Low-rank structure in semidefinite programs derived from the KYP lemma

Zhang Liu and Lieven Vandenberghe

***Abstract*— We extend a fast technique for solving semidefinite programs involving nonnnegative trigonometric polynomials to problems derived from the discrete-time Kalman-Yakubovich-Popov (KYP) lemma and some of its generalizations. The frequency-domain inequality associated with the generalized KYP lemma is first expressed as a weighted sum of squares of rational functions. By taking a sufficient number of samples of the sum-of-squares expression, an equivalent standard-form semidefinite program with low-rank structure is obtained. This low-rank structure is easily exploited in implementations of primal-dual interior-point algorithms. A complexity analysis and numerical examples are provided to support the performance improvement over standard semidefinite programming solvers.**

I. Introduction

The research on linear matrix inequalities during the last fifteen years has created a need for fast and reliable algorithms for semidefinite programming problems (SDPs) arising in control. High-quality general-purpose semidefinite programming packages [Stu99], [TTT03], [BY05], [YFK03], [Bor99] can currently handle dense semidefinite programs (SDPs) with up to a few thousand optimization variables (for SDPs in inequality form) and several linear matrix inequality (LMI) constraints of dimension up to a few hundred. They can also solve much larger problems if the data matrices are sufficiently sparse, and in fact most of the research on exploiting problem structure in semidefinite programming has focused on exploiting sparsity. SDPs in control, however, are usually quite dense, and often include matrix variables, so the number of optimization variables can easily run into several ten thousand, even when the underlying control problem is not particularly large (if problem size is measured in terms of the number of states, inputs and outputs).

This has prompted several researchers to develop special-purpose SDP algorithms targeted at control applications [Par99], [GHNV03], [DR05], [AV02], [RV06], [VBW+05], [KMJ03], [Hol06]. These algorithms are restricted to specific classes of SDPs, and exploit problem structure and control-theoretic properties to handle problem sizes that exceed the capabilities of the general-purpose packages. Such an approach has been particularly successful for SDPs derived from the KYP lemma (or 'KYP-SDPs'), a class of SDPs with widespread application in control [Par00], [KMJ03], [VBW+05]. A KYP-SDP is defined as

$$\begin{array}{ll} \text{min.} & c^T x \\ \text{s.t.} & K_i^H (\Phi \otimes P_i) K_i + M_i(x) \succeq 0, \ i = 1, \ldots, L, \end{array} \tag{1}$$

where $M_i : \mathbf{R}^p \to \mathbf{H}^{n_i+m_i}$ is an affine mapping, $c \in \mathbf{R}^p$,

$$K_i = \left[\begin{array}{cc} A_i & B_i \\ I & 0 \end{array} \right],$$

with $A_i \in \mathbf{C}^{n_i \times n_i}$, $B_i \in \mathbf{C}^{n_i \times m_i}$, $\otimes$ denotes the Kronecker product, and

$$\Phi = \left[\begin{array}{cc} 1 & 0 \\ 0 & -1 \end{array} \right], \qquad \Phi = \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right]$$

for discrete-time and continuous-time systems, respectively. The variables in (1) are $x \in \mathbf{R}^p$ and $P_i \in \mathbf{H}^{n_i}$, $i = 1, \ldots, L$. The name KYP-SDP refers to the Kalman-Yakubovich-Popov lemma, which states that if (A_i, B_i) is controllable, the existence of a P_i that satisfies the ith inequality in (1) is equivalent to the frequency-domain inequality

$$\left[\begin{array}{c} (\lambda I - A_i)^{-1} B_i \\ I \end{array} \right]^H M_i(x) \left[\begin{array}{c} (\lambda I - A_i)^{-1} B_i \\ I \end{array} \right] \succeq 0 \tag{2}$$

for all λ on the unit circle, respectively, the imaginary axis. Note that each (Hermitian) matrix variable P_i in (1) adds n_i^2 (real) variables to the SDP, making it very expensive to solve directly via general-purpose interior-point packages when n_i exceeds, say, fifty.

The different proposals for dedicated KYP-SDP solvers can be grouped in three categories. A first approach [Par99], [KMJ00], [KM01], [KMJ03] avoids the matrix variables P_i by working directly with the frequency-domain inequality constraints (2) instead of the LMIs in (1). This results in a semi-infinite SDP (or a semi-infinite linear program in the systems (A_i, B_i) are single-input). The problem is solved by localization methods such as the ellipsoid algorithm or the analytic centering cutting plane method. Each iteration of the cutting-plane algorithm requires determining whether a given x is feasible in (2), and, if not, generating a hyperplane that separates x from the feasible set. This can be done efficiently by standard methods for checking nonnegativity of a transfer function on the imaginary axis or the unit circle (via the eigenvalue decomposition of a Hamiltonian or symplectic matrix). The complexity of one iteration is low, $O(n^3)$ operations if we assume for simplicity that $L = 1$, A_1 is $n \times n$, B_1 is $n \times 1$ and $p = O(n)$. These algorithms are well suited if the number of optimization variables p is not too high.

A second class of methods are the barrier methods proposed by Kao and Megretski [KM01], [KM03]. These algorithms are based on new barrier functions for the convex set defined by the frequency inequalities (2), instead of the standard logarithmic barrier function for the constraints

Electrical Engineering Department, University of California, Los Angeles. Email: zhang@ee.ucla.edu, vandenbe@ee.ucla.edu. Research supported by NSF under grant ECS-0524663.

in (1). Thus the need for the auxiliary variables P_i is avoided. Under the same assumptions as above the algorithm in [KM03] has a complexity of $O(n^5)$ per iteration, but it requires many fewer iterations than a cutting-plane method, and it reaches an accurate solution much more quickly.

