METEOR:
A Constraint-based FIR Filter Design Program

K. Steiglitz†
Computer Science
Princeton University
Princeton, NJ 08544–2087

T. W. Parks*
Electrical Engineering
Cornell University
Ithaca, NY 14850

J. F. Kaiser
Bell Communications Research
445 South Street
Morristown, NJ 07962–1910

ABSTRACT

The usual way of designing a filter is to specify a filter length and a nominal response, and then to find a filter of that length which best approximates that response. In this paper we propose a different approach: specify the filter only in terms of upper and lower limits on the response, find the shortest filter length which allows these constraints to be met, and then find a filter of that order which is farthest from the upper and lower constraint boundaries in a mini-max sense.

Previous papers have described methods for using an exchange algorithm for finding a feasible linear-phase FIR filter of a given length if one exists, given upper and lower bounds on its magnitude response. The resulting filters touch the constraint boundaries at many points, however, and are not good final designs because they do not make best use of the degrees of freedom in the coefficients. We use the simplex algorithm for linear programming to find a best linear-phase FIR filter of minimum length, as well as to find the minimum feasible length itself. The simplex algorithm, while much slower than exchange algorithms, also allows us to incorporate more general kinds of constraints, such as concavity constraints (which can be used to achieve very flat magnitude characteristics).

We give examples that illustrate how the proposed and the usual approaches differ, and how the new approach can be used to design filters with flat passbands, filters which meet point constraints, minimum phase filters, and bandpass filters with controlled transition band behavior.

1. Introduction

The FIR linear-phase filter design problem begins with the formulation of specifications arising from the application at hand. Typical specifications include the desired stop-band attenuation, passband deviation, location of zeros of transmission, etc. Two methods for designing a filter to meet these specifications include the approximation approach and the limit approach. In the approximation approach, the length of the filter and a desired

---

†Work supported in part by NSF Grant MIP–8705454, U. S. Army Research Office – Durham Contract DAAG29-85-K-0191.

* Supported in part by ONR Contract N00014-91-1161

A preliminary version of this paper was presented in [22], and was published as "METEOR: A Constraint-Based FIR Filter Design Program," K. Steiglitz, T. W. Parks and J. F. Kaiser, *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1901-1909, August 1992.

This paper is dedicated to the Canyon Diablo Meteor Crater, AZ.

frequency response are specified. The filter coefficients are determined to minimize the maximum weighted error between the desired and actual responses over the frequency bands of interest. In the limit approach, a set of upper and lower limits are specified for the frequency response. The necessary number and values of filter coefficients for which the frequency response remains within the prescribed limits are then determined. The limit approach was used in the earliest work on analog filter design more than 50 years ago. Cauer [1] designed analog filters to meet prescribed, limit type tolerance schemes using elliptic functions. It is possible to use the approximation approach to meet limit constraints and to use the limit approach described in this paper to solve approximation problems.

In 1970, Herrmann published an article describing the equations which must be solved to obtain a filter with the maximum possible number of equal ripples [2] (later called extra-ripple [3] or maximal-ripple [4] filters). This maximal ripple design is neither an approximation approach nor a limit approach. Rather, it is a hybrid approach where the filter length and ripple size (equivalent to limits on the frequency response) are specified and the bandedges are determined by the algorithm. Schüssler, in 1970, presented the work he and Herrmann had been doing on the design of maximal-ripple filters at the Arden House Workshop [5]. Hofstetter developed an efficient algorithm for solving the equations proposed by Herrmann and Schüssler and presented papers with Oppenheim and Siegel at the 1971 Princeton conference [6] and the 1971 Allerton House conference [7] describing the algorithm and relating it to the Remes exchange algorithm.

Several papers on the Chebyshev approximation approach to filter design appeared at about the same time. Helms, in 1971 [8], described techniques, including linear programming, to solve the Chebyshev approximation problem for filter design. Parks and McClellan used the Remes exchange algorithm [9,10] to solve the Chebyshev approximation problem.

Hersey, Tufts, and Lewis described, at about the same time, an interactive method for designing filters with upper and lower constraints on the magnitude of the frequency response [11]. The limit approach was also used by McCallig and Leon in 1978 [12] and by Grenez in 1983 [13].

When a lowpass filter is designed using the Chebyshev approximation approach, the 5 interrelated parameters are the filter length N, the passband edge Fp, the stopband edge Fs, the passband error δ_p, and the stopband error δ_s. Relations among these parameters have been determined numerically for the Chebyshev approximation problem and design formulas have been published [14,15]. With the help of these design formulas it is possible to fix any 4 of these parameters and optimize the remaining parameter. Since these design formulas are not exact, several iterations of the design process are usually necessary. For example, when the bandedges and deviations are given, an estimate of the necessary filter length can be calculated using the design formulas. Usually the filter with this estimated length will not be exactly the minimum length required to meet the specifications and the filter will be designed again with a slightly different length until the minimum-length filter is obtained.

The use of transition bands will give good lowpass designs but may cause problems for multiband bandpass filters [16]. The frequency response is not controlled in the transition band and may make large, unexpected, excursions which make the design useless.

The design formulas can be used to modify the stopband specifications to eliminate the unwanted excursions in most cases, but the choice of stopband edges and appropriate error weighting functions is more of an art than a science. The limit approach offers a way to avoid unwanted excursions in multiband filter design. Upper and lower limits are imposed on the response for all frequencies. The limits imposed on the bands which otherwise would be unrestricted transition bands eliminate the possibility of large peaks in the magnitude of the frequency response, but do not impose any particular shape on the response in these bands.

In this paper we describe a very flexible design program which combines most of the useful characteristics of the approximation approach and the limit approach to FIR filter design. We use the simplex algorithm for linear programming to find the linear-phase filter of minimum length which meets prescribed limits on the frequency response and then maximize the distance from the constraints. For a fixed length filter, the bandedges can be adjusted to maximize or minimize the width of a frequency band while still meeting prescribed limits on the frequency response. The bands can consist of just one frequency so that the location of the zeros can be fixed in the stopband. Additional constraints, such as concavity of the response to give flat magnitude characteristics, can be imposed in appropriate frequency bands. First, we describe the algorithm and the Pascal program and then we give examples to show how this new approach can be used in a variety of situations.

