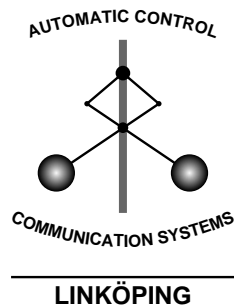


Utilizing low rank properties when solving KYP-SDPs

Janne Harju, 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: harju@isy.liu.se, ragnarw@isy.liu.se,
hansson@isy.liu.se

19th May 2006



Report no.: LiTH-ISY-R-2734

Submitted to IEEE Conference on Decision and Control, San
Diego , USA, 2006

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

Abstract

Semidefinite programs and especially those derived from the Kalman-Yakubovich-Popov lemma are quite common in control applications. KYPD is a dedicated solver for KYP-SDPs. It solves the optimization problem via the dual SDP. The solver is iterative. In each step a Hessian is formed and a linear system of equations is solved. The calculations can be performed much faster if we utilize sparsity and low rank structure. We show how to transform a dense optimization problem into a sparse one with low rank structure. A customized calculation of the Hessian is presented and investigated.

Keywords: Semidefinite programming, Kalman-Yakubovich-Popov lemma, low rank

Utilizing low rank properties when solving KYP-SDPs

Janne Harju, Ragnar Wallin and Anders Hansson

Abstract—Semidefinite programs and especially those derived from the Kalman-Yakubovich-Popov lemma are quite common in control applications. KYPD is a dedicated solver for KYP-SDPs. It solves the optimization problem via the dual SDP. The solver is iterative. In each step a Hessian is formed and a linear system of equations is solved. The calculations can be performed much faster if we utilize sparsity and low rank structure. We show how to transform a dense optimization problem into a sparse one with low rank structure. A customized calculation of the Hessian is presented and investigated.

I. INTRODUCTION

The need to solve semidefinite programs derived from the Kalman-Yakubovich-Popov lemma (KYP-SDPs) often arise in control and signal processing. In fact, some of the most important applications of semidefinite programming in control involve KYP-SDPs. Some examples are linear system design and analysis (Boyd and Barratt, 1991; Hindi *et al.*, 1998), robust control analysis using integral quadratic constraints (Rantzer, 1996; Megretski and Rantzer, 1997; Jönsson, 1996; Balakrishnan and Wang, 1999), quadratic Lyapunov function search (Boyd *et al.*, 1994) and filter design (Alkire and Vandenberghe, 2002).

As the size of the semidefinite programs (SDPs) in this problem class usually is very large it is often hard or even impossible to solve them using general-purpose software. The solver KYPD (Wallin and Hansson, 2004) is a dedicated solver for KYP-SDPs and utilizes the special structure in the problem. In this paper we will exploit sparseness and low rank properties in order to solve KYP-SDPs efficiently.

Other efficient solvers for KYP-SDPs have been developed. They are based on cutting plane methods (Parrilo, 2000; Kao *et al.*, 2001; Kao and Megretski, 2001; Hachez, 2003; Wallin *et al.*, 2005), interior-point methods with an alternative barrier (Kao *et al.*, 2001) and interior-point methods combined with conjugate gradients (Hansson and Vandenberghe, 2001; Wallin *et al.*, 2003; Gillberg and Hansson, 2003). Preliminary results on the approach used in KYPD has been presented in (Wallin *et al.*, 2003) and (Vandenberghe *et al.*, 2005). In (Vandenberghe *et al.*, 2005)

low rank properties are also exploited. However, the methodology used is only explained for systems with one input and a system matrix with real eigenvalues. To get such a matrix pole placement is used. The approach in this paper is more straightforward and possibly more numerically well-behaved.

A KYP-SDP in the variables $P \in \mathbb{S}^n$ and $x \in \mathbb{R}^p$ has the following structure

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

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

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

and

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

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{S}^n$ and $M_k \in \mathbb{S}^{n+m}$, $k = 0, 1, \dots, p$. In fact, there can be several constraints of the type above but for simplicity we only treat SDPs with one constraint in this paper. A generalization is straightforward. The number of variables n_{var} are $\frac{n(n+1)}{2} + p$.

