Peak-Constrained Least-Squares Optimization

John W. Adams, *Senior Member, IEEE,* and James L. Sullivan

***Abstract*— We presented the basic concepts for peak-constrained least-squares (PCLS) optimization in previous papers. We present advanced PCLS optimization concepts in this paper.**

I. Introduction

In the past, most digital filters were designed according to the minimax (MM) or least-squares (LS) optimality criteria. MM filters were used in applications where the peak errors were more important than the total squared errors. LS filters were used in applications where the total squared errors were more important than the peak errors.

In [1] and [2], we showed that MM and LS optimization problems can both be viewed as special cases in the class of peak-constrained least-squares (PCLS) optimization problems. In PCLS optimization problems, we constrain the peak error while minimizing the total squared error.

We pronounce PCLS as "pickles." When selecting the terminology in [1] and [2], we also considered maximum-constrained least-squares (MCLS). However, PCLS was easier to pronounce than MCLS.

We showed how to use PCLS optimization to design symmetric FIR digital filters and windows in [1] and [2]. We also showed that the tradeoff between the total squared error and the peak error has the fundamental shape shown in Fig. 1. In particular, we used Kuhn–Tucker multiplier theory in [1] and [2] to prove that the tradeoff monotonically decreases and terminates with zero slope for symmetric FIR digital filters and windows. We use a more general theory in Section III-D of this paper to prove the PCLS tradeoff theorem (PTT). It states that the tradeoff monotonically decreases and terminates with zero slope for *all* types of filters, including IIR digital filters, complex FIR digital filters, and analog filters.

The best solutions for most practical applications are in the knees of tradeoff curves. The LS and MM solutions are at the endpoints where the slopes are the most extreme. Therefore, the LS and MM solutions are the two special cases of PCLS solutions that have the worst performance tradeoffs. Ironically, the filter design literature and textbooks are dominated by these extremely bad special cases.

Starting from the LS solution, a very large reduction in the peak error can be obtained at the expense of a very small increase in the total squared error. Starting from the MM solution, a very large reduction in the total squared error can be obtained at the expense of a very small increase in the peak error. Therefore, we argued in [1] and [2] that LS and MM solutions are inherently inefficient.

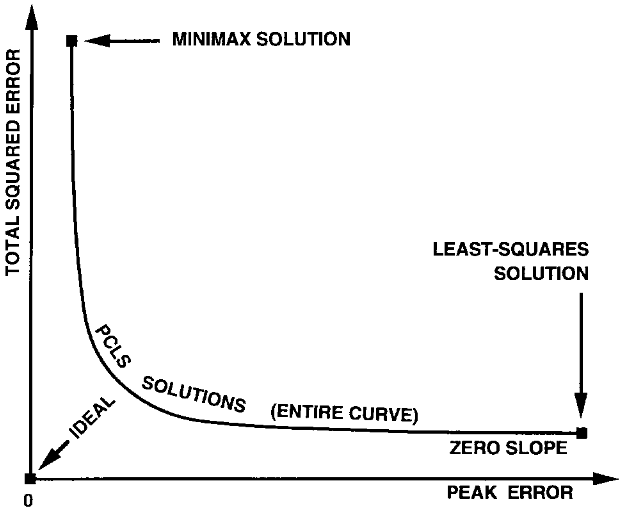


Fig. 1. Tradeoff between total squared error and peak error.

Very few filter design papers discuss systematic approaches to making tradeoffs between conflicting performance measures. In [10], we have one of the rare papers that deals with this important practical problem. In [10], systematic strategies for handling the tradeoff between the error energy in a filter's frequency response and its sensitivity to coefficient errors are discussed. The strategies in [10] are based on the theory of "multicriterion optimization." One strategy optimizes the weighted sum of normalized performance measures. Another strategy constrains one performance measure while optimizing the other. PCLS optimization is based on this strategy.

We first presented the PCLS optimality criterion in [29] along with our first algorithm for PCLS optimization. It was an iterative reweighted least-squares (IRLS) algorithm, and it was very slow. Moreover, it was not guaranteed to converge. On the other hand, the algorithm in Section IV of this paper is very fast, and it is guaranteed to converge to optimal PCLS solutions.

PCLS optimization problems are special forms of constrained least-squares (CLS) problems. A CLS problem is not in the PCLS category unless it includes inequality constraints that are used to control the error peaks on a smooth function. For the sake of brevity in this paper, maxima and minima of an error function are both called "error peaks."

Most constrained optimization algorithms use a single exchange of active constraints from one iteration to the next.

Manuscript received August 28, 1995; revised June 26, 1997. This work was supported in part by NSF Grant MIP-9200581. The associate editor coordinating the review of this paper and approving it for publication was Dr. Victor E. DeBrunner.

J. W. Adams is with the Electrical and Computer Engineering Department, California State University, Northridge, CA 91330-8346 USA.

J. L. Sullivan is with Allied Technical Services Corporation, Pasadena, CA 91107 USA.

Publisher Item Identifier S 1053-587X(98)01331-2.

Single exchange algorithms are appropriate for solving general CLS problems where the constraints are arbitrary. Unfortunately, single exchange algorithms converge very slowly. If a CLS problem includes peak-error constraints on a smooth function, then we can take advantage of multiple exchanges to improve the rate of convergence. Therefore, it is important to determine whether a CLS problem is in the PCLS category before selecting the optimization algorithm.

We presented a new algorithm for PCLS optimization, which is called the multiple exchange algorithm, in [1]–[3]. We showed how to use the method of Lagrange multipliers in a systematic sequence of multiple exchanges to quickly solve PCLS problems. In addition, we showed how to apply the Kuhn–Tucker conditions in the context of PCLS filters. (A generic discussion of Kuhn–Tucker conditions is in [28].)

The original multiple exchange algorithm usually converged to optimal solutions for lowpass FIR filter design problems. The examples in [1] were designed by the multiple exchange algorithm, and they were all confirmed to be optimal with the Kuhn–Tucker conditions. However, the original multiple exchange algorithm is not guaranteed to always converge to optimal solutions. In particular, it can converge to suboptimal solutions with negative Kuhn–Tucker multipliers.

A modified multiple exchange algorithm that inspected the polarities of the Kuhn-Tucker multipliers in each iteration was presented in [4] and [5]. If one or more Kuhn–Tucker multipliers were negative, it temporarily switched to single exchanges to drop the offending constraints until it obtained an active set with nonnegative Kuhn–Tucker multipliers, and then, it switched back to multiple exchanges. (It appears to us that this same modification of the multiple exchange algorithm was described as a new modification in [9] and was used in [13]. We assume that this modification was developed independently.) This modification guaranteed that the solution was optimal if the algorithm converged. Unfortunately, this modification was not sufficient to guarantee convergence in [4], [5], [9], or [13].

In order to guarantee convergence to the optimal solution, we introduced the generalized multiple exchange algorithm in [2] and provided more details in [6] and [7]. The algorithm is "generalized" in the sense that it can do both single and multiple exchanges. We would prefer to have a more specific and descriptive name for the algorithm because the term "generalized" is vague and overused. However, we continue to call it the generalized multiple exchange algorithm for lack of a better name.

In [6] and [7] it was proved that the generalized multiple exchange algorithm is guaranteed to converge to the unique optimal solution of any feasible positive-definite quadratic programming problem. (This type of problem naturally arises for real symmetric FIR digital filters and windows.) The generalized multiple exchange algorithm does multiple exchanges using the fundamental concepts in [1]–[3]. In addition, it includes the method from [4] and [5] that inspects the Kuhn–Tucker multiplier polarities and uses single exchanges to drop constraints with negative Kuhn–Tucker multipliers. It also uses single exchanges to exploit the convergence properties of the Goldfarb–Idnani algorithm [20].

We proposed combining the multiple exchange and Goldfarb-Idnani algorithms in [2], where we stated that "This is a natural combination because the Goldfarb-Idnani algorithm does not require primal feasibility until the last iteration is completed." Most quadratic programming algorithms in the mathematics literature require primal feasibility at the beginning and end of each iteration. They are inefficient when combined with the multiple exchange algorithm. We studied numerous (more than 30) single-exchange quadratic programming algorithms, and we concluded that the Goldfarb–Idnani algorithm is the best one to use in conjunction with multiple exchanges.

We presented several examples of optimal filters that were designed with the generalized multiple exchange algorithm in [12], including multiband filters that failed to converge with the original multiple exchange algorithm. We also discussed multirate applications for FIR PCLS filters in [12]. However, we called them FIR CLS filters in [12] to be consistent with the title of that conference paper. Coincidentally, the L_p approach to dealing with the tradeoff between peak error and error energy in FIR filters was presented in [11] at the same conference. The L_p optimality criterion permits the filter designer to obtain solutions that are between L_2 and L_∞, but it does not permit the designer to make a direct tradeoff between peak error and error energy.

II. Optimality Criteria

The PCLS optimality criterion is easy to customize for different applications. For example, in PCLS filter design problems, we can minimize the total weighted-squared error ε

$$\varepsilon = \int_{-0.5}^{0.5} W(e^{j2\pi f})|H(e^{j2\pi f}) - H_d(e^{j2\pi f})|^2\, df$$

subject to inequality constraints on the error magnitude

$$|H(e^{j2\pi f}) - H_d(e^{j2\pi f})| \le \delta(e^{j2\pi f})$$

and inequality constraints on the phase

$$PH_l(e^{j2\pi f}) \le PH(e^{j2\pi f}) \le PH_u(e^{j2\pi f})$$

and inequality constraints on the phase delay

$$PD_l(e^{j2\pi f}) \le \frac{-PH(e^{j2\pi f})}{2\pi f} \le PD_u(e^{j2\pi f})$$

and inequality constraints on the group delay

$$GD_l(e^{j2\pi f}) \le \frac{-dPH(e^{j2\pi f})}{2\pi\, df} \le GD_u(e^{j2\pi f})$$

and direct equality constraints on the variables

$$\boldsymbol{C}_e\boldsymbol{v} - \boldsymbol{c}_e = 0$$

and direct inequality constraints on the variables

$$\boldsymbol{C}_i\boldsymbol{v} - \boldsymbol{c}_i \ge 0.$$

$H_d(e^{j2\pi f})$ denotes the desired frequency response. $H(e^{j2\pi f})$ denotes the actual frequency response. $PH(e^{j2\pi f})$ denotes the actual phase response. The lower and upper inequality constraints are indicated with l and u subscripts. $\boldsymbol{v}$ denotes the vector of variables (filter coefficients). $W(e^{j2\pi f})$ denotes the squared-error weighting function.

