DISCRETE TRANSFORMS, SEMIDEFINITE PROGRAMMING AND SUM-OF-SQUARES REPRESENTATIONS OF NONNEGATIVE POLYNOMIALS*

TAE ROH AND LIEVEN VANDENBERGHE†

Abstract. We present a new semidefinite programming formulation of sum-of-squares representations of nonnegative polynomials, cosine polynomials and trigonometric polynomials of one variable. The parametrization is based on discrete transforms (specifically, the discrete Fourier, cosine and polynomial transforms) and has a simple structure that can be exploited by straightforward modifications of standard interior-point algorithms.

Key words. Semidefinite programming, interior-point methods, nonnegative polynomials

AMS subject classifications. 90C22, 90C25

1. Introduction.

We discuss fast algorithms for semidefinite programs (SDPs) derived from weighted sum-of-squares representations of polynomials, cosine polynomials and trigonometric polynomials of one variable.

Several well-known theorems state that a (generalized) polynomial $f : \mathbf{R} \to \mathbf{R}$ is nonnegative on an interval or a union of intervals I,

$$f(t) \geq 0, \qquad t \in I, \tag{1.1}$$

if and only if it can be expressed as a *weighted sum of squares*

$$f(t) = \sum_{k=1}^{r} w_k(t)(y_k^T q(t))^2, \tag{1.2}$$

where $w_k(t) \geq 0$ on I. (For trigonometric polynomials, q and y_k are complex-valued, and we replace $(y_k^T q)^2$ with $|y_k^H q|^2$, where y_k^H denotes the complex conjugate transpose of y_k.) The weight functions w_k, the required number of terms r, and the vector of basis functions q depend on I and the class of functions f under consideration. Specific examples of sum-of-squares theorems are given in §3.1, §4.1 and §5.1.

It is also well-known that the weighted sum-of-squares property (1.2) can be expressed as a set of linear equations and linear matrix inequalities (LMIs) in the coefficients of f and a number of auxiliary matrix variables. In other words, (1.2) is equivalent to a convex constraint of the form

$$x = \sum_{i=1}^{s} \mathcal{H}_i(X_i), \qquad X_i \succeq 0, \quad i = 1, \ldots, s, \tag{1.3}$$

where x is the vector of coefficients of f with respect to some basis, $\mathcal{H}_i$ is a linear mapping, and $s \leq r$ [24, 25, 21]. Combining these results, we can cast the constraint (1.1), which is an infinite number of linear inequalities in the coefficients x, as a finite number of linear equations and linear matrix inequalities. Thus we can solve a wide variety of optimization problems over polynomials, subject to piecewise-polynomial upper and lower bounds, as SDPs. Numerous applications of this idea can be found in signal processing and control [26, 23, 27, 11, 34, 4, 8, 9, 18].

*Research supported by NSF under grant ECS-0200320.

†Department of Electrical Engineering, University of California, Los Angeles (roh@ee.ucla.edu, vandenbe@ee.ucla.edu).

In this paper we propose a specific choice for the mappings $\mathcal{H}_i$ in (1.3). We show that the weighted sum-of-squares property can be expressed in the following common form or its complex-valued counterpart:

$$x = \sum_{i=1}^{s} A_i\, \mathbf{diag}(C_i X_i C_i^T), \qquad X_i \succeq 0, \quad i = 1, \dots, s, \tag{1.4}$$

where $\mathbf{diag}(C_i X_i C_i^T)$ denotes the vector of diagonal elements of $C_i X_i C_i^T$, and the matrices A_i and C_i are defined in terms of discrete orthogonal transforms and their inverses. This unified parametrization offers several advantages. First, we will see that SDPs with constraints of the form (1.4), in which x and the matrices X_i are variables, can be solved very efficiently by taking advantage of some simple properties of the **diag** operator. This allows one to develop a single solver that solves SDPs derived from weighted sum-of-squares representations much more quickly than general-purpose codes. Second, in many cases additional savings are possible by using fast discrete transform algorithms for the multiplications with A_i and C_i. Third, the matrices C_i can be chosen to be orthogonal, while A_i is generally a product of an orthogonal and a diagonal matrix. These orthogonality properties are attractive from a numerical stability viewpoint.

Our interest in numerical methods for SDPs derived from sum-of-squares representations is motivated by several recent papers. Nesterov in [24] pointed out the connections between sum-of-squares representations, semidefinite programming and classical results in moment theory. He also described a straightforward method for converting weighted sum-of-squares representations (1.2) into constraints of the form (1.3). We explain the method for the case with $w_i(t) = 1$. Let $q : \mathbf{R} \to \mathbf{R}^{m+1}$. Suppose $p_i(t)$, $i = 0, \dots, n$, are basis functions whose span contains all products $q_k(t) q_l(t)$, so there exist matrices $F_i \in \mathbf{S}^{m+1}$ such that

$$q(t)q(t)^T = \sum_{i=0}^{n} p_i(t) F_i.$$

A function f can be expressed as a sum of squares $f(t) = \sum_{k=1}^{r} (y_k^T q(t))^2$ for some r and y_k, if and only if

$$f(t) = \sum_{k=1}^{r} (y_k^T q(t))^2 = \mathbf{tr}(q(t)q(t)^T X) = \sum_{i=0}^{n} \mathbf{tr}(F_i X) p_i(t),$$

where $X = \sum_{k=1}^{r} y_k y_k^T$. We see that f is a sum of squares if and only if $f(t) = x_0 p_0(t) + \cdots + x_n p_n(t)$ where

$$x_i = \mathbf{tr}(F_i X), \quad i = 0, \dots, n, \qquad X \succeq 0 \tag{1.5}$$

for some $X \in \mathbf{S}^{m+1}$. Therefore (1.3) holds with $\mathcal{H}_1(X) = (\mathbf{tr}(F_0 X), \dots, \mathbf{tr}(F_n X))$, and $s = 1$.

As an example, it is well-known that a nonnegative polynomial of even degree

$$f(t) = x_0 + x_1 t + \cdots + x_{2m} t^{2m}$$

can be expressed as a sum of squares of two polynomials of degree m or less. To derive equivalent LMI conditions, we take $q(t) = (1, t, \dots, t^m)$, and note that

$$q(t)q(t)^T = \sum_{i=0}^{2m} t^i F_i, \qquad F_{i,kl} = \begin{cases} 1 & k + l = i \\ 0 & \text{otherwise.} \end{cases}$$

For this choice of F_i, (1.5) reduces to

$$x_i = \sum_{k+l=i} X_{kl}, \quad i = 0, \ldots, 2m, \qquad X \succeq 0. \tag{1.6}$$

We can conclude that $f(t)$ is nonnegative if and only if there exists an $X \in \mathbf{S}^{m+1}$ such that (1.6) holds. We refer to Nesterov [24] and Faybusovich [12, 13] for more examples and extensions of Nesterov's approach.

SDPs derived from sum-of-squares representations involve auxiliary matrix variables and are often large scale and difficult to solve using general-purpose solvers. This has spurred research into specialized implementations of interior-point methods. The most successful approaches have been based on dual barrier methods [14, 16, 4], and exploit properties of the logarithmic barrier function for the dual constraints associated with (1.3). Genin *et al.* [14] consider problems involving matrix-valued polynomials that are nonnegative on the unit circle, the real axis or the imaginary axis. They note that the dual variables have low displacement rank (for example, due to Toeplitz or Hankel structure) and use this property to reduce the cost of computing the gradient and Hessian of the dual barrier function. This results in a substantial reduction of the complexity per iteration, as compared to a general-purpose solver. In [4] similar gains are achieved for a more specific class of problems, involving nonnegative scalar trigonometric polynomials. As in the method of [14], the basic idea is to evaluate the gradient and Hessian of the dual barrier function fast. In [4] this is accomplished by using the discrete Fourier transform (DFT) of triangular factors of the inverses of the dual variables. The techniques discussed in this paper can be interpreted as an extension of the DFT method of [4] to a much wider class of problems and to general interior-point methods (primal, dual, or primal-dual). Several of the key ideas in this paper also extend to SDPs derived from sum-of-squares characterizations of multivariate polynomials. In this context, our techniques are related to recent work by Löfberg and Parrilo on improving the efficiency of SDP solvers for sum-of-squares programming (see [22], which appeared after the first submission of this paper).

Notation. The set of real symmetric $n \times n$ matrices is denoted $\mathbf{S}^n$; the set of Hermitian $n \times n$ matrices is denoted $\mathbf{H}^n$. $A \succeq 0$ means A is positive semidefinite; $A \succ 0$ means A is positive definite. $\mathbf{tr}(A)$ is the trace of A. For a square matrix A, $\mathbf{diag}(A)$ is the vector of diagonal elements of A. For an n-vector a, $\mathbf{diag}(a)$ is the diagonal matrix with the elements of a on its diagonal. A^T is the transpose of the matrix A, $\bar{A}$ is the complex conjugate, and $A^H = (\bar{A})^T$ is the complex conjugate transpose. $A \circ B$ denotes the Hadamard product of two matrices A and B of the same dimensions, *i.e.*, the matrix with elements $(A \circ B)_{ik} = A_{ik}B_{ik}$. The same notation is used for vectors: $(x \circ y)_i = x_i y_i$. For real matrices, $\mathbf{sqr}(A) = A \circ A$; for complex matrices, $\mathbf{sqr}(A) = A \circ \bar{A}$. We use the notation $(x_0, x_1, \ldots, x_n)$ for the (column) vector $\begin{bmatrix} x_0 & x_1 & \cdots & x_n \end{bmatrix}^T$. $\mathbf{1}$ is the vector with all components one, with dimension determined from the context. Throughout the paper the symbol j is reserved for the number $\sqrt{-1}$. We use $\deg(f)$ to denote the degree of a polynomial, cosine polynomial or trigonometric polynomial f. For a trigonometric polynomial $f(\omega) = x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega})$, we define $\deg(f) = n$ if $x_n \neq 0$.

2. A class of structured SDPs. Suppose the matrices F_i in the standard form SDP

$$
\begin{array}{ll}
\text{minimize} & \mathbf{tr}(DX) \\
\text{subject to} & \mathbf{tr}(F_i X) = b_i, \quad i = 1, \ldots, m \\
& X \succeq 0
\end{array}
\tag{2.1}
$$

can be factored as

$$
F_i = C^T \mathbf{diag}(a_i) C, \quad i = 1, \ldots, m, \tag{2.2}
$$

where $C \in \mathbf{R}^{q \times n}$ and $a_i \in \mathbf{R}^q$. In other words, the matrices F_i can be written as different linear combinations of q rank-one matrices $c_i c_i^T$, where c_i^T is the ith row of C. Substituting (2.2) in (2.1) we obtain

$$
\begin{array}{ll}
\text{minimize} & \mathbf{tr}(DX) \\
\text{subject to} & A\,\mathbf{diag}(CXC^T) = b \\
& X \succeq 0,
\end{array}
\tag{2.3}
$$

where $A \in \mathbf{R}^{m \times q}$ has rows a_i^T. In this section we will see that if $q \ll mn$, the SDP (2.3) can be solved very efficiently by taking advantage of the structure in the constraints. In §3–§5 we will then show that this type of structure arises in SDPs derived from sum-of-squares representations of nonnegative polynomials.

