INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR.

ECL 301 : DIGITAL SIGNAL PROCESSING .

PROJECT REPORT ON –

Smoothed Rectangular Function-Based FIR Filter Design.

SUBMITTED BY :

NAME : PRATIK R. ADLE
ENROLLMENT NO. : BT17ECE034

CONTENTS

ACKNOWLEDGEMENT

I would like to acknowledge the guidance and mentorship provided by **Dr. Ankit Bhurane** during the course of this semester, inspiring and motivating us throughout. He helped us conceive this project and provided me with solutions of various potential problems. Also I am very grateful to all my friends who supported me throughout this venture.

ABSTRACT

The Objective of this project is to Study and implement Smoothed Rectangular Function-Based FIR Filter Design . It is a technique for designing FIR filters where the desired frequency response is a smoothed rectangular function . Instead of choosing a suitable window in time domain, we will directly try smoothing the ideal desired response in frequency domain. The impulse response of the FIR filter is of the inverse Fourier transform of the frequency response. This technique provides the very good performance in terms of filter specifications .

INTRODUCTION

For FIR filter design, the main objective is to approximate a desired frequency response using a finite number of FIR filter coefficients. The ideal desired frequency response should look like a rectangular function, and the desired specifications in general are: low variation of the ripples in the passband, high attenuation in the stopband and sharp cutoff, which are competing parameters in FIR filter design. The popular methods that can be used for designing digital FIR filters are: window method, frequency sampling method and optimization methods. Each method has its own advantages and disadvantages.

Window functions can be used for designing digital FIR filters . The most used fixed windows are; rectangular, Hanning, Hamming and Blackman windows . FIR filter design using the window method exhibits oscillatory behavior around the discontinuity of the ideal frequency response and does not control the passband and stopband ripples. Even if we increase the number of coefficients in the FIR filter approximation, these ripples concentrate near the passband edge frequency and cannot be eliminated. To reduce this effect, we propose a solution in the frequency domain. In order to reduce the *Gibbs phenomenon*, we try smoothing the ideal response frequency to remove the discontinuities at abrupt transitions of the rectangular window by choosing a Gaussian function as a smoother filter response. The impulse response is obtained by applying inverse Fourier transform to the smoothed frequency response, and then the digital version of this response must be truncated and shifted to be causal. After this, a DFT is used to generate the frequency response of the FIR filter.

THEORY

The required frequency response of a desired filter of bandwidth B should look like a rectangular function , from $-B/2$ to $B/2$. In order to design this filter type, we propose to smooth the ideal response frequency to remove the discontinuities at abrupt transitions of the rectangular window. We choose a Gaussian function as a smoother filter response in the frequency domain. Consequently, the desired frequency response is given by the following convolution:

$$D(f) = R(f) * G(f) \qquad \text{………..} \ (1)$$

where the rectangular function is defined by:

$$R(f) = \begin{cases} 1\,, & -\frac{B}{2} \leq f \leq \frac{B}{2} \\ 0, & otherwise \end{cases} \qquad \text{……..} \ (2)$$

and the Gaussian function is defined by the following expression:

$$G(f) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-f^2}{2\sigma^2}} \qquad \text{………} \ (3)$$

The analytical expression of the frequency response for continuous-time filter can be expressed as:

$$D(f) = R(f) * G(f)$$

This frequency response has two independent parameters, namely the bandwidth of the rectangular function (*B)* and the adjustable shape parameter (σ), which is the standard deviation of the Gaussian function.

To study the effect of the parameter σ on the filter specifications, we compute and plot the frequency response for the continuous-time filter with fixed bandwidth B and different values of σ. By adjusting the parameter σ, the resulting frequency response can closely approximate the desired response. Figure 1 shows the frequency response of the continuous-time filter with fixed bandwidth $B = 4$ and different values of σ.

Once the desired frequency response has been obtained, the analytical expression of the impulse response is calculated using inverse Fourier transform.

$$dc(t) \ = \ B \ \mathrm{sinc}\ (Bt) \ \exp(-2\pi^2\sigma^2t^2) \ \text{…….} \ (4)$$

Figure 5 shows the sinc function with $B = 4$, Gaussian function with $\sigma = 0.5$ and the resultant impulse response which is defined as the multiplication of the Gaussian and the sinc function. As we know, the effective width of the desired impulse response should be as small as possible.

As illustrated in Fig. 3, the resulting filter is approximately band-limited. Consequently, the impulse response can be sampled with sampling frequency fs. To obtain the discrete-time filter's impulse response $d(n)$, the continuous-time filter's impulse response $dc(t)$ is sampled with sampling period $Ts = 1/fs$.

$$d(n) = Ts\,dc(nTs) \quad \ldots\ldots\ldots(5)$$

The frequency response of this FIR filter has three independent parameters: the bandwidth of the rectangular function B, the adjustable parameter σ and the filter order N. For desired value of B, both parameters σ and N can be adjusted to satisfy the requirements of a desired filter in frequency domain. To study the effect of the filter order N on the FIR filter specifications, we plot the frequency response for fixed values of B and σ and with different values of N and compare them with the frequency response of the continuous-time filter as shown in Fig. 6.

