EE387 – DISCRETE TIME SIGNALS

JAYATHUNGA W.W.K.

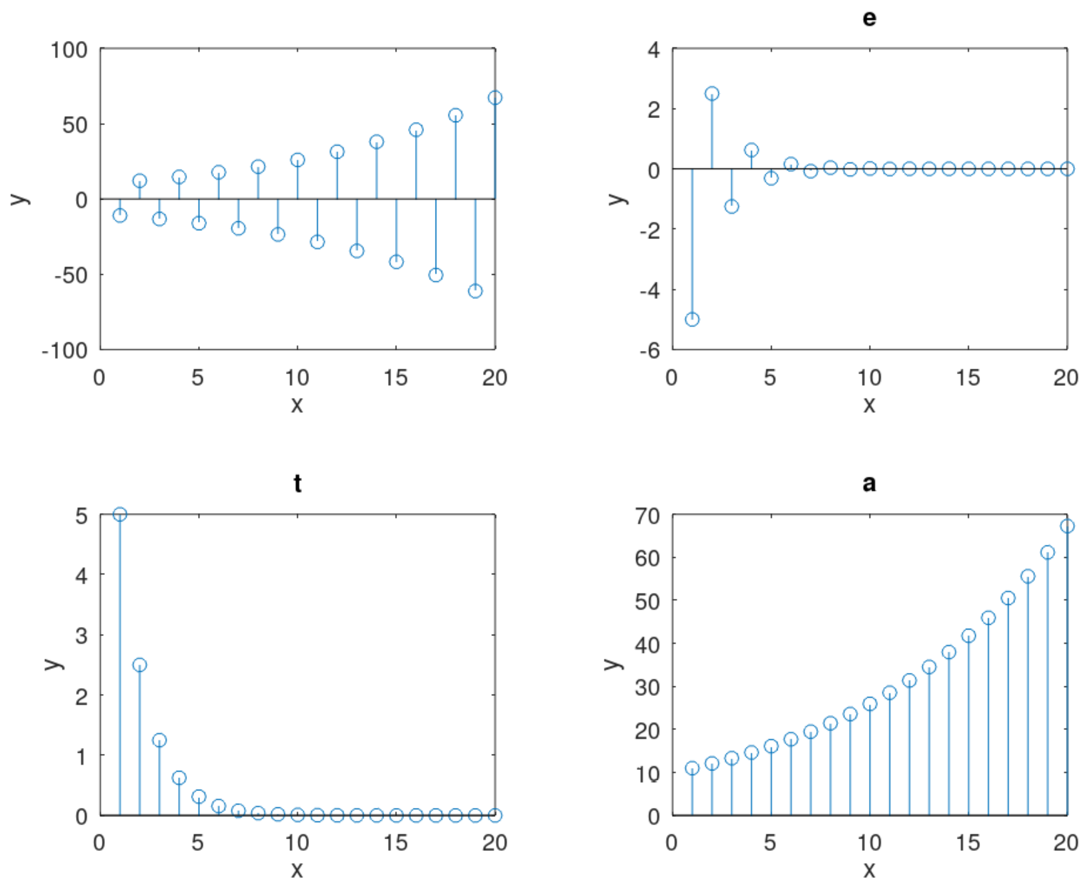
E/19/166

SEMESTER 06

30/04/2024

1. **Understanding properties of Discrete Time Sinusoidal signals**
   a. Code

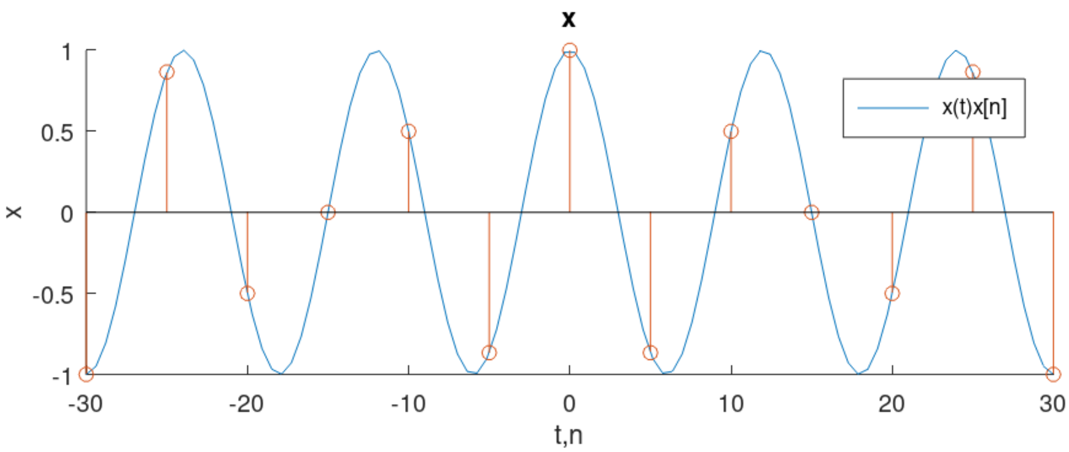
```
figure;
title('Ex 01 : A');
hold on;

% Different beta values
b = [-1.1, -0.5, 0.5, 1.1];
n = 1:20;

for idx = 1:4
  x = 10 * (b(idx).^n);
  subplot(2, 2, idx);
  stem(n, x);
  xlabel('x');
  ylabel('y');
  labels = ["\beta < -1", "-1 < \beta < 0", "0 < \beta < 1","1 < \beta"];
  title(labels(idx));
