Technical report from Automatic Control at Linköpings universitet

A Decomposition Algorithm for KYP-SDPs

Rikard Falkeborn, Anders Hansson

Division of Automatic Control

E-mail: falkeborn@isy.liu.se, hansson@isy.liu.se

19th October 2009

Report no.: LiTH-ISY-R-2919

Accepted for publication in European Control Conference (ECC) 2009, Budapest, Hungary

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

In this paper, a structure exploiting algorithm for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma, where some of the constraints appear as complicating constraints is presented. A decomposition algorithm is proposed, where the structure of the problem can be utilized. In a numerical example, where a controller that minimizes the sum of the H_2-norm and the H_∞-norm is designed, the algorithm is shown to be faster than SeDuMi and the special purpose solver KYPD.

Keywords: Optimization, Optimization Algorithms

A Decomposition Algorithm for KYP-SDPs

Rikard Falkeborn and Anders Hansson

***Abstract*— In this paper, a structure exploiting algorithm for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma, where some of the constraints appear as complicating constraints is presented. A decomposition algorithm is proposed, where the structure of the problem can be utilized. In a numerical example, where a controller that minimizes the sum of the H_2-norm and the H_∞-norm is designed, the algorithm is shown to be faster than SeDuMi and the special purpose solver KYPD.**

I. Introduction

Semidefinite programs (SDPs) arise in many applications in control and signal processing [1]. In many cases, the programs that need to be solved involve large matrix variables that can make the computational burden very large. In some cases, the structure of the problem can be utilized to reduce the computational demands.

SDPs derived from the Kalman-Yakubovich-Popov lemma (KYP-SDPs) [2] is one such example, where tailor-made solvers have been successfully developed, see for example [3], [4], [5]. In this paper an algorithm for solving KYP-SDPs where some of the constraints appear as *complicating constraints*, i.e. constraints which are such that the optimization program would have been much easier to solve, if they were not present. More specifically, we treat optimization programs with the following structure

$$\begin{aligned}\min_{x_i,P_i}\ & \sum_{i=0}^{N}\langle C_i,P_i\rangle + c_i^T x_i\\ \text{s.t. } & \mathcal{F}_0(P_0)+M_0+\mathcal{G}(x)\preceq 0\\ & \mathcal{F}_i(P_i)+M_{i_0}+\mathcal{H}_i(x_i)\preceq 0,\quad i=1,\ldots,N\end{aligned}\tag{1}$$

where $C_i, P_i \in \mathbf{S}^{n_i}$, and where $\mathbf{S}^n$ is the space of symmetric matrices of dimension $n\times n$. The inner product $\langle C_i, P_i\rangle$ is defined as $\operatorname{Trace} C_iP_i$. We define the operators

$$\mathcal{F}_i(P_i)=\begin{bmatrix}A_i^TP_i+P_iA_i & P_iB_i\\ B_i^TP_i & 0\end{bmatrix}, i=0,\ldots,N,$$

where $A_i\in\mathbf{R}^{n_i\times n_i}$, $B_i\in\mathbf{R}^{n_i\times m_i}$,

$$\mathcal{H}_i(x_i)=\sum_{j=1}^{p_i}M_{i_j}x_{i_j}, i=1,\ldots,N$$

where $M_{i_j}\in\mathbf{S}^{n_i+m_i}$, $x_i\in\mathbf{R}^{p_i}$, and where x_{i_j} denotes the jth component of the vector x_i. Let us also define the operators $\mathcal{G}_i(x_i)=\sum_{j=1}^{p_i}x_{i_j}M_{0_{i_j}}$, and $\mathcal{G}(x)=\sum_{i=0}^{N}\mathcal{G}_i(x_i)$ where $x=(x_0,x_1,\ldots,x_N)$.

We assume, that the pairs (A_i,B_i) are controllable. This implies that the operators $\mathcal{F}_i(P_i)$ have full rank. This can be relaxed to stabilizability of the pair (A_i,B_i) provided that the range of C_i is in the controllable subspace of (A_i,B_i), see [3] for details. We also assume the optimal value of (1) exists and is finite.

The constraint involving $\mathcal{G}(x)$ is a complicating constraint, since without it, the problem would decompose into several smaller problems, which all could be solved separately. Hence, a decomposition algorithm would be suitable to use. We remark that the algorithm we propose can easily be generalized to have several complicating constraints.