Figure 7 shows the frequency response of the FIR filter with $B = 4$, $N = 80$ and different values of σ. Figure 8 shows the frequency response of the resulting filter with $B = 4$, $\sigma = 0.0025$ and different values of N where the frequency response decreases rapidly and does not have any noticeable sidelobe.

MATLAB CODE

```
clc ;
clear ;
close all ;

Fs = 100 ;
B = 4 ;

f = -10 : 1/Fs : 10 ;

% Rectangular Function is given by R
R = rectpuls(f,B) ;

figure('Name','Rectangular Function')
plot(f,R) , grid on ;
xlabel('Frequency') ;
ylabel('Magnitude') ;
axis([-10 10 0 1.5]) ;


%% GAUSSIAN FUNCTION

sigma = [0.3 0.2 0.1 0.05 0.0025] ;

% Plotting Gaussian Function for Different Values of Sigma
% Gaussian Function is given by G
figure('Name','Gaussian Function for different Values of Sigma') ;

for i = 1:length(sigma)

    numr = exp(-(f.^2)./(2*((sigma(i))^2)));
    denr = sigma(i)*((2*pi).^0.5);

    G = numr/denr ;

    plot(f,G) ;
    xlabel('Frequency') ;
    ylabel('Magnitude') ;
    grid on ; hold on ;
    axis([-2 2 0 10]) ;

end

legend(sprintf('sigma = %g',sigma(1)),sprintf('sigma = %g',sigma(2)),...
  sprintf('sigma = %g',sigma(3)),sprintf('sigma = %g',sigma(4)),...
  sprintf('sigma = %g',sigma(5)))


%% DESIRED FREQUENCY RESPONSE for different Values of Sigma


% Plotting Desired Frequency Response for Different Values of Sigma
% D = convolution of R and G
```

```
figure('Name','Desired Frequency Response for Different Values of Sigma')

for i = 1:length(sigma)

    numr = exp(-(f.^2)./(2*((sigma(i))^2)));
    denr = sigma(i)*(sqrt(2*pi));

    G = numr./denr ;

    D = conv(R,G) ;

    f1 = -20:1/Fs:20 ;

    plot(f1/10,20*log10(D)) , grid on ;
    xlabel('Frequency') ;
    ylabel('Magnitude in dB') ;
    hold on ;
end

legend(sprintf('sigma = %g',sigma(1)),sprintf('sigma = %g',sigma(2)),...
  sprintf('sigma = %g',sigma(3)),sprintf('sigma = %g',sigma(4)),...
  sprintf('sigma = %g',sigma(5)))


%% R,G and D in Time Domain

figure('Name','R , G , D in Time Domain')
t = -10 : 1/500 : 10 ;

% r is ifft of R
% r = sinc function in time domain

r = B.*sinc(B.*t) ;

subplot(3,1,1)
plot(t,r) ;
xlabel('Time') ;
ylabel('Amplitude') ;
title('Rectangular function r(t)') ;


subplot(3,1,2)
% g is ifft of R
% g = gaussian function in time domain
g = exp(-2*(pi*0.5.*t).^2) ;
plot(t,g) ;
xlabel('Time') ;
ylabel('Amplitude') ;
title('Gaussian function g(t)');


subplot(3,1,3)
d = r.*g ;
dc_t = B*sinc(B.*t).*exp(-(2*pi*0.5.*t).^2)  ;
plot(t,dc_t) ;
xlabel('Time') ;
ylabel('Amplitude') ;
title('Desired Impulse Response d(t)');

```

```
figure('Name','R , G , D in Time Domain')
plot(t,r,t,g,t,dc_t) , grid on ;
xlabel('Time') ;
ylabel('Amplitude') ;
axis([-5 5 -1 4]) ;
legend(sprintf('sinc'),sprintf('gaussian function'),...
  sprintf('impulse'))



%% Varying N (Order of FIR Filter)

figure('Name','Frequency Response for sigma = 0.2 , B = 4 and diff
values of N')
N = [20 40 60 80] ;

for j = 1:length(N)

    n = -N(j)/2 : N(j)/2 ;

    dc_n = 0.1.*B*sinc(B.*n.*0.1).*exp(-(2*pi*0.2.*n.*0.1).^2) ;
    [H, w] = freqz(dc_n) ;
    plot(20*log10(abs(H))) , grid on ;
    xlabel('Frequency') ;
    ylabel('Magnitude in dB') ;
    hold on ;

end

legend(sprintf('N = %g',N(1)),sprintf('N = %g',N(2)),...
  sprintf('N = %g',N(3)),sprintf('N = %g',N(4)))

%%  Varying sigma

% Order = 80 and B = 4

M = 80 ;

if mod(M,2) == 0
    n = -M/2 : 1 : M/2 ;
else
    n = -(M-1)/2 : (M-1)/2 ;
end

dc_n = 0.1.*B*sinc(B.*n.*0.1).*exp(-(2*pi*0.5.*n.*0.1).^2) ;

figure('Name','Frequency Response for Order = 80 , B = 4 and diff
values of sigma')

for i = 1:length(sigma)

    dc_n = 0.1.*B*sinc(B.*n.*0.1).*exp(-(2*pi*sigma(i).*n.*0.1).^2)
;
    [H, w] = freqz(dc_n) ;
    plot(20*log10(abs(H))) , grid on ;
    xlabel('Frequency') ;
    ylabel('Magnitude in dB') ;
    hold on ;
end

