EE387 – SIGNAL PROCESSING

LAB 2

WIMALASIRI KPGP

E/14/403

SEMESTER 6

11/12/2018

1) Discrete time signals

a)

```
clear all;
close all;
subplot(2,2,1)
X = 0:1:20;
B = -1.2;
Y1= [10*B.^X];
stem(Y1);
title('beta < -1');
subplot(2,2,2)
B = -0.8;
Y2= [10*B.^X];
stem(Y2);
title('-1 < beta < 0');
subplot(2,2,3)
B = 0.8;
Y3= [10*B.^X];
stem(Y3);
title('0 < beta < 1');
subplot(2,2,4)
B = 1.2;
Y4= [10*B.^X];
stem(Y4);
title('1 < beta');
```

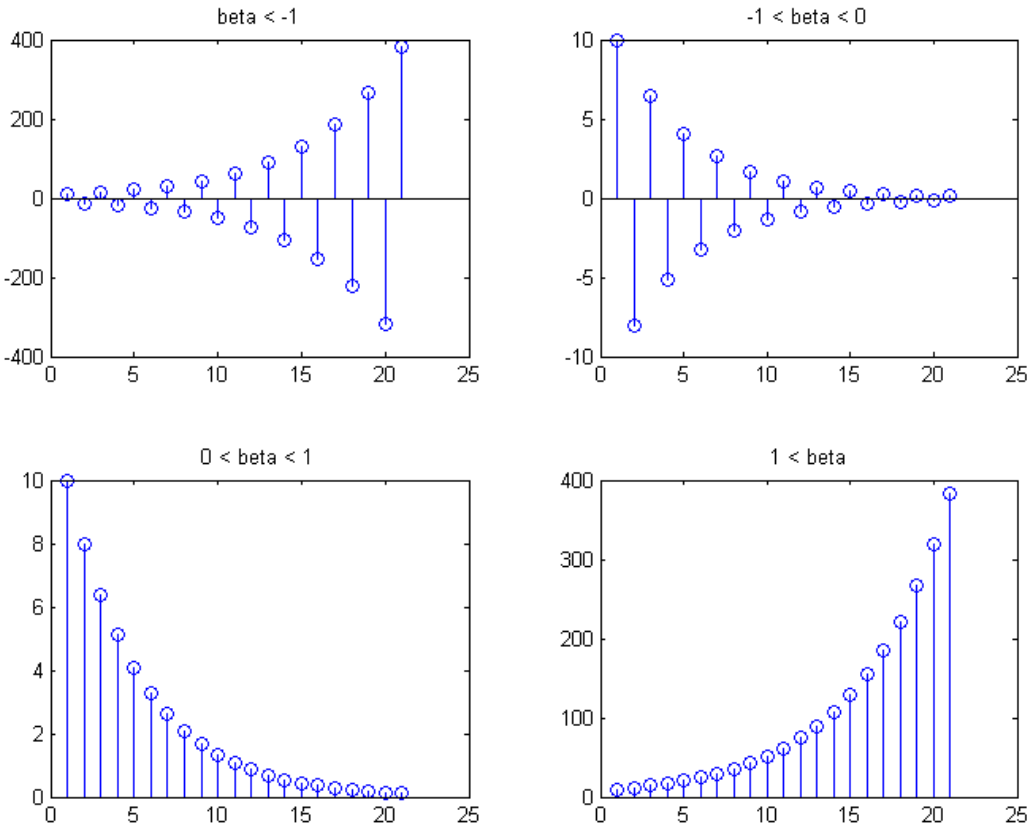
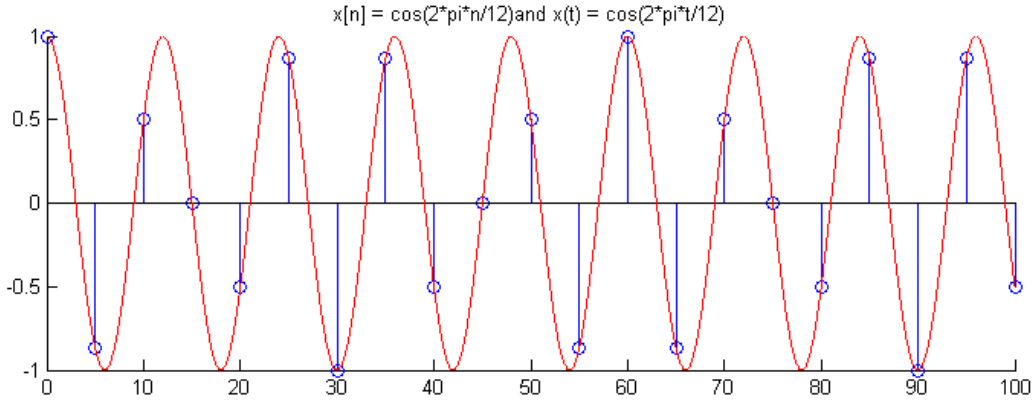
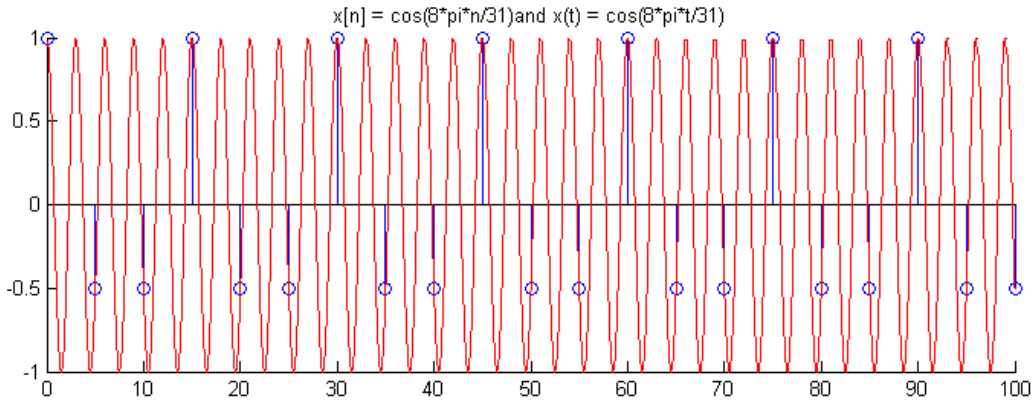


