Florida International University

College of Electrical Engineering

Digital Filters

A Practical Method to Design Equiripple FIR Filters

Author: *Pablo Gomez*, Ph.D. Candidate

Miami, November, 2001

Abstract

The design of FIR filters using Windows methods leads to good performance filters. However, sometimes there is a need to design a FIR filter that not only performs well but it is optimal. Optimization is the ability to specify a maximum error on each band of interest. This error is expressed as the absolute difference between the ideal or desired frequency response and the actual or resulting frequency response. One of the techniques to design optimal FIR filters is to minimize a *Chebyshev* error criterion. The resulting filters are known as *Equiripple* FIR Filters.

Introduction

This paper explains how to design *Equiriple* FIR Filters. We first cover some mathematical background necessary to understand how to evaluate and calculate the error function. The Remez Exchange Algorithm and its most common implementation by Parks, McClellan and Rabiner [1] are explained. Then, a practical guide to design Equiripple FIR filters is developed. Some filter design examples are provided, showing how to use the proposed technique.

Mathematical Background

Lets first begin by presenting a table of the four types of FIR filters [2][3].

Type	I	II	III	IV
Order	Even	odd	even	odd
F(θ)	1	cos(π/2)	sin(θ)	sin(θ/2)
M	N/2	(N-1)/2	(N-2)/2	(N-1)/2
θ_0	0	0	π/2	π/2

Table 1. Parameters of the four FIR filters types.

In order to minimize the error we need to define an error function *E(θ)* and a weight function *W(θ)* which defines the relative importance of the error at any given frequency θ. Then, the error function can be described as follows:

$$E(\theta) = W(\theta)\left[A_d(\theta) - A(\theta)\right] \qquad (1.0)$$

where $A_d(\theta)$ is the desired amplitude response, and *A(θ)* is the actual amplitude response.

A simple weight function *W(θ)*, could be defined as follows:

$$W(\theta) = \begin{cases} 1, \theta \in (passband) \\ 0, \theta \in (stopband) \end{cases} \qquad (2.0)$$

And the resulting amplitude response, A(θ) is defined by:

$$A(\theta) = F(\theta)G(\theta) \qquad (3.0)$$

and

$$G(\theta) = \sum_{k=0}^{M} b[k]\cos(k\theta) \qquad (4.0)$$

where *F(θ)* and *M* are obtained from Table 1.

The problem here is to obtain the coefficients *b[k]* that minimize the maximum absolute weighted error |E(θ)|, that is, to obtain

$\varepsilon = \max|E(\theta)|$ (5.0) where θ is in the operating frequency range of the filter.

The Alternation Theorem.

This theorem states that there exist at least $K+2$ frequencies θ_i, $\{0 \le i \le K+1\}$ where the maximum error, ε, occurs. That is,

$$|E(\theta_i)| = \varepsilon, \quad 0 \le i \le K+1 \qquad (6.0)$$

and

$$E(\theta_{i+1}) = -E(\theta), \quad 0 \le i \le K \qquad (7.0)$$

The last equation shows that the sign changes K+1 times, resulting in an oscillation or ripple on the band of interest.

The Remez Exchange Algorithm

The most common implementation of the Remez Exchange Algorithm is the version by Parks, McClellan and Rabiner [1][4]. Its objective is to obtain the coefficients b[k] that minimize ε. It uses the properties of the Alternation Theorem.

The first step is to find the order N of the desired filter. The following is an empirical formulae proposed by Kaiser:

$$N = \frac{-20\log_{10}(\sqrt{\delta_p \delta_s}) - 13}{2.32\left|\theta_p - \theta_s\right|} \quad (8.0)$$

where

θ_p is the passband-edge digital frequency,
θ_s is the stopband-edge digital frequency,
δ_p is the passband allowed deviation,
δ_s is the stopband allowed deviation,

and

$$\delta_p = (10^{A_p/20} - 1)/(10^{A_p/20} + 1) \quad (9.0)$$

$$\delta_s = 10^{-A_S/20} \quad (10.0)$$

where A_p and A_S are the attenuations on the passband and stopband respectively.

The following flowchart describes the steps required to implement the Remez Exchange Algorithm.

Practical Guide to Design Equiripple FIR Filters

In practice, the best way to design Equiripple FIR Filters is by using the functions ***remezord*** and ***remez***[5] included in the Signal Processing Toolbox of the MATLAB® software.