Arbitrary functions can be specified for $W(e^{j2\pi f})$ in PCLS optimization problems. In most practical applications, we specify the squared-error weighting to be zero in at least one band, such as a transition band. However, in some applications, we specify the squared-error weighting to be nonzero at all frequencies. For example, in [2, Sec. III-G], we minimize the error energy using a weighting of 1.0 in the entire digital frequency band from -0.5 to $+0.5$ cycles/sample. (In our opinion, the discussion in [13] implies that we always use zero weighting in at least one band when we do PCLS optimization. There are actually no restrictions on PCLS weighting functions.)

When designing symmetric FIR digital filters, we can impose constraints on the zero-phase response $H_0(e^{j2\pi f})$ as

$$\alpha_l(e^{j2\pi f}) \leq H_0(e^{j2\pi f}) \leq \alpha_u(e^{j2\pi f}).$$

$\alpha_l(e^{j2pf})$ and $\alpha_u(e^{j2\pi f})$ denote the lower and upper limits on the zero-phase response. (A nonnegative zero-phase response can be obtained by setting $\alpha_l(e^{j2\pi f}) = 0$ as in [30].) We can also specify constraints on the derivative of the zero-phase response

$$\beta_l(e^{j2\pi f}) \leq \frac{dH_0(e^{j2\pi f})}{df} \leq \beta_u(e^{j2\pi f}).$$

One of the anonymous reviewers of this paper questioned whether band edge frequencies are inflexible in practical design problems. We believe that band edge frequencies are often inflexible and must be controlled in many practical applications. For example, passband edge frequencies need to be controlled in communication filters to pass the channels of interest. Stopband edge frequencies need to be controlled in multirate filters to suppress the aliased signals in the appropriate frequency bands.

For the sake of simplicity in the following discussion about the importance of band edge frequencies, we will focus on lowpass filters, and we will use the notation in [1].

L impulse response length;
F_p passband edge frequency in cycles per sample;
F_s stopband edge frequency in cycles per sample;
DB_p passband variation in decibels;
DB_s peak stopband gain in decibels.

We believe that the primary reason why minimax optimization has generally been more popular than least-squares optimization is because of the ability to specify band edge frequencies. For example, in most lowpass MM filter design programs, the user can specify the passband edge frequency F_p and the stopband edge frequency F_s. In most lowpass LS filter design programs, the user can only specify a single "cut-off" frequency, which is usually denoted as F_0.

When using a lowpass LS filter design program, the user typically specified F_0 to be between the desired F_p and F_s. The resulting filter would typically have an unacceptably large attenuation for passband signals near F_p and an unacceptably large gain for stopband signals near F_s. Designers usually needed to control the gains at band edge frequencies, and this led them to prefer the MM method over the LS method.

On the other hand, there are some applications where band edge frequencies are flexible. For example, there are spectral analysis applications where stopband edge frequencies are flexible for windows. These windows can be designed with ripple-bounded least-squares (RBLS) optimization. (RBLS can be pronounced as "rebels.") In particular, the ripple bounded maximum directivity (RBMD) window is designed with RBLS optimization in [18]. In RBLS design problems, there is at least one frequency band where inequality constraints are used for ripple peaks (local extrema or stationary points) of the error function but not for band edges.

In [18, sec. III], the RBMD window with flexible stopband edge frequency was discussed, and it includes an example. We note that the RBMD window can be considered to be a special case of the peak-constrained maximum directivity (PCMD) window presented in [2, Sec. III-G]. Given an RBMD window, we can always find a PCMD window that is identical to it. In particular, the RBMD example in [18, Fig. 3] is identical to the PCMD example in [2, Fig. 11].

The PCMD example in [2] was designed to minimize the total energy for *all* frequencies subject to a unity dc gain constraint and a maximum gain of -30.0 dB in the interval $[F_s, 0.5]$, where F_s was specified to be 0.013 95. The RBMD example in [18] was designed to minimize the total energy for *all* frequencies subject to a unity dc gain constraint and a maximum gain of -30.0 dB at the frequencies of sidelobe ripple peaks (local extrema), but F_s was unspecified. The RBMD example in [18] is identical to the PCMD example in [2] because the inequality constraint at F_s is inactive in [2].

Another example of RBLS optimization was presented in [18, Sec. IV] for a lowpass FIR filter. The total energy was minimized in the interval $[F_p, 0.5]$ subject to $L = 95, F_p = 0.0625, DB_p \leq 1.0$, and $DB_s \leq -35.0$. F_s was unspecified. The resulting filter had a ripple-bounded stopband.

In practice, this type of RBLS filter can arise at an intermediate stage in the design of a lowpass decimation filter, where F_s depends on the decimation ratio, but the decimation ratio has not yet been determined. In the first cut and try, a ripple-bounded stopband filter is designed, and its resulting F_s (corresponding to DB_s) is measured. Based on this initial estimate for F_s, the nearest appropriate integer is selected for the decimation ratio DR, and DR is then used to determine the final specification for F_s. In most applications, we specify $F_s = (1/DR) - F_p$ so that only stopband signals are permitted to alias into the passband. In particular, a signal at F_s aliases to be at F_p after it is decimated by DR. Therefore, controlling the gain at F_s is very important in the final design.

The final PCLS filter is designed to minimize the total weighted energy of the important aliased signals (which are usually the signals that alias into the passband), given L, F_p, F_s, DB_p, DB_s, and DR. The squared-error weighting for the final PCLS filter is usually specified to be zero in the transition band because those frequencies do not alias into the passband. (Transition band signals alias back into the transition band.) If a nonzero squared-error weighting were used in the transition band, the resulting filter would reduce the energy of signals aliasing into the transition band at the expense of increasing the energy of signals aliasing into the passband. That would be an undesirable trade in most applications.

Given a RBLS filter, it is always possible to find a PCLS filter that is identical to it. On the other hand, given a PCLS filter, it is not always possible to find a RBLS filter that is identical to it. Therefore, RBLS filters can be considered to be special cases of PCLS filters.

In [13], an extensive discussion of RBLS filters is provided, and applications for filters with unspecified transition bandwidths and unspecified gains at band edge frequencies are described. Examples of lowpass filters are also included, where the passband and stopband edge frequencies are unspecified.

In [13], we are provided with interesting justifications for using RBLS filters in some special applications. However, for most practical applications, we believe that [13] overemphasizes the importance of RBLS filters compared with PCLS filters.

In [17], we find a discussion that refers to [13]. We believe that the discussion in [17] exaggerates the importance of RBLS filters significantly more than in [13]. Moreover, we believe that the overemphasis on RBLS filters is exaggerated in [17] to the point of excluding PCLS filters with specified band edge frequencies. We also believe that the constrained least-squares symmetric FIR filter design algorithms used in [17] do not always converge.

In our opinion, the discussion in [17] (especially on pp. 2–27) implies that the primary advantage of constrained least squares optimization is the ability to design filters with unspecified transition bandwidths and unspecified gains at band edge frequencies. We disagree with this implication for two reasons. First, we believe that the primary advantage of constrained least squares optimization is the ability to control the tradeoff between peak error and total weighted squared error. Second, in most practical applications, we believe that it is important for the filter designer to have the ability to specify inequality constraints on the gains at band edge frequencies. (Without this ability, the gain at a passband edge may be too small, and the gain at a stopband edge may be too high.) It appears to us that the constrained least squares filter design algorithms in [17] do not provide this ability. Moreover, it seems to us that the discussion in [17] implies that this inability is desirable.

III. PCLS Digital Filter Examples

A. FIR Filter with Monotonic Passband

We now consider the following symmetric FIR lowpass filter design problem: Minimize the stopband energy subject to $L = 118, F_p = 0.0625, F_s = 0.0804, DB_p \leq 1.0$ dB, $DB_s \leq -45.0$ dB and constrain the passband to be monotonically decreasing. The specifications in this example are identical to the ones used in [1, Sec. III-A] except that the passband is required to be monotone here, and the impulse response length is 118. The monotonic passband is obtained by including the following inequality constraints in the PCLS optimization:

$$\frac{dH_0(e^{j2\pi f})}{df} \leq 0 \quad \text{for } 0 \leq f \leq 0.0625$$

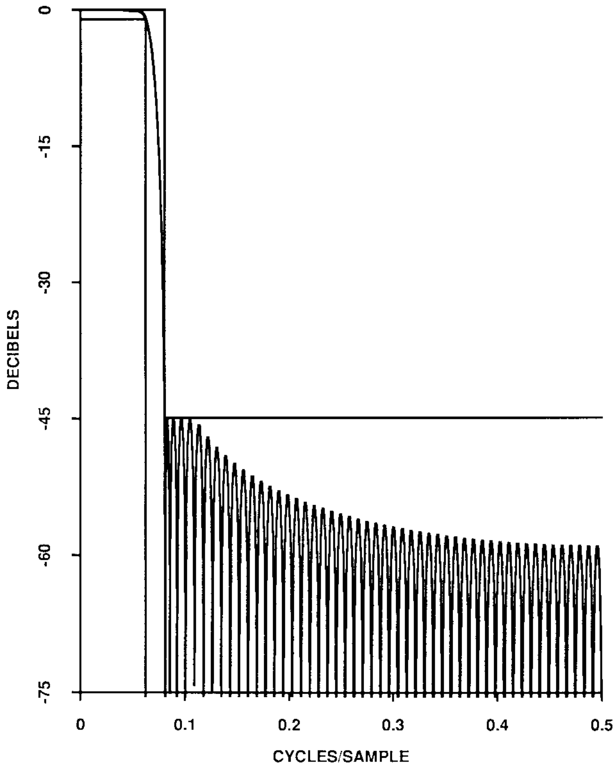


Fig. 2. Symmetric FIR filter with monotonic passband.

