Tutorials - 3

Objective - 3

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

Frequency Domain Response

Q1. Write a function that can synthesise a waveform in the form.

$$x(t) = Re\left\{\sum_{k=1}^{N} X_k e^{j2\pi k f_0 t}\right\}$$ f_0 = fundamental frequency

For $f_0 = 25\,Hz$, $X_k = \frac{j4}{k\pi}$ for k odd.

0 for k even.

Plot x(t) for N = 5, 10 and 25. Explain what happens when $N \to \infty$. Repeat the synthesis with $f_0 = 1\,kHz$ and listen to the cases N = 1, 2, 3, 4, 5, 10. Ensure that the sampling frequency f_s in sound (x, fs) is high to prevent aliasing.

→
```
clear; clc; clf;
T = 0.12; sum = 0; fo = 25; fs = 1000;
% change fo 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 * fo * t)
    k = k + 2;
end;
x(j) = real (sum);
t = t+1 / fs;
j = j+1
end;
t = 0: 1/fs : T-1 /fs ;
plot (t, x);
title ('Wave form for N = 25')
```

Q2. 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;
```

Q3. For the LTI systems described; generate its frequency response

- $y[n] - 0.3695\, y[n-1] + 0.1958\, y[n-2] = 0.2066\, x[n] + 0.4131\, x[n-1] + 0.2066\, x[n-2].$

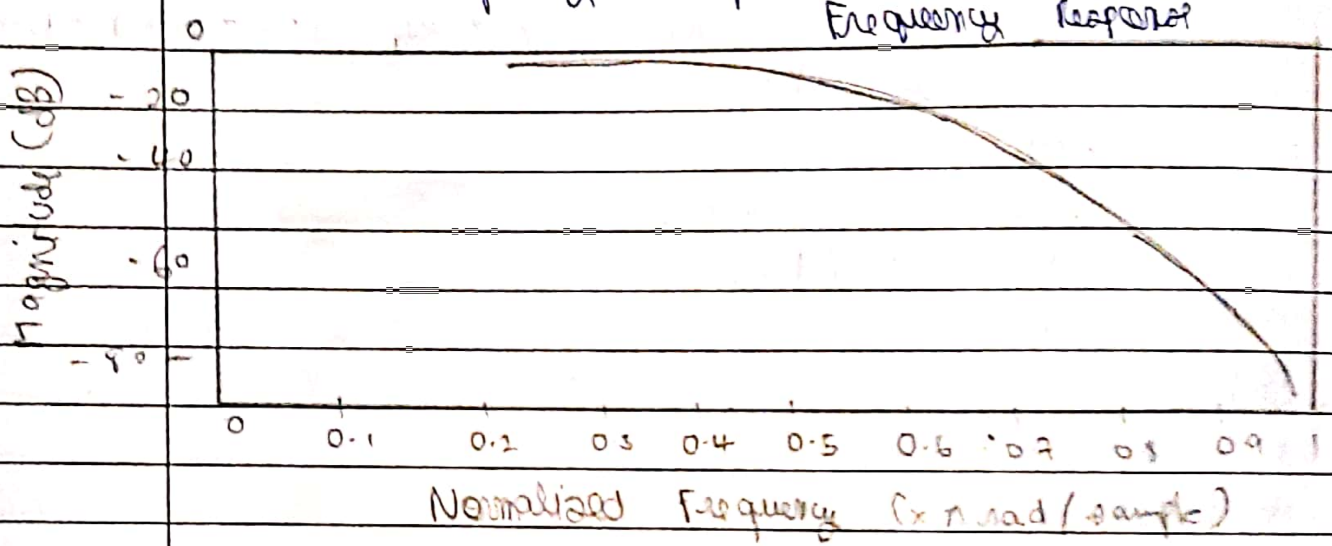
- $y[n] = 0.9\, y[n-1] + x[n]$
- $y[n] = 0.5x[n] + 0.5x[n-1].$

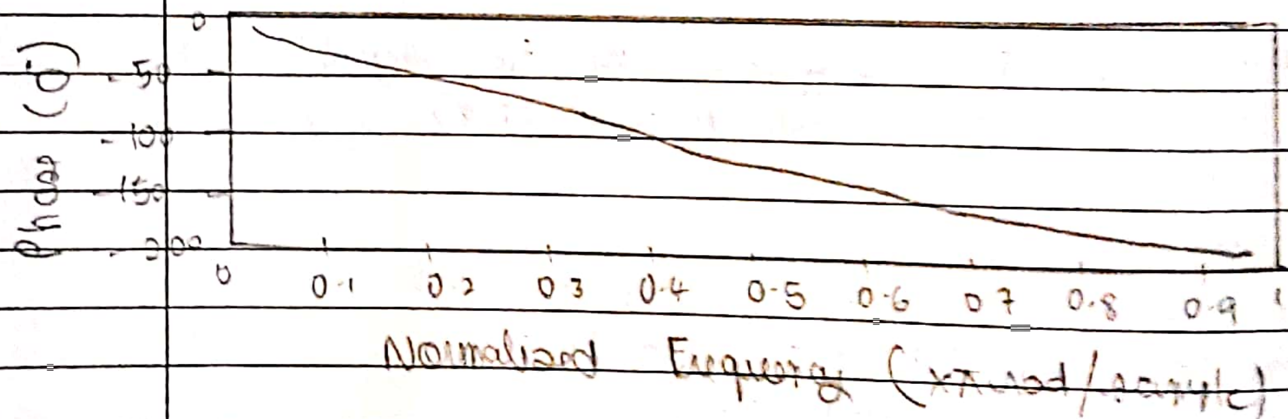
→ b1 = [0.2066 0.4131 0.2066]

a1 = [1 -0.3695 0.1958]

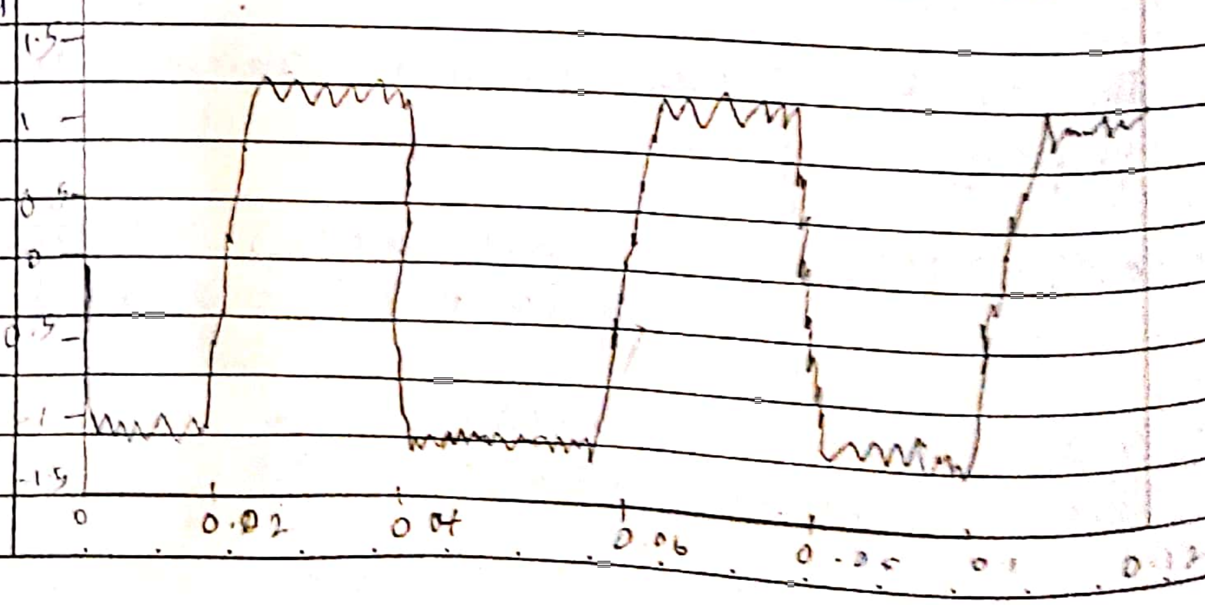
freqz (b1, a1, 64);

title ('Frequency Response')





D1 plot



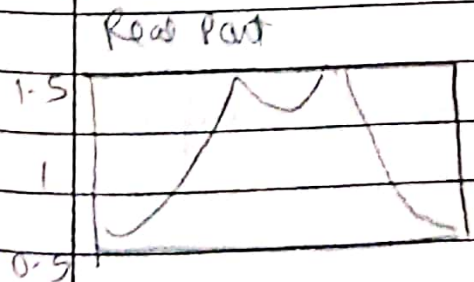
URBAN EDGE ___/___/___

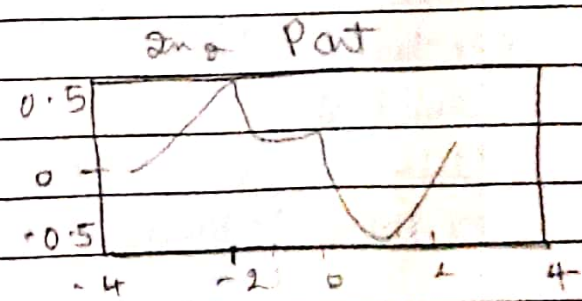
Q.4. Determine and plot the real and imaginary parts and the magnitude and phase analysis of the following DTFT for various values of r and θ.