Note that a factorization of the form (2.2) always exists. For example, one can use the eigenvalue decomposition to factor F_i as $F_i = V_i \,\mathbf{diag}(\lambda_i) V_i^T$ with $V_i \in \mathbf{R}^{n \times r_i}$, $\lambda_i \in \mathbf{R}^{r_i}$, where $r_i = \mathbf{rank}(F_i)$, and then take $q = \sum_i r_i$,

$$
C = \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_m^T \end{bmatrix}, \quad a_1 = \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad a_m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_m \end{bmatrix}. \tag{2.4}
$$

For general dense matrices, with $r_i = n$ and $q = mn$, there is no advantage in expressing the SDP as (2.3). If the matrices F_i are all low rank ($r_i \ll n$), then (2.4) provides a factorization (2.2) with $q \ll mn$. In this case our techniques are similar to known methods for exploiting low-rank structure [6]. Our focus in this paper, however, is on more general types of structure in which the matrices F_i are not low-rank.

2.1. Solution via interior-point methods. It will be convenient in later sections to use the problem format

$$
\begin{array}{ll}
\text{minimize} & \mathbf{tr}(DX) + c^T y \\
\text{subject to} & A\,\mathbf{diag}(CXC^T) + By = b \\
& X \succeq 0,
\end{array}
\tag{2.5}
$$

which includes a vector variable $y \in \mathbf{R}^p$. The problem parameters are $c \in \mathbf{R}^p$, $D \in \mathbf{S}^n$, $b \in \mathbf{R}^m$, $A \in \mathbf{R}^{m \times q}$, $B \in \mathbf{R}^{m \times p}$, and $C \in \mathbf{R}^{q \times n}$. The corresponding dual SDP is

$$
\begin{array}{ll}
\text{maximize} & b^T z \\
\text{subject to} & C^T \,\mathbf{diag}(A^T z) C \preceq D \\
& B^T z = c,
\end{array}
\tag{2.6}
$$

with variable $z \in \mathbf{R}^m$.

Interior-point methods for solving the pair of SDPs (2.5) and (2.6) typically require the solution of one or two sets of linear equations of the form

$$-T^{-1}\Delta X T^{-1} + C^T \,\mathbf{diag}(A^T\Delta z)C = R \tag{2.7}$$

$$A\,\mathbf{diag}(C\Delta X C^T) + B\Delta y = r_1 \tag{2.8}$$

$$B^T\Delta z = r_2 \tag{2.9}$$

at each iteration. The variables are Δy, Δz, ΔX; the matrix $T \succ 0$ and the righthand sides $R \in \mathbf{S}^n$, $r_1 \in \mathbf{R}^m$, and $r_2 \in \mathbf{R}^p$ are given. We refer to these equations as Newton equations, because they can be obtained by linearizing nonlinear equations that characterize the central path. The matrices T and the righthand sides R, r_1, r_2 change at each iteration, and depend on the particular method used. In some methods (for example, dual barrier methods) the matrix T may have additional structure that can be exploited [5, 14, 4]. In this paper, however, we will make no assumption about T, other than positive definiteness. The technique outlined below therefore applies to a wide variety of interior-point methods, including primal methods, dual methods, and primal-dual methods based on the Nesterov-Todd scaling [30]. Other primal-dual methods (in particular, the methods in [3, 17, 19]) involve Newton equations with a closely related structure.

It is well-known that the number of iterations in an interior-point method is typically in the range 10–50, almost independent of the problem dimensions, and that the overall cost is dominated by the cost of solving the Newton equations. An efficient method that takes advantage of the structure in the Newton equations (2.7)–(2.9) is as follows. We first eliminate ΔX from the first equation to get

$$A\,\mathbf{diag}(CTC^T\,\mathbf{diag}(A^T\Delta z)CTC^T) + B\Delta y = r_3 \tag{2.10}$$

$$B^T\Delta z = r_2, \tag{2.11}$$

where $r_3 = r_1 + A\,\mathbf{diag}(CTRTC^T)$. The 1,1-block can be written in matrix-vector form by using the identity $\mathbf{diag}(P\,\mathbf{diag}(u)Q^T) = (P \circ Q)u$:

$$\begin{aligned} A\,\mathbf{diag}\left(CTC^T\,\mathbf{diag}(A^T\Delta z)CTC^T\right) &= A\left((CTC^T)\circ(CTC^T)\right)A^T\Delta z \\ &= A\,\mathbf{sqr}(CTC^T)A^T\Delta z. \end{aligned}$$

The equations (2.10) and (2.11) therefore reduce to $m+p$ equations in $m+p$ variables

$$\begin{bmatrix} A\,\mathbf{sqr}(CTC^T)A^T & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_3 \\ r_2 \end{bmatrix}. \tag{2.12}$$

From the solution Δz, Δy, we find ΔX by solving (2.7).

To justify this approach, we can contrast it with the calculations used in common general-purpose implementations (such as Sedumi [28] or SDPT3 [31]). In a general-purpose code the Newton equations are also solved by eliminating ΔX and solving the reduced Newton equations (2.12). The difference lies in the way the $1,1$-block $H = A\,\mathbf{sqr}(CTC^T)A^T$ is assembled. In a general-purpose algorithm the linear mapping $C^T\,\mathbf{diag}(A^Tz)C$ is represented in the canonical form

$$C^T\,\mathbf{diag}(A^Tz)C = \sum_{i=1}^m z_iF_i$$

where $F_i = C^T \mathbf{diag}(a_i)C$ and a_i^T is the ith row of A. The matrix H is computed as

$$H_{ik} = \mathbf{tr}(TF_iTF_k), \qquad i,k = 1,\ldots,m.$$

These computations can be arranged in different ways, for example, by first computing the m matrices TF_i and then forming the $m(m+1)/2$ inner products $\mathbf{tr}(TF_iTF_k)$. If we assume that the matrices F_i are dense and full-rank and that the problem dimensions m, n, p are of the same order, this yields an $O(n^4)$ method for constructing the coefficient matrix in (2.12), which can then be solved in $O(n^3)$ operations. The direct formula $H = A\,\mathbf{sqr}(CTC^T)A^T$ is faster, because it requires $O(n^3)$ operations (again assuming that all problem dimensions are of the same order). Moreover, in the applications that we describe below, the matrices A and C represent discrete transforms or inverse discrete transforms, so fast methods often exist for multiplications with A and C.

2.2. Extension to complex data and variables. In applications involving trigonometric polynomials we will encounter SDPs in which some of the data and variables are complex numbers. It is therefore of interest to consider the complex counterpart of (2.5) and (2.6):

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DX) + c^Ty \\ \text{subject to} & A\,\mathbf{diag}(CXC^H) + By = b \\ & X \succeq 0 \end{array} \tag{2.13}$$

and

$$\begin{array}{ll} \text{maximize} & \Re(b^Hz) \\ \text{subject to} & C^H\,\mathbf{diag}(\Re(A^Hz))C \preceq D \\ & \Re(B^Hz) = c. \end{array}$$

The primal variables are $X \in \mathbf{H}^n$ and $y \in \mathbf{R}^p$. The dual variable is $z \in \mathbf{C}^m$. The problem parameters are $D \in \mathbf{H}^n$, $c \in \mathbf{R}^p$, $A \in \mathbf{C}^{m\times q}$, $C \in \mathbf{C}^{q\times n}$, $B \in \mathbf{C}^{m\times p}$ and $b \in \mathbf{C}^m$.

The Newton equations for (2.13) can be written as

$$\begin{aligned} -T^{-1}\Delta XT^{-1} + C^H\,\mathbf{diag}(\Re(A^H\Delta z))C &= R \\ A\,\mathbf{diag}(C\Delta XC^H) + B\Delta y &= r_1 \\ \Re(B^H\Delta z) &= r_2. \end{aligned}$$

Eliminating ΔX from the first equation gives

$$A\,\mathbf{diag}(CTC^H\,\mathbf{diag}(\Re(A^H\Delta z))CTC^H) + B\Delta y = r_3 \tag{2.14}$$

$$\Re(B^H\Delta z) = r_2, \tag{2.15}$$

where $r_3 = r_1 + A\,\mathbf{diag}(CTRTC^H)$. Again using the identity $\mathbf{diag}(P\,\mathbf{diag}(u)Q^T) = (P\circ Q)u$, we can write the 1,1-block as

$$\begin{aligned} A\,\mathbf{diag}\left(CTC^H\,\mathbf{diag}(\Re(A^H\Delta z))CTC^H\right) &= A\left((CTC^H)\circ(CTC^H)^T\right)\Re(A^H\Delta z) \\ &= A\,\mathbf{sqr}(CTC^H)\Re(A^H\Delta z). \end{aligned}$$

Plugging this in in the equations (2.14) and (2.15) and expanding complex data and variables in their real and imaginary parts ($A = A_\mathrm{r} + jA_\mathrm{i}$, et cetera), we obtain

$$\left[\begin{array}{ccc} A_\mathrm{r}\,\mathbf{sqr}(CTC^H)A_\mathrm{r}^T & A_\mathrm{r}\,\mathbf{sqr}(CTC^H)A_\mathrm{i}^T & B_\mathrm{r} \\ A_\mathrm{i}\,\mathbf{sqr}(CTC^H)A_\mathrm{r}^T & A_\mathrm{i}\,\mathbf{sqr}(CTC^H)A_\mathrm{i}^T & B_\mathrm{i} \\ B_\mathrm{r}^T & B_\mathrm{i}^T & 0 \end{array}\right]\left[\begin{array}{c} \Delta z_\mathrm{r} \\ \Delta z_\mathrm{i} \\ \Delta y \end{array}\right] = \left[\begin{array}{c} r_{3,\mathrm{r}} \\ r_{3,\mathrm{i}} \\ r_2 \end{array}\right]. \tag{2.16}$$

The extension to the case where only some of the rows of A and B (and the corresponding elements of Δz) in the equations (2.14) and (2.15) are complex, is straightforward: in (2.16) we simply delete the equations and variables corresponding to the zero rows in A_{i} and Δz_{i}.

3. Trigonometric polynomials. Let f be a trigonometric polynomial of degree n or less, *i.e.*, a function of the form

$$f(\omega) = \bar{x}_n e^{jn\omega} + \cdots + \bar{x}_1 e^{j\omega} + x_0 + x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega} \\ = x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}) \tag{3.1}$$

where $x = (x_0, \ldots, x_n) \in \mathbf{R} \times \mathbf{C}^n$. In this section we show that f is nonnegative on a subinterval of $[0, 2\pi]$ if and only if it satisfies an SDP constraint of the form

$$x = \sum_{k=1}^{r} A_k \,\mathbf{diag}(C_k X_k C_k^H), \qquad X_k \succeq 0, \quad k = 1, \ldots, r,$$

with $r = 1$ or $r = 2$. This result follows by reformulating classical sum-of-squares characterizations of nonnegative trigonometric polynomials via the discrete Fourier transform.