Function ***remezord*** calculates the optimal filter order, N, and the optimal frequency points and relative weights. Function ***remezord*** has 4 input parameters:

f, the vector of frequency-edges of the bands of interest (θ)
a, the vector of band amplitudes (1 to indicate passband, 0 to indicate stopband)
dev, the vector of allowed deviations on the bands (δ_p and δ_s in Equations 9.0 and 10.0)
fs, the sampling frequency

and returns the following output variables/vectors:

N, order of the filter
f0, vector of normalized frequency band edges (optimal points)
a0, vector of frequency band amplitudes
w0, vector of frequency band relative weights (optimal values for $W(\theta)$ in Equation 2.0)

Function ***remez*** calculates the coefficients b[k] in Equation 4.0. Its input parameters are exactly the output parameters of function ***remezord***. Therefore, these 2 functions have to be used together, in sequence. Function ***remez*** has 4 input parameters:

N, order of the filter
f0, vector of normalized frequency band edges (optimal points)
a0, vector of frequency band amplitudes
w0, vector of frequency band relative weights (optimal values for $W(\theta)$ in Equation 2.0)

and has one output parameter,

b, the vector of filter coefficients b[k] in Equation 4.0

The following is a general algorithm to design Equiripple FIR Filters using MATLAB:

1) User Input: Filter Type (LP,HP,BP,BR)
2) User Input: Frequency Edges (vector f, depending on the filter type)
3) User Input: Sampling Frequency (fs)
4) User Input: Attenuation on the passband (A_p)
5) User Input: Attenuation on the passband (A_s)
6) Calculate δ_p and δ_s using Equations 9.0 and 10.0 and populate vector **dev**.
7) If filter type is LP then a=[1 0]
8) If filter type is HP then a=[0 1]
9) If filter type is BP then a=[0 1 0]
10) If filter type is BR then a=[1 0 1]
11) Use the ***remezord*** function: [n,f0,a0,w] = remezord(f,a,dev,fs)
12) Use the ***remez*** function: b=remez(n,f0,a0,w)
13) Use the freqz function to obtain the h[k] coefficients
14) Plot the frequency response.

A MATLAB program, ***EquirippleFIR (***see Appendix A)**,** was written to implement the above algorithm. This program was used to design and verify the examples provided in the following section.

Equiripple FIR Filter Examples

Example 1.

Design a Low-Pass Equiripple FIR Filter with the following specifications:

1) Cutoff frequency of 1000 Hz
2) Stopband edge frequency = 1200 Hz
3) Sampling frequency = 4000 Hz
4) Passband attenuation = 0.1dB
5) Stopband attenuation = 40 dB

MATLAB program, ***EquirippleFIR*** was run with the specifications parameters. The frequency response plot in Figure 1 shows that the filter requirements were satisfied. However, the program had to be modified to increase the order of the filter. The first run of the program showed that the stopband attenuation requirement was not met (35dB as opposed to 40dB). It was found experimentally that the order of the filter has to be increased in 8 steps. The order of this filter is N=50.

To plot the passband details, the following MATLAB command was run:

*plot(f(1:520),20*log10(abs(h(1:520))));*

This command plots the first 520 points of vectors *f* and *h* as shown in Figure 2.

