Linköping University Post Print

An inexact interior-point method for system analysis

Janne Harju Johansson and Anders Hansson

N.B.: When citing this work, cite the original article.

This is an electronic version of an article published in:

Janne Harju Johansson and Anders Hansson, An inexact interior-point method for system analysis, 2010, INTERNATIONAL JOURNAL OF CONTROL, (83), 3, 601-616.

INTERNATIONAL JOURNAL OF CONTROL is available online at informaworldTM:

<http://dx.doi.org/10.1080/00207170903334813>

Copyright: Taylor and Francis

<http://www.tandf.co.uk/journals/default.asp>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54505>

RESEARCH ARTICLE

An Inexact Interior-Point Method for System Analysis

Janne Harju Johansson* and Anders Hansson

Automatic Control, Linköping University, SE-581 83 Linköping, Sweden

Email: {harju,hansson}@isy.liu.se

(Received 00 Month 200x; final version received 00 Month 200x)

In this paper a primal-dual interior-point algorithm for semidefinite programming that can be used for analyzing e.g. polytopic linear differential inclusions is tailored in order to be more computationally efficient. The key to the speedup is to allow for inexact search directions in the interior-point algorithm. These are obtained by aborting an iterative solver for computing the search directions prior to convergence. A convergence proof for the algorithm is given. Two different preconditioners for the iterative solver are proposed. The speedup is in many cases more than an order of magnitude. Moreover, the proposed algorithm can be used to analyze much larger problems as compared to what is possible with off-the-shelf interior-point solvers.

Keywords: Optimization; Linear Matrix Inequalities; Semidefinite programming; Interior-point methods; Iterative methods

1 Introduction and motivation

In this work a structured semidefinite programming (SDP) problem is investigated and a tailored algorithm is proposed and evaluated. The problem formulation, can for example be applied to analysis of polytopic linear differential inclusions (LDIs). The reformulation from systems analysis problems to SDPs is described in Boyd (1994) and Gahinet (1996). Another area of application is controller synthesis for parameter-dependent systems described in Wang and Balakrishnan (2002).

The software packages available to solve SDP problems are numerous. For example, if YALMIP, Löfberg (2004), is used as an interface, nine available solvers can be applied. Some examples of solvers are SDPT3, Toh (2006), DSDP, Benson and Yinyu (2005) and SeDuMi, Sturm (2001), Pólik (2005), all of which are interior point based solvers. These solvers solve the optimization problem on a general form. The problem size will increase with the number of constraints and the number of matrix variables. Hence, for large scale problems, generic solvers will not have an acceptable solution time or terminate within an acceptable number of function calls. It is necessary to utilize the problem structure to speed up the performance. Here an algorithm is described that uses inexact search directions for an infeasible interior-point method. A memory efficient iterative solver is used to compute the search directions in each step of the interior-point method. In each step of the algorithm, the error tolerance for the iterative solver decreases, and hence the initial steps are less expensive to calculate than the last ones.

Iterative solvers for linear systems of equations are well studied in the literature. For applications to optimization and preconditioning for interior-point methods see Chai and Toh (2007), Bonettini and Ruggiero (2007), Cafieri (2007), Bergamaschi (2004), Gillberg and Hansson (2003), Hansson and Vandenberghe (2001) and Vandenberghe and Boyd (1995). Here a SDP

problem is studied and hence the algorithms in Chai and Toh (2007), Bonettini and Ruggiero (2007), Cafieri (2007) and Bergamaschi (2004) are not applicable. In Gillberg and Hansson (2003), Hansson and Vandenberghe (2001) and Vandenberghe and Boyd (1995) a potential reduction method is considered and an iterative solver for the search directions is used. In Gillberg and Hansson (2003) a feasible interior-point method is used, and hence the inexact solutions to the search direction equations need to be projected onto the feasible set which is costly. In Hansson and Vandenberghe (2001) and Vandenberghe and Boyd (1995) this was circumvented by solving one linear system of equations for the primal search direction and another linear system of equations for the dual search direction, however also at a higher computational cost. Furthermore, solving the normal equations in Gillberg and Hansson (2003) resulted in an increasing number of iterations in the iterative solver when tending towards the optimum. In this paper the augmented equations are solved, which results in an indefinite linear system of equations. The number of outer iterations in the iterative solver does not increase close to the solution. The behavior of constant number of iterations in the iterative solver has also been observed in Hansson (2000) and Cafieri (2007). In Chai and Toh (2007) the same behavior was noted for linear programming. There the augmented equations are solved only when the iterate is close to the optimum.

The set of real-valued vectors with n rows is denoted $\mathbb{R}^n$. The space of symmetric matrices of size $n \times n$ is denoted $\mathbb{S}^n$. The space $\mathbb{S}^n$ has the inner product $\langle X, Y \rangle_{\mathbb{S}^n} = \text{Tr}(XY^T)$. We will with abuse of notation use the same notation $\langle \cdot, \cdot \rangle$ for inner products defined on different spaces when the inner product used is clear from context.

2 Problem Definition

Define the optimization problem to be solved as

$$\begin{aligned} \min \quad & c^T x + \langle C, P \rangle \\ \text{s.t.} \quad & \mathcal{F}_i(P) + \mathcal{G}_i(x) + M_{i,0} = S_i, \quad i = 1, \dots, n_i \\ & S_i \succeq 0 \end{aligned} \tag{1}$$

where the decision variables are $P \in \mathbb{S}^n$, $x \in \mathbb{R}^{n_x}$ and $S_i \in \mathbb{S}^{n+m}$. The variables S_i are slack-variables and only used to obtain equality constraints. Furthermore $c \in \mathbb{R}^{n_x}$ and $C \in \mathbb{S}^n$. Each constraint is composed of the operators

$$\mathcal{F}_i(P) = \begin{bmatrix} \mathcal{L}_i(P) & PB_i \\ B_i^T P & 0 \end{bmatrix} = \begin{bmatrix} A_i^T P + PA_i & PB_i \\ B_i^T P & 0 \end{bmatrix} \tag{2}$$

and

$$\mathcal{G}_i(x) = \sum_{k=1}^{n_x} x_k M_{i,k}, \tag{3}$$

with $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$ and $M_{i,k} \in \mathbb{S}^{n+m}$. The inner product $\langle C, P \rangle$ is $\text{Tr}(CP)$, and $\mathcal{L}_i : \mathbb{S}^n \rightarrow \mathbb{S}^n$ is the Lyapunov operator $\mathcal{L}_i(P) = A_i^T P + PA_i$ with adjoint $\mathcal{L}_i^*(X) = A_i X + X A_i^T$. Furthermore, the adjoint operators of $\mathcal{F}_i$ and $\mathcal{G}_i$ are

$$\mathcal{F}_i^*(Z_i) = \begin{bmatrix} A_i & B_i \end{bmatrix} Z_i \begin{bmatrix} I_n \\ 0 \end{bmatrix} + \begin{bmatrix} I_n & 0 \end{bmatrix} Z_i \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix} \tag{4}$$

and

$$\mathcal{G}_i^*(Z_i) = \begin{bmatrix} \langle M_{i,1}, Z_i \rangle \\ \vdots \\ \langle M_{i,n_x}, Z_i \rangle \end{bmatrix} \quad (5)$$

respectively, where the dual variable $Z_i \in \mathbb{S}^{n+m}$.

When we study (1) on a higher level of abstraction the operator

$$\mathcal{A}(P, x) = \oplus_{i=1}^{n_i} (\mathcal{F}_i(P) + \mathcal{G}_i(x)) \quad (6)$$

is used. Its adjoint is

$$\mathcal{A}^*(Z) = \sum_{i=1}^{n_i} \mathcal{F}_i^*(Z_i) \oplus \sum_{i=1}^{n_i} \mathcal{G}_i^*(Z_i), \quad (7)$$

where $Z = \oplus_{i=1}^{n_i} Z_i$. Also define $S = \oplus_{i=1}^{n_i} S_i$ and $M_0 = \oplus_{i=1}^{n_i} M_{i,0}$.

For notational convenience, define $z = (x, P, S, Z)$ and the corresponding finite-dimensional vector space $\mathcal{Z} = \mathbb{R}^{n_x} \times \mathbb{S}^n \times \mathbb{S}^{n_i(n+m)} \times \mathbb{S}^{n_i(n+m)}$ with its inner product $\langle \cdot, \cdot \rangle_{\mathcal{Z}}$. Furthermore, define the corresponding 2-norm $\| \cdot \|_2 : \mathcal{Z} \rightarrow \mathbb{R}$ by $\|z\|_2^2 = \langle z, z \rangle$. We notice that the 2-norm of a matrix with this definition will be the Frobenius norm and not the induced 2-norm.

Throughout the presentation it is assumed that the mapping $\mathcal{A}$ has full rank. Furthermore, the duality measure ν is defined as

$$\nu = \frac{\langle Z, S \rangle}{n_i(n+m)}. \quad (8)$$

3 Inexact interior-point method

In this work a primal-dual interior-point method is implemented. For such algorithms the primal and dual problems are solved simultaneously. The primal and dual for (1) with the higher level of notation are

$$\begin{aligned} \min \quad & c^T x + \langle C, P \rangle \\ \text{s.t.} \quad & \mathcal{A}(P, x) + M_0 = S \\ & S \succeq 0 \end{aligned} \quad (9)$$

and

$$\begin{aligned} \max \quad & -\langle M_0, Z \rangle \\ \text{s.t.} \quad & \mathcal{A}^*(Z) = C \oplus c \\ & Z \succeq 0 \end{aligned} \quad (10)$$

respectively. If strong duality holds, the Karush-Kuhn-Tucker conditions defines the solution to the primal and dual optimization problems, Boyd and Vandenberghe (2004). The Karush-Kuhn-

Tucker conditions for the optimization problems in (9) and (10) are

$$\mathcal{A}(P, x) + M_0 = S \quad (11)$$

$$\mathcal{A}^*(Z) = C \oplus c \quad (12)$$

$$ZS = 0 \quad (13)$$

$$S \succeq 0, Z \succeq 0 \quad (14)$$

Interior-point methods are iterative methods which compute iterates that approach solutions to (11)–(14). The equation in (13) is relaxed by replacing it with

$$ZS = \nu I. \quad (15)$$

The set of solutions defined by $\nu \geq 0$ is called the *central-path*. Interior-point methods proceed by computing search direction for the relaxed equations for smaller and smaller values of ν .

