COMPLEX FREQUENCY RESPONSE FIR FILTER DESIGN

Worayot Lertniphonphun, James H. McClellan

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332–0250, USA.
(Tel: 404-894-6291. Fax: 404-894-8363. Email: worayot, mcclella@ece.gatech.edu)

ABSTRACT

This paper provides an algorithm for designing FIR filters that approximate both magnitude and phase of the frequency response. The new algorithm produces a filter optimized under the weighted Chebyshev norm. The algorithm starts from a first stage unoptimized filter designed by a Remez-like algorithm and then uses shifted Dirichlet kernel functions to reduce large error peaks and converge to an equiripple set of peaks. The error function is modified at each iteration by subtracting a best-fit linear combination of kernel functions due to the large error peak(s). For one length-100 example, the computation of this algorithm was less than that of the complex Remez by two orders of magnitude.

1. INTRODUCTION

Finite impulse response (FIR) digital filters have been used for many applications. Linear-phase designs are popular and easy to obtain, but introduce a delay equal to $\frac{1}{2}(L-1)$ where L is the filter length. Low-delay linear-phase filters, therefore, have to be short. Other applications might not require the linear-phase property which constrains the filter's impulse response to be real and even. If we remove the linear-phase constraint, we must do filter design by approximating a complex-valued frequency response.

Complex Chebyshev approximation is a difficult mathematical problem, because the Alternation Theorem takes a much weaker form. Nevertheless, some algorithms have been developed for this case, including one the complex Remez [1, 2] (`cremez` in MATLAB). This algorithm has two main stages: The first stage generalizes the classical Parks-McClellan algorithm. This stage converges very fast, but for many specifications, the output of the first stage is not optimum. Therefore, a second stage is needed to refine the solution and drive it to the minimum Chebyshev error. This stage uses a ascent/decent algorithm that can be proven to converge. In practice, the `cremez` algorithm does converge to the optimum for a large set of specification, but it might require an unacceptable amount of computation.

In this paper, a new second stage algorithm is proposed. This alternative algorithm was adapted from the projection method of [3] for designing linear-phase FIR filters. The algorithm in [3] jumps back and forth between the time and frequency domain (using the FFT) and imposes constraints in each domain. The iteration is, in effect, a "projection onto convex sets" method. The algorithm in [3] could be applied to the complex case, but we have found it better to change the strategy somewhat. Starting from the output of the first-stage of `cremez`, we have information needed for setting the threshold used in the iteration. In addition, we can make the threshold adaptive to obtain convergence to the minimal Chebyshev error. Finally, we are using different frequency-domain constraints to modify the frequency response. Even though this algorithm can converge quickly to have a relatively small error, making it converge to the theoretical optimum is still very slow. Nonetheless, our empirical testing shows that the performance of this new method is better than the second stage in `cremez`.

Supported by Joint Services Electronics Program Contract DAAH-04-96-1-0161.

2. ALGORITHM DESCRIPTION

2.1. Preliminary Stage

The first step in the algorithm is to generate a FIR filter whose frequency response is close to the ideal case. There are two ways of generating a good starting filter: (1) The first stage algorithm from `cremez`. This stage sometimes produces the optimum filter, but if it does not, it still forces L of the error peaks to be equal to the minimum error peak value which is also a lower bound on the final Chebyshev error and is close to the optimum error. There exist only a few peaks that have magnitude greater than this minimum peak. (2) The clipped inverse FFT (`cifft`) of the ideal response. This is equivalent to a rectangular window design. Normally, this method gives a good response except for the band edges where we have discontinuities. One drawback of this initialization for multiband filters is that the ideal filter has no transition band, so we must adjust its cutoff frequency to match the given transition band. The quality of this starting filter is less reliable, but does work quite well in some examples.

No matter which starting filter we choose, its most important property is that the minimum error peak has to less than the optimum error. If we fail to have this property, we will not be able to run our algorithm to get the optimum filter. In addition we need an estimate of the final optimum error, which can be obtained from a linear-phase filter with similar specs, or from the minimum error from the first stage of `cremez`. In a sub-optimum error function we would like to measure how close to the optimum we might be, so we can use the ratio of the maximum error peak to the minimum error peak as an indicator of how close to equiripple we are.

2.2. Modify Error

The optimum Chebyshev-norm filter has an equiripple error and the number of its equal error peaks can be counted. To get the equiripple error shape, we iteratively modify the error function by subtracting a known response from the largest peak(s).

