = cos(2*pi*0.05*n); % A low-frequency sinusoid
s2 = cos(2*pi*0.47*n); % A high frequency sinusoid
x = s1+s2;
% Implementation of the moving average filter
M = input('Desired length of the filter = ');
num = ones(1,M);
y = filter(num,1,x)/M;
% Display the input and output signals
subplot(2,2,1);
plot(n, s1);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #1');
subplot(2,2,2);
plot(n, s2);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #2');
subplot(2,2,3);
plot(n, x);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Input Signal');
subplot(2,2,4);
plot(n, y);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Signal');
axis;
```

Q2.1 The output sequence generated by running the above program for M = 2 with x[n] = s1[n] + s2[n] as the input is shown below.



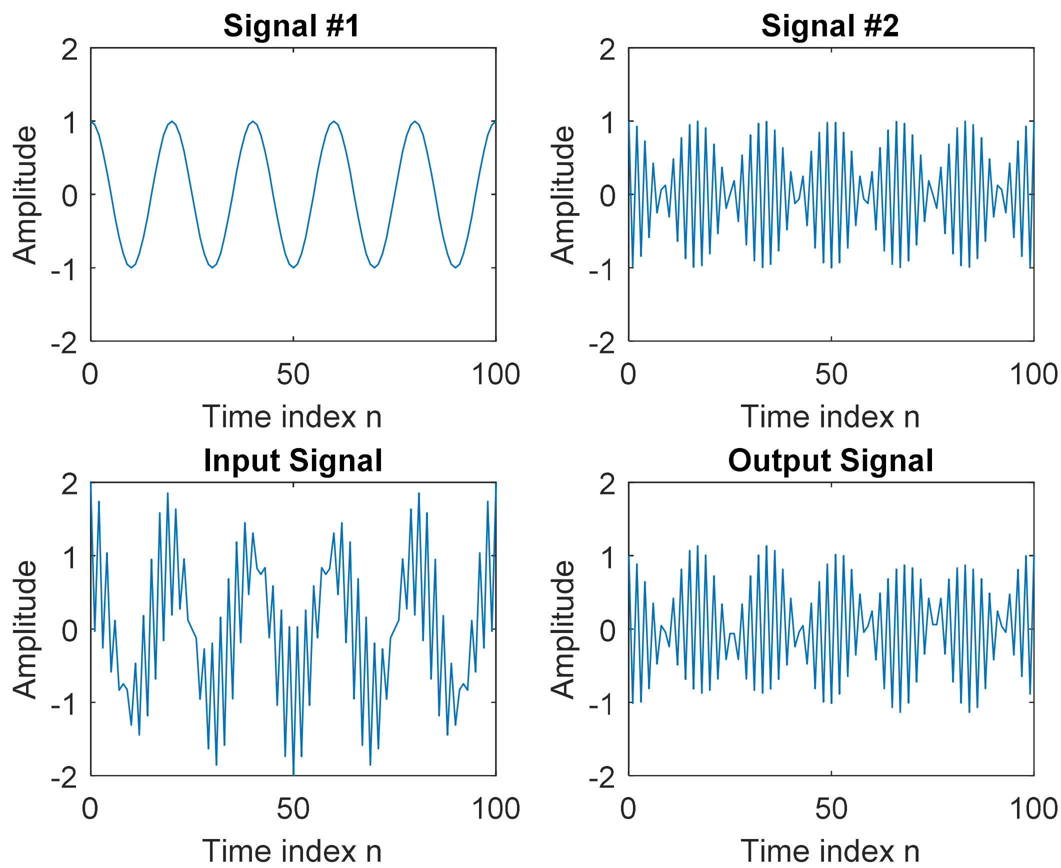
The component of the input x[n] suppressed by the discrete-time system simulated by this program is – ***Signal 2 with higher frequency of 0.47 → This filter is maybe a LPF.***

Q2.2 Program P2_1 is modified to simulate the LTI system $y[n] = 0.5\left(x[n] - x[n-1]\right)$ and process the input $x[n] = s_1[n] + s_2[n]$ resulting in the output sequence shown below:

```
% Simulation of an M-point Moving Average Filter
% Generate the input signal
clc; clear all; close all;
n = 0:100;
s1 = cos(2*pi*0.05*n); % A low-frequency sinusoid
s2 = cos(2*pi*0.47*n); % A high frequency sinusoid
x = s1+s2;
% Implementation of the moving average filter
M = input('Desired length of the filter = ');
num = [1,-1];
y = filter(num,1,x)/M;
```

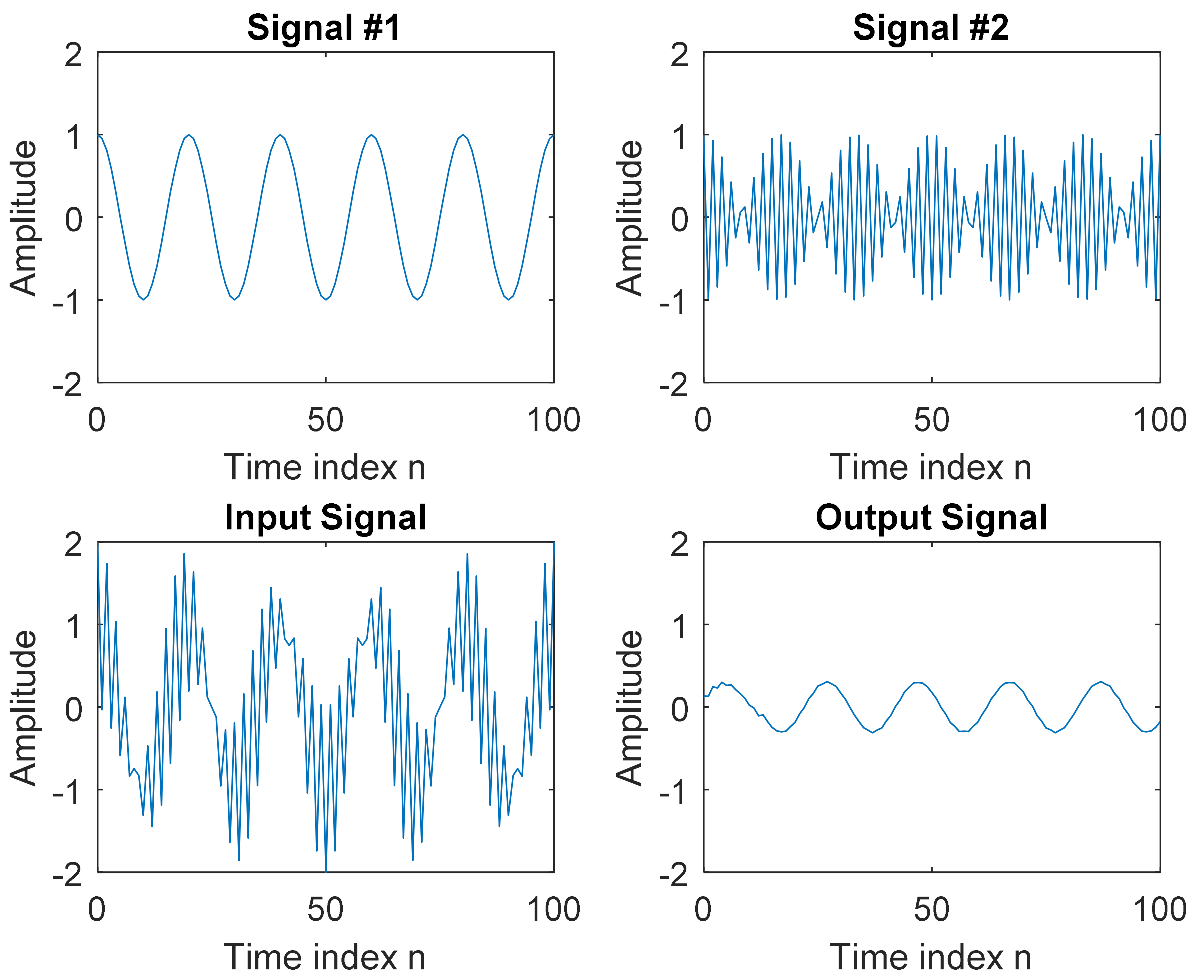
```
% Display the input and output signals
subplot(2,2,1);
plot(n, s1);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #1');
subplot(2,2,2);
plot(n, s2);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #2');
subplot(2,2,3);
plot(n, x);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Input Signal');
subplot(2,2,4);
plot(n, y);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Signal');
axis;
```

The effect of changing the LTI system on the input is ***making a HPF instead of LPF as before.***



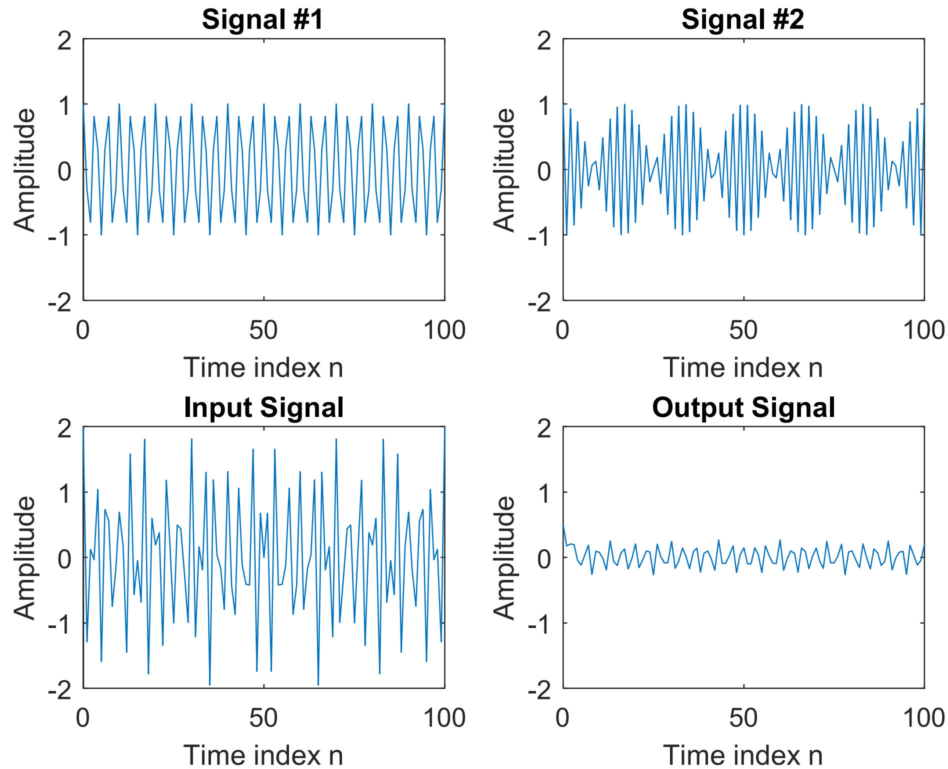
Q2.3 Program P2_1 is run for the following values of filter length M and following values of the frequencies of the sinusoidal signals s1[n] and s2[n]. The output generated for these different values of M and the frequencies are shown below.

- ***With*** $f_1 = 0.05;\;\; f_2 = 0.47;\;\; M = 15$ ***we obtain:***



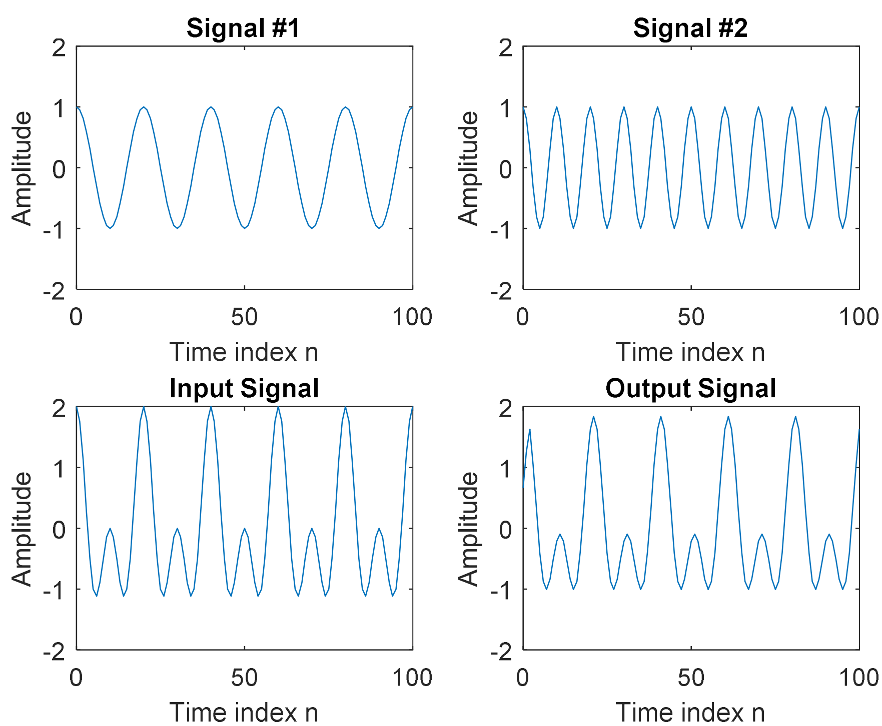
→ From this plots we make the following observation that: ***The high frequency s2 signal is still removed in the filter output. The low frequency signal s1 is passed but is attenuated in amplitude.***

- ***With*** $f_1 = 0.3;\;\; f_2 = 0.47;\;\; M = 4$ ***we obtain:***



→ From this plots we make the following observation that: ***Now, both signals s1 and s2 are high frequency signals so they are removed in the filter output.***

- ***With*** $f_1 = 0.05; \;\; f_2 = 0.1; \;\; M = 3$ ***we obtain:***

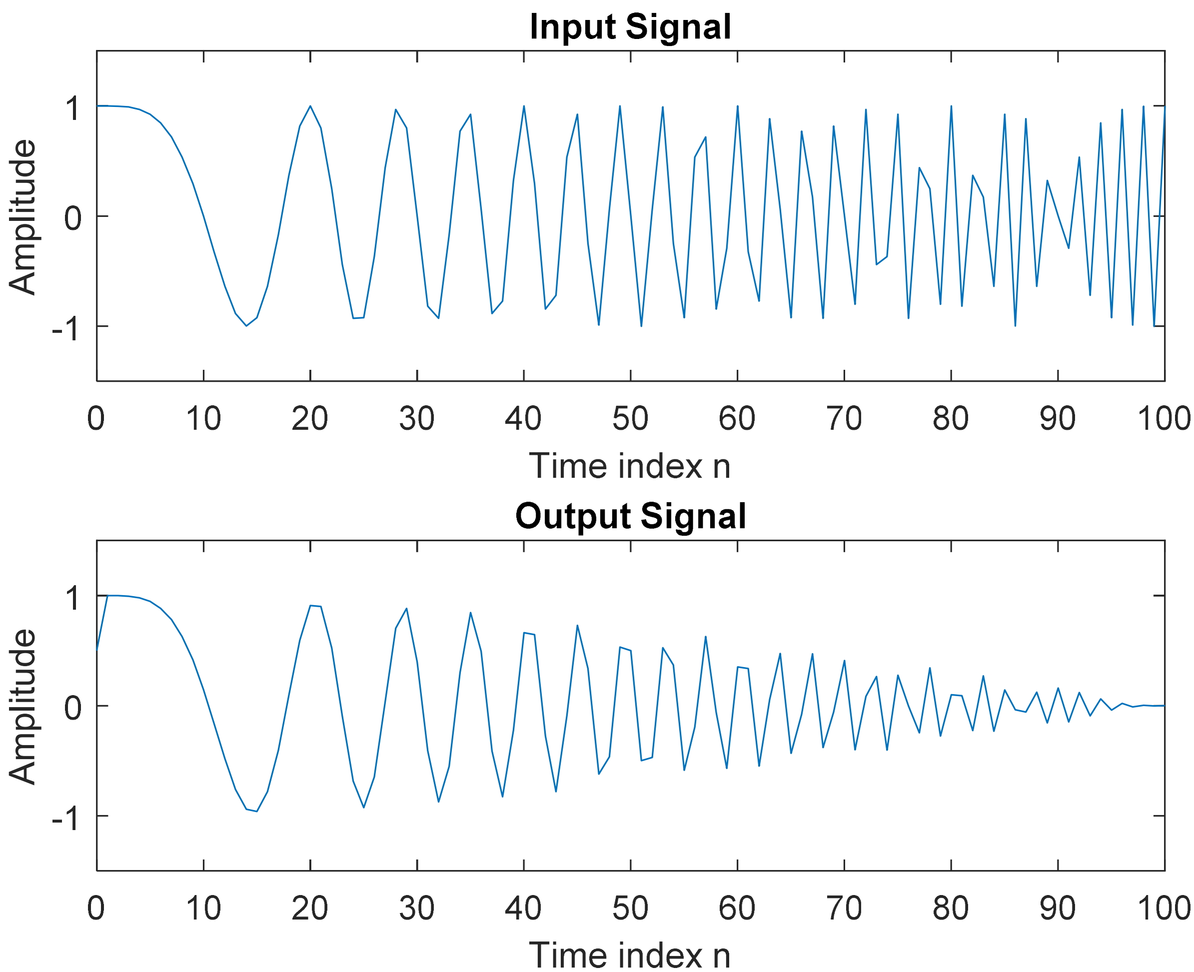


→ From this plots we make the following observation that: ***Now, both signals s1 and s2 are low frequency signals so they are passed in the filter output.***

Q2.4 The required modifications to Program P2_1 by changing the input sequence to ***a swept-frequency sinusoidal signal*** (length 101, minimum frequency 0, and a maximum frequency 0.5) as the input signal (see Program P1_7) are listed below:

```
% Generate the input signal
clc; clear all; close all;
n = 0:100;
a = pi/200;
b = 0;
arg = a*n.*n + b*n;
x = cos(arg);
% Implementation of the moving average filter
M = input('Desired length of the filter = ');
num = ones(1,M);
y = filter(num,1,x)/M;
subplot(2,1,1);
plot(n, x);
axis([0, 100, -1.5, 1.5]);
xlabel('Time index n'); ylabel('Amplitude');
title('Input Signal');
subplot(2,1,2);
plot(n, y);
axis([0, 100, -1.5, 1.5]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Signal');
axis;
```

The output signal generated by running this program is plotted below.



The results of Questions Q2.1 and Q2.2 from the response of this system to the swept-frequency signal can be explained as follows:

- ***Observing the input and output signals of the system we see that the low-frequency parts of the left signal are passed to the filter output, and the high-frequency parts of the signal are subsequently removed from the output. The frequency increases as n increases, resulting in the signal being attenuated along the increasing of n.***
- ***From*** $\arg = an^2 + bn \Rightarrow \omega = \left(an^2 + bn\right)' = 2an + b = 2.\frac{\pi}{200}.n \Rightarrow f = \frac{\omega}{2\pi} = \frac{n}{200}$. ***In Q2.1 we have the frequency of 0.05 at*** $n = 10$ ***.That's why the signal 1 is passed in the filter output. Likewise, the frequency of signal 2 is 0.47 at*** $n = 94$ ***, that's why the signal 2 is attenuated in the filter output.***
- ***The filter in Q2.2 is opposite to Q2.1 when it's HPF. That's why the signal s1 with low frequency is removed in the output where as the signal s2 is passed.***

Project 2.2 (Optional) A Simple Nonlinear Discrete-Time System

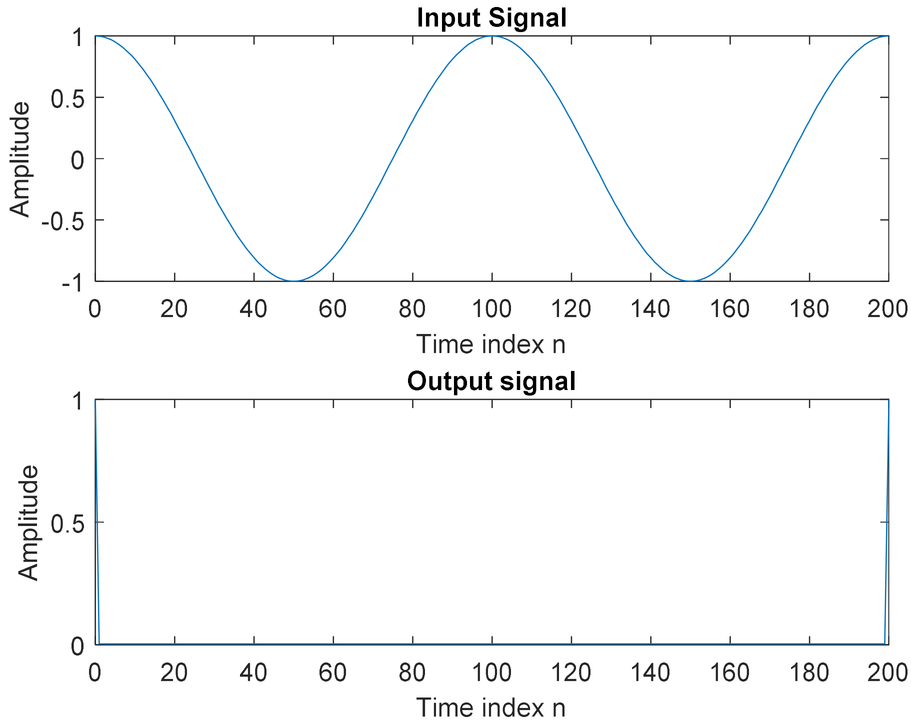
A copy of Program P2_2 is given below:

```
% Program P2_2
% Generate a sinusoidal input signal
clc; clear all; close all;
n = 0:200;
x = cos(2*pi*0.05*n);
% Compute the output signal
x1 = [x 0 0];          % x1[n] = x[n+1]
x2 = [0 x 0];          % x2[n] = x[n]
x3 = [0 0 x];          % x3[n] = x[n-1]
y = x2.*x2-x1.*x3;
y = y(2:202);
% Plot the input and output signals
subplot(2,1,1)
plot(n, x)
xlabel('Time index n');ylabel('Amplitude');
title('Input Signal')
subplot(2,1,2)
plot(n,y)
xlabel('Time index n');ylabel('Amplitude');
title('Output signal');
```

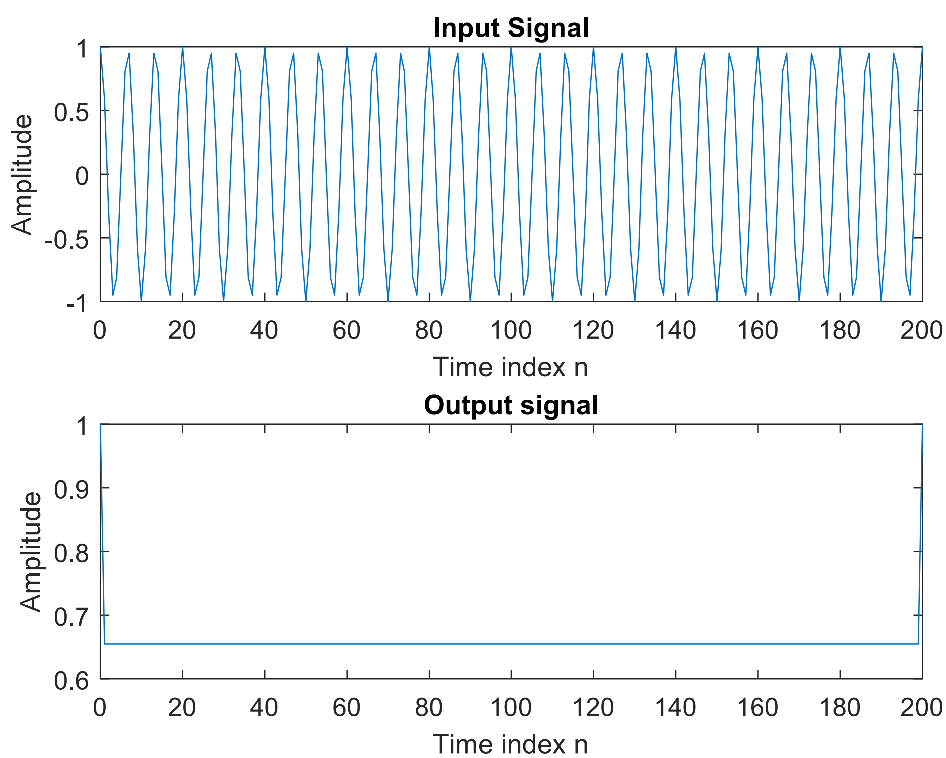
Q2.5 The sinusoidal signals with the following frequencies as the input signals were used to generate the output signals: $f = 0.01$; $f = 0.15$; $f = 0.25$

The output signals generated for each of the above input signals are displayed below:

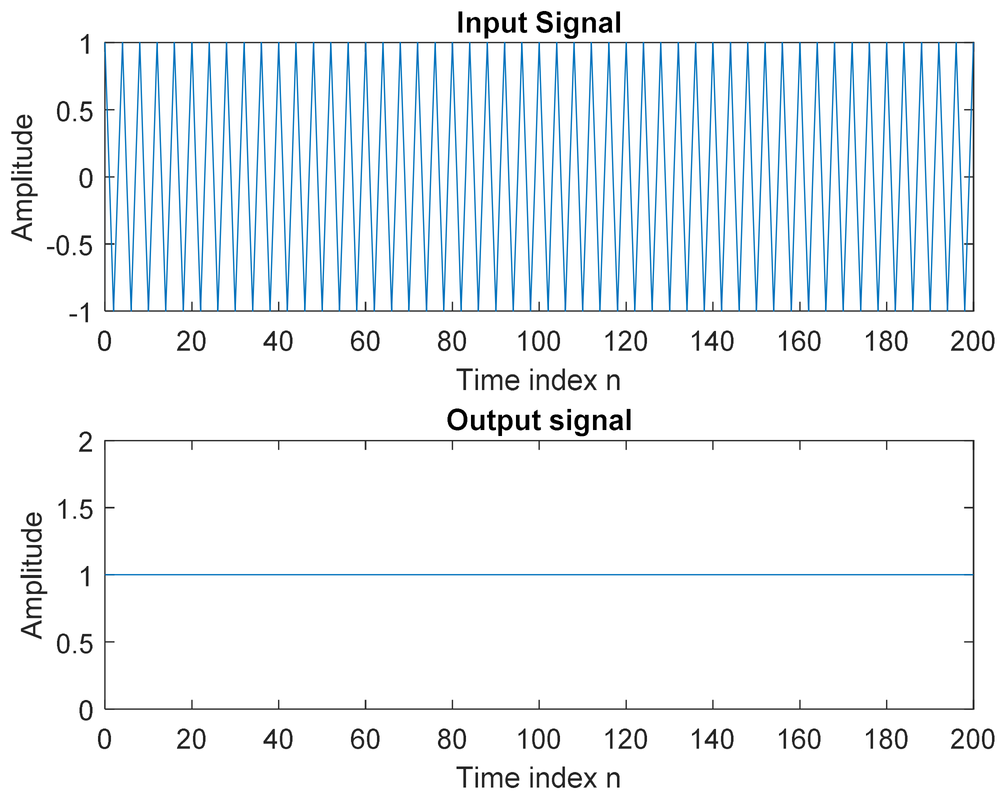
○ ***With*** $f = 0.01$***, we obtain:***



○ ***With*** $f = 0.15$***, we obtain:***



- ***With*** $f = 0.25$***, we obtain:***

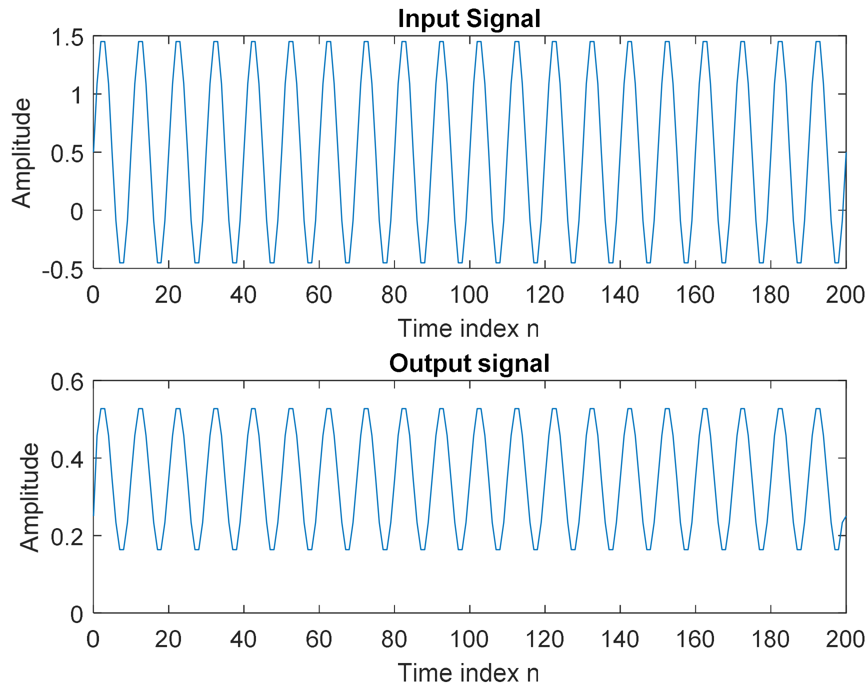


The output signals depend on the frequencies of the input signal according to the following rules: $y(n) = \cos^2(2\pi fn) - \cos[2\pi f(n+1)].\cos[2\pi f(n-1)]$

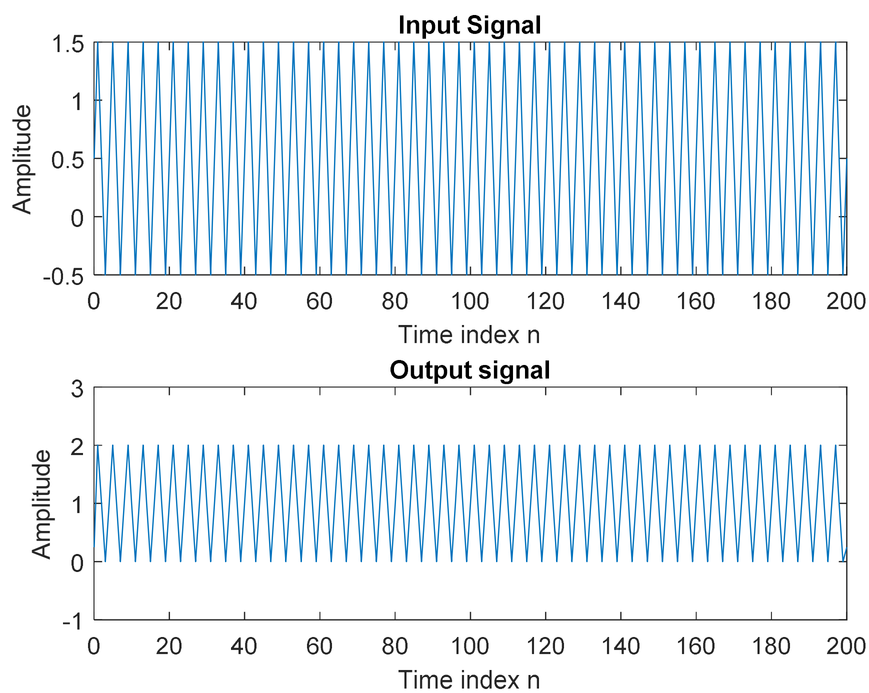
This observation can be explained mathematically as follows: ***By expand the above equation, we can get a conclusion about these previous results.***

Q2.6 The output signal generated by using sinusoidal signals of the form $x[n] = \sin(\omega_0 n) + K$ as the input signal is shown below for the following values of ω_0 and K :

- ***With*** $f = 0.1; \quad k = 0.5$***, we obtain:***



- ***With*** $f = 0.25; \quad k = 0.5$***, we obtain:***



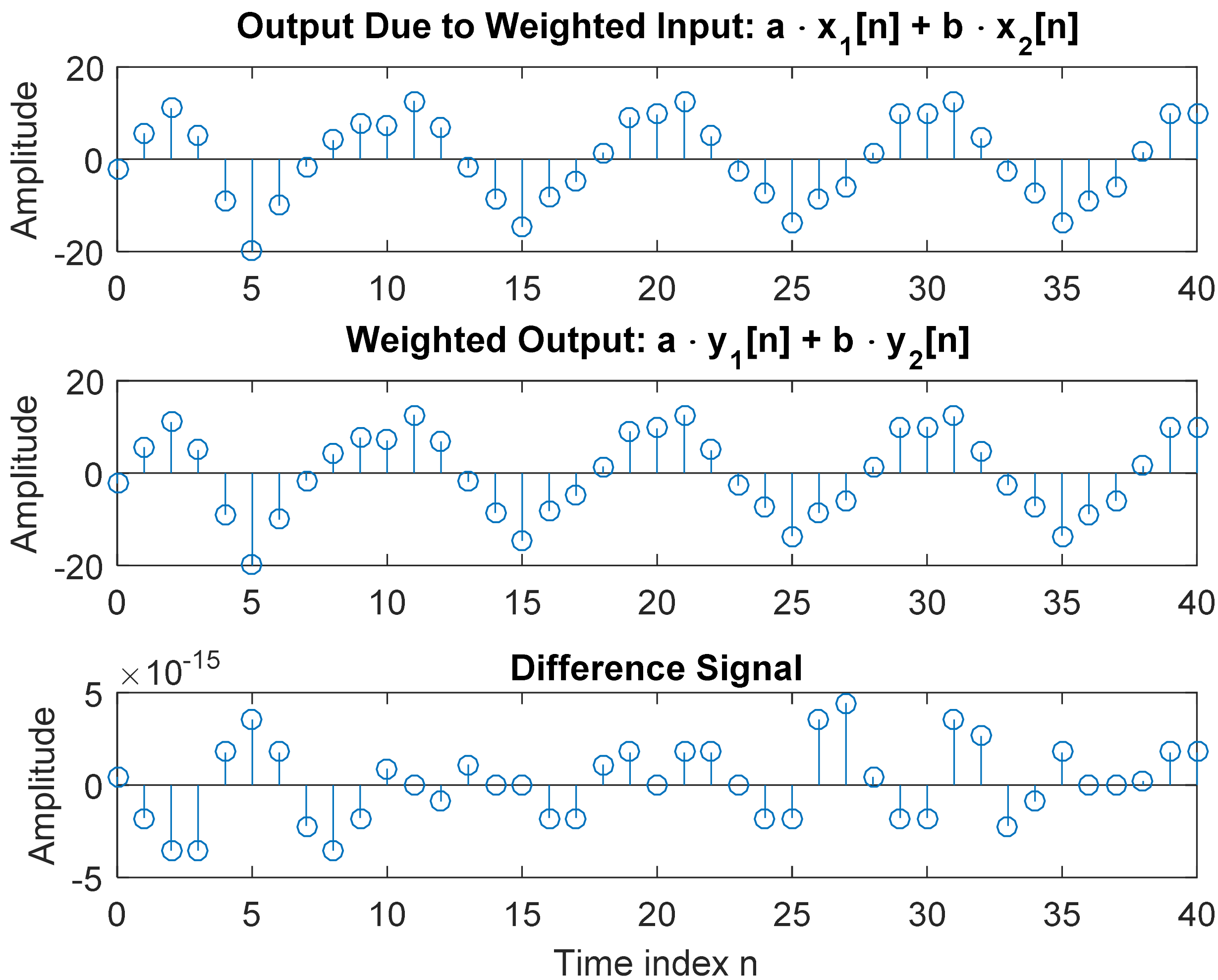
The dependence of the output signal yt[n] on the DC value K can be explained as ***the same way in Q2.5.***

Project 2.3 Linear and Nonlinear Systems

A copy of Program P2_3 is given below:

```
% Program P2_3
% Generate the input sequences
clc; clear all; close all;
n = 0:40;
a = 2; b = -3;
x1 = cos(2*pi*0.1*n);
x2 = cos(2*pi*0.4*n);
x = a*x1 + b*x2;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; % Set zero initial conditions
y1 = filter(num,den,x1,ic); % Compute the output y1[n]
y2 = filter(num,den,x2,ic); % Compute the output y2[n]
y = filter(num,den,x,ic); % Compute the output y[n]
yt = a*y1 + b*y2;
d = y - yt; % Compute the difference output d[n]
% Plot the outputs and the difference signal
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
title('Output Due to Weighted Input: a \cdot x_{1}[n] + b
\cdot x_{2}[n]');
subplot(3,1,2)
stem(n,yt);
ylabel('Amplitude');
title('Weighted Output: a \cdot y_{1}[n] + b \cdot
y_{2}[n]');
subplot(3,1,3)
stem(n,d);
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal');
```

Q2.7 The outputs y[n], obtained with weighted input, and yt[n], obtained by combining the two outputs y1[n] and y2[n] with the same weights, are shown below along with the difference between the two signals:



The two sequences are ***almost similar.***

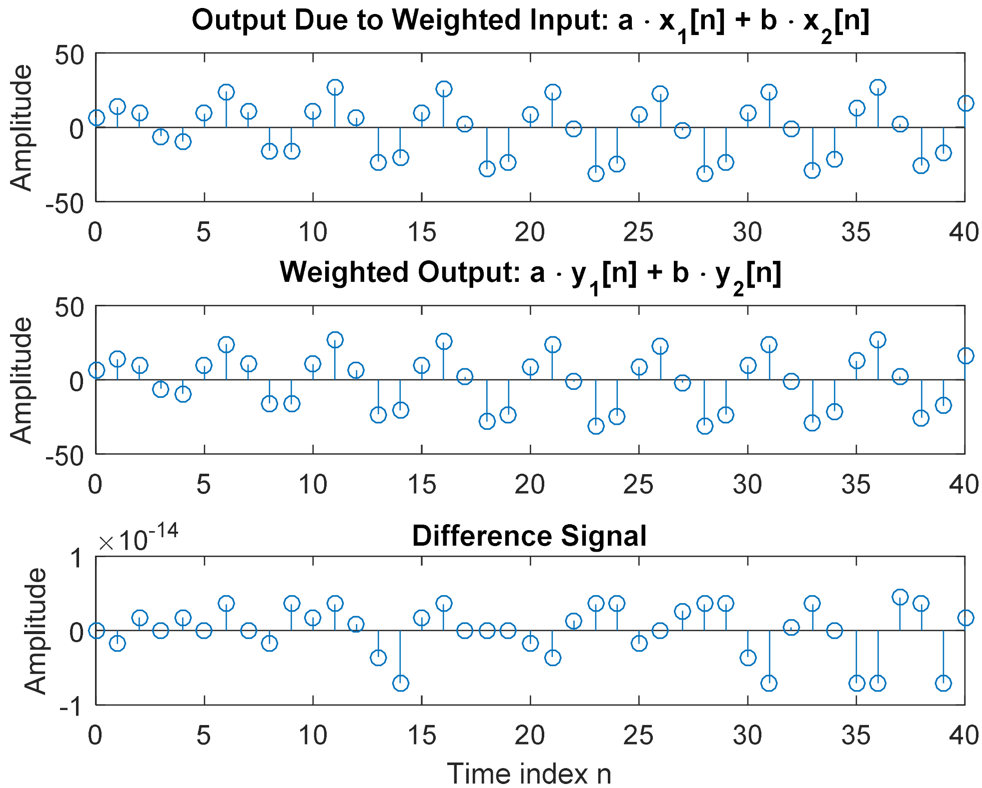
The system is ***linear.***

Q2.8 Program P2_3 was run for the following three different sets of values of the weighting constants, a and b, and the following three different sets of input frequencies:

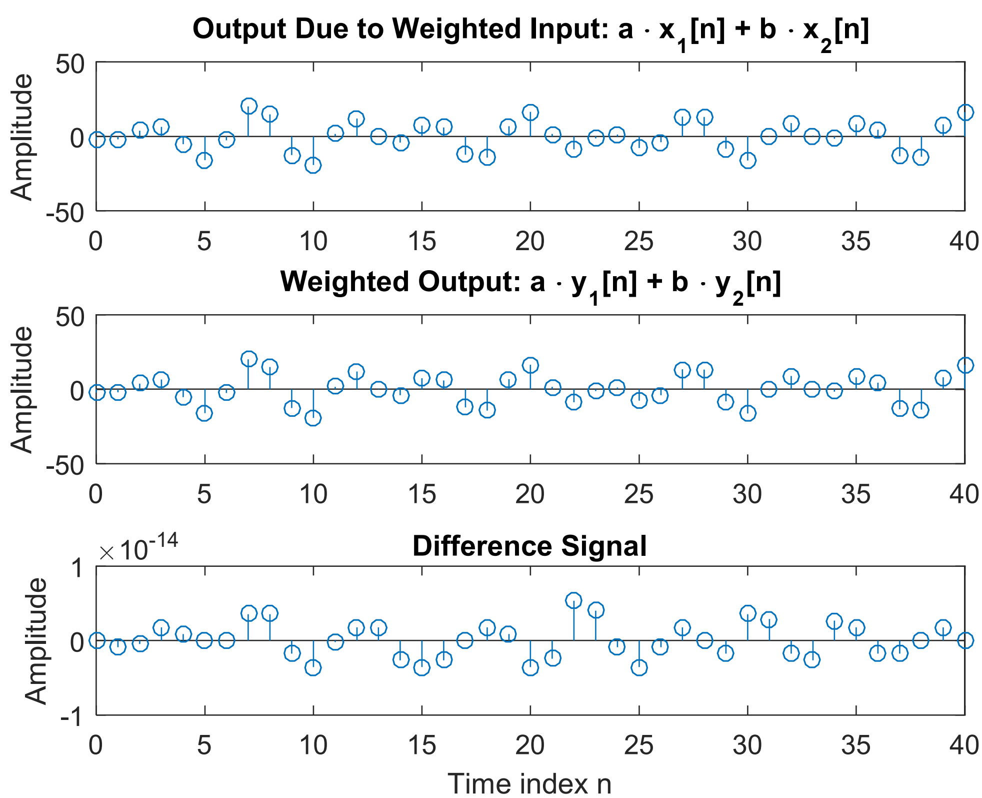
- $a = 1;\ b = 2;\ f_1 = 0.02;\ f_2 = 0.2$
- $a = 1;\ b = -2;\ f_1 = 0.15;\ f_2 = 0.25$
- $a = -2;\ b = -8;\ f_1 = 0.2;\ f_2 = 0.3$

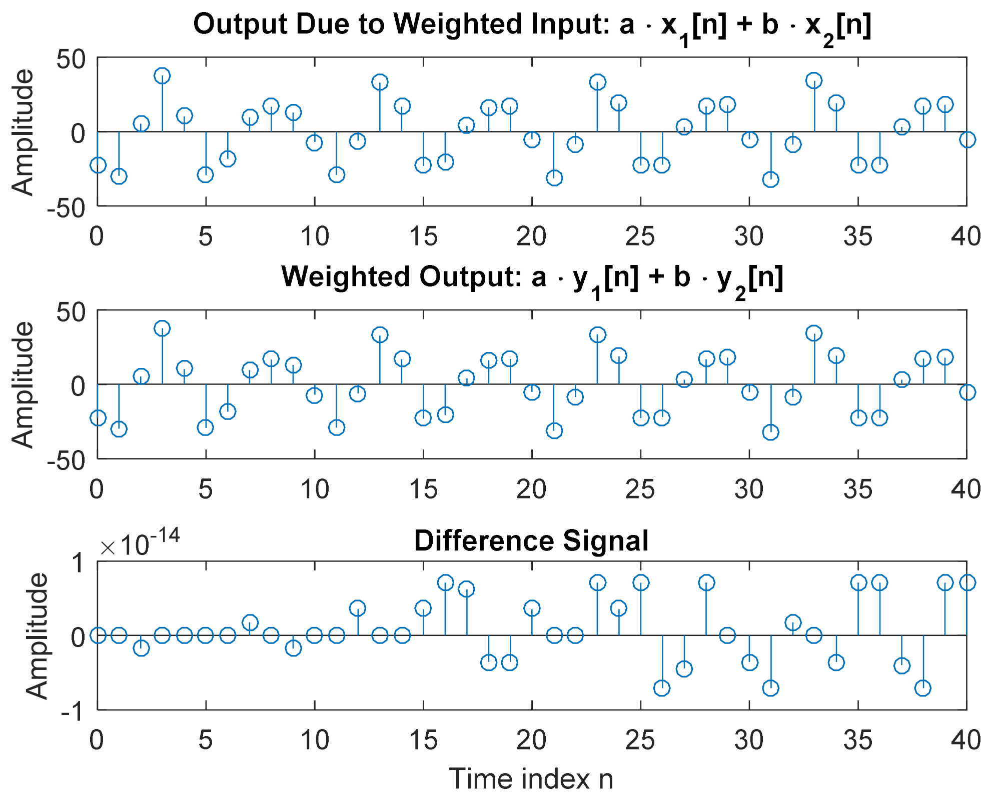
The plots generated for each of the above three cases are shown below:

- ***With*** $a = 1;\ b = 2;\ f_1 = 0.02;\ f_2 = 0.2$



- ***With*** $a = 1;\ b = -2;\ f_1 = 0.15;\ f_2 = 0.25$



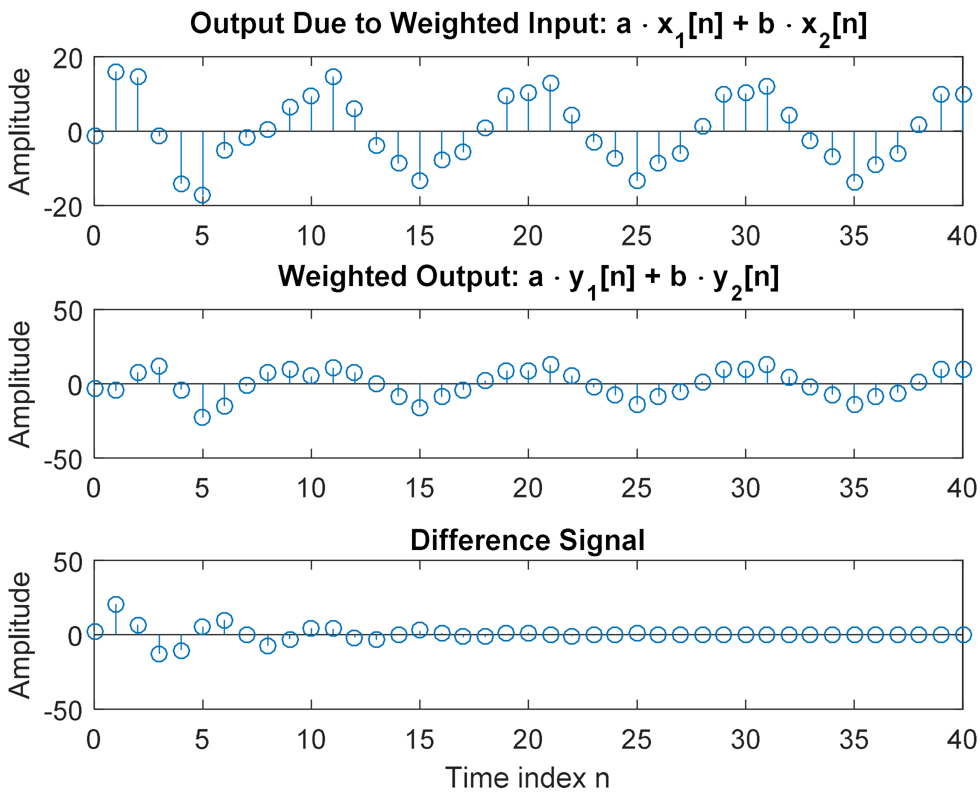


- ***With*** $a = -2;\ b = -8;\ f_1 = 0.2;\ f_2 = 0.3$

Based on these plots we can conclude that the system with different weights is ***Linear.***

Q2.9 Program 2_3 was run with the following non-zero initial conditions - `ic = [1 10];`

The plots generated are shown below:



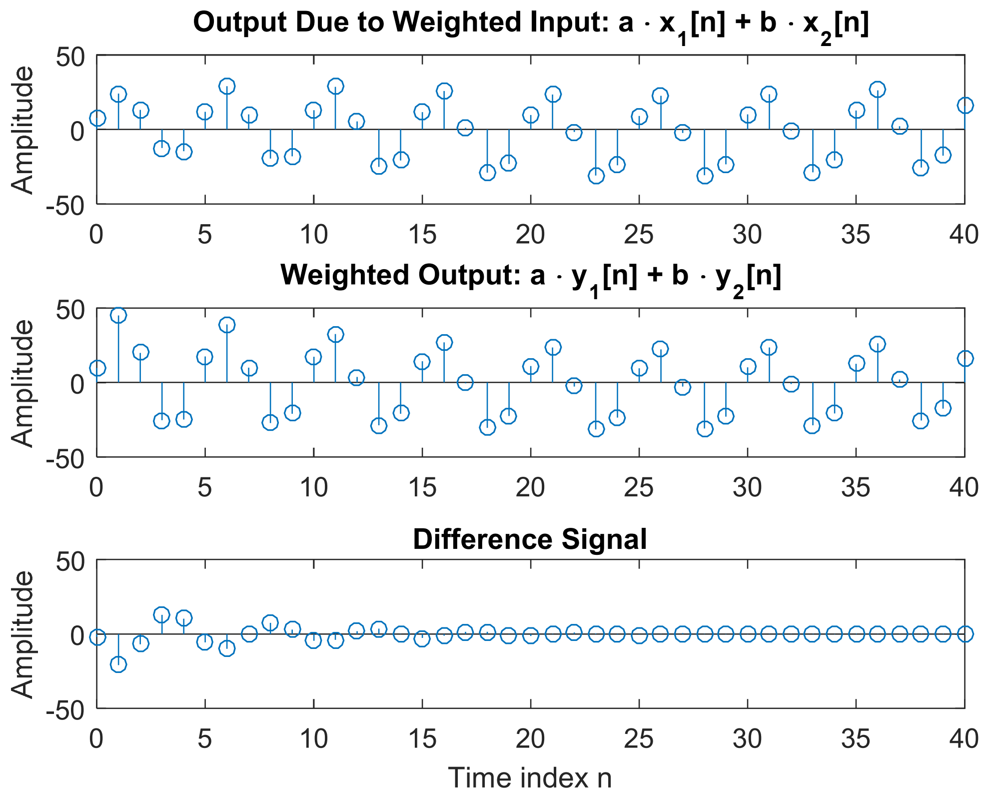
Based on these plots we can conclude that the system with nonzero initial conditions is ***Non Linear.***

Q2.10 Program P2_3 was run with nonzero initial conditions and for the following three different sets of values of the weighting constants, a and b, and the following three different sets of input frequencies:

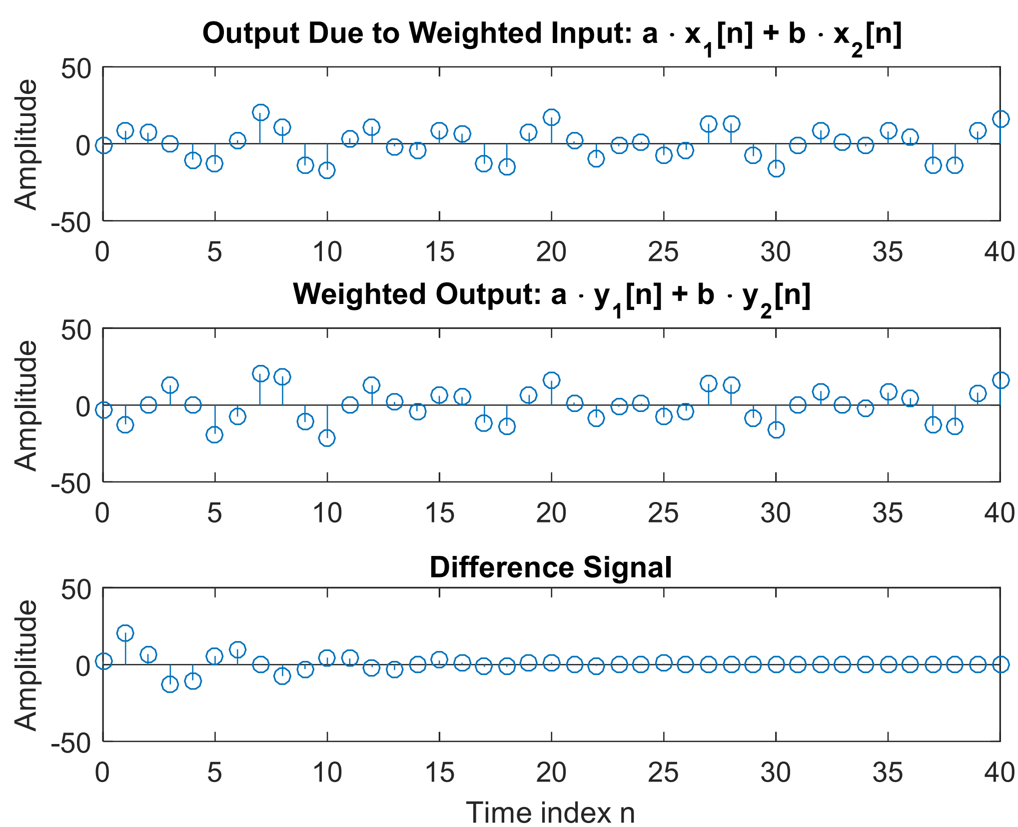
- $a = 1;\ b = 2;\ f_1 = 0.02;\ f_2 = 0.2$
- $a = 1;\ b = -2;\ f_1 = 0.15;\ f_2 = 0.25$
- $a = -2;\ b = -8;\ f_1 = 0.2;\ f_2 = 0.3$

The plots generated for each of the above three cases are shown below:

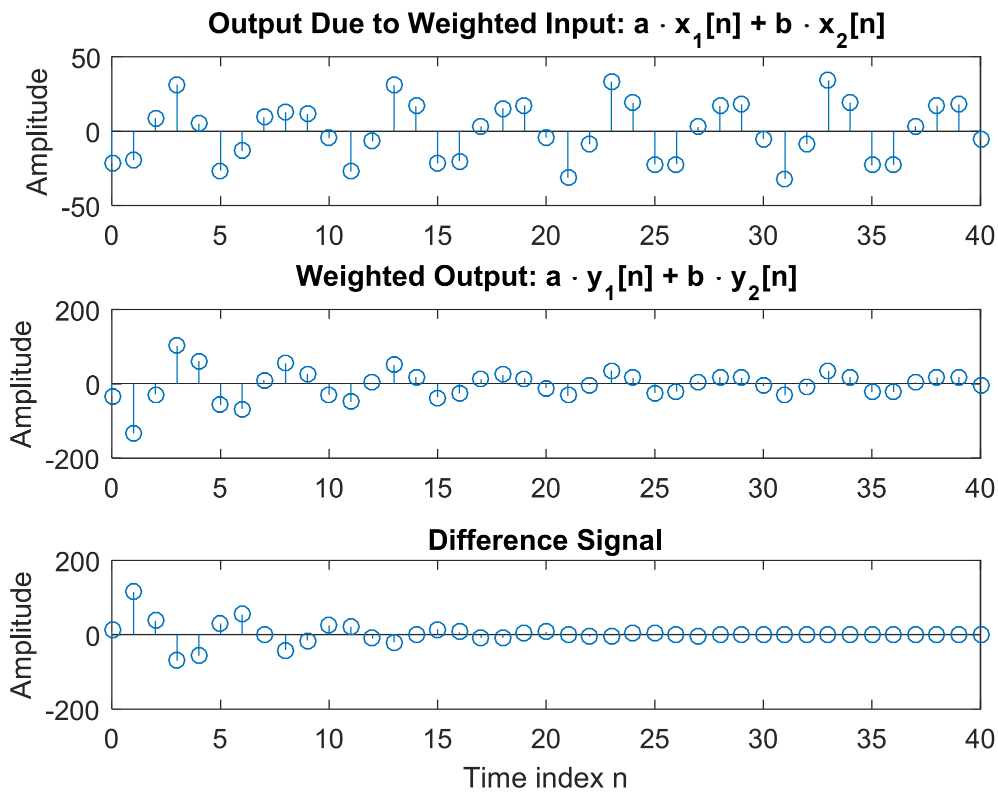
- ***With*** $a=1;\ b=2;\ f_1=0.02;\ f_2=0.2$



- ***With*** $a=1;\ b=-2;\ f_1=0.15;\ f_2=0.25$



- ***With*** $a = -2;\ b = -8;\ f_1 = 0.2;\ f_2 = 0.3$



Based on these plots we can conclude that the system with nonzero initial conditions and different weights is ***Non Linear.***

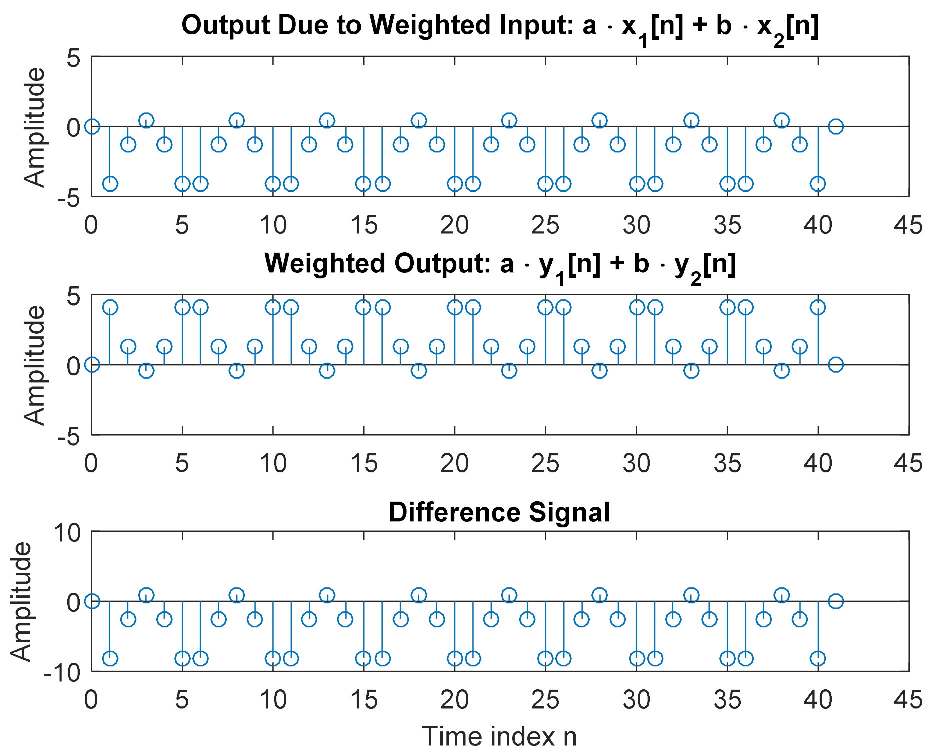
Q2.11 Program P2_3 was modified to simulate the system:

$$y[n] = x[n]x[n-1]$$

The output sequences y1[n], y2[n],and y[n] of the above system generated by running the modified program are shown below:

```
% Program P2_3
% Generate the input sequences
% clf;
clc;
clear all;
close all;
n = 0:40;
a = 2;b = -3;
x1 = cos(2*pi*0.1*n);
x11=[0 x1]; %x1(n-1)
x12=[x1 0]; %x1(n)
```

```
y1=x11.*x12;
x2 = cos(2*pi*0.4*n);
x21=[0 x2]; %x2(n-1)
x22=[x2 0]; %x2(n)
y2=x21.*x22;
x = a*x1 + b*x2;
xx1=[0 x];
xx2=[x 0];
y=xx1.*xx2;
yt = a*y1 + b*y2;
d = y - yt; % Compute the difference output d[n]
% Plot the outputs and the difference signal
subplot(3,1,1)
stem([n,41],y);
ylabel('Amplitude');
title('Output Due to Weighted Input: a \cdot x_{1}[n] + b \cdot
x_{2}[n]');
subplot(3,1,2)
stem([n,41],yt);
ylabel('Amplitude');
title('Weighted Output: a \cdot y_{1}[n] + b \cdot y_{2}[n]');
subplot(3,1,3)
stem([n,41],d);
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal');
```



Comparing y[n] with yt[n] we conclude that the two sequences are ***different.***

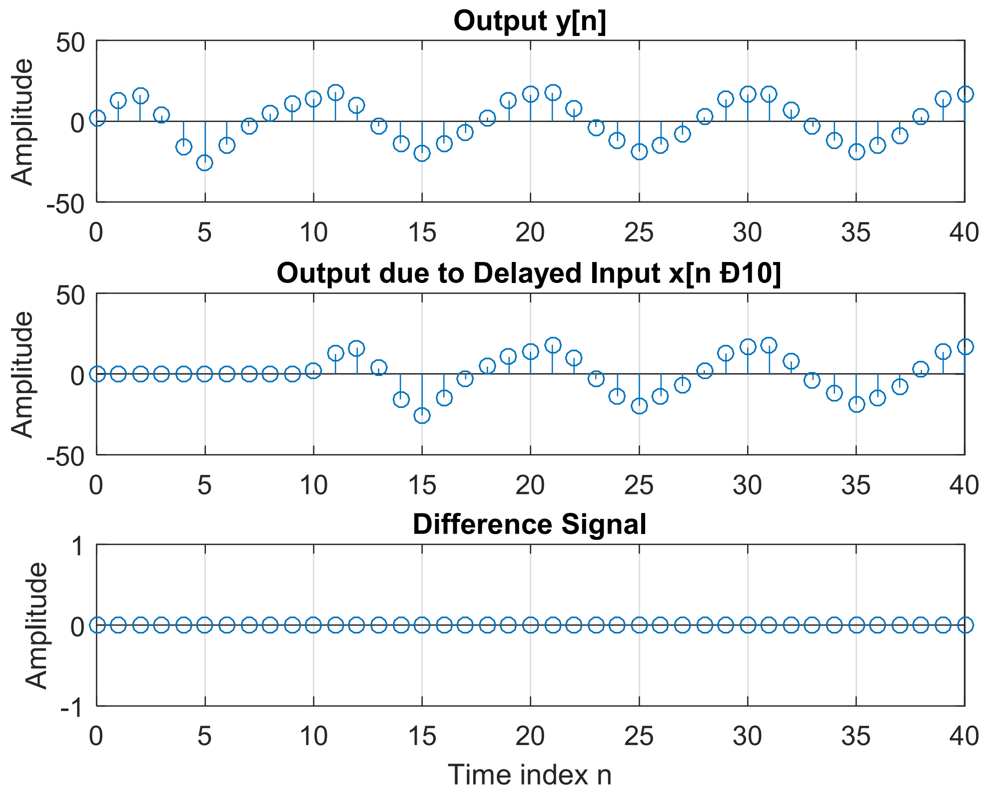
This system is ***Non Linear.***

Project 2.4 Time-invariant and Time-varying Systems

A copy of Program P2_4 is given below:

```
% Program P2_4
% Generate the input sequences
clc; clear all; close all;
n = 0:40; D = 10;a = 3.0;b = -2;
x = a*cos(2*pi*0.1*n) + b*cos(2*pi*0.4*n);
xd = [zeros(1,D) x];
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; % Set initial conditions
% Compute the output y[n]
y = filter(num,den,x,ic);
% Compute the output yd[n]
yd = filter(num,den,xd,ic);
% Compute the difference output d[n]
d = y - yd(1+D:41+D);
% Plot the outputs
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
title('Output y[n]'); grid;
subplot(3,1,2)
stem(n,yd(1:41));
ylabel('Amplitude');
title(['Output due to Delayed Input x[n Ð',
num2str(D),']']); grid;
subplot(3,1,3)
stem(n,d);
xlabel('Time index n'); ylabel('Amplitude');
title('Difference Signal'); grid;
```

Q2.12 The output sequences y[n] and yd[n-10] generated by running Program P2_4 are shown below



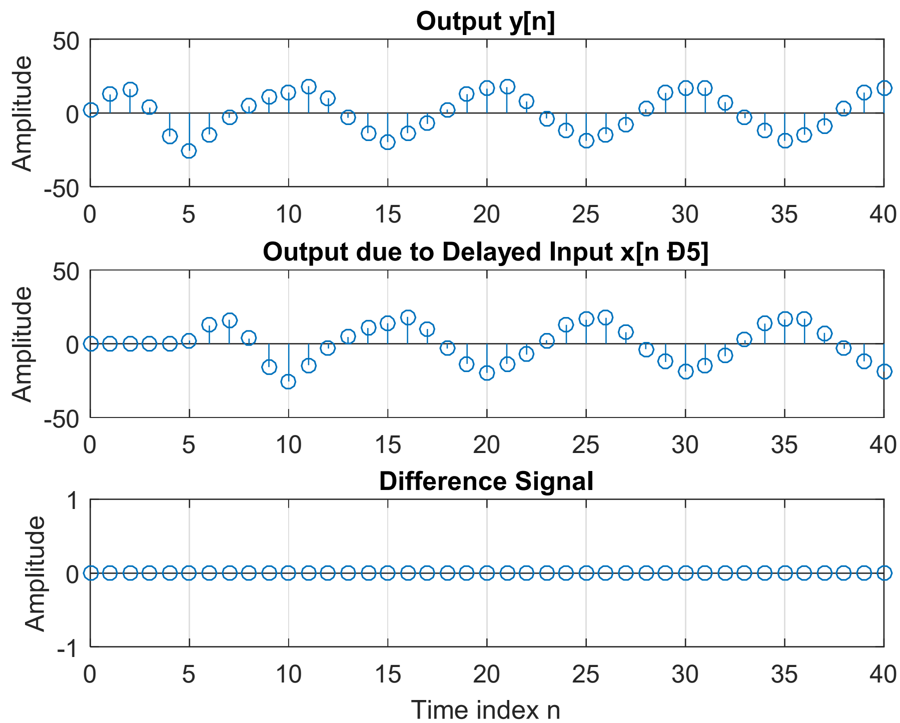
These two sequences are related as follows - $y(n) = y_d(n+10)$

The system is ***Time Invariant.***

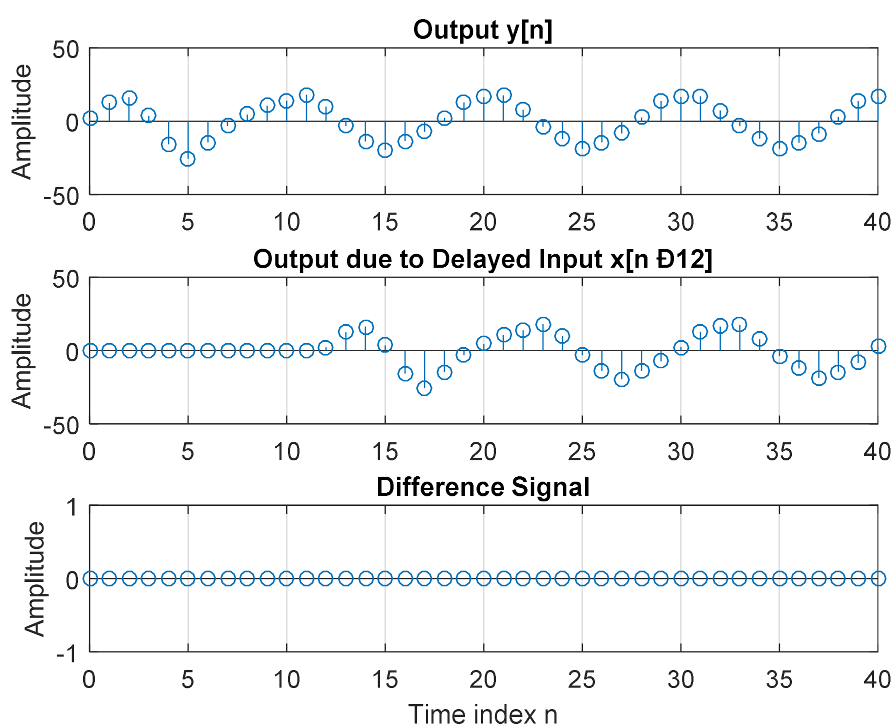
Q2.13 The output sequences y[n] and yd[n-D] generated by running Program P2_4 for the following values of the delay variable D - $5, 12, 15$

are shown below:

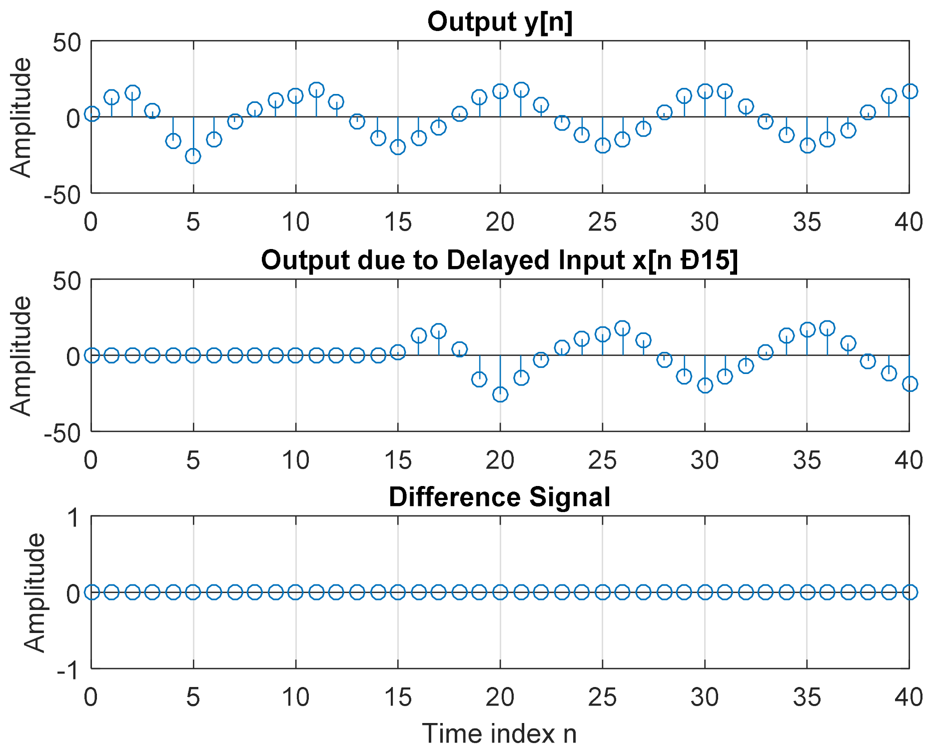
○ ***With*** $D = 5$



○ ***With*** $D = 12$



- ***With*** $D = 15$



In each case, these two sequences are related as follows - $y(n) = y_d(n + D)$

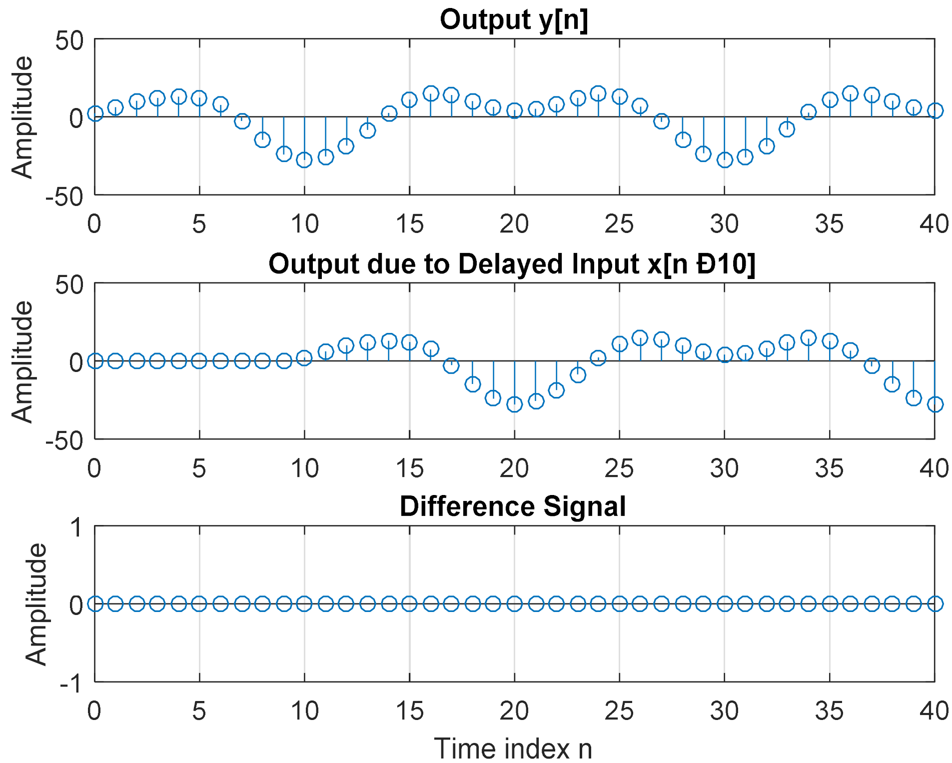
The system is ***Time Invariant.***

Q2.14 The output sequences y[n] and yd[n-10] generated by running Program P2_4 for the following values of the input frequencies:

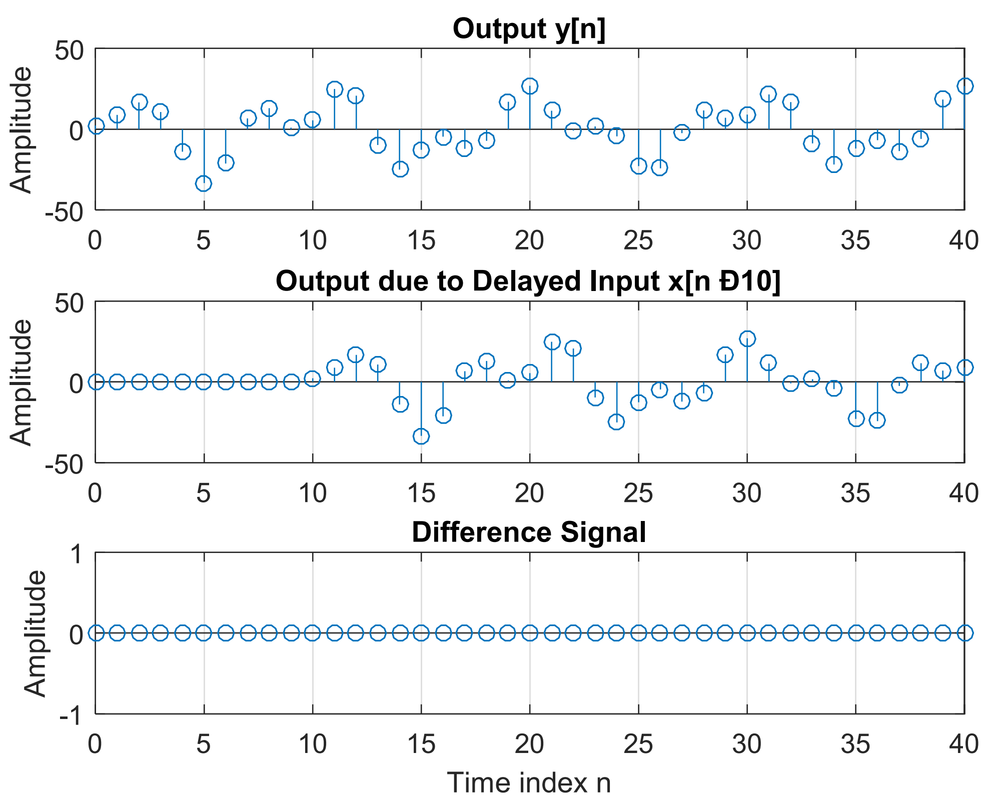
- $f_1 = 0.05;\ f_2 = 0.1$
- $f_1 = 0.1;\ f_2 = 0.25$
- $f_1 = 0.3;\ f_2 = 0.5$

are shown below :

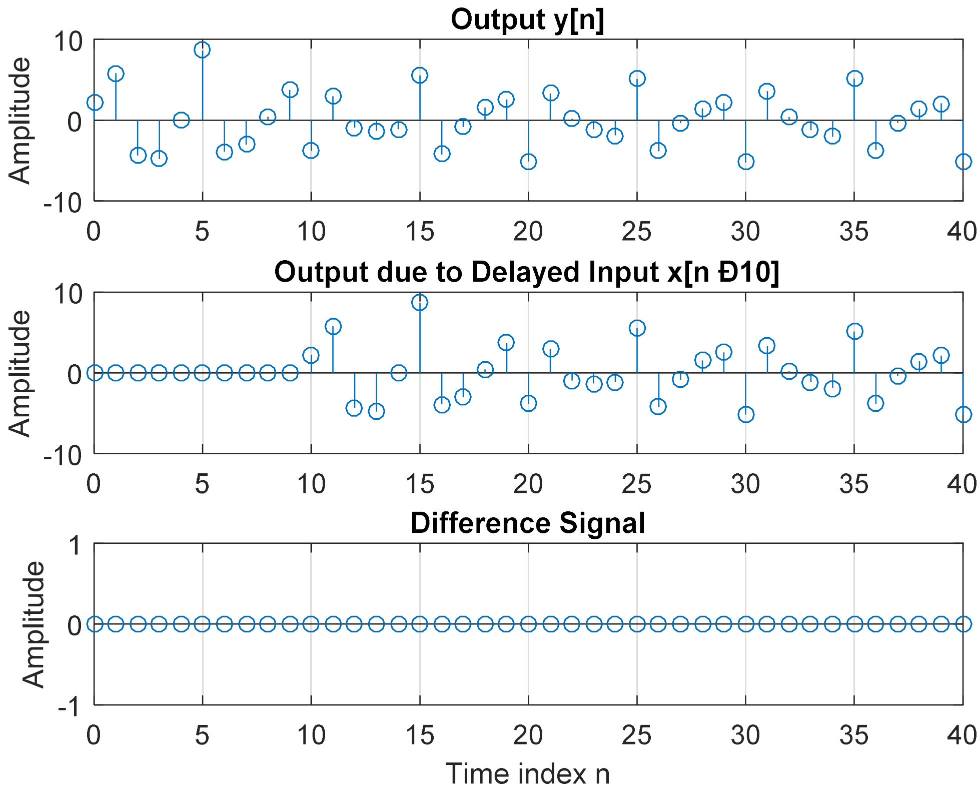
- ***With*** $f_1 = 0.05;\ f_2 = 0.1$



- ***With*** $f_1 = 0.1;\ f_2 = 0.25$



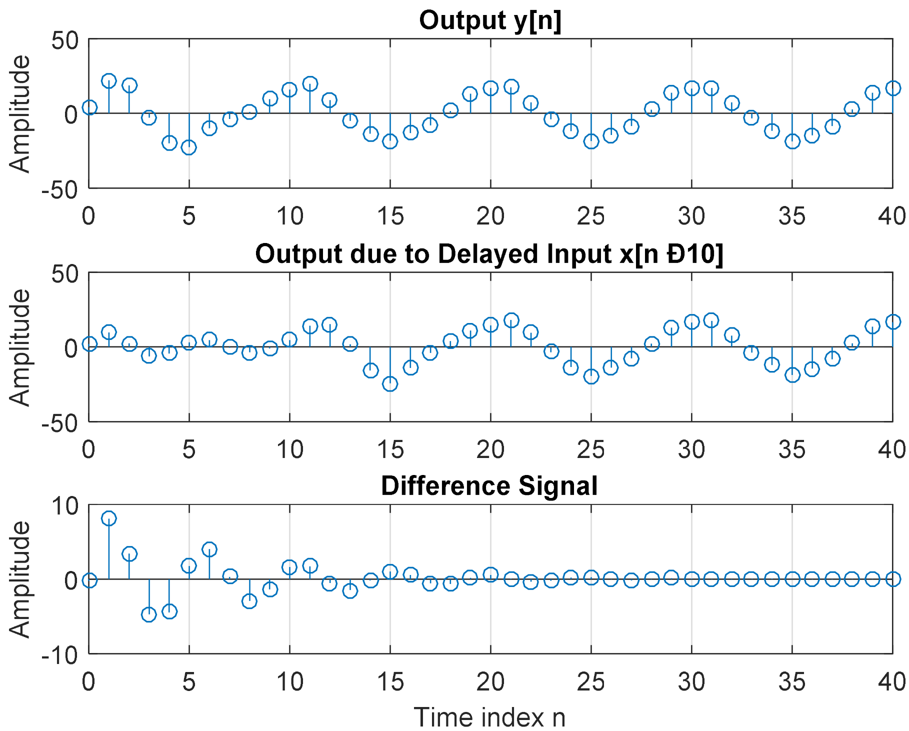
- ***With*** $f_1 = 0.3;\ f_2 = 0.5$



In each case, these two sequences are related as follows $y(n) = y_d(n+10)$

The system is ***Time Invariant.***

Q2.15 The output sequences y[n] and yd[n-10] generated by running Program P2_4 for non-zero initial conditions are shown below - `ic = [2 9];`



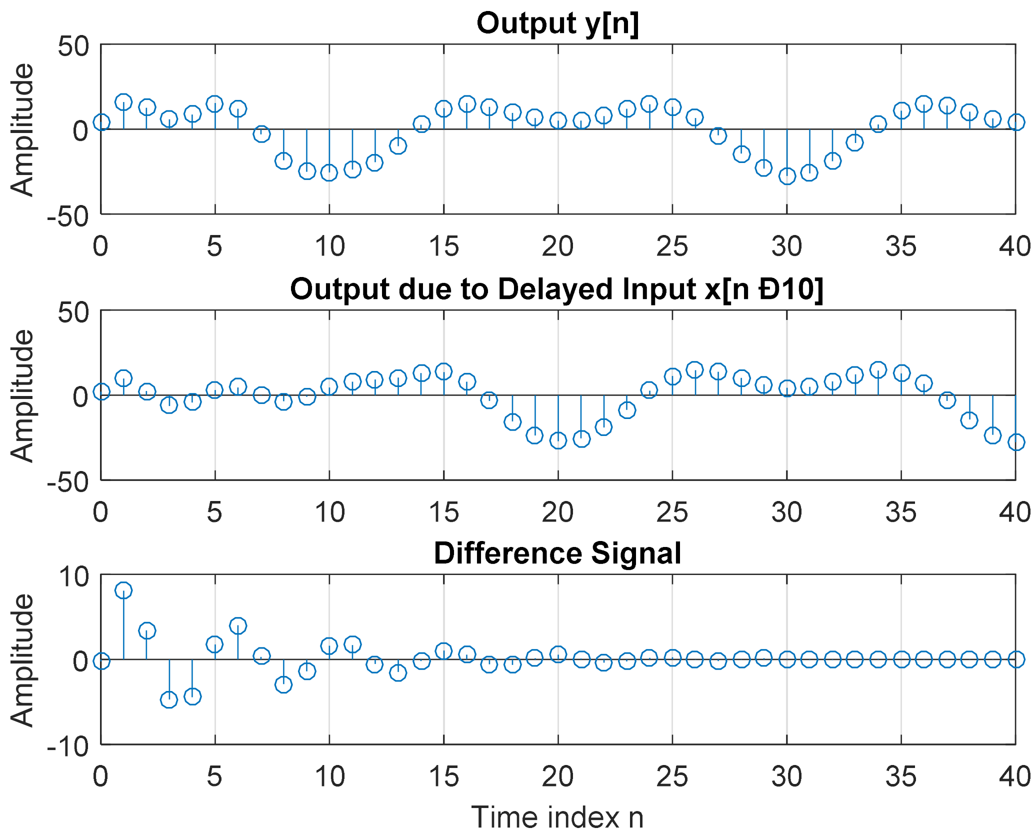
These two sequences are related as follows: $y(n)$ ***and*** $yd(n)$ ***are not the shifted versions each other.***

The system is ***Time Varying.***

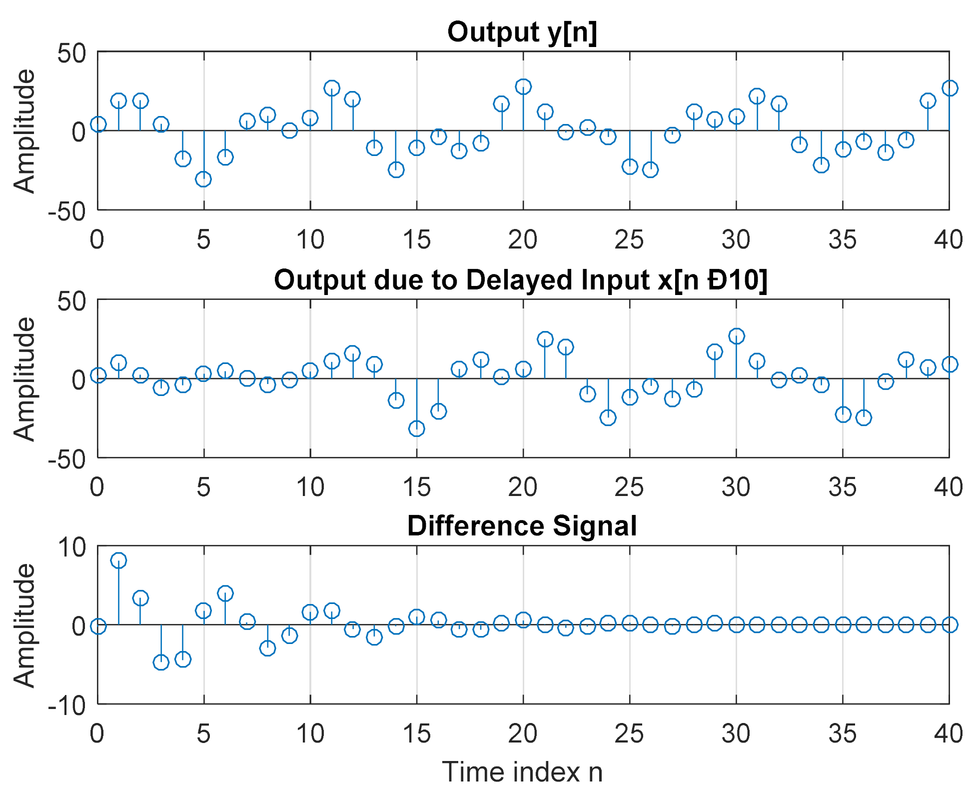
Q2.16 The output sequences y[n] and yd[n-10] generated by running Program P2_4 for non-zero initial conditions and following values of the input frequencies:

- $f_1 = 0.05;\ f_2 = 0.1$
- $f_1 = 0.1;\ f_2 = 0.25$
- $f_1 = 0.3;\ f_2 = 0.5$

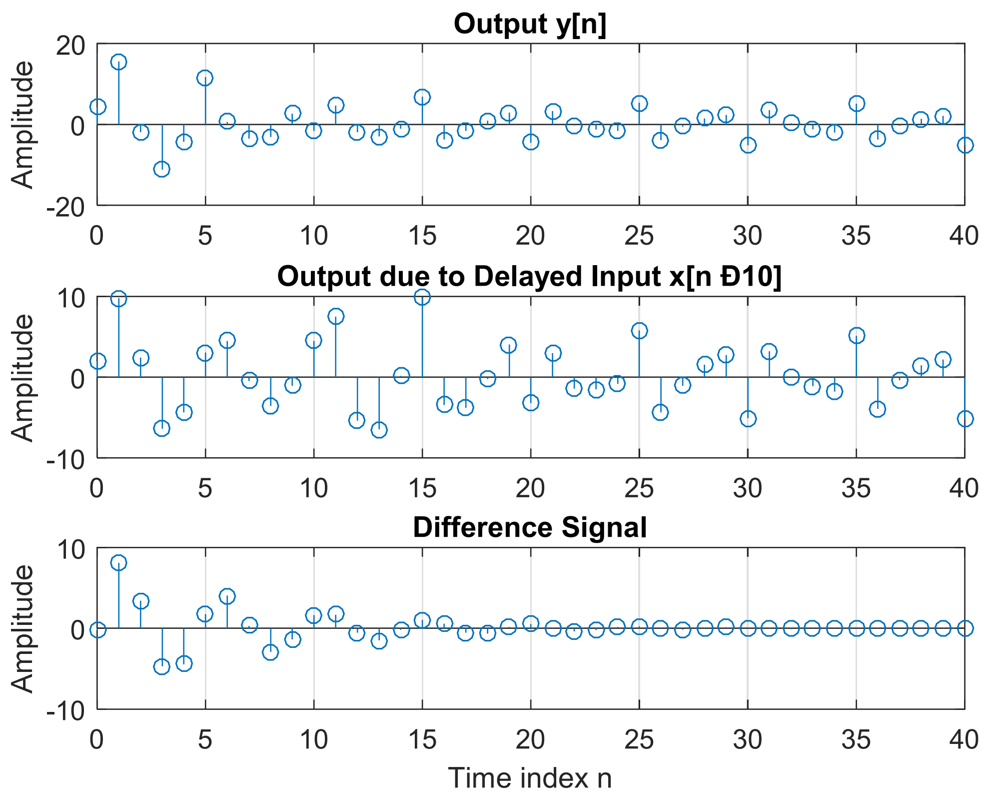
- ***With*** $f_1 = 0.05; f_2 = 0.1$



- ***With*** $f_1 = 0.1; f_2 = 0.25$



- ***With*** $f_1 = 0.3;\ f_2 = 0.5$



In each case, these two sequences are related as follows - $y(n)$ ***and*** $yd(n)$ ***are not the shifted versions each other.***

The system is ***Time Varying.***

Q2.17 The modified Program 2_4 simulating the system

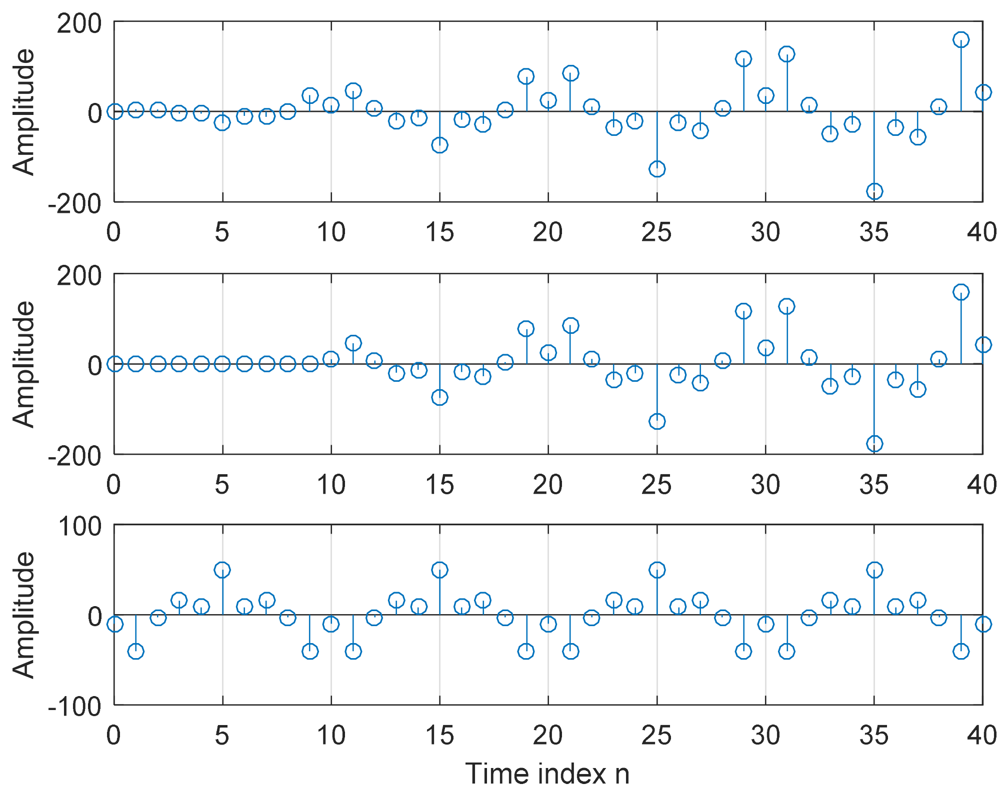
$$y[n] = nx[n] + x[n-1]$$

is given below:

```
% Generate the input sequences
clc; clear all; close all;
n = 0:40; D = 10;a = 3.0;b = -2;
x = a*cos(2*pi*0.1*n) + b*cos(2*pi*0.4*n);
xd = [zeros(1,D) x];
nd = 0:length(xd)-1;
% Compute the output y[n]
y = (n .* x) + [0 x(1:40)];
% Compute the output yd[n]
yd = (nd .* xd) + [0 xd(1:length(xd)-1)];
```

```
% Compute the difference output d[n]
d = y - yd(1+D:41+D);
% Plot the outputs
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
grid;
subplot(3,1,2)
stem(n,yd(1:41));
ylabel('Amplitude');
grid;
subplot(3,1,3)
stem(n,d);
xlabel('Time index n'); ylabel('Amplitude');
grid;
```

The output sequences y[n] and yd[n-10] generated by running modified Program P2_4 are shown below :



These two sequences are related as follows: $y(n)$ ***and*** $yd(n)$ ***are not the shifted versions each other.***

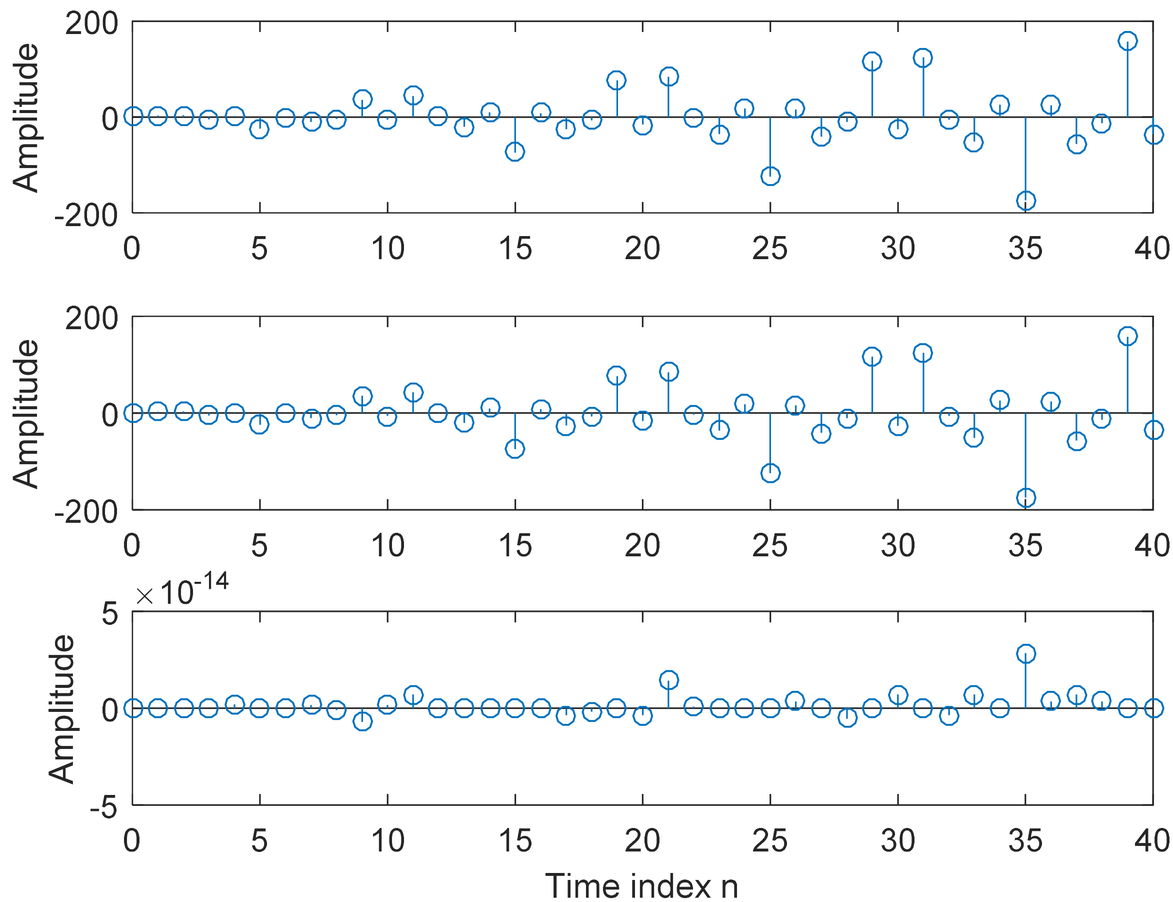
The system is ***Time Varying.***

Q2.18 (optional) The modified Program P2_3 to test the linearity of the system of Q2.18 is shown below:

```
% Program Q2_18
% Modify P2_3 for Q2.18.
% Generate the input sequences
clc; clear all; close all;
n = 0:40;
a = 2;b = -3;
x1 = cos(2*pi*0.1*n);
x2 = cos(2*pi*0.4*n);
x = a*x1 + b*x2;
y1 = (n .* x1) + [0 x1(1:40)]; % Compute the output y1[n]
y2 = (n .* x2) + [0 x2(1:40)]; % Compute the output y2[n]
y = (n .* x) + [0 x(1:40)]; % Compute the output y[n]
yt = a*y1 + b*y2;
d = y - yt; % Compute the difference output d[n]
% Plot the outputs and the difference signal
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
subplot(3,1,2)
stem(n,yt);
ylabel('Amplitude');
subplot(3,1,3)
stem(n,d);
xlabel('Time index n');ylabel('Amplitude');
```

The outputs y[n]and yt[n] obtained by running the modified program P2_3 are shown below:

The two sequences are ***almost similar.***



The system is ***Linear.***

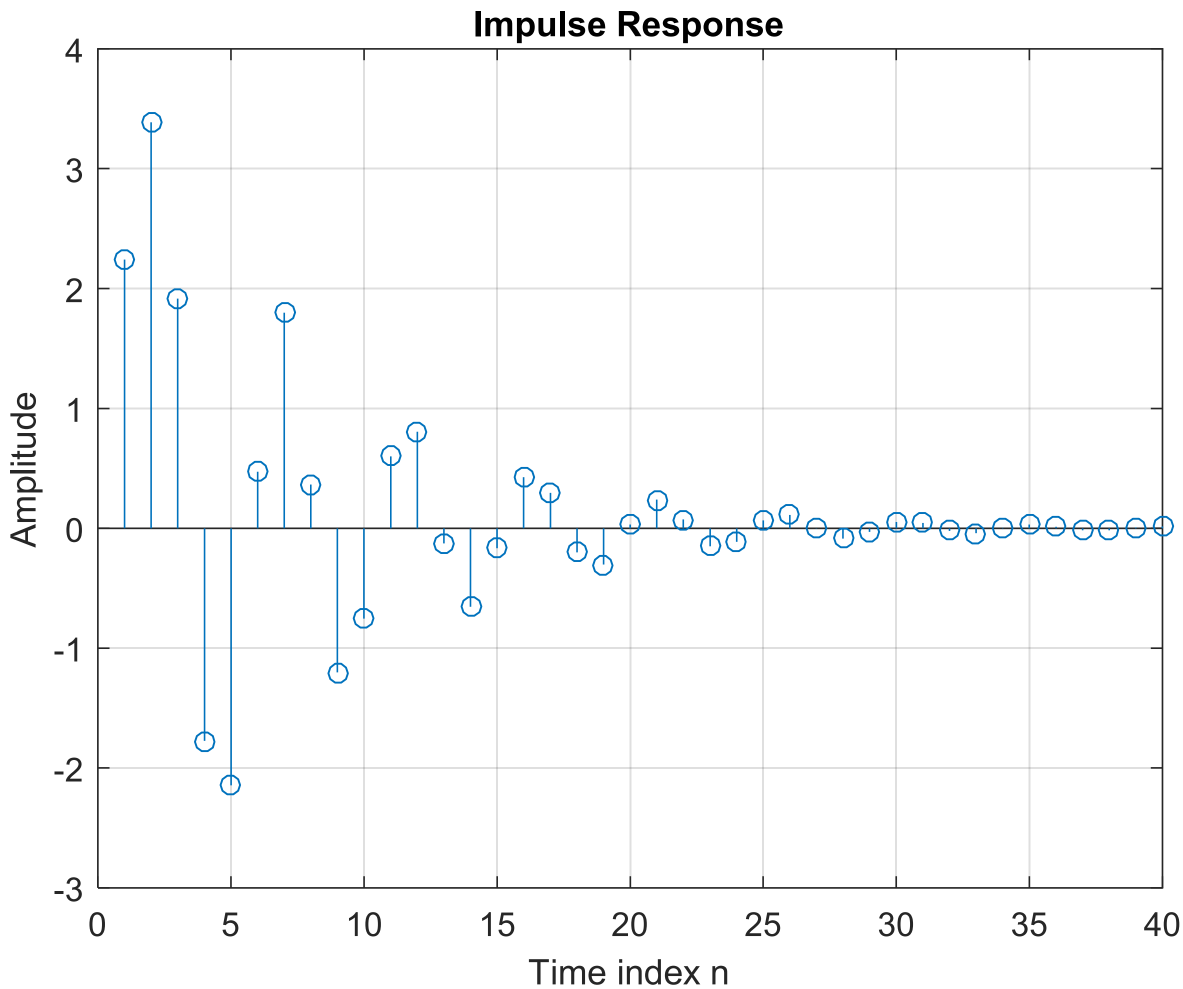
2.2 LINEAR TIME-INVARIANT DISCRETE-TIME SYSTEMS

Project 2.5 Computation of Impulse Responses of LTI Systems

A copy of Program P2_5 is shown below:

```
% Program P2_5
% Compute the impulse response y
clc; clear all; close all;
N = 40;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
y = impz(num,den,N);
% Plot the impulse response
stem(y);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

Q2.19 The first 41 samples of the impulse response of the discrete-time system of Project 2.3 generated by running Program P2_5 is given below:



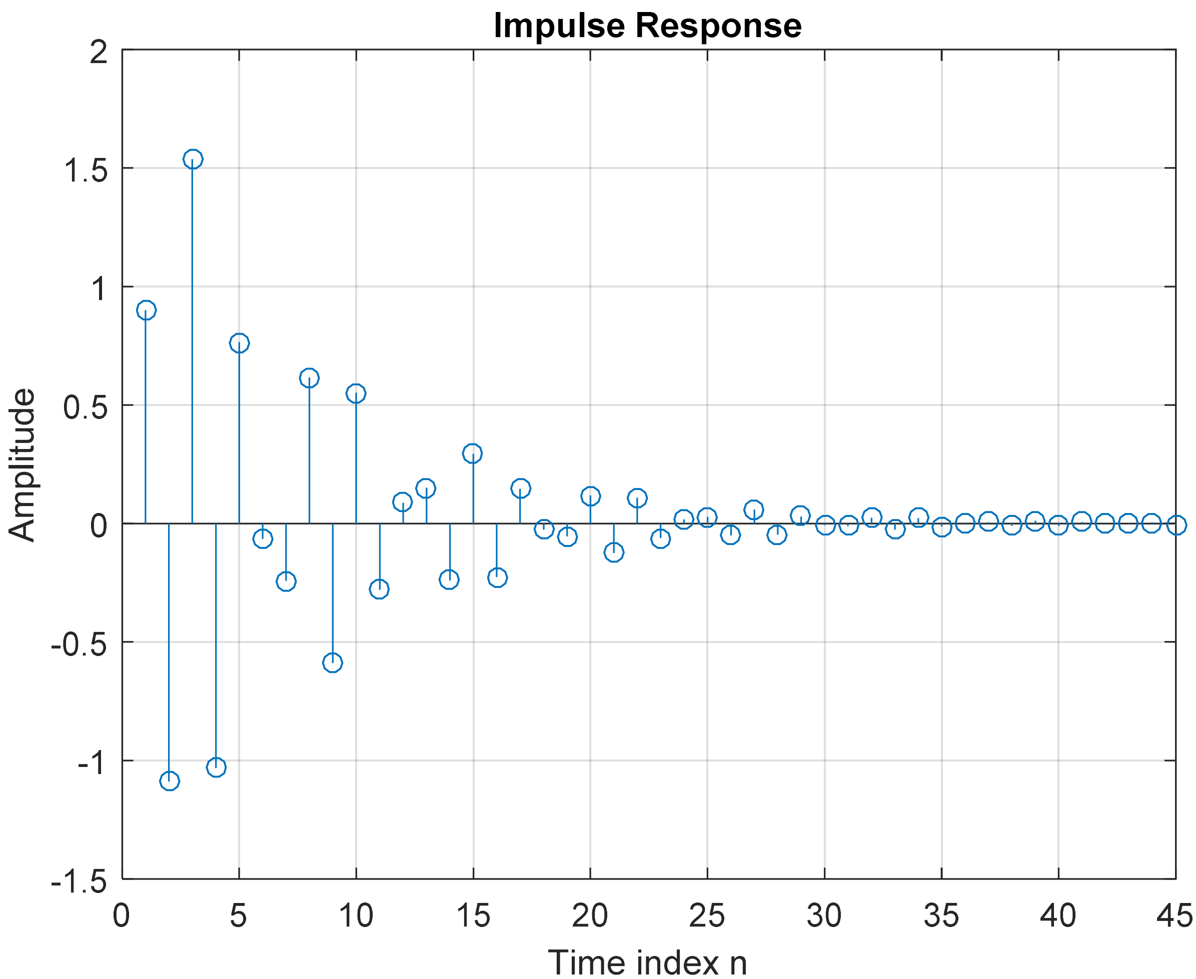
Q2.20 The required modifications to Program P2_5 to generate the impulse response of the following causal LTI system:

$$y[n]+0.71y[n-1]-0.46y[n-2]-0.62y[n-3]=0.9x[n]-0.45x[n-1]+0.35x[n-2]+0.002x[n-3]$$

are given below:

```
% Program Q2_20
% Compute the impulse response y
clc; clear all; close all;
N = 45;
num = [0.9 -0.45 0.35 0.002];
den = [1.0 0.71 -0.46 -0.62];
y = impz(num,den,N);
% Plot the impulse response
stem(y);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

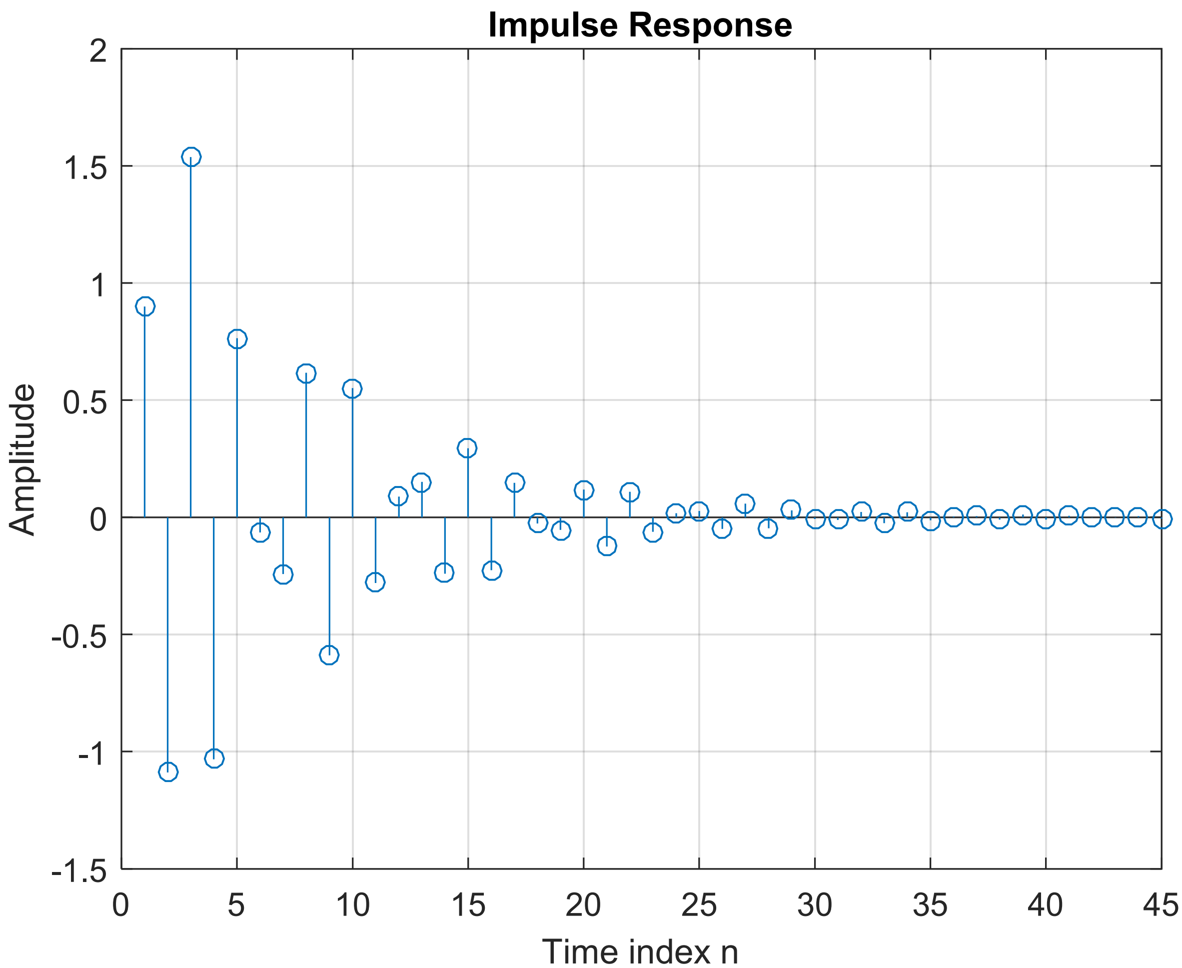
The first 45 samples of the impulse response of this discrete-time system generated by running the modified is given below:



Q2.21 The MATLAB program to generate the impulse response of a causal LTI system of Q2.20 using the filter command is indicated below:

```
% Program Q2_21
% Compute the impulse response y
clc; clear all; close all;
N = 40;
num = [0.9 -0.45 0.35 0.002];
den = [1.0 0.71 -0.46 -0.62];
% input: unit pulse
x = [1 zeros(1,N-1)];
y = filter(num,den,x);
stem(y);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

The first 40 samples of the impulse response generated by this program are shown below:

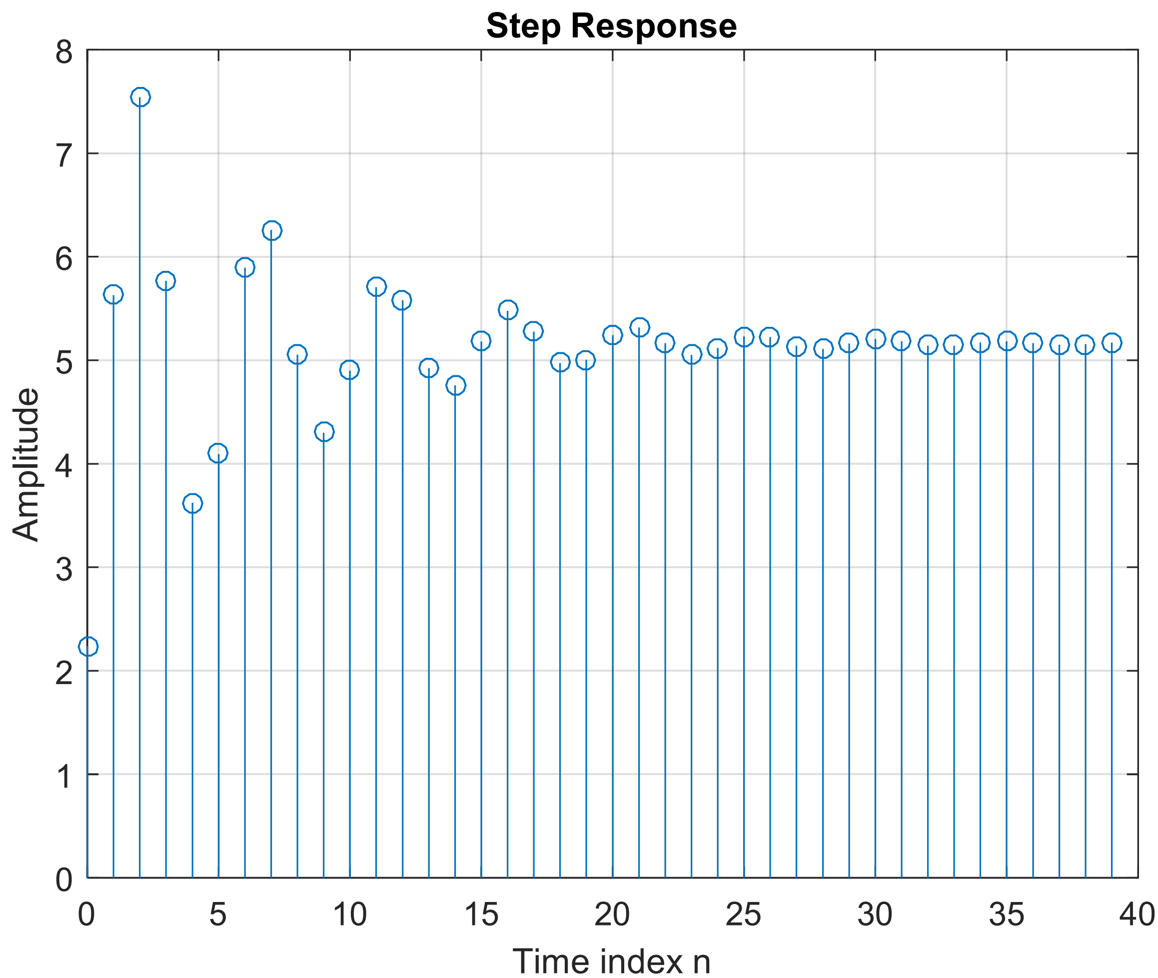


Comparing the above response with that obtained in Question Q2.20 we conclude that ***they are similar.***