To derive the equations for the search directions in the next iterate, $z^+ = z + \Delta z$, is defined and inserted into (11)–(12) and (15). Then a linearization of these equations is made. In order to obtain a symmetric update of the matrix variables we introduce the symmetrization operator $\mathcal{H} : \mathbb{R}^{n \times n} \rightarrow \mathbb{S}^n$ that is defined as

$$\mathcal{H}(X) = \frac{1}{2}(R^{-1}XR + (R^{-1}XR)^T) \quad (16)$$

where $R \in \mathbb{R}^{n \times n}$ is a so called scaling matrix and (15) is replaced with

$$\mathcal{H}(ZS) = \nu I. \quad (17)$$

For a thorough description of scaling matrices, see Wolkowicz (2000) and Zhang (1998). The described procedure results in a linear system of equations for the search directions

$$\mathcal{A}(\Delta P, \Delta x) - \Delta S = -(\mathcal{A}(P, x) + M_0 - S) \quad (18)$$

$$\mathcal{A}^*(\Delta Z) = C \oplus c - \mathcal{A}^*(Z) \quad (19)$$

$$\mathcal{H}(\Delta ZS + Z\Delta S) = \sigma \nu I - \mathcal{H}(ZS) \quad (20)$$

Here the centering parameter σ has been introduced to enable the algorithm to reduce the duality measure or to take a step towards the central path. For values of σ close to zero, a step to decrease the duality measure is taken, and for values of σ close to 1 a step to find an iterate closer to the central path is taken. These type of methods are called *predictor-corrector* methods.

3.1 Computing the Search Direction

To summarize the discussion so far, the resulting linear system of equations that is to be solved to obtain the search direction in an infeasible interior-point method is in the form

$$\mathcal{A}(\Delta x) - \Delta S = D_1 \quad (21a)$$

$$\mathcal{A}^*(\Delta Z) = D_2 \quad (21b)$$

$$\mathcal{H}(\Delta ZS + Z\Delta S) = D_3, \quad (21c)$$

for some residuals D_1 , D_2 and D_3 . The residuals depend on the chosen method, e.g., infeasible or feasible method. Now an important lemma that makes the solution of the search directions well-defined is presented.

Lemma 3.1: *If the operator $\mathcal{A}$ has full rank, i.e. $\mathcal{A}(x) = 0$ implies that $x = 0$, and if $Z \succ 0$ and $S \succ 0$, then the linear system of equations in (18)–(20) has a unique solution.*

Proof See Theorem 10.2.2 in Wolkowicz (2000) or (Todd 1998, p. 777). $\square$

Solving (21a)–(21c) is a well-studied problem. The fact that it is a linear system of equations enable a vectorization of the equations to obtain a more insightful description. If the symmetric vectorization is applied to (21a)–(21c) the following linear system is obtained:

$$\begin{pmatrix} A - I_{n(n+1)/2} & 0 \\ 0 & 0 & A^T \\ 0 & H_{\Delta S} & H_{\Delta Z} \end{pmatrix} \begin{pmatrix} \text{vec}(\Delta x) \\ \text{svec}(\Delta S) \\ \text{svec}(\Delta Z) \end{pmatrix} = \begin{pmatrix} \text{svec}(D_1) \\ \text{svec}(D_2) \\ \text{svec}(D_3) \end{pmatrix}, \quad (22)$$

where $A \in \mathbb{R}^{n(n+1)/2 \times n_x}$ is the vectorization of the operator $\mathcal{A}$, i.e.,

$$A \cdot \text{vec}(\Delta x) = \text{svec}(\mathcal{A}(\Delta x)). \quad (23)$$

Since only real valued functions and variables are studied the vectorization of the adjoint function $\mathcal{A}$ corresponds to coefficient matrices for the transposed matrix A^T . Furthermore, $H_{\Delta S} = R^{-1}Z \otimes_s R$ and $H_{\Delta Z} = R^{-1} \otimes_s SR$ are the vectorizations of the symmetrization operator $\mathcal{H}$.

It is easily seen that a non-singular scaling matrix R make the function $\mathcal{H}$ invertible and hence are $H_{\Delta S}$ and $H_{\Delta Z}$ invertible. The system to be solved, (22), is a system of equations with $n_x + n(n+1)$ variables. The invertibility of the symmetrization operator implies the elimination of either the slack variable ΔS or the dual variable ΔZ . Eliminating the slack variable ΔS and reordering the vectorized variables results in

$$\begin{pmatrix} H_{\Delta S}^{-1} H_{\Delta Z} & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \text{svec}(\Delta Z) \\ \text{vec}(\Delta x) \end{pmatrix} = \begin{pmatrix} D_1 - H_{\Delta S}^{-1} D_3 \\ D_2 \end{pmatrix}. \quad (24)$$

The system of equations in (24) is referred to as the *augmented* equations. The matrix $H_{\Delta S}^{-1} H_{\Delta Z}$ is positive definite. This is an indefinite linear system of equations, i.e., it has a coefficient matrix with both positive and negative eigenvalues.

The question of how to solve the augmented equations in optimization algorithms has received an increased research interest. A linear system of equations in the form (24) is also known as *a saddle point* problem in the literature and is a well-analyzed problem. Presenting a complete reference list in this subject is impossible, though Benzi (2005) give a very thorough discussion of saddle-point problems. Furthermore, an extensive overview of solution strategies is presented.

Although, (24) seems to have many tractable properties as was pointed out in the introduction, it has some disadvantages. Solving an indefinite system of equations might not be preferred. The number of variables is quite large, and the symmetric coefficient matrix enable a reduction of variables. Eliminating the dual variable ΔZ in (24) result in

$$-A^T (H_{\Delta S}^{-1} H_{\Delta Z})^{-1} A \cdot \text{vec}(\Delta x) = D_2 - A^T (H_{\Delta S}^{-1} H_{\Delta Z})^{-1} (D_1 - H_{\Delta S}^{-1} D_3). \quad (25)$$

This system of equations is referred to as the *normal* equations. The coefficient matrix in the normal equations is a positive definite matrix. This can be seen in (25), since $H_{\Delta S}^{-1} H_{\Delta Z}$ is positive definite and A has full rank. However, the advantages with the augmented system of equations, as mentioned in the introduction, are more important than the disadvantages pointed out above, and hence we will in this work use the augmented equations.

4 Algorithm

Now the optimization algorithm will be presented. The algorithm is an inexact infeasible primal-dual predictor-corrector interior-point method. Using an infeasible method allow the use of inexact search directions since the solution does not need to be exact or projected into the space of solutions. In order to make the algorithm converge, it is based on a neighborhood which decreases as the algorithm proceeds. The solver for the search directions solve a linear system of equations in each iterate with a decreasing error bound. Hence, the algorithm increases the accuracy in the solution of the search direction when tending toward the optimum.

For later use and to obtain an easier notation define $\mathcal{K} : \mathcal{Z} \rightarrow \mathbb{S}^{n_i(n+m)} \times \mathbb{S}^n \times \mathbb{R}^{n_x} \times \mathbb{S}^{n_i(n+m)}$ as

$$\mathcal{K}(z) = \begin{bmatrix} \mathcal{K}_p(z) \\ \mathcal{K}_d(z) \\ \mathcal{K}_c(z) \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix}. \quad (26)$$

Now we define the set Ω that will be used in the algorithm to define a neighborhood for an iterate. Define the set Ω as

$$\Omega = \{z \mid S \succeq 0, Z \succeq 0, \|\mathcal{K}_p(z)\|_2 \leq \beta\nu, \|\mathcal{K}_d(z)\|_2 \leq \beta\nu, \gamma\nu I_{n_i(n+m)} \succeq \mathcal{K}_c(z) \succeq \eta\nu I_{n_i(n+m)}\}, \quad (27)$$

where the scalars β , γ and η will be defined later on. Finally define the set $\mathcal{S}$ for which the Karush-Kuhn-Tucker conditions are fulfilled.

$$\mathcal{S} = \{z \mid \mathcal{K}_p(z) = 0, \mathcal{K}_d(z) = 0, \mathcal{K}_c(z) = 0, S \succeq 0, Z \succeq 0\}. \quad (28)$$

This is the set the algorithm is to converge toward. For a convex problem, it is noted that $\mathcal{S}$ is a single point.

In Algorithm 1 the overall algorithm is summarized, which is taken from Ralph and Wright (1997), and adapted to semidefinite programming.

Algorithm 1 Inexact primal-dual method

- 1: Choose $0 < \eta < \eta_{max} < 1$, $\gamma \geq n_i(n+m)$, $\beta > 0$, $\kappa \in (0, 1)$, $0 < \sigma_{min} < \sigma_{max} \leq 1/2$, $\epsilon > 0$, $0 < \chi < 1$ and $z_{(0)} \in \Omega$.
 - 2: $j \leftarrow 0$
 - 3: **while** $\min(\epsilon_{rel}, \epsilon_{abs}) > \epsilon_{break}$ **do**
 - 4: **if** j even **then**
 - 5: $\sigma \leftarrow \sigma_{max}$
 - 6: **else**
 - 7: $\sigma \leftarrow \sigma_{min}$
 - 8: **end if**
 - 9: Compute the scaling matrix R
 - 10: Solve (18)–(20) for search direction $\Delta z_{(j)}$ with a residual tolerance $\epsilon\sigma\beta\nu/2$
 - 11: Choose a step length $\alpha_{(j)}$ as the first element in the sequence $\{1, \chi, \chi^2, \dots\}$ such that $z_{(j+1)} = z_{(j)} + \alpha_{(j)}\Delta z_{(j)} \in \Omega$ and such that $\nu_{(j+1)} \leq (1 - \alpha\kappa(1 - \sigma))\nu_{(j)}$.
 - 12: Update the variable, $z_{(j+1)} = z_{(j)} + \alpha_{(j)}\Delta z_{(j)}$
 - 13: $j \leftarrow j + 1$
 - 14: **end while**
-

4.1 The Nesterov-Todd scaling matrix

A special choice of scaling matrix results in the so-called Nesterov-Todd (NT) direction which was introduced in Nesterov and Todd (1997). To obtain the NT direction, we first define

$$W_{nt} = S^{-1/2}(S^{1/2}ZS^{1/2})^{1/2}S^{-1/2}. \quad (29)$$

Then the NT scaling matrix is given as the solution to $R_{nt}^T R_{nt} = W_{nt}$. This scaling matrix has been widely tested and analyzed in detail. It has shown good results in practical evaluations. In Algorithm 2 an efficient implementation is presented, the description is due to Todd (1998) and follows the presentation in Vandenberghe (2005).

Algorithm 2 Computing the NT scaling matrix

- 1: Compute the Cholesky factorizations:

$$S = L_1 L_1^T$$

$$Z = L_2 L_2^T$$

- 2: Compute the singular value decomposition (SVD):

$$L_2^T L_1 = U \lambda V^T$$

where λ is a positive definite and diagonal matrix and U and V are unitary matrices.