2. The Algorithm

There are 4 different types of linear-phase filters. For both even and odd symmetry of the impulse response, we obtain linear phase with either even or odd number of coefficients. In Rabiner and Gold [17] it is shown that the frequency response for each of the 4 types of linear-phase filters has the form

$$H(\Omega) = e^{j\frac{\pi}{2}L} e^{-j\Omega(\frac{N-1}{2})} A(\Omega)$$

where the real-valued amplitude function $A(\Omega)$ is a weighted sum of trigonometric functions, where N is the length of the filter and $L = 0$ or $L = 1$, depending on the filter symmetry.

For convenience, in the following discussion we assume that the filter model is the following sum of cosines, corresponding to an odd-length, even symmetric impulse response, although any linear combination of known functions can be used.

$$A(\Omega_k) = \sum_{i=0}^{\frac{N-1}{2}} a_i \cos(i\Omega_k)$$

$A(\Omega_k)$ is the real-valued frequency response of the filter at frequency Ω_k, and the frequency points at which specifications are made, $\Omega_k, k = 1, 2, 3, \ldots$, need not be equally spaced.

An upper-limit constraint at Ω_k has the form

$$A(\Omega_k) \leq U(\Omega_k).$$

We introduce a parameter y which represents the distance between the frequency response and the upper bound, so that some of the constraints look like

$$A(\Omega_k) + y \leq U(\Omega_k).$$

Since we are maximizing y, we call those constraints which have y in them *optimized* constraints, and those that do not, *hugged* constraints. Similarly, lower bounds on the frequency response result in constraints of the form

$$-A(\Omega_k) \leq -L(\Omega_k)$$

or

$$-A(\Omega_k) + y \leq -L(\Omega_k),$$

depending on whether the constraint is hugged or optimized.

Putting constraints on the second derivative of the frequency response has been shown to be an effective way to obtain filters that are very flat [15]. The second derivative is a linear function of the coefficients, namely, for the case 1 filters considered here

$$A''(\Omega_k) = -\sum_{i=1}^{m-1} i^2 a_i \cos(i\Omega_k),$$

so that concavity constraints can be written as linear inequalities of the form

$$A''(\Omega_k) \leq 0$$

for a concave downward function, or

$$A''(\Omega_k) \geq 0,$$

for a concave upward function.

When all the constraints are written down, we obtain the linear programming problem

$$(PRIMAL) \quad \max y$$

subject to

$$C^T a + hy \leq b$$

where the matrix C is determined from the sampled trigonometric functions, the vector a is made up of the coefficients a_i, the vector b contains the bounds, and the vector h has a 1 wherever a constraint is optimized, and a 0 wherever it is hugged. The variables a and y are unconstrained in sign. We will call this the $PRIMAL$ problem. The dual of this linear program is in standard form, the most convenient for numerical solution:

$$(DUAL) \quad \min b^T x$$

subject to

$$Cx = 0, \quad h^T x = 1, \text{ and } x \geq 0.$$

We solve DUAL using the standard two-phase simplex algorithm [18]. Phase I searches for a feasible solution to DUAL, starting from an artificial basis, and phase II searches for an optimal solution.

It is a fundamental fact of linear programming theory that the cost function of the DUAL always satisfies $b^T x \geq y$, the cost function of the PRIMAL, with equality if and only if x and y are both optimal in their respective programs. Therefore, if the DUAL cost $b^T x$ ever falls below zero during pivoting, the optimal PRIMAL cost must be negative. This means that the original filter approximation problem is infeasible, and we stop the simplex algorithm whenever this condition is obtained. Application of the simplex algorithm to the DUAL problem therefore terminates in one of the following conditions:

a) Negative cost reached, implying that the original design problem is infeasible;
b) Optimality is reached in DUAL with non-negative cost, in which case the original design problem has a feasible solution;
c) DUAL is unbounded, which implies that PRIMAL (and the original design problem) is infeasible;
d) DUAL is infeasible, which implies that PRIMAL (and the original design problem) is either infeasible or unbounded.

A comment is in order as to why the variable y is introduced in those situations when we are interested only in whether there is a feasible solution to lower- and upper-bound constraints. Computational experience has shown that with a trivial cost function in the primal, the simplex method applied to the dual sometimes cycles in realistic filter-design problems, because of degeneracy. A non-trivial cost function seems to provide enough direction to the simplex algorithm to avoid such stagnation. Rather than take special precautions to avoid cycling, we chose always to maximize the distance y from the response to the constraint boundaries. (As we saw above, it is not always necessary to complete the optimization when the original problem is infeasible.) This has the additional advantage of being useful for the final design when the length is known, and also does not interfere with the resolution of ties based on size of the pivot elements, which is important for numerical stability (see [19]).

A special case arises unavoidably, however, when there are no constraints designated as *optimized*. In that case, $h = 0$ and DUAL is always infeasible. However, the constraint matrix of DUAL in this case is not of full rank, having a zero row, and phase I ends with an artificial basis element remaining in the basis. The redundant row is disregarded in phase II, and the optimization finds a solution to the original problem (if any exist) with zero cost, corresponding to a response that is allowed to touch any of the constraint boundaries. Thus, the algorithm functions in a useful way, even if a zero row is present in the DUAL constraint matrix.

The optimal value of the dual variable x has a well-known and interesting interpretation. Suppose the constraint values b are changed a small amount to $b + db$. This changes the cost function in the dual a small amount, but will not in general change the optimal solution x to the dual. The new value of the optimal cost function becomes $y = b^T x + db^T x$. Thus, $x$is the partial derivative of the optimal value of$y$ with respect to the constraint values b. Simplex finds an optimal value for x that has at most $m+1$ positive entries, and, by complementary slackness, each of these corresponds to an extremum of the distance between the frequency response and constraints (a "ripple") in the case of an upper or lower bound, or to a point where the second derivative is zero in the case of a concavity constraint.

The simplex algorithm is used in the following three modes, depending on what design task is desired:

a) Given $m_1 < m_2$, find the minimum-length m between them such that the original design problem is feasible (that is, such that DUAL has a non-negative optimal solution), and optimize y for that minimum length;

b) Solve the original optimization problem for fixed length m_0;

c) Given a particular right (left) bandedge and a set of constraints in which it occurs, find the largest (smallest) value for that bandedge for which the original design problem is feasible, and optimize y for that bandedge value. (The optimum value of y will in general be positive because the bandedge value is rounded to the nearest gridpoint.)