Subtracting the FIR modifier sequence is really the heart of this algorithm. First, we present the general idea of this modification.

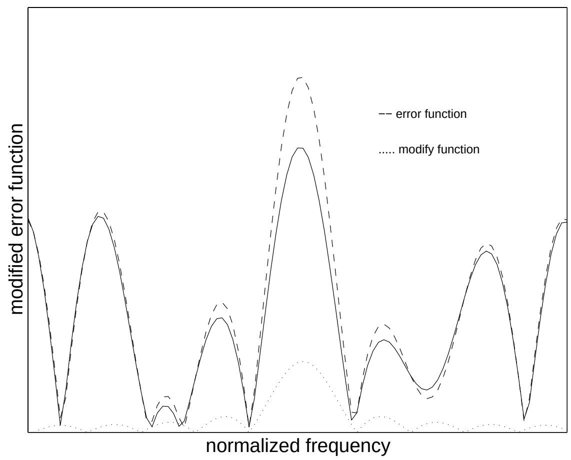

Figure 1: Error function before and after using the modification signal $m(n)$.

Consider a length-L sequence, $m(n)$, whose frequency response $M(\omega)$ has only one peak at ω_p. One of the sequences in this class is the modulated rectangular sequence,

$$m(n) = \begin{cases} ze^{-j\omega_p n} & \text{for } 0 \leq n \leq L-1 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where z is a complex constant and ω_p is an arbitrary frequency. The frequency response of this sequence is given by

$$M(\omega) = ze^{-j\frac{L}{2}(\omega-\omega_p)} \frac{\sin(\frac{L}{2}(\omega-\omega_p))}{\sin(\frac{1}{2}(\omega-\omega_p))}. \tag{2}$$

This function has its peak at ω_p with magnitude L and its mainlobe width is $\frac{4\pi}{L}$. It has regularly spaced zeros separated by $\frac{2\pi}{L}$, and a maximum sidelobe height that is approximately $L/5$.

We define the filter design error function by

$$E(\omega) = H(\omega) - I(\omega), \tag{3}$$

where $H(\omega)$ is the FIR frequency response and $I(\omega)$ is the ideal case. We define $H_m(\omega)$ and $E_m(\omega)$ to be the modified filter and the modified error function given by

$$\begin{aligned} H_m(\omega) &= H(\omega) - M(\omega) \\ \text{and} \quad E_m(\omega) &= E(\omega) - M(\omega) \end{aligned} \tag{4}$$

This $H_m(\omega)$ is also a frequency response of a FIR filter length L. The modification is shown in Fig. 1.

Under the condition that the ideal function is continuous and that the delay is between 0 to $L-1$, we claim that if there is only one maximum peak in the error function, we can find a value for z that reduces $||E_m(\omega)||_\infty$. To find this coefficient z, use the fact that only one maximum peak occurs at ω_p. Since $E(\omega) < E(\omega_p)$ when $\omega \neq \omega_p$, we can find ϵ such that $|E(\omega_p)| - \epsilon > |E(\omega)|$ for all points except possibly some small neighborhood of ω_p. This means that if we pick

$$z = -\frac{\epsilon}{L}\frac{E(\omega_p)}{|E(\omega_p)|} = -\frac{\epsilon}{L}e^{j\theta_p} \tag{5}$$

where θ_p is the angle of the complex error at $\omega = \omega_p$, then the error at ω_p will be reduced to $|E(\omega_p)| - \epsilon$. The other error maxima will be no larger than $\max\{|E(\omega_i)| + \epsilon\}$. When ϵ is small enough, the maximum error will still occur inside the neighborhood of ω_p, but the maximum will have been reduced. If we choose the largest possible ϵ, the modified error function will have two or more maximum peaks. An error function with several equal maxima can also be treated. Suppose that the error function has n equal maxima. First, define the Dirichlet kernel as:

$$D(\omega) = \frac{\sin(\frac{L}{2}\omega)}{\sin(\frac{1}{2}\omega)} e^{-j\frac{(L-1)}{2}\omega}$$

The maximum magnitude of this $D(\omega)$ is $D(0) = L$. To generate the modified error function needed in (4), we use

$$M(\omega) = \sum_{i=1}^{n} z_i\, D(\omega - \omega_{p_i}) \tag{6}$$

where ω_{p_i} are the equal-error peak locations.

