The preceding design methods have been based on designing an analog prototype filter and then converting it to a digital filter. This approach is appropriate for the class of approximations where analytical solutions are possible, but not for many others. In the remaining part of this chapter, methods will be developed that directly design the desired digital filter. Most approaches are extensions of methods used for FIR filters, but they are more complicated for the IIR case where rational approximation is being performed rather than polynomial approximation.

In this section a frequency-sampling design method is developed such that the frequency response of the IIR filter will pass through the given samples of a desired response. Since an IIR filter cannot have linear phase, the sampled response must contain both magnitude and phase. The extension of the frequency-sampling method to a LS-error approximation is not as simple as for the FIR filter. The method presented in this section uses a criterion based on the equation error rather than the more common error between the actual and desired frequency response[1]. Nevertheless, it is a useful noniterative design method. Finally, a general discussion of iterative design methods for LS-frequency response error is given.

1 Frequency-Sampling Design of IIR Filters

The method for calculating samples of the frequency response of an IIR filter presented in the section on Properties of IIR Filters can be reversed to design a filter much the same way it was for the FIR filter using frequency sampling. The z-transform transfer function for an IIR filter is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}.$$  

(1)

The frequency response of the filter is given by setting $z = e^{-j \omega}$. Using the notation

$$H(\omega) = H(z) |_{z=e^{-j\omega}}.$$  

(2)

Equally-spaced samples of the frequency response are chosen so that the number of samples is equal to the number of unknown coefficients in (1). These $L + 1 = M + N + 1$ samples of this frequency response are given by

$$H_k = H(\omega_k) = H\left(\frac{2\pi k}{L+1}\right)$$  

(3)
and can be calculated from the length-\((L + 1)\) DFTs of the numerator and denominator.

\[
H_k = \frac{\text{DFT}\{b_n\}}{\text{DFT}\{a_n\}} = \frac{B_k}{A_k}
\]

where the indicated division is term-by-term division for each value of \(k\). Multiplication of both sides of (4) by \(A_k\) gives

\[
B_k = H_k A_k
\]

If the length-\((L + 1)\) inverse DFT of \(H_k\) is denoted by the length-\((L + 1)\) sequence \(b_n\), equation (5) becomes cyclic convolution which can be expressed in matrix form by

\[
\begin{bmatrix}
  b_0 \\
  b_1 \\
  \vdots \\
  b_M \\
  0 \\
  \vdots \\
  0
\end{bmatrix}
= 
\begin{bmatrix}
  h_0 & h_L & h_{L-1} & \cdots & h_1 \\
  h_1 & h_0 & h_L & & \\
  \vdots & & & & \\
  h_2 & h_1 & h_0 & & \\
  0 & & & & \\
  \vdots & & & & \\
  0 & & & & \\
\end{bmatrix}
\begin{bmatrix}
  1 \\
  a_1 \\
  \vdots \\
  a_N \\
  0 \\
  \vdots \\
  0
\end{bmatrix}
\]

Note that the \(b_n\) in (6) are not the impulse response values of the filter as used in the FIR case. A more compact matrix notation of is

\[
\begin{bmatrix}
  b \\
  0
\end{bmatrix}
= 
\begin{bmatrix}
  H
\end{bmatrix}
\begin{bmatrix}
  a \\
  0
\end{bmatrix}
\]

where \(H\) is \((L + 1)\) by \((L + 1)\), \(b\) is length-\((M + 1)\), and \(a\) is length-\((N + 1)\). Because the lower \(L - N\) terms of the right-hand vector of (6) are zero, the \(H\) matrix can be reduced by deleting the right-most \(L - N\) columns to give \(H_0\) which causes (7) to become

\[
\begin{bmatrix}
  b \\
  0
\end{bmatrix}
= 
\begin{bmatrix}
  H_0
\end{bmatrix}
\begin{bmatrix}
  a \\
\end{bmatrix}
\]

