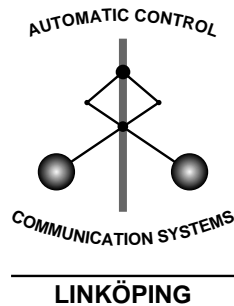


KYPD: A solver for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma

Ragnar Wallin, Anders Hansson

Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: <http://www.control.isy.liu.se>
E-mail: ragnarw@isy.liu.se, hansson@isy.liu.se

27th August 2004



Report no.: [LiTH-ISY-R-2622](#)

Submitted to IEEE Conference on Computer Aided Control
Systems Design, Taipei, Taiwan, 2004

Technical reports from the Control & Communication group in Linköping are
available at <http://www.control.isy.liu.se/publications>.

Abstract

Semidefinite programs derived from the Kalman-Yakubovich-Popov lemma are quite common in control and signal processing applications. The programs are often of high dimension making them hard or impossible to solve with general-purpose solvers. KYPD is a customized solver for KYP-SDPs that utilizes the inherent structure of the optimization problem thus improving efficiency significantly

Keywords: Semidefinite programming, Kalman-Yakubovich-Popov lemma

KYPD: A solver for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma

Ragnar Wallin and Anders Hansson

Abstract—Semidefinite programs derived from the Kalman-Yakubovich-Popov lemma are quite common in control and signal processing applications. The programs are often of high dimension making them hard or impossible to solve with general-purpose solvers. KYPD is a customized solver for KYP-SDPs that utilizes the inherent structure of the optimization problem thus improving efficiency significantly.

I. INTRODUCTION

Solving semidefinite programs derived from the Kalman-Yakubovich-Popov lemma is quite common in control and signal processing. In fact, some of the most important applications of SDPs in control involve KYP-SDPs. A far from complete list of applications include linear system design and analysis [1], [2], robust control analysis using integral quadratic constraints [3], [4], [5], [6], quadratic Lyapunov function search [7] and filter design [8].

In applications the size of the SDP is often very large making it hard or even impossible to solve with general-purpose software. However, the KYP-SDP has a very special structure that may be exploited to make efficient solvers. KYPD [9] is developed in YALMIP [10] and first utilizes the KYP-SDP structure to form an SDP with fewer variables. Then KYPD calls a general-purpose solver to actually solve the SDP. This solver can be any of YALMIPs interfaced primal-dual SDP-solvers, e.g. SeDuMi [11] or SDPT3 [12]. KYPD is also one of YALMIPs solvers.

There are other efficient solvers for KYP-SDPs. They are based on cutting plane methods [13], [14], [15], [16], interior-point methods with an alternative barrier [14] and interior-point methods combined with conjugate gradients [17], [18], [19]. Preliminary results on the approach used in KYPD has been presented in [18] and [20]

The KYP-SDP has the following structure

$$\begin{aligned} \min \quad & c^T x + \langle C, P \rangle \\ \text{s.t.} \quad & X = \mathcal{F}(P) + M_0 + \mathcal{G}(x) \geq 0 \end{aligned} \quad (1)$$

Where the inner product $\langle C, P \rangle$ is $\text{Trace}(CP)$,

$$\mathcal{F}(P) = \begin{bmatrix} A^T P + P A & P B \\ B^T P & 0 \end{bmatrix}$$

and

$$\mathcal{G}(x) = \sum_{k=1}^p x_k M_k$$

whith $A \in \mathbf{R}^{n \times n}$, $B \in \mathbf{R}^{n \times m}$, $C, P \in \mathbf{S}^n$ and $M_k \in \mathbf{S}^{n+m}$, $k = 0, 1, \dots, p$. In fact, there can be several constraints of the type above but for simplicity we only treat

R. Wallin and A. Hansson are with the Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden
ragnarw@isy.liu.se, hansson@liu.isy.se

SDPs with one constraint in this paper. A generalization is straightforward and incorporated in KYPD. Note that a standard LMI

$$M = M_0 + \sum_{k=1}^p x_k M_k$$

is a special case of the constraint with the size of A being 0×0 . Thus, in a general case, we can handle a mixture of KYP constraints and standard LMIs. At the moment KYPD cannot handle the term $\langle C, P \rangle$ in the objective function but this will be possible in the next version of the solver.

