Convex optimization problems involving finite autocorrelation sequences

Brien Alkire and Lieven Vandenberghe

Department of Electrical Engineering
University of California Los Angeles
brien@alkires.com, vandenbe@ee.ucla.edu

Abstract

We discuss convex optimization problems in which some of the variables are constrained to be finite autocorrelation sequences. Problems of this form arise in signal processing and communications, and we describe applications in filter design and system identification. Autocorrelation constraints in optimization problems are often approximated by sampling the corresponding power spectral density, which results in a set of linear inequalities. They can also be cast as linear matrix inequalities via the Kalman-Yakubovich-Popov lemma. The linear matrix inequality formulation is exact, and results in convex optimization problems that can be solved using interior-point methods for semidefinite programming. However, it has an important drawback: to represent an autocorrelation sequence of length n, it requires the introduction of a large number $(n(n+1)/2)$ of auxiliary variables. This results in a high computational cost when general-purpose semidefinite programming solvers are used. We present a more efficient implementation based on duality and on interior-point methods for convex problems with generalized linear inequalities.

This material is based upon work supported by the National Science Foundation under Grant No. ECS-9733450.

The website http://www.ee.ucla.edu/~vandenbe/alv01b.htm contains software and other supporting materials.

1 Introduction

A vector $x = (x_0, x_1, \ldots, x_n) \in \mathbf{R}^{n+1}$ is a *finite autocorrelation sequence* if there exists a vector $y = (y_0, y_1, \ldots, y_n) \in \mathbf{R}^{n+1}$ such that

$$x_k = \sum_{i=0}^{n-k} y_i y_{i+k}, \quad k = 0, \ldots, n. \tag{1}$$

In this paper we study optimization problems with the constraint that some of the variables, or an affine combination of the variables, form a finite autocorrelation sequence. As will become clear in §2, this is a convex constraint, because the set of autocorrelation sequences in $\mathbf{R}^{n+1}$ is a convex cone. Optimization problems involving autocorrelation sequences are common in signal processing, system identification, and communications. We will discuss several examples and applications in §4.

It is well known in system theory that the set of autocorrelation sequences can be described using linear matrix inequalities (LMIs). Specifically, we will see that x is a finite autocorrelation sequence if and only if there exists a matrix $P \in \mathbf{S}^n$ (the set of symmetric $n \times n$-matrices) such that

$$\left[\begin{array}{cc} P & \tilde{x} \\ \tilde{x}^T & x_0 \end{array}\right] - \left[\begin{array}{cc} 0 & 0 \\ 0 & P \end{array}\right]$$

is positive semidefinite, where $\tilde{x} \in \mathbf{R}^n$ is defined as $\tilde{x} = [x_n \; x_{n-1} \; \cdots \; x_1]^T$. This condition is an LMI in x and the auxiliary variable P. As a consequence, many convex problems involving autocorrelation sequences can be solved using interior-point methods for semidefinite programming (SDP). However this is not necessarily the most efficient solution, since the conversion to an SDP requires the introduction of a large number of auxiliary variables (the $n(n+1)/2$ elements of P). The goal of this paper is to develop more direct and much more efficient interior-point methods for handling autocorrelation sequences. For a typical example, these techniques reduce the cost per iteration from $O(n^6)$ floating-point operations when using the SDP formulation to $O(n^3)$ flops.

Two groups of researchers have recently and independently arrived at similar conclusions. Dumitrescu, Tabus and Stoica in [DTS01] have noted that the computational cost of representing autocorrelation sequences via the SDP embedding can be drastically reduced (typically from $O(n^6)$ to $O(n^4)$) by solving the dual SDP. The method presented here is based on the same observation, and achieves a further reduction (from $O(n^4)$ to $O(n^3)$) by taking advantage of the Toeplitz structure in the dual problem. Genin, Hachez, Nesterov and Van Dooren in [GHNV00a, GHNV00b] have developed efficient algorithms for convex optimization problems over pseudo-polynomial matrices, which include autocorrelation sequences as a special case. Their method is similar, but not identical, to the method presented in this paper. The idea in both methods is to take advantage of problem structure by evaluating the second derivatives of the dual barrier function fast. In our case the barrier function is the logarithmic barrier function of a positive definite Toeplitz matrix. We apply the Levinson-Durbin algorithm to factor of the inverse of the matrix and the discrete Fourier transform to assemble the gradient and Hessian fast. The method described in [GHNV00a] uses the generalized Schur algorithm to factor the displacement of certain matrices that occur in the expressions for the Hessian. This offers a more general method, as it applies to any type of LMI constraint with low displacement rank, such as Hankel matrices. The techniques presented in this paper provide an alternative and more direct approach, not based on displacement rank techniques, and leading to algorithms with the same efficiency.

Outline

The outline of the paper is as follows. We start in the next paragraph by listing a number of equivalent characterizations and geometrical properties of autocorrelation sequences. In particular we show that the set of finite autocorrelation sequences forms a closed convex cone. We also describe the corresponding dual cone. In §3 we describe the two most widely used methods for handling autocorrelation cone constraints in optimization problems and discuss their shortcomings. In §4 we present examples and applications in filter design and system identification. The section includes a new formulation of piecewise-constant filter magnitude constraints as linear generalized inequalities with respect to the autocorrelation cone. In §5 we present general background on barrier methods for convex optimization problems with cone constraints. In §6 we describe an efficient dual barrier method for problems with autocorrelation cone constraints. The main contribution in this section is an efficient method for evaluating the Hessian of the dual barrier function. We present numerical results in §7, and conclude with a summary and some topics for future research in §8.

Notation

Most of the notation is standard. We denote by $\mathbf{R}^n$ the space of n-dimensional real vectors, by $\mathbf{R}^{m\times n}$ the space of $m \times n$ real matrices, and $\mathbf{S}^n$ the space of real symmetric matrices of size $n \times n$. $\mathbf{C}^n$ denotes the space of n-dimensional complex vectors and $\mathbf{C}^{m\times n}$ the space of $m \times n$ complex matrices. If $A, B \in \mathbf{C}^{m\times n}$, then $C = A \circ B$ denotes the Hadamard product of A and B: $C \in \mathbf{C}^{m\times n}$ with $C_{ij} = A_{ij}B_{ij}$.

$\mathcal{C}_{n+1}$ is the cone of finite autocorrelation sequences of length $n+1$, and $\mathcal{C}^*_{n+1}$ represents the corresponding dual cone, defined in §2.2.

Indices for vectors and matrices start at 0. The first component of a vector x is x_0. The first element of the first row of a matrix X is X_{00}. If $x \in \mathbf{R}^m$ and $y \in \mathbf{R}^n$, then $(x, y) \in \mathbf{R}^{m+n}$ denotes the vector $(x, y) = [x^T \; y^T]^T$. In §4 and in Appendix A we define $j = \sqrt{-1}$. In the rest of the paper j is only used as an index.

The symbols $\succeq$, $\succ$, $\preceq$, $\prec$ will denote matrix inequality, or vector inequality with respect to the cone of autocorrelation vectors, depending on the context. In other words, if $x, y \in \mathbf{S}^n$, then $x \succeq y$ means $x - y$ is positive semidefinite, and $x \succ y$ means that $x - y$ is positive definite. If $x, y \in \mathbf{R}^{n+1}$, then $x \succeq y$ means $x - y \in \mathcal{C}_{n+1}$, and $x \succ y$ means $x - y \in \mathbf{int}\,\mathcal{C}_{n+1}$. (This notation is justified by the fact that $\mathcal{C}_{n+1}$ is a convex cone; see §2.1.) The symbols $\succeq_*$, $\succ_*$, $\preceq_*$, $\prec_*$ denote generalized inequality with respect to the dual cone $\mathcal{C}^*_{n+1}$: if $z, w \in \mathbf{R}^{n+1}$, then $z \succeq_* w$ means $z - w \in \mathcal{C}^*_{n+1}$, and $z \succ_* w$ means $z - w \in \mathbf{int}\,\mathcal{C}^*_{n+1}$.

2 The cone of finite autocorrelation sequences

In this section we present a number of characterizations and interpretations of the set of finite autocorrelation sequences $\mathcal{C}_{n+1}$. These results (with the possible exception of the LMI characterization, which is proved in the appendix) are well known (see, for example, [KN77, BEFB94, Gra72, SM97]).

2.1 Equivalent definitions

Frequency-domain characterization [KN77]

The Fourier transform of the sequence

$$\ldots, 0, 0, x_n, x_{n-1}, \ldots, x_1, x_0, x_1, \ldots, x_n, 0, 0, \ldots$$

is a function $X : \mathbf{R} \to \mathbf{R}$ defined as

$$X(\omega) = x_0 + 2\sum_{k=1}^{n} x_k \cos k\omega.$$

We have the following property: $x \in \mathcal{C}_{n+1}$ if and only if X is nonnegative, *i.e.*,

$$X(\omega) \geq 0, \quad \omega \in [0, \pi]. \tag{2}$$

Note that X is periodic with period 2π and even, so it is sufficient to consider $\omega \in [0, \pi]$.

This frequency-domain characterization of $\mathcal{C}_{n+1}$ has several important consequences. First, it immediately implies that $\mathcal{C}_{n+1}$ is a *cone*: if $x \in \mathcal{C}_{n+1}$, then obviously $tx \in \mathcal{C}_{n+1}$ for all $t \geq 0$. Secondly, we can note that for fixed ω, the inequality (2) is a linear inequality in x, *i.e.*, it defines a closed halfspace in $\mathbf{R}^{n+1}$. In (2) we express $\mathcal{C}_{n+1}$ as the intersection of infinitely many halfspaces, parametrized by ω. As a consequence, $\mathcal{C}_{n+1}$ is a closed convex cone.

Linear matrix inequality characterization

In the second characterization, we describe $\mathcal{C}_{n+1}$ as the image of the cone of positive semidefinite matrices in $\mathbf{S}^{n+1}$ under a linear transformation. We first note that we can write the equation (1) as

$$x_k = y^T E^k y = \mathbf{Tr}\, E^k yy^T, \quad k = 0, \ldots, n \tag{3}$$

where $\mathbf{Tr}\, A$ denotes the trace of a square matrix A, and the matrix E is the unit-shift matrix in $\mathbf{R}^{(n+1)\times(n+1)}$, defined as

$$E = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

In (3), E^k denotes the kth power of E, *i.e.*, E^0 is the identity matrix, and for $k > 0$, E^k has zeros everywhere, except on the kth subdiagonal. Multiplying a vector y with E^k corresponds to shifting the components of y over k positions:

$$E^k y = (0, \ldots, 0, y_0, y_1, \ldots, y_{n-k}).$$

In (3) we represent $\mathcal{C}_{n+1}$ as the image of a non-convex set (the set of positive semidefinite rank-one matrices yy^T) under a linear transformation. Remarkably, we obtain the same set if we relax the rank-one constraint. Indeed, we prove in the appendix that $x \in \mathcal{C}_{n+1}$ if and only if

$$x_k = \mathbf{Tr}\, E^k Y, \quad k = 0, \ldots, n \tag{5}$$

for some $Y \in \mathbf{S}^{n+1}$ with $Y \succeq 0$.

