TUTORIAL 3

Objectives:

Frequency domain representation of discrete-time signals – DFS, DTFT, DFT

Note: Students need to revise the theory covered in this tutorial
Write the code in the observation book (Calculation not required)

Frequency domain response

1. Write a function that can synthesise a waveform in the form

$$x(t) = \mathrm{Re}\left\{ \sum_{k=1}^{N} X_k e^{j2\pi k f_o t} \right\}$$ where f_o is the fundamental frequency.

For f_o = 25Hz, $X_k = j4/k\pi$ for k odd and 0 for k even, plot $x(t)$ for N=5,10 and 25. Explain what happens when $N \to \infty$. repeat the synthesis with f_o=1kHz and listen to the cases N=1,2,3,4,5,10. Ensure that the sampling frequency fs in sound(x,fs) is high to prevent aliasing.

```
clear; clc; clf;
T = 0.12; sum = 0; f0 = 25; fs = 1000;
%Change f0 and N values
N=25;
t = 0; i = 1;
while t < T
    sum = 0;
    k = 1;
    while k <= N
        X = j*4/(k*pi);
        sum = sum+X*exp(j*2*pi*k*f0*t);
        k=k+2;
    end;
    x(i)=real(sum);
    t=t+1/fs;
    i=i+1;
end;
t= 0:1/fs:T-1/fs;
plot(t,x)
title('Wave form for N=25')
```

2. Find the DFS coefficients of the signal $x[n]=1+\sin(\pi n/12 + 3\pi/8)$

```
clear; clc;
n= 0:23; phi=3*pi/8;
x=1+sin(pi*n/12 + phi);
X=fft(x,24)/24;
```

3. For the LTI systems described by the following difference equations, generate its frequency response. Comment on the type of response.
   i. *y[n] = 0.5x[n] + 0.5x[n-1]*
   ii. *y[n] = 0.9 y[n-1] + x[n]*
   iii. *y[n]-0.3695y[n-1]+0.1958y[n-2]=0.2066x[n]+0.4131x[n-1] +0.2066x[n-2]*

   Sample Solution (iii)

```
  % Frequency domain response of
  % difference equations
  % y[n]-0.3695y[n-1]+0.1958y[n-2] =
  % 0.2066x[n]+0.4131x[n-1]+
  % 0.2066x[n-2]

b1 = [0.2066 0.4131 0.2066];
a1 = [1 -0.3695 0.1958];
    freqz(b1,a1,64);
    title(' Frequency response')
```

Write the programme for the remaining three bits (i and ii) and plot the response

4. Determine and plot the real and imaginary parts and the magnitude and phase spectra of the following DTFT for various values of r and θ.

$$G(e^{j\omega}) = \frac{1}{1 - 2r\cos(\theta)e^{-j\omega} + r^2 e^{-j2\omega}} \qquad 0 < r < 1$$

```
clear; clc;
theta=60*pi/180;
r=0.5
w= -pi:pi/100:pi;
den = 1 -2*r*cos(theta)*exp(-j*w) +r^2*exp(-j*2*w);
G=1./den;
subplot (221), plot(w, real(G));
title('Real part')
subplot (222), plot(w, imag(G));
title('Imaginary part')
subplot (223), plot(w, abs(G));
title('Magnitude')
subplot (224), plot(w, angle(G));
title('Phase')
```

5. Compute and plot the DTFT of the following sequence and observe the properties
   $s[n] = A\cos(2\pi f_0 n + \phi)$ Try for f_o=100Hz, $\phi=\pi/6$ and different lengths of sequence.

```
clear; clc; clf;
A = 1; f0 = 100; phi = pi/6; fs = 2000;
N= input('Enter the  length of sequence:');
```

```
n = 0: 1/fs : (N-1)/fs;
sig = A * cos (2*pi*f0*n + phi);
w_axis = linspace(-1, 1, 1024);
time_axis=0:length(sig)-1;
F=fftshift(fft(sig,1024));
sig_dtft=abs(F);
subplot(2,1,1), stem(time_axis,sig);
title('Signal');
xlabel('Index');
subplot(2,1,2), plot(w_axis,sig_dtft);
title('DTFT Magnitude');
xlabel('Digital Frequency');
```

