MIT Open Access Articles

Linear Programming Algorithms for Sparse Filter Design

MIT
Massachusetts Institute of Technology

DSpace@MIT

Linear Programming Algorithms for Sparse Filter Design

Thomas Baran, *Member, IEEE*, Dennis Wei, *Member, IEEE*, and Alan V. Oppenheim, *Life Fellow, IEEE*

Abstract—In designing discrete-time filters, the length of the impulse response is often used as an indication of computational cost. In systems where the complexity is dominated by arithmetic operations, the number of nonzero coefficients in the impulse response may be a more appropriate metric to consider instead, and computational savings are realized by omitting arithmetic operations associated with zero-valued coefficients. This metric is particularly relevant to the design of sensor arrays, where a set of array weights with many zero-valued entries allows for the elimination of physical array elements, resulting in a reduction of data acquisition and communication costs. However, designing a filter with the fewest number of nonzero coefficients subject to a set of frequency-domain constraints is a computationally difficult optimization problem. This paper describes several approximate polynomial-time algorithms that use linear programming to design filters having a small number of nonzero coefficients, i.e., filters that are sparse. Specifically, we present two approaches that have different computational complexities in terms of the number of required linear programs. The first technique iteratively thins the impulse response of a non-sparse filter until frequency-domain constraints are violated. The second minimizes the 1-norm of the impulse response of the filter, using the resulting design to determine the coefficients that are constrained to zero in a subsequent re-optimization stage. The algorithms are evaluated within the contexts of array design and acoustic equalization.

Index Terms—Sparse filters, linear programming, FIR digital filters, linear arrays.

I. INTRODUCTION

IN THE efficient implementation of discrete-time filters, the focus has historically been on reducing the number of required arithmetic operations, and especially the number of multiplications. Some approaches eliminate multipliers altogether by restricting coefficient values to be binary [1] or composed of a limited number of power-of-two terms [2]–[5], thus permitting realizations requiring a relatively small number of shifters and adders. Among implementations that include multipliers, there exist a number of efficient techniques based on cascade structures. Examples include the use of recursive running-sum prefilters [6], [7], cyclotomic polynomial prefilters [8], a prefilter-

equalizer approach based on piecewise exponentials [9], interpolated FIR filters [10], [11], the frequency-response masking technique [12], [13], and multirate techniques [14]–[16]. For filters used in rate-conversion systems, efficient structures have been developed to perform the majority of the computation at a lower rate, e.g., polyphase structures and generalizations thereof [17], and multistage decompositions [18], which are also closely related to interpolated FIR filters and multirate filters.

In this paper, we focus on the design of sparse FIR filters, i.e., filters with relatively few nonzero impulse response coefficients, as a technique for reducing complexity. Sparse filters offer the opportunity to omit arithmetic operations corresponding to zero-valued coefficients. For instance, in an ASIC implementation, sparsity allows for the elimination or deactivation of circuit components, thereby reducing the area and power consumption. In addition, as demonstrated in [19], [20], sparse designs may be incorporated in cascade and multistage structures to yield even more efficient implementations.

Sparsity is also of interest in the design of linear sensor arrays, a problem that closely parallels the design of discrete-time FIR filters. In designing arrays, it is often desirable to achieve a given response pattern using as few array elements as possible. Indeed, sparse design methods may be especially relevant for arrays since methods based on cascading filters have no clear analogs.

The problem of determining a filter with maximal sparsity subject to a set of specifications is computationally difficult. The problem can be solved using integer programming methods such as branch-and-bound and more sophisticated techniques found in commercial software packages [19]–[21]. Generally speaking, these methods perform an intelligent search over all possible configurations of zero and nonzero coefficients, and hence are guaranteed in theory to obtain a maximally sparse filter. While much more efficient than an exhaustive search, they still exhibit non-polynomial complexity on average and can be quite computationally intensive for problems involving many coefficients.

The difficulty of optimal design has motivated the development of efficient approximate methods directed at obtaining reasonably sparse but not necessarily optimal designs with a reasonable amount of computation. Approximate methods are particularly useful for large problems or in applications in which the filter is frequently redesigned. A simple approach is to re-optimize a given non-sparse design after forcing some small-valued coefficients to zero [22], [23]. An alternative is to judiciously determine the subset of coefficients permitted to be nonzero, for example using orthogonal matching pursuit [24] or based on nth-band approximations to frequency-selective

filters with conveniently located cutoff frequencies [25].[1] An approximate, nonconvex measure of sparsity was proposed in [26], and an algorithm was developed to minimize it in a local sense.

In this paper, we present two types of approximate algorithms for sparse filter design. Both approaches make extensive use of linear programming, and the existence of efficient techniques for solving linear programs contributes significantly to their efficiency. In particular, the presented techniques use linear programming as a significant algorithmic component in determining both the subset of coefficients permitted to be nonzero as well as their values. Ongoing advancements in the execution times of linear program solvers may therefore be leveraged extensively by the presented techniques.

The first approach is based on successively thinning the impulse response of a predesigned filter and re-optimizing the remaining coefficients. Successive thinning algorithms require, in the worst case, the solution of a number of linear programming subproblems that is quadratic in the allowed order of the filter. However, since many of these linear programming subproblems are closely related, the solution to a given subproblem can be used to initialize other subproblems in a way that yields a dramatic speed increase over naive initialization.

The second approach involves minimizing the 1-norm of the impulse response, subject to a set of frequency-domain constraints, and using the result to indicate which of $M + 1$ coefficients to constrain to zero in a subsequent re-optimization stage. Specifically, the locations of the nonzero coefficients are chosen to be the locations of the J largest-magnitude coefficients in the solution to the 1-norm minimization. The algorithm is then iterated, varying J to determine the most-sparse design achievable. The overall design algorithm requires, at most, the solution of $1 + \lceil \log_2(M + 1) \rceil$ linear programs.

The organization of the paper is as follows. In Section II, the problem of sparse filter design is formulated in more detail and notation is established. In Section III, we develop algorithms based on the idea of successive thinning and discuss strategies for choosing the coefficients to be thinned. In Section IV, we outline the use of the 1-norm of the impulse response as an indication of the coefficients to be thinned, and we comment on a theoretical connection between the technique and the true sparse design problem. The performance of our algorithms is demonstrated through a number of design examples in Section V.

II. PROBLEM FORMULATION

For simplicity of presentation, we focus on the design of causal, Type I linear-phase FIR filters of length $N + 1$, i.e., filters with impulse responses $h[n]$ that are supported on an index set $n = 0, 1, \ldots, N$ and are even-symmetric about the integer index $M = N/2$. The methods to be described can be generalized with minor modifications to other types of linear-phase FIR filters. We regard N as a fixed parameter representing the maximum allowable number of delay elements, with the understanding that the final design may require fewer than N delays if coefficients at the ends of the impulse response are zero.

The impulse response values are parameterized by an $(M + 1)$-dimensional vector $\mathbf{b}$ with components $b_0, b_1, \ldots, b_M$ given by

$$b_0 = h[M],$$
$$b_n = 2h[M - n] = 2h[M + n], \quad n = 1, 2, \ldots, M. \quad (1)$$

The frequency response corresponding to (1) is given by

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega M}$$

where

$$A(e^{j\omega}) = \sum_{n=0}^{M} b_n \cos(n\omega) \quad (2)$$

is the real-valued amplitude response used to approximate a desired response $H_d(e^{j\omega})$. More precisely, we assume that $A(e^{j\omega})$ is chosen such that the maximum weighted error is no greater than a desired tolerance δ_d, i.e.,

$$W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \quad \forall \, \omega \in \mathcal{S} \quad (3)$$

where $W(\omega)$ is a strictly positive weighting function and $\mathcal{S}$ is a closed subset of $[-\pi, \pi]$.

We use the number of nonzero coefficients in $\mathbf{b}$ as a measure of computational complexity. In Sections III and IV, we make reference to the function $\|\mathbf{b}\|_p$ defined as

$$\|\mathbf{b}\|_p = \left(\sum_{n=0}^{M} |b_n|^p \right)^{1/p}$$

which, for $p \geq 1$, is known as the p-norm. It is also common to define

$$\|\mathbf{b}\|_0 = \lim_{p \to 0} \|\mathbf{b}\|_p^p$$

which corresponds exactly to the number of nonzero entries in $\mathbf{b}$. (The function $\|\mathbf{b}\|_0$ is often referred to as the 0-norm for convenience despite not being a true norm.) Thus, the problem of sparse filter design can be formulated as

$$\min_{b_0,\ldots,b_M} \|\mathbf{b}\|_0$$
$$\text{s.t.} \quad W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \quad \forall \, \omega \in \mathcal{S} \quad (4)$$

with $A(e^{j\omega})$ defined in (2) in terms of $\mathbf{b}$.

We emphasize that the problem of sparse filter design in (4) differs from the basic compressive sensing problem of reconstructing a signal, known to be sparse, from a small number of noise-free measurements (see, e.g., [27] and [28]). The latter can be formulated as follows:

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_0$$
$$\text{s.t.} \quad \mathbf{\Phi}\mathbf{x} = \mathbf{y} \quad (5)$$

where $\mathbf{y}$ is a vector of measurements and $\mathbf{\Phi}$ is a matrix representing the measurement process. In compressive sensing, it is assumed that the number of measurements is much lower than the dimension of the signal, i.e., $\mathbf{\Phi}$ has many fewer rows than columns, so that the problem in (5) has few constraints relative to the number of variables. This situation contrasts with that in

---

[1]An nth-*band filter* has the property that every nth coefficient in its impulse response is equal to zero except for the central coefficient.

(4) where the number of constraints is infinite. As we discuss in Section III, even when (3) is approximated by a finite number of constraints, the number of constraints must be much larger than the number of variables in order for the approximation to be valid.