What is the best search strategy to use in finding the minimum length in a)? We might expect, because the cost of testing feasibility increases with m, that the strategy with least expected cost (assuming uniformly distributed answers) probes to the left of the midpoint between the current left and right boundaries. However, computation of the optimal strategies for probe-cost functions that grow as a low-order polynomial in m shows that binary search is surprisingly near optimal. More work on this problem is in progress [20], but binary search appears adequate for this application. Mode b) allows us to do things like find the best stopband rejection, while keeping passband ripple within limits. Mode c) allows us to do things like extend the end of a stopband as far as possible, while keeping the other constraints fixed. Binary search is also used in c).

3. The Program

The algorithm described above was implemented in Pascal, and the current version is available from the authors. The authors' intent is that the program be read and modified by users, rather than used as a static package, and Pascal seems well suited to this purpose: it is widely available, cleanly designed, allows careful structuring, and hopefully, good readability.

As might be expected, the critical parts of the program involve the treatment of tests which theoretically determine whether quantities are positive, negative, or zero. These tests determine when each of the various termination conditions is reached, and roundoff error requires us to decide on how small a positive number is considered zero, how small a negative number is considered negative, and so on. Experience has shown that a single parameter *eps* can be used for these tests at several different places in the program, and that *eps* can be fixed at 10^{-8} for the range of problems used as examples in this paper.

The only cases observed so far where serious accumulation of roundoff error occurs is when a wide band of frequencies is unconstrained, and the frequency response is allowed to grow very large in those bands — say as large as 10^6. The problem is manifested by the cost in phase I reaching relatively large negative numbers before detecting optimality, even though the cost in phase I is theoretically non-negative. Of course, these designs are impractical, and the accuracy problem irrelevant, but the program continues to function in these cases.

Trading off space for running time is a serious issue in the program design. At one extreme, we can pre-compute and store the tableau entries, which avoids re-computation, but uses a great deal of storage. At the other extreme, we can generate the tableau entries on the fly, using the least space, but the most time. As a compromise between the two, we can pre-compute and store tables of the trigonometric functions used for the tableau entries. We chose the first alternative because it appears that execution time is a more serious limitation than storage for the kinds of design problems likely to be solved. If storage is a serious problem, references to the tableau entries must be replaced by procedure calls that compute the required values.

4. Examples

We present a series of 8 examples illustrating various features of the algorithm. Two separate programs are used to design a filter. The program FORM is an interactive program which requests information from the user and creates an input file for METEOR which solves the linear programming problem. The desired frequency response is specified by two kinds of specifications: *limit* and *concavity.* We call these "limit specifications" and "concavity specifications".

limit specifications: Each limit specification consists of the following information:

information	form
upper or lower?	"+" or "−"
left bandedge, right bandedge	$[F_1,\ F_2]$, real
bound at left edge, bound at right edge	$[B_1,\ B_2]$, real
hugged or not hugged?	"h" or "n"
arithmetic or geometric interpolation?	"a" or "g"

An upper bound on the frequency response is indicated by a "+", and a lower by a "−". The left and right bandedges, F_1 and F_2, are expressed in units of cycles/sampling interval, so that the Nyquist frequency corresponds to 0.5. The frequency response is constrained by the value B_1 at the left bandedge, and by B_2 at the right bandedge; the values in between are interpolated by the program either arithmetically (linearly) ("a"), or geometrically ("g"), (linearly in dB.). Finally, if a limit specification is "hugged", it is not included in the optimization criterion of the final linear program, and is included if it is "not hugged". Thus, the final design is pushed away as much as possible from those limit specifications that are not hugged, but may be arbitrarily close to the hugged limit specifications.

concavity specifications: Each concavity specification is determined by the following information:

information	form
concave up or down	"+" or "−"

left bandedge, right bandedge $[F_1,\ F_2]$, real

The frequency response is constrained to be concave up or down in the indicated band.

mode: The design program has three modes, "minimum-length", "optimize", and "push". In the "minimum-length" mode, the minimum length that satisfies the given constraints is found. The user specifies either even or odd length and either even or odd symmetry of the impulse response.

In the "optimize" mode, the response is pushed away from the non-hugged constraints for the fixed length specified by the user. If the design is not realizable at all for this fixed length, the program reports infeasibility.

In the "push" mode, a set of bandedges are pushed as far as possible while still respecting the constraints for the fixed length specified by the user. The set is pushed either to the left or the right.

In the following examples we first display the specification file in the format produced by the program FORM, then graph the resulting frequency response.

Example 1: Lowpass, Minimum-Length Filter

Here we use 4 limit specifications, which are displayed as follows by FORM:

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.200	1.100	1.100	n	a
2	limit	–	0.000	0.200	0.900	0.900	n	a
3	limit	+	0.250	0.500	0.100	0.100	n	a
4	limit	–	0.250	0.500	-0.100	-0.100	n	a

FINDING MINIMUM LENGTH
ODD LENGTHS from 7 to 21
COSINE MODEL (even symmetric coefficients)
201 grid points

The lower and upper limits N_1 and N_2 (7 and 21 in this case) for the filter length are estimated using formulas developed in [14] and [15]. Since we are specifying limits for the real-valued amplitude function which may be negative we must specify a negative lower limit in the stop band. Figure 1 shows the resulting amplitude response; the minimum length satisfying the specifications is 17. Note in Figure 1 that since the constraints are not hugged the optimized response is strictly within the limits. The resulting equiripple response is equivalent to that obtained with the Parks-McClellan algorithm. In Figure 1 we have shown the amplitude response to clearly display the negative as well as the positive limits. For the remaining examples we will display the magnitude of the frequency response.

