**EE387: Signal Processing**

**Lab 2: Laboratory on Discrete Time Signals**

Ranage R.D.P.R. - E/19/310

**1). Understanding properties of Discrete Time Sinusoidal signals**

**a. Plot the discrete time real sinusoidal signal 𝑥[𝑛] = 10𝛽n for positive 𝐶 when,**

**i. 𝛽 < −1**

**ii. −1 < 𝛽 < 0**

**iii. 0 < 𝛽 < 1**

**iv. 𝛽 > 1**

% Define values for n

n = linspace(0, 2\*pi, 50)';

% Define different values of B

Bs = [-2.5, -0.75, 0.75, 2.5];

% Create a new figure for subplots

figure;

% Loop through each value of B and plot in a subplot

for i = 1:length(Bs)

% Calculate result for current B

result = xnB(n, Bs(i));

% Create subplot for current result

subplot(length(Bs), 1, i);

stem(n, result);

title(['Plot of x[n] for B = ', num2str(Bs(i))]);

xlabel('n');

ylabel('x[n]');

grid on;

end

function result = xnB(n, B)

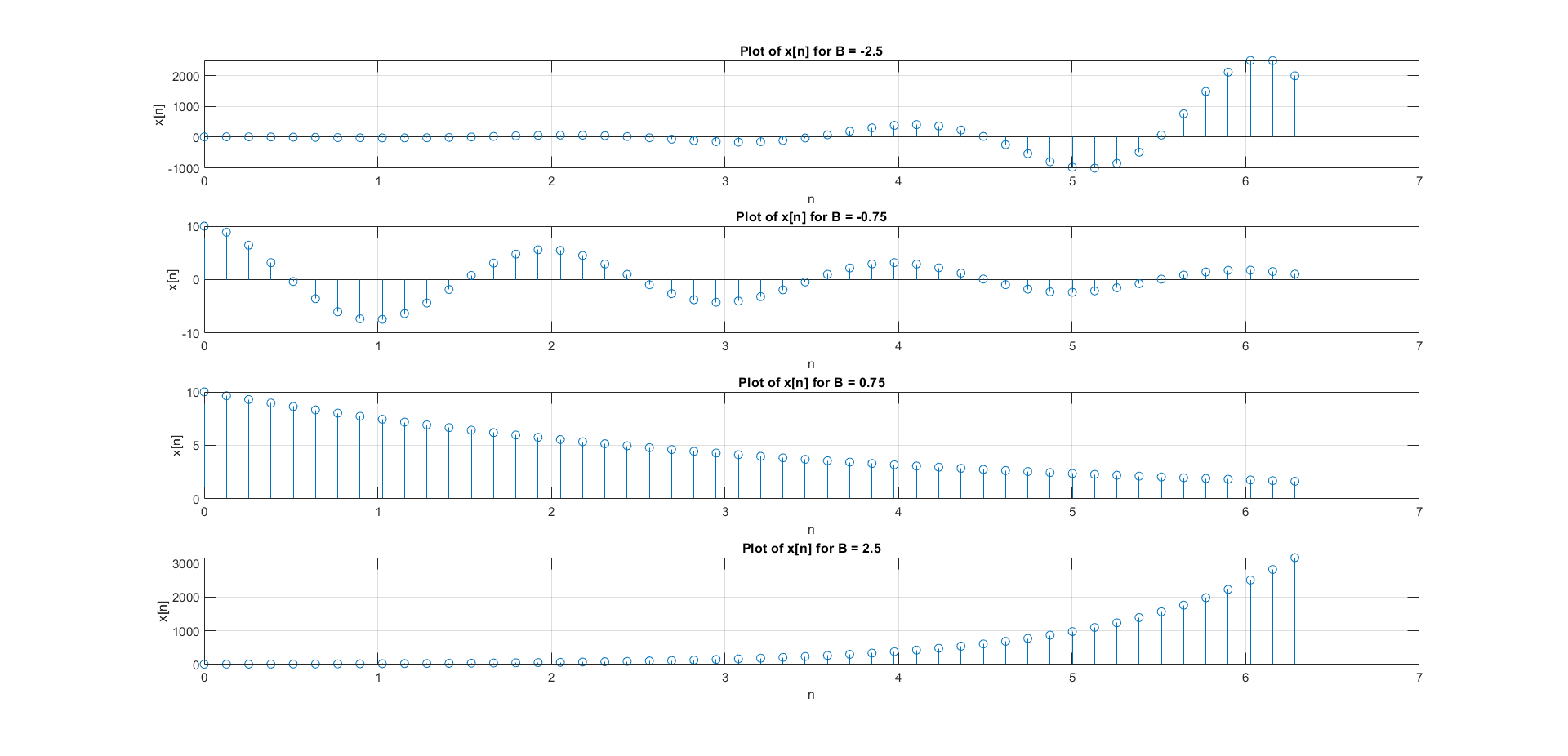
% generate the result as given in the equation