The problem in (4) also differs from that in (5) in having constraints that are inequalities as opposed to equalities. Even in the case of noisy measurements, the most common approach in compressive sensing is to replace the equality constraints in (5) by an upper bound on the 2-norm of the residual $\mathbf{y} - \mathbf{\Phi}\mathbf{x}$. In contrast, the weighted ∞-norm of the frequency response error is more typically of interest in filter design, leading to the linear inequality constraints in (4).

As stated in Section I, the exact problem in (4) is computationally difficult. In the following sections, we consider two approximate approaches to the problem that have the potential to yield very sparse designs.

III. SUCCESSIVE THINNING ALGORITHMS

In this section, we explore a class of heuristic techniques for sparse filter design that we refer to as *successive thinning*. To motivate the approach, we first observe that (4) can be solved exactly by finding, for each $R = M + 1, M, M - 1, \ldots$, a filter that minimizes the maximum weighted error among all filters with no more than R nonzero coefficients in the set $\{b_n\}$. If for some value of R the optimal error δ_R exceeds δ_d, then there exist no feasible solutions to (4) with R or fewer nonzero coefficients. Consequently, this procedure is guaranteed to result in an optimal solution to (4), specifically by determining the lowest value of R for which $\delta_R \leq \delta_d$. The associated computational complexity, however, can be very high since a naive search for a minimax optimal filter with R nonzero coefficients has complexity of order $\binom{M+1}{R}$, which becomes a high-degree polynomial in M as R decreases from $M + 1$. The complexity of the overall procedure remains non-polynomial in the total number of coefficients $M + 1$.

The class of successive thinning algorithms discussed in this section can be viewed as approximations to the iterative procedure just described. We give an outline of one such algorithm as an example and then discuss some generalizations. In the first iteration of our example algorithm, a minimax optimal filter with at most M nonzero coefficients can be found by constraining each b_n in turn to a zero value and choosing one that minimizes the maximum error δ. In the second iteration, rather than searching over all possible configurations with two zero-valued coefficients, we continue to constrain to zero the coefficient selected in the first iteration, and restrict the search to finding a second coefficient that yields the smallest increase in δ when also constrained to zero. In each subsequent iteration, all of the coefficients selected in previous iterations remain constrained to zero, limiting the task to minimizing δ over all remaining choices for one additional zero-valued coefficient. As before, the algorithm terminates when δ_R first exceeds δ_d for some R, at which point the last feasible solution with $R + 1$ nonzero coefficients is taken to be the final design. We refer to this algorithm as the *minimum-increase rule* since the objective in each iteration is to choose a coefficient to set to zero that minimizes the increase in δ. The algorithm resembles the class of greedy algorithms [29] in that decisions regarding zero-valued coefficients made in previous iterations are never revisited.

A successive thinning algorithm based on the minimum-increase rule greatly reduces the number of configurations that are explored compared to an exact method for solving (4). More precisely, since each iteration searches over a number of configurations no greater than $M + 1$, and the number of iterations is also bounded by $M + 1$, the total number of configurations grows quadratically with $M + 1$. Despite the loss of a guarantee of optimality, the minimum-increase rule can in many cases produce filters with significantly fewer nonzero coefficients than conventional designs, without the complexity inherent in solving the exact problem.

The idea behind the minimum-increase rule can be generalized, and in some cases, further simplified. For example, multiple coefficients can be chosen in each iteration to be constrained to zero instead of a single coefficient, while still preserving the simplification of never removing these zero-value constraints once imposed. It would seem reasonable that the likelihood of obtaining a near-optimal solution increases with the number of coefficients constrained per iteration, while the complexity of the algorithm also increases. We focus on the single-coefficient case, i.e., the minimum-increase rule just discussed, in the interest of maintaining computational simplicity.

To decrease the complexity even further, we propose in Section III-A a second rule known as the *smallest-coefficient rule*, which sets to zero the smallest coefficient in absolute value of the filter obtained in the current iteration. The smallest-coefficient rule is computationally easier than the minimum-increase rule since it evaluates a single coefficient configuration per iteration, for a total number that is linear in $M + 1$. In our experience, the two rules often yield filters of equal sparsity.

To describe the structure of a successive thinning algorithm in more detail, let $i = 0, 1, 2, \ldots$ denote the iteration number. The algorithm maintains a growing list $\mathcal{Z}^{(i)} \subset \{0, \ldots, M\}$ of indices at which b_n is constrained to be zero. Equivalently, we may consider the index set $\mathcal{N}^{(i)}$ of possibly nonzero coefficients, i.e., the complement of $\mathcal{Z}^{(i)}$ with respect to the full set $\{0, \ldots, M\}$. In most cases, the initial set $\mathcal{N}^{(0)} = \{0, \ldots, M\}$, but this is not necessarily so as certain coefficients may be fixed to zero *a priori*, for example in systems with broken multipliers or array elements. In each iteration after $i = 0$, an additional index $m^{(i)} \in \mathcal{N}^{(i)}$ is added to $\mathcal{Z}^{(i)}$, resulting in a new set $\mathcal{Z}^{(i+1)}$ that is one element larger. We defer discussion of rules for selecting the index $m^{(i)}$ until Section III-A.

In iteration i, we wish to solve the following minimax optimization problem:

$$\min_{\{b_n\}, \delta} \quad \delta$$
$$\text{s.t.} \quad W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta \quad \forall \, \omega \in \mathcal{S},$$
$$b_n = 0 \quad \forall \, n \in \mathcal{Z}^{(i)} \tag{6}$$

and we denote the minimal value of δ by $\delta^{(i)}$ and the corresponding coefficients by $b_n^{(i)}$. In the case where $\mathcal{N}^{(0)} = \{0, \ldots, M\}$, $\mathcal{Z}^{(0)}$ is empty and (6) can be solved

using the Parks–McClellan algorithm. We assume that M is sufficiently large and $\mathcal{N}^{(0)}$ is sufficiently inclusive so that the initial error $\delta^{(0)}$ is strictly less than the allowable tolerance δ_d.[2] When $\mathcal{Z}^{(i)}$ is not empty, (6) is approximated by enforcing the frequency-domain constraints only on a finite grid of frequencies $\omega_1, \omega_2, \ldots, \omega_K$,[3] resulting in

$$P^{(i)}: \quad \min_{\{b_n\},\delta} \delta$$

$$\text{s.t.} \quad W(\omega_k)\left|\sum_{n=0}^{M} b_n \cos(n\omega_k) - H_d(e^{j\omega_k})\right| \leq \delta,$$
$$k = 1, \ldots, K,$$
$$b_n = 0 \quad \forall n \in \mathcal{Z}^{(i)} \quad (7)$$

which can be readily converted to a linear programming problem.

The linear programming dual of problem $P^{(i)}$ in (7) is given by

$$D^{(i)}: \quad \max_{\{p_k^+\},\{p_k^-\}} \sum_{k=1}^{K} W(\omega_k)H_d(e^{j\omega_k})(p_k^+ - p_k^-) \quad (8)$$

$$\text{s.t.} \quad \sum_{k=1}^{K} W(\omega_k)\cos(n\omega_k)(p_k^+ - p_k^-) = 0$$
$$\forall n \in \mathcal{N}^{(i)} \quad (9)$$
$$\sum_{k=1}^{K} (p_k^+ + p_k^-) = 1,$$
$$p_k^+ \geq 0, p_k^- \geq 0, k = 1, \ldots, K$$

and has the same optimal value $\delta^{(i)}$ as the primal problem $P^{(i)}$. Depending on the linear program solver used, either the primal $P^{(i)}$ or the dual $D^{(i)}$ may be more efficient to solve. In the case of the dual, the optimal coefficients $b_n^{(i)}$ for $n \in \mathcal{N}^{(i)}$ are available as the Lagrange multipliers corresponding to the equality constraints in (9); the coefficients $b_n^{(i)}$ for $n \in \mathcal{Z}^{(i)}$ are zero by design.

Since problem $P^{(i+1)}$ differs from problem $P^{(i)}$ only in the additional constraint $b_{m^{(i)}} = 0$, the sequence $\{\delta^{(i)}\}$ is nondecreasing. Equivalently, it is seen that $D^{(i+1)}$ differs from $D^{(i)}$ in the elimination of the equality constraint in (9) corresponding to $n = m^{(i)}$. In Section III-B, we show how to exploit the close relationship between similar problems to improve the efficiency of the algorithm.

The algorithm terminates when $\delta^{(i)}$ first exceeds δ_d for some $i = I + 1$, at which point the last feasible solution $b_0^{(I)}, \ldots, b_M^{(I)}$ is taken to be the final design. Note that $b_n^{(I)}$ cannot be zero for any $n \in \mathcal{N}^{(I)}$, as otherwise we would have a feasible solution with $I + 1$ zero coefficients and the algorithm could continue. Thus, minimizing the number of nonzero coefficients is equivalent to maximizing the number I of feasible iterations. We dis-

cuss next some heuristic rules for selecting the indices $m^{(i)}$ with this goal in mind.

A. Selection Rules

In this section, we discuss two rules for selecting coefficients to constrain to zero in a successive thinning algorithm. The first, referred to as the minimum-increase rule, was outlined at the beginning of Section III and is described in more detail below. A simplification of the minimum-increase rule, referred to as the smallest-coefficient rule, is described next. Other selection rules are also possible.