We want to reduce each of the peaks by ϵ, so we need $M(\omega_{p_i}) = \epsilon\, e^{j\theta_{p_i}}$. We can find coefficient z_i by forming a matrix equation, $\mathbf{Dz} = \epsilon\,\mathbf{d}$, where

$$\mathbf{D} = \begin{bmatrix} L & d_{1,2} & d_{1,3} & \cdots & \\ d_{2,1} & L & d_{2,3} & & \\ \vdots & & \ddots & & \\ & & & & \\ & & & d_{n,n-1} & L \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \quad \text{and} \quad \mathbf{d} = \begin{bmatrix} e^{j\theta_{p_1}} \\ e^{j\theta_{p_2}} \\ \vdots \\ e^{j\theta_{p_n}} \end{bmatrix}$$

where $d_{i,j} = D(\omega_{p_i} - \omega_{p_j})$.

Note that the off-diagonal elements of $\mathbf{D}$ are much less than L, so $\mathbf{D}$ is unlikely to be singular. We can find z_i by computing $\mathbf{z} = \epsilon\,\mathbf{D}^{-1}\mathbf{d}$. We also have to choose ϵ to be positive and small enough to have the new maximum error equal to all reduced ones. This turns out to be possible until the error function is tight (min $\approx$ max), or we cannot add a nonzero modified function to reduce the peak error. Note that the equal peak height is not necessarily the stopping point, we also need a minimum number of ripples to satisfy the Alternation theorem criterion for the optimum filter.

So far, we have only proven the existence of a better error function but we do not yet have a practical algorithm, because we have to compute the inverse matrix $\mathbf{D}$ and find ϵ (by searching). We now propose another algorithm which has less comparison requirement and work on many peak simultaneously. This algorithm tries to make the large peak(s) be equal to an average peak value. Each iteration of this algorithm is described by these steps

1. calculate the error function, $E(\omega)$
2. extract all the peaks of $|E(\omega)|$
3. compute the average peak weighted error value
4. set a threshold and a desired error, (8) and (9)
5. compute the modifying function, $M(\omega)$
6. use $E(\omega) - M(\omega)$ to generate the new error function for the next iteration

The threshold and desired error can be computed from the average error, minimum error and maximum error. To make the algorithm as fast as possible, all peaks that are greater than the threshold are included in the set to be modified. Since the modifying function (6) changes the error function over all ω, including too many peak points can cause the algorithm to perform poorly—it might fail to converge and it will require much more computation. The desired error should be chosen smaller than the maximum error, but we cannot set it too small because it will then cause the algorithm to fail to converge. The average peak error is a compromise to use as the desired error, but in some of our experiments we have picked the desired error different from the average error to help the algorithm converge better.

The algorithm is formulate for the general complex case, but we can also use if to design complex-conjugate ideal frequency responses (where the filter has real coefficients). By using real coefficient first stage, we use the same threshold and desired error formula but find only peaks where $\omega \geq 0$, and set the modifying sequence to be real:

$$m(n) = \begin{cases} ze^{-j\omega_p n} + z^* e^{j\omega_p n} & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

3. IMPLEMENTATION

This paper used the modulated rectangular sequence as the modifying function because it has a small mainlobe which is close to the average distance between peaks of a typical error function. This inter-peak distance is much smaller near the transition band, but we can shift the peak of the modifying function into the transition band when the maximum error occurs at the band edge. Other types of modifying function could be also used. The threshold error and desired error are computed by

$$E_{\text{th}} = tE_{\text{avr}} + (1-t)E_{\max} \tag{8}$$
$$E_{\text{ds}} = dE_{\text{avr}} + (1-d)E_{\max} \tag{9}$$

The parameters t and d are used to compute the threshold and desired error as a linear combination of E_{avr} and $E_{\max}$. By using t and d between -1 and $+1$, we will have an algorithm where the maximum error converges to the average error. We must choose t small enough to make many peaks small simultaneously, but choosing this parameter too small can cause unnecessary computation. The desired error should be close to the average error, so d should chosen to be a small constant. One suggestion for setting these parameters is as follows: pick t between 0.4 and 0.6 and d between -0.2 and 0.2. A negative value for d can help the algorithm converge faster, but we have to limit E_{ds} to be greater than 0 to avoid divergence. When the bands are weighted, setting E_{ds} too small can make it hard to converge in the bands with small error.