For given x, (5) is a set of $n+1$ linear equations in Y, and it defines an affine set of dimension $n(n+1)/2$ in the space of symmetric $(n+1) \times (n+1)$-matrices. We can parametrize the same set explicitly as follows. It can be shown that Y satisfies (5) if and only if there exists a $P \in \mathbf{S}^n$ such that

$$Y = \begin{bmatrix} P & \tilde{x} \\ \tilde{x}^T & x_0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix}$$

where $\tilde{x} = [x_n \; x_{n-1} \; \cdots \; x_1]^T$. We therefore obtain the following equivalent LMI characterization for $\mathcal{C}_{n+1}$: $x \in \mathcal{C}_{n+1}$ if and only if there exists a $P \in \mathbf{S}^n$ such that

$$\begin{bmatrix} P & \tilde{x} \\ \tilde{x}^T & x_0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix} \succeq 0. \tag{6}$$

This is an LMI in the variables P and x.

This second form can also be derived directly from the Kalman-Yakubovich-Popov lemma in linear system theory; see the appendix for details.

Solution set of an infinite LMI [Gra72]

We can also characterize $\mathcal{C}_{n+1}$ as the solution set of an 'infinite' linear matrix inequality. It can be shown that $x \in \mathcal{C}_{n+1}$ if and only if the Toeplitz matrices $T_N(x) \in \mathbf{S}^N$, defined as

$$T_N(x)_{ij} = \begin{cases} x_{|i-j|} & \text{if } 0 \le |i-j| \le n \\ 0 & \text{otherwise,} \end{cases}$$

are positive semidefinite for all N:

$$T_N(x) \succeq 0, \quad N = 1, 2, \ldots. \tag{7}$$

Here we express $\mathcal{C}_{n+1}$ as the intersection of the solution sets of infinitely many linear matrix inequalities, which again confirms that $\mathcal{C}_{n+1}$ is a convex set.

Stochastic interpretation [SM97]

Our use of the term 'autocorrelation sequence' is justified by the following property: $x \in \mathcal{C}_{n+1}$ if and only if there exists a wide-sense stationary random sequence $w(t)$, $t = 0, 1, \ldots$, such that

$$\mathbf{E}\, w(t) w(t+k) = \begin{cases} x_k & 0 \le k \le n \\ 0 & k > n. \end{cases}$$

Nonnegative polynomials on $[-1,1]$

Any $x \in \mathcal{C}_{n+1}$ can be associated with a polynomial of degree n, defined as

$$P(t) = x_0 + 2x_1p_1(t) + 2x_2p_2(t) + \cdots + 2x_np_n(t)$$

where $p_k(t)$ is the Chebyshev polynomial of degree k. We have the following result: $P(t) \geq 0$ for $-1 \leq t \leq 1$ if and only if $x \in \mathcal{C}_{n+1}$. This follows immediately from the definition of Chebyshev polynomials [PM96, p.684],

$$p_k(t) = \cos(k\cos^{-1}(t)) \text{ for } t \in [-1,1].$$

As a consequence, we can identify $\mathcal{C}_{n+1}$ with the cone of polynomials of degree n that are nonnegative on $[-1,1]$.

2.2 The dual cone

The dual of the autocorrelation cone will play a fundamental role in the numerical techniques we discuss later. The dual cone of $\mathcal{C}_{n+1}$ is defined as

$$\mathcal{C}_{n+1}^{\star} = \{z \in \mathbf{R}^{n+1} \mid z^Tx \geq 0 \text{ for all } x \in \mathcal{C}_{n+1}\}.$$

An explicit characterization of the dual cone can be derived as follows [KN77]: $z^Tx \geq 0$ for all $x \in \mathcal{C}_{n+1}$ if and only if

$$\sum_{k=0}^{n} z_k \sum_{i=0}^{n-k} y_iy_{i+k} = \frac{1}{2}\left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_n \end{array}\right]\left[\begin{array}{ccccc} 2z_0 & z_1 & z_2 & \cdots & z_n \\ z_1 & 2z_0 & z_1 & \cdots & z_{n-1} \\ z_2 & z_1 & 2z_0 & \cdots & z_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & z_{n-2} & \cdots & 2z_0 \end{array}\right]\left[\begin{array}{c} y_0 \\ y_1 \\ \vdots \\ y_n \end{array}\right] \geq 0$$

for all $y \in \mathbf{R}^{n+1}$. Therefore, $z \in \mathcal{C}_{n+1}^{\star}$ if and only if

$$\left[\begin{array}{ccccc} 2z_0 & z_1 & z_2 & \cdots & z_n \\ z_1 & 2z_0 & z_1 & \cdots & z_{n-1} \\ z_2 & z_1 & 2z_0 & \cdots & z_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & z_{n-2} & \cdots & 2z_0 \end{array}\right] \succeq 0.$$

In other words, we can identify the dual cone with the cone of positive semidefinite Toeplitz matrices in $\mathbf{S}^{n+1}$.

2.3 Spectral factorization

Returning to the original definition (1), the question arises of checking whether a given x is a finite autocorrelation sequence. In other words, for a given x, we would like to be able to either construct a y that satisfies (1), or find a hyperplane that separates x from $\mathcal{C}_{n+1}$. This can be achieved very efficiently by a technique called *spectral factorization*. Several very efficient methods for spectral factorization exist, and we refer to [WBV98, p.242] for a brief survey.

For a given x, there may be multiple solutions y that satisfy (1). Spectral factorization methods usually compute the unique *minimum-phase* spectral factor, which satisfies the property that the complex function

$$y_0 + y_1 z^{-1} + \cdots + y_n z^{-n}$$

is nonzero for $|z| > 1$.

3 Embedding or approximating the autocorrelation cone

The frequency-domain and the LMI characterization given in §2 are the basis of two widely used methods for representing the autocorrelation cone. In this section we describe both representations, and discuss their shortcomings, as a motivation for the methods discussed later.

As a simple example, we will consider the optimization problem

$$\begin{array}{ll} \text{minimize} & \|x - \hat{r}\|^2 \\ \text{subject to} & x \succeq 0, \end{array} \tag{8}$$

with variable $x \in \mathbf{R}^{n+1}$, in which we approximate a given vector $\hat{r}$ by the nearest autocorrelation sequence.

Sampling

The definition (2) describes $\mathcal{C}_{n+1}$ as the intersection of an infinite number of linear inequalities, one for each value of ω. A popular method of handling the constraint $X(\omega) \geq 0$, is to sample this infinite set of linear inequalities. In this method we choose a large number of frequencies ω_i, $i = 0, \ldots, N$, in the interval $[0, \pi]$, and replace the constraint (2) by a finite set of N linear inequalities

$$x_0 + 2\sum_{k=1}^{n} x_k \cos k\omega_i \geq 0, \quad i = 0, \ldots, N. \tag{9}$$

Geometrically, we approximate the cone $\mathcal{C}_{n+1}$ with a polyhedral cone that contains it.

As an example, if we apply this approach to problem (8), we obtain a quadratic programming problem

$$\begin{array}{ll} \text{minimize} & \|x - \tilde{x}\|^2 \\ \text{subject to} & x_0 + 2\sum_{k=1}^{n} x_k \cos k\omega_i \geq 0, \quad i = 0, \ldots, N. \end{array}$$

The sampling method works well in practice, provided we choose N sufficiently large (typically $N = 15n$). However, it is not exact; it is possible to satisfy (9), and still have $X(\omega) < 0$ for some ω.

LMI embedding

The LMI characterizations (5) and (6) allow us to represent the constraint $x \in \mathcal{C}_{n+1}$ *exactly* by introducing new matrix variables Y or P. As an example, we can express (8) as the optimization problem

$$\begin{array}{ll} \text{minimize} & \|x - \tilde{x}\|^2 \\ \text{subject to} & x_k = \mathbf{Tr}\, E^k Y, \quad k = 0, \ldots, n \\ & Y \succeq 0. \end{array}$$

with variables $x \in \mathbf{R}^{n+1}$ and $Y \in \mathbf{S}^{n+1}$. An alternative form is

$$\begin{array}{ll} \text{minimize} & \|x - \tilde{x}\|^2 \\ \text{subject to} & \left[\begin{array}{cc} P & \tilde{x} \\ \tilde{x}^T & x_0 \end{array}\right] - \left[\begin{array}{cc} 0 & 0 \\ 0 & P \end{array}\right] \succeq 0 \end{array} \tag{10}$$

which has variables x and $P \in \mathbf{S}^n$. In both cases we obtain a convex optimization problem with a convex quadratic objective function and an LMI constraint.

No approximation is involved here, in contrast with the sampling method. The two representations may have a different complexity, depending on the algorithm used to solve them, but both suffer from the fact that a large number ($O(n^2)$) of auxiliary variables is introduced. (See [DTS01] for a detailed discussion of different SDP formulations of problem (8).)

4 Examples and applications

4.1 Enforcing nonnegativity of estimated spectral densities

Suppose we have measurements or estimates $\widehat{r}_k$ of the first $n+1$ autocorrelation coefficients $\mathbf{E}\, w(t)w(t+k)$ of a stationary time series $w(t)$. The estimates $\widehat{r}_k$ might be obtained, for example, by taking an average of sampled values of $w(t)w(t+k)$. Due to estimation errors, we may not have equality $\widehat{r}_k = \mathbf{E}\, w(t)w(t+k)$. Moreover, even if the values $\widehat{r}_k$ are exact, they only represent a finite part of an autocorrelation sequence of unknown and possibly infinite length. So we do not expect the sequence $\widehat{r}_0, \ldots, \widehat{r}_n$ to be a finite autocorrelation sequence, and as a result, the estimated power spectral density

$$\widehat{r}_0 + 2 \sum_{k=1}^{n} \widehat{r}_k \cos(k\omega)$$

might be negative for some ω. One approach to this problem is to approximate $\widehat{r}$ by the closest finite autocorrelation sequence, *i.e.*, to solve the problem (8). This problem was studied in [SMS93, ML91], where it was solved using semi-infinite optimization methods.