- 3: Compute the scaling matrix:

$$R_{nt} \leftarrow L_1 V \lambda^{-1/2}$$

From Algorithm 2, it can be derived that

$$R_{nt}^T S^{-1} R_{nt} = \lambda^{-1} \quad (30)$$

and

$$R_{nt}^T Z R_{nt} = \lambda. \quad (31)$$

5 Iterative Solver

It is the solution of (18)–(20), which is performed in Step 10 of the algorithm, that requires the most effort in an interior-point method. In order to study (18)–(20) in more detail rewrite them as

$$W_i \Delta Z_i W_i + \mathcal{F}_i(\Delta P) + \mathcal{G}_i(\Delta x) = D_1^i, \quad i = 1, \dots, n_i \quad (32a)$$

$$\sum_{i=1}^{n_i} \mathcal{F}_i^*(\Delta Z_i) = D_2 \quad (32b)$$

$$\sum_{i=1}^{n_i} \mathcal{G}_i^*(\Delta Z_i) = D_3, \quad (32c)$$

where $W_i = R_i R_i^T \in \mathbb{S}^n$ and $\oplus_{i=1}^{n_i} R_i = R_{nt}$. Note that the linear system of equations (32a)–(32c) is indefinite.

Here the symmetric quasi-minimal residual method (SQMR) is chosen as the solver to find the search directions. The SQMR method does not require inner products with the transposed coefficient matrix which makes it computationally cheap. In the SQMR method an indefinite preconditioner can be used. It should be stressed that most other iterative methods for solving indefinite linear systems of equations only allow for positive definite preconditioner, which are not good enough, and especially not when solving KKT systems, e.g. Hansson (2000). Another

positive property is that this method does not require as much storage as the theoretically optimal GMRES solver. Finally, the solver has better convergence properties than the BiCG method, (Greenbaum 1997, page 92). An undesirable property is that the residual is not included in the algorithm. Hence, it must be calculated since the norm of the residual is required by the interior-point solver.

In Freund and Nachtigal (1991) and Freund and Nachtigal (1994) the original SQMR algorithm description is presented. To simplify the description, we rewrite (32a)–(32c) as $\mathcal{B}(\Delta z) = b$ and denote the invertible preconditioner as $\mathcal{P}(\cdot)$. The algorithm described is SQMR without look-ahead for the linear system of equations using operator formalism. In Algorithm 1 the detailed description is presented.

Algorithm 1 SQMR on operator formalism

```

1: Choose  $\Delta z_{(0)} \in \mathcal{Z}$  and preconditioner  $\mathcal{P}(\cdot)$ 
2:  $r_0 \leftarrow b - \mathcal{B}(z_{(0)})$ 
3:  $t \leftarrow r_0$ 
4:  $\tau_0 \leftarrow \|t\|_2 = \sqrt{\langle r_0, r_0 \rangle}$ 
5:  $q_0 \leftarrow \mathcal{P}^{-1}(r_0)$ 
6:  $\vartheta_0 \leftarrow 0$ 
7:  $\rho_0 \leftarrow \langle r_0, q_0 \rangle$ 
8:  $d_0 \leftarrow 0$ 
9: for  $j = 1, 2, \dots$  do
10:    $t \leftarrow \mathcal{B}(q_{j-1})$ 
11:    $v_{j-1} \leftarrow \langle q_{j-1}, t \rangle$ 
12:   if  $v_{j-1} = 0$  then
13:     stop
14:   else
15:      $\alpha_{j-1} \leftarrow \frac{\rho_{j-1}}{v_{j-1}}$ 
16:      $r_j \leftarrow r_{j-1} - \alpha_{j-1}t$ 
17:   end if
18:    $t \leftarrow r_j$ 
19:    $\vartheta_j \leftarrow \|t\|_2 / \tau_{j-1}$ 
20:    $c_j \leftarrow 1 / \sqrt{1 + \vartheta_j^2}$ 
21:    $\tau_j \leftarrow \tau_{j-1} \vartheta_j c_j$ 
22:    $d_j \leftarrow c_j^2 \vartheta_{j-1}^2 d_{j-1} + c_j^2 \alpha_{j-1} q_{j-1}$ 
23:    $\Delta z_j \leftarrow \Delta z_{j-1} + d_j$ 
24:   if  $\Delta z_j$  has converged then
25:     stop
26:   end if
27:   if  $\rho_{j-1} = 0$  then
28:     stop
29:   else
30:      $u_j \leftarrow \mathcal{P}^{-1}(t)$ 
31:      $\rho_j \leftarrow \langle r_j, u_j \rangle$ 
32:      $\beta_j \leftarrow \frac{\rho_j}{\rho_{j-1}}$ 
33:      $q_j \leftarrow u_j + \beta_j q_{j-1}$ 
34:   end if
35: end for

```

Here $b, p, r, t, q, d \in \mathcal{Z}$ and $\tau, \vartheta, \rho, v, \alpha, c \in \mathbb{R}$.

6 Preconditioning

The construction of a good preconditioner is highly problem dependent. A preconditioner should reflect the main properties of the original system of equations and still be inexpensive to evaluate. There is a wide variety of preconditioners in the literature. In Benzi (2005) the general class of saddle point problems is studied and some preconditioners are discussed. In this section only the preconditioners applicable to an indefinite system of equations are discussed.

There are many strategies to approximate the linear system of equations to obtain a preconditioner. A popular choice is to approximate the symmetric and positive definite (1, 1)-block of the coefficient matrix with some less complicated structure. Common approximations are to use a diagonal matrix or a block-diagonal matrix. A special case of this strategy are the constraint preconditioners discussed in e.g. Cafieri (2007) and Rozloznik and Simoncini (2003). A collection of preconditioning strategies described previously can be found in Bonettini and Ruggiero (2007), Forsgren (2007), Dollar (2006), Bergamaschi (2004) and Keller (2000). The preconditioner used in the initial stage of Algorithm 1 is of this type.

Another strategy of preconditioning is to replace the coefficient matrix with a non-symmetric approximation that is easier to solve, as described in Benzi and Golub (2004) and Botchev and Golub (2006).

Finally, incomplete factorizations can be used. This is recommendable especially for sparse matrices, see Saad (1996) for further details.

In this work a two phase algorithm is described. The two separate phases are related to the change of properties of the linear system of equations when the iterates in the interior-point algorithm tend toward the optimum. The use of two separate preconditioners have previously been applied to linear programming problems in Chai and Toh (2007) and Bocanegra (2007).

6.1 Preconditioner I

To derive the preconditioner, assume that the constraints are closely related and in the sense that $A_i \approx A_j$, $B_i \approx B_j$, and $M_{i,k} \approx M_{j,k}$, $i \neq j$, and therefore an approximation assuming $A_i = \bar{A}$, $B_i = \bar{B}$ and $M_{i,k} = \bar{M}_k$ is applicable. Inserting the average system matrices $\bar{A}$, $\bar{B}$ and $\bar{M}_k$ into (32a)–(32c) results in the following equations

$$W_i \Delta Z_i W_i + \bar{\mathcal{F}}(\Delta P) + \bar{\mathcal{G}}(\Delta x) = D_1^i, \quad 1 = 1, \dots, n_i \quad (33a)$$

$$\bar{\mathcal{F}}^* \left(\sum_{i=1}^{n_i} \Delta Z_i \right) = D_2 \quad (33b)$$

$$\bar{\mathcal{G}}^* \left(\sum_{i=1}^{n_i} \Delta Z_i \right) = D_3. \quad (33c)$$

with obvious definition of $\bar{\mathcal{F}}$ and $\bar{\mathcal{G}}$. Approximate W_i with $w_i \cdot I_{n+m}$ and define $w_\Sigma = \sum_{i=1}^{n_i} 1/w_i^2$. Now rescale the equations and define the new variables $\Delta Z_{tot} = \sum_i \Delta Z_i$, $\Delta P_\Sigma = \Delta P \cdot w_\Sigma$ and $\Delta x_\Sigma = \Delta x \cdot w_\Sigma$. The simplified linear system of equations that is to be solved by the preconditioned iterative method is

$$\Delta Z_{tot} + \bar{\mathcal{F}}(\Delta P_\Sigma) + \bar{\mathcal{G}}(\Delta x_\Sigma) = \sum_{i=1}^{n_i} \frac{D_1^i}{w_i^2} \quad (34a)$$

$$\bar{\mathcal{F}}^*(\Delta Z_{tot}) = D_2 \quad (34b)$$

$$\bar{\mathcal{G}}^*(\Delta Z_{tot}) = D_3. \quad (34c)$$

Now define the partitioning

$$\Delta Z_{tot} = \begin{pmatrix} \Delta Z_{11} & \Delta Z_{12} \\ \Delta Z_{12}^T & \Delta Z_{22} \end{pmatrix} \quad (35)$$

where $\Delta Z_{11} \in \mathbb{S}^n$. To derive a method for solving (34a)–(34c) we use the following change of variables

$$\Delta \tilde{Z}_{11} = \Delta Z_{11} + \mathcal{L}^{-*}(\bar{B} \Delta Z_{12}^T + \Delta Z_{12} \bar{B}^T) \quad (36a)$$

$$\Delta \tilde{Z}_{12} = \Delta Z_{12} \quad (36b)$$

$$\Delta \tilde{Z}_{22} = \Delta Z_{22} \quad (36c)$$

$$\Delta \tilde{P} = \Delta P + \mathcal{L}^{-1} \left(\sum_{k=1}^{n_x} \Delta x_k \bar{M}_{k,11} \right) \quad (36d)$$

$$\Delta \tilde{x} = \Delta x, \quad (36e)$$

where $\bar{M}_{k,11}$ denotes the $(1,1)$ -block of the $\bar{M}_k$ matrices and $\mathcal{L}(X) = \bar{A}^T X + X \bar{A}$.