Before solving a KYP-SDP some preprocessing of the problem may be highly desirable or even necessary. The problem can in some cases be replaced by a reduced order equivalent problem resulting in a shorter computational time. Preprocessing can also lead to improved numerical conditioning of the problem. In Section II we declare what assumptions are usually made for KYP-SDPs, in Section III we explain the ideas behind the solver KYPD, in Section IV we show what preprocessing can be done if the assumptions are not fulfilled, in Section V it is shown how to use KYPD to solve semidefinite programs similar to KYP-SDPs, but with other stability regions, in Section VI we compare some solvers for KYP-SDPs, in Section VII possible future work is presented, and finally in Section VIII some conclusions are given.

II. ASSUMPTIONS

A common assumption is that the pair (A, B) is controllable. This implies that the operator \mathcal{F} has full rank, [20]. To see this we first notice that P will not change if we apply feedback, $\tilde{A} = A - BL$. The operator is transformed as

$$\begin{bmatrix} \tilde{A}^T P + P \tilde{A} & P B \\ B^T P & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix}^T \mathcal{F}(P) \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix} \quad (2)$$

As the pair (A, B) is controllable we can chose \tilde{A} Hurwitz. If the operator is equated with zero the (1,1)-block of the operator reads

$$\tilde{A}^T P + P \tilde{A} = 0$$

Because \tilde{A} is Hurwitz this implies that $P = 0$. Since the feedback can be interpreted as a congruence transformation on the operator we have shown that $\mathcal{F}(P) = 0$ implies $P = 0$, i.e. the operator has full rank.

In fact x is not changed either under the above congruence transformation, and the M_k -matrices are transformed analogously to $\mathcal{F}(P)$. This can be favourable as the numerical conditioning of the problem, in some cases, is improved

by this action. Another, standard assumption for SDPs is that the operator $\mathcal{A}(P, x)$ defined as

$$\mathcal{A}(P, x) = \mathcal{F}(P) + \mathcal{G}(x)$$

has full rank. The controllability assumption and the assumption that the operator \mathcal{A} has full rank can be relaxed. What is needed is stabilizability of the pair (A, B) , provided the range of C is in the controllable subspace of (A, B) . Moreover, if the operator $\mathcal{A}(P, x)$ does not have full rank it is possible to convert the problem to an equivalent reduced order problem for which the operator has full rank. More is said about this in Section IV.

III. KYPD–SOLVING AN EQUIVALENT DUAL PROBLEM

The solver KYPD is based on solving a problem equivalent to the dual problem of (1). The equivalent dual has considerably fewer variables than the primal SDP and can be solved using any primal-dual solver. Afterwards the original primal solution can be reconstructed. Solving the KYP-SDP in this fashion decreases the computational time from order n^6 to order n^4 . To reduce the number of variables in the dual some Lyapunov equations are solved. A set of basis matrices for the dual variable results. Solving the equivalent problem using a primal-dual SDP-solver yields a solution to both the equivalent dual problem and to an equivalent primal problem. Using the basis matrices and the solution to the equivalent primal problem it is possible to find x in (1) solving a small overdetermined but consistent system of linear equations. Having x the only remaining unknown variable in (1) is P . The matrix P can be found by solving a Lyapunov equation corresponding to the (1,1)-block of (1).

A. The dual problem

The dual of (1) is

$$\begin{aligned} \max \quad & -\langle M_0, Z \rangle \\ \text{s.t.} \quad & \mathcal{F}^*(Z) = C \end{aligned} \quad (3)$$

$$\mathcal{G}^*(Z) = c \quad k = 1, 2, \dots, p \quad (4)$$

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{bmatrix} \geq 0 \quad (5)$$

with

$$\mathcal{F}^*(Z) = AZ_{11} + Z_{11}A^T + BZ_{12}^T + Z_{12}B^T$$

$$\mathcal{G}_k^*(Z) = \langle M_k, Z \rangle$$

The operators \mathcal{F}^* and \mathcal{G}^* are the adjoint operators of \mathcal{F} and \mathcal{G} , respectively. As Z is a symmetric matrix of size $(n+m) \times (n+m)$ the dual problem has even more variables than the primal problem if

$$p < nm + \frac{m(m+1)}{2}$$

which is usually the case.

