Robust Control via Concave Minimization Local and Global Algorithms

Pierre Apkarian[1] *Hoang Duong Tuan*[3]

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 and many others can be formulated as the concave minimization of a nonlinear functional subject to Linear Matrix Inequalities 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 minimization programming methods are then introduced and combined with the local feasible direction method to secure and certify global optimality of the solutions. The implementation details of the algorithms are covered. A special focus is put on the development of new stopping criteria 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.

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 [28, 10, 20] that there exists a k-th order controller solving the H_∞ control problem iff one can find a pair of symmetric matrices (X, Y) such that for some H_∞ performance level γ the following holds.

$$\mathcal{L}(X, Y, \gamma) \quad < \quad 0\,, \tag{1}$$

$$\text{Rank}\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \quad \leq \quad n + k\,, \tag{2}$$

where n designates the plant's order and (1) is an LMI in X, Y and γ. The hardness of this problem stems from the rank condition (2) which is essentially non-convex. As a byproduct of the results in this paper, we show that this problem can be formulated as a concave minimization program over a convex set which establishes its NP-hardness. Different proofs of NP-hardness are also given in [33, 9].

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 [14], Grigoriadis and Skelton consider a method based on *alternating projections* for iteratively finding a solution to the rank constraint (2). In [19], Iwasaki derives an 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 [25]. These bounds can potentially be used in *branch and bound* refinement schemes to locate approximate solutions. In [12] 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 [13] 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 [8], 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 [34], the second author 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 can be recast as concave minimization problems. That is, problems involving a concave functional subject to convex constraints consisting of LMIs. A sample list of such problems 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 any combination of such problems. Although, this is not the central object of this paper, we reveal that BMI (Bilinear Matrix Inequality) problems and some other generalizations 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. These problems are generally difficult to deal with but exhibit some nice geometric concave structure that makes them more attractive and painless than general nonlinear optimization problems. Another distinguished characteristic of the concave problems under study is that whenever feasible, optimality occurs only at zeros of the concave functional. In this respect, such problems can be reinterpreted as zero finding concave programs which significantly reduces the difficulty of the search.

- Starting from this viewpoint, the work here provides first a full generalization of the technique in [13] to handle robust control problems for plants subject to time-varying LFT (Linear Fractional Transformation) uncertainties. More precisely, we show that a FW algorithm can be used to solve or find local solutions of robust synthesis problems involving either pairs of symmetric and skew-symmetric scalings or full generalized scalings as discussed in [30]. We demonstrate that these problems are equivalent to zero-seeking concave programming problems where the convex constraints express in terms of LMIs. We also indicate how the FW algorithm must be modified to handle other types of problems. We shall briefly prove that the FW algorithm is guaranteed to generate strictly decreasing sequences for the objective functional and that the sequence of points is either infinite or reach a local optimal solution. This follows from earlier results by Bennett and Mangasarian in [5].

An elementary step of the FW algorithm reduces to solving an LMI program whose cost is heavily dependent on the dimensions of the problem at hand (number of states and uncertainties). Therefore, an important part of the paper is dedicated to describing the implementation details of the proposed algorithms, including initialization, feasible descent directions and stopping criteria. 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. A key idea of these stopping criteria is to limit as far as possible the *zig-zagging* phenomenon which characterizes first-order descent methods such as the FW algorithm and hence to ensure reasonable computational time.

- It is important to stress out that the FW algorithm is of local nature and as such is not guaranteed to provide a global solution to our problems. This naturally leads us to combining recently available con-

[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

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

cave programming methods with the FW algorithm to certify global optimality of the solutions or invalidate feasibility. As concave programming is the best studied class of problems in global optimization [16, 17, 22, 36], one advantage of our formulation is that several efficient and practical algorithms are available for its resolution. In this paper, we have opted for a suitably built FW algorithm combined with several alternative concave programming techniques. The FW algorithm is much less costly but in return, is prone to non-global optimality. On the other hand, concave minimization 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 taking advantage of 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. We have paid special attention to the simplicial and conical Branch and Bound concave minimization methods which respectively divide the feasible set into simplices and cones of decreasing sizes. The main thrust of these techniques is that they rely heavily 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. On the other hand, the stopping criteria mentioned above are again useful to further reduce the computational burden.

As for the FW algorithm, the practical implementation of these methods in the robust control context of this paper is thoroughly investigated. 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.

