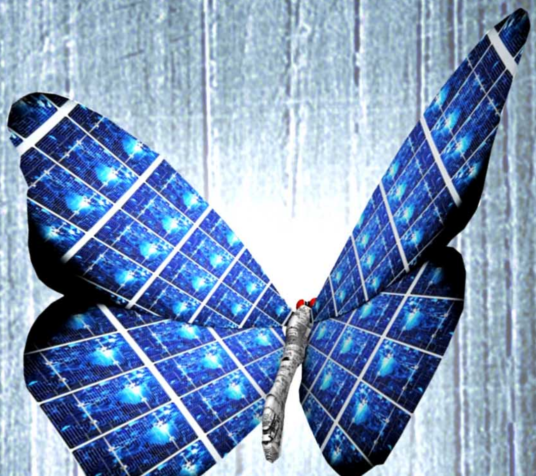
MATLAB NOTES
DIGITAL SIGNAL PROCESSING



AKSHANSH CHAUDHARY

Digital Signal Processing MATLAB Notes, First Edition

Presented by: Akshansh Chaudhary
Graduate of BITS Pilani, Dubai Campus
Batch of 2011

Course content by: Dr. R. Mary Lourde
Then Faculty, BITS Pilani, Dubai Campus

Layout design by: AC Creations © 2013

The course content was prepared during Fall, 2013.
More content available at: www.Akshansh.weebly.com

MATLAB BASICS
Digital Signal Processing

DSP

* MATLAB :- Matrix Laboratory.
  * mainly used for analysing in Z domain.

$\frac{o/p}{i/p}$ {
- efficiency η ≡ Mech. power.
- Gain ≡ electrical qty. V, I, P.
- TF ≡ model of any control sys. represented in freq. domain.

for a TF.

freq. at which o/p of sys is zero : Zero's

is max : Poles.

Root Locus :- Loc^n of roots.

* Gain margin & Phase margin :- How much to inc. gain & how much to change phase so that sys. is stabilized.

Time domain — freq. domain

Integration ≡ ~~Division~~ Multiplicⁿ

Differentiation ≡ Division

* NYQUIST Criterion :- Telling about stability

DSP LAB

Q. Consider a TF :- $\frac{s^2+2s+1}{s^3+s^2+3s+4} = \frac{NUM}{DEN}$

* each line terminⁿ with ;
  (If no ; the value typed will be shown again)
* CMD : WHO : to see all which all parameters of worspace ∃

  Cmd :- CLC : Clear screen.

Writing a parameter :- x.
Syntax

$x = [1 \ 2 \ 3]$

$(\equiv x = s^2 + 2s + 3)$

✡ Root Locus :-

Cmd :- rlocus (num, den)

Cmd :- help <fn name> → gives all cmds related to that fn cmd.

Cmd :- To see value of ~~ro~~ roots :-

Zeros :- zeros = roots (num)
poles = roots (den)

Cmd :- Put labels on x and y axis.

```
xlabel('This is x axis label')
ylabel('frequency')
```

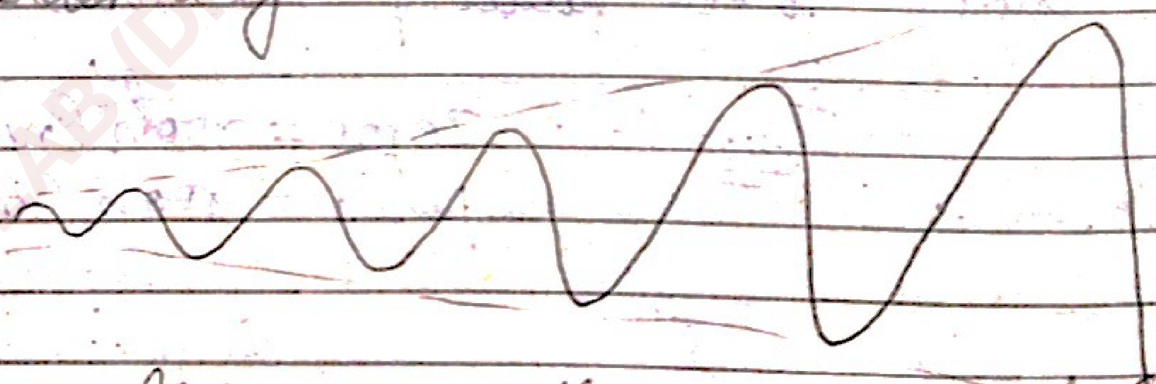
Cmd :- To give title :-

```
title('This is my first plot')
```

* ~~If~~ Cmd :- for step response

```
step(num, den)
```

* Consider a f^n :-



Analysis :- f^n is exponentially ↑ or ↓ :- e^{at} / e^{-at}

f^n is oscillating :- $\sin \omega t$ / $\cos \omega t$

* Whenever graph is plotted simultaneously, use HOLD cmd, Undo hold :- HOLD OFF

* To have 2 graphs in same screen

Cmd :- subplot ~~(2,1,2)~~ (2,1,1)

↓

make 2 graphs & refer to 1st one out of 2

subplot (2, 2, 1)

- 2 → Total no. of graphs
- 2 → column I want to refer
- 1 → which row I want to refer

DSP Lab

* Finding poles & zeros of a TF

Say, $TF = \dfrac{as^2 + bs + c}{ds^3 + es^2 + fs + g}$

So, let ~~a~~ ~~so~~ inputs be

A = [a , b c]

B = [d e f g] .

So, to get a graph of poles & zeros,

Cmd :- pzmap (A , B)

To know the values of poles & zero's →

let

C = roots (A) ↵ ; for zeros

D = roots (B) ↵ ; for poles

Inference :- If poles lie on the real axis, then, response is not oscillatory (either monotonically ↓ or ↑)

Pole lying on left half of s plane

⇒ sth like $\frac{1}{s+a}$ ⇒ $s = -a$ is a pole on left half.

⇒ in freq → e^{-at} → monotonically decreasing

⇒ it goes to zero i.e finishes

⇒ stable sys

⇒ Poles on left half of s plane : Stable
right half : Unstable

★ Step response analysis

Consider a TF :- $G(s) = \frac{s^2+2s+1}{2s^3+4s^2+3s+5}$

$t \longrightarrow s$

Find steady state response from TF given a step i/p.

↳ & finding initial & final value of a TF

As per graph, we get steady state at 0.2

$$\lim_{s \to 0} G(s) = \frac{1}{5} = 0.2$$

eg (2) $TF = \frac{s^2 + 2s + 2}{2s^3 + 4s^2 + 4s + 5}$

$$\lim_{s \to 0} (TF) = \lim_{s \to 0} \left(\frac{2}{5}\right) = 0.4$$

Finding ω

from the step response graph

Take 2 peaks, P_1, P_2.

See their time (sec), T_1, T_2

Time diff $= T_2 - T_1$. freq $= \frac{1}{T_2 - T_1}$

$$freq\Big|_{rad} = \omega = 2\pi\left(\frac{1}{T_2 - T_1}\right)$$

(To see from matlab, check the poles which are having a component in imaginary axis, That is ω)

---

* freq. of o/p of a natural control sys. gives natural freq.

for eq (2),

poles = −1.6914,
−0.1543 ± 1.2059i

zeros = −1 + i
−1 − i

Finding oscill^n freq. from graph :-

$T_1 = 1.98$ s

$T_2 = 7.19$ s.

$T_2 - T_1 = 5.21$ s.

$$\Rightarrow f = \frac{1}{T_2 - T_1} = \frac{1}{5.21} = 0.1919 \text{ Hz}$$

7.19
1.98
5.21

So, $w = 2\pi f = 2 \times 3.14 \times 0.1919$

$\omega = 1.2053$ rad/s → same as the complex part of poles.

* 2 plots on same graph:

eg: 2 step responses

① step(a, b) ↵

hold ↵

step(c, d) ↵

done

Note :- Freq. response depends on pole freq.

Even if we change zeros to be complex, ∃ no change in freq. of response. Only amp. changes.

* Pole freq: called as natural freq. of sys.

* Given a TF, find ~~freq.~~ magnitude response.

S1) Find magnitude from TF.

S2) Convert it to dB.

S3) Plot of magnitude vs ω

~~Freq. response~~

Magnitude response :- The magnitude of the o/p corresponding to diff+ values of freq (ω).

Basically, for any TF, take diff+ values of ω & substitute in TF. The o/p gives diff+ magnitudes. Then, convert them to dB.

eg :- let $TF = G(s) = \frac{s^2 + 2s + 1}{s^3 + 2s^2 + s + 1}$

Method 1 :- DISCRETE Method

a2 = [1 2 1];

b2 = [1 2 1 1];

step(a2, b2) ↵

Take different points on graph for diff+ times T_1, T_2, T_3, T_4, T_5, say

(S1) let $T_1 = 2.92\,s \Rightarrow 0.342\,Hz \Rightarrow 2.15$ rad/s ω_1

$T_2 = 7.02\,s \Rightarrow 0.142\,Hz \Rightarrow 0.89$ rad/s ω_2

$T_3 = 11.5\,s \Rightarrow 0.086\,Hz \Rightarrow 0.546$ rad/s ω_3

$T_4 = 15.6\,s \Rightarrow 0.064\,Hz \Rightarrow 0.402$ rad/s ω_4

$T_5 = 19.9\,s \Rightarrow 0.050\,Hz \Rightarrow 0.3155$ rad/s ω_5

Taken from step response for step (a2, b2)

Now, finding diff magnitudes by substituting ω in TF

(S2)

		dB
$G(s)\vert_{\omega_1}$	$= 0.444$	D_1 7.0523
$\vert_{\omega_2}$	$= 0.854$	D_2 1.370
$\vert_{\omega_3}$	$= 1.0369$	D_3 −0.3147
$\vert_{\omega_4}$	$= 1.098$	D_4 −0.8120
$\vert_{\omega_5}$	$= 1.1194$	D_5 −0.9797

Convert X to dB

$= -20 \log(X)$

(S3) Plot D vs ω.

So, Matlab:-

D = [7.0523 1.370 -0.3147 -0.812 -0.9797]

W = [2.15 0.89 0.546 0.402 0.315]

Plot (D, W) ↵

Method 2: MATLAB

Take num & den. matrices again from given TF.

a2 = [1 2 1];

b2 = [1 2 1 1];

Convert to magnitudes & frequencies

[h, W] = freqs(a2, b2)

Magnitude (h), freq (w)

Convert Magnitude to dB

H = 20 * log(abs(h))

Plot (H, W) ↵

absolute

* Doing convolution.

Cmd :- conv(a, b)↵

∀ TF :- [num den] = sos2tf(a, b);

a = [ ]; b = [ ];

(used for many TFs)

COMBINING 2 TFs,

Consider 2 TFs

$\frac{a}{b}$ & $\frac{c}{d}$.

Now, find $\frac{a}{b}\frac{c}{d}$

→ how?

A = [a ; c]
B = [b ; d]
} appending done ✓

Now, overall TF :-

[num den] = sos2tf([A , B])

makes TF to poly. form

So, now, we have combined TF in polynomial form.

PTO

eg Consider 2 TFs

$TF_1 = \frac{a}{b} \Rightarrow \frac{s^2+2s+1}{s^2+3s+2}$; $TF_2 = \frac{c}{d} = \frac{2s^2+s+3}{s^2+3s+1}$

A = [a, c] ; B = [b, d] ;

So, we get

A = [s^2+2s+1] [$2s^2+s+3$]

B = [s^2+3s+2] [s^2+3s+1]

Now, converting to polynomial form using cmd :-

```
[num den] = sos2tf([A, B])
```

we get

$$\frac{num}{den} = \frac{(s^2+2s+1)(2s^2+s+3)}{(s^2+3s+2)(s^2+3s+1)}$$

$$\Rightarrow \frac{num}{den} = \frac{2s^4+5s^3+7s^2+7s+3}{s^4+6s^3+12s^2+9s+2}$$

MATLAB Assignment 1
Digital Signal Processing

Akshansh Chaudhary

23/9/13

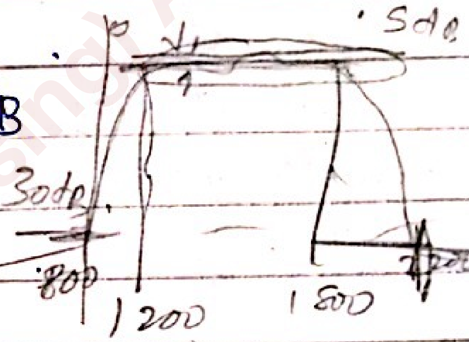
DSP Lab

rmldsplab
@gmail.com

Assignment-1

Q.1) Design an ~~low pass~~ analog filter to meet the specific^ns below:

Passband : 1200-1800 Hz
Stopband attenuation > 30 dB
Passband ripple < 0.5 dB
Transition width : 400 Hz



(a) Butterworth filter
(b) Chebyshev's filter
(c) Elliptical filter
(d)

Plot the freq. response of the filters and compare their performance.

(a) for Butterworth

Idea : find order

help butter ↵

[B, A] = BUTTER (N, Wn)

numerator   denominator

Part (a)

help buttord ↵ → ord. of butterworth for s-domain

★ INSTRUCTION ①

[N, Wn] = BUTTORD (Wp, Ws, Rp, Rs, 's')

- Wn → natural freq with which filter operates
- Wp → pass band edge freq
- Ws → stop band edge freq
- Rp → Max loss in pass band
- Rs → min attenuation of stop band

As per problem,

$W_p = [1200, 1800]$
$W_s = [800, 2200]$
$R_p = 0.5$ dB
$R_s = 30$

entering these values, we get

$n = 7$.

$W_n = 1.0e+003 *$ → Gain is multiplied with 2 natural frequencies

1.1441   1.8879

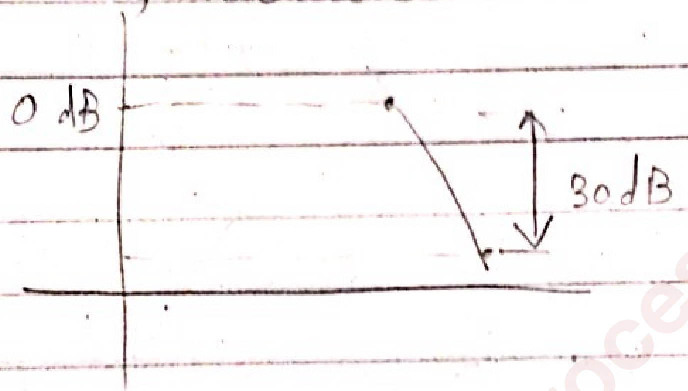
UNDERSTANDING THE BASICS.

☆ Stopband attenuation > 30 dB

⇓

how much value should be attenuated at the stop band.

i.e, decrease

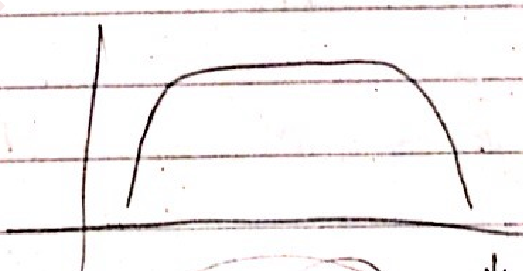


↓ decrease by 30 dB at stop band

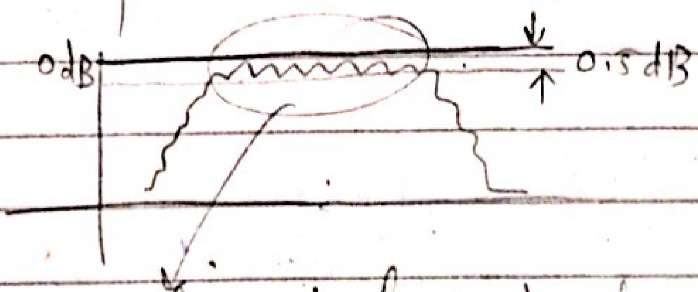
☆ Passband Ripple < 0.5 dB

⇓

for any filter, say



actually, it is:



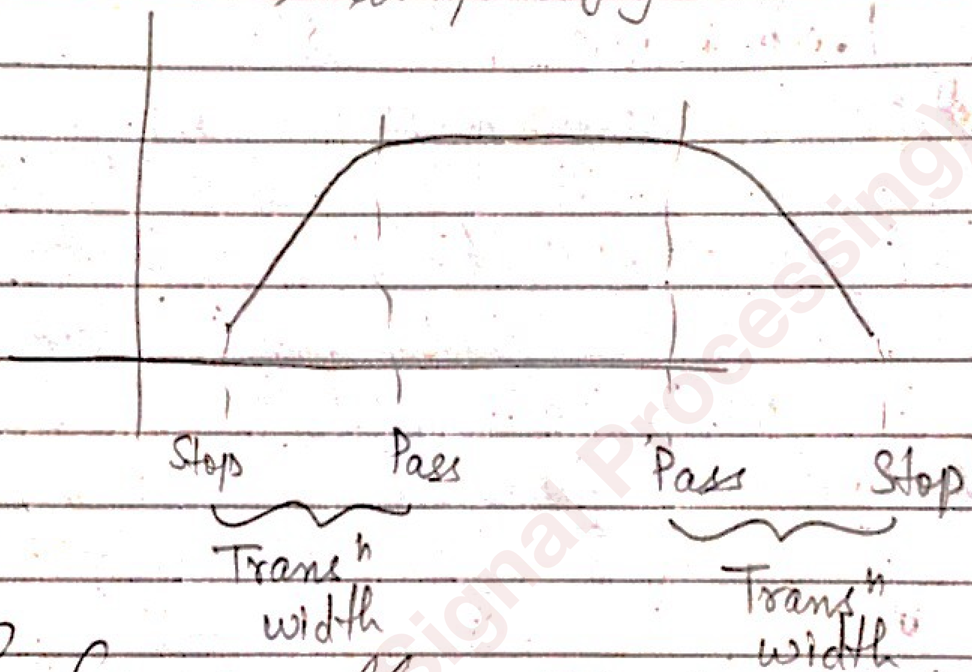
these ripples in passband can have amplitude (in dB) < 0.5 dB.

★ Trans^n width = 400 Hz

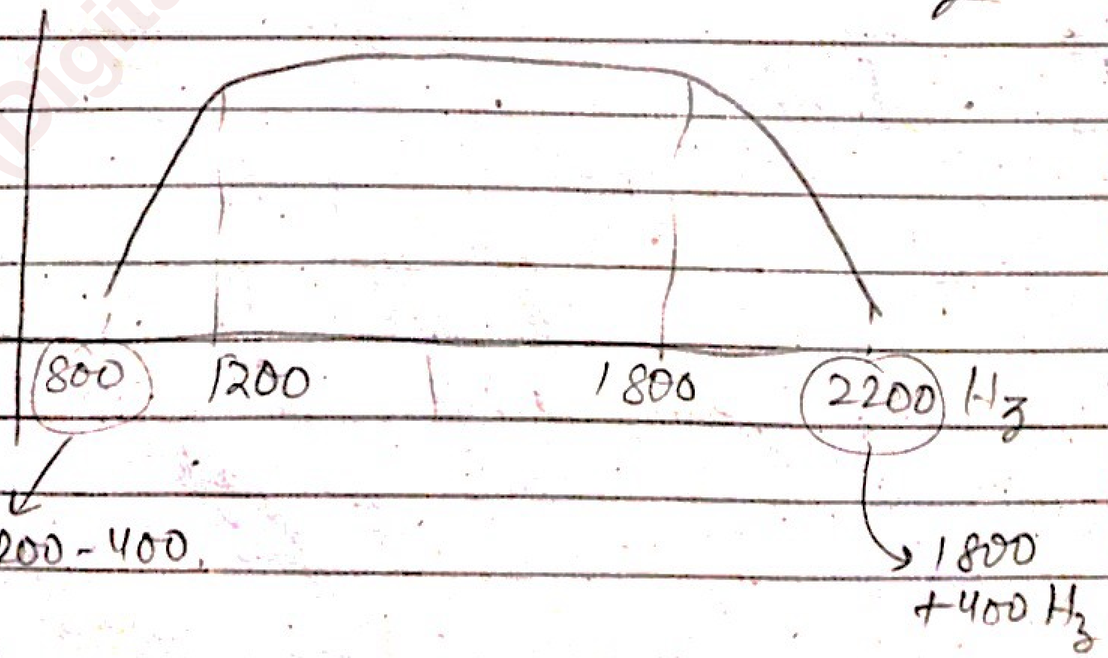
⇓

Basically, the freq req^d to change from passband to stopband.

or, as per fig.



So, for our problem :- Trans^n width = 400 Hz



★ INSTRUCTION ②

[a, b] = butter(n, ω_n, 's') ↵

We get

a =

1.0e+020 * (→ Gain)

Columns 1 through 7

0 0 0 0 0

b =

1.0e+044 * (→ Gain)

0 0 0 0

Columns 8 through 14

Colum 15

0

This instruction gives the TF of my filter

$$TF = \frac{b}{a}$$

✯ INSTRUCTION ③.

[h, f] = freqs(a, b) ↵

h : coeff. , f : frequency

We get different values of h & f on the screen.

TF = $\frac{b}{a}$

Put a = δ(t).

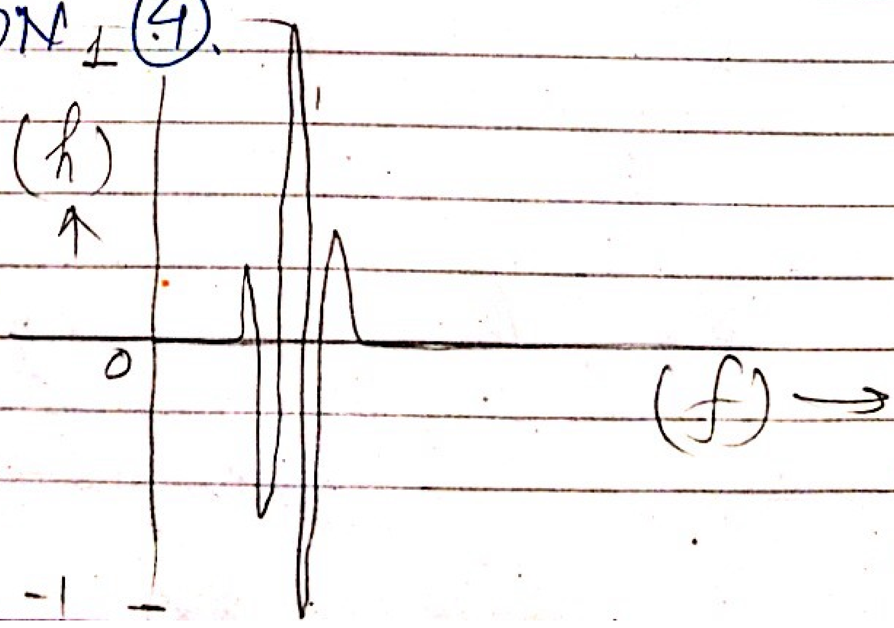
Then, output gives a poly fn s

The coeff. of s polynomial gives the coeff. of h.

✯ INSTRUCTION ④.

plot(f, h)

We get this plot.

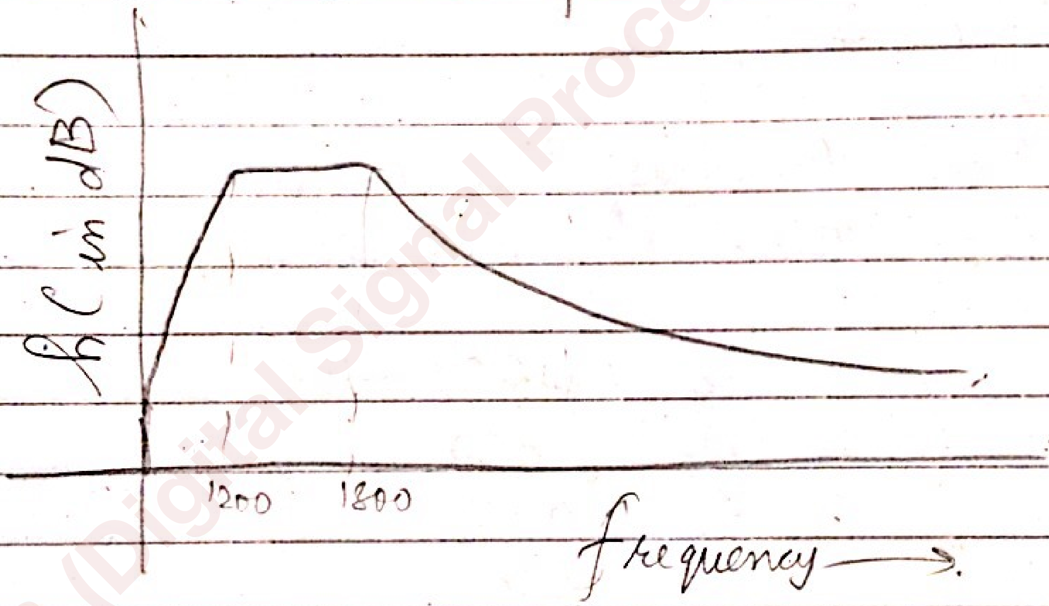


✯ INSTRUCTION ⑤

plot (f, 20* log 10 (abs(h))) ↵

↓ gives a plot of (h) → in dB vs f.

Plot looks something like this :-



Part (b)

Chebyshev's

Instruction ①

[N2, Wn2] = Cheb1ord (wp, ws, Rp, Rs)

N : ord. of lowest ord. analog Cheby. Type - I filter

Rp : Max. loss in pass band

Rs : Min. attenuation in stop band

Using $w_p = [1200, 1800]$

$w_s = [800, 2200]$

$R_p = 0.5$

$R_s = 30$.

We get

N2 = 4 . (order)

W_{n2} = 1200 1800

2 values of natural freq.

Instruction (2)

[B, A] = Cheby1 (N, R, Wn, 's') ↵

This gives values of numerator & den. of TF

A = ____ B = ____

Instruction (3)

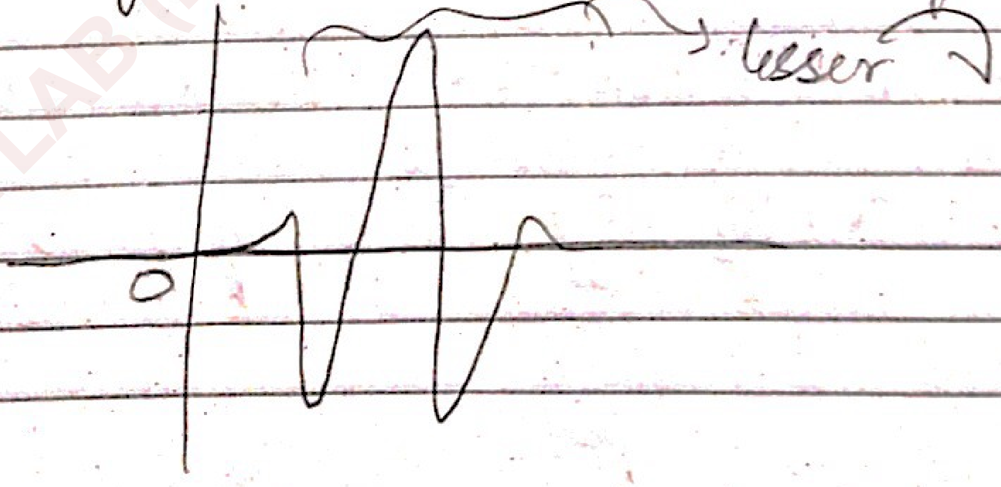
[h2, f2] = freqs(B, A)

This, just as before, gives many values of h, f

Instruction (4)

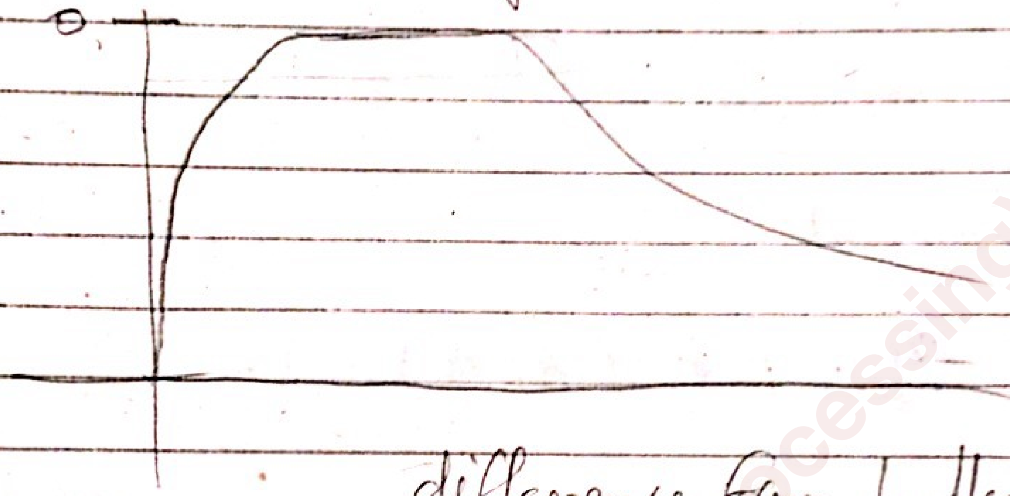
plot (f2, h2)

We get a plot which is slightly different



Instruction (5)

plot (f2, 20*log10(abs(h2)))



difference from butterworth :-
Touches 0 dB very nicely

Part (c) Elliptical

Instruction (1)

[N1, Wn] = Ellipord (Wp, Ws, Rp, Rs, 's')

Wp, Ws, Rp, Rs (same as before)

We get ord. $N_1 = 3$

$W_n1 = 1200 \ 1800$

Instruction ②.

[B1, A1] = ELLIP (N, Rp, Rs, Wn, 's') ↵

We get a TF = $\frac{B1}{A1}$

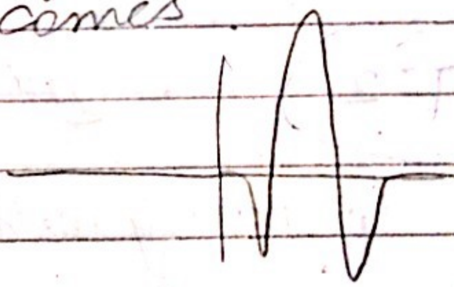
Instruction ③

[h1, f1] = freqs (B1, A1)

values of h1, f1 comes.

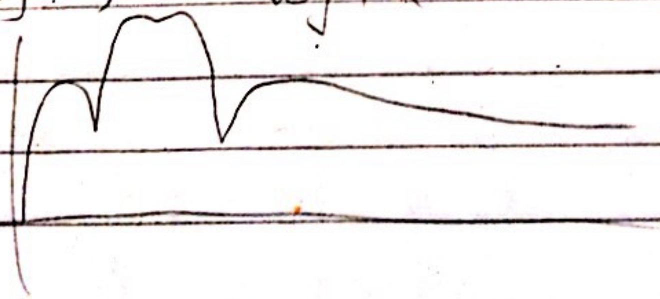
Instruction ④

plot(f1, h1)



Instruction ⑤

plot (f1, 20* log10(abs(h1))) ↵

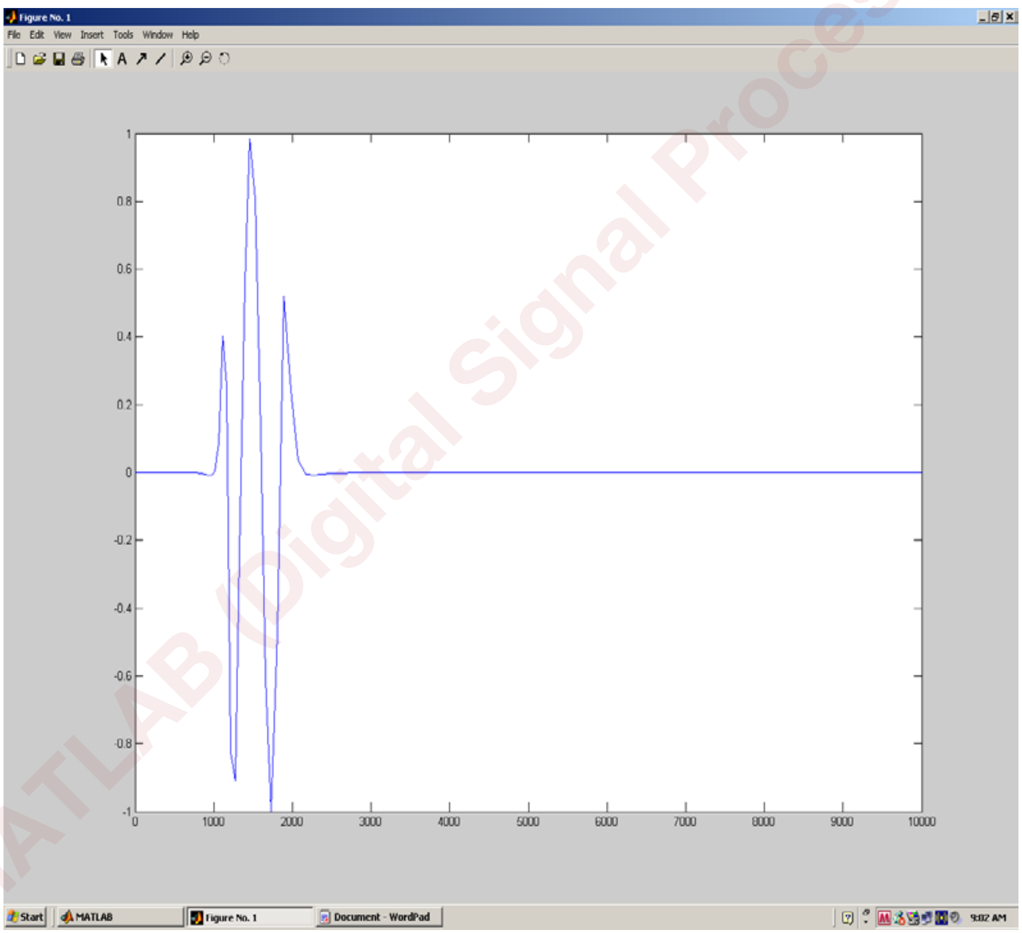


Akshansh Chaudhary
2011AAPS300U

Digital Signal Processing Lab
Assignment 1

Butterworth filter

Instructions -
```
[N1,Wn1]=buttord([1200 1800],[800 2200],0.5,30,'s')
[B1,A1]=butter(N1,Wn1,'s')
[h1,f1]=freqs(B1,A1)
plot(f1,h1)
plot(f1,20*log10(abs(h1)))
```



