A Prototype Primal-Dual LMI-Interior Algorithm for Nonconvex Robust Control Problems

Pierre Apkarian [*]   *Dominikus Noll* [‡]

Abstract

We present a new method for solving a broad class of problems in robust synthesis. The proposed method encompasses problems such as minimizing a nonlinear cost function subject to Linear Matrix Inequality (LMI) constraints and is easily generalized to harder problems where LMIs are replaced with Bilinear Matrix Inequalities (BMIs). At the core of the proposed method, we take advantage of a primal-dual formulation of the optimality conditions to construct a quadratic approximate model that is refined iteratively to solve the original (hard) problem. More specifically, the method consists in solving a sequence of indefinite quadratic programming problems in the primal space according to a trust region strategy followed by an appropriate updating scheme for the dual variables. This is an interior-point method in the sense that feasibility is maintained for both primal and dual variables. The potentials of this new method are evaluated through robust synthesis examples.

Keywords : interior-point methods, primal-dual algorithms, SDP, LMI techniques, BMI, robust synthesis.

1 Introduction

It is now widely accepted that LMI methods provide a powerful framework for formulating and solving problems in robust control theory [10]. Significant practical examples include:

- Lyapunov stability and performance analysis [20, 12],
- analysis of dynamic systems subject to IQC-defined (Integral Quadratic Constraint) components such as parametric uncertainties, delays, signal saturations [33, 19, 5].
- H_∞, H_2 and multi-channel/objective syntheses or relaxed variants [22, 30, 40].

[*]ONERA-CERT, Centre d'études et de recherche de Toulouse, Control System Department, 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 .

[‡]UPS-MIP, Mathmatiques pour l'Industrie et la Physique, 118, route de Narbonne, 31062 Toulouse, FRANCE - Email : `noll@dumbo.ups-tlse.fr` - Tel : +33 5.61.55.67.78 - Fax : +33 5.61.55.61.83 .

- gain-scheduling control via LPV (Linear Parameter-Varying) representations of the plant [35, 1, 39].

This list is far from being exhaustive and further examples can be found in recent control journals and conferences. Unfortunately, there are numerous problems of considerable practical impact for which an LMI formulation is not possible or remains unknown. A few samples are

- fixed- or reduced-order and decentralized syntheses,
- robust synthesis with different classes of scalings/multipliers or parameter-dependent Lyapunov functions,
- mixed control problems with both scheduled and uncertain parameters,
- simultaneous system structure and controller design,
- and combinations of the above.

These more delicate problems can be cast as either minimizing a linear objective subject to nonlinear equality constraints in tandem with LMIs:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & h(x) = 0 \\ & \mathcal{A}(x) \succeq 0\,, \end{array} \tag{1}$$

or, and this seems to be the most general situation, as minimizing a linear objective subject to BMIs

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \mathcal{B}(x) \succeq 0\,. \end{array} \tag{2}$$

In the previous formulations, x is the decision vector, $h(x)$ is a nonlinear vector-valued function, and $\mathcal{A}(x)$ and $\mathcal{B}(x)$ are linear, respectively bilinear symmetric matrix-valued expressions in x:

$$\begin{aligned} \mathcal{A}(x) &:= A_0 + \sum_{i=1}^{N} x_i A_i \\ \mathcal{B}(x) &:= A_0 + \sum_{i=1}^{N} x_i A_i + \sum_{1\le i<j\le N} x_i x_j B_{ij}\,. \end{aligned} \tag{3}$$

Recently, various interesting attempts have been made to solve at least locally these difficult problems. In [26], the authors use alternating projections to solve reduced-order design problems. A similar scheme which exploits two equivalent formulations of the fixed-order control problem is proposed in [29]. A successive linearization algorithm for static output feedback design is discussed in [25]. In [24] the authors develop a specialized algorithm again for static output feedback synthesis. A major issue in these works, as well as with customary $D-K$ iteration schemes, is that it appears delicate to establish global convergence to a local solution, that is convergence to a local solution for any even remote starting point. Also, the rate of convergence remains unknown in the vicinity of a local solution. A discussion on the difficulties encountered with $D-K$ schemes is provided in [28].

New potent and direct nonlinear-programming-based methods have been developped until recently. In [31], Jarre proposes a primal method which constructs a sequence of iterates through quadratic subproblems including a trust region strategy. A similar idea is developped by Leibfritz and Mostafa in [32]. They use an SQP (Sequential Quadratic Programming) scheme with trust regions applied to an appropriately defined barrier function to solve nonlinear SDP problems.

In [17], we discuss an algorithm for solving a class of robust synthesis problems which overcomes the above difficulties. The proposed method is of augmented Lagrangian type and iteratively constructs and refines approximations of the original problem. These approximations consist of quadratic programs suitably convexified in conjunction with LMI constraints, hence are readily solved using current SDP codes. In [18], we expand on our preliminary ideas and take advantage of duality to attain better local convergence rates. We used the terminology Succesive Semi-Definite Programming (SSDP) to refer to this class of algorithms since SDP is run sequentially on LMI constrained quadratic problems to approximate a local solution from the interior of the LMI feasible set.

For other reasons, however, we think a more direct approach is desirable. Previous techniques, though supported by highly reliable computational codes , hinge on a sequence of SDPs which entail overcomputations. An accurate solution of each individual SDP is, indeed, neither required nor recommended as they tend to drive the iterates close to the boundary of the feasible region. Also, these techniques suppose some sort of separation of the variables as in $D-K$ schemes or approximation with (convex) SDPs and consequently are unable to exploit directions of negative curvatures or concavity of functions or constraints. This weakness also has a direct impact on the efficiency on the method. The method in this work does not assume separation, nor rely on convex approximations. Progress is generated even when concavity is encountered and hence efficiency practically benefits from this feature.

The notation used throughout the paper is fairly standard. M^T is the transpose of the matrix M. The notation $\operatorname{Tr} M$ stands for the trace of M. For Hermitian or symmetric matrices, $M \succ N$ means that $M-N$ is positive definite and $M \succeq N$ means that $M-N$ is positive semi-definite. An LMI constraint is defined as

$$\mathcal{A}(x) := A_0 + \sum_{i=1}^{N} x_i A_i \succeq 0\,,$$

and its homogeneous part will be denoted $\mathcal{A}_*(x)$

$$\mathcal{A}_*(x) := \sum_{i=1}^{N} x_i A_i\,.$$

Also, for an LMI $\mathcal{A}(x) \succeq 0$, it is interesting, for computational efficiency reasons, to introduce the notation A, a matrix representation of the operator $\mathcal{A}(.)$, that is,

$$A = [\,\mathrm{col}(A_1), \mathrm{col}(A_2), \ldots, \mathrm{col}(A_N)\,]\,, \tag{4}$$

where col is the usual column operation on a matrix.

$\|M\|_F$ denotes the Frobenius norm of the matrix M. The notation $A \otimes B$ designates the Kronecker product of A and B and $\operatorname{vec} M$ stands for the column-wise vectorization operation on a matrix M. Finally, the gradient of a real-valued function $\Phi(x)$ is denoted $\nabla\Phi(x)$ and its Hessian $\nabla^2\Phi(x)$. We shall also use the notation x_+ to refer to the value of the variable x, matrix or vector, at the next iteration.

2 Nonlinear cost with LMI constraints

The focus of this section is on problems that can be cast as minimizing a nonlinear cost function subject to LMI constraints

$$\mathbf{P1} \qquad \begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & \mathcal{A}(x) \succeq 0\,, \end{array} \tag{5}$$

where $\mathcal{A}(x)$ is as in (3) and $\phi(x)$ is a twice continuously differentiable nonlinear function of x.

As will become clear in the subsequent derivations, this class of program is instrumental and will provide basic tools for the more practically interesting situation considered in Section 3. Also, it is worth mentioning that all arguments in the sequel generalize to the case where LMIs are replaced with BMIs, that is $\mathcal{A}(x)$ is replaced with $\mathcal{B}(x)$, c.f. (3).

A primal method for solving **P1** consists in introducing the barrier problem

$$\min_x \;\; F(x,\mu) := \phi(x) - \mu \log \det \mathcal{A}(x)\,, \tag{6}$$

which is well defined as long as the decision vector x belongs to the LMI feasible set. The primal method then generates a sequence of feasible iterates $x(\mu_k)$, local solutions to (6) for $\mu = \mu_k$, for a sequence of barrier parameters $\mu_k > 0$, whose limiting value is zero.

Primal interior-point method

Step 0 [*initialize*] Choose initial $\mu > 0$, a feasible x and an initial $\varepsilon > 0$.

Step 1 [*inner steps*] Given x, μ and ε, compute an approximate local solution x_+ to (6) satisfying

$$\|\nabla F(x_+, \mu)\| \leq \varepsilon$$

Step 2 [*update*] Update the parameters μ and ε to μ_+ and ε_+, respectively, so that they form strictly decreasing sequences converging to zero. Return to Step 1.