The optimal solution is shown in Fig. 2. Refer to [1, Fig. 3] to see the results for the corresponding example where the passband was not constrained to be monotone. The passband turned out to be equiripple in [1].

We also presented a PCLS digital filter example with a monotonic passband in [12, Fig. 2]. It was a multirate filter with multiple stopbands to attenuate signals that aliased into the passband. The monotonic-passband filter in [12] was motivated by a radar application where signal frequencies migrated through the passband. If the passband had ripples, they would have produced periodic amplitude modulations that would have created false radar echos.

As alternatives to filters with monotonic passbands, filters with maximally-flat passbands could be used in applications where passband ripples are objectionable. However, the number of derivatives that are set to zero must be an integer in a maximally-flat filter. Therefore, it is difficult for a maximally-flat filter to efficiently meet a specification on the passband gain variation, such as the 1.0-dB specification in this example. We believe that filters with maximally-flat passbands are discussed in many textbooks because they are easy to design and not because they are the best filters for practical applications.

B. Lowpass Asymmetric FIR Filter

If the impulse response is asymmetric, then we can simultaneously constrain the frequency response magnitude and group delay. (We note that an approach to constraining the magnitude of the complex error in asymmetric FIR filters was discussed in [9], but the phase and delay were unconstrained in [9].) As an example of simultaneous PCLS optimization of the frequency response magnitude and group delay, we designed a filter to minimize the stopband energy subject to $L = 95, F_p = 0.0625, F_s = 0.0804, DB_p \leq 1.0, DB_s \leq -45.0, GD_l \geq 47.24$ samples, and $GD_u \leq 47.26$ samples. Fig. 3(a) shows the overall frequency response. Figs. 3(b) and (c) show blowups of the passband gain and group delay. The group delay has an equiripple behavior centered around 47.25 samples and constrained within ± 0.01 samples. As a practical application, this type of asymmetric filter with specified group delay is useful for digital range gate tracking of a reference point to synchronize radar echos. (Each radar echo needs to be shifted by a specified fraction of a sample. The worst-case asymmetry occurs in the case of a 0.25 sample shift as in this example.) For purposes of comparison, we refer to [1, Fig. 3] to see the corresponding symmetric PCLS filter with the same specifications as in this example, except the group delay was 47.0 samples.

We presented the key concepts for the simultaneous PCLS optimization of the frequency response magnitude and group delay in [8] and [25]. The same methods can readily be used to perform simultaneous PCLS optimization of the frequency response magnitude, phase, phase delay, and group delay. However, in most applications, only one phase-related quantity needs to be included in the optimization.

C. Multiband FIR Filter with Symmetric Impulse Response

We now consider a multiband FIR filter design example that fails to converge with the original multiple exchange algorithm in [1]. The specifications in this example were developed to challenge the robustness of the generalized multiple exchange algorithm.

The objective is to minimize the stopband energy ε in the interval [0.35, 0.5] subject to the peak gain specification δ in the same stopband and subject to the following inequality constraints in two other stopbands:

$$|H(e^{j2\pi f}) \leq \begin{cases} 0.125f, & \text{for } 0.08 \leq f < 0.16 \\ 0.01, & \text{for } 0.16 \leq f \leq 0.22 \end{cases}$$

and subject to the following inequality constraints in two passbands:

$$0.9 \leq |H(e^{j2\pi f})| \leq 1.0 \quad \text{for } 0 \leq f \leq 0.05$$
$$\text{and} \quad 0.25 \leq f \leq 0.33$$

and subject to a stopband null constraint: $H(e^{j2\pi f}) = 0.0$ at $f = 0.414$ cycles/sample. The impulse response is required to be real and symmetric. The unique optimal solution is shown in Fig. 4(a) for the case where δ is specified to be 0.03. Fig. 4(b) shows the tradeoff between ε and δ.

The specifications in this example are challenging because of the discontinuity at 0.16 cycles/sample. The specifications require that $|H(e^{j2\pi f})| \leq 0.02$ for $f = 0.159\,999\cdots$ cycles/sample, and they require that $|H(e^{j2\pi f})| \leq 0.01$ for $f = 0.16$ cycles/sample.

Fig. 4(c) shows the minimax solution obtained from minimizing δ. Although the filter in Fig. 4(c) is a symmetric FIR filter with an equiripple frequency response, it cannot be designed with the Parks–McClellan algorithm. The filter in Fig. 4(c) is actually a "constrained minimax" (CMM) filter because the design problem includes constraints, such as the stopband null constraint at 0.414 cycles/sample and the specified limits on the ripples in the two passbands and the lower two stopbands. CMM problems are usually solved with a single-exchange linear programming algorithm such as the simplex algorithm. However, single-exchange linear programming algorithms are very slow.

We used multiple exchanges to design the filter in Fig. 4(c). We have developed several variations of the generalized multiple exchange algorithm for CMM problems, and we plan to present them in future papers. (One method systematically adjusts the parameter to be minimized, such as δ in this example, until the CMM solution is obtained. Another method converts the CMM problem into a QP problem.)

D. IIR Digital Filter

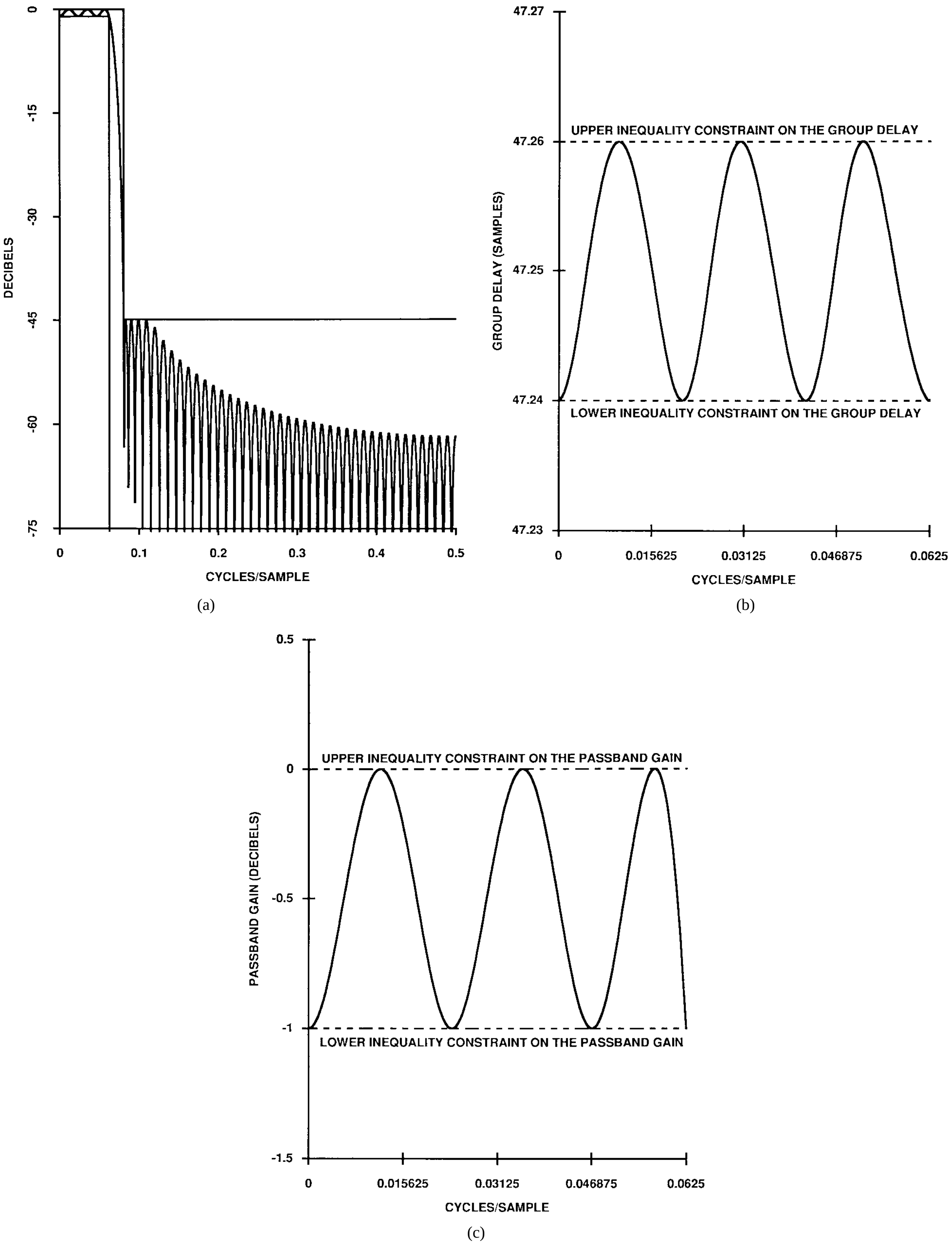
We now consider an eighth-order IIR lowpass filter. The filter in Fig. 5(a) was designed to minimize the stopband energy ε subject to the following specifications: $F_p = 0.0625, F_s = 0.065, DB_p \leq 1.0$, and $DB_s \leq -45.0$ (corresponding to $\delta = 0.005\,6234$). Fig. 5(b) shows a blowup of the passband. Fig. 5(c) shows the tradeoff between the stopband energy ε and the peak stopband gain δ. Fig. 5(d) shows the constrained minimax filter where the passband variation is constrained to be less than or equal to 1.0 dB, and δ is minimized. This IIR CMM filter was designed with a CMM variation of the recursive generalized multiple exchange algorithm. It corresponds to the top endpoint of the PCLS tradeoff curve in Fig. 5(c).

In [1] and [2], we proved that PCLS tradeoff curves must monotonically decrease and terminate with zero slope for symmetric FIR filters and windows. Although it corresponds to a set of IIR filters, the tradeoff curve in Fig. 5(c) seems to monotonically decrease and terminate with zero slope. The similarity between FIR and IIR tradeoff curves is more than a coincidence, as indicated by the following theorem.