result = 10 \* (B .^ n);

% Ensure that the result is real

result = real(result);

end

****

**b. Plot 𝑥[𝑛] and 𝑥(𝑡) in the same plot for the following sinusoidal signals. Let 𝑛 = 𝑘𝑇 where**

**𝑇 = 5𝑠 and 𝑘 ∈ 𝑍. That is 𝑥[𝑛] is obtained by sampling 𝑥[𝑡] at every 5 seconds. Determine the theoretical fundamental period of each signal.**

**i).**

**ii).**

**Is the observed period of the signal from the plot always equal to the theoretical period?**

% sampling frequency

Ts = 5;

% t to plot continuous time signals

t1 = 0:0.01:70;

% n = KTs

n1 = 0:Ts:70;

xt1 = cos(t1\*2\*pi/12);

xn1 = cos(n1\*2\*pi/12);

% t to plot continuous time signals

t2 = 0:0.01:100;

% n = KTs

n2 = 0:Ts:100;

xt2 = cos(t2\*8\*pi/31);

xn2 = cos(n2\*8\*pi/31);

figure;

subplot(2,1,1);

plot(t1,xt1,'linewidth',1.5);

hold on;

stem(n1,xn1);

grid on;

title('Plot of x\_n and its sampled signal part 1')

legend('X(t)','X(n)');

subplot(2,1,2);

plot(t2,xt2);

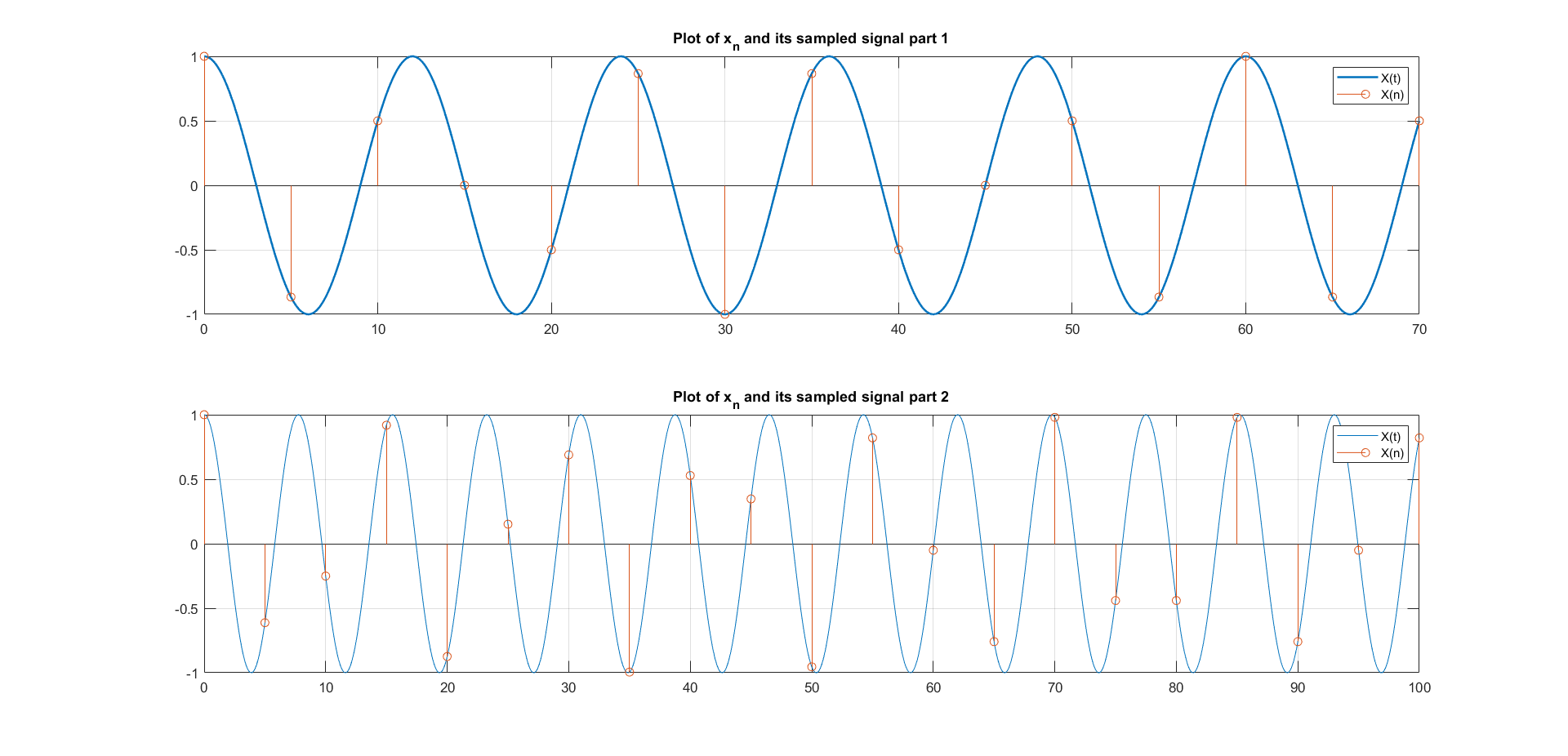
hold on;