```

```
legend(sprintf('sigma = %g',sigma(1)),sprintf('sigma = %g',sigma(2)),...
  sprintf('sigma = %g',sigma(3)),sprintf('sigma = %g',sigma(4)),...
  sprintf('sigma = %g',sigma(5)))


%% Varying N (Order of FIR Filter)

figure('Name','Frequency Response for sigma = 0.0025 , B = 4 and diff values of N')
N = [1000 2000 3000 4000 5000 6500] ;

for j = 1:length(N)

    n = -N(j)/2 : N(j)/2 ;

    dc_n = 0.1.*B*sinc(B.*n.*0.1).*exp(-(2*pi*0.0025.*n.*0.1).^2) ;
    [H, w] = freqz(dc_n) ;
    plot(20*log10(abs(H))) ,grid on ;
    xlabel('Frequency') ;
    ylabel('Magnitude in dB') ;
    hold on ;

end

legend(sprintf('N = %g',N(1)),sprintf('N = %g',N(2)),...
  sprintf('N = %g',N(3)),sprintf('N = %g',N(4)),sprintf('N = %g',N(5)),...
  sprintf('N = %g',N(6)))

%% sigma = 0.0225 ; order = 20 ;

sig = 0.0225 ;
ord = 20 ;

n = -ord/2:ord/2 ;
dc_n = 0.1.*B*sinc(B.*n.*0.1).*exp(-(2*pi*sig.*n.*0.1).^2) ;

figure('Name','Frequency Response for sigma = 0.0225 , N = 20')
[H, w] = freqz(dc_n) ;
plot(20*log10(abs(H))) ,grid on ;
xlabel('Frequency') ;
ylabel('Magnitude in dB') ;
```