end
```

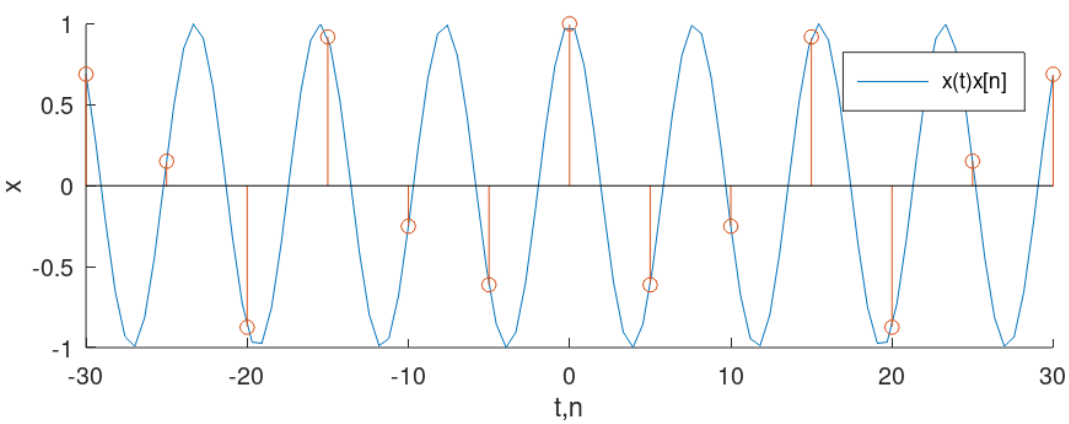


b. <u>Code</u>

```
clear all;close all;clc;
tStart=-30;
tEnd=30;
w=[pi/6, 8*pi/31];
t=linspace(tStart,tEnd,100);%CT variable
T=5;
k=(tStart/T):(tEnd/T);
n=k*T;%DT variable
figure;
title('Ex 01 : B');
hold on;
for idx=1:2
 x_t=cos(w(idx)*t);
 x_n=cos(w(idx)*n);

 subplot(2,1,idx);
```

```
hold on;
plot(t,x_t);%Plot the CT
stem(n,x_n);%Plot the DT
xlabel('t,n');
ylabel('x');
legend(["x(t)","x[n]"]);
labels=["x = coe(2 \pi t /12)","x = coe(8 \pi n /31)"];title(labels(idx));
end
```