stem(n2,xn2);

grid on;

title('Plot of x\_n and its sampled signal part 2')

legend('X(t)','X(n)');

****

i). Theoretical fundamental period of *x*(*t*)

*w*0*t* = 2 \* *pi* \* *t* / 12

*T* = 2 \* *pi* / *w*0

*T* = 12*s*

Observed fundamental period of *x*[*n*] = 60

ii). Theoretical fundamental period of x(t)

*w*0*t* = 8 \* *pi* \* *t* / 31

*T* = 2 \* *pi* / *w*0

*T* = 31/4*s*

*T* = 7.5*s*

Observed fundamental period of x[n] is not detectable, (> 100)

**c. Plot the following nine discrete time signals in the same graph (use subplot command)**

% sampling frequency

Ts = 3;

% n = KTs

n = (0:Ts:70)';

x1n = cos(n\*0.1);

subplot(3,3,1);

stem(n,x1n);

grid on;

xlabel('n');

ylabel('x\_1n');

x2n = cos(n\*pi/8);

subplot(3,3,2);

stem(n,x2n);

grid on;

xlabel('n');

ylabel('x\_2n');

x3n = cos(n\*pi/4);

subplot(3,3,3);

stem(n,x3n);

grid on;

xlabel('n');

ylabel('x\_3n');

x4n = cos(n\*pi/2);

subplot(3,3,4);

stem(n,x4n);

grid on;

xlabel('n');

ylabel('x\_4n');

x5n = cos(n\*pi);

subplot(3,3,5);

stem(n,x5n);

grid on;

xlabel('n');

ylabel('x\_5n');

x6n = cos(n\*pi\*3/2);

subplot(3,3,6);

stem(n,x6n);

grid on;

xlabel('n');

ylabel('x\_6n');

x7n = cos(n\*pi\*7/4);

subplot(3,3,7);

stem(n,x7n);

grid on;

xlabel('n');

ylabel('x\_7n');

x8n = cos(n\*pi\*15/8);

subplot(3,3,8);

stem(n,x8n);

grid on;

xlabel('n');

ylabel('x\_8n');

x9n = cos(n\*pi\*2);