Note that the inner steps involve constructing quadratic approximations of the function $F(x, \mu)$ and generating steps using either a modified Newton method or a trust region strategy. See, for instance, [8] for a general treatment of the latter techniques. It has been demonstrated that this scheme is guaranteed to converge to a local solution under very mild assumptions [21, 37]. In spite of this simplicity, primal barrier methods are often delicate to implement in practice. More precisely, the Hessian matrix (see Appendix A) is given as

$$\nabla^2(x, \mu) := \nabla^2 \phi(x) + \mu A^T (\mathcal{A}(x)^{-1} \otimes \mathcal{A}(x)^{-1}) A\,, \tag{7}$$

where A is defined in (4). The second term in this expression becomes dominant in the course of the algorithm as the iterates come closer to the LMI boundary, and tends to generate distorted search directions. Also, as μ tends to zero, the underlying quadratic problem defining the Newton step becomes increasingly ill-conditioned. A consequence is that the starting point for each outer iteration is required to be more and more accurate which may, and often will, thwart the nice theoretical convergence properties of the method. Facing these difficulties, a number of safeguarding techniques are necessary for a reliable implementation of the primal method. A comprehensive discussion on that matter is provided in the expository paper [15] in the restricted case of classical vector inequalities. These safeguarding techniques include extrapolation to improve the starting point of each outer iteration and also primal-dual formulations to bypass the difficulties due to ill-conditioning. This is what we consider hereafter. We extend the work of [11] to the SD cone and show that similar constructions are possible.

Before going further, let us write down the first-order optimality conditions for the program (6). This yields

$$\nabla\phi(x) - \mu \begin{bmatrix} \operatorname{Tr}\mathcal{A}(x)^{-1}A_1 \\ \operatorname{Tr}\mathcal{A}(x)^{-1}A_2 \\ \vdots \\ \operatorname{Tr}\mathcal{A}(x)^{-1}A_N \end{bmatrix} = 0 \tag{8}$$

Alternatively, the Lagrangian function for **P1** in (5) is described as

$$\mathcal{L}(x, Z) := \phi(x) - \operatorname{Tr} Z\mathcal{A}(x)\,,$$

and the corresponding stationarity conditions are given as

$$\nabla\phi(x) - \begin{bmatrix} \operatorname{Tr} ZA_1 \\ \operatorname{Tr} ZA_2 \\ \vdots \\ \operatorname{Tr} ZA_N \end{bmatrix} = 0 \tag{9}$$

$$Z\mathcal{A}(x) = 0 \tag{10}$$

$$\mathcal{A}(x) \succeq 0, \qquad Z \succeq 0 \tag{11}$$

where Z is the dual (Lagrange multiplier) variable. The reader is referred to [41] for additional details[1]. The first of these conditions is the standard stationarity conditions, while (10) and (11) are the complementary and the primal-dual feasibility conditions, respectively. Direct examination shows that critical points of pure primal problems parameterized by μ correspond to points satisfying the *perturbed* stationarity conditions in the original problem (5) defined as (9), (11) and

$$Z\mathcal{A}(x) = \mu I\,. \tag{12}$$

The idea of primal-dual algorithms is therefore to base the iteration scheme on (9), (11) and (12) in place of (7) and (8). Following this idea, a quadratic model for problem (5) about the iterate x is obtained as

$$\mathcal{M}(x+d, \mu) := F(x, \mu) + (\nabla\phi(x) - \mu A^T \mathrm{vec}\,\mathcal{A}(x)^{-1})^T d + \frac{1}{2} d^T \left[\nabla^2\phi(x) + A^T(Z \otimes \mathcal{A}(x)^{-1})A\right] d\,, \tag{13}$$

[1]Similar conditions are valid for BMIs whenever the Mangasarian-Fromovitz constraint qualification holds, see [41]. The latter are always satisfied for LMI constraints provided that they admit a non-empty interior.

where we have used the fact that

$$\begin{bmatrix} \mathrm{Tr}\, MA_1 \\ \mathrm{Tr}\, MA_2 \\ \vdots \\ \mathrm{Tr}\, MA_N \end{bmatrix} = A^T \mathrm{vec}\,(M^T) = A^T \mathrm{vec}\,(M)\,,$$

for any matrix M. Note that d is the step or displacement from x and that the model (13) is a true approximation of the quadratic model of the purely primal problem. This approximation is more accurate when the dual variable Z comes closer to $\mu\mathcal{A}(x)^{-1}$. According to our analysis, the displacement in the primal space can be determined using a modified Newton method or a trust region strategy applied to the quadratic model (13). On the other hand, Z should satisfy dual feasibility and the complementarity condition in order to meet the necessary criticality conditions of a local solution. This will determine the displacement in the dual space.

The displacement in the dual space can be determined by applying Newton method to the conditions (9) and (12). This is obtained by calculating a first-order approximation of the relationships

$$\nabla\phi(x+d) - \begin{bmatrix} \mathrm{Tr}\,(Z+\Delta Z)A_1 \\ \mathrm{Tr}\,(Z+\Delta Z)A_2 \\ \vdots \\ \mathrm{Tr}\,(Z+\Delta Z)A_N \end{bmatrix} = 0$$

$$\mathcal{A}(x+d)(Z+\Delta Z) = \mu I$$

which gives

$$\nabla^2\phi(x)d - \begin{bmatrix} \mathrm{Tr}\,\Delta Z A_1 \\ \mathrm{Tr}\,\Delta Z A_2 \\ \vdots \\ \mathrm{Tr}\,\Delta Z A_N \end{bmatrix} = -\nabla\phi(x) + \begin{bmatrix} \mathrm{Tr}\, Z A_1 \\ \mathrm{Tr}\, Z A_2 \\ \vdots \\ \mathrm{Tr}\, Z A_N \end{bmatrix}, \tag{14}$$

and

$$\Delta Z \mathcal{A}(x) + Z\mathcal{A}_*(d) = -Z\mathcal{A}(x)\,. \tag{15}$$

Note that due to the lack of symmetry in the equation (15), the dual displacement ΔZ is not generally symmetric and thus cannot be considered an acceptable step. As is common in primal-dual SDP, we shall use a symmetrization procedure to enforce symmetry of the dual step. There are many ways to carry out the symmetrization of the complementary conditions and a comprehensive analysis of these techniques is given in [42] for SDP. In this paper, we have used a dual search direction proposed by Monteiro and discussed in length by Zhang in [43], as it leads to simple formulas that can be efficiently computed. Monteiro-Zhang's symmetrization is based on the formulas

$$\mathcal{A}(x)(Z+\Delta Z)\mathcal{A}(x+d) + (\mathcal{A}(x)(Z+\Delta Z)\mathcal{A}(x+d))^T = 2\mu\mathcal{A}(x)\,.$$

This leads to the Newton step

$$\Delta Z = -Z + \mu\mathcal{A}(x)^{-1} - \frac{1}{2}\left[Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1} + (Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1})^T\right]\,, \tag{16}$$

and thus the updating expression

$$Z_+ = \mu\mathcal{A}(x)^{-1} - \frac{1}{2}\left[Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1} + (Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1})^T\right] . \tag{17}$$

Then, by eliminating ΔZ in (14), we infer the primal-dual Newton equations:

$$\left[\nabla^2\phi(x) + A^T(Z \otimes \mathcal{A}(x)^{-1})A\right] d = -\nabla\phi(x) + \mu A^T \mathrm{vec}\,\mathcal{A}(x)^{-1} , \tag{18}$$

$$Z_+ = \mu\mathcal{A}(x)^{-1} - \frac{1}{2}\left[Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1} + (Z\mathcal{A}_*(d)\mathcal{A}(x)^{-1})^T\right] . \tag{19}$$

It is interesting to mention that the primal step d can be generated either by using the quadratic model (13) or the Newton equation (18). The quadratic programming formulation is generally preferable as it guarantees descent in the barrier function (6) and global convergence is guaranteed provided that the trust region radius (if this option is taken) is appropriately selected at each iteration. To sum up, the proposed method is described below.

Primal-dual interior-point method with trust regions

Step 0 [*initialize*] Fix parameters $0 < \gamma_1 < 1 \leq \gamma_2$, η_1, η_2, $\mu_{\min}$ and function $\varepsilon(\mu)$. Initialize μ and feasible x, Z.

Step 1 [*inner steps*] For given μ and Z perform the following steps.

Step 1.1 [*primal step with trust regions*] Choose trust region radius $r > 0$ and solve quadratic program

$$\begin{array}{l} \min\ \mathcal{M}(x+d,\mu) \\ \|d\| \leq r \end{array}$$

Step 1.2 [*acceptance of primal step*] If $\mathcal{A}(x+d) \succeq 0$ compute

$$\rho = \frac{F(x,\mu) - F(x+d,\mu)}{\mathcal{M}(x,\mu) - \mathcal{M}(x+d,\mu)} ,$$

else set $\rho = -\infty$.
If $\rho \geq \eta_1$ (retain the step) $x_+ = x + d$ and update Z to Z_+ according to (19), otherwise $x_+ = x$ and $Z_+ = Z$.

Step 1.3 [*update radius*]

$$r_+ = \begin{cases} \gamma_1\|d\| & \text{if} \quad \rho \leq \eta_1 \\ \gamma_2\|d\| & \text{if} \quad \rho \geq \eta_2 \\ r & \text{if} \quad \eta_1 \leq \rho \leq \eta_2 \, . \end{cases}$$

Step 1.4 [*stationarity stop test*] Stop if

$$\max\left\{\|Z_+\mathcal{A}(x_+) - \mu I\|, \|\nabla\phi(x_+) - A^T\mathrm{vec}\,Z_+\|\right\} \leq \varepsilon(\mu) ,$$

else go to Step 1.1.