This type of programs appear for example in robust control analysis using integral quadratic constraints [6], and linear system design and analysis [7], [8].

We remark that a standard linear matrix inequality (LMI)

$$M=M_0+\sum_{k=1}^{n}x_kM_k\preceq 0$$

is a special case of a KYP-LMI with the size of A being 0×0. Hence we can handle a mixture of KYP constraints and standard LMIs. In fact, as we will see in Section III, in some cases, where the complicating constraint is a regular LMI, the ability to solve the regular LMI using a standard solver and use tailormade solvers for the KYP-constraints can, as we will see, reduce the computational time.

A. Eliminating variables from KYP-SDPs

We here show how the structure of the KYP-SDP can be utilized to eliminate dual variables and thus formulate a smaller problem which can be solved in less time. The details can be found in [3]. For simplicity, we only show how the elimination is done for the case with one KYP-constraint, but a generalization is straightforward. Consider the problem

$$\begin{aligned}\min_{x,P}\ & \langle C,P\rangle + c^Tx\\ \text{s.t. } & \mathcal{F}(P)+M_0+\mathcal{G}(x)\preceq 0.\end{aligned}\tag{2}$$

The dual formulation of this problem is [3]

$$\begin{aligned}\max_{Z}\ & \langle M_0,Z\rangle\\ \text{s.t. } & \mathcal{F}^*(Z)+C=0\\ & \mathcal{G}^*(Z)+c=0\\ & Z=\begin{bmatrix}Z_{11} & Z_{12}\\ Z_{12}^T & Z_{22}\end{bmatrix}\succeq 0\end{aligned}\tag{3}$$

where the adjoint operators $\mathcal{F}^*(Z)$ and $\mathcal{G}^*(Z)$ are defined as

$$\begin{aligned}\mathcal{F}^*(Z)&=AZ_{11}+Z_{11}A^T+BZ_{12}^T+Z_{12}B^T\\ \mathcal{G}^*(Z)&=\begin{bmatrix}\langle M_1,Z\rangle\\ \vdots\\ \langle M_p,Z\rangle\end{bmatrix}.\end{aligned}\tag{4}$$

By computing a basis for the nullspace of the adjoint operator $\mathcal{F}^*(Z)$, it is possible to reduce the number of variables. In [3] it is shown how such a basis can be easily found by solving Lyapunov equations which can be done in an efficient and numerically stable way [9].

When the reduction of variables is done, the reduced dual problem can be solved, and when a solution is found, the original optimal variables can be computed, see [3] for details.

II. Lagrangian relaxation

Optimization programs with complicating constraints have been extensively studied within the field of optimization and operations research and is usually tackled with different decomposition algorithms. We will employ one of these, *Lagrangian relaxation* [10] pioneered in [11], [12], and show how the specific structure of (1) can be used to speed up the computational time.

A. Forming the Lagrangian decomposes the problem

In order to make the presentation more streamlined, we derive the algorithm in a slightly more general form than (1). Consider the problem

$$\begin{aligned} \min_X \ & \sum_{i=0}^N \langle f_i, X_i \rangle \\ \text{s.t. } & g(X) = g_c + \sum_{i=0}^N g_i(X_i) \preceq 0 \qquad (5a) \\ & h_i(X_i) \preceq 0, \quad i = 1, \dots, N, \qquad (5b) \end{aligned}$$

where g_i and h_i are assumed to be symmetric linear matrix functions of X, and where f_i is assumed to be a symmetric matrix. We let $X_i = (P_i, x_i)$ and $X = (X_0, \dots, X_N)$.

The Lagrangian function to (5) with respect to the complicating constraint (5a) is [13]

$$L(X, Z) = \sum_{i=0}^N \langle f_i, X_i \rangle + \left\langle Z, g_c + \sum_{i=0}^N g_i(X_i) \right\rangle. \qquad (6)$$