Figure 1

b)

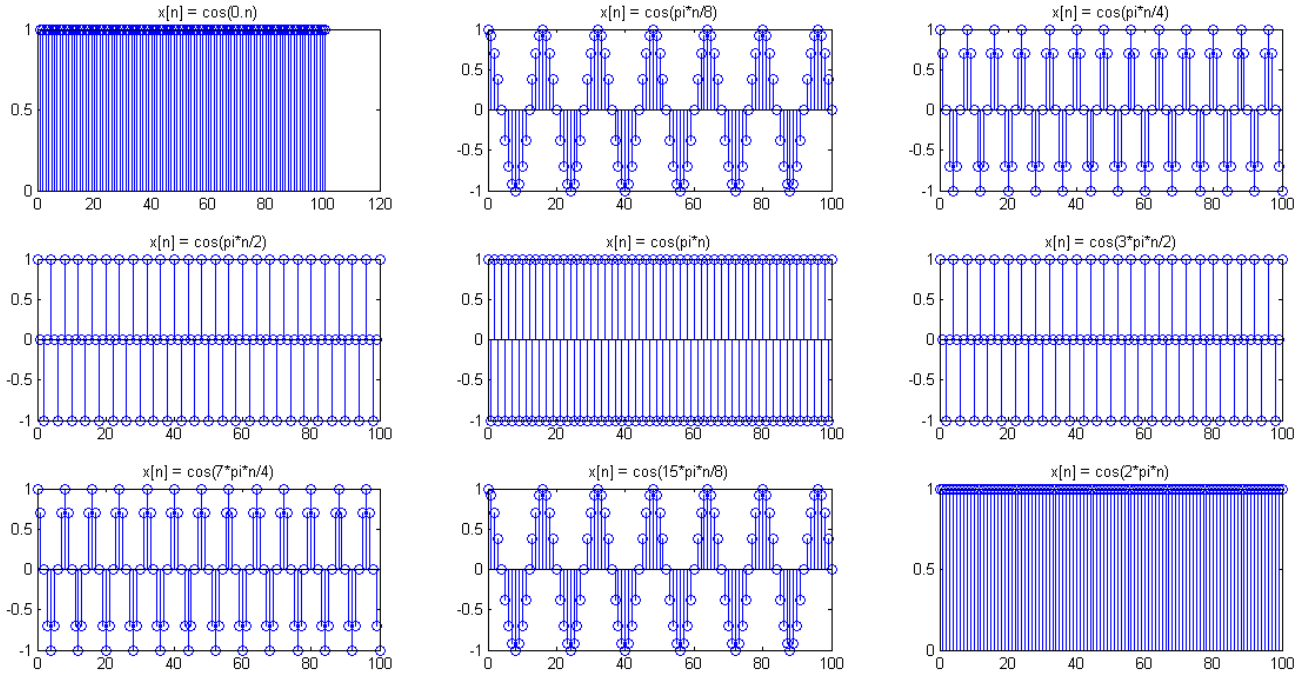
```
clear all;
close all;
subplot(2,1,1)
hold on;
Xn = 0:5:100;
Y1= [cos(2*pi*Xn/12)];
stem(Xn,Y1);
Xt = 0:0.01:100;
Y2= [cos(2*pi*Xt/12)];
plot(Xt,Y2,'r');
title('x[n] = cos(2*pi*n/12)and x(t) = cos(2*pi*t/12)');
subplot(2,1,2)
hold on;
Y3 = [cos(8*pi*Xn/12)];
stem(Xn,Y3);
Y4= [cos(8*pi*Xt/12)];
plot(Xt,Y4,'r');
title('x[n] = cos(8*pi*n/31)and x(t) = cos(8*pi*t/31)');
```





c)

```
clear all;
close all;

figure

subplot(3,3,1)
X = 0:1:100;
Y1= [cos(0*X)];
stem(Y1);
title('x[n] = cos(0.n)');

subplot(3,3,2)
Y2= [cos(pi*X/8)];
stem(X,Y2);
title('x[n] = cos(pi*n/8)');

subplot(3,3,3)
Y3= [cos(pi*X/4)];
stem(X,Y3);
title('x[n] = cos(pi*n/4)');

subplot(3,3,4)
Y4= [cos(pi*X/2)];
stem(X,Y4);
title('x[n] = cos(pi*n/2)');

subplot(3,3,5)
Y5= [cos(pi*X)];
stem(X,Y5);
title('x[n] = cos(pi*n)');

subplot(3,3,6)
Y6= [cos(3*pi*X/2)];
stem(X,Y6);
title('x[n] = cos(3*pi*n/2)');

subplot(3,3,7)
Y7= [cos(7*pi*X/4)];
stem(X,Y7);
title('x[n] = cos(7*pi*n/4)');

subplot(3,3,8)
Y8= [cos(15*pi*X/8)];
stem(X,Y8);
title('x[n] = cos(15*pi*n/8)');

subplot(3,3,9)
Y9= [cos(2*pi*X)];
stem(X,Y9);
title('x[n] = cos(2*pi*n)');
```



d) According to the plots we can identify that the pattern will repeat at 2*pi cycles

2) Discrete convolution

a)