3.1. Sum-of-squares characterizations. If the trigonometric polynomial (3.1) is nonnegative and of degree n (*i.e.*, $x_n \neq 0$), then it can be expressed as

$$f(\omega) = |g(e^{-j\omega})|^2$$

where $g(s) = u_0 + u_1 s + \cdots + u_n s^n$ is a polynomial of degree n, with (in general) complex coefficients u_k. This is known as the *Riesz-Fejér theorem* or the *spectral factorization theorem* [29, page 3], [20, page 60]. Several efficient methods exist for computing g from x; see, for example, [32, Appendix D].

The following generalization of the Riesz-Fejér theorem can be found in [2, page 133], [20, page 294], [8, Theorem 2], [16, page 44], [12, 13]. If f is nonnegative on $[\alpha - \beta, \alpha + \beta]$, where $0 < \beta < \pi$, then it can be expressed as

$$f(\omega) = |g(e^{-j\omega})|^2 + (\cos(\omega - \alpha) - \cos\beta)\, |h(e^{-j\omega})|^2,$$

where g and h are polynomials with $\deg(g) \leq n$ and $\deg(h) \leq n - 1$. In other words, f is the sum of two nonnegative trigonometric polynomials. The first trigonometric polynomial $|g(e^{-j\omega})|^2$ is nonnegative everywhere; the second term is the product of a nonnegative trigonometric polynomial $|h(e^{-j\omega})|^2$ with the trigonometric polynomial $\cos(\omega - \alpha) - \cos\beta$, which is nonnegative on $[\alpha - \beta, \alpha + \beta]$.

3.2. Discrete Fourier transform. The discrete Fourier transform (DFT) offers a convenient way to map the coefficients of a pseudo-polynomial

$$F(s) = x_{-n} s^{-n} + \cdots + x_{-1} s^{-1} + x_0 + x_1 s + \cdots + x_n s^n \tag{3.2}$$

to its values at equidistant points on the unit circle, and vice-versa. Let $W_{\mathrm{DFT}} \in \mathbf{C}^{N \times N}$ be the length-N DFT matrix, with $N \geq 2n + 1$:

$$W_{\mathrm{DFT}} = \begin{bmatrix} w_0 & w_1 & \cdots & w_{N-1} \end{bmatrix},$$

where

$$w_k = (1, e^{-jk\omega_N}, e^{-j2k\omega_N}, \ldots, e^{-j(N-1)k\omega_N}), \qquad \omega_N = 2\pi/N.$$

For the pseudo-polynomial F given by (3.2), define

$$\tilde{x} = (x_0, x_1, \ldots, x_n, 0, \ldots, 0, x_{-n}, \ldots, x_{-1}) \in \mathbf{C}^N$$
$$y = (F(1), F(e^{-j\omega_N}), \ldots, F(e^{-j(N-1)\omega_N})) \in \mathbf{C}^N.$$

Then it is easily verified that

$$y = W_{\mathrm{DFT}}\tilde{x}, \qquad \tilde{x} = \frac{1}{N} W_{\mathrm{DFT}}^H y.$$

In other words, the DFT maps the coefficients of F to the values of F at N equidistant points on the unit circle; the inverse DFT maps these sample values back to the coefficients.

If $x_{-k} = \bar{x}_k$, then $F(e^{-j\omega})$ is the trigonometric polynomial

$$F(e^{-j\omega}) = f(\omega) = x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega})$$

and the relation between $x = (x_0, x_1, \ldots, x_n)$ and $y = (f(0), f(\omega_N), \ldots, f((N-1)\omega_N))$ simplifies to

$$x = \frac{1}{N} W^H y,$$

where the columns of W are the first $n+1$ columns of W_{DFT}:

$$W = \begin{bmatrix} w_0 & w_1 & \cdots & w_n \end{bmatrix} \in \mathbf{C}^{N\times(n+1)}. \tag{3.3}$$

3.3. Semidefinite representations. We now combine the observations in the previous two paragraphs to obtain SDP characterizations of nonnegative trigonometric polynomials. Let f be the trigonometric polynomial (3.1). Suppose $N \geq 2n+1$, W is defined as in (3.3), and $W_1 \in \mathbf{C}^{N\times n}$ is the matrix formed by the first n columns of W_{DFT}.

THEOREM 3.1. *f is nonnegative everywhere if and only if there exists an $X \in \mathbf{H}^{n+1}$ such that*

$$x = W^H \mathbf{diag}(WXW^H), \qquad X \succeq 0. \tag{3.4}$$

The result follows directly from the following fact: two vectors $x \in \mathbf{R} \times \mathbf{C}^n$ and $u \in \mathbf{C}^{n+1}$ satisfy

$$x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}) = |u_0 + u_1 e^{-j\omega} + \cdots + u_n e^{-jn\omega}|^2 \tag{3.5}$$

for all ω if and only if

$$x = \frac{1}{N} W^H \mathbf{diag}(Wuu^H W^H). \tag{3.6}$$

To see this, we simply note that the elements of $\mathbf{diag}(Wuu^H W^H)$ are the righthand side of (3.5) evaluated at $\omega = 2\pi k/N$, for $k = 0, 1, \ldots, N-1$. As we observed in §3.2, the inverse DFT of this vector gives the (unique) coefficients of the trigonometric polynomial that assumes those specified values. Therefore the coefficients x defined in (3.5) are given by (3.6). Since every nonnegative trigonometric polynomial can be expressed as (3.5), (3.4) holds with $X = (1/N)uu^H$.

Conversely, if (3.4) holds, then by factoring X as $X = (1/N)\sum_{k=0}^{n} u_k u_k^H$, with $u_k = (u_{k0}, u_{k1}, \ldots, u_{kn})$, we express f in the form

$$f(\omega) = \sum_{k=0}^{n} |u_{k0} + u_{k1}e^{-j\omega} + \cdots + u_{kn}e^{-jn\omega}|^2,$$

which shows $f(\omega) \geq 0$. This completes the proof of theorem 3.1.

THEOREM 3.2. *f is nonnegative on $[\alpha - \beta, \alpha + \beta]$ where $0 < \beta < \pi$ if and only if there exist $X_1 \in \mathbf{H}^{n+1}$, $X_2 \in \mathbf{H}^n$, such that*

$$x = W^H \left(\mathbf{diag}(W X_1 W^H) + d \circ \mathbf{diag}(W_1 X_2 W_1^H)\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0, \tag{3.7}$$

where $d \in \mathbf{R}^N$ has elements $d_k = \cos(2\pi k/N - \alpha) - \cos\beta$ for $k = 0, \ldots, N-1$.

The proof of this theorem is similar to the proof of theorem 3.1. We have

$$\begin{aligned} & x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}) \\ & = \left|\sum_{k=0}^{n} u_k e^{-jk\omega}\right|^2 + (\cos(\omega - \alpha) - \cos\beta)\left|\sum_{k=0}^{n-1} v_k e^{-jk\omega}\right|^2 \end{aligned} \tag{3.8}$$

for all ω if and only if

$$x = \frac{1}{N} W^H(\mathbf{diag}(W u u^H W^H) + d \circ \mathbf{diag}(W_1 v v^H W_1^H).$$

According to the extension of the Riesz-Fejér theorem mentioned in §3.1, if f is nonnegative on $[\alpha - \beta, \alpha + \beta]$, then it can be represented as (3.8), so (3.7) holds with $X_1 = (1/N)uu^H$, $X_2 = (1/N)vv^H$. Conversely, if (3.7) holds, then f can be expressed as a sum of functions of the form (3.8), so it is clearly nonnegative on $[\alpha - \beta, \alpha + \beta]$. This proves theorem 3.2.

The constraint (3.4) is better known in a different form [14, 4, 11]. Let E_i be the ith 'shift' matrix, *i.e.*, $E_i \in \mathbf{R}^{(n+1)\times(n+1)}$ with elements

$$E_{i,kl} = \begin{cases} 1 & k - l = i \\ 0 & \text{otherwise.} \end{cases}$$

It is easily seen that $E_i = (1/N)W^H \mathbf{diag}(w_i) W$ where W and w_i are defined in (3.3) with $N \geq 2n+1$. Therefore (3.4) holds if and only if

$$x_i = w_i^H \mathbf{diag}(W X W^H) = \mathbf{tr}(\mathbf{diag}(w_i)^H W X W^H) = N\,\mathbf{tr}(E_i^T X) = N \sum_{k-l=i} X_{kl}.$$

Hence the linear mapping $\mathcal{H} : \mathbf{H}^{n+1} \to \mathbf{R} \times \mathbf{C}^n$ defined by

$$\mathcal{H}(X) = \frac{1}{N} W^H \mathbf{diag}(W X W^H), \tag{3.9}$$

can also be expressed as

$$\mathcal{H}(X) = (\mathbf{tr}(E_0^T X), \mathbf{tr}(E_1^T X), \ldots, \mathbf{tr}(E_n^T X)). \tag{3.10}$$

We obtain the well-known result that $f(\omega) \geq 0$ if and only if there exists an $X \succeq 0$ such that $x_i = \sum_{k-l=i} X_{kl}$ for $i = 0, \ldots, n$.

The adjoint of $\mathcal{H}$ (with respect to the inner products $\Re(x^H z)$ on $\mathbf{R} \times \mathbf{C}^n$ and $\mathbf{tr}(XZ)$ on $\mathbf{H}^{n+1}$) can be derived using either one of the two expressions for $\mathcal{H}$. From (3.10),

$$\mathcal{H}^{\mathrm{adj}}(z) = \frac{1}{2} \begin{bmatrix} 2z_0 & \bar{z}_1 & \cdots & \bar{z}_n \\ z_1 & 2z_0 & \cdots & \bar{z}_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & \cdots & 2z_0 \end{bmatrix}, \tag{3.11}$$

the Hermitian Toeplitz matrix with first column $(z_0, z_1/2, \ldots, z_n/2)$. From (3.9),

$$\begin{aligned} \Re(z^H \mathcal{H}(X)) &= \frac{1}{N} \Re(z^H W^H \mathbf{diag}(W X W^H)) \\ &= \frac{1}{N} \Re(\mathbf{tr}(\mathbf{diag}(Wz)^H W X W^H)) \\ &= \frac{1}{N} \mathbf{tr}((W^H \mathbf{diag}(\Re(Wz)) W) X), \end{aligned}$$

so

$$\mathcal{H}^{\mathrm{adj}}(z) = \frac{1}{N} W^H \mathbf{diag}(\Re(Wz)) W.$$

Although it is not immediately clear that this is equal to the Toeplitz matrix (3.11), it is sufficient to note that the convolution of z with an arbitrary $y \in \mathbf{C}^{n+1}$ is given by

$$\frac{1}{N} W^H((Wz) \circ (Wy)) = \frac{1}{N} W^H \mathbf{diag}(Wz) W y.$$

The matrix $(1/N) W^H \mathbf{diag}(Wz) W$ is therefore the lower triangular Toeplitz matrix with $(z_0, z_1 \ldots, z_n)$ as its first column. Adding the complex conjugate transpose and dividing by two gives

$$\frac{1}{2N} W^H (\mathbf{diag}(Wz) + \mathbf{diag}(Wz)^H) W = \frac{1}{N} W^H \mathbf{diag}(\Re(Wz)) W,$$

so this is indeed the Hermitian Toeplitz matrix with first column $(z_0, z_1/2, \ldots, z_n/2)$.