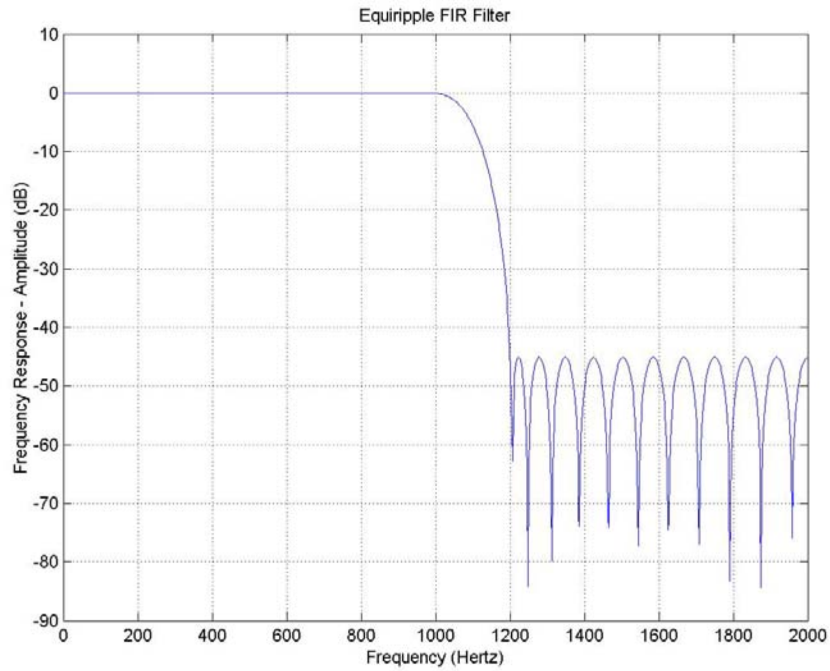

Figure 1. Equiripple Low Pass Filter Frequency Response

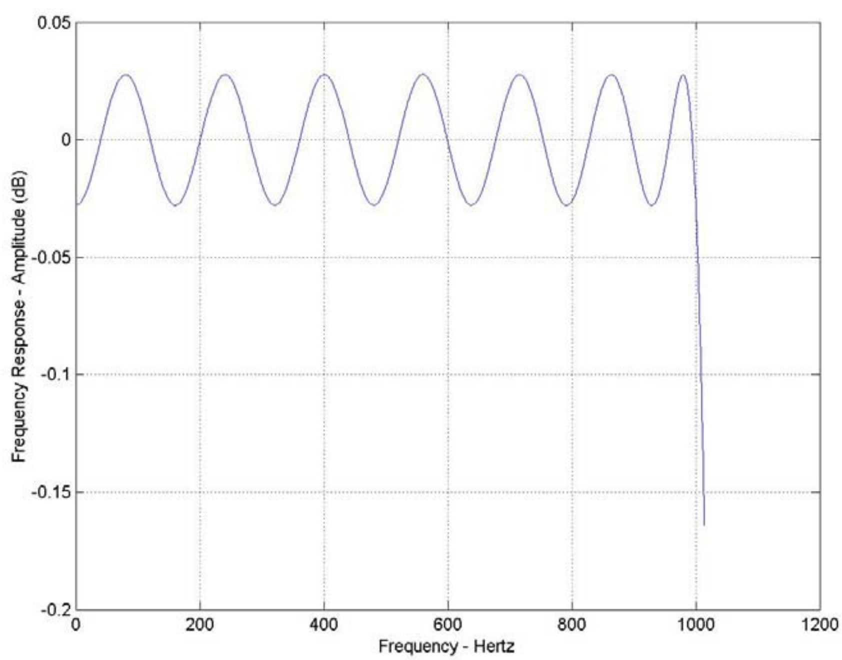

Figure 2. Equiripple Low Pass Filter - Passband details

Example 2.

Design a Band-Pass Equiripple FIR Filter with the following specifications:

1) A passband attenuation of 0.1dB in the range 1000-1200Hz
2) A stopband attenuation of 40dB for frequencies <= 800 Hz
3) A stopband attenuation of 40dB for frequencies >= 1400 Hz
4) Sampling Frequency = 4000 Hz

MATLAB program, ***EquirippleFIR*** was run with the specifications parameters. Figure 3 shows the amplitude response of a FIR filter of order $N = 50$.

To plot the passband details, the following MATLAB command was run:

*plot(f(510:620),20*log10(abs(h(510:620))));*

This command plots points 510 to 620 of vectors f and h as shown in Figure 4.

