EE387 – LABORATORY ON DISCRETE TIME SIGNALS

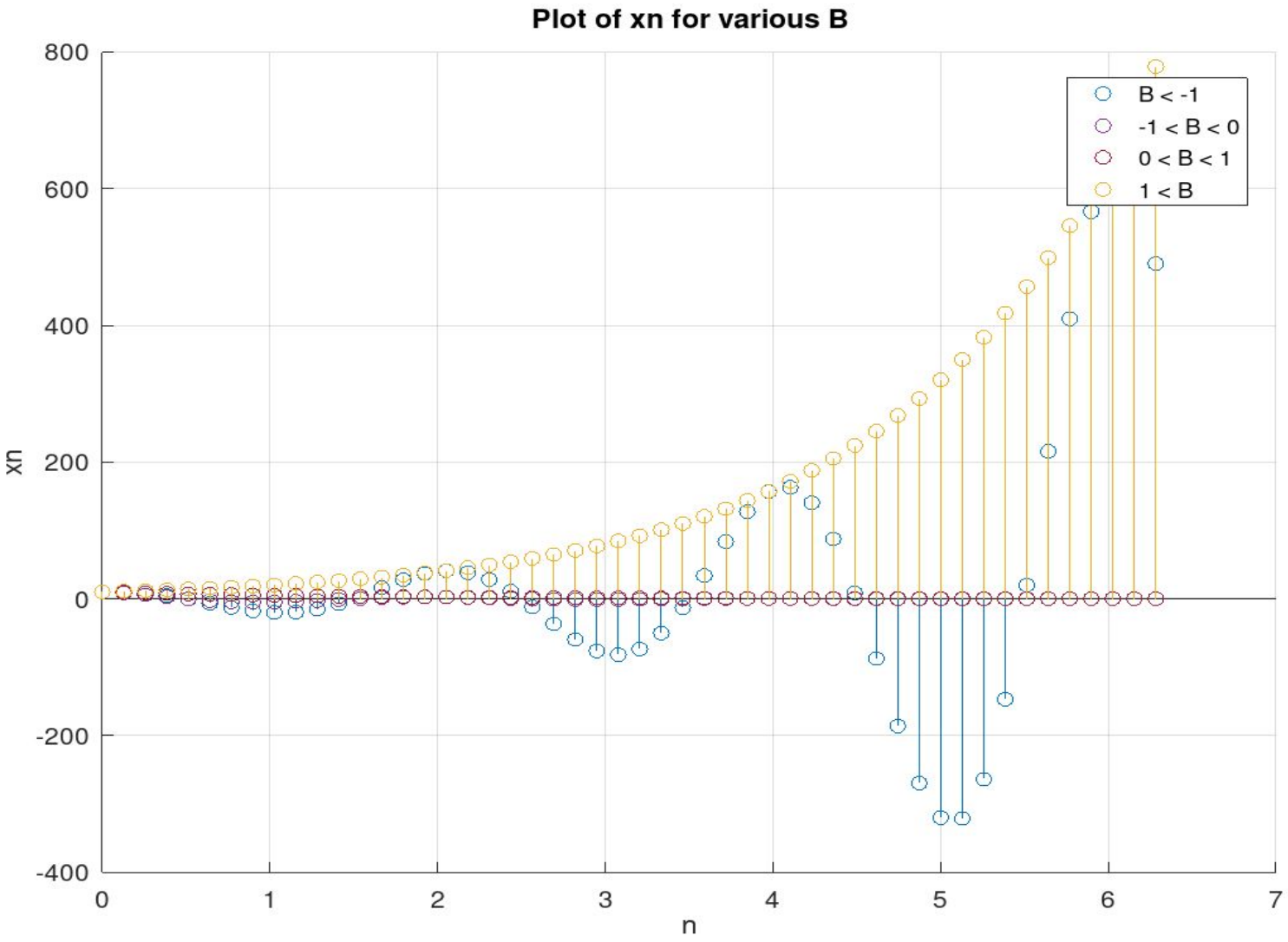
IRFAN M.M.M.
E/15/138
SEMESTER 06
10/04/2020

1) Understanding properties of Discrete Time Sinusoidal signals
a)Plot the discrete time real sinusoidal signal $x[n] = 10B^n$ for positive C.