OUTPUT

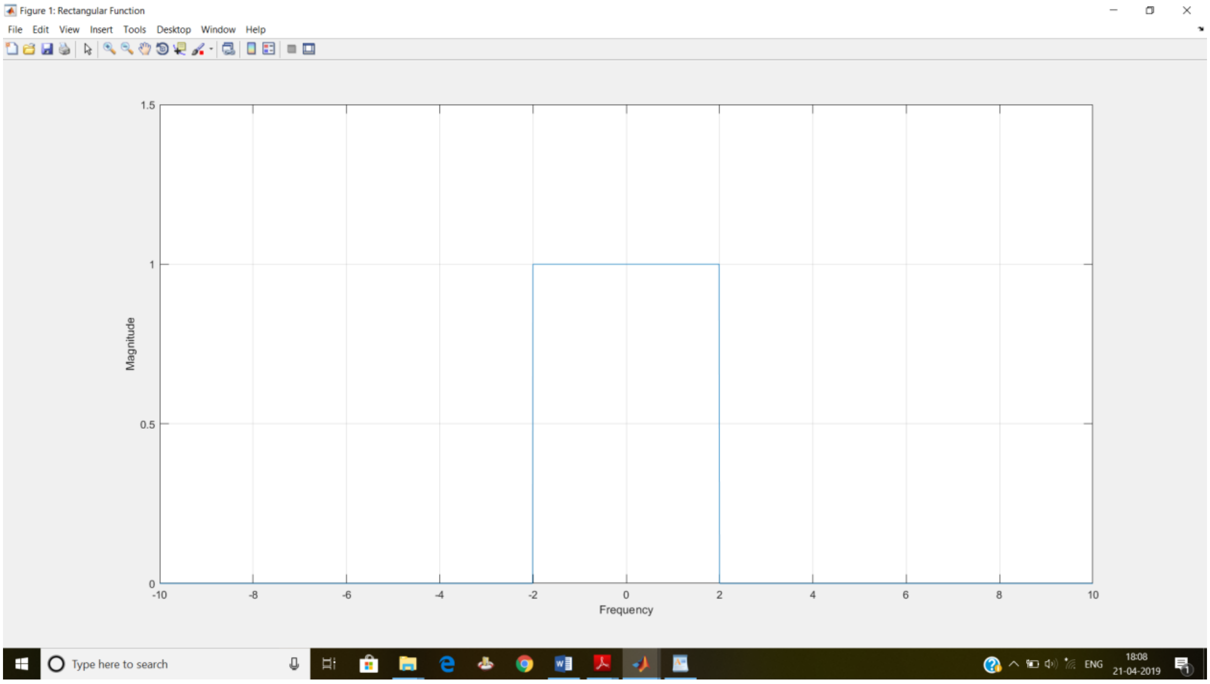


Figure 1 : Rectangular Function

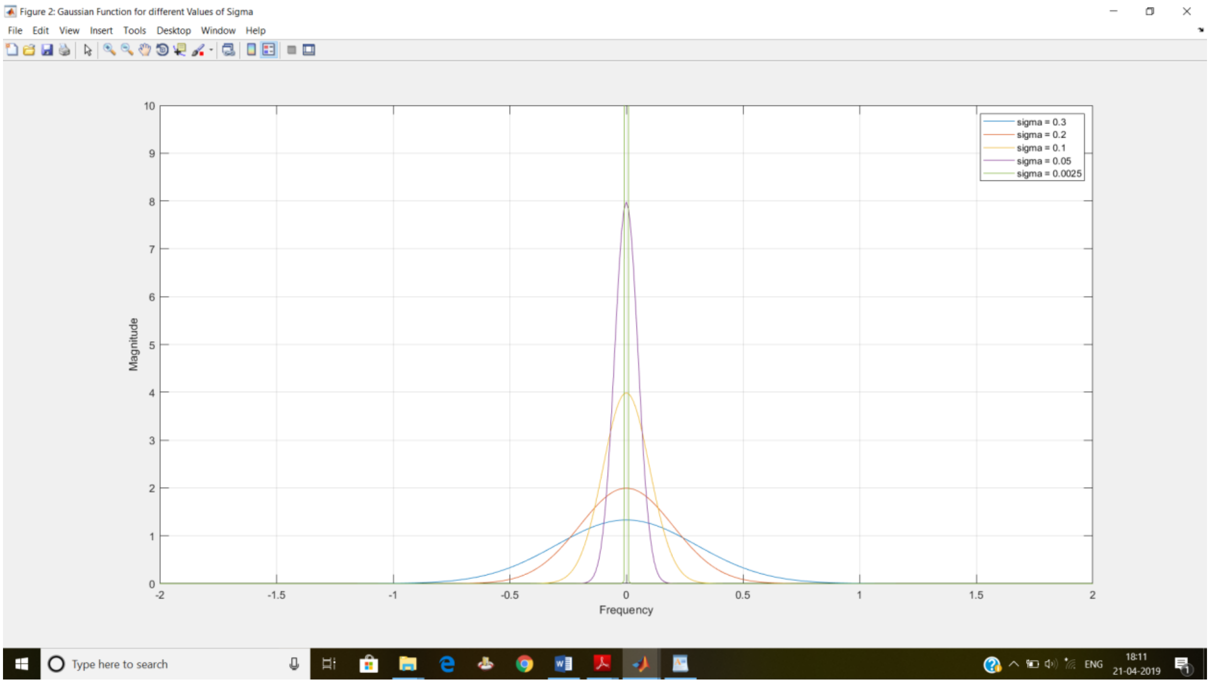