4. Cosine polynomials. In this section we consider semidefinite formulations of the constraint

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_n \cos n\omega \geq 0, \qquad \omega \in [\alpha, \beta],$$

where $x \in \mathbf{R}^{n+1}$ and $0 \leq \alpha < \beta \leq \pi$. This is in fact a special case of the constraints considered in the previous section, since f is a trigonometric polynomial with real coefficients. For example, using theorem 3.1, we can say that $f(\omega) \geq 0$ for all ω if and only if

$$(x_0, x_1/2, \ldots, x_n/2) = W^H \mathbf{diag}(W X W^H)$$

for some $X \succeq 0$, where $N \geq 2n+1$ and W is formed by the first $n+1$ columns of the length-N DFT matrix. The purpose of this section is to show that simpler semidefinite parametrizations, using smaller matrices, can be obtained for cosine polynomials.

4.1. Sum-of-squares characterizations. Let f be a cosine polynomial of degree n, *i.e.*,

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_n \cos n\omega, \tag{4.1}$$

with $x \in \mathbf{R}^{n+1}$ and $x_n \neq 0$. If f is nonnegative on $[\alpha, \beta]$, where $0 \leq \alpha < \beta \leq \pi$, then it can be expressed as

$$f(\omega) = \begin{cases} g(\omega)^2 + (\cos\omega - \cos\beta)(\cos\alpha - \cos\omega)h(\omega)^2 & n \text{ even} \\ (\cos\omega - \cos\beta)g(\omega)^2 + (\cos\alpha - \cos\omega)h(\omega)^2 & n \text{ odd} \end{cases}$$

where g and h are cosine polynomials wtih $\deg(g) \leq \lfloor n/2 \rfloor$, $\deg(h) \leq \lfloor (n-1)/2 \rfloor$. This result can be derived from the characterization of nonnegative polynomials on $[-1, 1]$ (see §5.1), by making a change of variables $t = \cos\omega$.

If $\alpha = 0$, $\beta = \pi$, *i.e.*, f is nonnegative everywhere, these expressions can be simplified. If $n = 2m$, we have

$$\begin{aligned} f(\omega) &= g(\omega)^2 + (1 - \cos^2\omega)h(\omega)^2 \\ &= g(\omega)^2 + (\sin\omega)^2 h(\omega)^2 \\ &= g(\omega)^2 + \tilde{h}(\omega)^2, \end{aligned} \tag{4.2}$$

where $\tilde{h}$ is of the form $\tilde{h}(\omega) = v_1 \sin\omega + v_2 \sin 2\omega + \cdots + v_m \sin m\omega$. This follows from the fact that the function $\sin k\omega / \sin\omega$ is a cosine polynomial of degree $k - 1$.

If $n = 2m + 1$, we have

$$\begin{aligned} f(\omega) &= (\cos\omega + 1)g(\omega)^2 + (1 - \cos\omega)h(\omega)^2 \\ &= 2(\cos(\omega/2))^2 g(\omega)^2 + 2(\sin(\omega/2))^2 h(\omega)^2 \\ &= \tilde{g}(\omega)^2 + \tilde{h}(\omega)^2, \end{aligned} \tag{4.3}$$

where $\tilde{g}$ and $\tilde{h}$ have the form

$$\tilde{g}(\omega) = \sum_{k=0}^{m} u_k \cos((k + 1/2)\omega), \qquad \tilde{h}(\omega) = \sum_{k=0}^{m} v_k \sin((k + 1/2)\omega).$$

This follows from the fact that $\cos((k+1/2)\omega)/\cos(\omega/2)$ and $\sin((k+1/2)\omega)/\sin(\omega/2)$ are cosine polynomials of degree k.

4.2. Discrete cosine transform. The matrices

$$V_{\mathrm{DCT}} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \cos(\pi/N) & \cdots & \cos((N-1)\pi/N) & \cos(\pi) \\ 1 & \cos(2\pi/N) & \cdots & \cos(2(N-1)\pi/N) & \cos(2\pi) \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos(\pi) & \cdots & \cos((N-1)\pi) & \cos(N\pi) \end{bmatrix} \in \mathbf{S}^{N+1}$$

and

$$W_{\mathrm{DCT}} = \frac{2}{N} D V_{\mathrm{DCT}} D,$$

where $D = \mathbf{diag}(1/2, 1, 1, \ldots, 1, 1, 1/2)$, are inverses:

$$W_{\mathrm{DCT}} V_{\mathrm{DCT}} = I \tag{4.4}$$

(see, for example, [7, page 124]). The mapping $V_{\rm DCT}Du$ is sometimes referred as the discrete cosine transform (DCT) of u.

Suppose $N \geq n$, and let W and V be the matrices formed by taking the first $n+1$ columns of $W_{\rm DCT}$ and $V_{\rm DCT}$, respectively. These matrices satisfy $W^TV = I$ as a consequence of (4.4) and the symmetry of $W_{\rm DCT}$. The matrix V maps the coefficients $x_0, \ldots, x_n$ of the cosine polynomial (4.1) to its values at $\omega = k\pi/N$, $k = 0, \ldots, N$. Multiplying with W^T maps these sample values to the coefficients. In other words, if $y = (f(0), f(\pi/N), \ldots, f((N-1)\pi/N), f(\pi))$, then

$$y = Vx, \qquad x = W^Ty.$$

4.3. Semidefinite representations. We now use the DCT and the sum-of-squares theorems in §4.1 to express constraints on a cosine polynomial

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_n \cos n\omega$$

in semidefinite form. Assume $N \geq n$ and define $\omega_N = \pi/N$. As in §4.2, $W \in \mathbf{R}^{(N+1)\times(n+1)}$ denotes the matrix formed with the first $n+1$ columns of $W_{\rm DCT}$.

THEOREM 4.1. *$f(\omega) \geq 0$ on $[\alpha, \beta]$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$ and $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^T\left(d_1 \circ \mathbf{diag}(V_1X_1V_1^T) + d_2 \circ \mathbf{diag}(V_2X_2V_2^T)\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0, \tag{4.5}$$

where $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$ and $d_1, d_2 \in \mathbf{R}^{N+1}$ are defined as

$$d_{1,k} = \begin{cases} 1 & n \text{ even} \\ \cos k\omega_N - \cos\beta & n \text{ odd}, \end{cases}$$

$$d_{2,k} = \begin{cases} (\cos k\omega_N - \cos\beta)(\cos\alpha - \cos k\omega_N) & n \text{ even} \\ \cos\alpha - \cos k\omega_N & n \text{ odd}, \end{cases}$$

for $k = 0, \ldots, N$. The columns of $V_1 \in \mathbf{R}^{(N+1)\times(m_1+1)}$ and $V_2 \in \mathbf{R}^{(N+1)\times(m_2+1)}$ are the first m_1+1, resp. m_2+1, columns of $V_{\rm DCT}$.

We prove the theorem for n even ($n = 2m$). By the sum-of-squares characterization in §4.1, if f is nonnegative on $[\alpha, \beta]$, then it can be expressed as

$$f(\omega) = g(\omega)^2 + (\cos\omega - \cos\beta)(\cos\alpha - \cos\omega)h(\omega)^2 \tag{4.6}$$

for some cosine polynomials

$$g(\omega) = \sum_{k=0}^{m} u_k \cos k\omega, \qquad h(\omega) = \sum_{k=0}^{m-1} v_k \cos k\omega.$$

From §4.2, we can express the righthand side of (4.6) as a cosine polynomial by computing the values at $\omega = k\pi/N$, $k = 0, \ldots, N$, which gives the vectors

$$d_1 \circ \mathbf{diag}(V_1uu^TV_1^T) + d_2 \circ \mathbf{diag}(V_2vv^TV_2^T),$$

and then multiplying on the left with W^T. In other words, (4.6) is equivalent to

$$x = W^T\left(d_1 \circ \mathbf{diag}(V_1uu^TV_1^T) + d_2 \circ \mathbf{diag}(V_2vv^TV_2^T)\right).$$

Therefore (4.5) holds with $X_1 = uu^T$ and $X_2 = vv^T$. Conversely, if (4.5) holds, with X_1 and X_2 of rank greater than two, then f is a sum of cosine polynomials that are nonnegative on $[\alpha, \beta]$, so it is also nonnegative.

If $\alpha = 0$ and $\beta = \pi$, we can start from (4.2) and (4.3), and express the semidefinite constraints in a slightly simpler form.

THEOREM 4.2. *$f(\omega) \geq 0$ everywhere if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$, $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^T \left(\mathbf{diag}(V_1 X_1 V_1^T) + \mathbf{diag}(V_2 X_2 V_2^T)\right), \qquad X_1 \succeq 0, \qquad X_2 \succeq 0, \tag{4.7}$$

where $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$. If n is even, we define $V_1 \in \mathbf{R}^{(N+1)\times(m_1+1)}$ as the matrix formed by the first m_1+1 columns of V_{DCT} and

$$V_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \sin(\omega_N) & \sin(2\omega_N) & \cdots & \sin(m\omega_N) \\ \sin(2\omega_N) & \sin(4\omega_N) & \cdots & \sin(2m\omega_N) \\ \vdots & \vdots & & \vdots \\ \sin(N\omega_N) & \sin(2N\omega_N) & \cdots & \sin(mN\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_2+1)}.$$

If n is odd, we define V_1 and V_2 as

$$V_1 = \begin{bmatrix} 1 & \cdots & 1 \\ \cos(\omega_N/2) & \cdots & \cos((m+1/2)\omega_N) \\ \cos(\omega_N) & \cdots & \cos(2(m+1/2)\omega_N) \\ \vdots & & \vdots \\ \cos(N\omega_N/2) & \cdots & \cos(N(m+1/2)\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_1+1)}$$

$$V_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \sin(\omega_N/2) & \cdots & \sin((m+1/2)\omega_N) \\ \sin(\omega_N) & \cdots & \sin(2(m+1/2)\omega_N) \\ \vdots & & \vdots \\ \sin(N\omega_N/2) & \cdots & \sin(N(m+1/2)\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_2+1)}.$$

Note that the matrices X_1 and X_2 in the constraints (4.5) and (4.7) have dimension roughly $n/2$, as opposed to the constraints for general trigonometric polynomials of degree n given in §3, which involve matrix variables of size n. It is also interesting to note that the matrices V_1, V_2 and W are orthogonal or nearly orthogonal (*i.e.*, have a condition number close to 1).