Now apply $\mathcal{L}^{-1}(\cdot)\bar{B}$ to the $(1,1)$ -block of (34a) and subtract it from the $(1,2)$ -block in (34a). Additionally apply $\langle \mathcal{L}^{-*}(\cdot), \bar{M}_{k,11} \rangle$ to (34b) and subtract it from the k :th element in (34c). Using the variable change defined in (36a)–(36e) results in the following linear system of equations

$$\Delta \tilde{Z}_{11} - \mathcal{L}^{-*}(\bar{B} \Delta \tilde{Z}_{12}^T + \Delta \tilde{Z}_{12} \bar{B}^T) + \mathcal{L}(\Delta \tilde{P}) = \sum_{i=1}^{n_i} \frac{D_{1,11}^i}{w_i^2} \quad (37a)$$

$$\begin{aligned} & \Delta \tilde{Z}_{12} + \mathcal{L}^{-1}(\mathcal{L}^{-*}(\bar{B} \Delta \tilde{Z}_{12}^T + \Delta \tilde{Z}_{12} \bar{B}^T)) \bar{B} \\ & - \mathcal{L}^{-1}(\Delta \tilde{Z}_{11}) \bar{B} + \sum_{k=1}^{n_x} \Delta x_k (\bar{M}_{k,12} - \mathcal{L}^{-1}(\bar{M}_{k,11}) \bar{B}) = \sum_{i=1}^{n_i} \frac{D_{1,12}^i}{w_i^2} - \mathcal{L}^{-1} \left(\sum_{i=1}^{n_i} \frac{D_{1,11}^i}{w_i^2} \right) \bar{B} \end{aligned} \quad (37b)$$

$$\Delta \tilde{Z}_{22} + \sum_{k=1}^{n_x} \Delta x_k \bar{M}_{k,22} = \sum_{i=1}^{n_i} \frac{D_{1,22}^i}{w_i^2} \quad (37c)$$

$$\mathcal{L}^*(\Delta \tilde{Z}_{11}) = D_2 \quad (37d)$$

$$\left\langle \begin{bmatrix} -\mathcal{L}^{-*}(\bar{B} \Delta \tilde{Z}_{12}^T + \Delta \tilde{Z}_{12} \bar{B}^T) & \Delta \tilde{Z}_{12} \\ \Delta \tilde{Z}_{12}^T & \Delta \tilde{Z}_{22} \end{bmatrix}, \begin{bmatrix} \bar{M}_k^{11} & \bar{M}_k^{12} \\ \bar{M}_k^{12^T} & \bar{M}_k^{22} \end{bmatrix} \right\rangle = D_{3,k} - \langle \mathcal{L}^{-*}(D_2), \bar{M}_{k,11} \rangle. \quad (37e)$$

The resulting linear system of equations can be solved in a five step procedure.

Algorithm 1 Preconditioner I

- 1: Solve (37d) for $\Delta \tilde{Z}_{11}$ using a Lyapunov solver.
 - 2: Vectorize and solve (37b), (37c) and (37e) to retrieve $\Delta \tilde{Z}_{12}$, $\Delta \tilde{Z}_{22}$ and Δx .
 - 3: Solve (37a) using a Lyapunov solver to obtain $\Delta \tilde{P}$.
 - 4: Compute the untransformed variables ΔZ_{tot} , ΔP and Δx using (36a)–(36e).
 - 5: Distribute the solution such that

$$\Delta Z_i = (D_1^i - \mathcal{F}_i(\Delta P) - \mathcal{G}_i(\Delta x))/w_i^2.$$
-

The vectorized linear system of equations in step 4 has $(nm+m^2) + (m(m+1)/2) + n_x$ variables and can thus be solved in $\mathcal{O}(n^3)$ flops (assuming $n \gg m$ and $n \gg n_x$).

Note that the coefficient matrix for the vectorized system of equations in Step 2 in the algorithm needs only to be constructed once. The main cost is the solution of Lyapunov equations

and the solution of the vectorized linear system of equations in step 4. This can be done at a total cost of $\mathcal{O}(n^3)$.

It is noted that the assumptions made to derive the preconditioner is not guaranteed to be fulfilled for a general problem. It is obvious that if these assumptions are violated the convergence speed will deteriorate for the iterative solver.

6.2 Preconditioner II

The inspiration to Preconditioner II is found in Gill (1992). In that work the analysis does not consider block structure in the coefficient matrix. Furthermore, the problem is reformulated to obtain a definite coefficient matrix since the chosen solver requires a definite preconditioner. Here we will construct an indefinite preconditioner by identifying the constraints that indicate large eigenvalues for the scaling matrices and look at the linear system of equations and the block structure introduced by the constraints.

First recall the definition of the symmetric vectorization operator

$$\mathbf{svec}(X) = (X_{11}, \sqrt{2}X_{12}, \dots, X_{22}, \sqrt{2}X_{23}, \dots, X_{nn})^T. \quad (38)$$

The $\mathbf{svec}$ operator yield a symmetric coefficient matrix when applied to (32a)–(32c). For notational convenience define

$$\mathbf{D}_{vec} = \begin{pmatrix} \mathbf{svec}(D_1^1) \\ \vdots \\ \mathbf{svec}(D_1^{n_i}) \\ \mathbf{svec}(D_2) \\ D_3 \end{pmatrix}$$

and

$$\Delta = \begin{pmatrix} \mathbf{svec}(\Delta Z_1) \\ \vdots \\ \mathbf{svec}(\Delta Z_{n_i}) \\ \mathbf{svec}(\Delta P) \\ \Delta x \end{pmatrix}.$$

To illustrate how Preconditioner II works, the vectorized version of (32a)–(32c) is studied. The linear system of equations for the search directions in a vectorized form can be written as

$$\begin{pmatrix} H_1 & & F_1 & G_1 \\ & \ddots & \vdots & \vdots \\ & & H_{n_i} & F_{n_i} & G_{n_i} \\ F_1^T & \dots & F_{n_i}^T & & \\ G_1^T & \dots & G_{n_i}^T & & \end{pmatrix} \Delta = \mathbf{D}_{vec}, \quad (39)$$

where H_i , F_i and G_i denote appropriate sub-matrices. To simplify the expressions in this section, define

$$N_i = (F_i \ G_i). \quad (40)$$

Simple matrix manipulations give the solution of (39) as

$$\begin{pmatrix} \text{svec}(\Delta P) \\ \Delta x \end{pmatrix} = \left(\sum_i N_i^T H_i^{-1} N_i \right)^{-1} \left(\sum_i N_i^T H_i^{-1} \text{svec}(D_1^i) - \begin{pmatrix} \text{svec}(D_2) \\ D_3 \end{pmatrix} \right) \quad (41)$$

and

$$\text{svec}(Z_i) = H_i^{-1} \left(\text{svec}(D_1^i) - N_i \begin{pmatrix} \text{svec}(\Delta P) \\ \Delta x \end{pmatrix} \right). \quad (42)$$

It has been observed in simulations that the eigenvalues of W_i grow large when the iterates tend towards the optimum for some of the constraints. This implies that the eigenvalues of H_i grow large and hence will $H_i^{-1} N_i \approx 0$ since F_i and G_i do not change during the iterations and have element-wise moderate values.

To derive Preconditioner II, assume that $H_i^{-1} N_i \approx 0$ is valid for all $i \neq s$. As an intermediate result, note that

$$\sum_i N_i^T H_i^{-1} N_i \approx N_s^T H_s^{-1} N_s. \quad (43)$$

Then the approximate solution is

$$\begin{pmatrix} \text{svec}(\Delta P) \\ \Delta x \end{pmatrix} \approx \left(N_s^T H_s^{-1} N_s \right)^{-1} \left(N_s^T H_s^{-1} \text{svec}(D_1^s) - \begin{pmatrix} \text{svec}(D_2) \\ D_3 \end{pmatrix} \right) \quad (44)$$

and

$$\text{svec}(Z_i) \approx \begin{cases} H_i^{-1} \text{svec}(D_1^i), & i \neq s \\ H_i^{-1} \left(\text{svec}(D_1^i) - N_i \begin{pmatrix} \text{svec}(\Delta P) \\ \Delta x \end{pmatrix} \right), & i = s. \end{cases} \quad (45)$$

This can be interpreted as the solution to an approximation of (39). Written on vectorized form, the approximate solution (44)–(45) is the solution to

$$\begin{pmatrix} H_1 & & & & & \\ & \ddots & & & & \\ & & H_s & & F_s & G_s \\ & & & \ddots & & \\ & & & & H_{n_i} & \\ & F_s^T & & & & \\ & G_s^T & & & & \end{pmatrix} \Delta = \mathbf{D}_{vec}. \quad (46)$$

This linear system of equations have a nice structure. ΔP , Δx and ΔZ_s can be found by solving a system of equations, as if we had a single constraint. The remaining dual variables ΔZ_i , $i \neq s$ are easily found by matrix inversions.

The constraint s is found by studying the eigenvalues of the W_i matrices. For each constraint the mean of the eigenvalues is calculated. The constraint with the smallest mean of the eigen-

values is defined to be s . The resulting equations for the preconditioner to solve are

$$W_i \Delta Z_i W_i = D_1^i, \quad i \neq s \quad (47a)$$

$$W_s \Delta Z_s W_s + \mathcal{F}_s(\Delta P) + \mathcal{G}_s(\Delta x) = D_1^s \quad (47b)$$

$$\mathcal{F}_s^*(\Delta Z_s) = D_2 \quad (47c)$$

$$\mathcal{G}_s^*(\Delta Z_s) = D_3 \quad (47d)$$

The procedure is described in Algorithm 2. The solution of (47b)–(47d) is a well-studied problem. By using the results in Wallin (2008), (47b)–(47d) can be solved at a total cost of $\mathcal{O}(n^3)$. Finally, the dual variables ΔZ_i in (47a) are easily obtained by matrix inversions.

Algorithm 2 Preconditioner II

1: Calculate the mean of the eigenvalues:

$$\bar{\lambda}_i \leftarrow \frac{1}{n+m} \sum_{j=1}^{n+m} \lambda_j(W_i), \quad i = 1, \dots, n_i$$

2: Identify the constraint:

$$s \leftarrow \underset{i}{\operatorname{argmin}} \bar{\lambda}_i$$

3: Solve (47b)–(47d) to obtain ΔP , Δx and ΔZ_s

4: Compute $\Delta Z_i = W_i^{-1} D_1^i W_i^{-1}$, $i \neq s$

7 Numerical Evaluation

All experiments are performed on a Dell Optiplex GX620 with 2GB RAM, Intel P4 640 (3.2GHz) CPU running under CentOS 4.1. Matlab version 7.4 (R2007a) is used with YALMIP version 3 (R20070810), Löfberg (2004), as interface to the solver. As comparison SDPT3 Version 4.0 (beta), Toh (2006), is used as underlying solver. Since the intention is to solve large scale optimization problems, the tolerance for termination is set to 10^{-3} for both the relative

$$\epsilon_{rel} = \frac{\|\mathcal{A}(P, x) + M_0 - S\|_2 + \|\mathcal{A}^*(Z) - C \oplus c\|_2 + \|ZS\|_2}{\|P\|_2 + \|x\|_2} \quad (48)$$

and absolute residual

$$\epsilon_{abs} = \|\mathcal{A}(P, x) + M_0 - S\|_2 + \|\mathcal{A}^*(Z) - C \oplus c\|_2 + \|ZS\|_2 \quad (49)$$

The residual is calculated as the 2-norm of the KKT equations (11)–(14). It is noted that both SDPT3 and the proposed solver terminate due to the relative residual being below the desired tolerance in all the problems in this simulation study.

A comparison of absolute solution times is not always fair since the choice of implementation language is crucial. In SDPT3 the expensive calculations are implemented in C while the overview algorithm is written in Matlab. For the algorithm described and evaluated in this work the main algorithm, the iterative solver and Preconditioner I are written in Matlab. However the

construction of the coefficient matrix in Preconditioner II is implemented in C and that is the operation that requires the most computational effort. Obviously an implementation in C of all sub-steps would improve the described solver. The similarity in the level of implementation in C makes a comparison in absolute computational time applicable. At least is the proposed solver not in favour in such a comparison.