Hence, if we let $S_i = \{X_i : h_i(X_i) \preceq 0\}$, the dual function to (5) is

$$h(Z) = \min_{X_0, X_i \in S_i} L(X, Z). \qquad (7)$$

For fixed Z, the minimization can be formulated as

$$\min_{X_0, X_i \in S_i} \left\{ \langle X_0, f_0 + g_0^*(Z) \rangle + \langle Z, g_c \rangle + \sum_{i=1}^N \langle f_i, X_i \rangle + \langle g_i^*(Z), X_i \rangle \right\} \qquad (8)$$

where $g_i^*(Z)$ denotes the dual of $g_i(X_i)$. In order for the minimal value to be bounded from below when minimizing $L(X, Z)$ with respect to X_0, we have to require that Z fulfills the constraint

$$g_0^*(Z) + f_0 = 0. \qquad (9)$$

For the problem studied in this paper, this corresponds to

$$\begin{aligned} \mathcal{F}_0^*(Z) + C_0 &= 0 \\ \mathcal{G}_0^*(Z) + c_0 &= 0, \end{aligned} \qquad (10)$$

where the adjoint operators $\mathcal{F}_0^*(Z)$ and $\mathcal{G}_0^*(Z)$ are defined as in (4). After minimizing with respect to X_0, we obtain, since we have required Z to fulfill the constraint (9),

$$\begin{aligned} h(Z) = \min_{X_0, X_i \in S_i} L(X_0, \dots, X_N, Z) = \\ \sum_{i=1}^N \min_{X_i \in S_i} \langle f_i + g_i^*(Z), X_i \rangle . \end{aligned} \qquad (11)$$

Hence, for fixed Z, the problem of minimizing the Lagrangian under the constraints (5b) is a separable problem in $X_i = (P_i, x_i)$ for $i = 1, \dots, N$. In our problem, each minimization is equal to

$$\begin{aligned} \min_{x_i, P_i} \ & \langle C_i, P_i \rangle + \bar{c}_i^T x_i \\ \text{s.t. } & \mathcal{F}_i(P_i) + M_{i_0} + \mathcal{H}_i(x_i) \preceq 0, \end{aligned} \qquad (12)$$

where $\bar{c} = c_i + \mathcal{G}_i^*(Z)$.

We remark that $h(Z)$ is less than or equal to $\sum_{i=0}^N \langle f_i, Z \rangle$ for $Z \succeq 0$ which fulfills (9).

One should not solve the separable optimization programs as they stand, but use a tailor-made solver for KYP-SDPs. We take the same approach as in [3] and eliminate variables in the dual problem as described in Section I-A. Note that the reduction of the dual variables for the ith subproblem can be reused if we need to solve the same subproblem but for a different Z. This will be used in Section II-B.1.

The boundedness of the subproblems (12) can be an issue and is ensured by bounding the optimal value of (12), see Appendix. This is done in the numerical example we present in Section III.

B. Updating Z

The dual function $h(Z)$ is a lower bound on the optimal value of (5). We know that if the problem is convex and Slaters condition [13] holds, the maximum of the dual function is equal to the optimal objective value of the optimization problem. Hence, we want to maximize the dual function to get a good lower bound on the optimal value. That is, we want to solve the optimization problem

$$\max_Z h(Z) = \max_Z \min_X L(X, Z). \qquad (13)$$

A problem is that we do not have an explicit expression for the dual function in (11) since it depends on X_i.

1) Dual formulation: To be able to compute a lower bound on the optimal objective function, [14] proposes a *tangential approximation* method, first outlined in [15]. We note that, if we have solved the Lagrangian problem r times for r different fixed Z^k satisfying (9), and then have r

different solutions X_i^k, the functions

$$h(Z^k) + \left\langle Z - Z^k, g_c + \sum_{i=1}^{N} g_i(X_i^k) \right\rangle = \sum_{i=1}^{N} \left\langle f_i, X_i^k \right\rangle + \left\langle Z, g_c + \sum_{i=1}^{N} g_i(X_i^k) \right\rangle \quad (14)$$

are *linear supporting functions* to $h(Z)$ at Z^k. A linear supporting function to $h(Z)$ at Z^k is a linear function which never lies below $h(Z)$ and contacts it at Z^k. We can therefore use the piecewise linear function

$$v^r(Z) = \min_{1 \le k \le r} \sum_{i=1}^{N} \left\langle f_i, X_i^k \right\rangle + \left\langle Z, g_c + \sum_{i=1}^{N} g_i(X_i^k) \right\rangle \quad (15)$$

as an approximation to $h(Z)$. Instead of maximizing $h(Z)$, we maximise the approximation v^r. By using the epigraph formulation of (15), an equivalent problem is

$$\begin{aligned} \max_{\sigma, Z} \ & \sigma \\ \text{s.t.} \ & \sum_{i=1}^{N} \left\langle f_i, X_i^k \right\rangle + \left\langle Z, g_c + \sum_{i=1}^{N} g_i(X_i^k) \right\rangle \ge \sigma, \\ & \qquad\qquad k = 1, \ldots, r \\ & f_0^* + g_0^*(Z) = 0 \\ & Z \succeq 0. \end{aligned} \quad (16)$$

Since it is not certain that the optimal value of (16) is bounded, it is necessary to add the constraint

$$\sigma < \sigma_{\max} \quad (17)$$

in order to get a bounded optimal value. Here $\sigma_{\max}$ is chosen such that $\sigma_{\max}$ is larger than the optimal value of the original problem. We remark that this value is unknown, and one should pick a large value of $\sigma_{\max}$.

