Design of Linear-Phase Recursive Digital Filters by Optimization

S. Saab, A. Antoniou, and W.-S. Lu

Department of Electrical and Computer Engineering
University of Victoria
PO Box 3055 Stn CSC, Victoria BC V8W 3P6 Canada
ssaab@ece.uvic.ca, wslu@ece.uvic.ca, andreas@ece.uvic.ca

Abstract— An optimization method for the design of constant delay recursive digital filters is presented. The stability problem associated with current recursive-filter optimization algorithms is handled within the objective function which specifies the amplitude and phase characteristics simultaneously. Minimizing this objective function using specific starting points yields filter designs that satisfy arbitrary amplitude and phase specifications to within a preset degree of accuracy.

I. Introduction

A popular approach for the design of recursive digital filters is to use optimization techniques [1]–[2]. Current methods focus on minimizing the difference between a fixed, ideal or close-to-ideal response and the response of a practical filter. However, sometimes the ideal response is not known and must therefore be approximated. Such an instance arises when designing recursive filters with linear phase response. These filters would have a constant group delay but the lowest delay that would yield the lowest-order filter which would satisfy prescribed amplitude-response specifications is unknown at the outset.

This paper presents an optimization method that allows the ideal response to change during the design of a recursive filter. In addition, the issue of instability is addressed and alleviated. However, the resulting objective function is extremely nonlinear, and is therefore very sensitive to the choice of starting points. A method for finding suitable initialization points is presented along with design examples of a linear-phase lowpass filter, a differentiator, and a Hilbert transformer.

II. Objective Function

Practical recursive digital filters can be represented by the transfer function

$$H(\mathbf{x}_1, z) = \frac{N(z)}{D(z)} = H_0 \frac{z + a_{00}}{z + b_{00}} \prod_{l=1}^{L} \frac{z^2 + a_{1l} z + a_{0l}}{z^2 + b_{1l} z + b_{0l}} \quad (1)$$

where

$$\mathbf{x}_1 = \begin{bmatrix} H_0 & a_{00} & b_{00} & a_{11} & a_{01} & b_{11} & b_{01} & \ldots & b_{1L} & b_{0L} \end{bmatrix}^{\mathrm{T}}$$

and the substitution $z = e^{j\omega T}$, where ω is the frequency in rad/s and T is the sampling period in s, gives the frequency response. The frequency response of an ideal linear-phase recursive digital filter can be represented by

$$D(\tau, \omega) = M_0(\omega) e^{j\omega\tau}$$

where $M_0(\omega)$ is the amplitude response, and τ is an unknown delay. The *error function* to be minimized can be expressed as

$$e(\mathbf{x}, \omega) = |H(\mathbf{x}_1, e^{j\omega T}) - D(\tau, \omega)|$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} & \tau \end{bmatrix}^{\mathrm{T}}$$

The optimization algorithm used is the *quasi-Newton* algorithm described in ch. 14 of [1] because of its flexibility. The objective function

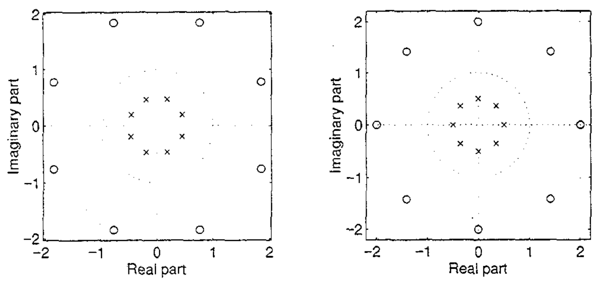
$$\Psi(\mathbf{x}) = \hat{E}(\mathbf{x}) \left\{ \sum_{i=1}^{K} \left[\frac{e(\mathbf{x}, \omega_i)}{\hat{E}(\mathbf{x})} \right]^p \right\}^{1/p}$$

where

$$\hat{E}(\mathbf{x}) = \max_{1 \le i \le K} e(\mathbf{x}, \omega_i)$$

is minimized using the *least-pth* algorithm (see p. 510 of [1]) to achieve a *minimax* solution.