Solving this optimization problem using an interior-point solver involves forming and solving a linear system of equations in each iteration. Let us assume that n is large compared to p and m which is most often the case. The number of variables are then of order n^2 . Hence, solving the system of equations can be done in order n^6 operations but forming the system has a computational complexity proportional to n^8 . Our approach has complexity of order n^3 .

II. ASSUMPTIONS

To make the presentation of the theory more streamlined we make two assumptions. The first assumption is that the pair (A, B) is controllable. This implies that the operator \mathcal{F} has full rank, (Vandenberghe *et al.*, 2005). The second assumption is that the operator $\mathcal{A}(P, x)$ defined as

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

has full rank, see (Wallin and Hansson, 2004) for details.

Neither of the assumptions are restrictive. The controllability assumption can be relaxed to stabilizability of the pair (A, B) , provided the range of C is in the controllable subspace

This work was supported by the Swedish Research Council under Grant No. 40469101 and by CENIIT.

J. Harju, A. Hansson and R. Wallin all are with the Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden {harju, ragnarw, hansson}@isy.liu.se

of (A, B) and 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 corresponding operator has full rank.

III. THE IDEA BEHIND KYPD

The solver KYPD is based on solving a problem equivalent to the SDP dual of (1). This equivalent dual has considerably fewer variables than the primal SDP and can be solved using any primal-dual solver.

The dual SDP

The dual of (1) is

$$\begin{aligned} \max \quad & -\langle M_0, Z \rangle \\ \text{s.t.} \quad & \mathcal{F}^*(Z) = C \end{aligned} \quad (2)$$

$$\mathcal{G}^*(Z) = c \quad (3)$$

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{bmatrix} \geq 0 \quad (4)$$

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. Hence, we should reduce the number of variables in order to solve the dual SDP efficiently.

Reduction of the number of dual variables

We want to find a parsimonious parametrization

$$Z = F_0 + \sum_{k=1}^{k_{\max}} z_k F_k \quad (5)$$

of all feasible Z . One such parametrization is given by

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

where $k_{\max} = mn + \frac{m(m+1)}{2}$, $E_{k(12)}$ is the standard basis for unstructured $n \times m$ matrices, $E_{k(22)}$ is 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 parsimonious parametrization we have to solve $mn + 1$ Lyapunov equations with respect to $E_{k(11)}$.

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 (3). However, this extra reduction will destroy any additional structure present in the basis matrices as, for example, sparsity or low rank. As the purpose of this paper is to exploit such properties we will not further reduce the number of variables. The interested reader can find a description of the procedure in (Wallin and Hansson, 2004).

The SDP equivalent to the dual is

$$\begin{aligned} \min \quad & d^T z \\ \text{s.t.} \quad & Gz = c \end{aligned} \quad (6)$$

$$Z = F_0 + \sum_{k=1}^{k_{\max}} z_k F_k \geq 0$$

where the entries of d are

$$d_k = \langle M_0, F_k \rangle$$

the entries of G are

$$G_{ik} = \langle M_i, F_k \rangle$$

$$\text{and } z = [z_1 \quad z_2 \quad \dots \quad z_{k_{\max}}]^T.$$

Reconstructing x and P

Primal-dual SDP solvers deliver the dual as well as the primal variable. The dual solution, X , of the above SDP is actually also a solution to (1), see (Vandenberghe *et al.*, 2005). Hence, we have 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 (6) and

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

When this overdetermined but consistent system of equations is solved we have x and can compute P , for example, by solving the Lyapunov function corresponding to the (1,1)-block of the constraint in (1).

IV. INTRODUCING ADDITIONAL STRUCTURE

By insisting on the system matrix to have a special structure, for example being diagonal or block diagonal the basis matrices F_k will be both sparse and have low rank. Both properties can be utilized to form the Hessian more efficiently. A diagonal A -matrix will in addition let us solve the Lyapunov equations in a computationally cheaper way.

