ROBUST DIGITAL FILTERS
MINIMAX FIR FILTER

Department of Electrical and Computer Engineering

ECE 503 Optimization for Machine Learning

PROJECT REPORT
Submitted by

Alinstein Jose, (V00918748)

University of Victoria

Date: August 16th, 2019

Table of Contents

ABSTRACT

This project implements a robust digital filter, where robust digital filters are the filters that offers optimal performance under variation of filters parameters. The characteristics performance of robust digital filters against the parameter uncertainties are discussed in project report. The performance and design formulations of robust FIR filters in L_∞ (minimax) are discussed as nonsmoothed convex problems. In this project, L_∞ -robust FIR filter is designed (optimized) using an accelerated sub-gradient algorithm and implemented in MATLAB.

Keyword: accelerated sub-gradient algorithm, FIR filters

1.INTRODUCTION

Digital filters are used in most day to day used electronic devices. Compared to their analog counterparts, digital filters offer outstanding performance and flexibility. Designing digital filters can seem a daunting task, however, because of its seemingly endless range of implementation choices. In normal cases, a filter is designed with assuming that filter is having infinite precision and is followed by an implementation phase to deal with finite word length effects which include, for example, roundoff noise and power to two constraints imposed upon filter parameters. Many of these filters with P02 coefficient designed involving mixed-integer linear programming that does not guarantee optimal performance. However, in this project, we implement a digital filter with performance guarantee even when the filter parameters are varied independently to a certain extent. This filter combines the two procedure in previous design namely filter design and finite precision implementation phase with both Po2 and constrain and finite word length issue in terms of parameter uncertainties, into one single step.

2. THEORY

The transfer function of digital filter is represented as H(z) and x is a vector representing the coefficients of H(z) (time domain). H(ω,x) represent the frequency response of the filter with respect to coefficient x. And the desired frequency response is given by $H_d(\omega)$. The performance of the filter is determined finding the error measure, that is by quantifies the closeness between H(ω,x) and $H_d(\omega)$. Normally two method are used to compute the error, namely Least-square error or minimax error between the transfer functions. The least-square (L_2) and minimax (L_∞) is shown below:

$$\left[\int_{\Omega} W(\omega)|H(\omega,x) - H_d(\omega)|^2\, d\omega\right]^{1/2} \quad 1.1$$

$$\max_{\omega\in\Omega}[W(\omega)|H(\omega,x) - H_d(\omega)|] \quad 1.2$$

Where the Ω stands for the frequency of interest and W (Ω)>=0 is the weight function defined over Ω. For example, x* stands for the minimizer of the optimization problem (L_∞) or (L_2). The filter obtain optimal performance at H(ω, x*) , would be achieve only if its implementations are perfectly accurate. Since neither hardware nor software utilized practical implementation are not infinitely precise, thus the approximate model of H(ω, x*) is realized. The approximate model has frequency response of H (ω, x*+ δ) for some variation of δ due to

some reason ranging from power of two constraints on filter coefficients to rounding errors in multiplication using fixed point arithmetic. The design represented by (x*+ δ) is no longer a minimizer of the error measure in (1.1), the performance degradation at x*+ δ is inevitable for small δ.

2.1. Measure of Performance Robustness

Least-square (L_2) and minimax (L_∞) algorithm are used to measure error of filter with the desired filter, also considering robust performance against coefficient variations. Least-square (L_2) and minimax (L_∞) error considering coefficient variations are shown below:

$$e_2(\boldsymbol{x}) = \max_{\delta \in Br}\left[\int_\Omega W(\omega)|H(\omega, \boldsymbol{x}+\boldsymbol{\delta}) - H_d(\omega)|^2\, d\omega\right]^{1/2} \quad 1.3$$

$$e_\infty(\boldsymbol{x}) = \max_{\delta \in Br}\max_{\omega \in \Omega}[W(\omega)|H(\omega, \boldsymbol{x}+\boldsymbol{\delta}) - H_d(\omega)|] \quad 1.4$$

Here B_r denotes a small bounding box or bounding ball so that x+ δ belongs to $\delta \in Br$ defines a small region centered at the give design x in parameter space. Therefore, the function e2 and e∞ from equation (1.3) and (1.4) respectively are the largest L2 and L∞ errors over the region (**x***+ $\boldsymbol{\delta}$ where $\boldsymbol{\delta}$ belongs to B_r). The dimension of **x** in set B_r is equal to K+1. And $\boldsymbol{\delta}$ =[δ_1+ δ_2 …… δ_K].