B. Basis matrices for the dual variable Z

To solve the dual SDP efficiently a reduction of the number of variables is necessary. We want to find a basis for Z that satisfies (3). The cost for computing the basis must be moderate. As the pair (A, B) is controllable the operator \mathcal{F} has full rank. Hence, it follows that there exists a mapping of dimension $k_{\max} = mn + m(m+1)/2$ that spans the nullspace of the adjoint operator \mathcal{F}^* . We can thus find a basis for Z which satisfies (3) and where the last k_{\max} basis matrices span the nullspace of \mathcal{F}^* . One such basis is

$$F_k = \begin{cases} \begin{bmatrix} E_{k(11)} & 0 \\ 0 & 0 \end{bmatrix} & k = 0 \\ \begin{bmatrix} E_{k(11)} & E_{k(12)} \\ E_{k(12)}^T & 0 \end{bmatrix} & k = 1, 2, \dots, mn \\ \begin{bmatrix} 0 & 0 \\ 0 & E_{k(22)} \end{bmatrix} & k = mn + 1, \dots, k_{\max} \end{cases}$$

The matrices $E_{k(12)}$ form the standard basis for unstructured $n \times m$ matrices, the matrices $E_{k(22)}$ form the standard basis for symmetric $m \times m$ matrices and each $E_{k(11)}$, $k = 1, 2, \dots, mn$, is related to $E_{k(12)}$ through

$$\mathcal{F}^*(F_k) = 0$$

Moreover, F_0 solves

$$\mathcal{F}^*(F_0) = C$$

Thus, to get the basis we have to solve $mn + 1$ Lyapunov equations with respect to $E_{k(11)}$. This can be done in order n^3 operations by transforming A to upper triangular form and in order n^2 if A is diagonalized. Hence, the total cost is at most of the order n^4 and n^3 , respectively. Now we can express Z as

$$Z = F_0 + \sum_{k=1}^{k_{\max}} z_k F_k \quad (6)$$

As the operator \mathcal{A} has full rank we know that the nullspace of \mathcal{A}^* is spanned by $k_{\max} - p$ basis matrices. Thus, we can further reduce the number of variables by finding basis matrices that also fulfil (4). This extra reduction will destroy the structure, for example sparsity, that may be present in the basis matrices F_k . As some solvers utilize sparsity this, normally very modest (p usually small), reduction may not be desirable. Hence, it is optional in KYPD. To make the reduction we note that (4), with Z expressed as in (6), can be written as

$$Gz = b \quad (7)$$

where the entries of G are

$$G_{ij} = \langle M_i, F_j \rangle,$$

the entries of b are

$$b_i = c_i - \langle M_i, F_0 \rangle$$

and $z = [z_1 \ z_2 \ \dots \ z_{k_{\max}}]^T$. If we make a QR factorization of G^T , [21]

$$G^T = [Q_1 \ Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where Q_1 and Q_2 are orthogonal, we can express z as

$$z = Q_1 R^{-T} b + Q_2 \tilde{z}$$

The vector \tilde{z} contains our new variables. The dual variable can be represented in the new basis

$$Z = (F_0 + H_0) + \sum_{k=1}^{k_{\max}-p} \tilde{z}_k H_k$$

where the H_k :s are defined by

$$\begin{bmatrix} \text{vec} H_0 & \dots & \text{vec} H_{k_{\max}-p} \end{bmatrix} \\ = \begin{bmatrix} \text{vec} F_1 & \dots & \text{vec} F_{k_{\max}} \end{bmatrix} [Q_1 R^{-T} b \quad Q_2]$$

The dual objective can be written as

$$-\langle M_0, (F_0 + H_0) + \sum_{k=1}^{k_{\max}-p} \tilde{z}_k H_k \rangle = -d_0 - d^T \tilde{z}$$

As the new basis matrices are linear combinations of the old basis matrices they satisfy both (3) and (4). Hence, the dual SDP can be reformulated as

$$\begin{aligned} \min \quad & d^T \tilde{z} \\ \text{s.t.} \quad & Z = (F_0 + H_0) + \sum_{k=1}^{k_{\max}-p} \tilde{z}_k H_k \geq 0 \end{aligned}$$

This standard SDP can be solved with any primal-dual SDP solver. The computational cost for this is typically of the order n^4 . If we are only interested in the objective value we are ready, but if we want the primal variables P and x we have to reconstruct them. To this end we need the fact that the dual solution, X , of the above SDP is actually also a solution to (1), see [20].