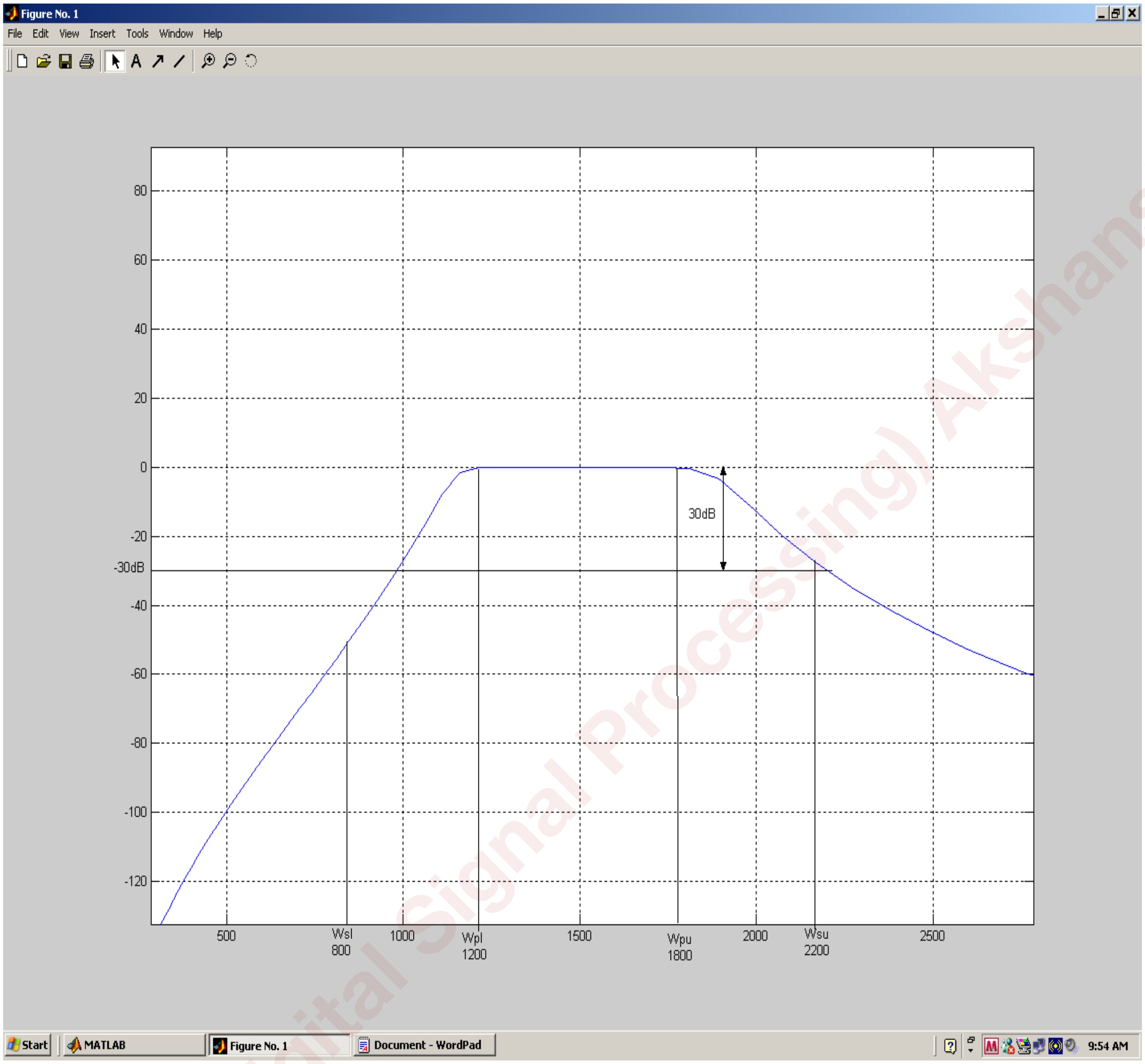


Figure No. 1
File Edit View Insert Tools Window Help
80
60
40
20
0
-20
-30dB
-40
-60
-80
-100
-120
30dB
500
Wsl
800
1000
Wpl
1200
1500
Wpu
1800
2000
Wsu
2200
2500
Start
MATLAB
Figure No. 1
Document - WordPad
9:54 AM

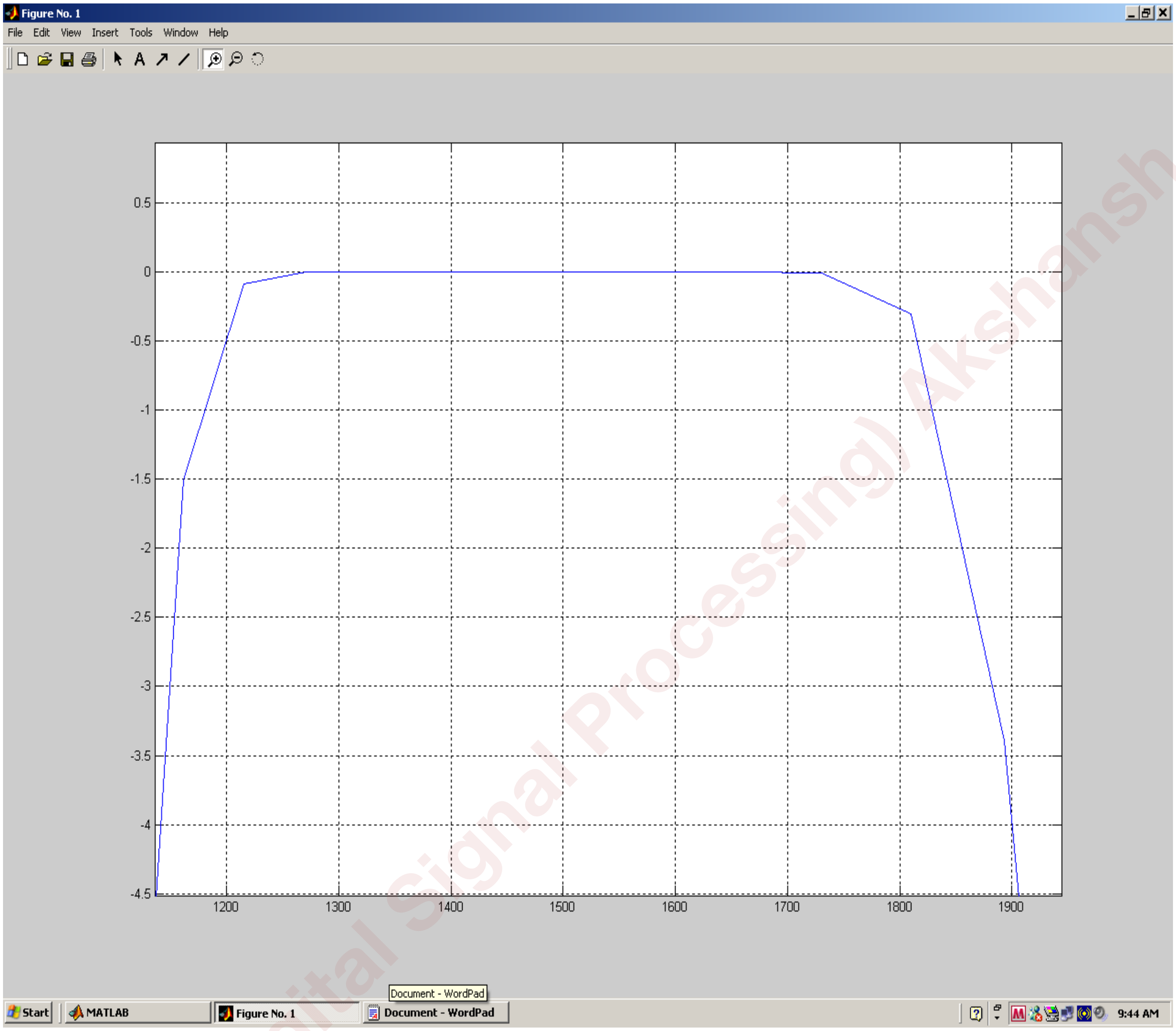


Figure No. 1
File Edit View Insert Tools Window Help
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
1200
1300
1400
1500
1600
1700
1800
1900
Document - WordPad
Start
MATLAB
Figure No. 1
Document - WordPad
9:44 AM

Chebyshev's filter

Instructions -

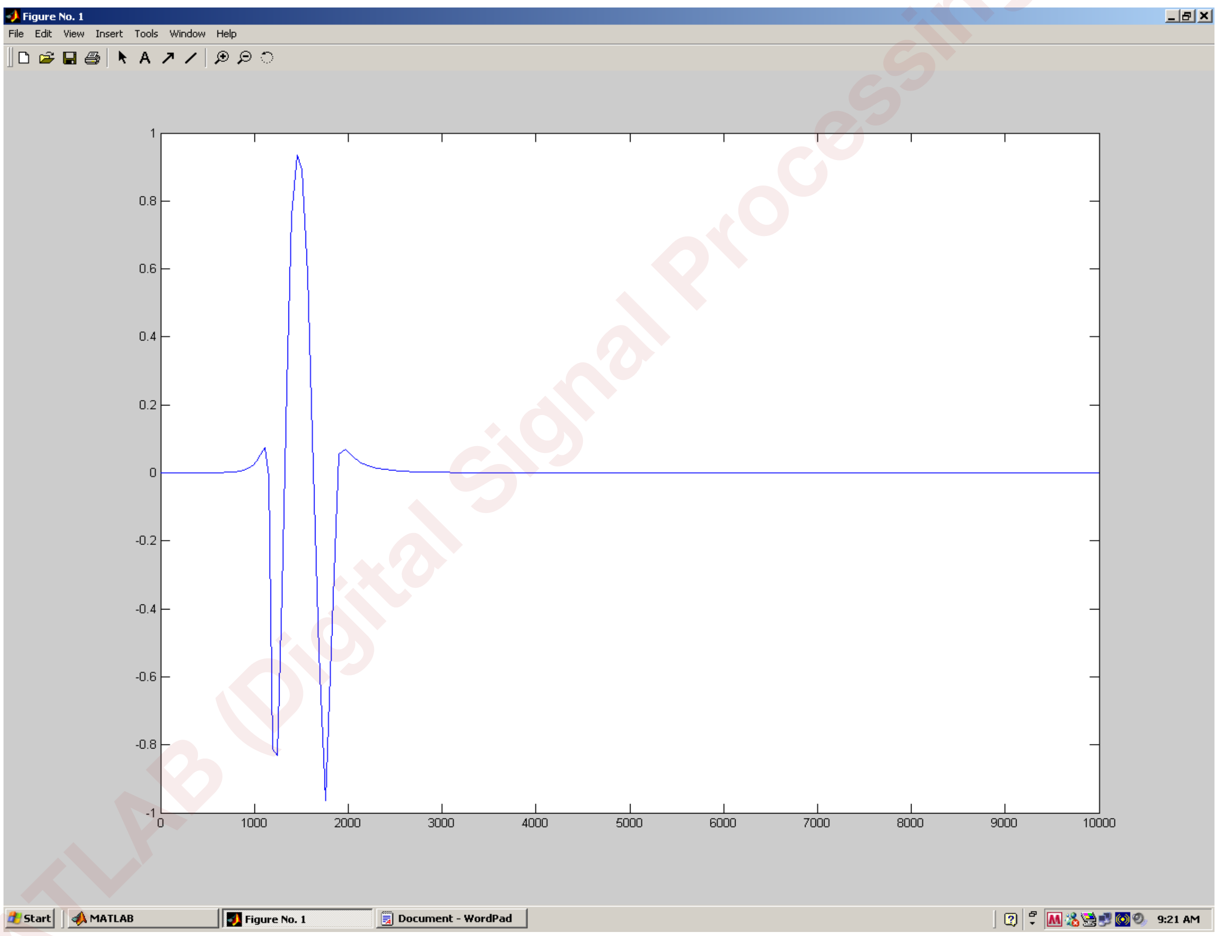
[N2,Wn2]=cheb1ord([1200 1800],[800 2200],0.5,30,'s')

[B2,A2]=cheby1(4,0.5,[1200 1800],'s')

[h2,f2]=freqs(B2,A2)

plot(f2,h2)

plot(f2,20*log10(abs(h2)))



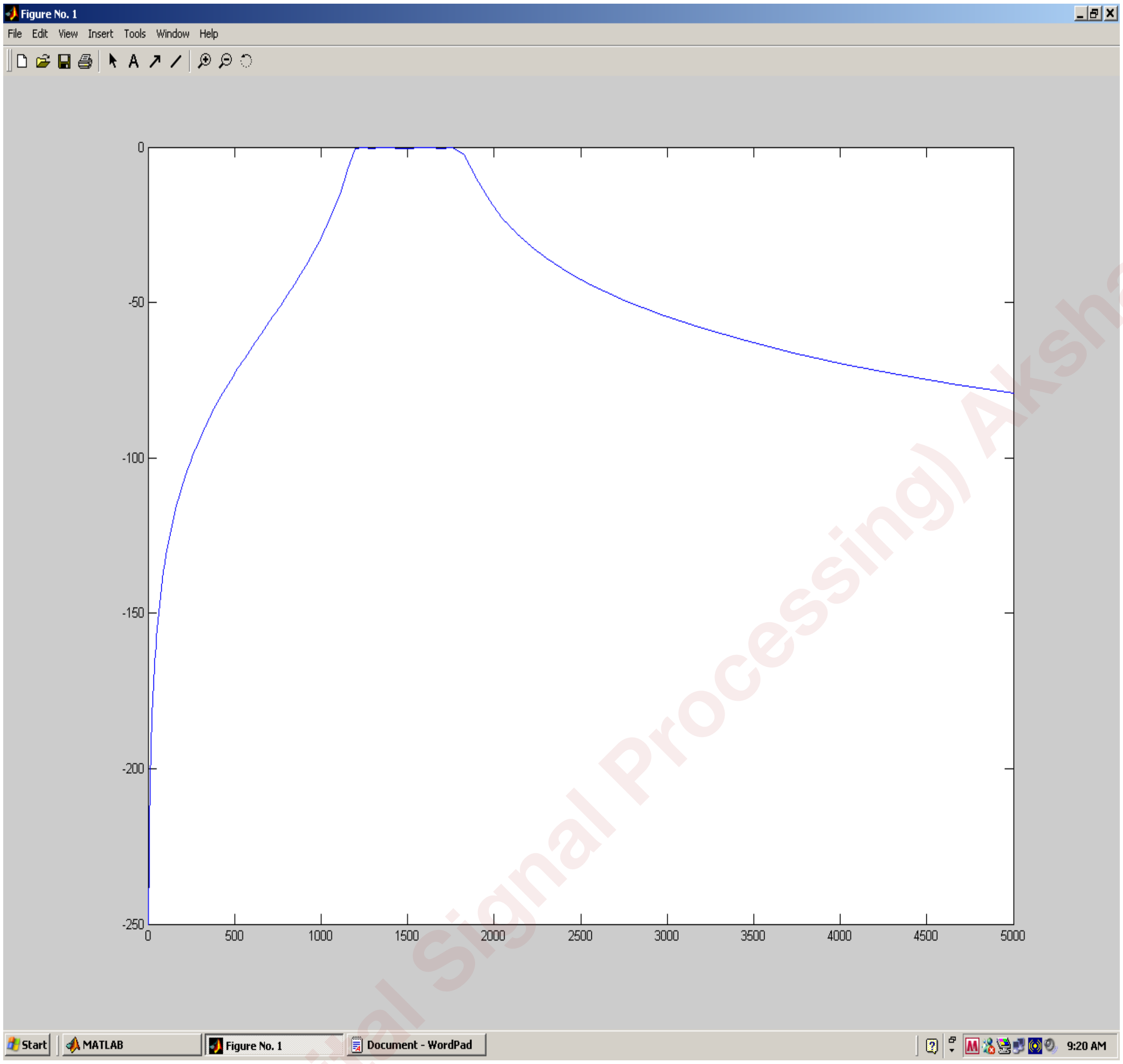


Figure No. 1
File Edit View Insert Tools Window Help
0
-50
-100
-150
-200
-250
0 500 1000 1500 2000 2500 3000 3500 4000 4500 5000
Start
MATLAB
Figure No. 1
Document - WordPad
9:20 AM

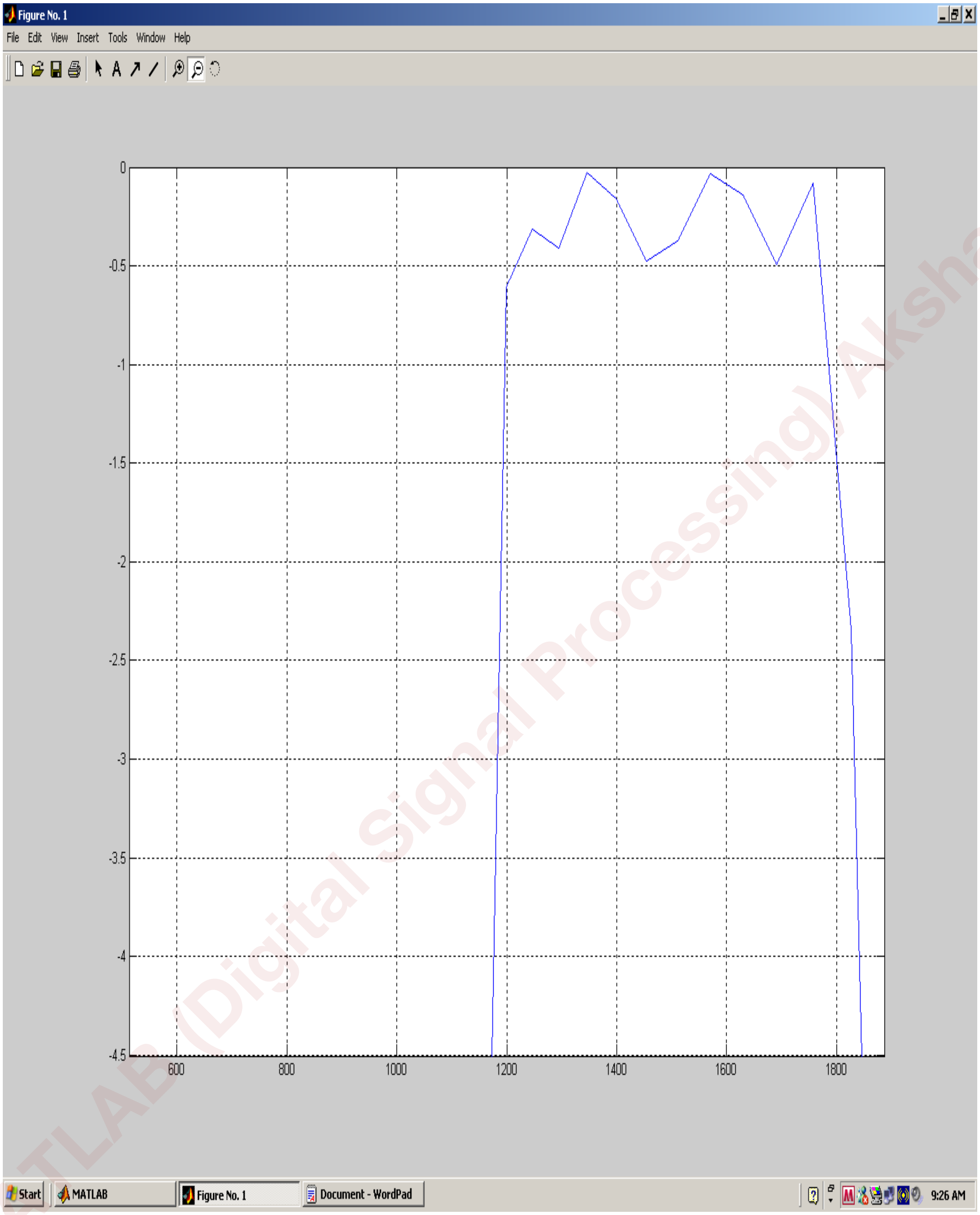


Figure No. 1
File Edit View Insert Tools Window Help
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
600
800
1000
1200
1400
1600
1800
Start
MATLAB
Figure No. 1
Document - WordPad
9:26 AM

Elliptical filter

Instructions -

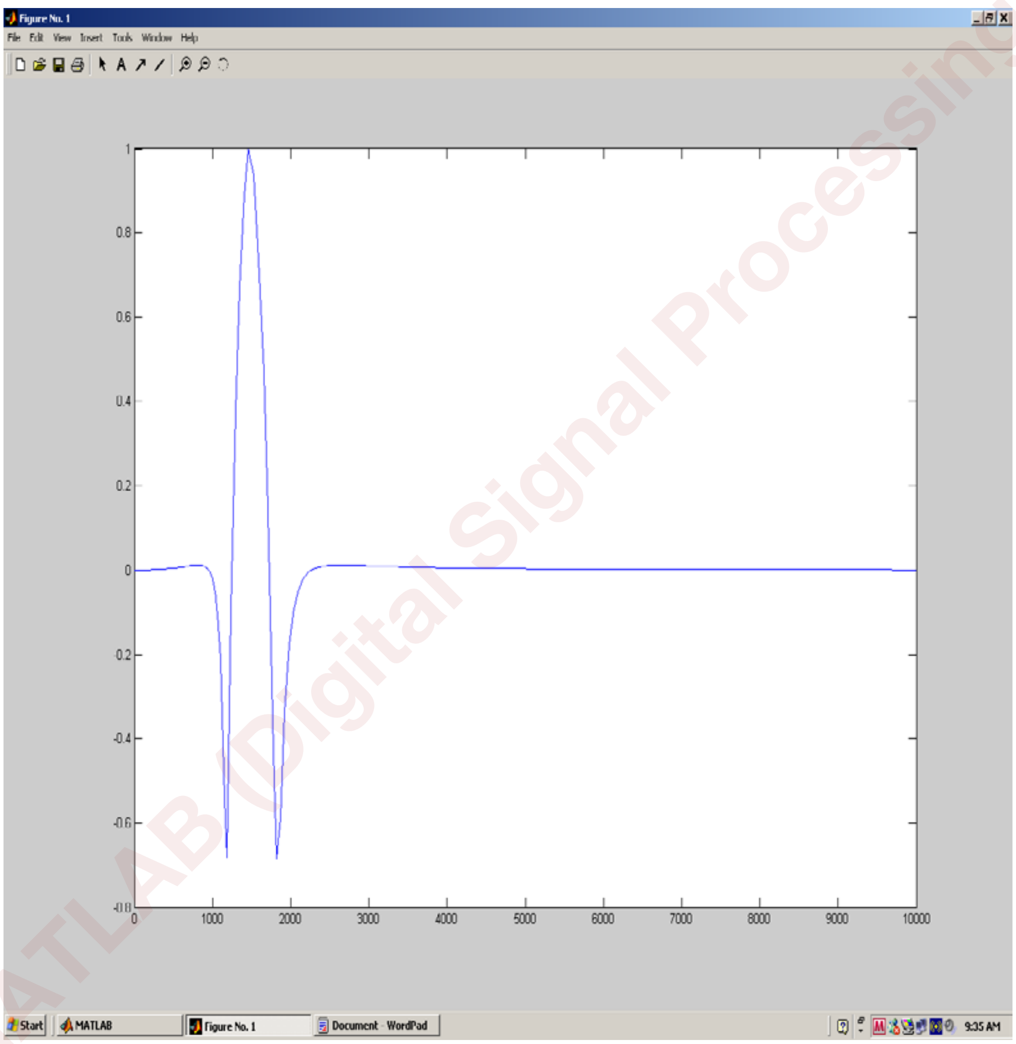
[N3,Wn3]=ellipord([1200 1800],[800 2200],0.5,30,'s')

[B3,A3]=ellip(N3,0.5,30,[1200 1800],'s')

[h3,f3)=freqs(B3,A3)

plot(f3,h3)

plot(f3,20*log10(abs(h3)))



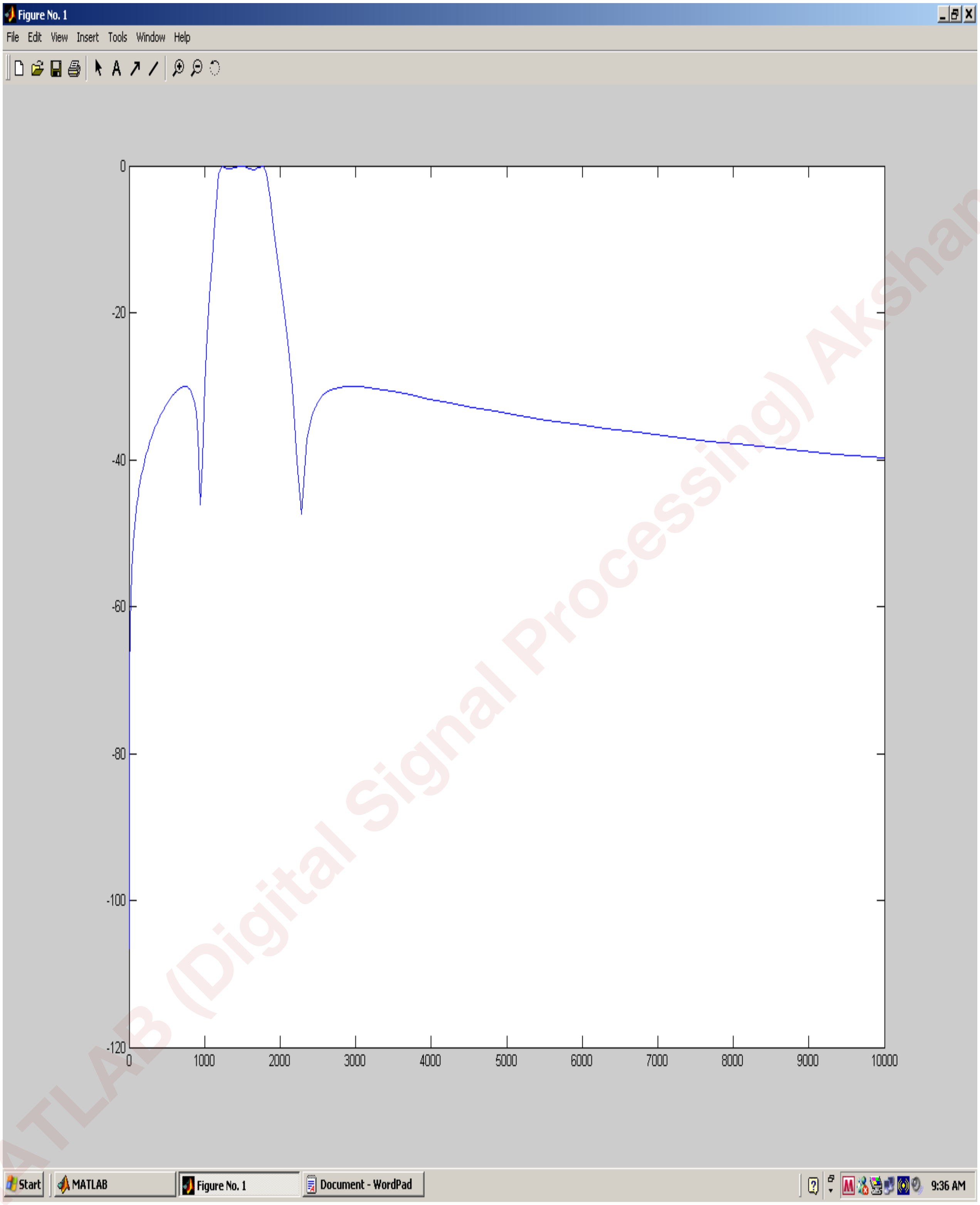


Figure No. 1
File Edit View Insert Tools Window Help
0
-20
-40
-60
-80
-100
-120
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Start
MATLAB
Figure No. 1
Document - WordPad
9:36 AM

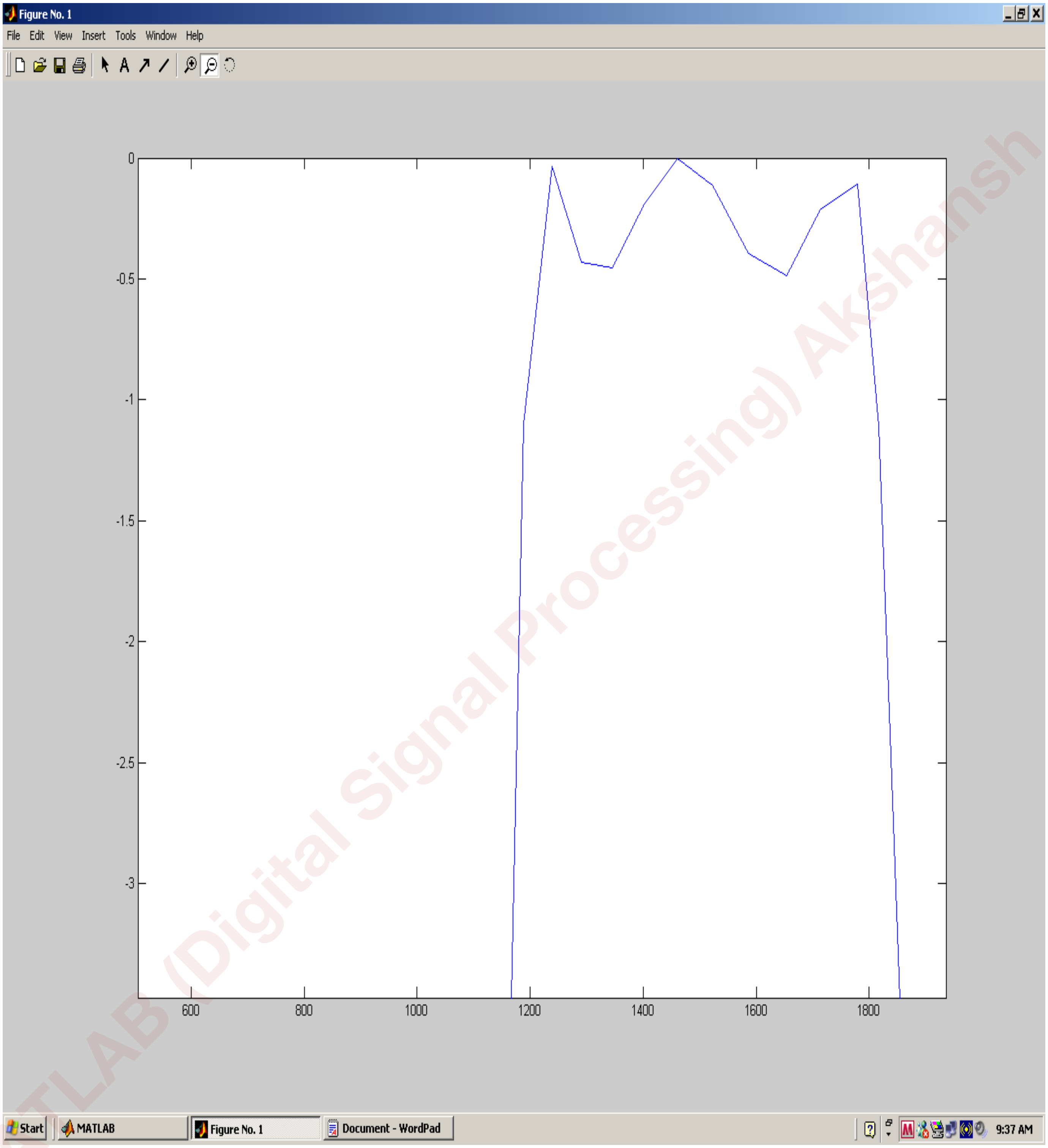


Figure No. 1
File Edit View Insert Tools Window Help
0
-0.5
-1
-1.5
-2
-2.5
-3
600
800
1000
1200
1400
1600
1800
Start
MATLAB
Figure No. 1
Document - WordPad
9:37 AM

Comparison

As the order decreases, frequency decreases.
For Butterworth filter, n=7.
For Chebyshev's filter, n=4
For Elliptical Filter, n=3.

MATLAB Assignment 2
Digital Signal Processing

Akshansh Chaudhary

DSP - Lab

Assignment - 2

Q. The TF of a discrete-time sys. is given by

(TF1) $H(z) = \dfrac{z^2 - z}{z^2 - 0.9051z + 0.4096}$

Determine location of poles and zeros. Plot the pole zero map of function.

First, convert to -ve powers of z

So, $H(z) = \dfrac{z^2(1 - z^{-1})}{z^2(1 - 0.9051z^{-1} + 0.4096z^{-2})}$ (TF2)

Comparing with std. form

$H(z) = TF = \dfrac{b_0 + b_1 z^{-1} + \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots}$

we get

$b_0 = 1 \quad b_1 = -1$

$a_0 = 1 \quad a_1 = -0.9051 \quad a_2 = 0.4096$

Using (TF 2) : Taking coeff.

Numerator matrix gives zeros
Denominator matrix gives poles

b_0 b_1 → highest power = 1, power = −1

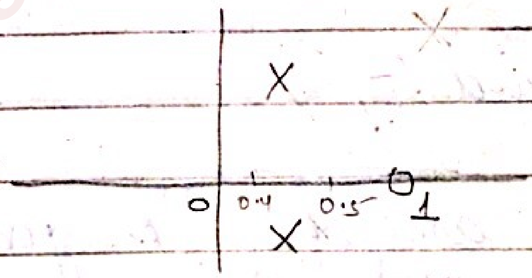
So, f^n :- num = [1 −1]
den = [1 −0.9051 0.4096]
a_0 a_1 a_2

★ Some useful commands :-
for superimposing s-plane & z-plane grids for root locus or pole/zero maps.
Sgrid zgrid

★ Now, for plotting poles & zeros on z-plane

Cmd :- pzmap(num, den)

we get



from TF 2

Conclusion :- Poles are outside unit circle of z plane,
So, sys. is unstable with z^{-1} powers in TF.

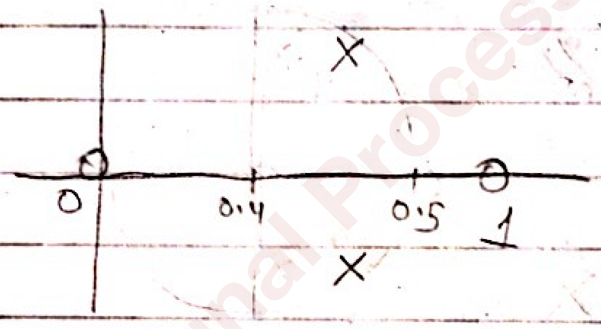
Using TF (1) : Taking coeff

writing coeff. of (TF, 1) in matrix

a = [1 -1 0]
b = [1 -0.9051 0.4096]

pzmap (a, b)

we get



Conclusion:

Poles are inside unit circle z plane. So, sys. is stable with +ve powers of z in TF

Question continued :-

(b) Repeat the same if f^n is given by

$$H(z) = \frac{1 - z^{-1}}{1 - 0.9051 z^{-1} + 0.4096 z^{-2}}$$

(c) plot pole-zero diagram of H(z) in both cases

(i) with numerator & denominator polynomial coefficients as inputs.

(ii) with poles and zeroes as inputs.

(d) Plot the frequency response of the DT sys. with a sampling frequency of

(i) 1 kHz

(ii) 10 kHz

(iii) 100 kHz.

Digital Signal Processing
MATLAB Assignment 2
Akshansh Chaudhary
Id no. – 2011AAPS300U

Question: The transfer function of a discrete time system is given by

(a) $H(z)=(z^2-z)/(z^2-0.9051z+0.4096)$
    Determine the location of poles and zeroes.
    Plot the pole zero map of function.
(b) Repeat the same if function is given by
    $H(z)=(1-z^{-1})/1-0.9051z^{-1}+0.4096z^{-2})$
(c) Plot pole-zero diagram of H(z) in both cases
    (i.) With numerator and denominator polynomial coefficients as inputs
    (ii.) With poles and zeroes as inputs
(d) Plot the frequency response of the DT system with a sampling frequency of
    (i.) 1 kHz
    (ii.) 10kHz
    (iii.) 100kHz

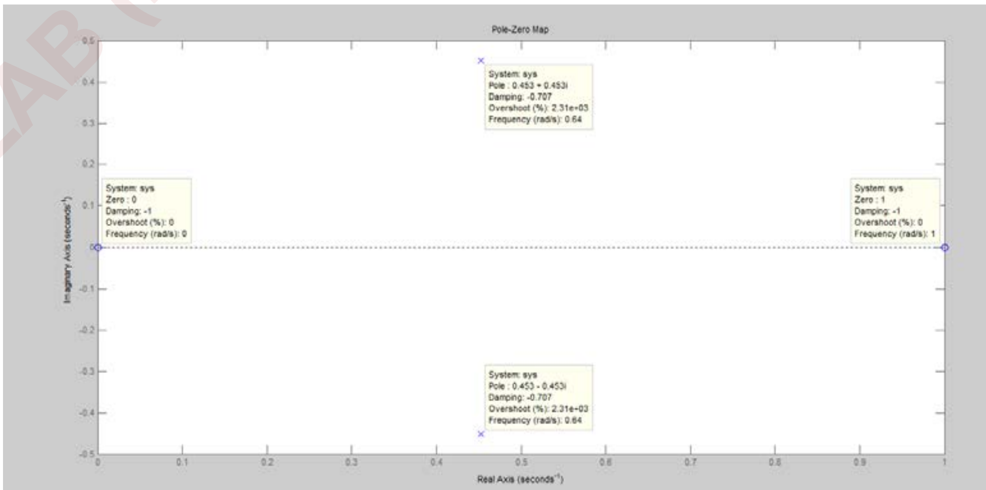
Solution:

Let the transfer functions be named as

(a) $TF1 = (z^2-z)/(z^2-0.9051z+0.4096)$
(b) $TF2 = (1-z^{-1})/1-0.9051z^{-1}+0.4096z^{-2})$

Subpart 1: Polynomial coefficients of TF1 and TF2 as inputs:

a. For TF1,
   Commands:
   A=[1 -1 0]; B=[1 -0.9051 0.4096];
   pzmap(A,B)
   From this, we get the following graph –



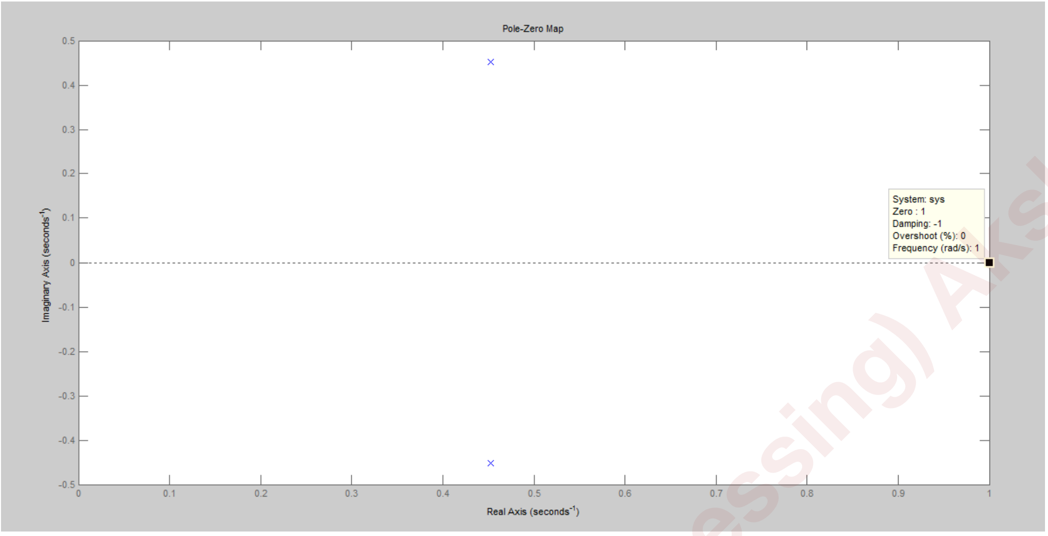
b. For TF2,

Commands:

num=[1 -1]; den=[1 -0.9051 0.4096];

pzmap(num,den)

We get the following graph –



Subpart 2: Poles and Zeroes as inputs:

a. For TF1,

Finding the poles and zeroes of the TF.