2 Problem presentation

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

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ z_\Delta \\ z \\ y \end{bmatrix} &= \begin{bmatrix} A & B_\Delta & B_1 & B_2 \\ C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2} \\ C_1 & D_{1\Delta} & D_{11} & D_{12} \\ C_2 & D_{2\Delta} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w_\Delta \\ w \\ u \end{bmatrix} \\ w_\Delta &= \Delta(t)\, z_\Delta\,, \end{aligned} \tag{3}$$

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

$$\Delta(t) = \mathrm{diag}\,(\ldots, \delta_i(t)I, \ldots, \Delta_j(t), \ldots) \in \mathbf{R}^{N\times N} \tag{4}$$

and normalized such that

$$\Delta(t)^T \Delta(t) \le I, \qquad t \ge 0\,. \tag{5}$$

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 [7, 6]. 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 (3)-(5) the robust control problem consists in seeking a Linear Time-Invariant (LTI) controller

$$\begin{aligned} \dot{x}_K &= A_K x_K + B_K y\,, \\ u &= C_K x_K + D_K y\,, \end{aligned} \tag{6}$$

such that for all parameter trajectories $\Delta(t)$ defined by (5)
- the closed-loop system (3)-(5) and (6) is internally stable,
- the L_2-induced gain of the operator connecting w to z is bounded by γ.

It is now well-known that such problems can be handled via a suitable generalization of the Bounded Real Lemma. The reader is referred to references [27, 26, 1, 2, 15, 32] for more details and additional results.

2.1 Solvability conditions for LFT plants

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 (4). Denoting this structure as $\mathbf{\Delta}$, the following scaling sets can be introduced. The set of symmetric scalings associated with the parameter structure $\mathbf{\Delta}$ is defined as

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

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

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

Solvability conditions are given in the next theorem.

Theorem 2.1 *Consider the LFT plant governed by (3) and (5) with Δ assuming a block-diagonal structure as in (4). 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*

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

hold and the matrix inequalities below, denoted $\mathbf{LMI}\,[i]$, $i = 1, \ldots, 4$,

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

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

$$-\begin{bmatrix} X & I \\ I & Y \end{bmatrix} < 0\,, \tag{10}$$

$$-\begin{bmatrix} S & 0 \\ 0 & \Sigma \end{bmatrix} < 0 \tag{11}$$

subject to the algebraic constraints

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

are feasible.

Note that due to the algebraic constraints (12), the problem under consideration is non-convex and has been even shown to have non-polynomial (NP) complexity. See [5] and references therein. Simpler instances of this problem as those considered in [24] are NP complete . 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 condition (12) fully disappears.

3 Concave minimization programs subject to LMI constraints

For tractability reasons, it is interesting to find alternate formulations that are amenable to numerical computations. A potential technique was introduced in [13] and amounts to constructing a nonlinear functional whose feasible optimal points satisfy the algebraic constraints (12). Hereafter, we develop a suitable extension of this technique that is applicable to structured μ-scalings S and T, but also to full-block generalized scalings as considered in [30]. We state this more formally in the next Lemma.

Lemma 3.1 *Introduce the concave LMI-constrained minimization program*

$$\textbf{Pb1:} \qquad \textit{minimize } \mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) \tag{13}$$

subject to LMIs (8)-(11) and

$$\mathbf{LMI}\,[5]\ :\ -\begin{bmatrix} Z_1 & Z_3 & S+T & I \\ Z_3^T & Z_2 & I & \Sigma + \Gamma \\ (S+T)^T & I & I & 0 \\ I & (\Sigma+\Gamma)^T & 0 & I \end{bmatrix} \le 0\,. \tag{14}$$

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

$$\mathrm{Tr}\,(Z_1 - Z_3 Z_2^{-1} Z_3^T) = 0, \qquad (15)$$

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

The concavity of the trace function in (13) follows by examination of its hypograph which turns out to be convex. 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 scalings S and T. This simple observation reduces the number of "nonconvex variables" and avoid a wasteful search in an unduly large space [4].

One advantage of the formulation of the problem as in Lemma 3.1 is that one completely gets rid of the hard set constraints (12) and the non-convexity is reflected in the functional to be optimized. A disadvantage of the general formulation in Lemma 3.1 is that the problem does no longer fall within the class of *bilinear* or *cone complementary* problems for which specialized algorithms are already available. See [5] for a survey. A central purpose of this paper is to point out and discuss adequate algorithms for solving this class of problems.