PCLS Tradeoff Theorem (PTT): ε versus δ tradeoff curves for all types of optimal PCLS filters must monotonically decrease and terminate with zero slope.

PTT Proof: The feasible set for $\delta = \delta_a$ must be a subset of the feasible set for $\delta = \delta_b$ if $\delta_a \leq \delta_b$. Therefore, $\varepsilon(\delta_b) \leq \varepsilon(\delta_a)$ if $\delta_a \leq \delta_b$, and both solutions are optimal. This proves that ε must be a monotonically decreasing function of δ for optimal PCLS filters. The slope at the LS solution must be zero because all δ-inequality constraints are inactive, and their KT multipliers vanish.

Although the PTT proof is very simple, the PTT is a very general and useful theorem. The PTT is true for all types of optimal PCLS filters, including IIR digital filters, asymmetric FIR filters, complex FIR filters, and analog filters. Moreover, it is applicable to filters with very complicated nonlinear inequality constraints. For example, the tradeoff between the stopband energy and the peak stopband gain in optimal PCLS filters must satisfy the PTT, regardless of the passband constraints, such as inequality constraints on the passband magnitude, phase, and delay. The PTT is not restricted to analyzing the tradeoffs between the stopband



(a)

(b)

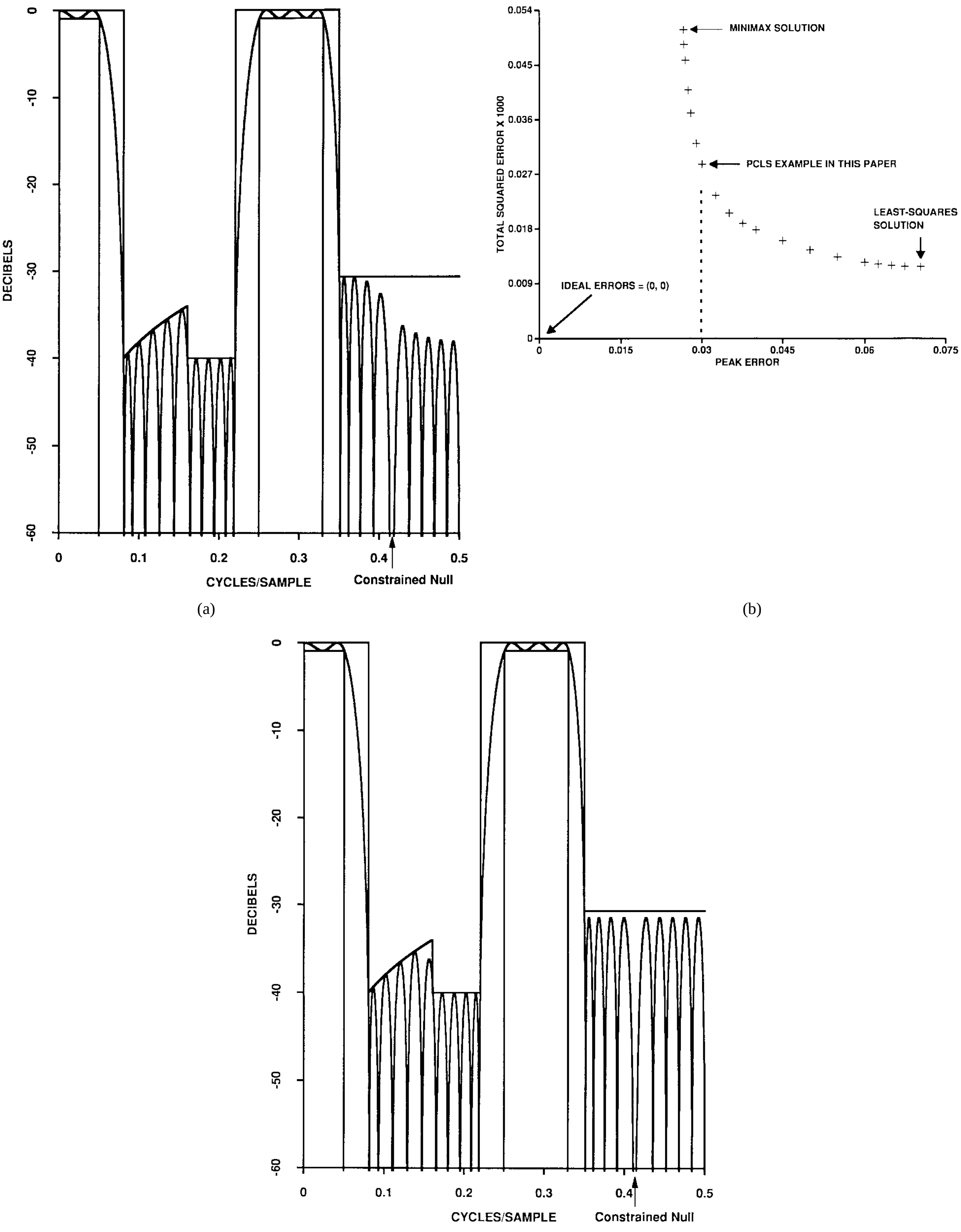
(c)

Fig. 3. Asymmetric FIR filter with constrained group delay. (a) Frequency response magnitude. (b) Passband group delay. (c) Passband magnitude.

energies and the peak stopband gains in filters. It can also be used for analyzing tradeoffs between other types of peak errors and total squared errors.

The PTT can be used as one-way optimality test. (It states a necessary but not a sufficient condition for optimality.) A set of filters is definitely not a set of optimal PCLS filters if its tradeoff curve has “bumps,” indicating that it is not monotonically decreasing.

The tradeoff curve in Fig. 5(c) was obtained by systematically designing a large number of PCLS IIR digital filters.



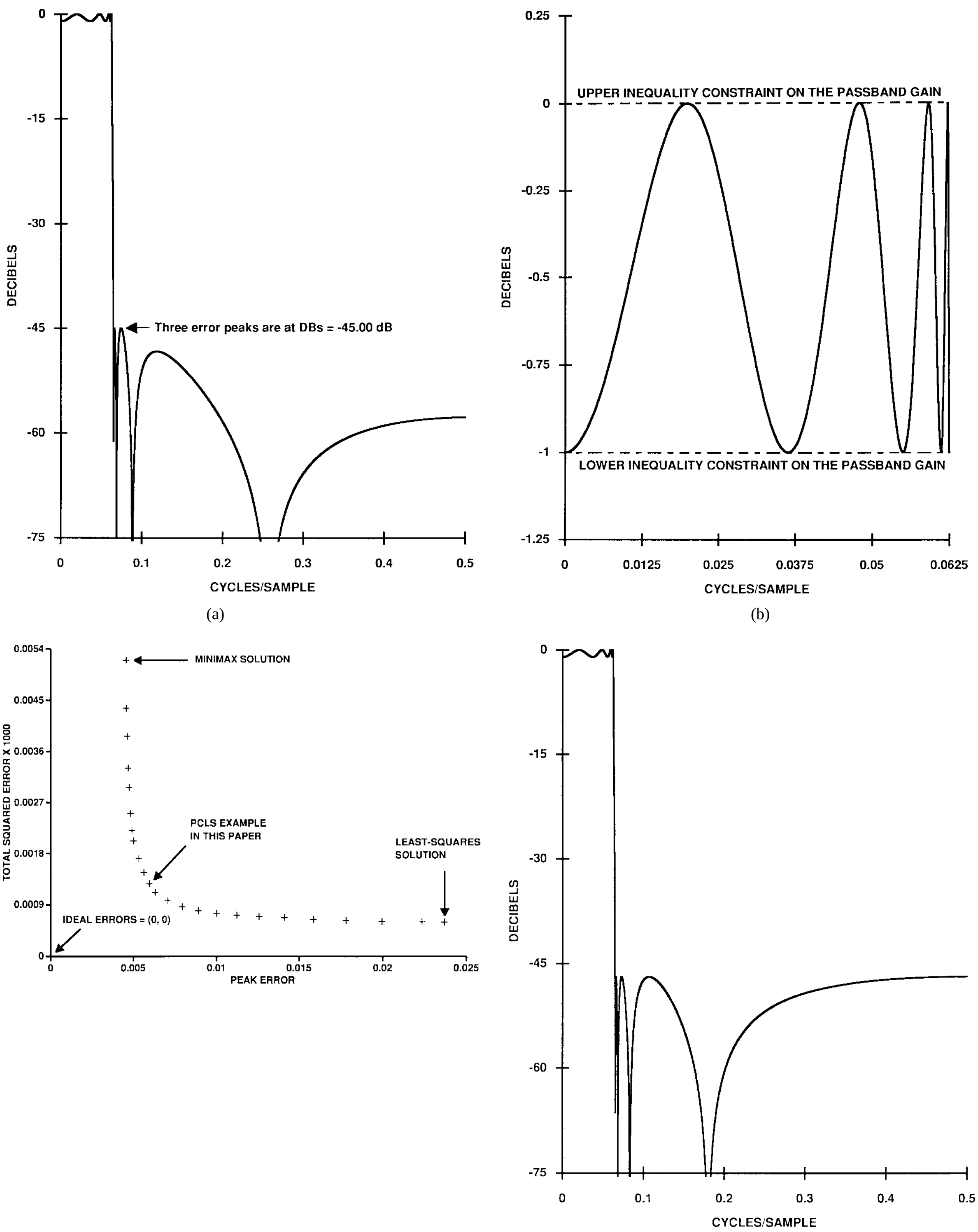
(a)

(b)

(c)

Fig. 4. Multiband FIR filter. (a) Frequency response for the PCLS filter with $\delta = 0.03$. (b) Tradeoff between the total squared error and the peak error in the upper stopband. (c) Minimax filter.

