Robust Control via Concave Minimization
Local and Global Algorithms

Pierre Apkarian[*] *Hoang Duong Tuan*[‡]

Abstract

This paper is concerned with the robust control problem of LFT (Linear Fractional Representation) uncertain systems depending on a time-varying parameter uncertainty. Our main result exploits an LMI (Linear Matrix Inequality) characterization involving scalings and Lyapunov variables subject to an additional essentially non-convex algebraic constraint. The non-convexity enters the problem in the form of a rank deficiency condition or matrix inverse relation on the scalings only. It is shown that such problems but also more generally rank inequalities and bilinear constraints can be formulated as the minimization of a concave functional subject to Linear Matrix Inequality constraints. First of all, a *local* Frank and Wolfe feasible direction algorithm is introduced in this context to tackle this hard optimization problem. Exploiting the attractive concavity structure of the problem, several efficient *global* concave programming methods are then introduced and combined with the local feasible direction method to secure and certify global optimality of the solutions. Convergence and practical implementation details of the algorithms are covered. Stopping criteria are introduced in order to reduce the overall computational overhead.

Computational experiments indicate the viability of our algorithms, and that in the worst case they require the solution of a few LMI programs. Power and efficiency of the algorithms are demonstrated through realistic and randomized numerical experiments.

Key words. Linear Matrix Inequalities, parametric uncertainty, global concave minimization, Frank and Wolfe algorithms.

1 Introduction

A number of challenging problems in robust control theory fall within the class of rank minimization problems subject to LMI (convex) constraints. An important example is provided by the reduced-order H_∞ control problem. It has been shown in [24, 8, 18] that there exists a k-th order controller solving the H_∞ control problem if and only if one can find a pair of symmetric matrices (X, Y) satisfying LMIs constraint with

$$\text{Rank} \begin{bmatrix} X & I \\ I & Y \end{bmatrix} \leq n + k \,, \tag{1}$$

[*]ONERA-CERT, Control System Dept., 2 av. Edouard Belin, 31055 Toulouse, FRANCE - Email : apkarian@cert.fr - Tel : +33 5.62.25.27.84 - Fax : +33 5.62.25.27.64

[‡]Department of Electronic-Mechanical Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN - Email : tuan@nuem.nagoya-u.ac.jp

where n designates the plant's order. The hardness of this problem stems from the rank condition (1) which is essentially non-convex. Different proofs of NP-hardness are given in [29, 7].

As it plays a central role in robust control theory, many researchers have devoted their efforts to developing adequate algorithms and heuristics for determining solutions to this class of problems. In [13], Grigoriadis and Skelton consider a heuristic method based on *alternating projections* for iteratively finding a solution to the rank constraint (1). In [17], Iwasaki derives an heuristic iterative scheme taking advantage of *primal and dual* formulations of the fixed-order control problem and demonstrates its practicality by extensive tests and investigations. Non-trivial lower and upper bounds of the above problem are obtained in [21] which however are used for a relaxation rather than for *Branch and Bound* (BB) refinement schemes to locate approximate solutions. Earlier works on the use of general-purpose global optimization for solving BMI problems can be found in [25, 12] and references therein. In [10] Geromel et al. introduce a *min/max algorithm* for solving the reduced-order stabilization problem and discuss its convergence properties. A closely related algorithm, referred to as the *cone complementary linearization algorithm* is elaborated in [11] by El Ghaoui et al. The authors introduce a nonlinear objective functional whose optimal value corresponds to solutions to the lower-order stabilization problem. Following the ideas of Frank and Wolfe (FW) in [6], each step of the algorithm utilizes a local linearization of the functional to determine a "best" feasible descent direction and therefore a feasible line segment in the constraint set. In addition to convergence, it is shown that the algorithm enforces some rank deficiency at each step. In [30], we developed a global optimization technique based upon d.c. (difference of convex functions/sets) optimization techniques exploiting the fact that the reverse convex constraints are of relatively low-rank, which is of primary importance to ensure practicality of the algorithm. This technique is however currently limited to the case of symmetric scalings and hardly generalizes to more complex structures.

The contribution of this paper is threefold.

• It is first shown that several important problems in robust control theory which involve bilinear constraints, equality and inequality rank constraints or matrix inverse constraints, can be recast as finding zero optimal solutions to generalized concave programs. These generalized concave programs consist in the minimization of a concave functional subject to convex constraints consisting of LMIs. A distinguished characteristic of these problems is that only *zero* solutions are of interest. This significantly reduces the difficulty of the search and thus makes the problems much more computationally attractive and painless than the conventional concave programs which seek an arbitrary minimum of a concave function over a convex set. A sample list of control applications of this new formulation includes robust control and robust multi-objective problems based on any kind of scalings or multipliers, robust fixed- or reduced-order control problems, multi-objective Linear Parameter-Varying (LPV) control, reduction of LFT representations, and more generally combinations of such problems. Starting from this viewpoint, the work here provides first a full generalization of the technique in [11] to handle robust control problems for plants subject to time-varying LFT (Linear Fractional Transformation) uncertainties. More precisely, we show that the robust synthesis problems involving either pairs of symmetric and skew-symmetric scalings or full generalized scalings as discussed in [26] are equivalent to zero-seeking concave programming problems where the convex constraints express in terms of LMIs. Although, this is not the central object of this paper, we reveal that BMI (Bilinear Matrix Inequality) problems can also be formulated in the same fashion, so that in this respect, concavity appears to be the most prominent feature of a very vast array of problems in control theory.

• It develops generalizations of local and global optimization methods for solving these zero-seeking concave programs. In this respect, we indicate how the FW algorithm must be modified to handle our problems. We prove that due to the concave structure, the FW algorithm is not only guaranteed to generate strictly decreasing sequences for the objective functional but also that the sequence of points is either infinite or reaches a local optimal solution. Also the traditional line search at every iteration can be bypassed as a consequence of concavity. However, the FW algorithm is a local method and is not guaranteed to provide a global solution. This naturally leads us to combining recently available global search techniques with the FW algorithm to certify global optimality of the solutions or invalidate feasibility of the problem.

As concave programming is the best studied class of problems in global optimization [15, 16, 20, 32], we have exploited several key basic concepts for developing efficient and practical algorithms suitably generalized to the matrix context of our problems. As mentioned previously, the properties of zero-seeking concave programs make them much more computationally tractable than conventional concave programs. Our efforts in this direction are thus to maximally exploit this fact. Namely, we have paid special attention for developing extensions of the simplicial and conical BB concave minimization methods which work with matrices and over the positive semidefinite cone. These methods respectively divide the feasible set into matrix simplices and matrix cones of decreasing sizes. Their main thrust is that they rely heavily on our specific matrix structures, on concavity and convexity geometric concepts which make them particularly appropriate for our problems. Each step of the proposed techniques exploits both the convexity of the constraint set and the concavity of the functional and also the fact that only zero optimal values are of interest. This allows large portions of the feasible set to be eliminated at each iteration. The most computationally demanding operation in each step comes down to solving one LMI program, hence the practicality of the methods.

There is a obvious trade-off between local and global search techniques. The FW algorithm is much less costly but in return, is prone to non-global optimality. On the other hand, concave programming techniques provide global optimal solutions but generally require intensive computations. Therefore, an important target of this paper is to maintain a reasonable computational cost by combining local and global techniques. Hence, the global concave programming techniques are used either to refine a local solution issued from the FW algorithm until global optimality is achieved or to provide a certificate of global optimality.

• As with many other methods, both FW and concave programming algorithms may have slow convergence in the viccinity of a local or global solution. Therefore, again based on the fact that we are only interested in zeros of the functional, an important part of the paper is dedicated to a thorough description of the practical implementation of algorithms, including initialization, feasible descent directions and stopping criteria to avoid slow final convergence. A special emphasis is put on developing accurate and non-conservative stopping criteria that do not require modification of the LMI characterization of the problem but use perturbation techniques on the *non-convex* variables (that are responsible of the nonconvexity/hardness of the problem). A key idea of these stopping criteria is to limit as far as possible the *zigzagging* phenomenon which characterizes first-order descent methods such as the FW algorithm or to reduce the computational burden in global search and hence to ensure reasonable computational time.

This description is followed by a set of numerical experiments for a realistic and randomized robust control problems. Interestingly enough, in almost all of our computational experiments, the local solutions found by FW algorithms are very close to optimality and are either certified global or quickly improved to optimality after a few iterations of the simplicial and conical techniques.

The remainder of the paper is organized as follows. A description of the robust control problem, its solvability conditions, motivations and difficulties are given in Section 2. Section 3 focuses on deriving a new formulation of the robust control problem as a concave minimization program where the constraints consist of LMIs. This section starts with a general result for converting BMI problems into rank constrained LMI problems. In turn, rank constrained LMI problems are shown to be equivalent to generalized concave programs where the usual linear vector inequalities are replaced with inequalities over the cone of positive semidefinite matrices. Extensions of the technique to other classes of scalings and problems are also discussed. A detailed presentation of a Frank and Wolfe feasible direction algorithm for solving the concave program is given in Section 4. Various stopping tests based on simple perturbation techniques of the scaling are derived in order to maintain reasonable computational cost. Section 5 is devoted to global concave optimization algorithms. More precisely, we generalize simplicial and conical concave minimization techniques over the positive semidefinite cone of symmetric matrices and provide their formulation for the robust control problem under consideration. A special emphasis is placed on convergence and implementation issues. This discussion has a general value and can be applied with minor modifications to a wide class of problems. Computational experiments are conducted in Section 6.

The following definitions and notations are used throughout the paper. M^T is the transpose of the matrix M, and M^* denotes its complex-conjugate transpose. The notation $\mathrm{Tr}\, M$ stands for the trace of M. For Hermitian or symmetric matrices, $M > N$ means that $M - N$ is positive definite and $M \geq N$ means that $M - N$ is positive semidefinite. The notation $\mathrm{co}\,\{p_1, \ldots, p_L\}$ stands for the convex hull of the set $\{p_1, \ldots, p_L\}$. The notation $\mathsf{vert}\,(P)$ is used to denote the set of vertices of a polyhedron P. Simplices and cones are defined in the usual way. In symmetric block matrices or long matrix expressions, we use $\bigstar$ as an ellipsis for terms that are induced by symmetry, e.g.,

$$\bigstar \begin{bmatrix} S & M \\ \bigstar & Q \end{bmatrix} K \equiv K^T \begin{bmatrix} S & M \\ M^T & Q \end{bmatrix} K\,.$$

We shall also use $\nabla f(x)$ to denote the gradient of the function f. Finally, in algorithm descriptions the notation X^k is used to designate the k-th iterate of the variable X. The notations $\mathrm{int}\, S$ and ∂S are used for the relative interior and the boundary of the set S.