3.1 Extended class of scalings and other problems

With only minor modifications, it is possible to formulate a concave minimization program subject to LMI constraints for the robust control problem involving full generalized scalings, as discussed in [31]. The formulation is very close in spirit to that of Lemma 3.1 and is omitted to save a space (see [4]). The reader is referred to [18, 31] for thorough discussions on generalized scalings. Simple manipulations tools to convert BMI problems and some other nonlinear problems into concave programs are examined in [3]. See also [3] for a catalog of applications of the proposed techniques including robust multi-objective, robust fixed- or reduced-order control, multi-objective LPV and aggregation of these.

4 A local Frank and Wolfe algorithm

In this section, we discuss a Frank and Wolfe algorithm for finding solutions to Lemma 3.1. An analogous algorithm can be derived in the context of full generalized scalings. Such algorithms are of local nature in the sense that they cannot guarantee global optimality but have proven very efficient in practice [5, 13].

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 [8]. Consider the minimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}, \qquad (16)$$

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

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

$$\begin{array}{l} d^k \in \mathrm{argmin}\ \nabla f(x^k)\, d \\ \text{subject to}\ \ d \in \mathcal{X} \end{array}$$

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

$$\begin{array}{l} x^{k+1} = (1-\alpha^k)x^k + \alpha^k d^k, \\ \text{where } \alpha^k \in \mathrm{argmin}\, f((1-\alpha)x^k + \alpha d^k) \\ \text{subject to}\ \ 0 \le \alpha \le 1\,. \end{array}$$

Under the above very mild assumptions Bennett and Mangasarian have proved in [5] 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 [4]. The reader may also consult references [29, 23] for refinements of the Frank and Wolfe algorithm.

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. The ideas of the algorithm can be rephrased for any of the control problems mentioned earlier in Section 3. The basic ingredients of the algorithm are as follows:

- **Initialization**: this step simply consists in obtaining a feasible point of the constraints.
- **Phase I - FW step**: at this stage, we compute the gradients and perform an LMI optimization to get a best feasible descent direction in the feasible set.
- **Stopping tests**: we apply different stopping criteria in order to avoid long sequences of iterates.

Hereafter, we discuss the implementation details of each step of the general algorithm. 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 sides of the corresponding LMI in (8)-(11) and (14), respectively.

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 (14) maximally negative. This is easily formulated as the LMI program

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

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.

Phase I - FW step: In this phase, we determine a feasible segment pointing towards a descent direction. Remarking that the gradients of

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

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:

$$\begin{array}{lll} \text{minimize}\ \mathrm{Tr}\,(G_1 Z_1 + G_2 Z_2 + G_3 Z_3) & & \\ \text{subject to} & & \\ \quad \mathbf{LMI}[i] \ \le -\varepsilon I, & & i = 1,2,3,4 \\ \quad \mathbf{LMI}[5] \ \le 0, & & \\ \quad \|Z_l\| \le \rho, & & l = 1,2,3 \end{array}$$

Note that this problem is always solvable, since we are only manipulating feasible points and directions. The parameter ρ is only for preventing infinite solutions and is generally fixed to a large numerical value. As will be clarified later, we shall use perturbations of the LMIs (8) and (9) in the stopping criteria. The parameter $\varepsilon > 0$ maintains the projection conditions (LMIs (8) and (9)) in Theorem 2.1 "unsaturated". This will ease the controller construction but also leave room for applying our perturbation techniques in the next section. Note that for a general function, a line search on the matrix segment

$$\left[\begin{bmatrix} Z_1^k \\ Z_2^k \\ Z_3^k \end{bmatrix}, \begin{bmatrix} Z_1^{\text{feas}} \\ Z_2^{\text{feas}} \\ Z_3^{\text{feas}} \end{bmatrix} \right],$$

will be required, where Z_1^{feas}, Z_2^{feas} and Z_3^{feas} are solutions of the FW step above. In virtue of the concavity of the objective function (13), the line search can be completely bypassed and one can perform a *full step size of one*, hence reducing the overall computational overhead.