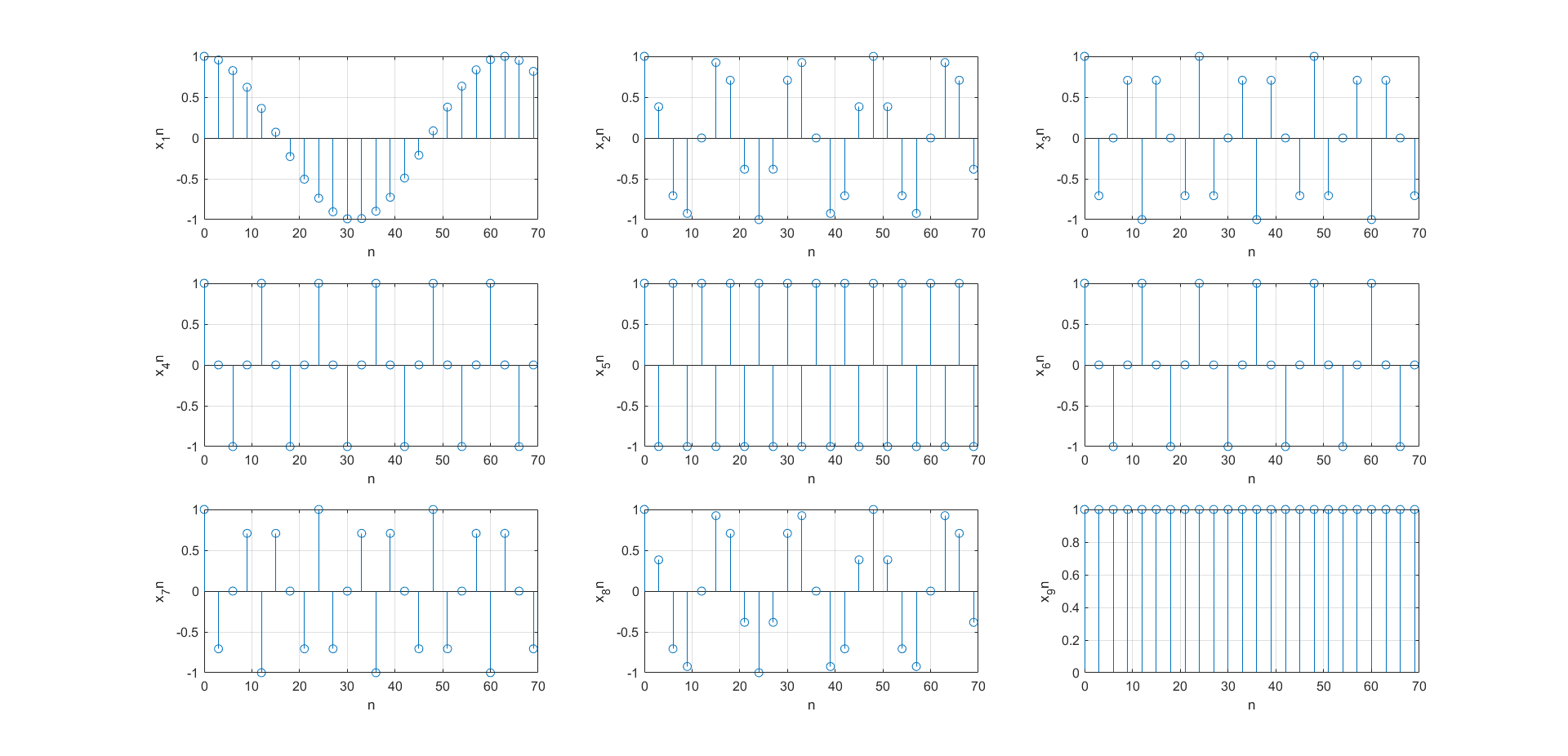
subplot(3,3,9);

stem(n,x9n);

grid on;

xlabel('n');

ylabel('x\_9n');

****

**d. By observing the plots you have obtained in question 1.c, what can you tell about the shape of the signal as discrete frequency is varied?**

When the sampling frequency is decreased, fewer data points are captured per unit of time, resulting in information loss. This leads to the original form of the signal being altered, as some data is no longer represented. Consequently, the signal's fidelity is compromised, and distortions may arise, especially at lower frequencies, due to inadequate data representation.