Commands:

```
>> A1=roots(A)
A1 =
     0
     1
>> B1=roots(B)
B1 =
   0.4526 + 0.4525i
   0.4526 - 0.4525i
```

b. For TF2,

Commands:

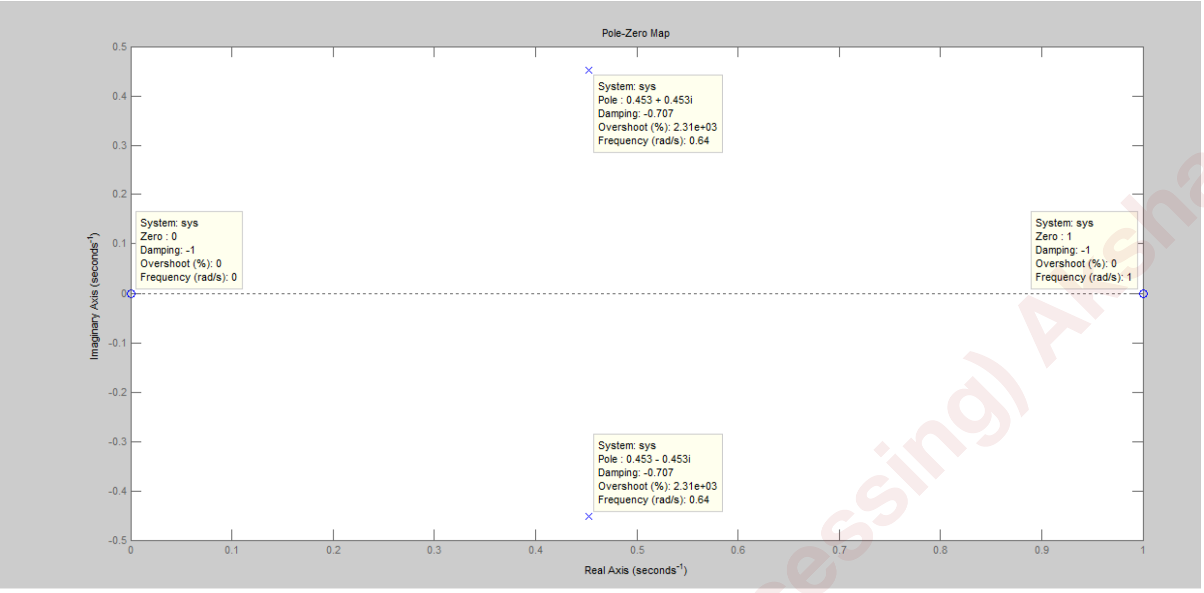
```
>> num1=roots(num)
num1 =
     1
>> den1=roots(den)
den1 =
   0.4526 + 0.4525i
0.4526 - 0.4525i
```

Now, commands for making the pole zero plot for Subpart 2:

Commands:

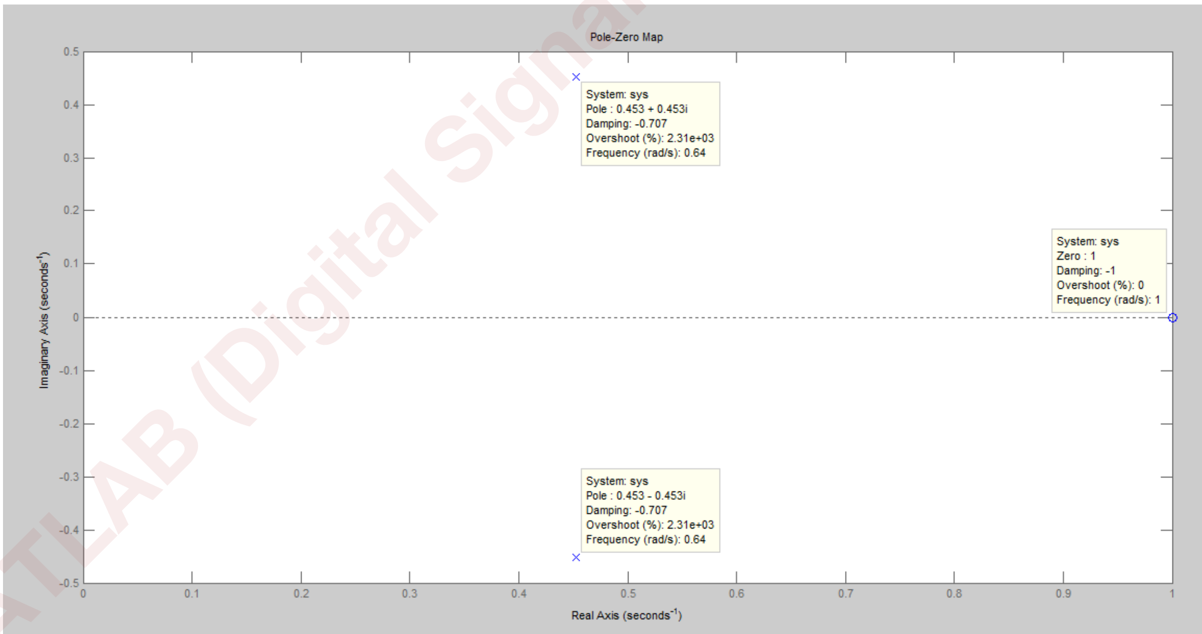
a. For TF1

>> pzmap(B1,A1)



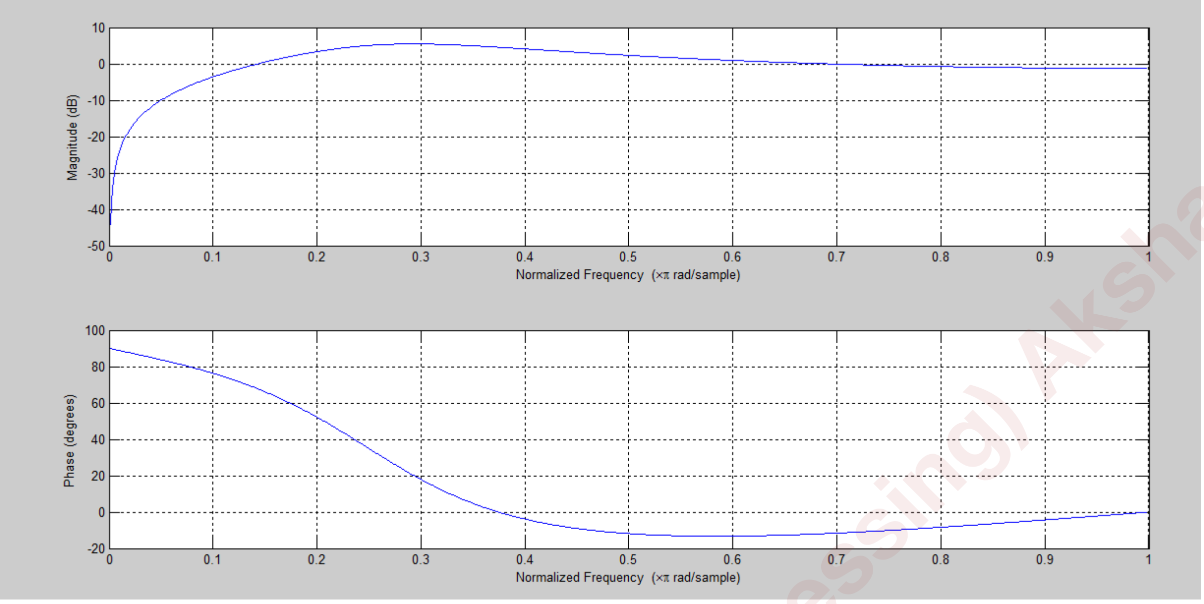
b. For TF2

>> pzmap(den1,num1)

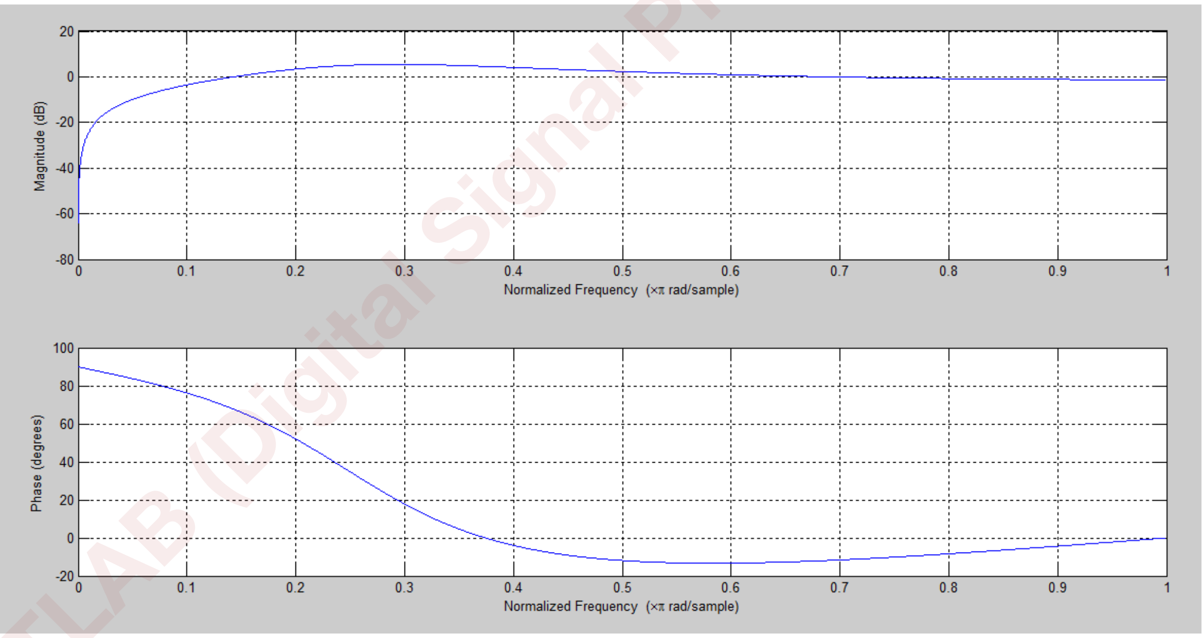


Part (d). Finding frequency response of DT system with a sampling frequency of

i. 1kHz
   Command: freqz(num,den,1000)



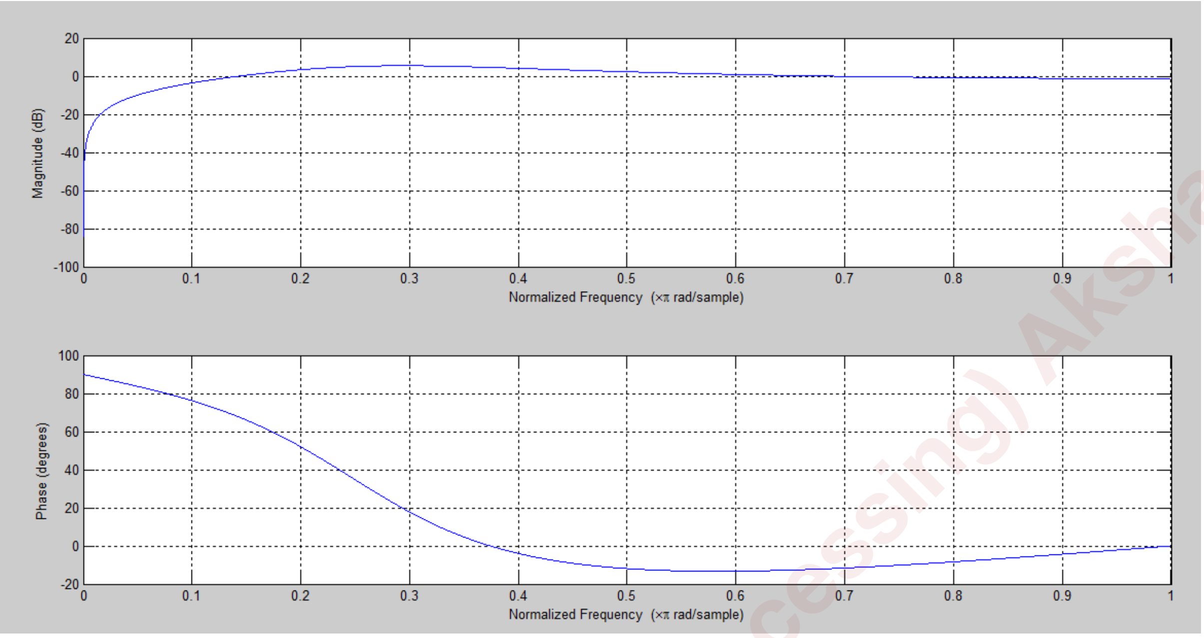
ii. 10kHz
   Command: freqz(num,den,10000)



iii. 100kHz

Command:

Freqz(num,den,100000)



Observation and Analysis –

It was found that the order of the given transfer function (in z domain) changes when the transfer function is rearranged from positive powers of z to negative powers of z.

So, TF1 (positive powers of z) has order =2 (w.r.t numerator)

And, TF2 (negative powers of z) has order =1 (w.r.t numerator)

Also, it was found that the pole zero plot using the coefficients of the transfer function as matrix and the poles and zeroes as matrix is same for both TF1 and TF2.

Finally,

In the frequency response plots, the plots for both TF1 and TF2 were found to be the same. And, on seeing the plots on different frequencies, the plots were nearly the same with little difference in the magnitude part of the plot on changing the frequency.

MATLAB Assignment 3
Digital Signal Processing

DSP – Lab
Assignment – 3

Q. Compare impulse invariant and bilinear z-transform methods in terms of
(a) the Nyquist effect on the magnitude, phase & group delay responses.
(b) Distribution for pole-zero diagrams.
Do this for LP, HP, BP & BS filters.

Case (1) (LP)

Given following specs:-

Passband	0 – 1 kHz
Stopband	3 – 5 kHz
passband ripple	1 dB
Stopband attenuation	60 dB
Sampling frequency	10 kHz

Now, we have to make
(a) Magnitude Plot
(b) Phase plot
(c) Group Delay
(d) Pole-zero diagram

for (i) Impulse Invariant & (ii) BZT.

(a) freqz (bz, az, fs)
(b) Phasedelay (bz, az, 512)
(c) grpdelay (bz, az, 512)
(d) zplane (bz, az)

(i) [bz, az] = impinvar (b, a, fs)
num den

(ii) [b1, a1] = bilinear (b, a, fs)

codes to convert analog to digital & plotting

Case (4) :- BS filter

Given specs :- passband 0 - 15 kHz
30 - 50 kHz
stopband 20 - 25 kHz
passband ripple 0.2 dB
stopband attenuation = 40 dB
sampling freq. = 100 kHz

Idea :- for any case,
first, it is told that we have to use Elliptical filter
Then, we are given specs.
So, just like in assignment 1, design an analog filter using these specs.
After that convert it into digital domain by using (i) Impulse invariant
(ii) BZT method.
} codes given above

8 Solving case (4)

Part (1) :- Designing analog elliptical filter.

Part (2) :- Converting analog to digital using

(i) Impulse invariant method
↳ Find plots (a), (b), (c), (d)

(ii) BZT method
↳ Find plots (a), (b), (c), (d)

Part (1)

Instruction (1)

[N1, Wn1] = ellipord(Wp, Ws, Rp, Rs, 's')

Wp = [15000/50000, 30000/50000]

Ws = [20000/50000, 25000/50000]

Rp = 0.2

Rs = 40.

fs = 100000

Instruction gives value of Wn1 & N1.

Wn1 = 0.3333 0.6.

N1 = 4.

Instruction (2)

[B1, A1] = ellip(N1, Rp, Rs, Wn1, 'stop')

(we get) B1 = . . . — —

A1 = — — —

Conversion to analog done

Part (2) :- (i) Impulse invariant method.

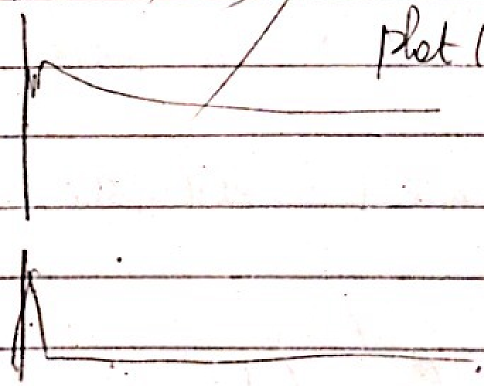
[bz, az] = impinvar (B1, A1, Fs)

(a) Magnitude plot

freqz(bz, az, Fs) / [H, F] = freqz(bz, az, 512, Fs)

plot (F, abs(H))

we get



(b) Phase plot

phasedelay(bz, az, 512)

we get

(c) Group delay

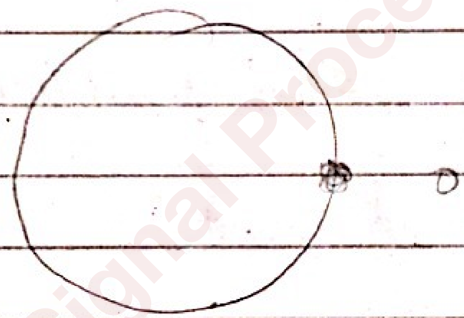
grpdelay(bz, az, 512)

we get :-

(d) Pole zero diagram

zplane(bz, az)

we get



(ii) BZT method

[b1, a1] = bilinear(B1, A1, Fs)

(a) Magnitude plot

freqz(b1, a1, Fs)

(b) Phase plot
phasedelay (b1, a1, 512)

(c) Group delay
grpdelay (b1, a1, 512)

(d) Pole zero diagram

zplane (b1, a1)

Assignment 3
Akshansh Chaudhary
ID – 2011AAPS300U
Dated - 7.10.2013
Question-

Given the specifications of a band stop filter, make it an analog filter and convert it to digital filter using impulse invariant method and Bilinear z transform method. Also, find the magnitude plot, phase plot, group delay and pole zero diagrams for both the methods.

Solution -

Part (1)
Making analog elliptical filter

Initializing the value of Wp,Ws,Rp, Rs for the elliptical filter

```
Rp=0.2
Rp =
   0.2000
>> Rs=40
Rs =
   40
>> Fs=100000
Fs =
     100000
>> Wp=[15000/50000, 30000/50000]
Wp =
   0.3000    0.6000
```

```
>> Ws=[20000/50000,25000/50000]
Ws =
    0.4000    0.5000
```

Instructions after initialization

```
>> [N1,Wn1]=ellipord(Wp,Ws,Rp,Rs,'s')
N1 =
     4
Wn1 =
    0.3333    0.6000

[B1,A1]=ellip(N1,Rp,Rs,Wn1,'stop')
B1 =
  Columns 1 through 3
    0.3198   -0.4313    1.3982
  Columns 4 through 6
   -1.2748    2.1536   -1.2748
  Columns 7 through 9
    1.3982   -0.4313    0.3198
A1 =
  Columns 1 through 3
    1.0000   -0.9838    1.9828
  Columns 4 through 6
   -1.4369    1.8869   -0.8671
  Columns 7 through 9
    0.7043   -0.2039    0.1459
```

Part (2)
Converting to Digital

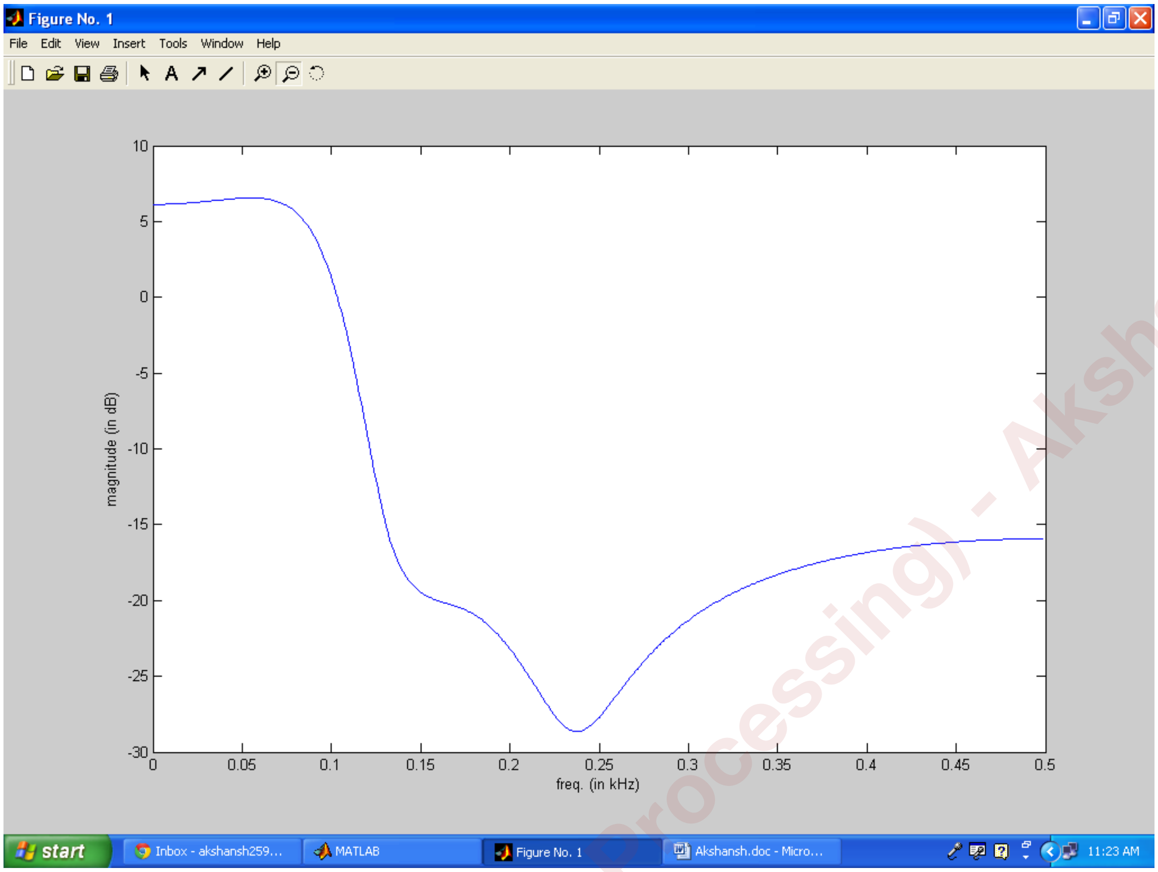
(i.) Impulse Invariant Method

```
[bz,az]=impinvar(B1,A1,Fs)
bz =
  1.0e-003 *
  Columns 1 through 8
    0.0029   -0.0238    0.0859   -0.1772    0.2284   -0.1882    0.0968   -0.0284
  Column 9
    0.0037
az =
  Columns 1 through 8
    1.0000   -8.0000   28.0000  -56.0001   70.0002  -56.0002   28.0001   -8.0000
  Column 9
    1.0000
```

(a.) Magnitude Plot

```
>> [H,F]=freqs(bz,az)
>> plot(F,20*log10(abs(H)))
>> xlabel('freq. (in kHz)')
>> ylabel('magnitude (in dB)')
```

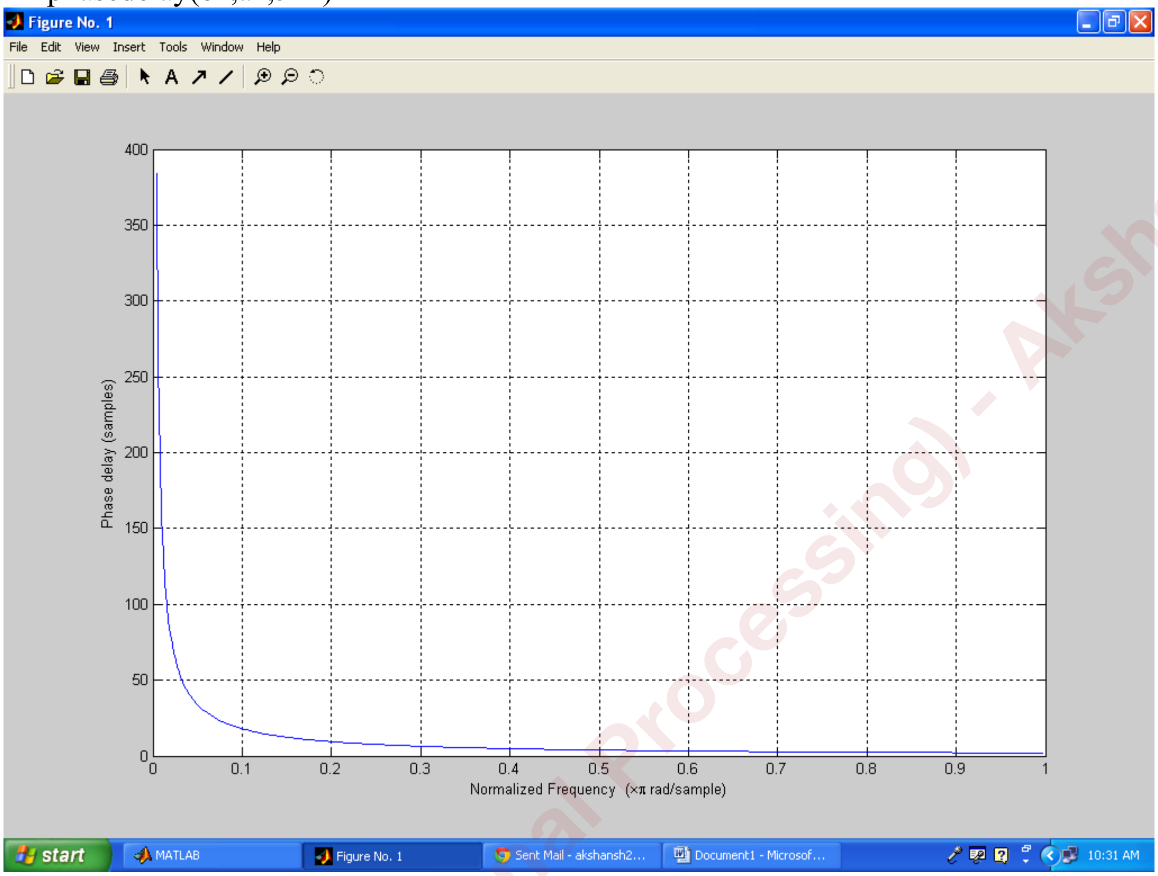
(Note: Graphs got in one version of MATLAB may differ in other versions)



Observations from the graph:
The stop band is between 0.2 and 0.25 kHz as required.
Pass band is between 0-0.15 kHz and then 0.3-0.5 kHz as in problem.
The pass band ripple is nearly 0.2 (after zooming the pass band area)
The stop band ripple has a little deviation. Its coming as 30 db (instead of 40 dB given.)
The problem becomes more ideal when the sampling frequency is increased.

(b.) Phase delay

>> phasedelay(bz,az,512)



(c.) Group Delay

>> grpdelay(bz,az,512)
>>

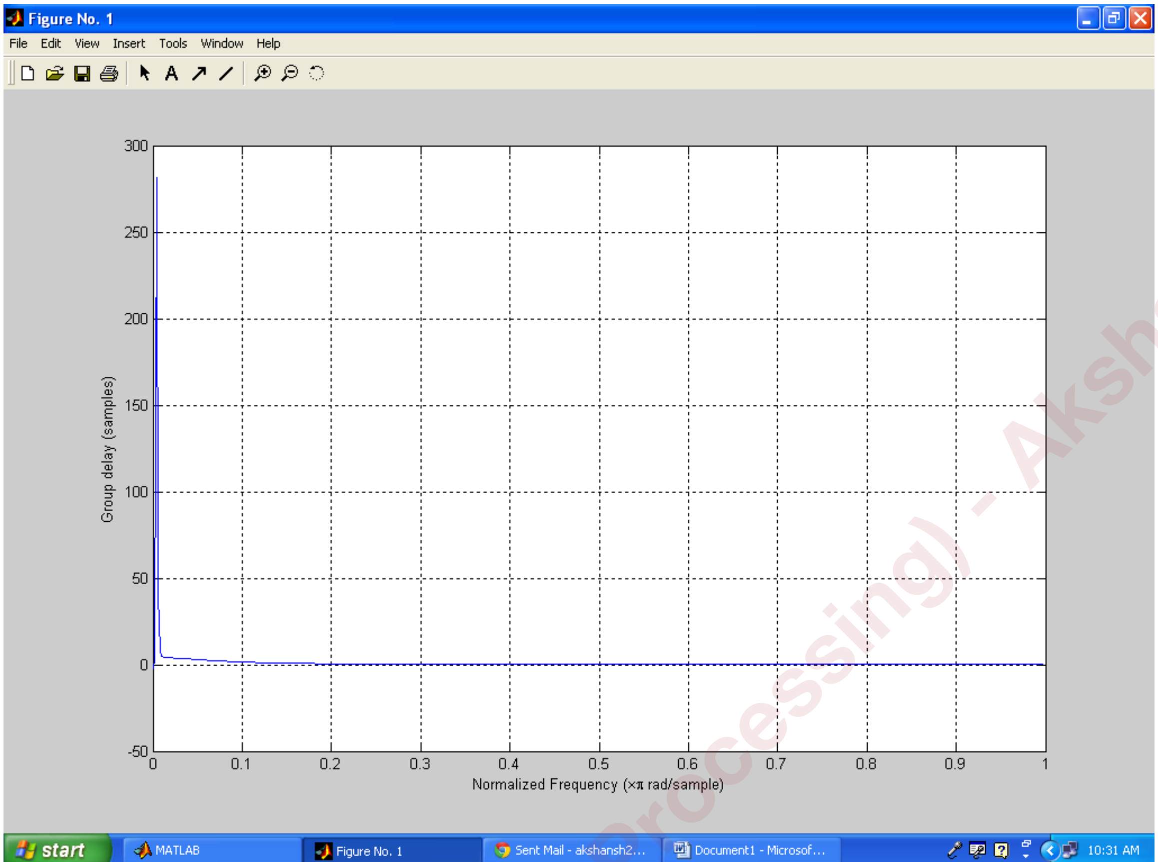


Figure No. 1
File Edit View Insert Tools Window Help
300
250
200
150
100
50
0
-50
Group delay (samples)
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
start
MATLAB
Figure No. 1
Sent Mail - akshansh2...
Document1 - Microsof...
10:31 AM