In Figure 1, tangential approximation of the concave function $h(Z)$ is shown.

If $h(Z^r) = L(X_1^r, \ldots, X_N^r)$, we know that we have found the optimum. If not, we can use the "new" Z^{r+1} to solve the subproblems again and obtain a new X_i^{r+1}. In practice, it is not possible to find the exact optimum due to numerics, and we have to settle for when $h(Z^r) - L(X_1^r, \ldots, X_N^r)$ is less than ϵ. This implies that the duality gap is ϵ or less.

An iterative procedure to solve the original problem (1) can be outlined as follows, starting from iteration r.

1) Solve the problem (16), and obtain an optimal solution Z^{r+1}.
2) Solve the Lagrangian subproblems (12) and obtain an optimal solution $\left(X_0^{r+1}, \ldots, X_N^{r+1}\right)$.
3) If $|h(Z^{r+1}) - L(X_0^{r+1}, \ldots, X_N^{r+1})| < \epsilon$, terminate.
4) Add a linear constraint like in (16) and return to step 1.

Here we can note that σ is a non-increasing function in iterations r, but no such guarantees can be given for L [14, p. 433].

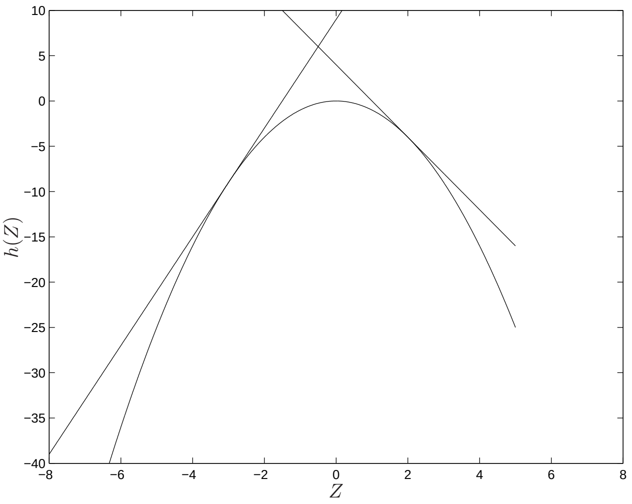

Fig. 1. Tangential approximation of $h\,(Z)$.

2) Utilizing the structure of Z: Solving the problem (16) as it stands is not a good idea. Instead, it is possible to reformulate the optimization problem as an equivalent problem with fewer variables. We take same approach as described in Section I-A. However, note that the computation of the basis can be reused in all iterations.

3) Primal formulation: It is also possible to formulate the dual of (16) and solve that problem instead. The dual problem of (16) is, note that P_i^k and x_i^k are fixed,

$$\begin{aligned} \min_{y,Q} \ & c_0^T y_0 + \langle C_0, Q \rangle + \sum_{k=1}^{r} \sum_{i=1}^{N} \left(c_i^T x_i^k + \langle C_i, P_i^k \rangle \right) y_k \\ \text{s.t.} \ & \mathcal{F}_0(Q) + \mathcal{G}_0(y_0) + M_0 \underbrace{\sum_{k=1}^{r} y_k}_{=1} + \sum_{k=1}^{r} \sum_{i=1}^{N} \mathcal{G}_i(x_i^k) y_k \preceq 0 \\ & \sum_{k=1}^{r} y_k = 1 \\ & y_k \ge 0, \quad k = 1, \ldots, r \end{aligned} \quad (18)$$

where $y_0 \in \mathbf{R}^{p_0}$, $Q \in \mathbf{S}^{n_0}$, and where $y_i \in \mathbf{R}, i = 1, \ldots, r$. The dual variable Z needed in the next iteration can be returned by primal-dual solvers. In order to avoid the equality constraint we can eliminate the "last" variable y_r and replace it with $1 - \sum_{k=1}^{r-1} y_k$. Thus, tailor-made solvers for KYP-problems [3], [4] can be used to solve the problem.