**2). Discrete convolution**

**a. Write a matlab function to implement discrete convolution for 𝑛 > 0. Note that**

**𝑦[𝑛] = 𝑥[𝑛] ∗ ℎ[𝑛] is given by the convolution summation**

function y = conv\_disc(xn, hn)

% Determine the lengths of xn and hn

sizeX = length(xn);

sizeH = length(hn);

% Pad xn and hn with zeros to make their lengths equal to the sum of their original lengths minus one

xn = [xn, zeros(1, sizeH - 1)];

hn = [hn, zeros(1, sizeX - 1)];

% Determine the size of the output

out\_size = sizeH + sizeX - 1;

% Pre-allocate memory for the output array for efficiency

y = zeros(1, out\_size);

% Perform convolution using nested loops

for n = 1:out\_size

kmin = max(1, n - sizeH + 1);

kmax = min(n, sizeX);

for k = kmin:kmax

y(n) = y(n) + xn(k) \* hn(n - k + 1);

end

end

end

**b. Using the function written in section a, convolve 𝑥[𝑛] = 0.5n𝑢(𝑛) with ℎ[𝑛] = 𝑢[𝑛].**

**Plot the output signal along with the two input signals.**

% Define the input signals x[n] and h[n]

n = 0:20; % Define the range of n

x = 0.5.^n .\* (n >= 0); % x[n] = 0.5^n \* u[n]

h = (n >= 0); % h[n] = u[n]

y = conv\_disc(x, h);

subplot(3,1,1);

stem(n, x, 'b', 'filled'); % Plot x[n]

xlabel('n');

ylabel('Amplitude');

title('Input Signal x[n]');

subplot(3,1,2);

stem(n, h, 'r', 'filled'); % Plot h[n]

xlabel('n');

ylabel('Amplitude');

title('Impulse Response h[n]');

ny = 0:length(y)-1;

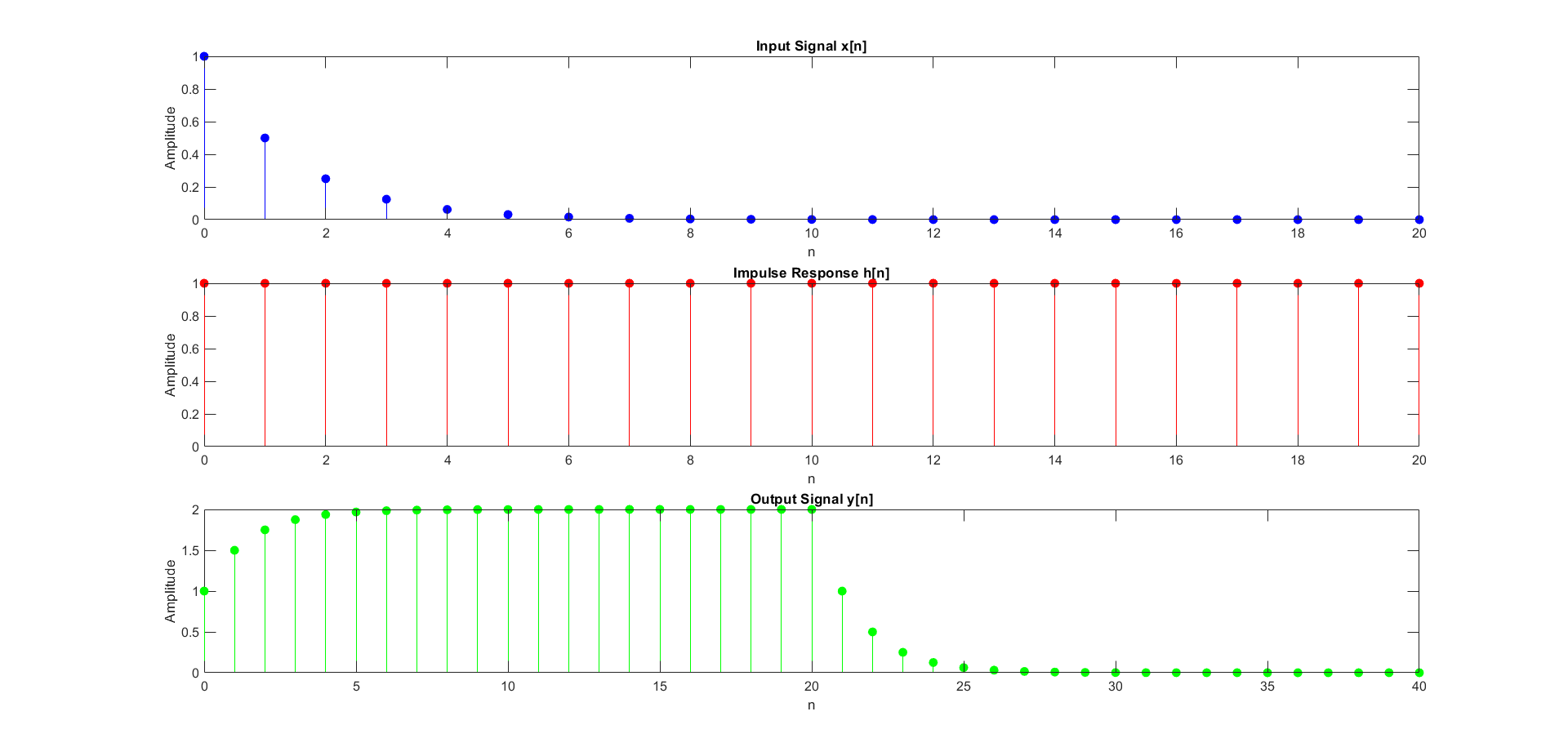
subplot(3,1,3);