If the prescribed specifications concern only the amplitude response, the filter can be designed using an unconstrained optimization and if an unstable filter is obtained, it can be readily stabilized by replacing any poles outside the unit circle of the z plane by their reciprocals (see p. 226 of [1]). If the prescribed specifications also concern the phase response, stabilization



(a) (b)

Fig. 1. Zero-pole plots of general $8th$-order initial points.

is not possible without changing the phase response. However, the problem can be alleviated by generating the denominator of the transfer function in (1) by using the modified bilinear transformation (MBT) described in [3]. We can write

$$D(z) = D(s)\Big|_{s=\frac{z-1+\delta}{z+1-\delta}}$$

where

$$D(s) = (s + d_{00}^2)\prod_{l=1}^{L}(s^2 + d_{1l}^2 s + d_{0l}^2)$$

and δ is a small positive constant. Since the zeros of $D(s)$ are always in the left-half s plane, the zeros of $D(z)$, and hence the poles of $H(z)$, must be located inside or on the circle of radius $1-\delta$ in the z plane. A nonzero value of δ is used to provide a certain stability margin and to avoid high sensitivity to coefficient quantization and possible artifacts in the amplitude response.

III. Initialization

The MBT causes the objective function to become highly nonlinear and, as a consequence, the choice of an initialization point becomes very critical in achieving a good local minimum point for the optimization problem. However, good initialization points do exist and can be found by initially using a general initialization point that would reveal the general characteristics of the desired filter.

The allpass filters represented by the zero-pole plots in Fig. 1 have a constant delay equal to their order times T. The magnitudes of the poles and zeros are half and two, respectively. The angle difference between adjacent poles and zeros in an Nth-order filter with N even is $2\pi/N$, and odd-order filters require an additional pole at the origin and a single zero at either plus or minus two.

Although one of the benefits of using a low-order filter is its low delay, one may expect that the constraints on the phase will increase the delay. This, indeed, is the case, and this is one of the reasons why the initialization points in Fig. 1 tend to give good results.

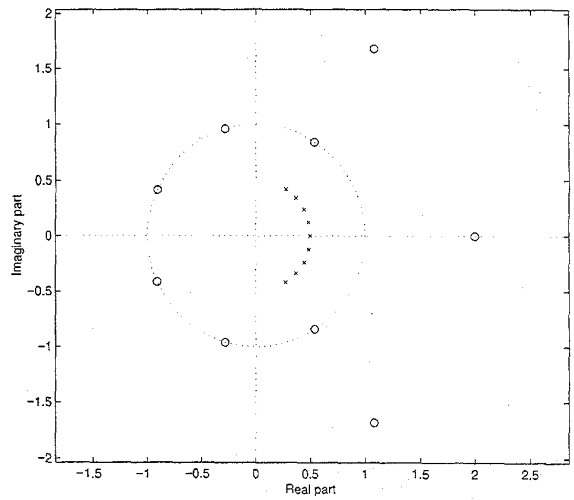


Fig. 2. Initial point for a 9th-order lowpass filter.

Inspecting the form of equation (1) reveals its limitations to zero-pole movement. If a pair of zeros are complex conjugates of each other, and the optimal point requires the pair of zeros to be real with one positive and the other negative, both outside of the unit circle, this point will not be reached. This is because as a zero enters the unit circle, the error function will increase, and a quasi-Newton algorithm will only decrease an error function. A similar analogy can be used for pole movement within the unit circle. These problems can be overcome by using both of the general initialization points in Fig. 1 (a) and (b).

After using these general initialization points for specific types of filters such a lowpass filters, certain zero-pole patterns emerge. These patterns can then be copied and used as initialization points. These new tailored initialization points can yield lower errors, and potentially lower-order filters which satisfy the same specifications. When designing these tailored initialization points, it helps to catagorize the zeros as *phase zeros* and *amplitude zeros*. Phase zeros are located outside the unit circle, and are mainly used to keep the phase linear. Amplitude zeros are located on or near the unit circle, and are mainly used to provide attenuation in the stopbands.

Algorithms 1 to 4 below provide general methods for generating initialization points for lowpass, highpass, bandpass, and bandstop filters. Fig. 2 gives an example of an initialization point for a 9th-order lowpass filter with passband edge at 1 rad/s and $T = 1$ s.