2 Problem presentation and motivations

This section provides a brief review of a basic result that will be exploited throughout the paper. We are concerned with the robust control problem of an uncertain plant subject to LFT uncertainty. In other words, the uncertain plant is described as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ z_\Delta \\ z \\ y \end{bmatrix} &= \begin{bmatrix} A & B_\Delta & B_1 & B_2 \\ C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2} \\ C_1 & D_{1\Delta} & D_{11} & D_{12} \\ C_2 & D_{2\Delta} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w_\Delta \\ w \\ u \end{bmatrix} \\ w_\Delta &= \Delta(t)\, z_\Delta\,, \end{aligned} \tag{2}$$

where $\Delta(t)$ is a time-varying matrix-valued parameter and is usually assumed to have a block-diagonal structure in the form

$$\Delta(t) = \mathrm{diag}\,(\ldots, \delta_i(t) I, \ldots, \Delta_j(t), \ldots) \in \mathbf{R}^{N \times N} \tag{3}$$

and normalized such that

$$\Delta(t)^T \Delta(t) \leq I, \qquad t \geq 0\,. \tag{4}$$

Blocks denoted $\delta_i I$ and Δ_j are generally referred to as repeated-scalar and full blocks according to the μ analysis and synthesis literature [5, 4]. Note that straightforward computations lead to the state-space representation

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \left\{ \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} + \begin{bmatrix} B_\Delta \\ D_{1\Delta} \\ D_{2\Delta} \end{bmatrix} \Delta(t)(I - D_{\Delta\Delta}\Delta(t))^{-1} \left[C_\Delta \quad D_{\Delta 1} \quad D_{\Delta 2} \right] \right\} \begin{bmatrix} x \\ w \\ u \end{bmatrix},$$

hence the plant with inputs w and u and outputs z and y has state-space data entries which are fractional functions of the time-varying parameter $\Delta(t)$. Hereafter, we are using the following notation: u for the control signal, w for exogenous inputs, z for controlled or performance variables and y for the measurement signal.

For the uncertain plant (2)-(4) the robust control problem consists in seeking a Linear Time-Invariant (LTI) controller

$$\begin{aligned} \dot{x}_K &= A_K x_K + B_K y\,, \\ u &= C_K x_K + D_K y\,, \end{aligned} \tag{5}$$

such that

- the closed-loop system (2)-(4) and (5) is internally stable,
- the L_2-induced gain of the operator connecting w to z is bounded by γ,

for all parameter trajectories $\Delta(t)$ defined by (4).

It is now well-known that such problems can be handled via a suitable generalization of the Bounded Real Lemma which expresses as the existence of a Lyapunov matrix $X_{c\ell}$ and scalings S and T with adequate structure such that $X_{c\ell} > 0$ and

$$\begin{bmatrix} A_{c\ell}^T X_{c\ell} + X_{c\ell} A_{c\ell} & \star & \star \\ B_{c\ell}^T X_{c\ell} + \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix} C_{c\ell} & -\begin{bmatrix} S & 0 \\ 0 & \gamma I \end{bmatrix} + \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix} D_{c\ell} + D_{c\ell}^T \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}^T & \star \\ C_{c\ell} & D_{c\ell} & -\begin{bmatrix} S & 0 \\ 0 & \gamma I \end{bmatrix}^{-1} \end{bmatrix} < 0,$$

where the state-space data $A_{c\ell}$, $B_{c\ell}$, $C_{c\ell}$ and $D_{c\ell}$ determine the closed-loop system (2)-(5) with the loop $w_\Delta = \Delta(t)\, z_\Delta$ open. The following LMI characterizations for the solvability of such problems is then obtained. The reader is referred to references [23, 22, 1, 2, 14, 28] for more details and additional results.

The characterization of the solutions to the robust control problem for LFT plants requires the definitions of scaling sets compatible with the parameter structure given in (3). Denoting this structure as $\boldsymbol{\Delta}$, the following scaling sets can be introduced. The set of symmetric scalings associated with the parameter structure $\boldsymbol{\Delta}$ is defined as

$$S_{\boldsymbol{\Delta}} := \left\{ S : \, S^T = S, \quad S\Delta = \Delta S, \quad \forall \Delta \text{ with structure } \boldsymbol{\Delta} \right\}.$$

Similarly, the set of skew-symmetric scalings associated with the parameter structure $\boldsymbol{\Delta}$ is defined as

$$T_{\boldsymbol{\Delta}} := \left\{ T : \, T^T = -T, \quad T\Delta = \Delta^T T, \quad \forall \Delta \text{ with structure } \boldsymbol{\Delta} \right\}.$$

Equivalently, it is easily verified that with $S > 0$, the uncertain matrix Δ satisfies the quadratic constraints

$$\begin{bmatrix} \Delta \\ I \end{bmatrix}^T \begin{bmatrix} -S & T^T \\ T & S \end{bmatrix} \begin{bmatrix} \Delta \\ I \end{bmatrix} \geq 0, \quad \forall \Delta \text{ s. t. } \Delta^T \Delta \leq I, \quad \text{with structure } \boldsymbol{\Delta}. \tag{6}$$

With the above definitions and notations in mind, the following algebraically constrained LMI characterization for the solvability of the problem can be established.

Theorem 2.1 *Consider the LFT plant governed by (2) and (4) with Δ assuming a block-diagonal structure as in (3). Let $\mathcal{N}_X$ and $\mathcal{N}_Y$ denote any bases of the null spaces of $[C_2, D_{2\Delta}, D_{21}, 0]$ and $[B_2^T, D_{\Delta 2}^T, D_{12}^T, 0]$, respectively. Then, there exists a controller such that the (scaled) Bounded Real Lemma conditions hold for some L_2 gain performance γ if and only if there exist pairs of symmetric matrices (X, Y), (S, Σ) and a pair of skew-symmetric matrices (T, Γ) such that the structural constraints*

$$S,\ \Sigma \in S_{\boldsymbol{\Delta}} \text{ and } T,\ \Gamma \in T_{\boldsymbol{\Delta}} \tag{7}$$

hold and the matrix inequalities

$$\textbf{LMI}\,[1]\ :\ \star \begin{bmatrix} A^TX + XA & XB_\Delta + C_\Delta^T T^T & XB_1 & C_\Delta^T S & C_1^T \\ B_\Delta^T X + TC_\Delta & -S + TD_{\Delta\Delta} + D_{\Delta\Delta}^T T^T & TD_{\Delta 1} & D_{\Delta\Delta}^T S & D_{1\Delta}^T \\ B_1^T X & D_{\Delta 1}^T T^T & -\gamma I & D_{\Delta 1}^T S & D_{11}^T \\ SC_\Delta & SD_{\Delta\Delta} & SD_{\Delta 1} & -S & 0 \\ C_1 & D_{1\Delta} & D_{11} & 0 & -\gamma I \end{bmatrix} \mathcal{N}_X \ < \ 0, \tag{8}$$

$$\textbf{LMI}\,[2]\ :\ \star \begin{bmatrix} AY + YA^T & YC_\Delta^T + B_\Delta \Gamma^T & YC_1^T & B_\Delta \Sigma & B_1 \\ C_\Delta Y + \Gamma B_\Delta^T & -\Sigma + \Gamma D_{\Delta\Delta}^T + D_{\Delta\Delta}\Gamma^T & \Gamma D_{1\Delta}^T & D_{\Delta\Delta}\Sigma & D_{\Delta 1} \\ C_1 Y & D_{1\Delta}\Gamma^T & -\gamma I & D_{1\Delta}\Sigma & D_{11} \\ \Sigma B_\Delta^T & \Sigma D_{\Delta\Delta}^T & \Sigma D_{1\Delta}^T & -\Sigma & 0 \\ B_1^T & D_{\Delta 1}^T & D_{11}^T & 0 & -\gamma I \end{bmatrix} \mathcal{N}_Y \ < \ 0, \tag{9}$$

$$\textbf{LMI}\,[3]\ :\ -\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \ < \ 0, \tag{10}$$

$$\textbf{LMI}\,[4]\ :\ -\begin{bmatrix} S & 0 \\ 0 & \Sigma \end{bmatrix} \ < \ 0 \tag{11}$$

subject to the algebraic constraints

$$(S + T)^{-1} = (\Sigma + \Gamma), \tag{12}$$

or equivalently,

$$\begin{bmatrix} S & T \\ T^T & -S \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma & \Gamma^T \\ \Gamma & -\Sigma \end{bmatrix}, \tag{13}$$

are feasible.

Note that due to the algebraic constraints (12), the problem under consideration is non-convex and has been even shown to be NP-hard. See [3] and references therein. This feature is in sharp contrast with the associated Linear Parameter-Varying control problem for which the LMI constraints (8)-(11) are the same but the nonlinear conditions (12) or alternatively (13) fully disappears.

3 Rank constraints, BMIs and concave programs

For tractability reasons, it is interesting to find alternate formulations that are amenable to numerical computations. A potential technique was introduced in [11] and amounts to constructing a nonlinear functional whose feasible optimal points satisfy the algebraic constraints (12). Hereafter, we develop different extensions of this technique that is applicable to structured μ-scalings (6), to full-block generalized scalings as considered in [26] but also more importantly to bilinearly constrained LMI problems. We begin the presentation by a more general result which reveals the close connections between BMIs, rank constrained LMI problems and concave programming.

Lemma 3.1 (Rank formulation) *Introduce the bilinear constrained LMI problem*

$$\begin{aligned} \mathcal{L}(x) &< 0 &\quad (14)\\ W(x) &= L(x)CR(x) &\quad (15) \end{aligned}$$

where x denotes the vector of decision variables, the inequality (14) is a general LMI constraint and $L(x)$ and $R(x)$ are matrix-valued functions of x. The matrix C is constant and assumes a minimal rank factorization of rank r, that is $C = UV^T$ where the column dimension of U is r. Then, the feasibility problem (14)-(15) is equivalent to

$$\begin{aligned} \mathcal{L}(x) &< 0 &\quad (16)\\ \operatorname{Rank}\begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I \end{bmatrix} &= r\,. &\quad (17) \end{aligned}$$

Proof: The equivalence follows from the rank-perserving transformations

$$\begin{bmatrix} I & -L(x)U \\ 0 & I_r \end{bmatrix} \begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -V^T R(x) & I_r \end{bmatrix} = \begin{bmatrix} W(x) - L(x)CR(x) & 0 \\ 0 & I_r \end{bmatrix}.$$

■

Lemma 3.1 has important algorithmic consequences that we examine in the sequel. It also provides a direct link between BMI problems and rank constrained LMI problems. A more easily implementable form of Lemma 3.1 is as follows.

Lemma 3.2 (Concave representation) *With the notations of Lemma 3.1, we assume without loss of generality that $W(x) \in \mathbf{R}^{l\times c}$ with $l \geq c$. The bilinearly constrained LMI problem (14)-(15) is equivalent to the existence of a symmetric (slack) matrix Z and x such that $\mathcal{L}(x) < 0$ and*

$$Z := \begin{bmatrix} Z_1 & Z_3 \\ Z_3^T & Z_2 \end{bmatrix}, \quad \begin{bmatrix} Z_1 & Z_3 & W(x)^T & R(x)^T V \\ Z_3^T & Z_2 & U^T L(x)^T & I \\ W(x) & L(x)U & I & 0 \\ V^T R(x) & I & 0 & I \end{bmatrix} \geq 0, \qquad (18)$$

with the additional Schur complement constraint,

$$\operatorname{Tr}(Z_1 - Z_3 Z_2^{-1} Z_3^T) = 0\,. \qquad (19)$$

Moreover, the trace function in (19) is concave over the cone of positive semidefinite matrices and is bounded below by zero.