Stoica, McKelvey, and Mari [SMM00] mention similar constraints in the context of Moving Average (MA) signal estimation. For example, they consider the problem

$$\begin{array}{ll} \text{minimize} & (x - \widehat{r})^T W (x - \widehat{r}) \\ \text{subject to} & x \succeq 0 \end{array} \tag{11}$$

where $W \in \mathbf{S}^{n+1}$ and $\hat{r} \in \mathbf{R}^{n+1}$ are given with $W \succ 0$, and $x \in \mathbf{R}^{n+1}$ is the variable. This problem was solved via the SDP embedding and general-purpose SDP software in [SMM00]. In [DTS01, AV00] it was pointed out that the dual of the resulting SDP has a much smaller number of variables than the primal, so the problem can be solved more efficiently via the dual. In [AV00] the complexity was further reduced by taking advantage of the Toeplitz structure in the dual problem, as described below in §6.

4.2 Frequency-domain system identification

Consider the problem of fitting a rational function $H : \mathbf{C} \to \mathbf{C}$,

$$H(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}, \tag{12}$$

with real coefficients a_i, b_i, to measurements of its power spectrum $|H(e^{j\omega})|^2$. Specifically, we are interested in the problem

$$\begin{array}{ll} \text{minimize} & \max_{i=0,\ldots,N} \left| |H(e^{j\omega_i})|^2 - P_i \right| \\ \text{subject to} & b^T b = 1, \end{array} \tag{13}$$

where the variables are the coefficients $a \in \mathbf{R}^{n+1}$ and $b \in \mathbf{R}^{m+1}$, and the problem data are the measurement pairs ω_i, P_i for $i = 0, \ldots, N$. The equality constraint in (13) is added to normalize the solution.

Problem (13) is not convex in the variables a and b, but can be reformulated as a quasi-convex problem via a nonlinear change of variables. Define $u \in \mathbf{R}^{n+1}$, $v \in \mathbf{R}^{m+1}$ as

$$u_k = \sum_{i=0}^{n-k} a_i a_{i+k}, \quad v_k = \sum_{i=0}^{m-k} b_i b_{i+k},$$

i.e., u and v are the autocorrelation sequences of the vectors a and b. It is easily shown that

$$|H(e^{j\omega})|^2 = \frac{U(\omega)}{V(\omega)}$$

for all ω, where

$$\begin{aligned} U(\omega) &= u_0 + 2u_1 \cos\omega + \cdots + 2u_n \cos n\omega, \\ V(\omega) &= v_0 + 2v_1 \cos\omega + \cdots + 2v_m \cos m\omega. \end{aligned}$$

We can therefore express problem (13) as

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=0,\ldots,N} \left| \frac{U(\omega_i)}{V(\omega_i)} - P_i \right| \\ \text{subject to} & v_0 = 1 \\ & u \succeq 0, \quad v \succeq 0, \end{array}$$

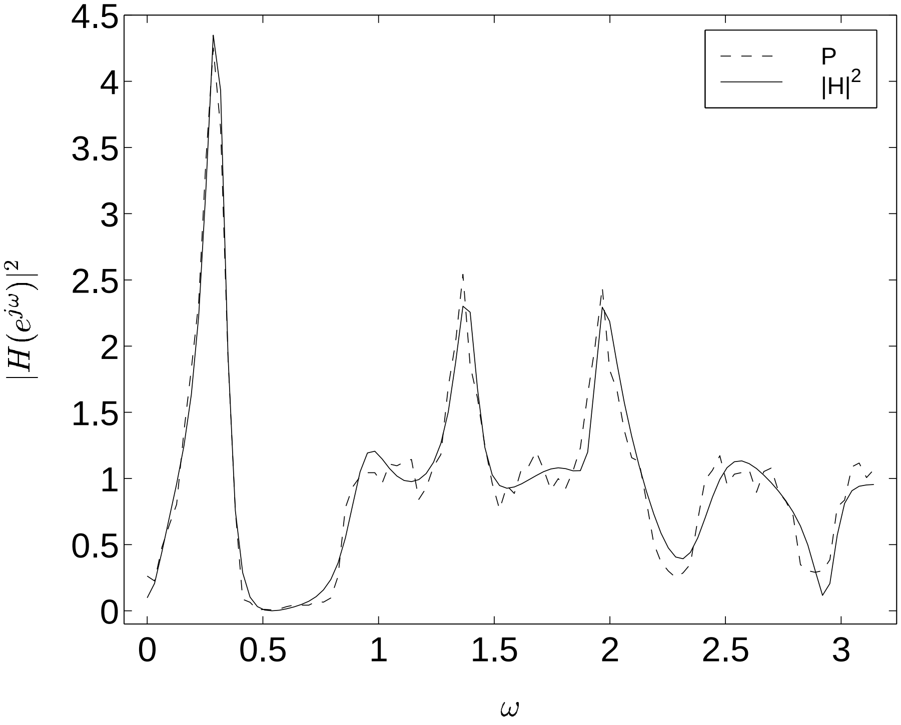

Figure 1: Power spectrum of an order 15 rational transfer function obtained via a minimax fit to the data shown with a dashed line.

This problem can be reformulated as

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & -\gamma V(\omega_i) \leq U(\omega_i) - P_i V(\omega_i) \leq \gamma V(\omega_i), \quad i = 0, \ldots, N \\ & v_0 = 1 \\ & u \succeq 0, \quad v \succeq 0, \end{array} \tag{14}$$

and solved via bisection on γ. From the optimal u and v we can obtain a globally optimal solution a and b by spectral factorization. Choosing minimum phase spectral factors yields a function H with no poles and zeros outside the unit disk. Note that it is possible to have poles on the unit circle at the optimal solution. This can be avoided by adding the constraint

$$V(\omega) - \delta \geq 0, \quad \omega \in [0, \pi],$$

where δ is a small positive constant, That is, we replace the constraint $v \succeq 0$ with

$$v - \delta\, e \succeq 0$$

where $e = (1, 0, \ldots, 0) \in \mathbf{R}^{m+1}$.

Figure 1 shows an example. The dimensions of the problem are $N = 99$, $m = 15$ and $n = 14$. The data were generated by adding noise to a 25th order transfer function.

4.3 FIR magnitude filter design

A *finite impulse response* (FIR) filter is a linear system described by the input-output relation

$$y(t) = \sum_{k=0}^{n} h_k u(t-k), \quad t \in \mathbf{Z}$$

where $u(t)$ is the input at time t, $y(t)$ is the output at time t, and $h_0, \ldots, h_n$ are the filter coefficients. The transfer function $H : \mathbf{C} \to \mathbf{C}$ of the filter is

$$H(z) = h_0 + h_1 z^{-1} + \cdots + h_n z^{-n}.$$

Pointwise upper and lower bounds on the magnitude of the frequency response $|H(e^{j\omega})|$, *i.e.* constraints of the form

$$L \leq |H(e^{j\omega})| \leq U, \quad \omega \in [\alpha, \beta]$$

are usually not convex in the filter coefficients h. However, they are convex as a function of the autocorrelation coefficients

$$x_k = \sum_{i=0}^{n-k} h_i h_{k+i}, \quad k = 0, \ldots, n, \tag{15}$$

and, as a consequence, a wide variety of FIR filter design problems can be solved via convex optimization [Sam88, WBV96, WBV98, DLW00].

As an example, consider the problem of designing a multiband FIR filter with $N+1$ bands $[\alpha_k, \beta_k]$, $k = 0, 1, \ldots, N$. We assume that $0 \leq \alpha_k < \beta_k \leq \pi$, and that none of the intervals overlap. In each band, we have a lower bound $L_k \geq 0$ and an upper bound $U_k > L_k$ on the filter magnitude. We are interested in minimizing the stopband energy (*i.e.* squared error) subject to peak constraints on the magnitude response. This is a variation on the peak-constrained least-squares filter design formulation introduced by Adams (see [AS98]).

The problem can be expressed as

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=0}^{N} w_k \int_{\alpha_k}^{\beta_k} \left|H(e^{j\omega})\right|^2 d\omega \\ \text{subject to} & L_k \leq |H(e^{j\omega})| \leq U_k, \quad \omega \in [\alpha_k, \beta_k], \ k = 0, \ldots, N, \end{array}$$

where $w_k = 0$ if band k is a passband, and $w_k = 1$, $L_k = 0$ if band k is a stopband. (Taking stopband weights w_k different from one allows us to trade-off the energy in different stopbands.) This problem is not convex in the filter coefficients h. However, in terms of the autocorrelation coefficients x, defined by (15), the problem reduces to

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=0}^{N} w_k \int_{\alpha_k}^{\beta_k} X(\omega) d\omega \\ \text{subject to} & L_k^2 \leq X(\omega) \leq U_k^2, \quad \omega \in [\alpha_k, \beta_k], \ k = 0, \ldots, N \\ & X(\omega) \geq 0, \ \omega \in [0, \pi], \end{array} \tag{16}$$

where X is defined as $X(\omega) = x_0 + 2x_1 \cos\omega + \cdots + 2x_n \cos n\omega$. The objective function is linear in x; the constraints are an infinite set of linear inequalities, so this problem is convex in the variable x.

Two methods exist for dealing with the semi-infinite nature of the constraints in (16). The most popular method is the sampling method of §3. We replace the constraint

$$L^2 \leq X(\omega) \leq U^2, \quad \omega \in [\alpha, \beta] \tag{17}$$

with a large finite set of inequality constraints

$$L^2 \leq X(\omega_k) \leq U^2, \quad k = 0, \ldots, m$$

where $\alpha \leq \omega_0 < \omega_2 < \cdots < \omega_m \leq \beta$.

A second and more recent method is based on semidefinite programming. We have already seen that for $\alpha = 0$ and $\beta = \pi$, the constraint (17) can be cast as two linear matrix inequalities. Davidson *et al.* in [DLS00] and Genin *et al.* in [GHNV00a] have recently extended this formulation, and derived LMI formulations of the constraints (17) for *arbitrary* α and β. Using these methods, a problem such as (16) can be cast as a semidefinite programming problem (SDP), without any approximation or sampling, and solved via general-purpose semidefinite programming software.