Matlab Script for part a) ex1a.m

```
% n from 1 to 100
n = linspace(0,2*pi,50)';

% B < -1
B = -2;
result1 = xnB(n,B);


% -1 < B < 0
B = -0.5;
result2 = xnB(n,B);

% 0 < B < 1
B = 0.5;
result3 = xnB(n,B);

% 1 < B
B = 2;
result4 = xnB(n,B);

Y = [result1',result2',result3',result4'];

stem(n,Y);
title('Plot of xn for various B')
xlabel('n')
ylabel('xn')
legend('B < -1','-1 < B < 0','0 < B < 1','1 < B');
grid on;
```

Plots Created by the code snippet ex1a.m



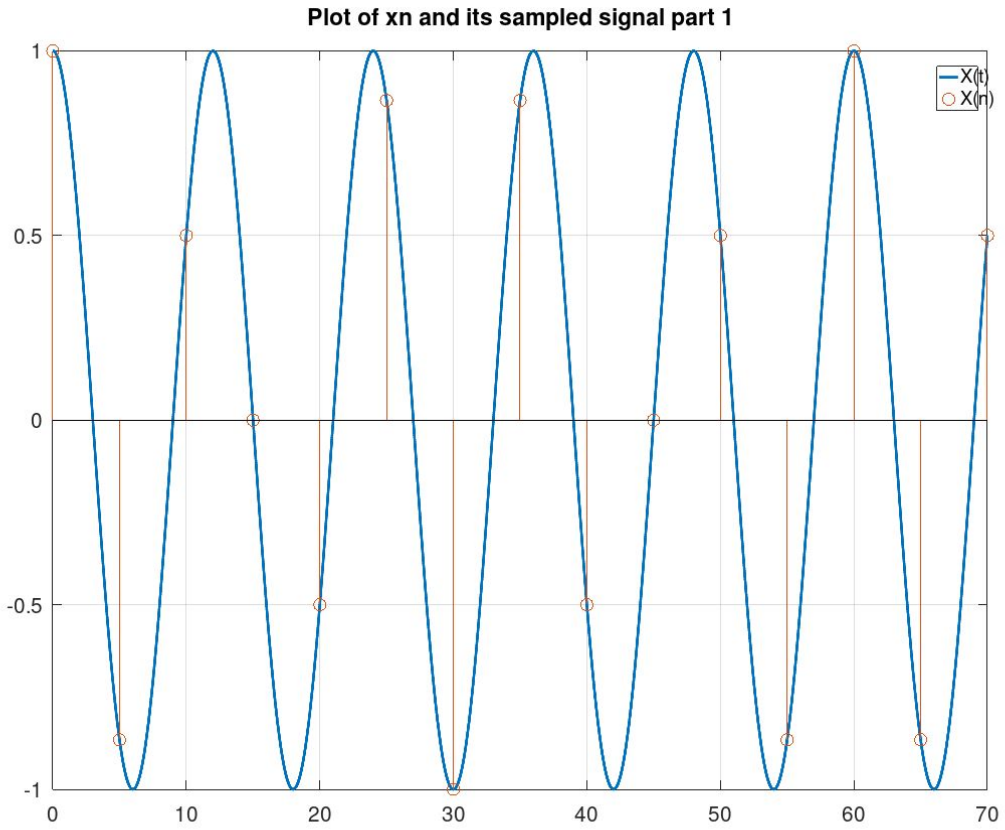
b) Plot x[n], $x(t)$ in same plot for the following sinusoidal signals. Let $n = kT$ where $T = 5s$ and $k \in Z$. That is x[n] is obtained by sampling x[t] every 5 seconds. Determine the theoretical fundamental period of each signal. Is the observed period of the signal from the plot always equal to the theoretical period?

Matlab Script for part b) ex1b.m

```
% sampling frequency
Ts = 5;

% part 1

% t to plot continuous time signals
t = 0:0.01:70;
```

```
% n = KTs
n = 0:Ts:70;

% generate X(t)
xt = cos(t*2*pi/12);
% generate X(n)
xn = cos(n'*2*pi/12);

% plot them in the same graph
plot(t,xt,'linewidth',1.5);
hold on;
stem(n,xn);
grid on;
title('Plot of xn and its sampled signal part 1')
legend('X(t)','X(n)');

% part 2

% t to plot continuous time signals
t = 0:0.01:100;
% n = KTs
n = 0:Ts:100;



% generate X(t)
xt = cos(t*8*pi/31);
% generate X(n)
xn = cos(n'*8*pi/31);

% plot in new figure for the new x(t),x(n)
figure;


% plot them in the same graph
plot(t,xt);
hold on;
stem(n,xn);
grid on;
title('Plot of xn and its sampled signal part 2')
legend('X(t)','X(n)');
```