Figure 2 : Gaussian Function for different Values of Sigma

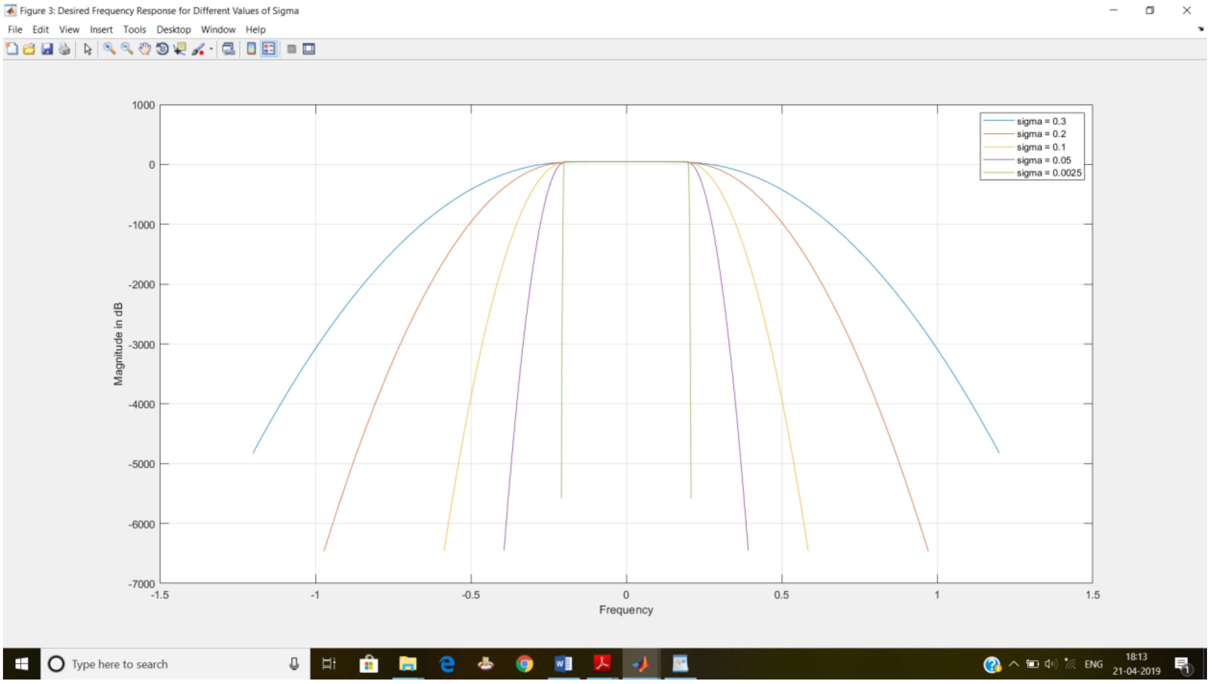


Figure 3 : Desired Frequency Response for different Values of Sigma