1) Minimum-Increase Rule: In this method, the index $m^{(i)}$ is chosen to minimize the increase in the maximum error δ. The algorithm maintains a list $\mathcal{C}^{(i)}$ of indices that are candidates for addition to the list $\mathcal{Z}^{(i)}$. The set $\mathcal{C}^{(i)}$ is always a subset of the set $\mathcal{N}^{(i)}$, and initially $\mathcal{C}^{(0)}$ is equal to $\mathcal{N}^{(0)}$. In iteration i, we determine for every $p \in \mathcal{C}^{(i)}$ the optimal error $\delta^{(i)}(p)$ that results from adding p to $\mathcal{Z}^{(i)}$, namely by solving (7) with $\mathcal{Z}^{(i)}$ replaced by $\mathcal{Z}^{(i)} \cup \{p\}$. The index $m^{(i)}$ is chosen to yield the smallest error value, i.e.,

$$m^{(i)} = \arg \min_{p \in \mathcal{C}^{(i)}} \delta^{(i)}(p).$$

Then $m^{(i)}$ is added to $\mathcal{Z}^{(i)}$ and removed from $\mathcal{C}^{(i+1)}$, the candidate list for the next iteration. In addition, to improve efficiency, we also remove from $\mathcal{C}^{(i+1)}$ those indices p for which $\delta^{(i)}(p) > \delta_d$, i.e., those coefficients that yield errors greater than the tolerance when set to zero. There is no further need to consider these coefficients in later iterations because the errors associated with setting them to zero can never decrease with the addition of new constraints.

2) Smallest-Coefficient Rule: In the case of the smallest-coefficient rule, the index $m^{(i)}$ in the ith iteration is chosen to correspond to the smallest of the optimal coefficients $b_n^{(i)}$ for $n \in \mathcal{N}^{(i)}$, i.e.,

$$m^{(i)} = \arg \min_{n \in \mathcal{N}^{(i)}} \left|b_n^{(i)}\right|.$$

The smallest-coefficient rule can be regarded as a simplification of the minimum-increase rule. It can be justified by recalling the interpretation of the coefficients $b_n^{(i)}, n \in \mathcal{N}^{(i)}$, as the Lagrange multipliers associated with the optimal solution to the dual problem $D^{(i)}$ given in (8). According to this interpretation, if the right-hand side of constraint n in (9) is changed from zero to a small value y, the corresponding optimal value is changed from $\delta^{(i)}$ to $\delta^{(i)} + b_n^{(i)}y$. Hence, if coefficient b_n is constrained to be zero, or equivalently, if constraint n in (9) is relaxed, the marginal rate of increase of the optimal error is given by $|b_n^{(i)}|$. Choosing $m^{(i)}$ to correspond to the smallest $|b_n^{(i)}|$ thus yields the smallest marginal rate of increase. The smallest-coefficient rule is therefore an approximation to the minimum-increase rule in a marginal or local sense.

A successive thinning algorithm that uses the smallest-coefficient rule evaluates a single coefficient configuration per iteration, for a total number that is linear in $M + 1$. In contrast, the minimum-increase rule enumerates a number of configurations that is quadratic in $M + 1$, since the lengths of the candidate lists $\mathcal{C}^{(i)}$ are of order $M + 1$. Nevertheless, in our experience the

---

[2]We have typically chosen M to be 10%–50% larger than the minimum order required to satisfy (3).

[3]In our experience with several examples, it is sufficient to set $K \sim 10M$ and to distribute the frequencies $\omega_1, \ldots, \omega_K$ uniformly over $\mathcal{S}$ to ensure that (3) is well-approximated. This is consistent with previous work on filter design using linear programming [24], [25], [30].

two rules seem to yield comparable levels of sparsity as demonstrated in Section V.

B. Efficiency Enhancements

This section discusses methods for improving the efficiency of successive thinning algorithms by exploiting relationships among the linear programming subproblems. These methods are based on linear programming techniques directed at situations in which an optimal solution to a problem P has been obtained and a solution to a similar problem $\tilde{P}$ is desired. In the context of successive thinning algorithms, problem P has the form of (7) for some set $\mathcal{Z}$ of zero-valued coefficients, while $\tilde{P}$ differs from P only in the additional constraint $b_m = 0$ for an index $m \notin \mathcal{Z}$, i.e., the corresponding index set $\tilde{\mathcal{Z}} = \mathcal{Z} \cup \{m\}$. Such a situation arises in the case of the minimum-increase rule when solving (7) with $\mathcal{Z}^{(i)}$ replaced by $\mathcal{Z}^{(i)} \cup \{p\}$ for $p \in \mathcal{C}^{(i)}$ given an existing optimal solution to the problem with $\mathcal{Z}^{(i)}$ alone. In the case of the smallest-coefficient rule, the problems $P^{(i)}$ and $P^{(i+1)}$ in consecutive iterations also differ by only one constraint.

For notational convenience in what follows, denote by $\mathbf{b}_{\mathcal{N}}$ the column vector composed of the coefficients b_n for $n \in \mathcal{N}$, where $\mathcal{N}$ is the index set of coefficients not constrained to be zero in problem P. The vector $\mathbf{b}_{\tilde{\mathcal{N}}}$ is defined similarly with the sets $\tilde{\mathcal{N}}$ and $\mathcal{N}$ related by

$$\mathcal{N} = \tilde{\mathcal{N}} \cup \{m\}. \tag{10}$$

We also introduce the matrix $\mathbf{A}$ and the vectors $\mathbf{h}$ and $\mathbf{e}$ with components

$$\mathbf{A}_{kn} = W(\omega_k) \cos(n\omega_k),$$
$$k = 1, \ldots, K, \quad n = 0, \ldots, M \tag{11a}$$
$$\mathbf{h}_k = W(\omega_k) H_d(e^{j\omega_k}), \quad k = 1, \ldots, K \tag{11b}$$
$$\mathbf{e}_k = 1, \quad k = 1, \ldots, K. \tag{11c}$$

Let $\mathbf{A}_{\mathcal{N}}$ and $\mathbf{A}_{\tilde{\mathcal{N}}}$ denote matrices obtained by extracting from $\mathbf{A}$ the columns corresponding to $\mathcal{N}$ and $\tilde{\mathcal{N}}$, respectively. Then problem P can be written in the following block matrix form:

$$P : \min_{\delta, \mathbf{b}_{\mathcal{N}}} \delta$$
$$\text{s.t.} \quad \begin{bmatrix} \mathbf{e} & \mathbf{A}_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\mathcal{N}} \end{bmatrix} \geq \mathbf{h}$$
$$\begin{bmatrix} \mathbf{e} & -\mathbf{A}_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\mathcal{N}} \end{bmatrix} \geq -\mathbf{h} \tag{12}$$

where the inequalities are to be interpreted as component-wise inequalities. Problem $\tilde{P}$ can be similarly rewritten with $\mathbf{b}_{\mathcal{N}}$ replaced by $\mathbf{b}_{\tilde{\mathcal{N}}}$ and $\mathbf{A}_{\mathcal{N}}$ by $\mathbf{A}_{\tilde{\mathcal{N}}}$. The dual of problem P in (12) is given by

$$D : \max_{\mathbf{p}^+, \mathbf{p}^-} \mathbf{h}^T (\mathbf{p}^+ - \mathbf{p}^-)$$
$$\text{s.t.} \quad \mathbf{e}^T (\mathbf{p}^+ + \mathbf{p}^-) = 1,$$
$$\mathbf{A}_{\mathcal{N}}^T (\mathbf{p}^+ - \mathbf{p}^-) = \mathbf{0},$$
$$\mathbf{p}^+ \geq \mathbf{0}, \mathbf{p}^- \geq \mathbf{0}, \tag{13}$$

and similarly for the dual $\tilde{D}$ of problem $\tilde{P}$.

We first discuss how the dual problem $\tilde{D}$ can be solved more efficiently when a solution $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ to problem D in (13) has already been determined. Equation (10) implies that $\mathbf{A}_{\tilde{\mathcal{N}}}$ has one fewer column than $\mathbf{A}_{\mathcal{N}}$, and therefore problem $\tilde{D}$ has one fewer constraint than problem D as seen by comparing (13) with its analog. As a consequence, $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ is also feasible for problem $\tilde{D}$ and can be used to initialize the solution of $\tilde{D}$. The reader is referred to [31] for the details of this procedure.

The corresponding situation with the primal problems is not as straightforward. Since the coefficient b_m is not constrained to zero in problem P, an optimal solution $(\delta^*, \mathbf{b}_{\mathcal{N}}^*)$ to P is usually infeasible for problem $\tilde{P}$ and cannot be used directly as an initialization. To address this, we propose solving an alternative problem $\hat{P}$, which is identical to P except for an additional penalty in the objective function on the absolute value of b_m. To derive the form of $\hat{P}$, we first use (10) and (11a) to decompose the product $\mathbf{A}_{\mathcal{N}} \mathbf{b}_{\mathcal{N}}$ as

$$\mathbf{A}_{\mathcal{N}} \mathbf{b}_{\mathcal{N}} = \mathbf{A}_{\tilde{\mathcal{N}}} \mathbf{b}_{\tilde{\mathcal{N}}} + \mathbf{A}_m b_m$$
$$= \mathbf{A}_{\tilde{\mathcal{N}}} \mathbf{b}_{\tilde{\mathcal{N}}} + \mathbf{A}_m (b_m^+ - b_m^-), \quad b_m^+ \geq 0, b_m^- \geq 0$$

where $\mathbf{A}_m$ is the column of $\mathbf{A}$ corresponding to $n = m$, and b_m is represented as $(b_m^+ - b_m^-)$. Then problem $\hat{P}$ can be expressed as

$$\hat{P} : \min_{\delta, \mathbf{b}_{\tilde{\mathcal{N}}}, b_m^+, b_m^-} \delta + C(b_m^+ + b_m^-)$$
$$\text{s.t.} \quad \begin{bmatrix} \mathbf{e} & \mathbf{A}_{\tilde{\mathcal{N}}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\tilde{\mathcal{N}}} \end{bmatrix}$$
$$+ \mathbf{A}_m(b_m^+ - b_m^-) \geq \mathbf{h},$$
$$\begin{bmatrix} \mathbf{e} & -\mathbf{A}_{\tilde{\mathcal{N}}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\tilde{\mathcal{N}}} \end{bmatrix}$$
$$- \mathbf{A}_m(b_m^+ - b_m^-) \geq -\mathbf{h},$$
$$b_m^+ \geq 0, \quad b_m^- \geq 0 \tag{14}$$

where C is a penalty constant.