The parameters in the algorithm are set to $\kappa = 0.01$, $\sigma_{max} = 0.9$, $\sigma_{min} = 0.01$, $\eta = 10^{-6}$, $\chi = 0.9$, $\epsilon = 10^{-8}$ and $\beta = 10^7 \cdot \beta_{lim}$, where $\beta_{lim} = \max(\|\mathcal{A}(P, x) + M_0 - S\|_2, \|\mathcal{A}^*(Z) - C \oplus c\|_2)$. The choice of parameter values are based on knowledge obtained during the development of the algorithm and through continuous evaluation. Note that the choice of σ_{max} is not within the range of values in the convergence proof given in the appendix. However, the choice is motivated by as good convergence as if $\sigma_{max} \in (\sigma_{min}, 0.5]$. This can be motivated by the fact that values close to zero and one correspond to predictor and corrector steps respectively.

Switching between the preconditioners is made after ten iterations. This is equivalent to five predictor-corrector steps. A more elaborate switching technique could improve the convergence but for practical use, the result is satisfactory.

The only information that Preconditioner II is given, is if the active constraint has changed. This information is obtained from the main algorithm.

To obtain the solution times, the Matlab command `cputime` is used. Input to the solvers are the system matrices, so any existing preprocessing of the problem is included in the total solution time.

In order to monitor the progress of the algorithm the residual for the search directions is calculated in each iteration in the iterative solver. To obtain further improvement in practice, one could use the in SQMR for free available bi-conjugate gradient (BCG) residual. However this results in that the exact residual is not known and hence the convergence might be affected.

Initialization

For comparable results, the initialization scheme given in Toh (2006) is used for the dual variables,

$$Z_i = \max \left(10, \sqrt{n+m}, \max_{k=1, \dots, n_x} \frac{(n+m)(1+|c_k|)}{1 + \|M_{i,k}\|_2} \right) I_{n+m},$$

where $\|\cdot\|_2$ denotes the Frobenius norm of a matrix. The slack variables are chosen as $S_i = Z_i$ while the primal variables are initialized as $P = I_n$ and $x = 0$.

Examples

To evaluate the suggested inexact algorithm with an iterative equations solver, randomly generated optimization problems are solved. The procedure to generate the examples is described below. For the examples in this section all matrices generated by `gallery.m` have a condition number of ten and are random. The procedure to generate an example is presented in Algorithm 1.

Algorithm 1 Generate Example

- 1: Define the scalar value δ .
 - 2: Generate the mean system matrices $\bar{A}, \bar{B}$ and $\bar{M}_k$ using `gallery.m`.
 - 3: Generate the system matrices A_i, B_i and $M_{i,k}$ as $A_i = \bar{A} \pm \delta \cdot \Delta_A$, $B_i = \bar{B} \pm \delta \cdot \Delta_B$ and $M_{i,k} = \bar{M}_{i,k} \pm \delta \cdot \Delta_{M_{i,k}}$. The matrix Δ_A is a diagonal matrix where the diagonal is generated by `rand.m` while Δ_B and $\Delta_{M_{i,k}}$ are generated by `gallery.m`.
 - 4: Define c and C such that a feasible optimization problem is obtained.
-

Results

To make an exhaustive investigation, different problem parameters have been investigated:

$$n \in \{10, 16, 25, 35, 50, 75, 100, 130, 165\}$$

$$m \in \{1, 3, 7\}$$

$$n_i \in \{2, 5\}$$

$$\delta \in \{0.01, 0.02, 0.05\}$$

For each case there are 15 generated examples in order to find the average solution time.

The simulation results are presented in Figure 1, Figure 2 and Figure 3. There the solution times for randomly generated problems, as described in Algorithm 1 are presented. The compared solvers are the suggested algorithm and the SDP solver SDPT3.

First the properties of the SDPT3 solver, Toh (2006), are discussed. This solver is well-tested and numerically stable. It solves all the problems generated up to $n = 75$. However, the solver does not solve the larger problems since the solution times tend to be unacceptable and for even larger problem the solver cannot proceed. When n and/or the number of constraints n_i is large, the solver will terminate due to memory restrictions. This motivate the use of inexact methods using an iterative solver since an iterative solver will require a substantial less amount of memory.

The suggested algorithm can solve large problems with lower computational time required than for the SDPT3 solver. A negative property that has been noted is that it will not converge on all the generated problems. Here it is the iterative solver for the search direction that fails to converge. However, when convergence occurs the solver is always faster than SDPT3 for large problems, $n \geq 50$. In the figures, only solution times for problems where a solution is obtained are presented.

For the 2430 generated problems, 11% does not converge due to numerical problems in the calculations to find the search direction. The inability to converge is not uniformly distributed. For $\delta = 0.01$ every problem is solved and the worst case is when $n = 165$, $n_i = 5$, $m = 7$ and $\delta = 0.05$ with a failure rate of 47%. This is natural since this example is the one where the assumptions made in Preconditioner II might not be valid. See appendix for further details.

8 Conclusions

For the investigated optimization problem a tailored inexact primal-dual interior-point method has been proposed and investigated. Convergence has been established for the algorithm. The use of an infeasible inexact method raised the possibility to use an iterative solver in each iterate in the algorithm to find the search direction. In order to obtain good performance in the iterative solver, two separate preconditioners were developed.

The proposed algorithm has been investigated in a simulation study. In the study the solution time was reduced when compared to a state of the art SDP solver. Numerical problems were noted for the iterative solver. However, the proposed solver solved large-scale problems with a size unmanageable for the generic solver.

Appendix A: Convergence Proof

A global proof of convergence for the proposed algorithm will be presented. This proof of convergence is due to Polak (1971). It has been applied to variational inequalities in Ralph and Wright (1997). Inexact solutions of the equations for the search direction were considered in Hansson (2000) for the application to model predictive control and in Gasparo (2002) for variational

Figure 1. Solution times for randomly generated problems. The suggested inexact primal-dual interior-point method is compared to the SDPT3 solver. In the examples $\delta = 0.01$. For each plot the solution time is plotted as a function of system order n .

inequalities. The convergence result is that either the sequence generated by the algorithm terminates at a solution to the Karush-Kuhn-Tucker conditions in a finite number of iterations, or all limit points, if any exist, are solutions to the Karush-Kuhn-Tucker conditions. Here the proof is extended to the case of semidefinite programming.

Figure 2. Solution times for randomly generated problems. The suggested inexact primal-dual interior-point method is compared to the SDPT3 solver. In the examples $\delta = 0.02$. For each plot the solution time is plotted as a function of system order n .

A.1 Convergence of Inexact Interior-Point Method

In order to prove convergence some preliminary results are presented in the following lemmas. First we define the set Ω_+ as

$$\Omega_+ = \{z \in \Omega \mid S \succ 0, Z \succ 0\}. \quad (\text{A1})$$

Figure 3. Solution times for randomly generated problems. The suggested inexact primal-dual interior-point method is compared to the SDPT3 solver. In the examples $\delta = 0.05$. For each plot the solution time is plotted as a function of system order n .

Lemma A.1: *Any iterate generated by the algorithm in Algorithm 1 is in Ω , which is a closed set. If $z \notin \mathcal{S}$ and $z \in \Omega$ then $z \in \Omega_+$.*

Proof Any iterate generated by the algorithm is in Ω from the definition of the algorithm. The set Ω is closed, since it is defined as an intersection of closed sets. For a detailed proof, see Section A.3.

The rest of the proof follows by contradiction. Assume that $z \notin \mathcal{S}$, $z \in \Omega$ and $z \notin \Omega_+$. Now study the two cases $\nu = 0$ and $\nu > 0$ separately. Note that $\nu \geq 0$ by definition.

First assume that $\nu = 0$. Since $\nu = 0$ and $z \in \Omega$ it follows that $\mathcal{K}_c(z) = 0$, $\|\mathcal{K}_p(z)\|_2 = 0$ and $\|\mathcal{K}_d(z)\|_2 = 0$. This implies that $\mathcal{K}_p(s) = 0$ and $\mathcal{K}_d(s) = 0$. Note that $z \in \Omega$ also implies that $Z \succeq 0$ and $S \succeq 0$. Combining these conclusions gives that $z \in \mathcal{S}$, which is a contradiction.

Now assume that $\nu > 0$. Since $\nu > 0$ and $z \in \Omega$ it follows that $\mathcal{K}_c(z) = \mathcal{H}(ZS) \succ 0$. To complete the proof two inequalities are needed. First note that $\det(\mathcal{H}(ZS)) > 0$. To find the second inequality the Ostrowski-Taussky inequality, see page 56 in Lütkepohl (1996),

$$\det\left(\frac{X + X^T}{2}\right) \leq |\det(X)|, \quad (\text{A2})$$

is applied to the symmetry transformation. This gives

$$\det(\mathcal{H}(ZS)) \leq |\det(R^{-1}ZSR)| = |\det(Z)| \cdot |\det(S)| = \det(Z) \cdot \det(S),$$

where the last equality follows from $z \in \Omega$. Combining the two inequalities gives

$$0 < \det(\mathcal{H}(ZS)) \leq \det(Z) \cdot \det(S). \quad (\text{A3})$$

Since the determinant is the product of all eigenvalues and $z \in \Omega$, (A3) shows that the eigenvalues are nonzero and therefore $Z \succ 0$ and $S \succ 0$. This implies that $z \in \Omega_+$, which is a contradiction. $\square$

The linear system of equations in (18)-(20) for the step direction is now rewritten as

$$\frac{\partial \text{vec}(\mathcal{K}(z))}{\partial \text{vec}(z)} \text{vec}(\Delta z) = \text{vec}(r). \quad (\text{A4})$$

Note that the vectorization is only used for the theoretical proof of convergence. In practice solving the equations is preferably made with a solver based on the operator formalism.

Lemma A.2: Assume that $\mathcal{A}$ has full rank. Let $\hat{z} \in \Omega_+$, and let $\epsilon \in (0, 1)$. Then there exist scalars $\hat{\delta} > 0$ and $\hat{\alpha} \in (0, 1]$ such that if

$$\left\| \frac{\partial \text{vec}(\mathcal{K}(z))}{\partial \text{vec}(z)} \text{vec}(\Delta z) - \text{vec}(r) \right\|_2 \leq \frac{\epsilon}{2} \sigma \beta \nu, \quad (\text{A5})$$

$$\frac{\partial \text{vec}(\mathcal{K}_c(z))}{\partial \text{vec}(z)} \text{vec}(\Delta z) - \text{vec}(r_c) = 0, \quad (\text{A6})$$

and if the algorithm takes a step from any point in

$$\mathcal{B} = \{z \mid \|z - \hat{z}\|_2 \leq \hat{\delta}\}, \quad (\text{A7})$$

then the calculated step length α will satisfy $\alpha \geq \hat{\alpha}$.