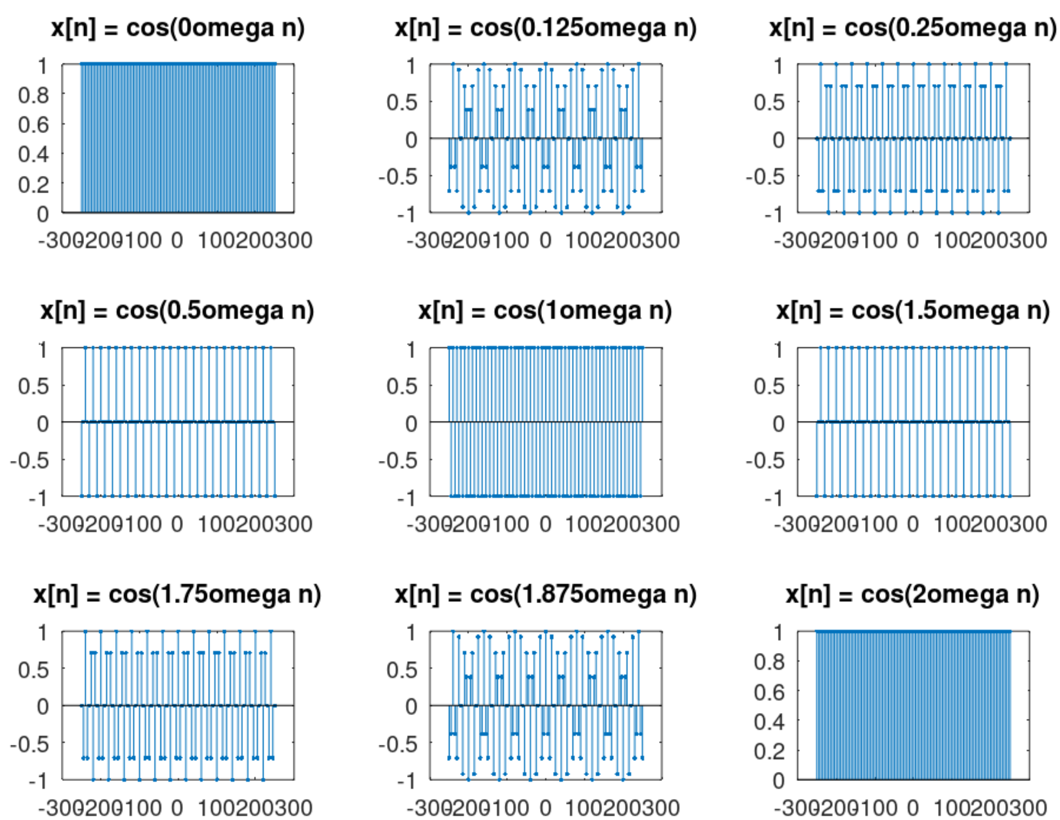
CT Signal	cos(2*pi*t / 12)	cos(8*pi*t / 31)
Theoretical Time Period	12 time units	31 / 4 time units

DT Signal	cos(2*pi*n / 12)	cos(8*pi*n / 31)
Theoretical Time Period	12	31

For both the CT and DT forms for the first signal, the observed period is the DT form's theoretical period. For the second signal, observed period of the CT form is the theoretical period of the DT form.

c. <u>Code</u>

```
clear all;
close all; clc;
tStart=-500;
tEnd=500;
w=[0,pi/8,pi/4,pi/2,pi,3*pi/2,7*pi/4,15*pi/8,2*pi];
t=linspace(tStart,tEnd,1000);%Continuoue variable
T=5;
k=(tStart/T):(tEnd/T);
n=k*T;%Diecrete variable
for idx=1:length(w)
 subplot(3,3,idx);
 stem(n,cos(w(idx).*n),'.');
 title((["x[n] = cos(",num2str(w(idx)/pi),"\omega n)"]));
end
```



d. When x[n] = cos(0 * omega * n), the waveform is a constant.
Then, the frequency of the waveform increases.
The peak frequency is at x[n] = cos(1 * omega * n)
After that, again the frequency decreases.