4.4. Example: Linear-phase Nyquist filter. We consider the lowpass filter design problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leq H(\omega) \leq t, \quad \omega_{\mathrm{s}} \leq \omega \leq \pi, \end{array} \tag{4.8}$$

in which H is the frequency response of a linear-phase *Nyquist-M* filter [32, §4.6]:

$$H(\omega) = h_0 + h_1 \cos\omega + \cdots + h_n \cos n\omega$$

with

$$h_0 = 1/M, \qquad h_{kM} = 0, \quad k = 1, 2, \ldots, \lfloor n/M \rfloor. \tag{4.9}$$

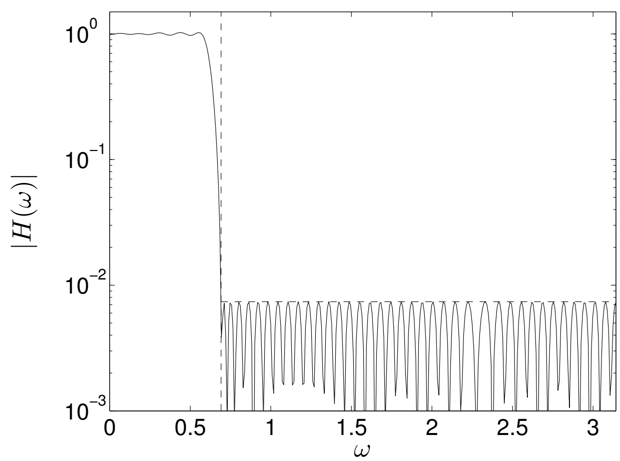

FIG. 4.1. *Frequency response of a linear-phase Nyquist-5 filter of length 51 and stopband edge* $\omega_s = 1.1\pi/5 = 0.69$.

The variables in (4.8) are t and the $n - \lfloor n/M \rfloor$ coefficients h_i that are not determined by (4.9). Since H is a cosine polynomial, we can apply theorem 4.1 to formulate this problem as an SDP

$$\begin{array}{ll} \text{min.} & t \\ \text{s.t.} & h + te_0 = W^T\left(d_1 \circ \mathbf{diag}(V_1 X_1 V_1^T) + d_2 \circ \mathbf{diag}(V_2 X_2 V_2^T)\right) \\ & -h + te_0 = W^T\left(d_1 \circ \mathbf{diag}(V_1 X_3 V_1^T) + d_2 \circ \mathbf{diag}(V_2 X_4 V_2^T)\right) \\ & X_1 \succeq 0, \quad X_2 \succeq 0, \quad X_3 \succeq 0 \quad X_4 \succeq 0, \end{array} \tag{4.10}$$

where $e_0 = (1, 0, \ldots, 0) \in \mathbf{R}^{n+1}$ and W, d_1, d_2, V_1, V_2 are defined as in theorem 4.1, with $\alpha = \omega_s$, $\beta = \pi$. The variables are t, the $n - \lfloor n/M \rfloor$ unknown entries of $h = (h_0, h_1, \ldots, h_n)$ and four symmetric matrices X_i, which have dimension roughly $n/2$. Figure 4.1 shows an example with $n = 50$, $M = 5$, $\omega_s = 1.1\pi/M$.

5. Real polynomials.

5.1. Sum-of-squares characterizations.

Let f be a polynomial of degree n with real coefficients. If f is nonnegative on $\mathbf{R}$, then n is even and f can be expressed as

$$f(t) = g(t)^2 + h(t)^2, \tag{5.1}$$

where $\deg(g) \le n/2$ and $\deg(h) \le n/2$. If f is nonnegative on $[a, \infty)$, then f can be expressed as

$$f(t) = g(t)^2 + (t - a)h(t)^2,$$

where $\deg(g) \le \lfloor n/2 \rfloor$ and and $\deg(h) \le \lfloor (n-1)/2 \rfloor$. Finally, if f is nonnegative on $[a, b]$, where $a < b$, then it can be expressed as

$$f(t) = \begin{cases} g(t)^2 + (t-a)(b-t)h(t)^2 & n \text{ even} \\ (t-a)g(t)^2 + (b-t)h(t)^2 & n \text{ odd}, \end{cases} \tag{5.2}$$

where g and h are polynomials with $\deg(g) \le \lfloor n/2 \rfloor$ and $\deg(h) \le \lfloor (n-1)/2 \rfloor$. This last result is known as the *Markov-Lukács theorem* [29, §1.21], [20, §3.2].

5.2. Discrete polynomial transforms. Let $p_k(t)$, $k = 0, 1, \ldots$, be a system of orthogonal and normalized polynomials on a bounded or unbounded interval $I \subseteq \mathbf{R}$, with respect to a nonnegative weight function $w(t)$:

$$\int_I p_k(t)p_l(t)w(t)\,dt = \begin{cases} 0 & k \neq l \\ 1 & k = l. \end{cases}$$

The kth polynomial p_k has degree k, with a positive leading coefficient a_k. It is well-known that orthogonal polynomials satisfy a three-term recursion

$$p_{k+1}(t) = (\alpha_k t - \beta_k)p_k(t) - \gamma_k p_{k-1}(t), \tag{5.3}$$

where we define $p_{-1}(t) = 0$. The coefficients α_k, γ_k are positive and satisfy

$$\alpha_k = \frac{a_{k+1}}{a_k} > 0, \qquad \frac{\alpha_k \gamma_{k+1}}{\alpha_{k+1}} = 1. \tag{5.4}$$

The recursion (5.3) for $k = 0, \ldots, N$ can be written in matrix-vector form as

$$tp(t) = Jp(t) + (1/\alpha_N)p_{N+1}(t)e_N, \tag{5.5}$$

where $p(t) = (p_0(t), p_1(t), \ldots, p_N(t))$, $e_N = (0, 0, \ldots, 0, 1) \in \mathbf{R}^{N+1}$, and

$$J = \begin{bmatrix} \beta_0/\alpha_0 & 1/\alpha_0 & 0 & \cdots & 0 & 0 \\ \gamma_1/\alpha_1 & \beta_1/\alpha_1 & 1/\alpha_1 & \cdots & 0 & 0 \\ 0 & \gamma_2/\alpha_2 & \beta_2/\alpha_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{N-1}/\alpha_{N-1} & 1/\alpha_{N-1} \\ 0 & 0 & 0 & \cdots & \gamma_N/\alpha_N & \beta_N/\alpha_N \end{bmatrix}.$$

It follows from (5.4) that J is symmetric. Another well-known property of orthogonal polynomials is that p_k has exactly k distinct roots in the interior of I [10, page 236]. From (5.5) we see that this implies

$$\lambda_i p(\lambda_i) = Jp(\lambda_i), \quad i = 0, \ldots, N,$$

where λ_0, λ_1, $\ldots$, λ_N are the roots of p_{N+1}. In other words $p(\lambda_i)$ is an eigenvector of J with eigenvalue λ_i [15].

These properties provide an efficient method for computing the matrix

$$V_{\mathrm{DPT}} = \begin{bmatrix} p_0(\lambda_0) & p_1(\lambda_0) & \cdots & p_N(\lambda_0) \\ p_0(\lambda_1) & p_1(\lambda_1) & \cdots & p_N(\lambda_1) \\ \vdots & \vdots & & \vdots \\ p_0(\lambda_N) & p_1(\lambda_N) & \cdots & p_N(\lambda_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(N+1)}$$

directly from the coefficients α_k, β_k, γ_k in the recursion (5.3). Let

$$J = Q\,\mathbf{diag}(\lambda)Q^T$$

be the eigenvalue decomposition of J, with normalized eigenvectors ($QQ^T = Q^TQ = I$) and the signs in the first row of Q chosen to be positive. The ith column of Q is then a positive multiple of $p(\lambda_i)$, and therefore

$$V_{\mathrm{DPT}} = DQ^T$$

with D positive diagonal. The matrix D is easily determined by dividing the first column of $V_{\rm DPT}$, which is a constant $p_0(t) = (\int w(t)\,dt)^{-1/2}$, by the elements in the first row q_1^T of Q: $D = p_0(t)\,\mathbf{diag}(q_1)^{-1}$. It follows that

$$V_{\rm DPT}^T D^{-2} V_{\rm DPT} = I,$$

so the matrix

$$W_{\rm DPT} = D^{-1}Q^T = D^{-2}V_{\rm DPT} \tag{5.6}$$

satisfies $W_{\rm DPT}^T V_{\rm DPT} = I$. The matrices $V_{\rm DPT}$ and $W_{\rm DPT}$ thus define a pair of forward and inverse 'discrete polynomial transforms' [7, §8.5].

Now suppose $N \geq n$, and let W and V be the matrices formed by the first $n+1$ columns of $W_{\rm DPT}$ and $V_{\rm DPT}$. Since $V_{\rm DPT}$ and $W_{\rm DPT}^T$ are inverses, we have $W^TV = I$. The linear transformations Vx and W^Ty map the coefficients of the polynomial

$$f(t) = x_0p_0(t) + x_1p_1(t) + \cdots + x_np_n(t)$$

to $N+1$ values at $\lambda_0,\ldots,\lambda_N$ and vice-versa: If

$$y = (f(\lambda_0), f(\lambda_1), \ldots, f(\lambda_N))$$

then $y = Vx$ and $x = W^Ty$.

5.3. Semidefinite representations. We can apply the discrete transform associated with the orthogonal polynomials p_k, combined with the sum-of-squares results in §5.1, to derive LMI conditions for nonnegativity of the polynomial

$$f(t) = x_0p_0(t) + x_1p_1(t) + \cdots + x_np_n(t).$$

Assume $N \geq n$. Let $W \in \mathbf{R}^{(N+1)\times(n+1)}$ be the matrix formed by the first $n+1$ columns of $W_{\rm DPT}$ in (5.6), and let $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_N)$ be the vector of zeros of p_{N+1}.

THEOREM 5.1. *$f(t) \geq 0$ for $t \in \mathbf{R}$ if and only if n is even and there exists an $X \in \mathbf{S}^{n/2+1}$ such that*

$$x = W^T\,\mathbf{diag}(V_1XV_1^T), \qquad X \succeq 0.$$

Here V_1 is the matrix formed by the first $n/2+1$ columns of $V_{\rm DPT}$.

THEOREM 5.2. *$f(t) \geq 0$ on $[a,\infty)$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$ and $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^T\left(\mathbf{diag}(V_1X_1V_1^T) + (\lambda - a)\circ\mathbf{diag}(V_2X_2V_2^T)\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0.$$

Here $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$, and V_1 and V_2 are the matrices formed by the first m_1+1, respectively m_2+1, columns of $V_{\rm DPT}$.

THEOREM 5.3. *$f(t) \geq 0$ on $[a,b]$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$, $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^T\left(d_1\circ\mathbf{diag}(V_1X_1V_1^T) + d_2\circ\mathbf{diag}(V_2X_2V_2^T)\right), \qquad X_1 \succeq 0, \qquad X_2 \succeq 0.$$

Here $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$, and V_1 and V_2 are the matrices formed by the first m_1+1, resp. m_2+1, columns of $V_{\rm DPT}$. The vectors $d_1, d_2 \in \mathbf{R}^{N+1}$ are defined as

$$d_1 = \begin{cases} \mathbf{1} & n \text{ even} \\ \lambda - a\mathbf{1} & n \text{ odd}, \end{cases} \qquad d_2 = \begin{cases} (\lambda - a\mathbf{1})\circ(b\mathbf{1} - \lambda) & n \text{ even} \\ b\mathbf{1} - \lambda & n \text{ odd}. \end{cases}$$

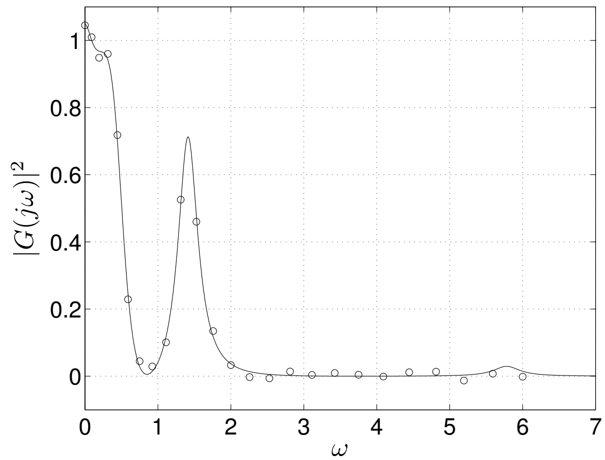

FIG. 5.1. *Minimax magnitude fit of a rational transfer function to 25 data points.*

The proofs follow exactly the same pattern as in §3.3 and §4.3, and are omitted.