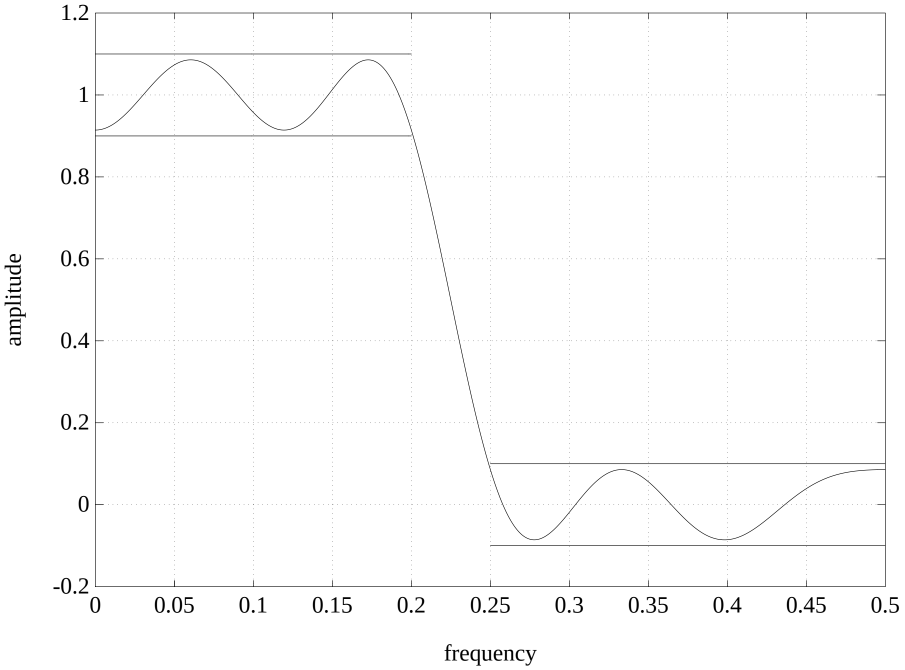

Figure 1. Frequency Response for Example 1, a Length-17 Lowpass Filter

Example 2: Flat Passband, Lowpass, Minimum-Length Filter

Suppose we want a lowpass filter with the same bandedges as in Example 1, but we want the passband to be flat. One simple way to do this is to add a concavity specification that forces the frequency response to be concave down ("$-$") in the passband. We can also relax the upper limit specification in the passband to be hugged, and change the upper limit to 1.0, so that the frequency response can decrease monotonically in the passband from a value of 1. The new specifications are shown below.

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.200	1.000	1.000	h	a
2	limit	−	0.000	0.200	0.900	0.900	n	a
3	limit	+	0.250	0.500	0.100	0.100	n	a
4	limit	−	0.250	0.500	-0.100	-0.100	n	a
5	concave	−	0.000	0.200				

FINDING MINIMUM LENGTH
ODD LENGTHS from 21 to 31
COSINE MODEL (even symmetric coefficients)
201 grid points

The resulting frequency response, shown in Figure 2, has a zero frequency gain of exactly 1 because the upper limit in the passband is hugged. The stopband has the same upper and lower limits as Example 1. The price we pay for the flat passband is an increase in filter length from N=17 for Example 1, to N=29 for this example.

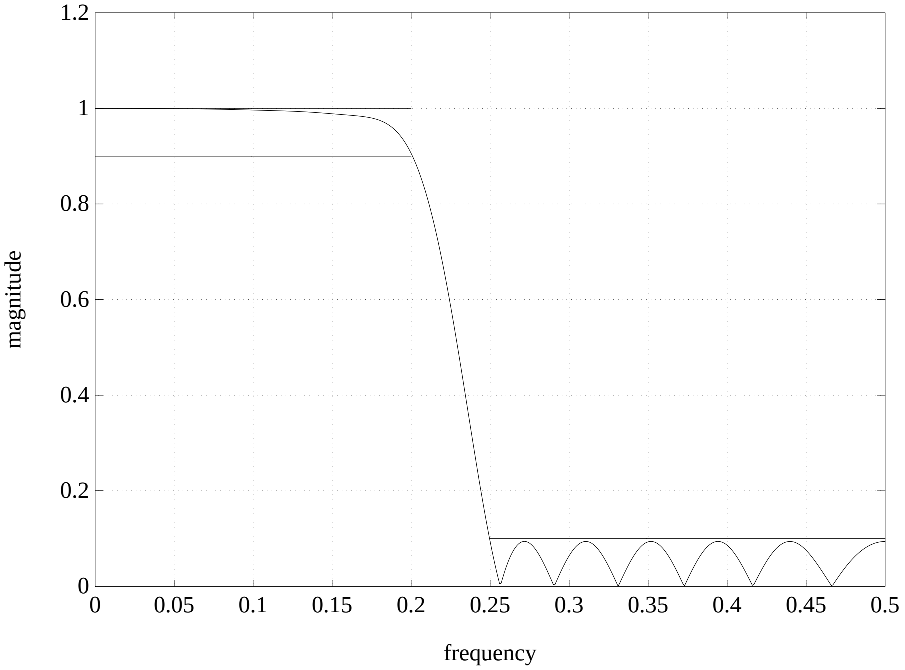

Figure 2. Length-29 Filter with Monotonically Decreasing Passband Response .

Example 3: Flat Passband, Minimum-Phase Filter

If a minimum-phase filter is desired with the same magnitude performance as the linear-phase filter in Example 2, the factorization approach of Herrmann and Schüssler[21] can be used beginning with the length-43 filter which resulted from the following specifications:

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.200	1.000	1.000	h	a
2	limit	−	0.000	0.200	0.810	0.810	n	a
3	limit	+	0.250	0.500	0.010	0.010	n	a
4	limit	−	0.250	0.500	0.000	0.000	h	a
5	concave	−	0.000	0.200				

FINDING MINIMUM LENGTH
ODD LENGTHS from 37 to 55
COSINE MODEL (even symmetric coefficients)
201 grid points

The lower limit of 0.81 in the passband and the upper limit of 0.01 in the stopband are used in anticipation of the square root involved in the mimimum-phase design, while the lower limit of 0.0 in the stopband guarantees a non-negative response. Half of the 42 roots of the length-43 filter, 10 roots inside the unit circle and 1 each of the 11 double roots on the unit circle, are retained to give the length-22 mimimum-phase filter with response shown in Figure 3. This minimum-phase filter, length-22 is slightly shorter than the linear-phase, length-29, filter of Example 2 which meets the same magnitude specifications.