C. Reconstructing x and P

Primal-dual SDP solvers deliver the dual as well as the primal variable. Hence, we get X in (1), but we are really interested in P and x . It turns out that they can be reconstructed using the basis matrices. Remember that $\mathcal{F}^*(F_k) = 0$ for all $k = 1, 2, \dots, k_{\max}$. Hence, from the definition of adjoint operators it follows that

$$\langle F_k, \mathcal{F}(P) \rangle = \underbrace{\langle \mathcal{F}^*(F_k), P \rangle}_{=0} = 0$$

Thus we have

$$\begin{aligned} \langle F_k, X \rangle &= \langle F_k, \mathcal{F}(P) + M_0 + \mathcal{G}(x) \rangle \\ &= \langle F_k, M_0 \rangle + \langle F_k, \mathcal{G}(x) \rangle \\ &= \langle F_k, M_0 \rangle + \sum_{j=1}^p x_j \langle F_k, M_j \rangle \end{aligned}$$

This can be rewritten as

$$G^T x = g$$

where G is the same matrix as in (7) and

$$g_k = \langle F_k, X - M_0 \rangle$$

This is a overdetermined but consistent system of equations. Now, when we have x we can compute P by solving the Lyapunov function corresponding to the (1,1)-block of the constraint in (1).

IV. PREPROCESSING

It is common in applications that the assumptions given in Section II are not fulfilled. Hence, preprocessing the problem may be necessary. Preprocessing can improve computational efficiency and numerical conditioning. All of the preprocessing in this section can be performed by KYPD.

A. The pair (A, B) only stabilizable

Assume that the pair (A, B) is only stabilizable and that the range of C in (1) is in the controllable subspace of (A, B) . Then, it is possible to get x and the (1,1)-block of P by solving an equivalent reduced order problem, for which the pair (A, B) is controllable. The size of the problem is dependent on the size of the controllable subspace of (A, B) . As the problem is of reduced order this results in a shorter computational time. To get the solution to the original problem, that is the whole P -matrix, an additional Lyapunov equation has to be solved.

If a primal-dual interior-point solver is used to solve the KYP-SDP, this reduction may be necessary as no strictly feasible dual point exists if the pair (A, B) is only stabilizable. Let T be a unitary state transformation such that

$$\begin{aligned} \tilde{A} &= T^T A T = \begin{bmatrix} \tilde{A}_1 & \tilde{A}_{12} \\ 0 & \tilde{A}_2 \end{bmatrix} \\ \tilde{B} &= B T = \begin{bmatrix} \tilde{B}_1 \\ 0 \end{bmatrix} \end{aligned}$$

where the pair $(\tilde{A}_1, \tilde{B}_1)$ is controllable and \tilde{A}_2 is Hurwitz. Define

$$\begin{aligned} \tilde{P} &= T^T P T = \begin{bmatrix} \tilde{P}_1 & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_2 \end{bmatrix} \\ \tilde{C} &= T C T^T = \begin{bmatrix} \tilde{C}_1 & \tilde{C}_{12} \\ \tilde{C}_{12}^T & \tilde{C}_2 \end{bmatrix} \\ \tilde{M} &= \begin{bmatrix} T^T & 0 \\ 0 & I \end{bmatrix} (M_0 + \mathcal{G}(x)) \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} \tilde{M}_1 & \tilde{M}_{12} & \tilde{M}_{13} \\ \tilde{M}_{12}^T & \tilde{M}_2 & \tilde{M}_{23} \\ \tilde{M}_{13}^T & \tilde{M}_{23}^T & \tilde{M}_3 \end{bmatrix} \end{aligned}$$

Some calculations yield that (1) is equivalent to

$$\begin{aligned} & \min c^T x + \text{Tr}(\tilde{C}_1 \tilde{P}_1) + 2\text{Tr}(\tilde{C}_{12} \tilde{P}_{12}) + \text{Tr}(\tilde{C}_2 \tilde{P}_2) \\ & \text{s.t.} \begin{bmatrix} \tilde{A}_1^T \tilde{P}_1 + \tilde{P}_1 \tilde{A}_1 & \tilde{P} \tilde{B}_1 \\ \tilde{B}_1^T \tilde{P}_1 & 0 \end{bmatrix} + \begin{bmatrix} \tilde{M}_1 & \tilde{M}_{13} \\ \tilde{M}_{13}^T & \tilde{M}_3 \end{bmatrix} > 0 \\ & \tilde{A}_2 \tilde{P}_2 + \tilde{P}_2 \tilde{A}_2 + \tilde{P}_{12}^T \tilde{A}_{12} + \tilde{A}_{12}^T \tilde{P}_{12} + \tilde{M}_2 \\ & - \begin{bmatrix} \tilde{P}_1 \tilde{A}_{12} + \tilde{P}_{12} \tilde{A}_2 + \tilde{A}_1^T \tilde{P}_{12} + \tilde{M}_{12} \\ \tilde{B}_1^T \tilde{P}_{12} + \tilde{M}_{23}^T \end{bmatrix}^T \\ & \times \begin{bmatrix} \tilde{A}_1^T \tilde{P}_1 + \tilde{P}_1 \tilde{A}_1 + \tilde{M}_1 & \tilde{P} \tilde{B}_1 + \tilde{M}_{13} \\ \tilde{B}_1^T \tilde{P}_1 + \tilde{M}_{13}^T & \tilde{M}_3 \end{bmatrix}^{-1} \\ & \times \begin{bmatrix} \tilde{P}_1 \tilde{A}_{12} + \tilde{P}_{12} \tilde{A}_2 + \tilde{A}_1^T \tilde{P}_{12} + \tilde{M}_{12} \\ \tilde{B}_1^T \tilde{P}_{12} + \tilde{M}_{23}^T \end{bmatrix} > 0 \end{aligned}$$