Bounding ball is like:

$$B_r = (\ \boldsymbol{\delta} = \|\boldsymbol{\delta}\|_2 \leq r\ ) \quad (1.5)$$

Where r is radius of the ball.

In this project we are considering $|\delta_i| \leq r_{i.}$ and $\boldsymbol{\delta}$ belongs to $B_{r.}$ (1.6)

Where i=1,…,K.

2.2 Design Formulation:

In this project we will implement FIR filter using minimax (L∞) method. The minimax error for performance robustness can be minimized as shown below,

Minimize $e_\infty(\mathbf{x})$ (2.1.1)

Or

Minimize $\underset{\delta\in Br}{\max}\underset{\omega\in\Omega}{\max}[W(\omega)|H(\omega,x+\delta)-H_d(\omega)|]$ (2.1.2)

The objective function is minimized with respect to parameter vector **x.** Since we are considering robust filter, we have to choose linear phase FIR filter only. The length of FIR filter is odd length N, and magnitude of filter is symmetric with respect to axis. The transfer function of filter is given by:

$$H(z) = \sum_{i=0}^{N-1} h_i(z^i) \quad (2.2)$$

$$H(\omega,\boldsymbol{x}) = e^{-jK\omega}c(\omega)^T\boldsymbol{x} \quad (2.3)$$

Where K=(N-1)/2, which denotes the group delay of the filter.

$$c(\omega) = [1,\cos(\omega),\cos(2\omega),\dots\dots\dots,cos(K\omega)\,]^T \quad (2.4.1)$$

$$\boldsymbol{x} = [h_k, 2h_{k-1}, 2h_{k-2}, \dots\dots\dots 2h_0]T \quad (2.4.2)$$

In the paper [1], it is proved that both objective functions $e_\infty(\mathbf{x})$ in (1.3) and $e_2(\mathbf{x})$ (1.4) are convex function with respect to **x**. However, the functions are not differentiable because each function is defined as maximum of the certain family of functions involving either square roots or operation of taking absolute value. So, the objective functions are $e_\infty(\mathbf{x})$ in (1.3) and $e_2(\mathbf{x})$ in (1.4) nondifferentiable convex functions. Since the objective function ($e_\infty(\mathbf{x})$) are not differentiable, we must find subdifferential rather than differential of the function to minimize function.

The subdifferential of $e_\infty(\mathbf{x})$ is:

$$g = \partial\, e_\infty(\mathbf{x}) = W(\omega^*)\, c(\omega *)(\nabla|y|) \quad (2.5)$$

Where: ω^* and δ^* are determined by equation (2.6). Detailed explanation is shown in algorithm part.

ω^* and $\delta^* = \arg_{\omega^*,\delta^*} \left(\max_{\delta \in Br} \max_{\omega \in \Omega} W(\omega)|c(\omega)^T(\boldsymbol{x}+\boldsymbol{\delta}) - A_d(\omega)|\right)$

$$y=[c(\omega^*)^T x-[A_d(\omega^*)-c(\omega^*)^T)\ \delta *]] \quad (2.7)$$

and, $$\nabla|y| = \begin{cases} 1 & if\ y > 0 \\ -1 & if\ y < 0 \\ 0 & if\ y = 0 \end{cases}$$

The $W(\omega *)|c(\omega *)^T(\boldsymbol{x}+\boldsymbol{\delta}) - A_d(\omega *)|$ is convex means that

$W(\omega *)|c(\omega *)^T(\boldsymbol{x}+\boldsymbol{\delta}) - A_d(\omega *)| \geq e_\infty(\mathbf{x})+(\mathbf{y}-\mathbf{x})^T\mathbf{g}$

For any value **y** and **x,** where **g** is the sub-differential of

$W(\omega *)|c(\omega *)^T(\boldsymbol{x}+\boldsymbol{\delta} *) - A_d(\omega *)|$.