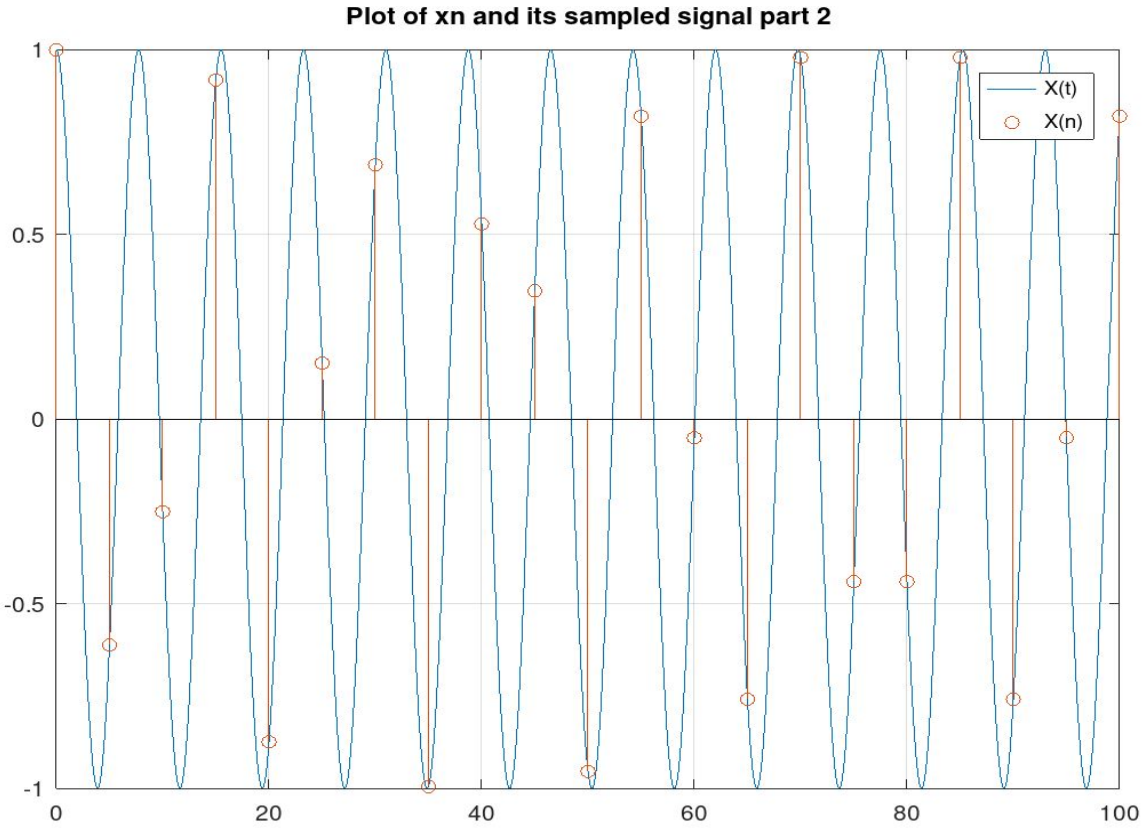
Plots Created by the code snippet ex1b.m