When C is sufficiently large, it is expected that the optimal solution to $\hat{P}$ will have $b_m^+ = b_m^- = 0$, and consequently solving $\hat{P}$ becomes equivalent to solving $\tilde{P}$. The following theorem specifies values of C sufficient for the equivalence between $\hat{P}$ and $\tilde{P}$ to be exact.

Theorem I: If

$$C > \max_{k=1, \ldots, K} W(\omega_k) \tag{15}$$

then $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, \hat{b}_m^-)$ is an optimal solution to problem $\hat{P}$ if and only if $\hat{b}_m^+ = \hat{b}_m^- = 0$ and $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}})$ is an optimal solution to problem $\tilde{P}$.

The proof of Theorem 1 can be found in Appendix A.

In MATLAB implementations, the techniques presented in this section increased the speed of successive thinning algorithms by about 2–3 times compared to an implementation in which all linear programming problems are solved independently. Similar gains are expected for more specialized linear programming solvers.
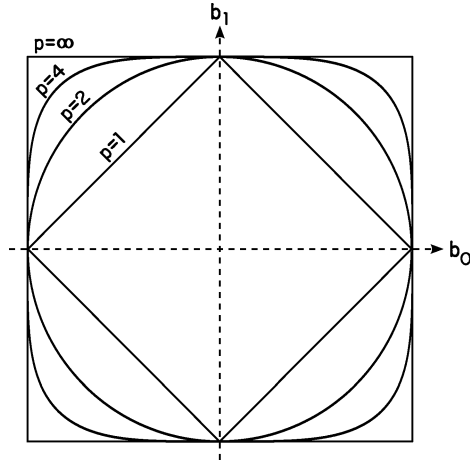
Fig. 1. Contours of equal $\|\mathbf{b}\|_p$ for various values of p. Determining the minimum-cost filter that meets a set of frequency-domain specifications involves expanding a given contour until it intersects with the polyhedron defining the set of feasible designs. Smaller values of p tend to promote solutions that lie close to the coordinate axes, i.e., that tend to be nearly sparse.

IV. MINIMUM 1-NORM DESIGN

Within the context of the successive thinning algorithms in Section III, the smallest-coefficient rule was proposed as a method for obtaining a sparse filter from a design having a coefficient b_n whose magnitude was small. In a particular iteration of the successive thinning process, the smallest coefficient was constrained to zero, along with the other coefficients constrained from previous iterations. In this section, we apply a similar strategy to filters having many small coefficients. In particular, we discuss a sparse filter design algorithm in which a filter is first designed to have many coefficients that are small in magnitude, followed by a single re-optimization stage where a filter with $\|\mathbf{b}\|_0 = J$ is obtained by constraining the $M + 1 - J$ smallest-magnitude coefficients to zero.

We approach designing a filter that is suitable for the re-optimization stage, i.e., one that has many small coefficients, by minimizing the 1-norm of $\mathbf{b}$

$$\|\mathbf{b}\|_1 = \sum_{n=0}^{M} |b_n|$$

subject to a set of pre-specified constraints on the magnitude response of the filter.

The 1-norm is a natural cost function to minimize in several respects. Because the 1-norm of $\mathbf{b}$ is equal to the sum of the magnitudes of the impulse response samples, minimizing the sum should intuitively tend to promote designs having many small coefficients. In general, minimizing $\|\mathbf{b}\|_p$ tends to produce increasingly sparse solutions as p is decreased, and $p = 1$ is the smallest p that results in an optimization solvable by a polynomial-time algorithm. The principle is illustrated in Fig. 1.

Choosing $\|\mathbf{b}\|_1$ as a cost function also results in a linear relaxation of the true minimum-$\|\mathbf{b}\|_0$ problem formulated in (4) and in that sense is closely related to the exact sparse filter design problem. The subsections that follow outline the design algorithm and discuss the details of this argument.

A. Proposed Design Technique

In obtaining a filter with many small coefficients, we formulate the filter design problem as

$$\min_{\{b_n\}} \quad \|\mathbf{b}\|_1$$
$$\text{s.t.} \quad W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \quad \forall \, \omega \in \mathcal{S},$$
$$b_n = 0 \quad \forall \, n \in \mathcal{Z} \tag{16}$$

which may be cast as the following linear program:

$$\min_{\mathbf{m},\mathbf{b}} \quad [\,1 \quad 1 \quad \cdots \quad 1\,]\mathbf{m}$$
$$\text{s.t.} \quad \mathbf{A}_{\mathcal{N}}\mathbf{b}_{\mathcal{N}} \geq \mathbf{h} - \delta_d \mathbf{e},$$
$$-\mathbf{A}_{\mathcal{N}}\mathbf{b}_{\mathcal{N}} \geq -\mathbf{h} - \delta_d \mathbf{e},$$
$$\mathbf{m} + \mathbf{b}_{\mathcal{N}} \geq 0, \quad \mathbf{m} - \mathbf{b}_{\mathcal{N}} \geq 0,$$
$$\mathbf{m} \geq 0. \tag{17}$$

Here, $\mathcal{Z}$ represents the index set for those coefficients constrained to zero a priori. The convention for $\mathbf{A}_{\mathcal{N}}, \mathbf{b}_{\mathcal{N}}, \mathbf{e}$ and $\mathbf{h}$ is consistent with the use of the variables in (12). Specifically, the vector $\mathbf{b}_{\mathcal{N}}$ contains the filter coefficients whose indices are elements of the set $\mathcal{N}$. $\mathbf{A}_{\mathcal{N}}$ transforms the vector $\mathbf{b}_{\mathcal{N}}$ to samples of $A(e^{j\omega})$, the discrete-time Fourier transform of the designed impulse response.

The minimization of the sum of the coefficient magnitudes in (17) is formulated by introducing a vector $\mathbf{m}$ that is constrained to bound the magnitudes of the coefficients b_n. As (17) is solved, the minimization of the cost function involving $\mathbf{m}$ causes its elements to decrease in value and eventually reach the magnitudes of the optimal coefficients. The cost function in (17) is the sum of the elements of $\mathbf{m}$, and consequently an optimal $\mathbf{b}_{\mathcal{N}}$ minimizes $\|\mathbf{b}_{\mathcal{N}}\|_1$ and in turn $\|\mathbf{b}\|_1$.

Although (17) is designed to yield filters having coefficients that are small in magnitude, these coefficients will generally be nonzero. Using the result from (17), the following optimization is next run to thin the impulse response of the filter:

$$\min_{\{b_n\},\delta} \quad \delta$$
$$\text{s.t.} \quad W(\omega_k)|A(e^{j\omega_k}) - H_d(e^{j\omega_k})| \leq \delta,$$
$$k = 1, \ldots, K,$$
$$b_n = 0 \quad \forall \, n \notin \mathcal{N}_J. \tag{18}$$

In particular, the result from (17) is used to determine $\mathcal{N}_J$, the index set of the J largest-magnitude coefficients that are allowed to be nonzero.

The linear program formulation of (18) is equivalent to problem P in (12). Again, (12) involves an index set $\mathcal{N}$ that specifies the coefficients that are allowed to be nonzero. Here, we use $\mathcal{N} = \mathcal{N}_J$.

The overall design algorithm consists of three stages.
  I) Solve (17).
 II) Choose $\mathcal{N}_J$ as the index set of the J largest-magnitude coefficients from the solution to Stage I.
III) Thin the impulse response of the filter by solving (12) using $\mathcal{N} = \mathcal{N}_J$.
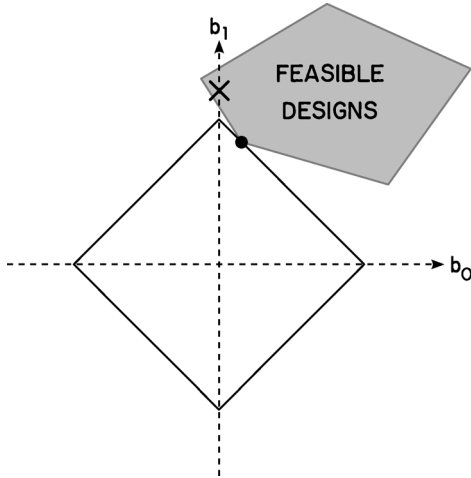
Fig. 2. Depiction of design process for the 2-coefficient case where a design with 1 nonzero coefficient is found. Dot indicates solution to Stage I, and $\mathbf{x}$ indicates solution to Stage III.

To determine the most-sparse filter achievable with the algorithm, Stages II and III may be repeated, iterating over J. In contrast to the successive thinning algorithms presented in Section III, the result from a given iteration does not depend on previous iterations. Furthermore, the set of the J largest-magnitude coefficients is contained in the set of the $J+1$ largest-magnitude coefficients, and the existence of a feasible solution to (12) for some $J = J_0$ consequently implies that a feasible solution exists for all $J > J_0$. Likewise, the lack of a feasible solution to (12) for some $J = J_0$ implies that no feasible solution exists for any $J < J_0$. Determining the smallest J that results in a feasible design may therefore be accomplished using a binary search that requires the solution of no more than $1 + \lceil \log_2(M+1) \rceil$ linear programs: a single linear program corresponding to solving the optimization in Stage I and a maximum of $\lceil \log_2(M+1) \rceil$ linear programs corresponding to repeating Stages II and III.

The behavior of the algorithm is summarized in Fig. 2 for the $M = 1$ (2 coefficient) case where a sparse design exists for $J = 1$. The dot indicates a solution to Stage I of the algorithm, and the $\mathbf{x}$ indicates the final design found in Stage III. In this example, the resulting filter is a true most-sparse design since the axis that lies closest to the dot intersects the set of feasible designs, i.e., the index of the smallest coefficient is the index of the zero-valued coefficient in a true most-sparse filter. This illustrates the motivation for the heuristic underlying the minimum 1-norm algorithm: we expect small coefficients resulting from the solution to (16) to occur at the same locations as zero-valued coefficients in the solution to (4). Although this expectation is not always met, it has been our experience that the algorithm still succeeds at thinning many different types of designs. Examples illustrating the performance of the algorithm are given in Section V.