Diagonalization of A

If the A -matrix is not diagonalizable we can always, as the pair (A, B) is controllable, apply a congruence transformation to the KYP-LMI to make $\tilde{A} = A - BL$ diagonalizable. The operator $\mathcal{F}(P)$ will be transformed as

$$\begin{bmatrix} \tilde{A}^T P + P \tilde{A} & PB \\ B^T P & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix}^T \mathcal{F}(P) \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix} \quad (7)$$

The matrix M_0 and the operator $\mathcal{G}(x)$ will be transformed analogously. Then apply another congruence transformation

$$\begin{bmatrix} \tilde{A}^H \tilde{P} + \tilde{P} \tilde{A} & \tilde{P} \tilde{B} \\ \tilde{B}^H \tilde{P} & 0 \end{bmatrix} = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}^H \begin{bmatrix} \tilde{A}^T P + P \tilde{A} & PB \\ B^T P & 0 \end{bmatrix} \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}$$

to make $\tilde{A} = T^{-1} \tilde{A} T$ diagonal. We also have that $\tilde{P} = T^H P T$ and $\tilde{B} = T^{-1} B$.

Two negative aspects with the diagonalization are that the basis matrices will be complex if A has complex valued eigenvalues and not every matrix can be diagonalized in a numerically well conditioned way. There are two remedies to the first dilemma. Either we can solve a real SDP involving LMIs with twice as many rows and columns as the original one (Boyd and Vandenberghe, 2004) or we can transform the complex diagonal A -matrix into a real block diagonal one of the same size. We prefer the second alternative. The numerical issues with diagonalizing the A -matrix may not be as severe as it seems though. As is mentioned above we can always do a congruence transformation to change the system matrix to $\tilde{A} = A - BL$. We thus have a freedom to choose L to get a matrix that has good numerical properties when it comes to diagonalization. How to choose L has however to be investigated.

Block diagonal A -matrix

Let us first assume that the eigenvalues are ordered on the diagonal. First we have all real eigenvalues and then the complex ones follow in complex conjugated pairs. To transform the A -matrix from being complex diagonal to being real block diagonal we only have to do a congruence transformation $\tilde{A} = V^H A V$. The matrix V has ones on the diagonal for all rows with real eigenvalues and blocks

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix}$$

on the diagonal for rows with complex conjugated eigenvalues. If we have a complex conjugated block in the A -matrix it will be transformed as

$$S^H A_k S = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} \frac{a+ib}{2} & 0 \\ 0 & \frac{a-ib}{2} \end{bmatrix} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

The congruence transformation will also result in a real B -matrix. The Lyapunov matrices we have to solve to get the basis matrices can be solved in order n^2 operations when the A -matrix is diagonal. Hence, the total cost for forming the basis is of order n^3 . If the matrix $E_{k(12)}$ has a one in a row corresponding to a block of dimension one, i.e. a real eigenvalue, the resulting basis matrix will be of at most rank two and can be written as

$$F_k = u_1 e_j^T + e_j u_1^T = v_1 v_1^T + v_2 v_2^T \quad (8)$$

where e_j is the j th unit vector. Thus, in addition to having low rank the basis matrix is also sparse, having only one row and one column with nonzero elements. If the matrix $E_{k(12)}$ has a one in a row corresponding to a block of dimension two the resulting basis matrix will be of at most rank four and can be written as

$$\begin{aligned} F_k &= u_1 e_j^T + e_j u_1^T + u_2 e_{j+1}^T + e_{j+1} u_2^T \\ &= v_1 v_1^T + v_2 v_2^T + v_3 v_3^T + v_4 v_4^T \end{aligned} \quad (9)$$

Also in this case the basis matrices are sparse.