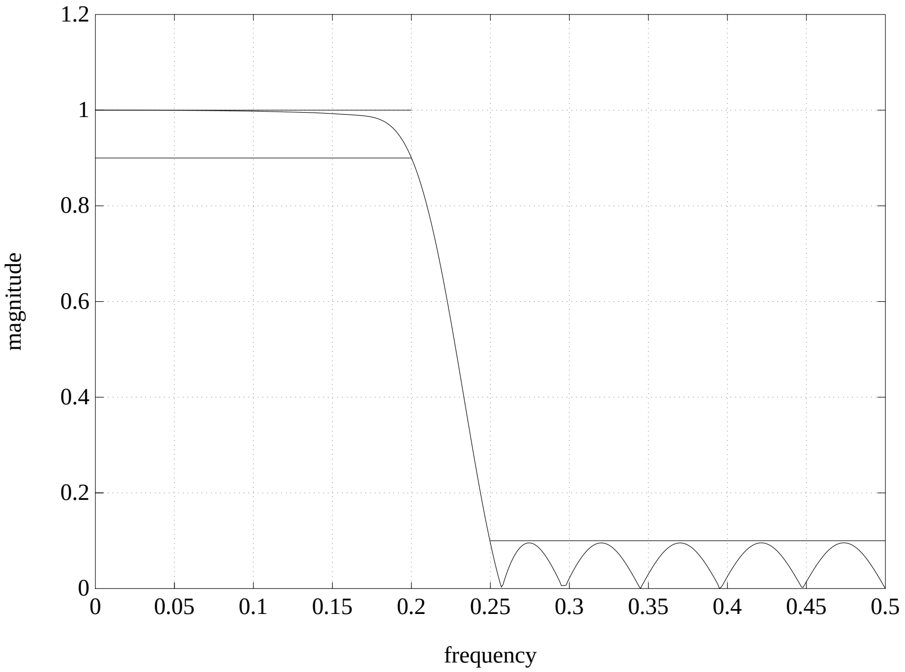

Figure 3. Flat Passband, Length-22, Minimum-Phase Filter.

Example 4: Point Constraints with a Flat Passband Filter

If there are specific frequencies in the stopband where zeros are desired to null out interference, the following specifications which require zeros at frequencies of 0.3 and 0.4 would be appropriate:

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.200	1.000	1.000	h	a
2	limit	−	0.000	0.200	0.900	0.900	n	a
3	limit	+	0.250	0.500	0.100	0.100	n	a
4	limit	−	0.250	0.500	-0.100	-0.100	n	a
5	concave	−	0.000	0.200				
6	limit	+	0.300	0.300	0.000	0.000	n	a
7	limit	−	0.300	0.300	0.000	0.000	n	a
8	limit	+	0.400	0.400	0.000	0.000	n	a
9	limit	−	0.400	0.400	0.000	0.000	n	a

FINDING MINIMUM LENGTH
ODD LENGTHS from 21 to 31
COSINE MODEL (even symmetric coefficients)
201 grid points

In this case there was no increase in length over the length of 29 in example 2, required to meet these additional point constraints; the zeros of the response were simply shifted as shown in Figure 4. Generally, however an increase in length would be required to meet these additional constraints.

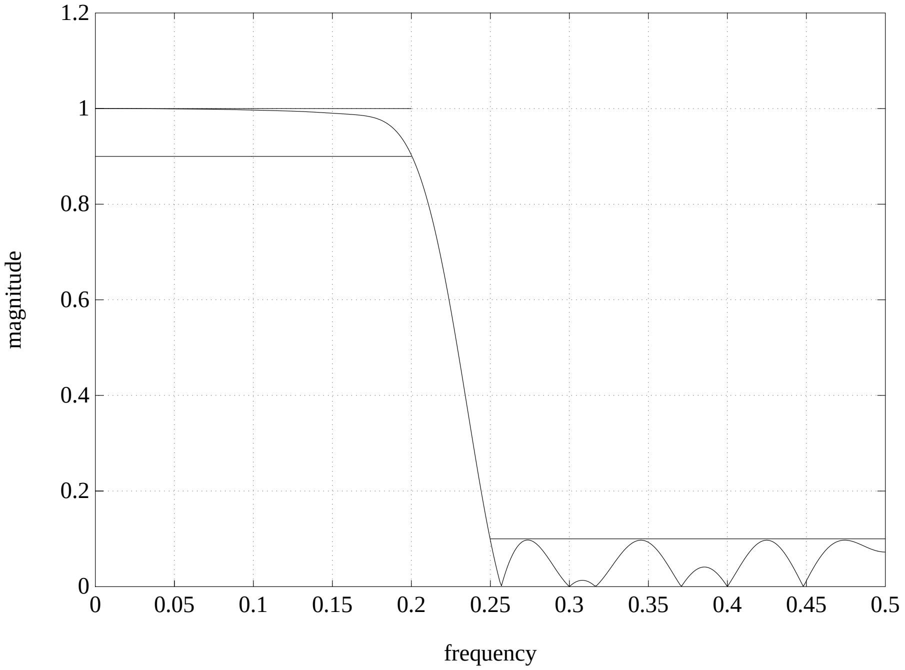

Figure 4. Length-29 Filter with Point Constraints .

Example 5: Partial-Band Differentiator, Pushing the Stopband.

Suppose next we want a linearly increasing magnitude response, followed by rejection at higher frequencies. We know we want a linearly increasing response up to 0.25 cycles/sample, and we want as wide a stopband as possible with a length of 16. We do this by specifying the differentiating band by an upper constraint linearly interpolated from 0.01 to 0.26 that is optimized (pushed away from), and a lower constraint from 0.0 to 0.25 that is hugged. The left bandedges of the upper and lower stopband constraints are then pushed left in the mode "push".

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.250	0.010	0.260	n	a
2	limit	−	0.000	0.250	0.000	0.250	h	a
3	limit	+	0.400	0.500	0.0100	0.0100	n	a
4	limit	−	0.400	0.500	-0.0100	-0.0100	n	a

PUSHING 2 BANDEDGES LEFT, fixed length = 16, bands: 3 4
SINE MODEL (odd symmetric coefficients)
201 grid points

The resulting bandedge is 0.3555, and Figure 5 shows the frequency response.