There exist several other interesting choices for the matrices V_1, V_2, and W. First, we can define V_1 and V_2 as the first columns of the matrix Q^T (instead of the first columns of $V_{\text{DPT}} = DQ^T$), if we change the definition of W accordingly and construct W from the first columns of DQ^T. With this choice, V_1 and V_2 are orthogonal. Alternatively, we can define W to be the first columns of the matrix Q^T, and redefine V_1 and V_2 as the first columns of $D^{1/2}Q^T$. With this choice W is orthogonal.

Second, we can note that the basis polynomials used in the definitions of V_1 and V_2 need not be the same as in the definition of W. This follows from the fact that in (5.1)–(5.2), we can use a different basis to represent the polynomials f, g and h. We could therefore define V_1 and V_2 as generalized Vandermonde matrices with k, l elements $q_l(t_k)$ where t_k are the zeros of p_N, and q_0, q_1, ..., is any polynomial basis. This is equivalent to replacing the matrices V_k by V_kT_k where T_k is nonsingular. In particular, we can replace V_1 and V_2 with orthogonal matrices that have the same column spaces.

5.4. Example: Minimax magnitude fit of rational transfer function. We consider the problem of fitting the magnitude of a rational transfer function

$$G(s) = \frac{a_0 + a_1 s + \cdots + a_n s^n}{b_0 + b_1 s + \cdots + b_m s^m}$$

to data points, *i.e.*, choosing the (real) coefficients a_i, b_i so that $|G(j\omega_k)|^2 \approx \gamma_k$ for $k = 1, \ldots, K$, where ω_k and γ_k are given. Using a minimax criterion and introducing an auxiliary variable δ we can formulate this problem as

$$\begin{array}{ll} \text{minimize} & \delta \\ \text{minimize} & -\delta \leq |G(j\omega_k)|^2 - \gamma_k \leq \delta, \quad k = 1, \ldots, K. \end{array}$$

Figure 5.1 shows an example with $n = 6$, $m = 8$ and $K = 25$.

This problem can be posed as a quasiconvex optimization problem. We first express the magnitude squared of the transfer function as

$$|G(j\omega)|^2 = \frac{f_1(\omega^2)}{f_2(\omega^2)}$$

where f_1 and f_2 are the real polynomials

$$f_1(t) = a_{\rm e}(t)^2 + t a_{\rm o}(t)^2, \qquad f_2(t) = b_{\rm e}(t)^2 + t b_{\rm o}(t)^2, \tag{5.7}$$

with

$$a_{\rm e}(t) = \sum_{k=0}^{\lfloor n/2 \rfloor} a_{2k}(-t)^k, \qquad a_{\rm o}(t) = \sum_{k=0}^{\lfloor (n-1)/2 \rfloor} a_{2k+1}(-t)^k,$$

$$b_{\rm e}(t) = \sum_{k=0}^{\lfloor m/2 \rfloor} b_{2k}(-t)^k, \qquad b_{\rm o}(t) = \sum_{k=0}^{\lfloor (m-1)/2 \rfloor} b_{2k+1}(-t)^k.$$

Clearly $f_1(t) \geq 0$ and $f_2(t) \geq 0$ for $t \geq 0$. Conversely, if f_1 and f_2 are nonnegative on the nonnegative real axis, then by the result mentioned in §5.1, they can be expressed as (5.7). The fitting problem is therefore equivalent to

$$\begin{array}{ll} \text{min.} & \delta \\ \text{s.t.} & (\gamma_k - \delta) f_2(\omega_k^2) \leq f_1(\omega_k^2) \leq (\gamma_k + \delta) f_2(\omega_k^2), \quad k = 1, \ldots, K \\ & f_1(t) \geq 0, \quad f_2(t) \geq 0 \quad \text{for } t \geq 0. \end{array} \tag{5.8}$$

The variables are δ and the coefficients of the polynomials

$$f_1(t) = x_0 p_0(t) + x_1 p_1(t) + \cdots + x_n p_n(t), \qquad f_2(t) = p_0(t) + y_1 p_1(t) + \cdots + y_m p_m(t)$$

for some choice of orthogonal basis polynomials $p_k(t)$. We normalize the first coefficient of f_2 to rule out the trivial solution $f_1(t) = f_2(t) = 0$. (Alternatively, one might prefer to replace $f_2(t) \geq 0$ with $f_2(t) \geq \epsilon$ for some small positive ϵ, which would also ensure that there are no poles on the imaginary axis.)

Problem (5.8) can be solved via bisection on δ. In each bisection step we fix δ and determine whether the constraints are feasible or not. This feasibility problem can be cast as an SDP feasibility problem

$$\begin{array}{c} (\gamma_k - \delta) f_2(\omega_k^2) \leq f_1(\omega_k^2) \leq (\gamma_k + \delta) f_2(\omega_k)^2, \quad k = 1, \ldots, K \\ x = W^T \left(\mathbf{diag}(V_1 X_1 V_1^T) + \lambda \circ \mathbf{diag}(V_2 X_2 V_2^T) \right) \\ y = \tilde{W}^T \left(\mathbf{diag}(\tilde{V}_1 \tilde{X}_1 \tilde{V}_1^T) + \tilde{\lambda} \circ \mathbf{diag}(\tilde{V}_2 \tilde{X}_2 \tilde{V}_2^T) \right) \\ X_1 \succeq 0, \quad X_2 \succeq 0, \quad \tilde{X}_1 \succeq 0, \quad \tilde{X}_2 \succeq 0, \end{array} \tag{5.9}$$

where $x = (x_0, x_1, \ldots, x_n)$ and $y = (1, y_1, \ldots, y_m)$. The variables are x_k, y_k, and the matrices X_i and $\tilde{X}_i$. The matrices W, V_1, V_2 and the vector λ are defined as in theorem 5.2 with $a = 0$. The matrices $\tilde{W}$, $\tilde{V}_1$, $\tilde{V}_2$ and $\tilde{\lambda}$ are defined similarly but with n replaced by m.

6. Numerical examples. The SDP characterizations of nonnegative polynomials derived in the previous sections can be expressed in the following common form. A (trigonometric, cosine, real) polynomial with coefficients x is nonnegative on a given interval if and only if there exist Hermitian matrices X_k such that

$$x = \sum_{k=1}^{s} A_k \,\mathbf{diag}(C_k X_k C_k^H), \qquad X_k \succeq 0, \qquad k = 1, \ldots, s.$$

In the case of cosine polynomials or real polynomials, the matrices A_k, C_k and the variables x and X_k are real. This representation allows us to formulate a wide variety of optimization problems involving polynomials as SDPs of the form

$$\begin{array}{ll} \text{minimize} & c^T y \\ \text{subject to} & \sum_{k=1}^{s_i} A_{ik} \,\mathbf{diag}(C_{ik} X_{ik} C_{ik}^H) + B_i y = b_i, \quad i = 1, \ldots, L \\ & X_{ik} \succeq 0, \quad k = 1, \ldots, s_i, \quad i = 1, \ldots, L. \end{array} \tag{6.1}$$

The variables are $y \in \mathbf{R}^p$ and the Hermitian matrices X_{ik}. Each of the L constraints expresses a nonnegativity condition on a polynomial with coefficients $b_i - B_i y$.

The SDP (6.1) is a special case of (2.5) or (2.13) if we interpret X as a block-diagonal matrix with diagonal blocks X_{ik}, and define A, C, and B as block matrices constructed from A_{ik}, C_{ik}, and B_i. In this section we present numerical results for a primal-dual interior-point method that uses the fast method for solving the Newton equations described in §2.1. We first provide some details of the implementation.

6.1. Implementation. All examples are instances of the SDP (6.1), with real data and variables. Applying the method of §2.1 to an SDP with block-diagonal structure (6.1) leads to a reduced Newton system (2.12)

$$\begin{bmatrix} H_1 & 0 & \cdots & 0 & B_1 \\ 0 & H_2 & \cdots & 0 & B_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & H_L & B_L \\ B_1^T & B_2^T & \cdots & B_L^T & 0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \vdots \\ \Delta z_L \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{3,1} \\ r_{3,2} \\ \vdots \\ r_{3,L} \\ r_2 \end{bmatrix}, \tag{6.2}$$

where

$$H_i = \sum_{k=1}^{s_i} A_{ik} \,\mathbf{sqr}(C_{ik} T_{ik} C_{ik}^T) A_{ik}^T.$$

When solving (6.2), we can exploit the 'block-arrow' structure by first eliminating the variables Δz_i and then solving a positive definite set of linear equations in the variables Δy:

$$\left(\sum_{i=1}^{L} B_i^T H_i^{-1} B_i \right) \Delta y = \sum_{i=1}^{L} B_i^T H_i^{-1} r_{3,i} - r_2. \tag{6.3}$$

From the solution Δy we obtain Δz_i by solving $H_i \Delta z_i = r_{3,i} - B_i \Delta y$.

In the numerical experiments described below we implemented this idea as follows. We compute the Hadamard products $\mathbf{sqr}(C_{ik} T_{ik} C_{ik}^T)$ and factor them as

$$\mathbf{sqr}(C_{ik} T_{ik} C_{ik}^T) = V_{ik} V_{ik}^T.$$

The eigenvalue decomposition is used for this purpose, since the matrix $\mathbf{sqr}(C_{ik}T_{ik}C_{ik}^T)$ is often rank-deficient. We then factor the matrices

$$H_i = \sum_{k=1}^{s_i} A_{ik}V_{ik}V_{ik}^T A_{ik}^T$$

as $H_i = R_i^T R_i$ via QR factorizations of the matrices

$$\left[\begin{array}{cccc} A_{i1}V_{i1} & A_{i2}V_{i2} & \cdots & A_{ir_i}V_{ir_i} \end{array}\right]^T = Q_i R_i.$$

This is more stable than using a Cholesky factorization of H_i, since it allows us to compute the triangular factors R_i without explicitly forming H_i. The equation (6.3) now reduces to

$$\left(\sum_{i=1}^{L} B_i^T R_i^{-1} R_i^{-T} B_i\right) \Delta y = \sum_{i=1}^{L} B_i^T R_i^{-1} R_i^{-T} r_{3,i} - r_2.$$

To improve the numerical stability, we again avoid forming the coefficient matrix and use a QR factorization

$$\left[\begin{array}{cccc} B_1^T R_1^{-1} & B_2^T R_2^{-1} & \cdots & B_L^T R_L^{-1} \end{array}\right]^T = QR$$

instead. Given Q and R we can find Δy by solving

$$R\Delta y = Q^T \tilde{r}_3 - R^{-T} r_2,$$

where

$$\tilde{r}_3 = \left[\begin{array}{cccc} (R_1^{-T} r_{3,1})^T & (R_2^{-T} r_{3,2})^T & \cdots & (R_L^{-T} r_{3,L})^T \end{array}\right]^T.$$

Except for the algorithm used for solving the Newton equations, the code is a rudimentary implementation of an SDPT3-style path-following method [30, 31], following the outline given in the appendix of [33]. Infeasible starting points are used: we take $y = 0$, $X_{ik} = I$ in the primal problem; in the dual problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^L b_i^T z_i \\ \text{subject to} & C_{ik}^T \,\mathbf{diag}(A_{ik}^T z_i) C_{ik} + Z_{ik} = 0, \quad Z_i \succeq 0, \quad k = 1, \ldots, s_i, \quad i = 1, \ldots, L \\ & \sum_{i=1}^L B_i^T z_i = c \end{array}$$

we take $Z_{ik} = I$ and for z_i the least-norm solution of the last equality constraint. The stopping criterion is based on the following quantities.