$$G(e^{jw}) = \frac{1}{1 - 2r\cos(\theta)e^{-jw} + r^2 e^{-j2w}} \qquad 0<r<1$$

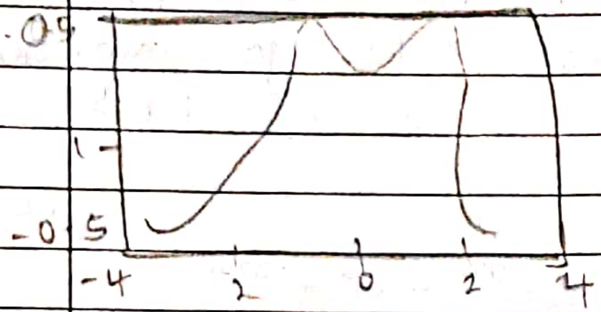
→
```
clear, clc;
theta = 60 * pi / 180
r = 0.5;
w = -pi : pi/100 : pi;
den = 1 - 2 * r * cos (theta) * exp (-j*w) + r^2 exp (-j*2*w);

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')
```

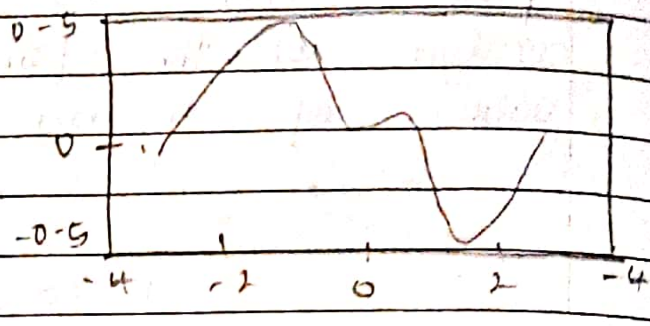




Magnitude



Phase



Q5. Compute and plot the DTFT of the following sequence and observe the properties

$x[n] = A\cos(2\pi f_0 n + \phi)$.

$f_0 = 100$ Hz, $\phi = \pi/6$ and different length sequence.

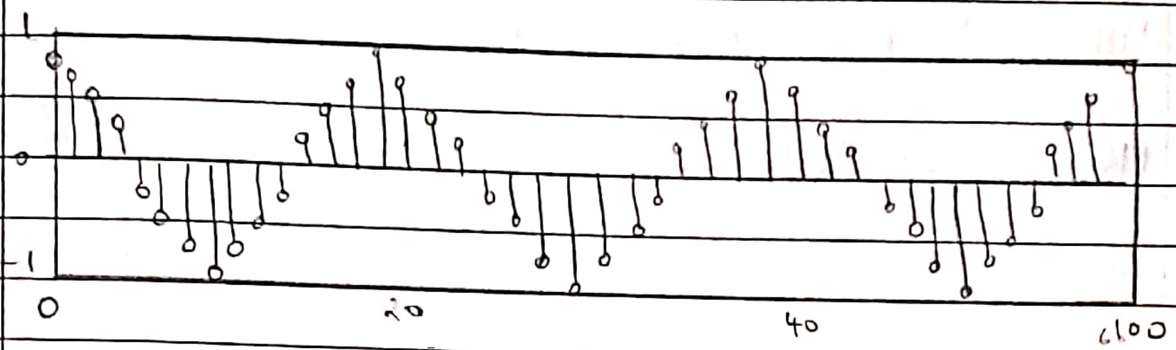
→

```
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 (212), plot (w_axis, sig_dtft);
title ('DTFT Magnitude');