Algorithm 1: Lowpass and highpass filters

1. Set: n = filter order
2. Set: ω_p to passband edge
3. If lowpass: Set: $\Omega_p = \omega_p$
4. If highpass: Set: $\Omega_p = \pi/T - \omega_p$
5. Set: $\Omega_a = \pi/T - \Omega_p$
6. Choose an even or odd number of phase zeros.
   If even:
      Set: $Z_p = 2\lfloor \Omega_p nT/(2\pi) + 1/2 \rfloor$
   If odd:
      Set: $Z_p = 2\lfloor \Omega_p nT/(2\pi) \rfloor + 1$
7. Set: $Z_a = n - Z_p$
8. Call Algorithm 2 with the following parameters:
   If lowpass:
      (a) n = order, $\Omega = -\Omega_p$, $\omega_1 = \omega_p$, $m = 1/2$
      (b) $n = Z_p$, $\Omega = -\Omega_p$, $\omega_1 = \omega_p$, $m = 2$
      (c) $n = Z_a$, $\Omega = \Omega_a$, $\omega_1 = \omega_p$, $m = 1$
   If highpass:
      (a) n = order, $\Omega = \Omega_p$, $\omega_1 = \omega_p$, $m = 1/2$
      (b) $n = Z_p$, $\Omega = \Omega_p$, $\omega_1 = \omega_p$, $m = 2$
      (c) $n = Z_a$, $\Omega = -\Omega_a$, $\omega_1 = \omega_p$, $m = 1$
9. Assign H_0 to give an average gain of unity.

Algorithm 2: Pole and zero placement

1. Set: $\omega = \omega_1 + \Omega$
   If $\omega = 0$ or $\omega = \pi$:
      Set: $\Delta\omega = 2\Omega/(n-1)$
   Else:
      Set: $\Delta\omega = \Omega/(n/2 - 1)$
2. For $i = 1$ to $i = \lfloor (n+1)/2 \rfloor$:
   If $\omega_i = 0$ or $\omega_i = \pi$:
      If $m = 1/2$:
         Place a pole at $0.5e^{j\omega_i}$
      Else:
         Place a zero at $me^{j\omega_i}$
   Else:
      If $m = 1/2$:
         Place two poles at $0.5e^{\pm j\omega_i}$
      Else:
         Place two zeros at $me^{\pm j\omega_i}$
   Assign $\omega_{i+1} = \omega_i + \Delta\omega$

Algorithm 3: Bandpass filters

1. Set: n = filter order
2. Set: ω_{p_1} to 1st passband edge
3. Set: ω_{p_2} to 2nd passband edge
4. Set: $\Omega_p = \omega_{p_2} - \omega_{p_1}$
5. Set: $\Omega_a = \pi/T - \Omega_p$
6. Set: $\Omega_{a_1} = \omega_{p_1}$
7. Set: $\Omega_{a_2} = \Omega_a - \Omega_{a_1}$
8. Set: $Z_p = 2\lfloor \Omega_p nT/(2\pi) + 1/2 \rfloor$
9. Set: $Z_{a_1} = \lfloor \Omega_{a_1}(n - Z_p)/\Omega_a + 1/2 \rfloor$
10. Choose an even or odd number for Z_{a_1}
    If assignment of Z_{a_1} in step 9 does not satisfy choice:
       If $\Omega_{a_2} Z_{a_1} < \Omega_{a_1}(N - Z_p - Z_{a_1})$:
          Assign: $Z_{a_1} = Z_{a_1} + 1$
       Else:
          Assign: $Z_{a_1} = Z_{a_1} - 1$
11. Set: $Z_{a_2} = n - Z_p - Z_{a_1}$
12. Call Algorithm 2 with the following parameters:
    (a) n = order, $\Omega = \Omega_p$, $\omega_1 = \omega_{p_1}$, $m = 1/2$
    (b) $n = Z_p$, $\Omega = \Omega_p$, $\omega_1 = \omega_{p_1}$, $m = 2$
    (c) $n = Z_{a_1}$, $\Omega = -\Omega_{a_1}$, $\omega_1 = \omega_{p_1}$, $m = 1$
    (d) $n = Z_{a_2}$, $\Omega = \Omega_{a_2}$, $\omega_1 = \omega_{p_2}$, $m = 1$
13. Assign H_0 to give an average gain of unity.