- The duality gap

$$\eta_{\mathrm{abs}} = \sum_{i=1}^{L} \sum_{k=1}^{s_i} \mathbf{tr}(X_{ik} Z_{ik}).$$

  (This is only truly the duality gap when the primal and dual iterates are feasible.)
- The relative duality gap

$$\eta_{\mathrm{rel}} = \begin{cases} -\eta_{\mathrm{abs}}/c^T y & c^T y < 0 \\ \eta_{\mathrm{abs}}/\sum_i b_i^T z_i & \sum_i b_i^T z_i > 0 \\ \infty & \text{otherwise.} \end{cases}$$

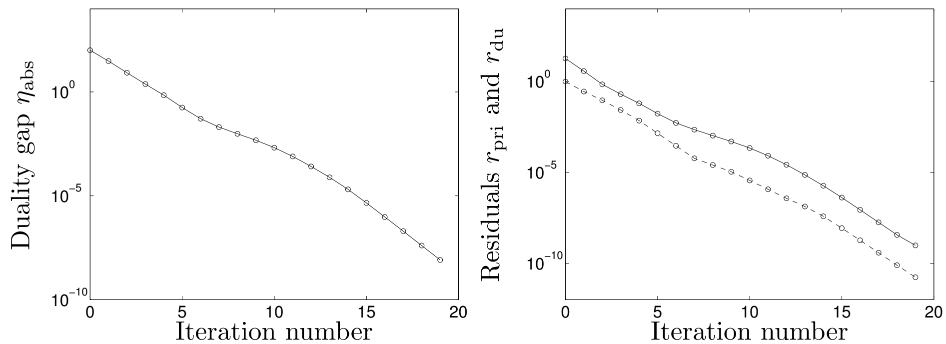

FIG. 6.1. *Progress of the primal-dual method for the design of a lowpass Nyquist-5 filter. The left plot shows the duality gap versus iteration number. The right plot shows the primal residual (solid line) and the dual residual (dashed line).*

- The primal residual

$$r_{\rm pri} = \max_{i=1,\ldots,L} \frac{\|b_i - B_i y - \sum_{k=1}^{s_i} A_{ik}\,\mathbf{diag}(C_{ik}X_{ik}C_{ik}^T)\|_2}{\max\{1, \|b_i\|_2\}}.$$

- The dual residual

$$r_{\rm du} = \max\left\{\frac{\|c - \sum_{i=1}^L B_i^T z_i\|_2}{\max\{1, \|c\|_2\}},\ \max_{i,k} \|S_{ik} + C_{ik}^T\,\mathbf{diag}(A_{ik}^T z_i) C_{ik}\|_2\right\}.$$

In these expressions, $\|\cdot\|_2$ denotes the Euclidean norm for vectors, and the matrix norm (maximum singular value norm) for matrices. The algorithm terminates if

$$r_{\rm pri} \le \epsilon_{\rm feas} \quad \text{and} \quad r_{\rm du} \le \epsilon_{\rm feas} \quad \text{and} \quad (\eta_{\rm abs} \le \epsilon_{\rm gap} \quad \text{or} \quad \eta_{\rm rel} \le \epsilon_{\rm gap})$$

where $\epsilon_{\rm feas} = 10^{-7}$ and $\epsilon_{\rm gap} = 10^{-8}$. The code was implemented in Matlab version 6.5.1 on a 2.4GHz Pentium IV PC with 1GB of memory.

6.2. Linear-phase FIR filter design. We first illustrate the behavior of the algorithm with the example problem of §4.4. Figure 6.1 shows the progres of the algorithm applied to the SDP (4.10) with the same parameters as used for figure 4.1. The algorithm terminates after 19 iterations, with a CPU time of 0.05 seconds per iteration.

6.3. Minimax magnitude fit of transfer function. The example in §5.4 was solved by bisection on δ. The optimal value of δ was computed with an absolute accuracy of 10^{-5}. We used the basis of Laguerre polynomials to construct the SDP constraints (5.9). The feasibility problems (for fixed δ) were solved by applying the interior-point method to the 'phase-I' problem

$$\begin{array}{ll} \text{min.} & u \\ \text{s.t.} & (\gamma_k - \delta) f_2(\omega_k^2) - u \le f_1(\omega_k^2) \le (\gamma_k + \delta) f_2(\omega_k)^2 + u, \quad k = 1, \ldots, K \\ & x = W^T\left(\mathbf{diag}(V_1 X_1 V_1^T) + \lambda \circ \mathbf{diag}(V_2 X_2 V_2^T)\right) \\ & y = \tilde{W}^T\left(\mathbf{diag}(\tilde{V}_1 \tilde{X}_1 \tilde{V}_1^T) + \tilde{\lambda} \circ \mathbf{diag}(\tilde{V}_2 \tilde{X}_2 \tilde{V}_2^T)\right) \\ & X_1 \succeq 0, \quad X_2 \succeq 0, \quad \tilde{X}_1 \succeq 0, \quad \tilde{X}_2 \succeq 0, \end{array} \tag{6.4}$$

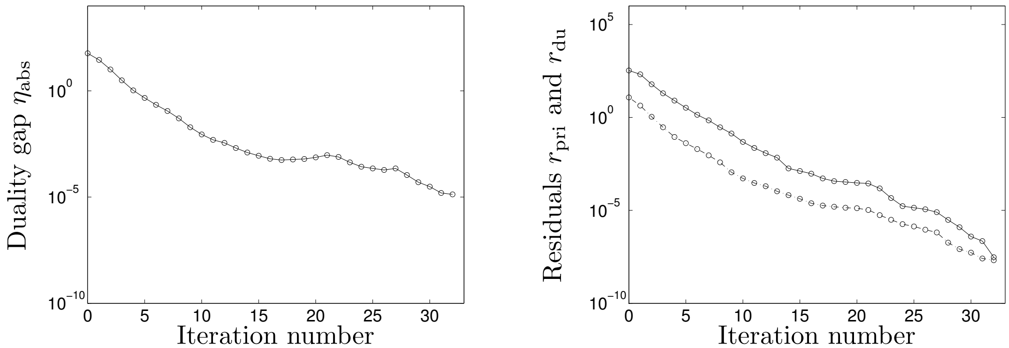

FIG. 6.2. *Progress of the primal-dual method applied to the phase-I problem in the last bisection step for computing the function in figure 5.1. The left plot shows the duality gap versus iteration number. The right plot shows the primal residual (solid line) and the dual residual (dashed line).*

with variables u, x, y, X_i and $\tilde{X}_i$.

Figure 6.2 shows the convergence of the primal-dual path-following method applied to the SDP (6.4) in the final bisection step. Although a primal feasible point for problem (6.4) is known, the algorithm was started at the default infeasible starting points. Instead of using the stopping criterion based on the duality gap described in §6.1, we terminated the interior-point algorithm as soon as the sign of the optimal value of (6.4) was known.

We observed that the convergence of the algorithm for this example problem was much more sensitive to the choice of problem parameters than for the other numerical examples. While the stability of our interior-point implementation certainly leaves room for improvement, optimization problems over real polynomials on unbounded intervals appear to be much more difficult to solve than problems with cosine polynomials.

6.4. Magnitude FIR filter design. The next example is a family of a lowpass filter design problem similar to examples described in [1] and [8]. The design variables are the (real) filter coefficients h_i of an FIR filter of length $n+1$, with transfer function

$$H(\omega) = h_0 + \sum_{k=0}^{n} h_k e^{-jk\omega}.$$

The objective is to minimize the stopband energy

$$\int_{\omega_{\text{s}}}^{\pi} |H(\omega)|^2 \, d\omega.$$

The constraints include upper and lower bounds on the filter magnitude $|H(\omega)|$ in the passband, and an upper bound on the magnitude in the stopband:

$$1/\delta_{\text{p}} \le |H(\omega)|^2 \le \delta_{\text{p}}, \quad 0 \le \omega \le \omega_{\text{p}}, \qquad |H(\omega)|^2 \le \delta_{\text{s}}, \quad \omega_{\text{s}} \le \omega \le \pi.$$

This problem can be formulated as a convex problem by expressing the constraints in terms of $Y(\omega) = |H(\omega)|^2$, which is a cosine polynomial

$$Y(\omega) = y_0 + y_1 \cos\omega + \cdots + y_n \cos n\omega.$$

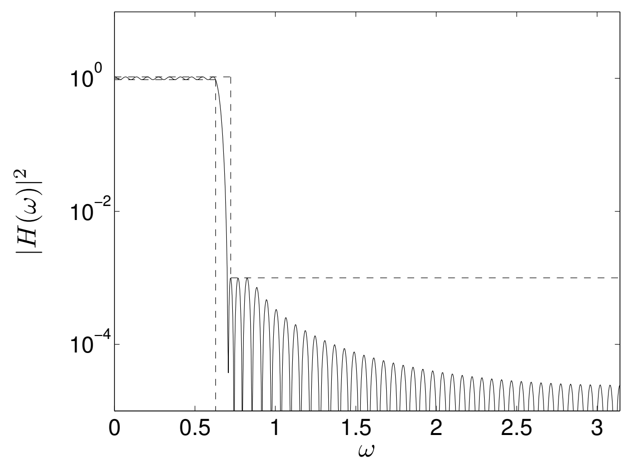

FIG. 6.3. *Frequency response of lowpass filter with length 102. The filter minimizes the stopband energy subject to the upper and lower bounds shown in dashed lines.*

The resulting problem is

$$\begin{array}{ll} \text{minimize} & \int_{\omega_s}^{\pi} Y(\omega)\, d\omega \\ \text{subject to} & 1/\delta_p \leq Y(\omega) \leq \delta_p, \quad 0 \leq \omega \leq \omega_p \\ & Y(\omega) \leq \delta_s, \quad \omega_s \leq \omega \leq \pi \\ & Y(\omega) \geq 0, \quad 0 \leq \omega \leq \pi, \end{array} \tag{6.5}$$

with variables $y \in \mathbf{R}^{n+1}$. From the optimal y, the filter coefficients h_k can be computed via spectral factorization [34].