Theoretical fundamental period of $x(t)$

$$w_0 t = 2 * pi * t / 12$$
$$T = 2 * pi / w_0$$
$$= 12s$$

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



Theoretical fundamental period of $x(t)$

$$w_0 t = 8 * pi * t / 31$$
$$T = 2 * pi / w_0$$
$$= 31/4s$$
$$= 7.5s$$

Observed fundamental period of $x[n] = not\ detectable, (> 100)$

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

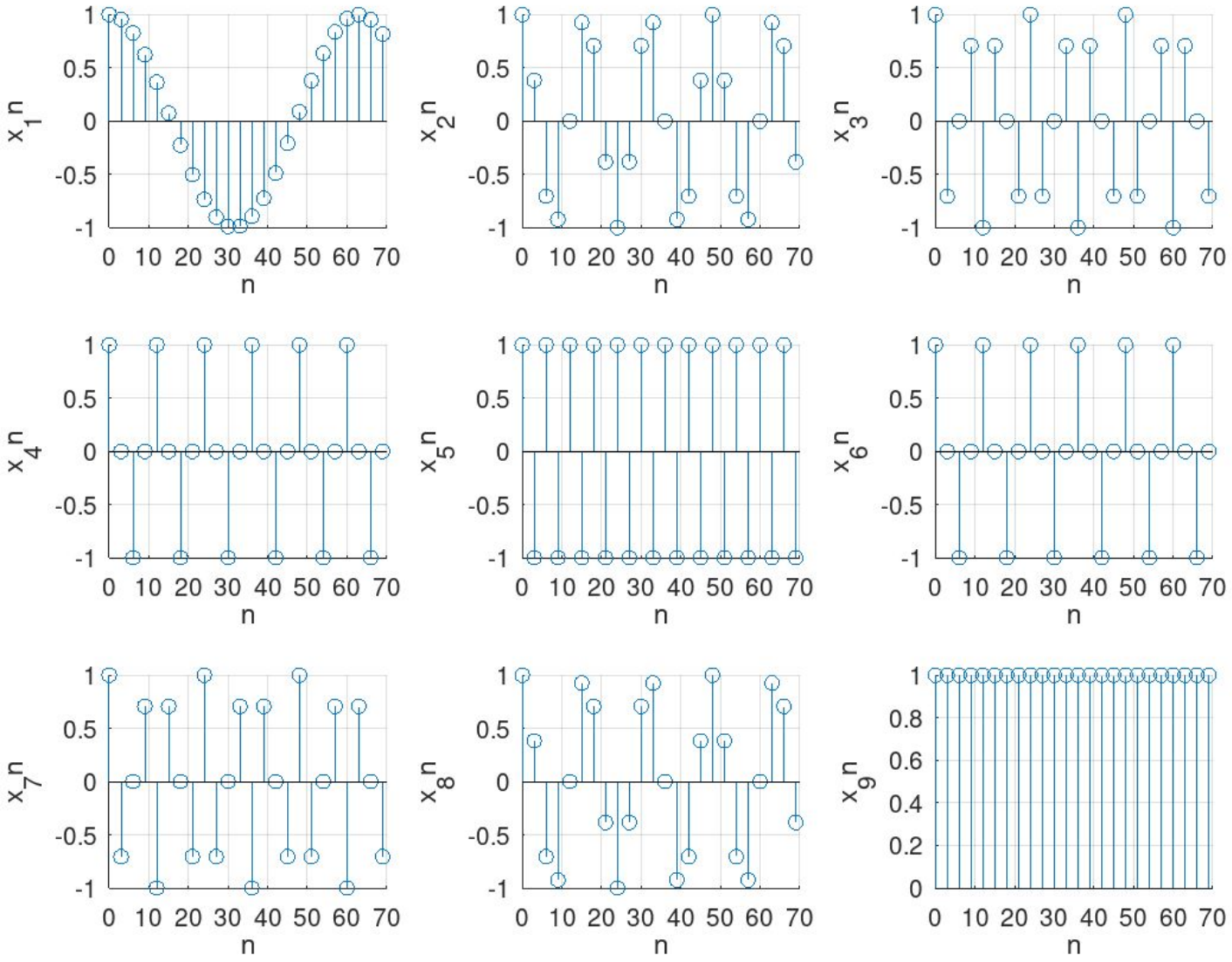
Matlab Script for part c) ex1c.m

```
% sampling frequency
Ts = 3;

% n = KTs
n = (0:Ts:70)';
```

```
% generate plot for part 1
x1n = cos(n*0.1);

subplot(3,3,1);
stem(n,x1n);
grid on;
xlabel('n');
ylabel('x_1n');


% generate plot for part 2
x2n = cos(n*pi/8);

subplot(3,3,2);
stem(n,x2n);
grid on;
xlabel('n');
ylabel('x_2n');

% generate plot for part 3
x3n = cos(n*pi/4);

subplot(3,3,3);
stem(n,x3n);
grid on;
xlabel('n');
ylabel('x_3n');

% generate plot for part 4
x4n = cos(n*pi/2);

subplot(3,3,4);
stem(n,x4n);
grid on;
xlabel('n');
ylabel('x_4n');


% generate plot for part 5
x5n = cos(n*pi);

subplot(3,3,5);
stem(n,x5n);
grid on;
xlabel('n');
```

```
ylabel('x_5n');

% generate plot for part 6
x6n = cos(n*pi*3/2);

subplot(3,3,6);
stem(n,x6n);
grid on;
xlabel('n');
ylabel('x_6n');

% generate plot for part 7
x7n = cos(n*pi*7/4);

subplot(3,3,7);
stem(n,x7n);
grid on;
xlabel('n');
ylabel('x_7n');


% generate plot for part 8
x8n = cos(n*pi*15/8);

subplot(3,3,8);
stem(n,x8n);
grid on;
xlabel('n');
ylabel('x_8n');


% generate plot for part 9
x9n = cos(n*pi*2);

subplot(3,3,9);
stem(n,x9n);
grid on;
xlabel('n');
ylabel('x_9n');
```

Plots Created by the code snippet ex1c.m



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?

It can be seen that the original form of the signal get lost when we decrease the sampling frequency , in other words we lose some data when we decrease the discrete frequency

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 $y[n]= \Sigma\, x[k]\, h[n-k]$

Matlab Function conv_disc (conv_disc.m)

```
function y = conv_disc(xn, hn)

 % get initial sizes of xn, hn
 sizeX = length(xn);
 sizeH = length(hn);

 % fill rest of the array with zeros until both has same size
 xn = [xn,zeros(1,sizeH)];
 hn = [hn,zeros(1,sizeX)];

 % get the size of the output
 out_size = sizeH+sizeX-1 ;

 % generate elements of output on for loop
 for n=1:out_size
    y(n) = 0;
    for k=1:sizeX
      if( n+1 > k )
        % getting the sum of Xn(n) * Hn(n-k)
        y(n) = y(n)+xn(k)*hn(n-k+1);
      endif
    endfor
 endfor

endfunction
```

b) Using the function written in section a, convolve $x[n]= 0.5^n u(n)$ with $h[n]=u[n]$. Plot the output signal along with the two input signals.