Proof: Necessity is trivial and follows from the choice

$$\begin{bmatrix} Z_1 & Z_3 \\ Z_3^T & Z_2 \end{bmatrix} = \begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I \end{bmatrix}^T \begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I \end{bmatrix}$$

for which (18) and (19) hold.

Sufficiency: It follows from (18) and (19) that Z has a loss of rank of dimension c and that Z_2 is invertible. Select a basis $\mathcal{N} \in \mathbf{R}^{(c+r)\times c}$ of the nullspace of Z. We infer by a Schur complement argument with respect to the identity term in the inequality (18) that

$$\begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I_r \end{bmatrix} \mathcal{N} = 0\,.$$

But since $\mathcal{N}$ is a full rank matrix, we deduce that

$$\text{Rank} \begin{bmatrix} W(x) & L(x)U \\ V^T R(x) & I_r \end{bmatrix} = r\,.$$

It then follows from Lemma 3.1 that (15) holds.

The concavity of the trace function in (19) is easily seen by looking at its hypograph. Using (22) one can rewrite the inequality

$$\text{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) \geq t\,,$$

in the form

$$\begin{bmatrix} Z_1 - P & Z_3 \\ Z_3^T & Z_2 \end{bmatrix} \geq 0, \qquad \text{Tr}\,(P) \geq t\,,$$

which defines a convex set. Also, we deduce that the trace function is bounded below by zero for any Z such that (18) holds. This terminates the proof. ■

Note that when $W(x)$, $L(x)$ and $R(x)$ are affine functions of x, which is the case of interest, then inequality (18) reduces to an LMI and thus the new formulation is a concave program where usual vector linear inequalities are replaced by inequalities over the positive semidefinite cone.

One important consequence of Lemma 3.2 is that BMI problems can be equivalently formulated as the search of zero optimal solutions of concave programs. These problems however exhibit a high degree of nonconvex dimensionality and consequently are generally harder to solve than the problems investigated in this paper. Important advantages lie in the simplicity of this new formulation but also in the fact that matrix structures are preserved in the concave program. This is an important factor for efficient implementation of algorithms that we shall consider in the sequel. Because of the special properties of concave programs, it is possible to develop algorithms local or global which take advantage of the problem properties to enhance efficiency. A fairly extensive discussion of concave programs is provided in Sections 4 and 5. Before going further, we must point out that feasibility problems involving LMIs and rank inequalities can be handled in the same fashion. This is achieved by remarking that

$$\text{Rank}\, W(x) \leq k$$

is equivalent to the existence of a (slack) matrix $U \in \mathbf{R}^{l\times k}$

$$W(x)^T W(x) - UU^T \geq 0, \qquad \text{Tr}\,(W(x)^T W(x) - UU^T) = 0.$$

Then one can linearize the terms W^TW using Lemma 3.2 while the term UU^T is linearized using Schur complements. Similarly, when $W(x)$ and $L(x)$ are affine matrix-valued functions of x, the inversion constraint

$$W(x) = L(x)^{-1},$$

can be given the concave programming representation

$$\begin{bmatrix} Z_1 & Z_3 & W(x) & I \\ Z_3^T & Z_2 & I & L(x) \\ W(x)^T & I & I & 0 \\ I & L(x)^T & 0 & I \end{bmatrix} \geq 0, \quad \mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) = 0. \tag{20}$$

This is readily obtained from Lemma 3.2 by noting that

$$\mathrm{Rank} \begin{bmatrix} W(x) & I \\ I & L(x) \end{bmatrix} = \mathrm{Rank} \begin{bmatrix} W(x) & L(x)^{-1} \\ I & I \end{bmatrix}.$$

The outcome of this discussion is that a non-exhaustive list of potential control applications of the proposed algorithms include also

- reduced- and fixed-order robust control,
- multi-objective robust and Linear Parameter-Varying control,
- reduction of LFT representations,

Moreover, since positive combinations of concave functions remain concave these problems can be aggregated in many different ways to formulate more complex practical requirements.

3.1 Concave representations of robust control problems

An immediate application of Lemma 3.2 leads to a concave programming formulation of the robust control problem introduced in Section 2 and characterized in Theorem 2.1.

Corollary 3.3 *Introduce the concave LMI-constrained minimization program*

$$\textbf{Pb1:} \qquad \textit{minimize}\ \mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) \tag{21}$$

subject to LMIs (8)-(11) and

$$\textbf{LMI}\,[5]\ :-\begin{bmatrix} Z_1 & Z_3 & S+T & I \\ Z_3^T & Z_2 & I & \Sigma+\Gamma \\ (S+T)^T & I & I & 0 \\ I & (\Sigma+\Gamma)^T & 0 & I \end{bmatrix} \leq 0. \tag{22}$$

Then, any feasible point to **Pb1** *which further satisfies*

$$\mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) = 0, \tag{23}$$

is optimal and is a solution to the problem described in Theorem 2.1 and conversely.

Proof: The result follows from the inversion form of Lemma 3.2 in (20) and the fact that S and Σ are invertible. ■

Note that without loss of generality, it can be assumed that the matrix

$$Z := \begin{bmatrix} Z_1 & Z_3 \\ Z_3^T & Z_2 \end{bmatrix}$$

has a structure conformable with that of the particular block-diagonal structure of the scalings. This simple observation reduces the number of "nonconvex variables" and avoids a wasteful search in an unduly large space. The number of nonconvex variables is also reduced when some subblocks T_i and Γ_i in the skew-symmetric matrices T and Γ vanish. This is the case when the corresponding Δ_i in Δ is scalar or is considered as a complex block. In such case, one can remove this block from both LMI (22) and the objective functional (21). The sizes of Z_1, Z_2 and Z_3 are then reduced accordingly and the (concave) objective functional becomes

$$\mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) + \mathrm{Tr}\,(S_i - \Sigma_i^{-1}) \tag{24}$$

with the additional LMI

$$\begin{bmatrix} S_i & I \\ I & \Sigma_i \end{bmatrix} \geq 0\,.$$

One advantage of the formulation of the problem as in Corollary 3.3 is that one completely gets rid of the hard set constraints (12) and the non-convexity is reflected in the functional to be optimized. It is also important to note that the approach considered in [11] is not directly applicable since the above rank constrained LMI problems cannot be reduced to standard *bilinear* or *cone complementary* problems for which specialized algorithms are already available. See [3] for a survey. This is easily verified on simple examples. A central target of this paper is to point out and discuss adequate algorithms for solving this class of problems. Before going into the details of the algorithm, we must stress out that the proposed concave reformulations apply with the same degree of simplicity to other classes of scalings such as the full block scalings introduced in [27] and also to dynamic scalings or multipliers hence providing a complete concave formulation of the μ synthesis problem.

4 A local search: Frank and Wolfe algorithm

In this section, we discuss a Frank and Wolfe algorithm for finding solutions to Corollary 3.3. Analogous algorithms can be derived in the context of any of the control problems mentioned previously. Such algorithms are of local nature in the sense that they cannot guarantee global optimality but have proven very efficient in practice [3, 11].

4.1 Basic principle

The basic principle of Frank and Wolfe (FW) algorithms is to determine a segment line in the feasible set pointing towards a "best" descent direction and then to perform a line search on this segment to minimize the cost function [6]. Consider the minimization problem

$$\text{minimize} \;\; f(Z) \;\; \text{subject to} \;\; Z \in \mathcal{X} \tag{25}$$

where the function f has continuous first-order partial derivatives on $\mathcal{X}$ and is bounded below on the matrix set $\mathcal{X}$, a convex subset of the space of symmetric matrices. The algorithm of Frank and Wolfe can be detailed as follows:

1. Find a steepest descent direction by solving the convex programming problem

$$D^k \in \arg\min_{D \in \mathcal{X}} \ \mathrm{Tr}\,(\nabla f(Z^k)\, D)$$

2. Perform a line search on the segment $[Z^k, D^k]$ to get

$$\begin{aligned} Z^{k+1} &= (1-\alpha^k)Z^k + \alpha^k D^k, \\ \text{where } \alpha^k &\in \arg\min_{0 \le \alpha \le 1} f((1-\alpha)Z^k + \alpha D^k) \end{aligned} \tag{26}$$

Under the above very mild assumptions Bennett and Mangasarian have proved in [3] that for a general differentiable f the algorithm terminates at a point that satisfies the minimum principle necessary optimality conditions, or each accumulation point of the generated sequence satisfies also the minimum principle. Hence, there is a risk of *cycling* or *jamming* with such algorithms though it turns out to be very low in practice. Interestingly, when f is moreover concave, the algorithm generates *strictly decreasing* sequences that can only terminate to a point satisfying the minimum principle local optimality conditions. This can be clarified as follows. Let Z^k denote the k-th iterate of the FW algorithm, then from the concavity of f, we have

$$f(Z) - f(Z^k) \le \mathrm{Tr}\,(\nabla f(Z^k)(Z - Z^k)), \quad \forall Z \in \mathcal{X}\,.$$

Since f is bounded from below on $\mathcal{X}$, we can write

$$-\infty < \inf_{Z \in \mathcal{X}} f(Z) - f(Z^k) \le \mathrm{Tr}\,(\nabla f(Z^k)(Z - Z^k))\,,$$

so that the FW step

$$\text{minimize} \ \ \mathrm{Tr}\,(\nabla f(Z^k)(Z - Z^k)) \text{ subject to } Z \in \mathcal{X}$$

is well defined and generate a new iterate Z^{k+1} in $\mathcal{X}$. We also infer $\mathrm{Tr}\,(\nabla f(Z^k)(Z^{k+1} - Z^k)) \le 0$, since Z^k is feasible. Thus, only two situations can occur. Either $\mathrm{Tr}\,(\nabla f(Z^k)(Z^{k+1} - Z^k)) < 0$ and consequently

$$f(Z^{k+1}) \le f(Z^k) + \mathrm{Tr}\,(\nabla f(Z^k)(Z^{k+1} - Z^k)) < f(Z^k)\,.$$

The sequence is therefore strictly decreasing from Z^k and can only stop when the second situation $\mathrm{Tr}\,(\nabla f(Z^k)(Z^{k+1} - Z^k)) = 0$ occurs. In such case, we obtain

$$\mathrm{Tr}\,(\nabla f(Z^k)(Z - Z^k)) \ge 0, \quad \forall Z \in \mathcal{X}\,,$$

which is nothing else than the minimum principle local optimality condition for symmetric matrices. To sum up, the sequence $f(Z^k)$ is strictly decreasing and Z^k is either infinite or reaches a local optimum.

Also for a general function, a line search on the matrix segment

$$\left[Z^k,\, D^k\right]\,,$$

will be required, where D^k is a solution of the FW step above. In virtue of the concavity of the objective function (21), as well as for all the functions introduced in Section 3, the line search can be completely bypassed and one can perform a *full step size of one*, hence reducing the overall computational overhead.

4.2 Implementation of FW algorithm for robust control

In this section, we reexamine the algorithm of Frank and Wolfe in the context of the robust control problem introduced in Section 2. In order to facilitate the presentation, we shall assume that the notation $\mathbf{LMI}[i]$, $i = 1, \ldots, 5$ is nothing else than the difference between the left-hand and the right-hand side of the corresponding LMI in (8)-(11) and (22), respectively.