(We have developed algorithms for efficiently automating the generation of tradeoff curves, and we plan to present them in future papers.) All of the filters corresponding to Fig. 5(c) have nonnegative Kuhn–Tucker (KT) multipliers, which is a necessary condition for optimality. Unfortunately, these filters cannot be proven to be globally optimal because their objective functions are nonlinear. However, we believe that they are globally optimal for reasons discussed in [26]. Our belief is



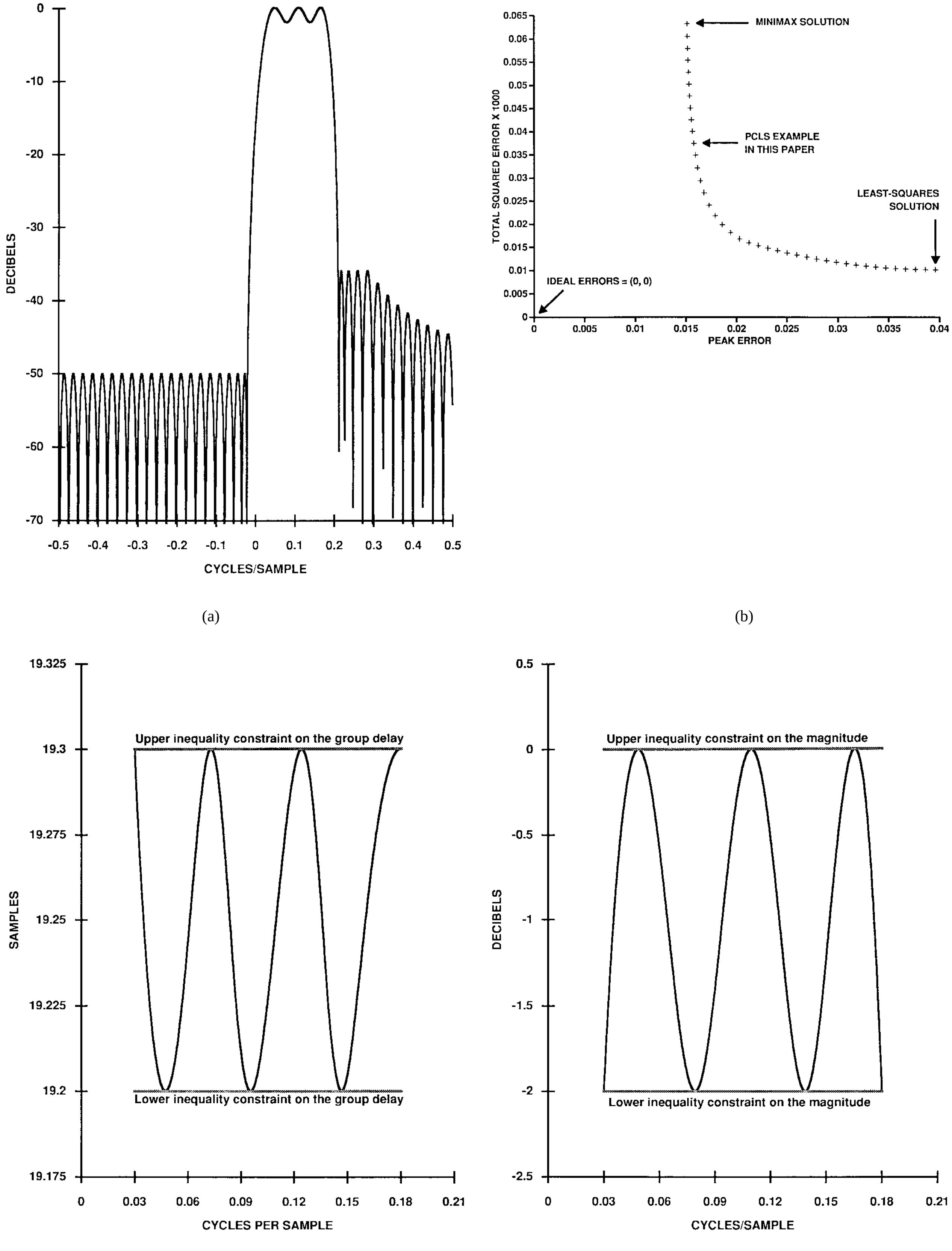
(a) (b)

(c) (d)

Fig. 5. IIR digital filter. (a) PCLS frequency response. (b) PCLS passband details. (c) Tradeoff between the total squared error and the peak error in the stopband. (d) Minimax filter.

strengthened by the fact that the tradeoff curve in Fig. 5(c) satisfies the PTT.

We discuss PCLS IIR digital filters that meet simultaneous specifications on the frequency response magnitude and group delay in [25] and [26]. In particular, we consider Example 1 in Deczky's classic IIR digital filter paper [22]. The same example also appears in the popular textbook by Oppenheim and Schafer [23, pp. 442–443]. In addition, the same example



(a)

(b)

(c)

(d)

Fig. 6. Complex FIR filter. (a) Frequency response magnitude of the complex PCLS filter with $DB_{us} = -36.0$ dB and constant group delay of 19.5 samples. (b) Tradeoff between the total squared error and the peak error in the upper stopband. (c) Passband group delay of the complex PCLS filter with $DB_{us} = -36.0$ dB and passband group delay constrained between 19.2 and 19.3 samples. (d) Passband magnitude for the filter in Fig. 6(c).

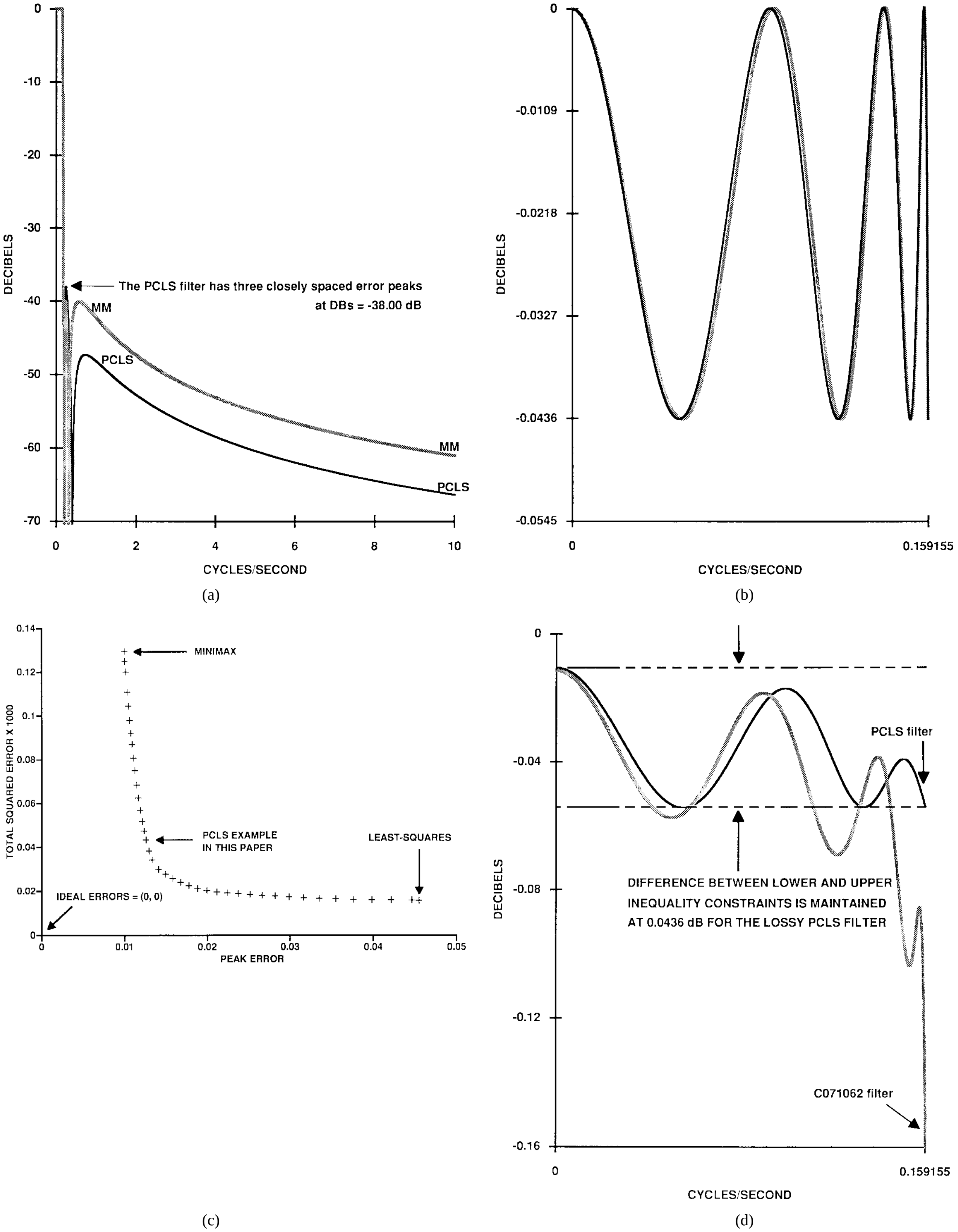


Fig. 7. Analog filter. Solid curve: PCLS filter. Shaded curve: Minimax filter C071 062. (a) Frequency response. (b) Passband details. (c) Tradeoff between the total squared error and the peak error in the stopband. (d) Passband details for the PCLS filter and C071 062 filter when the inductors have $Q_L = 1000$.

appears in the recent *Handbook for Digital Signal Processing* [24]. Simultaneous PCLS optimization of the frequency response magnitude and group delay provides a dramatic improvement in the solution of this classic IIR filter design problem. Using the same number of quadratic sections and the same specifications for the frequency response magnitude as in [22]–[24], we reduce the group delay ripple by a factor of 35 in [25] and [26]. In [26], the simultaneous optimization of the frequency response magnitude, phase, phase delay, and group delay is also discussed.

For [22, Example 3], PCLS optimization can reduce the group delay ripple to only ± 0.002 samples (giving an improvement factor of 40) at the same time the stopband energy is reduced by 6 dB, without sacrificing any performance measure. More details are in [26].