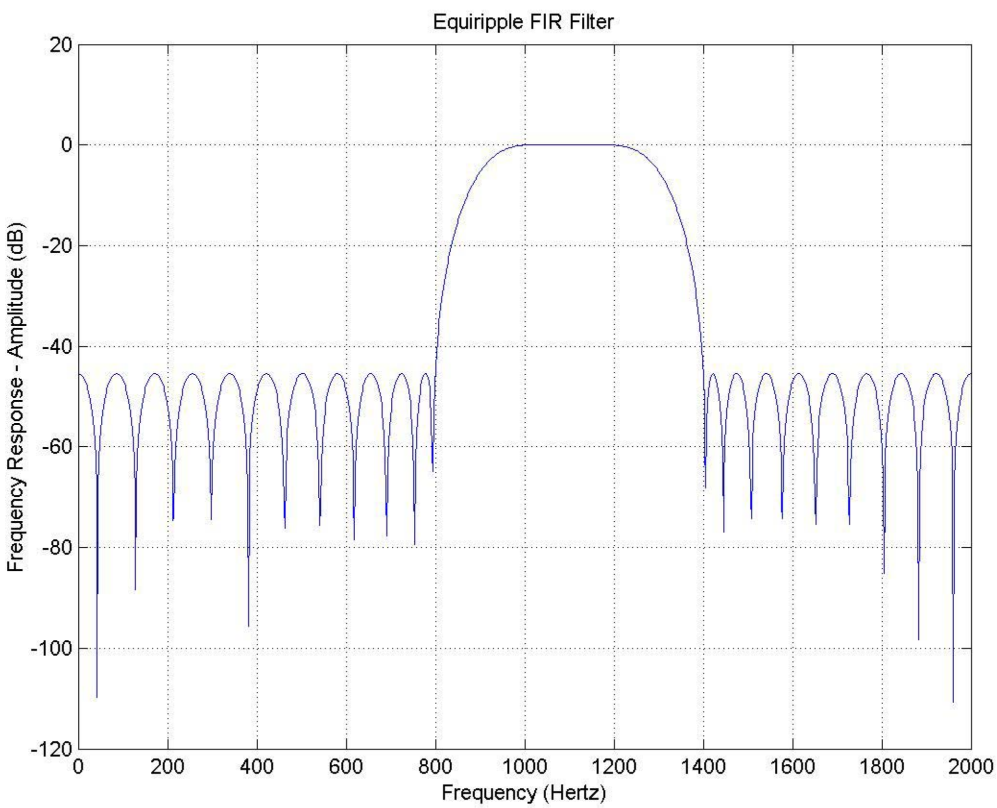

Figure 3. Equiripple Band Pass Filter Frequency Response

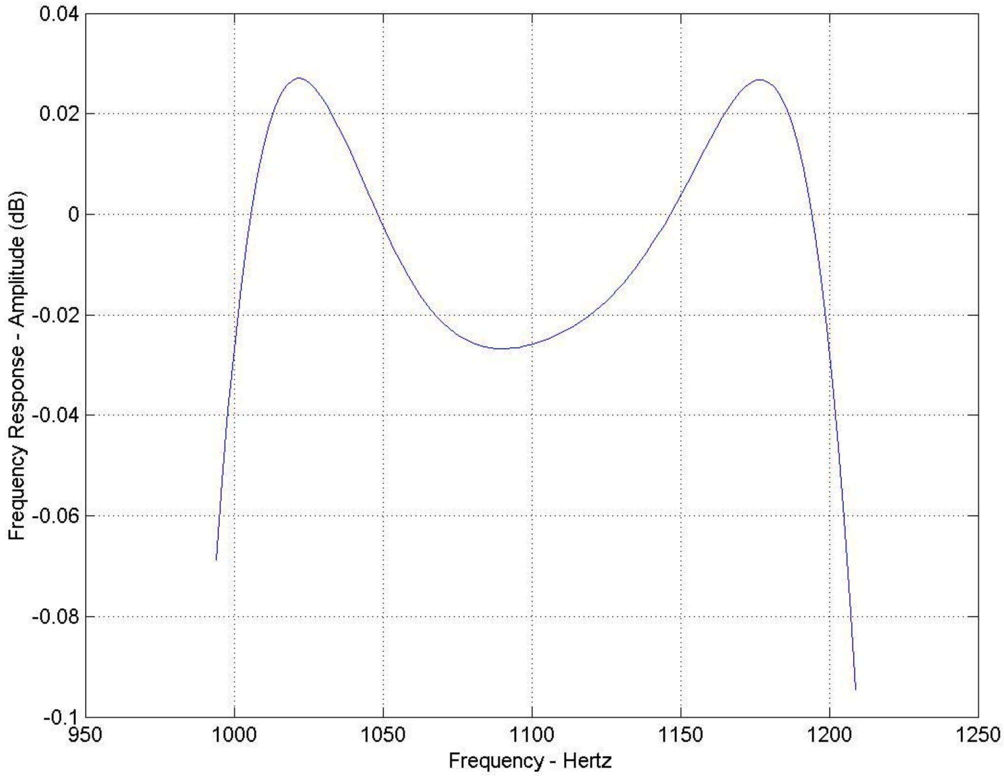

Figure 4. Equiripple Band Pass Filter –Passband Details

Results and Conclusions

The methodology to design Equiripple FIR Filters is simple and leads to good optimal FIR filters with respect to the Chebyshev norm. This technique allows the designer to explicitly control the band edges and relative ripple sizes on each band of interest. A practical guide to design these filters was developed successfully. However, the order of the filter, *N*, obtained by using the MATLAB function ***remezord*** does not yield the best results. Some experimentation is required to obtain the best value of *N*, the filter order. It was found that it is necessary to increase the order of the filter to meet the stopband attenuation requirement.