The constraints (17) can also be represented as generalized linear inequalities with respect to the cone $\mathcal{C}_{n+1}$, as follows. We first consider the constraint

$$X(\omega) \geq 0, \quad \omega \in [\alpha, \beta]. \tag{18}$$

To simplify notation, we make a change of variable $t = \cos\omega$. This maps the interval $0 \leq \omega \leq \pi$ to $-1 \leq t \leq 1$, and the function $\cos k\omega$ to the kth *Chebyshev polynomial* $p_k(t) = \cos(k \cos^{-1} t)$. Therefore $X(\omega)$ is mapped to the polynomial

$$P(t) = x_0 p_0(t) + 2\sum_{k=1}^{n} x_k p_k(t),$$

so it is clear that $P(t) \geq 0$ for $t \in [-1, 1]$ if and only if $x \succeq 0$. Now consider the constraint (18). It is satisfied if and only if $P(t) \geq 0$ for $t \in [\cos\beta, \cos\alpha]$. Let $A(\alpha, \beta) \in \mathbf{R}^{(n+1)\times(n+1)}$ be defined as follows: the components of $A(\alpha, \beta)x$ are the coordinates of P in the basis

$$p_0(at - b), \quad 2p_1(at - b), \quad \ldots, \quad 2p_n(at - b),$$

where $a = 2/(\cos\alpha - \cos\beta)$, and $b = (\cos\alpha + \cos\beta)/(\cos\alpha - \cos\beta)$. In other words, if we take $y = A(\alpha, \beta)x$, we can express $P(t)$ as

$$P(t) = y_0\, p_0(at - b) + 2\sum_{k=1}^{n} y_k\, p_k(at - b). \tag{19}$$

The matrix $A(\alpha, \beta)$ is readily constructed based on the recursion for Chebyshev polynomials

$$p_0(t) = 1, \quad p_1(t) = t, \quad p_k(t) = 2tp_{k-1}(t) - p_{k-2}(t), \quad k \geq 2.$$

We refer to the appendix for the details of constructing the matrix $A(\alpha, \beta)$ for given α, β. From (19) and the definition of a and b, it is clear that $P(t) \geq 0$ for $t \in [\cos\beta, \cos\alpha]$ if and only if

$$y_0\, p_0(\tau) + 2\sum_{k=1}^{n} y_k\, p_k(\tau) \geq 0, \quad \tau \in [-1, 1].$$

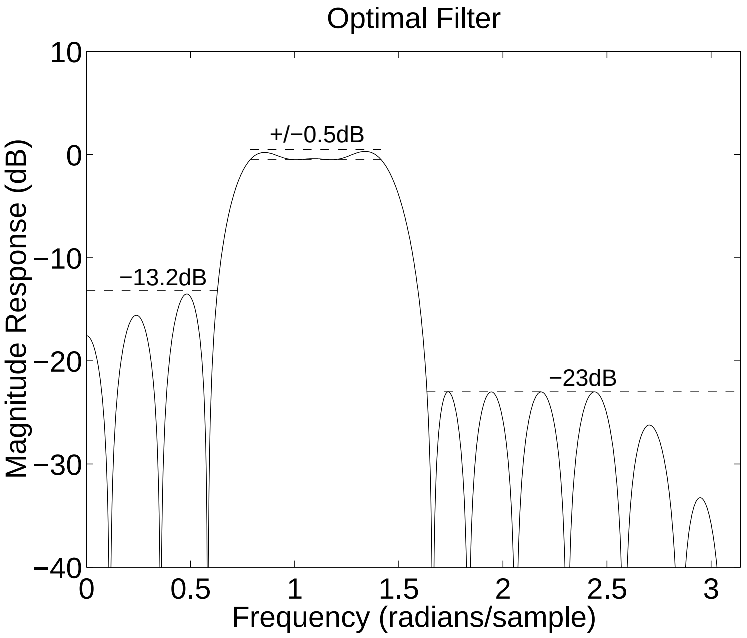

Figure 2: Length 24 bandpass filter.

In other words, x satisfies the constraint (17) if and only if $A(\alpha,\beta)x \succeq 0$, which is a generalized linear inequality with respect to $\mathcal{C}_{n+1}$.

More generally, we can express the magnitude constraints (17) as a pair of generalized linear inequalities

$$L^2 e \preceq A(\alpha,\beta)x \preceq U^2 e$$

where $e = (1, 0, \ldots, 0) \in \mathbf{R}^{n+1}$.

Returning to the multiband filter design example, we can now use the generalized inequality notation to express problem (16) as follows:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & L_k^2 e \preceq A(\alpha_k,\beta_k)x \preceq U_k^2 e, \quad k = 0,\ldots,N \\ & x \succeq 0 \end{array} \tag{20}$$

where

$$c_0 = \sum_{i=0}^{N} w_i(\beta_i - \alpha_i), \qquad c_k = 2\sum_{i=0}^{N} w_i \int_{\alpha_i}^{\beta_i} \cos k\omega \, d\omega, \quad k = 1,\ldots,n.$$

Figure 2 shows the magnitude response of a length $n = 24$ bandpass filter designed via this formulation, and using the algorithm described later in this paper. The first stopband is the interval $[0, 0.2\pi]$ (radians-per-sample), with an upper bound constraint of -13.2dB. The passband is the interval $[0.25\pi, 0.45\pi]$ (radians-per-sample), with passband gain constrained to ± 0.5dB. The second stopband is the interval $[0.52\pi, \pi]$ (radians-per-sample), with an upper bound constraint of -23dB. We minimize the weighted sum of the two stopband squared errors, with weights inversely proportional to the stop band widths. The problem can be formulated in terms of the filter coefficients h as

$$\begin{array}{ll} \text{minimize} & (1/(0.2\pi))\int_0^{0.2\pi}|H(e^{j\omega})|^2 d\omega + (1/(0.48\pi))\int_{0.52\pi}^{\pi}|H(e^{j\omega})|^2 d\omega \\ \text{subject to} & 20\log_{10}|H(e^{j\omega})| \le -13.2, \quad \omega\in[0,0.2\pi] \\ & -0.5 \le 20\log_{10}|H(e^{j\omega})| \le 0.5, \quad \omega\in[0.25\pi,0.45\pi] \\ & 20\log_{10}|H(e^{j\omega})| \le -23, \quad \omega\in[0.52\pi,\pi]. \end{array}$$

We should mention that the conditioning of the matrices $A(\alpha,\beta)$ appears to rapidly deterioriate with increasing n. Similar numerical problems were observed with the bandpass filter design formulation of [GHNV00a].

4.4 IIR magnitude filter design

The techniques described in the previous paragraph are readily extended to magnitude design of *infinite impulse response* (IIR) filter design, *i.e.* filters with a transfer function

$$H(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}.$$

The design variables are the coefficientsd a_i, $b_i \in \mathbf{R}$. For example, we can design a lowpass IIR filter with maximum stopband attenuation, subject to peak constraints on the magnitude response in the passband, by solving

$$\begin{array}{ll} \text{minimize} & \sup_{\omega\in[\omega_s,\pi]}|H(e^{j\omega})| \\ \text{subject to} & 1-\delta \le |H(e^{j\omega})| \le 1+\delta, \quad \omega\in[0,\omega_p] \\ & b^T b = 1. \end{array}$$

The variables are $a\in\mathbf{R}^{n+1}$ and $b\in\mathbf{R}^{m+1}$. The problem data are the passband limit ω_p, the stopband limit ω_s, and the allowable passband ripple α. The equality constraint is added to normalize the solution.

This problem is not convex in the variables a and b, but can be reformulated as a quasiconvex problem via the nonlinear change of variables described in §4.2. If we define $u\in\mathbf{R}^{n+1}$ and $v\in\mathbf{R}^{m+1}$ as the autocorrelation coefficients associated with a and b, respectively, the problem can be reformulated as

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & (1-\delta)^2 V(\omega) \le U(\omega) \le (1+\delta)^2 V(\omega), \quad \omega\in[0,\omega_p] \\ & -\gamma V(\omega) \le U(\omega) \le \gamma V(\omega), \quad \omega\in[\omega_s,\pi] \\ & v_0 = 1 \\ & v\succeq 0, \quad u\succeq 0 \end{array}$$

or equivalently (if for simplicity we assume $m=n$),

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & (1-\delta)^2 A(0,\omega_p)v \preceq A(0,\omega_p)u \preceq (1+\delta)^2 A(0,\omega_p)v \\ & -\gamma A(\omega_s,\pi)v \preceq A(\omega_s,\pi)u \preceq \gamma A(\omega_s,\pi)v \\ & v_0 = 1 \\ & v\succeq 0, \quad u\succeq 0 \end{array}$$

where the matrices $A(0,\omega_p)$ and $A(\omega_s,\pi)$ are constructed as described in the appendix. This is a quasiconvex problem with variables u, v and γ.

As in §4.2, it may be necessary to replace the constraint $v \succeq 0$ with $v \succeq \delta\, e$, in order to avoid poles on or very close to the unit circle.

5 Cone programming

In the remainder of this paper we will use the *cone programming* framework as a standard description of a general convex optimization problem. A cone program (CP) is defined as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F_i x + g_i \preceq_{K_i} 0, \quad i = 1, \ldots, L \\ & Ax = b, \end{array} \tag{21}$$

where the optimization variable is $x \in \mathbf{R}^n$ and the problem parameters are $c \in \mathbf{R}^n$, $F_i \in \mathbf{R}^{m_i \times n}$, $g_i \in \mathbf{R}^{m_k}$, $A \in \mathbf{R}^{p \times n}$, and $b \in \mathbf{R}^p$. The inequalities are generalized inequalities with respect to closed, pointed, and solid convex cones K_i, *i.e.*, $F_i x + g_i \preceq_{K_i} 0$ means $-F_i x - g_i \in K_i$ (see [NN94, §4.2], [BV99], [LSZ00]). Without loss of generality we can assume that $\mathbf{Rank}\, A = p$ and that

$$\mathbf{Rank} \left[\begin{array}{ccccc} F_1^T & F_2^T & \cdots & F_L^T & A^T \end{array} \right] = n.$$

The most commonly used cones in cone programming are the nonnegative real axis $\mathbf{R}_+ = \{x \in \mathbf{R} \mid x \geq 0\}$, the positive semidefinite cone, and the second-order cone $L_{n+1} = \{(x,t) \mid x \in \mathbf{R}^n, \|x\| \leq t\}$. However, cone programming is in fact very general, and any convex optimization problem can be expressed as a cone program (see [NN94, p.103]).