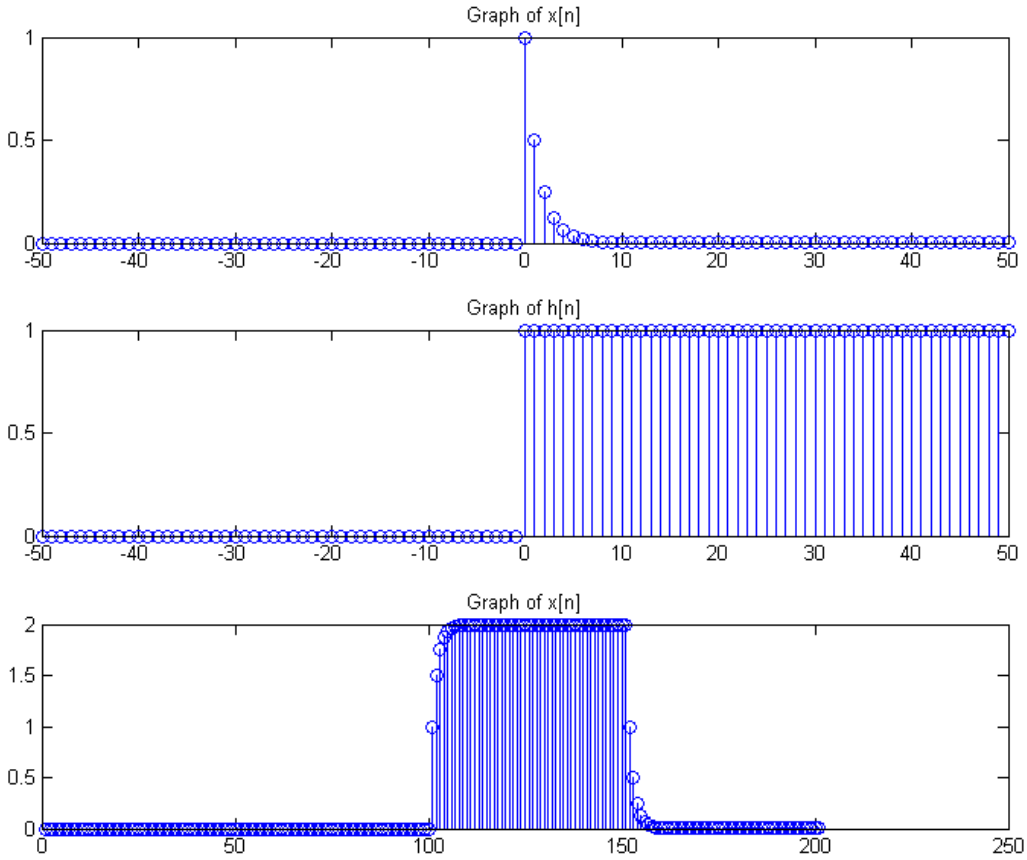
```
%part2_a
function y = convolution(x,h)
xa = length(x);
hb = length(h);
n = xa+hb-1;
y = zeros(1,n); %initial values are given as zero

for p = 0:n
    for q = 0:n
        if((p-q+1)>0 && (p-q+1)<=hb && (q+1)<=xa)
           y(p+1) = y(p+1)+ x(q+1).*h(p-q+1);
        end
    end
end
```

b)

```
clear all;
close all;

X = -50:1:50;
xn = ((0.5).^(X)).*uStep(X);
hn = uStep(X);

yn = convolution(xn,hn);

subplot(3,1,1);
stem(X,xn);
title('Graph of x[n]');

subplot(3,1,2);
stem(X,hn);
title('Graph of h[n]');

subplot(3,1,3);
stem(yn);
title('Graph of x[n]');
```