xlabel ('Digital Frequency');
```

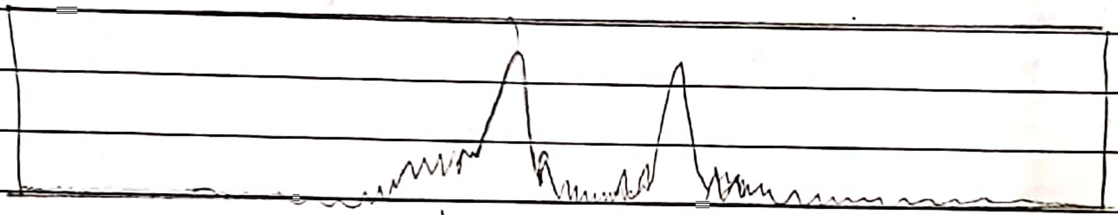
→ Enter length of sequence

100.

Signal



DTFT magnitude.



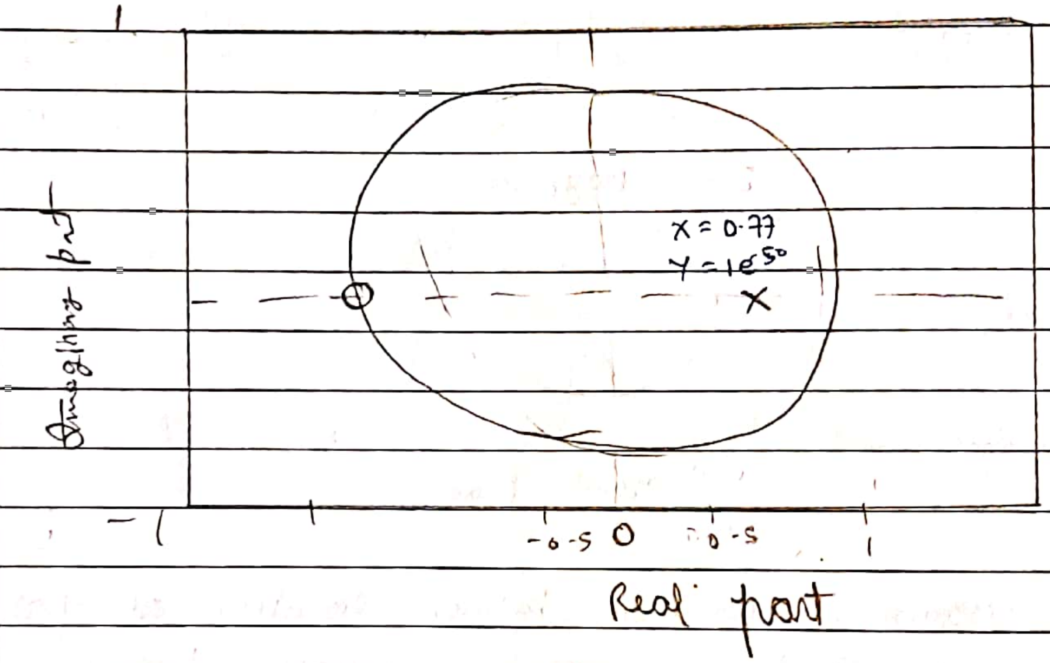
Q6. Three domains : Relation between location of poles and zeroes in z plane, impulse response / frequency response

(1.) $y(n) = 0.77\,y(n-1) + x(n) + x(n-1)$

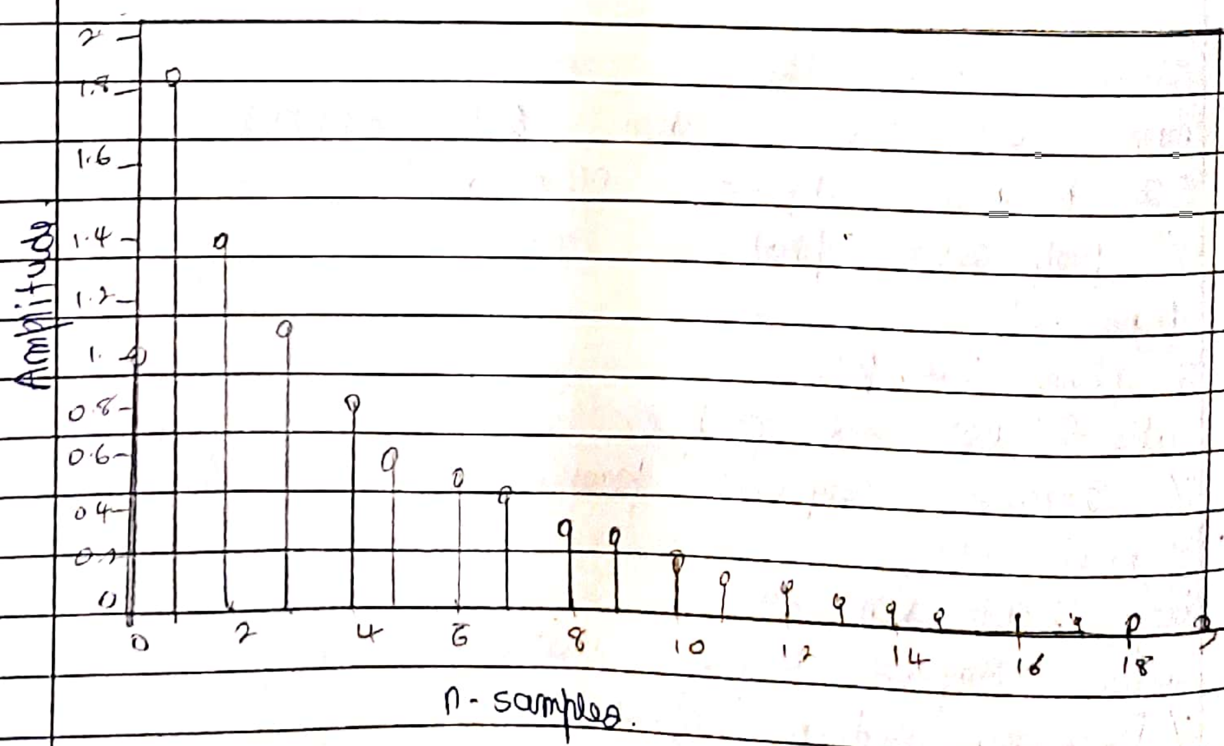
→
```
clear; clc; clf;
num = [1 1];        den = [1 -0.77];
[z, p, k] = tf2zp (num, den);
% pole zero plot
figure (1);
zplane (z, p);
title ('Pole zero plot');
% Impulse response plot
figure (2);
impz (num, den, 20);
title ('Impulse Response Plot').
% Frequency response plot
figure (3);
```

freqz (num, den, 64).
title ('Frequency Response Plot').