4.2.1 Initialization

The initialization phase simply consists in determining a feasible point of the constraints. In order to favor large step sizes in the course of the algorithm and avoid stucking initially to the boundary of the constraints, it is advisable to perform a "centering step". It amounts to seeking an initial point that renders the LMIs (8)-(11) and (22) maximally negative. This is easily formulated as the LMI program

$$\text{minimize } \; t \text{ subject to } \mathbf{LMI}[i] \; < t, \qquad i = 1, \ldots, 5$$

We also mention that for all LMI runs used throughout, we put a norm constraint on the decision variables for preventing solutions at infinity. This is easily done with currently available LMI solvers.

4.2.2 Phase I - FW step

In this phase, we determine a feasible segment pointing towards a descent direction. Remarking that the gradients of

$$J = \operatorname{Tr}(Z_1 - Z_3 Z_2^{-1} Z_3^T),$$

at the k-th iterate are given as

$$G_1 := \frac{\partial J}{\partial Z_1} = I, \; G_2 := \frac{\partial J}{\partial Z_2} = Z_2^{k^{-1}} Z_3^{k^T} Z_3^k Z_2^{k^{-1}}, \; G_3 := \frac{\partial J}{\partial Z_3} = -2 Z_2^{k^{-1}} Z_3^{k^T},$$

the FW step can be described by the following LMI program:

$$\begin{array}{l} \text{minimize } \operatorname{Tr}(G_1 Z_1 + G_2 Z_2 + G_3 Z_3) \\ \text{subject to } \; \mathbf{LMI}[i] \; < 0, \; i = 1,2,3,4; \; \mathbf{LMI}[5] \; \leq 0. \end{array}$$

Note that this problem is always solvable, since we are only manipulating feasible points and directions.

4.2.3 Stopping criteria

Given the current point of the algorithm determined by the variables (X^k, Y^k), (S^k, T^k), (Σ^k, Γ^k), Z_1^k, Z_2^k and Z_3^k our goal is to verify whether this point or a closely related point is a solution to the LMIs (8)-(11) subject to the algebraic constraint (12). In such case the algorithm will terminate and a controller solution to the problem in Section 2 can be constructed, avoiding long sequences of iterates. In our new notation, our test takes the form

$$\mathbf{LMI}[i] \; < 0, \qquad i = 1,2,3,4 \tag{27}$$

$$(S^k + T^k)^{-1} = (\Sigma^k + \Gamma^k). \tag{28}$$

Note that in the course of the algorithm, the current point is not generally optimal so that the constraint (12) does not hold. It is, however, possible to terminate the program without reaching optimality. Our stopping criteria are based on the following perturbations techniques. We assume that a current feasible point of LMIs (8)-(11) and (22) is given. There exists a controller for which the conditions in Theorem 2.1 hold whenever one of the following perturbation techniques is successful.

- Compute $W = (S^k + T^k)^{-1}$ and update Σ^k and Γ^k using the substitutions

$$\widetilde{\Sigma}^k := \frac{W + W^T}{2}, \qquad \widetilde{\Gamma}^k := \frac{W - W^T}{2}. \tag{29}$$

  Then, stop if new point passes the test (27).

- If previous test fails, then compute $W = (\Sigma^k + \Gamma^k)^{-1}$ and update S^k and T^k using the substitutions

$$\widetilde{S}^k := \frac{W + W^T}{2}, \qquad \widetilde{T}^k := \frac{W - W^T}{2}. \tag{30}$$

  Then, stop if new point passes the test (27).

Note that since we do not alter the original characterization of the solutions in Theorem 2.1, our stopping criteria are generally less conservative than those in [11] which necessitate a modification of the problem.

5 Global concave programming based methods

Concave programming constitutes a class of well-developed methods in global optimization whose foundations were mostly layed in [31]. It offers a wealth of practically efficient techniques for solving difficult problems which seem, however, to have been overlooked by the control community. Reasons for this disinterest lie in the fact that most successfully developed concave programming algorithms [15, 16, 20, 32] deal with (linear) polytopic constraints, thus having a finite number of extreme points, and are restricted to the usual vector space $\mathbf{R}^n$ which could be an obstacle for applicability of these methods to robust control problems. In this section, we shall show that several important basic concepts of concave programming carry over matrix spaces and the positive semidefinite cone of symmetric matrices and that these generalizations can be exploited to handle our problems. The discussion here is deliberately very short and avoids the abstract convergence theory that can be found in textbooks. The reader is referred to the recent book of Tuy [32] for further details on concave programming.

Return to the problem of checking whether there exists

$$Z^* \in \mathcal{X} = \{(Z_1, Z_2, Z_3) : \exists (X, Y, S, T, \Sigma, \Gamma) \text{ s.t. } \mathbf{LMI}[i] < 0,\ i = 1, \ldots, 4;\ \mathbf{LMI}[5] \le 0\} \tag{31}$$

satisfying $f(Z^*) = 0$ where $f(Z) := \mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T)$ is concave. Such a Z^* when it exists will be called a *zero* of f. It is important to note that since f satisfies $f(Z) \ge 0, \forall Z \in \mathcal{X}$, any zero of f is also a global optimal solution of (25), and consequently, our problem is much more computationally attractive than conventional concave programs in which minimal values of the cost function are unknown. In the methods presented hereafter, we can stop the search as soon as either such a zero is found in which case global optimality is ensured, or the minimum cost value is strictly positive in which case our problem has no solution.

In view of the recent developments in global optimization, it seems that a BB method is the most suitable for our global search. Our intention in the present work is to maximally exploit the structure and properties of the problem to make our search algorithms much more efficient than general BB schemes. The overall scheme goes as follows.

Branching: The function f is not only concave in (Z_1, Z_2, Z_3) but is also linear in Z_1 with (Z_2, Z_3) held fixed, i.e. only (Z_2, Z_3) are the "complicating" variables, responsible for the nonconvexity/hardness of the problem. The global search thus is concentrated on the reduced-dimensional space $\mathcal{Z}$ of variables (Z_2, Z_3). Accordingly, the feasible set can be interpreted as the projection of the convex set defined by the LMIs (8)-(11) and (22) on the space $\mathcal{Z}$. This space is partitioned into finitely many matrix polyhedrons of the same kind (simplices, cones etc.). At each iteration, a partition polyhedron M is selected and subdivided further into several subpolyhedrons according to a specified rule.

Bounding: With the branching strategy determined and given a partition set M, the convexity of $\mathcal{X}$, the concavity of f and its linearity in Z_1 are further exploited in the search of a zero of f over $(Z_2, Z_3) \in M$. This is carried out through computing a number $\beta(M)$ by a convex program such that

$$\beta(M) \leq \nu(M) := \inf\{f(Z_1, Z_2, Z_3) : \ (Z_1, Z_2, Z_3) \in \mathcal{X}, (Z_2, Z_3) \in M\}. \tag{32}$$

Clearly, the partition sets M with $\beta(M) > 0$ cannot contain any zero of f and therefore are discarded from further consideration. On the other hand, the partition set with smallest $\beta(M) < 0$ can be considered the most promising one. To concentrate further investigation on this set, we subdivide it into more refined subsets. With a given tolerance $\varepsilon > 0$, the stop criterion of the BB algorithm is

$$\min_{M} \beta(M) \geq \varepsilon. \tag{33}$$

Stopping rule: The branching operation is devised for speeding up the convergence. The optimal solution $Z(M)$ of the problem for computing $\beta(M)$ is used for the stopping test developed above to reduce the time of global search.

Based on the kind of polyhedrons which are used in branching, we develop 2 different BB algorithms. As one may see, each of them has its own advantage depending on the more specific structure of the objective $f(Z)$. It is important to mention that all branching and bounding operations must be developed consistently to secure global convergence of the search to a global solution. Global convergence is often a delicate issue in BB techniques. Proofs are provided in Appendix A.

5.1 Simplicial algorithm

In the simplicial algorithm, the space $\mathcal{Z}$ is partitioned into simplices. From now on, N will denote the dimension of $\mathcal{Z}$. For every simplex M with vertices $u^1, u^2, ..., u^{N+1}$ in $\mathcal{Z}$, the affine function $\phi_M(Z)$ defined for every Z_1 and $x = (Z_2, Z_3) = \sum_{i=1}^{N+1} \lambda_i u^i, \ \lambda_i \geq 0, \ \sum_{i=1}^{N+1} \lambda_i = 1$ by

$$\phi_M(Z_1, x) := \mathrm{Tr}\,(Z_1) + \phi_M(\sum_{i=1}^{N+1} \lambda_i u^i) = \mathrm{Tr}\,(Z_1) + \sum_{i=1}^{N+1} \lambda_i f(0, u^i),$$

satisfies $\phi_M(Z_1, x) = f(Z_1, x)\ \forall x \in \mathrm{vert} M$, and any Z_1 and thus $\phi_M(Z_1, x) \leq f(x)\ \forall x \in M, Z_1$, i.e. $\phi_M(Z)$ is an affine minorant of f in M (in fact the convex envelope of $f(Z)$ over M). On the other hand, if there is a zero (Z_1, Z_2, Z_3) with $(Z_2, Z_3) \in M$ then again by the concavity of f, one must have

$$\min_{i=1,2,...,N+1} f(Z_1, u^i) \leq 0 \Leftrightarrow \mathrm{Tr}\,(Z_1) + \min_{i=1,2,...,N+1} f(0, u^i) \leq 0\,. \tag{34}$$

Thus a lower bound $\beta(M)$ satisfying (32) is defined by the convex (LMI) program

$$\beta(M) := \min\{\phi_M(Z_1, \sum_{i=1}^{N+1} \lambda_i u^i) : \ (34), \ \sum_{i=1}^{N+1} \lambda_i = 1, \ \lambda_i \geq 0, \ (Z_1, \sum_{i=1}^{N+1} \lambda_i u^i) \in \mathcal{X}\} \tag{35}$$

Keeping in mind that the algorithm will stop when the current best value is 0 or there is evidence that the lower bound of (25) is positive (infeasibility), we can state the simplicial algorithm as follows. The proof of global convergence of this algorithm is deferred to appendix A.

Step 0. (*Initialization*) In the $\mathcal{Z}$-space take an N-simplex M_0 large enough such that f is still concave for $(Z_2, Z_3) \in M_0$ as the initial simplex. Let Z^0 be an initial feasible point (the best available), $\nu_0 = f(Z^0)$, $\mathcal{S}_0 = \{M_0\}$, $\mathcal{P}_0 = \mathcal{S}_0$, $k = 0$.

Step 1. (*Bounding*) For each simplex $M = [u^1, \ldots, u^{N+1}] \in \mathcal{P}_k$ compute $\beta(M)$ by (35) and let $Z_1(M)$ and $\omega(M) = \sum_{i=1}^{N+1} \lambda_i(M) u^i$ be an optimal solution of this (convex) LMI program.

Step 2. (*Incumbent*) Let Z^k be the best among: Z^{k-1} and all $(Z_1(M), \omega(M))$ for $M \in \mathcal{P}_k$. Let $\nu_k = f(Z^k)$. If $\nu_k = 0$, then terminate (a zero has been found). Otherwise, $\nu_k > 0$ (since $f(Z^k) \geq 0$, $\forall k$), then go to Step 3.