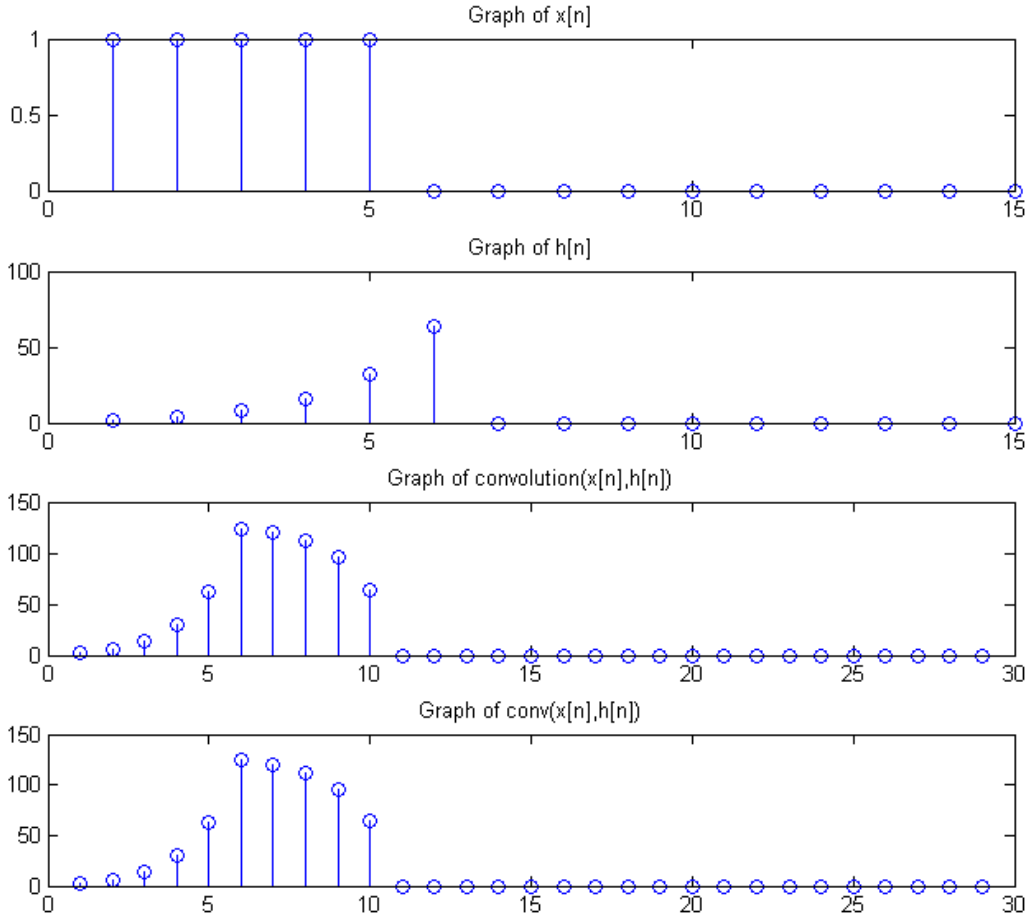
uStep function

```
function u = uStep(n)

u = [];

for i = n
    if (i>=0)
        u = [u 1];
    else
        u = [u 0];
    end
end
```



c)

```
clear all;
close all;

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];

yn = convolution(xn,hn);
yn2 = conv(xn,hn);

subplot(4,1,1);
stem(xn);
title('Graph of x[n]');

subplot(4,1,2);
stem(hn);
title('Graph of h[n]');

subplot(4,1,3);
stem(yn);
title('Graph of convolution(x[n],h[n])');

subplot(4,1,4);
stem(yn2);
title('Graph of conv(x[n],h[n])');
```


Graph of x[n]
1
0.5
0
0
5
10
15
Graph of h[n]
100
50
0
0
5
10
15
Graph of convolution(x[n],h[n])
150
100
50
0
0
5
10
15
20
25
30
Graph of conv(x[n],h[n])
150
100
50
0
0
5
10
15
20
25
30

3)

a) i.

```
function newBalance = bankBalance(balance, netSaving, month)

newBalance = [];

for i = 1:length(month)

    balance = (balance+balance*0.01+netSaving(i));
    newBalance = [newBalance balance];

end
```

ii.

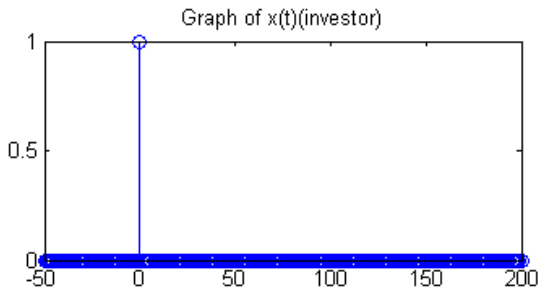
```
function newBalance = merchant(saving, money, month)

newBalance = [];

for i = 1:length(month)
    saving = saving+(money/2);
    newBalance = [newBalance saving];
end
```