Since Y is a cosine polynomial, problem (6.5) can be cast as an SDP of the form (6.1) as explained in §4. The problem dimensions are $L = 4$ and $s_i = 2$ for $i = 1, \ldots, L$. The primal variables are the $n+1$-vector y, and eight symmetric matrices X_{ik} of size $\lfloor n/2 \rfloor$ or $\lfloor (n-1)/2 \rfloor$.

We first consider an instance with parameters

$$n = 101, \qquad \delta_p = 1.05, \qquad \delta_s = 0.001, \qquad \omega_p = 0.2\pi, \qquad \omega_s = 0.23\pi.$$

Figure 6.3 shows the specifications and the optimal filter magnitude. Figure 6.4 shows the duality gap and the relative primal and dual residuals versus the iteration number. The code terminates after 20 iterations and requires 0.41 seconds per iteration.

Table 6.1 show the solution times for twelve filter design problems from the same family, with $\omega_p = 0.23\pi$ and $\delta_p = 1.1$ and n ranging from 25 to 300. The stopband parameters ω_s and δ_s are modified to tighten the specifications as n increases. The last two columns show the CPU time per iteration for the specialized interior-point implementation and for the general-purpose solver SDPT3, applied to the primal problem (6.1). (To express this problem as a standard form SDP, we split the y variable as a difference of two nonnegative vectors before passing it to SDPT3.) Figure 6.5 shows a graph of the CPU time versus n. The results clearly illustrate the benefits of exploiting problem structure when solving the Newton equations.

7. Conclusion. We have described a new SDP formulation of sum-of-squares theorems of nonnegative polynomials, cosine polynomials and trigonometric polynomials. The formulation results in structured SDPs that can be solved very efficiently, by taking advantage of simple properties of the **diag** operator.

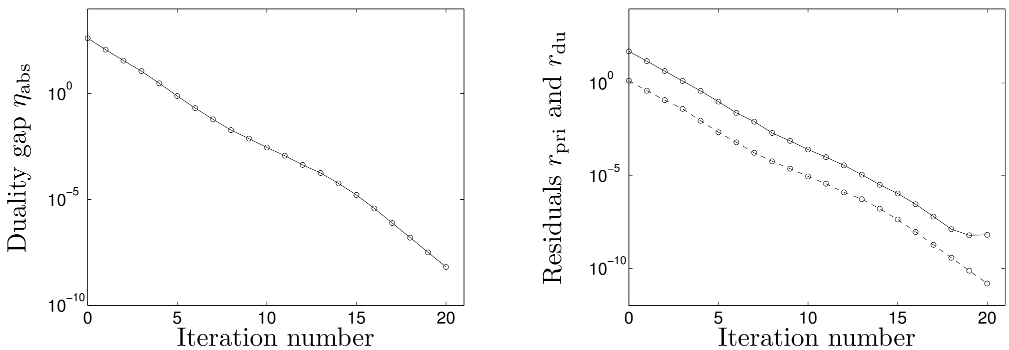

Fig. 6.4. *Progress of a primal-dual method for the lowpass filter design problem. The left plot shows the duality gap versus iteration number. The right plot shows the primal residual (solid line) and the dual residual (dashed line).*

Design parameters			Time per iteration (sec.)	
n	ω_s	δ_s	fast impl.	SDPT3
25	0.300π	$5.62 \cdot 10^{-3}$	0.04	0.17
50	0.280π	$3.16 \cdot 10^{-3}$	0.10	1.81
75	0.270π	$1.00 \cdot 10^{-3}$	0.21	5.78
100	0.260π	$1.00 \cdot 10^{-3}$	0.41	14.2
125	0.255π	$1.00 \cdot 10^{-3}$	0.71	29.0
150	0.250π	$1.00 \cdot 10^{-3}$	1.15	55.7
175	0.248π	$1.00 \cdot 10^{-3}$	1.77	86.5
200	0.248π	$3.16 \cdot 10^{-4}$	2.46	137
225	0.244π	$2.24 \cdot 10^{-4}$	3.50	203
250	0.244π	$1.78 \cdot 10^{-4}$	4.79	302
275	0.244π	$1.78 \cdot 10^{-4}$	6.57	
300	0.244π	$1.78 \cdot 10^{-4}$	8.56	

Table 6.1

Numerical results for a family of magnitude filter design problems. The first three columns specify the design parameters. The last two columns show the CPU time per iteration in seconds, for a special-purpose interior-point implementation that exploits problem structure, and for the general-purpose solver SDPT3.

The SDP parametrizations involve discrete transform matrices that are often orthogonal, or products of orthogonal and diagonal matrices. This should benefit the numerical stability of interior-point algorithms based on the parametrization. Although we have not analyzed the numerical properties, the numerical experiments are encouraging. In particular, the FIR filter examples that we solved successfully are much larger than those reported with other fast implementations of interior-point methods [16, 4].

REFERENCES

[1] J. W. Adams, *FIR digital filters with least-squares stopbands subject to peak-gain constraints*, IEEE Transactions on Circuits and Systems, 39 (1991), pp. 376–388.

[2] N. I. Ahiezer and M. Krein, *Some Questions in the Theory of Moments*, vol. 2 of Translations of Mathematical Monographs, American Mathematical Society, Providence, Rhode Island, 1962. Translated from 1938 Russian manuscript by W. Fleming and D. Prill.

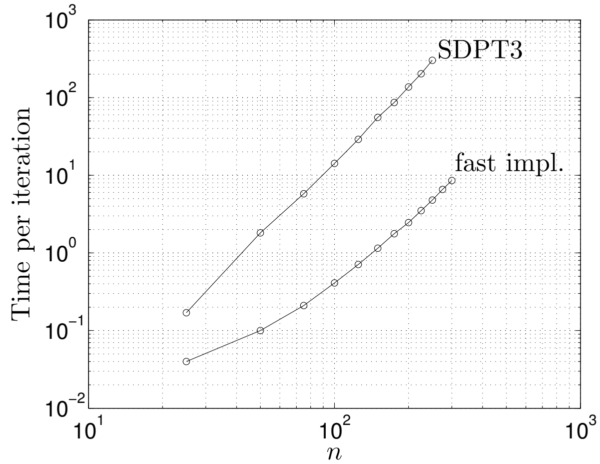

Fig. 6.5. *CPU time per iteration versus problem dimension for the results in table 6.1.*

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results*, SIAM J. on Optimization, 8 (1998), pp. 746–768.

[4] B. Alkire and L. Vandenberghe, *Convex optimization problems involving finite autocorrelation sequences*, Mathematical Programming Series A, 93 (2002), pp. 331–359.

[5] S. J. Benson and Y. Ye, *DSDP5: A software package implementing the dual-scaling algorithm for semidefinite programming*, Tech. Report ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory, 2004.

[6] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM Journal on Optimization, 10 (2000), pp. 443–461.

[7] W. L. Briggs and V. E. Henson, *The DFT. An Owner's Manual for the Discrete Fourier Transform*, SIAM, 1995.

[8] T. N. Davidson, Z.-Q. Luo, and J. F. Sturm, *Linear matrix inequality formulation of spectral mask constraints*, IEEE Transactions on Signal Processing, 50 (2002), pp. 2702–2715.

[9] T. N. Davidson, Z.-Q. Luo, and K. M. Wong, *Design of orthogonal pulse shapes for communications via semidefinite programming*, IEEE Transactions on Signal Processing, 48 (2000), pp. 1433–1445.

[10] P. J. Davis, *Interpolation and Approximation*, Dover Publications, 1975. First published by Blaisdell Publishing Company in 1963.

[11] B. Dumitrescu, I. Tabus, and P. Stoica, *On the parametrization of positive real sequences and MA parameter estimation*, IEEE Transactions on Signal Processing, 49 (2001), pp. 2630–9.

[12] L. Faybusovich, *On Nesterov's approach to semi-infinite programming*, Acta Applicandae Mathematicae, 74 (2002), pp. 195–215.

[13] ———, *Semidefinite descriptions of cones defining spectral mask constraints*, European Journal of Operations Reserach, (2005). To appear.

[14] Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren, *Optimization problems over positive pseudopolynomial matrices*, SIAM Journal on Matrix Analysis and Applications, 25 (2003), pp. 57–79.

[15] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules*, Mathematics of Computation, 23 (1969), pp. 221–230.

[16] Y. Hachez, *Convex Optimization over Non-Negative Polynomials: Structured Algorithms and Applications*, PhD thesis, Université catholique de Louvain, Belgium, 2003.

[17] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. on Optimization, 6 (1996), pp. 342–361.

[18] D. Henrion, D. Arzelier, and D. Peaucelle, *Positive polynomial matrices and improved LMI robustness conditions*, Automatica, 39 (2003), pp. 1479–1485.

[19] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone linear complementarity problem in symmetric matrices*, SIAM J. on Optimization, 7 (1997), pp. 86–125.

[20] M. G. Krein and A. A. Nudelman, *The Markov Moment Problem and Extremal Problems*,

vol. 50 of Translations of Mathematical Monographs, American Mathematical Society, Providence, Rhode Island, 1977.

[21] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11 (2001), pp. 796–817.

[22] J. Löfberg and P. A. Parrilo, *From coefficients to samples: a new approach to SOS optimization*, in Proceedings of the 43rd IEEE Conference on Decision and Control, 2004, pp. 3154–3159.

[23] P. Moulin, M. Anitescu, K. O. Kortanek, and F. A. Potra, *The role of linear semi-infinite programming in signal-adapted QMF bank design*, IEEE Transactions on Signal Processing, 45 (1995), pp. 2160–2174.

[24] Y. Nesterov, *Squared functional systems and optimization problems*, in High Performance Optimization Techniques, J. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds., Kluwer Academic Publishers, 2000, pp. 405–440.

[25] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Mathematical Programming Series B, 96 (2003), pp. 293–320.

[26] A. Stoica, R. Moses, and P. Stoica, *Enforcing positiveness on estimated spectral densities*, Electronics Letters, 29 (1993), pp. 2009–2011.

[27] P. Stoica, T. McKelvey, and J. Mari, *MA estimation in polynomial time*, IEEE Transactions on Signal Processing, 48 (2000), pp. 1999–2012.

[28] J. F. Sturm, *Using SEDUMI 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11-12 (1999), pp. 625–653.

[29] G. Szegö, *Orthogonal Polynomials*, American Mathematical Society, 2003. First published in 1939.

[30] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. on Optimization, 8 (1998), pp. 769–796.

[31] R. H. Tütüncü and M. J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming Series B, 95 (2003), pp. 189–217.

[32] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[33] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. H. Hanson, and T. Roh, *Interior point algorithms for semidefinite programming problems derived from the KYP lemma*, in Positive Polynomials in Control, D. Henrion and A. Garulli, eds., vol. 312 of Lecture Notes on Control and Information Sciences, Springer Verlag, 2005.

[34] S.-P. Wu, S. Boyd, and L. Vandenberghe, *FIR filter design via spectral factorization and convex optimization*, in Applied and Computational Control, Signals, and Circuits, B. Datta, ed., vol. 1, Birkhauser, 1998, pp. 215–245.