Finally, one can consider customized versions of standard primal-dual interior-point methods, applied directly to the SDP (1). The method presented in [VBW$^+$05] is a primal-dual interior-point method, similar to the algorithms used in [TTT03], [Stu99], with a complexity of $O(n^4)$ per iteration (again, assuming that $L = 1$, A_1 is $n \times n$, B_1 is $n \times 1$ and $p = O(n)$). This is achieved by exploiting problem structure in the linear equations that are solved at each step of the interior-point algorithm and form the most time-consuming part of the algorithm. The complexity can be further reduced by applying the conjugate gradient algorithm [HV00], [GH03] to solve the linear equations, or by transforming the matrices A_i to a canonical form (diagonal or companion) [VBW$^+$05]. However these last techniques are numerically less stable than the $O(n^4)$ method and may adversely influence the total number of interior-point iterations or the obtainable accuracy. The third approach has the advantage that it is based on the most popular interior-point methods for semidefinite programming, and that it allows us to take advantage of the wealth of theoretical and experimental knowledge about primal-dual interior-point methods gathered during the last decade.

In this paper we describe a new method for solving KYP-SDPs, as well as SDPs based on the generalized KYP lemma introduced by Iwasaki and Hara [IH05]. We adopt the third approach mentioned above and present a fast implementation of a standard primal-dual interior-point method applied to a reformulation of (1). Under the same assumptions as above, the algorithm has a complexity of $O(n^3)$ operations per iteration. Contrary to the $O(n^3)$ method in [VBW$^+$05], the new method does not require the reduction of the state-space model to a canonical form with questionable numerical properties. It also uses less memory and a much simpler preprocessing than the algorithms in [VBW$^+$05].

This work is a continuation of the papers [AV02], [RV06]. In [AV02] a fast algorithm was presented for convex optimization problems involving nonnegative trigonometric polynomials (KYP-SDPs for FIR systems). The algorithm is a dual barrier method, and achieves an $O(n^3)$ complexity per iteration by using the discrete Fourier transform to compute the Hessian of the dual barrier function. This idea is closely related to the techniques in [GHNV03], which use the generalized Schur method to evaluate the derivatives of the dual barrier function fast. In [RV06] the results of [AV02] were extended to other types of interior-point methods (primal and primal-dual methods), and a larger class of applications. The key observation is that the discrete transform technique of [AV02] can be interpreted as derived from a reformulation of the KYP-SDP. The constraints in the SDP express the fact that a nonnegative trigonometric polynomial can be factored as a sum of squares [Nes00]. An equivalent formulation of this constraint is obtained by first sampling the sum-of-squares expression, and then interpolating the samples to obtain the coefficients of the nonnegative polynomial. This yields an SDP with a low-rank structure that is easily exploited in any interior-point algorithm, resulting in a complexity of $O(n^3)$ per iteration. In addition, the use of discrete transforms offers the possibility of some further improvements using fast transforms techniques. In [RV06] the details were worked out for trigonometric polynomials, cosine polynomials, and real polynomials of one variable. Similar ideas have been applied in [LP04], [RDV07] to multidimensional sum-of-squares optimization. The purpose of this paper is to extend the method of [RV06] to state-space representations of nonnegative rational functions.

Notation: $\mathbf{S}^n$ denotes the set of real symmetric matrices of order n; $\mathbf{H}^n$ is the set of complex Hermitian matrices of order n. The symbols $\succeq$ and $\succ$ are used for matrix inequalities. $\bar{A}$ is the complex conjugate of the matrix A; $A^H = \bar{A}^T$ is the complex conjugate transpose. The paraconjugate of a transfer matrix $H(z)$ is $H_*(z) = H(1/\bar{z})^H$. If $H(z) = C(zI - A)^{-1}B + D$, then $H_*(z) = B^H(z^{-1}I - A^H)^{-1}C^H + D^H$. Note that $H_*(z) = H(z)^H$ for z on the unit circle.

II. The KYP and generalized KYP lemmas

Assume that (A, B) is controllable, with $A \in \mathbf{C}^{n\times n}$ and $B \in \mathbf{C}^{n\times m}$, and that A does not have eigenvalues on the unit circle. We define

$$U(z) = (zI - A)^{-1}B, \qquad V(z) = \begin{bmatrix} U(z) \\ I \end{bmatrix}.$$

The discrete-time KYP lemma and its generalizations provide necessary and sufficient conditions for a rational function of the form

$$F(z) = V_*(z)MV(z), \tag{3}$$

with $M \in \mathbf{H}^{n+m}$, to be positive semidefinite on the unit circle or a segment of the unit circle. A function of the form (3) is often called a *Popov function* [IW93]. In this section we review the variations of the KYP lemma needed for our purposes.

A. Identically zero Popov functions

We first consider Popov functions that are identically zero, *i.e.*, satisfy

$$V_*(z)MV(z) = 0 \quad \forall z. \tag{4}$$

A well-known explicit parametrization is as follows: M satisfies (4) if and only if there exists a $P \in \mathbf{H}^n$ such that

$$M = K^H(\Phi \otimes P)K \tag{5}$$

where $\otimes$ denotes the Kronecker product and

$$K = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}, \qquad \Phi = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{6}$$

(see [HSK99, p.340]). The equation (5) is a set of linear equations in M and the auxiliary variable P.

Another convenient way to convert (4) to a finite set of equations follows from the fact that

$$V_*(z)MV(z) = \frac{1}{z^n \det(zI - A)\det(z^{-1}I - \bar{A})} Y(z)$$

where Y is a matrix polynomial of degree $2n$. Therefore $V_*(z)MV(z)$ is identically zero if it is zero at $2n+1$ or more points (that are not zeros of $z^n \det(zI - A)\det(z^{-1}I - \bar{A})$), for example, $2n+1$ distinct points on the unit circle. In other words, if we choose a set

$$\mathcal{C} = \{e^{j\omega_k} \mid 0 \le \omega_1 < \cdots < \omega_q < 2\pi\} \qquad (7)$$

of $q \ge 2n+1$ distinct points on the unit circle, then (4) is equivalent to the linear equations

$$V_*(z)MV(z) = 0 \quad \forall z \in \mathcal{C}. \qquad (8)$$

(See [AG93] for other examples of similar techniques.)