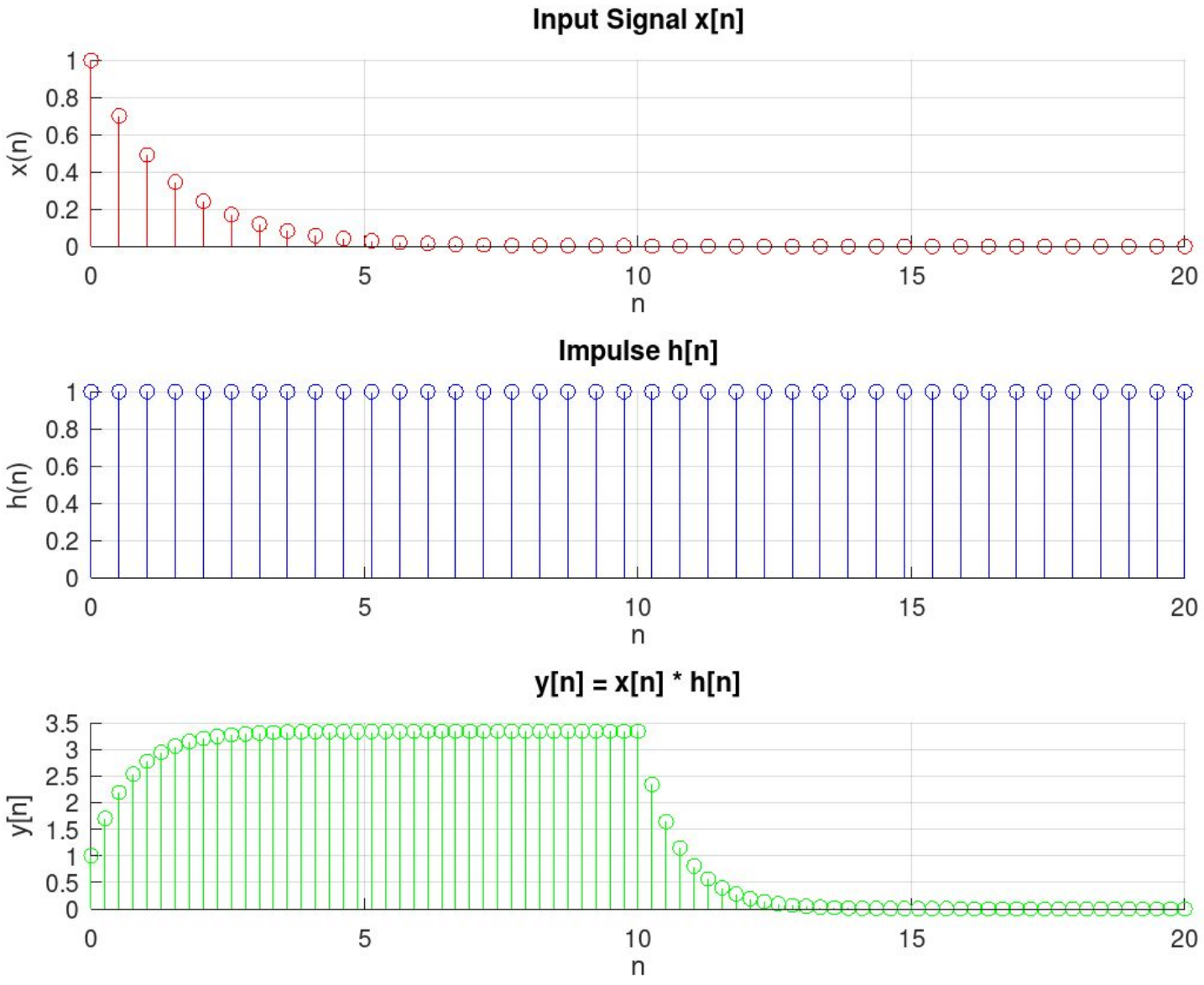
Unit Step Function ustep.m

```
function y = ustep(t,ad)
% t: length of time
% ad: advance (positive), delay (negative) factor% Write your code

% shorthand conditional statement which gives 1 or 0 based on the
condition
% this conditional statement is evaluated for each element in t
 y = (t+ad) >= 0;
Endfunction
```

Matlab Script for part b) ex2b.m

```
% get some discrete points
x = linspace(0,20,40);

% generate xn
xn = 0.5.^x.*ustep(x,0);

% generate hn
hn = ustep(x,0);
% do discrete convolution
yn = conv_disc(xn,hn);

% plot xn
subplot(3,1,1);
stem(x,xn,'r');
grid on;
title('Input Signal x[n]');
xlabel('n');
ylabel('x(n)');

% plot hn
subplot(3,1,2);
stem(x,hn,'b');
grid on;
title('Impulse h[n]');
xlabel('n');
ylabel('h(n)');

% plot y(n)
x2 = linspace(0,20,length(yn));

subplot(3,1,3);
stem(x2,yn,'g');
grid on;
title('y[n] = x[n] * h[n]');
xlabel('n');
ylabel('y[n]');
```