Algorithm 4: Bandstop filters

1. Set: n = filter order
2. Set: ω_{p_1} to 1st passband edge
3. Set: ω_{p_2} to 2nd passband edge
4. Set: $\Omega_p = \pi/T + \omega_{p_1} - \omega_{p_2}$
5. Set: $\Omega_a = \pi/T - \Omega_p$
6. Set: $\Omega_{p_1} = \omega_{p_1}$
7. Set: $\Omega_{p_2} = \Omega_p - \Omega_{p_1}$
8. Set: $Z_a = 2\lfloor \Omega_a nT/(2\pi) + 1/2 \rfloor$
9. Set: $Z_{p_1} = \lfloor \Omega_{p_1}(n - Z_a)/\Omega_p + 1/2 \rfloor$
10. Choose an even or odd number for Z_{p_1}
    If assignment of Z_{p_1} in step 9 does not satisfy choice:
       If $\Omega_{p_2} Z_{p_1} < \Omega_{p_1}(n - Z_a - Z_{p_1})$:
          Assign: $Z_{p_1} = Z_{p_1} + 1$
       Else:
          Assign: $Z_{p_1} = Z_{p_1} - 1$
11. Set: $Z_{p_2} = n - Z_a - Z_{p_1}$
12. Set: $P_1 = \lfloor \Omega_{p_1} n/\Omega_p + 1/2 \rfloor$
13. Choose an even or odd number for P_1
    If assignment of P_1 in step 12 does not satisfy choice:
       If $\Omega_{p_2} P_1 < \Omega_{p_1}(n - P_1)$:
          Assign: $P_1 = P_1 + 1$
       Else:
          Assign: $P_1 = P_1 - 1$
14. Set: $P_2 = n - P_1$
15. Call Algorithm 2 with the following parameters:
    (a) $n = P_1$, $\Omega = -\Omega_{p_1}$, $\omega_1 = \omega_{p_1}$, $m = 1/2$
    (b) $n = P_2$, $\Omega = \Omega_{p_2}$, $\omega_1 = \omega_{p_2}$, $m = 1/2$
    (c) $n = Z_{p_1}$, $\Omega = -\Omega_{p_1}$, $\omega_1 = \omega_{p_1}$, $m = 2$
    (d) $n = Z_{p_2}$, $\Omega = \Omega_{p_2}$, $\omega_1 = \omega_{p_2}$, $m = 2$
    (e) $n = Z_a$, $\Omega = \Omega_a$, $\omega_1 = \omega_{p_1}$, $m = 1$
16. Assign H_0 to give an average gain of unity.

Another parameter which must be optimized is the minimum distance δ of the poles from the unit circle. Normally, the amplitude response is not specified in transition bands. However, for constant delay filters, the optimization can introduce amplitude-response artifacts in the transition region, such as that in Fig. 3, which are caused by high sensitivity due to one or more poles being located close to the unit circle.

It is reasonable to presume that when δ is increased, the amplitude of artifacts decreases, and the value of

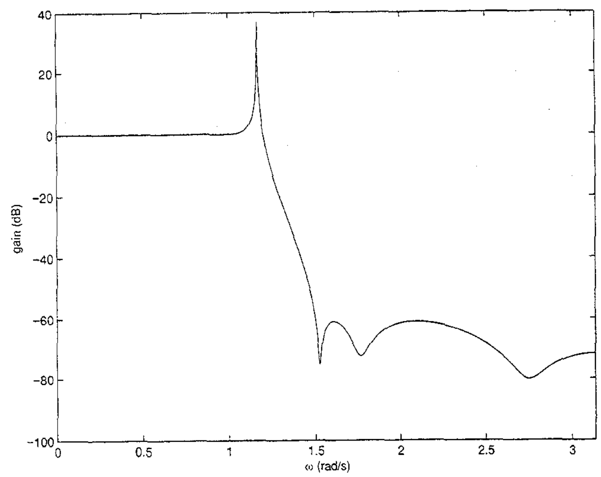


Fig. 3. Amplitude response of a potentially unstable filter.

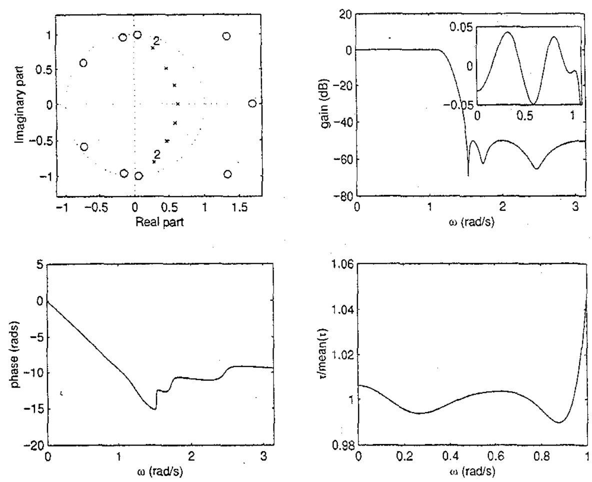


Fig. 4. Lowpass constant-delay recursive filter.

the objective function increases. This is, in fact, what happens almost all the time. However, it is quite possible that the minimum point does not place the poles at the boundary set by δ. In such a case, the solution can have less pronounced artifacts compared to a restriction from a high δ. However, the difference is small and δ can be adjusted to effectively find the optimal solution for a particular set of specifications.