By definition we also have :

$e_\infty(\mathbf{x}) \geq W(\omega *)|c(\omega *)^T(\mathbf{y}+\boldsymbol{\delta}) - A_d(\omega)|$.

From both equations we can conclude that $e_\infty(\mathbf{y}) \geq e_\infty(\mathbf{x})+ (\boldsymbol{y}-x)^T\mathbf{g}$.

Hence, we can conclude that **g** is the sub-differential of the objective function.

3.Algorithm:

Step 1)

Desired Amplitude response variable $A_d(\omega)$, width of the filter is considered as N=21 are created, where passband edge is $\omega_p=0.4\pi$ and stopband edge is $\omega_s=0.5\pi$ are initialized . $W(\omega)$ is made as 1 since we consider only in region [0 : 4π U 0.5π : 1]. The bounding box as (1.6) with $r_i=0.005$ for i=0,1,2,…,10;

$B_r=(\ \boldsymbol{\delta}=|\ \delta_i\ | \leq 0.005$, for i=0,1,2,….,10)

The set Ω_d consist of 100 frequency grids that are between [0, 0.4π] and [0.5π, π].

Step 2)
Generated a least square lowpass filter with the same passband and stopband edges with MATLAB inbuild function (fir1). And we initialize $\mathbf{x}_0$ as output coefficient of filter from least square lowpass filter.

Step 3)

Start a loop with k=0,1,….N_t, Nt=10000.

Step 3.1) Initially, we compute sub-gradient **g** in (2.5) at point x_0. Since our objective function is non-smooth convex function, we have to use sub-gradient rather than the gradient of the function.

The subdifferential of $e_\infty(\mathbf{x})$ is:

$$\partial e_\infty(\mathbf{x}) = W(\omega^*)\, c(\omega *)(\nabla|y|)$$

Where: y can be compute from the following equation.

$$y = [c(\omega^*)^T\mathbf{x}-[A_d(\omega^*)-c(\omega^*)^T)\,\delta^*]]$$

for given x,

$$\nabla|y| = \begin{cases} 1 & if\ y > 0 \\ -1 & if\ y < 0 \\ 0 & if\ y = 0 \end{cases}$$

and optimal ω : ω^*(frequency with maximum error) and optimal δ: δ^* (δ with maximum error) can be calculated using the equations shown below.

$$\omega^* = \max_{\omega \in \Omega d} W(\omega)\left(\sum_{i=0}^{K} r_i|\cos(i\omega)| + |A_d(\omega) - c(\omega)^T\boldsymbol{x}|\right)$$

$$\delta^* = -sign(A_d(\omega *) - c(\omega)^T\boldsymbol{x})\left(sign(c(\omega)^T)\right).\boldsymbol{r}$$

Step 3.2)

Update the filter coefficient x by computing the equation shown below. At first iteration, consider x_k and x_{k-1} are equal. Compute the value gamma and beta value as shown below. Both γ_k and β_k are proportional to 1/(k+1).

$$\mathbf{x}_{k+1} = x_k - \alpha_k g_k + \beta_k(x_k - x_{k-1})$$

Last term in above equation contributes an acceleration or momentum, which pushes the current iteration like a heavy ball moving down hill faster. This helps in better convergence of the objective function.

Update the value of gamma and beta in each iteration.

$$\gamma_k = \frac{0.16}{k+1} \quad \text{and} \quad \beta_k = \frac{0.06}{k+1}$$

The step size α_k is calculated using equation shown below, where e_{best}^{k} is minimum of $e\infty(\boldsymbol{x_k})$ for k=1,….,(last iteration).

$$\alpha_k = \frac{e\infty(\boldsymbol{x_k}) - e_{best}^{k} + \gamma k}{||\boldsymbol{g}_k||_2^2}$$

The values of α_k is always greater than 0.

Step 3.3) If the k is less than the number of iterations required then go back to step 3.1, else break loop and go to next step.

Step 4) Display the amplitude response of the coefficient computed in final iteration. Also, plot the objective function with respect to iteration number.