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. In our new notation, our test takes the form

$$\textbf{LMI}\,[i]\ < 0, \qquad i = 1, 2, 3, 4 \tag{17}$$

$$(S^k + T^k)^{-1} = (\Sigma^k + \Gamma^k)\,. \tag{18}$$

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 (14) 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

$$\widetilde{\Sigma}^k := \frac{W + W^T}{2}, \qquad \widetilde{\Gamma}^k := \frac{W - W^T}{2}\,. \tag{19}$$

  Then, stop if new point passes the test (17).

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

$$\widetilde{S}^k := \frac{W + W^T}{2}, \qquad \widetilde{T}^k := \frac{W - W^T}{2}\,. \tag{20}$$

  Then, stop if new point passes the test (17).

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 [13] which necessitate a modification of the problem.

5 Concave global optimization algorithms

Since concave minimization algorithms have been to some extent overlooked by the control community, we present some essential elements of this theory and examine its use in the context of this paper. The reader is referred to the recent book of Tuy [36] for further details.

Returning to the generic program (16) where the concave function f and convex set $\mathcal{X}$ satisfy $f(x) \geq 0,\ \forall x \in \mathcal{X}$, our goal is to check whether the optimal value of (16) is 0. In the sequel, let Z define the minimal linear space of dimension N containing $\mathcal{X}$.

A branch and bound (BB) method for solving (16) is an iterative procedure in which the space $\mathcal{Z}$ is iteratively partitioned into smaller sets (*branching*) and the search over each partition set M is carried out through estimating a lower bound $\beta(M)$ of the value of f over $x \in M \cap \mathcal{X}$ (*bounding*). At each iteration k, a feasible solution x^k is known which is the best among all feasible solutions so far obtained (x^k and the value $\nu_k := f(x^k)$ are often referred to as the current best solution and the current best value, resp.) Clearly, the partition sets M with $\beta(M) > \nu_k$ cannot contain any better feasible solution than x^k. They are therefore discarded from further consideration. On the other hand, from the information so far obtained, the partition set with smallest $\beta(M)$ can be considered the most promising one. To concentrate further investigation on this set, we subdivide it into more refined subsets. A lower bound is then computed for each of these newly generated partition sets, and the procedure goes to the next iteration. Thus, a BB method for solving (16) involves two basic operations:

Branching: The space $\mathcal{Z}$ is partitioned into finitely many polyhedrons of the same kind (simplices, cones or hyperrectangles). At each iteration, a partition polyhedron M is selected and subdivided further into several subpolyhedrons according to a specified rule.

Bounding: Given a partition set M, one has to compute a number $\beta(M)$ such that

$$\beta(M) \leq \nu(M) := \inf\{f(x) :\ x \in M \cap \mathcal{X}\}. \tag{21}$$

Let M_k be the candidate for further partition at iteration k (as mentioned above, M_k is the partition set with smallest lower bound at iteration k, so $\beta(M_k) \leq f(x),\ \forall x \in \mathcal{X}$). To ensure convergence, the operations of branching and bounding must be *consistent* in the following sense: as $k \to +\infty$, the difference $\nu_k - \beta(M_k)$ must tend to zero, i.e. the smallest lower bound at iteration k must tend to the sought global minimum of (16). Thus, with a given tolerance $\varepsilon > 0$, the stop criterion of the BB algorithm is

$$\nu_k - \beta(M_k) \leq \varepsilon. \tag{22}$$

The main features of the simplicial and conical techniques discussed hereafter are the following.

- They take advantage of the fact that the branching operation needs only be performed with respect to the "nonconvex variables" which evolve in a space of lower dimension than the full decision vector.
- They exploit the concave nature of the functional as well as the convexity of the constraint set for the computation of bounds.
- They also make use of the fact $f(x) \geq 0,\ \forall x \in X$, so that any portion of the space of the "nonconvex variables" where f is bounded below by a positive value is irrelevant and in addition, the stopping criterion (22) is replaced with

$$\nu^k \leq \varepsilon. \tag{23}$$

- The branching operation is devised so that the situation (23) is brought about likely as rapidly as possible.

As we shall see, these techniques are tailored to the particular properties of the problem to make them far more efficient than in the general case.

5.1 Simplicial algorithm

In this subsection, we discuss in more detail the simplicial algorithm together with its convergence properties.