Step 3. (*Pruning*) Delete every simplex $M \in \mathcal{S}_k$ such that $\beta(M) > 0$ (this means that f cannot attain zero with $(Z_2, Z_3) \in M \cap \mathcal{X}$). Let $\mathcal{R}_k$ be the collection of remaining members of $\mathcal{S}_k$.

Step 4. (*Termination criterion*) If $\mathcal{R}_k = \emptyset$, then terminate: there is no zero of $f(Z)$ in $\mathcal{X}$.

Step 5. (*Branching*) Select $M_k \in \text{argmin}\{\beta(M) |\ M \in \mathcal{R}_k\}$. Subdivide M_k according to a chosen normal rule described the Appendix A. Let $\mathcal{P}_{k+1}$ be the partition of M_k.

Step 6. (*New net*) Set $\mathcal{S}_{k+1} = (\mathcal{R}_k \setminus \{M_k\}) \cup \mathcal{P}_{k+1}$, $k \leftarrow k+1$ and return to Step 1.

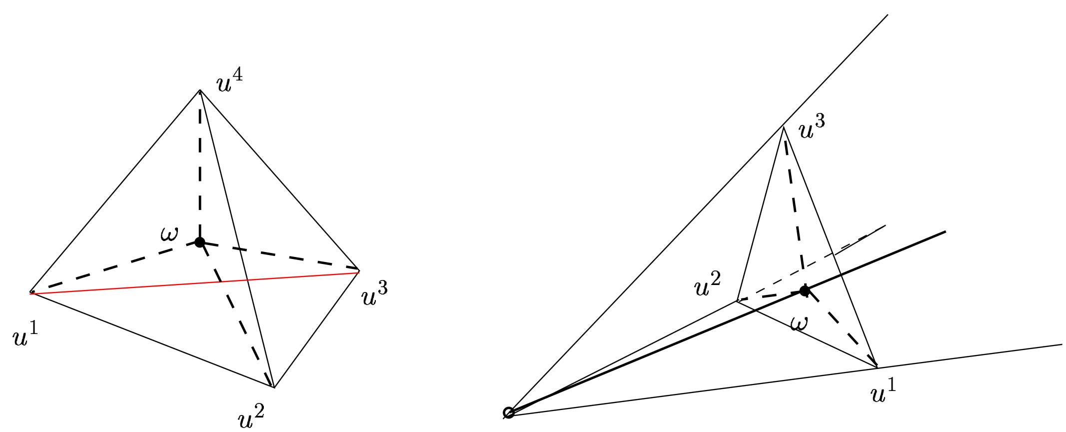

Figure 1: Simplicial and conical ω-subdivisions

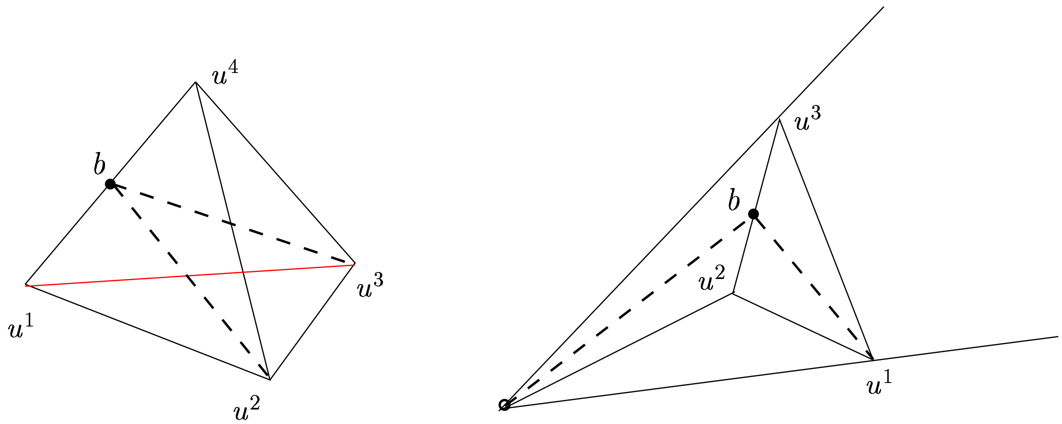

Figure 2: Simplicial and conical bisections

5.2 Conical algorithm

Close scrutiny of the objective function properties ($\mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T)$) in Corollary 3.3 reveals the following.

(i) If (Z_1, Z_2, Z_3) is the solution of **Pb1** with the zero optimal value then (tZ_1, tZ_2, tZ_3) with $t \geq 1$ is also a solution satisfying the same conditions. Thus, without loss of generality, we can set $\mathrm{Tr}\,(Z_1) = L$, with L a constant large enough.

(ii) $Z_2 \geq I$ which means that we can use the change of variable $Z_2 \to Z_2 + \varepsilon I$ with $Z_2 \geq 0$ instead of $Z_2 > 0$.

As a consequence, problem **Pb1** can be reduced to minimizing the objective function

$$f(Z_2, Z_3) = L - \mathrm{Tr}\,(Z_3(Z_2 + \varepsilon I)^{-1} Z_3^T) \tag{36}$$

and LMIs (8)-(22) are changed accordingly using the substitution $Z_2 \to Z_2 + \varepsilon I$. The function f in (36) is concave in the cone $\mathcal{C}_+^{m_2} \times \mathcal{C}^{m_3}$ where $\mathcal{C}_+^{m_2}$ is the cone of nonnegative definite matrices with the same structure as Z_2 and $\mathcal{C}^{m_3}$ is the space of symmetric matrices having the same structure as Z_3. It is sufficent to take $\bar{\mathcal{Z}}$ as a large enough finite family of canonical cones approximating $\mathcal{C}_+^{m_2} \times \mathcal{C}^{m_3}$ with some tolerance. Perhaps, the most essential property of a concave function f is that its level sets $C_0 = \{Z = (Z_2, Z_3) \in \bar{\mathcal{Z}} : \ f(Z) \geq 0\}$ are convex and therefore an alternative formulation of our problem is to find $Z \in \mathcal{X} \setminus \mathrm{int}\, C_0$ or else prove that $\mathcal{X} \subset \mathrm{int}\, C_0$, where both $\mathcal{X}, C_0$ are convex sets. All these facts are taken into account in the following global search which uses the so-called concavity cut or Tuy's cut [31].

In what follows, by a cone we mean a cone with vertex at 0 and exactly N edges. Consider an initial family $\mathcal{P}_0$ of cones covering $\bar{\mathcal{Z}}$ and with pairwise disjoint interiors. For each initial cone in $\mathcal{P}_0$ take a fixed hyperplane cutting all its edges. Then, the intersection of each subcone of this initial cone with the above mentioned hyperplane is a simplex with N vertices and is called the base of the subcone. Let M be a cone with base $[u^1, u^2, ..., u^N]$. Since $f(0) = L > 0$, we have $0 \in \mathrm{int}\, C_0$ and by the convexity of C_0 the ray from 0 through u^i meets the boundary of C_0 at a unique point $\bar{u}^i = \theta_i u^i$ with $\theta_i > 0$ determined by

$$\theta_i = \sup\{\theta > 0 : \ f(\theta u^i) \geq 0\}. \tag{37}$$

Then, since the convex set C_0 is closed, $\bar{u}^i \in C_0$ and

$$\text{co}\{0, \bar{u}^i,\ i = 1, 2, ..., N\} \subset C_0. \tag{38}$$

Consider then the convex (LMI) program

$$\max\{\sum_{i=1}^{N} \lambda_i :\ \lambda_i \geq 0,\ \sum_{i=1}^{N} \lambda_i \bar{u}^i \in \mathcal{X}\}, \tag{39}$$

and let $\mu(M)$ and $\lambda(M)$ be the optimal value and the optimal solution of this program. Also let $\omega(M) = \sum_{i=1}^{N} \lambda_i(M)\bar{u}^i \in \mathcal{X}$. Only one of the following mutually exclusive possibilities can occur:

(i) $\mu(M) < 1$. Then it easily follows that $M \cap \mathcal{X} \subset \text{int}\, C_0$, i.e. there is no zero optimal solution in $M \cap \mathcal{X}$ and so M can be discarded from further consideration;

(ii) $\omega(M) \in \mathcal{X} \setminus \text{int}\, C_0$ (i.e. $f(\omega(M)) = 0$): then we have obtained a zero optimal solution;

(iii) $\mu(M) \geq 1$ and $\omega(M) \in C_0$. In this case $\omega(M)$ does not lie on any edge of M (so that the subdivision of M by the ray through $\omega(M)$ is possible). Indeed, if $\omega(M)$ lies on some edge u^i of M then we must have $\omega(M) = \mu(M)\bar{u}^i = \mu(M)\theta_i u^i$ with $\mu(M)\theta_i > \theta_i$ and $f(\mu(M)\theta_i u^i) = f(\omega(M)) \geq 0$, which contradicts the definition (37) for θ_i.

Actually, $\mu(M)$ is not a lower bound for $f(x)$ on $M \cap \mathcal{X}$ but because of the above property, $1 - \mu(M)$ plays essentially the same role as a lower bound for eliminating portions of the constraint set. Therefore, using a partition of the cone via the ray through a point in its simplex base defined according to the normal subdivision rule, the conical algorithm can be described (see Figures 1 and 2). Its global convergence can be shown similarly to that of the simplicial algorithm, and is omitted for brevity .

5.3 Trade-off of two global searches

Let us briefly mention the relative advantages of each of these two global algorithms. Clearly, by concentrating the search on the boundary of the feasible set, the conical algorithm better exploits the fact that the global minimum is attained at a boundary point and is therefore more efficient than the simplicial algorithm in the case of problem **Pb.1**.

However, the simplicial algorithm is convenient for exploiting the partial linearity of the objective. For instance, in the case when all skew-symmetric matrices T and Γ vanish, the objective for (24) can be reduced to the form

$$\text{Tr}\,(S) - \text{Tr}\,(\Sigma^{-1}), \tag{40}$$

which means that it is concave in Σ and linear in S. Since the optimal solution may now project to an interior point of $\mathcal{X}$, the conical algorithm would require preliminary transformations of the problem by the introduction of one extra variable, whereas the simplicical algorithm can be applied directly, with branching operations on the $\Sigma-$space as previously. Thus in this case, the simplicial algorithm might be preferred.

6 Numerical experiments

This section provides a set of illustrations of the local and global techniques proposed in the paper. As mentioned in the introduction, the overall algorithm can be detailed as follows. The FW algorithm is computationally cheaper than simplicial and conical global techniques, and hence is used first to find a good suboptimal value γ. Then, the simplicial/conical algorithm are employed to further reduce γ, or to certify global optimality. As discussed hereafter, in realistic and randomly generated examples, the FW algorithm is able to locate a suboptimal solution, up to 8% of the global optimal value, after only a few iterations. The simplicial/conical algorithms starting from this good initial guess find a global optimal solution very quickly, less than 5 iterations when the problem is feasible. For infeasible problems, they obtain a positive lower bound of **Pb.1** after less than 10 iterations. It is also important to emphasize that for feasible γ, the use of the stopping criteria in Section 4.2.3 substantially reduces the computational cost since only an approximate solution is required for termination. This fact and the power of simplicial/conical techniques explains why so few iterations (LMI runs) are needed and thus the relatively cheap cost of our global algorithms.