Plots Created by the code snippet ex2b.m



c) Consider the following two signals

I. x[n] = [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0]

II. h[n] = [2 4 8 16 32 64 0 0 0 0 0 0 0 0 0]

III. Convolve the two signals using the function written in part a. Use MATLAB conv command to verify your answer.

Matlab Script for part c) ex2c.m

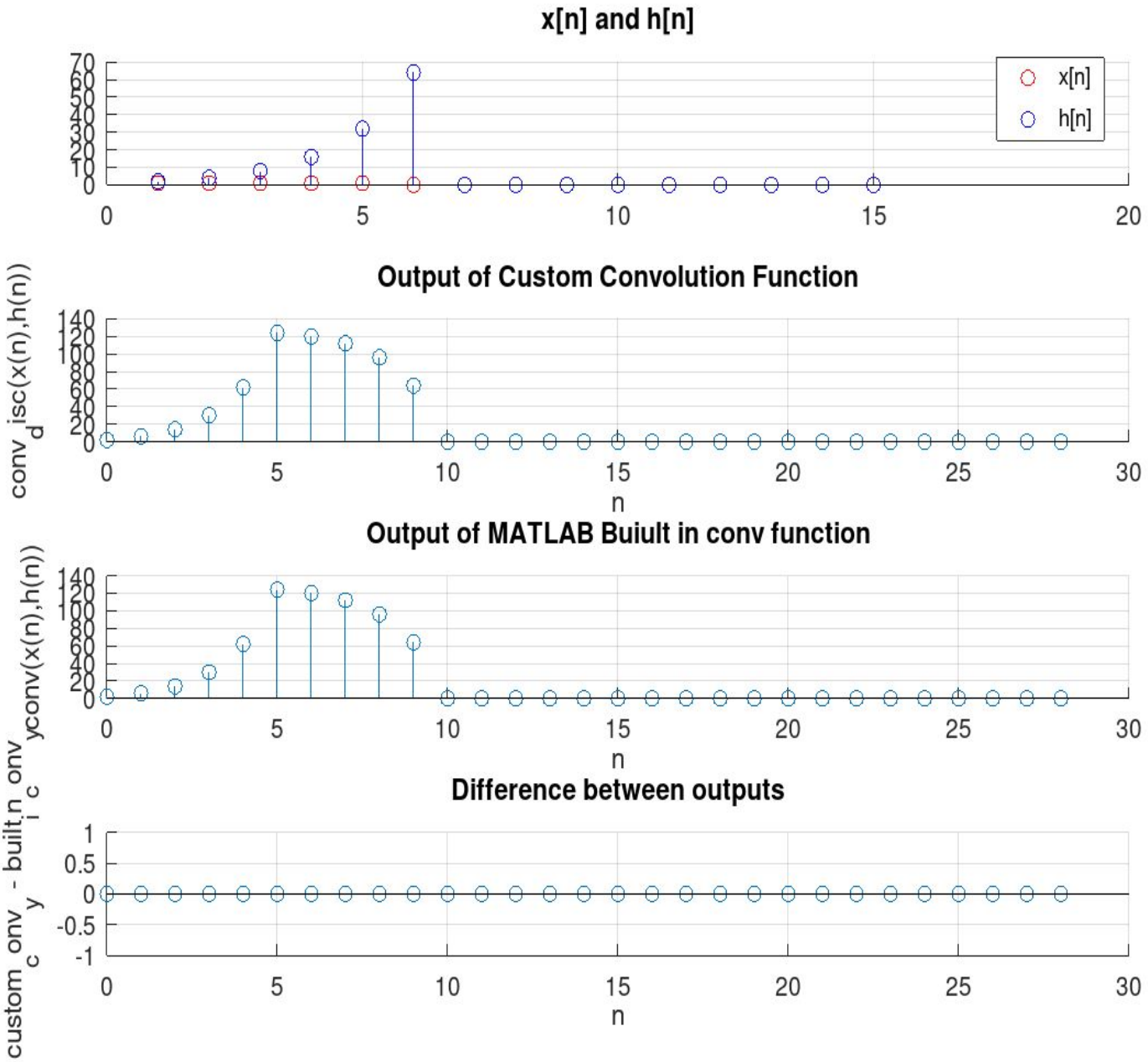
```
% initialize data
n = 1:15;
xn = [1,1,1,1,1,0,0,0,0,0,0,0,0,0,0];
hn = [2,4,8,16,32,64,0,0,0,0,0,0,0,0,0];
custom_conv_y = conv_disc(xn,hn);
built_in_conv_y = conv(xn,hn);
% check whether the two outputs are identical
if(isequal(custom_conv_y,built_in_conv_y))
display("The two functions give the same output")
endif
% graphical proof
% plot x(n) h(n)
subplot(4,1,1);
stem(n,xn,'r');
hold on;
stem(n,hn,'b');
grid on;
title('x[n] and h[n]');
legend('x[n]','h[n]')

% plot custom function output
n_1 = 0:length(custom_conv_y)-1;
subplot(4,1,2);
stem(n_1,custom_conv_y)
grid on;
title('Output of Custom Convolution Function')
ylabel('conv_disc(x(n),h(n))');
xlabel('n');

% plot Built in function output

subplot(4,1,3);
stem(n_1,built_in_conv_y)
grid on;
title('Output of MATLAB Built in conv function');
ylabel('conv(x(n),h(n))');xlabel('n');
% plot difference
```

```
subplot(4,1,4);
stem(n_1,custom_conv_y-built_in_conv_y)
grid on;
title('Difference between outputs')
ylabel('custom_conv_y - built_in_conv_y');xlabel('n')
```