Because the first element of \(a\) is unity, it is partitioned to remove the unity term and the remaining length-\(N\) vector is denoted \(a^*\). The simultaneous equations represented by (8) are uncoupled by further partitioning of the \(H\) matrix as shown in

\[
\begin{bmatrix}
  b \\
  0
\end{bmatrix}
= 
\begin{bmatrix}
  H_1 & 1 \\
  h_1 & H_2
\end{bmatrix}
\begin{bmatrix}
  a^*
\end{bmatrix}
\]

where \(H_1\) is \((M + 1)\) by \((N + 1)\), \(h_1\) is length-\((L - M)\), and \(H_2\) is \((L - M)\) by \(N\). The lower \((L - M)\) equations are written

\[
0 = h_1 + H_2 a^*
\]

or

\[
h_1 = -H_2 a^*
\]
which must be solved for \( a^* \). The upper \( M + 1 \) equations of (10) are written

\[
b = H_1 a
\]

which allows the calculation of \( b \).

If \( L = N + M \), \( H_2 \) is square. If \( H_2 \) is nonsingular, (11) can be solved exactly for the denominator coefficients in \( a^* \), which are augmented by the unity term to give \( a \). From (12), the numerator coefficients in \( b \) are found. If \( H_2 \) is singular [15] and there are multiple solutions, a lower order problem can be posed. If there are no solutions, the approximation methods must be used.

Note that any order numerator and denominator can be prescribed. If the filter is in fact an FIR filter, \( a \) is unity and \( a^* \) does not exist. Under these conditions, (12) states that \( b_n = h_n \), which is one of the cases of FIR frequency sampling covered [19]. Also note that there is no control over the stability of the filter designed by this method.

1.1 Summary

In this section, an interpolation design method was developed and analyzed. Use of the DFT converted the frequency-domain specifications to the time domain. A matrix partitioning allowed uncoupling the solution for the numerator from the solution of the denominator coefficients. The use of the DFT prevents the possibility of unequally spaced frequency samples as was possible for FIR filter design. The solution of simultaneous equations would allow unequal spacing which is not as troublesome as with the FIR filter because IIR filters are usually of lower order.

The frequency-sampling design of IIR filters is somewhat more complicated than for FIR filters because of the requirement that \( H_2 \) be nonsingular. As for the FIR filter, the samples of the desired frequency response must satisfy the conditions to insure that \( h_n \) are real. The power of this method is its ability to interpolate arbitrary magnitude and phase specification. In contrast to most direct IIR design methods, this method does not require any iterative optimization with the accompanying convergence problems.

As with the FIR version, because this design approach is an interpolation method rather than an approximation method, the results may be poor between the interpolation points. This usually happens when the desired frequency-response samples are not consistent with what an IIR filter can achieve. One solution to this problem is the same as for the FIR case [19], the use of more frequency samples than the number of filter coefficients and the definition of an approximation error function that can be minimized. There is no simple restriction that will guarantee stable filters. If the frequency-response samples are consistent with an unstable filter, that is what will be designed.

2 Discrete Least-Squared Equation-Error IIR Filter Design

In order to obtain better practical filter designs, the interpolation scheme of the previous section is extended to give an approximation design method [19]. It should be noted at the outset that the method developed in this section minimizes an equation-error measure and not the usual frequency-response error measure.

The number of frequency samples specified, \( L+1 \), will be made larger than the number of filter coefficients, \( M+N+1 \). This means that \( H_2 \) is rectangular and, therefore, (11) cannot in general be satisfied. To formulate an approximation problem, a length-\((L+1)\) error vector \( \epsilon \) is introduced in (8) and (9) to give

\[
\begin{bmatrix}
    b \\
    0
\end{bmatrix} = \begin{bmatrix}
    H_0 \\
    a
\end{bmatrix} + [\epsilon]
\]

Equation (11) becomes

\[
h_1 - \epsilon = -H_2 a^*
\]
where now $H_2$ is rectangular with $(L - M) > N$. Using the same methods as used to derive (11), the error $\epsilon$ is minimized in a least-squared error sense by the solution of the normal equations

$$H_2^T h_1 = -H_2^T H_2 a^*$$  \hfill (15)

If the equations are not singular, the solution is

$$a^* = -[H_2^T H_2]^{-1} H_2^T h_1.$$  \hfill (16)

If the normal equations are singular, the pseudo-inverse [15] can be used to obtain a minimum norm or reduced order solution.