The above approach has also been used to develop algorithms for differentiators and Hilbert transformers. Algorithms that can generate these initialization points are as follows:

Algorithm 5: Differentiators

1. Set: $n =$ filter order
2. If n is even:
   Set: $\alpha = 0$ and place one zero at 3
3. If n is odd:
   Set: $\alpha = 1$ and place one pole at -0.4
4. Place poles at 0.4 and -0.5, and a zero at 1.
5. Set: $\omega_1 = \Delta\omega = 2\pi/(n-\alpha)$
6. For $i = 1$ to $i = (n-2-\alpha)/2$:
   (a) Place two poles at $0.4e^{\pm j\omega_i}$
   (b) Assign $\omega_{i+1} = \omega_i + \Delta\omega$
7. Set: $\omega_1 = (2-\alpha)\pi/(n-1)$ and $\Delta\omega = 2\pi/(n-1)$
8. For $i = 1$ to $i = (n-2+\alpha)/2$:
   (a) Place two zeros at $3e^{\pm j\omega_i}$
   (b) Assign $\omega_{i+1} = \omega_i + \Delta\omega$
9. Assign H_0 to give an average gain of half the frequency range of interest within the frequency range of interest.

Algorithm 6: Hilbert transformers

1. Set: $n =$ filter order
2. If n is even:
   Set: $\alpha = 0$, place a pole at $-5/9$ and a zero at $-9/5$.
3. If n is odd:
   Set: $\alpha = 1$
4. Place a pole at $5/6$ and a zero at $6/5$.
5. Assign $\omega_1 = \Delta\omega = 2\pi/n$
6. For $i = 1$ to $i = (n-2+\alpha)/2$:
   (a) Place two poles at $5e^{\pm j\omega_i}/9$
   (b) Place two zeros at $9e^{\pm j\omega_i}/5$
   (c) Assign $\omega_{i+1} = \omega_i + \Delta\omega$
7. Assign H_0 to give an average gain of unity.

IV. Results

The optimization algorithm was implemented in C++ and was then linked to MATLAB where m-files were written to create initialization points, to track convergence, and manage the overall algorithm. Several lowpass, highpass, bandstop, differentiators, and Hilbert transformers were designed in order to test the program. Following are examples of a lowpass filter and a differentiator. In general, each recursive filter could meet the same amplitude specifications as a corresponding nonrecursive filter with a third of the order, and two thirds of the delay.

A lowpass filter with passband ripple $A_p = 0.1$ dB, minimum stopband attenuation $A_a = 50$ dB, $\omega_p = 1.0$ rad/s, $\omega_a = 1.5$ rad/s, and $\omega_s = 2\pi$ rad/s can be obtained with a 9th-order recursive filter with a group delay of approximately 9.45 s and the characteristics obtained are shown in Fig. 4. The same specifications would require a 32nd-order linear-phase nonrecursive filter which would have a delay of 16.0 s.

A bandstop filter with passband ripple $A_p = 0.1$ dB, minimum stopband attenuation $A_a = 50$ dB, $\omega_{p_1} = 0.5$ rad/s, $\omega_{a_1} = 1$ rad/s, $\omega_{p_2} = 2$ rad/s, $\omega_{a_2} =$

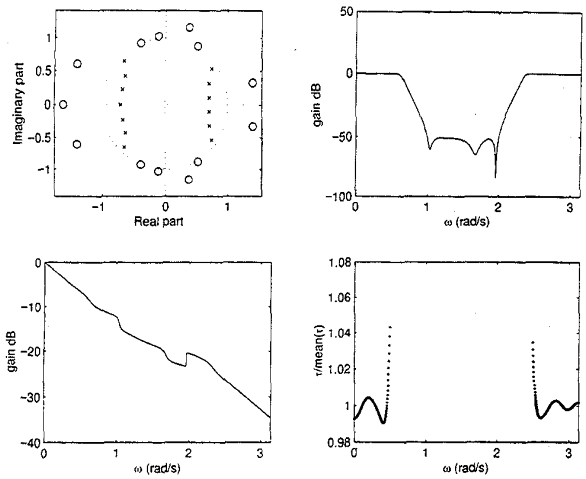


Fig. 5. Bandstop constant-delay recursive filter.