Note that if a feasible point for iteration r is known, then a feasible point for $r+1$ is also known (by letting $y_{r+1} = 0$). Hence, if the underlying solver uses a method which requires a strictly feasibly point, the first phase where such a point is found can be skipped. This might save some computational time.

In the first iterations of the algorithm, it is not certain that the LMI (18) is feasible since there are only a few x_i^k. This corresponds to the case where σ is unbounded from above in (16). A remedy to this is to include the constraint (17) when formulating the dual of (16). The dual is then

$$\begin{aligned}\min_{y,Q,w}\;& w\sigma_{\max} + c_o^T y_0 + \langle C_0, Q\rangle + \\ & \sum_{k=1}^{r}\sum_{i=1}^{N}\left(c_i^T x_i^k + \langle C_i, P_i^k\rangle\right) y_k \\ \text{s.t. }& \mathcal{F}_0(Q) + \mathcal{G}_0(y_0) + M_0\sum_{k=1}^{r} y_k + \sum_{k=1}^{r}\sum_{i=1}^{N}\mathcal{G}_i(x_i^k) y_k \preceq 0 \\ & \sum_{k=1}^{r} y_k + w = 1 \\ & y_k \geq 0, \quad k = 1,\dots,r \\ & w \geq 0\end{aligned} \tag{19}$$

which, if $w = 0$ is the same problem as (18). If $w = 0$ in the solution to (19), the problem (18) is feasible. For numerical reasons, it is better to switch to solving (18) when the optimal value of w is zero. We remark that for problems where n_0 is small compared to m_0, that is, when the number of columns in A_0 is small compared to the number of columns in B_0, it may be beneficial to solve the problem in (19) instead of its dual (16). We also remark that one should bound the optimal value of the problem in (19) in the same fashion as we show in the Appendix.

III. Numerical example

The efficiency of the algorithm is investigated in a numerical example borrowed from [8], see also [7], [16], and it is compared to the generic solver SeDuMi 1.21, [17], and the tailor-made solver for KYP-problems, KYPD, version 1.2 [3]. All solvers were interfaced via YALMIP version 3, release 20090505, [18].

Using the Youla parametrization [19], the set of all stable closed-loop plants of a system can be written as

$$G_{cl} = T_1 + T_2 Q T_3 \tag{20}$$

where T_i are stable and depends on the system matrices and Q is an arbitrary stable function transfer matrix. The corresponding controller is then, assuming positive feedback

$$K = Q(I + GQ)^{-1}. \tag{21}$$

By restricting Q to lie in a finite-dimensional subspace in such a way that the parameters enter linearly, i.e. $Q = Q(\theta) = \sum_{i=1}^{n} Q_i\theta_i$, convex constraints on the closed loop system result in convex optimization problems where the optimum can be easily found by polynomial time methods [13].

In order to formulate constraints on the closed loop system as LMIs, it is necessary to write (20) in state space form. It is also necessary that this state space realization has all θ_i in the C and D matrix in order for the constraints to be convex. The realization of (20) is typically obtained using system Kronecker products [20], [7] and yields a system of a much higher order than the original plant order. The resulting closed loop system matrices can then be written as A, B, $C(\theta)$ and $D(\theta)$ where $C(\theta)$ and $D(\theta)$ depends affinely on the parameters chosen in the parametrization of $Q(\theta)$.

Many design specifications can be cast as LMIs. We can mention for example specifications and constraints on the H_2-norm, H_∞-norm, dispassivity constraints and the location of the closed loop poles. Some of these are in the form of KYP-SPDs. As an example, we will solve the joint minimization of the H_2-norm and the H_∞-norm of a system [7], [8]. The design problem results in the optimization problem

$$\min_{\gamma^2,X,\theta,P}\; \operatorname{Trace} X + \gamma^2$$

$$\text{s.t.}\; \begin{bmatrix} X & C(\theta)W^{\frac{1}{2}} \\ W^{\frac{1}{2}}C(\theta)^T & I\end{bmatrix} \succeq 0 \tag{22a}$$

