Bangladesh Army University of Science and Technology (BAUST)
CSE-4204 Digital Signal Processing
Lab Day-4

- Addition
- Subtraction
- Multiplication
- Shifting a Signal
- Reversing a Signal
- Linear Convolution of 2 signal

1. **Magnitude and Phase:** 1 - Plot the magnitude and phase of the function $X(j\omega)=j\omega 1+j\omega$ as a function of ω for $-10\pi\leq\omega\leq 10\pi$

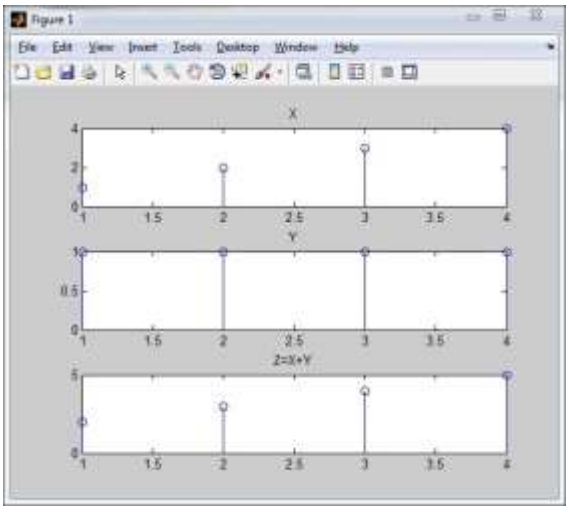
```
omega = linspace(-10*pi,10*pi,1000);
x = (j*omega)./(1+j*omega);
subplot(2,1,1),plot(omega,abs(x));
xlabel('w'),ylabel('|x(j w)|');
subplot(2,1,2),plot(omega,phase(x));
xlabel('w'),ylabel('phase(x(j w))');
```

2. Plot the magnitude and phase of the function $x(t)=imag(3-e^{(1-j2\pi)t})$ as a function of t for $-10\leq t\leq 10$

```
t = -10:001:10;
x = imag(3-exp((1-2*j*pi)*t))
subplot(2,1,1),plot(t,abs(x));
xlabel('t'),ylabel('|x(t)|');
subplot(2,1,2),plot(t,phase(x));
xlabel('t'),ylabel('phase(x(j w))');
```

3. **Signal addition**

```
x=[1 2 3 4];
subplot(3,1,1);
stem(x);
title('X');
y=[1 1 1 1];
subplot(3,1,2);
stem(y);
title('Y');
z=x+y;
subplot(3,1,3);
stem(z);
title('Z=X+Y');
```



4. Addition:

$$y(n) = x_1(n) + x_2(n)$$

```
n1 = 0:4;
x1 = [0 1 2 3 4];
n2 = -2:2;
x2 = [2 2 2 2 2];
n =min(min(n1),min(n2)):max(max(n1),max(n2)); % duration of y(n)
y1 =zeros(1,length(n));y2 = y1;              %initialization
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;    % x1 with duration of y
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;    % x2 with duration of y
y = y1+y2;                                    %sequence addition
%addition
stem(n,y)
```

5.

m-file

```
%Discrete-time addition y(n)=x1(n)+x2(n)
n1 = 0:4;
x1 = [0 1 2 3 4];
subplot(3,1,1);stem(n1,x1); title('x_1(n) signal');
xlabel('n'); ylabel('x_1(n)');

n2 = -2:2;
x2 = [2 2 2 2 2];
subplot(3,1,2);stem(n2,x2); title('x_2(n) signal');
xlabel('n'); ylabel('x_2(n)');

n = min(min(n1),min(n2)):max(max(n1),max(n2)); % duration of y(n)
y1 = zeros(1,length(n)); y2 = y1;              % initialization
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;     % x1 with duration of y
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;     % x2 with duration of y
y = y1+y2;                                     % sequence addition
% addition
subplot(3,1,3); stem(n,y); title('Addition of the discrete-time signals');
xlabel('n'); ylabel('y(n)');
```

6. **Subtraction**

```
n1=-2:1;
x=[1 2 3 4];
subplot(3,1,1);
stem(n1,x);
title('X') ;
axis([-4 4 -5 5]);
n2=0:3;
```

```
y=[1 1 1 1];
subplot(3,1,2);
stem(n2,y);
title('Y');
axis([-4 4 -5 5]);
n3 =min (min(n1) ,min( n2 ) ) : max ( max ( n1 ) , max ( n2 ) ); %
finding the duration of output signal
s1 =zeros(1,length (n3) );
s2 =s1;
s1 (find ( ( n3>=min( n1 ) ) & ( n3 <=max ( n1 ) )==1 ) )=x;
% signal x with the duration of output signal 'sub'
s2 (find ( ( n3>=min ( n2 ) ) & ( n3 <=max ( n2 ))==1) )=y;
% signal y with the duration of output signal 'sub'
sub=s1 - s2; % subtraction
subplot(3,1,3)
stem(n3,sub)
title('Z=X-Y');
axis([-4 4 -5 5]);
```

7. Multiplication

By using ‘ * ‘ (asterisk) operator we can perform multiplication of signals.