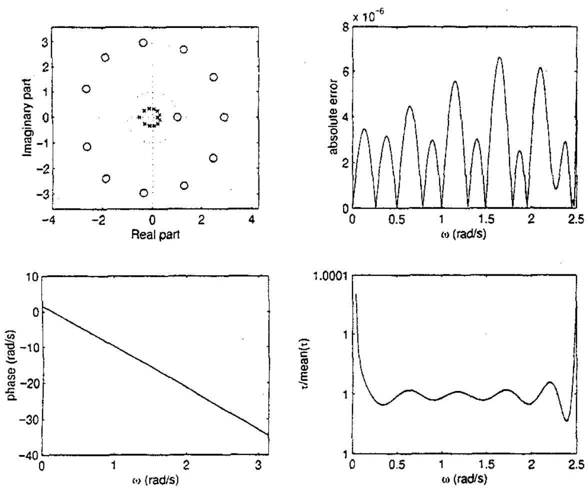


Fig. 6. Recursive differentiator filter.

2.5 rad/s, and $\omega_s = 2\pi$ rad/s can be obtained with a 13th-order recursive filter with a group delay of approximately 13.00 s and the characteristics obtained are shown in Fig. 5. The same specifications would require a 34th-order linear-phase nonrecursive filter which would have a delay of 17.0 s.

A differentiator with $\omega_p = 2.5$ rad/s, $\omega_s = 2\pi$ rad/s, and maximum weighted error of 10^{-3} with weighting function

$$W(\omega) = \mathrm{sinc}(\omega/\pi)$$

can be satisfied with a 12th-order recursive filter with a group delay of approximately 11.49 s and the characteristics obtained are shown in Fig. 6. The same specifications would require a 34th-order nonrecursive filter which would have a delay of 17 s.

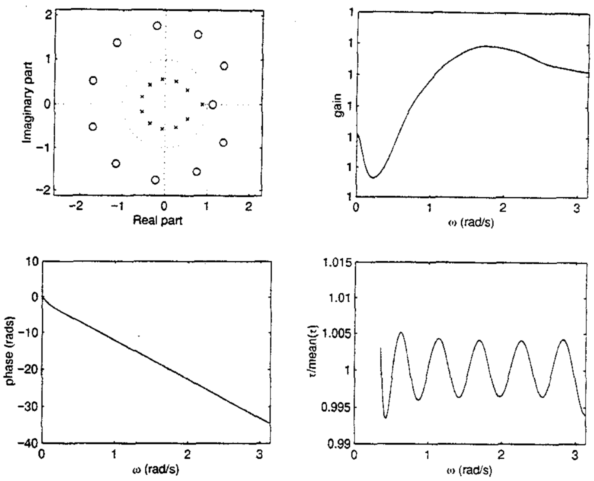


Fig. 7. Recursive Hilbert transformer.

A Hilbert transformer $\omega_p = 0.3$ rad/s and $\omega_s = 2\pi$ rad/s can be satisfied with an 11th-order recursive filter with a group delay of approximately 10.50 s and the characteristics obtained are shown in Fig. 7. The same specifications would require a 35th-order nonrecursive filter which whould have a delay of 17.5 s.

V. Conclusions

An optimization method for the design of constant-delay recursive filters was presented. The method determines the optimum average filter delay for the specifications supplied and it incorporates a mechanism that prevents unstable designs. The method can by used to achieve both amplitude and phase response specifications simultaneously and arbitrary amplitude and phase responses are possible.

The problem of initialization has been studied. By using certain general allpass filters as initialization points for any type of filter, customized initialization points can be formulated for filter types which have not been covered in this paper.

Filters designed with this method tend to have delays that are approximately two-thirds the delays of nonrecursive filters satisfying the same amplitude specifications with approximately one-third of the filter order.

References

[1] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, McGraw-Hill, New York, 2nd edition, 1993.

[2] W.-S. Lu, "Design of stable IIR digital filters with equiripple passbands and peak-constrained least squares stopbands," *IEEE Int. Symp. Circuits Syst.*, pp. 2192–2195, June 1997.

[3] W.-S. Lu, "A parameterization method for the design of IIR digital filters with prescribed stability margin," *IEEE Int. Symp. Circuits Syst.*, pp. 2188–2191, June 1997.