V. PRIMAL-DUAL SOLVERS

A general-purpose primal-dual solver applied to (6) generates iterates of $z \in \mathbb{R}^{k_{\max}}$, $\lambda \in \mathbb{R}^p$ and the positive semidefinite matrix $X \in \mathbb{S}^{m+n}$. The vector λ and the matrix X are variables in the Dual to (6). At each iteration a linear system of

equations

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

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

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

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

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

In general the cost for solving this system of equations is proportional to $(k_{\max} + p)^3$ and the cost for forming H is proportional to k_{\max}^4 . If the number of variables in Z are not reduced $k_{\max} = \frac{(m+n)(m+n+1)}{2}$ and after the reduction we have $k_{\max} = mn + \frac{m(m+1)}{2} + p$ variables. This yields a considerable reduction in computational complexity. However, when the F_k -matrices are low rank we can do even better. The cost for forming H will only be cubic in k_{\max} .

Utilizing low rank of the basis matrices

To utilize low rank the basis matrices are written as a sum of rank one matrices. Below two separate forms are presented.

$$F_k = \sum_{i=1}^{2r_k} v_{ik} v_{ik}^T = \sum_{i=1}^{r_k} e_{ik} u_{ik}^T + u_{ik} e_{ik}^T \quad (11)$$

Rewriting the expression for H with the low rank expression for F_k and using properties for the inner product gives

$$\begin{aligned} H_{ij} &= \sum_{k=1}^{2r_i} \sum_{l=1}^{2r_j} v_{ik}^T W^{-1} v_{jl} v_{jl}^T W^{-1} v_{ik} \\ &= 2 \sum_{k=1}^{r_i} \sum_{l=1}^{r_j} u_{ik}^T W^{-1} e_{jl} u_{jl}^T W^{-1} e_{ik} + u_{ik}^T W^{-1} u_{jl} e_{jl}^T W^{-1} e_{ik}, \\ i, j &= 1, 2, \dots, k_{\max} \end{aligned}$$

Note that preprocessing can be done by calculating $v^T W^{-1}$ once. Exploiting sparsity to form H is implemented in SDPT3. However, if p is small compared to n , the worst-case

cost to form H is proportional to $m^5 n^4$. This is independent of utilizing sparsity or not. Tests imply though that the sparsity utilizing algorithm is much faster in practice and this is used when a system is block-diagonalized. Calculating H_{ij} using low rank matrices will reduce the cost to $m^2 n^3$, see (Toh *et al.*, 1999) for details.

VI. NUMERICAL EXAMPLES

To evaluate the algorithm we compare the computational times for some numerical examples. The examples are run on a Sun Sunfire V20z computer with 2Gb RAM running Linux under CentOS 4.1. They are solved using KYPD using SDPT3 version 3.1 as the underlying solver. SDPT3 is interfaced using YALMIP version 3 (R20050720). Matlab version 7.0.1 (R14) is used.

The options for YALMIP defined by `sdpsettings.m` were given an extra option to enable block-diagonalization or block-diagonalization combined with low rank calculations of the Hessian. In SDPT3 the possibility to use function-handles for a Hessian calculation was added in `NTpred.m`. To utilize sparsity the setting `sdpdensity` is used in SDPT3. The diagonalization is activated via YALMIP. To utilize low rank structure the file that performs the calculation of the Hessian is provided to SDPT3 as a function-handle. Lyapunov equations in the first example are solved through a diagonalization of the A -matrix.

In order to improve numerical issues feedback is performed in the Seismic isolation example. SDPT3 terminates when the primal-dual gap is less than 10^{-7} .

In the examples KYP-SDPs are solved using four different settings. SDPT3 denotes that the primal problem is solved using SDPT3 interfaced via YALMIP. KYPD denotes that the equivalent dual is solved using KYPD with SDPT3 as an underlying solver. Sparsity denotes that the dual is solved using KYPD after a transformation is done. This transformation block-diagonalizes the system matrix A . Lowrank is similar to Sparsity but `sdpdensity` is set to 1 and in every step the interior-point method forms the Hessian using a special low rank algorithm.