Step 1 [*update*] Update μ to μ_+. Stop if $\mu \leq \mu_{\min}$ else update go to Step 1.

Note that a step d is retained whenever it is feasible, $\mathcal{A}(x+d) \succeq 0$, and there is a good agreement between the quadratic predicted decrease and the decrease in the barrier function $\rho \geq \eta_1$. The update of the trust region radius obeys a standard rule [38].

2.1 Exploiting Dikin's geometry

It may occur that due to the elongated shape of the feasible set $\mathcal{A}(x) \succeq 0$, iterates get stuck to the boundary and unduly short steps take place. As is well-known in affine-scaling linear programming methods, one can overcome this difficulty by rescaling the step coordinates in a basis that, at least locally, captures the geometry of the feasible region. For the SD cone, an interesting coordinate system is that given by Dikin's ellipsoid [13, 9, 23]. It defines a region in the feasible set where the current iterate can move without losing feasibility and is described as

$$d^T A^T (\mathcal{A}(x)^{-1} \otimes \mathcal{A}(x)^{-1}) A d \leq 1\,.$$

Up to a scalar multiple, the latter can be approximated by

$$d^T A^T (Z \otimes \mathcal{A}(x)^{-1}) A d \leq 1\,,$$

which is nothing else but the second term in the Hessian matrix (13) and (18). The scaling defined by the new coordinate system is given as

$$\widetilde{d} = Rd,$$

where R Cholesky factorizes $A^T (Z \otimes \mathcal{A}(x)^{-1}) A$. The new trust region problem is therefore

$$\begin{array}{l} \min\ (\nabla\phi(x) - \mu A^T \mathrm{vec}\, \mathcal{A}(x)^{-1})^T R^{-1} \widetilde{d} + \frac{1}{2} \widetilde{d}^T R^{-T} \left[\nabla^2 \phi(x) + A^T (Z \otimes \mathcal{A}(x)^{-1}) A\right] R^{-1} \widetilde{d}\,, \\ \text{subject to } \ \|\widetilde{d}\| \leq r\,, \end{array}$$

while the rest of the algorithm remains unchanged.

3 Nonlinear equality and LMI constraints

In this section, we investigate a further complication of **P1** in (5) in which LMI constraints are in tandem with nonlinear equality constraints:

$$\mathbf{P2} \qquad \begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & h(x) = 0, \\ & \mathcal{A}(x) \succeq 0\,. \end{array} \tag{20}$$

The second equation in this formulation determines the nonlinear constraints and the function h from $\mathbf{R}^N$ to $\mathbf{R}^m$ is twice continuously differentiable in its argument over some open set containing $\{x :\ \mathcal{A}(x) \succeq 0\}$.

A straightforward extension of the primal method in (6) leads us to the penalty/barrier problem

$$\min_x \ F(x,\mu) := \phi(x) + \frac{c_0}{2\,\mu}\, \|h(x)\|^2 - \mu \log \det \mathcal{A}(x)\,. \tag{21}$$

As before, this function incorporates a barrier term to secure LMI feasibility of iterates but also a penalty term $\frac{c_0}{2\,\mu}\,\|h(x)\|^2$ which asymptotically drives the iterates towards the algebraic manifold $h(x) = 0$. Note that penalty and barrier parameters are not independent as one could expect but are inverse of each other up to a scalar multiple. The benefit of this restriction is mainly a simplification in the algorithm since only one parameter needs to be updated at each outer iteration.

Using a primal method for solving (21) for a decreasing sequence μ_k, leads to the difficulties already discussed in Section 2. It is therefore advisable to introduce dual SD cone variables to improve the algorithm numerical behavior. This can be accomplished in exactly the same way as for the simpler problem considered in Section 2 and thereby a similar method can be constructed. This method, however, would not completely cope with the difficulties associated with the penalty term which also gives rise to complications when μ is small. It is well known that the latter can be alleviated by augmenting the primal function with Lagrangian terms associated with equality constraints. This method is referred to as the Augmented Lagrangian method and a detailed treatment is given in Bertsekas's master book [7]. In place of the penalty/ barrier function in (21), we therefore consider the modified program

$$\min_x \quad F(x, \lambda, \mu) := \phi(x) + \lambda^T h(x) + \frac{c_0}{2\,\mu}\, \|h(x)\|^2 - \mu \log \det \mathcal{A}(x)\,, \tag{22}$$

where λ are Lagrange multipliers associated with equality constraints. Also, we update the multipliers using the first-order classical rule (see [7, 17] for details):

$$\lambda_+ = \lambda + c_0/\mu\; h(x)\,. \tag{23}$$

The associated quadratic model about the point x is given as

$$\mathcal{M}(x + d, \lambda, \mu) := F(x, \lambda, \mu) + g(x)^T d + \frac{1}{2} d^T \left[H(x)\right] d\,, \tag{24}$$

where

$$g(x) := \nabla\phi(x) + \nabla h(x)\lambda + \frac{c_0}{\mu}\nabla h(x)\, h(x) - \mu A^T \mathrm{vec}\, \mathcal{A}(x)^{-1},$$

and

$$H(x) := \nabla^2\phi(x) + \sum_{i=1}^{m} \nabla^2 h_i(\lambda_i + \frac{c_0}{\mu} h_i) + \frac{c_0}{\mu}\nabla h(x)\nabla h(x)^T + A^T(Z \otimes \mathcal{A}(x)^{-1})A\,.$$

The outlined strategy is made more precise in the following algorithm. For simplicity, we introduce the definition:

$$\psi(x) := \phi(x) + \lambda^T h(x) + \frac{c_0}{2\,\mu}\, \|h(x)\|^2\,.$$

Primal-dual interior-point method
with equality constraints and trust regions

Step 0 [*initialize*] Fix parameters c_0, $0 < \gamma_1 < 1 \leq \gamma_2$, η_1, η_2, $\mu_{\min}$, ε_2 and a function $\varepsilon_1(\mu)$. Initialize μ, λ and feasible x, Z, i.e. $\mathcal{A}(x) \geq 0$, $Z \geq 0$.

Step 1 [*inner steps*] For given μ perform the steps.

Step 1.1 [*primal step with trust regions*] Solve the quadratic program

$$\begin{array}{l}\min\ \mathcal{M}(x+d,\lambda,\mu)\\ \|d\|\le r\end{array}$$

Step 1.2 [*acceptance of primal step*] If $\mathcal{A}(x+d)\succeq 0$ compute

$$\rho=\frac{F(x,\lambda,\mu)-F(x+d,\lambda,\mu)}{\mathcal{M}(x,\lambda,\mu)-\mathcal{M}(x+d,\lambda,\mu)},$$

else set $\rho=-\infty$.
If $\rho\ge\eta_1$ (retain the step) $x_+=x+d$, update Z to Z_+, otherwise $x_+=x$ and $Z_+=Z$.

Step 1.3 [*update radius*]

$$r_+=\begin{cases}\gamma_1\|d\| & \text{if} \quad \rho\le\eta_1\\ \gamma_2\|d\| & \text{if} \quad \rho\ge\eta_2\\ r & \text{if} \quad \eta_1\le\rho\le\eta_2\,.\end{cases}$$

Step 1.4 [*stationarity stop test*] Stop the inner iterations if

$$\max\left\{\|Z_+\mathcal{A}(x_+)-\mu I\|,\ \|\nabla\psi(x_+)-A^T\mathrm{vec}\,Z_+\|\right\}\le\varepsilon_1(\mu)\,, \tag{25}$$

else return to Step 1.1.

Step 1 [*update*] Stop if $\mu\le\mu_{\min}$ and $\|h(x)\|<\varepsilon_2$ else

$$\begin{array}{ll}\text{update } \lambda \text{ to } \lambda_+ & \text{if } h(x_+)<\beta\,h(x),\\ \text{update } \mu \text{ to } \mu_+ & \text{otherwise}\\ \text{and go to Step 1}\,. & \end{array}$$

The inner steps of the algorithm are nearly the same as those in Section 2. The algorithm stops when both the barrier parameter and the nonlinear constraints are small enough. Indeed, we know from our earlier studies [3, 17] that the nonlinear equality constraints do not need to be satisfied exactly and that perturbations techniques can be applied to end up on the algebraic variety $h(x)=0$. Also, we only decrease μ when the decrease in the equality constraints is not satisfactory, otherwise we update the Lagrange multipliers λ associated with equality constraints in accordance with the first-order rule in (23).

4 Robust synthesis

In this section, we show how the proposed techniques can be used for robust control of LFT (Linear Fractional Transformation) uncertain systems. The uncertain plant is described as

$$\begin{array}{rcl}\begin{bmatrix}\dot{x}\\ z_\Delta\\ z\\ y\end{bmatrix} & = & \begin{bmatrix}A & B_\Delta & B_1 & B_2\\ C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2}\\ C_1 & D_{1\Delta} & D_{11} & D_{12}\\ C_2 & D_{2\Delta} & D_{21} & 0\end{bmatrix}\begin{bmatrix}x\\ w_\Delta\\ w\\ u\end{bmatrix}\\ w_\Delta & = & \Delta(t)\,z_\Delta\,,\end{array} \tag{26}$$

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

$$\Delta(t) = \operatorname{diag}(\ldots, \delta_i(t)I, \ldots, \Delta_j(t), \ldots) \in \mathbf{R}^{L\times L} \tag{27}$$

and normalized such that

$$\Delta(t)^T\Delta(t) \le I, \qquad t \ge 0\,. \tag{28}$$

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 [16, 14]. 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 (26)-(28) the robust control problem consists in seeking a Linear Time-Invariant (LTI) controller

$$\begin{array}{rcl} \dot{x}_K &=& A_K x_K + B_K y\,, \\ u &=& C_K x_K + D_K y\,, \end{array} \tag{29}$$

such that

- the closed-loop system (26)-(28) and (29) 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 (28).