E. Complex FIR Digital Filter

We now consider an FIR filter with complex impulse response. The objective is to minimize the total squared error in the upper stopband from 0.21 to 0.5 cycles/sample, subject to the peak gain specification of DB_{us} dB in the upper stopband and subject to the following inequality constraints in the lower stopband:

$$\begin{gathered}20\log_{10}|H(e^{j2\pi f})| \leq -50.0 \text{ dB}\\ \text{for } -0.5 \leq f \leq -0.02 \text{ cycles/sample}\end{gathered}$$

and subject to the following inequality constraints in the passband:

$$\begin{gathered}-2.0 \text{ dB} \leq 20\log_{10}|H(e^{j2\pi f})| \leq 0.0 \text{ dB}\\ \text{for } 0.03 \leq f \leq 0.18 \text{ cycles/sample}\end{gathered}$$

and the passband group delay is required to be 19.5 samples.

Fig. 6(a) shows the unique optimal solution to this design problem for the case, where DB_{us} is specified to be −36.0 dB. Unlike other plots in this paper, the plot in Fig. 6(a) spans the digital frequency band from −0.5 to +0.5 cycles/sample. The frequency response magnitude is not symmetric about dc because the impulse response is complex valued. Fig. 6(b) shows the tradeoff between the total squared error and the peak error in the upper stopband as the DB_{us} specification is varied.

If we modify the specifications for the filter shown in Fig. 6(a) to have a desired group delay of 19.25 samples with lower and upper inequality constraints of 19.2 and 19.3 samples, we obtain the passband group delay plot in Fig. 6(c). The corresponding passband magnitude is plotted in Fig. 6(d). The stopbands for this complex filter with passband group delay constrained between 19.2 and 19.3 samples are virtually identical to the stopbands for the filter with 19.5 sample delay shown in Fig. 6(a). The lower stopband is equiripple at −50.0 dB, and the upper stopband has four sidelobe peaks that touch the DB_{us} specification of −36.0 dB. This paper does not include a separate plot for the overall frequency response magnitude of this filter because it is virtually identical to the plot in Fig. 6(a).

We present details for simultaneous PCLS optimization of the frequency response magnitude, phase, phase delay, and group delay in asymmetric and complex FIR filters in [19].

F. Analog Filter

We include an analog filter example to show that PCLS optimization is not restricted to digital filters. As the basis for this example, we consider the LC ladder filter used by Orchard *et al.* in [27]. The schematic is labeled Filter C071 062 in [27, Fig. 4]. Orchard *et al.* obtained the component values from a table in a filter handbook. C071 062 was the table address. C071 062 is a lowpass filter with an equiripple frequency response in both the passband and stopband. It has the following characteristics: $F_p = 0.159\,155$ Hz, $F_s = 0.180\,254$ Hz, $DB_p = 0.0436$ dB, and $DB_s = -40.1$ dB.

We designed a PCLS filter to match the C071 062 values for F_p, F_s, and DB_p, but we minimized the stopband energy subject to $DB_s \leq -38.00$ dB and subject to component inequality constraints to ensure nonnegative values for the ten capacitors and inductors comprising the filter. The resulting PCLS frequency response is shown in Fig. 7(a) and (b).

For purposes of comparison, the frequency response for the C071 062 filter is overlaid in Fig. 7(a) and (b). The PCLS filter is shown with a solid curve, and the C071 062 filter is shown with a shaded curve. The PCLS filter has slightly higher stopband sidelobes within a very narrow frequency range adjacent to the transition band, but its frequency response is much lower over the remainder of the enormously wide stopband. (Of course, the analog stopband extends to infinity.) When computing the stopband energy, we used numerical integration for moderate frequencies and an analytical asymptotic approximation for frequencies approaching infinity.

We used PCLS optimization to design the filters comprising the tradeoff curve in Fig. 7(c). These filters were designed to match the C071 062 values for F_p, F_s, and DB_p, but the stopband energy was minimized subject to a variety of DB_s specifications, and the components were constrained to be nonnegative. The minimax filter at the top endpoint of the tradeoff curve in Fig. 7(c) corresponds to the C071 062 filter studied by Orchard *et al.* in [27].

PCLS optimization has special advantages over the traditional minimax optimization methods for analog filters because the stopbands usually have infinite bandwidths. The stopband energy is especially important in an antialiasing filter ahead of an A/D converter because the energy of the aliased signals depends on the energy in the analog filter's infinitely wide stopband. It is difficult for us to understand why stopband energy is ignored in most of the analog filter literature. The emphasis has clearly been on the peak error as evidenced by the many analog filter papers based on elliptic and Chebyshev solutions.

Analog filter handbooks provide tables based on ideal components because the classical analog network synthesis methods are based on the assumption that components are ideal. Engineers often accept the degradation resulting from component values that were optimized under this false assumption.

PCLS optimization is especially powerful for dealing with lossy components such as inductors. Inductors have predictable wire resistance losses. The resistance is proportional to the inductance. It is easy to include predictable losses (and parasitics) in PCLS optimization problems.

Fig. 7(d) shows the passband gain for the C071 062 filter when the inductors have quality factors of $Q_L = 1000$. The passband variation DB_p degrades from 0.0436 dB to 0.16

dB when Q_L is changed from ∞ to 1000. The C071 062 filter, along with most filters in the analog literature, was optimized under the assumption that $Q_L = \infty$. Fig. 7(d) includes the passband gain for the PCLS filter that was optimized to maintain $DB_p = 0.0436$ dB and nonnegative component values when $Q_L = 1000$. (The DB_s specification was increased to -25.0 dB to obtain a feasible solution in the knee of the tradeoff curve for $Q_L = 1000$. The first three stopband peaks are at -25.0 dB, and the fourth peak is at -30.8 dB.) In this example, PCLS optimization permitted us to control the effects of lossy components, rather than accepting the degradation resulting from an optimization based on a false assumption.

We have found that letting load resistances be variables in PCLS optimization of analog filters permits very lossy components to be optimized to meet surprisingly stringent frequency response specifications. We plan to present more details for PCLS optimization of nonideal components in future papers.

IV. Generalized Multiple Exchanges

In order to efficiently implement multiple exchanges, it is important to divide the inequality constraints into two categories: smooth and nonsmooth. In filter design problems, inequality constraints that vary smoothly inside of each frequency band are in the smooth category. On the other hand, inequality constraints at the edges of frequency bands are in the nonsmooth category. Conventional quadratic programming algorithms do not distinguish between smooth and nonsmooth inequality constraints. They are much less efficient than the generalized multiple exchange (GME) algorithm.

The GME algorithm includes the following parameters: $NDF, INCR$, and NKT. NDF denotes the number of degrees of freedom. $INCR$ denotes the limit on the increase in the number of constraints in the active set S_A from one iteration to the next. ($INCR$ stands for "increase.") The original multiple exchange algorithm in [1] effectively had $INCR$ corresponding to infinity because it did not control the increase in the number of active constraints. $INCR$ can be set to any value from 1 to infinity, and the GME algorithm will converge to the same unique optimal solution. However, if the number of active constraints is permitted to suddenly increase by a large amount, the algorithm can waste time by dropping many constraints in Step 2. (If the algorithm is permitted to suddenly activate a large number of new constraints, this usually means that it has been sidetracked to a bad path and will waste time getting back to a good path.) We use the $INCR$ parameter to avoid this inefficiency. We recommend using *INCR*$= 4 + \text{Int}NDF/8\}$, where $\text{Int}\{x\}$ denotes the integer part of x.

The NKT parameter is used to control the number of constraints dropped in Step 2. (NKT stands for "negative Kuhn–Tucker multipliers.") The GME algorithm will converge to the same optimal solution if NKT is set to any number from 1 to infinity. However, if it drops many constraints in Step 2, then it will usually find that the Step 2 error energy test eventually fails and it will be forced to go to Step 3 after wasting time with many drops. We use the *NKT* parameter to avoid this inefficiency. We recommend using $NKT = 4 + \text{Int}\{NDF/16\}$.

The GME algorithm obtains its initial guess in Step 0. It performs multiple exchanges in Step 1, and it performs single exchanges in Steps 2 and 3. These steps are described in the following paragraphs.

Step 0a) Use the method of Lagrange multipliers to minimize ε subject to the set of equality constraints S_E. If the solution is 0, then go to Step 0b. Otherwise, initialize $S_A(0)$ to the null set, N_A to 0, and k to 0. Test for optimality using the KT conditions. Terminate if the solution is optimal. Else compute $\varepsilon(0)$ and go to Step 1.

Step 0b) Select any inequality constraint that yields a nonzero solution, put it into $S_A(0)$, and use the method of Lagrange multipliers to minimize ε subject to S_E and $S_A(0)$. (As an example of selecting a constraint corresponding to a nonzero solution in a filter design problem, we can select a passband edge frequency and activate the constraint corresponding to the minimum passband gain specification. This allows us to get a nontrivial solution, even when the passband squared-error weighting is zero.) Set k to 0, and set N_A to 1. Test for optimality using the KT conditions. Terminate if the solution is optimal. Otherwise, compute $\varepsilon(0)$ and go to Step 1.

Step 1) Let S_{AN} denote the subset of $S_A(k)$ that corresponds to nonsmooth inequality constraints. Let S_{AS} denote the subset of $S_A(k)$ that corresponds to smooth inequality constraints at local error extrema. Define $S_{AC} = S_{AN} \cup S_{AS}$. Let NAC denote the number of constraints in S_{AC}. Let $S_{TV} = (S_{VN} \cup S_{VS}) - S_{AC}$. S_{VN} denotes the nonsmooth inequality constraints that violate the specifications. S_{VS} denotes the smooth inequality constraints at local error extrema that violate the specifications. At first, it may seem unnecessary to exclude S_{AC} from S_{TV} because any constraint that was active in the previous iteration should now be satisfied with exact equality and, theoretically, should not violate the specifications. However, due to machine rounding errors, we may encounter small violations of constraints that were active in the previous iteration. It is useful to separately keep track of the constraints in S_{TV} and S_{AC} because the constraints in S_{AC} were previously active, and the computations associated with them do not need to be recalculated in the next iteration.