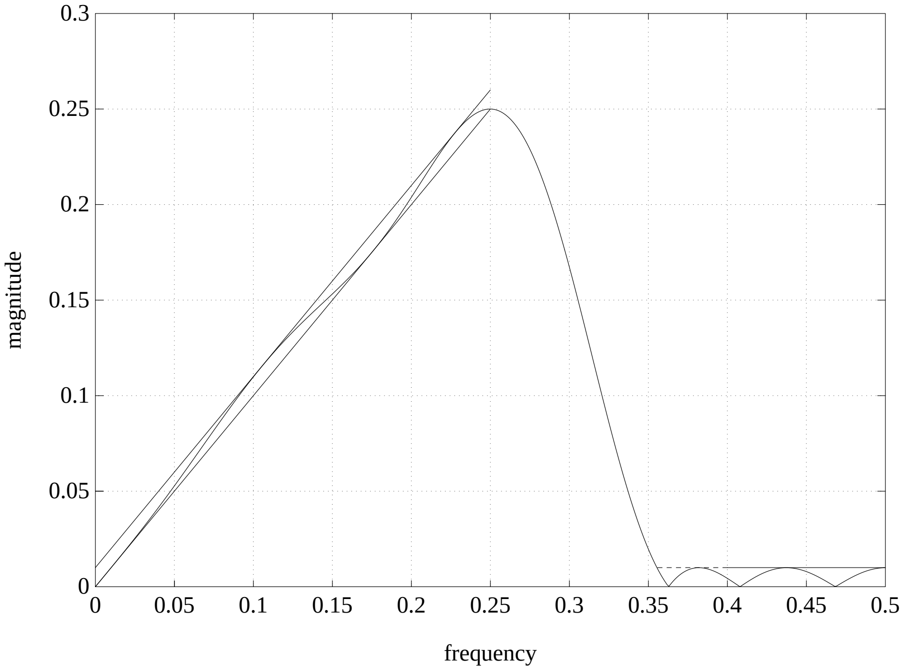

Figure 5. Frequency Response for Example 5, a Length-16 Lowpass Differentiator with a Minimum-Width Transition Band.

Note that the bandedges in the stopband have been pushed to lower frequencies as far as possible until the constraints are hugged. The specifications of any one bandedge for any type of filter, lowpass, bandpass, etc., can be pushed in this manner.

Example 6: Bandpass Filter

This example shows how to find the minimum-length linear-phase filter which meets the frequency specifications listed below and has well-behaved transition bands.

#	type	sense	edge1	edge2	bound1	bound2	hugged?	interp
1	limit	+	0.000	0.080	0.100	0.100	n	a
2	limit	−	0.000	0.080	-0.100	-0.100	n	a
3	limit	+	0.250	0.370	1.100	1.100	n	a
4	limit	−	0.250	0.370	.900	.900	n	a
5	limit	+	0.400	0.500	0.100	0.100	n	a
6	limit	−	0.400	0.500	-0.100	-0.100	n	a

FINDING MINIMUM LENGTH
ODD LENGTHS from 21 to 29
COSINE MODEL (even symmetric coefficients)
201 grid points

The initial design using METEOR, maximizing the distance from all the constraints, produces a filter of length 25, the shortest length that meets these specifications, and a deviation of 0.097846.

The frequency response for this design is shown in Figure 6, and is essentially the same as that produced by the Parks-McClellan program [4]. On the scale of Figure 6, there appears to be a problem in the transition band, but in the bands where the Chebyshev error was minimized, the response looks good.

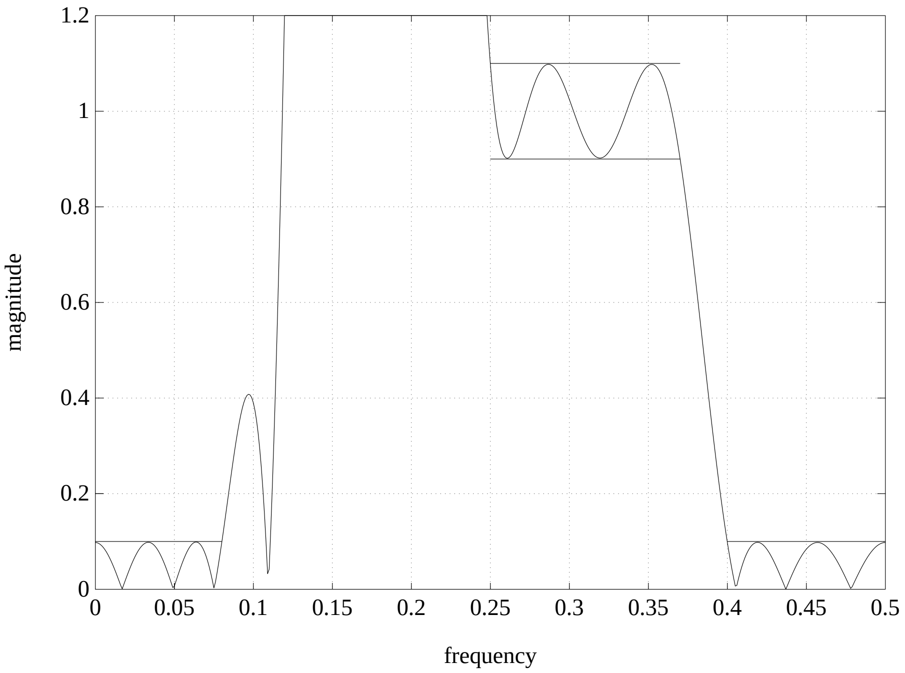

Figure 6. Length-25 Bandpass Filter with a Transition-Band Excursion

To eliminate the transition band excursion, new limits were introduced which constrained the response in the first transition band to lie between -1.1 and $+1.1$. The new algorithm found that the filter length must be increased to 27 in order to meet these new, stricter, limits. The response of this length-27 filter is shown in Figure 7. As in Figure 6, the distance from all the original constraints is maximized, but the response is allowed to touch the new constraints.