B. The KYP lemma

The KYP lemma states that $F(e^{j\omega}) \succeq 0$ for all ω if and only if there exists a $P \in \mathbf{H}^n$ such that

$$M + K^H(\Phi \otimes P)K \succeq 0. \qquad (9)$$

This is a linear matrix inequality in M and the variable $P \in \mathbf{H}^n$. We can apply the results of the previous section to reformulate the constraint (9) in a form that is better suited for inclusion in semidefinite programming problems solved by interior-point algorithms. First, if we denote the slack in (9) by X, then (9) means that the Popov function $V_*(z)(M - X)V(z)$ is identically zero. Therefore (9) holds for some P if and only if there exists an X such that

$$F(z) = V_*(z)XV(z) \quad \forall z, \quad X \succeq 0. \qquad (10)$$

As in the previous paragraph, it is sufficient to impose the equality at a finite set of at least $2n+1$ points on the unit circle. This yields the following form of the KYP lemma: $F(e^{j\omega}) \succeq 0$ for all ω if and only if there exists an X such that

$$F(z) = V_*(z)XV(z) \quad \forall z \in \mathcal{C}, \quad X \succeq 0, \qquad (11)$$

where $\mathcal{C}$ is defined as in (7).

It is important to note the difference between (11) and constraint generation as used, for example, in cutting-plane methods. Sampling the constraint $F(e^{j\omega}) \succeq 0$ by replacing it with

$$F(z) \succeq 0 \quad \forall z \in \mathcal{C},$$

where $\mathcal{C}$ is a finite set of points on the unit circle, is a *relaxation* of the original constraint. The constraint (11), on the other hand, is exactly equivalent to $F(e^{j\omega}) \succeq 0$ for all ω.

C. The generalized KYP lemma

Iwasaki and Hara [IH05] describe several extensions of the KYP lemma that characterize nonnegativity of a rational function on a segment of the unit circle. We mention the following theorem from [IH05]. Suppose α and β are given with $0 < \beta < \pi$. Then $F(e^{j\omega}) \succeq 0$ for $|\omega - \alpha| \le \beta$ if and only if there exist $P, Q \in \mathbf{H}^n$ such that

$$M + K^H\left(\Phi \otimes P - \Psi \otimes Q\right)K \succeq 0, \quad Q \succeq 0. \qquad (12)$$

where

$$\Psi = \begin{bmatrix} 0 & e^{j\alpha} \\ e^{-j\alpha} & -2\cos\beta \end{bmatrix}. \qquad (13)$$

From section II-A, we know that there exists a P that satisfies (12) if and only if

$$\begin{aligned} V_*(z)MV(z) &= V_*(z)\left(X_1 + K^H(\Psi \otimes Q)K\right)V(z) \\ &= V_*(z)X_1V(z) + 2g(z)U_*(z)QU(z) \end{aligned}$$

where $g(z) = (e^{j\alpha}z^{-1} + e^{-j\alpha}z - 2\cos\beta)/2$, and X_1 is the slack in the first inequality in (12). If we denote $X_2 = 2Q$, we can conclude that $F(e^{j\omega}) \succeq 0$ for $|\omega - \alpha| \le \beta$ if and only if there exist matrices X_1 and X_2 such that

$$\begin{gathered} F(z) = V_*(z)X_1V(z) + g(z)U_*(z)X_2U(z) \quad \forall z \\ X_1 \succeq 0, \quad X_2 \succeq 0. \end{gathered} \qquad (14)$$

Note that $g(e^{j\omega}) = \cos(\omega - \alpha) - \cos\beta \ge 0$ for $|\omega - \alpha| \le \beta$. One direction of the generalized KYP lemma is therefore obvious. If F can be expressed as (14) for some $X_1 \succeq 0$, $X_2 \succeq 0$, then clearly $F(e^{j\omega}) \succeq 0$ for $|\omega - \alpha| \le \beta$, since $g(e^{j\omega}) \ge 0$ on this interval.

As in the standard KYP lemma we can express (14) in sampled form as

$$\begin{gathered} F(z) = V_*(z)X_1V(z) + g(z)U_*(z)X_2U(z) \quad \forall z \in \mathcal{C} \\ X_1 \succeq 0, \quad X_2 \succeq 0. \end{gathered} \qquad (15)$$

III. Semidefinite programming

This section provides some background on interior-point methods for semidefinite programming.

A. Standard form and inequality form SDPs

The following optimization problems are Lagrange duals:

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(QX) + d^Tx \\ \text{subject to} & \mathcal{F}(X) + Gx = h \\ & X \succeq 0, \end{array} \qquad (16)$$

with as variables $X \in \mathbf{H}^s$, $x \in \mathbf{R}^r$, and

$$\begin{array}{ll} \text{maximize} & h^Tz \\ \text{subject to} & \mathcal{F}^{\text{adj}}(z) \preceq Q \\ & G^Tz = d, \end{array} \qquad (17)$$

with variable $z \in \mathbf{R}^t$. The coefficients in the cost functions are $Q \in \mathbf{H}^s$, $d \in \mathbf{R}^r$, $h \in \mathbf{R}^t$. The equality constraint in the primal problem is defined in terms of a linear mapping $\mathcal{F} : \mathbf{H}^s \to \mathbf{R}^t$. $\mathcal{F}^{\text{adj}}$ is the adjoint of $\mathcal{F}$.