stem(ny, y, 'g', 'filled'); % Plot the output signal

xlabel('n');

ylabel('Amplitude');

title('Output Signal y[n]');



**c. Consider the following two signals**

1. **X[n] = [ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0]**
2. **h[n]= [2 4 8 16 32 64 0 0 0 0 0 0 0 0 0 ]**
3. **Convolve the two signals using the function written in part a. Use matlab conv command to verify your answer.**

% Define the input signals X[n] and h[n]

X = [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0];

h = [2 4 8 16 32 64 0 0 0 0 0 0 0 0 0];

y\_conv\_disc = conv\_disc(X, h);

y\_conv\_matlab = conv(X, h);

n\_x = 0:length(X)-1;

n\_h = 0:length(h)-1;

n\_y = 0:length(y\_conv\_disc)-1;

subplot(4,1,1);

stem(n\_x, X, 'b', 'filled'); % Plot X[n]

xlabel('n');

ylabel('Amplitude');

title('Input Signal X[n]');

subplot(4,1,2);

stem(n\_h, h, 'r', 'filled'); % Plot h[n]

xlabel('n');

ylabel('Amplitude');

title('Impulse Response h[n]');

subplot(4,1,3);

stem(n\_y, y\_conv\_disc, 'g', 'filled'); % Plot output signal from conv\_disc

xlabel('n');

ylabel('Amplitude');

title('Output Signal (conv\\_disc)');

subplot(4,1,4);

stem(n\_y, y\_conv\_matlab, 'm', 'filled'); % Plot output signal from MATLAB's conv

xlabel('n');

ylabel('Amplitude');

title('Output Signal (conv)');

function y = conv\_disc(xn, hn)

% Determine the lengths of xn and hn

sizeX = length(xn);

sizeH = length(hn);

xn = [xn, zeros(1, sizeH - 1)];

hn = [hn, zeros(1, sizeX - 1)];

% Determine the size of the output

out\_size = sizeH + sizeX - 1;

% Pre-allocate memory for the output array for efficiency

y = zeros(1, out\_size);

% Perform convolution using nested loops

for n = 1:out\_size

kmin = max(1, n - sizeH + 1);

kmax = min(n, sizeX);