The examples in §4 are all readily expressed as cone programs as follows. Problem (11) can be reformulated as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{c} W^{1/2}(x - \hat{r}) \\ t \end{array} \right] \succeq_{L_{n+2}} 0 \\ & x \succeq 0 \end{array} \tag{22}$$

where t is an auxiliary variable, *i.e.*,

$$\left[\begin{array}{c} W^{1/2}(x - \hat{r}) \\ t \end{array} \right] \succeq_{L_{n+2}} 0 \Longleftrightarrow \|W^{1/2}(x - \hat{r})\| \leq t.$$

To solve problem (14) we can solve a sequence of feasibility problems with (scalar) linear inequalities (*i.e.* inequalities with respect to the nonnegative real axis) and generalized inequalities with respect to the cone of finite autocorrelation sequences. Problem (20) is already a cone program.

Efficient interior-point methods for solving cone programs with scalar linear inequalities, linear matrix inequalities (*i.e.* with respect to the positive semidefinite cone), and second-order cone constraints can be found in the literature on linear, semidefinite, and second-order cone programming. In this paper, we focus on special techniques for problems where one or more of the cones K_i are autocorrelation cones.

We first briefly review the general theory on duality and interior-point methods for cone programming. More details can be found in [NN94, BV99]. In §6 we discuss an efficient method for problems with autocorrelation cone constraints.

5.1 The dual cone program

We can associate with the cone program (21) the dual cone program

$$\begin{array}{ll} \text{maximize} & \sum_{k=1}^{L} g_k^T z_k - b^T \nu \\ \text{subject to} & \sum_{k=1}^{L} F_k^T z_k + A^T \nu + c = 0 \\ & z_k \succeq_{K_i^\star} 0, \quad k = 1, \ldots, L, \end{array} \tag{23}$$

where $K_i^\star$ is the dual cone of K_i, *i.e.*,

$$K_i^\star = \{ z \in \mathbf{R}^{m_i} \mid x^T z \geq 0 \text{ for all } x \in K_i \}.$$

We refer to (21) as the *primal* problem of the (dual) cone program (23).

It can be shown that if $p^\star$ and $d^\star$ are the optimal values of (21) and (23) respectively, then $p^\star \geq d^\star$. Moreover, $p^\star = d^\star$ if the primal or the dual problem is strictly feasible.

As an example, the dual of problem (22) is

$$\begin{array}{ll} \text{maximize} & \hat{r}^T W^{1/2} u \\ \text{subject to} & -W^{1/2} u - z = 0, \quad v = 1 \\ & \begin{bmatrix} u \\ v \end{bmatrix} \succeq_{L_{n+2}^*} 0, \quad z \succeq_* 0, \end{array}$$

where the variables are $u \in \mathbf{R}^{n+1}$, $v \in \mathbf{R}$, and $z \in \mathbf{R}^{n+1}$. The dual cone L_{n+2}^* of the second-order cone is itself, so the problem can be written more compactly as

$$\begin{array}{ll} \text{maximize} & -\hat{r}^T z \\ \text{subject to} & \|W^{-1/2} z\| \leq 1 \\ & z \succeq_* 0. \end{array}$$

5.2 Logarithmic barrier functions

Let $K \subseteq \mathbf{R}^m$ be a closed, pointed, and solid convex cone. A *logarithmic barrier function* for K is a function $\psi : \mathbf{R}^m \to \mathbf{R}$ with the following properties:

- $\mathbf{dom}\, \psi = \mathbf{int}\, K$ and $\psi(y) \to \infty$ as $y \in \mathbf{int}\, K$ approaches the boundary of K
- ψ is smooth and strictly convex

- there exists a constant $\theta > 0$ such that

$$\psi(sy) = \psi(y) - \theta \log s$$

for all $y \in \mathbf{int}\, K$ and all $s > 0$. The parameter θ is called the *degree* of ψ.

It can be shown that for all $y \in \mathbf{int}\, K$,

$$\nabla\psi(y) \prec_{K^\star} 0 \tag{24}$$

and

$$y^T \nabla\psi(y) = -\theta. \tag{25}$$

In practice, some additional properties are necessary or desirable. It is important that ψ and its first and second derivatives can be cheaply evaluated. Secondly, polynomial worst-case complexity results for barrier methods have only been developed under the assumption that the barrier functions are *self-concordant*, *i.e.* satisfy a certain condition involving second and third derivatives [NN94].

Examples of self-concordant logarithmic barriers include the function $\psi(y) = -\log y$, which is a logarithmic barrier function for the nonnegative real axis and has degree $\theta = 1$, $\psi(Y) = -\log\det Y$, which is a logarithmic barrier function for the positive semidefinite cone

$$K = \{Y \in \mathbf{S}^l \mid Y \succeq 0\}$$

and has degree $\theta = l$, and $\psi(y,t) = -\log(t^2 - y^T y)$, which is a logarithmic barrier for the second-order cone, with degree $\theta = 2$.

5.3 Dual barrier method

The dual barrier method for solving (21) applies when the problem (21) and its dual (23) are strictly feasible. It is based on the Sequential Unconstrained Minimization Technique (SUMT) [FM68] and relies on the following observation.

Let ψ_i be a logarithmic barrier function for the dual cone K_i^* in problem (23), with degree θ_i. Consider the problem of minimizing a weighted sum of the dual objective in (23) and the dual barrier functions

$$\begin{array}{ll} \text{minimize} & t(-\sum_{k=1}^{L} g_k^T z_k + b^T\nu) + \sum_{k=1}^{L} \psi_k(z_k) \\ \text{subject to} & \sum_{k=1}^{L} F_k^T z_k + A^T\nu + c = 0 \end{array} \tag{26}$$

where $t > 0$ is a parameter. It can be shown that the minimum is attained and that the minimizer $(z_1, \ldots, z_k, \nu)$ satisfies the following optimality conditions:

$$-tg_k + \nabla\psi_k(z_k) - F_k y = 0, \quad k = 1, \ldots, L, \qquad tb - Ay = 0 \tag{27}$$

for some $y \in \mathbf{R}^p$. We also have $\nabla\psi_k(z_k) \prec_{K_i} 0$ (by the general property (24)), and therefore

$$F_k(y/t) + g_k \prec_{K_i} 0, \quad A(y/t) = b.$$

In other words, $x = y/t$ is strictly feasible for the primal problem (21). Furthermore, using the property (25) we can evaluate the duality gap between this primal feasible point $x = y/t$ and the minimizer $(z_1, \ldots, z_L, \nu)$ of (26), *i.e.* the difference between the primal objective evaluated at x and the dual objective evaluated at $(z_1, \ldots, z_L, \nu)$:

$$c^T(y/t) - (-\sum_{k=1}^{L} g_k^T z_k + b^T \nu) = -\frac{1}{t} \sum_{k=1}^{L} z_k^T \nabla \psi_k(z_k) = (\sum_{k=1}^{L} \theta_k)/t.$$

This allows us to place a bound on how suboptimal $x = y/t$ is:

$$c^T x - p^\star \leq c^T x - (-\sum_{k=1}^{L} g_k^T z_k + b^T \nu) = (\sum_{k=1}^{L} \theta_k)/t,$$

where $p^\star$ is the optimal value of (21). In summary, we can find primal feasible points with $c^T x - p^\star \leq \epsilon$, by minimizing (26) with $t \geq \sum_k \theta_k / \epsilon$. Problem (26) is a smooth, convex optimization problem with equality constraints, and can be efficiently solved using Newton's method.

The dual barrier method is based on this idea, but solves (26) for a sequence of increasing values of t until $t \geq \sum_k \theta_k/\epsilon$, where ϵ is the desired accuracy. This often requires a smaller total number of Newton iterations than solving (26) directly for $t = \sum_k \theta_k/\epsilon$.

```
Dual barrier method
given tolerance $\epsilon$, initial $t > 0$, $\mu > 1$.
repeat
  1. Calculate the solution $(z_1, \ldots, z_L, \nu)$ of (26).
  2. Determine $x$ by solving $Ax = b$, $F_k x = -g_k + \nabla\psi_k(z_k)/t$, $k = 1, \ldots, L$.
  3. If $\sum_k \theta_k/t < \epsilon$, return $x$; otherwise set $t = \mu t$ and go to step 1.
```

The minimization in step 1 is usually implemented using Newton's method with a backtracking line search. It requires a strictly feasible starting point $(z_1, \ldots, z_L, \nu)$ in the first iteration. In subsequent iterations we can use the previous minimizer as starting point. In step 2 we solve x from an overdetermined set of linear equations, that is guaranteed to have a solution if $(z_1, \ldots, z_L, \nu)$ satisfies the optimality conditions (27). The parameter μ in step 3 controls the rate at which we increase t. Typical values are $\mu = 20, \ldots, 50$.

The complexity of the barrier method is dominated by the cost of calculating the Newton directions $(\Delta z_1, \ldots, \Delta z_L, \Delta \nu)$, which are given by the solution of the set of linear equations

$$\begin{bmatrix} \nabla^2\psi_1(z_1) & 0 & \cdots & 0 & 0 & F_1 \\ 0 & \nabla^2\psi_2(z_2) & \cdots & 0 & 0 & F_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \nabla^2\psi_L(z_L) & 0 & F_L \\ 0 & 0 & \cdots & 0 & 0 & A \\ F_1^T & F_2^T & \cdots & F_L^T & A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \vdots \\ \Delta z_L \\ \Delta \nu \\ v \end{bmatrix} = \begin{bmatrix} tg_1 - \nabla\psi_1(z_1) \\ tg_2 - \nabla\psi_2(z_2) \\ \vdots \\ tg_L - \nabla\psi_L(z_L) \\ -tb \\ 0 \end{bmatrix}.$$

6 Dual barrier for autocorrelation cone constraints

In this section we apply the dual barrier method to problems of the form (21) and (23), where some of the inequalities are with respect to the autocorrelation cone. This requires a logarithmic barrier for the dual cone, and efficient methods to evaluate its gradient and Hessian.