Proof See Section A.2. $\square$

Now the global convergence proof is presented.

Theorem A.3: Assume that $\mathcal{A}$ has full rank. Then for the iterates generated by the interior-point algorithm either

- $z_{(j)} \in \mathcal{S}$ for some finite j .
- all limit points of $\{z_{(j)}\}$ belongs to $\mathcal{S}$.

Remark A1: Note that nothing is said about the existence of a limit point. It is only stated that if a convergent subsequence exists, then its limit point is in $\mathcal{S}$. A sufficient condition for the existence of a limit point is that $\{z_{(j)}\}$ is uniformly bounded, Polak (1971).

Proof Suppose that the sequence $\{z_{(j)}\}$ is infinite and it has a subsequence which converges to $\hat{z} \notin \mathcal{S}$. Denote the corresponding subsequence $\{j_i\}$ with $\mathcal{K}$. Then for all $\delta' > 0$, there exist a k such that for $j \geq k$ and $j \in \mathcal{K}$ it holds that

$$\|z_{(j)} - \hat{z}\|_2 \leq \delta'. \quad (\text{A8})$$

Since $\hat{z} \notin \mathcal{S}$ it holds by Lemma A.1 that $\hat{z} \in \Omega_+$ and hence from Lemma A.2 and (8) that there exist a $\hat{\delta} > 0$ and $\hat{\alpha} \in (0, 1]$ such that for all $z_{(j)}$, $j \geq k$ such that $\|z_{(j)} - \hat{z}\|_2 \leq \hat{\delta}$ it holds that

$$\nu_{(j+1)} - \nu_{(j)} \leq -\hat{\alpha}\kappa(1 - \sigma_{\max})\nu_{(j)} < -\hat{\alpha}\kappa(1 - \sigma_{\max})\hat{\delta}^2 < 0.$$

Now take $\delta' = \hat{\delta}$. Then for two consecutive points $z_{(j)}$, $z_{(j+i)}$ of the subsequence with $j \leq k$ and $j \in \mathcal{K}$ it holds that

$$\begin{aligned} \nu_{(j+i)} - \nu_{(j)} &= (\nu_{(j+i)} - \nu_{(j+i-1)}) + (\nu_{(j+i-1)} - \nu_{(j+i-2)} + \dots) + (\nu_{(j+1)} - \nu_{(j)}) \\ &< \nu_{(j+1)} - \nu_{(j)} \leq -\hat{\alpha}\kappa(1 - \sigma_{\max})\hat{\delta}^2 < 0. \end{aligned} \quad (\text{A9})$$

Since $\mathcal{K}$ is infinite it holds that $\{\nu_{(j)}\}_{j \in \mathcal{K}}$ diverges to $-\infty$. The assumption $\hat{z} \in \Omega_+$ gives that $\hat{\nu} > 0$, which is a contradiction. $\square$

To simplify the proof of Lemma A.2, two preliminary lemmas are presented.

Lemma A.4: Assume $A \in \mathbb{S}^k$ is symmetric. If $\|A\|_2 \leq a$, then

$$-aI_k \preceq A \preceq aI_k. \quad (\text{A10})$$

Proof Denote the i :th largest eigenvalue of a matrix A with $\lambda_i(A)$. Then the norm fulfills

$$\|A\|_2^2 = \sum_{i=1}^k \lambda_i^2(A), \quad (\text{A11})$$

see (Boyd and Vandenberghe 2004, p. 647). Then

$$\|A\|_2 \leq a \Leftrightarrow \sum_{i=1}^k \lambda_i^2(A) \leq a^2 \Rightarrow \lambda_i^2(A) \leq a^2 \Rightarrow |\lambda_i(A)| \leq a \Leftrightarrow -a \leq \lambda_{\min}(A) \leq \lambda_i(A) \leq \lambda_{\max}(A) \leq a \quad (\text{A12})$$

and hence

$$-aI_k \preceq A \preceq aI_k. \quad (\text{A13})$$

$\square$

Lemma A.5: Assume $z \in \mathcal{B}$ as defined in (A7). Then $Z \succeq \hat{Z} - \hat{\delta}I_{n_i(n+m)}$ and $S \succeq \hat{S} - \hat{\delta}I_{n_i(n+m)}$.

Proof We have $\|z - \hat{z}\|_2 \leq \hat{\delta}$. Hence $\|z - \hat{z}\|_2^2 \leq \hat{\delta}^2$ and

$$\|S - \hat{S}\|_2^2 + \|Z - \hat{Z}\|_2^2 + \|x - \hat{x}\|_2^2 \leq \hat{\delta}^2. \quad (\text{A14})$$

Therefore $\|S - \hat{S}\|_2^2 \leq \hat{\delta}^2$ and $\|Z - \hat{Z}\|_2^2 \leq \hat{\delta}^2$. Hence by Lemma A.4 above $-\hat{\delta}I_{n_i(n+m)} \leq S - \hat{S} \leq \hat{\delta}I_{n_i(n+m)}$ and $-\hat{\delta}I_{n_i(n+m)} \leq Z - \hat{Z} \leq \hat{\delta}I_{n_i(n+m)}$. $\square$

A.2 Proof of Lemma A.2

Here a proof of Lemma A.2 is presented. What is needed to show is that there exist an α such that $\alpha \geq \hat{\alpha} > 0$ so that $z(\alpha) = z + \alpha\Delta z \in \Omega$ and that

$$\nu(\alpha) \leq (1 - \alpha\kappa(1 - \sigma))\nu. \quad (\text{A15})$$

First a property for symmetric matrices is stated for later use in the proof.

$$A \succeq B, C \succeq D \Rightarrow \langle A, C \rangle \geq \langle B, D \rangle, \quad (\text{A16})$$

where $A, B, C, D \in \mathbb{S}_+^n$, page 44, Property (18) in Lütkepohl (1996).

Define

$$\hat{\delta} = \frac{1}{2} \min \left(\min_i \lambda_i(\hat{S}), \min_i \lambda_i(\hat{Z}) \right). \quad (\text{A17})$$

Notice that $z \in \mathcal{B}$ by Lemma A.5 implies that $S \succeq \hat{S} - \hat{\delta}I_{n_i(n+m)}$ and $Z \succeq \hat{Z} - \hat{\delta}I_{n_i(n+m)}$. Then it holds that

$$\nu = \frac{\langle Z, S \rangle}{n_i(n+m)} \geq \frac{\langle \hat{Z} - \hat{\delta}I_{n_i(n+m)}, \hat{S} - \hat{\delta}I_{n_i(n+m)} \rangle}{n_i(n+m)} \geq \frac{\langle \hat{\delta}I_{n_i(n+m)}, \hat{\delta}I_{n_i(n+m)} \rangle}{n_i(n+m)} = \hat{\delta}^2 > 0, \quad (\text{A18})$$

where the first inequality follows from (A16) and what was mentioned above. The second inequality is due to (A17) and (A16). Similarly as in the proof of Lemma A.1 it also holds that $Z, S \succ 0$.

Now since $\mathcal{A}$ has full rank it follows by Lemma 3.1 that $(\partial \text{vec}(\mathcal{K}(z))/\partial \text{vec}(z))$ is invertible in an open set containing $\mathcal{B}$. Furthermore, it is clear that $(\partial \text{vec}(\mathcal{K}(z))/\partial \text{vec}(z))$ is a continuous function of z , and that r is a continuous function of z and σ . Therefore, for all $z \in \mathcal{B}$, $\sigma \in (\sigma_{\min}, 1/2]$, it holds that the solution Δz_0 of

$$\frac{\partial \text{vec}(\mathcal{K}(z))}{\partial \text{vec}(z)} \text{vec}(\Delta z_0) = \text{vec}(r), \quad (\text{A19})$$

satisfies $\|\Delta z_0\|_2 \leq C_0$ for a constant $C_0 > 0$. Introduce $\delta\Delta z = \Delta z - \Delta z_0$. Then from the bound on the residual in the assumptions of this lemma it follows that

$$\left\| \frac{\partial \text{vec}(\mathcal{K}(z))}{\partial \text{vec}(z)} \text{vec}(\delta\Delta z) \right\|_2 \quad (\text{A20})$$

is bounded. Hence there must be a constant $\delta C > 0$ such that $\|\delta\Delta z\|_2 \leq \delta C$. Let $C = C_0 + \delta C > 0$. It now holds that $\|\Delta z\|_2 \leq C$ for all $z \in \mathcal{B}$, $\sigma \in (\sigma_{\min}, 1/2]$. Notice that it also holds that $\|\Delta Z\|_2 \leq C$ and $\|\Delta S\|_2 \leq C$. Define $\hat{\alpha}^{(1)} = \hat{\delta}/2C$. Then for all $\alpha \in (0, \hat{\alpha}^{(1)}]$ it holds that

$$\begin{aligned} Z(\alpha) &= Z + \alpha\Delta Z \succeq \hat{Z} - \hat{\delta}I_{n_i(n+m)} + \alpha\Delta Z \succeq \\ &2\hat{\delta}I_{n_i(n+m)} - \hat{\delta}I_{n_i(n+m)} - \frac{\hat{\delta}}{2}I_{n_i(n+m)} \succ 0, \end{aligned} \quad (\text{A21})$$

where the first inequality follows by Lemma A.5 and $z \in \mathcal{B}$, and where the second inequality follows from (A17). The proof for $S(\alpha) \succ 0$ is analogous. Hence it is possible to take a positive step without violating the constraint $S(\alpha) \succ 0$, $Z(\alpha) \succ 0$.

Now we prove that

$$\mathcal{K}_c(z) = \mathcal{H}(Z(\alpha)S(\alpha)) \succeq \eta\nu(\alpha)I_{n_i(n+m)} \quad (\text{A22})$$

for some positive α . This follows from two inequalities that utilize the fact that

$$\begin{aligned} \frac{\partial \text{vec}(\mathcal{K}_c(z))}{\partial \text{vec}(z)} \Delta z - \text{vec}(r_c) &= 0 \Leftrightarrow \\ \mathcal{H}(\Delta ZS + Z\Delta S) + \mathcal{H}(ZS) - \sigma\nu I_{n_i(n+m)} &= 0 \end{aligned} \quad (\text{A23})$$

and the fact that the previous iterate z is in the set Ω . Note that $\mathcal{H}(\cdot)$ is a linear operator. Hence

$$\begin{aligned} \mathcal{H}(Z(\alpha)S(\alpha)) &= \mathcal{H}(ZS + \alpha(\Delta ZS + Z\Delta S) + \alpha^2\Delta Z\Delta S) \\ &= (1 - \alpha)\mathcal{H}(ZS) + \alpha(\mathcal{H}(ZS) + \mathcal{H}(\Delta ZS + Z\Delta S)) + \alpha^2\mathcal{H}(\Delta Z\Delta S) \\ &\succeq (1 - \alpha)\eta\nu I_{n_i(n+m)} + \alpha\sigma\nu I_{n_i(n+m)} + \alpha^2\mathcal{H}(\Delta Z\Delta S) \\ &\succeq ([(1 - \alpha)\eta + \alpha\sigma]\nu - \alpha^2|\lambda_{\min}^\Delta|)I_{n_i(n+m)}, \end{aligned} \quad (\text{A24})$$

where $\lambda_{\min}^\Delta = \min_i \lambda_i(\mathcal{H}(\Delta Z\Delta S))$. Note that $\lambda_{\min}^\Delta$ is bounded. To show this we note that in each iterate in the algorithm Z and S are bounded and hence R is bounded since it is calculated from Z and S . Hence $\mathcal{H}(\Delta Z\Delta S)$ is bounded since $\|\Delta Z\|_2 < C$ and $\|\Delta S\|_2 < C$.