The numerator coefficients are found by the same techniques as before in (12)

$$b = H_1 a$$  \hfill (17)

which results in the upper $M + 1$ terms in $\epsilon$ being zero and the total squared equation error being minimum.

As is true for LS-error design of FIR filters, (15) is often numerically ill-conditioned and (16) should not be used to solve for $a^*$. Special algorithms such as those used by Matlab and LINPACK [18], [8] should be employed.

The error $\epsilon$ defined in (13) can better be understood by considering the frequency-domain formulation. Taking the DFT of (13) gives

$$B_k = H_k A_k + \epsilon$$  \hfill (18)

where $\epsilon$ is the error in trying to satisfy (8) when the equations are over-specified. This can be reformulated in terms of $\mathcal{E}$, the difference between the frequency response samples of the designed filter and the desired response samples, by dividing (8) by $A_k$ to give

$$\mathcal{E}_k = \frac{B_k}{A_k} - H_k = \frac{\epsilon}{A_k}$$  \hfill (19)

$\mathcal{E}$ is the error in the solution of the approximation problem, and $\epsilon$ is the error in the equations defining the problem. The usual statement of a frequency-domain approximation problem is in terms of minimizing some measure of $\mathcal{E}$, but that results in solving nonlinear equations. The design procedure developed in this section minimizes the squared error $\epsilon$, thus only requiring the solution of linear equations. There is an important relation between these problems. (19) shows that minimizing $\epsilon$ is the same as minimizing $\mathcal{E}$ weighted by $A$. However, $A$ is unknown until after the problem is solved.

Although this is posed as a frequency-domain design method, the method of solution for both the interpolation problem and the LS equation-error problem is the same as the time-domain Prony’s method, discussed in Complex and Minimum Phase Approximation \(^1\) of reference [19].

Numerous modifications and extensions can be made to this method. If the desired frequency response is close to what can be achieved by an IIR filter, this method will give a design approximately the same as that of a true least-squared solution-error method. It can be shown that $\epsilon = 0 \leftrightarrow \mathcal{E} = 0$. In some cases, improved results can be obtained by estimating $A_k$ and using that as a weight on $\epsilon$ to approximate minimizing $\mathcal{E}$. There are iterative methods based on solving (16) and (17) to obtain values for $A_k$. These values are used as weights on $\epsilon$ to solve for a new set of $A_k$ used as a new set of weights to solve again for $A_k$[19][25]. We found this approach to converge slowly, but a recent paper using the log-magnitude [13] was more successful. Other approaches are given in [22], [23], [11]. The solution of (16) and (17) is sometimes used to obtain starting values for other iterative optimization algorithms that need good starting values for convergence.

An interesting iterative design algorithm that can design to approximate complex or magnitude frequency responses has been recently proposed by Jackson [11]. A different approach to the same problem was posed by Soewito [25], [29].

\(^1\)‘Least Squared Error Design of FIR Filters”: Section Complex and Minimum Phase Approximation

<http://cnx.org/content/m16892/latest/#uid90>

http://cnx.org/content/m16902/1.2/
To illustrate this design method a sixth-order lowpass filter was designed with 41 frequency samples to approximate. The magnitude of those less than 0.2 Hz is one and of those greater than 0.2 is zero. The phase was experimentally adjusted to result in a good magnitude response. The design was performed with Program 9 in the appendix of [19] and the frequency response is shown in Figure 7-33 of [19]. Matlab programs have recently been written which are smaller and easier to understand than those in FORTRAN.