Recall from our discussion of the dual cone $\mathcal{C}_{n+1}^{\star}$ in §2.2 that $z \succeq_{\star} 0$ if and only if

$$F(z) = \begin{bmatrix} 2z_0 & z_1 & z_2 & \cdots & z_n \\ z_1 & 2z_0 & z_1 & \cdots & z_{n-1} \\ z_2 & z_1 & 2z_0 & \cdots & z_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & z_{n-2} & \cdots & 2z_0 \end{bmatrix} = \sum_{k=0}^{n} z_k (E^k + (E^k)^T) \succeq 0, \qquad (28)$$

where the matrices E^k are defined in (4). We will use the function

$$\psi(z) = -\log\det F(z), \qquad (29)$$

with domain $\mathbf{dom}\,\psi = \{z \mid F(z) \succ 0\}$, as a barrier function for the dual cone $\mathcal{C}_{n+1}^{\star}$. This is the standard log-det barrier for the cone of positive semidefinite matrices, and it is well known that this function is convex and self-concordant. It is also logarithmic with degree $n+1$, since

$$\psi^{\star}(sz) = \psi^{\star}(s) - (n+1)\log s$$

for $z \prec_{*} 0$ and $s > 0$. For future reference, we note that the first and second derivatives of ψ are given by

$$\nabla\psi(z)_j = -\,\mathbf{Tr}((E^j + (E^j)^T)F(z)^{-1}) = -2\,\mathbf{Tr}(E^j F(z)^{-1}), \qquad (30)$$

and

$$\begin{aligned} \nabla^2\psi(z)_{ij} &= \mathbf{Tr}(F(z)^{-1}(E^i + (E^i)^T)F(z)^{-1}(E^j + (E^j)^T)) \qquad (31) \\ &= 2\,\mathbf{Tr}(E^i F(z)^{-1} E^j F(z)^{-1}) + 2\,\mathbf{Tr}(E^i F(z)^{-1} (E^j)^T F(z)^{-1}) \end{aligned}$$

for $i, j = 0, 1, \ldots, n$.

Note that for *general* (dense, unstructured) matrices $A_i \in \mathbf{S}^{(n+1)}$, the cost of evaluating the function

$$-\log\det(z_0 A_0 + z_1 A_1 + \cdots + z_n A_n),$$

and its gradient is $O(n^3)$ flops, and the cost of evaluating the Hessian is $O(n^4)$ flops. In fact the cost of evaluating the Hessian in barrier methods for semidefinite programming often dominates the cost of solving the Newton equations. The purpose of this section is to describe an algorithm for evaluating the barrier $\psi(z)$ and its gradient in $O(n^2)$ flops, and the Hessian in $O(n^3)$ flops, by taking advantage of the Toeplitz structure of $F(z)$.

The Levinson-Durbin algorithm

The Levinson-Durbin algorithm allows us to efficiently calculate the Cholesky factorization of $F(z)^{-1}$. We write the Cholesky factorization as

$$F(z)^{-1} = RR^T$$

where the matrix

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0,n-1} & r_{0n} \\ 0 & r_{11} & r_{12} & \cdots & r_{1,n-1} & r_{1n} \\ 0 & 0 & r_{22} & \cdots & r_{2,n-1} & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{n-1,n-1} & r_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & r_{nn} \end{bmatrix}$$

is upper triangular with positive diagonal elements. We denote the kth column of R as r_k and the kth column flipped upside down and shifted up over $n-k$ positions, as $\tilde{r}_k$:

$$\tilde{r}_k = (r_{kk}\,,\, r_{k-1,k}\,, r_{k-2,k}, \;\ldots,\; r_{0k}, 0, \ldots, 0).$$

The algorithm proceeds as follows.

```
Levinson-Durbin algorithm
$r_{00} = 1/\sqrt{2z_0}$
for $k = 0$ to $n-1$ {
    $r_{k+1} = \frac{1}{\sqrt{1-\alpha_k^2}}(Er_k + \alpha_k \tilde{r}_k)$ where $\alpha_k = -r_{kk} z^T E r_k$
}
```

Each new column r_{k+1} is constructed as a linear combination Er_k (*i.e.* the previous column r_k shifted down over one position) and $\tilde{r}_k$. It can be shown that $|\alpha_k| < 1$ if $F(z) \succ 0$.

From the Cholesky factors we immediately obtain the value of the barrier function

$$\psi(z) = -\log\det F(z) = 2\sum_{k=0}^{n} \log r_{kk}.$$

The cost of one iteration of the Levinson-Durbin algorithm is $O(n)$ flops, so the total cost is $O(n^2)$. The algorithm therefore provides a very efficient way of evaluating the barrier function ψ. Below we examine the cost of calculating the gradient and the Hessian of ψ.

Formulas for the gradient and Hessian of ψ

We start by rewriting expressions (30) and (31) in terms of the Cholesky factors of $F(z)^{-1}$:

$$\begin{aligned} \nabla\psi(z)_j &= -2\,\mathbf{Tr}(R^T E^j R) \\ &= -2\sum_{k=0}^{n} r_k^T E^j r_k \end{aligned}$$

$$\begin{aligned}\nabla^2\psi(z)_{ij} &= 2\,\mathbf{Tr}(R^TE^iRR^TE^jR) + 2\,\mathbf{Tr}(R^TE^iRR^TE^{j^T}R)\\ &= 2\sum_{k=0}^{n}\sum_{l=0}^{n}\Big((r_k^TE^ir_l)(r_l^TE^jr_k) + (r_k^TE^ir_l)(r_l^TE^{j^T}r_k)\Big)\\ &= 2\sum_{k=0}^{n}\sum_{l=0}^{n}(r_k^TE^ir_l)(r_l^TE^jr_k + r_k^TE^jr_l).\end{aligned}$$

More compactly,

$$\begin{aligned}\nabla\psi(z) &= -2\sum_{k=0}^{n}c(k,k) && (32)\\ \nabla^2\psi(z) &= 2\sum_{k=0}^{n}\sum_{l=0}^{n}c(k,l)(c(l,k)+c(k,l))^T, && (33)\end{aligned}$$

where $c(k,l)\in\mathbf{R}^{n+1}$ denotes the *crosscorrelation* between the vectors r_k and r_l, *i.e.* a vector with components

$$c_i(k,l) = r_k^TE^ir_l = \sum_{j=0}^{n-i} r_{k,j+i}r_{lj}, \quad i=0,1,\ldots,n.$$

The cost of a straightforward evaluation of the expressions (32) and (33) is $O(n^3)$ flops and $O(n^4)$ flops, respectively. It takes $O(n^3)$ flops to calculate the autocorrelation vectors $c(k,k)$ by working out the inner products in the definition, and the addition in (32) costs $O(n^2)$ flops. Evaluating all crosscorrelation vectors $c(k,l)$ would take $O(n^4)$ flops, and the sum in (33) requires another $O(n^4)$ flops.

We now outline a more efficient method based on the *discrete Fourier transform* (DFT). The DFT is defined as follows. We define a complex matrix $W\in\mathbf{C}^{N\times(n+1)}$, where $N\geq 2(n+1)$, and with elements

$$W_{ik} = e^{-ik(2\pi\sqrt{-1}/N)}, \quad i=0,\ldots,N-1, \quad k=0,\ldots,n.$$

(The choice of N is not important for our present purposes, as long as $N\geq 2(n+1)$; we can assume that $N=2(n+1)$. In practice, if we use the fast Fourier transform to implement the DFT, we choose for N the smallest power of two greater than or equal to $2(n+1)$.)

The DFT of a vector $x\in\mathbf{R}^{n+1}$ is the vector $X\in\mathbf{C}^N$, defined as

$$X = Wx.$$

(More precisely, X is the DFT of the N-vector obtained by appending $N-n-1$ zeros to x.) The cost of evaluating the DFT or the inverse DFT is $O(n\log n)$ flops (assuming that $N=O(n)$).

We now return to the expressions for the gradient and Hessian in (32) and (33). Let $R_k = Wr_k$ be the DFT of r_k. It is well known that

$$c(k,l) = \frac{1}{N}W^*(R_k\circ\overline{R}_l)$$

where $\overline{R}_l$ denotes the complex conjugate of R_l and $A \circ B$ denotes the Hadamard product of two matrices or vectors A and B ($(A \circ B)_{ij} = A_{ij}B_{ij}$) (see [OS89, §8]). We therefore have the following identities:

$$\nabla\psi(z) = -2\sum_{k=0}^{n} c(k,k) = -\frac{2}{N}W^{\star}\sum_{k=0}^{n} R_k \circ \overline{R}_k \tag{34}$$

and

$$\begin{aligned}
\nabla^2\psi(z) &= 2\sum_{k=0}^{n}\sum_{l=0}^{n} c(k,l)(c(k,l)+c(l,k))^T \\
&= \frac{2}{N^2}W^*\sum_{k=0}^{n}\sum_{l=0}^{n}(R_k \circ \overline{R}_l)(R_k \circ \overline{R}_l + \overline{R}_k \circ R_l)^*W \\
&= \frac{2}{N^2}W^*\left((\sum_{k=0}^{n} R_k R_k^*) \circ (\sum_{l=0}^{n} R_l R_l^*)^T + (\sum_{k=0}^{n} R_k R_k^T) \circ (\sum_{l=0}^{n} R_l R_l^T)^*\right)W. \qquad (35)
\end{aligned}$$

The formulas (34) and (35) suggest a much more efficient way of evaluating gradient and Hessian. Calculating the gradient from the vectors R_k requires only $O(n^2)$ flops, while calculating the Hessian via (35) takes $O(n^3)$ flops.

Summary

In summary, the proposed algorithm for evaluating the dual barrier function $\psi(z)$, its gradient $\nabla\psi(z)$ and Hessian $\nabla^2\psi(z)$, proceeds as follows:

1. calculate the Cholesky factorization $F(z)^{-1} = RR^T$ via the Levinson-Durbin algorithm ($O(n^2)$ flops)
2. the value of the barrier function is given by $\psi(z) = 2\sum_{k=0}^{n}\log r_{kk}$
3. choose an integer $N \geq 2(n+1)$ (for example, the smallest power of 2 greater than $2(n+1)$), and calculate the DFTs R_k of the columns of R ($O(n^2\log n)$ flops)
4. evaluate the gradient via the expression (34) ($O(n^2)$ flops)
5. evaluate the Hessian via (35) ($O(n^3)$ flops)

The total cost is $O(n^3)$.

We note that several further improvements are possible. For example, we can eliminate step 3 if we adapt the Levinson-Durbin algorithm so that it directly computes the DFTs R_k rather than the vectors r_k. Such a modification is straightforward, using basic properties of the DFT. Other possible improvements of the basic algorithm would take advantage of the fact that z and R are real, both in the implementation of the Levinson-Durbin algorithm and the DFT. Finally, we could speed up the formation of the gradient and Hessian, by using the recursion that relates the vectors R_k.