As the range of C is in the controllable subspace of (A, B) we have that $\tilde{C}_2 = 0$ and $\tilde{C}_{12} = 0$. The matrices \tilde{P}_{12} and \tilde{P}_2 only appear in the latter constraint. Hence, it is possible to solve the reduced order problem involving \tilde{P}_1 and x for which the pair (A_1, B_1) is controllable and then solve a feasibility problem involving \tilde{P}_{12} and \tilde{P}_2 . Without loss of generality \tilde{P}_{12} may be chosen to zero. Then, the matrix \tilde{P}_2 can then be found by solving a Lyapunov equation.

B. Checking full rank of $\mathcal{A}(P, x)$

The full rank property of $\mathcal{A}(P, x)$ can be difficult to verify. We get a sufficient condition which is much easier to check if we make a change of variables. First, partition the M_k -matrices in (1) as

$$M_k = \begin{bmatrix} M_{1,k} & M_{12,k} \\ M_{12,k}^T & M_{2,k} \end{bmatrix}$$

Let P_k solve

$$A^T P_k + P_k A = M_{1,k}, \quad k = 1, 2, \dots, p$$

and let $\bar{M}_{12,k} = M_{12,k} - P_k B$, $k = 1, 2, \dots, p$. Then with $\bar{P} = P - \sum_{k=1}^p x_k P_k$ it holds that the KYP-LMI in (1) is fulfilled for some P and x if and only if \bar{P} and x satisfy

$$\begin{aligned} \tilde{\mathcal{A}}(\bar{P}, x) + M_0 &= \begin{bmatrix} A^T \bar{P} + \bar{P} A & \bar{P} B \\ B^T \bar{P} & 0 \end{bmatrix} \\ &+ \sum_{k=1}^p x_k \bar{M}_k + M_0 \geq 0 \end{aligned}$$

where

$$\bar{M}_k = \begin{bmatrix} 0 & \bar{M}_{12,k} \\ \bar{M}_{12,k}^T & M_{2,k} \end{bmatrix}, \quad k = 1, 2, \dots, p$$

If A is Hurwitz and the matrices \bar{M}_k are linearly independent $\mathcal{A}(P, x) = 0$ is equivalent to $\tilde{\mathcal{A}}(\bar{P}, x) = 0$. This is true as in the (1,1) block we have

$$A^T \bar{P} + \bar{P} A = 0$$

and since A is Hurwitz it follows that $\bar{P} = 0$. This implies that

$$\sum_{k=1}^p x_k \bar{M}_k = 0$$

and as \bar{M}_k , $k = 1, 2, \dots, p$ are linearly independent it follows that $x = 0$. By the definition of the change of variables it is then true that $(P, x) = 0$.

C. Linearly dependent M_k -matrices

If the \bar{M}_k -matrices are linearly dependent, then either the objective function is not bounded from below or the problem can be reduced to an equivalent problem with fewer variables and new linearly independent matrices. To this end define

$$\bar{M} = [\text{svec}(\bar{M}_1) \quad \text{svec}(\bar{M}_2) \quad \dots \quad \text{svec}(\bar{M}_p)]$$

Make a singular value decomposition of \bar{M}

$$\bar{M} = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

where Σ_1 has full column rank. Notice that we do not have to include the zero rows of \bar{M} which will reduce the computational complexity significantly. Define a change of variables for x via