Let NTV denote the number of constraints in S_{TV}, and let $NT = NTV + NAC$. If $NT > NDF - 1$ or if $k > 1$ and $NT > NA + INCR$, then go to Step 3. Otherwise, define $S_T = S_{TV} \cup S_{AC}$. Use the method of Lagrange multipliers to minimize ε subject to the constraints in S_E and S_T and obtain ε_T. Terminate if the solution is optimal. If any KT multiplier is negative, then set IKT to 1 and go to Step 2. Otherwise, if $\varepsilon_T > \varepsilon(k)$ and all KT multipliers are nonnegative, then let $k \leftarrow k+1, S_A(k) \leftarrow S_T, NA \leftarrow NT, \varepsilon(k) \leftarrow \varepsilon_T$, and repeat Step 1. Otherwise, if $\varepsilon_T \leq \varepsilon(k)$, go to Step 3.

Step 2) If $IKT > NKT$, then go to Step 3. Otherwise, drop the constraint with the most negative KT multiplier from S_T and increment $IKT \leftarrow IKT + 1$. Use the method of Lagrange multipliers to minimize ε subject to the constraints in S_E and S_T, and obtain ε_T. Terminate if the solution is optimal. Otherwise, if $\varepsilon_T > \varepsilon(k)$ and all KT multipliers are nonnegative, then let $k \leftarrow k+1, S_A(k) \leftarrow S_T, NA \leftarrow NT, \varepsilon(k) \leftarrow \varepsilon_T$, and go to Step 1. If $\varepsilon_T > \varepsilon(k)$ and any KT multiplier is negative and $IKT < NKT$, then repeat Step 2. If $\varepsilon_T \leq \varepsilon(k)$ or if any KT multiplier is negative and $IKT > NKT$, then go to Step 3.

Step 3) Starting from $S_A(k)$, perform an iteration based on the Goldfarb–Idnani algorithm, and obtain the corresponding constraints S_{GI} and error energy ε_{GI}. (For improved efficiency, our implementation of the Goldfarb–Idnani algorithm in Step 3 exploits the special structures of the vectors and matrices in filter design problems instead of using the numerical implementation in [20]. It is also customized to exploit the calculations in Steps 1 and 2. We plan to discuss the numerical details in a future publication.) If the solution is optimal, then terminate. If the problem is infeasible, then notify the user and terminate. (We plan to discuss other options for infeasible problems in future papers.) Otherwise, let $k \leftarrow k+1, S_A(k) \leftarrow S_{GI}, \varepsilon(k) \leftarrow \varepsilon_{GI}$, and go to Step 1.

The GME algorithm must converge to the unique optimal solution of any feasible positive-definite quadratic programming problem within a finite number of iterations. (Error energies are always positive definite in filter design problems.) The number of iterations is finite because each iteration terminates with a different set of active constraints and because the number of different active constraint sets is finite. (We can use a finite number of frequency grid points in filter problems.) Each iteration terminates with a different set of active constraints because the objective function increases monotonically. The final solution must be optimal because the algorithm cannot stop until the KT conditions are satisfied.

The ε tests may be replaced by tests for violations of previously active constraints as discussed in [6] and [7]. This approach may increase the number of iterations, but it can reduce the number of computations within each iteration. (Fewer computations are usually required to check for violations of previously active constraints than to check ε.) Both approaches are guaranteed to converge to the unique optimal solution when used in the GME algorithm.

The GME algorithm is inherently very fast. Step 1 is very efficient because it uses multiple exchanges. Steps 2 and 3 usually reduce the rate of convergence for the iterations where they are used, but they are not used unless they are needed to guarantee convergence to the optimal solution. Even in difficult design problems, the GME algorithm usually does Step 1 multiple exchanges in most iterations.

The GME algorithm is especially fast for symmetric FIR filter design problems because we can exploit the special structures of $\boldsymbol{R}, \boldsymbol{C}$, and $\boldsymbol{p}$, where

$\boldsymbol{R}$ Hessian matrix;
$\boldsymbol{C}$ working constraint matrix;
$\boldsymbol{p}$ linear component of the objective.

For example, we discussed a fast method for solving FIR filter equations in [21]. We are now working on improvements to it along with fast Toeplitz-plus-Hankel methods to include in the GME algorithm. In addition to permitting us to use fast matrix-vector methods for solving equations in the GME algorithm, symmetric FIR filters also permit us to use fast discrete cosine transform (FDCT) methods for computing the zero-phase response.

For the sake of numerical efficiency in the GME algorithm, it is important to compute static quantities only once and reuse them in later iterations. For example, FDCT coefficients (twiddle factors) should only be computed once because the grid frequencies are the same in every iteration. Moreover, the $\boldsymbol{R}$ matrix and $\boldsymbol{p}$ vector are static, and computations should be performed on them only one time. The equality constraints are also static, and their intermediate solution vectors should be saved.

Some dynamic quantities should be saved until the next iteration in case they can be reused. For example, the intermediate solution vectors for the active constraint equations should be saved until the next iteration because some of the active constraints may remain active. In addition, the intermediate computations for solving the KT-multiplier system of equations should be saved until the next iteration because updating methods can be used. Of course, brute-force methods such as matrix inversion should not be used.

We have observed that the GME algorithm has three basic phases of iterations. In the early (Phase I) iterations, most of the smooth active constraints have significant frequency changes from one iteration to the next. In the middle (Phase II) iterations, only a few of the smooth active constraints have significant frequency changes from one iteration to the next. In the final (Phase III) iterations close to convergence, none of the smooth active constraints have significant frequency changes from one iteration to the next.

We have found that the overall computational cost can be reduced by preventing minor frequency changes for smooth active constraints during iterations where there are major frequency changes for smooth active constraints. This modification of the GME algorithm tends to be activated during Phase II iterations. It permits more computations to be reused at the expense of a small increase in the number of iterations. The tradeoff between the number of recycled computations and the number of iterations can be controlled by adjusting the thresholds for minor and major frequency changes. We are continuing to explore modifications of the GME algorithm to trade off the amount of work inside each iteration with the number of iterations. The goal is to find the best balance to minimize the total computational cost.

The GME algorithm was used directly to design the symmetric FIR digital filter examples in this paper. The GME algorithm was used as a subroutine inside of recursive quadratic programming (RQP) loops to design the other examples. We use RQP to convert nonlinear programming problems (for example, group-delay inequality constraints are highly nonlinear functions of filter coefficients) into chains of quadratic programming problems that we solve with the GME algorithm. The overall algorithm is called the recursive generalized multiple exchange (RGME) for lack of a better name.

The RGME algorithm cannot be guaranteed to converge to the optimal solutions for all problems in the very broad class of nonlinear programming problems. However, we have observed it to be very robust for practical engineering applications. Details about its convergence properties are included in [26].

We are developing computer programs that optimize the component values in digital and analog schematics with arbitrary topologies. They are based on the component optimization using the recursive generalized multiple exchange (CORGME) algorithm. The programs use nodal analysis and read disk files containing branch descriptions in the format From Node, To Node, Component Type, Initial Component Value, Lower Limit on the Component Value, Upper Limit on the Component Value. We usually set the lower limits on the component values to zero in analog filters. The component values in the input file are used as the initial guess for the CORGME optimization. The output file has the same format as the input file, except Initial Component Value is replaced by Optimized Component Value. The CORGME algorithm is very useful for optimizing circuits that were derived from human intuition and approximations. It is also useful for optimizing "recipe book" circuits obtained from handbooks.

If the initial guess is very far from meeting the specifications, then we usually need to use the CORGME algorithm in multiple stages. We relax the specifications so that they are almost met by the initial guess, and then we tighten the inequality constraints in each stage as we nudge the design toward the solution that meets the original specifications. This multistage strategy is useful for preventing the occurrence of infeasible intermediate design problems.

The multistage approach is usually needed when making large changes to many specifications. For example, if we start with an analog handbook filter as an initial guess and we wish to significantly change the band edge frequencies and the ripple sizes, then we usually need multiple optimization stages. In addition to permitting large changes to handbook designs, the multistage approach permits CORGME to start from rough and intuitive (human) initial guesses, even when designing high-order filters.

The CORGME algorithm converges very rapidly because it uses multiple exchanges. The quick turnaround significantly reduces the importance of finding accurate initial guesses because many initial guesses can be optimized in a short time.

PCLS optimization with the CORGME algorithm can liberate the designer from many of the intricate details of classical analog network synthesis theory. It is ideal for designing analog circuits because it can easily include inequality constraints on the component values to ensure that they are nonnegative and realizable. Moreover, it can perform simultaneous PCLS optimization in the time and frequency domains. For example, the overshoot of the step response in the time domain can be constrained while the gain and phase are optimized in the frequency domain.

PCLS optimization is not restricted to one-dimensional (1-D) responses. In [15], two-dimensional (2-D) PCLS filters and windows are discussed, and several design examples are included. We plan to publish the work in [15] along with numerical refinements for the 2-D implementation of the GME algorithm in a future journal paper. (The publication of the work based on [15] was disrupted by the infamous 1994 Northridge earthquake and two deaths in the family of J. W. Adams.) We presented brief discussions of 2-D PCLS optimization in conference papers [16], [18]. A 2-D PCLS example was included in [16] and [18]. The example showed a 2-D PCLS window with a mainlobe peak constrained to be equal to 0.0 dB while the sidelobes were constrained to be less than or equal to -34.0 dB across the 2-D frequency plane. The first three error peaks along each frequency axis turned out to be exactly -34.0 dB, and the remaining sidelobes decayed to minimize the total sidelobe energy in the 2-D frequency plane.

Conventional multidimensional (M-D) minimax design algorithms are notoriously slow. Moreover, the alternation theorem cannot test the optimality of M-D MM solutions. On the other hand, it is easy to use the KT theorem to test the optimality of M-D PCLS solutions. The GME algorithm is ideal for solving M-D problems. We note that steps 0–3, which define the GME algorithm, do not depend on the number of dimensions. The primary differences between the 1-D and M-D implementations are in the details of formulating ε and the constraint equations. For example, in the 2-D implementation of the GME algorithm for FIR filters and windows, we formulate ε using a 2-D integral instead of a 1-D integral, and we define the inequality constraint equations on a 2-D frequency grid instead of a 1-D grid. The GME algorithm is guaranteed to converge to the unique optimal solution regardless of the dimensionality of the problem.