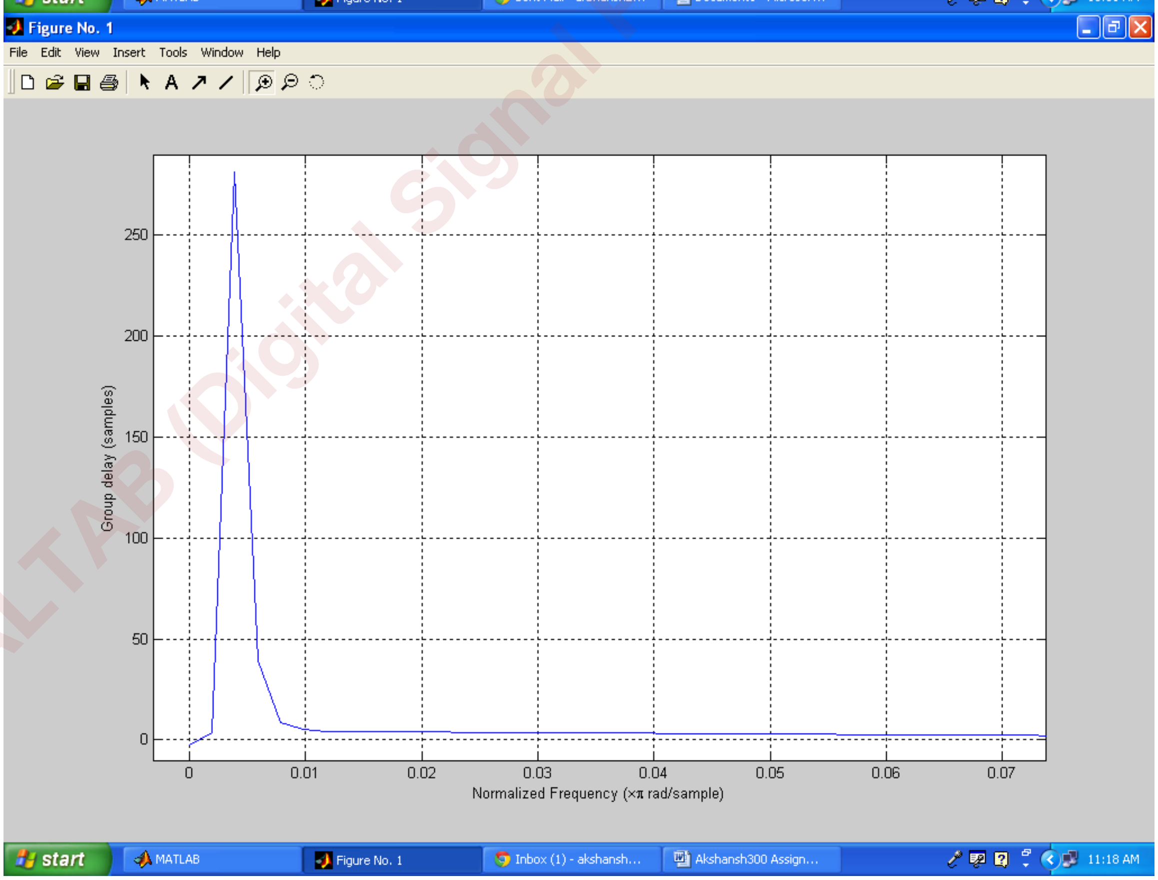
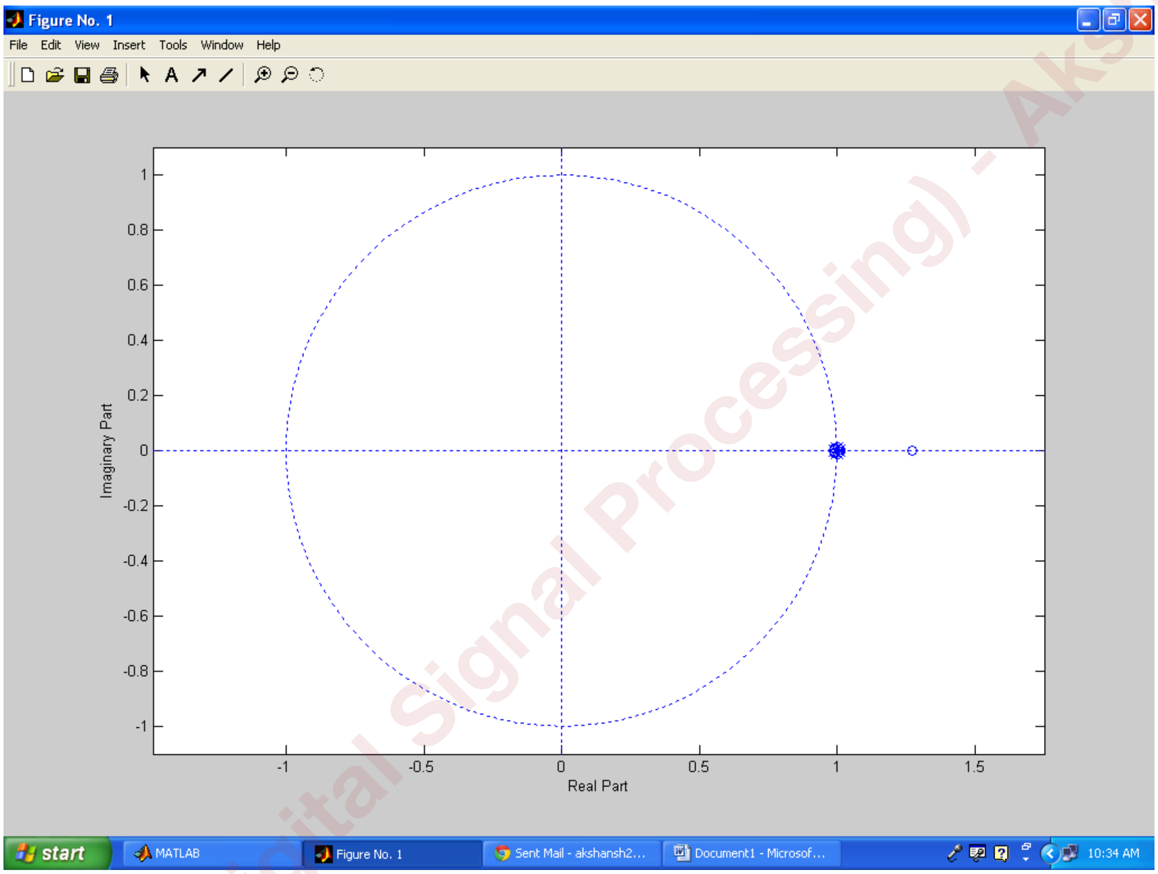
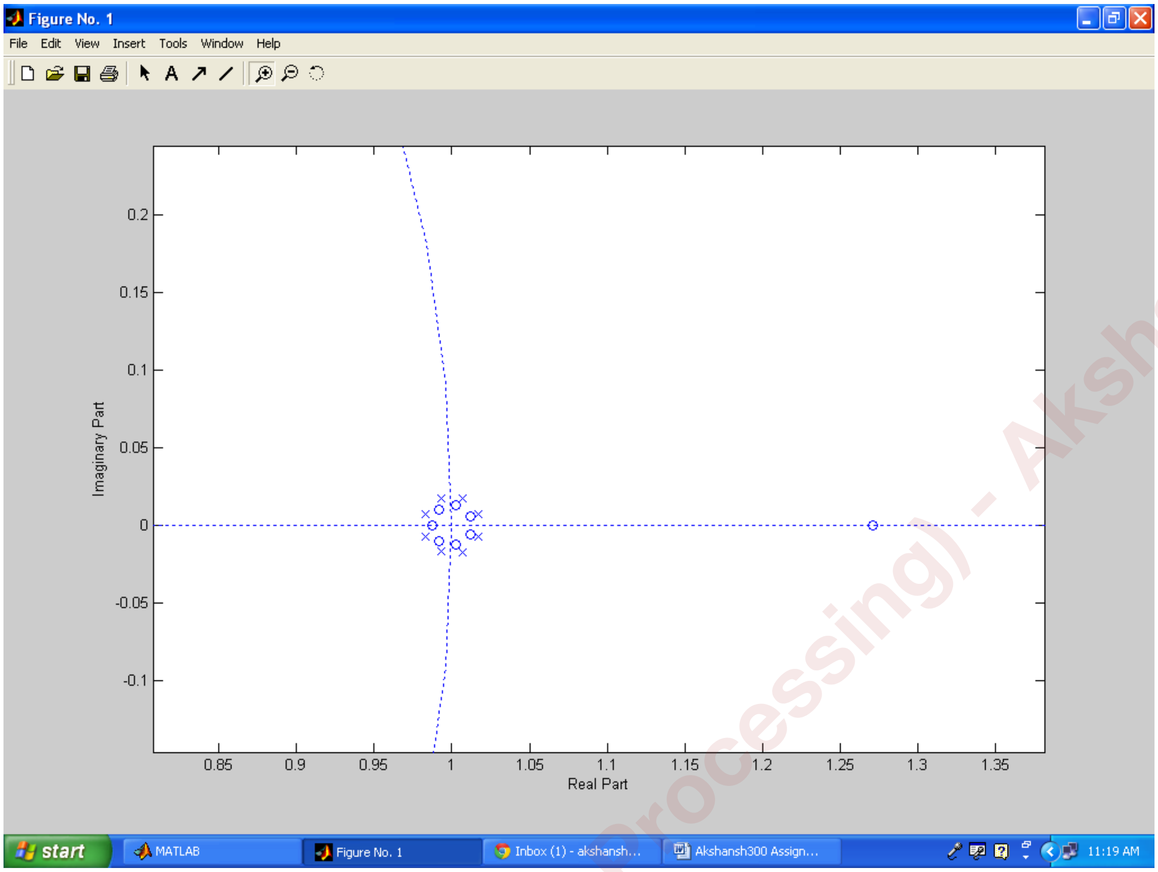


Figure No. 1
File Edit View Insert Tools Window Help
250
200
150
100
50
0
Group delay (samples)
0 0.01 0.02 0.03 0.04 0.05 0.06 0.07
Normalized Frequency (×π rad/sample)
start
MATLAB
Figure No. 1
Inbox (1) - akshansh...
Akshansh300 Assign...
11:18 AM

(d.) Pole zero diagrams

zplane(bz,az)



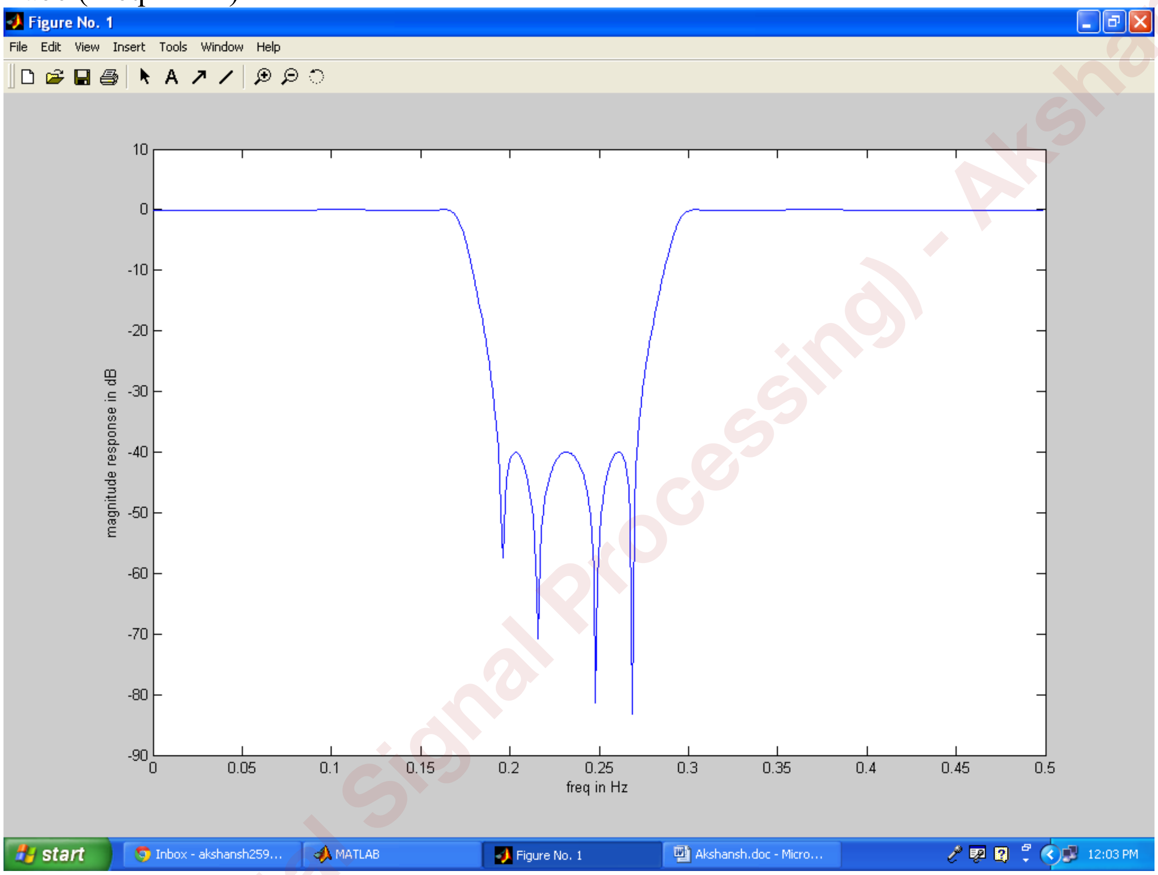


(ii.) BZT Method

```
>> [b1,a1]=bilinear(B1,A1,Fs)
b1 =
  Columns 1 through 8
    0.3652   -2.9218   10.2263  -20.4526   25.5658  -20.4527   10.2263   -2.9218
  Column 9
    0.3652
a1 =
  Columns 1 through 8
    1.0000   -8.0000   28.0000  -56.0001   70.0002  -56.0002   28.0001   -8.0000
  Column 9
```

(a) Magnitude Plot

```
>> [H,F]=freqz(B1,A1,512,Fs)
>> plot(F,20*log10(abs(H)))
>> ylabel('magnitude response in dB')
>> xlabel('freq in Hz')
```



(b.) Phase Plot

```
>> phasedelay(b1,a1,512)
>>
```

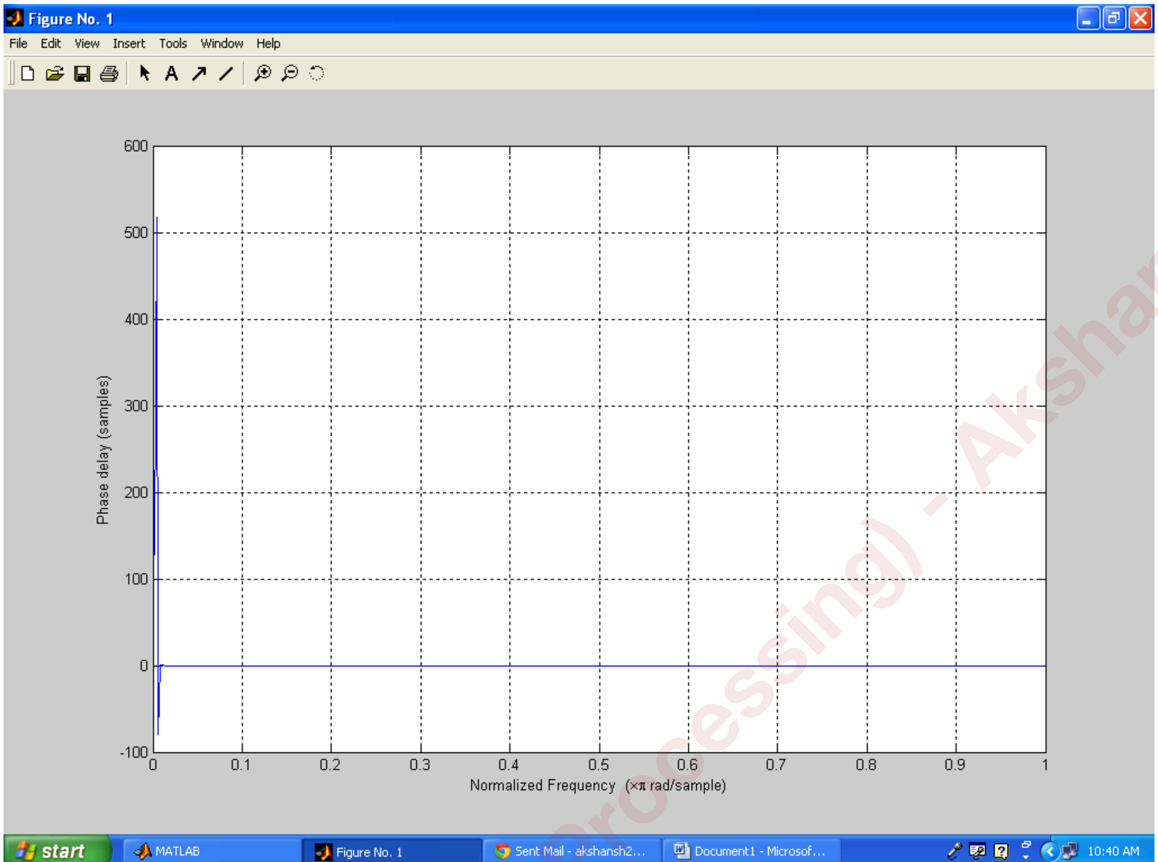


Figure No. 1
File Edit View Insert Tools Window Help
600
500
400
300
200
100
0
-100
Phase delay (samples)
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
start
MATLAB
Figure No. 1
Sent Mail - akshansh2...
Document1 - Microsof...
10:40 AM

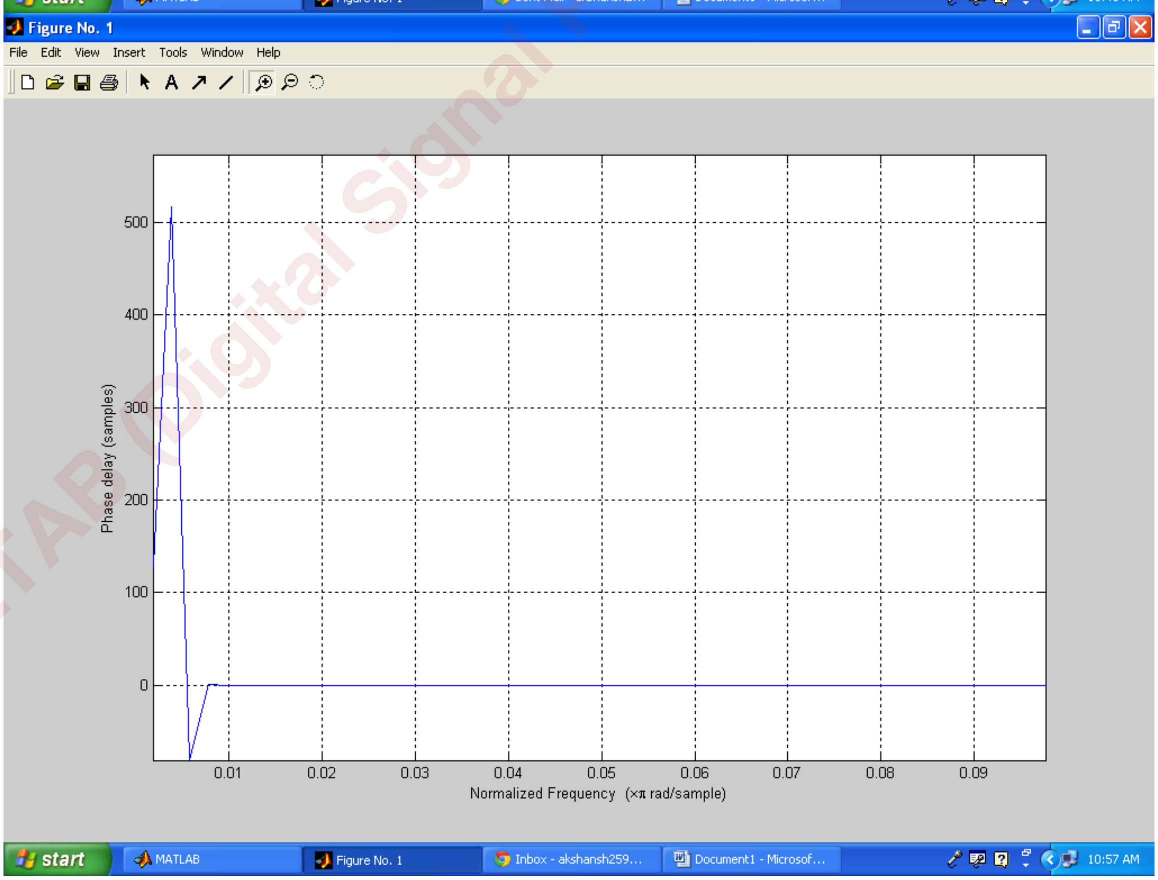
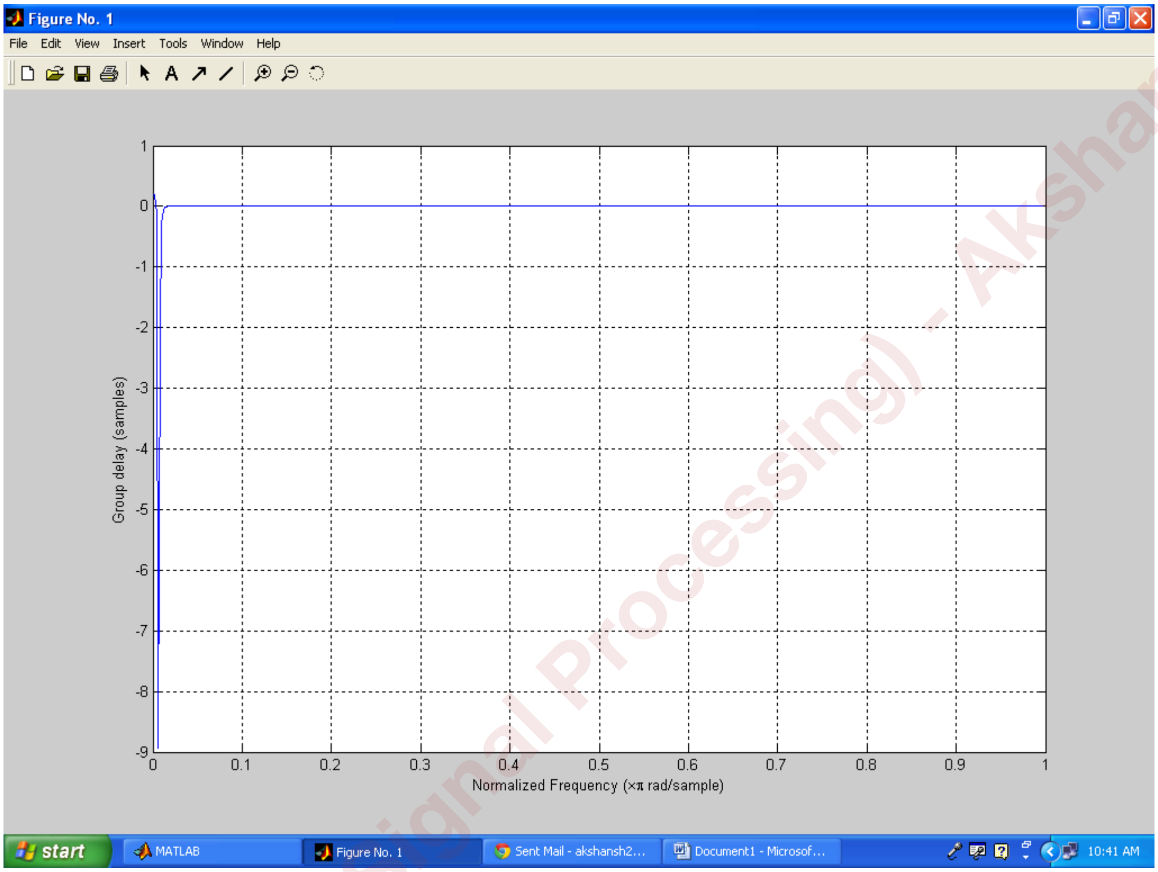
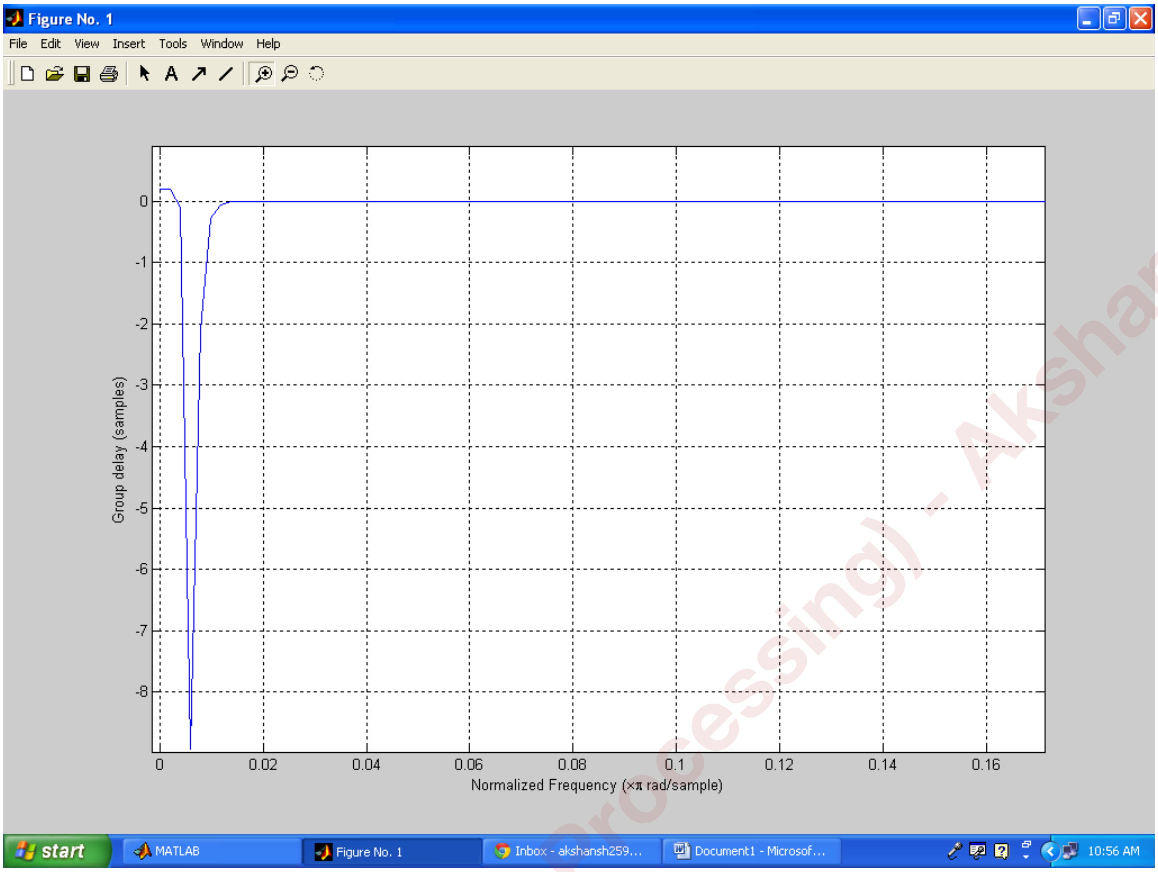


Figure No. 1
File Edit View Insert Tools Window Help
500
400
300
200
100
0
Phase delay (samples)
0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
Normalized Frequency (×π rad/sample)
start
MATLAB
Figure No. 1
Inbox - akshansh259...
Document1 - Microsof...
10:57 AM

(c.) Group Delay

>> grpdelay(b1,a1,512)





(d.) Pole Zero Plot

```
>> zplane(b1,a1)
```

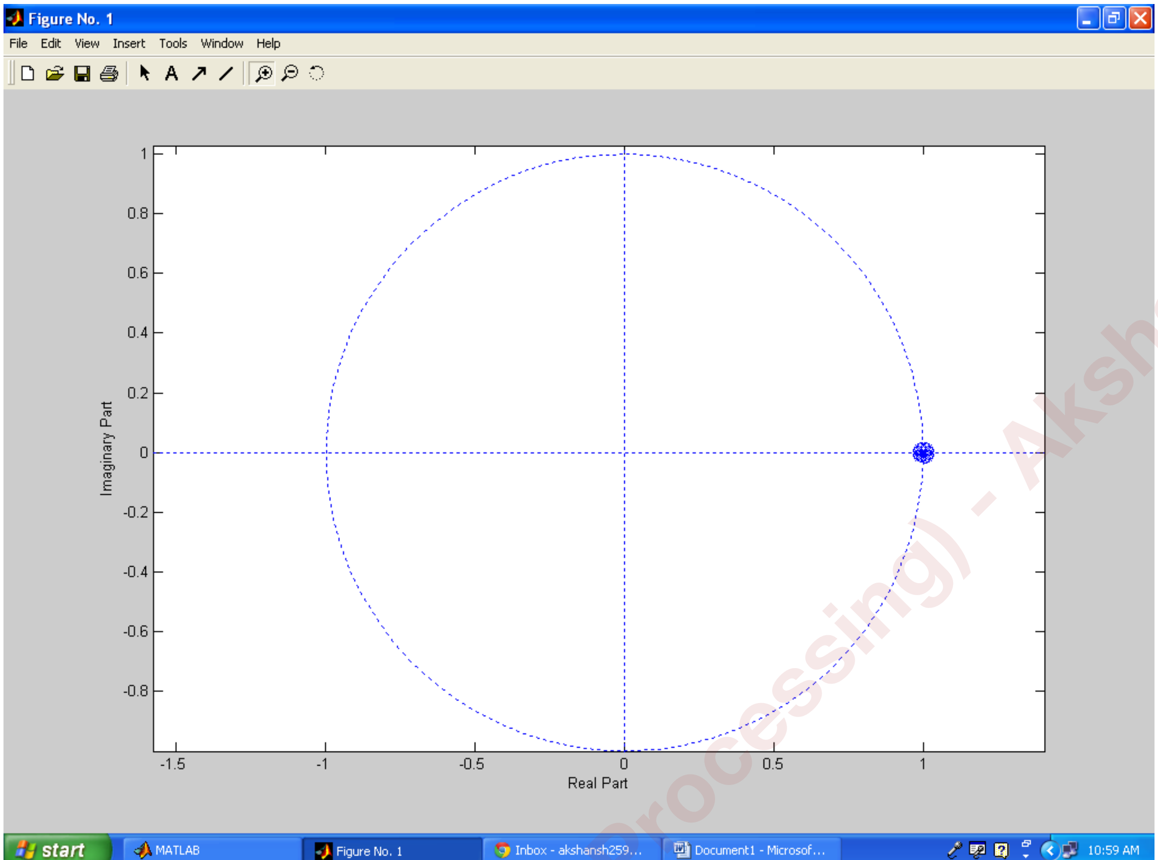


Figure No. 1
File Edit View Insert Tools Window Help
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.5
-1
-0.5
0
0.5
1
start
MATLAB
Figure No. 1
Inbox - akshansh259...
Document1 - Microsof...
10:59 AM

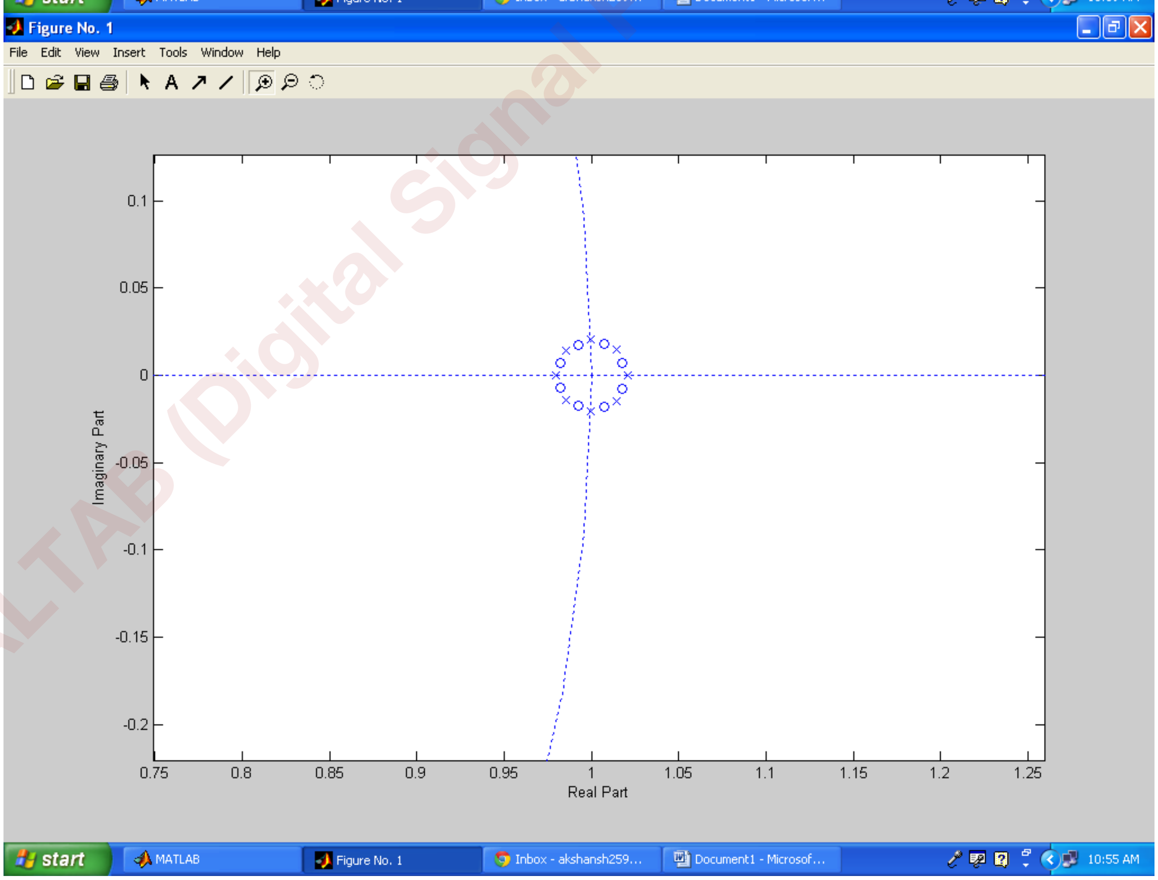
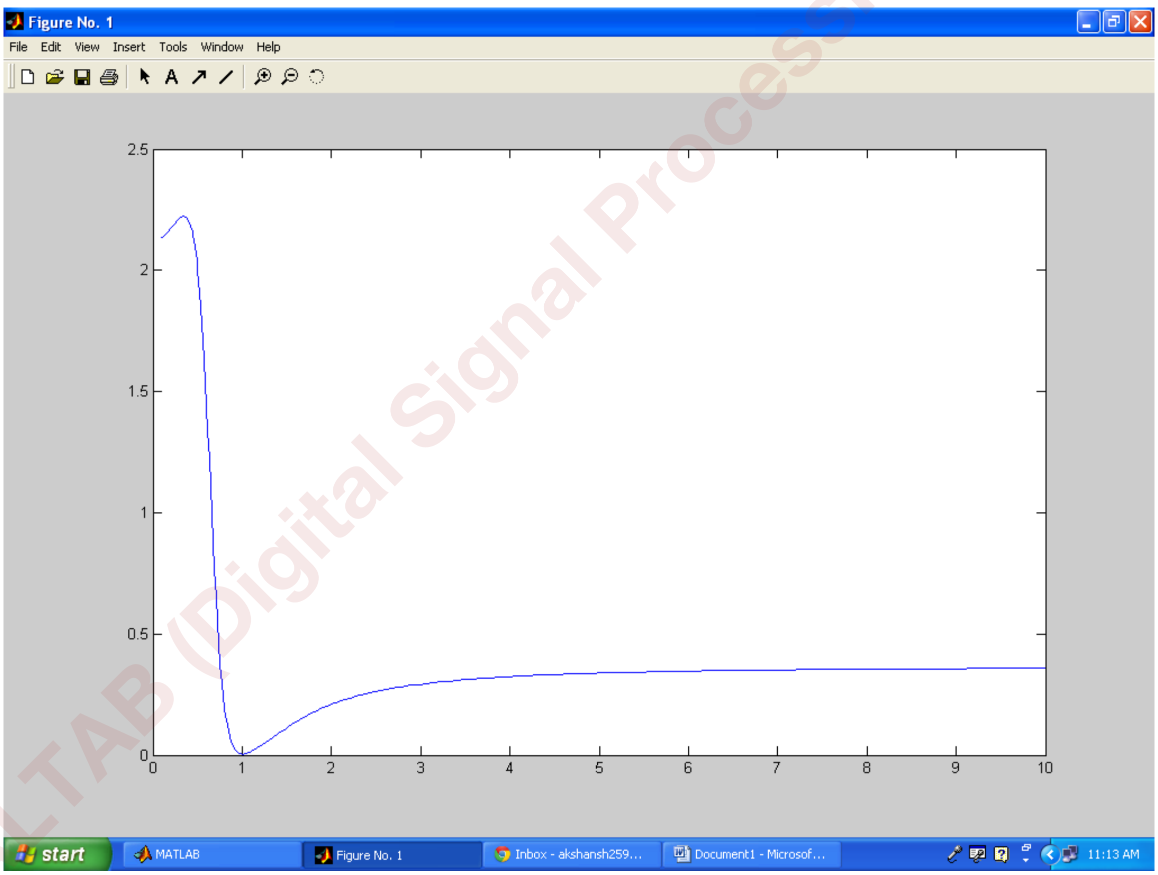


Figure No. 1
File Edit View Insert Tools Window Help
Imaginary Part
Real Part
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.75
0.8
0.85
0.9
0.95
1
1.05
1.1
1.15
1.2
1.25
start
MATLAB
Figure No. 1
Inbox - akshansh259...
Document1 - Microsof...
10:55 AM

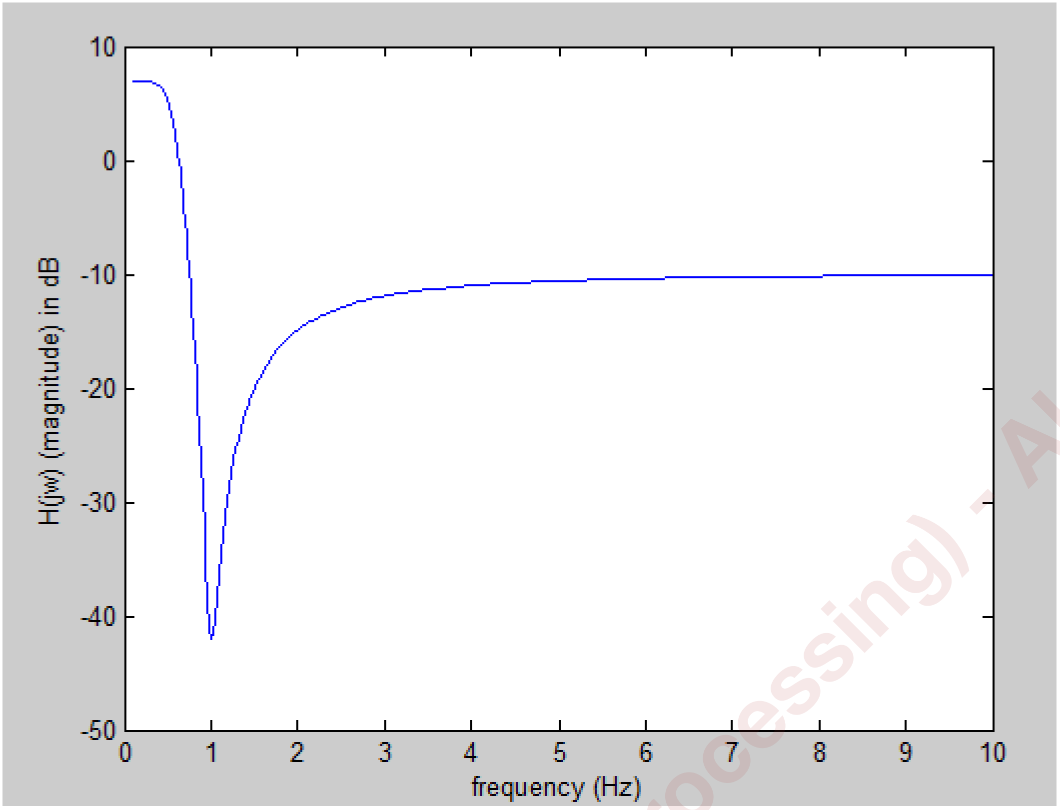
Showing that we have Band Stop Filter

(Seeing in s domain)

```
>> [N1,Wn1]=ellipord(Wp,Ws,Rp,Rs,'s')

>> [B1,A1]=ellip(N1,Rp,Rs,Wn1,'stop')

>> [H,F]=freqs(B1,A1)

>>plot(F,abs(H))
```




```
>> plot(F, 20*log10(abs(H)))
>> ylabel('H(jw) (magnitude) in dB')
>> xlabel('frequency (Hz)')
```



The figure shows the magnitude plot for elliptical filter with the given specifications in the ANALOG DOMAIN.
From the figure, it is visible that it's a band stop filter.