2.1 Summary

In this section an LS-error approximation method was posed to design IIR filters. By using an equation-error rather than a solution-error criterion, a problem resulted that required only the solution of simultaneous linear equations.

Like the FIR filter version, the IIR frequency sampling design method and the LS equation-error extension can be used for complex approximation and, therefore, can design with both magnitude and phase specifications.

If the desired frequency-response samples are close to what an IIR filter of the specified order can achieve, this method will produce a filter very close to what a true least-squared error method would. However, when the specifications are not consistent with what can be achieved and the approximating error is large, the results can be very poor and in some cases, unstable. It is particularly difficult to set realistic phase response specifications. With this method, it is even more important to have a design environment that will allow easy trial-and-error procedure.

Newly published works which will be discussed here are [13], [12], [27], [10], [20], [17], [14]. Other references can be found in [19], [13], [6], [25], [29], [5]. The Matlab command \texttt{invfreqz()} which is an inverse to the \texttt{freqz()} command gives a similar or, perhaps, the same result as the method described in this note but uses a different formulation [16], [24].

2.2 more

Practical problems occur in the design of a filter to separate signals according to their energy. Because the energy content of a signal is the integral or sum of the square of the signal, a mean-squared-error measure is natural. Unfortunately, for the IIR filter design problem, the optimization procedure is nonlinear. This was pointed out in the last section where the equation error was used in order to have a linear problem.

Because of the nonlinear nature of the least-squared-error minimization, the method of solution becomes dependent on the desired frequency response, and therefore, there is no single method for design. The mean-squared error for magnitude approximation is defined as

$$q(x) = \sum_{i=0}^{L} |H(\omega_i)| - |H_d(\omega_i)|^2$$

(20)

where $x$ is a vector of filter parameters chosen to minimize $q$, and the error is sampled at $L+1$ frequencies $\omega_i$. Steiglitz [19] chose the parameter vector $x$ to be the coefficients of a cascade structure in order to best fit an iterative optimization scheme. He applied a standard optimization algorithm, the Fletcher-Powell method, to the minimization of (19). Other methods which are more directly related to a squared-error measure can also be used.

Practical difficulties exist in solving this approximation problem. In some cases, local minima are found rather than the global minimum. In other cases, convergence of the minimization algorithm is slow or does not occur at all. Numerical problems can result from ill-conditioned equations, and there is no guarantee that the designed filter will be stable.

An important factor is the choice of a desired frequency-response function $H_d(\omega)$ that does not result in the optimum approximation having a large error. This often means not having an abrupt discontinuity between the passband and stopband.

Another factor is the starting of the iterative optimization algorithm with a set of coefficients in $x$ that is close to the optimum. This can be accomplished by using the frequency sampling method to give a
design that can be used to start a least-squares algorithm. Because the error defined in (19) is in terms of magnitudes, an unstable design can be converted to a stable one by moving the unstable pole at a radius of \( r \) in the \( z \)-plane to a radius of \( 1/r \). This does not change the magnitude frequency response and does stabilize the effect of that pole \([19]\).

A generalization of the idea of a squared-error measure is defined by raising the error to the \( p \) power where \( p \) is a positive integer. This error is defined by

\[
q(x) = \sum_{i=0}^{L} |H(\omega) - H_d(\omega)|^p
\]  

(21)

Deczky \([7]\) developed this approach and used the Fletcher-Powell method to minimize (21). He also applied this method to the approximation of a desired group-delay function. An important characteristic of this formulation is that the solution approaches the Chebyshev or mini-max solution as \( p \) becomes large. Initial work shows the method of iteratively reweighted least squared error (IRLS) as was applied to the FIR filter design in can also be used for \( L_p \) and constrained least squared error optimal design of IIR filters \([28]\).

3 The Chebyshev Error Criterion for IIR Filter Design

The error measure that often best meets filter design specifications is the maximum error in the frequency response that occurs over a band. The filter design problem becomes the problem of minimizing the maximum error (the min-max problem).