The solution time is obtained by using the matlab command `cputime` before and after a call to the solver. Each solver has obtained the problem data on the primal form to make a comparison fair. Preprocessing such as transformations and any rewritings of the problem are included in the solution time.

Randomly generated KYPs

This numerical example is based on randomly generated KYPs. The problem to be solved is 1 where the matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^n$ are generated with the Matlab command `rand`. In order to get comparable results infeasible problems and problems where the condition number of the controllability matrix exceeds 10^6 are rejected. The components of

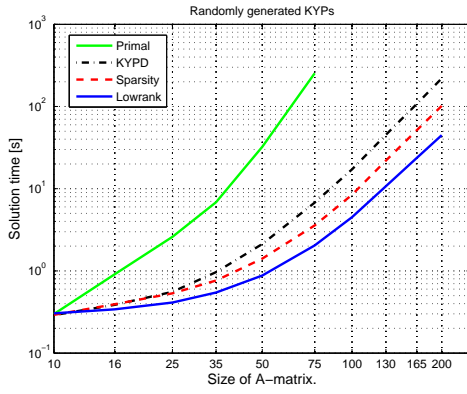


Fig. 1. Solution times for the randomly generated KYPs example vs system order. For every system order n ten generated problems are solved. The average solution time is presented. Used solvers are primal problem using SDPT3, solving the dual with KYPD, solving the problem using a block-diagonalization and finally solving the problem using a block-diagonalization and low rank calculations of the Hessian.

c is drawn from a uniformly distributed (.2 – 1.2) random variable. The matrices $M \in \mathbb{S}^{n \times n}$ are linearly independent and also generated by `rand`. For every system size n , ten generated problems are solved and the mean time is calculated. In Figure 1 it can be seen that we reach the theoretical total cost of order n^3 for the block-diagonalization combined with a low rank calculation of the Hessian. The initial cost for these calculations is due to the preprocessing done when calculating the low rank matrices in Equation 11. An implementation in C of these calculations would lower the preprocessing time significantly. Doing the block-diagonalization and utilizing sparsity when forming the Hessian also has complexity n^3 .

Seismic isolation control

This example deals with seismic isolation control of a n story building and is taken from (Kao, 2002). The building is modeled as a series connection of masses, springs and dampers as is illustrated in Figure 2. The equations describing the dynamics of the system are

$$\begin{aligned} m_1 \ddot{x}_1 + c_1 \dot{x}_1 + k_1 x_1 - c_2 (\dot{x}_2 - \dot{x}_1) - k_2 (x_2 - x_1) &= -u + v \\ m_r \ddot{x}_r + c_r \dot{x}_r + k_r x_r - c_{r+1} (\dot{x}_{r+1} - \dot{x}_r) - k_{r+1} (x_{r+1} - x_r) &= 0, \\ (\text{for } r = 2, 3, \dots, n-1) \\ m_n \ddot{x}_n + c_n (\dot{x}_n - \dot{x}_{n-1}) + k_n (x_n - x_{n-1}) &= 0 \end{aligned}$$

where u is the control force applied between the ground and the first floor of the building, and v is the earthquake's force applied to the ground. The spectrum of v lies in the frequency span 1/3 to 3 Hz. Seismic isolation controllers are designed for buildings of 6, 8 and 10 stories. An accelerometer is available at each floor of the building. The values of m_r , c_r and k_r in the examples are given in Table I. The controller is based on H_2 design and is carried out using the Matlab