4.MATLAB CODE:

```
clc
clear all
clear,
close


%%
%fp:passband edge
%fa:stopband edge
%N: filter length
%gam:gama variable
%b:Beta variable
%M:Number of frequency grids
%K:number of iteration for
%Size of bounding box

%%
```

```
%Intializing variables
ri=ones(11,1)*0.005;
fp=0.4;
fa=0.5;
N=21;
M=100;
K=10000;
gam=.16;
b=0.08;
N1 = (N + 1)/2;

fp = fp*pi;
fa = fa*pi;
Mp = round(M*fp/(fp+pi-fa));
Ma = M - Mp;
f1= 0:fp/(Mp-1):fp;
f2 = fa:(pi-fa)/(Ma-1):pi;
f = [f1(:); f2(:)];

%Desired Omega values
omegad = [linspace(0,0.4*pi,44),linspace(0.5*pi,pi,56)];
omegad=f';


N=21;
K=(N-1)/2;

%Weight
W=ones(1,100);
%W(41:60)=0;

%intialize x0 with least-squares lowpass filter coeffcient with 0-0.4pi as  passband and 0.6pi to
Pi stopband edges

x0= fir1(20,0.45);

x = [x0(11) 2*x0((11+1):21)]';
x2=x;

%Desired Amplitude Ad
Ad=[ones(1,44),zeros(1,56)];

%Intialize C(w) and Ri*cos(w)
for i=1:length(omegad)
```

```
    for j=0:K
       c_omega(j+1,i)=cos(omegad(i)*j);
    end

    for j=0:K
       ricos(j+1,i)=ri(j+1)*abs( cos(j*omegad(i)) );
    end

end

 alpha(1)=0.1/(1+1);
 for k=1:10000
     k

     %calculates E(x+d) for all values of Omega_Desired

     [omegastar(k),omegastar_ind(k)]=max( sum(ricos(:,:))'+abs(Ad' - c_omega(:,:)'*x(:,k))  );

     %Find the maximum delta point
     delta(:,k)=-sign(Ad(omegastar_ind(k)) -
c_omega(:,omegastar_ind(k))'*x(:,k))*sign(c_omega(:,omegastar_ind(k))).*ri;

     y(k)= c_omega(:,omegastar_ind(k))'*x(:,k)  -  (Ad(omegastar_ind(k))  -
c_omega(:,omegastar_ind(k))'*delta(:,k));


     %Gradient function

     if y(k)>0
        grad_y(k)=1;
     elseif y(k)<0
        grad_y(k)=-1;
     elseif y(k)==0
        grad_y(k)= 0;
     end


     g(:,k) = W(omegastar_ind(k))*c_omega(:,omegastar_ind(k))*(grad_y(k));

     %alpha beta Gamma calculation

     einfinty(k)= W(omegastar_ind(k)) * abs(c_omega(:,omegastar_ind(k))' * (x(:,k)+delta(:,k))  -
Ad(omegastar_ind(k)) );
     [minerror(k),minindex]=min(einfinty);

     beta(k)= 0.08/(k);
```

```
    gamma(k)=0.16/(k);
    alpha(k)=(einfinty(k)-minerror(k)+gamma(k))/((norm(g(:,k),2))^2);


    %Update the value of x
     if k==1
        x(:,k+1)= x(:,k)- alpha(k)*g(:,k);
     else
         x(:,k+1)= x(:,k)- alpha(k)*g(:,k) + beta(k)*(x(:,k) - x(:,k-1));
     end
    [minerror(k),minindex]=min(omegastar);

 end


xfinal=[ flipud(x(2:11,minindex))'/2 x(1,minindex)' (x(2:11,minindex)'/2) ];

omegad1 = [linspace(0,pi,1000)];
[h,w] = freqz(xfinal,1,1000);
plot(omegad1,(20*log10(abs(h))))
grid
title("Amplitude response of the robust minimax FIR filter")
ylabel("Amplitude")
xlabel("Normalized frequency")

figure
plot(20*log10(abs(c_omega(:,:)'*x(:,minindex))) )
grid
title("Amplitude response of the robust minimax FIR filter")
ylabel("Amplitude")
xlabel("Frequncy Grid")

iteration_num=[1:10000];

figure
plot(iteration_num,einfinty)
axis([-100 10000 0 .4])
grid
title("Plot of error(norm infinity) with repesct to iteration")
ylabel("Error(norm infinity)")
xlabel(" iteration ")

iterationnum1=[1:200]
figure
plot(iterationnum1,einfinty(1:200))
grid
```

```
axis([-10 200 0 .4])
title("Plot of error(norm infinity) with repesct to first 200 iteration")
ylabel("error(norm infinity)")
xlabel(" iteration ")

figure
plot(xfinal)
grid
axis([0 22 -.2 0.6])
hold on
plot(x0)
legend('Intial coefficients of Filter', 'Final coefficients of Filter')
title("Coefficients of Filter")
fprintf('The Coefficients of Filter are')
xfinal
```