6.1 Robust control of an inverted pendulum

The first illustration consists of the robust control problem of an arm-driven inverted pendulum (ADIP) which is depicted in Figure 3. This is a two-link system comprising an actuated arm (first link) and a non-actuated pendulum (second link). The main control objective is to maintain the pendulum in the vertical position using the rotation of the arm. Moreover, this stabilization must be accomplished on a wide range with respect to the angular position of the arm. A detailed description of the plant as well as the corresponding physical experiment is given in [19].

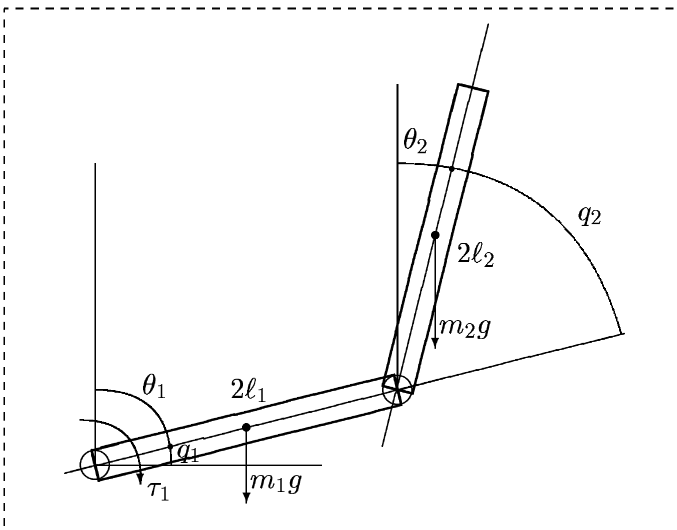

Figure 3: Inverted pendulum

By selecting as state vector $x := [\, z \quad \dot{z} \quad r_x \quad \varphi_1 \,]^T$, where r_x is the horizontal position of the arm tip (r_y is the vertical position), φ_1 and φ_2 are the angular positions of the arm and the pendulum, respectively, and $z := r_x + \frac{4}{3} l_2 \varphi_2$, the following simplified LFT state-space representation is obtained [19].

$$\begin{array}{rcl} \dot{x} & = & Ax + B_\Delta w_\Delta + Bu \\ z_\Delta & = & C_\Delta x \\ w_\Delta & = & \Delta z_\Delta \,, \end{array}$$

where the parameter structure is given as

$$\Delta := \begin{bmatrix} r_y & 0 & 0 \\ 0 & \varphi_2 & 0 \\ 0 & 0 & \varphi_2 \end{bmatrix}.$$

Therefore, the inverted pendulum admits LPV dynamics and can be controlled using either LPV or robust control techniques, as those considered here. Given an operating range for the inverted pendulum, the parameters are normalized such that $\Delta = \text{diag}(\delta_1, \delta_2 I_2)$ with $|\delta_i| \le 1$, $i = 1, 2$.

The synthesis structure used to achieve the design requirements is shown in Figure 4. It simply translates performance tracking ($\omega_I x_I$) and high-frequency gain attenuation ($\omega_d \dot{r}_x$).

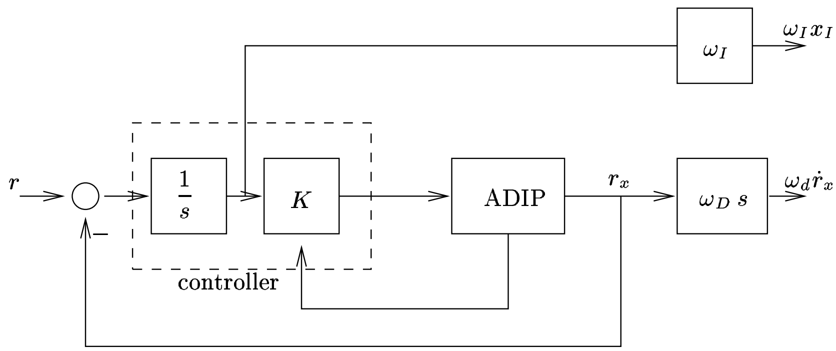

Figure 4: Synthesis structure for the inverted pendulum

Formulated in this way, the local and global robust control techniques discussed in this paper are immediately applicable. The numerical data of the synthesis interconnection are given in Appendix B.

The following table displays the performance of each algorithm in terms of number of iterations and cputime. The computations were performed on a PC with CPU Pentium II 330 Mhz and all LMI-related computations were performed using the *LMI Control Toolbox* [9]. Remember that the simplicial and conical algorithms are used only after the FW algorithm has failed ($\gamma = 0.1903$ in this case). The symbol 'f' indicates a failure of the FW algorithm to achieve the corresponding value of γ, first column, whereas the symbol 'inf' is used to specify infeasibility of γ.

From Table 1, we see that the performance found by the FW algorithm is within 5.5 % of the global optimal value of γ. It is also worth noticing that with the same γ, there are many solutions obtained by the global algorithms. For instance, for $\gamma = 0.1838$, the scaling solutions with the simplicial and conical algorithms are given as

$$S = \begin{bmatrix} 1.2261 \times 10^{-5} & 0 & 0 \\ 0 & 0.5110 & -0.0231 \\ 0 & -0.0231 & 0.0042 \end{bmatrix}, \; T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.0014 \\ 0 & 0.0014 & 0 \end{bmatrix},$$

and

$$S = \begin{bmatrix} 1.2261 \times 10^{-5} & 0 & 0 \\ 0 & 0.1719 & 0.0010 \\ 0 & 0.0010 & 4.2145 \times 10^{-5} \end{bmatrix}, \; T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.0073 \\ 0 & -0.0073 & 0 \end{bmatrix},$$

	FWA		SA		CA	
γ	# iter.	cputime	# iter.	cputime	# iter.	cputime
0.2	3	65.74 sec.	-	-	-	-
0.1910	10	148.03 sec.	-	-	-	-
0.1905	10	152.09 sec	-	-	-	-
0.1904	2	56.08 sec	-	-	-	-
0.1903	f	f	1	12.3 sec.	1	18.73 sec
0.1838	-	-	2	84.80 sec.	1	18.95 sec
0.18375	-	-	12(inf)	793.01 sec.	1	18.840 sec.
0.18370	-	-	1(inf)	13.03 sec	1(inf)	16.04 sec

Table 1: FWA: Frank and Wolf Algorithm; SA: simplicial algorithm; CA: conical algorithm; f: the test fails; inf: no zero optimal value (infeas.)

respectively. The optimal scalings with $\gamma = 0.18375$ and the conical algorithm are

$$S = \begin{bmatrix} 1.2264 \times 10^{-5} & 0 & 0 \\ 0 & 0.1748 & 0.0010 \\ 0 & 0.0010 & 3.3449 \times 10^{-5} \end{bmatrix}, \; T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.0074 \\ 0 & -0.0074 & 0 \end{bmatrix}$$

The optimal value of γ achieved with both the simplicial and conical algorithms are very close to that obtained using LPV synthesis ($\gamma = 0.1830$), which indicates that one will hardly find a better linear time-invariant controller for the specified control objectives.

6.2 Randomly generated examples

Furthermore, our algorithms were tested over a hundred randomly generated robust control problems with sizes around those of the inverted pendulum (Table 2) and also for problems of much larger dimensions. Computational experience shows that the number of iterations is almost not sensitive to the problem dimensions while the cputime is strongly depending on the efficiency of the LMI solver used in the FW steps and also for lower bound computations in the simplicial and conical algorithms.

The optimal γ is computed using a bisection scheme in the interval $[\gamma_{lpv}, \gamma_{fw}]$, where γ_{fw} is the best performance reached by the FW algorithm and γ_{lpv} is the optimal performance achieved using LPV synthesis. Note that γ_{lpv} obviously provides a global lower bound on the performance level.

Step 0. Set $\gamma_{opt} = \gamma_{fw}, \gamma_{if} = \gamma_{lpv}$;

Step 1. Take $\gamma = (\gamma_{opt} + \gamma_{if})/2$ and use the simplicial/conical algorithm to solve **Pb. 1**. If LMIs (8)-(22) are feasible then set $\gamma_{opt} \leftarrow \gamma$. Otherwise set $\gamma_{if} \leftarrow \gamma$.

In our experiments, the conical algorithm is slower than the simplicial one for finding a feasible solution but it is faster for proving infeasibility of some γ. The average performance for tolerance samples from 8% to 2.5% is presented in Table 2.

γ	FWA		SA		CA	
	# iter.	cputime	# iter.	cputime	# iter.	cputime
γ_{fw}	25.3	436.72 sec.	-	-	-	-
$\gamma_{8\%}$	-	-	2.1	75.63 sec.	2.3	137.86 sec.
$\gamma_{5\%}$	-	-	7.0	405.24 sec.	5.3	300.11 sec.
$\gamma_{2.5\%}$	-	-	20.3(t)	830.85 sec. (t)	36.7 (t)	1007.33(t)

Table 2: FWA: Frank and Wolf Algorithm; SA: simplicial algorithm; CA: conical algorithm; t: total iterations/cputime needed for arriving at the optimal solution in the γ bisection.

6.2.1 A larger problem

Finally, we consider a larger problem with the data given in Appendix C. This example is intended to serve as a test example for competing techniques. It is also interesting since standard $D-K$ iteration schemes fail to reach an optimal solution. The uncertainty structure is described as

$$\Delta(t) = \text{diag}(\delta_1(t)I_2, \delta_2(t)I_2, ..., \delta_5(t)I_2)\,.$$

This gives rise to the following scaling structure in (7)

$$S = \text{diag}(S_1, S_2, \ldots, S_5),\ T = \text{diag}(T_1, T_2, \ldots, T_5)\,,$$

where S_i and T_i are 2×2 symmetric and skew-symmetric matrices. The performance in terms of iterations of our algorithms is very much like that of the inverted pendulum examples and is omitted here to save a space. The best performance found by the Frank and Wolfe algorithm is $\gamma_{fw} = 1.7890$ while the best (global optimal) performance found by both simplicial and conical algorithms is $\gamma_{opt} = 1.7835$. Here again the FW algorithm provides a very good suboptimal value. At γ_{opt} both simplicial and conical algorithms need just one iteration with the stopping criterion to find the corresponding optimal scaling with cputime 266.66 sec. and 566.61 sec., respectively. The computional times for proving infeasibility of a smaller γ are 236.67 sec. and 299.70 sec.. As for the random examples in Table 2, the iteration count remains reasonable and the cputime performance is dictated by the time required for solving an individual LMI problem. The optimal scaling found by the simplicial algorithm is

$$\begin{aligned}
S_1 &= \begin{bmatrix} 0.6231 & 0.0078 \\ 0.0078 & 0.8834 \end{bmatrix}, \quad & S_2 &= \begin{bmatrix} 24.5429 & 0.2083 \\ 0.2083 & 24.2111 \end{bmatrix}, \quad & S_3 &= \begin{bmatrix} 24.5255 & 0.2913 \\ 0.2913 & 23.5374 \end{bmatrix} \\
S_4 &= \begin{bmatrix} 2.5050 & 0.2320 \\ 0.2320 & 2.9766 \end{bmatrix}, \quad & S_5 &= \begin{bmatrix} 24.7840 & -0.0371 \\ -0.0371 & 24.4618 \end{bmatrix}, & & \\
T_1 &= \begin{bmatrix} 0 & -0.0319 \\ 0.03190 & \end{bmatrix}, \quad & T_2 &= \begin{bmatrix} 0 & -0.2848 \\ 0.2848 & 0 \end{bmatrix}, \quad & T_3 &= \begin{bmatrix} 0 & -0.2694 \\ 0.26940 & \end{bmatrix}, \\
T_4 &= \begin{bmatrix} 0 & -0.0829 \\ 0.0829 & 0 \end{bmatrix}, \quad & T_5 &= \begin{bmatrix} 0 & -0.3887 \\ 0.3887 & 0 \end{bmatrix}, & &
\end{aligned}$$