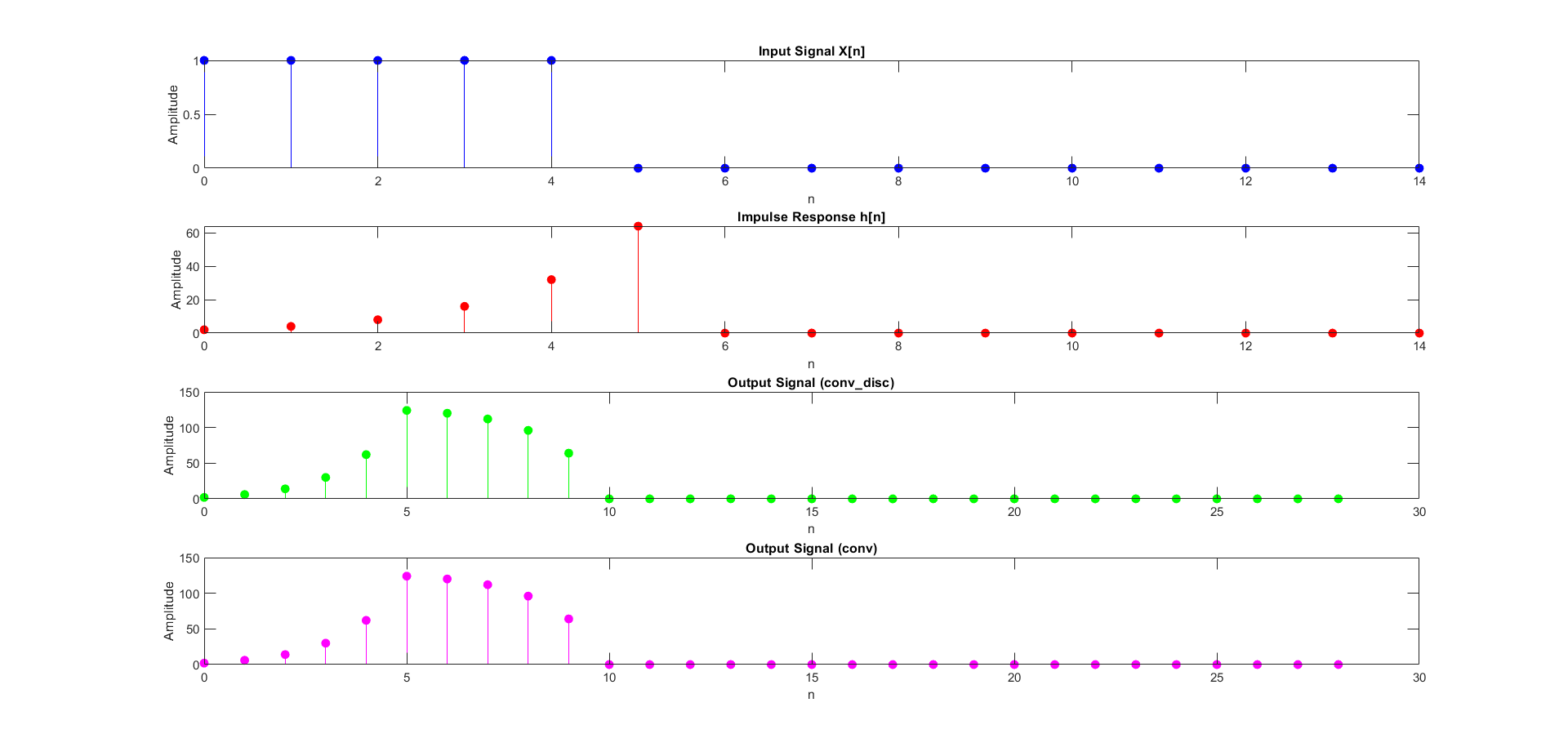
for k = kmin:kmax

y(n) = y(n) + xn(k) \* hn(n - k + 1);

end

end

end



**iv. Considering the shape of the signal h[n] and the output signal, what sort of a transformation has been applied through the convolution operation?**

There has been a linear transformation happened for h[n]

[Y n] = 2 \* h[n] − 2 for x[n] = 1;

**3). LTI Systems**

**a. Consider the following processes. Identify intput x[n] and the output y[n] for each case.**

**Implement a matlab function to implement the given system.**

**i. An investor is maintaining a bank account. The bank pays him a monthly interest of 1%. It is**

**given that the net savings he makes is P. Write a function to calculate his current bank balance B in terms of B and P.**

**ii. A merchant earns M amount of money monthly. He spends half of it and retains the rest of it as savings. Write a function to calculate the amount of money he has as savings.**

function y = bank\_balance(initial\_balance)

interest\_rate = 0.01; % interest rate

temp\_balance = initial\_balance(1); % store initial balance in a temporary variable

y(1) = temp\_balance \* (1 + interest\_rate); % first month's balance

for i = 2:length(initial\_balance)

temp\_balance = y(i - 1) + initial\_balance(i); % update temporary balance

y(i) = temp\_balance \* (1 + interest\_rate); % generate balance for each month

end

end

% M will be an array something like [100,10,0,0]

function y = merchant\_savings(M)

y(1) = M(1)/2; % first month's savings

current\_savings = y(1); % store the current savings

% calculate next savings based on savings given and add the previous savings

for i = 2:length(M)

current\_savings = current\_savings + (M(i)/2); % update current savings

y(i) = current\_savings;

end

end

**b. Find the impulse response of the above two LTI systems. Hint: you may use convolution function to obtain the impulse response**