The applications discussed in this paper have an additional block-diagonal structure and take the form

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{L} \mathbf{tr}(Q_k X_k) + d^T x \\ \text{subject to} & \mathcal{F}_k(X_k) + G_k x = h_k, \;\; k = 1, \ldots, L \\ & X_k \succeq 0, \;\; k = 1, \ldots, L. \end{array} \qquad (18)$$

There are L sets of equality constraints of dimension r_k, and L matrix variables $X_k \in \mathbf{H}^{s_k}$, each appearing in one equality. The equality constraints are coupled only by the variable x. The corresponding dual problem is

$$\begin{array}{ll} \text{maximize} & \sum_{k=1}^{L} h_k^T z_k \\ \text{subject to} & \mathcal{F}_k^{\text{adj}}(z_k) \preceq Q_k, \quad k = 1, \ldots, L \\ & \sum_{k=1}^{L} G_k^T z_k = d. \end{array} \qquad (19)$$

It is a matter of convention which of the two problems (16) and (17) is called the primal or the dual. We therefore refer to an SDP of the form (16) as a *standard form* SDP (with free variables) and to an SDP of the form (17) as an *inequality form* SDP.

B. Interior-point methods

To estimate the cost of solving the SDPs (16) and (17) by a primal-dual interior-point algorithm of the type used in the popular solvers [Stu99], [TTT03], it is sufficient to know that the number of iterations of an interior-point method is relatively small (usually less than 50) and grows slowly with problem size. Each iteration requires the solution of a set of linear equations

$$\begin{array}{rcl} -T^{-1}\Delta X T^{-1} + \mathcal{F}^{\text{adj}}(\Delta z) & = & R \\ \mathcal{F}(\Delta X) + G\Delta x & = & r_1 \\ G^T \Delta z & = & r_2, \end{array}$$

where $T \succ 0$. The values of T and the righthand sides R, r_1 and r_2 change at each iteration. These equations are often referred to as the *Newton equations*, because they can be interpreted as obtained by linearizing the conditions that characterize the primal and dual central paths.

The Newton equations are solved by eliminating ΔX from the first equation, and then solving

$$\left[\begin{array}{cc} H & G \\ G^T & 0 \end{array}\right] \left[\begin{array}{c} \Delta z \\ \Delta x \end{array}\right] = \left[\begin{array}{c} r_3 \\ r_2 \end{array}\right] \qquad (20)$$

where $r_3 = r_1 + \mathcal{F}(TRT)$ and H is the matrix that satisfies

$$\mathcal{F}(T\mathcal{F}^{\text{adj}}(u)T) = Hu \qquad (21)$$

for all u. The cost of solving (20) is $O(\max\{r^3, t^3\})$. The cost of forming the matrix H depends on the structure of $\mathcal{F}$, and often exceeds the cost of solving (20).

The Newton equations for the block-diagonal SDP (18) can be reduced to an equation with coefficient matrix

$$\left[\begin{array}{ccccc} H_1 & 0 & \cdots & 0 & G_1 \\ 0 & H_2 & \cdots & 0 & G_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & H_L & G_L \\ G_1^T & G_2^T & \cdots & G_L^T & 0 \end{array}\right] \qquad (22)$$

where H_i is defined by the identity $\mathcal{F}_i(T_i\mathcal{F}_i^{\text{adj}}(u)T_i) = H_i u$.

C. General-purpose solvers

General-purpose software packages require that $\mathcal{F}$ is expressed as a vector of inner products

$$\mathcal{F}(X) = (\mathbf{tr}(F_1X), \, \mathbf{tr}(F_2X), \, \ldots, \, \mathbf{tr}(F_tX)). \qquad (23)$$

In this case, the elements of H are given by

$$H_{ij} = \mathbf{tr}(F_iTF_jT), \quad i, j = 1, \ldots, t. \qquad (24)$$

If no sparsity in the matrices F_i is exploited, this requires $\max\{ts^3, t^2s^2\}$ operations, since each matrix product TF_iT requires $O(s^3)$ operations, and the t^2 inner products $\mathbf{tr}(F_iTF_jT)$ cost $O(s^2)$ each.

D. Rank-one structure

Important savings are possible when the mapping $\mathcal{F}$ has rank-one structure [BYZ00], [LP04], [RV06]. Suppose

$$\mathcal{F}(X) = D\,\mathbf{diag}(E^HXE), \qquad (25)$$

where $D \in \mathbf{R}^{t\times q}$, $E \in \mathbf{C}^{s\times q}$. In (25) we express the components of $\mathcal{F}(X)$ as linear combinations of the diagonal entries of E^HXE. These diagonal entries are inner products with rank-one matrices. It can be shown that

$$H = D((E^HTE) \circ \overline{(E^HTE)})D^H$$

where $\circ$ denotes the Hadamard product. From this we see that H can be assembled in order $\max\{s^2q, sq^2, tq^2, t^2q\}$ operations. If $q = O(\max\{t, s\})$, this represents an important gain over the complexity of general-purpose solvers discussed in the last paragraph.

IV. SDPs derived from the KYP lemma

We now consider optimization problems of the form

$$\begin{array}{ll} \text{min.} & c^T x \\ \text{s.t.} & F_i(e^{j\omega}, x) \succeq 0, \; |\omega - \alpha_i| \leq \beta_i, \; i = 1, \ldots, L. \end{array} \qquad (26)$$

The optimization variable is $x \in \mathbf{R}^p$. The constraints are frequency domain inequalities defined in terms of L Popov functions

$$F_i(z, x) = V_{i*}(z)M_i(x)V_i(z)$$

that are affine in x, *i.e.*,

$$M_i(x) = M_{i0} + \sum_{k=1}^{p} x_k M_{ik}$$

where $M_{ij} \in \mathbf{H}^{n_i+m_i}$, and

$$V_i(z) = \left[\begin{array}{c} U_i(z) \\ I \end{array}\right], \qquad U_i(z) = (zI - A_i)^{-1}B_i,$$

with $A_i \in \mathbf{C}^{n_i\times n_i}$, $B_i \in \mathbf{C}^{n_i\times m_i}$. We assume that the pairs (A_i, B_i) are controllable and that A_i has no eigenvalues on the unit circle. Each inequality is defined on an interval $[\alpha_i - \beta_i, \alpha_i + \beta_i]$, where $0 < \beta_i \leq \pi$.