2. Discrete Convolution

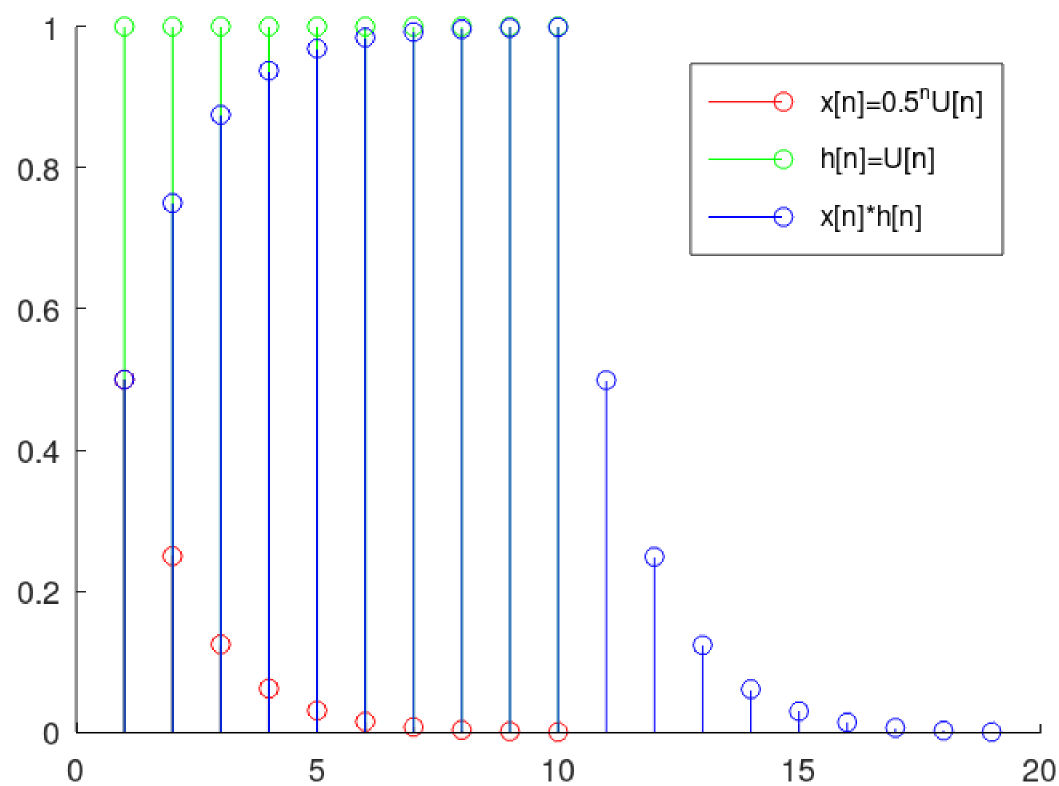
a. Code

```
clear all;
close all;
clc;

function [ y] = myConv( x1,x2)
 y=zeros(1,length(x1)+length(x2));%resulting vector
 N=length(y);

 for n=1:N
 for k=1:N
 if (k <= length(x1)) && (n-k >= 1) && (n-k <=length(x2))
 %Checking to see if the variables goes out of the finite
 %array (in which case they are zero)
 y(n)=y(n)+x1(k)*x2(n-k);
 end
 end
 end

 y=y(1,2:length(y));

% disp('DEBUG OUTPUT:')
% disp([' x1 = ',num2str(x1)]);
% disp([' x2 = ',num2str(x2)]);
% disp([' x1CONVx2 = ',num2str(y)]);

End
```

b. Code

```
pkg load symbolic; % Load the symbolic package

clear all;
close all;
clc;

function [ y] = myConv( x1,x2)
 y=zeros(1,length(x1)+length(x2));%resulting vector
```