6. *Three domains:* Relation between location of poles and zeroes in z plane, impulse response and frequency response.
   - **i.** **$y(n) = 0.77y(n-1)+x(n)+x(n-1)$**
   - **ii.** **$y(n)=0.77y(n-1)+0.77x(n)-x(n-1)$**
   - **iii.** **$H(z)=1-z^{-1}/1+0.77z^{-1}$**
   - **iv.** **$H(z)=1-z^{-1}+z^{-2}-z^{-3}+z^{-4}-z^{-5}$**
   - **v.** **$y(n)=x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)+x(n-5)$**
   - **vi.** **$H(z)=3-3z^{-1}$**

```
%solution for 6(i)
clear; clc; clf;
num=[1 1]; den=[1 -0.77];
[z,p,k]=tf2zp(num,den);
% pole-zero plot
figure(1);
zplane(z,p);
title('Pole-zero plot');
% impulse response plot
figure(2);
impz(num,den,20);
title('Impulse response plot');
% Frequency response plot
figure(3);
freqz(num,den,64);
title('Frequency response plot');
```

Write the programme for the remaining three bits (ii, iii, iv, v, and vi) and plot the response

7. Compute and plot the DFT of the following sequences and observe their properties
   - i. Unit impulse signal; *xi = {1 0 0 0 0 0 0 0}*
   - ii. All ones; *x1 = {1 1 1 1 1 1 1 1}*
   - iii. Three point boxcar; *xb = {1 1 1 0 0 0 0 0}*
   - iv. Symmetric boxcar; *xbsy = {1 1 0 0 0 0 0 1}*

```
%solution for 7(iii)
N = 8; nn = 0:(N-1); kk = nn;
xb = [1 1 1 0 0 0 0 0];
Xb = fft(xb,N);

subplot(221), stem(nn,xb);
title(' x(n) '); xlabel(' Index (n) ');
axis([0 7 0 1]);
subplot(222), stem(kk,real(Xb));
title(' Real part of DFT ');
xlabel(' Index (k) ');
axis([0 7 -1 4]);
subplot(224), stem(kk,imag(Xb));
title(' Imag part of DFT ');
xlabel(' Index (k) ');
axis([0 7 -2 2]);
```

Write the programme for the remaining three bits (i, ii, and iv) and document the result

8. Generate a real-valued test signal *v[n]* using the MATLAB function rand of length *N=15 or 16*. Compute DFT of *v[n]* to get *V[k]* and then try the following for even and odd lengths

   a) Compute the even and odd parts of *v[n]*,

   ```
   ve[n] = 0.5 [v(n) + v[-n mod N] ]
   v(-n mod N) = [v(1), v(N:-1:2)]
   vo[n] = 0.5 [v[n] - v[-n mod N] ]
   ```

   b) Compare DFT $v_e[n]$ with *Re V[k]* and DFT $v_o[n]$ with *Im V[k]*

```
clear;clc;clf;
N=15;nn=0:(N-1);kk=nn;
v=rand(1,N);
V=fft(v,N);
vfold=[v(1) v(N:-1:2)];
veven=0.5*(v+vfold);
vodd=0.5*(v-vfold);
Veven=fft(veven,N);
Vodd=fft(vodd,N);

subplot(321), stem(nn,veven);
title('veven(n)');
axis([0 N-1 0 1]);
subplot(322), stem(nn,vodd);
title('vodd(n)');
axis([0 N-1 -1 1]);
subplot(323), stem(kk,real(V));
title('ReV(k)');
axis([0 N-1 -2 10]);
subplot(324), stem(kk,Veven);
title('Ve(k)');
axis([0 N-1 -2 10]);
subplot(325), stem(kk,imag(V));
```

```
title('ImV(k)');
axis([0 N-1 -2 2]);
subplot(326), stem(kk,imag(Vodd));
title('Im(Vo(k))');
axis([0 N-1 -2 2]);
```

9. Generate a signal *s[n]* with three sinusoidal components at 50, 120 and 240Hz corrupted by AWGN. Plot the spectrum and identify the signal components.

```
% identification of sinusoids in noise
fs=2000;
t = (0:199)/fs;
s = sin(2*pi*50.*t) + sin(2*pi*120.*t) +sin(2*pi*240.*t);

awgn = (0.5*randn(1,200)+.25);   % N(0.25, 0.25)
sn = s+awgn;
subplot(211), plot(t,sn);
title(' Sinusoid with noise'); grid;

Sn = fft(sn,200);
f = 0:10:990;
sfmag = abs(Sn);
subplot(212), plot(f,sfmag(1:100));
title(' Spectral estimation'); grid;
```