B. Relationship to LP Relaxation of Minimum-$\|\mathbf{b}\|_0$ Problem

The optimization (16) bears a theoretical connection to the true minimum $\|\mathbf{b}\|_0$ problem in (4). Specifically, the minimum

$\|\mathbf{b}\|_1$ problem is a linear relaxation of the mixed-integer linear formulation of the minimum $\|\mathbf{b}\|_0$ problem. We derive this result and comment on the relationship of the optimal cost of (16) to the optimal cost of (4).

In arriving at a linear relaxation, (4) is cast as the equivalent mixed integer linear program

$$
\min_{\{b_n\},\{i_n\}} \quad \sum_{n=0}^{M} i_n
$$
$$
\text{s.t.} \quad \{b_n\} \in \mathcal{F},
$$
$$
|b_n| \leq B i_n, \quad n = 0, \dots, M,
$$
$$
i_n \in \{0, 1\} \tag{19}
$$

where the set of feasible designs $\mathcal{F}$ is defined as

$$
\mathcal{F} = \{\{b_n\} : \ W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \ \forall \ \omega \in \mathcal{S}\}. \tag{20}
$$

The i_n are binary variables that indicate whether a given coefficient is nonzero, and the strategy in (19) is to minimize the sum of the i_n, thereby arriving at the most-sparse design. In arriving at an optimal solution to (19), each of the inequalities involving B will either constrain the coefficient b_n to have value zero or will constrain its magnitude to $|b_n| \leq B$, depending on the value of i_n. The value of B is therefore chosen so that for any feasible design $\{b_n\} \in \mathcal{F}, B \geq |b_n|$ for all n, and consequently the optimizations (4) and (19) admit the same set of solutions.

In determining a linear relaxation of (19), we first remove the integer constraints on the i_n, resulting in the following linear program:

$$
\min_{\{b_n\},\{i_n\}} \quad \sum_{n=0}^{M} i_n
$$
$$
\text{s.t.} \quad \{b_n\} \in \mathcal{F},
$$
$$
|b_n| \leq B i_n, \ n = 0, \dots, M,
$$
$$
0 \leq i_n \leq 1. \tag{21}
$$

Making the substitution $\ell_n = B i_n$, we obtain

$$
\frac{1}{B} \min_{\{b_n\},\{\ell_n\}} \quad \sum_{n=0}^{M} \ell_n
$$
$$
\text{s.t.} \quad \{b_n\} \in \mathcal{F},
$$
$$
|b_n| \leq \ell_n, \ n = 0, \dots, M,
$$
$$
0 \leq \ell_n \leq B. \tag{22}
$$

Again B is chosen to be large enough that $|b_n| \leq B$ for any $\{b_n\} \in \mathcal{F}$, and the constraints $\ell_n \leq B$ can be removed without affecting any solution to (22). The minimization is separable into two optimizations involving the b_n and the ℓ_n respectively, and (22) becomes

$$
\frac{1}{B} \min_{\{b_n\} \in \mathcal{F}} \left\{ \begin{array}{ll} \min_{\{\ell_n\}} & \sum_{n=0}^{M} \ell_n \\ \text{s.t.} & |b_n| \leq \ell_n \\ & \ell_n \geq 0 \end{array} \right\}. \tag{23}
$$

The bracketed subproblem in (23) is optimized for $\ell_n = |b_n|$, resulting in

$$\frac{1}{B} \min_{\{b_n\} \in \mathcal{F}} \sum_{n=0}^{M} |b_n| = \frac{1}{B} \min_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_1. \tag{24}$$

The optimization (24) differs from (16) only by a constant scaling of the cost function when we additionally impose on (16) that $\mathcal{Z} = \emptyset$. An optimal $\mathbf{b}$ for (24) is consequently an optimal solution to (16) under this condition, and a similar argument applies in a straightforward way to (16) for arbitrary $\mathcal{Z}$ by introducing in (24) additional equalities that constrain appropriate coefficients to zero. The minimum-$\|\mathbf{b}\|_1$ problem is therein considered a linear relaxation of the mixed-integer formulation of the true minimum-$\|\mathbf{b}\|_0$ problem.

As a linear relaxation of the true minimum-$\|\mathbf{b}\|_0$ problem, the optimal cost of (24) may be used as a lower bound on the optimal cost of (4) and consequently may appear to be useful in the development of sparse filter design algorithms. However, the lower bound prescribed by (24) is fairly loose and is therefore not used within the context of the presented techniques. The details of this argument are presented in Appendix B.

V. DESIGN EXAMPLES

In this section, we present a number of examples to illustrate the performance of the algorithms in Sections III and IV.

A. Uniform Linear Beamformer

As is well known, the design of uniform linear beamformers is mathematically identical to the design of discrete-time FIR filters [32]. For a length N linear array with uniform spacing d, the beam pattern at a wavelength λ is given by

$$B(\theta, \lambda) = \sum_{n=0}^{N-1} w_n e^{jn\left[\frac{2\pi d}{\lambda} \sin\theta\right]}, \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \tag{25}$$

where θ is the angle from the array normal. Equation (25) has the form of a discrete-time Fourier transform of the array weights w_n with

$$\psi = \frac{2\pi d}{\lambda} \sin\theta \tag{26}$$

playing the role of the frequency variable. The problem of beamformer design consists of choosing weights to approximate a desired beam pattern. When the magnitude of the desired beam pattern is symmetric about $\psi = 0$, it is typical to restrict attention to weights that are real and even-symmetric, in which case the problem is equivalent to that of linear-phase filter design. Such symmetry occurs in the case of a beam directed normal to the array (broadside) with no nulls required at specific angles. In addition, beam patterns steered in other directions are frequently obtained by first designing a symmetric broadside beam pattern and then modulating the corresponding weights by an appropriate complex exponential.

For this example, we have chosen a desired beam pattern with the property that the mainlobe response is equal to unity over

TABLE I
SPECIFICATIONS FOR THE BEAMFORMER EXAMPLE

mainlobe region	$\psi \in [0, 0.0436\pi]$
sidelobe region	$\psi \in [0.0872\pi, \pi]$
mainlobe magnitude	within ± 0.5 dB of unity
sidelobe magnitude	below $-20, -30, -40$ dB

a range of angles as opposed to a single angle. The specifications for the desired beam pattern (assumed to be symmetric) are listed in Table I. In the case $d = \lambda/2$, the width of the mainlobe region is $5°$ at broadside. Beam patterns with a relatively wide and flat mainlobe find use in a number of contexts, which are sometimes grouped under the label *robust beamforming* [33]. A common feature of these application areas is the presence of uncertainty in the direction of interest and the consequent desire for a mainlobe shape that can accommodate the uncertainty.

In this example, we consider sidelobe levels of $-20, -30$, and -40 dB. For each sidelobe level, array weights are designed using the algorithms we have developed, namely successive thinning algorithms employing the minimum-increase and smallest-coefficient rules and the minimum 1-norm algorithm, as well as the Parks–McClellan algorithm for comparison. For the sparse design algorithms, we initially allow 50% more length than that required by the Parks–McClellan design. The actual length is determined by the positions of the nonzero weights.

Table II lists the number of nonzero weights (corresponding to the number of required physical array elements) and the array length returned by the algorithms in each case. For the successive thinning algorithms, the decrease in the number of nonzeros relative to the Parks–McClellan designs ranges from 13% to 33%, with the largest relative decrease at a sidelobe level of -20 dB. The minimum-increase rule gives slightly better results than the smallest-coefficient rule but requires significantly more computation. The minimum 1-norm algorithm, which has the lowest complexity of the sparse algorithms, yields relative decreases in the 8% to 33% range. Note also that the amount of extra length used in the sparse designs is never more than 19% of the Parks–McClellan length and can actually be zero as in the -30-dB case.

In a related experiment, we fix the number of nonzero weights at the values required by the Parks–McClellan algorithm (43, 55, and 79 for the different sidelobe levels) and determine how much additional sidelobe attenuation can be achieved using the sparse methods by increasing the sidelobe attenuation in 0.1-dB increments until an infeasible design is obtained. Here we allow 50% more length than is required for the respective initial Parks–McClellan designs. Due to their approximate nature, it is possible that the algorithms may return a feasible design at still greater levels of sidelobe attenuation beyond the point where an infeasible design is first encountered. The experiment is therefore intended as an indicator of the worst case increase when no further searching is performed.