Our discussion of the KYP lemma in section II suggests two routes to formulating (26) as an SDP.

A. Inequality form KYP-SDP

In the control literature, problems of the form (26) are usually converted to SDPs using the inequality form of the generalized KYP lemma (12). The ith constraint in (26) is replaced with an LMI in x and two auxiliary variables P_i and Q_i:

$$M_i(x) + K_i^H(\Phi \otimes P_i - \Psi \otimes Q_i)K_i \succeq 0, \quad Q_i \succeq 0$$

where

$$K_i = \begin{bmatrix} A_i & B_i \\ I & 0 \end{bmatrix}, \qquad \Psi_i = \begin{bmatrix} 0 & e^{j\alpha_i} \\ e^{-j\alpha_i} & -2\cos\beta_i \end{bmatrix}.$$

If $\beta_i = \pi$ for some i, we can remove the variable Q_i:

$$M_i(x) + K_i^H(\Phi \otimes P_i)K_i \succeq 0.$$

We obtain an SDP in inequality form, with variables x, P_i, Q_i. The resulting SDP is expensive to solve by interior-point methods. Each matrix variable P_i or Q_i contributes n_i^2 variables, and as we have seen, the cost of solving an inequality form SDP grows at least as fast as the cube of the number of variables. If for simplicity we assume that $L = 1$, $m_i = 1$ and $n_1 = n$, $p = O(n)$, we obtain a complexity of at least $O(n^6)$ per iteration (not including the cost of assembling the coefficient matrix defined in (24).

B. Standard form KYP-SDP

Using the equality form of the KYP lemma, we can replace each constraint in (26) with a constraint

$$\begin{gathered} F_i(z,x) = V_{i*}(z)X_{i1}V_i(z) + g_i(z)U_{i*}(z)X_{i2}U_i(z) \\ X_{i1} \succeq 0, \quad X_{i2} \succeq 0, \end{gathered} \qquad (27)$$

where $g_i(z) = (e^{j\alpha_i}z^{-1} + e^{-j\alpha_i}z - 2\cos\beta_i)/2$, or with

$$F_i(z,x) = V_{i*}(z)X_iV_i(z) \qquad X_i \succeq 0 \qquad (28)$$

if $\beta_i = \pi$. If we then sample these equality constraints on sets

$$\mathcal{C}_i = \{e^{j\omega_k} \mid 0 \le \omega_1 \le \cdots \le \omega_{q_i} < 2\pi\}$$

of $q_i \ge 2n_i + 1$ or more distinct points on the unit circle, we obtain an SDP in standard form, with free variables x.

For single-input KYP-SDPs ($m_i = 1$), the problem can be expressed succinctly as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & D_i\,\mathbf{diag}(E_i^H X_i E_i) + G_i x = h_i, \;\; i = 1, \ldots, L \\ & X_i \succeq 0, \;\; i = 1, \ldots, L \end{array}$$

where $h_i - G_i x = (F_i(z_{i1}, x), \ldots, F_i(z_{iq_i}, x))$,

$$D_i = \begin{bmatrix} I & \mathbf{diag}(g_i(z_{i1}), \ldots, g_i(z_{iq_i})) \end{bmatrix}$$

$$E_i = \begin{bmatrix} E_{i1} & 0 \\ 0 & E_{i2} \end{bmatrix}, \qquad X_i = \begin{bmatrix} X_{i1} & 0 \\ 0 & X_{i2} \end{bmatrix},$$

and

$$\begin{aligned} E_{i1} &= \begin{bmatrix} V_i(z_{i1}) & V_i(z_{i2}) & \cdots & V_i(z_{i,q_i}) \end{bmatrix} \\ E_{i2} &= \begin{bmatrix} U_i(z_{i1}) & U_i(z_{i2}) & \cdots & U_i(z_{i,q_i}) \end{bmatrix}. \end{aligned}$$

Note that the dimensions of the matrices D_i, E_i, X_i are all $O(n_i)$. If $L = 1$, $n_1 = O(n)$, $p = O(n)$, the SDP can therefore be solved at a cost of order n^3 per iteration.

For multi-input KYP-SDPs ($m_i > 1$), sampling the $m_i \times m_i$ matrix equations (27) or (28) on $2n_i + 1$ points results in $O(n_i m_i^2)$ scalar equalities with rank-one coefficients. The complexity per iteration therefore grows rapidly (at least as m_i^6) with the number of inputs.

V. Basis matrices and selection of sample points

The sampled KYP-SDPs formulations described in the last section require the matrix

$$\begin{bmatrix} U(z_1) & U(z_2) & \cdots & U(z_q) \end{bmatrix}, \qquad (29)$$

where $U(z) = (zI - A)^{-1}B$, the pair (A, B) is controllable, and $z_1, \ldots, z_q$ are $q \ge 2n + 1$ distinct points on the unit circle. The sample points z_i can be chosen arbitrarily, but their choice clearly affects the condition number of (29). It is also important to be able to compute (29) fast.

A. Single-input systems

We first consider single-input systems, so throughout this section we assume that $m = 1$. We make two assumptions that can be satisfied at the expense of a simple preprocessing step with a complexity of $O(n^3)$.

First, we assume A is stable. If A is unstable, we can find a stabilizing state feedback, for example, by pole placement or by solving an LQR problem, and replace the KYP constraint with an equivalent constraint defined in terms of a stable matrix $A + BF$ (see [VBW+05]).

Second, we assume that (A, B) is input balanced, *i.e.*, $AA^H + BB^H = I$. An input-balanced realization is easily obtained by computing the controllability Grammian P from $APA^H - P + BB^H = 0$, and redefining A and B as $A := P^{-1/2}AP^{1/2}$, $B := P^{-1/2}B$.