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} x$$

Now clearly $\bar{M}x = \tilde{M}\tilde{x}_1$, where $\tilde{M} = U\Sigma_1$. Therefore we can rewrite the constraint in the variables \tilde{x}_1 and with a set of linearly independent matrices, \tilde{M}_k , given by the inverse symmetric vectorization of \bar{M} . The part of the objective function involving x can be rewritten as

$$c^T x = c^T \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} x = \tilde{c}_1^T \tilde{x}_1 + \tilde{c}_2^T \tilde{x}_2$$

where $\tilde{c}_1 = V_1^T c$ and $\tilde{c}_2 = V_2^T c$. As \tilde{x}_2 is not present in the constraint the objective function is bounded from below only if $\tilde{c}_2 = 0$.

V. USING KYPD FOR OTHER STABILITY REGIONS

KYPD is originally designed for analysis of continuous-time systems. However, it can be used for other stability regions. For the discrete-time case the structure of the KYP-LMI is

$$\begin{bmatrix} A^T P A - P & A^T P B \\ B^T P A & B^T P B \end{bmatrix} + M_0 + \sum_{k=1}^p x_k M_k \geq 0 \quad (8)$$

and for any stability region in the complex plane defined by

$$\mathcal{D} = \left\{ s \in \mathbf{C} : \begin{bmatrix} 1 \\ s \end{bmatrix}^H \underbrace{\begin{bmatrix} d_{11} & d_{12} \\ d_{12}^* & d_{22} \end{bmatrix}}_D \begin{bmatrix} 1 \\ s \end{bmatrix} < 0 \right\} \quad (9)$$

where D has one strictly negative and one strictly positive eigenvalue, we can define a generalized KYP-LMI as [22]

$$\begin{bmatrix} I & 0 \\ A & B \end{bmatrix}^T D \otimes P \begin{bmatrix} I & 0 \\ A & B \end{bmatrix} + M_0 + \sum_{k=1}^p x_k M_k \geq 0 \quad (10)$$

The continuous-time KYP-LMI, where the stability region is the left half-plane, is a special case with $d_{11} = 0$, $d_{12} =$

1 and $d_{22} = 0$. The discrete-time KYP-LMI, where the stability region is the unit disk, corresponds to $d_{11} = -1$, $d_{12} = 0$ and $d_{22} = 1$. If we start with a generalized KYP-LMI with

$$D = \begin{bmatrix} d_{11} & d_{12} \\ d_{12}^* & d_{22} \end{bmatrix}$$

but only have a solver for the region defined by

$$\bar{D} = \begin{bmatrix} \bar{d}_{11} & \bar{d}_{12} \\ \bar{d}_{12}^* & \bar{d}_{22} \end{bmatrix}$$

we can convert the problem to the latter region, with the same P and x by doing the transformation

$$\begin{aligned} \bar{A} &= (tA - yI)(zI - xA)^{-1} \\ \bar{B} &= (x\bar{A} + tI)B \\ \bar{M}_k &= T^T M_k T, \quad k = 0, 1, \dots, p \end{aligned}$$

where

$$T = (tz - xy) \begin{bmatrix} (zI - xA)^{-1} & x(zI - xA)^{-1}B \\ 0 & I \end{bmatrix}$$

From an eigenvalue decomposition of D and \bar{D}

$$\begin{aligned} D &= V \begin{bmatrix} \lambda^- & 0 \\ 0 & \lambda^+ \end{bmatrix} V^H \\ \bar{D} &= \bar{V} \begin{bmatrix} \bar{\lambda}^- & 0 \\ 0 & \bar{\lambda}^+ \end{bmatrix} \bar{V}^H \end{aligned}$$

we get t, x, y, z as

$$\begin{bmatrix} t & x \\ y & z \end{bmatrix} = V \begin{bmatrix} \sqrt{\frac{\bar{\lambda}^-}{\lambda^-}} & 0 \\ 0 & \sqrt{\frac{\bar{\lambda}^+}{\lambda^+}} \end{bmatrix} \bar{V}^H$$

If the inversion of the matrix $(zI - xA)$ is numerically ill-conditioned a transformation of the generalized KYP-LMI, similar to the one done in (2), will change the eigenvalues of A .