Table III lists the sidelobe levels and array lengths yielded by our algorithms for each number of nonzeros. The sparse algorithms increase the level of attenuation by 2–8 dB over the Parks–McClellan designs. Fig. 3 shows the beam patterns given

TABLE II
NUMBERS OF NONZERO WEIGHTS AND ARRAY LENGTHS FOR DIFFERENT SIDELOBE LEVELS

sidelobe level [dB]	algorithm	non-zero weights	array length (in units of d)
−20	Parks-McClellan	43	42
	minimum-increase	29	48
	smallest-coefficient	31	46
	minimum 1-norm	29	50
−30	Parks-McClellan	55	54
	minimum-increase	47	54
	smallest-coefficient	47	54
	minimum 1-norm	47	54
−40	Parks-McClellan	79	78
	minimum-increase	65	82
	smallest-coefficient	69	82
	minimum 1-norm	73	82

TABLE III
SIDELOBE LEVELS AND ARRAY LENGTHS FOR DIFFERENT NUMBERS OF NONZERO WEIGHTS

| non-zero weights	algorithm	sidelobe level [dB]	array length (in units of d)
43	Parks-McClellan	−20.0	42
	minimum-increase	−28.3	56
	smallest-coefficient	−27.7	52
	minimum 1-norm	−26.7	54
55	Parks-McClellan	−30.0	54
	minimum-increase	−35.3	78
	smallest-coefficient	−34.9	78
	minimum 1-norm	−32.3	74
79	Parks-McClellan	−40.0	78
	minimum-increase	−45.5	90
	smallest-coefficient	−45.5	88
	minimum 1-norm	−42.8	88
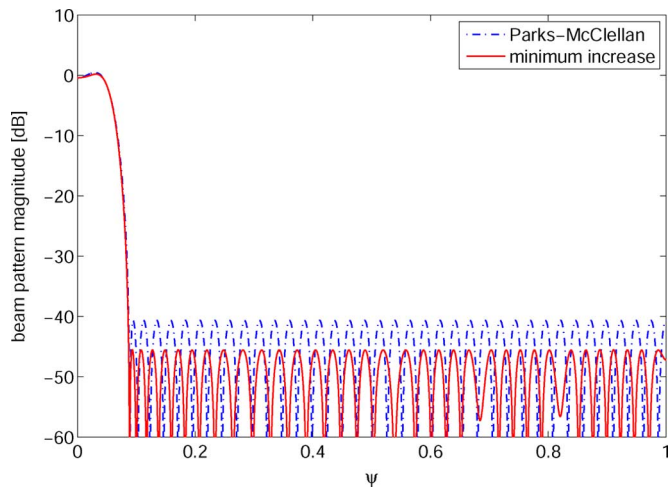
Fig. 3. Beam patterns produced by the Parks–McClellan and minimum-increase algorithms for the case of 79 nonzeros.

Fig. 4. Specifications used in Section V-B for designing an equalizing filter.

by the Parks–McClellan and minimum-increase algorithms for the case of 79 nonzeros.

B. Linear-Phase Acoustic Equalizer

As a second example, we consider the design of an acoustic equalizer. In equalizing the magnitude response of acoustic systems such as loudspeakers and microphones, a linear-phase discrete-time filter may be used to attain the desired magnitude response while preserving the group delay of the original system to within a constant offset. Specifications for the equalizing filter are often given in terms of minimum and maximum constraints on the desired magnitude response of the overall system, which in turn specify constraints on the magnitude response of the filter.
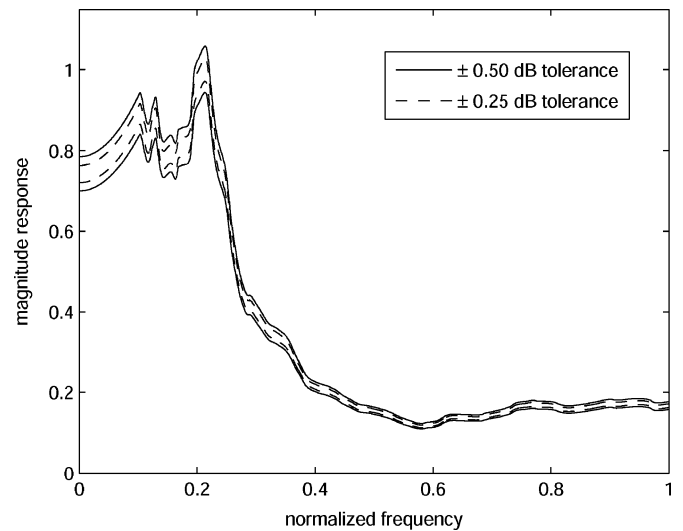
In this example, we design a sparse linear-phase discrete-time equalizer for use in a low-frequency portion of a public address system for which the sampling rate is 400 Hz. Two sets of specifications are considered, corresponding to magnitude tolerances of ±0.25 dB and ±0.50 dB about an ideal response. Fig. 4 depicts the specifications that are used in designing the system. For the sparse design algorithms, we allow 50% more length than the minimum required to obtain a corresponding feasible Parks–McClellan design.

TABLE IV
NUMBERS OF NONZERO IMPULSE RESPONSE VALUES AND DELAY
ELEMENTS FOR DIFFERENT TOLERANCES ON THE SPECIFICATIONS
FOR AN EQUALIZING FILTER

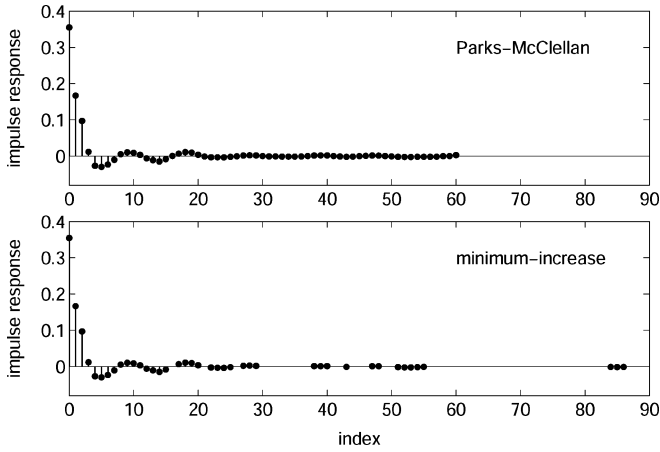| tolerance [dB]	algorithm	non-zero impulse response values	delay elements
±0.50	Parks-McClellan	45	44
	minimum-increase	39	56
	smallest-coefficient	39	56
	minimum 1-norm	43	48
±0.25	Parks-McClellan	121	120
	minimum-increase	81	172
	smallest-coefficient	81	172
	minimum 1-norm	83	176

Fig. 5. Equalizer impulse responses given by the Parks–McClellan and minimum-increase algorithms for a magnitude tolerance of ±0.25 dB. Only half of each impulse response is shown, re-indexed to allow easier comparison. Zero-valued coefficients are omitted.

Table IV lists the number of nonzero values in the impulse responses returned by the successive thinning and minimum 1-norm algorithms for each tolerance, as well as the number of required delay elements. The reported number of delay elements reflects the minimum number required to implement the filter design up to a constant offset in group delay using a causal direct-form FIR structure. Results from the Parks–McClellan algorithm are also included for comparison. For the successive thinning algorithms, the decreases in the number of nonzeros relative to the Parks–McClellan designs are 13% and 33%, with the largest relative decrease at the ±0.25 dB tolerance. The minimum 1-norm algorithm yields relative decreases of 4% and 31% for the ±0.50-dB and ±0.25-dB tolerances, respectively.

The impulse responses given by the Parks–McClellan and minimum-increase algorithms for a ±0.25-dB tolerance are shown in Fig. 5. Only half of each impulse response is shown, re-indexed to allow easier comparison. It can be observed that the minimum-increase algorithm tends to produce zero values at locations where the Parks–McClellan impulse response has small values. In this particular example, the minimum-increase algorithm has also introduced nonzero values at locations well beyond the support of the Parks–McClellan impulse response.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented two heuristic approaches to designing sparse FIR filters. In the first approach, an initial impulse response is successively thinned according to certain rules and the remaining coefficients are recomputed after each thinning to minimize the error in the frequency domain.

In the second approach, an impulse response having minimal 1-norm is used to determine which coefficients are constrained to have zero value in subsequent optimizations aimed at increasing sparsity. All of our algorithms rely on linear programming and the highly developed tools available for solving linear programs. Consequently, our algorithms are able to leverage ongoing advances in linear programming algorithms as well as in the implementation of solvers. The algorithms differ in complexity as measured by the required number of linear programs and accordingly also differ in the sparsity of the final designs. In terms of complexity and in the use of linear programming, our algorithms are comparable to a number of existing methods, notably [24]–[26] ([26] uses a simplex-like algorithm to solve a nonlinear problem). A careful study and comparison involving a wide range of examples and various linear program solvers could contribute to a better understanding of how all these methods compare.

The techniques presented in this work can in principle be extended to design nonlinear-phase FIR filters and IIR filters. A straightforward approach is to retain the overall structure of the algorithms while modifying only the optimization problems contained within, namely the minimization of the frequency-domain error with certain coefficients constrained to zero and the minimization of the 1-norm of the coefficients. This approach is likely to be successful for nonlinear-phase FIR filters since the optimization problems remain convex with respect to the coefficients and are thus tractable (see, e.g., [34] and [35]). Convexity is not preserved however in the IIR case if the numerator and denominator of the system function are optimized jointly.

Alternatively, in some IIR design methods, a filter is first designed to approximate a desired squared-magnitude response, followed by spectral factorization to obtain a stable IIR filter (e.g., [36]). However, sparsity in the square of the system function is generally not preserved by spectral factorization, further suggesting that the application of the presented algorithms to the problem of sparse IIR design may present additional challenges.

APPENDIX A
PROOF OF THEOREM 1

To facilitate the proof, we first show that the inequality

$$\left| \mathbf{A}_n^T (\mathbf{p}^+ - \mathbf{p}^-) \right| < C \qquad (27)$$

holds for any column $\mathbf{A}_n$ of the matrix $\mathbf{A}$ defined in (11a), any feasible solution $(\mathbf{p}^+, \mathbf{p}^-)$ to problem D in (13), and C satisfying (15). Using the fact that the magnitude of a sum is bounded by the sum of the magnitudes of individual terms

$$\left| \mathbf{A}_n^T (\mathbf{p}^+ - \mathbf{p}^-) \right|$$
$$= \left| \sum_{k=1}^{K} W(\omega_k) \cos(n\omega_k)(p_k^+ - p_k^-) \right|$$
$$\leq \sum_{k=1}^{K} W(\omega_k) |\cos(n\omega_k)| |p_k^+ - p_k^-|$$
$$\leq \sum_{k=1}^{K} W(\omega_k) |\cos(n\omega_k)| (p_k^+ + p_k^-)$$

noting that $W(\omega_k), p_k^+$ and p_k^- are non-negative. Bounding $W(\omega_k)$ by its maximum value and $|\cos(n\omega_k)|$ by 1, we obtain

$$\left| \mathbf{A}_n^T (\mathbf{p}^+ - \mathbf{p}^-) \right| \le \max_{k=1,\dots,K} W(\omega_k) \sum_{k=1}^{K} (p_k^+ + p_k^-) < C$$

as desired, where the second inequality follows from the first constraint in (13) and the assumption (15) on C.