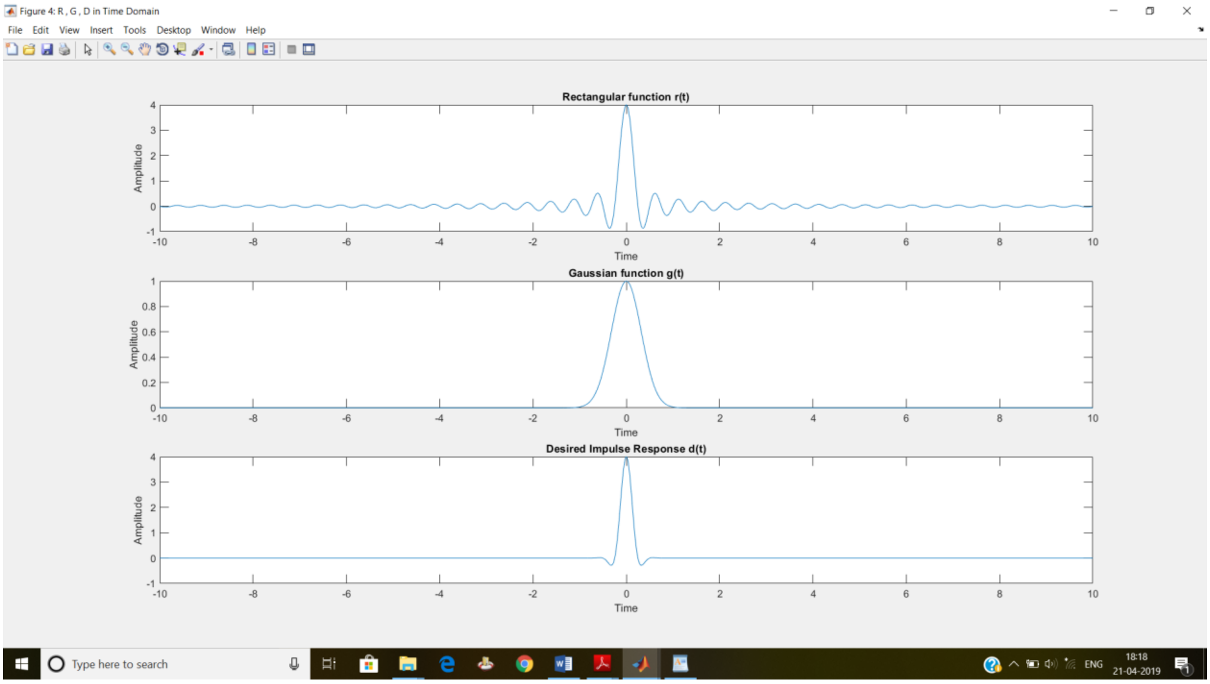
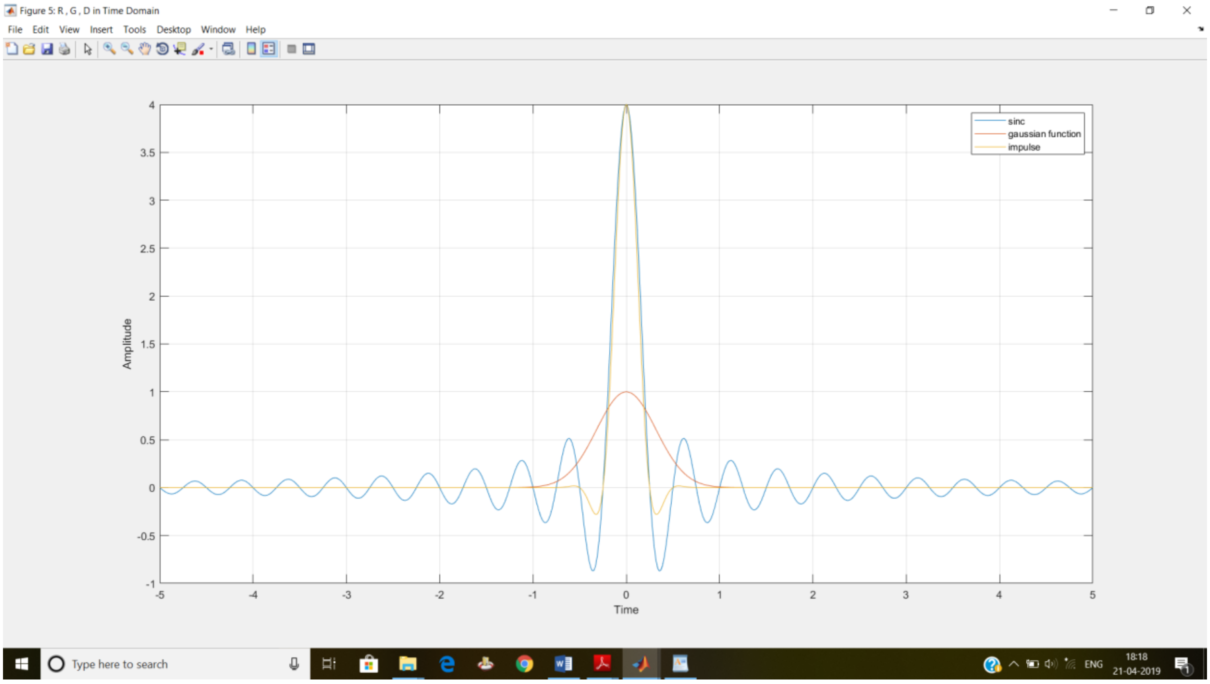
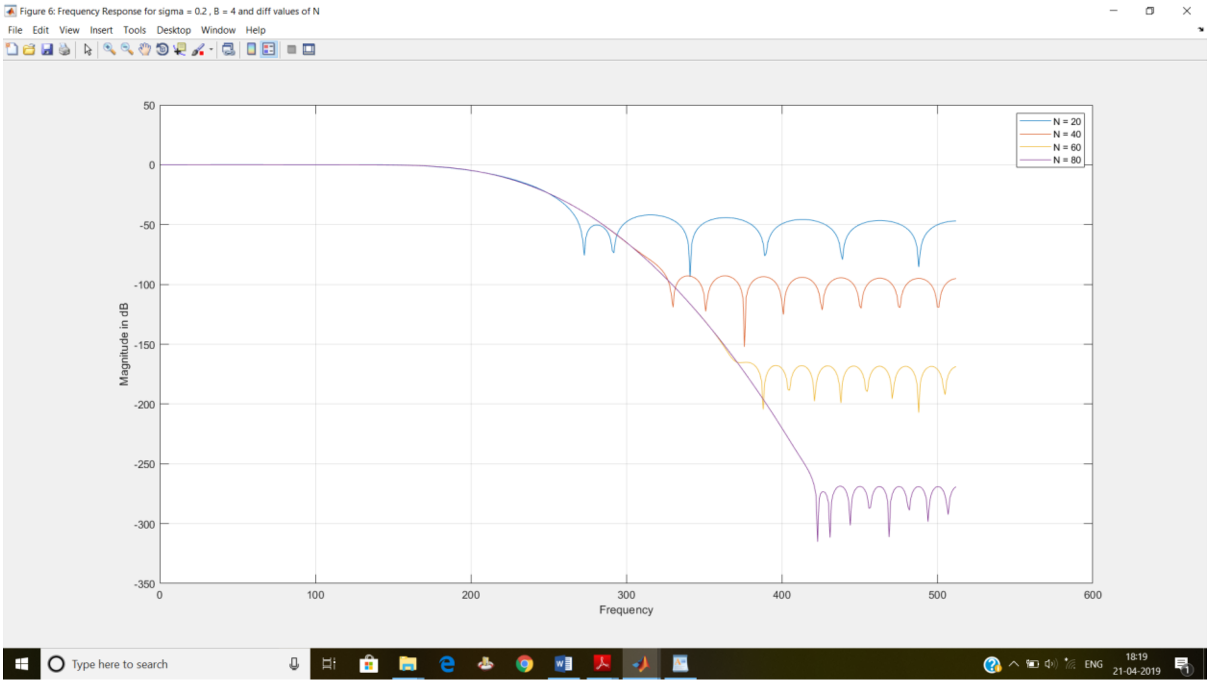