Console Output of the above program

```
>> ex2c
The two functions give the same output
```

Plots Created by the code snippet [ex2c.m]



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 input 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.

bank_balance.m

```
% initial_balance will be an array something like [100,10,0,0]
function y = bank_balance(initial_balance)
  intrest_rate = 0.01; % interest rate
  y(1) = initial_balance(1)*(1+intrest_rate); % first months balance
  for i=2:length(initial_balance)
    y(i) = (y(i-1)+ initial_balance(i))* (1 + intrest_rate); % generate
balance for each month
  endfor
endfunction
```

Input x[n] : savings put into bank each month
Output y[n] : bank balance after each month

II. A merchant earns an 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.

merchant_savings.m

```
% M will be an array something like [100,10,0,0]
function y = merchant_savings(M)
  y(1) = M(1)/2; % first months savings
 % calculate next savings based on savings given and add the previous
savings
  for i = 2 : length(M)
    y(i) = y(i-1) + (M(i)/2) ;
  endfor
endfunction
```

Input x[n] : Merchant earned money each month
Output y[n] : Merchant saving after each month

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

Matlab Script for part b) ex3b.m

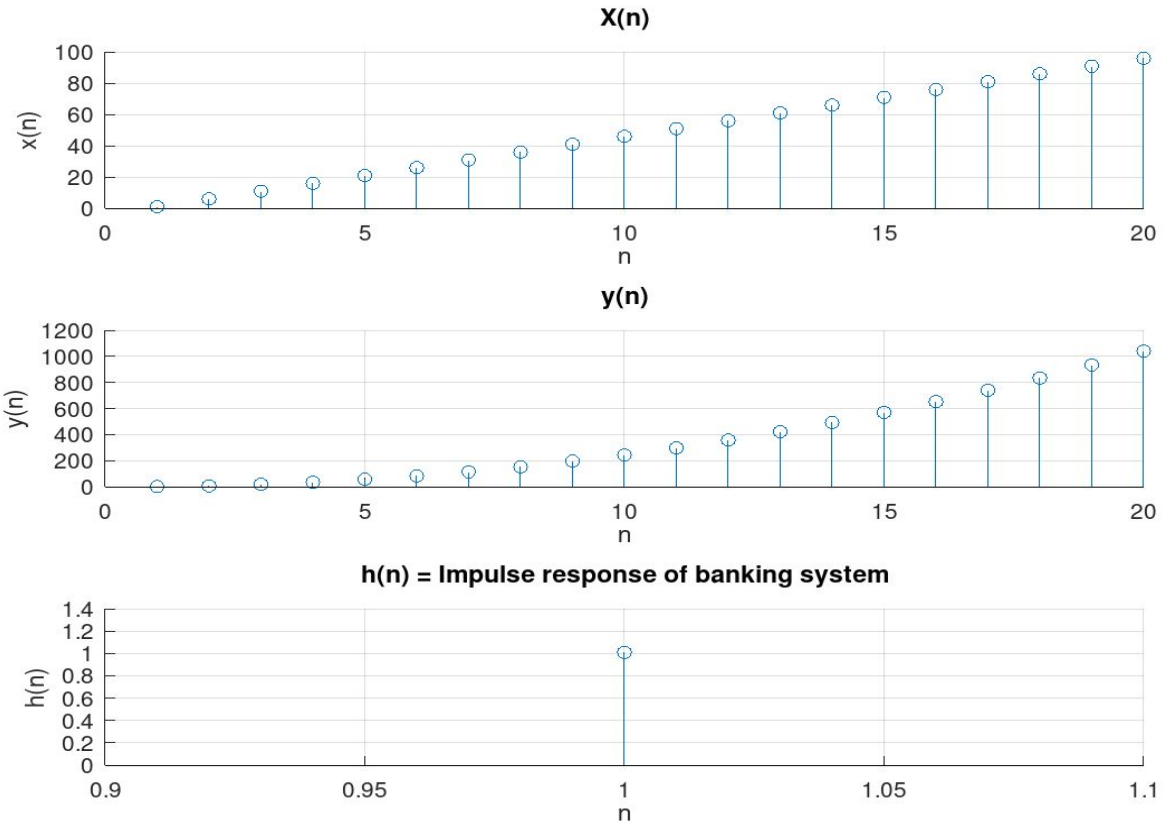
```
% for bank system
xn = 1:5:100;
yn = bank_balance(xn); % get yn

% deconv to get hn - impulse response
hn = deconv(yn,xn);
display("Impulse response of banking system");
display(hn);

% plot xn
subplot(3,1,1);
stem(xn);
grid on;
title('X(n)');xlabel('n');ylabel('x(n)');

% plot yn
subplot(3,1,2);
stem(yn);
title('y(n)');xlabel('n');ylabel('y(n)');
grid on;

% plot hn
subplot(3,1,3);
stem(hn);
title('h(n) = Impulse response of banking
system');xlabel('n');ylabel('h(n)');
grid on;

% for merchant savings

xn = 1:5:100;
yn = merchant_savings(xn); % get yn
% deconv to get hn - impulse response
hn = deconv(yn,xn);
display("Impulse response of merchant savings system");
display(hn);
```