Proceeding with the proof of the main result, suppose $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, \hat{b}_m^-)$ is an optimal solution to problem $\hat{P}$. Then $\hat{b}_m^+$ and $\hat{b}_m^-$ cannot both be nonzero, as otherwise both could be decreased by $\min\{\hat{b}_m^+, \hat{b}_m^-\}$, reducing the objective value without affecting feasibility. Assume first that $\hat{b}_m^- = 0$ and $\hat{b}_m^+ > 0$. Let $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*)$ and $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ be optimal solutions to problem $\tilde{P}$ and its corresponding dual $\tilde{D}$. We wish to show that $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*, 0, 0)$, which is a feasible solution to problem $\hat{P}$, has a strictly lower objective value than the assumed optimal solution $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, \hat{b}_m^-)$, thus establishing a contradiction.

First, we use strong duality to equate the optimal values for $\tilde{P}$ and $\tilde{D}$

$$\delta^* = \mathbf{h}^T (\mathbf{p}^{+*} - \mathbf{p}^{-*}). \tag{28}$$

Since $\mathbf{p}^{+*}$ and $\mathbf{p}^{-*}$ are non-negative and $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, \hat{b}_m^-)$ satisfies the constraints for problem $\hat{P}$ (14)

$$\mathbf{h}^T \mathbf{p}^{+*} \le \left(\begin{bmatrix} \mathbf{e} & \mathbf{A}_{\tilde{\mathcal{N}}} \end{bmatrix} \begin{bmatrix} \hat{\delta} \\ \hat{\mathbf{b}}_{\tilde{\mathcal{N}}} \end{bmatrix} + \mathbf{A}_m \hat{b}_m^+ \right)^T \mathbf{p}^{+*} \tag{29}$$

$$-\mathbf{h}^T \mathbf{p}^{-*} \le \left(\begin{bmatrix} \mathbf{e} & -\mathbf{A}_{\tilde{\mathcal{N}}} \end{bmatrix} \begin{bmatrix} \hat{\delta} \\ \hat{\mathbf{b}}_{\tilde{\mathcal{N}}} \end{bmatrix} - \mathbf{A}_m \hat{b}_m^+ \right)^T \mathbf{p}^{-*}. \tag{30}$$

Combining (28)–(30)

$$\begin{aligned} \delta^* &\le \hat{\delta} \mathbf{e}^T (\mathbf{p}^{+*} + \mathbf{p}^{-*}) + \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}^T \mathbf{A}_{\tilde{\mathcal{N}}}^T (\mathbf{p}^{+*} - \mathbf{p}^{-*}) \\ &\quad + \hat{b}_m^+ \mathbf{A}_m^T (\mathbf{p}^{+*} - \mathbf{p}^{-*}) \\ &= \hat{\delta} + \hat{b}_m^+ \mathbf{A}_m^T (\mathbf{p}^{+*} - \mathbf{p}^{-*}) \end{aligned} \tag{31}$$

where the simplifications result from the feasibility of $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ for problem D (13). Applying the bound in (27) to (31)

$$\delta^* < \hat{\delta} + C\hat{b}_m^+.$$

The left-hand side represents the objective value of the feasible solution $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*, 0, 0)$, while the right-hand side represents the value of the optimal solution $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, 0)$. This contradicts the optimality of $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, \hat{b}_m^+, 0)$, and hence $\hat{b}_m^+$ must be zero. The case $\hat{b}_m^+ = 0, \hat{b}_m^- > 0$ is similarly excluded.

The conclusion that $\hat{b}_m^+ = \hat{b}_m^- = 0$ has two consequences: First, the pair $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}})$ becomes a feasible solution to problem $\tilde{P}$. Second, the inequality in (31) becomes $\delta^* \le \hat{\delta}$, and in fact equality must hold in order for $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}}, 0, 0)$ to be an optimal solution to $\hat{P}$. Therefore, $(\hat{\delta}, \hat{\mathbf{b}}_{\tilde{\mathcal{N}}})$ is also an optimal solution to problem $\tilde{P}$, completing the proof of the forward direction.

To prove the converse, suppose that $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*)$ is an optimal solution to problem $\tilde{P}$. It was shown in the proof of the forward direction, specifically in (31), that the optimal objective value in problem $\hat{P}$ can be no less than δ^*. Furthermore, $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*, 0, 0)$ is

a feasible solution for $\hat{P}$ and achieves a value of δ^*. We conclude that $(\delta^*, \mathbf{b}_{\tilde{\mathcal{N}}}^*, 0, 0)$ is an optimal solution to $\hat{P}$. ∎

APPENDIX B
RELATIONSHIP BETWEEN OPTIMAL COSTS OF $\|\mathbf{b}\|_1$ AND $\|\mathbf{b}\|_0$

The optimal cost of (24) may be used as a lower bound on the optimal cost of (4), i.e., the smallest value of $\|\mathbf{b}\|_0$ for any $\mathbf{b}$ in the set of feasible designs, since constraints were removed in arriving at (24) from (4) while the set of feasible designs remained unchanged. Specifically,

$$\frac{1}{B} \min_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_1 \le \min_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_0. \tag{32}$$

The optimization (19), which is the mixed-integer linear formulation of (4), was made under the assumption that $B \ge |b_n|$ for any $\{b_n\} \in \mathcal{F}$. This condition may equivalently be formulated as $B \ge \max_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_\infty$, where $\|\mathbf{b}\|_\infty$ denotes the maximum magnitude of the elements of $\mathbf{b}$.

In tightening the bound on the value of $\min_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_0$ in (32), we choose $B = \max_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_\infty$ as the smallest allowable value of B, and we apply the ceiling function to both sides of the inequality. The right-hand side of (32) takes on integer values and is unaffected by the ceiling function, resulting in

$$\left\lceil \frac{\min_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_1}{\max_{\{b_n\} \in \mathcal{F}} \|\mathbf{b}\|_\infty} \right\rceil \le \|\mathbf{b}^*\|_0 \tag{33}$$

where $\mathbf{b}^*$ denotes an optimal set of coefficients for the minimum-$\|\mathbf{b}\|_0$ problem (4). The lower bound on $\|\mathbf{b}^*\|_0$ prescribed by the left-hand side of (33) may be computed by solving linear programs to determine the values in the numerator and denominator or may alternately be estimated using the design parameters of the filter.

The bound given by (33) becomes looser as the value of $\|\mathbf{b}^*\|_0$ increases. In illustrating this, consider a set $\mathcal{F} = \{\{b_0^*, 0, b_2^*\}\}$ that contains a single set of feasible (and optimal) coefficients with the property that $\|\mathbf{b}^*\|_0 = 2$. Equation (33) reduces to

$$\left\lceil \frac{|b_0^*| + 0 + |b_2^*|}{\max(|b_0^*|, 0, |b_2^*|)} \right\rceil \le \|\mathbf{b}^*\|_0$$

which, as can be verified, is met with equality under these assumptions for any choice of b_0^* and b_2^*. Adding an additional coefficient to the problem so that $\|\mathbf{b}^*\|_0 = 3$ results in the feasible set $\mathcal{F} = \{\{b_0^*, b_1^*, b_2^*\}\}$. Equation (33) in turn reduces to

$$\left\lceil \frac{|b_0^*| + |b_1^*| + |b_2^*|}{\max(|b_0^*|, |b_1^*|, |b_2^*|)} \right\rceil \le \|\mathbf{b}^*\|_0$$

which is tight only when the two smallest coefficient magnitudes add up to strictly more than the largest coefficient magnitude. The general trend continues; as the number of nonzero coefficients in a true most-sparse filter $\|\mathbf{b}^*\|_0$ increases, the conditions under which (33) is met with equality become more specific.

Fig. 6 depicts the minimization and maximization involved in computing the lower bound on $\|\mathbf{b}^*\|_0$ given by (33) for cases where the bound is tight. We illustrate the behavior of the minimization and maximization for $M = 1$, corresponding to filters with two coefficients. The bound is determined by expanding the diamond-shaped contour of equal $\|\mathbf{b}\|_1$ and reducing the square-shaped contour of equal $\|\mathbf{b}\|_\infty$ until each intersects the
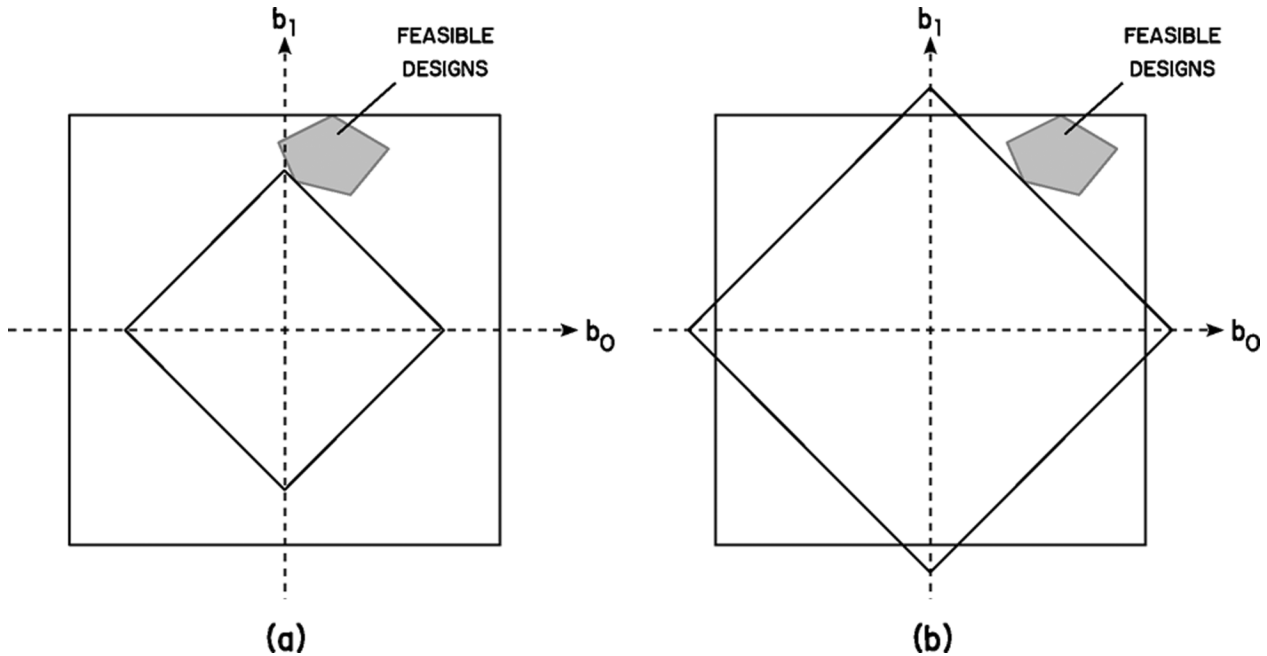
Fig. 6. Depiction of the minimization and maximization involved in computing the lower bound on $\|\mathbf{b}^*\|_0$ prescribed by (33), for two situations where the computed bound is the true value of $\|\mathbf{b}^*\|_0$. (a) The ratio of the norms corresponding to the two contours is positive and less than 1, resulting in the bound $\|\mathbf{b}^*\|_0 \geq 1$. (b) The ratio of the norms corresponding to the two contours is strictly greater than 1 and less than 2, resulting in the bound $\|\mathbf{b}^*\|_0 \geq 2$.