APPENDIX A: MATLAB CODE

```
%=================================================================
% Florida International University
% College of Electrical Engineering
% Digital Filters
% Purpose of Program:
%    To Design Equiripple FIR Filters using the Remez Exchange Algorithm
% Author: Pablo Gomez
% Date: 11/19/2001
%=================================================================
% Prompt User for Type of Filter (LP,HP,BP,BR)
filter_type=input('Filter Type(1=LP,2=HP,3=BP,4=BR) ');

%Obtain Frequencies:
switch filter_type
  case 1
     %LP Filter:
     Fpass = input('Enter cutoff frequency= ');
     Fstop = input('Enter stopband frequency= ');
     f = [Fpass Fstop];
  case 2
     %HP Filter:
     Fpass = input('Enter cutoff frequency= ');
     Fstop = input('Enter stopband frequency= ');
     f = [Fstop Fpass];
  case 3
     %BP Filter:
     F1 = input('Enter Lower frequency= ');
     F2 = input('Enter Upper frequency= ');
     F3 = input('Enter Lower Stopband edge frequency= ');
     F4 = input('Enter Upper Stopband edge frequency= ');
     f = [F3 F1 F2 F4];
  case 4
     %BR Filter:
     F1 = input('Enter Lower Stopband edge frequency= ');
     F2 = input('Enter Upper Stopband edge frequency= ');
     F3 = input('Enter Lower Passband edge frequency= ');
     F4 = input('Enter Upper Passband edge frequency= ');
     f = [F3 F1 F2 F4];
end

%Get Sampling Frequency from User:
fs = input('Enter Sampling Frequency= ');

%Get Attenuations:
```

```
Ap = input('Enter Passband Attenuation (dB) = ');
As = input('Enter Stopband Attenuation (dB) = ');

%Calculate Deviations (Deltap, Deltas):
DELTAp = (10^(Ap/20)-1)/(10^(Ap/20)+1);
DELTAs = 10^(-As/20);

%Populate vectors a(amplitudes) and dev(deviations) according to filter type:
switch filter_type
  case 1 %LP
     a   = [1 0];
     dev = [DELTAp DELTAs];
  case 2 %HP
     a   = [0 1];
     dev = [DELTAs DELTAp];
  case 3 %BP
     a   = [0 1 0];
     dev = [DELTAs DELTAp DELTAs];
  case 4 %BR
     a   = [1 0 1];
     dev = [DELTAp DELTAs DELTAp];
end

%Obtain the optimal parameters:
[n,f0,a0,w] = remezord(f,a,dev,fs);
%Increase the order of the filter (+8 was experimentally found to be enough
%to satisfy the design criteria):
n = n + 8;

%Obtain the best Remez approximation:
b=remez(n,f0,a0,w);

%Plot frequency response:
[h,f]=freqz(b,1,1024,fs);
plot(f,20*log10(abs(h)));
grid on;
xlabel('Frequency (Hertz)');
ylabel('Frequency Response - Amplitude (dB)');
title('Equiripple FIR Filter');
%******************* End of Program **********************************
```

References.

[1] Rabiner, L.R., McCLellan, J.H., and Parks, T.W. *FIR digital filter design techniques using weighted Chebyshev approximation*, Proc. IEEE, 63, 595-610, April 1975.

[2] Boaz Porat, *A Course in Digital Signal Processing*, John Wiley & Sons, 1997

[3] Mitra, Kaiser, *Handbook for Digital Signal Processing*, John Wiley & Sons, 1993, Table 4.84

[4] Vijay K. Madisetti, Douglas B. Williams, *The Digital Signal Processing Handbook*, CRC Press and IEEE Press, 1998.

[5] MATLAB Online Help, *Digital Signal Processing Toolbox Help.*