User’s guide to kypd_solver

Ragnar Wallin, Janne Harju

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, harju@isy.liu.se

19th May 2006

Report no.: LiTH-ISY-R-2733

Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications
Abstract
This package contains software for solving semidefinite programs (SDPs) originating from the Kalman-Yakubovich-Popov lemma. A presentation of the software is given and the options included are presented and described.

Keywords: Semidefinite programming, Kalman-Yakubovich-Popov lemma,
User manual for \texttt{kypd\_solver}

Ragnar Wallin, Janne Harju

May 19, 2006

1 Introduction

This package contains software for solving semidefinite programs (SDPs) originating from the Kalman-Yakubovich-Popov lemma. Those SDPs have the following structure

\[
\begin{align*}
\text{minimize} & \quad c^T x + \sum_{i=1}^{N} \langle C_i, P_i \rangle \\
\text{subject to} & \quad X_i = \begin{bmatrix} P_i A_i + A_i^T P_i & P_i B_i \\
B_i^T P_i & 0 \end{bmatrix} + M_{i0} + \sum_{j=1}^{K} x_j M_{ij} \geq 0 \\
i & = 1, 2, \ldots, N
\end{align*}
\]

The problem data are the vector $c \in \mathbb{R}^K$, the matrices $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, $C_i \in \mathbb{S}^{n_i \times m_i}$ and the matrices $M_{ij} \in \mathbb{S}^{(n_i+m_i) \times (n_i+m_i)}$. The inequality $X_i \geq 0$ means that $X_i$ is positive semidefinite. To make the code efficient the dual problem is solved using SeDuMi [3] or SDPT3 [5], after a reduction of the number of variables. The SDP is formulated using YALMIP [4]. The primal variables $P$ and $x$ are reconstructed afterwards. For more information, see [1] and [2].

2 Matlab routines

The main routines are \texttt{kypd.m} and \texttt{kypd\_solver.m}. To solve a given SDP you can proceed as follows:

1. if you know that the $M_{ij}$ matrices are linearly independent and the pairs $(A_i, B_i)$ are controllable use \texttt{kypd.m}.

2. otherwise use \texttt{kypd\_solver.m}.
kypd.m

[u,P,x,Z]=kypd(matrix_info,options)

**Purpose**

The function kypd solves the problem

\[
\begin{align*}
\text{minimize} & \quad c^T x + \sum_{i=1}^{N} \langle C_i, P_i \rangle \\
\text{subject to} & \quad X_i = \begin{bmatrix} P_i A_i + A_i^T P_i & P_i B_i \\ B_i^T P_i & 0 \end{bmatrix} + M_{i0} + \sum_{j=1}^{K} x_j M_{ij} \geq 0 \\
& \quad i = 1, 2, \ldots, N
\end{align*}
\]

and its dual

\[
\begin{align*}
\text{maximize} & \quad -\sum_{i=1}^{N} \text{Tr} M_{i0} Z_i \\
\text{subject to} & \quad A_i Z_{i1}^{11} + Z_{i1}^{11 T} A_i^T + B_i Z_{i2}^{12 T} + Z_{i2}^{12 T} B_i^T = C_i \\
& \quad \sum_{i=1}^{N} \text{Tr} M_{ij} Z_i = c_j \quad j = 1, 2, \ldots, K \\
& \quad Z_i \geq 0
\end{align*}
\]

where \( Z_i \) is partitioned as

\[
\begin{bmatrix}
Z_{i1}^{11} & Z_{i1}^{12} \\
Z_{i2}^{12 T} & Z_{i2}^{22}
\end{bmatrix}
\]

**Input arguments**

All matrix input information is stored in a cell structure called `matrix_info`. All fields are described below along with the other input arguments.

1. **matrix_info.N**: Number of constraints.
2. **matrix_info.K**: Number of elements in the vector \( x \) of the primal objective function.
3. **matrix_info.c**: The vector \( c \) in the primal objective.
4. **matrix_info.C{[i]}**: The \( C_i \) matrix in the primal objective.
• matrix_info.A{i}: The A matrix appearing in constraint i of the primal problem.

• matrix_info.B{i}: The B matrix appearing in constraint i of the primal problem.

• matrix_info.M0{i}: The M_0 matrix appearing in constraint i of the primal problem.

• matrix_info.M{i,j}: The jth M matrix appearing in constraint i of the primal problem.

• matrix_info.n: A vector containing the sizes of the A matrices, i.e. \( n = [n_1, n_2, \ldots, n_N]^T \).

• matrix_info.nm: A vector containing the sizes of the M matrices, i.e. \( nm = [nm_1, nm_2, \ldots, nm_N]^T \). \( nm_i = n_i + m_i \).