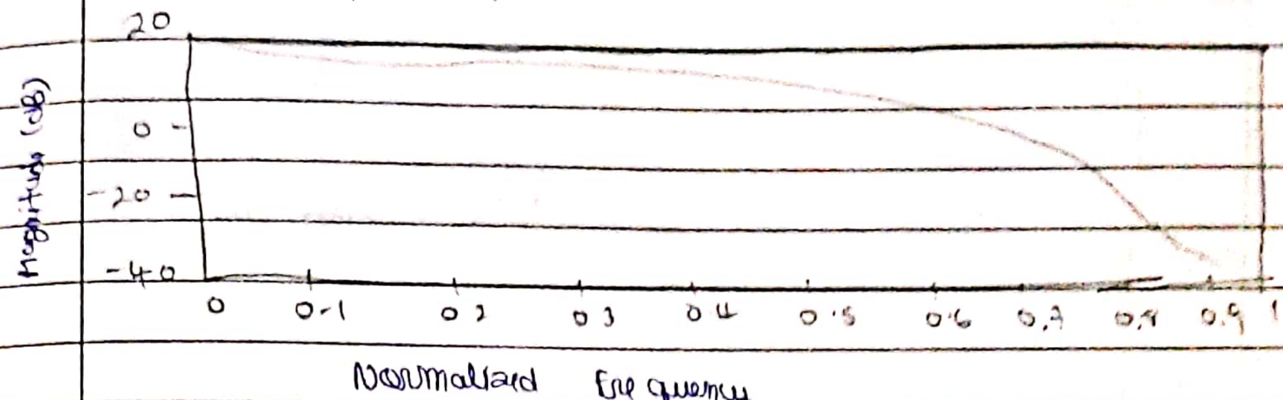
~~Plo~~ Pole zero plot

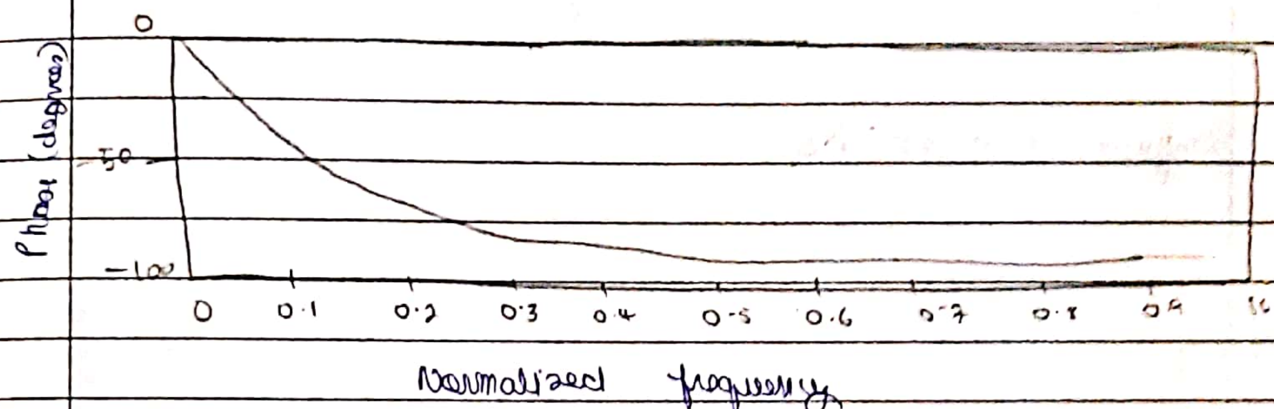


Impulse Response Plot



Frequency Response plot

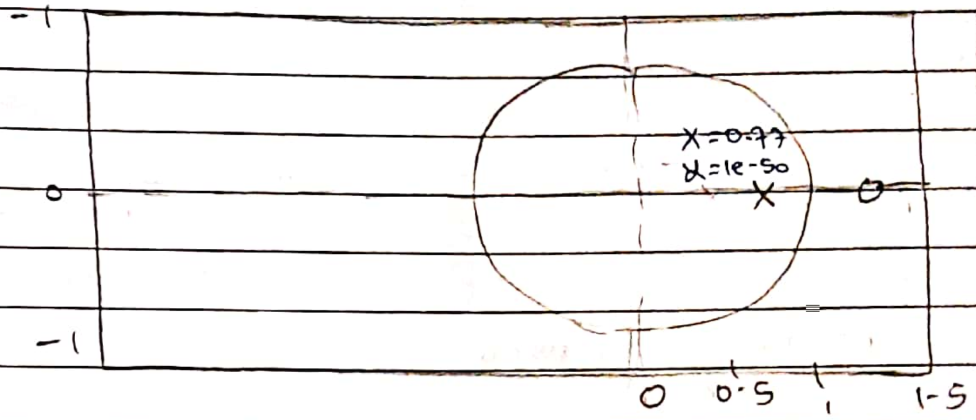




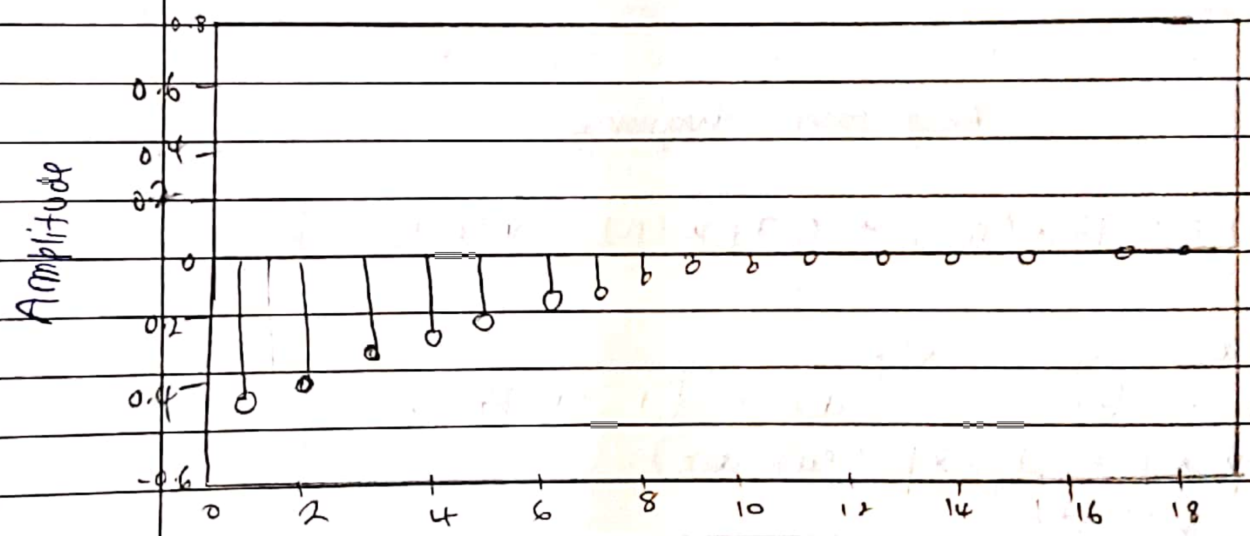
(ii) $y(n) = 0.77y(n-1) + 0.77x(n) - x(n-1)$