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 (27). Denoting this structure as $\boldsymbol{\Delta}$, the following scaling sets can be introduced. The set of block-structured scalings associated with the parameter structure $\boldsymbol{\Delta}$ is defined as

$$W_{\boldsymbol{\Delta}} := \{W : \quad W\Delta = \Delta W, \quad \forall \Delta \text{ with structure } \boldsymbol{\Delta}\}\,.$$

With the above definitions and notations in mind, it is now well-known that such problems can be handled via a suitable generalization of the Bounded Real Lemma [35, 1, 2] which expresses as the existence of a Lyapunov matrix, a scaling W and controller state-space data satisfying BMI constraints. Then, using the Projection Lemma [22] as the basic tool for reducing nonconvex terms and variables the following algebraically constrained LMI characterization for the solvability of the problem can be established. See [27, 36, 4] for proofs and insights on problems in the same class.

Theorem 4.1 *Consider the LFT plant governed by (26) and (28) with Δ assuming a block-diagonal structure as in (27). 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 a pair of symmetric matrices (X, Y) and a pair of scalings $(W, \widetilde{W})$ such that the structural constraints*

$$W,\ \widetilde{W} \in W_{\boldsymbol{\Delta}} \tag{30}$$

hold and the LMIs

$$\mathcal{N}_X^T \begin{bmatrix} A^T X + XA & XB_\Delta + C_\Delta^T T^T & XB_1 & C_\Delta^T S & C_1^T \\ B_\Delta^T X + TC_\Delta & -S + TD_{\Delta\Delta} + D_{\Delta\Delta}^T T^T & TD_{\Delta 1} & D_{\Delta\Delta}^T S & D_{1\Delta}^T \\ B_1^T X & D_{\Delta 1}^T T^T & -\gamma I & D_{\Delta 1}^T S & D_{11}^T \\ SC_\Delta & SD_{\Delta\Delta} & SD_{\Delta 1} & -S & 0 \\ C_1 & D_{1\Delta} & D_{11} & 0 & -\gamma I \end{bmatrix} \mathcal{N}_X \prec 0, \tag{31}$$

$$\mathcal{N}_Y^T \begin{bmatrix} AY + YA^T & YC_\Delta^T + B_\Delta \Gamma^T & YC_1^T & B_\Delta \Sigma & B_1 \\ C_\Delta Y + \Gamma B_\Delta^T & -\Sigma + \Gamma D_{\Delta\Delta}^T + D_{\Delta\Delta}\Gamma^T & \Gamma D_{1\Delta}^T & D_{\Delta\Delta}\Sigma & D_{\Delta 1} \\ C_1 Y & D_{1\Delta}\Gamma^T & -\gamma I & D_{1\Delta}\Sigma & D_{11} \\ \Sigma B_\Delta^T & \Sigma D_{\Delta\Delta}^T & \Sigma D_{1\Delta}^T & -\Sigma & 0 \\ B_1^T & D_{\Delta 1}^T & D_{11}^T & 0 & -\gamma I \end{bmatrix} \mathcal{N}_Y \prec 0, \tag{32}$$

$$-\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \prec 0, \qquad -\begin{bmatrix} S & 0 \\ 0 & \Sigma \end{bmatrix} \prec 0 \tag{33}$$

hold with the definitions

$$S := \frac{1}{2}(W + W^T),\ T := \frac{1}{2}(W - W^T),\ \Sigma := \frac{1}{2}(\widetilde{W} + \widetilde{W}^T),\ \Gamma := \frac{1}{2}(\widetilde{W} - \widetilde{W}^T)$$

and subject to the algebraic constraints

$$W^{-1} = \widetilde{W}. \tag{34}$$

Note that due to the algebraic constraints (34), the problem under consideration is nonconvex and has been even shown to be NP-hard. See [6] and references therein. This feature is in stark contrast with the associated Linear Parameter-Varying control problem for which the LMI constraints (31)-(33) are the same but the nonlinear conditions (34) fully disappears. Finally, the problem takes the form discussed in Section 3:

$$\begin{array}{l} \min_x \gamma \\ \text{subject to LMIs } (31) - (33) \\ \text{ and algebraic constraint } W\widetilde{W} = I\,, \end{array}$$

where x gathers all variables into a single vector $x := (X, Y, W, \widetilde{W}, \gamma)$.

According to our discussion in Section 3, the penalty-barrier function associated with this problem is given as

$$F(x, \mu) := \gamma + \operatorname{Tr}(\Lambda^T(W\widetilde{W} - I)) + \frac{c_0}{\mu}\|W\widetilde{W} - I\|_F^2 - \mu \log\det\mathcal{A}(x)\,.$$

Derivative informations on this function are provided in Appendix B.

5 Applications

In this section, we consider a simple illustration of the proposed algorithm. The example consists of a mass-spring system depicted in Figure 1.

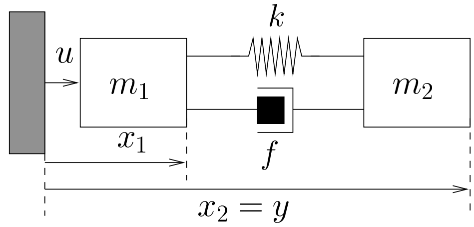

FIGURE 1: Flexible system

It is characterized by the nominal values of the parameters

$$m_1 = m_2 = 0.5\text{Kg} \;, k = 1\text{N/m} \;, f = 0.0025\text{Ns/m}.$$

We assume that relative uncertainties corrupt both k and m_2 in the form

$$k = k_{\text{nominal}}\,(1+\delta_k), \qquad m_2 = m_{2,\text{nominal}}\,(1+\delta_{m_2})\,.$$

An LFT representation of the uncertain mass-spring system is easily inferred from its diagram representation in Figure 2.

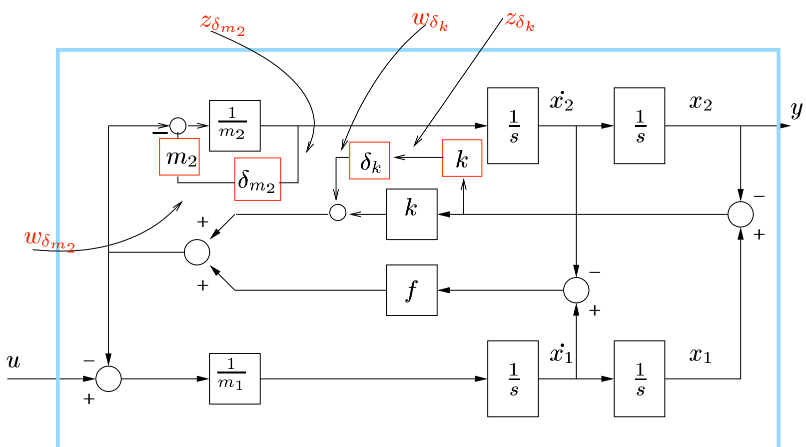

FIGURE 2: Diagram representation of mass-spring system

This leads to the LFT model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_1 \\ \ddot{x}_2 \\ \hline z_{\delta_k} \\ z_{\delta_{m_2}} \\ \hline y \end{bmatrix} = \left[\begin{array}{cccc|cc|c} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -k/m_1 & k/m_1 & -f/m_1 & f/m_1 & -1/m_1 & 0 & 1/m_1 \\ k/m_2 & -k/m_2 & f/m_2 & -f/m_2 & 1/m_2 & -1 & 0 \\ \hline k & -k & 0 & 0 & 0 & 0 & 0 \\ k/m_2 & -k/m_2 & f/m_2 & -f/m_2 & 1/m_2 & -1 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \\ \hline w_{\delta_k} \\ w_{\delta_{m_2}} \\ \hline u \end{bmatrix}, \qquad (35)$$

with

$$\begin{bmatrix} w_{\delta_k} \\ w_{\delta_{m_2}} \end{bmatrix} = \overbrace{\begin{bmatrix} \delta_k & 0 \\ 0 & \delta_{m_2} \end{bmatrix}}^{\Delta} \begin{bmatrix} z_{\delta_k} \\ z_{\delta_{m_2}} \end{bmatrix} .$$

In this problem, we have used an optimal-control-related quadratic criterion in the form

$$\max_{w \in L_2} \int_0^\infty (x_2^2 + (0.1\, u)^2)\, \mathrm{dt}\,, \tag{36}$$

where the exogenous signal w is a disturbance at the controller output. The first term in the integral reflects damping and high gain requirements for good tracking properties of the position of the second mass x_2, whereas the second term translates constraints in control signal energy. Standard manipulations exploiting (35) and (36) then gives the LFT synthesis plant in the form (26).