Among several approaches to this error minimization, one is by Deczky which minimizes the \( p \)-power error of (21) for large \( p \). Generally, \( p = 10 \) or greater approximates a Chebyshev result \([19]\). Another is by Dolan and Kaiser which uses a penalty-function approach.

Linear programming can be applied to this error measure by linearizing the equations in much the same way as in (15)\([21]\). In contrast to the FIR case, this can be a practical design method because the order of practical IIR filters is generally much lower than for FIR filters. A scheme called differential correction has also proven to be effective.

Although the rational approximation problem is nonlinear, an application of the Remes exchange algorithm can be implemented \([19]\). Since the zeros of the numerator of the transfer function mainly control the stopband characteristics of a filter, and the zeros of the denominator mainly control the passband, the effects of the two are somewhat uncoupled. An application of the Remes exchange algorithm, alternating between the numerator and denominator, gives an effective method for designing IIR filters with a Chebyshev error criterion. If the order of the numerator and denominator are the same and the desired filter is an ideal lowpass filter, the Remes exchange should give the same result as the elliptic function filter. However, this approach allows any order numerator or denominator to be set and any shape passband to be approximated. There are cases where a lower-order denominator than numerator results in a filter with fewer required multiplications than an elliptic-function filter \([19]\).

4 Prony’s Method for Time-Domain Design of IIR Filters

The problem of designing an IIR digital filter with a prescribed time-domain response is addressed in this section. Most formulations of time-domain design of IIR filters result in nonlinear equations for the same reasons as for frequency-domain design. Prony, in 1790, derived a special formulation for the analysis of elastic properties of gases, which resulted in linear equations. A more general form of Prony’s method can be applied to the IIR filter design by use of a matrix description \([4]\, [19]\).

The transfer function of an IIR filter is given by

\[
H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots
\]  

(22)
and the impulse response \( h(n) \) is related to \( H(z) \) by the \( z \) transform.

\[
H(z) = \sum_{n=0}^{\infty} h(n) \ z^{-n}
\]

(23) can be written

\[
B(z) = H(z) \ A(z)
\]

which is the \( z \)-transform version of convolution. This convolution can be written as a matrix multiplication. Using the first \( K+1 \) terms of the impulse response, this is written

\[
\begin{bmatrix}
  b_0 \\
  b_1 \\
  \vdots \\
  b_M \\
  0 \\
  \vdots \\
  0
\end{bmatrix} =
\begin{bmatrix}
  h_0 & 0 & 0 & \cdots & 0 \\
  h_1 & h_0 & 0 & \cdots \\
  h_2 & h_1 & h_0 & \cdots \\
  0 & \vdots & \vdots & \ddots & 0 \\
  \vdots & \ddots & \ddots & \ddots & \vdots \\
  0 & & & \cdots & h_L & \cdots & h_0
\end{bmatrix}
\begin{bmatrix}
  1 \\
  a_1 \\
  \vdots \\
  a_N
\end{bmatrix}
\]

(25)

In order to uncouple the calculations of the \( a_n \) and the \( b_n \), the matrices are partitioned to give

\[
\begin{bmatrix}
  b \\
  0
\end{bmatrix} =
\begin{bmatrix}
  h_1 & H_2 \\
  H_1 & H_2
\end{bmatrix}
\begin{bmatrix}
  1 \\
  a^*
\end{bmatrix}
\]

(26)

where \( b \) is the vector of the \( M + 1 \) numerator coefficients of (22), \( a^* \) is the vector of the \( N \) denominator coefficients (\( a_0 = 1 \)), \( h_1 \) is the vector of the last \((K - M)\) terms of the impulse response, \( H_1 \) is the \( M + 1 \) by \( N + 1 \) partition of (25), and \( H_2 \) is the \((K - M)\) by \( N \) remaining part. The lower \( K - M \) equations are written

\[
0 = h_1 + H_2 a^*
\]

(27)
or

\[
h_1 = -H_2 a^*
\]

(28)

which must be solved for \( a^* \). The upper \( M + 1 \) equations of (26) are written

\[
b = H_1 a
\]

(29)

which allows the calculation of \( b \).