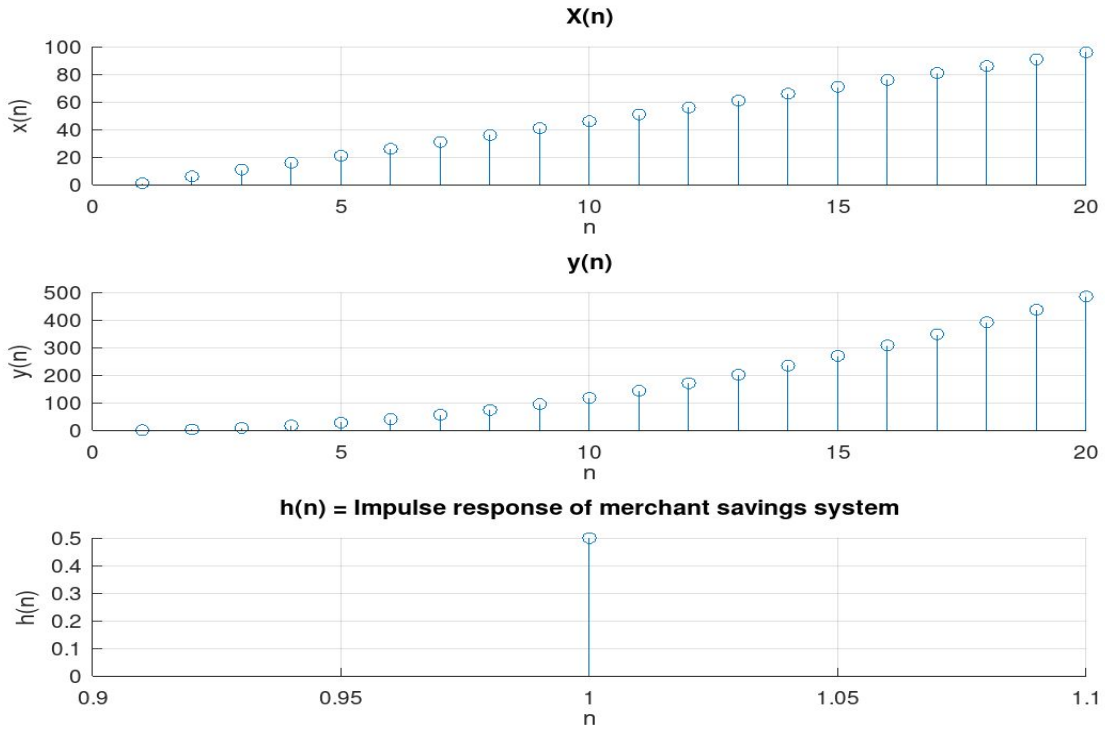
```
figure;
% plot xn
subplot(3,1,1);
stem(xn);
grid on;title('X(n)');xlabel('n');ylabel('x(n)');
% plot yn
subplot(3,1,2);
stem(yn);
title('y(n)');xlabel('n');ylabel('y(n)');grid on;
% plot hn
subplot(3,1,3);
stem(hn);
title('h(n) = Impulse response of merchant savings
system');xlabel('n');ylabel('h(n)');grid on;
```

Console Output of the above program

```
>> ex3b
Impulse response of banking system
hn =  1.0100
Impulse response of merchant savings system
hn =  0.50000
```

Plots Created by the code snippet ex3b.m





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

Both of the system's outputs depend on the previous outputs. So the output will not settle to zero in finite time.as a result we can classify both systems as **Infinite Impulse Response** Systems.

REMARKS

The matlab code and the image files can be found at,
https://github.com/irfanm96/EE387-Signal-Processing/tree/master/Lab02