while that found by the conical one is

$$
\begin{aligned}
S_1 &= \begin{bmatrix} 0.6489 & -0.7276 \\ -0.7276 & 1.0803 \end{bmatrix}, & S_2 &= \begin{bmatrix} 6.4585 & -9.4148 \\ -9.4148 & 14.1924 \end{bmatrix}, & S_3 &= \begin{bmatrix} 3.3073 & -4.6313 \\ -4.6313 & 6.9327 \end{bmatrix}, \\
S_4 &= \begin{bmatrix} 2.4087 & -0.5502 \\ -0.5502 & 2.6384 \end{bmatrix}, & S_5 &= \begin{bmatrix} 10.9920 & -16.1119 \\ -16.1119 & 24.0839 \end{bmatrix}, \\
T_1 &= \begin{bmatrix} 0 & 0.6988 \\ -0.6988 & 0 \end{bmatrix}, & T_2 &= \begin{bmatrix} 0 & -0.0206 \\ 0.0206 & 0 \end{bmatrix}, & T_3 &= \begin{bmatrix} 0 & -0.0678 \\ 0.0678 & 0 \end{bmatrix}, \\
T_4 &= \begin{bmatrix} 0 & 0.6785 \\ -0.6785 & 0 \end{bmatrix}, & T_5 &= \begin{bmatrix} 0 & -0.0108 \\ 0.0108 & 0 \end{bmatrix}.
\end{aligned}
$$

7 Concluding remarks

In this paper, we show that many important problems in robust control theory can be formulated as the minimization of a concave functional over a convex set determined by LMI constraints. In this respect, concavity appears to play a central role in a broad class of problems. This is the departure point which motivates the development of a comprehensive technique which provides a global solution of robust control problems admitting scaling-based characterizations. Although, we do not pursue the vein further, it appears that the technique is applicable with only modest changes to many other difficult problems encompassing fixed-order robust control, multi-objective LPV control, ... and any aggregation of these problems. We also derive new results, interesting in their own, which clarify the equivalence between BMIs, rank constrained LMI problems and zero-seeking concave programs.

The proposed optimization method comprises a local search algorithm combined with extensions of global concave minimization techniques which at the final stage secure global optimality of the solutions or invalidate feasibility of the problem. The method takes advantage of the concavity and convexity characteristics of the problem. It is also aided by adequate stopping criteria to reduce as far as possible the overall computational overhead.

Surprisingly, the theoretically predicted high degree of complexity of the problems under consideration never shows up both in realistic and randomized experiments. Therefore, the only limitation of the method turns out to be the power of currently available semidefinite programming solvers for handling repeated LMI problems. Experience on large problems demonstrates that it constitutes a tractable approach for realistic applications. The good results obtained in this paper are not exception in general nonconvex optimization. For the the geometric problem of point sets bilinear separation, Bennett and Mangasarian note that experimentally, the FW algorithm provides optimal solutions without a single failure [3]. Konno, Thach and Tuy point out the fact that for concave problems with low rank nonconvex structures the time required to get a global solution is often not much than the time of a few linear programs [20].

This work also raises several important directions for future research.

- The use of the algorithms for handling general BMI problems is currently under study.
- Extensions of the technique to Popov multipliers are also of practical interest to tackle more sophisticated uncertainty descriptions.
- From an optimization viewpoint, there are different ways for further improving the efficiency of the algorithms, for instance, by exploiting monotonicity properties of the objective function. The FW algorithm can be locally accelerated by a Newton-like method with quadratic local

convergence. There are also possibilities of using multiple restarts of the FW algorithm inside the simplicial/conical techniques while preserving global convergence and optimality.

A Convergence of simplicial algorithm

Let M_k be the simplex chosen for subdivision at iteration k and $(Z_1(M_k), \omega(M_k))$ be the optimal solution of problem (35) with $M = M_k$, i.e. $\omega(M_k) \in M_k \cap M_0$ and $\phi_{M_k}(Z_1(M_k), \omega(M_k)) = \beta(M_k)$. Note that, as $(Z_1(M_k), \omega(M_k))$ is feasible, we must have $\nu_k \le f(Z_1(M_k), \omega(M_k))$, so if it so happens that $\omega(M_k) \in \text{vert}(M_k)$ then $\beta(M_k) = \phi_M(Z_1(M_k), \omega(M_k)) = f(Z_1(M_k), \omega(M_k)) = \nu_k$ and therefore $\beta(M_k)$ will be the exact minimum of f over $\mathcal{X}$ and according to the stop criterion (33) the algorithm will terminate. This suggests that to accelerate the convergence one should subdivide M_k via $\omega(M_k)$. Such a subdivision strategy, called the *$\omega-$subdivision strategy* [32], has long been used [31] and is known to work well in practice though its theoretical convergence is still an open question [32]. Another subdivision strategy called the *bisection strategy* consists in subdividing M via the midpoint of its longest edge. This subdivision guarantees convergence but the convergence speed is most often much slower than the previous one. Therefore, the following so called *normal subdivision rule* which combines $\omega-$subdivision with bisections in a mixed strategy is a recognized good trade-off between convergence and efficiency.

Normal subdivision rule. *Let M_k be the candidate simplex for subdivision at iteration k. Select an infinite increasing sequence Π of natural numbers and define the generation index of every simplex M by setting $\tau(M_0) = 0$ and $\tau(M') = \tau(M) + 1$ whenever M' is a child of M (i.e. M' is one member of the partition of M). Then: if $\tau(M_k) \in \Pi$ then bisect M_k. Otherwise ω-subdivide M_k.*

The idea of the normal rule is to use ω-subdivision in most iterations and bisection occasionally, in such a way that any infinite nested sequence of generated simplices involves infinitely many bisections. In practical implementation, it suffices to do one or two bisections only when the procedure seems to slow down.

A basic property of the normal rule ensuring its convergence is the following [32, Th. 5.1].

Lemma A.1 *Let $\{M_k\},\ k = 0, 1, 2, ...$ be any infinite nested sequence of simplices generated by a given normal rule. Then at least one accumulation point ω^∞ of the sequence $\{\omega^k\} = \{\omega(M_k)\}$ will be a vertex of $M_\infty = \cap_{k=1}^{\infty} M_k$.*

Theorem A.2 *Either the simplicial algorithm terminates after finitely many iterations, yielding a zero optimal solution of (25) (termination at Step 2) or providing evidence that (25) has no zero optimal solution (termination at Step 4). Or it generates an infinite sequence of feasible solutions (Z_1^k, ω^k) converging to a zero optimal solution.*

Proof: Suppose the algorithm is infinite. One of the children of M_0 must have infinitely many descendants, hence must be splitted at some iteration k_1, i.e. must be M_{k_1} for some k_1 (following our notation, M_k is the candidate for further partition at iteration k). Analogously, one child of M_{k_1} must be M_{k_2} at some iteration $k_2 > k_1$ and so on. Proceeding that way we see that there exists a nested sequence $M_{k_\nu},\ \nu = 1, 2, ...$ as in Lemma A.1. For short, let us write M_ν for M_{k_ν} and (Z_1^ν, ω^ν) for $(Z_1(M_{k_\nu}), \omega(M_{k_\nu}))$. Without loss of generality, we can assume that $(Z_1^\nu, \omega^\nu) \to (Z_1^\infty, \omega^\infty)$ with $\omega^\infty \in \text{vert}(\cap_{\nu=1}^{\infty} M_\nu)$. But it is easy to see that any vertex of $M_\infty := \cap_{\nu=1}^{\infty} M_\nu$ is an accumulation point of some sequence $\{u^{\nu,i} \in \text{vert}(M_\nu), \nu = 1, 2, \ldots\}$, where the second index stands for the vertex number. By passing to subsequences if necessary, we can assume without loss of generality that $u^{\nu,1} \to \omega^\infty$. Then $\phi_{M_\nu}(Z_1^\nu, \omega^\nu) - \phi_{M_\nu}(Z_1^\nu, u^{\nu,1}) \to 0$ because this difference equals $\sum_{i=1}^{N+1}[\lambda_{\nu,i} - \mu_{\nu,i}]f(0, u^{\nu,i})$ with $u^{\nu,i} \to u^{*,i}$ and $\lambda_{\nu,i} - \mu_{\nu,i} \to 0$ as $\nu \to \infty$ (in view of the fact $\omega^\nu - u^{\nu,1} \to 0$). Therefore,

$$f(Z_1^\infty, \omega^\infty) = \lim_{\nu\to\infty} f(Z_1^\nu, u^{\nu,1}) = \lim_{\nu\to\infty} \phi_{M_\nu}(Z_1^\nu, u^{\nu,1}) = \lim_{\nu\to\infty} \phi_{M_\nu}(Z_1^\nu, \omega^\nu) = \lim_{\nu\to\infty} \beta(M_\nu).$$

Now, as was pointed out earlier, $\beta(M_\nu) \le \min\{f(Z) : \ Z \in \mathcal{X}\}$. Hence $f(Z_1^\infty, \omega^\infty) \le \min\{f(Z) : Z \in \mathcal{X}\}$. On the other hand, Step 3 guarantees that $\beta(M_{k_\nu}) < 0$, while the termination criterion in Step 2 implies $f(Z_1^\nu, \omega^\nu) \ge 0$. Taking the limits yields $f(Z_1^\infty, \omega^\infty) \le 0$ and $f(Z_1^\infty, \omega^\infty) \ge 0$, hence $f(Z_1^\infty, \omega^\infty) = 0$. ■

B State-space data for inverted pendulum

$$\left[\begin{array}{c|c|c|c} A & B_\Delta & B_1 & B_2 \\ \hline C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2} \\ \hline C_1 & D_{1\Delta} & D_{11} & D_{12} \\ \hline C_2 & D_{2\Delta} & D_{21} & 0 \end{array}\right] :=$$

$$\left[\begin{array}{ccccc|ccc|cc|c}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
48.9844 & 0 & -48.9844 & 0 & 0 & 0 & -.35634 & -.015548 & 0 & 0 & 0 \\
0 & 0 & 0 & .184940 & 0 & .0750596 & 0 & 0 & 0 & 0 & 50.0 \\
0 & 0 & 0 & -50.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -.50 & 0 & 0 & 0 & 0 & 0 & 0 & .50 & 0 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
436.33231 & 0 & -.043633 & 0 & 0 & 0 & 0 & .043633 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & .0036988 & 0 & .001501 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