VI. COMPARISON OF SOLVERS

The LMI Control Toolbox [23] was for a long time considered the standard SDP solver for control related problems. It is still one of the most numerically stable solvers around but has been surpassed by many solvers when it comes to computational speed. In recent years the fast and reliable solver SeDuMi has become a popular choice. In this section we will compare how efficient some solvers are for KYP-SDPs. We will solve the SDPs with the LMI Control Toolbox, and SeDuMi, KYPD calling SeDuMi. Originally, the analytic cutting plane method solver for KYP-SDPs [15] was going to be part of the comparison. This was actually the fastest solver for most of the tests we run, but often it had to terminate early because of numerical problems. The relative error in the objective function was in some cases as high as 10^{-3} . However, we believe that there are remedies to the difficulties and have contacted the developers of this solver. If everything turns out well the analytic cutting plane method solver will be a formidable competitor.

A. Example

This is the same example as Example 2 in [24] except for the numerical values. It involves one KYP-LMI of size $(n+2) \times (n+2)$ and a scalar linear constraint. There are two x -variables and the number of variables in P is $n(n+1)/2$. The solvers stop when the objective function agrees with the optimal value with an absolute error less than $0.5 \cdot 10^{-6}$ or a relative error less than 10^{-6} . The results from the comparison is shown in Table I.

n	LMI Tool	SeDuMi	KYPD
10	504 s	2.2 s	2.6 s
18	746 s	4.5 s	4.2 s
32	TIME	14 s	10 s
56	TIME	384 s	75 s
100	TIME	SWAP	635 s

TABLE I

COMPUTATIONAL TIMES FOR THE THREE SOLVERS. TIME INDICATES THAT THE TIME PER ITERATION WAS MORE THAN AN HOUR, SWAP INDICATES THAT MEMORY SWAPPING OCCURED.

VII. FURTHER IMPROVEMENTS OF THE SOLVER

A general-purpose primal-dual solver applied to

$$\begin{aligned} \min & f^T z \\ \text{s.t.} & Gz = b \\ & Z = F_0 + \sum_{k=1}^{k_{\max}} z_k F_k \geq 0 \end{aligned}$$

generates iterates z, λ and $X \geq 0$, where λ and X are the dual variables. At each iteration a linear system of equations

$$-W \Delta X W - \sum_{k=1}^{k_{\max}} \Delta z_k F_k = R$$

$$\begin{bmatrix} \langle F_1, \Delta X \rangle \\ \vdots \\ \langle F_{k_{\max}}, \Delta X \rangle \end{bmatrix} + G^T \Delta \lambda = r_1$$

$$G^T \Delta \lambda = r_2$$

is solved. The positive definite matrix W and the righthand sides R, r_1 and r_2 change at each iteration and also depend on the particular algorithm used. These equations are solved by eliminating ΔX from the first equation and substituting $\Delta X = \sum_{k=1}^{k_{\max}} \Delta z_k W^{-1} F_k W^{-1} - R$ into the second. This yields

$$\begin{bmatrix} H & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} r_1 + h \\ r_2 \end{bmatrix}$$

where

$$\begin{aligned} H_{ij} &= \langle W^{-1} F_i W^{-1} F_j \rangle, & i, j &= 1, 2, \dots, k_{\max} \\ h_i &= \langle W^{-1} F_i W^{-1} R \rangle, & i &= 1, 2, \dots, k_{\max} \end{aligned}$$

The cost for solving this system of equations is proportional to n^3 but the cost for forming H is proportional to n^4 . However, if the F_k -matrices are low-rank the computational complexity only has to be cubic. If we, for example, require that the A -matrix is diagonal the basis matrices in (6) would have at most rank two. In Section 2.1 of [25] an algorithm for utilizing low rank is discussed where one starts by writing F_k as a sum of rank one matrices

$$F_k = \sum_{i=1}^r \alpha_{ik} v_{ik} v_{ik}^T$$

Note, however, that this is a case where the second equality constraint in the dual SDP should not be eliminated as this would destroy the low rank property of the basis matrices. Also notice that the algorithm described in [25] is a purely dual algorithm mostly suitable for max-cut problems having very sparse structure not only in the F_k -matrices but also in Z . However, some primal-dual solvers, which are known to be more efficient than purely dual solvers, already have routines exploiting low rank for special problems. In the code referenced in [12] there are specialized routines for max-cut problems, graph partitioning problems and Lovasz θ number problems. Unfortunately that version of the code does not handle equality constraints. Hence, it would not be possible to write customized code, applicable to KYP-SDPs, for that solver. Other structures on the A -matrix can also produce low rank basis matrices. One way to utilize the low-rank property, when forming H , is to write a customized solver. This is explained in [20] and produces very good results. However, the code referenced in [20] does not detect if the SDP is infeasible. Also in the code referenced in [23] low rank is utilized, but as has been seen in Section VI the code still cannot compete. Moreover, it is worth noting that numerical experiments indicate that the computational complexity for KYPD, if the A matrix is diagonalized, is already cubic for n up to approximately 100.