If \( L = N + M \), then \( H_2 \) is square. If \( H_2 \) is nonsingular, (28) can be solved exactly for the denominator coefficients in \( a^* \), which are augmented by the unity term to give \( a \). From (29), the numerator coefficients in \( b \) are found. If \( H_2 \) is singular [15] and there are multiple solutions, a lower order problem can be posed. If there are no solutions, the methods of the next section must be used.

Note that any order numerator and denominator can be prescribed. If the filter is in fact an \( \text{FIR} \) filter, \( a \) is unity and \( a^* \) does not exist. Under these conditions, (29) states that \( b_n = h_n \), which is one of the cases of \( \text{FIR} \) frequency sampling covered in Section 3.1 of [19]. Also note that there is no control over the stability of the filter designed by this method.

Although Prony’s method, applied to the time-domain design problem here, is similar to the solution of the frequency-sampling \( \text{IIR} \) design problem, there are important differences. The inverse DFT is used to
obtain the matrix in the frequency domain problem, which is cyclic convolution. Equation (25) is noncyclic convolution and the \( K + 1 \) terms of \( h(n) \), used to form \( H \), result from a truncation of the infinitely long sequence.

### 4.1 An Approximate Solution or the Least Equation Error Problem

In order to obtain better practical filter designs, the interpolation scheme of the previous section is extended to give an approximation design method [19]. It should be noted at the outset that the method developed in this section minimizes an equation-error measure and not the usual frequency-response error measure.

The number of samples specified, \( L + 1 \), will be made larger than the number of filter coefficients, \( M + N + 1 \). This means that \( H_2 \) is rectangular and, therefore, cannot in general be satisfied. To formulate an approximation problem, a length-(\( L + 1 \)) error vector \( \epsilon \) is introduced in and to give

\[
\begin{bmatrix}
  b \\
  0
\end{bmatrix} = \begin{bmatrix} H_0 \end{bmatrix} \begin{bmatrix} a \end{bmatrix} + [\epsilon] \quad (30)
\]

(28) becomes

\[
h_1 - \epsilon = -H_2 a^* \quad (31)
\]

where now \( H_2 \) is rectangular with \((L - M) > N\). Using the same methods as used to derive (28), the error \( \epsilon \) is minimized in a least-squared error sense by the solution of the normal equations [15]

\[
H_2^T h_1 = -H_2^T H_2 a^* \quad (32)
\]

If the equations are not singular, the solution is

\[
a^* = -[H_2^T H_2]^{-1} H_2^T h_1. \quad (33)
\]

If the normal equations are singular, the pseudo-inverse [15] can be used to obtain a minimum norm or reduced order solution.

The numerator coefficients are found by the same techniques as before in (29)

\[
b = H_1 a \quad (34)
\]

which results in the upper \( M + 1 \) terms in \( \epsilon \) being zero and the total squared equation error being minimum.

As is true for LS-error design of FIR filters, (32) is often numerically ill-conditioned and (33) should not be used to solve for \( a^* \). Special algorithms such as those used by Matlab and LINPACK [18], [8] should be employed.

Various modifications can be made to the form of Prony’s method presented. After the denominator is found by minimizing the equation error, the numerator can be found by minimizing the solution error. It is possible to mix the exact and approximate methods. The details can be found in [19], [2], [3].

Several modifications to Prony’s method have been made to use it to minimize the solution error. Most of these iteratively minimize a weighted-equation error with Prony’s method and update the weights from the previous determination of \( a[9] \), [26].

If an LS-error, time-domain approximation is the desired result, a minimization technique can be applied directly to the solution error. The most successful method seems to be the Gauss-Newton algorithm with a step-size control. This combined with Prony’s method to find starting parameters is an effective design tool.

http://cnx.org/content/m16902/1.2/
References


http://cnx.org/content/m16902/1.2/


http://cnx.org/content/m16902/1.2/