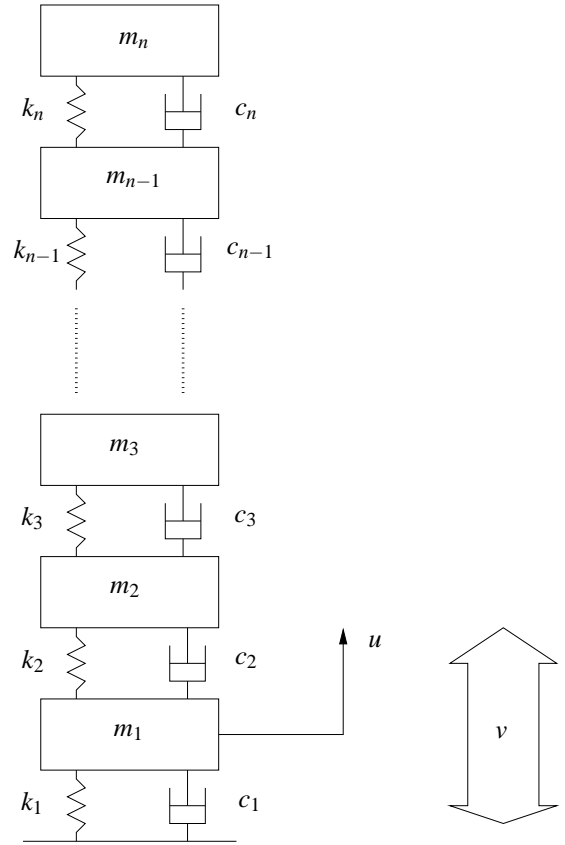


Fig. 2. Each story is modeled as a mass, a spring and a damper. The stories are then connected in series. The force u is the control force and the force v is moving the ground and is due to the earthquake

r	m_r	c_r	k_r
1	44.2	18.3	91.6
2	44.2	18.3	91.6
3	44.2	17.7	88.3
4	44.2	17.8	89.2
5	44.2	15.8	79.1
6	44.2	14.6	73.1
7	44.2	13.2	66.1
8	44.2	11.6	58.0
9	44.2	9.8	48.8
10	44.2	7.6	38.1

TABLE I

THE VALUES OF MASSES, SPRING AND DAMPER CONSTANTS FOR THE DIFFERENT STORIES

the μ -Analysis and Synthesis Toolbox (Balas *et al.*, 1993). We assume that the constants c_r and k_r have a 10 % uncertainty for $r = 1$ to 5. We analyze the system robustness by computing an upper bound on the induced \mathcal{L}_2 -gain from v to the acceleration vector \ddot{x} , as described in (Kao, 2002). If the number of stories are s the number of KYP-LMIs are 11, out of which the first ten are due to the uncertainties in the damping coefficients and spring constants and are of size 2×2 and the last one is of size $(4s + 34) \times (4s + 34)$.

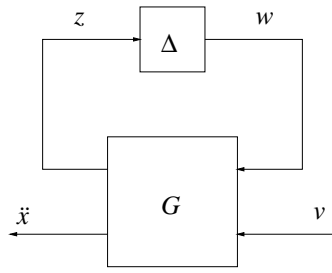


Fig. 3. Block diagram for the robustness analysis of the seismic isolation control system.

The decision variables are $x \in \mathbb{R}^{41}$, $P_i \in \mathbb{R}^1$, $i=1, 2, \dots, 10$, $P_{11} \in \mathbb{S}^{4s+23}$. We compute an upper bound on the induced \mathcal{L}_2 -gain with three correct digits. The computational results are shown in Table II. The first column shows the number of storys and then the solution times in seconds are shown. It can be seen that solving the equivalent dual as is done in KYPD gives a major improvement. To block diagonalize the A-matrix and utilize sparsity further improves the efficiency. The numerical issues are severe in this example and therefore the low rank utilization is not reliable.

# storys	SDPT3 [s]	KYPD [s]	Sparsity [s]
6	1438.9	113.2	71.3
8	4569.0	203.0	159.2
10	9831.6	385.5	187.1

TABLE II

COMPUTATIONAL RESULTS FOR THE SEISMIC CONTROL PROBLEM. FIRST COLUMN SHOWS THE NUMBER OF STORYS AND THEN THE SOLUTION TIMES IN SECONDS ARE SHOWN. TIME INDICATES A SOLUTION TIME LARGER THAN 10^4 .