4. COMPUTATION AND CONVERGENCE

This algorithm formulation is simple, but it has a computational advantage because it has only one FFT per iteration and many fewer comparisons than [3]. Normally, this algorithm converges faster than `cremez`, but the algorithm might fail to converge when peaks of the error are very small which is the case when we have a wide transition band. The main factor inhibiting convergence in this case is that the modifying function (6) is very small, so it becomes hard to change the frequency response in the transition band.

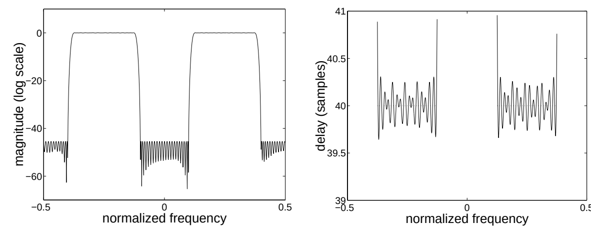

Figure 2: Frequency response of example 1. The optimized Chebyshev error is $\delta_{\text{opt}} = 0.0053$.

The error modification step has two other problems with respect to convergence. First, the algorithm might converge to an unoptimized filter when the average error of the first stage is too large. We would have to change the initializing filter to solve this problem. The second problem is that the algorithm might diverge or stop short of convergence. This problem occurs rarely, but happens when the modified error function increases at the peak-error maximum point. This can occur when the summation in (6) consists of several shifted Dirichlet kernels whose tails cause the peak error to increase. We can solve this problem by changing the threshold level or setting the desired error to be closer to the maximum error thereby reducing the effect of other peak-error points.

5. EXAMPLE

This part will show three filter designs: bandpass filter with delay less than half of it length and equal error weighting in the stop band and pass band, a lowpass filter with wide transition band and low delay, and a filter with 4 complex curves on 4 specific bands. All these cases cannot be designed by only first stage `cremez`. The first example shows a primary application of this filter design algorithm. The second example shows how the first stage affects the output. The third example shows that the algorithm can converge in any specifications.

Example 1. A length-100 bandpass filter with stop bands at normalized frequency 0 to 0.1 and 0.4 to 0.5, pass band at normalized frequency 0.125 to 0.375, and overall delay equal to 40 samples. Using `cremez` as the first stage, frequency response and delay are shown in Fig. 2. Because the modifying function is very small when we approach convergence, we stop the algorithm when $\delta_{\max} = 0.00536$ and $\delta_{\min} = 0.00534$. So, the error response is not exactly equiripple. Compared to a linear-phase filter with 0.0042 maximum error, this filter has nearly the same maximum error but a lower delay. The output of the first stage `cremez` is not purely real, but the imaginary part of its coefficients is very small. Consequently, we can use the real part to implement the filter and get almost the same frequency response.

The computation of this algorithm is 100 times faster than that of `cremez`. To illustrate further the advantage of the new algorithm, a length-1000 bandpass filter with its stop bands at normalized frequency 0 to 0.1 and 0.4 to 0.5, pass band at normalized frequency 0.1025 to 0.3975, overall delay equal to 300 was designed. Table I summarizes the design times which differ by 10^3.

We can also specify weighting in the design. For this specification, changing the weights to be 10 in stop band and 1 in pass band,

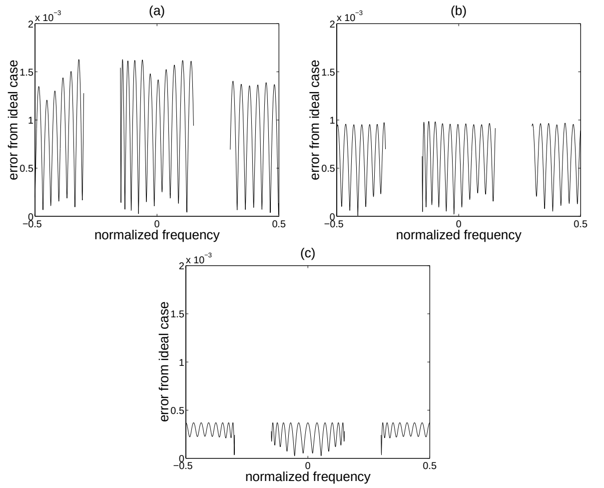

Figure 3: Example 2. (a) error of complex Remez, (b) error of new algorithm using `cremez` initialization, (c) error of new algorithm using `cifft` initialization.

the algorithm gives a filter that has a stop band deviation of 0.0015.