Bounding operation: Recall that N is the dimension of $\mathcal{Z}$. For every affine function $\phi(x)$, the function $f(x) - \phi(x)$ is still concave and as a consequence, its maximum over a simplex M is attained on the vertex set vert$\,(M)$ of M, i.e. $\phi(x) \leq f(x),\ \forall x \in \text{vert}\,(M) \Rightarrow \phi(x) \leq f(x),\ \forall x \in M$. Thus in every simplex M with vertices $u^1, u^2, \ldots, u^{N+1}$ in $\mathcal{Z}$ the affine function $\phi_M(x)$ defined for every $x = \sum_{i=1} \lambda_i u^i,\ \lambda_i \geq 0, \sum_{i=1}^{N+1} \lambda_i = 1$ by

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

obviously matches f at all vertices of M and satisfies $\phi(x) = \sum_{i=i}^{N+1} \lambda_i f(u^i) \leq f(x)$ for every $x \in M$. Thus $\phi(x)$ is an affine minorant of f in M (in fact the convex envelope of $f(x)$ over M). We then have that

$$\begin{aligned}\beta(M) \quad &:= \quad \min\{\phi_M(x) : x \in M \cap \mathcal{X}\} \\ &= \quad \min\{\phi_M(\sum_{i=1}^{N+1} \lambda_i u^i) :\ \sum_{i=1}^{N} \lambda_i = 1,\ \lambda_i \geq 0,\ \sum_{i=1}^{N} \lambda_i u^i \in \mathcal{X}\} \\ &\leq \quad \min\{f(x) :\ x \in M \cap \mathcal{X}\}.\end{aligned} \tag{24}$$

and therefore $\beta(M)$ is a lower bound of $f(x)$ over $M \cap \mathcal{X}$. It is worth noticing that the minimization problem in (24) is a convex program since ϕ_M is affine and the constraint set $\mathcal{X}$ is convex by hypothesis. Also, if $f(u^i) > 0$ for every $i = 1, \ldots, N+1$ then by concavity of $f(x)$ it follows immediately that $f(x) > 0,\ \forall x \in M$, i.e. for our purpose M can be discarded from further consideration.

Branching: Let M_k be the simplex chosen for subdivision at iteration k and $\omega(M_k)$ be the optimal solution of problem (24) with $M = M_k$, i.e. $\omega(M_k) \in M_k \cap \mathcal{X}$ and $\phi_{M_k}(\omega(M_k)) = \beta(M_k)$. Note that, as $\omega(M_k)$ is feasible, we must have $\nu_k \leq f(\omega(M_k))$, so if it so happens that $\omega(M_k) \in \text{vert}(M_k)$ then $\beta(M_k) = \phi_M(\omega(M_k)) = f(\omega(M_k)) = \nu_k$ and therefore $\beta(M_k)$ will be the exact minimum of f over $\mathcal{X}$ and according to the stop criterion (22) 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* [36], has long been used [35] and is known to work well in practice though its theoretical convergence is still an open question [36]. 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 [36, Th. 5.1].

Lemma 5.1 *Let $\{M_k\}$, $k = 0, 1, 2, \ldots$ 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 $M_\infty = \cap_{k=1}^{\infty} M_k$.*

Algorithm organization and convergence properties: Keeping in mind that the algorithm will stop when the current best value is 0 or there is evidence that the lower bound of (16) is positive (infeasibility), we can state the simplicial algorithm as follows:

Step 0. (*Initialization*) In the $\mathcal{Z}$-space take an N-simplex $M_0 \supset \mathcal{X}$ such that f is still concave on M_0. Let x^0 be an initial feasible point (the best available), $\nu_0 = f(x^0)$, $\mathcal{S}_0 = \{M_0\}$, $\mathcal{P}_0 = \mathcal{S}_0$, $k = 0$.

Step 1. (*Bounding*) For each simplex $M = [u^1, \ldots, u^{N+1}] \in \mathcal{P}_k$ if $f(u^i) > 0$, $\forall i$, then set $\beta(M) > 0$; otherwise, compute

$$\beta(M) = \min\{\sum_{i=1}^{N+1} \lambda_i f(u^i) : \sum_{i=1}^{N+1} \lambda_i u^i \in \mathcal{X},\ \sum_{i=1}^{N+1} \lambda_i = 1,\ \lambda_i \geq 0\} \quad (25)$$

and let $\omega(M) = \sum_{i=1}^{N+1} \lambda_i(M) u^i$ be an optimal solution of this convex program.