$$\begin{bmatrix} A^TP + PA & PB & 0 \\ B^TP & 0 & 0 \\ 0 & 0 & 0\end{bmatrix} + \begin{bmatrix} 0 & 0 & C(\theta)^T \\ 0 & -\gamma^2 I & D(\theta)^T \\ C(\theta) & D(\theta) & -I\end{bmatrix} \preceq 0 \tag{22b}$$

where X and P are symmetric positive definite matrices, A, B, $C(\theta)$ and $D(\theta)$ are the state space matrices of the closed loop system, W is the controllability gramian of the system, i.e. W solves

$$AW + WA^T + BB^T = 0. \tag{23}$$

This problem can be transformed to the form (1) where the complicating constraint is (22a) since without it, tailor-made solvers for KYP-problems that use the dual formulation could solve the problem. However, formulating the dual of the entire problem would create a very large matrix variable corresponding to the LMI (22a). Hence, using a decomposition algorithm that alternates between solving the KYP-SDP in it's dual form, after eliminating variables and solving the H_2-LMI (22a) in it's primal form with much fewer variables than the dual form will lower the computational burden.

To test the algorithm, we create random systems using `rss` in Matlab with one input and one output. We chose the Youla parameter $Q(\theta)$ to be

$$Q(\theta) = \sum_{i=0}^{n_Q}\frac{\theta_i}{(s + 0.5)^i} \tag{24}$$

as suggested in [19]. We choose to have $n_Q = 10$ in all examples. The reason for letting $n_Q = 10$ is seen in Figure 2, where the normalized objective value is plotted for 10 different systems of state dimension 20 with increasing n_Q. The computations were done using SeDuMi, i.e. no decomposition was used. We stopped the increase in n_Q when the objective function was not improving more than 0.1% or if SeDuMi ran into numerical problems that were too severe.

We solve the resulting LMIs using our algorithm, the tailor-made solver KYPD calling SeDuMi and the generic solver SeDuMi. All computations were terminated when the absolute error or the relative error was less than 10^{-3}. We create 10 different systems for each n, where n is the number of states, and let n vary. The number n_Q was set to 10 for all computations. The computational times were averaged and

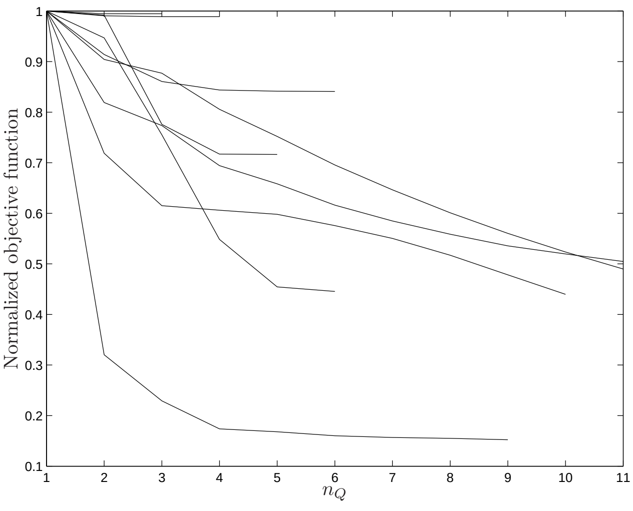

Fig. 2. Normalized objective values for different sizes of $Q(\theta)$.

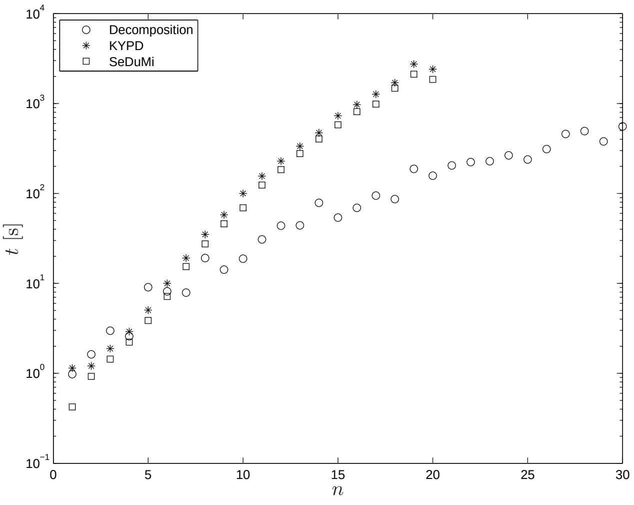

Fig. 3. Averaged computational times for the controller synthesis problem of SISO systems. Note that n denotes the number of states in the original plant and not the size of the matrices that are involved in the actual computations.

the averaged times can be seen in Figure 3. We remark that in practice, n would include both the states of the original plant as well as states from various weighting filters that is commonly used in H_∞-synthesis. We also remark that the number n in Figure 3 is the number of states in the original plant, not the dimension of the A-matrix in (1). As an example, for $n = 20$, the A-matrix is has 90 rows and columns.