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

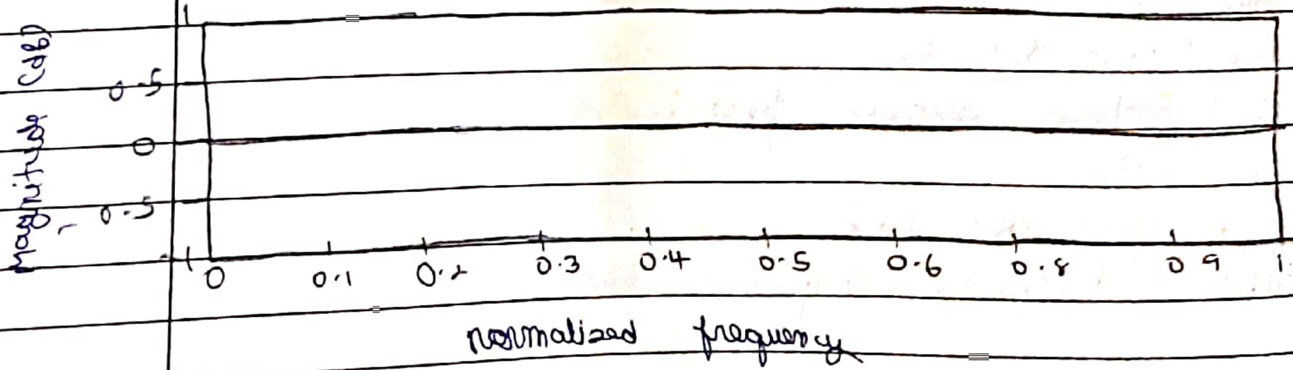
Pole zero Plot

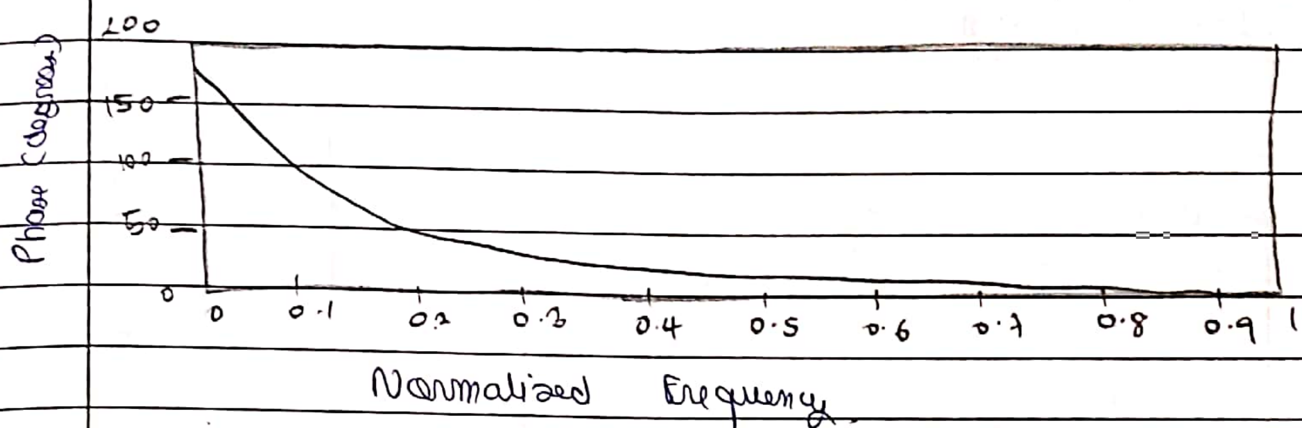


Impulse Response Plot



Frequency Response Plot



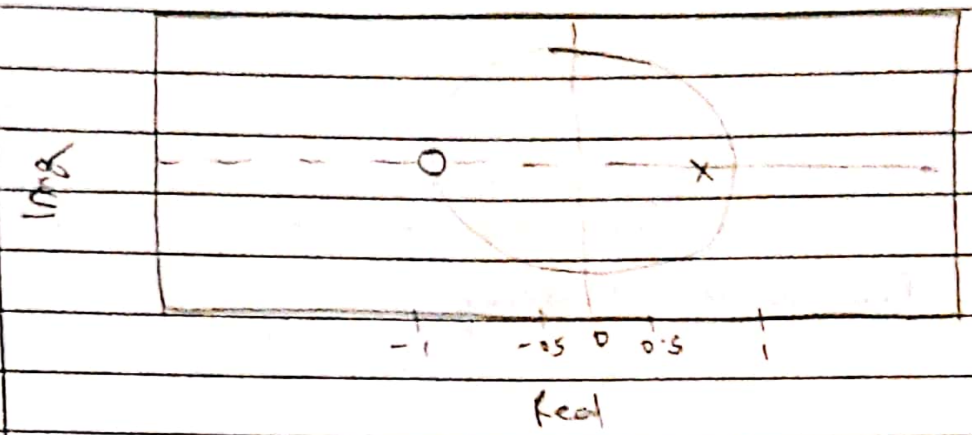


Q7.