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

Step 5. (*Branching*) Select $M_k \in \operatorname{argmin}\{\beta(M) |\ M \in \mathcal{R}_k\}$. Subdivide M_k according to a chosen normal rule (bisect M_k or split it through $\omega(M)$). 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.

The convergence properties of the simplicial algorithm are clarified by the following theorem.

Theorem 5.2 *Either the simplicial algorithm terminates after finitely many iterations, yielding a zero optimal solution of (16) (termination at Step 2) or providing evidence that (16) has no zero optimal solution (termination at Step 4). Or it generates an infinite sequence of feasible solutions ω^k converging to a zero optimal solution.*

The conical algorithm which makes use of subdivision into cones is discussed in [4].

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 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 1. 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 [21].

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 [21].

$$\begin{aligned} \dot{x} &= Ax + B_\Delta w_\Delta + Bu \\ z_\Delta &= C_\Delta x \\ w_\Delta &= \Delta z_\Delta, \quad \Delta := \begin{bmatrix} r_y & 0 & 0 \\ 0 & \varphi_2 & 0 \\ 0 & 0 & \varphi_2 \end{bmatrix} \end{aligned}$$

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 $\Delta = \operatorname{diag}(\delta_1, \delta_2 I_2)$ with $|\delta_i| \leq 1$, $i = 1, 2$.

The synthesis structure used to achieve the design requirements is shown in Figure 2. It simply translates performance tracking ($\omega_I x_I$) and high-frequency gain attenuation ($\omega_d \dot{r}_x$).

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 [4].

Table 1 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* [11]. 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. See [4].

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 and larger problems

The algorithms have also been shown to perform well on a set of random and larger size problems (up to 30 states and 5 repeated parameters) [4].

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. 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.

Acknowledgment to Professor Hoang Tuy and Dr. Hiroyuki Kajiwara

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] P. Apkarian and H. D. Tuan, *Concave Programming in Control Theory*, (1998). to appear in Jour. of Global Optimization.

[4] ———, *Robust Control via Concave Minimization - Local and Global Algorithms*, (1998). submitted to IEEE Trans. on Automatic Control.

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

[6] 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.

[7] 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.

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

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

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

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

[12] 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.

[13] 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.

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

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

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

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

[18] T. Iwasaki, *Robust Stability Analysis with Quadratic Separator: Parametric Time-Varying Uncertainty Case*, in Proc. IEEE Conf. on Decision and Control, San Diego,USA, 1997, pp. 4880–4885.

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

[20] 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.

[21] 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.

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

[23] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass., 2nd ed., 1984.

[24] O. L. Mangasarian and J. S. Pang, *The Extended Linear Complementary Problem*, SIAM J. on Matrix Analysis and Applications, 16 (1995), pp. 359–368.

[25] 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.

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

[27] 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.

[28] 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.

[29] E. Polak, *Computational Methods in Optimization*, Academic Press, New York, 1971.

[30] 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.

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

[32] 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.

[33] 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.

[34] H. D. Tuan, S. Hosoe, and H. Tuy, *Global Optimization for Robust Performance Problem in Robust Controls*, (1997). Submitted to Automatica for publication.

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

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

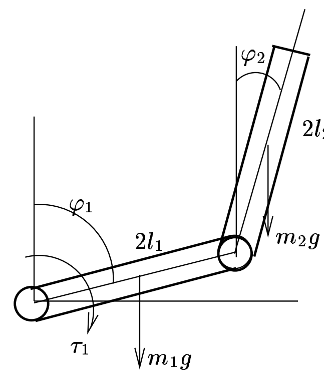

Figure 1: Inverted pendulum

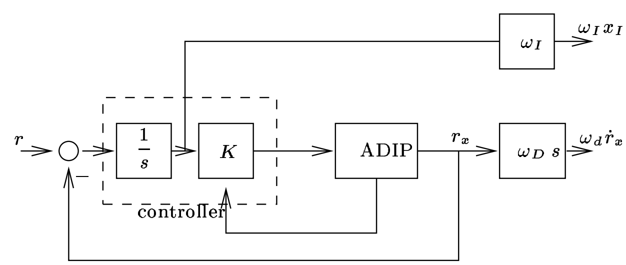

Figure 2: Synthesis structure for the inverted pendulum

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

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