In Figure 3 we see that the decomposition algorithm in this case outperforms both the tailormade solver KYPD and the generic solver SeDuMi. It can also be seen that the tailor-made solver for KYP-problems actually performs worse than the generic solver SeDuMi. This can be explained as follows. SeDuMi solves an optimization problem where the majority of variables come from P which is $n \times n$ and therefore yield $(n^2 + n)/2$ variables assuming the order of $Q(\theta)$ and the number of variables in X, which is determined by the number of outputs, can be neglected. KYPD formulates the dual problem and eliminates variables. For a constraint of the type

$$\mathcal{F}(P) + M_0 + \mathcal{G}(x) \preceq 0 \tag{25}$$

the number of remaining dual variables, after the reduction, is $mn + m\,(m+1)\,/2$ where m is the number of rows in the B matrix. This is usually a lot lower than $\left(n^2 + n\right)/2$. However, for this numerical example, we also have the constraint (22a). When we formulate the dual problem of the numerical example, there will be a dual variable Z_2 corresponding the LMI (22a). This variable has no special structure that is used in KYPD, and no variables will be reduced, adding an extra matrix variable with $\Big((n+p)^2 + n + p\Big)/2$ scalar variables, where p is the number of outputs to the system. This is even more variables than the original primal formulation, and hence KYPD will usually be slower than SeDuMi for this specific example.

IV. Conclusions

In this paper, a structure exploiting algorithm for KYP-SDPs where some of the constraints appear as complicating constraints is proposed. The structure of the KYP-SDP is utilized to reduce the computational complexity. The algorithm is basically the Kelley-Cheney-Goldstein cutting plane method [11], [12]. The convergence of the method is established if the technical but important condition that Z_k is bounded holds, see for example [10]. A sufficient condition for this is that there exist an interior point for the problem (19) [21, Thm. 4.1.3]. That this is indeed the case for all possible problems is still an open question. However, our experience is that we have had no problems with convergence using the algorithm.

We remark that the worst case complexity of the number of iterates is proportional to $\mathcal{O}(1/\epsilon^m)$ where m is the dimension of the dual variable, which is very poor.

In a numerical example, it is shown that it is beneficial to use the proposed algorithm in some cases. It is advantageous to use the algorithm in cases where one or more constraints is associated with a large dual variable that has no specific structure that can be exploited.

References

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, ser. Studies in Applied Mathematics. SIAM, 1994, vol. 15.

[2] A. Rantzer, "On the Kalman-Yakubovich-Popov lemma," *Systems and Control Letters*, vol. 28, no. 1, pp. 7 – 10, 1996.

[3] R. Wallin, A. Hansson, and J. H. Johansson, "A structure exploiting preprocessor for semidefinite programs derived from the kalman-yakubovich-popov lemma," *Automatic Control, IEEE Transactions on*, vol. 54, no. 4, pp. 697–704, April 2009.

[4] R. Wallin, C.-Y. Kao, and A. Hansson, "A cutting plane method for solving KYP-SDPs," *Automatica*, vol. 44, no. 2, pp. 418 – 429, 2008.

[5] C.-Y. Kao, A. Megretski, and U. Jönsson, "Specialized fast algorithms for IQC feasibility and optimization problems," *Automatica*, vol. 40, no. 2, pp. 239 – 252, 2004.

[6] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819 – 830, 1997.

[7] H. Hindi, B. Hassibi, and S. Boyd, "Multiobjective H_2/H_∞-optimal controlvia finite dimensional Q-parametrization and linear matrix inequalities," *American Control Conference, 1998. Proceedings of the 1998*, vol. 5, 1998.

[8] J. Oishi and V. Balakrishnan, "Linear controller design for the NEC laser bonder via LMI optimization," *Advances in Linear Matrix Inequality Methods in Control, Advances in Control and Design. SIAM*, 1999.

[9] R. Bartels and G. Stewart, "Solution of the matrix equation $AX + XB = C$ [F4]," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[10] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization*, M. Jünger and D. Nadded, Eds. Heidelberg: Springer Verlag, 2001, pp. 112 – 156.