Q2.22 The MATLAB program to generate and plot the step response of a causal LTI system is indicated below:

```
% Program Q2_21
% Compute the impulse response
clc; clear all; close all;
N = 40;
num = [0.9 -0.45 0.35 0.002];
den = [1.0 0.71 -0.46 -0.62];
% input: unit pulse
x = [1 zeros(1,N-1)];
y = filter(num,den,x);
stem(y);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

The first 40 samples of the step response of the LTI system of Project 2.3 are shown below:



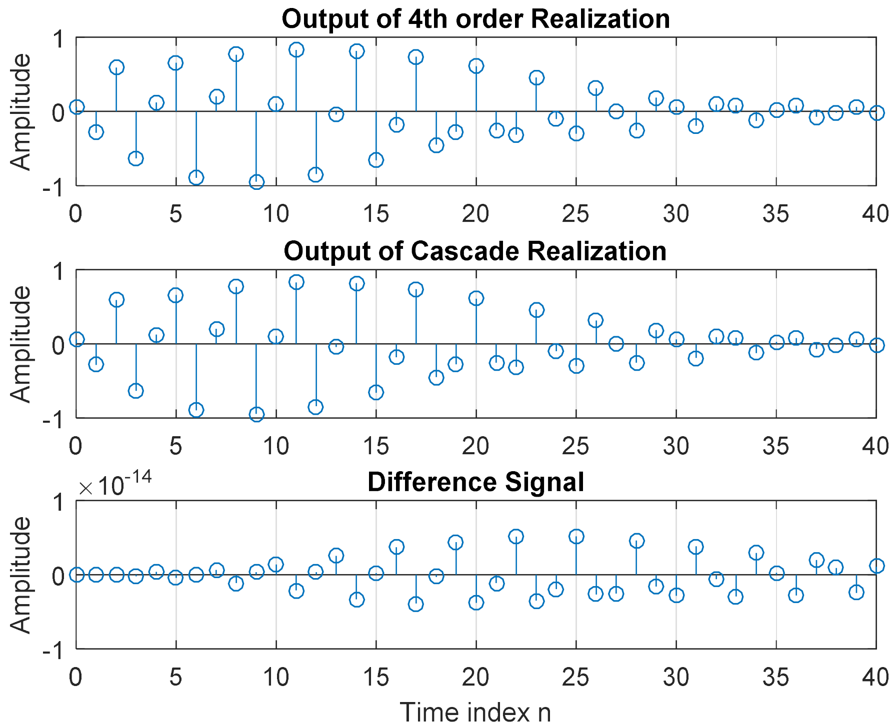
Project 2.6 Cascade of LTI Systems

A copy of Program P2_6 is given below:

```
% Program P2_6
% Cascade Realization
clc; clear all; close all;
x = [1 zeros(1,40)]; % Generate the input
n = 0:40;
% Coefficients of 4th order system
den = [1 1.6 2.28 1.325 0.68];
num = [0.06 -0.19 0.27 -0.26 0.12];
% Compute the output of 4th order system
y = filter(num,den,x);
% Coefficients of the two 2nd order systems
num1 = [0.3 -0.2 0.4];den1 = [1 0.9 0.8];
num2 = [0.2 -0.5 0.3];den2 = [1 0.7 0.85];
```

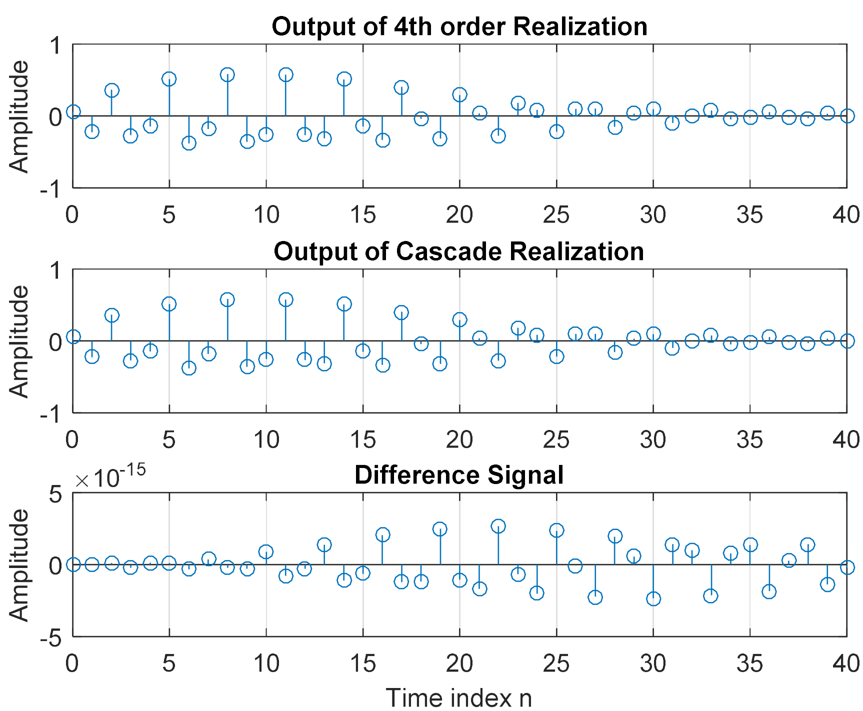
```
% Output y1[n] of the first stage in the cascade
y1 = filter(num1,den1,x);
% Output y2[n] of the second stage in the cascade
y2 = filter(num2,den2,y1);
% Difference between y[n] and y2[n]
d = y - y2;
% Plot output and difference signals
subplot(3,1,1);
stem(n,y);
ylabel('Amplitude');
title('Output of 4th order Realization'); grid;
subplot(3,1,2);
stem(n,y2)
ylabel('Amplitude');
title('Output of Cascade Realization'); grid;
subplot(3,1,3);
stem(n,d)
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal'); grid;
```

Q2.23 The output sequences y[n], y2[n], and the difference signal d[n] generated by running Program P2_6 are indicated below:



The relation between y[n] and y2[n] is ***almost similar.***

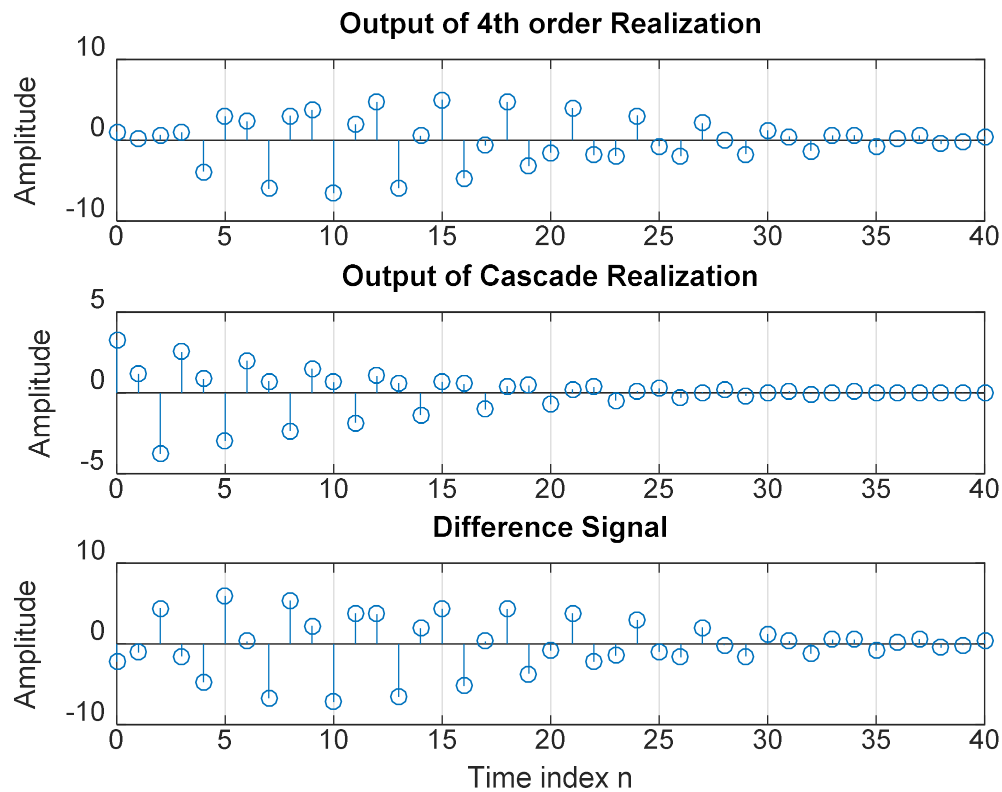
Q2.24 The sequences generated by running Program P2_6 with the input changed to a sinusoidal sequence are as follows:



The relation between y[n] and y2[n] in this case is ***almost similar.***

Q2.25 The sequences generated by running Program P2_6 with non-zero initial condition vectors are now as given below:

```
% Cascade Realization
clc; clear all; close all;
x = [1 zeros(1,40)]; % Generate the input
n = 0:40;
ic=[1 2 3 4];
ic1=[1 2];
ic2=[3 4];
% Coefficients of 4th order system
den = [1 1.6 2.28 1.325 0.68];
num = [0.06 -0.19 0.27 -0.26 0.12];
% Compute the output of 4th order system
y = filter(num,den,x,ic);
% Coefficients of the two 2nd order systems
num1 = [0.3 -0.2 0.4];den1 = [1 0.9 0.8];
num2 = [0.2 -0.5 0.3];den2 = [1 0.7 0.85];
% Output y1[n] of the first stage in the cascade
y1 = filter(num1,den1,x,ic1);
% Output y2[n] of the second stage in the cascade
y2 = filter(num2,den2,y1,ic2);
% Difference between y[n] and y2[n]
d = y - y2;
% Plot output and difference signals
subplot(3,1,1);
stem(n,y);
ylabel('Amplitude');
title('Output of 4th order Realization'); grid;
subplot(3,1,2);
stem(n,y2)
ylabel('Amplitude');
title('Output of Cascade Realization'); grid;
subplot(3,1,3);
stem(n,d)
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal'); grid;
```



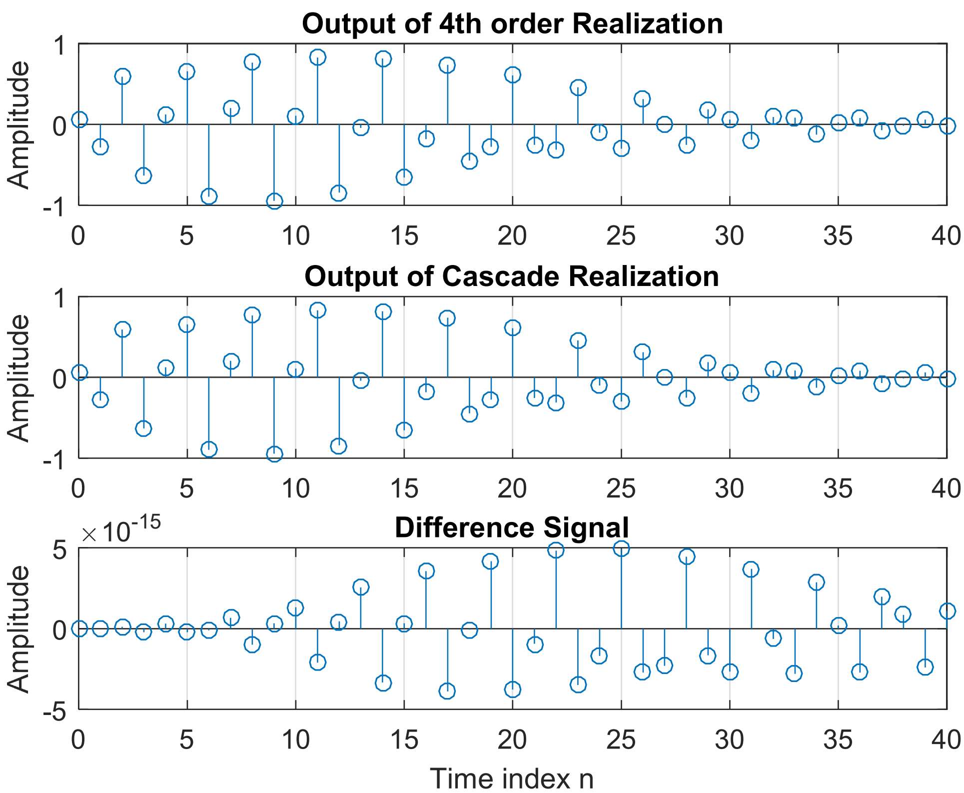
The relation between y[n] and y2[n] in this case is ***different.***

Q2.26 The modified Program P2_6 with the two 2nd-order systems in reverse order and with zero initial conditions is displayed below:

```
% Cascade Realization
clc; clear all; close all;
x = [1 zeros(1,40)]; % Generate the input
n = 0:40;
% Coefficients of 4th order system
den = [1 1.6 2.28 1.325 0.68];
num = [0.06 -0.19 0.27 -0.26 0.12];
% Compute the output of 4th order system
y = filter(num,den,x);
% Coefficients of the two 2nd order systems
num1 = [0.3 -0.2 0.4];den1 = [1 0.9 0.8];
num2 = [0.2 -0.5 0.3];den2 = [1 0.7 0.85];
% Output y1[n] of the first stage in the cascade
y1 = filter(num2,den2,x);
% Output y2[n] of the second stage in the cascade
y2 = filter(num1,den1,y1);
% Difference between y[n] and y2[n]
d = y - y2;
% Plot output and difference signals
```

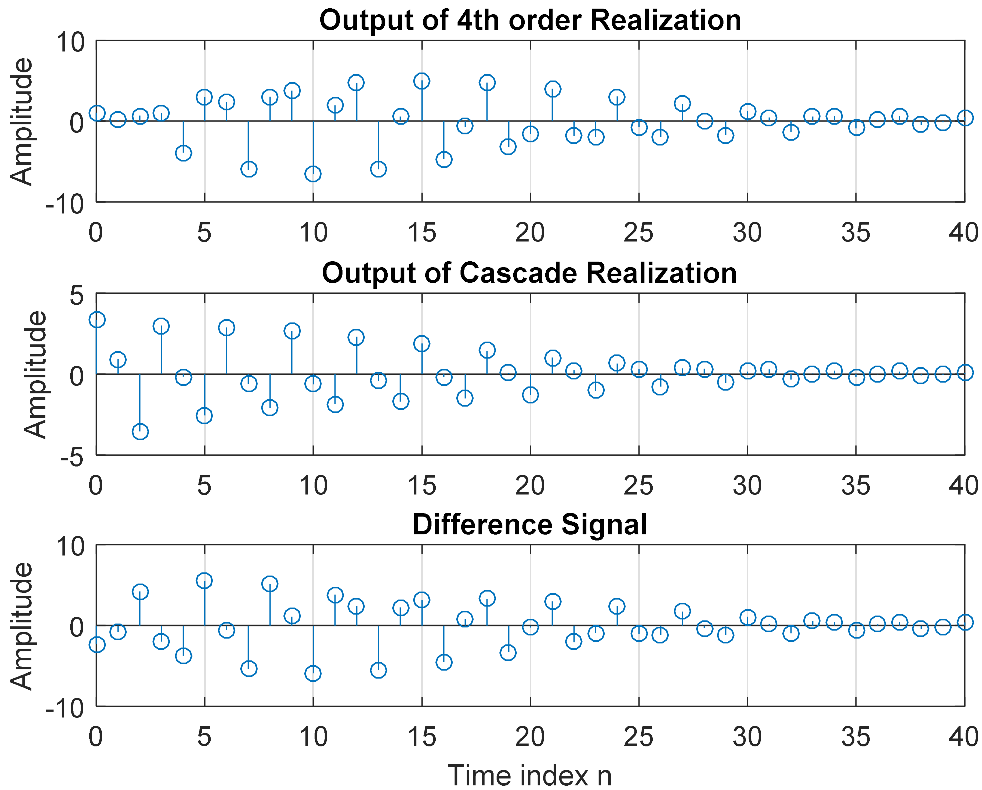
```
subplot(3,1,1);
stem(n,y);
ylabel('Amplitude');
title('Output of 4th order Realization'); grid;
subplot(3,1,2);
stem(n,y2)
ylabel('Amplitude');
title('Output of Cascade Realization'); grid;
subplot(3,1,3);
stem(n,d)
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal'); grid;
```

The sequences generated by running the modified program are sketched below:



The relation between y[n] and y2[n] in this case is ***almost similar.***

Q2.27 The sequences generated by running the modified Program P2_6 with the two 2nd-order systems in reverse order and with non-zero initial conditions are displayed below:



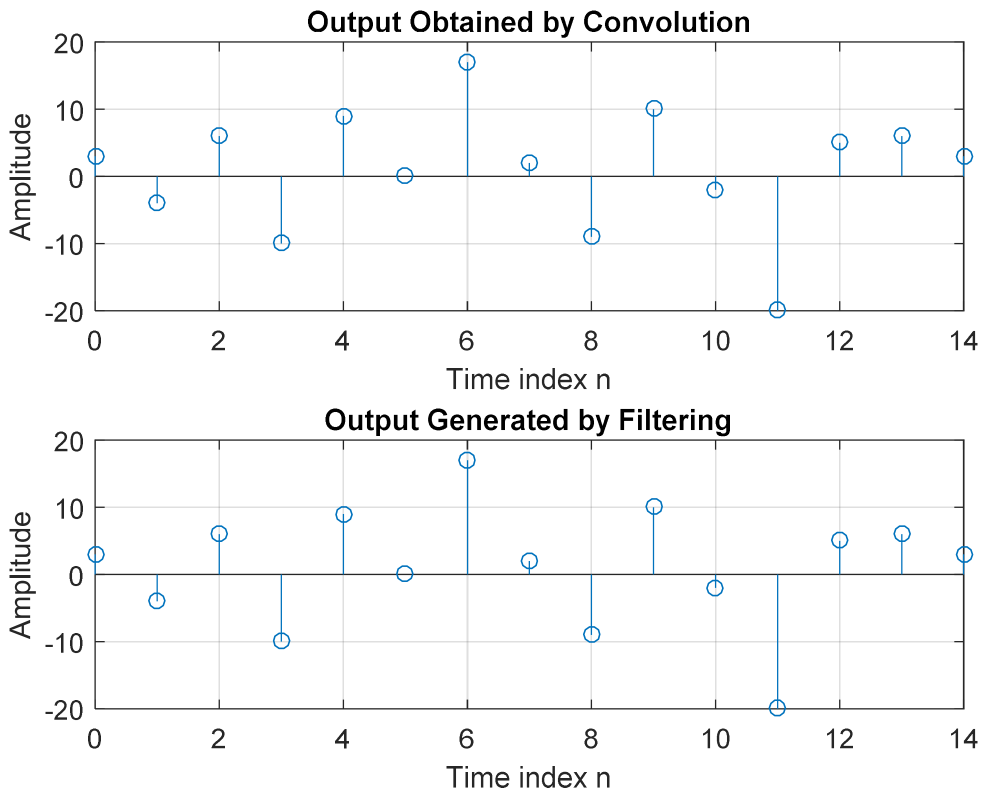
The relation between y[n] and y2[n] in this case is ***different.***

Project 2.7 Convolution

A copy of Program P2_7 is reproduced below:

```
% Program P2_7
clc; clear all; close all;
h = [3 2 1 -2 1 0 -4 0 3];   % impulse response
x = [1 -2 3 -4 3 2 1];       % input sequence
y = conv(h,x);
n = 0:14;
subplot(2,1,1);
stem(n,y);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Obtained by Convolution'); grid;
x1 = [x zeros(1,8)];
y1 = filter(h,1,x1);
subplot(2,1,2);
stem(n,y1);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Generated by Filtering'); grid;
```

Q2.28 The sequences y[n] and y1[n] generated by running Program P2_7 are shown below:



The difference between y[n] and y1[n] is that: ***they are similar.***

The reason for using x1[n] as the input, obtained by zero-padding x[n], for generating y1[n] is:

- ***For two sequences of length N1 and N2, "`conv`" returns the resulting sequence of length N1+N2-1.*** $9+7-1=15$
- ***"`Filter`" returns the resulting sequence of the same length as the input signal. To have a resulting sequences of length 15, we have to append 8 zero elements into x (with length of 7).***

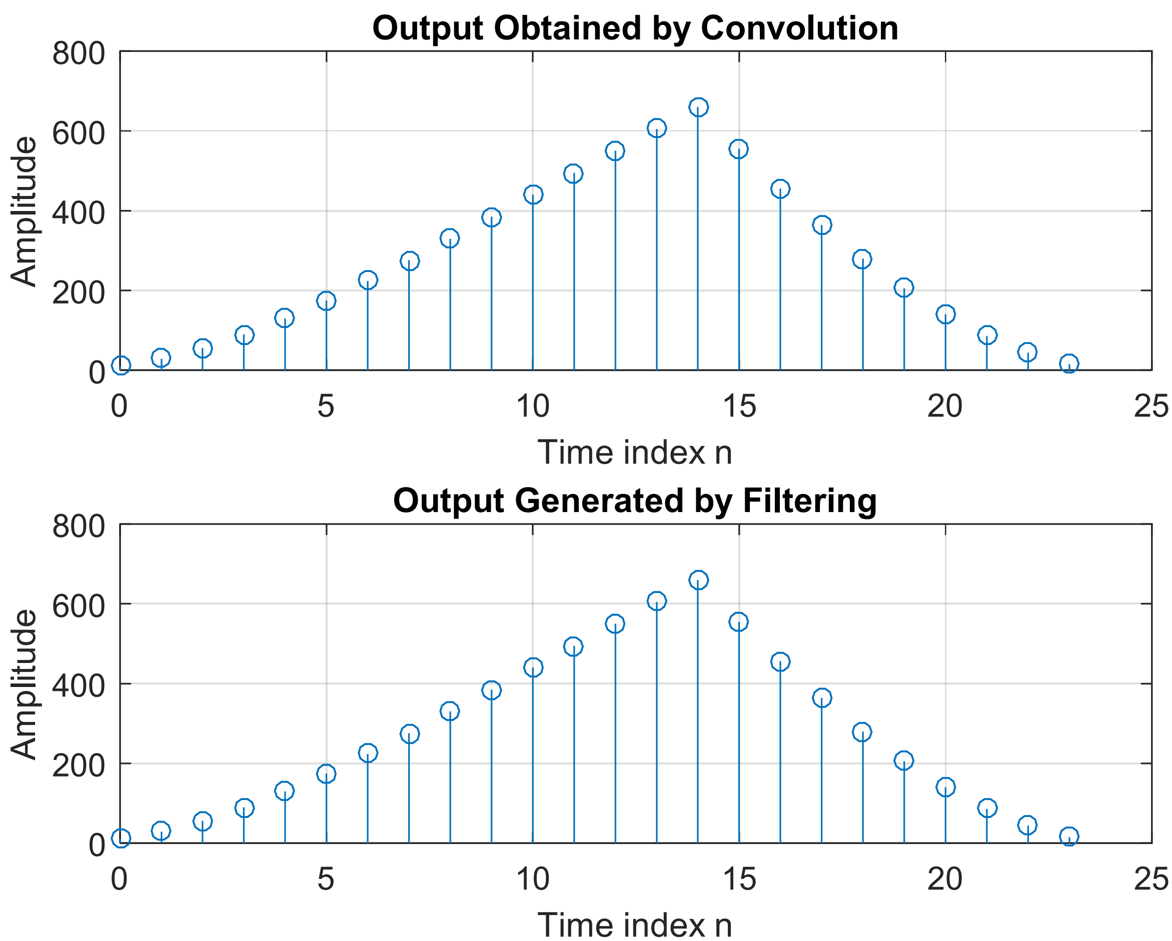
Q2.29 The modified Program P2_7 to develop the convolution of a length-15 sequence h[n] with a length-10 sequence x[n]is indicated below:

```
clc; clear all; close all;
h = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15];  % impulse
response
x = [10 9 8 7 6 5 4 3 2 1];      % input sequence
y = conv(h,x);
n = 0:23;
subplot(2,1,1);
stem(n,y);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Obtained by Convolution'); grid;
```

```
x1 = [x zeros(1,14)];
y1 = filter(h,1,x1);
subplot(2,1,2);
stem(n,y1);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Generated by Filtering'); grid;
```

The sequences y[n] and y1[n] generated by running modified Program P2_7 are shown below:

The difference between y[n] and y1[n] is that: ***they are similar.***



Project 2.8 Stability of LTI Systems

A copy of Program P2_8 is given below:

```
% Program P2_8
% Stability test based on the sum of the absolute
% values of the impulse response samples
clc; clear all; close all;
num = [1 -0.8]; den = [1 1.5  0.9];
N = 200;
```

```
h = impz(num,den,N+1);
parsum = 0;
for k = 1:N+1;
    parsum = parsum + abs(h(k));
    if abs(h(k)) < 10^(-6), break, end
end
% Plot the impulse response
n = 0:N;
stem(n,h)
xlabel('Time index n'); ylabel('Amplitude');
% Print the value of abs(h(k))
disp('Value =');disp(abs(h(k)));
```

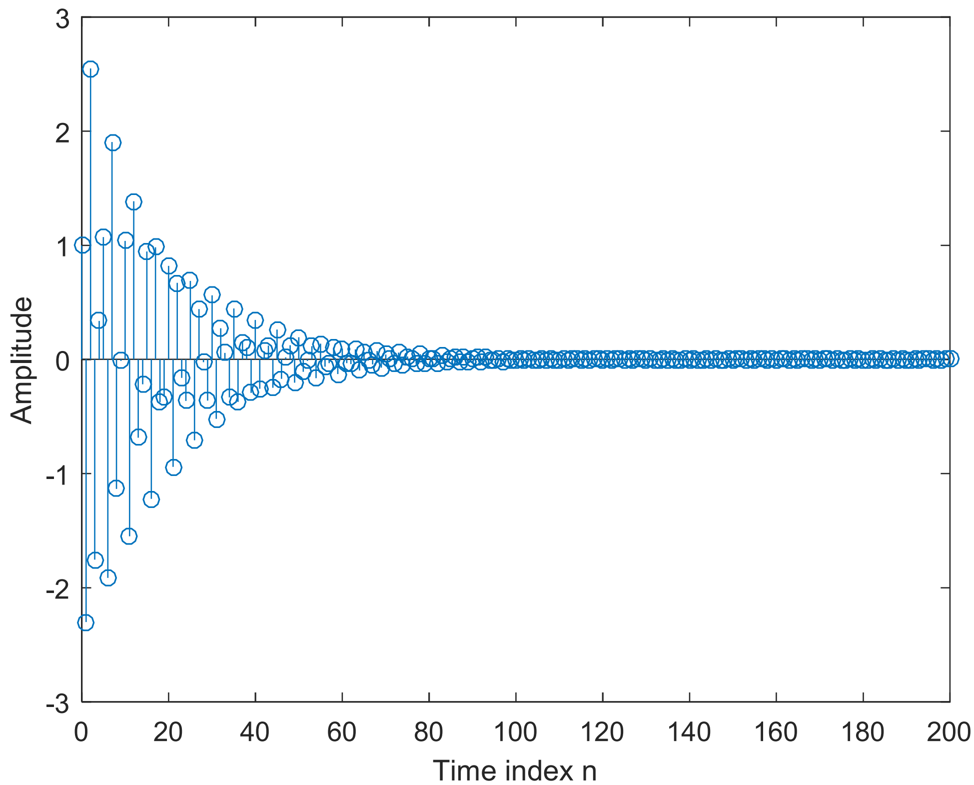
Q2.30 The purpose of the `for` command is ***used for loop to repeat specified number of times.***

The purpose of the `end` command is ***used to terminate block of code, or indicate last array index.***

Q2.31 The purpose of the `break` command is ***used to terminate execution of for or while loop.***

Q2.32 The discrete-time system of Program P2_8 is - `y[n]+1.5y[n-1]+0.9y[n-2]=x[n]-0.8x[n-1]`

The impulse response generated by running Program P2_8 is shown below:



The value of |h(K)| here is - `1.6761e-05`

From this value and the shape of the impulse response we can conclude that the system is ***stable.***

By running Program P2_8 with a larger value of N=500 the new value of |h(K)| is - `9.1752e-07`

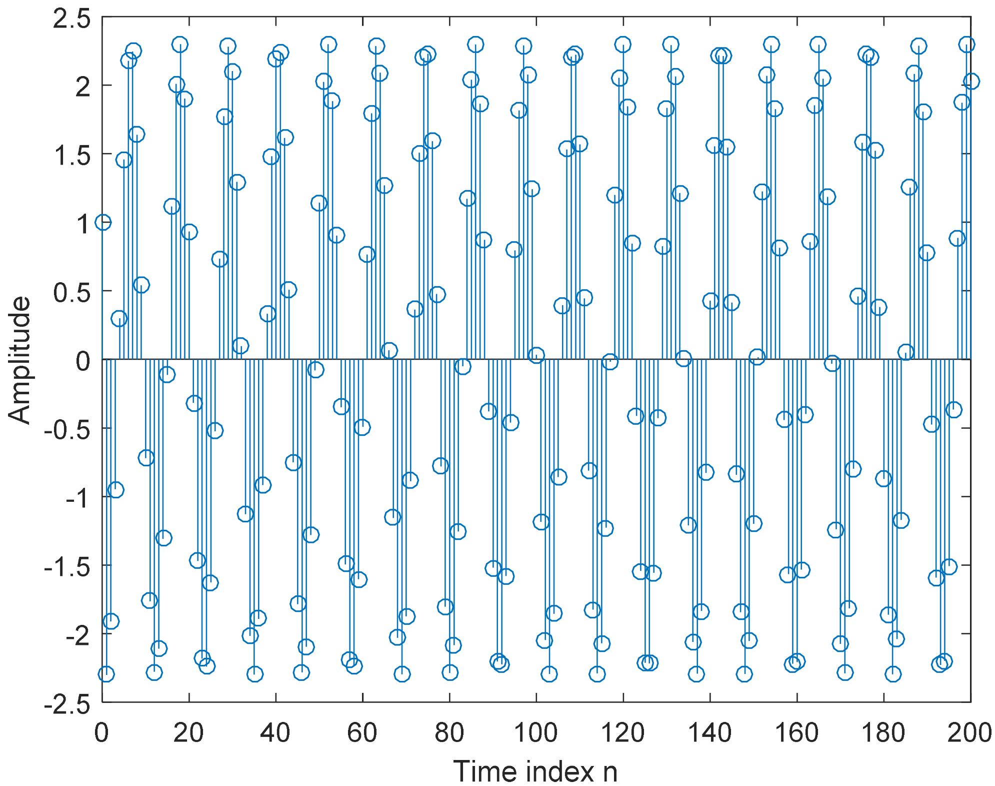
From this value we can conclude that the system is ***stable.***

Q2.33 The modified Program P2_8 to simulate the discrete-time system of Q2.33 is given below:

```
clc; clear all; close all;
num = [1 -4 3]; den = [1 -1.7 1.0];
N = 200;
h = impz(num,den,N+1);
parsum = 0;
for k = 1:N+1;
 parsum = parsum + abs(h(k));
 if abs(h(k)) < 10^(-6), break, end
end
% Plot the impulse response
n = 0:N;
```

```
stem(n,h)
xlabel('Time index n'); ylabel('Amplitude');
% Print the value of abs(h(k))
disp('Value =');disp(abs(h(k)));
```

The impulse response generated by running the modified Program P2_8 is shown below:



The values of |h(K)| here are - `2.0321`

From this value and the shape of the impulse response we can conclude that the system is ***most likely unstable.***

Project 2.9 Illustration of the Filtering Concept

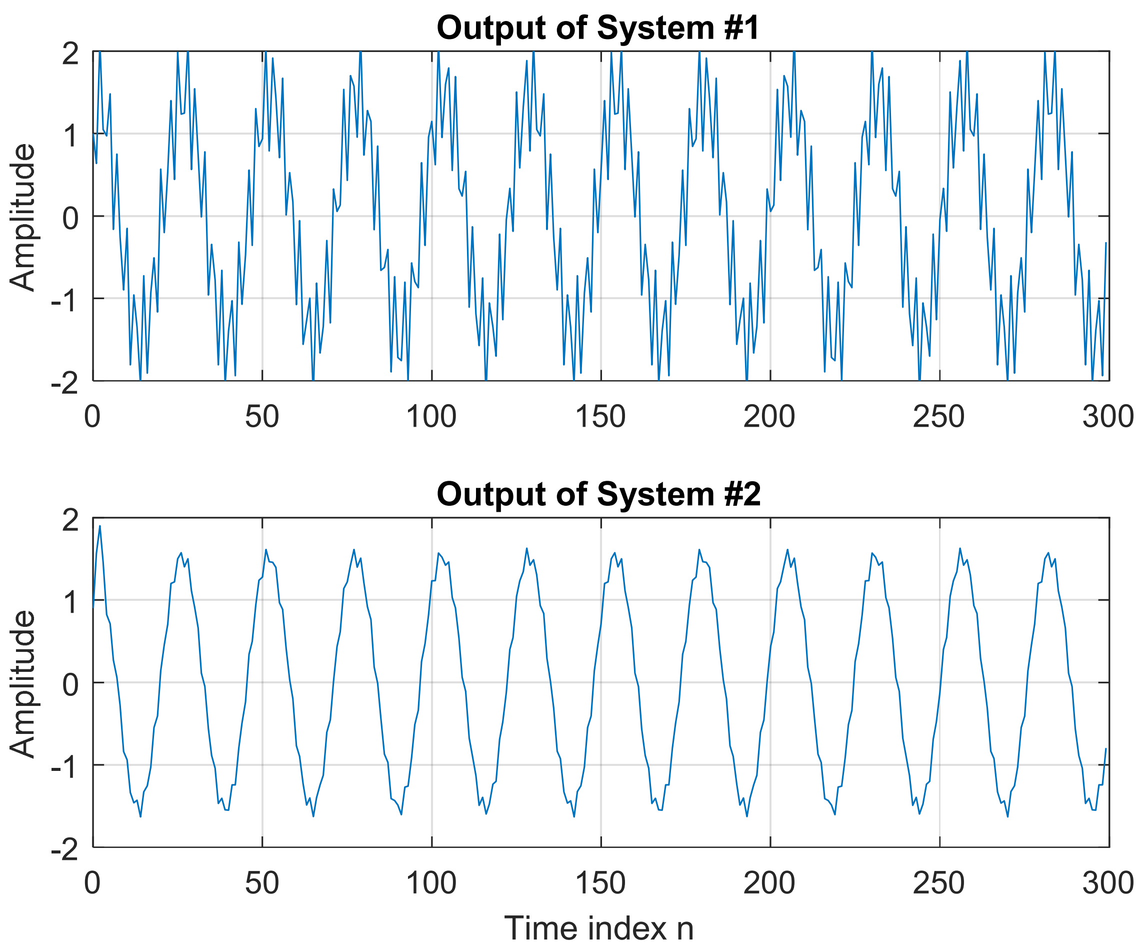
A copy of Program P2_9 is given below:

```
%Program P2_9
% Generate the input sequence
clc; clear all; close all;
n = 0:299;
x1 = cos(2*pi*10*n/256);
x2 = cos(2*pi*100*n/256);
x = x1+x2;
% Compute the output sequences
num1 = [0.5 0.27 0.77];
y1 = filter(num1,1,x); % Output of System #1
den2 = [1 -0.53 0.46];
num2 = [0.45 0.5 0.45];
```

```
y2 = filter(num2,den2,x); % Output of System #2
% Plot the output sequences
subplot(2,1,1);
plot(n,y1);axis([0 300 -2 2]);
ylabel('Amplitude');
title('Output of System #1'); grid;
subplot(2,1,2);
plot(n,y2);axis([0 300 -2 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output of System #2'); grid;
```

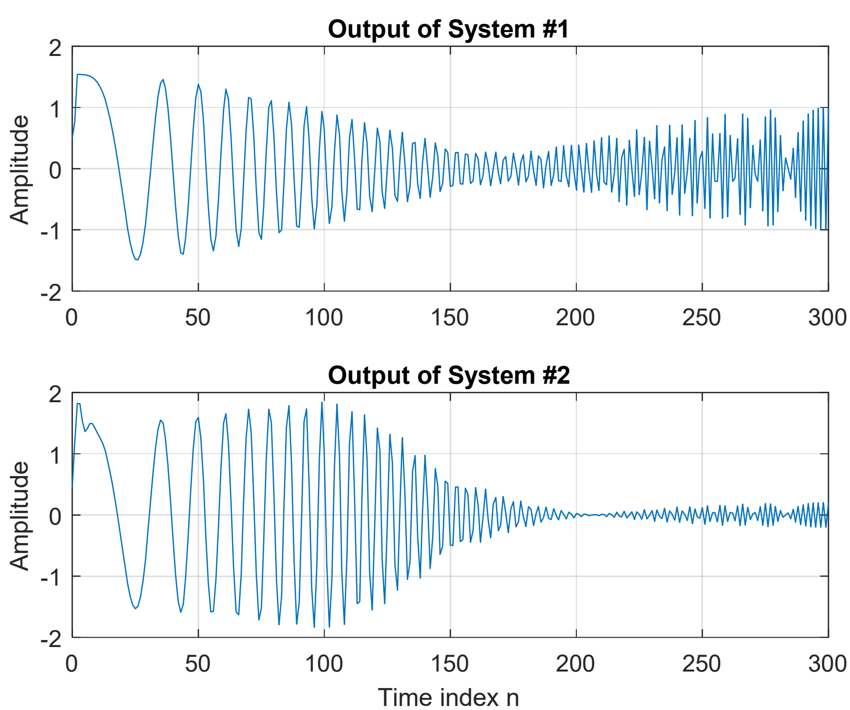
Q2.34 The output sequences generated by this program are shown below:

The filter with better characteristics for the suppression of the high frequency component of the input signal x[n] is – ***System #2.***



Q2.35 The required modifications to Program P2_9 by changing the input sequence to a swept sinusoidal sequence (length 301, minimum frequency 0, and maximum frequency 0.5) are listed below along with the output sequences generated by the modified program:

```
clc; clear all; close all;
n = 0:300;
a = pi/600;
b = 0;
arg = a*n.*n + b*n;
x = cos(arg);
% Compute the output sequences
num1 = [0.5 0.27 0.77];
y1 = filter(num1,1,x); % Output of System #1
den2 = [1 -0.53 0.46];
num2 = [0.45 0.5 0.45];
y2 = filter(num2,den2,x); % Output of System #2
% Plot the output sequences
subplot(2,1,1);
plot(n,y1);axis([0 300 -2 2]);
ylabel('Amplitude');
title('Output of System #1'); grid;
subplot(2,1,2);
plot(n,y2);axis([0 300 -2 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output of System #2'); grid;
```



The filter with better characteristics for the suppression of the high frequency component of the input signal x[n] is – ***System #2.***

Date: 17/9/2023 **Signature: Do Trung Hau**