A 2-D PCLS filter design algorithm was proposed in [14]. However, the algorithm proposed in [14] is not guaranteed to converge, and it is only a single exchange algorithm. Single exchange algorithms are very slow in 2-D problems.

V. Conclusion

In this paper, we extended our previous discussions of PCLS optimization. We also presented details for the GME algorithm, and we introduced the RGME and CORGME algorithms. The GME and RGME algorithms solve quadratic and nonlinear programming problems, respectively. The CORGME algorithm is a schematic-based version of the RGME algorithm. Unlike the conventional algorithms in the optimization literature, the algorithms in this paper use both single and multiple exchanges.

We have developed another algorithm that can be used for PCLS and RBLS design problems, but we did not discuss it in the main body of this paper because it is generally not as powerful as the GME algorithm. It is called the ripple-weighted least-squares (RWLS) algorithm in [31].

Unfortunately, the RWLS algorithm converges slowly when designing PCLS filters with specified band edge frequencies. It converges more rapidly for RBLS filters with unspecified band edge frequencies, but as previously discussed, we believe that RBLS filters are not appropriate for most applications.

(In our early experiments with the RWLS algorithm, we used it initially to design RBLS filters because it had the most difficulty controlling the gains at band edge frequencies.) Another problem with the RWLS algorithm is that it is not guaranteed to converge.

On the other hand, the RWLS algorithm has an important advantage compared with the GME algorithm. It easily accommodates nonlinear inequality constraints on the error magnitude. (We must use the GME algorithm recursively in order to accommodate nonlinear constraints.) The RWLS algorithm has not yet been fully developed, but we believe that it may be useful for designs with problematic nonlinear constraints. It may also be useful in combination with the RGME algorithm. For example, we may use the RWLS algorithm to compute a RBLS solution as the initial guess to a nonlinear programming problem and then use the RGME algorithm to obtain the final PCLS solution. This is an area for future research.

We have published over 25 papers on PCLS optimization and supervised over 30 graduate student theses on this topic, but much work remains to be done. We believe that many researchers will become interested in PCLS optimization because there are many practical applications for it.

This paper discussed only a few of the many ways that PCLS design problems can be formulated. We believe that the PCLS optimality criterion can be customized to fit virtually any practical application. For example, PCLS optimization can readily be used to design equalizers in the prefilter-equalizer or interpolated FIR (IFIR) implementations of filters.

Our previous PCLS research emphasized analog and digital filter design problems. However, PCLS optimization is certainly not restricted to filters. For example, PCLS optimization could be used to solve many classical problems in numerical analysis, such as curve fitting, where textbooks now recommend minimax and least-squares optimization. It could also be used in the field of antenna design to minimize total sidelobe energy subject to peak-sidelobe inequality constraints and directional-null equality constraints. There are also many applications for PCLS optimization in other fields.

ACKNOWLEDGMENT

The authorswish to thank the reviewers for their comments and suggestions.

REFERENCES

[1] J. W. Adams, "FIR digital filters with least-squares stopbands subject to peak-gain constraints," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 376–388, Apr. 1991.

[2] ——, "A new optimal window," *IEEE Trans. Signal Processing*, vol. 38, pp. 1753–1769, Aug. 1991.

[3] ——, "A new window for spectral analysis with comparisons to conventional windows," *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1989, pp. 638–642.

[4] ——, "Quadratic programming approach to new problems in digital filter design," *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1990, pp. 307–310.

[5] ——, "Quadratic programming approach to new optimal windows and antennas," in *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1990, pp. 69–72.

[6] J. W. Adams, E. A. Gauer, J. L. Sullivan, and B. Tucker, "Digital filters with peak-constrained weighted least-squared errors," *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1992, pp. 256–260.

[7] J. W. Adams, J. Sullivan, R. Hashemi, and C. Ghadimi, "New approaches to constrained optimization of digital filters," *Proc. IEEE Int. Symp. Circuits Syst.*, May 1993, pp. 80–83.

[8] J. L. Sullivan and J. W. Adams, "A new nonlinear optimization algorithm for asymmetric FIR digital filters," *Proc. IEEE Int. Symp. Circuits Syst.*, London, U.K., May 1994, pp. 541–544.

[9] M. Lang and J. Bamberger, "Nonlinear phase FIR filter design according to the l2 norm with constraints for the complex error," *Signal Process.*, vol. 36, pp. 259–268, July 1994.

[10] V. E. DeBrunner, "The design of low sensitivity digital filters using multi-criterion optimization strategies," *Proc. ICASSP*, Mar. 1992, vol. IV, pp. 317–320.

[11] C. S. Burrus and J. Barreto, "Least p-power error design of FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, San Diego, CA, May 1992, pp. 545–548.

[12] J. W. Adams, "Constrained least-squares digital filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, San Diego, CA, May 1992, pp. 565–568.

[13] I. W. Selesnick, M. Lang, and C. S. Burrus, "Constrained least square design of FIR filters without specified transition bands," in*Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 1995, vol. 2, pp. 1260–1263.

[14] M. Lang, I. W. Selesnick, and C. S. Burrus, "Constrained least-squares design of 2-D FIR filters," *IEEE Trans. Signal Processing*, vol. 44, pp. 1234–1241, May 1996.

[15] R. Hashemi, *Two-Dimensional PCLS FIR Filter and Window Design*, Master's thesis, Calif. State Univ., Northridge, CA, pp. 1–158, Dec. 1994.

[16] J. W. Adams, P. Kruethong, R. Hashemi, J. L. Sullivan, and D. Gleeson, "New quadratic programming algorithms for designing FIR digital filters," in *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1993, pp. 1206–1210.

[17] *Signal Processing Toolbox User's Guide*. Natick, MA: The Math Works, Dec. 1996.

[18] J. W. Adams, J. L. Sullivan, D. R. Gleeson, P. H. Chang, and R. Hashemi, "Application of quadratic programming to FIR digital filter design problems," in *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1994, pp. 314–319.

[19] J. Sullivan and J. W. Adams, "PCLS optimization of asymmetric and complex FIR digital filters and windows," submitted fofr publication.

[20] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Math. Programming*, vol. 27, pp. 1–33, 1983.

[21] J. L. Sullivan and J. W. Adams, "An algorithm for solving the toeplitz systems of equations in FIR digital filter design problems," *IEEE Trans. Signal Processing*, vol. 45, pp. 992–993, Apr. 1996.

[22] A. G. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 257–263, Oct. 1972.

[23] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[24] W. E. Higgins and D. C. Munson, "Infinite impulse response digital filter design," in *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser, Eds. New York: Wiley, 1993, ch. 5.

[25] J. L. Sullivan and J. W. Adams, "Peak-constrained least-squares IIR digital filters using recursive quadratic programming," in *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1995, pp. 980–984.

[26] ——, "PCLS IIR digital filters with simultaneous frequency response magnitude and group delay specifications," submitted for publication.

[27] H. J. Orchard, G. C. Temes, and T. Cataltepe, "Sensitivity formulas for terminated lossless two-ports," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 459–466, May 1985.

[28] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1989.

[29] J. W. Adams, J. Nelson, and J. Moncada, "FIR digital filter design with multiple optimality criteria and constraints," in *Proc. IEEE Int. Conf. Circuits Syst.*, Apr. 1989, pp. 343–346.

[30] J. W. Adams, R. W. Sucher, and J. L. Sullivan, "Optimal weights for autocorrelation sequences," *Proc. IEEE Conf. Signals, Syst., Comput.*, Oct. 1993, pp. 1255–1259.

[31] J. W. Adams, C. Ghadimi, B. Tucker, Q. Gao, and J. Sullivan, "Ripple-weighted approach to designing peak-constrained least-squares digital filters," in *Proc. IEEE Conf. Signals, Syst., Computers*, Oct. 1993, pp. 692–696.

John W. Adams (S'76–M'81–SM'87) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Los Angeles, in 1976, 1976, and 1982, respectively.

Since 1984, he has been a Professor in the Electrical and Computer Engineering Department, California State University, Northridge. (The university is located one half mile from the epicenter of the 1994 Northridge earthquake.) Since December 1997, he has also been doing part-time work for the Sensors and Communications Segment of the Raytheon Systems Company. His main research interests are in the areas of analog and digital signal processing, image processing, radar, super-resolution spectral analysis, optimization, antenna design, and computer-aided circuit design.

Dr. Adams served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1993 to 1997. He received the 1996 IEEE Fred Nathanson Memorial Radar Education Award and the 1995 Fred Nathanson Memorial Radar Engineer of the Year Award from the IEEE Aerospace and Electronic Systems Society. He received the 1988 Trustees Outstanding Professor Award for the Northridge campus of the California State University system "for excellence in teaching." He was awarded the 1987 Texas Instruments Founders Prize, including an honorarium of $50 000, "for the individual exhibiting the most outstanding record of achievement and potential for future achievement in the physical sciences, health sciences, management sciences, engineering, or mathematics." He was co-recipient of the 1985 IEEE W.R.G. Baker Prize "for the most outstanding paper presenting original work in all of the journals, transactions, magazines, and proceedings of the IEEE." He is a member of Tau Beta Pi and Phi Beta Kappa.

James L. Sullivan was born in the state of Washington on February 3, 1942. He received the B.S. degree from the University of Washington, Seattle, in 1965 and the M.S. degree in electrical engineering (with honors) from California State University, Northridge, in 1993. From 1966 to 1995, he was employed at the Pacific Missile Test Center, Point Mugu, CA, working on missile ground support systems and missile-borne telemetry design. He is presently working for Allied Technical Services Corporation, Pasadena, CA, developing and maintaining the Deep Space Network for JPL/NASA. He has published 21 conference and journal papers. His main research interests are in the areas of digital and analog filter design, digital controls design, and optimization.