VII. ACKNOWLEDGMENTS

The authors would like to thank Johan Löfberg, for a helping hand in YALMIP, K.C Toh, who has been helpful with SDPT3, Gustaf Hendeby and Henrik Tidefeldt for writing C-code in a previous project and David Lindgren for constructing numerical examples.

REFERENCES

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

- Balakrishnan, V. and F. Wang (1999). Efficient computation of a guaranteed lower bound on the robust stability margin for a class of uncertain systems. *IEEE Transactions on Automatic Control* **44**(11), 2185–2190.
- Balas, G., J. C. Doyle, K. Glover, A. Packard and R. Smith (1993). *μ -Analysis and Synthesis Toolbox*. The MathWorks Inc.
- Boyd, S. and C. Barratt (1991). *Linear controller design: Limits of performance*. Prentice Hall.
- Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. New York, New York, USA.
- Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan (1994). *Linear matrix inequalities in system and control theory*. SIAM. Philadelphia, USA.
- Gillberg, J. and A. Hansson (2003). 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.
- Hachez, Y. (2003). Convex optimization over nonnegative polynomials: structured algorithms and applications. PhD thesis. Université catholique de Lovain. Louvain, Belgium.
- Hansson, A. and L. Vandenberghe (2001). A primal-dual potential reduction method for integral quadratic constraints. In: *Proceedings of the American Control Conference*. Arlington, Virginia, USA. pp. 3013–3017.
- Hindi, H., B. Hassibi and S. Boyd (1998). Multiobjective H_2/H_∞ -optimal control via finite-dimensional Q-parameterization and linear matrix inequalities. In: *Proceedings of the American Control Conference*. Vol. 5. Philadelphia, Pennsylvania, USA. pp. 3244–3249.
- Jönsson, U. (1996). Robustness analysis of uncertain and nonlinear systems. PhD thesis. Lund Institute of Technology. Lund, Sweden.
- Kao, C.Y. (2002). Efficient computational methods for robustness analysis. PhD thesis. Massachusetts Institute of Technology.
- Kao, C.Y., A. Megretski and U.T. Jönsson (2001). A cutting plane algorithm for robustness analysis of time-varying systems. *IEEE Transactions on Automatic Control* **46**(4), 579–592.
- Kao, C.Y. and A. Megretski (2001). Fast algorithms for solving IQC feasibility and optimization problems. In: *Proceedings of the American Control Conference*. Vol. 4. Arlington, Virginia, USA. pp. 3019–3024.
- Megretski, A. and A. Rantzer (1997). System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control* **42**(6), 819–830.
- Parrilo, P. (2000). Structured semidefinite programs and semialgebraic geometry imethods in robustness and optimization. PhD thesis. California Institute of Technology. Pasadena, California, USA.
- Rantzer, A. (1996). On the Kalman-Yakubovich-Popov lemma. *Systems & Control Letters* **28**(1), 7–10.
- Toh, K. C., M. J. Todd and R. H. Tütüncü (1999). SDPT3—a Matlab software package for semidefinite programming. *Optimization Methods and Software* **11**, 545–581.
- Vandenberghe, L., V.R. Balakrishnan, R. Wallin and A. Hansson (2005). *Positive Polynomials in Control*. Chap. Interior-point algorithms for semidefinite programming problems derived from the KYP lemma. Lecture Notes on Control and Information Sciences. Springer Verlag.
- Wallin, R. and A. Hansson (2004). KYPD: A solver for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma. In: *IEEE Symposium on Computer Aided Control Design*. Taipei, Taiwan.
- Wallin, R., C. Y. Kao and A. Hansson (2005). A decomposition approach for solving KYP-SDPs. In: *Proceedings of the 16th IFAC World Congress*. Prague, Czech Republic.
- Wallin, R., H. Hansson and L. Vandenberghe (2003). Comparison of two structure exploiting algorithms for integral quadratic constraints. In: *4th IFAC Symposium on Robust Control Design*. Milan, Italy.