Moreover

$$\begin{aligned} n_i(n+m)\nu(\alpha) &= \langle Z(\alpha), S(\alpha) \rangle = \langle \mathcal{H}(Z(\alpha)S(\alpha)), I_{n_i(n+m)} \rangle \\ &= \langle (1 - \alpha)\mathcal{H}(ZS) + \alpha\sigma\nu I_{n_i(n+m)} + \alpha^2\mathcal{H}(\Delta Z\Delta S), I_{n_i(n+m)} \rangle \\ &= (1 - \alpha)\langle Z, S \rangle + n_i(n+m)\alpha\sigma\nu + \alpha^2\langle \Delta Z, \Delta S \rangle \\ &\leq (1 - \alpha)n_i(n+m)\nu + n_i(n+m)\alpha\sigma\nu + \alpha^2C^2. \end{aligned} \quad (\text{A25})$$

The last inequality follows from $\langle \Delta Z, \Delta S \rangle \leq \|\Delta Z\|_2 \|\Delta S\|_2 \leq C^2$. Rewriting (A25) gives that

$$\eta\nu(\alpha) \leq \left((1 - \alpha + \alpha\sigma)\nu + \frac{\alpha^2C^2}{n_i(n+m)} \right) \eta. \quad (\text{A26})$$

Clearly

$$\left([(1 - \alpha)\eta + \alpha\sigma]\nu - \alpha^2|\lambda_{\min}^\Delta| \right) I_{n_i(n+m)} \succeq \left([(1 - \alpha)\eta + \alpha\sigma\eta]\nu + \frac{\alpha^2C^2\eta}{n_i(n+m)} \right) I_{n_i(n+m)} \quad (\text{A27})$$

implies (A22), which (assuming that $\alpha > 0$) is fulfilled if

$$\alpha \leq \frac{\sigma\nu(1 - \eta)}{C^2\eta/n_i(n+m) + |\lambda_{\min}^\Delta|}. \quad (\text{A28})$$

Recall that $\sigma \geq \sigma_{\min} > 0$, $\eta < \eta_{\max} < 1$ by assumption and that $0 < \hat{\delta}^2 \leq \nu$ by (A18). Hence

with

$$\hat{\alpha}^{(2)} = \min \left(\hat{\alpha}^{(1)}, \frac{\sigma_{\min} \hat{\delta}^2 (1 - \eta_{\max})}{C^2 \eta_{\max} / n_i (n + m) + |\lambda_{\min}^\Delta|} \right), \quad (\text{A29})$$

(A22) is satisfied for all $\alpha \in (0, \hat{\alpha}^{(2)}]$.

We now show that

$$\gamma \nu(\alpha) I_{n_i(n+m)} \succeq \mathcal{H}(Z(\alpha)S(\alpha)). \quad (\text{A30})$$

First note that $\gamma \geq n_i(n + m)$. Then

$$\begin{aligned} \frac{\gamma}{n_i(n + m)} \sum_i \lambda_i(\mathcal{H}(Z(\alpha)S(\alpha))) &\geq \lambda_{\max}(\mathcal{H}(Z(\alpha)S(\alpha))) \Leftrightarrow \\ \frac{\gamma}{n_i(n + m)} \sum_i \lambda_i(\mathcal{H}(Z(\alpha)S(\alpha))) I_{n_i(n+m)} &\succeq \lambda_{\max}(\mathcal{H}(Z(\alpha)S(\alpha))) I_{n_i(n+m)} \Leftrightarrow \\ \frac{\gamma}{n_i(n + m)} \text{Tr}(\mathcal{H}(Z(\alpha)S(\alpha))) I_{n_i(n+m)} &\succeq \lambda_{\max}(\mathcal{H}(Z(\alpha)S(\alpha))) I_{n_i(n+m)}, \end{aligned}$$

where the first expression is fulfilled by definition. The second equivalence follows from a property of the trace of a matrix, see page 41 in Lütkepohl (1996). From the definition of duality measure in (8) it now follows that (A30) holds.

Now we prove that (A15) is satisfied. Let

$$\hat{\alpha}^{(3)} = \min \left(\hat{\alpha}^{(2)}, \frac{n_i(n + m) \hat{\delta}^2 (1 - \kappa)}{2C^2} \right). \quad (\text{A31})$$

Then for all $\alpha \in [0, \hat{\alpha}^{(3)}]$ it holds that

$$\begin{aligned} \alpha^2 C^2 &\leq \alpha n_i(n + m)(1 - \kappa) \hat{\delta}^2 / 2 \\ &\leq \alpha n_i(n + m)(1 - \kappa) \nu / 2 \\ &\leq \alpha n_i(n + m)(1 - \kappa)(1 - \sigma) \nu, \end{aligned} \quad (\text{A32})$$

where the second inequality follows from (A18) and the third inequality follows from the assumption $\sigma \leq 1/2$. This inequality together with (A25) implies that

$$\begin{aligned} \nu(\alpha) &\leq ((1 - \alpha) + \alpha \sigma) \nu + \frac{\alpha^2 C^2}{2} \\ &\leq ((1 - \alpha) + \alpha \sigma) \nu + \alpha(1 - \kappa)(1 - \sigma) \nu \\ &= (1 - \alpha \kappa(1 - \sigma)) \nu, \end{aligned} \quad (\text{A33})$$

where the second inequality is due to $\sigma < 1/2$ and hence (A15) is satisfied.

It now remains to prove that

$$\|\mathcal{K}_p(z(\alpha))\|_2 \leq \beta \nu(\alpha) \quad (\text{A34a})$$

$$\|\mathcal{K}_d(z(\alpha))\|_2 \leq \beta \nu(\alpha). \quad (\text{A34b})$$

Since the proofs are similar, it will only be proven that (A34a) holds true. Use Taylor's theorem

to write

$$\mathbf{vec}(\mathcal{K}_p(z(\alpha))) = \mathbf{vec}(\mathcal{K}_p(z)) + \alpha \frac{\partial \mathbf{vec}(\mathcal{K}_p(z))}{\partial \mathbf{vec}(z)} \mathbf{vec}(\Delta z) + \alpha R$$

where

$$R = \int_0^1 \left(\frac{\partial \mathbf{vec}(\mathcal{K}_p(z(\theta\alpha)))}{\partial \mathbf{vec}(z)} - \frac{\partial \mathbf{vec}(\mathcal{K}_p(z))}{\partial \mathbf{vec}(z)} \right) \cdot \mathbf{vec}(\Delta z) d\theta.$$

Using $\Delta z = \Delta z_0 + \delta \Delta z$ and the fact that

$$\frac{\partial \mathbf{vec}(\mathcal{K}_p(z))}{\partial \mathbf{vec}(z)} \mathbf{vec}(\Delta z_0) = -\mathbf{vec}(\mathcal{K}_p(z)), \quad (\text{A35})$$

it follows that

$$\mathbf{vec}(\mathcal{K}_p(z(\alpha))) = (1 - \alpha) \mathbf{vec}(\mathcal{K}_p(z)) + \alpha \frac{\partial \mathbf{vec}(\mathcal{K}_p(z))}{\partial \mathbf{vec}(z)} \mathbf{vec}(\delta \Delta z) + \alpha R. \quad (\text{A36})$$

Furthermore

$$\|R\|_2 \leq \max_{\theta \in (0,1)} \left\| \frac{\partial \mathbf{vec}(\mathcal{K}_p(z(\theta\alpha)))}{\partial \mathbf{vec}(z)} - \frac{\partial \mathbf{vec}(\mathcal{K}_p(z))}{\partial \mathbf{vec}(z)} \right\|_2 \cdot \|\mathbf{vec}(\Delta z)\|_2. \quad (\text{A37})$$

Since $\|\Delta z\|_2 \leq C$ and since $\partial \mathbf{vec}(\mathcal{K}_p(z))/\partial \mathbf{vec}(z)$ is continuous, there exists an $\hat{\alpha}^{(4)} > 0$ such that for $\alpha \in (0, \hat{\alpha}^{(4)})$, $\epsilon \in (0, 1)$ and $z \in \mathcal{B}$ it holds that

$$\|R\|_2 < \frac{1 - \epsilon}{2} \sigma \beta \nu. \quad (\text{A38})$$

Using the fact that $\|\mathcal{K}_p(z)\|_2 \leq \beta \nu$, it now follows that

$$\|\mathcal{K}_p(z(\alpha))\|_2 \leq (1 - \alpha) \beta \nu + \alpha \frac{\sigma}{2} \beta \nu, \quad (\text{A39})$$

for all $\alpha \in [0, \hat{\alpha}^{(4)}]$. By reducing $\hat{\alpha}^{(4)}$, if necessary, it follows that

$$\alpha C^2 < \frac{\sigma}{2} n_i(n+m) \nu, \quad (\text{A40})$$

for all $\alpha \in [0, \hat{\alpha}^{(4)}]$. Similar to (A25), but bounding below instead, it holds that

$$n_i(n+m) \nu(\alpha) \geq (1 - \alpha(1 - \sigma)) n_i(n+m) \nu - \alpha^2 C^2 \Leftrightarrow (1 - \alpha) \nu \leq \nu(\alpha) - \alpha \sigma \nu - \frac{\alpha^2 C^2}{n_i(n+m)}. \quad (\text{A41})$$

Hence

$$\|\mathcal{K}_p(z(\alpha))\|_2 \leq \beta \left(\nu(\alpha) - \alpha \sigma \nu + \frac{\alpha^2 C^2}{n_i(n+m)} \right) + \alpha \frac{\sigma}{2} \beta \nu = \beta \nu(\alpha) - \alpha \beta \left(\sigma \nu - \frac{\alpha C^2}{n_i(n+m)} - \frac{\sigma \nu}{2} \right) \leq \beta \nu(\alpha), \quad (\text{A42})$$

where the last inequality follows from (A40). The proof for $\mathcal{K}_d(z(\alpha)) \leq \beta \nu(\alpha)$ is done analogously, which gives an $\hat{\alpha}^{(5)} > 0$.