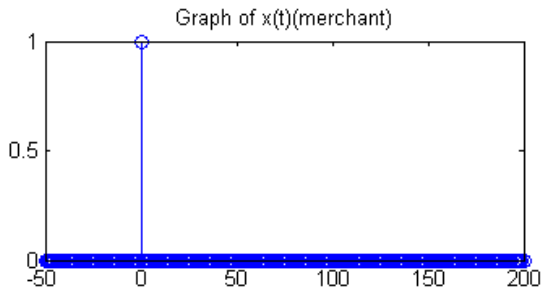
b)

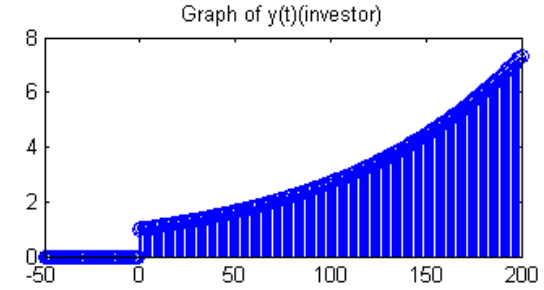
```
clear all;
close all;

month = -50:200;
y = unitImpulse(month);
investorBalance = bankBalance(0,y,month);
ht = conv(investorBalance,y);

subplot(3,2,1);
stem(month,y);
title('Graph of x(t)(investor)');

subplot(3,2,3);
stem(month,investorBalance);
title('Graph of y(t)(investor)');

subplot(3,2,5);
stem(ht);
title('Graph of h(t)(investor)');

merchantBalance = merchant(0, 100, month);
ht = conv(merchantBalance, y);

subplot(3,2,2);
stem(month,y);
title('Graph of x(t)(merchant)');

subplot(3,2,4);
stem(month,merchantBalance);
title('Graph of y(t)(merchant)');

subplot(3,2,6);
stem(ht);
title('Graph of h(t)(merchant)');
```

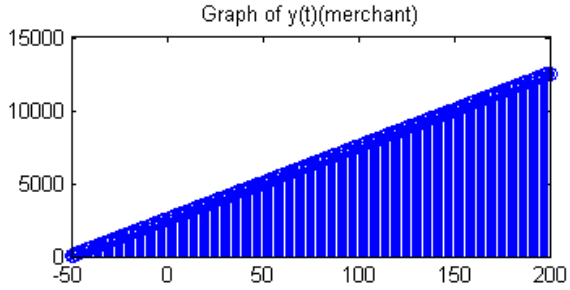
unitImpulse function

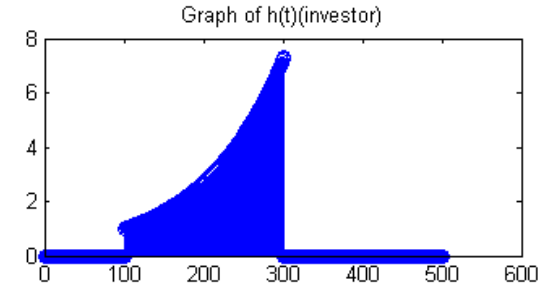
```
function y = unitImpulse(n)

y = [];
for i=n
    if(i==0)
        y = [y 1];
    else
        y = [y 0];
    end
end

end
```

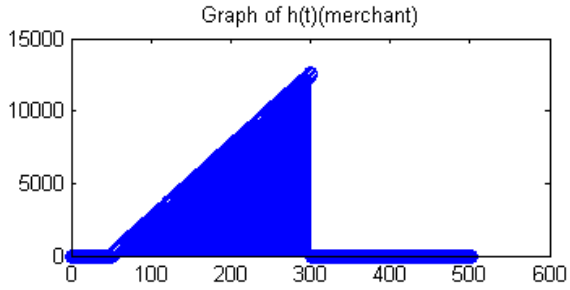












c) IIR- Infinite Impulse Response FIR – Finite Impulse Response

In the investor side function depends on both previous outputs and previous inputs. Therefore investor side is an IIR system. But in merchant side only inputs are taken into the function. Therefore, merchant side is an FIR system.