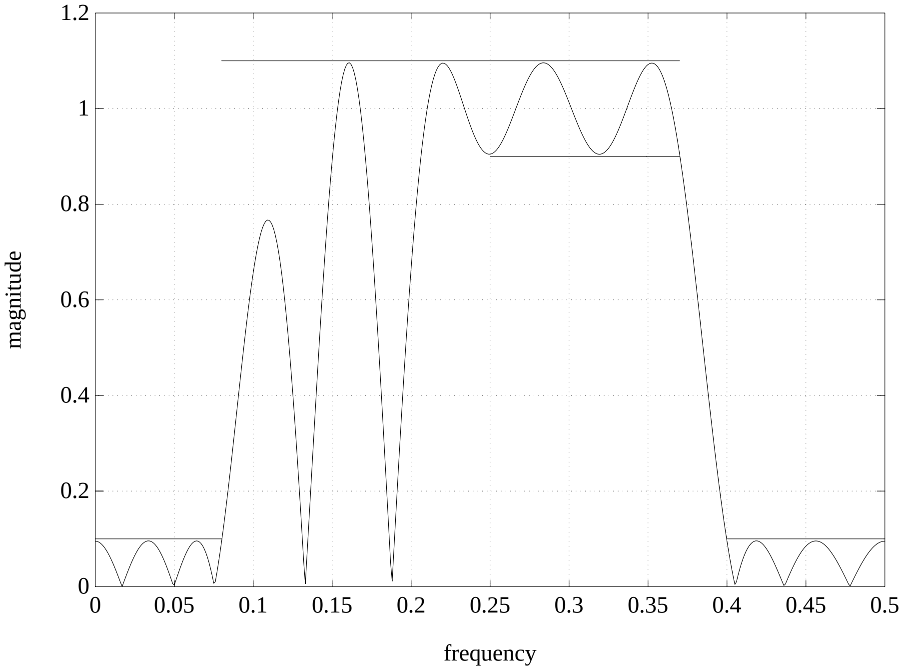

Figure 7. Length-27 Bandpass Filter with Transition Limits.

Another way to eliminate the transition band peak is to fix the length at 27 and push the upper edge of the lower stopband to the right, maximizing the width of the first stopband, thus reducing the width of the first transition band and eliminating the transition band peak. The bandedge found is 0.1667 cycles/sample, and corresponds to a deviation of 0.099998 . The resulting frequency response is shown in Figure 8.

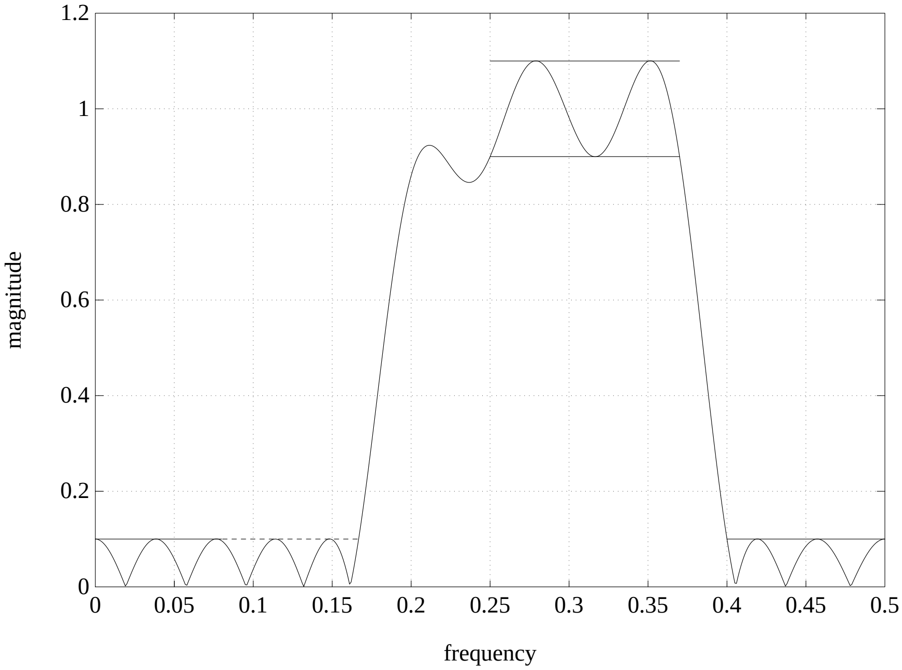

Figure 8. Length-27 Bandpass Filter with Mimimum-Width Transition Band.

5. Timing Comparison with the Parks-McClellan Program

The Parks-McClellan program runs much faster than METEOR, as we would expect given METEOR's greater generality, and the fact that METEOR uses the simplex algorithm instead of the Remes exchange algorithm. However, the running time of METEOR on present-day computers is not prohibitive even for reasonably large problems. To illustrate this, we give some timing comparisons on a SPARCstation 1+ (Model 4/65) using an f77 compiler at optimization level 03, and a Pascal compiler at optimization level 2.

The examples run were simple fixed-length lowpass filters of length L, with $L = 2^i, i = 4, ..., 8$; passband [0, 0.1]; and stopband $[x,\ 0.5]$, where $x = 0.1 * (1 + 2^{(4-i)})$. Thus the filters were of length-16 with stopband [0.2, 0.5]; length-32 with stopband [0.15, 0.5]; etc. The left edge of the stopband was moved left as the filter was made longer to keep the deviation from specifications roughly constant over the examples. The number of grid points was kept comparable in the programs by choosing grid density 10 in the Parks-McClellan program, and using $10 * (L/2) + 1$ grid points in METEOR. The upper and lower bounds in the passband were 1.5 and 0.5; and 0.5 and -0.5 in the stopband. Table 1 shows the optimal deviations from 1 in the passband (or 0 in the stopband) found by each program, as well as the user cpu times.

SPARCstation 1+ Model 4/65

	Parks-McClellan		Meteor		Meteor (convex passband)	
length	deviation	cpu seconds	deviation	cpu seconds	deviation	cpu seconds
16	0.0283516	0.1	0.0283515	0.2	0.0397526	0.2
32	0.0235960	0.1	0.0235959	0.7	0.0467533	0.7
64	0.0213816	0.5	0.0213815	3.8	0.0509093	4.3
128	0.0185210	2.0	0.0185210	32.7	0.0550728	31.6
256	0.0186837	3.9	0.0186838	184.6	0.0589392	207.0

Table 1. Timing Comparison

The deviations of the two programs check to 6 significant figures. The Parks-McClellan program is clearly much faster, by a factor increasing with filter length, to a factor of 16 for the length-128 problem, and 47 for the length-256 problem. However, the cpu time of about 3 minutes on a modern workstation for a length-256 filter would hardly be prohibitive in most situations. Of course when the minimum length is sought, with no prior estimate, binary search on the length will result in as many as $8 = \log(256)$ instances of optimizations.