5. RESULTS AND EVALUATIONS

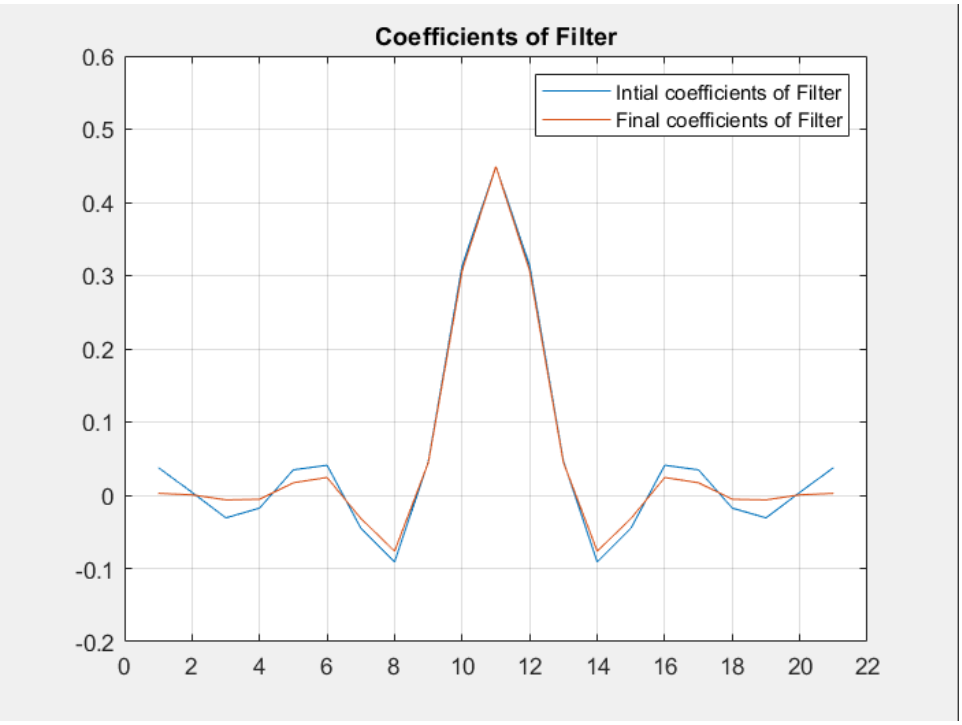

Figure 1:Coefficients of the Initial filter and final filter

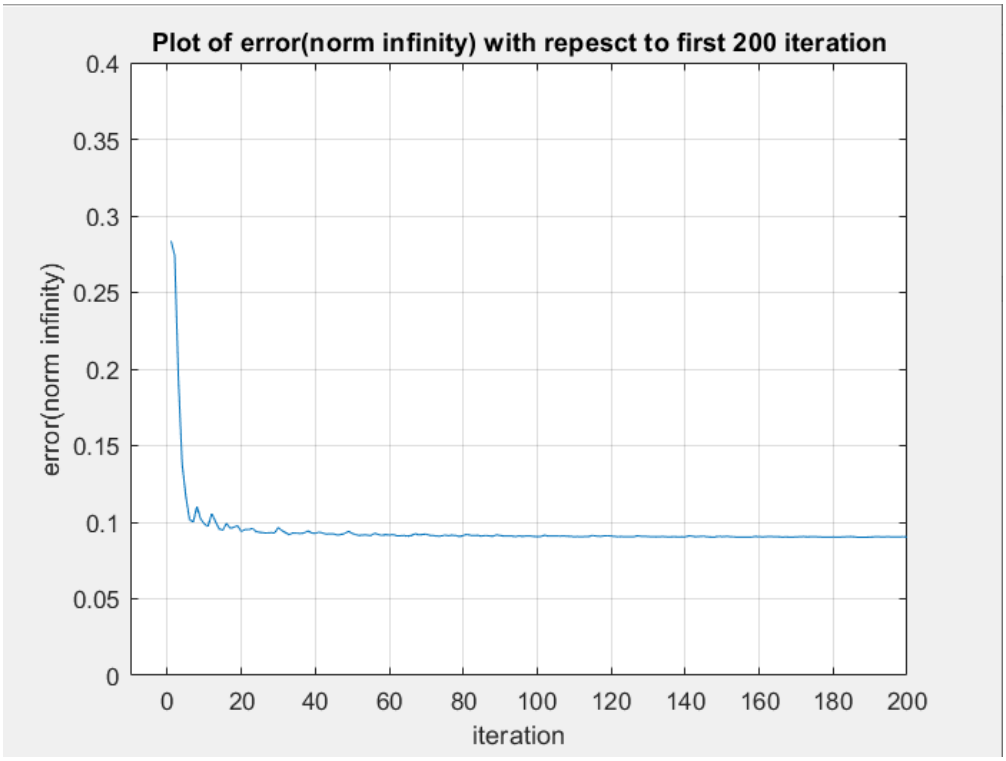

Figure 2:First 200 iterations of objective function (INFINITY NORM)