A.3 Proof of Closed Set

Here the proof of that the set Ω defined in (27) defines a closed set is presented.

Definition A.6: Let X and Y denote two metric spaces and define the mapping $f : X \rightarrow Y$. Let $C \subseteq Y$. Then the inverse image $f^{-1}(C)$ of C is the set $\{x \mid f(x) \in C, x \in X\}$.

Lemma A.7: A mapping of a metric space X into a metric space Y is continuous if and only if $f^{-1}(C)$ is closed in X for every closed set $C \subseteq Y$.

Proof See page 81 in Rudin (1976). $\square$

Lemma A.8: The set $\{z \mid \|\mathcal{K}_p(z)\|_2 \leq \beta\nu, z \in \mathcal{Z}\}$ is a closed set.

Proof Consider the mapping $\mathcal{K}_p(z) : \mathcal{Z} \rightarrow \mathbb{S}^n$ and the set $\mathcal{C} = \{C \mid \|C\|_2 \leq \beta\nu, C \in \mathbb{S}^n\}$ that is closed since the norm defines a closed set. This follows from that we have $\mathcal{C} = f^{-1}([0, \beta\nu])$, where $f(\cdot) = \|\cdot\|_2$. Now the claim follows by Lemma A.7. To see this note that the mapping $\mathcal{K}_p(z)$ is continuous. Then the inverse image $\{z \mid \mathcal{K}_p(z) \in \mathcal{C}, z \in \mathcal{Z}\} = \{z \mid \|\mathcal{K}_p(z)\|_2 \leq \beta\nu, z \in \mathcal{Z}\}$ is a closed set. $\square$

Lemma A.9: The set $\{z \mid \|\mathcal{K}_d(z)\|_2 \leq \beta\nu, z \in \mathcal{Z}\}$ is a closed set.

Proof Analogous with the proof of Lemma A.8. $\square$

Lemma A.10: The set $\{z \mid \gamma\nu I_{n_i(n+m)} \succeq \mathcal{H}(ZS) \succeq \eta\nu I_{n_i(n+m)}, z \in \mathcal{Z}\}$ is a closed set.

Proof Define the mapping $h(z) = \mathcal{H}(ZS) : \mathcal{Z} \rightarrow \mathbb{S}^n$ and the set $\mathcal{C}_1 = \{C_1 \mid C_1 \succeq \eta\nu I_{n_i(n+m)}, C_1 \in \mathbb{S}^n\}$ that is closed, page 43 in Boyd and Vandenberghe (2004). Since the mapping is continuous the inverse image $\{z \mid \mathcal{H}(ZS) \succeq \eta\nu I_{n_i(n+m)}, z \in \mathcal{Z}\}$ is a closed set. Now define the set $\mathcal{C}_2 = \{C_2 \mid C_2 \preceq \gamma\nu I_{n_i(n+m)}, C_2 \in \mathbb{S}^n\}$. Using continuity again gives that $\{z \mid \mathcal{H}(ZS) \preceq \gamma\nu I_{n_i(n+m)}, z \in \mathcal{Z}\}$ is a closed set. Since the intersection of closed sets is a closed set, Theorem 2.24 in Rudin (1976), $\{z \mid \gamma\nu I_{n_i(n+m)} \succeq \mathcal{H}(ZS) \succeq \eta\nu I_{n_i(n+m)}, z \in \mathcal{Z}\}$ is a closed set. $\square$

Lemma A.11: The set Ω is a closed set.

Proof Note that $(\mathcal{Z}, \rho)$ is a metric space with distance $\rho(u, v) = \|u - v\|_2$. Hence is $\mathcal{Z}$ a closed set. Lemmas A.8, A.9 and A.10 gives that each constraint in Ω defines a closed set. Since the intersection of closed sets is a closed set, Theorem 2.24 in Rudin (1976), shows that Ω is a closed set. $\square$

References

- Boyd, S., Laurent, E.G., Feron, E., and Balakrishnan, V., *Linear Matrix inequalities in System and Control Theory*, SIAM (1994).
- Gahinet, P., Apkarian, P., and Chilali, M. (1996), “Parameter-Dependent Lyapunov Functions for Real Parametric Uncertainty,” *IEEE Transactions on Automatic Control*, 41(3), 436 – 442.
- Wang, F., and Balakrishnan, V. (2002), “Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems,” *Automatic Control, IEEE Transactions on*, 47(5), 720–734.
- Löfberg, J. (2004), “YALMIP : A Toolbox for Modeling and Optimization in Matlab,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan.
- Toh, K.C., Todd, M.J., and Tütüncü, R.H. (2006), “On the implementation and usage of SDPT3 — a Matlab software package for semidefinite-quadratic-linear programming, version 4.0,”
- Benson, S.J., and Yinyu, Y. (2005), “DSDP5: Software For Semidefinite Programming,” Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL.

- Sturm, J.F., "Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones," (2001).
- Pólik, I., "Addendum to the SeDuMi user guide, version 1.1," (2005).
- Chai, J.S., and Toh, K.C. (2007), "Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming," *Computational Optimization and Applications*, 36(2-3), 221–247.
- Bonettini, S., and Ruggiero, V. (2007), "Some iterative methods for the solution of a symmetric indefinite KKT system," *Computational Optimization and Applications*, 38(1), 3 – 25.
- Cafieri, S., D'Apuzzo, M., De Simone, V., and di Serafino, D. (2007), "On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems," *Computational Optimization and Applications*, 38(1), 27 – 45.
- Bergamaschi, L., Gondzio, J., and Zilli, G. (2004), "Preconditioning indefinite systems in interior point methods for optimization," *Computational Optimization and Applications*, 28(2), 149 – 171.
- Gillberg, J., and Hansson, A. (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*, Dec., Maui, Hawaii, USA, p. 6.
- Hansson, A., and Vandenberghe, L. (2001), "A primal-dual potential reduction method for integral quadratic constraints," in *Proceedings of the American Control Conference*, Arlington, Virginia, USA, pp. 3013–3017.
- Vandenberghe, L., and Boyd, S. (1995), "A primal-dual potential reduction method for problems involving matrix inequalities," *Mathematical Programming*, 69, 205 – 236.
- Hansson, A. (2000), "A primal-dual interior-point method for robust optimal control of linear discrete-time systems," *Automatic Control, IEEE Transactions on*, 45(9), 1639–1655.
- Boyd, S., and Vandenberghe, L., *Convex optimization*, Cambridge University Press (2004).
- Wolkowicz, H., Saigal, R., and Vandenberghe, L. (eds.) *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Vol. 27 of *International series in operations research & management science*, KLUWER (2000).
- Zhang, Y. (1998), "On Extending Some Primal–Dual Interior-Point Algorithms From Linear Programming to Semidefinite Programming," *SIAM Journal on Optimization*, 8(2), 365–386.
- Todd, M.J., Toh, K.C., and Tütüncü, R.H. (1998), "On the Nesterov-Todd direction in semidefinite programming," *SIAM Journal on Optimization*, 8(3), 769–796.
- Benzi, M., Golub, G.H., and Liesen, J. (2005), "Numerical solution of saddle point problems," *Acta numerica*, 14, 1 – 137.
- Ralph, D., and Wright, S.J. (1997), "Superlinear convergence of an interior-point method for monotone variational inequalities," *Complementarity and Variational Problems: State of the Art*.
- Nesterov, Y., and Todd, M.J. (1997), "Self-scaled barriers and interior-point methods for convex programming," *Mathematics of Operations Research*, 22(1), 1 – 42.
- Vandenberghe, L., Balakrishnan, V.R., Wallin, R., Hansson, A., and Roh, T. (2005), Lecture Notes on Control and Information Sciences, "Interior-point algorithms for semidefinite programming problems derived from the KYP lemma," *Positive Polynomials in Control*, Springer Verlag.
- Greenbaum, A., *Iterative methods for solving linear systems*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (1997).
- Freund, R.W., and Nachtigal, N.M. (1991), "QMR: a quasi-minimal residual method for non-Hermitian linear systems," *Numerische Mathematik*, 60(3), 315 – 339.
- Freund, R.W., and Nachtigal, N.M. (1994), "A New Krylov-Subspace Method for Symmetric Indefinite Linear Systems," in *Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics*, IMACS, pp. 1253–1256.
- Rozloznik, M., and Simoncini, V. (2003), "Krylov subspace methods for saddle point problems

- with indefinite preconditioning," *SIAM journal on matrix analysis and applications*, 24(2), 368 – 391.
- Forsgren, A., Gill, P.E., and Griffin, J.D. (2007), "Iterative Solution of Augmented Systems Arising in Interior Methods," *SIAM Journal on Optimization*, 18(2), 666–690.
- Dollar, H.S., Gould, N.I.M., Schilders, W.H.A., and Wathen, A.J. (2006), "Implicit-Factorization Preconditioning and Iterative Solvers for Regularized Saddle-Point Systems," *SIAM Journal on Matrix Analysis and Applications*, 28(1), 170–189.
- Keller, C., Gould, N.I.M., and Wathen, A.J. (2000), "Constraint Preconditioning for Indefinite Linear Systems," *SIAM Journal on Matrix Analysis and Applications*, 21(4), 1300 – 1317.
- Benzi, M., and Golub, G.H. (2004), "A Preconditioner for Generalized Saddle Point Problems," *SIAM Journal on Matrix Analysis and Applications*, 26(1), 20–41.
- Botchev, M.A., and Golub, G.H. (2006), "A Class of Nonsymmetric Preconditioners for Saddle Point Problems," *SIAM Journal on Matrix Analysis and Applications*, 27(4), 1125–1149.
- Saad, Y., *Iterative Methods for Sparse Linear Systems*, Boston, USA: PWS Publishing Company (1996).
- Bocanegra, S., Campos, F.F., and Oliveira, A.R.L. (2007), "Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods," *Computational Optimization and Applications*, 36(2-3), 149–164.
- Gill, P.E., Murray, W., Ponceleón, D.B., and Saundeeers, M. (1992), "Preconditioners for indefinite systems arising in optimization," *SIAM journal on matrix analysis and applications*, 13(1), 292 – 311.
- Wallin, R., Hansson, A., and Harju Johansson, J. (2008), "A structure exploiting preprocessor for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma," *IEEE Transactions on Automatic Control*, Provisionally accepted.
- Polak, E., *Computational methods in optimization*, Academic press (1971).
- Gasparo, M.G., Pieraccini, S., and Armellini, A. (2002), "An Infeasible Interior-Point Method with Nonmonotonic Complementarity Gaps," *Optimization Methods and Software*, 17(4), 561 – 586.
- Lütkepohl, H., *Handbook of matrices*, John Wiley and sons (1996).
- Rudin, W., *Principles of Mathematical Analysis*, 3 ed., McGraw-Hill (1976).