Also shown are the deviations and running times for the same problems, using METEOR, but with concave-down passbands in [0, 0.1]. The passband was specified by a single-point upper bound at the left-edge, a single-point lower bound at the right edge, and a concavity constraint in the entire band. The upper and lower bounds are redundant within the passband, and this strategy reduces the number of columns generated by METEOR. The running times are roughly the same as those for the traditional design, and the results illustrate the price paid in increased deviation by constraining the passband to be concave down.

6. Conclusion

A new approach to filter design, using the simplex method of linear programming, was proposed which is very general and can incorporate a wide variety of constraints on the frequency response of the filter. Several examples were presented to illustrate the wide range of applications of this approach to linear-phase filter design. We are presently working on extensions of this approach to the design of filters with constraints on group delay and/or phase as well as magnitude.

Acknowledgement

The authors would like to thank Professor H.W. Schüssler for his careful review of this manuscript and for his helpful suggestions.

References

[1] W. Cauer, *Siebschaltungen*, V.D.I. Verlag G.M.B.H., Berlin, 1931.

[2] O. Herrmann, "Design of Nonrecursive Digital Filters With Linear Phase," *Electronics Letters*, vol. 6, pp. 328–329, May 28, 1970 (Reprinted in *Digital Signal Processing*, L.R. Rabiner and C.M. Rader, editors, IEEE Press, New York, NY 1972).

[3] T.W. Parks, L.R. Rabiner, and J.H. McClellan, "On the Transition Width of Finite Impulse-response Digital Filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU–21, pp. 1–4, February, 1971.

[4] L.R. Rabiner, J.H. McClellan, and T.W. Parks, "FIR Digital Filter Design Techniques using Weighted Chebyshev Approximation," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 595–610, April 1975.

[5] O. Herrmann and H.W. Schüssler, "On the Design of Selective Nonrecursive Digital Filters," presented at the IEEE Arden House Workshop on Digital Filtering, January 12, 1970.

[6] E.M. Hofstetter, A.V. Oppenheim, and J. Siegel, "A New Technique For the Design of Non-recursive Digital Filters," *Proc. Fifth Annual Princeton Conference on Information Sciences and Systems*, Princeton, N.J., pp. 64–72, March 1971 (Reprinted in *Digital Signal Processing*, L.R. Rabiner and C.M. Rader, editors, IEEE Press, New York, NY 1972).

[7] E.M. Hofstetter, A.V. Oppenheim, and J. Siegel, "On Optimum Nonrecursive Digital Filters," *Proc. Ninth Annual Allerton Conference on Circuit and System Theory*, pp. 789–798, October 1971 (Reprinted in *Digital Signal Processing*, L.R. Rabiner and C.M. Rader, editors, IEEE Press, New York, NY 1972).

[8] H.D. Helms, "Digital Filters with Equiripple or Minimax Responses," *IEEE Trans. Audio and Electroacoustics*, vol. AU–19, pp. 87–94, March 1971. (Reprinted in *Digital Signal Processing*, L.R. Rabiner and C.M. Rader, editors, IEEE Press, New York, NY 1972.)

[9] T.W. Parks, "Extensions of Chebyshev Approximation for Finite Impulse Response Filters," presented at the IEEE Arden House Workshop on Digital Filtering, January 10, 1972.

[10] T.W. Parks and J.H. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory*, vol. CT–19, pp. 189–194, March 1972.

[11] H.S. Hersey, D.W. Tufts, and J.T. Lewis, "Interactive Minimax Design of Linear Phase Nonrecursive Digital Filters Subject to Upper and Lower Function Constraints," *IEEE Trans. Audio and Electroacoustics*, vol. AU–20, pp. 171–173, June 1972.

[12] M.T. McCallig and B.J. Leon, "Constrained Ripple Design of FIR Digital Filters," *IEEE Trans. Circuits and Systems*, vol. CAS–25, pp. 893–902, November, 1978.

[13] F. Grenez, "Constrained Chebyshev Approximation for FIR Filters," *Proc. 1983 Int. Conf. on Acoustics, Speech, and Signal Processing*, Boston, Mass., pp. 194–196.

[14] O. Herrmann, L.R. Rabiner, and D.S.K. Chan, "Practical Design Rules for Optimum Finite Impulse Response Low-pass Digital Filters," *Bell Syst. Tech. J.*, vol. 52, pp. 769–799, July-August 1973.

[15] J.F. Kaiser and K. Steiglitz, "Design of FIR Filters with Flatness Constraints," *Proc. 1983 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Boston, Mass., pp. 197–200, April 14–16, 1983.

[16] L.R. Rabiner, J.F. Kaiser, and R.W. Schafer, "Some Considerations in the Design of Multiband Finite-Impulse-Response Digital Filters," *IEEE Trans. Audio and Electroacoustics*, vol. ASSP–22, pp. 462–472, December 1974.

[17] L.R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975; see pp. 75–84.

[18] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[19] K. Steiglitz, "Optimal Design of FIR Digital Filters with Monotone Passband Response," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP–27, no. 6, pp. 643–649, December 1979.

[20] W.J. Knight, "Search in an Ordered Array Having Variable Probe Cost," *SIAM J. Comput.*, vol. 17, no. 6, pp. 1203–1214, December 1988.

[21] O. Herrmann and H.W. Schüssler,"Design of Nonrecursive Digital Filters with Minimum Phase", *Electronics Letters*, vol. 26, no. 11, May 28, 1970 (Reprinted in *Digital Signal Processing*, L.R. Rabiner and C.M. Rader, editors, IEEE Press, New York, NY 1972).

[22] K. Steiglitz and T. W. Parks, "What is the Filter Design Problem?," *Proc. 1986 Princeton Conf. on Information Sciences and Systems*, Princeton, NJ, pp. 604–609, March 1986.