2. options: The options are in YALMIPs format. To set, for example, tol to \( 10^{-6} \) you write \texttt{sdpsettings('kypd.tol',1e-6)}. Possible options are:

(a) tol The solution will not violate feasibility and optimality conditions with more than tol. See \texttt{par.eps} in the SeDuMi user’s manual [3] for details. Default value \( 10^{-8} \).

(b) maxiters: Maximum number of iterations. maxiters\( \geq 0 \). Default value 100.

(c) lyapunovsolver When solving Lyapunov equations two solvers can be chosen. The first one diagonalizes the A-matrices and is more efficient. It may not be numerically safe to use it in all cases though. This solver is chosen by setting \texttt{lyapunovsolver='diago'}. The other choice, which is the default choice, is \texttt{'schur'}.

(d) reduce If a minimal order dual is preferred the second equality constraint in the dual can be eliminated [?]. This may destroy some structure, like sparsity, and is not always recommended. To eliminate the constraint set \texttt{reduce=1}. Otherwise set it to zero. The default value is zero.
Output arguments

1. **u**: The primal objective $c^T x$ is stored in $u(1)$. The dual objective $- \sum_{i=1}^{N} \text{Tr} M_{i0} Z_i$ is stored in $u(2)$.

2. **P**: The last iterate of the primal variable $P$.

3. **x**: The last iterate of the primal variable $x$.

4. **Z**: The last dual iterate $Z$.

Caveats

- The matrices $M_{ij}$ must be linearly independent and the pairs $(A_i, B_i)$ must be controllable. If this is not the case the problem is not dual feasible or can be reduced to a problem with fewer variables. Additionally, all $A_i$ matrices must be Hurwitz.

**kypd_solver.m**

```
[u,P,x,Z]=kypd_solver(matrix_info,options)
```

**Purpose**

To solve the KYP based SDP using a primal-dual method the $M_{ij}$ matrices have to be linearly independent, the pairs $(A_i, B_i)$ have to be controllable and the $A_i$s Hurwitz. If the matrices $M_{ij}$ are linearly dependent the problem is not strictly dual feasible and can be reduced to a problem with fewer variables which is strictly dual feasible or the primal is unbounded from below. If the system is stabilizable but not controllable there does not exist a strictly feasible dual point. Also in this case the problem can be reduced to a problem with fewer variables which is controllable and for which a strictly feasible dual point exists. If the system is not Hurwitz state feedback can be applied to make it so. Sometimes we want to apply feedback to improve numerical properties even if it is not necessary for any other reason. **kypd_solver** will handle this and produce a correct solution to the original problem, that is the problem before reduction and feedback.
Feedback

Linear quadratic feedback is applied if the system is not Hurwitz or if \texttt{transform=1}. The criterion we minimize is

$$J = \int_0^\infty (x_i^T x_i + \rho u_i^T u_i) dt$$

where $\dot{x}_i = A_i x_i + B_i u_i$. The optimal solution is given by $u_i = -L_i x_i$ where $L_i = \frac{1}{\rho} B_i^T S_i$ and $S_i$ is the solution to the algebraic Riccati equation

$$S_i A_i + A_i^T S_i - \rho S_i B_i B_i^T S_i + I = 0$$

The resulting closed loop system is given by

$$\dot{x}_i = (A_i - B_i L_i)x_i$$

and the modified system matrix $A_i - B_i L_i$ is Hurwitz.

Input arguments

1. \texttt{matrix\_info}: Information about the matrices in the problem. See the input arguments for \texttt{kypd} for details.

2. \texttt{options} In addition to the options for the function \texttt{kypd} a few more can be set for \texttt{kypd\_solver}. These are:

   (a) \texttt{transform}: If \texttt{transform=0} state feedback is only done when the system is not Hurwitz. If \texttt{transform=1} state feedback is done even if it is not necessary. It may improve numerical properties. Default value is 0.

   (b) \texttt{rho}: A nonnegative parameter that is used for the state feedback. Default value is 1.

   (c) \texttt{lowrank} Make a block-diagonalization of the system matrices and/or utilize low-rank structure when forming the Hessian. To make a block-diagonalization set \texttt{lowrank=1}, to make a block-diagonalization and utilize lowrank structure when forming the hessian set \texttt{lowrank=2}. Note that this requires the use of SDPT3. Otherwise set it to zero. Default is zero.

If a default value of an input parameter is wanted it can be put to the empty matrix \([ \ ]\).
Output arguments

1. **u**: The primal objective value \( c^T x \).
2. **P**: The last primal iterate \( P \).
3. **x**: The last primal iterate \( x \).
4. **Z**: The last dual iterate \( Z \).

3 Installation

Install SeDuMi [3], SDPT3 [6] and YALMIP [4] according to the manuals. Unpack the matlab files and set the matlab paths right. If the lowrank option is to be used, the C-files has to be compiled. See `help mex` in Matlab for details.

References