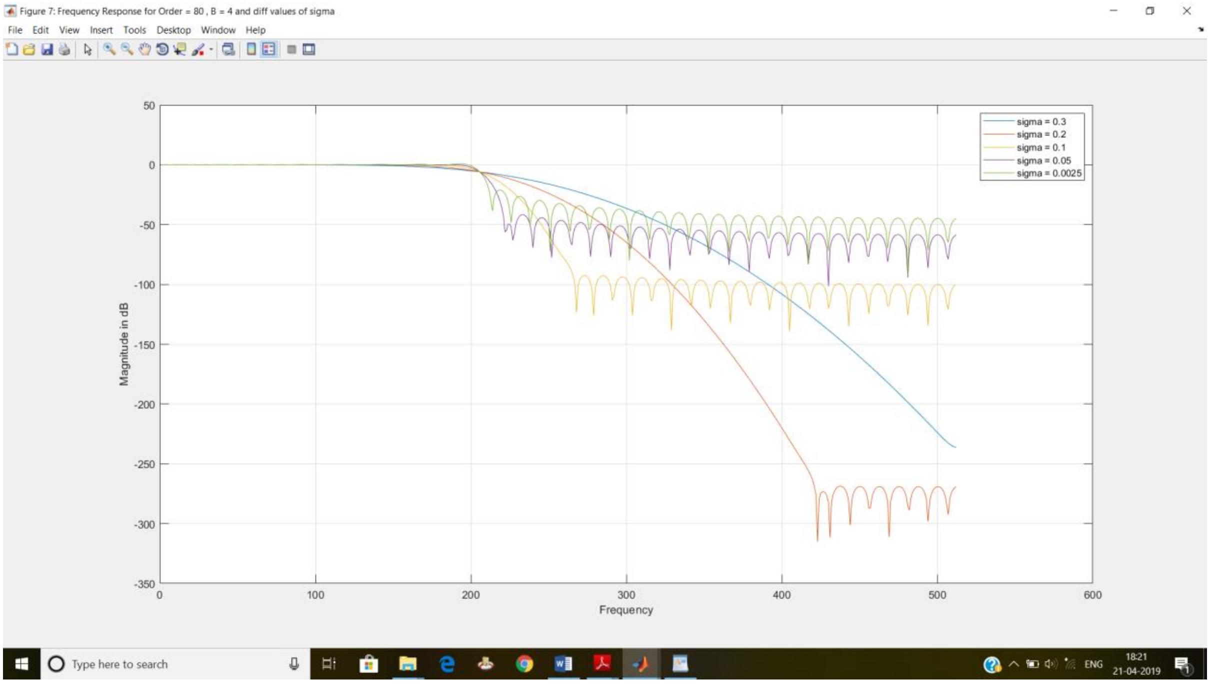
Figure 4 : R , G , D in Time Domain



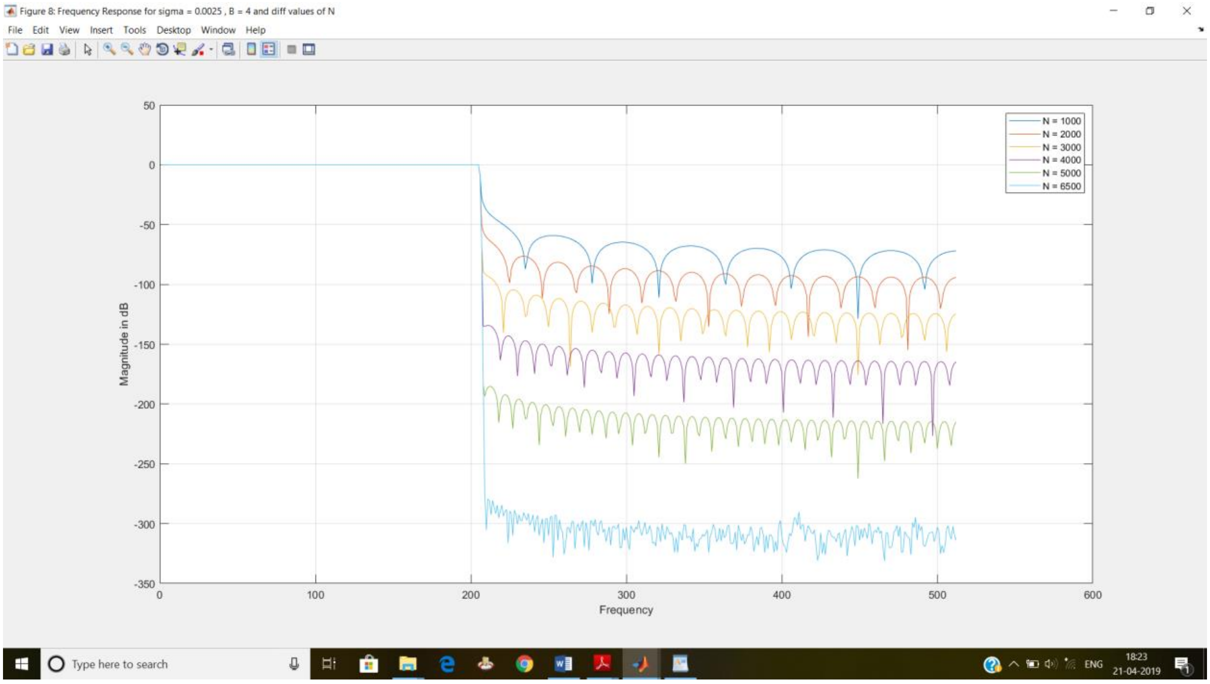
<u>Figure 5 :</u> R , G , D in Time Domain



<u>Figure 6 :</u> Frequency Response for sigma = 0.2 , B = 4 and different values of N .



<u>Figure 7 :</u> Frequency Response for Order = 80 , B = 4 and different values of sigma .



<u>Figure 8 :</u> Frequency Response for sigma = 0.0025 , B = 4 and different values of N .

OBSERVATIONS

1) (i) We can notice that the effect of convolution with a Gaussian is to smooth out the sharp transitions of the rectangular function according to the standard deviation value (σ).

   (ii) From Figure 3 , For larger values of σ, the frequency response becomes narrower at the top and wider at the bottom, whereas for small values of σ the frequency response becomes very close to the ideal response.