xn = 1:5:100;

yn = bank\_balance(xn); % get yn

% deconv to get hn - impulse response

hn = deconv(yn,xn);

display("Impulse response of banking system");

display(hn);

figure; % New figure for banking system

subplot(3,1,1);

stem(xn, 'Color', [0.8 0.2 0.2]); % Red color for X(n)

grid on;

title('X(n)');xlabel('n');ylabel('x(n)');

subplot(3,1,2);

stem(yn, 'Color', [0.2 0.8 0.2]); % Green color for y(n)

title('y(n)');xlabel('n');ylabel('y(n)');

grid on;

subplot(3,1,3);

stem(hn, 'Color', [0.2 0.2 0.8]); % Blue color for h(n)

title('h(n) = Impulse response of banking system');xlabel('n');ylabel('h(n)');

grid on;

xn = 1:5:100;

yn = merchant\_savings(xn); % get yn

hn = deconv(yn,xn);

display("Impulse response of merchant savings system");

display(hn);

figure;

subplot(3,1,1);

stem(xn, 'Color', [0.8 0.2 0.2]); % Red color for X(n)

grid on;

title('X(n)');xlabel('n');ylabel('x(n)');

subplot(3,1,2);

stem(yn, 'Color', [0.2 0.8 0.2]); % Green color for y(n)

title('y(n)');xlabel('n');ylabel('y(n)');

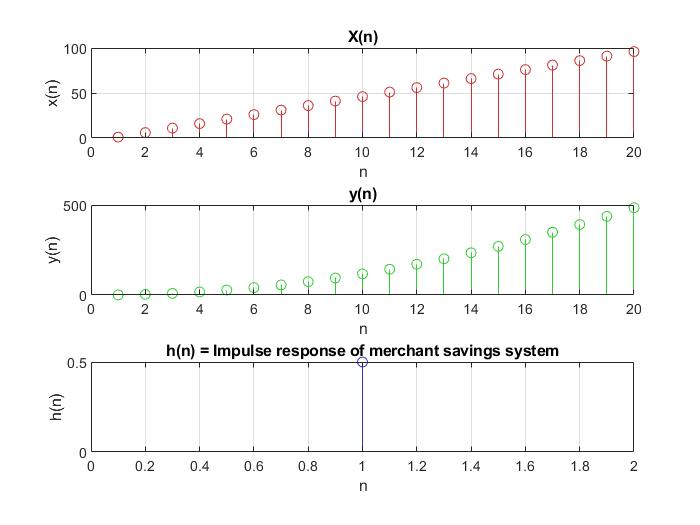
grid on;

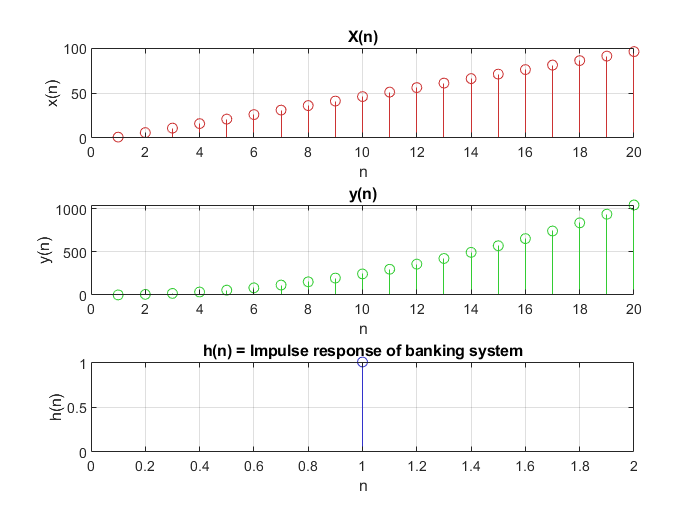
subplot(3,1,3);

stem(hn, 'Color', [0.2 0.2 0.8]); % Blue color for h(n)

title('h(n) = Impulse response of merchant savings system');xlabel('n');ylabel('h(n)');

grid on;





**c. Based on the results obtained at part b, classify the two LTI systems into IIR or FIR.**

The system's two outputs are reliant on its earlier outputs. As a result, the output will not eventually zero out. Consequently, both systems fall within the category of infinite impulse response systems.