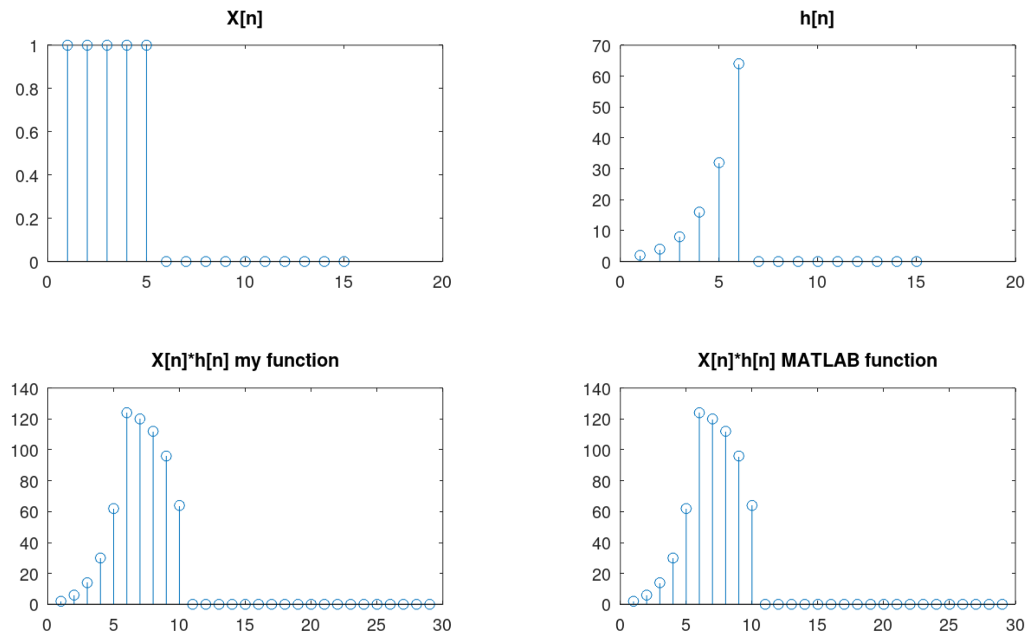
```
 N=length(y);

 for n=1:N
 for k=1:N
 if (k <= length(x1)) && (n-k >= 1) && (n-k <=length(x2))
 %Checking to see if the variables goes out of the finite
 %array (in which case they are zero)
 y(n)=y(n)+x1(k)*x2(n-k);
 end
 end
 end

 y=y(1,2:length(y));

% disp('DEBUG OUTPUT:')
% disp([' x1 = ',num2str(x1)]);
% disp([' x2 = ',num2str(x2)]);
% disp([' x1CONVx2 = ',num2str(y)]);

end


n=1:10;
x=0.5.^n .* heaviside(n);
h=heaviside(n);
xh=myConv(x,h);
figure;
hold on;
stem(n,x,'r')
stem(n,h,'g');
stem((1:length(xh)),xh,'b');
legend(["x[n]=0.5^nU[n]","h[n]=U[n]","x[n]*h[n]"]);
```



c. <u>Code</u>

```
close all;
clear all;
clc;

pkg load symbolic; % Load the symbolic package

clear all;
close all;
clc;

function [ y] = myConv( x1,x2)
 y=zeros(1,length(x1)+length(x2));%resulting vector
 N=length(y);

 for n=1:N
 for k=1:N
 if (k <= length(x1)) && (n-k >= 1) && (n-k <=length(x2))
 %Checking to see if the variables goes out of the finite
 %array (in which case they are zero)
 y(n)=y(n)+x1(k)*x2(n-k);
 end
```

```
 end
 end

 y=y(1,2:length(y));

% disp('DEBUG OUTPUT:')
% disp([' x1 = ',num2str(x1)]);
% disp([' x2 = ',num2str(x2)]);
% disp([' x1CONVx2 = ',num2str(y)]);

end

X=[1 1 1 1 1 0 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];
Xh=myConv(X,h);%My implementation
Xhh=conv(X,h);%MATLAB function

subplot(2,2,1);
stem(X);
title("X[n]");

subplot(2,2,2);
stem(h);
title("h[n]");

subplot(2,2,3);
stem(Xh);
title("X[n]*h[n] my function");

subplot(2,2,4);
stem(Xhh);
title("X[n]*h[n] MATLAB function");
```



(iv) The convolution has acted like a range sum for the h[n] sequence.

3. LTI Systems

a. (i) The input sequence is P[n], which is the monthly net savings.
The output sequence is B[n], which represents the bank balance at the end of each month.
Since the bank balance at the conclusion of the first month only consists of net savings, B[1]=P[1].
B[n] is equal to B[n-1].The new bank balance for every other month is equal to *101% + P[n] plus the prior month's bank balance plus interest and net savings.

Code

```
function [ B ] = investor(P)
%P[n], the net savinge per month is the input sequence.

%B[n], the bank balance at the end of every month is the output sequence.

B=zeros(1,length(P));


 %since the bank balance at the end of the firet month is only the net savings.

B(1)=P(1);


for m=2:length(P)
```

%the new bank balance ie the previoue bank balance, intereet for it and the net eavinge of the month.

B(m)=1.01*B(m-1)+P(m);

end

end

(ii) M[n], the monthly income, is the input list.The output sequence is S[n], the savings at the end of each month.S[1]= 0.5*M[1] because half of the first month's profits are preserved and the balance brought forward is 0.For all other months, S[n]= S[n-1] + 0.5*M[n], as the savings account is augmented by half of each month's profits.