   (iii) To reduce the transition band, we must diminish the value of σ.

2) From Fig. 5 , we notice that the impulse response of the filter is a narrow function where the two functions have a mutual contribution.

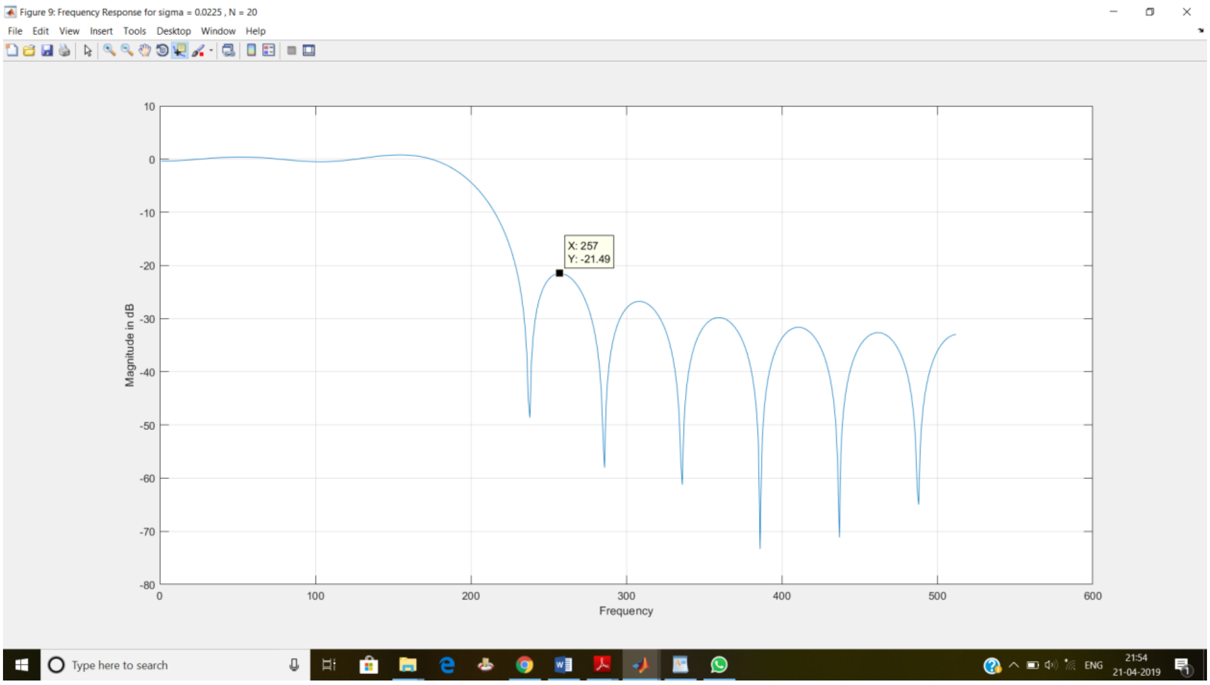
3) The Gaussian function reduces the side lobes of the sinc function .

4) From Figure 6 , The filter has no noticeable ripples in the passband and preserves practically the shape of the frequency response of the continuous filter according to the filter order N. We can also see that the larger N is, the better the frequency response of the FIR filter.

5) From figure 7 , For fixed N, we can notice that the transition width is reduced when σ diminishes, whereas the stop band attenuation is diminished.

6) From figure 8 , To increase the stop band attenuation, we must increase the filter order N .

RESULT



1) For order $N = 20$, this algorithm achieves practically the maximum stop band attenuation of -21.49 dB .
2) Compared to the other algorithms, our filter has no noticeable ripples in the passband .
3) To reduce the transition width and increase the stop band attenuation, we must diminish the value of σ and increase the filter order N .

CONCLUSION

In this project , an efficient way of designing linear-phase FIR filters using a smoothed rectangular function as a frequency response is implemented . The smoothed rectangular function should be as close as possible to the ideal frequency response. In the approach, an analytical study in the frequency domain and once the desired frequency response is obtained, the impulse response is calculated directly as inverse Fourier transform of the frequency response.

The main advantage of this algorithm is its flexibility due to the three tunable parameters, which allows the adjustment of the compromise between the stop band attenuation and the transition width.

This algorithm can be used to design reconfigurable filters for many applications in communication systems. On the other hand, the major drawback of optimization algorithms is that they require a high computational cost to achieve the desired specifications. They depend on a lot of parameters, and the choice of them has an important impact on optimization performance. As conclusion, the proposed algorithm do well in designing FIR filters of any order.

BIBLIOGRAPHY

Websites :

1. https://wolframalpha.com/
2. https://www.stackoverflow.com/
3. https://in.mathworks.com/

Books :

1. (Prentice Hall Signal Processing Series) Alan V. Oppenheim, Ronald W. Schafer - Discrete-Time Signal Processing-Prentice Hall (2010) .

2. John G Proakis , Dimitris G Manolakis - Digital Signal Processing-Pearson Education (2013).