set of feasible designs. The ceiling of the ratio of the norms corresponding to these contours is the lower bound on $\|\mathbf{b}\|_0$ prescribed by (33).

REFERENCES

[1] M. Bateman and B. Liu, "An approach to programmable CTD filters using coefficients 0, +1, and −1," *IEEE Trans. Circuits Syst.*, vol. CAS-27, no. 6, pp. 451–456, Jun. 1980.

[2] Y. Lim and S. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 3, pp. 583–591, Jun. 1983.

[3] Y. C. Lim, "Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude," *IEEE Trans. Circuits Syst.*, vol. 37, no. 12, pp. 1480–1486, Dec. 1990.

[4] D. Li, Y. C. Lim, Y. Lian, and J. Song, "A polynomial-time algorithm for designing FIR filters with power-of-two coefficients," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1935–1941, Aug. 2002.

[5] C.-L. Chen and A. N. Willson, Jr., "A trellis search algorithm for the design of FIR filters with signed-powers-of-two coefficients," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 1, pp. 29–39, Jan. 1999.

[6] J. Adams and A. Willson, Jr., "A new approach to FIR digital filters with fewer multipliers and reduced sensitivity," *IEEE Trans. Circuits Syst.*, vol. CAS-30, no. 5, pp. 277–283, May 1983.

[7] J. Adams and A. Willson, Jr., "Some efficient digital prefilter structures," *IEEE Trans. Circuits Syst.*, vol. CAS-31, no. 3, pp. 260–266, Mar. 1984.

[8] R. J. Hartnett and G. F. Boudreaux-Bartels, "On the use of cyclotomic polynomial prefilters for efficient FIR filter design," *IEEE Trans. Signal Process.*, vol. 41, no. 5, pp. 1766–1779, May 1993.

[9] G. Boudreaux and T. Parks, "Thinning digital filters: A piecewise-exponential approximation approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 2, pp. 105–113, Feb. 1983.

[10] Y. Neuvo, D. Cheng-Yu, and S. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 3, pp. 563–570, Jun. 1984.

[11] T. Saramaki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. 35, no. 1, pp. 70–88, Jan. 1988.

[12] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits Syst.*, vol. ASSP-33, no. 4, pp. 357–364, Apr. 1986.

[13] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization," *IEEE Trans. Circuits Syst. II*, vol. 41, no. 8, pp. 518–525, Aug. 1994.

[14] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 2, pp. 155–162, Apr. 1981.

[15] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[16] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 5, pp. 444–456, Oct. 1975.

[17] T. A. Baran and A. V. Oppenheim, "Design and implementation of discrete-time filters for efficient rate-conversion systems," in *Proc. 41st Asilomar Conf. Signals, Syst., Comput.*, Nov. 2007, pp. 1088–1092.

[18] D. B. Turek, "Design of efficient digital interpolation filters for integer upsampling," M.S. thesis, Mass. Inst. of Technol., Cambridge, MA, Jun. 2004.

[19] J. T. Kim, W. J. Oh, and Y. H. Lee, "Design of nonuniformly spaced linear-phase FIR filters using mixed integer linear programming," *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 123–126, Jan. 1996.

[20] Y.-S. Song and Y. H. Lee, "Design of sparse FIR filters based on branch-and-bound algorithm," in *Proc. 40th Midwest Symp. Circuits and Systems*, 1997, vol. 2, pp. 1445–1448.

[21] O. Gustafsson, L. DeBrunner, V. DeBrunner, and H. Johansson, "On the design of sparse half-band like FIR filters," in *Proc. 41st Asilomar Conf. Signals, Syst., Comput,*, Nov. 2007, pp. 1098–1102.

[22] J.-K. Liang, R. De Figueiredo, and F. Lu, "Design of optimal Nyquist, partial response, Nth band, and nonuniform tap spacing FIR digital filters using linear programming techniques," *IEEE Trans. Circuits Syst.*, vol. CAS-32, no. 4, pp. 386–392, Apr. 1985.

[23] M. Smith and D. Farden, "Thinning the impulse response of FIR digital filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1981, vol. 6, pp. 240–242.

[24] D. Mattera, F. Palmieri, and S. Haykin, "Efficient sparse FIR filter design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2002, vol. 2, pp. 1537–1540.

[25] J. L. H. Webb and D. C. Munson Jr., "Chebyshev optimization of sparse FIR filters using linear programming with an application to beamforming," *IEEE Trans. Signal Process.*, vol. 44, no. 8, pp. 1912–1922, Aug. 1996.

[26] R. M. Leahy and B. D. Jeffs, "On the design of maximally sparse beamforming arrays," *IEEE Trans. Antennas Propag.*, vol. 39, no. 8, pp. 1178–1187, Aug. 1991.

[27] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[28] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[29] T. H. Gormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

[30] L. Rabiner, "Linear program design of finite impulse response (FIR) digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AE-20, no. 4, pp. 280–288, Oct. 1972.

[31] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Nashua, NH: Athena Scientific, 1997.

[32] H. L. Van Trees, *Optimum Array Processing*, ser. Detection, Estimation, and Modulation Theory. New York: Wiley , 2002, vol. 4.

[33] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Trans. Signal Process.*, vol. 47, pp. 2677–2684, Oct. 1999.

[34] L. J. Karam and J. H. McClellan, "Complex Chebyshev approximation for FIR filter design," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 3, pp. 207–216, Mar. 1995.

[35] D. Burnside and T. W. Parks, "Optimal design of FIR filters with the complex Chebyshev error criteria," *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 605–616, Mar. 1995.

[36] L. Rabiner, N. Graham, and H. Helms, "Linear programming design of IIR digital filters with arbitrary magnitude function," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-22, no. 2, pp. 117–123, Apr. 1974.

Thomas Baran (M'01) received the S.B. degree (*summa cum laude*) in electrical engineering and in biomedical engineering from Tufts University, Medford/Somerville, MA, in 2004 and the M.S. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 2007. He is currently pursuing the Ph.D. in the Digital Signal Processing Group, MIT.

He was with MIT Lincoln Laboratory, Hewlett-Packard Laboratories, and Bose Corporation. Current research interests include filter design, speech and audio enhancement, and signal processing theory.

Mr. Baran is a member of Sigma Xi, Eta Kappa Nu, and Tau Beta Pi.

Dennis Wei (S'09) received the S.B. degree in electrical engineering and physics and the M.Eng. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 2006 and 2007, respectively. He is currently pursuing the Ph.D. degree in the Digital Signal Processing Group at MIT. Current research interests include filter design, non-uniform sampling, and optimization and its application to signal processing.

Mr. Wei is a member of Phi Beta Kappa, Sigma Xi, Eta Kappa Nu, and Sigma Pi Sigma. He has been a recipient of the William Asbjornsen Albert Memorial Fellowship at MIT and a Siebel Scholar.

Alan V. Oppenheim (M'65–SM'71–F'77–LF'03) was born in New York, on November 11, 1937. He received S.B. and S.M. degrees and an Sc.D. degree, all in electrical engineering, from the Massachusetts Institute of Technology (MIT), Cambridge, in 1961 and 1964, respectively.

In 1964, he joined the faculty at MIT, where he is currently Ford Professor of Engineering. Since 1967, he has been affiliated with MIT Lincoln Laboratory and since 1977 with the Woods Hole Oceanographic Institution. His research interests are in the general area of signal processing and its applications. He is coauthor of the widely used textbooks *Discrete-Time Signal Processing* and *Signals and Systems*. He is also editor of several advanced books on signal processing.

Dr. Oppenheim is a member of the National Academy of Engineering and a member of Sigma Xi and Eta Kappa Nu. He is also the recipient of an honorary doctorate from Tel Aviv University. He has been a Guggenheim Fellow and a Sackler Fellow. He has received a number of awards for outstanding research and teaching, including the IEEE Education Medal, the IEEE Jack S. Kilby Signal Processing Medal, the IEEE Centennial Award and the IEEE Third Millennium Medal. From the IEEE Signal Processing Society, he has been honored with the Education Award, the Society Award, the Technical Achievement Award, and the Senior Award. He has also received a number of awards at MIT for excellence in teaching, including the Bose Award and the Everett Moore Baker Award and being named as a MacVicar Faculty Fellow.