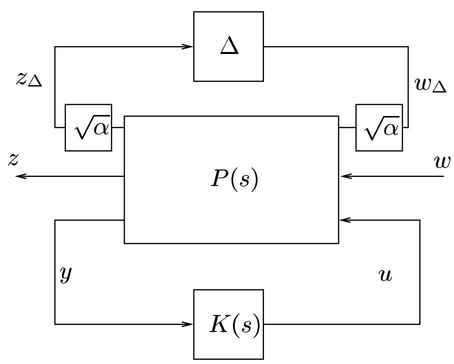

FIGURE 3: Synthesis structure with tradeoff parameter

We have used the primal-dual algorithm in Section 3 to find a solution to this problem. Our choice of parameters is $\eta_1 = 0.05$, $\eta_2 = 0.9$, $\gamma_1 = 1/3$ and $\gamma_2 = 3$ for updating the trust region radius. We have chosen $\beta = 0.5$ to update λ and μ. The inner stop test is based on the following rule

$$\varepsilon_1(\mu) := \max\{1e - 3,\ \mu\}\ ,$$

and $\mu_{\min} = 1e - 2$ for the outer stop test. Also, μ_+ is updated with $\mu_+ = \mu/2$.

In order to tradeoff the relative contributions of the uncertainty channel and the performance channel, an ajustment α parameter has been introduced as in Figure 3. The case $\alpha = 0$ corresponds to a pure nominal H_∞ synthesis with no parametric uncertainty while the uncertainty level increases with α.

Results for $\alpha = 0$ and $\alpha = 0.1$ are depicted in Figures 4 and 5, respectively. For each controller, we provide its root locus, its Nichols plot (upper right picture). The bottom left picture shows the parametric robustness of the controller. The grey area corresponds in the parameter space (δ_k, δ_{m_2}) to the stability region which has been estimated by an exhaustive search on a grid of points. The square delimits variations up to 30% of the nominal values.

Hence, the closed-loop system is stable for variations up to 30% in the parameters whenever the square is entirely contained in the grey area. The bottom right picture displays a superposition of the x_2 step responses associated with the nominal and corner values of the parameters for the 30% percent variation square. As can be expected, the pure H_∞ design is very sensitive to parameter variations due to pole/zero cancellation. The robust design, however, provides a very satisfactory answer in this respect.

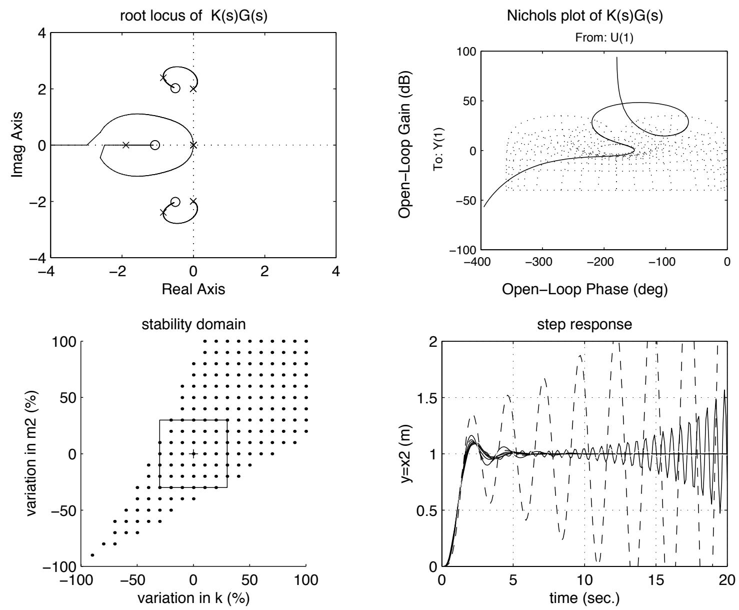

FIGURE 4: Results with $\alpha = 0$, H_∞ nominal design

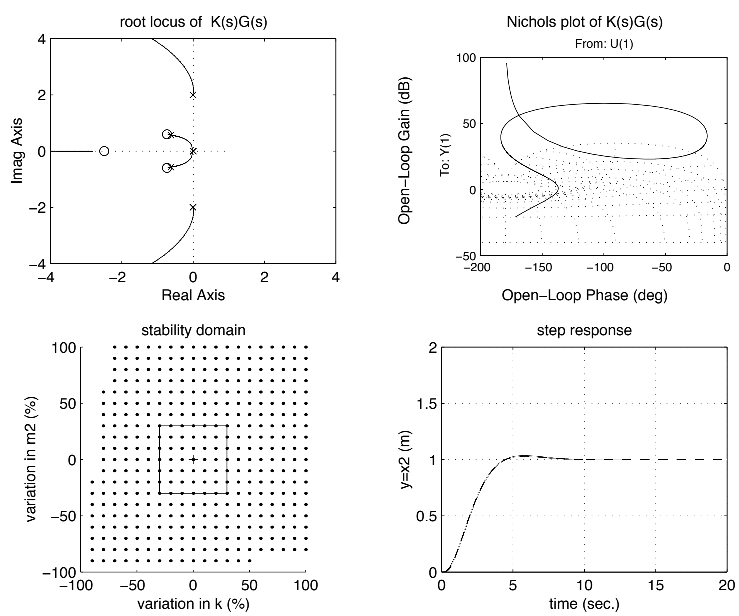

FIGURE 5: Results with $\alpha = 0.1$, robust design

It is worth mentioning that the proposed technique recovers the pure H_∞ controller in the case $\alpha = 0$. Hence, there is no loss due to non-global optimality for $\alpha = 0$. The algorithm histogram is shown in Table 1, Appendix C for the case $\alpha = 0.1$. These results are typical of what we observed in 8 different control problems. Columns in the table display the time histories of the penalty parameter, the barrier parameter, the current iterate feasibility (< 0 for feasibility), the trust region radius, the nonlinear constraint norm, γ, the value of the barrier/penalty function in (22), the maximum norm of the stationnarity conditions in (25).

With our choice of parameters, the problem is solved in 149 inner iterations corresponding to 27 outer iterations. This number compares favorably to the 645 inner iterations required with the SSDP method in [17]. A signicant amount ($> 20\%$) of inner iterations correspond to rejection of the step in the trust region strategy, hence, are much less costly than accepted steps since gradients and Hessians need not to be recomputed. When μ is updated, we observe a jump in the value of the stationnarity conditions. This means that at least in the early iterations, the computed point is not a good starting point for the next outer iteration and that the algorithm can be significantly accelerated through extrapolation techniques as suggested in [15]. In the final steps, however, there is only one inner iteration for each outer iteration corresponding to updating the Lagrange multiplier Λ and convergence to a local solution is obtained very quickly.

6 Conclusion

In this paper, we have described a very promising algorithm for solving hard problems in robust control theory. It takes advantage of a primal-dual formulation of the stationnary conditions and uses a trust region strategy for generating the primal steps. Each primal step is followed by an appropriate adjustment of the dual SD cone variables. Algthough, the proposed technique has proved satisfactory in a number of examples, it remains a prototype and additional algorithmic and numerical issues are of concerned. This will be discussed in the companion paper [34]. Also, application of the proposed ideas to the more general class of BMI problems will be investigated.

A Derivatives of LMI logarithmic barrier

The natural barrier function for an LMI

$$\mathcal{A}(x) := A_0 + \sum_{i=1}^{N} x_i A_i \succeq 0,$$

is defined as

$$b(x) := -\log\det(\mathcal{A}(x))\,.$$

Its gradient vector and Hessian matrix are easily computed using straightforward variational arguments. The gradient vector is obtained as

$$\nabla b(x) = - \begin{bmatrix} \operatorname{Tr} \mathcal{A}(x)^{-1} A_1 \\ \operatorname{Tr} \mathcal{A}(x)^{-1} A_2 \\ \vdots \\ \operatorname{Tr} \mathcal{A}(x)^{-1} A_N \end{bmatrix} = -A^T \operatorname{vec} \mathcal{A}(x)^{-1} \,,$$

and the Hessian matrix obeys the form

$$\nabla^2 b(x) = ((\operatorname{Tr} \mathcal{A}(x)^{-1} A_i \mathcal{A}(x)^{-1} A_j)) = A^T (\mathcal{A}(x)^{-1} \otimes \mathcal{A}(x)^{-1}) A \,,$$

where the matrix A is defined in (4).

B Derivatives of penalty/barrier for robust control problem

We assume here that the decision vector x gathers all variables into a single vector with the arrangement $x := (X, Y, W, \widetilde{W}, \gamma)$.

B.1 Gradient

The gradient of the penalty/barrier function can be computed as follows.

- Compute

$$E := W\widetilde{W} - I, \; g_W := \operatorname{vec} (\Lambda + \frac{c_0}{\mu} E)\widetilde{W}^T, \; g_{\widetilde{W}} := \operatorname{vec} W^T (\Lambda + \frac{c_0}{\mu} E) \,.$$

- Define a linear transformation T mapping $\operatorname{vec} W$ into a vector of smaller size which only retains its nonzero entries (W is a block-diagonal matrix).
- Form total gradient as

$$\nabla F(x, \mu) = \begin{bmatrix} 0_{\bar{n},1} \\ \operatorname{diag}(T, T) \begin{bmatrix} g_W \\ g_{\widetilde{W}} \end{bmatrix} \\ 1 \end{bmatrix} - \mu A^T \operatorname{vec} \mathcal{A}(x)^{-1} \,,$$

  with the definition $\bar{n} := \frac{n(n+1)}{2}$ and n is the plant's order.