Conclusion

In the analog domain, the sampling interval is from 0 to infinity.
But, as we change it to the digital domain, the sampling interval changes form 0 to the sampling frequency. Basically, this reduces the sampling interval, or, in a way, compresses it.
In such a condition, we find distortion in the magnitude, phase, group and phase delay plots.
This effect is called as **Nyquist effect.**

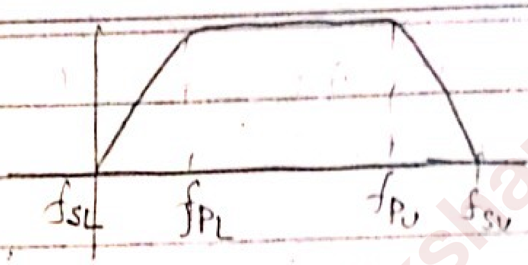
Matlab Assignment 4
Digital Signal Processing

Q.7.28 (1) Assignment - 4

(1) Doing on paper

We are given $f_{PL} = 200\,Hz$

$f_s = 2000\,Hz$; $f_{SU} = 500\,Hz$,



Now f can't be -ve. So, assume $f_{SL} = 0\,Hz$

Hence, Trans^n width $= f_{PL} - f_{SL} = 200\,Hz$.

So, $||^{ly}$, $f_{SU} - f_{PU} = 200\,Hz$

$\Rightarrow 500 - f_{PU} = 200\,Hz$

$\Rightarrow f_{PU} = 300\,Hz$.

So,

let $f_{c1} = \frac{(200-100)\,Hz}{(2000)}$ & $f_{c_2} = \frac{(300+100)\,Hz}{(2000)}$ — Smearing effect; Normalizing wrt f_s

Given Bandpass filter, so,

$$h_D(n) = \begin{cases} 2f_2 \frac{\sin(n\omega_2)}{n\omega_2} - 2f_1 \frac{\sin(n\omega_1)}{n\omega_1} & ; n \neq 0 \\ 2(f_2 - f_1) & ; n = 0 \end{cases}$$

& , for hamming window,

$$\omega(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right)$$

Now, finding coefficients,

$$h(n) = h_D(n) \times \omega(n)$$

for 7 point, $n = 0, 1, 2, 3$

(∵ its symmetric)

so don't find for 4, 5, 6

$$\because \begin{cases} h(0) = h(6) \\ h(1) = h(5) \\ h(2) = h(4) \\ h(3) = h(3) \end{cases}$$

for $n = 0$

$$h_D(0) = 2\left(\frac{300+100}{2000}\right) - \left(\frac{200-100}{2000}\right) = h_D(6)$$

$$\omega(0) = 0.54 + 0.46(1) = 1 = \omega(6)$$

Now

$$h(0) = h_D(0) \times \omega(0) = h(6)$$

So, IIly, find all coeffs.

Taking Z transform of $h(n)$, we get $H(z)$ values. Now, we have $\omega(0) \ldots \omega(6)$

We know

$$\omega = 2\pi f$$

So, $f(0) \ldots f(6)$. ✓

Now, plot for

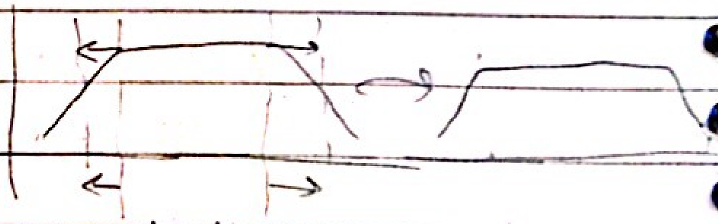
$H(z)$ vs f values.

we should get something like this

☆ Smearing effect

Δf = Trans^n width = 200

$\frac{\Delta f}{2} = \frac{1}{2}(Tw) = \frac{200}{2} = 100.$



extending PB. to include smearing effect.

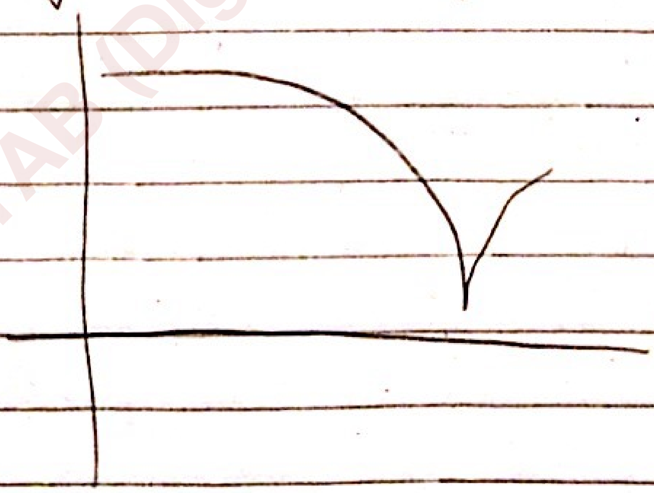
$$f(c_1)_{new} = 200 - \frac{\Delta f}{2} = 100$$

$$f(c_2)_{new} = 300 + \frac{\Delta f}{2} = 400.$$

(2) Doing in MATLAB

```
fs = 2000;          % sampling freq
fN = fs/2;          % Nyquist freq
N = 7
fc1 = 100/fN;
fc2 = 400/fN;
FC = [fc1 fc2];
hn = fir1(N-1, FC, hamming(N));
[H,f] = freqz(hn, 1, 512, fs);
mag = 20*log10(abs(H));
plot(f, mag), grid on
xlabel('Freq (HZ)');
ylabel('Magnitude Response (dB)');
```

Using this, we get a graph



corresponding to $N = 7$.

Now, changing N.

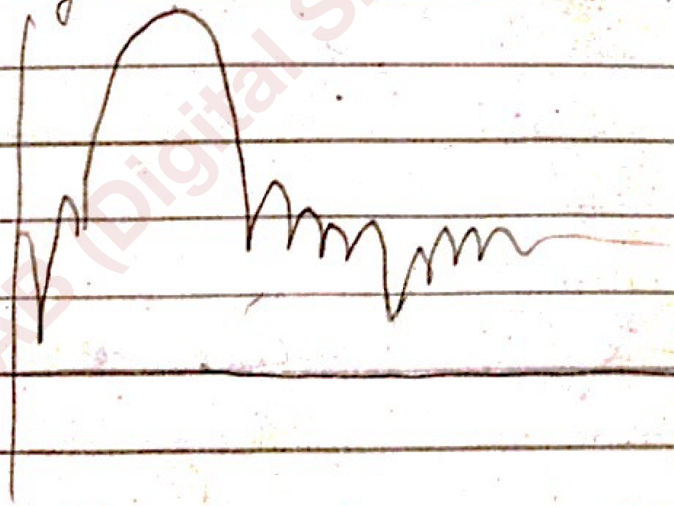
we have Δf = transⁿ width = 200.

& $\Delta f = \frac{3.3}{N} = \frac{200}{2000}$

$\frac{200}{f_s}$ → sampling freq

$\Rightarrow N = \frac{3.3}{200} \times 2000 = 33$

So, Using same commands for N=33, we get

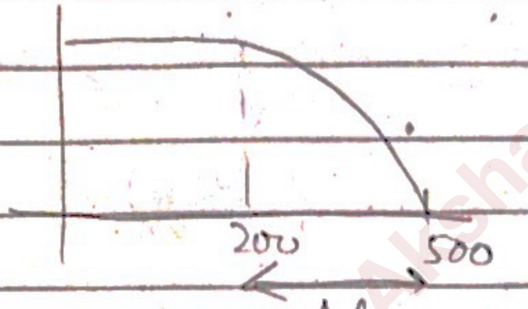


Band pass ✓

(Extra):

Assuming low Pass from given specs

```
fs = 2000;
N = 7;
fn = fs/2;
fc = 450/1000;
hn = fir1(N-1, fc, hamming(N));
[H, f] = freqz(hn, 1, 512, fs);
mag = 20*log10(abs(H));
plot(f, mag); grid on;
```



$\Delta f = 300$

So, including smearing effect,

$f_c = \left(f_{c2}\right) = \dfrac{\Delta f + \frac{\Delta f}{2}}{f_s/2}$ new passband

$= \dfrac{300 + 150}{2000/2}$

$\Rightarrow f_c = \dfrac{450}{1000}$

So, $N = \dfrac{3.3}{300} \times 2000 = 22$

We got

for N = 7 N = 8

for N = 22

Q. 7.29 given $N = 41$

$$H(f) = \begin{cases} 1 & f \in [2\,kHz, 4\,kHz] \\ 0 & \text{otherwise} \end{cases}$$

⇓

$f_{P_1} = 2000\ Hz$
$f_{P_2} = 4000\ Hz$
(without including
· normalizⁿ
· Smearing effect)

Assume sampling freq = 10 kHz
(not given).

Now,

Case (1) :- Hamming ★

So, $\Delta f \Big|_{\text{normalised}} = \frac{3.3}{N} = \frac{3.3}{41} = 0.0804$

& $\Delta f \Big|_{\text{denormalised}} = 0.0804 \times 10\ kHz$
$\simeq 804\ Hz$.

So, $\frac{\Delta f}{2} = 402\ Hz$

Now, including normalizⁿ & smearing effect:

$f_{P_1} = \frac{2000 - 402}{(10000/2)} = 0.3196$

Taking Nyquist effect

$f_{P_2} = \frac{4000 + 402}{(10000/2)} = 0.8804$

Now, plotting for BP, using commands

```
fs = 10000;
N = 41;
fp1 = 0.3196
fp2 = 0.8804
fp = [fp1 fp2];
hn = fir1(N-1, fP, hamming(N));
[H, f] = freqz(hn, 1, 512, fs);
mag = 20*log10(abs(H));
plot(f, mag, 'magenta'); grid on;
```

Case (2) :- Rectangular

$$\Delta f \Big|_{normalised} = \frac{0.9}{N} = \frac{0.9}{41} = 0.02195$$

$$\Delta f \Big|_{denormalised} = \frac{0.9}{41} \times 10000 = 219.512$$

$$\frac{\Delta f}{2} = 109.756$$

So, including normalisⁿ & smearing effect

$$f_{c1} = \frac{2000 - 109.756}{(10000/2)} = 0.37804$$

$$f_{c2} = \frac{4000 + 87.804}{(10000/2)} = 0.81756$$

→ Taking Nyquist effect

Now, plotting BP using commands;

$f_s = 10000;$

$N = 41$

↓

↓

```
fc1 = 0.37804
fc2 = 0.81756

fc = [fc1 fc2];

hn2=fir1(N-1, fc, rectwin(N));
[H2,f2] = freqz(hn2, 1, 512, fs);
mag2= 20* log10(abs(H2));

plot(f2, mag2, 'green'); grid on;
```

Assignment 4

MATLAB problems

7.28 Use MATLAB to compute the coefficients, plot the magnitude–frequency response in dB, and determine the locations of the zeros of each of the following window-based filters (assume a sampling frequency of 2 kHz and a Hamming window function):

(1) A 7-point, bandpass FIR filter with pass- and stopband edge frequencies of 200 Hz and 500 Hz.

(2) An 8-point, bandpass FIR filter with pass- and stopband edge frequencies of 200 Hz and 500 Hz.

In the given problem, it has been told that the pass band frequency is given as 200 Hz and stop band frequency is given as 500 Hz.
Now, as it's told to design a band pass filter, are left to assume it as:

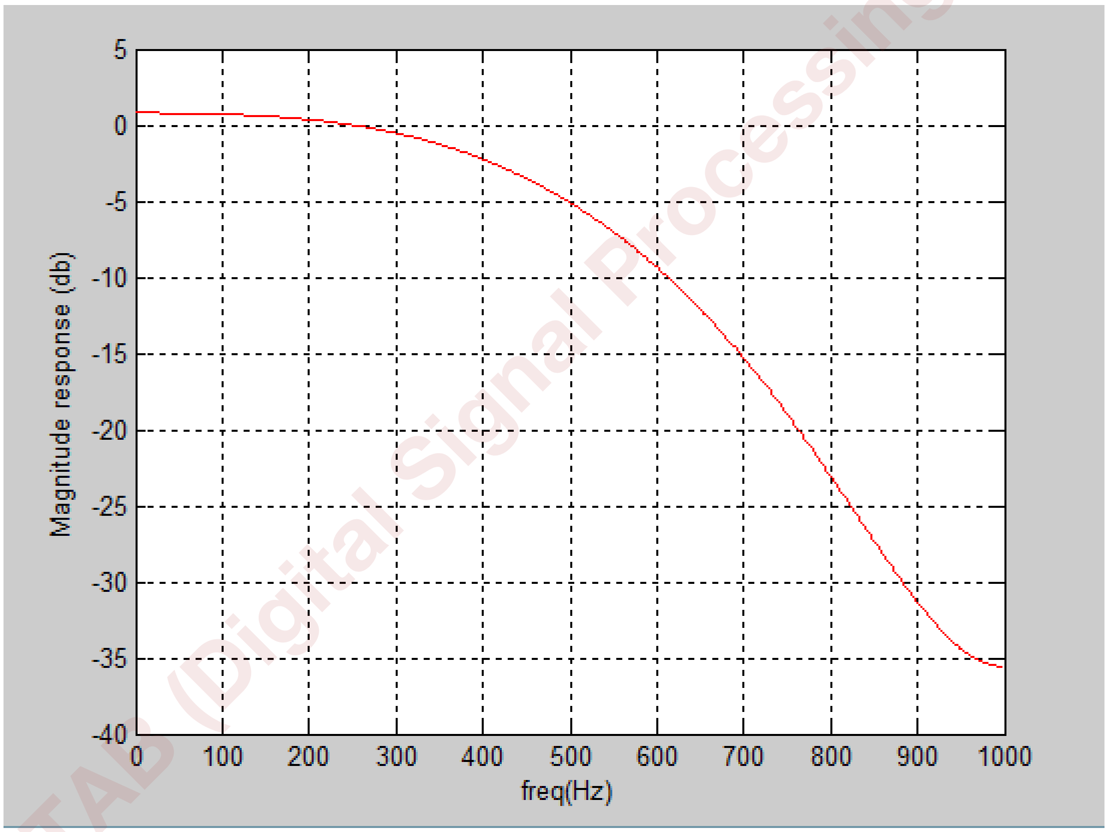
a. A low pass filter with pass band edge frequency of 200 Hz and stop band edge frequency of 500 Hz
b. A band pass filter with lower stop band frequency of 0 Hz, lower pass band frequency of 200 Hz and upper stop band frequency of 500 Hz.

Note: it is left to the designer to choose how to design the filter using the specs given.

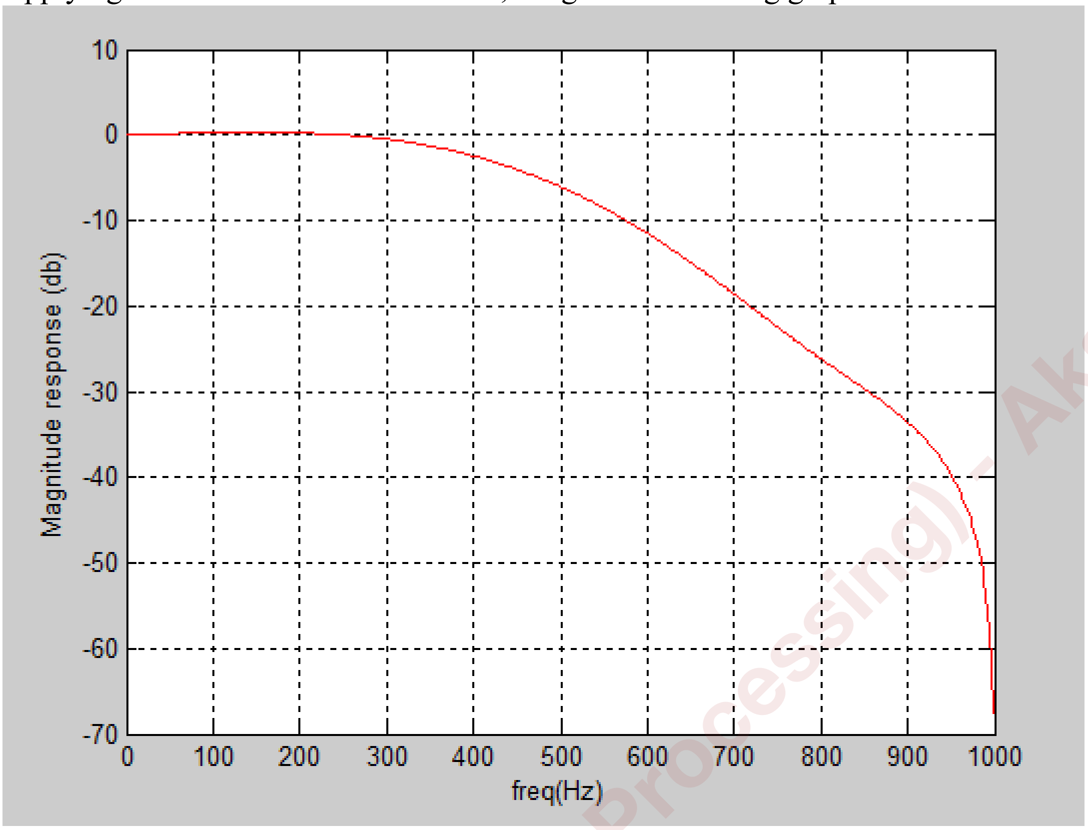
Now, assuming the case, b. we assume:
Wsl= 0 Hz
Wpl= 200 Hz
That means, the transition width (Wpl-Wsl)= Delta f (denormalised)=200 Hz
Wsu=500 Hz
So, Wpu= Wsu-200 Hz = 300 Hz

Commands:

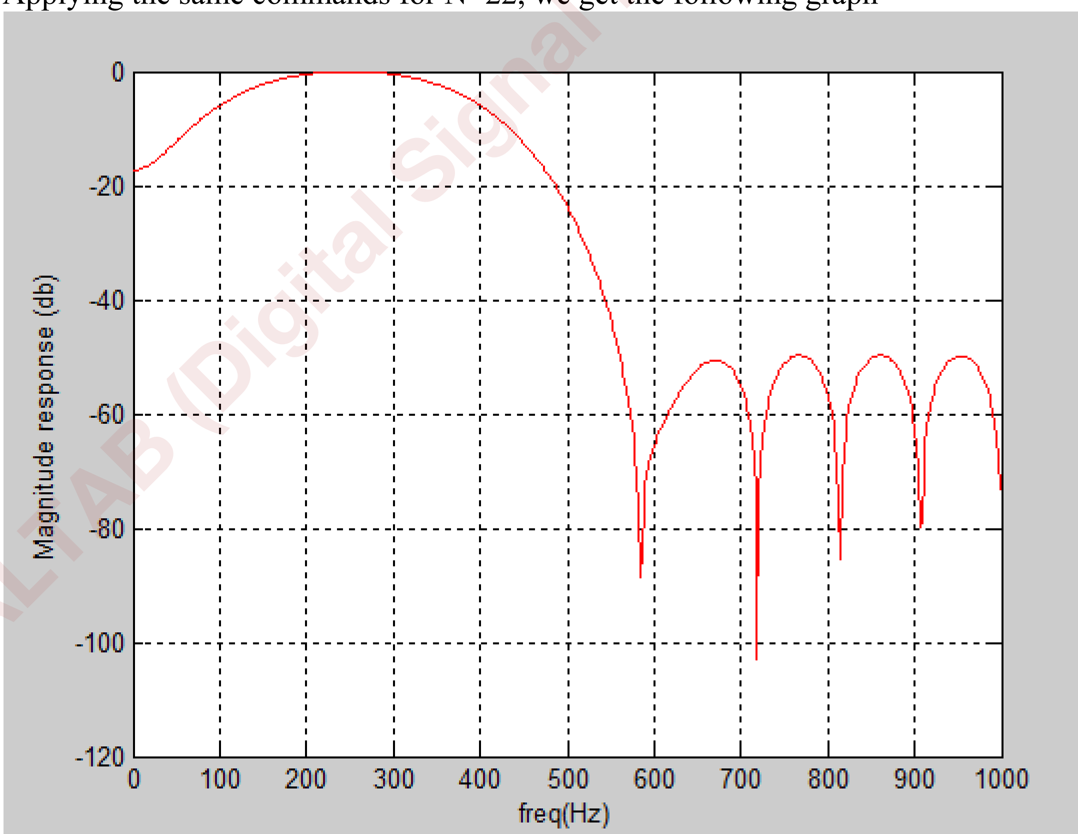
```
>> fs=2000;
>> fn=fs/2; % fn= Nyquist freq.
>> N=7;
>> fc1=100/fn; % 100, as its Wpl - (delta f/2); and /fn, as normalising w.r.t nyquist
>> fc2=400/fn; % 400, as its Wpu + (delta f/2)
>> FC=[fc1 fc2];
>> hn=fir1(N-1, FC, hamming(N));
>> [H,f]=freqz(hn, 1,512,fs);
>> mag=20*log10(abs(H));
>> plot(f,mag,'red');
grid on;
xlabel('freq(Hz)')
>> ylabel('Magnitude response (db)')
>>
```



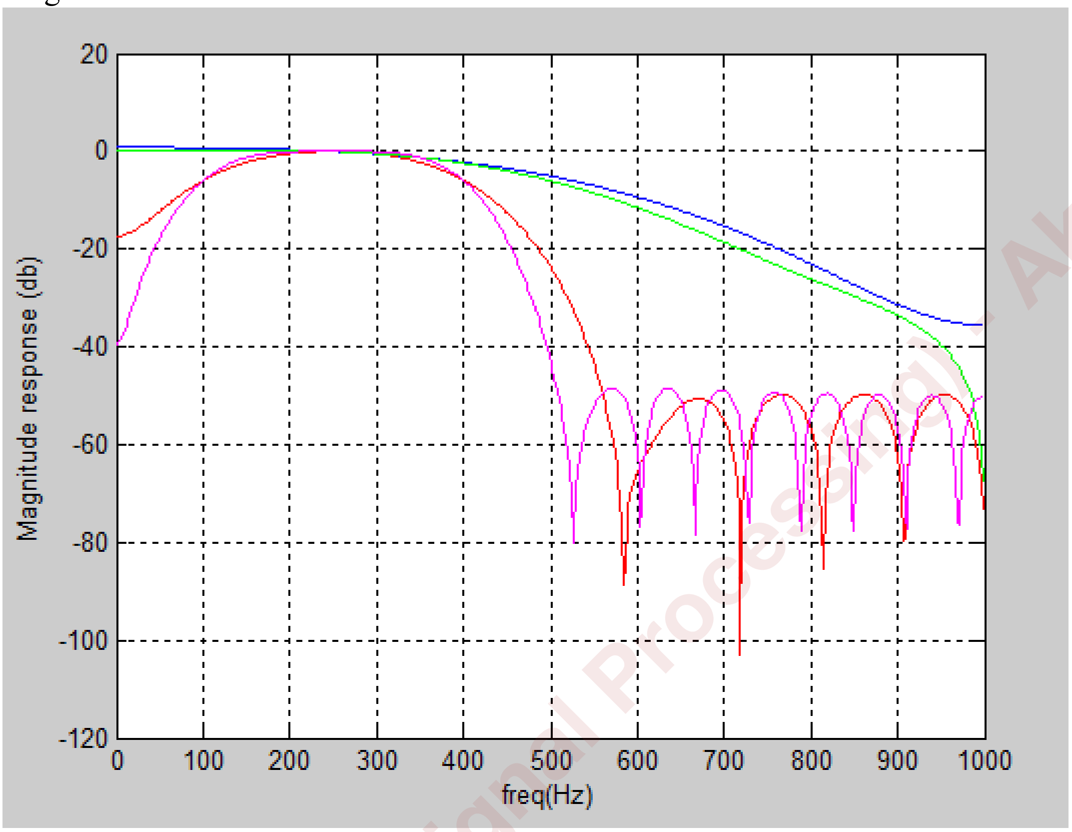
Applying the same commands for N=8, we get the following graph



Applying the same commands for N=22, we get the following graph



Proceeding similarly and then, holding all the graphs together, we see:
Blue: N=7
Green: N=8
Red: N=22
Magenta: N=33



Note: finding N by using delta f formula for hamming window
Delta f = 3.3/N
Where Delta f = 200/fs
Using this, we get N=33

7.29 A 41-point bandpass FIR filter is to be designed to approximate the following ideal magnitude response characteristics using the window method:

$$H(f) = 1 \quad 2\text{ kHz} \leq f \leq 4\text{ kHz}$$
$$0 \quad \text{otherwise}$$

Determine the impulse response coefficients of the filter and plot its magnitude and phase frequency responses with the aid of MATLAB for each of the following cases:

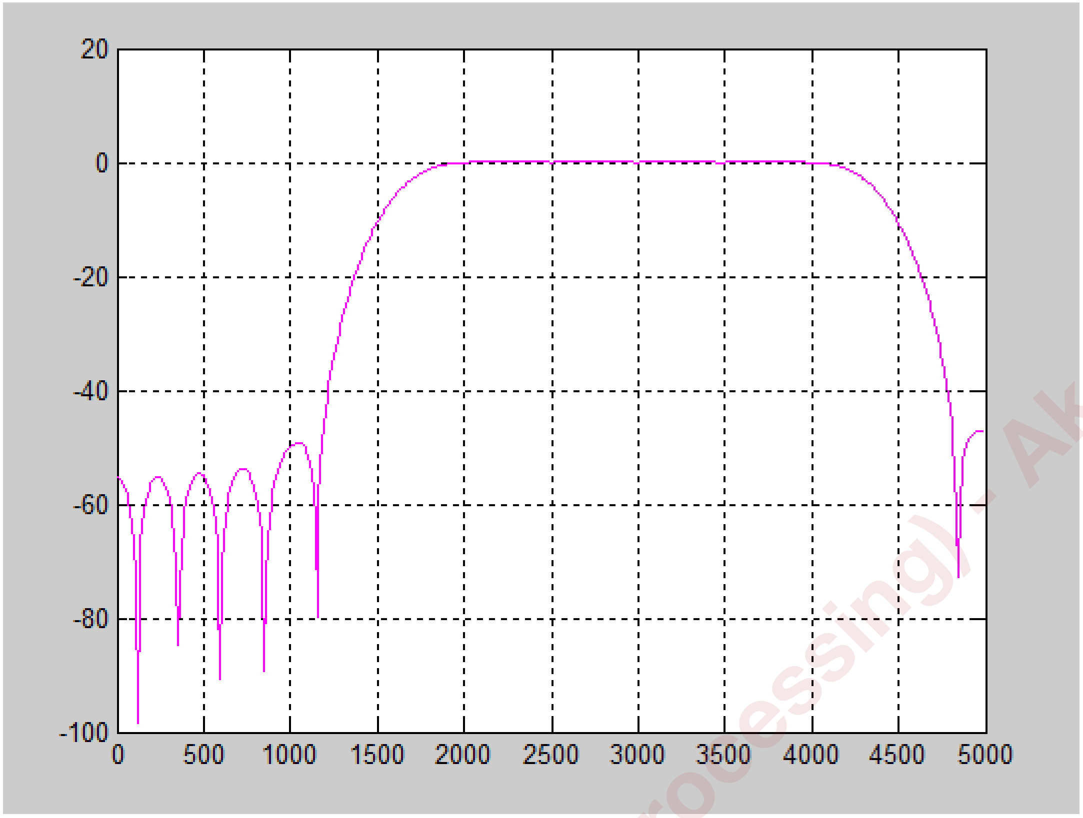
(1) Using a rectangular window.

(2) Using a Hamming window.

Clearly, here fpl=2kHz and fpu=4kHz.
For the given problem, assume the sampling frequency as 10kHz, as its not mentioned.
(We know that for denormalizing using Nyquist theorem, we have to divide the frequencies by (fs/2), where fs is the sampling frequency.
Also, the sampling frequency should be greater than 4kHz.)

For hamming window

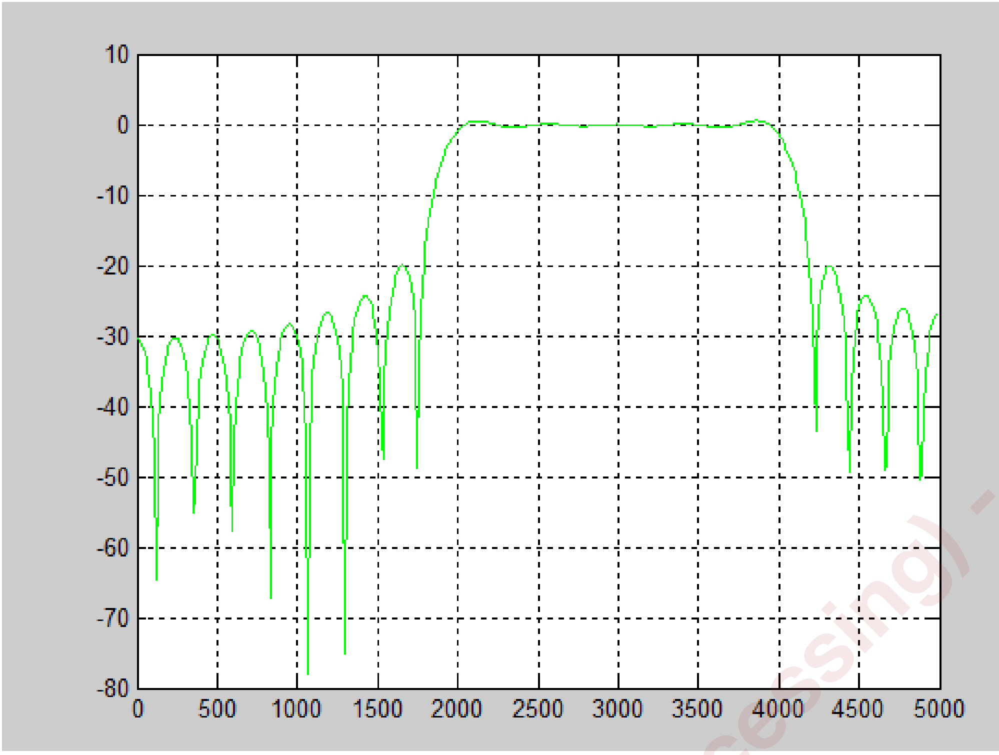
```
N=41;
>> fp1=0.3196; % value got after normalizing it w.r.t Nyquist, and using smearing effect
>> fp2=0.8804;
>> fs=10000;
>> fp=[fp1 fp2];
>> hn=fir1(N-1,fp,hamming(N));
>> hn=fir1(N-1,fp,hamming(N));
>> [H,f]=freqz(hn,1,512,fs);
>> mag=20*log10(abs(H));
>> plot(f,mag,'magenta'); grid on;
>>
```



Clearly, it is visible that the pass band is between 2000 to 5000 Hz (nearly)

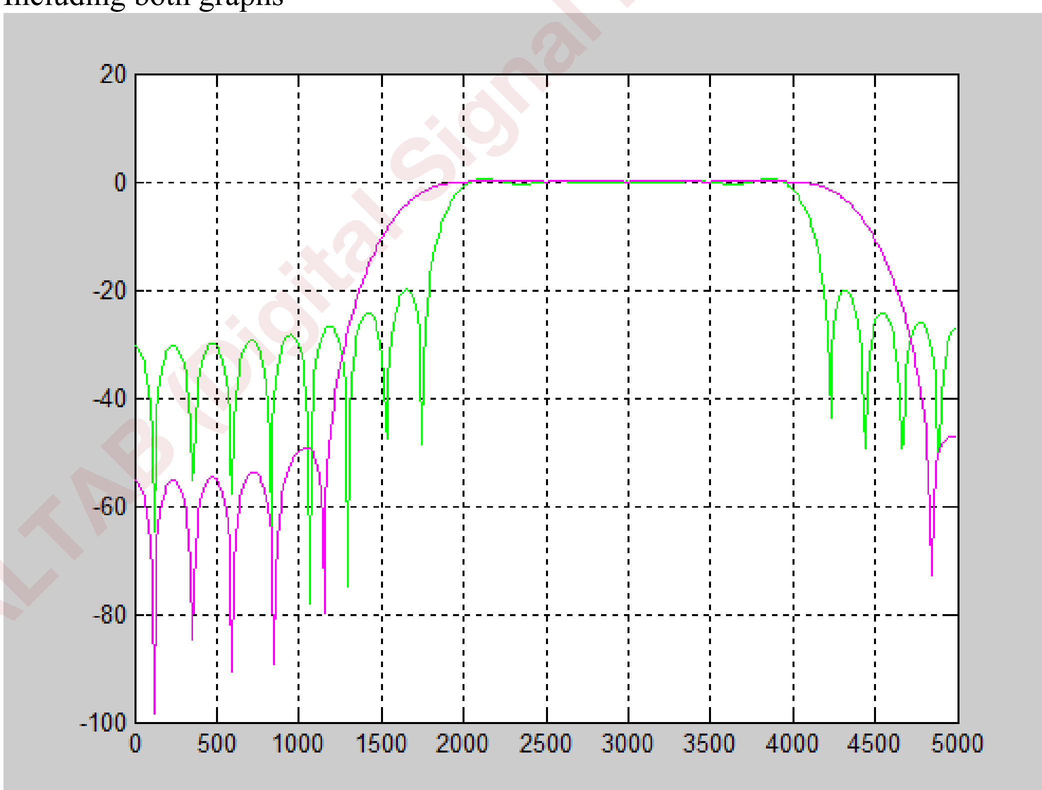
For rectangular window

```
>> fc1=0.37804;
>> fc2=0.81756;
>> fc=[fc1 fc2];
>> hn2=fir1(N-1,fc,rectwin(N));
>> [H2,f2]=freqz(hn2,1,512,fs);
>> mag2=20*log10(abs(H2));
>> plot(f2,mag2,'green');grid on;
```



It is also having pass band in 2000 to 5000 Hz.

Including both graphs –



Inference:
It can be clearly seen that increasing the no. of samples (N) makes the frequency response more ideal.
Moreover, the graph compresses as N is increased.

Matlab Assignment 5
Digital Signal Processing

classmate

DSP - MATLAB.
Assignment - 5.

Ifeacher : Q. 7.30

(LPF)

Given :- $N = 21$

$f_p = 2\ kHz$ } ⇒ $TW = \Delta f = 1\ kHz$.

$f_c = 3\ kHz$ }

$f_s = 10\ kHz$

Idea : Finding FIR filter coeff, use 'remez' command.

Syntax :-

b = remez (N-1, F, M)

N : filter length

F : Vector of normalised band edge freq.

M : " of desired magnitude response of filter

If weight is specified,

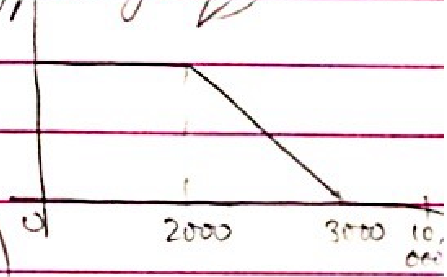
b = remez (N-1, F, M, WT)
↳ vector.