We also point out that the $O(n^4)$ to $O(n^3)$ reduction in complexity results from the use of the DFT in forming the Hessian, and not from the application of the Levinson-Durbin algorithm. In fact, for numerical reasons, a standard Cholesky algorithm (with a cost of $O(n^3)$) flops might be preferable. The Levinson-Durbin algorithm is interesting when only the gradient is needed, for example, in line searches.

7 Numerical results

The numerical results in this section were obtained using a laptop computer running the Windows XP operating system and equipped with a 1GHz Pentium III processor and 256MB of RAM. The Matlab results were obtained using Matlab version V12.

The C and C++ codes were compiled using Microsoft Visual C++ version 6.0. External libraries were used for Fourier transform and linear algebra operations. The FFTW library version 2.1.3 was used for the Fourier transform operations (available at www.fftw.org). The library was compiled from the C source code. BLAS and Lapack version 3 libraries were used for the linear algebra operations (available at netlib.org). These libraries were compiled from the Fortran source code using Digital Visual Fortran version 5.0. No attempt was made to optimize the FFTW, BLAS or Lapack libraries for the computer architecture.

Dual barrier evaluations

Table 1 compares the performance of the fast $O(n^3)$ algorithm for evaluating the Hessian of the barrier, with a straightforward ($O(n^4)$) evaluation, as used in most general-purpose SDP solvers.

The middle column shows the CPU times for the fast algorithm as a function of n. The algorithm was implemented in C++ with a Matlab mex-file interface. FFTs of length N were used, where N is the smallest power of two greater than $2(n+1)$. This explains the jump in CPU time when the problem size exceeds a power of two.

The right column shows the CPU times required for the same evaluations using the routines of the general purpose SDP solver SP [VB94], which is implemented in C with calls to BLAS and Lapack libraries. The elements of the Hessian (31) are computed as follows. An efficient Fortran routine based on level 2 BLAS is used to perform the congruence operations

$$\tilde{F}_j = F(z)^{-1}\left(E^j + \left(E^j\right)^T\right)F(z)^{-1}, \quad j = 0, \ldots, n,$$

which takes $O(n^4)$ operations. We can then form the second derivatives as

$$\nabla^2\psi(z)_{ij} = \mathbf{Tr}((E^i + (E^i)^T)\tilde{F}_j) = 2\,\mathbf{Tr}(E^i\tilde{F}_j), \qquad i, j = 0, \ldots, n,$$

in $O(n^3)$ operations, by adding the elements on the ith diagonal of $\tilde{F}_j$.

The results clearly demonstrate the gain in efficiency achieved by the fast algorithm of §6.

Projection on the autocorrelation cone

Table 2 lists typical CPU times required for solving (8). The results for each n are averaged over five randomly generated vectors $\hat{r}$. The CPU time per Newton iteration is more relevant than the total CPU time, because we used a basic implementation (taking 69 Newton iterations on average) of the dual barrier method, with calls to the C++ code implementation of the algorithm of §6 for evaluating the barrier, gradient and Hessian. A more sophisticated interior-point method, with the same optimized code for evaluating the barrier function, would require fewer iterations, and roughly the same amount of time per iteration.

	$O(n^3)$ algorithm	$O(n^4)$ algorithm
$n+1$	CPU time (sec.)	CPU time (sec.)
100	0.11	3.06
200	2.54	54.34
300	15.24	481.94
400	20.29	1572.08
500	25.48	3896.21
600	125.32	8133.43
700	142.05	15062.33
800	158.41	28805.36
900	178.14	41270.38
1000	195.35	62968.16

Table 1: CPU times for evaluating the barrier, gradient and Hessian of the dual barrier using the fast method presented in §6 and a straightforward method as used in general-purpose SDP solvers.

Note that the CPU times per iteration listed in Table 2 are about the same as the CPU times of Table 1. Indeed, assembling the Hessian of the barrier function dominates the complexity of one Newton iteration in the dual barrier method. (The CPU times listed in Table 1 are actually slightly higher than the times per iteration in Table 2 because they include overhead for memory allocation. This overhead is only needed in the first iteration of the barrier method, so in Table 2 it is averaged out over many iterations.)

These results illustrate that exploiting problem structure allows us to solve much larger problems than via the SDP embedding and general-purpose SDP solvers. Note, for example, that for $n = 600$ the SDP formulation (10) of this problem involves about $180,000$ variables.

8 Conclusions

Semidefinite programming problems in system and control are often derived from the Kalman-Yakubovich-Popov (KYP) lemma. These LMIs involve an auxiliary matrix variable, introduced to express a semi-infinite frequency-domain inequality as a convex constraint in a finite-dimensional space with a finite number of variables. The number of auxiliary variables introduced this way is often very large compared to the number of original optimization variables. This has important consequences for the computational efficiency of interior-point methods, since the amount of work per iteration grows at least as the cube of the number of variables. For this reason, several researchers have recently proposed using cutting-plane methods as an efficient method to take advantage of the specific structure in LMI problems derived from the KYP lemma (see, for example, [Par00, Chapter 3]). In this paper we have shown for a specific constraint (the LMI representation (6) follows from the KYP lemma), that the same goal can be achieved using interior-point methods. In a typical application (such as problem (11)), the cost of applying a general-purpose SDP solver to the SDP derived

$n+1$	total time (sec.)	time/Newton iter. (sec.)
100	5.3	0.13
200	123.8	2.47
300	780.1	14.56
400	1510.6	19.27
500	2556.6	24.12
600	9599.2	114.28

Table 2: CPU times for solving the projection problem (8) using the dual barrier method. The second column gives the total CPU time; the right column gives the CPU time per Newton iteration. For each value of n the average over 5 randomly generated problem instances is given.

from the KYP lemma is $O(n^6)$ flops per iteration. Using the implementation described in this paper, the complexity is reduced to $O(n^3)$ per iteration. This reduction in complexity results from two observations. First, the number of variables in the dual problem is usually much smaller than in the primal problem, so the complexity of solving the dual problem is lower. Secondly, when solving the dual problem, one can take advantage of the Toeplitz structure in the dual LMI constraints.

As an interesting topic for future research, we can mention the possibility of developing primal or primal-dual methods (for example, using the primal-dual methods for nonsymmetric cones developed by Tunçel [Tun01]). A suitable logarithmic barrier for the (primal) autocorrelation cone is the Legendre transform of the dual barrier ψ, which is defined as

$$\phi(x) = \sup_z(-x^T z - \psi(z)),$$

for $x \succ 0$. It follows from [NN94, §2.4] that ϕ is a logarithmic and self-concordant barrier function for the autocorrelation cone. Although it is not clear what the most efficient method is to evaluate ϕ, it can certainly be evaluated numerically, by maximizing $-x^T z - \psi(z)$ over z using Newton's method, at a cost of $O(n^3)$ operations per Newton iteration. This allows us to implement a primal barrier method, with a similar gain in efficiency over general purpose SDP methods, as the dual method described in this paper.

Acknowledgments

We thank Ragu Balakrishnan, Bogdan Dumitrescu, Petre Stoica, and Ioan Tabus for their comments. We also thank Yvan Hachez and Paul Van Dooren for sending us preprints of references [GHNV00a, GHNV00b].

A LMI characterization of autocorrelation sequences

In this appendix, we prove the two LMI characterizations of autocorrelation sequences given in §2. We give two different proofs, although the results are equivalent, as mentioned in §2.

First form

We show that $x \in \mathcal{C}_{n+1}$ if and only if there exists a positive semidefinite $Y \in \mathbf{S}^{n+1}$ such that $x_k = \mathbf{Tr}\, E^k Y$ for $k = 0, 1, \ldots, n$.

The first part is obvious. Assume $x \in \mathcal{C}_{n+1}$. We can rewrite the definition (1) as

$$x_k = y^T E^k y = \mathbf{Tr}\, E^k y y^T, \quad k = 0, \ldots, n,$$

i.e., (3) holds for $Y = yy^T$.

Conversely, suppose that there exists a $Y = Y^T \succeq 0$ that satisfies (2), *i.e.*,

$$x_k = \mathbf{Tr}\, E^k Y, \;\; k = 0, \ldots, n, \quad Y = Y^T \succeq 0. \tag{36}$$

One can verify that for any $z \in \mathbf{C}$,

$$x_0 + x_1(z^{-1} + z) + x_2(z^{-2} + z^2) + \cdots + x_n(z^{-n} + z^n) = \mathbf{Tr}\, ZY$$

where Z is defined as

$$Z = \begin{bmatrix} 1 & z^{-1} & z^{-2} & \cdots & z^{-n} \\ z & 1 & z^{-1} & \cdots & z^{-(n-1)} \\ z^2 & z & 1 & \cdots & z^{-(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z^n & z^{n-1} & z^{n-2} & \cdots & 1 \end{bmatrix}.$$

For $z = e^{j\omega}$ the matrix Z is Hermitian positive semidefinite. By assumption $Y \succeq 0$. Therefore the inner product $\mathbf{Tr}\, ZY \geq 0$ for $z = e^{j\omega}$, *i.e.*,

$$X(\omega) = x_0 + x_1(e^{-j\omega} + e^{j\omega}) + x_2(e^{-2j\omega} + e^{2j\omega}) + \cdots + x_n(e^{-nj\omega} + e^{nj\omega}) \geq 0$$

for all ω. Therefore $x \in \mathcal{C}_{n+1}$.

Second form

We show that $x \in \mathcal{C}_{n+1}$ if and only if there exists a P such that the LMI (6) holds.

First of all we can verify that for all ω, and *any* $P \in \mathbf{S}^n$, the following identity holds:

$$X(\omega) = \begin{bmatrix} e^{-jn\omega} & e^{-j(n-1)\omega} & \cdots & e^{-j\omega} & 1 \end{bmatrix} \left(\begin{bmatrix} P & \tilde{x} \\ \tilde{x}^T & x_0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix} \right) \begin{bmatrix} e^{jn\omega} \\ e^{j(n-1)\omega} \\ \vdots \\ e^{j\omega} \\ 1 \end{bmatrix}.$$

Therefore if P satisfies the LMI (6), then $X(\omega) \geq 0$ for all ω.