B.2 Hessian

The Hessian matrix of the penalty/barrier function can be computed according to the following procedure.

- Compute

$$H_{W\widetilde{W}} := \begin{bmatrix} c_0/\mu\, \widetilde{W}\widetilde{W}^T \otimes I & * \\ J^T(\Lambda + c_0/\mu\, E \otimes I) + c_0/\mu\, \widetilde{W} \otimes W & c_0/\mu\, I \otimes W^T W \end{bmatrix}$$

where J is a permutation matrix such that $\mathrm{vec}\,(M^T) = J\,\mathrm{vec}\,(M)$.

- Construct the augmentation

$$\bar{H}_{W\widetilde{W}} := \mathrm{diag}\left(0_{\bar{n},\bar{n}}, H_{W\widetilde{W}}, 0\right)\,.$$

- Deduce the total Hessian as

$$H := \bar{H}_{W\widetilde{W}} + \mu A^T(\mathcal{A}(x)^{-1} \otimes \mathcal{A}(x)^{-1})A\,.$$

Note that the second term in the last expression is approximated by

$$A^T(Z \otimes \mathcal{A}(x)^{-1}))A$$

in the proposed primal-dual technique.

C Algorithm histogram

it.	c_0/μ	μ	feas.	rad.	nl	γ	f	stat.
1	5.000e-03	1.000e-01	-9.551e-06	3	3.851e-04	5.024e+00	-2.053e+01	7.308e+01
2	5.000e-03	1.000e-01	-9.551e-06	1	3.851e-04	5.024e+00	-2.053e+01	7.308e+01
3	5.000e-03	1.000e-01	-9.551e-06	3.333e-01	3.851e-04	5.024e+00	-2.053e+01	7.308e+01
4	5.000e-03	1.000e-01	-9.551e-06	1.111e-01	3.851e-04	5.024e+00	-2.053e+01	7.308e+01
5	5.000e-03	1.000e-01	-9.551e-06	3.704e-02	3.851e-04	5.024e+00	-2.053e+01	7.308e+01
6	5.000e-03	1.000e-01	-1.518e-04	3.704e-02	9.519e-04	5.021e+00	-2.083e+01	7.597e+03
7	5.000e-03	1.000e-01	-1.543e-04	1.111e-01	1.210e-03	5.023e+00	-2.088e+01	4.782e+02
8	5.000e-03	1.000e-01	-1.887e-04	3.333e-01	4.841e-03	5.033e+00	-2.107e+01	5.743e+02
9	5.000e-03	1.000e-01	-2.760e-04	1	2.356e-02	5.071e+00	-2.138e+01	2.270e+02
10	5.000e-03	1.000e-01	-7.097e-04	3	3.034e-01	5.051e+00	-2.209e+01	1.778e+01
11	5.000e-03	1.000e-01	-7.097e-04	1	3.034e-01	5.051e+00	-2.209e+01	1.778e+01
12	5.000e-03	1.000e-01	-7.097e-04	3.333e-01	3.034e-01	5.051e+00	-2.209e+01	1.778e+01
13	5.000e-03	1.000e-01	-4.088e-03	3.333e-01	1.865e+00	5.222e+00	-2.294e+01	1.246e+02
14	5.000e-03	1.000e-01	-4.983e-03	1	4.382e+00	4.975e+00	-2.330e+01	1.291e+01
15	5.000e-03	1.000e-01	-1.260e-02	3	4.041e+01	4.108e+00	-2.490e+01	7.944e+00
16	5.000e-03	1.000e-01	-1.260e-02	1	4.041e+01	4.108e+00	-2.490e+01	7.944e+00
17	5.000e-03	1.000e-01	-5.636e-03	1	7.259e+01	3.144e+00	-2.548e+01	5.722e+00
18	5.000e-03	1.000e-01	-5.636e-03	3.333e-01	7.259e+01	3.144e+00	-2.548e+01	5.722e+00
19	5.000e-03	1.000e-01	-5.636e-03	1.111e-01	7.259e+01	3.144e+00	-2.548e+01	5.722e+00
20	5.000e-03	1.000e-01	-5.636e-03	3.704e-02	7.259e+01	3.144e+00	-2.548e+01	5.722e+00
21	5.000e-03	1.000e-01	-2.210e-02	3.704e-02	6.935e+01	3.108e+00	-2.561e+01	1.740e+01
22	5.000e-03	1.000e-01	-2.305e-02	1.111e-01	6.348e+01	3.080e+00	-2.568e+01	4.634e+00
23	5.000e-03	1.000e-01	-2.994e-02	3.333e-01	6.979e+01	2.981e+00	-2.589e+01	3.459e+00
24	5.000e-03	1.000e-01	-4.613e-02	1	7.360e+01	2.639e+00	-2.627e+01	1.138e+00
25	5.000e-03	1.000e-01	-4.613e-02	3.333e-01	7.360e+01	2.639e+00	-2.627e+01	1.138e+00
26	5.000e-03	1.000e-01	-3.841e-02	3.333e-01	8.474e+01	2.284e+00	-2.672e+01	1.793e+00
27	5.000e-03	1.000e-01	-7.160e-02	1	8.137e+01	2.060e+00	-2.692e+01	6.394e-01
28	5.000e-03	1.000e-01	-1.356e-01	1	5.575e+01	1.769e+00	-2.723e+01	8.579e-01
29	5.000e-03	1.000e-01	-1.743e-01	3	1.508e+02	1.801e+00	-2.751e+01	4.345e-01
30	5.000e-03	1.000e-01	-1.743e-01	1	1.508e+02	1.801e+00	-2.751e+01	4.345e-01
31	5.000e-03	1.000e-01	-1.743e-01	3.333e-01	1.508e+02	1.801e+00	-2.751e+01	4.345e-01
32	5.000e-03	1.000e-01	-1.826e-01	3.333e-01	7.233e+01	1.694e+00	-2.753e+01	5.809e-01
33	5.000e-03	1.000e-01	-1.952e-01	3.333e-01	1.608e+02	1.712e+00	-2.765e+01	8.571e-02
34	1.000e-02	5.000e-02	-1.952e-01	3.333e-01	1.608e+02	1.712e+00	-1.237e+01	4.841e-01
35	1.000e-02	5.000e-02	-1.039e-01	1	4.638e+01	1.327e+00	-1.285e+01	2.791e-01
36	1.000e-02	5.000e-02	-1.039e-01	3.333e-01	4.638e+01	1.327e+00	-1.285e+01	2.791e-01
37	1.000e-02	5.000e-02	-1.039e-01	1.111e-01	4.638e+01	1.327e+00	-1.285e+01	2.791e-01
38	1.000e-02	5.000e-02	-1.011e-01	1.111e-01	3.089e+01	1.210e+00	-1.292e+01	1.944e-01
39	1.000e-02	5.000e-02	-8.311e-02	1.111e-01	2.685e+01	1.101e+00	-1.295e+01	1.104e-01
40	1.000e-02	5.000e-02	-7.421e-02	3.333e-01	3.171e+01	1.020e+00	-1.297e+01	2.014e-02
41	1.000e-02	5.000e-02	-7.421e-02	3.333e-01	3.171e+01	1.020e+00	-1.265e+01	2.602e-01
42	1.000e-02	5.000e-02	-7.421e-02	1.111e-01	3.171e+01	1.020e+00	-1.265e+01	2.602e-01
43	1.000e-02	5.000e-02	-7.207e-02	3.333e-01	1.403e+01	1.017e+00	-1.272e+01	5.358e-02
44	1.000e-02	5.000e-02	-7.207e-02	1.111e-01	1.403e+01	1.017e+00	-1.272e+01	5.358e-02
45	1.000e-02	5.000e-02	-6.965e-02	1.111e-01	1.096e+01	1.014e+00	-1.272e+01	3.034e-02
46	1.000e-02	5.000e-02	-6.965e-02	3.333e-01	1.096e+01	1.014e+00	-1.261e+01	1.294e-01
47	1.000e-02	5.000e-02	-6.965e-02	1.111e-01	1.096e+01	1.014e+00	-1.261e+01	1.294e-01
48	1.000e-02	5.000e-02	-6.852e-02	1.111e-01	6.426e+00	1.025e+00	-1.263e+01	5.421e-02
49	1.000e-02	5.000e-02	-6.787e-02	1.111e-01	6.103e+00	1.018e+00	-1.263e+01	1.193e-02
50	2.000e-02	2.500e-02	-6.787e-02	3.333e-01	6.103e+00	1.018e+00	-5.650e+00	8.312e-02
51	2.000e-02	2.500e-02	-5.261e-03	3.333e-01	1.888e-01	7.629e-01	-5.715e+00	2.720e-01
52	2.000e-02	2.500e-02	-2.685e-02	3.333e-01	1.692e-01	7.757e-01	-5.776e+00	6.348e-01
53	2.000e-02	2.500e-02	-2.521e-02	3.333e-01	1.190e+00	7.496e-01	-5.805e+00	1.461e-01
54	2.000e-02	2.500e-02	-2.485e-02	1	9.183e-01	7.529e-01	-5.809e+00	7.678e-03
55	2.000e-02	2.500e-02	-2.447e-02	3	5.694e-01	7.528e-01	-5.800e+00	9.161e-03
56	4.000e-02	1.250e-02	-5.687e-03	3	5.685e-02	6.424e-01	-2.531e+00	3.669e-02
57	4.000e-02	1.250e-02	-1.028e-02	9	3.634e-02	6.507e-01	-2.542e+00	8.083e-02
58	4.000e-02	1.250e-02	-8.660e-03	27	2.776e-02	6.273e-01	-2.567e+00	1.959e-01
59	4.000e-02	1.250e-02	-2.222e-03	27	1.308e-02	5.793e-01	-2.606e+00	9.338e-01
60	4.000e-02	1.250e-02	-9.364e-03	27	2.394e-02	5.652e-01	-2.635e+00	9.345e-01
61	4.000e-02	1.250e-02	-7.163e-03	81	1.640e-02	5.480e-01	-2.660e+00	8.511e-01
62	4.000e-02	1.250e-02	-6.618e-03	81	1.002e-02	5.140e-01	-2.705e+00	1.068e+00
63	4.000e-02	1.250e-02	-8.162e-03	243	6.322e-03	4.998e-01	-2.734e+00	2.785e+00
64	4.000e-02	1.250e-02	-8.162e-03	81	6.322e-03	4.998e-01	-2.734e+00	2.785e+00
65	4.000e-02	1.250e-02	-1.044e-02	81	2.053e-03	4.827e-01	-2.753e+00	1.247e+00
66	4.000e-02	1.250e-02	-9.973e-03	243	4.181e-03	4.744e-01	-2.771e+00	1.298e-01
67	4.000e-02	1.250e-02	-9.697e-03	729	4.832e-03	4.531e-01	-2.804e+00	1.260e+00
68	4.000e-02	1.250e-02	-9.697e-03	243	4.832e-03	4.531e-01	-2.804e+00	1.260e+00
69	4.000e-02	1.250e-02	-1.112e-02	729	1.641e-03	4.418e-01	-2.825e+00	7.513e-01
70	4.000e-02	1.250e-02	-4.696e-03	2.060e+03	1.601e-04	4.130e-01	-2.855e+00	7.474e+00
71	4.000e-02	1.250e-02	-4.696e-03	6.866e+02	1.601e-04	4.130e-01	-2.855e+00	7.474e+00
72	4.000e-02	1.250e-02	-7.809e-03	6.866e+02	4.282e-03	4.003e-01	-2.869e+00	7.017e+00
73	4.000e-02	1.250e-02	-1.135e-02	2.060e+03	5.136e-05	4.004e-01	-2.889e+00	6.938e+00
74	4.000e-02	1.250e-02	-1.275e-02	2.060e+03	2.250e-04	3.901e-01	-2.920e+00	9.015e+00
75	4.000e-02	1.250e-02	-8.182e-03	6.180e+03	8.408e-03	3.775e-01	-2.937e+00	5.673e+00
76	4.000e-02	1.250e-02	-2.476e-03	6.180e+03	2.455e-02	3.573e-01	-2.953e+00	2.003e+01
77	4.000e-02	1.250e-02	-1.065e-02	6.180e+03	5.401e-04	3.631e-01	-2.979e+00	2.404e+01
78	4.000e-02	1.250e-02	-1.092e-02	6.180e+03	1.050e-03	3.586e-01	-2.994e+00	8.453e+00
79	4.000e-02	1.250e-02	-8.745e-03	1.854e+04	1.751e-03	3.494e-01	-3.004e+00	3.130e+00
80	4.000e-02	1.250e-02	-8.234e-03	5.562e+04	3.591e-05	3.409e-01	-3.024e+00	8.047e+00