First, normalise given freq (wrt Nyquist freq)

i.e., 0 - 2 kHz PB

2 - 3 kHz Trans^n

3 - 10 kHz SB.



$$F = \left(\frac{0}{10000/2}, \frac{2000}{10000/2}, \frac{3000}{10000/2}, \frac{10000}{10000/2} \right)$$

So, F = [0, 0.4, 0.6, 1]

N = 21 ; % telling no. of filters

Fs = 10000 ;

Page No

Now, For ideal LPF, the magnitude is 1 → PB
0 → Transⁿ
0 → SB

So,
M = [1 1 0 0]

Now computing filter coeff,

b = remez(N-1, F, M);

Computing freq. response :-

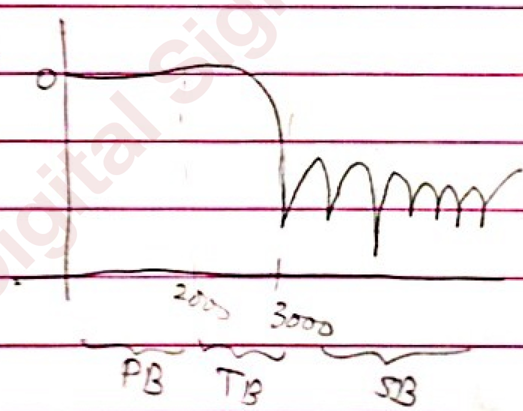
[H, f] = freqz(b, 1, 512, Fs)

Plotting freq response in dB.

% for Magnitude response

plot (f, 20* log 10(abs(H)));

we get :-



Now, use the command for finding all plots :

fvtool(b, 1);

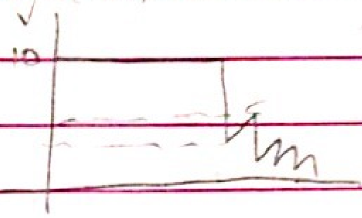
b → numerator coeffs

1 → ∵ den. eqⁿ is not there for FIR

It displays separate buttons & plots

classmate

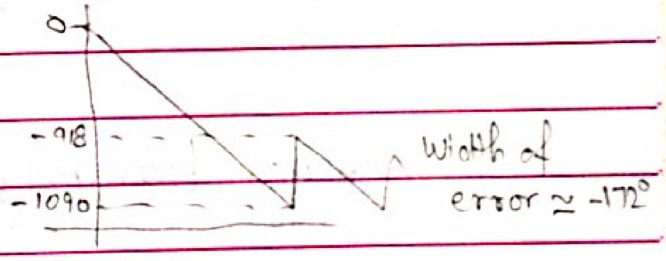
✓ For finding phase delay :
>> phasedelay (b, 1, 512)
We get



✓ Seeing impulse response :-
Command : impz(b)

✓ Pole zero plot (z-domain)
zplane (b, 1)
(s-domain)
pzmap(b, x)
x = [0 0 - - - 1]

✓ Phase response
phasez(b, 1)



✓ Group delay
grpdelay (b, 1, 512)

MATLAB Assignment 6
Digital Signal Processing

classmate

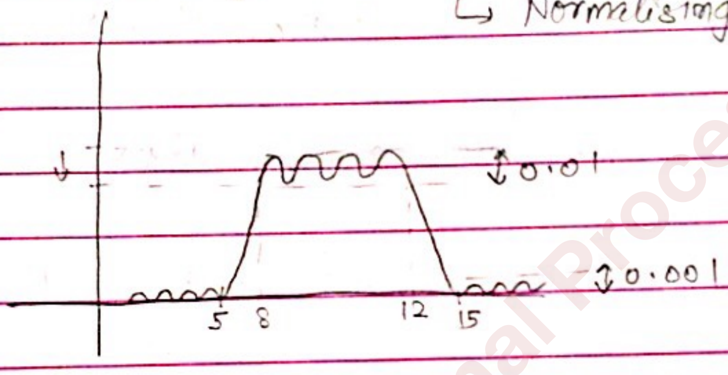
18.11.13

DSP LAB
Assignment - 6

Q.7.32 ✱ Given a linear phase FIR BANDPASS filter with:
- passband 8 - 12 kHz
- SB ripple 0.001
- PB ripple 0.01
- f_s 48 kHz

Δf = Tran^n width 3 kHz

↳ Normalising wrt Nyquist $= \frac{3}{48/2} = \frac{3}{24}$



(1) Using Hamming Window

Tran^n width $= \frac{3.3}{N} \Rightarrow \frac{3.3}{N} = \frac{3}{48}$ → Normalising

$\Rightarrow N = 1.1 \times 48 = 52.8$

So, n varies from -26 to 26.

Now, normalising all freq w.r.t Nyquist + including Smearing effect

$f_{SL} = \frac{5}{48/2}$

$f_{PL} = \frac{8}{48/2} - \frac{\Delta f}{2} = \frac{16}{48} - \frac{3}{48} = 13/48$;

$f_{PU} = \frac{12}{48/2} + \frac{\Delta f}{2} = \frac{24}{48} + \frac{3}{48} = 27/48$;

$f_{SU} = 15/48$;

Page No

Now, the cut off freq. will be f_{PL} & f_{PU}
This can be entered in an array,

fc = [fPL fPU];

N = 53; (filter length)
fs = 48000;

For making truncated impulse response, $h_D(n)$, use :-

hd = fir1(N-1, fc, boxcar(N));

[The same formula in theory becomes

$$h_D(n) = \begin{cases} 2f_2 \frac{\sin(n\omega_2)}{n\omega_2} - 2f_1 \frac{\sin(n\omega_1)}{n\omega_1}, & n \neq 0 \\ 2(f_2 - f_1), & n = 0 \end{cases}$$
]

Now, finding w(n)
use :-

wn = hamming(N);

[In theory, for hamming window, we use
$$\omega(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right)$$
]

Now, finding coeff, h(n)
use :-

hn = fir1(N-1, fc, wn);

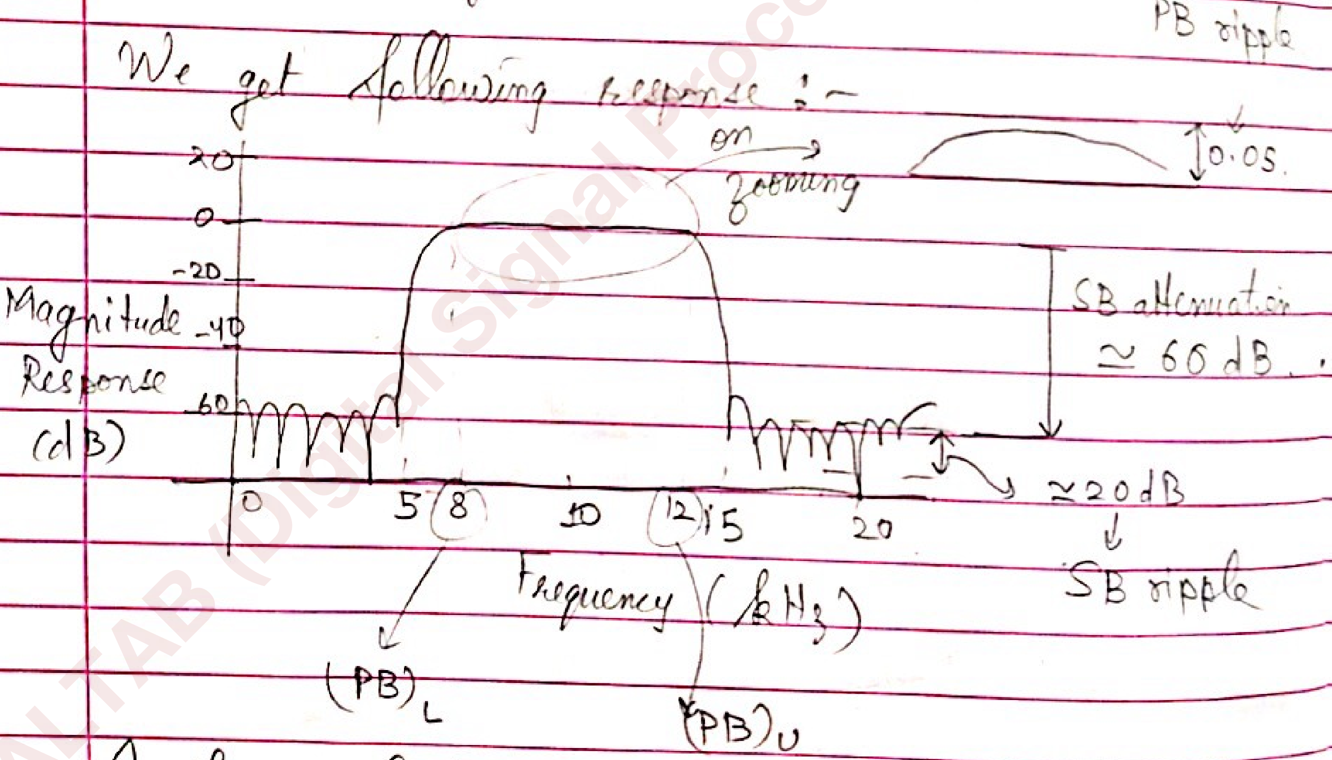
[In theory, we do $h(n) = h_D(n) \times \omega(n)$ $\forall\, n \in (0, N)$]

→ Now, obtaining freq response
i.e, we have h(n) values.
Find its dB values.
Now, find corresponding freq w.r.t these dB values.
Then, plot dB (h(n)) vs freq

Using MATLAB:

```
[H,f] = freqz(hn, 1, 512, fs);
mag = 20*log10(abs(H));
plot(f, mag), grid on
xlabel('Frequency (Hz)');
ylabel('Magnitude Response (dB)');
```

We get following response :-

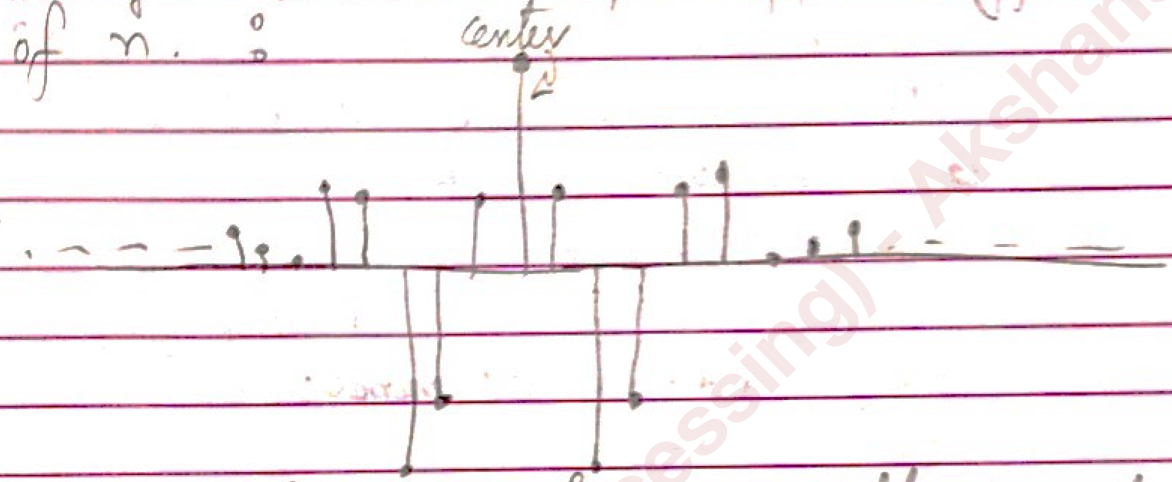


In theory, finding PB & SB ripples
PB : $20 \log(1+\delta_p) = 0.01$, given.
So, $\delta_p \simeq 0.08$. (We got 0.05)
& $20 \log(\delta_s) = 0.001$
So, $\delta_s \simeq 60$ dB
(we got ≃ 20 dB)

→ Now, finding impulse response,
use :-

imp z(hn, 1)

we get the values of coeff. at diff^t values of n. :



Counting them, we find on coeff. in center & 26 coeff. on either side of it. So, 53 coeff. implemented.

(2) Using Kaiser Window :-

SB attenuation as found before,
from δ_s = 60 dB

Now, for Kaiser window, for Attenuation(A) ≥ 50

$$\beta = 0.1102(A - 8.7)$$

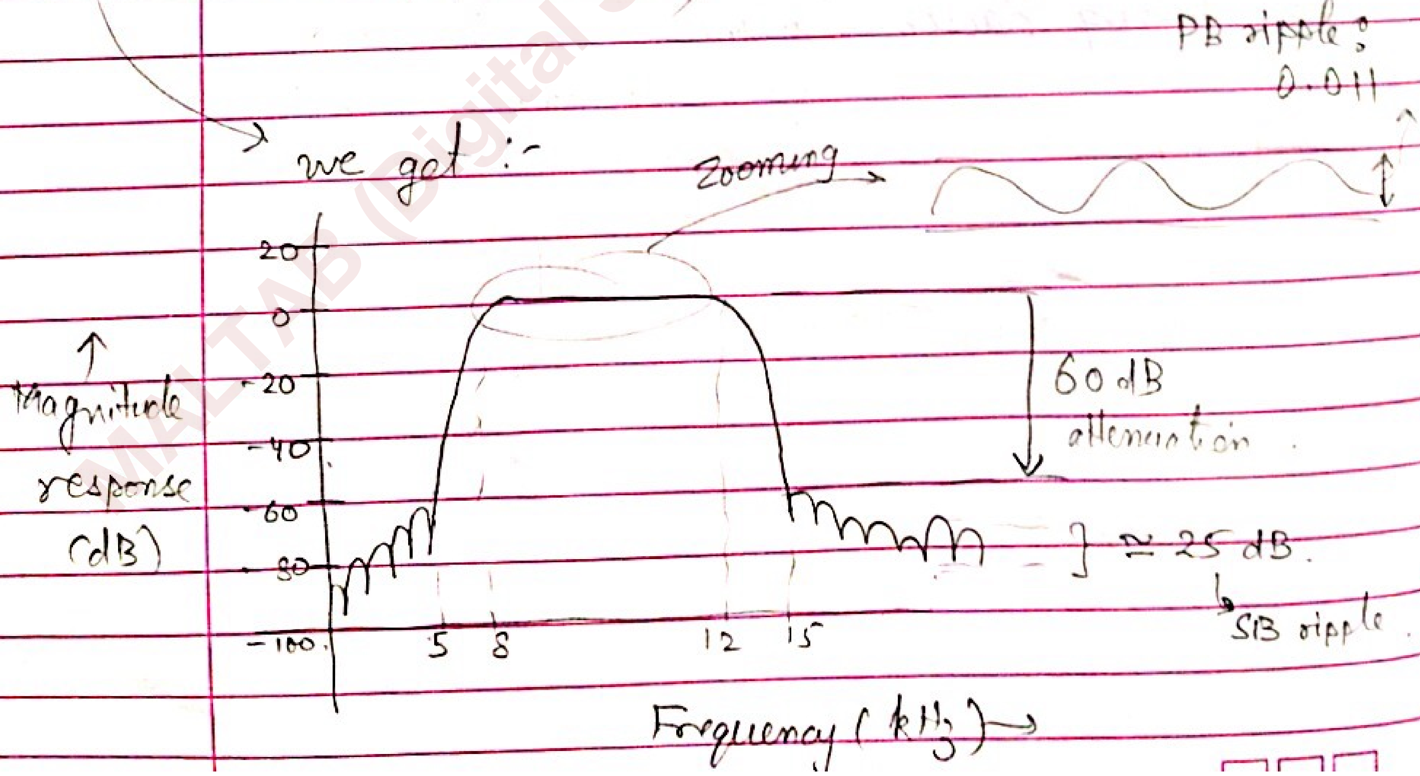
$$\& \; N \geq \frac{A - 7.95}{14.36 \Delta f}$$

Using this, we get $N \simeq 58$.

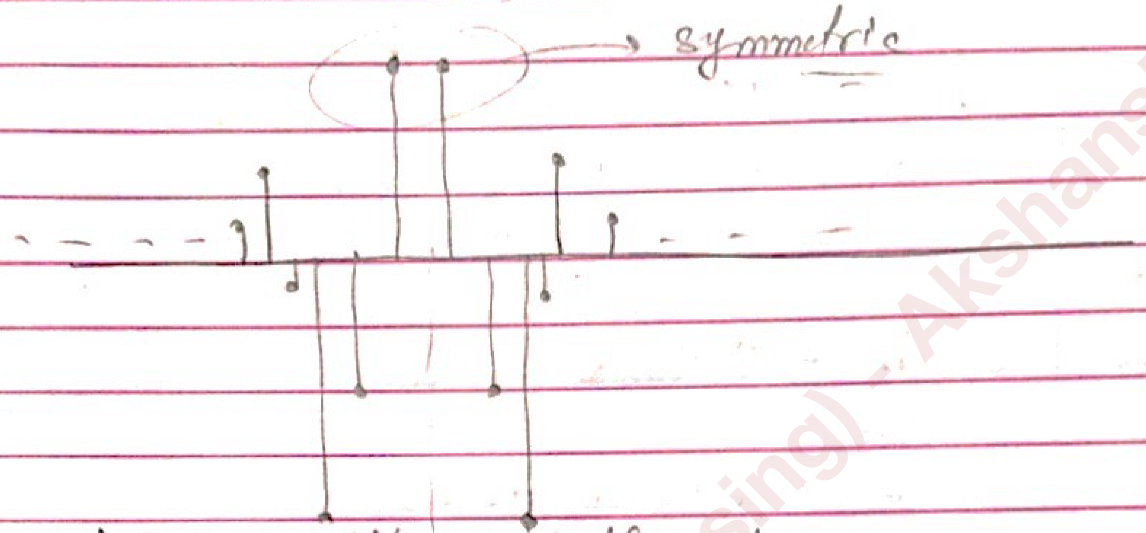
Now, continuing with same program :

```
N = 58;
beta = 5.65;
hn = fir1(N-1, fc, kaiser(N, beta));
[H, f] = freqz(hn, 1, 512, Fs);
mag = 20*log10(abs(H));
subplot(4, 2, 1)    % Making all plots & parts in same plots
plot(f, mag), grid minor;  % frequency response
xlabel('Frequency (Hz)');
ylabel('Magnitude Response (dB)');

% Finding impulse response :-
subplot(4, 2, 2);
impz(hn, 1);
```

we got :-



Impulse response, got from command :-
N = 58 (even)

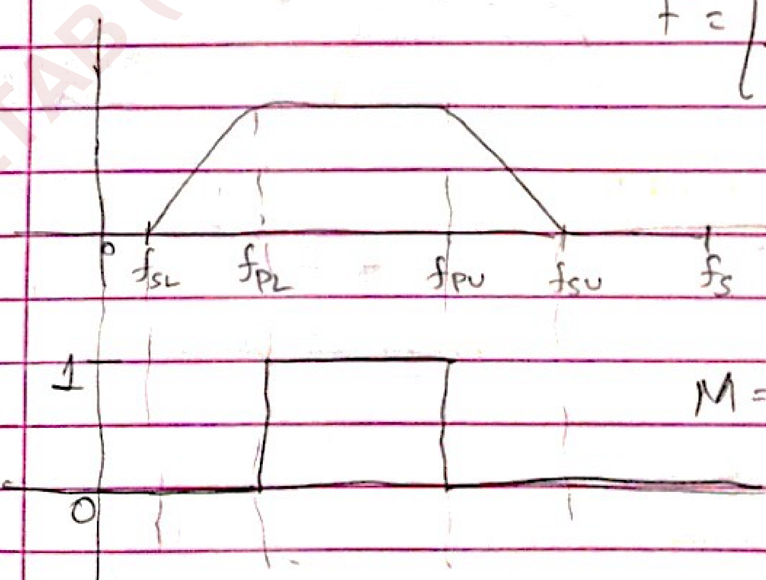


Total 29 coeff. on both sides.

(3) Using optimal method :-

In this filter, we find an array of all normalised frequencies. let the array be F
Now, we make another array considering the case that its ideal. let the array be M
So :-

$$F = \left[\frac{0}{f_s} \quad \frac{f_{sL}}{f_s} \quad \frac{f_{pL}}{f_s} \quad \frac{f_{pU}}{f_s} \quad \frac{f_{sU}}{f_s} \quad \frac{f_s}{f_s} \right]$$



$$M = [\,0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0\,]$$

Commands :

F = [5000, 8000, 12000, 15000]

M = [0 1 0]

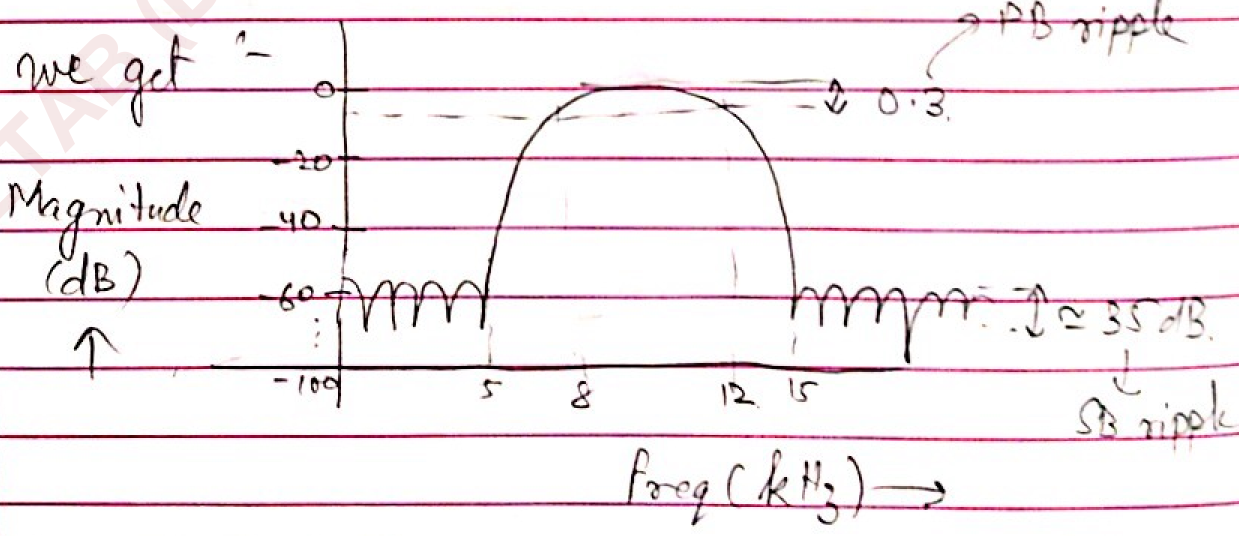
- 0 → from 5k to 8k
- 1 → from 8k to 12k
- 0 → from 12k to 15k

```
dp = 0.01 ;
ds = 0.001 ;
dev = [ds dp ds] ;
[N1, F0, M0, W] = remezord (F, M, dev, fs)
```

↳ for determining order

```
[b delta] = remez (N1, F0, M0, W) ;
[H,f] = freqz(b,1,1024,fs);
mag = 20*log10(abs(H)) ;
plot (f, mag), grid minor ;
xlabel ('Frequency (Hz)') ;
ylabel ('Magnitude (dB)') ;
```

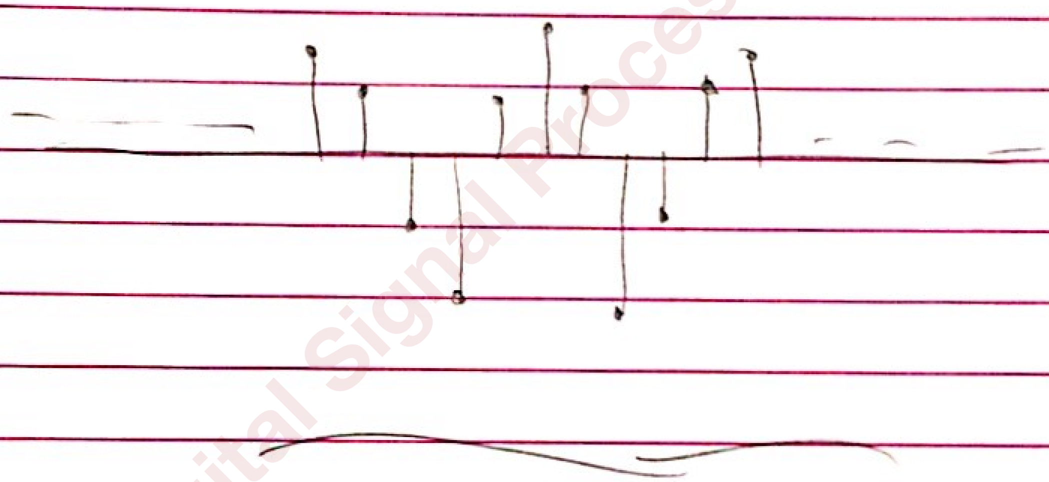
we get :-



Impulse response

impz(b, 1)

For N = 40, we get

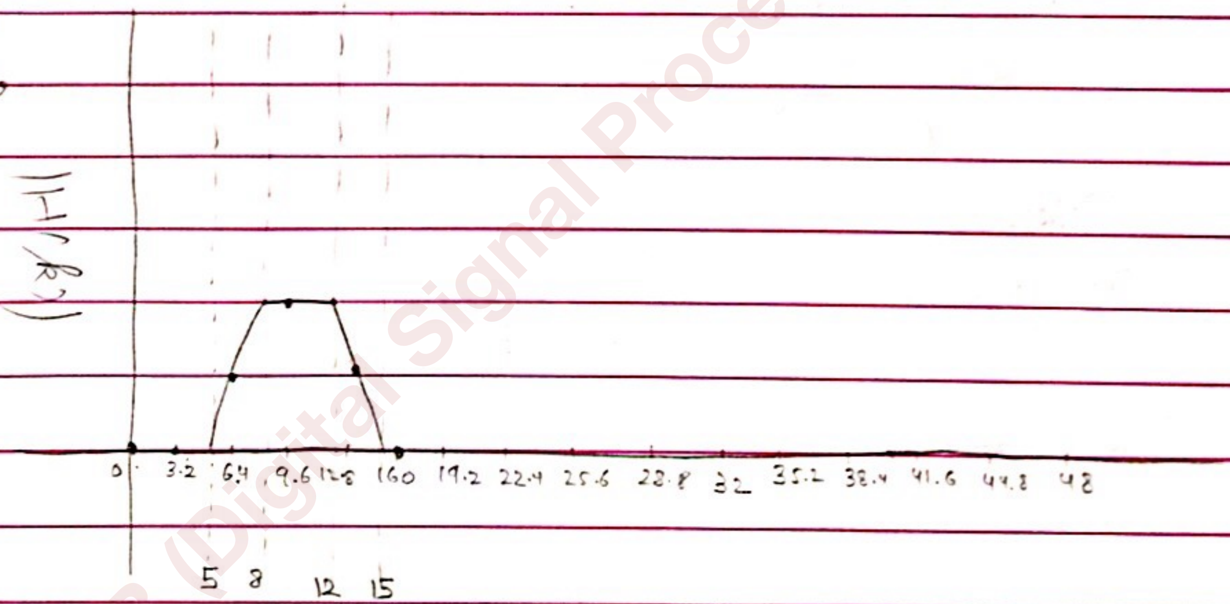


(4) Using Frequency Sampling method :-

Assuming N = 15 i.e., a 15 pt. FIR filter, we have

Now, with f_s = 48 kHz, distance (→ freq. difference) b/w each samples = $\frac{48}{15}$ = 3.2 kHz .

So,



Page No

So, we have

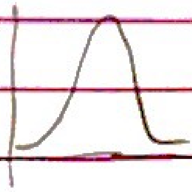
$$|H(k)| = \begin{cases} 0 & k = 0, 1 \\ 0.41793 & k = 2 \\ 1 & k = 3 \\ 0.404058 & k = 4 \\ 0 & k = 5, 6, 7 \end{cases}$$

(values taken from table 7.11)

normalising freq. points w.r.t. half of sampling freq :-

$0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1$ (taking half of 15 pts due to symmetry)

```
Fs = 48000;
N = 15;
fd = [0 1/7 2/7 3/7 4/7 5/7 6/7 1];
Hd = [0 0 0.41793 1 0.404058 0 0 0];
hn = fir2(N-1, fd, Hd);
[H, f] = freqz(hn, 1, 512, Fs)
plot(f, abs(H)), grid on
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

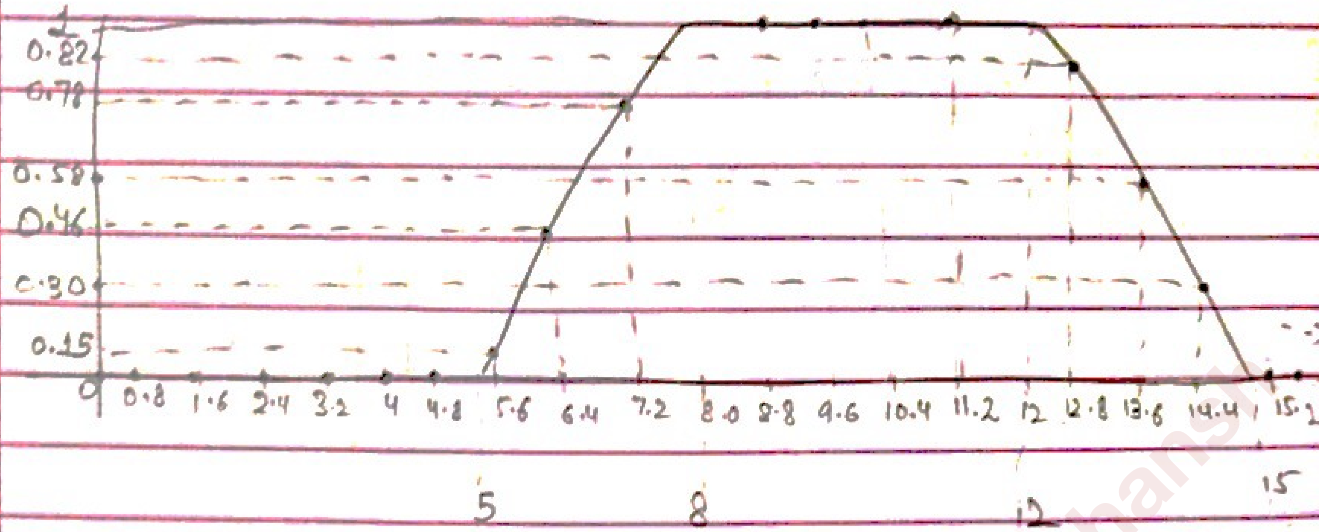
We get 

Now,

*Actual work : Taking N = 58.

So, 58 point FIR filter. So, we take $\frac{58}{2} = 29$ samples.

Now, with $f_s = 48$ kHz, distance b/w each sample $= \frac{48}{58} \simeq 0.8$ kHz



Normalised freq points are

$\frac{0}{29}, \frac{1}{29}, \frac{2}{29}, \frac{3}{29}, \ldots \frac{27}{29}, \frac{28}{29}, 1$

f_d = [0 1/29 2/29 3/29 . . . 28/29 1]

OR →