VIII. CONCLUSIONS

We have presented a structure-exploiting solver for KYP-SDPs. It decreases the computational complexity from n^6 to n^4 . Some preprocessing that may be beneficial is described. How to use the solver for analysis with respect to other stability regions is explained. Possible improvements for future versions of the solver are indicated.

REFERENCES

- [1] S. Boyd and C. Barratt, *Linear controller design: Limits of performance*. Prentice Hall, 1991.
- [2] H. Hindi, B. Hassibi, and S. Boyd, "Multiobjective H_2/H_∞ -optimal control via finite-dimensional Q-parametrization and linear matrix inequalities," in *Proceedings of the American Control Conference*, vol. 5, Philadelphia, Pennsylvania, USA, 1998, pp. 3244–3249.
- [3] A. Rantzer, "On the Kalman-Yakubovich-Popov lemma," *Systems & Control Letters*, vol. 28, no. 1, pp. 7–10, 1996.
- [4] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819–830, 1997.
- [5] U. Jönsson, "Robustness analysis of uncertain and nonlinear systems," Ph.D. dissertation, Lund Institute of Technology, Lund, Sweden, 1996.
- [6] V. Balakrishnan and F. Wang, "Efficient computation of a guaranteed lower bound on the robust stability margin for a class of uncertain systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2185–2190, November 1999.
- [7] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Philadelphia, USA: SIAM, 1994.
- [8] B. Alkire and L. Vandenberghe, "Convex optimization problems involving finite autocorrelation sequences," *Mathematical Programming Series A*, vol. 93, pp. 331–359, 2002.
- [9] R. Wallin, "User's guide to kypd_solver," Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Tech. Rep. LiTH-ISY-R-2517, 2003.
- [10] J. Löfberg, *YALMIP 3*, 2004. [Online]. Available: <http://control.ee.ethz.ch/~joloef/yalmip.msql>
- [11] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.
- [12] K. Toh, M. Todd, and R. Tütüncü, "SDPT3—a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999.
- [13] P. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA, 2000.
- [14] C. Kao, A. Megretski, and U. Jönsson, "A cutting plane algorithm for robustness analysis of time-varying systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, pp. 579–592, 2001.
- [15] C. Kao and A. Megretski, "Fast algorithms for solving IQC feasibility and optimization problems," in *Proceedings of the American Control Conference*, vol. 4, Arlington, Virginia, USA, 2001, pp. 3019–3024.
- [16] Y. Hachez, "Convex optimization over nonnegative polynomials: structured algorithms and applications," Ph.D. dissertation, Université catholique de Lovain, Louvain, Belgium, 2003.
- [17] A. Hansson and L. Vandenberghe, "A primal-dual potential reduction method for integral quadratic constraints," in *Proceedings of the American Control Conference*, Arlington, Virginia, USA, 2001, pp. 3013–3017.
- [18] R. Wallin, H. Hansson, and L. Vandenberghe, "Comparison of two structure exploiting algorithms for integral quadratic constraints," in *4th IFAC Symposium on Robust Control Design*, Milan, Italy, 2003.
- [19] 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*, Maui, Hawaii, USA, 2003.
- [20] L. Vandenberghe, V. Balakrishnan, R. Wallin, and A. Hansson, *Positive Polynomials in Control*, ser. Lecture Notes on Control and Information Sciences. Springer Verlag, ch. Interior-point algorithms for semidefinite programming problems derived from the KYP lemma, to be published.
- [21] J. Nocedal and S. Wright, *Numerical Optimization*. New York, USA: Springer Verlag, 1999.
- [22] D. Henrion, M. Šebek, and V. Kučera, "Positive polynomials and robust stabilization with fixed order controllers," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1178–1186, 2003.
- [23] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali, *LMI Control Toolbox*. The MathWorks Inc., 95.
- [24] 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*, vol. 3, Phoenix, Arizona, USA, 1999, pp. 2334–2338.
- [25] S. J. Benson and Y. Ye, "DSDP3: Dual Scaling Algorithm for General Positive Semidefinite Programming." [Online]. Available: citeseer.ist.psu.edu/444754.html