Eg:

```
n1=-2:1;
x=[1 2 3 4];
subplot(3,1,1);
stem(n1,x);
title('X') ;
axis([-4 4 -5 5]);
n2=0:3;
y=[1 1 1 1];
subplot(3,1,2);
stem(n2,y);
title('Y');
axis([-4 4 -5 5]);
n3 =min (min(n1) ,min( n2 ) ) : max ( max ( n1 ) , max ( n2 ) ); %
finding the duration of output signal (out)
s1 =zeros(1,length (n3) );
s2 =s1;
s1 (find ( ( n3>=min( n1 ) ) & ( n3 <=max ( n1 ) )==1 ) )=x;
% signal x with the duration of output signal 'mul'
s2 (find ( ( n3>=min ( n2 ) ) & ( n3 <=max ( n2 ))==1) )=y;
% signal y with the duration of output signal 'mul'
mul=s1 .* s2; % multiplication
subplot(3,1,3)
stem(n3,mul)
title('Z=X*Y');
axis([-4 4 -5 5]);
```

8. **Multiplication**

m-file

```
% implements y(n) = x1(n)*x2(n)
n1 = 0:4;
x1 = [0 1 2 3 4];
subplot(3,1,1);stem(n1,x1); title('x_1(n) signal');
xlabel('n'); ylabel('x_1(n)');

n2 = -2:2;
x2 = [2 2 2 2 2];
subplot(3,1,2);stem(n2,x2); title('x_2(n) signal');
xlabel('n'); ylabel('x_2(n)');

n = min(min(n1),min(n2)):max(max(n1),max(n2)); % duration of y(n)
y1 = zeros(1,length(n)); y2 = y1;              % initialization
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;     % x1 with duration of y
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;     % x2 with duration of y
y = y1 .* y2;                                  % sequence multiplication
subplot(3,1,3); stem(n,y); title('Multiplication of the discrete-time
signals');
xlabel('n'); ylabel('y(n)');
```

9. Shifting a Signal

```
n1=input('Enter the amount to be delayed');
n2=input('Enter the amount to be advanced');
n=-2:2;
x=[-2 3 0 1 5];
subplot(3,1,1);
stem(n,x);
title('Signal x(n)');
m=n+n1;
y=x;
subplot(3,1,2);
stem(m,y);
title('Delayed signal x(n-n1)');
t=n-n2;
z=x;
subplot(3,1,3);
stem(t,z);
title('Advanced signal x(n+n2)');
```

10. **Shifting operation**

```
%Shiftinganon-function Discrete-time signal
n = 0:8;
x = [0 1 5 2 1 36 4 5];
subplot(2,1,1);stem(n,x);title('x(n) signal');
xlabel('n');ylabel('x(n)');

m=n+2; y=x;
subplot(2,1,2);stem(m,y);title('y(n)=x(n-2) signal');
xlabel('n');ylabel('y(n)');
```

11. Folding

m-file

```
%Folding a Discrete-time signal
n = 0:8;
x = [0 0 1 2 3 4 5 4 3];
subplot(2,1,1);stem(n,x); title('x(n) signal');
xlabel('n'); ylabel('x(n)');

m=-fliplr(n); y=fliplr(x);
subplot(2,1,2);stem(m,y); title('y(n)=x(-n)
signal');
xlabel('n'); ylabel('y(n)');
```

12. Reversing a Signal

Syntax:

- fliplr(a) : if a is **row vector** it returns a vector with the same size of a but with reversed order.
  if a is **column vector** it flips the elements one column to the other.

```
n=-1:2;
x=[3 -1 0 -4];
subplot(2,1,1)
stem(n,x);
axis([-3 3 -5 5]);
title('Signal x(n)');
c=fliplr(x);
y=fliplr(-n);
subplot(2,1,2);
stem(y,c);
axis([-3 3 -5 5]);
title('Reversed Signal x(-n)') ;
```

13. Linear Convolution of Signals

Linear convolution between signals can be easily performed in MATLAB using **conv()** function. I hope you are familiar with the linear convolution of 2 signals.

Syntax:

1. conv (a,b)- Convolves the vectors a and b.

```
p=input('Enter the limit for x');
q=input('Enter the limit for y');
x=input('Enter the elements for x');
y=input('Enter the elements for y');
n1=0:p ;
n2=0:q;
subplot(3,1,1);
stem(n1,x);
title('Signal - x(n)');
subplot(3,1,2);
stem(n2,y);
title('Signal - h(n)');
z=conv(x,y);
t=length(n1)+length(n2)-1;
s=0:t-1;
subplot(3,1,3);
```

```
stem(s,z);
title('Output - y(n)');
```