Reference codes for lab 6

dfs function

```
function [Xk]=dfs(xn,N)
% Compute Discrete Fourier Series Co-efficients
% [Xk]=DFS co-efficients array over 0<=k<=N-1
% xn= one of period of the periodic sequence
% N= fundamental period
n=[0:1:N-1]
k=[0:1:N-1]
WN=exp(-j*2*pi/N);
nk=n'*k
WNnk=WN.^nk
Xk=xn*WNnk
end
```

idfs function

```
function [Xk]=idfs(Xk,N)
% Compute Inverse Discrete Fourier Series Co-efficients
% [xn]=one of period of the periodic sequence
% [Xk]=DFS co-efficients error
% N= fundamental period
n=[0:1:N-1]
k=[0:1:N-1]
WN=exp(-j*2*pi/N);
nk=n'*k;
WNnk=WN.^(-nk);
Xk=(Xk*WNnk)/N
end
```

1. Now, for example 1, save the above "dfs" function written in editor and saved as the name of the function (here, it's "dfs") and then call them in command window as x(n) = [0 1 2 3] (because rest of the samples are just a copy of this fundamental values) and N = 4. Then, write the command "xk = dfs(xn,N) and it will provide the DFT of the given periodic signal.
2. We can also find one period of signal using the DFS coefficients. First, define "N", and "xk coefficients" in a matrix form. Then write the command "xn = idfs(xk, N)."

Codes for example 2:

```
L=5;
N=20;
```

```
n=[-N:N+1];
x1=[ones(1,L), zeros(1,N-L+1)];
x=[x1 x1]
subplot(221)
stem(n,x)
title('Signal')
% find the DFS with period N=20 and duty cycle =5
k=[-N/2:N/2]
xn=[ones(1,L), zeros(1,N-L)]
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)])
subplot(222)
stem(k,magXk)
title('DFS with N=20, L=5')
% find the DFS with period N=50 and duty cycle=5
N=50
L=5
k=[-N/2:N/2]
xn=[ones(1,L), zeros(1,N-L)]
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)])
subplot(223)
stem(k,magXk)
title('DFS with N=50, L=5')
% find the DFS with period N=100 and duty cycle=5
N=200
L=5
k=[-N/2:N/2]
xn=[ones(1,L), zeros(1,N-L)]
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)])
subplot(224)
stem(k,magXk)
title('DFS with N=200, L=5')
```

dft function

```
function [Xk]=dft(xn,N)
% Compute Discrete Fourier Series Co-efficients
% [Xk]=DFS co-efficients array over 0<=k<=N-1
% xn= one of period of the periodic sequence
% N= fundamental period
n=[0:1:N-1]
k=[0:1:N-1]
WN=exp(-j*2*pi/N);
```

```
nk=n'*k
WNnk=WN.^nk
Xk=xn*WNnk
end
```

idft function

```
function [Xk]=idft(Xk,N)
% Compute Inverse Discrete Fourier Series Co-efficients
% [xn]=one of period of the periodic sequence
% [Xk]=DFS co-efficients error
% N= fundamental period
n=[0:1:N-1]
k=[0:1:N-1]
WN=exp(-j*2*pi/N);
nk=n'*k;
WNnk=WN.^(-nk);
Xk=(Xk*WNnk)/N
end
```

save these two functions properly using the file name same as the function name.

then, in separate editor file, write the following codes for example 3.

```
w=0:pi/200:2*pi
H=1+exp(-j*w)+exp(-j*2*w)+exp(-j*3*w)
subplot(221)
plot(w,abs(H))
title('Magnitude of DTFT of x(n)')
axis([0 6 0 4])
% computer the 4-point DFT
x=[1 1 1 1]
N=4
n=0:1:(N-1)
X4=dft(x,4)
subplot(222)
stem(n,abs(X4))
title('Magnitude of 4-point DFT')
axis([0 3 0 4])
% compute the 16-point DFT
N=16
x=[ones(1,4) zeros(1,12)] % Note that x has been zero padded
n=0:1:(N-1)
X8=dft(x,N)
subplot(223)
```

```
stem(n,abs(X8))
hold on
plot(n,abs(X8),'-.k')
title('Magnitude of 16-point DFT')
axis([0 16 0 4])
% compute 32 point DFT
N=32
x=[ones(1,4) zeros(1,28)] % Note that x has been zero padded
n=0:1:(N-1)
X8=dft(x,N)
subplot(224)
stem(n,abs(X8))
hold on
plot(n,abs(X8),'-.k')
title('Magnitude of 32-point DFT')
axis([0 32 0 4])
```

Part B

A. FFT computation

A1. Take the value of sample as [0:29]. Then, define the signal "x = cos(2*pi*n/10). Here, we have used the format 2*pi*n*(1/T) as 10 samples per period is asked to be taken.

A2. Define 3 different values of "N", say N1 = 64, N2 = 128 and N3 = 256 (take it as you wish). Then, you can use these values to find 3 different "X" for the predefined "x" using the following command format: X1 = abs(fft(x,N1)). Use the same format to find X2 and X3.

A3. To normalize the frequency scale, use command like the following for each of the three "N" range.

F1 = [0 : N1-1]/N1

In similar way, find F2 and F3.

A4. In the same figure, plot X1 vs F1, X2 vs F2 and X3 vs F3 using "subplot.

B. Define the values of "n" just like the previous. And choose a larger value for "N" and then used previously. Use following command to define three signals as per question:

```
x1 = cos(2*pi*n/10); % 3 periods
x2 = [x1 x1]; % 6 periods
x3 = [x1 x1 x1]; % 9 periods
```

Now, find the FFT X1, X2 and X3 just like the previous cases. Take "F = `[0:N-1]/N`".

Notice, we have taken only one "F" instead of three now. Why? ☺

C. Use the following command as reference for part "C"

```
n = [0:149];
x1 = cos(2*pi*n/10);
N = 2048;
X = abs(fft(x1,N));
X = fftshift(X);
F = [-N/2:N/2-1]/N;
plot(F,X),
xlabel('frequency / f s')
```