[11] J. Kelley, "The cutting plane method for solving convex programs," *Journal of the SIAM*, vol. 8, no. 4, pp. 703–712, 1960.

[12] E. Cheney and A. Goldstein, "Newton's method for convex programming and Tchebycheff approximation," *Numerische Mathematik*, vol. 1, no. 1, pp. 253–268, 1959.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] L. S. Lasdon, *Optimization Theory for Large Systems*, ser. MacMillan Series in Operations Research. MacMillan Publishing, 1970.

[15] A. M. Geoffrion, "Primal resource-directive approaches for optimizing nonlinear decomposable systems," *Operations Research*, vol. 18, no. 3, pp. 375–403, may 1970.

[16] M. Farhoodi and M. Beheshti, "A Case Study of the Multiobjective H_2/H_∞ Control via Finite Dimensional Youla Parameterization and LMI Optimization," *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pp. 493–497, 2007.

[17] J. Sturm, "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1, pp. 625–653, 1999.

[18] J. Löfberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, 2004.

[19] S. Boyd and C. Barratt, *Linear controller design: limits of performance*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[20] P. Khargonekar and M. Rotea, "Multiple objective optimal control of linear systems: the quadratic norm case," *Automatic Control, IEEE Transactions on*, vol. 36, no. 1, pp. 14–24, 1991.

[21] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, 2000.

Appendix
Bounding x_i

Each of the subproblems in (12) are on the form

$$\begin{aligned} \min_{x,P} \ & \langle C, P\rangle + c^T x \\ \text{s.t. } & \mathcal{F}(P) + M_0 + \mathcal{G}(x) \preceq 0 \end{aligned} \tag{26}$$

One way to make this optimization problem bounded is to add the constraint

$$\langle C, P\rangle + c^T x - f_{\min} \geq 0 \tag{27}$$

The dual of (26) with the extra constraint (27) is

$$\begin{aligned} \max_{\lambda,z} \ & \langle M_o, Z\rangle - f_{\min}\lambda \\ \text{s.t. } & \mathcal{F}^*(Z) + C(1+\lambda) = 0 \\ & \mathcal{G}^*(Z) + c(1+\lambda) = 0 \\ & Z \succeq 0 \\ & \lambda \leq 0. \end{aligned} \tag{28}$$

We can eliminate some of the dual variables Z using the constraint $\mathcal{F}^*(Z) + C(1+\lambda) = 0$ in a very similar way as was done in [3]. In [3], the number of variables in Z is reduced by finding a basis for the nullspace of $\mathcal{F}^*(Z) + C = 0$. The only difference is that we have the term $C(1+\lambda)$ instead of C. Hence the only difference compared to [3] is that we need to find a solution to the Lyapunov equation

$$AF_0 + F_0A^T + C(1+\lambda) = 0 \tag{29}$$

where λ is a variable. A solution to this is $F_0 = F_c(1+\lambda)$ where F_c solves the Lyapunov equation

$$AF_c + F_cA^T + C = 0. \tag{30}$$

To see this, just insert F_0 in (29) and we get

$$\begin{aligned} AF_0 + F_0A^T + C(1+\lambda) &= \\ AF_c(1+\lambda) + (1+\lambda)F_cA^T + C(1+\lambda) &= \\ \underbrace{AF_c + F_cA^T + C}_{=0} + \lambda\left(\underbrace{AF_c + F_cA^T + C}_{=0}\right) &= 0. \end{aligned} \tag{31}$$

Having this basis we can solve the dual problem using the reduced dual variables. The reconstruction of P and x is not affected by this and can be found in the exact same fashion as in [3].

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2009-10-19

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ ________________

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-2919

URL för elektronisk version

`http://www.control.isy.liu.se`

Titel
Title

A Decomposition Algorithm for KYP-SDPs

Författare
Author

Rikard Falkeborn, Anders Hansson

Sammanfattning
Abstract

In this paper, a structure exploiting algorithm for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma, where some of the constraints appear as complicating constraints is presented. A decomposition algorithm is proposed, where the structure of the problem can be utilized. In a numerical example, where a controller that minimizes the sum of the H_2-norm and the H_∞-norm is designed, the algorithm is shown to be faster than SeDuMi and the special purpose solver KYPD.

Nyckelord
Keywords

Optimization, Optimization Algorithms