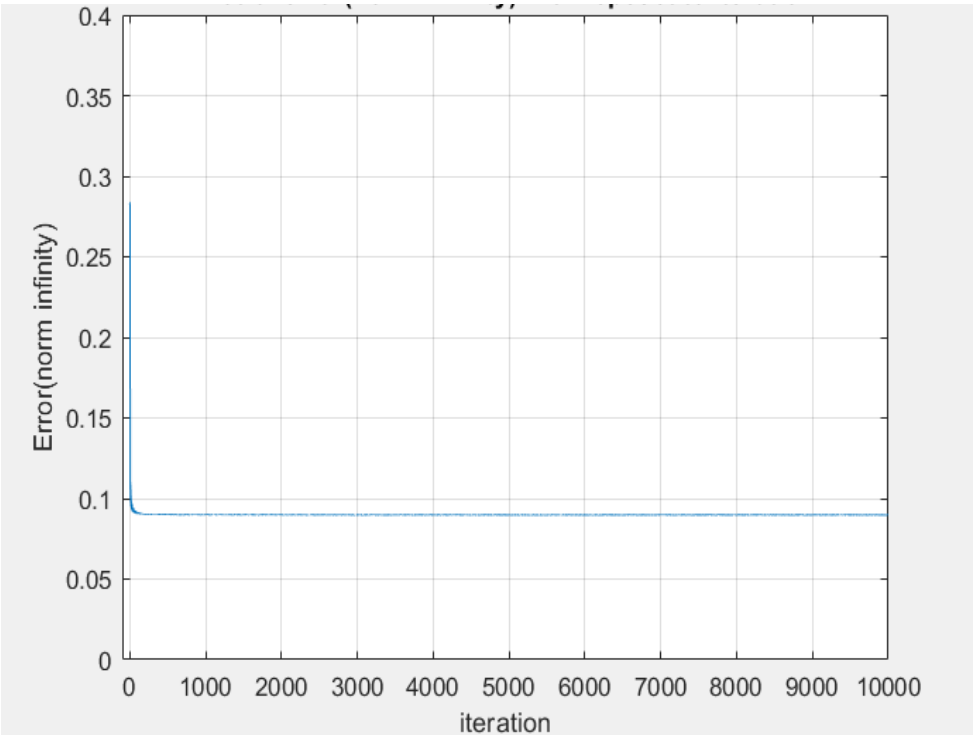

Figure 3:Figure of Error (infinity norm) with respect to 10000 iterations

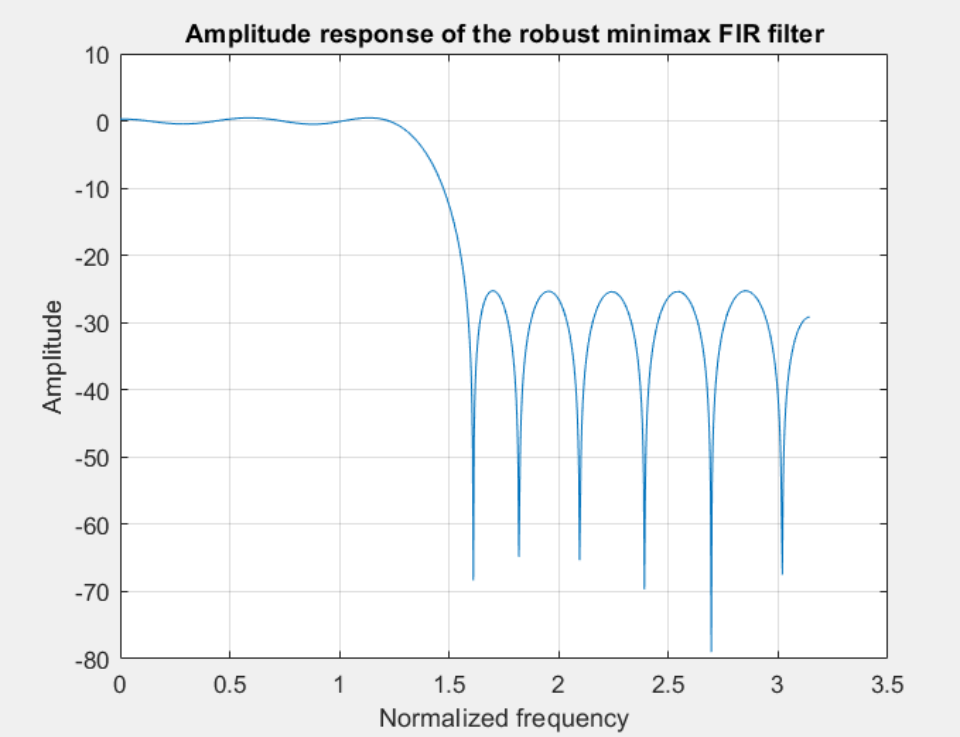

Figure 4: Amplitude Response of the optimized Filter

0.0377392576114746
0.00395589130332158
-0.0309478250752698
-0.0176170780166618
0.0346560230787361
0.0409448046086712
-0.0450878601275328
-0.0911166165644512
0.0462158743487858
0.313562923929549
0.449135853464320

Figure 5:First 11 Coefficients of the filter

6. DISCUSSION

The objective function at first iteration is ($e_\infty(x_0)$) equal to 0.283738. The error was drastically reduced in the first 50 iterations. At 50^{th} iteration the $e_\infty(x_0)$ is reduced to 0.0916314. This can be depicted from the figure 2, which shows error for the first 200 iteration. After 50^{th} iteration the objective function remained almost same. A gradual reduction was observed till 492^{nd} iteration, and then then the objective function retained almost same value till the 10000^{th} iteration. The objective function ($e_\infty(x_0)$) at 492^{nd} iteration is equal to 0.089966732924528. The figure 4 depicts the amplitude response for the robust filter that is obtained after 10000 iterations. The objective function at final iteration is ($e_\infty(x^*)$)=0.089847696835561. The 11 coefficients of the total 21 coefficients of filters are shown in Figure 5, and coefficients of the filter is symmetric with respect to axis of filter. From the amplitude response of the filter in figure 4, we depict that amplitude response resembles that of amplitude response of desired filter. Figure 1 shows the comparison of coefficients of the filter before and after optimization.

7. CONCLUSION

In this project we have designed and implemented a robust FIR filter. Such filters are expected to be of use in practice as their performance are more robust with respect to change in various constraints in their implementation. The filter's coefficients are optimized to optimal performance using accelerated sub-gradient method considering it as non-smooth convex optimization.

8.REFERENCES

[1]Robust Digital Filters Part 1 — Minimax FIR Filters
By Wu-Sheng, Takao Hinamoto
[2] W.-S. Lu, Applications of Digital Processing Techniques. 2019.
[3]W.-S. Lu and A. Antoniou, Two-Dimensional Digital Filters, New York: Marcel Dekker, July 1992.
[4] W.-S. Lu, Course notes of Advanced Mathematical optimizations.