Example 2. A length-30 lowpass filter with pass band at normalized frequency 0 to 0.15, stop band at normalized frequency 0.3 to 0.5, and overall delay equal to 12 samples. We use two starting filters: first stage `cremez` and `cifft`. The three error responses are shown in Fig. 3. Note that this specification can be designed to have its maximum error less than 1.5×10^{-4} (the first stage of `cremez` has $\delta = 1.2 \times 10^{-4}$). Figure 3 shows that `cremez` has, in fact, failed to converge for these specs. The second stage of `cremez` uses an ascent/decent algorithm which was proven to converge to optimum case for a continuous function. The numerical implementation, unfortunately, does not obtain the converged solution. We can see that the convergence of the new algorithm is strongly dependent on the first stage. These types of filters have very small error which is always a problem when designing filters with large transition bands. The starting filters from the `cremez` first stage are not close enough to the optimum response.

Example 3. A length-80 filter specified by

$$A(f) = \begin{cases} 2|f| & \text{delay} = 15 \quad -0.48 < f < -0.27 \\ 2+i & \text{delay} = 50 \quad -0.23 < f < -0.02 \\ 4f^2 & \text{delay} = 70 \quad 0.02 < f < 0.23 \\ -ln(2f) & \text{delay} = 25 \quad 0.27 < f < 0.48, \end{cases}$$

Other intervals are don't care (transition bands), and $f = \omega/2\pi$ is normalized frequency.

The optimized response of the filter is shown in Fig. 4. This illustrates an efficient way of designing a filter for some applications. We can specify a filter with different delay in each frequency band. We also can design a filter with any desired complex gains for each frequency.

6. SUMMARY

This paper presents a new algorithm for designing FIR filters with complex frequency response and possibly a complex impulse response. Since the real and even symmetry property of linear-phase filters is not needed, the iteration is actually much simpler in this case.

Initialization of the algorithm uses any available efficient algorithm to return a filter whose peaks are close to equiripple. Then the large error peaks are modified by using a weighted sum of shifted Dirichlet kernels. This reduces magnitude of maximum error and makes it converge to the average peak error. A threshold and desired error are introduced as parameters of the algorithmic implementation.

The algorithm will dramatically improve the output from the first stage of `cremez`, but the solution is not guaranteed to be the true Chebyshev optimum. The big difference between this algorithm and `cremez` is the amount of computation. We can also design very long filters by this algorithm. Table 1 shows the comparison between the algorithm and `cremez`.

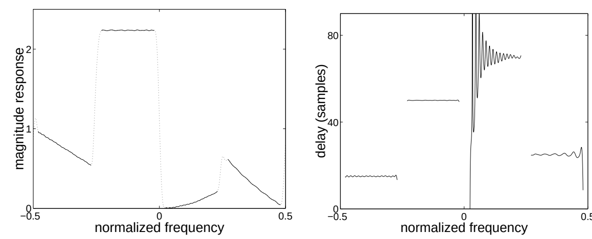

Figure 4: Frequency response of example 3. The maximum error is 0.00537 while `cremez` gets 0.00538.

FILTER	`cremez` (flops)	**new algorithm** (flops)	**# iterations**
Example 1	1.8×10^{10}	8.2×10^7	90
Example 2	1.9×10^7	1.9×10^7	223
Example 3	8.4×10^8	2.9×10^7	58
Long filter 1	2.24×10^{13}	5.2×10^{10}	220
Long filter 2	2.24×10^{13}	2×10^9	70

Table 1: Computation comparison between the new algorithm and `cremez`. Long filter 1 is a length-1000 band pass filter with `cremez` initialization. Long filter 2 is a length-1000 band pass filter with `cifft` initialization.

7. REFERENCES

[1] L. J. Karam, J. H. McClellan, "Design of optimal digital FIR filters with arbitrary magnitude and phase responses," *Proc. Intl. Symp. Circuits Syst.,* vol. 2, pp. 385–388, 1996

[2] L. J. Karam, J. H. McClellan, "Complex chebyshev approximation for FIR filter design," *IEEE Trans Circuits Syst II Analog Digital Sig. Proc.,* vol. 42, pp. 207–216, Mar 1995

[3] A. E. Cetin, O. N. Gerek, Y. Yardimci, "Equiripple FIR filter design by the FFT algorithm," *IEEE Signal Processing Magazine,* vol. 14, pp. 60–64, Mar 1997