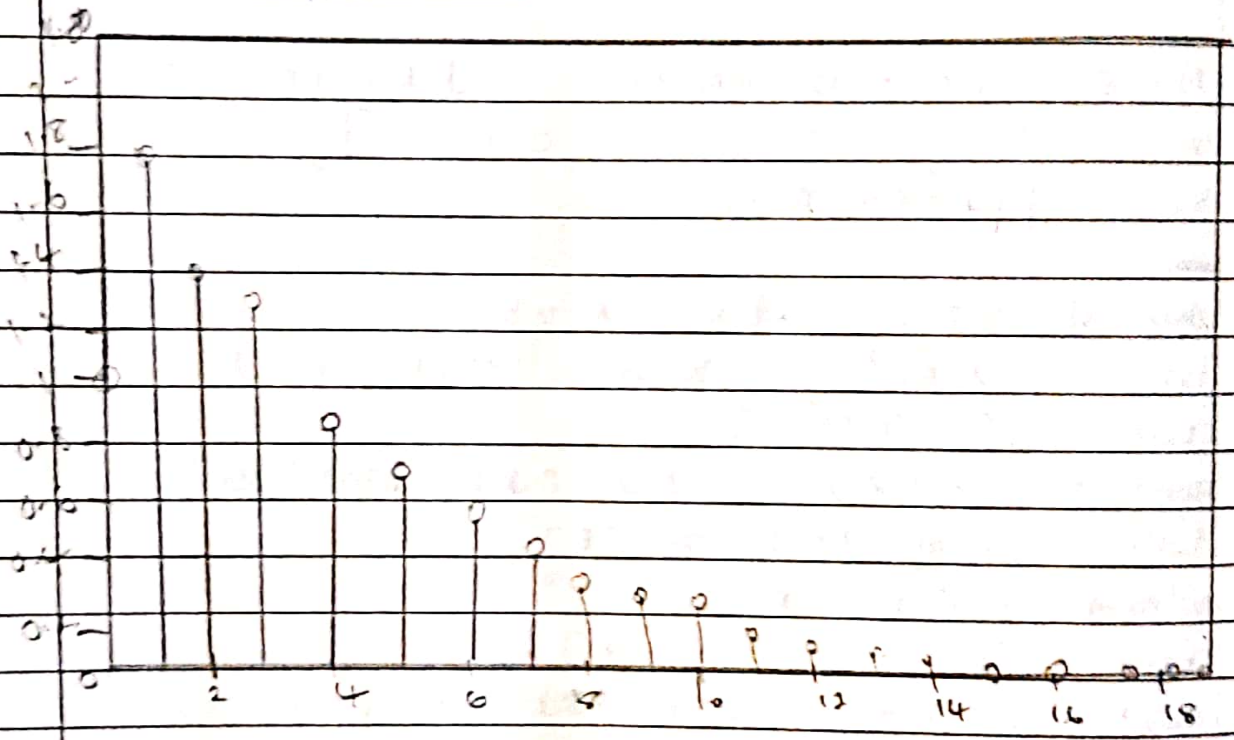
(iii) Three point boxcar, $xb = \{111\ 00000\}$

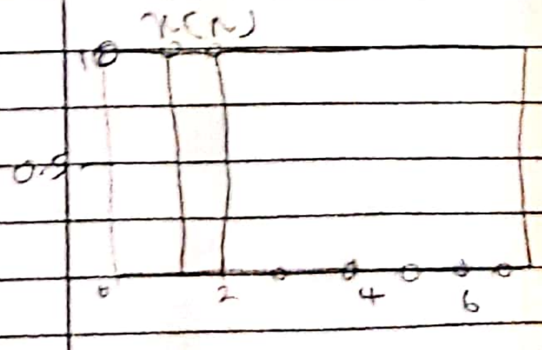
```
N = 8;     nn = 0:(N-1);      kk = nn;
xb = [1 1 1  0 0  0 0 0];
Xb = fft(xb, N);
subplot(221),  stem(nn, xb);
title('x(n)');  xlabel('index (n)');
axis([0 7 0 1]);
subplot(222)   stem(kk, real(Xb));
title('Real part of DFT');
xlabel('index (k)');
axis([0 7 -1 4]);
subplot(224),  stem(kk, imag(Xb));
title('Imag part of DFT');
xlabel('index (k)');
axis([0 7 -2 2]);
```

Pole zero Plot

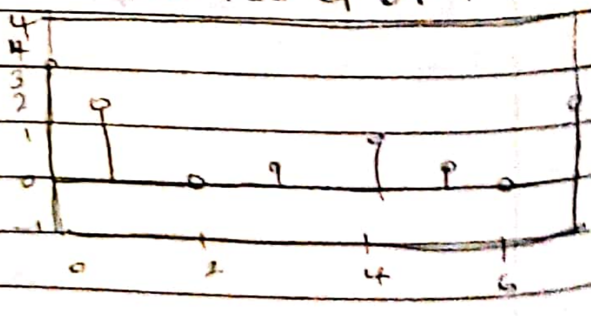


Impulse Response Plot





Real Part of DFT



Q8. 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 signals lengths.