The following results are related to orthogonal transformation techniques used in system identification [HVW05, Chapter 3]. Since $AA^H + BB^H = I$, there exist C and D such that the matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

is unitary. (This means that $G(z) = D + C(zI - A)^{-1}B$ is an *inner function*, *i.e.*, $G_*(z)G(z) = 1$.) Consider the generalized eigenvalue problem of order $2n + 1$

$$Mv = zNv \qquad (30)$$

where

$$M = \begin{bmatrix} A & B & 0 \\ 0 & 0 & I \\ C & D & 0 \end{bmatrix}, \qquad N = \begin{bmatrix} I & 0 & 0 \\ 0 & \bar{B} & \bar{A} \\ 0 & \bar{D} & \bar{C} \end{bmatrix}. \qquad (31)$$

The eigenvalues and a unitary matrix of eigenvectors can be obtained by a Schur factorization

$$N^H M = Q\,\mathbf{diag}(z_1, \ldots, z_{2n+1})Q^H. \qquad (32)$$

This factorization, with Q unitary and z_k on the unit circle, exists because the lefthand side is a product of unitary matrices, hence unitary itself. From (30) we see that the (non-normalized) generalized eigenvectors satisfy

$$v = \left((zI - A)^{-1}B,\ 1,\ (z^{-1}I - \bar{A})^{-1}\bar{B}\right)$$

and the eigenvalues are solutions of $G(z) = zG_*(z)$. The unitary matrix of eigenvectors Q is therefore

$$Q = \left[\begin{array}{c} Q_1 \\ \mathbf{1} \\ \bar{Q}_1 \end{array}\right]\Lambda \qquad (33)$$

where Λ is diagonal and

$$Q_1 = \left[\begin{array}{ccc} U(z_1) & \cdots & U(z_{2n+1}) \end{array}\right]. \qquad (34)$$

Thus from the Schur factorization (32), we obtain $2n{+}1$ sample points z_k and the matrix of sampled basis functions Q_1.

B. Discrete transform interpretation

The matrix Q in (33) defines a discrete transform matrix similar to the discrete Fourier transform (DFT). Let $F : \mathbf{C} \to \mathbf{C}$ be a function of the form

$$\begin{aligned} F(z) &= V_*(z)\left[\begin{array}{cc} 0 & y_1 \\ y_3^T & y_2 \end{array}\right]V(z) \\ &= U_*(z)y_1 + y_2 + y_3^T U(z) \end{aligned} \qquad (35)$$

where $y = (y_1, y_2, y_3) \in \mathbf{C}^{2n+1}$.

Suppose $x = (F(z_1), \ldots, F(z_{2n+1}))$ are the samples of F at the $2n+1$ points z_k defined by (30). From (33) and (34) we see that $x = Wy$ where $W = (Q\Lambda^{-1})^H$. Conversely, if a vector $x \in \mathbf{C}^{2n+1}$ contains the samples of a function of the form (35), then the coefficients of F are $y = W^{-1}x$ where $W^{-1} = Q\Lambda^H$. We can interpret this as a pair of forward and inverse discrete transforms. Multiplication with W^{-1} provides the forward transform and maps a vector of samples of F to the corresponding coefficients y. The inverse transform $x = Wy$ maps the coefficients to the samples.

As an example, if we take

$$A = \left[\begin{array}{cc} 0 & 0 \\ I & 0 \end{array}\right], \qquad B = \left[\begin{array}{c} 1 \\ 0 \end{array}\right], \qquad C = \left[\begin{array}{cc} 0 & 1 \end{array}\right]$$

and $D = 0$, then we have $U(z) = (z^{-1}, z^{-2}, \ldots, z^{-n})$. The matrix N in (31) is the identity. The matrix M, permuted as

$$\left[\begin{array}{ccc} A & 0 & B \\ C & 0 & D \\ 0 & I & 0 \end{array}\right] = \left[\begin{array}{cccc} 0 & 0 & 0 & 1 \\ I & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & I & 0 \end{array}\right],$$

is the circulant shift matrix of order $2n+1$. Therefore the eigenvalues z_k are the DFT sample points $e^{j2\pi k/(2n+1)}$, and the eigenvector matrix Q is the DFT matrix with its columns and rows permuted. We therefore retrieve as a special case the results on nonnegative trigonometric polynomials in [RV06].

n	uniform sampling (unbalanced)	uniform sampling (balanced)	from Schur decomposition
5	3.8 10^7	1.3 10^3	8.2
10	6.1 10^7	1.4 10^3	6.7
15	6.4 10^7	2.1 10^3	6.5
20	1.8 10^8	1.4 10^4	6.3
25	2.5 10^8	2.3 10^4	6.1

TABLE I

CONDITION NUMBER OF THE BASIS MATRICES Q_1 CONSTRUCTED BY THREE DIFFERENT METHODS.

C. Numerical example

In table I we compare the condition numbers of the matrices Q_1 constructed by three different methods. For each model order we randomly generate 500 controllable discrete-time state-space models of orders up to 25, using the Matlab command `drss`. The second column ('uniform sampling (unbalanced)') shows the maximum condition number if we sample uniformly on the unit circle, using the original state-space model. The third column ('uniform sampling (balanced)') shows the maximum condition number if we sample uniformly and use an input-balanced realization. The last column shows the condition numbers based on the input-balanced realization and the Schur decomposition. The last method results in better conditioned basis matrices, and the condition number appears not to increase with n.

The precise connection between the condition number of Q_1 and the condition number of the KYP-SDPs is obviously complicated to analyze. In our practical experience with the interior-point solver applied to problems generated with `drss`, uniform sampling often led to numerical difficulties that were not observed with the Schur decomposition sampling.