TABLE 1: Algorithm histogram for mass-spring system
Computations with MINQ quadratic solver
on SUN ultra 5.

it.	c_0/μ	μ	feas.	rad.	nl	γ	f	stat.
81	4.000e-02	1.250e-02	-6.624e-03	1.669e+05	3.261e-03	3.271e-01	-3.052e+00	3.840e+01
82	4.000e-02	1.250e-02	-6.624e-03	5.562e+04	3.261e-03	3.271e-01	-3.052e+00	3.840e+01
83	4.000e-02	1.250e-02	-5.490e-03	5.562e+04	1.790e-03	3.228e-01	-3.066e+00	4.504e+01
84	4.000e-02	1.250e-02	-5.876e-03	1.669e+05	4.964e-05	3.219e-01	-3.077e+00	1.462e+01
85	4.000e-02	1.250e-02	-5.410e-03	5.006e+05	5.088e-05	3.169e-01	-3.097e+00	8.288e+00
86	4.000e-02	1.250e-02	-3.852e-03	1.477e+06	1.547e-02	3.058e-01	-3.121e+00	1.000e+02
87	4.000e-02	1.250e-02	-3.852e-03	4.924e+05	1.547e-02	3.058e-01	-3.121e+00	1.000e+02
88	4.000e-02	1.250e-02	-3.325e-03	4.924e+05	2.478e-03	3.023e-01	-3.133e+00	9.806e+01
89	4.000e-02	1.250e-02	-3.648e-03	1.477e+06	1.625e-04	3.034e-01	-3.142e+00	1.403e+01
90	4.000e-02	1.250e-02	-3.440e-03	4.432e+06	1.595e-04	3.000e-01	-3.158e+00	3.580e+01
91	4.000e-02	1.250e-02	-2.703e-03	1.330e+07	6.518e-03	2.929e-01	-3.180e+00	1.606e+02
92	4.000e-02	1.250e-02	-2.320e-03	1.330e+07	3.425e-03	2.886e-01	-3.204e+00	8.814e+01
93	4.000e-02	1.250e-02	-2.255e-03	3.989e+07	2.903e-04	2.870e-01	-3.215e+00	2.444e+02
94	4.000e-02	1.250e-02	-1.974e-03	3.989e+07	1.513e-03	2.850e-01	-3.224e+00	8.107e+01
95	4.000e-02	1.250e-02	-1.912e-03	3.930e+07	3.694e-04	2.841e-01	-3.227e+00	3.944e+01
96	4.000e-02	1.250e-02	-1.863e-03	2.569e+07	3.264e-04	2.837e-01	-3.228e+00	1.343e+00
97	4.000e-02	1.250e-02	-1.843e-03	1.407e+07	3.014e-04	2.835e-01	-3.228e+00	1.325e+00
98	4.000e-02	1.250e-02	-1.833e-03	7.129e+06	2.984e-04	2.834e-01	-3.228e+00	2.997e-01
99	4.000e-02	1.250e-02	-1.829e-03	3.648e+06	2.983e-04	2.833e-01	-3.228e+00	7.855e-02
100	4.000e-02	1.250e-02	-1.826e-03	1.688e+06	2.985e-04	2.833e-01	-3.228e+00	1.721e-02
101	4.000e-02	1.250e-02	-1.825e-03	7.860e+05	2.987e-04	2.833e-01	-3.228e+00	3.635e-03
102	4.000e-02	1.250e-02	-1.833e-03	3	1.429e-04	2.832e-01	-3.228e+00	3.510e-05
103	4.000e-02	1.250e-02	-1.838e-03	3	6.791e-05	2.831e-01	-3.228e+00	2.896e-05
104	4.000e-02	1.250e-02	-1.841e-03	3	3.216e-05	2.830e-01	-3.228e+00	2.549e-05
105	4.000e-02	1.250e-02	-1.843e-03	3	1.519e-05	2.829e-01	-3.228e+00	2.073e-05
106	4.000e-02	1.250e-02	-1.845e-03	3	7.158e-06	2.829e-01	-3.228e+00	1.611e-05
107	4.000e-02	1.250e-02	-1.845e-03	3	3.366e-06	2.829e-01	-3.228e+00	1.277e-05
108	4.000e-02	1.250e-02	-1.846e-03	3	1.580e-06	2.829e-01	-3.228e+00	9.882e-06
109	4.000e-02	1.250e-02	-1.846e-03	3	7.399e-07	2.829e-01	-3.228e+00	7.194e-06
110	4.000e-02	1.250e-02	-1.846e-03	1	3.483e-07	2.829e-01	-3.228e+00	9.338e-07
111	4.000e-02	1.250e-02	-1.846e-03	3.333e-01	3.483e-07	2.829e-01	-3.228e+00	1.818e-03
112	8.000e-02	6.250e-03	-8.622e-04	3	5.563e-02	2.358e-01	-1.491e+00	3.797e-01
113	8.000e-02	6.250e-03	-1.040e-03	3	6.344e-02	2.440e-01	-1.493e+00	1.950e+00
114	8.000e-02	6.250e-03	-1.041e-03	9	6.371e-02	2.435e-01	-1.493e+00	2.374e-02
115	8.000e-02	6.250e-03	-1.038e-03	27	6.361e-02	2.434e-01	-1.493e+00	2.286e-03
116	1.600e-01	3.125e-03	-5.951e-04	3	7.064e-02	2.231e-01	-6.402e-01	1.346e-01
117	1.600e-01	3.125e-03	-6.371e-04	9	7.183e-02	2.241e-01	-6.405e-01	6.890e-01
118	1.600e-01	3.125e-03	-6.393e-04	27	7.093e-02	2.240e-01	-6.405e-01	4.551e-03
119	3.200e-01	1.563e-03	-3.882e-04	3	3.474e-02	2.114e-01	-2.179e-01	5.178e-02
120	3.200e-01	1.563e-03	-3.999e-04	9	3.574e-02	2.114e-01	-2.181e-01	2.966e-01
121	3.200e-01	1.563e-03	-4.023e-04	27	3.489e-02	2.114e-01	-2.181e-01	3.301e-02
122	3.200e-01	1.563e-03	-4.006e-04	81	3.486e-02	2.114e-01	-2.181e-01	1.033e-02
123	3.200e-01	1.563e-03	-3.959e-04	243	3.492e-02	2.112e-01	-2.182e-01	7.428e-02
124	3.200e-01	1.563e-03	-3.821e-04	729	3.544e-02	2.109e-01	-2.185e-01	2.875e-01
125	3.200e-01	1.563e-03	-3.439e-04	729	3.961e-02	2.099e-01	-2.191e-01	9.849e-01
126	3.200e-01	1.563e-03	-3.220e-04	2187	3.706e-02	2.092e-01	-2.197e-01	9.343e-01
127	3.200e-01	1.563e-03	-3.220e-04	729	3.706e-02	2.092e-01	-2.197e-01	9.343e-01
128	3.200e-01	1.563e-03	-2.941e-04	2187	3.805e-02	2.087e-01	-2.200e-01	6.385e-01
129	3.200e-01	1.563e-03	-2.941e-04	729	3.805e-02	2.087e-01	-2.200e-01	6.385e-01
130	3.200e-01	1.563e-03	-2.746e-04	2187	3.669e-02	2.083e-01	-2.203e-01	7.848e-01
131	3.200e-01	1.563e-03	-2.746e-04	729	3.669e-02	2.083e-01	-2.203e-01	7.848e-01
132	3.200e-01	1.563e-03	-2.563e-04	2187	3.635e-02	2.080e-01	-2.205e-01	6.320e-01
133	3.200e-01	1.563e-03	-2.563e-04	729	3.635e-02	2.080e-01	-2.205e-01	6.320e-01
134	3.200e-01	1.563e-03	-2.412e-04	2187	3.576e-02	2.078e-01	-2.206e-01	5.287e-01
135	3.200e-01	1.563e-03	-2.412e-04	729	3.576e-02	2.078e-01	-2.206e-01	5.287e-01
136	3.200e-01	1.563e-03	-2.280e-04	2187	3.537e-02	2.076e-01	-2.207e-01	4.667e-01
137	3.200e-01	1.563e-03	-2.108e-04	2187	3.633e-02	2.074e-01	-2.207e-01	7.787e-01
138	3.200e-01	1.563e-03	-2.078e-04	6561	3.383e-02	2.074e-01	-2.207e-01	1.463e-01
139	3.200e-01	1.563e-03	-2.049e-04	19683	3.377e-02	2.074e-01	-2.207e-01	4.030e-02
140	3.200e-01	1.563e-03	-2.048e-04	59049	3.368e-02	2.074e-01	-2.207e-01	6.520e-05
141	3.200e-01	1.563e-03	-2.394e-04	3	5.032e-04	2.018e-01	-2.145e-01	1.793e-01
142	3.200e-01	1.563e-03	-2.430e-04	3	6.513e-04	2.022e-01	-2.146e-01	4.179e-01
143	3.200e-01	1.563e-03	-2.437e-04	9	6.307e-04	2.023e-01	-2.146e-01	2.965e-03
144	3.200e-01	1.563e-03	-2.488e-04	3	7.836e-06	2.016e-01	-2.144e-01	7.588e-03
145	3.200e-01	1.563e-03	-2.499e-04	3	1.753e-07	2.016e-01	-2.144e-01	3.167e-04
146	3.200e-01	1.563e-03	-2.500e-04	3	1.643e-09	2.015e-01	-2.144e-01	6.549e-06
147	3.200e-01	1.563e-03	-2.499e-04	3	1.778e-11	2.015e-01	-2.144e-01	1.012e-06
148	3.200e-01	1.563e-03	-2.499e-04	3	3.059e-15	2.015e-01	-2.144e-01	1.192e-06
149	3.200e-01	1.563e-03	-2.499e-04	3	1.815e-13	2.015e-01	-2.144e-01	1.138e-06