The converse can be derived from a theorem of alternatives for semidefinite programming (see [BV01, Theorem 3]), which states that if the LMI is infeasible, then there exists a $Z \in \mathbf{S}^{n+1}$ such that $Z \succeq 0$,

$$\mathbf{Tr}\, Z \left[\begin{array}{cc} 0 & \tilde{x} \\ \tilde{x}^T & x_0 \end{array} \right] < 0 \tag{37}$$

and

$$\left[\begin{array}{cc} I_n & 0 \end{array} \right] Z \left[\begin{array}{c} I_n \\ 0 \end{array} \right] = \left[\begin{array}{cc} 0 & I_n \end{array} \right] Z \left[\begin{array}{c} 0 \\ I_n \end{array} \right], \tag{38}$$

i.e., the upper-left $n \times n$ block of Z is identical to its lower-right $n \times n$ block. In other words, Z is Toeplitz. Let us write Z as

$$Z = \left[\begin{array}{ccccc} 2z_0 & z_1 & z_2 & \cdots & z_n \\ z_1 & 2z_0 & z_1 & \cdots & z_{n-1} \\ z_2 & z_1 & 2z_0 & \cdots & z_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & z_{n-2} & \cdots & 2z_0 \end{array} \right].$$

We know that $Z \succeq 0$ and, from (37),

$$x_0 z_0 + x_1 z_1 + \cdots x_n z_n < 0.$$

Now assume x is a finite autocorrelation sequence, *i.e.*, $x_i = \sum_k y_i y_{k+i}$. Then we have a contradiction:

$$(1/2) y^T Z y = \sum_{k=0}^{n} z_k (\sum_{i=0}^{n-k} y_i y_{i+k}) = \sum_{k=0}^{n} z_k x_k < 0$$

and on the other hand, $Z \succeq 0$, *i.e.*, $y^T Z y \geq 0$. In other words, if the LMI (6) is infeasible, then $x \notin \mathcal{C}_{n+1}$.

Connection with the KYP-lemma

The Kalman-Yakubovich-Popov lemma [AM79] states that a transfer function $H = C(zI - A)^{-1}B + D$ with A stable, (A, B) controllable, (A, C) observable, and $D + D^T \geq 0$ satisfies

$$H(e^{j\omega}) + H(e^{-j\omega}) \geq 0$$

for all ω, if and only if there exists a $P \in \mathbf{S}^n$ such that

$$\left[\begin{array}{cc} P - A^T P A & C^T - A^T P B \\ C - B^T P A & D + D^T - B^T P B \end{array} \right] \succeq 0.$$

This reduces to the LMI (6) if we take

$$A = \left[\begin{array}{ccccc} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{array} \right] \in \mathbf{R}^{n \times n}, \quad B = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{array} \right] \in \mathbf{R}^n, \quad C = \left[\begin{array}{ccccc} x_n & x_{n-1} & \cdots & x_2 & x_1 \end{array} \right]$$

and $D = x_0/2$, for which

$$H(e^{j\omega}) = \frac{1}{2}x_0 + x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}$$

(see [WBV96, AHD74]).

B Representing spectral mask constraints

We have shown in §4.3 that the constraint

$$X(\omega) \geq 0, \quad \omega \in [\alpha, \beta], \tag{39}$$

where $0 \leq \alpha < \beta \leq \pi$, can be expressed as a generalized linear inequality $A(\alpha,\beta)x \succeq 0$, where $A(\alpha,\beta) \in \mathbf{R}^{(n+1)\times(n+1)}$ depends only on α and β. (Some authors refer to (39) as a *spectral mask constraint* [DLS00].) In this appendix we describe a method for constructing $A(\alpha,\beta)$.

Recall that $y = A(\alpha,\beta)x$ is the linear transformation defined by

$$x_0 + 2x_1p_1(t) + \cdots + 2x_np_n(t) = y_0 + 2y_1p_1(at-b) + \cdots + 2y_np_n(at-b),$$

where $a = 2/(\cos\alpha - \cos\beta)$ and $b = (\cos\alpha + \cos\beta)/(\cos\alpha - \cos\beta)$, and $p_k(t)$ is the kth Chebyshev polynomial.

The matrix $A(\alpha,\beta)$ must satisfy

$$\begin{bmatrix} p_0(t) \\ 2p_1(t) \\ 2p_2(t) \\ \vdots \\ 2p_n(t) \end{bmatrix} = A(\alpha,\beta)^T \begin{bmatrix} p_0(at-b) \\ 2p_1(at-b) \\ 2p_2(at-b) \\ \vdots \\ 2p_n(at-b) \end{bmatrix}.$$

We can calculate $A(\alpha,\beta)$ as $A(\alpha,\beta) = C^{-1}D$ where C and D are defined by

$$\begin{bmatrix} p_0(t) \\ 2p_1(t) \\ 2p_2(t) \\ \vdots \\ 2p_n(t) \end{bmatrix} = D^T \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^{n-1} \end{bmatrix}, \qquad \begin{bmatrix} p_0(at-b) \\ 2p_1(at-b) \\ 2p_2(at-b) \\ \vdots \\ 2p_n(at-b) \end{bmatrix} = C^T \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^{n-1} \end{bmatrix}.$$

The columns of $D = [d_0 \; d_1 \; \cdots \; d_n]$ can be constructed recursively as

$$d_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad d_1 = \begin{bmatrix} 0 \\ 2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad d_2 = \begin{bmatrix} -2 \\ 0 \\ 4 \\ \vdots \\ 0 \end{bmatrix}, \quad d_k = 2Ed_{k-1} - d_{k-2}, \; k = 3, \ldots, n,$$

where E is the unit-shift matrix defined in (4). This follows from the fact that $p_0(t) = 1$, $p_1(t) = t$, and from the recursion

$$p_k(t) = 2tp_{k-1}(t) - p_{k-2}(t), \quad k = 2, \ldots, n.$$

Similarly, the columns of $C = [c_0 \; c_1 \; \cdots \; c_n]$ can be constructed as

$$c_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad c_1 = \begin{bmatrix} -2b \\ 2a \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad c_2 = \begin{bmatrix} 4b^2 - 2 \\ -8ab \\ 4a^2 \\ \vdots \\ 0 \end{bmatrix}, \quad c_k = 2(aE - bI)c_{k-1} - c_{k-2}, \; k = 3, \ldots, n.$$

References

[AHD74] B. Anderson, K. Hitz, and N. Diem. Recursive algorithm for spectral factorization. *IEEE Transactions on Circuits and Systems*, 21:742–750, 1974.

[AM79] B. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[AS98] J. W. Adams and J. L. Sullivan. Peak-constrained least-squares optimization. *IEEE Transactions on Signal Processing*, 46(2):306–321, February 1998.

[AV00] B. Alkire and L. Vandenberghe. Handling nonnegative constraints in spectral estimation. In *Proceedings of the 34th Asilomar Conference on Signals, Systems, and Computers*, pages 202–206, 2000.

[BEFB94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics.* SIAM, Philadelphia, PA, June 1994.

[BV99] S. Boyd and L. Vandenberghe. Convex optimization. UCLA Academic Publishing, January 1999. Course Reader for EE236B: Nonlinear Programming.

[BV01] V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. 2001. Submitted to *IEEE Transactions on Automatic Control.* Expanded version available as Technical Report TR-ECE 02-02, Purdue University School of Electrical and Computer Engineering.

[DLS00] T. N. Davidson, Z.-Q. Luo, and J. F. Sturm. Linear matrix inequality formulation of spectral mask constraints. *Submitted to IEEE Transactions on Signal Processing*, 2000.

[DLW00] 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(5):1433–1445, 2000.

[DTS01] B. Dumitrescu, Ioan Tabus, and Petre Stoica. On the parametrization of positive real sequences and MA parameter estimation. *IEEE Transactions on Signal Processing*, 49(11):2630–9, November 2001.

[FM68] A. Fiacco and G. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques.* Wiley, 1968. Reprinted 1990 in the SIAM Classics in Applied Mathematics series.

[GHNV00a] Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren. Convex optimization over positive polynomials and filter design. In *Proceedings of the 2000 UKACC International Conference on Control*, Cambridge University, 2000.

[GHNV00b] Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren. Optimization problems over positive pseudo-polynomial matrices. *Submitted to the SIAM Journal on Matrix Analysis and Applications*, 2000.

[Gra72] R. M. Gray. On the asymptotic eigenvalue distribution of Toeplitz matrices. *IEEE Transactions on Information Theory*, 18(6):725–730, 1972.

[KN77] M. G. Krein and A. A. Nudelman. *The Markov Moment Problem and Extremal Problems*, volume 50 of *Translations of Mathematical Monographs.* American Mathematical Society, Providence, Rhode Island, 1977.

[LSZ00] Z.-Q. Luo, J. F. Sturm, and S. Zhang. Conic convex programming and self-dual embedding. *Optimization Methods and Software*, 14(3):169–218, 2000.

[ML91] R. L. Moses and D. Liu. Optimal nonnegative definite approximations of estimated moving average covariance sequences. *IEEE Transactions on Signal Processing*, 39(9):2007–2015, September 1991.

[NN94] Yu. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics.* SIAM, Philadelphia, PA, 1994.

[OS89] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[Par00] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, Pasadena, California, May 2000.

[PM96] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing.* Prentice Hall, 1996.

[Sam88] H. Samueli. Linear programming design of digital data transmission filters with arbitrary magnitude specifications. In *Conference Record, International Conference on Communications*, pages 30.6.1–30.6.5. IEEE, June 1988.

[SM97] P. Stoica and R. Moses. *Introduction to Spectral Analysis.* Prentice Hall, 1997.

[SMM00] P. Stoica, T. McKelvey, and J. Mari. MA estimation in polynomial time. *IEEE Transactions on Signal Processing*, 48(7):1999–2012, July 2000.

[SMS93] A. Stoica, R. Moses, and P. Stoica. Enforcing positiveness on estimated spectral densities. *Electronics Letters*, 29(23):2009–2011, November 1993.

[Tun01] L. Tunçel. Generalization of primal-dual interior-point methods to convex optimization problems in conic form. *IEEE Transactions on Signal Processing*, 1(3):229–254, March 2001.

[VB94] L. Vandenberghe and S. Boyd. SP*: Software for Semidefinite Programming. User's Guide, Beta Version.* Stanford University, October 1994. Available at `http://www-isl.stanford.edu/people/boyd`.

[WBV96] S.-P. Wu, S. Boyd, and L. Vandenberghe. FIR filter design via semidefinite programming and spectral factorization. In *Proc. IEEE Conf. on Decision and Control*, pages 271–276, 1996.

[WBV98] S.-P. Wu, S. Boyd, and L. Vandenberghe. FIR filter design via spectral factorization and convex optimization. In B. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, pages 215–245. Birkhauser, 1998.