a.) Compute the even and odd parts of $v[n]$.

$v_e[n] = 0.5[v(n) + v[-n \bmod N]]$

$v(-n \bmod N) = [v(1), v(N:-1:2)]$

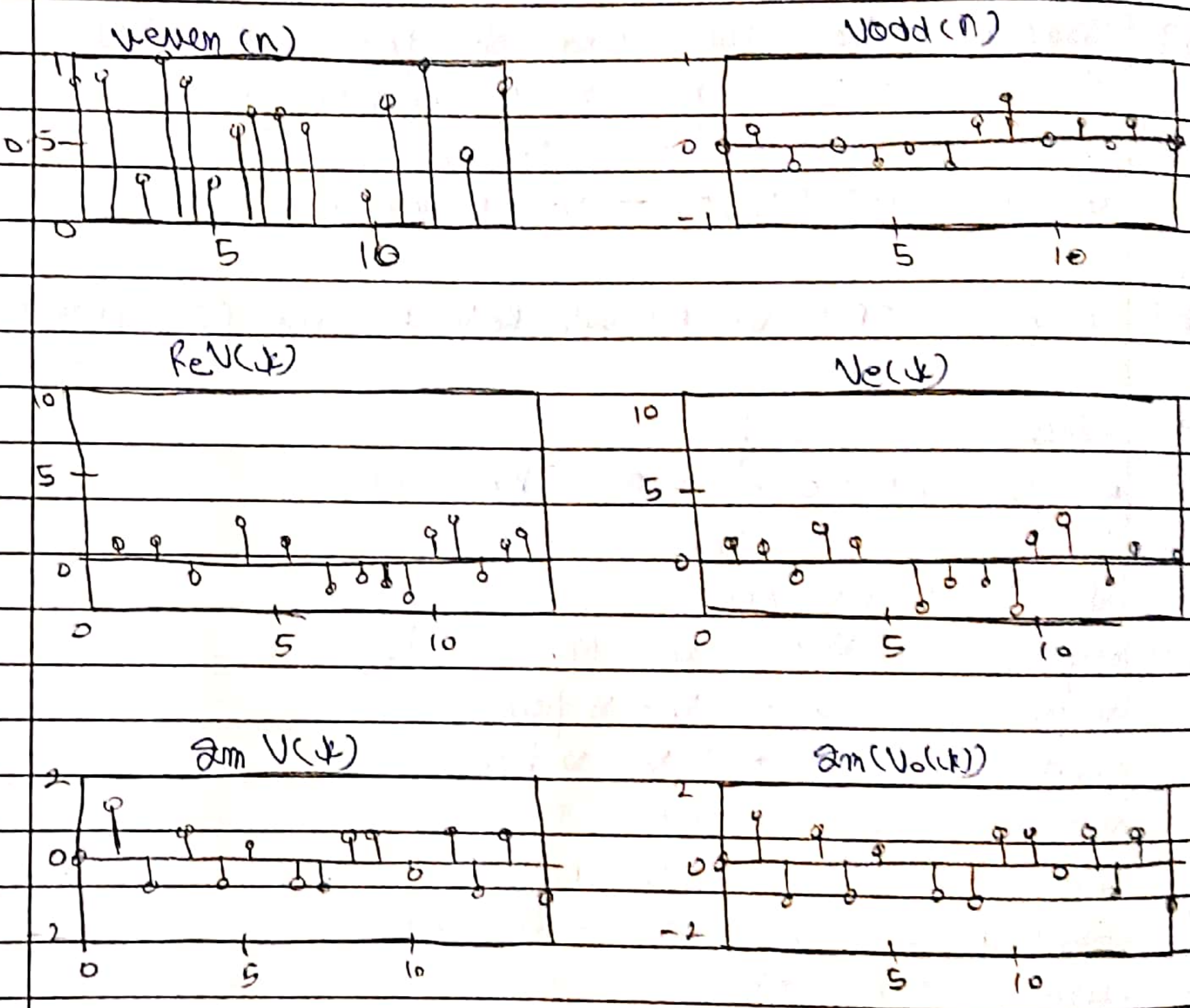
$v_o[n] = 0.5[v[n] - v[-n \bmod N]]$.

b.) Compare DFT $v_e[n]$ with $Re\,V[k]$ and DFT of $v_o[n]$ with ...

→

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



Q9. 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.

```
→ 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);
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;
```