TABLE 2: Algorithm histogram continued

Acnowledgment

We would like to thank A. Conn, N. Gould, D. Orban and P. Toint for providing an early version of their papers.

References

[1] P. APKARIAN AND P. GAHINET, *A Convex Characterization of Gain-Scheduled H_∞ Con-*

trollers, 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, *A Sequential SDP/Gauss-Newton Algorithm for Rank-Constrained LMI Problems*, in Proc. IEEE Conf. on Decision and Control, 1999.

[4] ——, *Concave Programming in Control Theory*, Journal of Global Optimization, 15 (1999), pp. 343–370.

[5] V. Balakrishnan, *Linear matrix inequalities in robustness analysis with multipliers*, Syst. Control Letters, 25 (1995), pp. 265–272.

[6] K. P. Bennett and O. L. Mangasarian, *Bilinear Separation of Two Sets in n-Space*, Computational Optimization and Applications, 2 (1993), pp. 207–227.

[7] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Academic Press, London, 1982.

[8] ——, *Nonlinear Programming*, Athena Scientific, USA, Belmont, Mass., 1995.

[9] S. Boyd and L. ElGhaoui, *Method of Centers for Minimizing Generalized Eigenvalues*, Linear Algebra and Appl., 188 (1992), pp. 63–111.

[10] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, vol. 15 of SIAM Studies in Applied Mathematics, SIAM, Philadelphia, 1994.

[11] A. R. Conn, N. Gould, D. Orban, and P. L. Toint, *A Primal-Dual Trust-Region Algorithm for Non-convex Nonlinear Programming*, Math. Programming, 2 (2000), pp. 215–249.

[12] M. C. de Oliveira, J. Bernussou, and J. C. Geromel, *A New Discrete-Time Robust Stability Condition*, Syst. Control Letters, 37 (1999), pp. 261–265.

[13] I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[14] 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.

[15] J. P. Dussault, *Numerical stability and efficiency of penalty algorithms*, SIAM J. on Numerical Analysis, 32 (1995), pp. 296–317.

[16] 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.

[17] B. Fares, P. Apkarian, and D. Noll, *An Augmented Lagrangian Method for a Class of LMI-Constrained Problems in Robust Control Theory*, (2000). to appear in Int. Journal of Control.

[18] B. Fares, D. Noll, and P. Apkarian, *Primal-dual Lagrangian methods for robust control synthesis* , (2000). submitted to SIAM Journal On Control and Optimization.

[19] E. Feron, *Analysis of robust* $\mathbf{H}_2$ *performance with multipliers*, in Proc. IEEE Conf. on Decision and Control, Lake Buena Vista, Florida, Dec. 1994.

[20] E. Feron, P. Apkarian, and P. Gahinet, *Analysis and Synthesis of Robust Control Systems via Parameter-Dependent Lyapunov Functions*, IEEE Trans. Aut. Control, 41 (1996), pp. 1041–1046.

[21] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 1987.

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

[23] P. Gahinet and A. Nemirovski, *General-Purpose LMI Solvers with Benchmarks*, in Proc. IEEE Conf. on Decision and Control, San Antonio, TX, 1993, pp. 3162–3165.

[24] 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.

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

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

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

[28] J. W. Helton and O. Merino, *Coordinate optimization for bi-convex matrix inequalitites*, in Proc. IEEE Conf. on Decision and Control, San Diego, CA, 1997, pp. 3609–3613.

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

[30] 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.

[31] F. Jarre, *A qqp-minimization algorithm method for semidefinite and smooth nonconvex programs*, tech. rep., The University of Notre Dame, Dec. 1999.

[32] F. Leibfritz and E. M. E. Mostafa, *An interior point constrained trust region method for a special class of nonlinear semi-definite programming problems*, SIAM J. on Control and Optimization, (2000). Submitted.

[33] A. Megretski and A. Rantzer, *System analysis via integral quadratic constraints*, IEEE Trans. Aut. Control, 42 (1997), pp. 819–830.

[34] D. Noll and P. Apkarian, *A Primal-Dual Technique for Robust Control Design: Convergence and Numerical Implementation*, (2000). under preparation.

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

[36] 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, Brighton,UK, 1991, pp. 1245–1250.

[37] E. Polak, *Optimization : Algorithms and Consistent Approximations*, Applied Mathematical Sciences, 1997.

[38] A. Sartenaer, *Automatic determination of an initial trust region in nonlinear programming*, J. sisci, 18 (1997), pp. 1788–1803.

[39] 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.

[40] C. Scherer, P. Gahinet, and M. Chilali, *Multi-Objective Output-Feedback Control via LMI Optimization*, IEEE Trans. Aut. Control, 42 (1997), pp. 896–911.

[41] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming, (1997), pp. 301–320.

[42] M. Todd, *A Study of Search Directions in Primal-Dual Interior-Point Methods for Semidefinite Programming* , Optimization Methods and Software , (1999), pp. 1–46.

[43] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. on Optimization, 8 (1998), pp. 365–386.