D. Multi-input systems

The construction in section V-A applies to state-space models with one input. For systems with more than one input, the $2n + m$ eigenvalues of (30) include m repeated eigenvalues $z = 1$, but the eigenvector matrix does not directly provide the matrix (29). However, we can note that if the sample points z_k are given (computed from (30) or selected otherwise), then the matrices $(z_kI - A)^{-1}B$ are easily computed (in $O(n^2m)$ operations), if A is first transformed to triangular form using a Schur decomposition.

VI. NUMERICAL EXAMPLES

A. Randomly generated KYP-SDPs

We first consider a family of randomly generated problems with one KYP-LMI constraint:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & M(x) + K^H(\Phi \otimes P)K \preceq 0, \end{array} \qquad (36)$$

where $c \in \mathbf{R}^p$ and K is defined in (6) with $A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^n$. The system orders n range from 20 to 1000; the dimension of the variable x is $p = n/5$. The state-space

n	Inequality form SeDuMi	 SDPT3	Standard form SeDuMi	 SDPT3	KYPD	This paper
20	0.12	0.20	0.03	0.03	0.02	0.01
30	0.72	0.59	0.05	0.05	0.04	0.02
45	5.7	3.3	0.14	0.10	0.09	0.04
70	64	33	0.63	0.31	0.41	0.12
100			2.6	0.91	1.2	0.31
150			10	3.6	4.7	0.94
220			44	15	18	2.9
350						11.1
500						30.2
750						98.2
1000						227

TABLE II

TIMES PER ITERATION (SEC.) OF DIFFERENT SOLVERS FOR KYP-SDPS.

models are constructed by randomly generating orthogonal matrices

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

The coefficients of the linear mapping $M(x)$ are randomly generated in such a way that the problem is strictly feasible. Instances that are dual infeasible (unbounded below) are discarded.

The table shows the CPU times in seconds on a 3.0 GHz Pentium 4 with 3.0 GB of memory, using Matlab 7.4 (R2007a). All times are averaged over five randomly generated instances. The number of iterations itself is not reported but was roughly 10–15 for all the algorithms. Blank entries in the table indicate that the simulation was aborted due to excessive execution time or an out-of-memory error.

Column 2 and 3 show the times per iteration for solving the inequality form SDP (36) using the general-purpose solver SeDuMi (version 1.1R3) and SDPT3 (version 4.0 beta) [TTT03], via the YALMIP interface [Löf04]. The YALMIP pre-processing time was excluded when calculating the times per iteration. Column 4 and 5 show the times per iteration using the SeDuMi and SDPT3 solvers directly for solving the equivalent standard form SDP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & V_*(z)(M(x)-X)V(z)=0 \;\; \forall z\in\mathcal{C} \\ & X\succeq 0 \end{array} \tag{37}$$

where $\mathcal{C}$ is a set of $2n+1$ sample points on the unit circle, generated as described in section V.

The next column, labeled KYPD, shows the results for the KYPD Matlab package [Wal03], which implements the algorithm of [VBW$^+$05]. This method requires a significant amount of processing before the start of the first iteration, and we excluded the preprocessing time when calculating the time per iteration.

The last column shows the results of a Matlab implementation of a primal-dual interior-point method that exploits rank-one structure in the standard form SDP (37). The algorithm is similar to [TTT03] and follows [VBW$^+$05, Appendix A] and [RV06].

Figure 1 shows the average times per iteration versus n. We can note that the complexity of the fast algorithm is almost exactly $O(n^3)$.

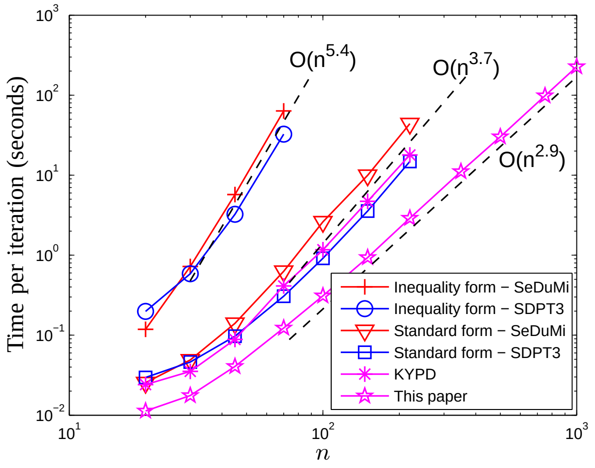

Fig. 1. Graph of the results in table II.

n	Inequality form SeDuMi	 SDPT3	Standard form SeDuMi	 SDPT3	This paper
20	0.89	1.2	0.08	0.16	0.02
30	8.0	6.1	0.18	0.17	0.03
45	77	44	0.71	0.54	0.09
70			3.8	2.1	0.24
100			11	7.5	0.61
150			60	38	1.9
220					5.7
350					23
500					63
750					205

TABLE III

TIMES PER ITERATION (SEC.) OF SOLVERS FOR GKYP-SDPS.

B. Randomly generated GKYP-SDPs

The second experiment (table III and figure 2) is based on a family of SDPs with one generalized KYP constraint,

$$\begin{array}{ll} \text{min.} & c^Tx \\ \text{s.t.} & K^H(\Phi\otimes P+\Psi\otimes Q)K^H+M(x)\preceq 0 \\ & Q\succeq 0, \end{array}$$

with Ψ defined as in (13). The problem data (including α and β) are randomly generated, and the problem is strictly feasible by construction. Dual infeasible problems are discarded. The system orders range from $n=20$ to $n=750$, and $p=n/5$.