```
fd = zeros(1,29)
for i = 0:28
fd(i+1) = fd(i+1) + (i * 0.8)
end ; fd = fd./22.4 ;
Fs = 48000;
```

→ for normalising → only needed for "for loop"

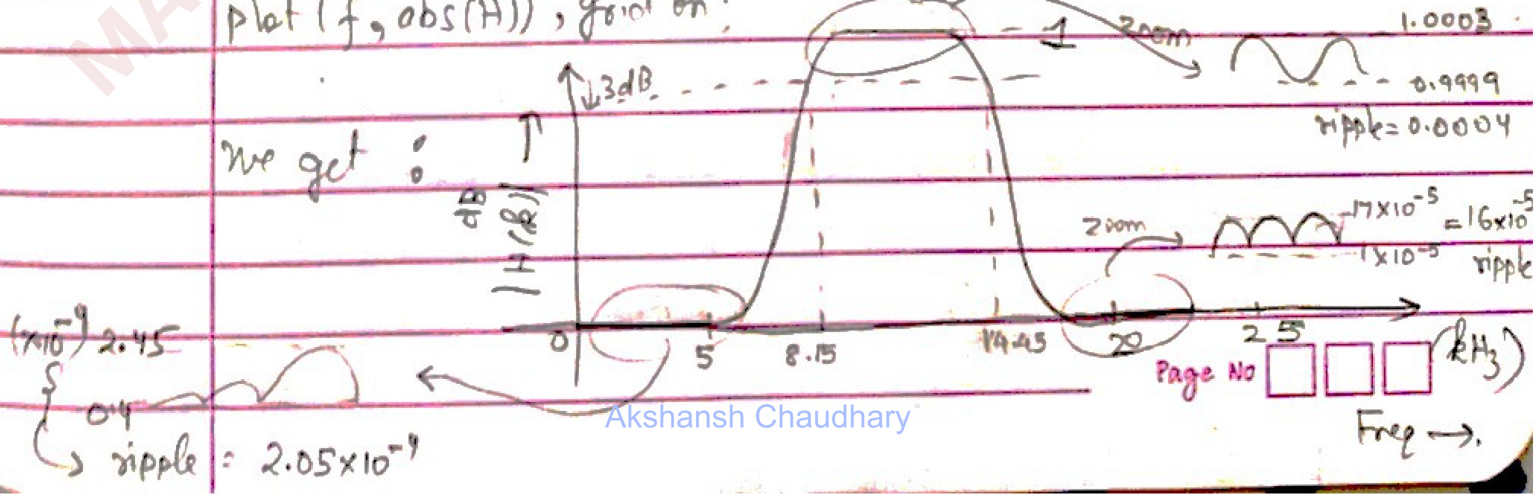
Now,

H_d = [0 0 0 0 0 0 0 0 (0.15) (0.46) (0.78)
1 1 1 1 1 1 (0.82) (0.58) (0.30)
0 0 0 0 0 0 0 0 0]

The values in ( ) lie in the Trans^n band

Now, hn = fir2(N-1, fd, Hd) ; [H, f] = freqz(hn, 1, 512, Fs);
plot (f, abs(H)) ; grid on ;

we get :

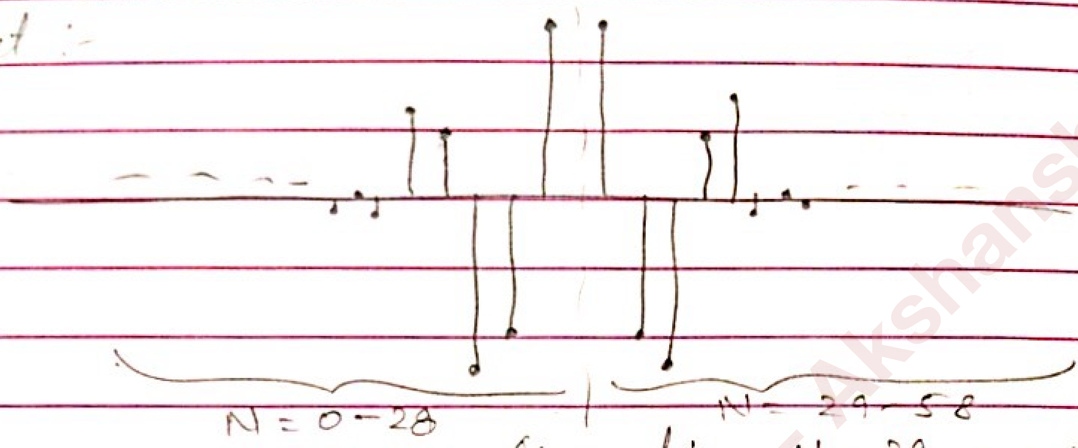


Finding Impulse response

Cmd :-

impz(hn);

we get :-



(symmetric, N = 29, one side
29-58, other side)

Observations :

- ✓ Frequency sampling method is giving less/lower ripples in PB as well as SB.
- ✓ The values for cutoff freq. of PB & SB are nearly same as given in the question.
- ✓ Value of N=58 was taken as the no. of samples, corresponding to the value that was got in Kaiser window.
- ✓ The attenuation is 100% i.e., corresponding to the values taken in "Hd". ⇒ Frequency sampling method gives a more ideal curve., because, we get :- magnitude = 1 : for PB, 0 : for SB } ideal.
- ✓ The PB cut off freq. seen 3dB down, we see

PB → f_{Pl} = 8.15 kHz → given 8

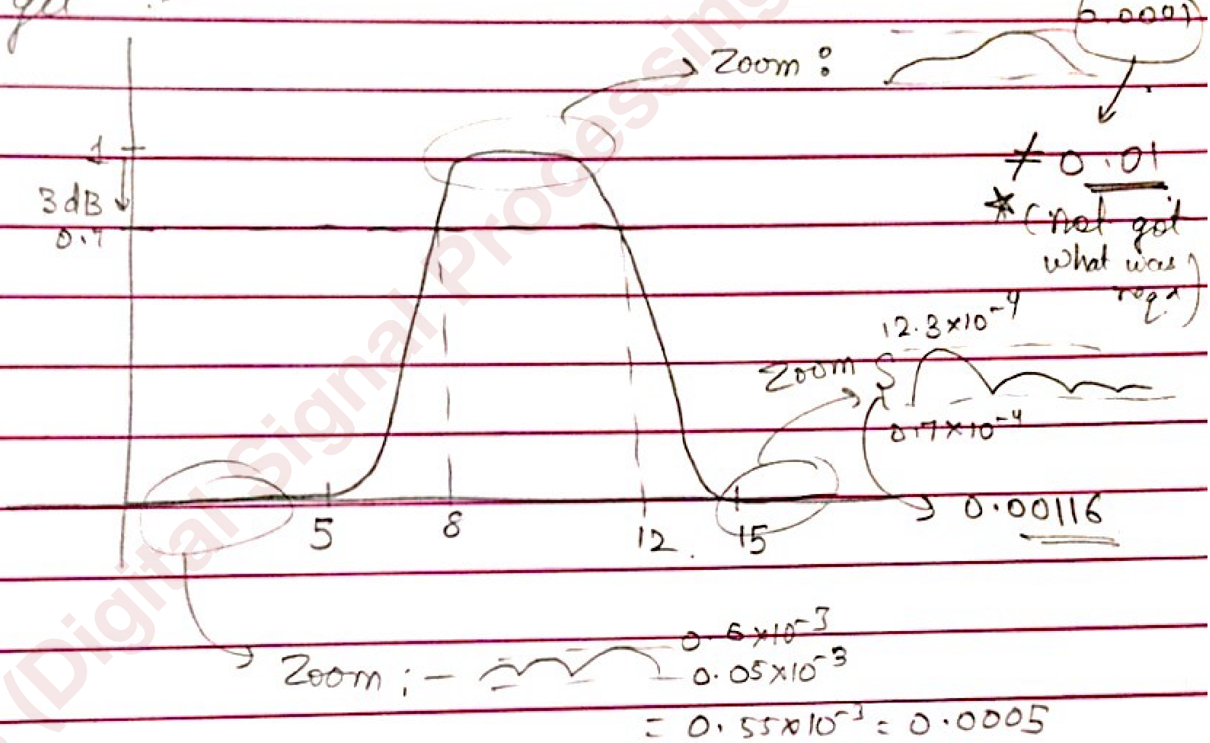
f_{Pu} = 14.45 kHz → given 12 (larger deviation.)

Page No

Trying to match config :

H_d = [0 0 0 0 0 0 0 0 0 0.3 1 1 1 1 1 0.5
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

We get :-

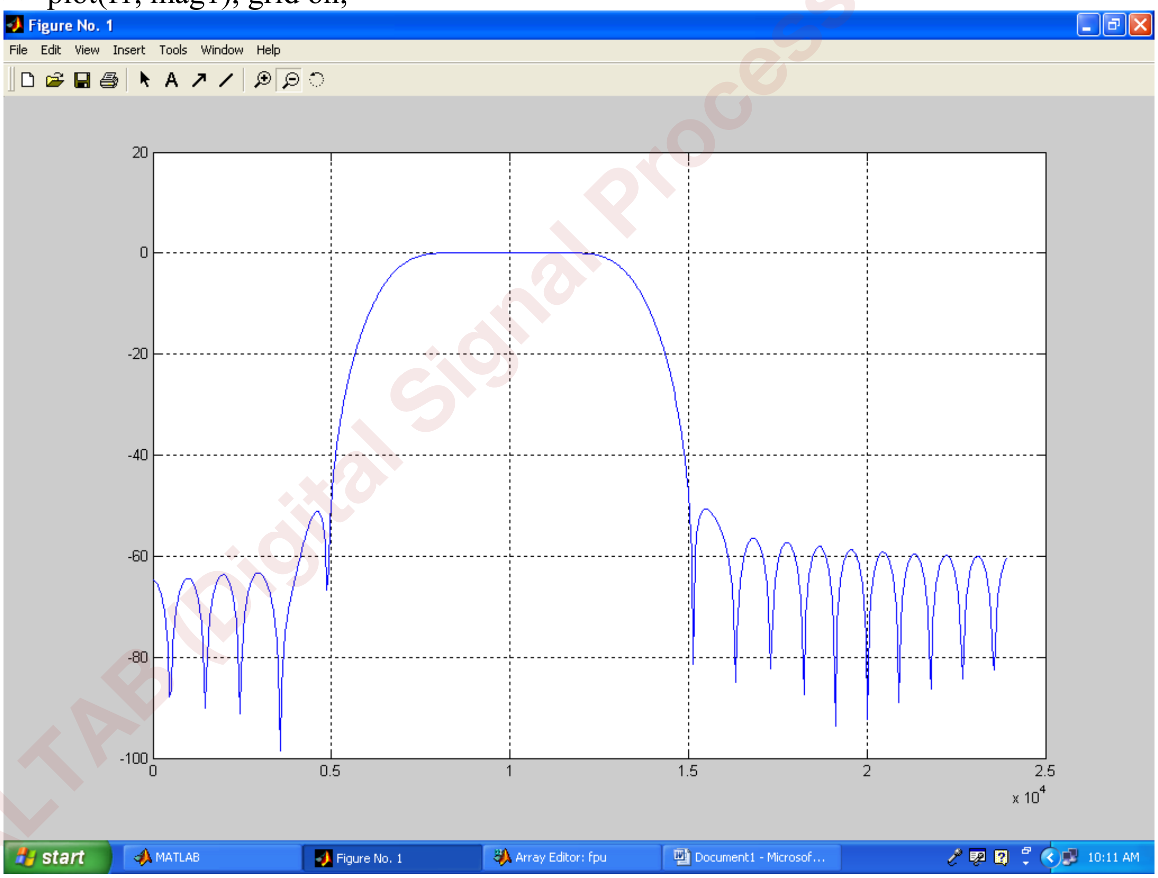


Assignment 6

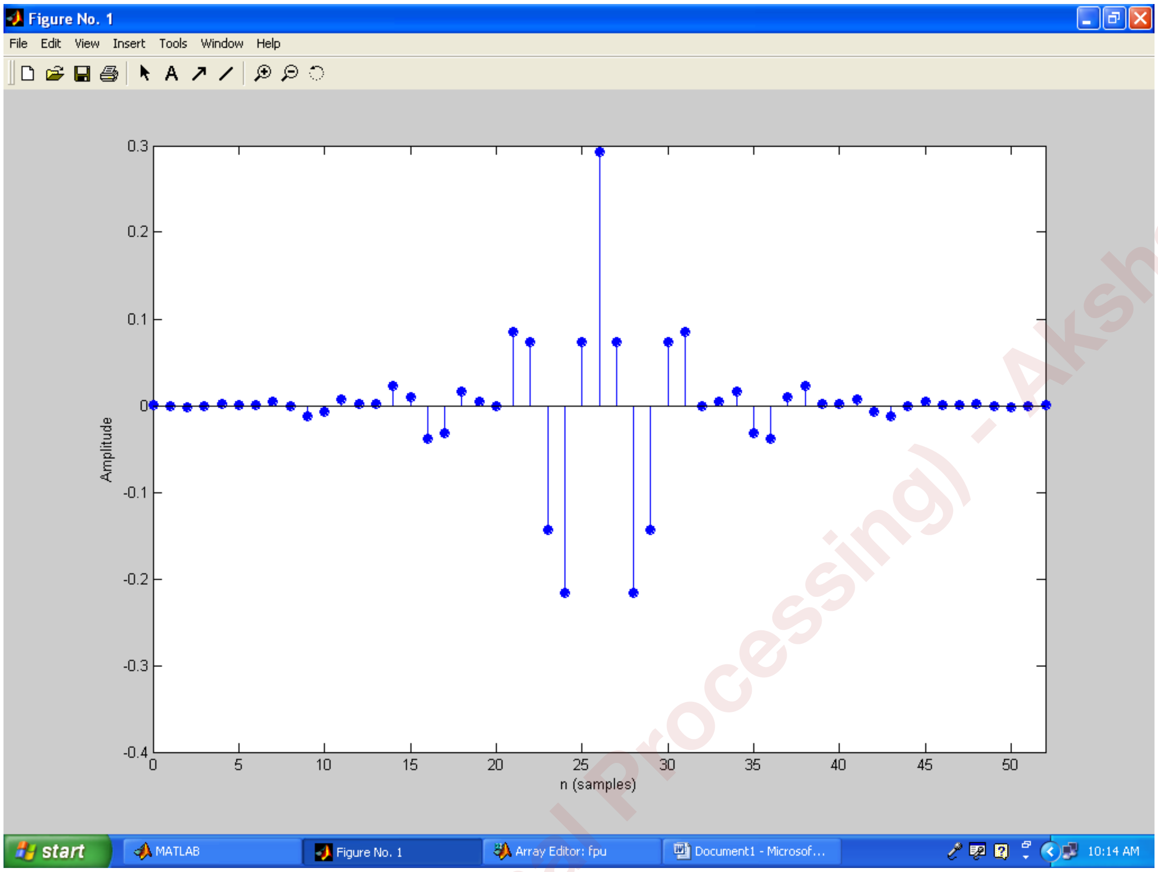
Q. 7.32

1.

```
fpl=13/48;
>> fpu=27/48;
>> fsl=12/48;
>> fsu=15/48;
>> fc=[fpl fpu];
>> N1=53;
>> hd=fir1(N1-1, fc, boxcar(N1));
>> wn=hamming(N1);
>> hn1=fir1(N1-1, fc, wn);
>> fs=48000;
>> [H1,f1]=freqz(hn1, 1, 512, fs);
>> mag1=20*log10(abs(H1));
>> plot(f1, mag1), grid on;
```



```
>> impz(hn1,1)
>>
```



2.

```
N2=58;
>> beta=5.65;
>> hn2=fir1(N2-1, fc, kaiser(N2, beta));
>> [H2, f2]=freqz(hn2, 1, 512, fs);
>> mag2=20*log10(abs(H2));
>> plot(f2, mag2, ‘red’), grid minor;
```

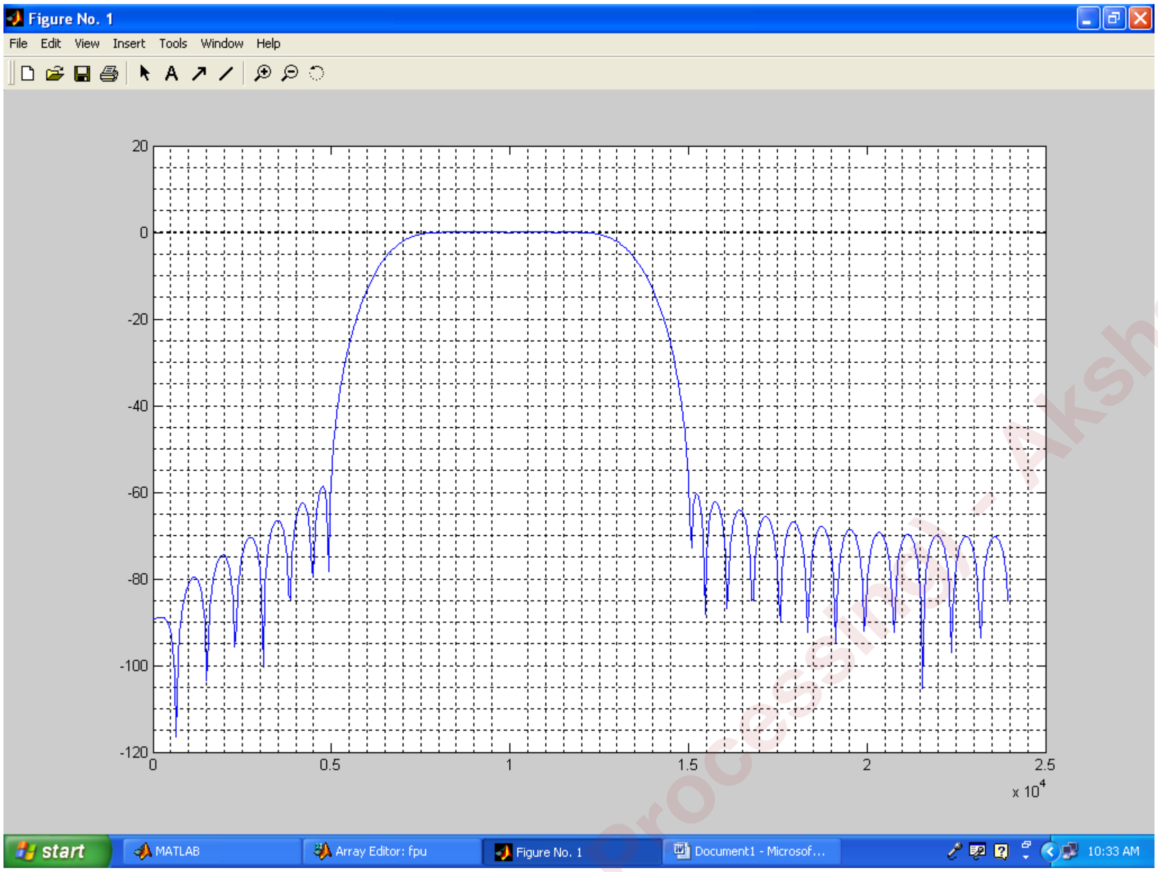


Figure No. 1
File Edit View Insert Tools Window Help
20
0
-20
-40
-60
-80
-100
-120
0
0.5
1
1.5
2
2.5
x 10^4
start
MATLAB
Array Editor: fpu
Figure No. 1
Document1 - Microsof...
10:33 AM

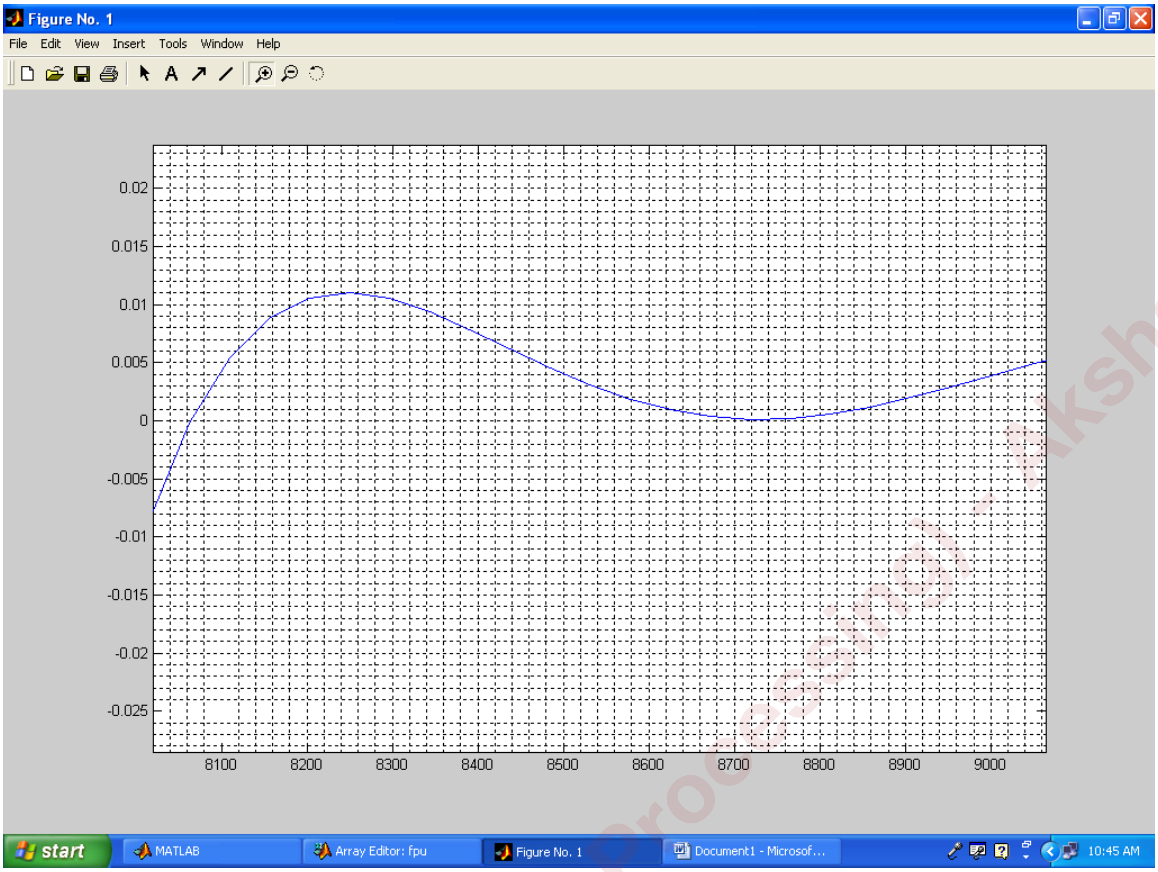
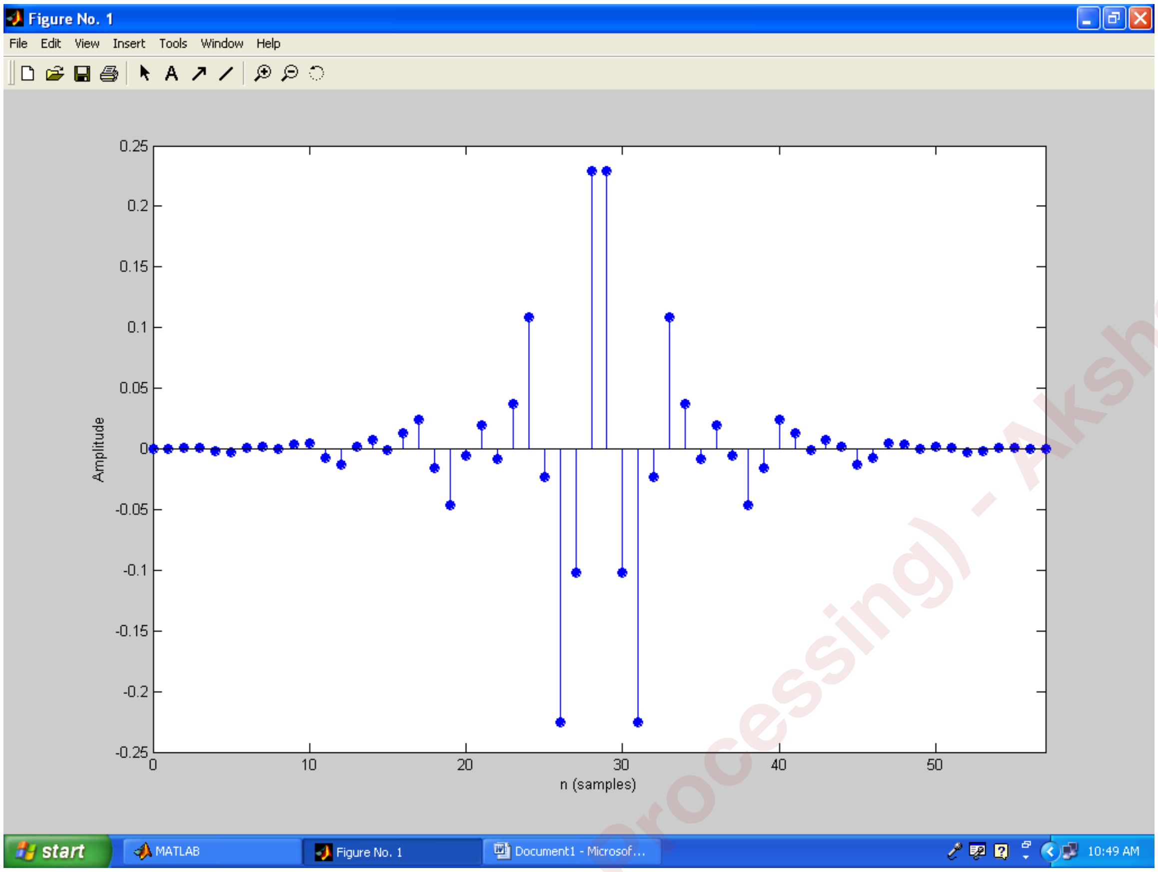


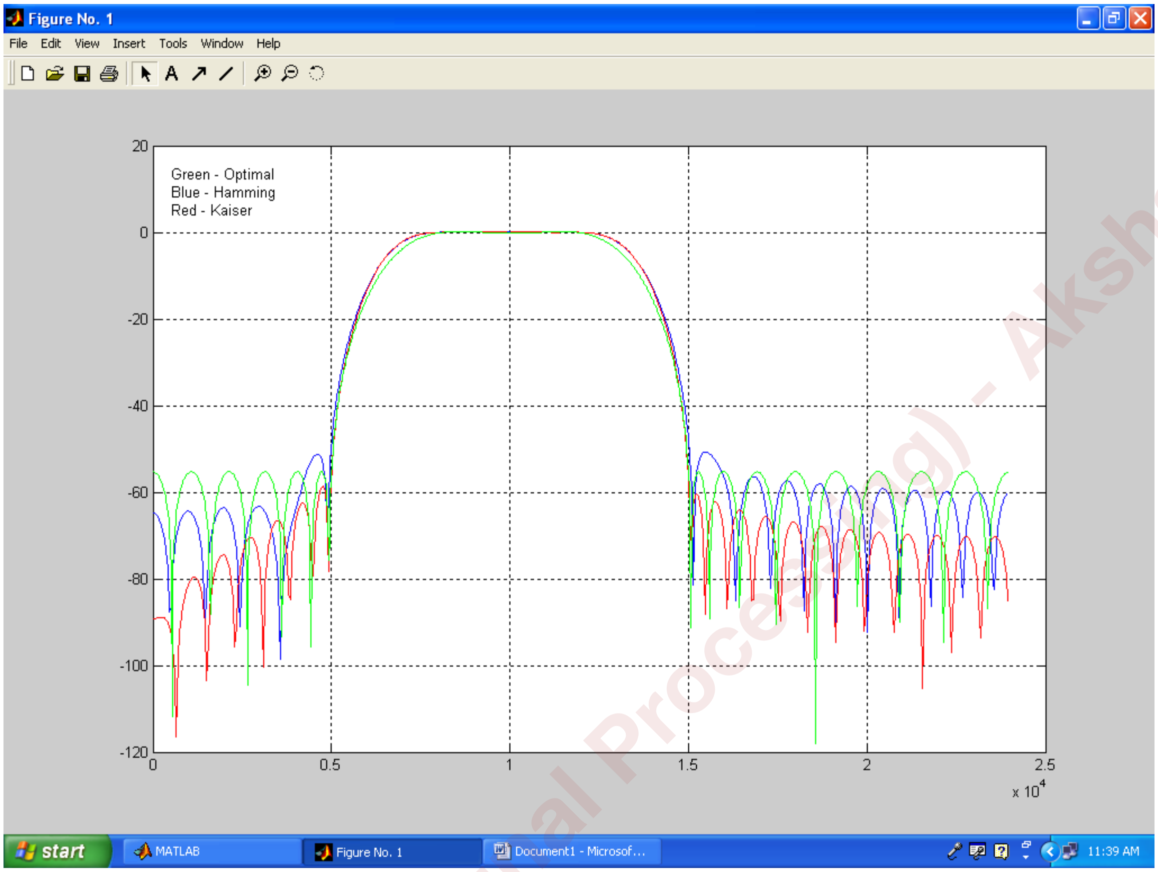
Figure No. 1
File Edit View Insert Tools Window Help
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
-0.025
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
start
MATLAB
Array Editor: fpu
Figure No. 1
Document1 - Microsof...
10:45 AM



3.

```
> F=[5000, 8000, 12000, 15000];
>> M=[0 1 0];
>> dp=0.01;
>> ds=0.001;
>> dev=[ds dp ds];
>> [N3, F0, M0, W]=remezord(F, M, dev, fs);
>> [b delta]=remez(N3, F0, M0, W);
>> [H3, f3]=freqz(b, 1, 1024, fs);
>> mag3=20*log10(abs(H3));
>> plot(f3, mag3, ‘green’), grid minor;
```

Plotting all freq. resaponse on same graph



MATLAB Assignment 7
Digital Signal Processing

Akshansh Chaudhary

CLASSMATE

9/12/13

Assignment - 7 (Lab 8)

Multirate Digital Signal Processing