<u>Code</u>

function [S] = merchant(M)

%M[n], monthly earnings is the input sequence.

%S[n], the savinge at the end of every month is the output sequence.

S=zeros(1,length(M));

S(1)=0.5*M(1);

%since the balance brought forward is 0 and half the first month earnings is saved.

for m=2:length(M)

S(m)=S(m-1)+0.5*M(m);

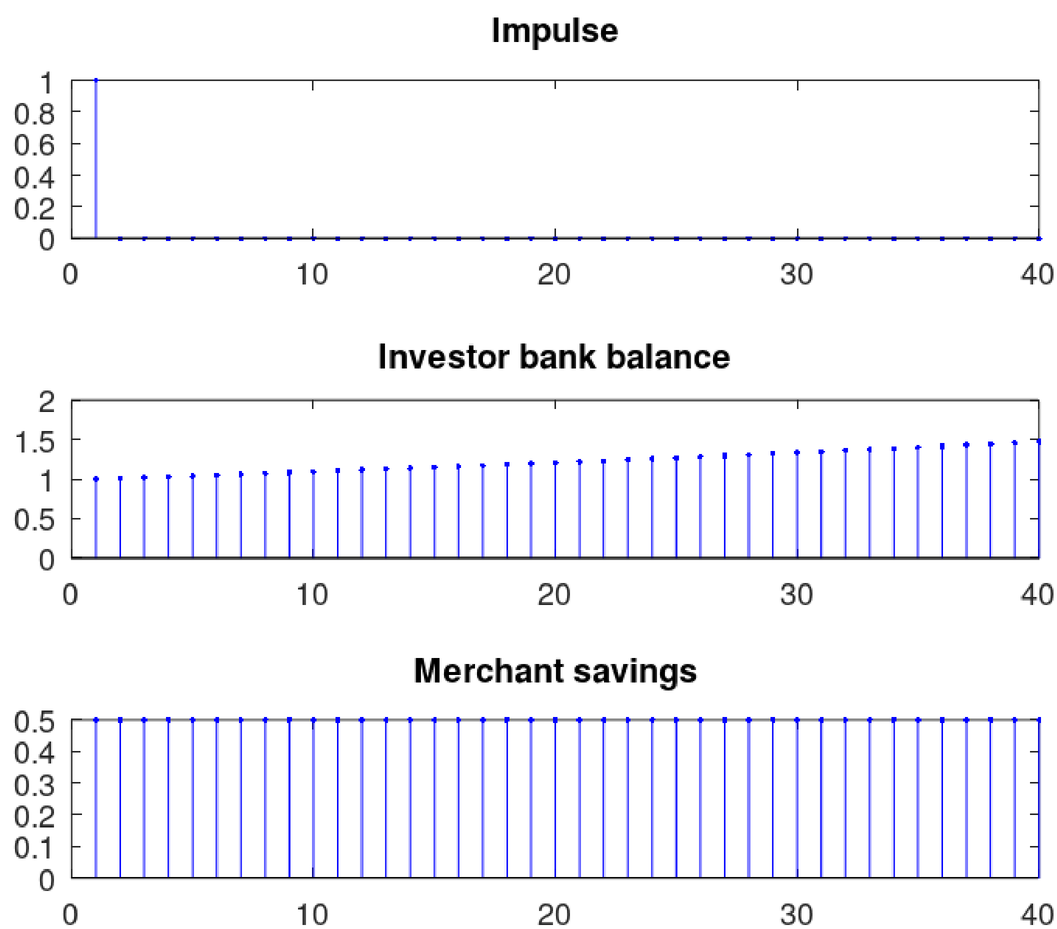
%since half of every months earninge is added to the savings.

end

end

b. <u>Code</u>

```
im=zeros(1,40);
im(1) = 1;

balance = investor(im);
savings = merchant(im);

subplot(3,1,1);
stem(im,'b.');
title("Impulse");

subplot(3,1,2);
stem(balance,'b.');
title("Investor bank balance");

subplot(3,1,3);
stem(savings,'b.');
title("Merchant savings");
```



c. The investor's bank balance is IIR, or infinite in time when the previous output is added recursively.
   FIR is the merchant's savings.