VII. SUMMARY AND CONCLUSIONS

We have described a fast technique for solving SDPs derived from the discrete-time (generalized) KYP lemma. The key idea is to represent the frequency-domain inequality as a weighted sum of squares of rational functions, and then take a sufficient number of samples on the unit circle. The resulting SDP has a low-rank structure, which is easily exploited in interior-point algorithms. The technique is particularly efficient if the number of inputs is small: for single-input systems, the complexity is $O(n^3)$ operations per iteration if the number of optimization variables is of the same order as the system order n. This allows us to

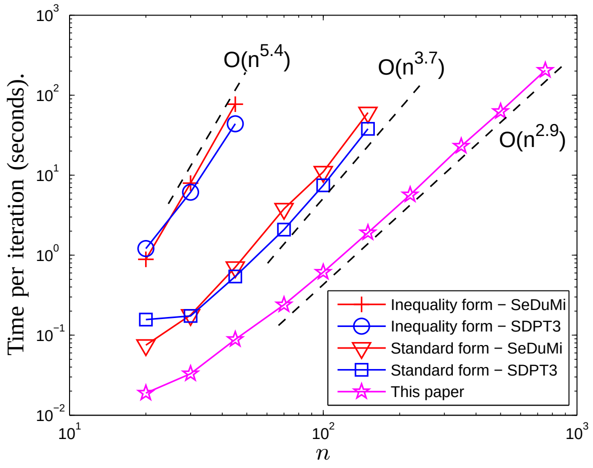

Fig. 2. Graph of the result in table III.

solve much larger problems than using standard SDP solvers applied to the matrix inequality form of the KYP lemma.

We have also presented an efficient method for selecting sample points on the unit circle and for computing the associated generalized discrete transform matrix.

Acknowledgements

We thank Tae Roh for his help in the implementation of the interior-point solver. We also thank the conference reviewers for their very detailed comments and helpful suggestions.

References

[AG93] P. J. Antsaklis and Z. Gao. Polynomial and rational matrix interpolation: theory and control applications. *International Journal of Control*, 58(2):349–404, 1993.

[AV02] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Mathematical Programming Series A*, 93:331–359, 2002.

[Bor99] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[BY05] S. J. Benson and Y. Ye. DSDP5 user guide — software for semidefinite programming. Technical Report ANL/MCS-TM-277, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005.

[BYZ00] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10:443–461, 2000.

[DR05] F. J. D'Amato and M. A. Rotea. LFTB: An efficient algorithm to bound linear fractional transformations. *Optimization and Engineering*, 6(2):177–201, 2005.

[GH03] J. Gillberg and A. Hansson. Polynomial complexity for a Nesterov-Todd potential-reduction method with inexact search directions. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 3, pages 3824–3829, 2003.

[GHNV03] Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren. Optimization problems over positive pseudopolynomial matrices. *SIAM Journal on Matrix Analysis and Applications*, 25(1):57–79, 2003.

[Hol06] C. W. J. Hol. *Structured Controller Synthesis for Mechanical Servo-systems. Algorithms, Relaxations and Optimality Certificates*. PhD thesis, Delft University of Technology, 2006.

[HSK99] B. Hassibi, A. H. Sayed, and T. Kailath. *Indefinite-Quadratic Estimation and Control. A Unified Approach to H^2 and H^∞ Theories*. Society for Industrial and Applied Mathematics, 1999.

[HV00] A. Hansson and L. Vandenberghe. Efficient solution of linear matrix inequalities for integral quadratic constraints. In *Proc. IEEE Conf. on Decision and Control*, pages 5033–5034, 2000.

[HVW05] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg, editors. *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer, 2005.

[IH05] T. Iwasaki and S. Hara. Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Transactions on Automatic Control*, 50(1):41–59, 2005.

[IW93] V. Ionescu and M. Weiss. Continuous and discrete-time Riccati theory: a Popov function approach. *Linear Algebra and Its Applications*, 193:173–209, 1993.

[KM01] C.-Y. Kao and A. Megretski. Fast algorithms for solving IQC feasibility and optimization problems. In *Proceedings of the American Control Conference*, volume 4, pages 3019–3024, 2001.

[KM03] C.-Y. Kao and A. Megretski. A new barrier function for IQC optimization problems. In *Proceedings of the American Control Conference*, volume 5, pages 4281–4286, 2003.

[KMJ00] C.-Y. Kao, A. Megretski, and U. Jönsson. An algorithm for solving optimization problems involving special frequency dependent LMIs. In *Proceedings of the American Control Conference*, volume 1, pages 307–311, 2000.

[KMJ03] C.-Y. Kao, A. Megretski, and U. Jönsson. Specialized fast algorithms for IQC feasibility and optimization problems. *Automatica*, 40(2):239–252, 2003.

[Löf04] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[LP04] 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*, pages 3154–3159, 2004.

[Nes00] Y. Nesterov. Squared functional systems and optimization problems. In J. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization Techniques*, pages 405–440. Kluwer Academic Publishers, 2000.

[Par99] P. A. Parrilo. On the numerical solution of LMIs derived from the KYP lemma. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2334–2338, 1999.

[Par00] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[RDV07] T. Roh, B. Dumitrescu, and L. Vandenberghe. Interior-point algorithms for sum-of-squares optimization of multidimensional trigonometric polynomials. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 905–908, 2007.

[RV06] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming and sum-of-squares representations of nonnegative polynomials. *SIAM Journal on Optimization*, 16:939–964, 2006.

[Stu99] J. F. Sturm. Using SEDUMI 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[TTT03] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Series B*, 95:189–217, 2003.

[VBW+05] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. H. Hansson, and T. Roh. Interior point algorithms for semidefinite programming problems derived from the KYP lemma. In D. Henrion and A. Garulli, editors, *Positive Polynomials in Control*, volume 312 of *Lecture Notes on Control and Information Sciences*. Springer Verlag, 2005.

[Wal03] R. Wallin. User's guide to kypd_solver. Technical Report LiTH-ISY-R-2517, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, 2003.

[YFK03] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (Semidefinite Programming Algorithm 6.0). *Optimization Methods and Software*, 18:491–505, 2003.