```
Fs = 5000;
A = 2;
B = 1;
f1 = 50;
f2 = 100;
t = 0:1/Fs:1;
x = A*cos(2*pi*f1*t) + B*cos(2*pi*f2*t);
```

(a)
```
% Taking 1000 samples.
stem(x(1:1000))
xlabel('Discrete time, nT');
ylabel('Input signal level');
```

(b)
```
% Now, decimation by a factor of 10 so, 100 samples
y = decimate(x,10);
stem(y(1:100));
xlabel('Discrete time, nT/10')
ylabel('Decimated output signal level')
```

(c)
```
% Now, interpolating signal by factor of 4.
y1 = interp(y, 4)
stem(y1(1:400))
xlabel('Discrete time, 4*nT')
ylabel('Decimated output signal level')
```

Using function

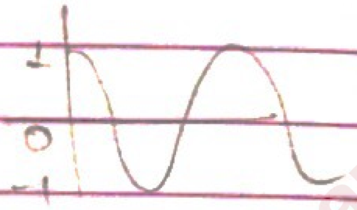
↳ or function

(S1) File > New > function file
(Tab opens)

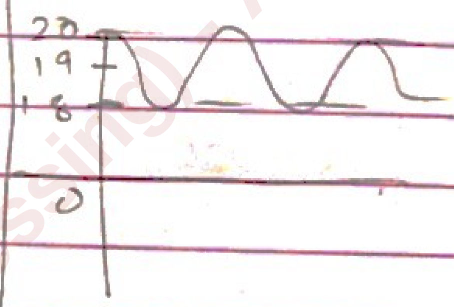
function name(~)
{

}
end

y = sin x



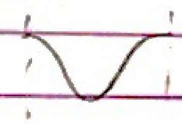
y = 20 + sin x



(S2) Save it

Calling function

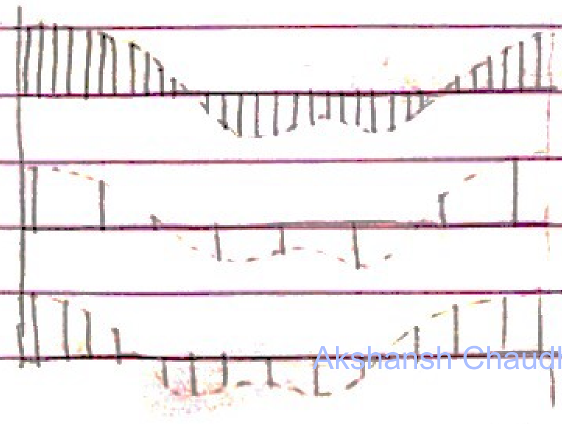
In Cmd. window :
name ↵

How to see? → See in one cycle ()

Original

Decimated

Interpolated.



Assignment 7
Decimation and Interpolation
9.12.13

Commands –
```
>> Fs=5000;
>> A=2;
>> B=1;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
>> x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:100))
>> xlabel('Discrete time, nT/10')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
>>
```

Graphs -
Part a – 1000 samples

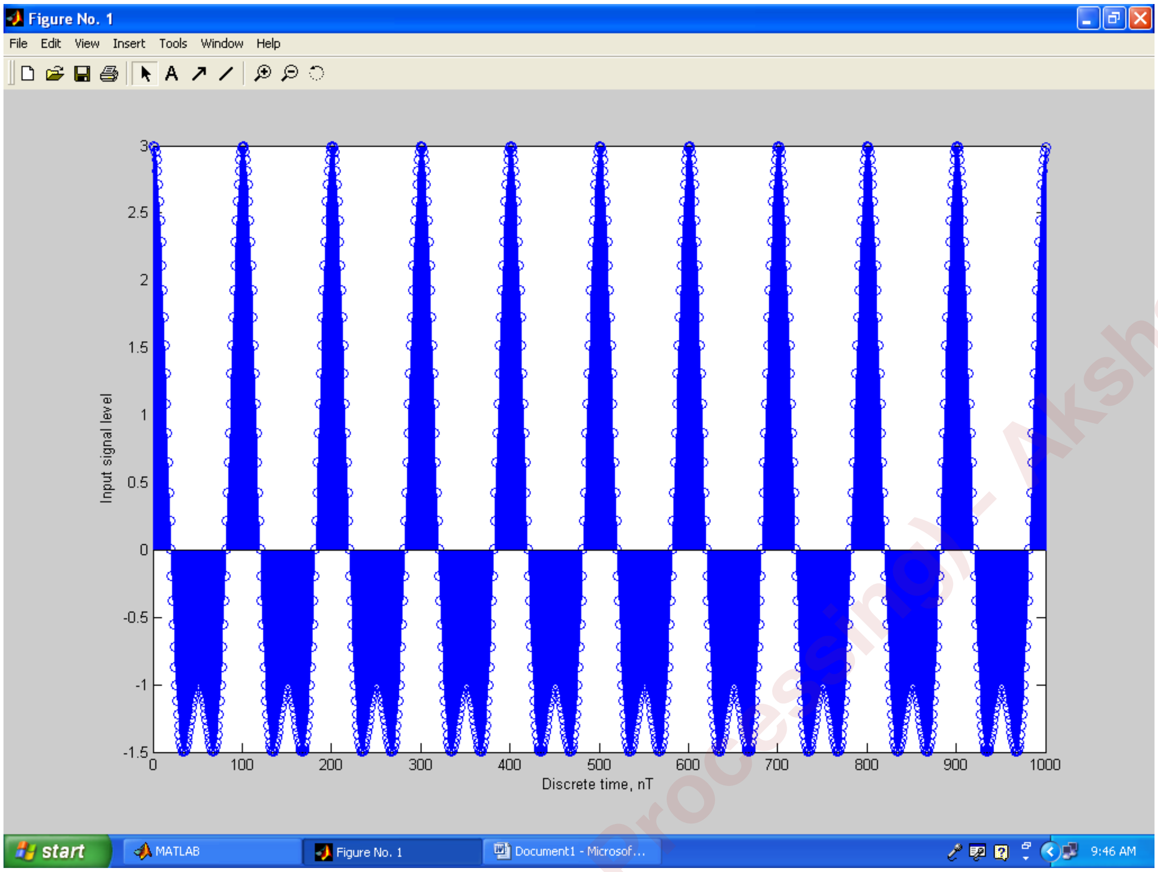
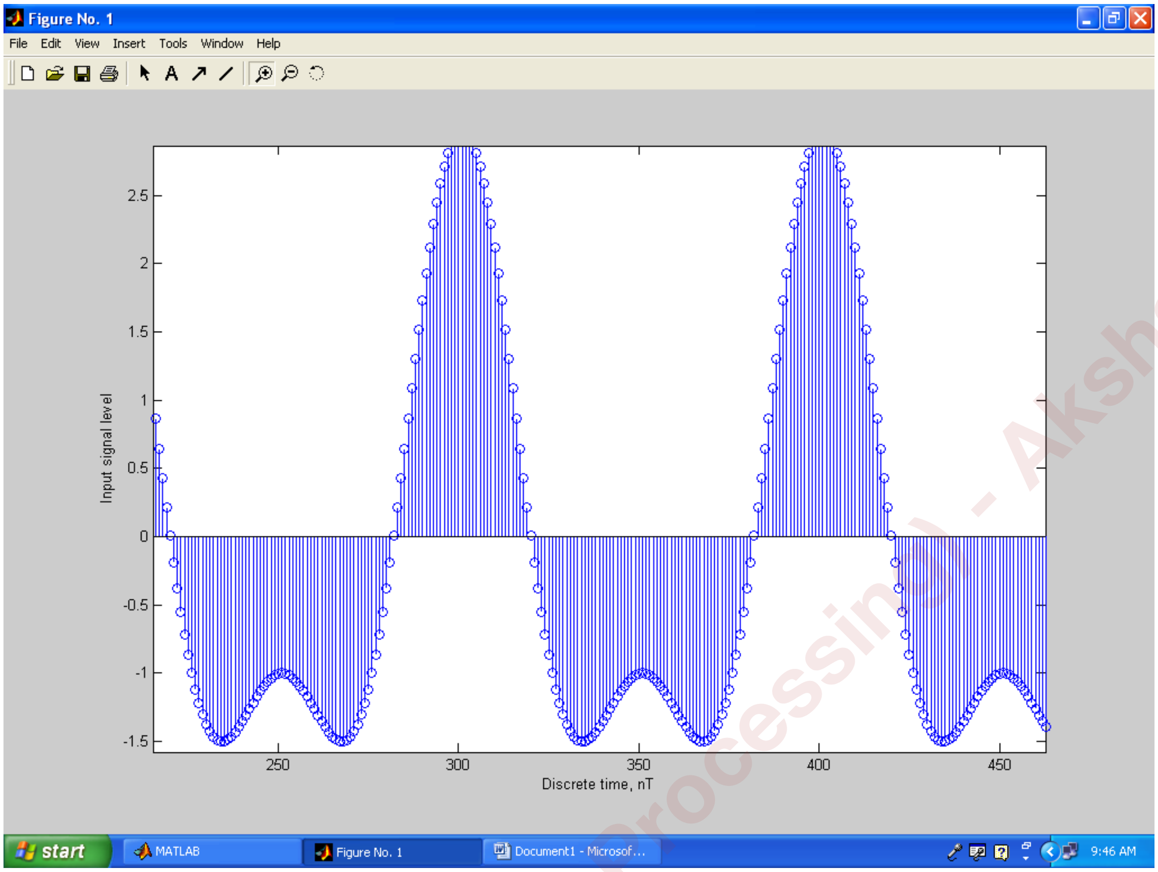
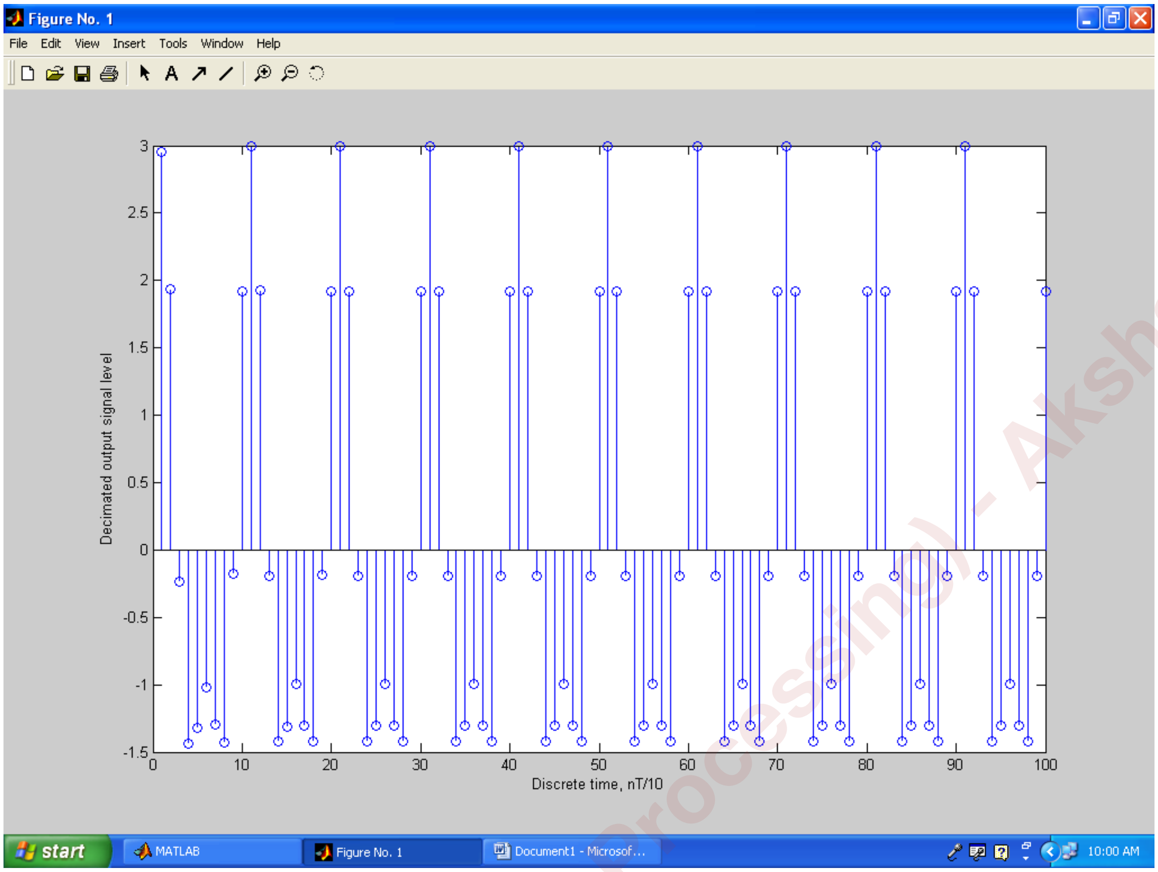


Figure No. 1
File Edit View Insert Tools Window Help
Input signal level
Discrete time, nT
start
MATLAB
Figure No. 1
Document1 - Microsof...
9:46 AM

Zooming –



Part b – Decimation (1000/10.e. 100 samples)



Part c- Interpolation (250*10 i.e. 2500 samples)

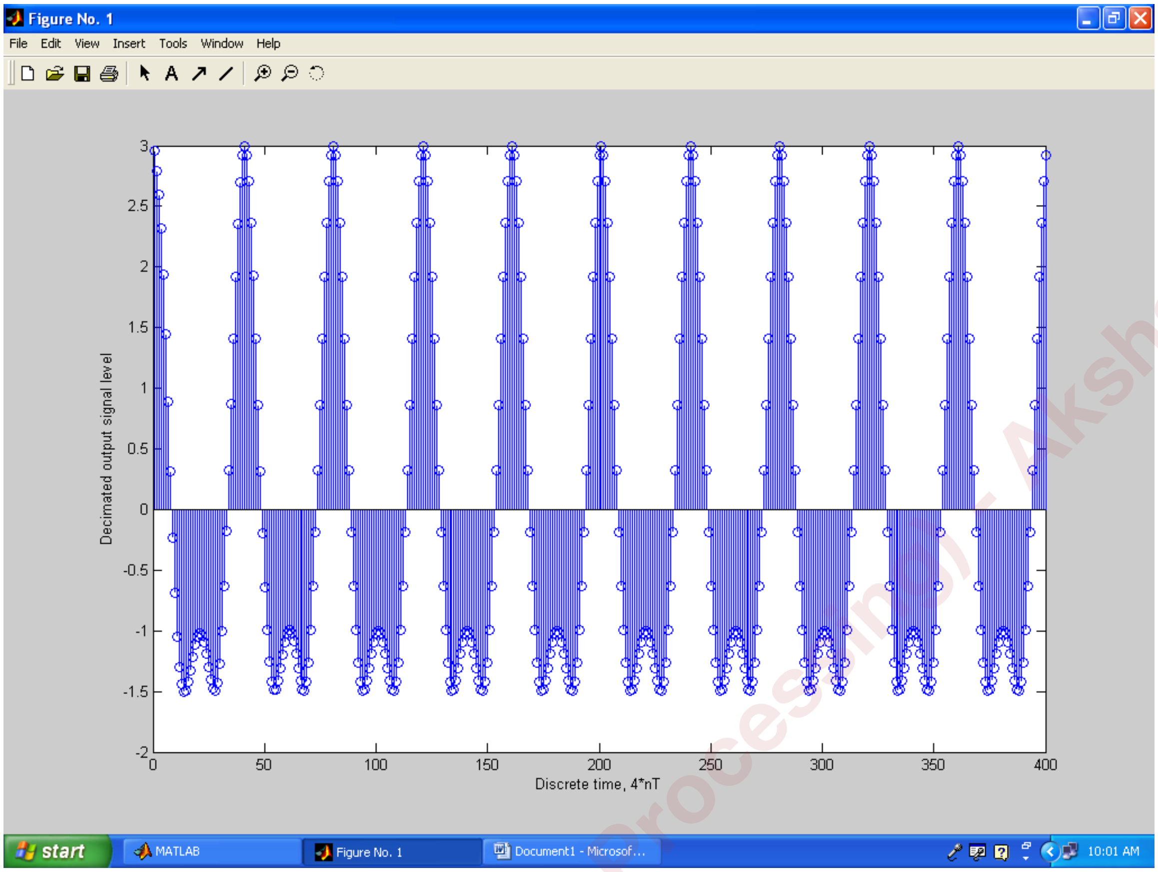
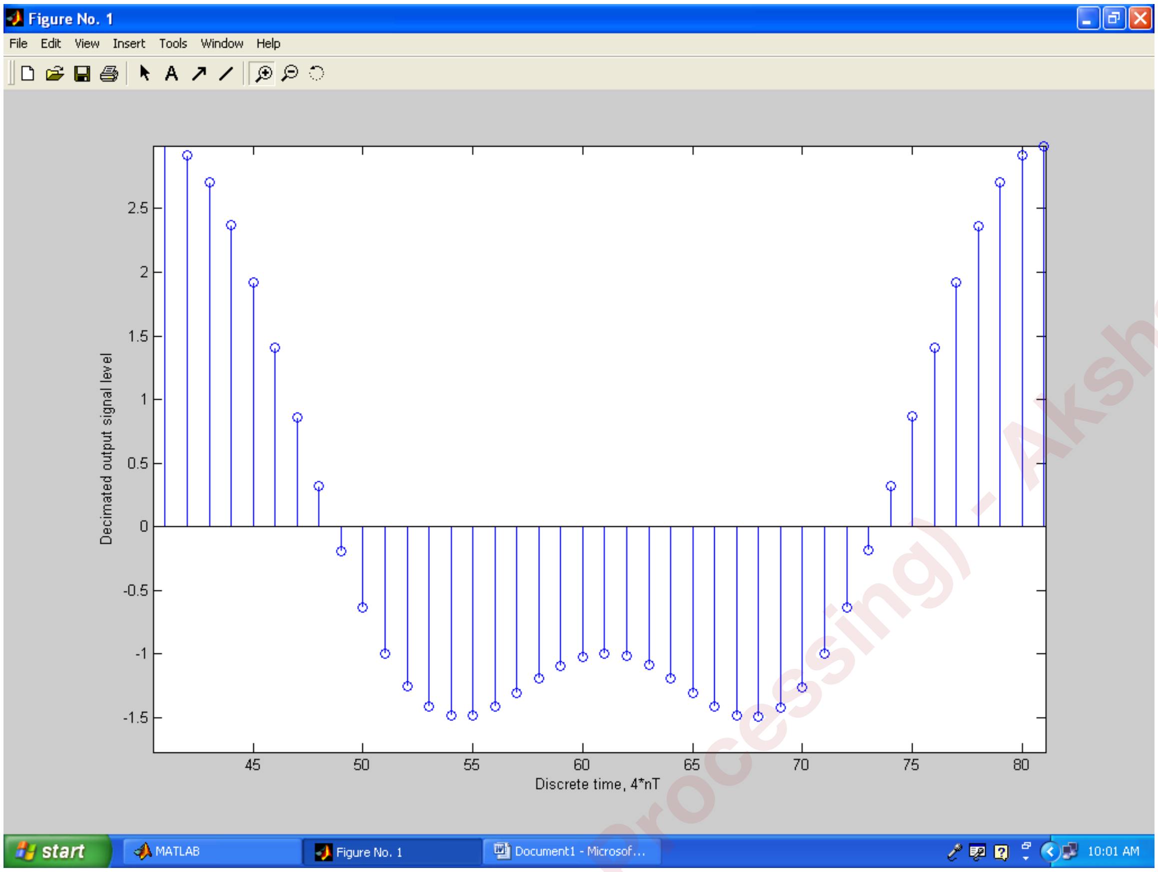
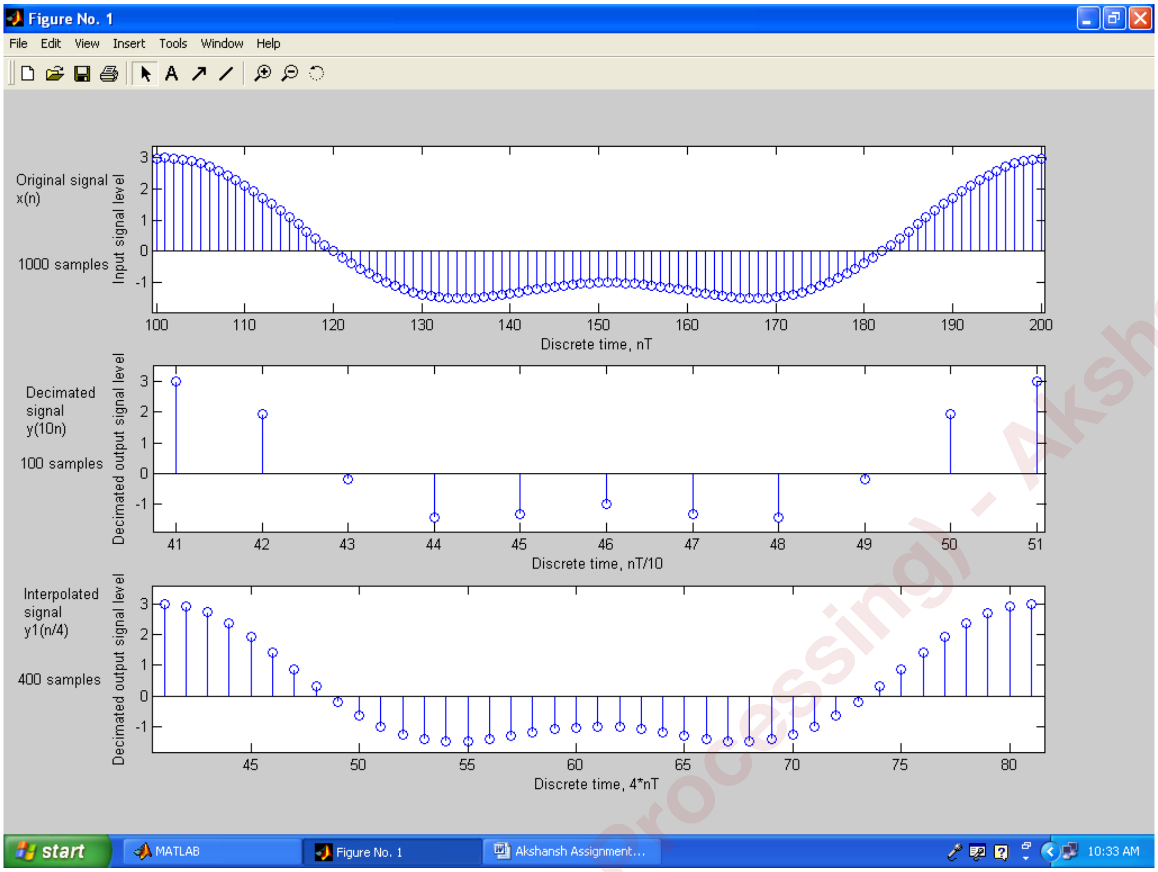


Figure No. 1
File Edit View Insert Tools Window Help
Decimated output signal level
Discrete time, 4*nT
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
50
100
150
200
250
300
350
400
start
MATLAB
Figure No. 1
Document1 - Microsof...
10:01 AM

Zooming –



Combined graph-
Seeing the change in the number of samples in one cycle.

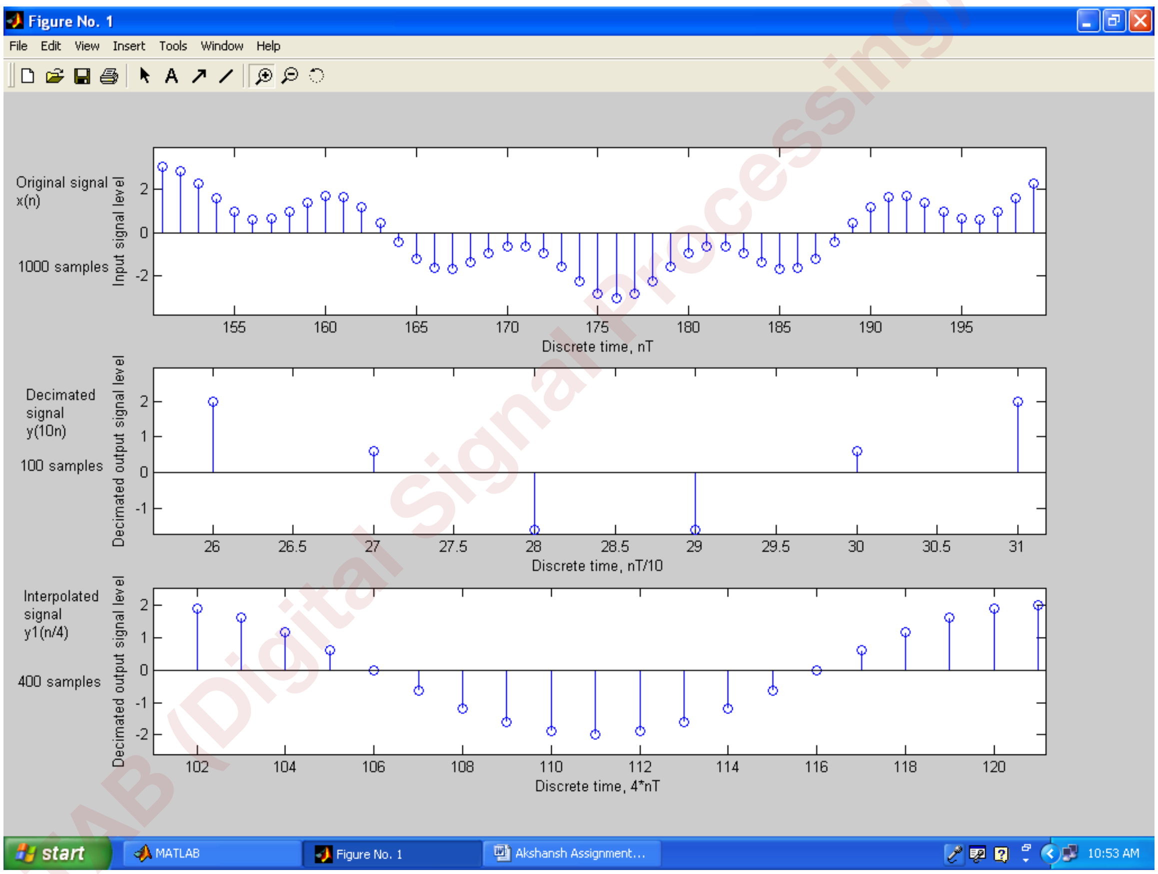


OBSERVATIONS:

Change 1

```
>> Fs=5000;
>> A=2;
>> B=1;
>> f1=100
>> f2=500;
>> t=0:1/Fs:1;
>> x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
```

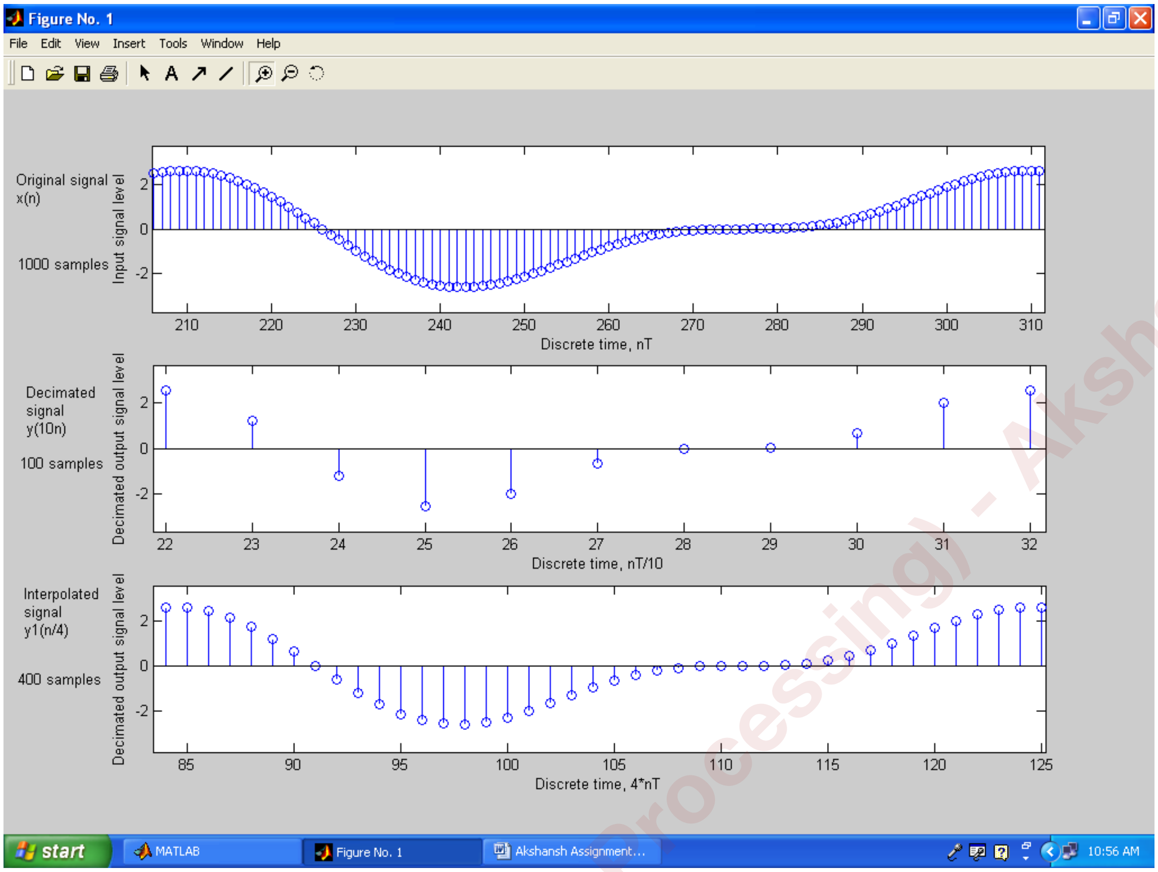
```
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:100))
>> xlabel('Discrete time, nT/10')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
>>
```



Change 2

```
Fs=5000;
>> A=2;
```

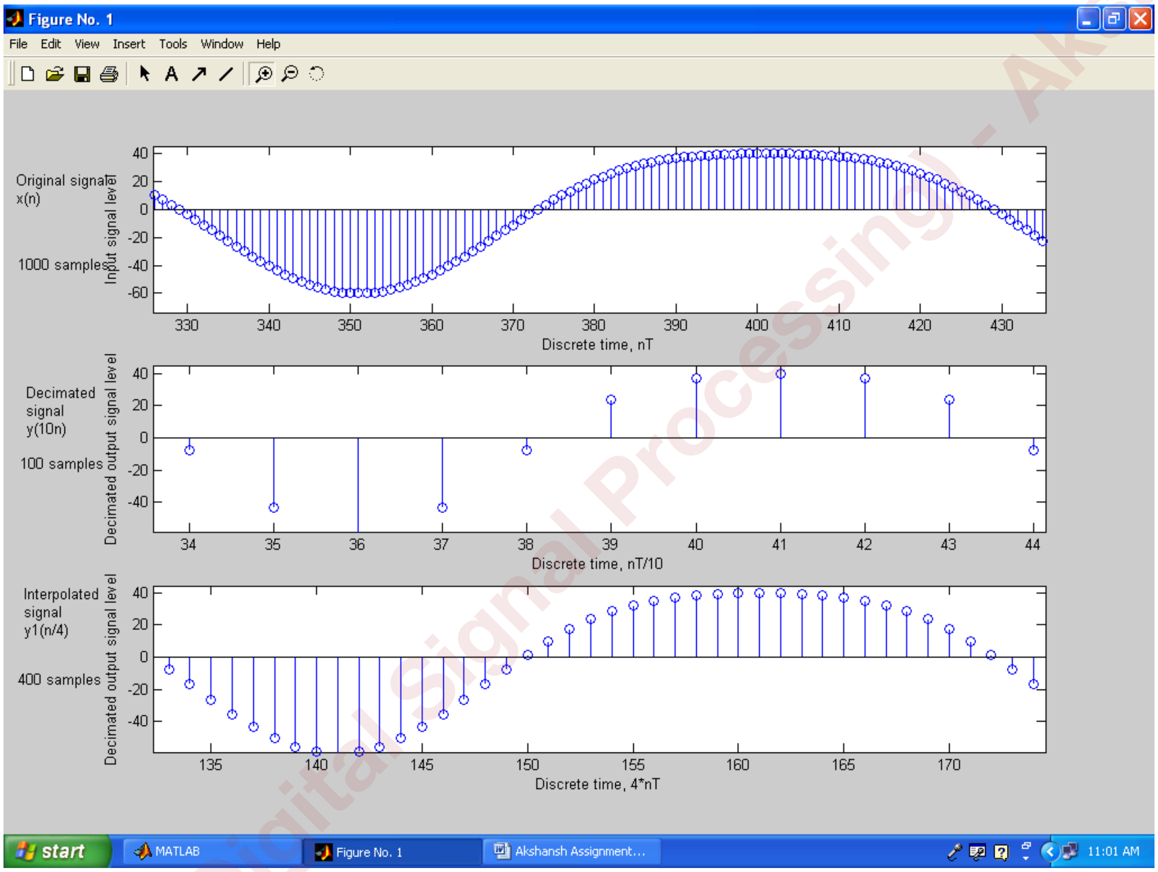
```
>> B=1;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
>> x=A*cos(2*pi*f1*t)+B*sin(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:100))
>> xlabel('Discrete time, nT/10')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
```



Change 3

```
Fs=5000;
>> A=50;
>> B=-10;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
>> x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:100))
```

```
>> xlabel('Discrete time, nT/10')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
>>
```

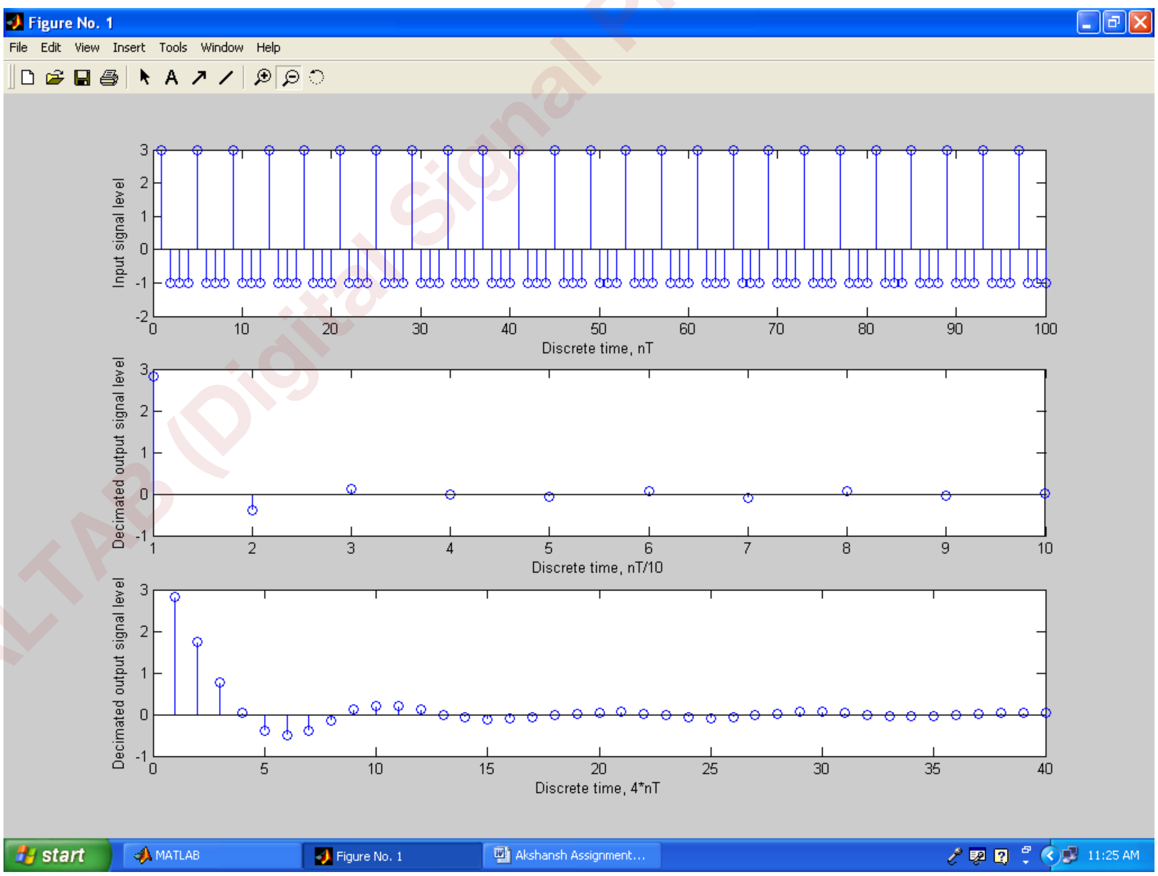


Change 4

Idea – No. of samples taken should always be less than the sampling frequency that we take.

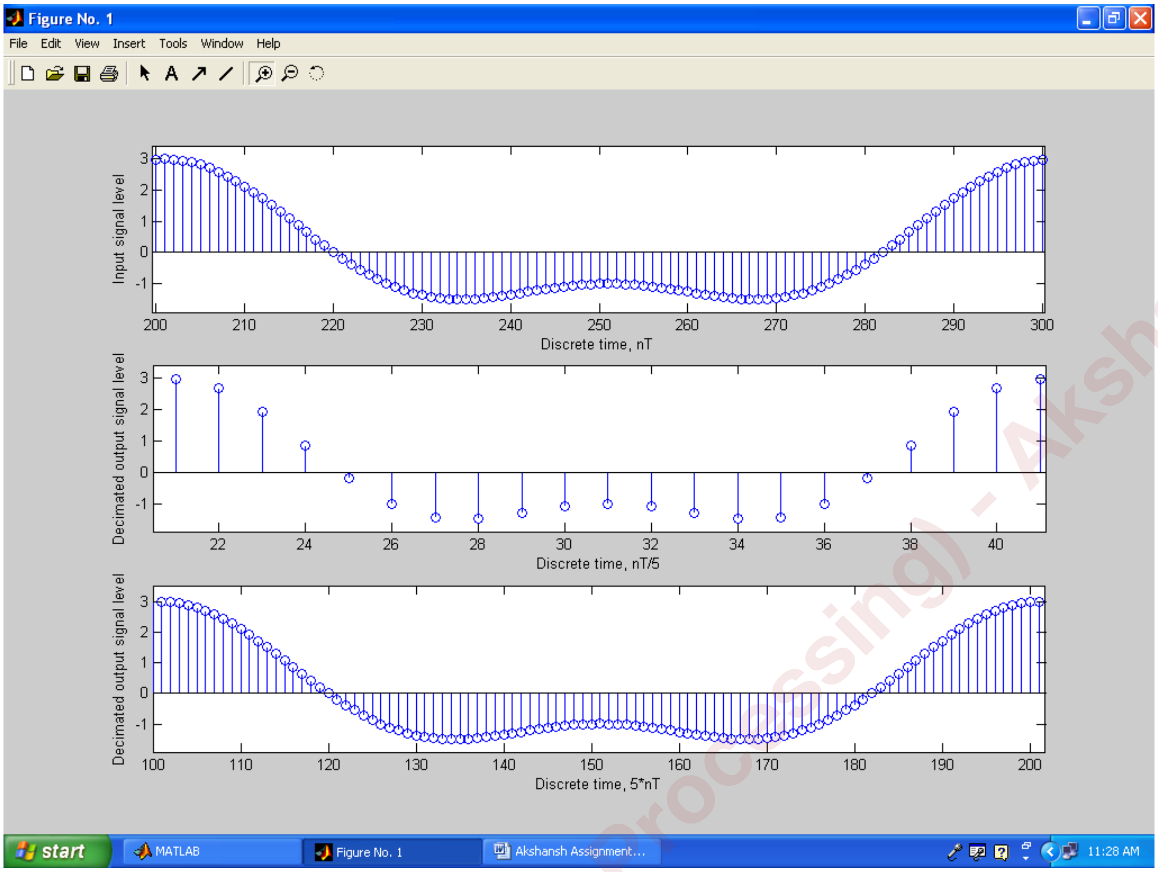
```
Fs=200;
>> A=2;
>> B=1;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
```

```
>> x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:100))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:10))
>> xlabel('Discrete time, nT/10')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:40))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
>>
```



Change 5

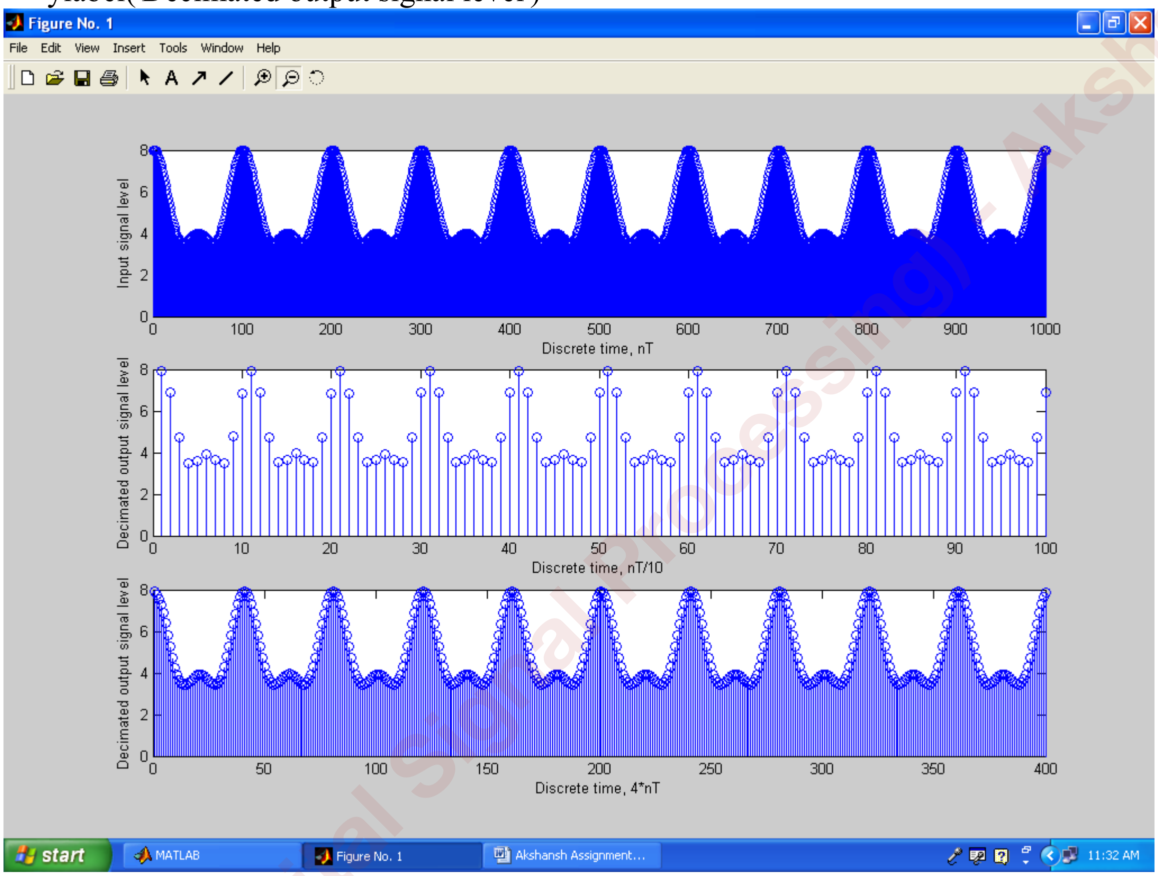
```
>> Fs=5000;
>> A=2;
>> B=1;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
>> x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,5);
>> subplot(3,1,2)
>> stem(y(1:100))
>> xlabel('Discrete time, nT/5')
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,5);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 5*nT')
>> ylabel('Decimated output signal level')
```



Change 6

```
Fs=5000;
>> A=2;
>> B=1;
>> f1=50;
>> f2=100;
>> t=0:1/Fs:1;
>> x=5+A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
>> subplot(3,1,1)
>> stem(x(1:1000))
>> xlabel('Discrete time, nT')
>> ylabel('Input signal level')
>>
>>
>> %Now, decimating for part b.
>> y=decimate(x,10);
>> subplot(3,1,2)
>> stem(y(1:100))
>> xlabel('Discrete time, nT/10')
```

```
>> ylabel('Decimated output signal level')
>>
>> %Now, interpolating for part c.
>> y1=interp(y,4);
>> subplot(3,1,3)
>> stem(y1(1:400))
>> xlabel('Discrete time, 4*nT')
>> ylabel('Decimated output signal level')
```



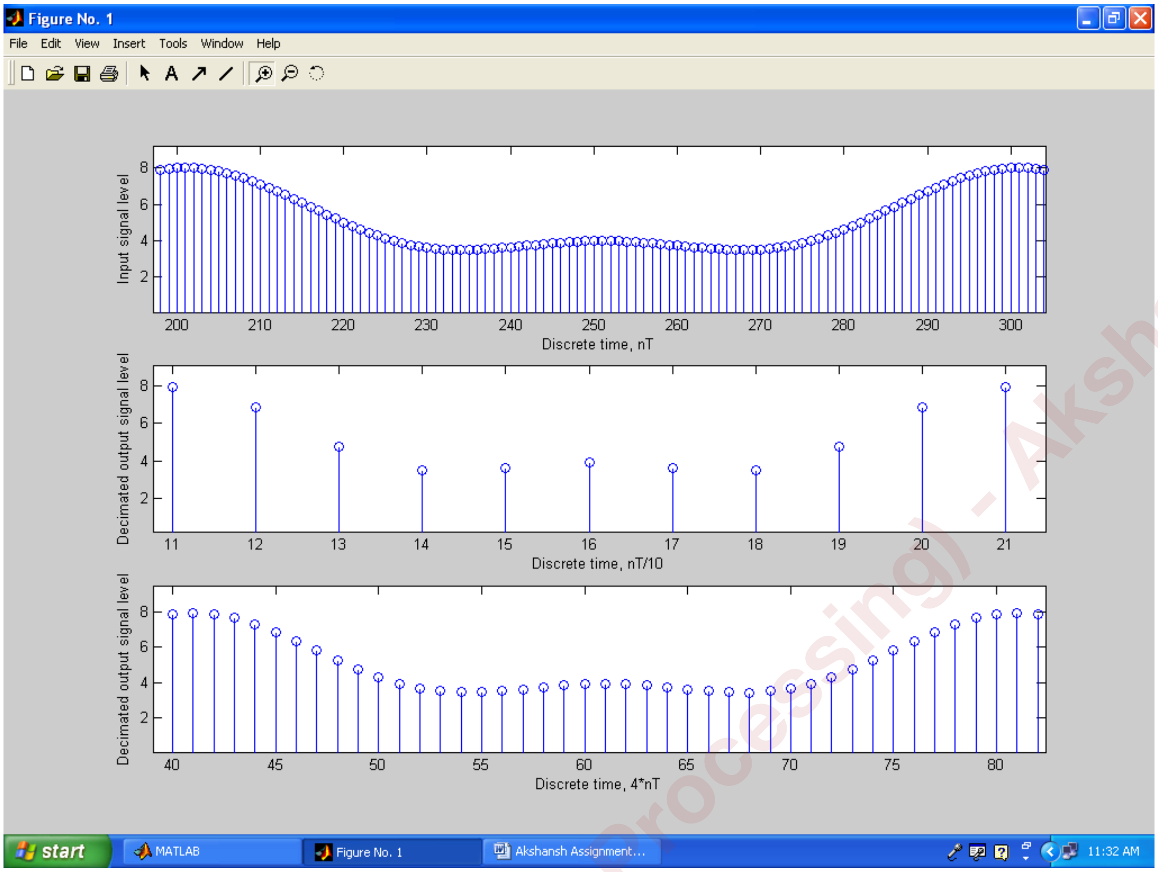


Figure No. 1
File Edit View Insert Tools Window Help
Input signal level
Discrete time, nT
Decimated output signal level
Discrete time, nT/10
Decimated output signal level
Discrete time, 4*nT
start
MATLAB
Figure No. 1
Akshansh Assignment...
11:32 AM