C State-space data for large problem

$$A = [\, A_1 \quad A_2 \,],$$

$$A1 = \begin{bmatrix}
-101 & -99.90000 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -101 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -101 & -99.90000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -101 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 427.09800 & -46.83410 \\
0 & 0 & 0 & 0 & 0 & 0 & 232.07190 & 120.46490 \\
0 & 0 & 0 & 0 & 0 & 0 & -764.24560 & 85.41540 \\
0 & 0 & 0 & 0 & 0 & 0 & 166.82700 & -264.77390 \\
0 & 0 & 0 & 0 & 0.31620 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.12500 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.31620 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.12500 & 0 & 0
\end{bmatrix}$$

$$A2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0.42710 & -0.04680 & 0 & 0 & 0 & 0 \\
0 & -1 & 0.23210 & 0.12050 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.76420 & 0.08540 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.16680 & -1.26480 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1.10000 & -0.07590 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1.10000 & -0.07590 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}$$

$$B_\Delta = \begin{bmatrix}
0 & 0 & 0 & 0 & -0.00100 & 0 & 0 & 0 & -0.00100 & 0 \\
0 & 0 & 0 & 0 & -0.00100 & 0 & 0 & 0 & -0.00100 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.00100 & 0 & 0 & 0 & -0.00100 \\
0 & 0 & 0 & 0 & 0 & -0.00100 & 0 & 0 & 0 & -0.00100 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.63080 & 0.17870 & 0 & 0 & 0.00030 & 0.00010 & -0.63080 & 0.17870 & 0.00030 & 0.00010 \\
0.63200 & -0.83640 & 0 & 0 & 0.00010 & 0.00030 & 0.63200 & -0.83640 & 0.00010 & 0.00030 \\
2.22270 & 0.08180 & 0 & 0 & -0.00050 & -0.00030 & 2.22270 & 0.08180 & -0.00050 & -0.00030 \\
-2.06740 & 0.35770 & 0 & 0 & 0.00010 & -0.00030 & -2.06740 & 0.35770 & 0.00010 & -0.00030 \\
0 & 0 & -0.31620 & 0 & 0 & 0 & 0 & 0 & -0.31620 & 0 \\
0 & 0 & 0.12500 & 0 & 0 & 0 & 0 & 0 & 0.12500 & 0 \\
0 & 0 & 0 & -0.31620 & 0 & 0 & 0 & -0.31620 & 0 & 0 \\
0 & 0 & 0 & 0.12500 & 0 & 0 & 0 & 0 & 0 & 0.12500
\end{bmatrix}$$

$$[B_1 \, B_2] = \left[\begin{array}{ccc|ccc}
0 & -0.00100 & 0 & 0 & -9.99500 & 0 \\
0 & -0.00100 & 0 & 0.19900 & -9.99500 & 0 \\
0 & 0 & -0.00100 & 0.21100 & 0 & -9.99500 \\
0 & 0 & -0.00100 & -0.23300 & 0 & -9.99500 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0.17870 & 0.00030 & 0.00010 & 0 & 2.71730 & 1.42740 \\
-0.83640 & 0.00010 & 0.00030 & 0 & 1.42740 & 2.83820 \\
0.08180 & -0.00050 & -0.00030 & 0 & -4.79090 & -2.60320 \\
0.35770 & 0.00010 & -0.00030 & 0 & 1.02610 & -2.63930 \\
0 & -0.31620 & 0 & 0.11000 & 0 & 0 \\
0 & 0.12500 & 0 & 0 & 0 & 0 \\
-0.31620 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.12500 & 0.01000 & 0 & 0
\end{array}\right]$$

$$C_\Delta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -6.11370 & 2.91140 & 0 & 0 & -0.00610 & 0.00290 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.55640 & 3.48340 & 0 & 0 & 0.00160 & 0.00350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.47430 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.34790 & 0 \\ 9.99500 & 9.99500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.99500 & 9.99500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.55640 & 3.48340 & 0 & 0 & 0.00160 & 0.00350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.47430 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.34790 & 0 \\ 9.99500 & 9.99500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1.55640 & 3.48340 & 0 & 0 & 0.00160 & 0.00350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.47430 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.34790 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.31620 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.31620 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.55640 & 3.48340 & 0 & 0 & 0.00160 & 0.00350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.47430 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.34790 & 0 \end{bmatrix}$$

$$[\, D_{\Delta\Delta} \quad D_{\Delta 1} \quad D_{\Delta 2} \,] = \begin{bmatrix} 0.59860 & -0.00390 & 0 & 0 & 0 & 0 & 0.59860 & -0.00390 & 0 & 0 & 0.59860 & 0.02730 & 0 & 0 & 0 & 0 \\ 0.02730 & -0.56130 & 0 & 0 & 0 & 0 & 0.02730 & -0.56130 & 0 & 0 & -0.00390 & -0.56130 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.59860 & 0.02730 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.00390 & -0.56130 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$[\, D_{1\Delta} \quad D_{11} \quad D_{12} \,] = \begin{bmatrix} 0.59860 & -0.00390 & 0 & 0 & 0 & 0 & 0.59860 & -0.00390 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.02730 & -0.56130 & 0 & 0 & 0 & 0 & 0.02730 & -0.56130 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$[\, D_{2\Delta} \quad D_{21} \,] = 0 \,.$$

Acknowledgment. The authors are grateful to Professor Hoang Tuy for his invaluable suggestions and advice on concave minimization algorithms. We also wish to thank Dr. Hiroyuki Kajiwara at Kyutech for providing the arm-driven inverted pendulum example.

References

[1] P. Apkarian and P. Gahinet, *A Convex Characterization of Gain-Scheduled H_∞ Controllers*, IEEE Trans. Aut. Control, 40 (1995), pp. 853–864. See also pp. 1681.

[2] P. Apkarian, P. Gahinet, and G. Becker, *Self-Scheduled H_∞ Control of Linear Parameter-Varying Systems: A Design Example*, Automatica, 31 (1995), pp. 1251–1261.

[3] K. P. Bennett and O. L. Mangasarian, *Bilinear Separation of Two Sets in n-Space*, Computational Optimization and Applications, (1993), pp. 207–227.

[4] J. Doyle, A. Packard, and K. Zhou, *Review of LFT's, LMI's and μ*, in Proc. IEEE Conf. on Decision and Control, vol. 2, Brighton, Dec. 1991, pp. 1227–1232.

[5] M. K. H. Fan, A. L. Tits, and J. C. Doyle, *Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics*, IEEE Trans. Aut. Control, AC-36 (1991), pp. 25–38.

[6] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Log. Quart., 3 (1956), pp. 95–110.

[7] M. Fu and Z.-Q. Luo, *Computational complexity of a problem arising in fixed-order output feedback design*, Syst. Control Letters, (1998). to appear.

[8] P. Gahinet and P. Apkarian, *A Linear Matrix Inequality Approach to H_∞ Control*, Int. J. Robust and Nonlinear Control, 4 (1994), pp. 421–448.

[9] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox* , The MathWorks Inc., 1995.

[10] J. C. Geromel, C. C. de Souza, and R. E. Skelton, *Static Output Feedback Controllers : Stability and Convexity*, IEEE Trans. Aut. Control, 43 (1998), pp. 120–125.

[11] L. E. Ghaoui, F. Oustry, and M. AitRami, *A Cone Complementary Linearization Algorithm for Static Output-Feedback and Related Problems*, IEEE Trans. Aut. Control, 42 (1997), pp. 1171–1176.

[12] K. C. Goh, M. G. Safonov, and G. P. Papavassilopoulos, *A Global Optimization Approach to the BMI Problem*, in Proc. IEEE Conf. on Decision and Control, Dec. 1994, pp. 45–49.

[13] K. M. Grigoriadis and R. E. Skelton, *Low-order Control Design for LMI Problems Using Alternating Projection Methods*, Automatica, 32 (1996), pp. 1117–1125.

[14] A. Helmersson, *Methods for Robust Gain-Scheduling*, Ph. D. Thesis, Linkoping University, Sweden, 1995.

[15] R. Horst and P. Pardalos, eds., *Handbook of Global Optimization*, Kluwer Academic Publishers, 1995.

[16] R. Horst and H. Tuy, *Global optimization: deterministic approaches*, Springer (3rd edition), 1996.

[17] T. Iwasaki, *The dual iteration for fixed-order control*, in Proc. American Control Conf., 1997, pp. 62–66.

[18] T. Iwasaki and R. E. Skelton, *All Controllers for the General H_∞ Control Problem: LMI Existence Conditions and State Space Formulas*, Automatica, 30 (1994), pp. 1307–1317.

[19] H. Kajiwara, P. Apkarian, and P. Gahinet, *Wide-Range Stabilization of an Arm-Driven Inverted Pendulum Using Linear Parameter-Varying Techniques*, in AIAA Guid., Nav. and Control Conf., 1998. to appear.

[20] H. Konno, P. T. Thach, and H. Tuy, *Optimization on low rank nonconvex structures*, Kluwer Academic Publishers, Dordrecht/Boston/London, 1996.

[21] M. Mesbahi and G. P. Papavassilopoulos, *Solving a Class of Rank Minimization Problems via Semi-Definite Programs, with Applications to the Fixed Order Output Feedback Synthesis*, in Proc. American Control Conf., 1997.

[22] A. Packard, *Gain Scheduling via Linear Fractional Transformations*, Syst. Control Letters, 22 (1994), pp. 79–92.

[23] A. Packard and G. Becker, *Quadratic Stabilization of Parametrically-Dependent Linear Systems using Parametrically-Dependent Linear, Dynamic Feedback*, Advances in Robust and Nonlinear Control Systems, DSC-Vol. 43 (1992), pp. 29–36.

[24] A. Packard, K. Zhou, P. Pandey, and G. Becker, *A collection of robust control problems leading to LMI's*, in Proc. IEEE Conf. on Decision and Control, vol. 2, Brighton, Dec. 1991, pp. 1245–1250.

[25] M. G. Safonov, K. C. Goh, and J. H. Ly, *Control System Synthesis via Bilinear Matrix Inequalities*, in Proc. American Control Conf., 1994, pp. 45–49.

[26] C. Scherer, *Robust Generalized H_2 Control for Uncertain and LPV Systems with General Scalings*, in Proc. IEEE Conf. on Decision and Control, Kobe, JP, 1996, pp. 3970–3975.

[27] C. W. Scherer, *A Full Block $\mathcal{S}$-Procedure with Applications*, in Proc. IEEE Conf. on Decision and Control, San Diego,USA, 1997, pp. 2602–2607.

[28] G. Scorletti and L. E. Ghaoui, *Improved Linear Matrix Inequality Conditions for Gain-Scheduling*, in Proc. IEEE Conf. on Decision and Control, New Orleans, LA, Dec. 1995, pp. 3626–3631.

[29] O. Toker and H. Ozbay, *On the np-hardness of the purely complex μ computation, analysis/synthesis, and some related problems in multidimensional systems*, in Proc. American Control Conf., 1995, pp. 447–451.

[30] H. Tuan, S. Hosoe, and H. Tuy, *New global optimization algorithms for solving the robust performance problem of robust controls*, in 2nd Asian Control Conference, vol. 1, 1997, pp. 350–353.

[31] H. Tuy, *Concave programming under linear constraints*, Soviet Mathematics, (1964), pp. 1437–1440.

[32] ——, *Convex analysis and global optimization*, Kluwer Academic Publishers, 1998.