NON LINEAR SPECTRAL SDP METHOD FOR BMI-CONSTRAINED PROBLEMS : APPLICATIONS TO CONTROL DESIGN

Jean-Baptiste Thevenet
ONERA-CERT, 2 av. Edouard Belin, 31055 Toulouse, France
and UPS-MIP (Mathmatiques pour l'Industrie et la Physique), CNRS UMR 5640
118, route de Narbonne 31062 Toulouse, France

Dominikus Noll
UPS-MIP,

Pierre Apkarian
ONERA-CERT and UPS-MIP,

Keywords: Bilinear matrix inequality, spectral penalty function, trust-region, control synthesis.

Abstract: The purpose of this paper is to examine a nonlinear spectral semidefinite programming method to solve problems with bilinear matrix inequality (BMI) constraints. Such optimization programs arise frequently in automatic control and are difficult to solve due to the inherent non-convexity. The method we discuss here is of augmented Lagrangian type and uses a succession of unconstrained subproblems to approximate the BMI optimization program. These tangent programs are solved by a trust region strategy. The method is tested against several difficult examples in feedback control synthesis.

1 INTRODUCTION

Minimizing a linear objective function subject to bilinear matrix inequality (BMI) constraints is a useful way to describe many robust control synthesis problems. Many other problems in automatic control lead to BMI feasibility and optimization programs, such as filtering problems, synthesis of structured or reduced-order feedback controllers, simultaneous stabilization problems and many others. Formally, these problems may be described as

$$\begin{array}{ll} \text{minimize} & c^T x, \ x \in \mathbb{R}^n \\ \text{subject to} & \mathcal{B}(x) \preceq 0, \end{array} \tag{1}$$

where $\preceq 0$ means negative semi-definite and

$$\mathcal{B}(x) = A_0 + \sum_{i=1}^{n} x_i A_i + \sum_{1 \le i < j \le n} x_i x_j B_{ij} \tag{2}$$

is a bilinear matrix-valued operator with data $A_i, B_{ij} \in \mathbb{S}^m$.

The importance of BMIs was recognized over the past decade and different solution strategies have been proposed. The earliest ideas use interior point methods (imported from semidefinite programming), concave programming or coordinate descent methods employing successions of SDP subproblems. The success of these methods is up to now very limited, and even moderately sized programs with more than 100 variables usually make such methods fail.

Significant progress was achieved by two recent approaches, both using successions of SDP tangent subproblems. In (Fares et al., 2002), a sequential semidefinite is used, programming (S-SDP) algorithm, expanding on the sequential quadratic programming (SQP) technique in traditional nonlinear programming. In (Noll et al., 2002; Apkarian et al., 2003b; Fares et al., 2001) an augmented Lagrangian strategy is proposed, which was since then successfully tested against a considerable number of difficult synthesis problems (Fares et al., 2000; Apkarian et al., 2002; Apkarian et al., 2003a; Noll et al., 2002). This approach is particularly suited when BMI problems are transformed to LMI-constrained programs with an additional nonlinear equality constraints:

$$\begin{array}{ll} \text{minimize} & c^T x, \ x \in \mathbb{R}^n \\ \text{subject to} & \mathcal{A}(x) \preceq 0 \\ & g(x) = 0. \end{array} \tag{3}$$

Here $\mathcal{A}$ is an affine operator (i.e., (2) with $B_{ij} = 0$) with values in $\mathbb{S}^m$, and $g : \mathbb{R}^n \to \mathbb{R}^p$ incorporates a finite number of equality constraints. Notice that *every* BMI problem may in principle be transformed to the form (3), but the change will only be fruitful if it is motivated from the control point of view. The latter is for instance the case in robust or reduced order synthesis, where the projection lemma is successfully brought into play. For semidefinite programming, (Zibulevsky, 1996; Ben-Tal and Zibulevsky, 1997) propose a modified augmented Lagrangian method,

Thevenet J., Noll D. and Apkarian P. (2004).
NON LINEAR SPECTRAL SDP METHOD FOR BMI-CONSTRAINED PROBLEMS : APPLICATIONS TO CONTROL DESIGN.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 237-248
DOI: 10.5220/0001128402370248

which in (Kocvara and Stingl, 2003) has been used to build a software tool for SDP called PENNON. The idea is further expanded in (Henrion et al., 2003) to include BMI-problems.

The present paper addresses cases where the passage from (1) to (3) seems less suited and a direct approach is required. Our algorithm solves (1) by a sequence of unconstrained subproblems via a trust region technique. Using trust regions is of the essence, since we have to bear in mind that (1) is usually highly non-convex. In consequence, local optimal methods risk failure by getting stalled at a local minimum of constraint violation. This refers to points x satisfying $\mathcal{B}(x) \not\preceq 0$, where the algorithm makes no further progress. Such a case means failure to solve the underlying control problem, which requires at least x with $\mathcal{B}(x) \preceq 0$. This phenomenon may be avoided (or at least significantly reduced) with genuine second order strategies. In particular, trust region techniques do not require convexification of the tangent problem Hessian, a major advantage over line search methods. (Notice that in this situation, any approach to (1) using a succession of SDPs is bound to face similar difficulties due to the need to convexify the Hessian. On the other hand, if LMI-constrained subproblems with non-convex quadratic objectives $f(x)$ can be handled, such an approach is promising. This idea has been developed successfully in (Apkarian et al., 2003a).

While our technique could be applied basically to *any* BMI-constrained program, we presently focus on automatic control issues. In particular, we give applications in static output feedback control under $\mathcal{H}_2$, $\mathcal{H}_\infty$ or mixed $\mathcal{H}_\infty/\mathcal{H}_2$ performance indices. Two other difficult cases, multi-model and multi-objective synthesis, and synthesis of structured controllers, are presented.

The structure of the paper is as follows. In section 2, we present our spectral SDP method and discuss its principal features. Section 3 presents theory needed to compute first and second derivatives of spectral functions. Finally, in section 4, numerical experiments on automatic control issues are examined.

NOTATION

Our notation is standard. We let $\mathbb{S}^m$ denote the set of $m \times m$ symmetric matrices, M^T the transpose of the matrix M and $\operatorname{Tr} M$ its trace. We equip $\mathbb{S}^m$ with the euclidean scalar product $\langle X, Y\rangle = X \bullet Y = \operatorname{Tr}(XY)$. For symmetric matrices , $M \succ N$ means that $M - N$ is positive definite and $M \succeq N$ means that $M - N$ is positive semi-definite. The symbol $\otimes$ stands for the usual Kronecker product of matrices and vec stands for the columnwise vectorization on matrices. We shall make use of the properties:

$$\begin{aligned} \operatorname{vec}(AXB) &= (B^T \otimes A)\operatorname{vec} X, & (4)\\ \operatorname{Tr}(AB) &= \operatorname{vec}^T A^T \operatorname{vec} B, & (5)\end{aligned}$$

which hold for matrices A, X and B of compatible dimensions. The Hadamard or Schur product is defined as

$$A \circ B = ((A_{ij}B_{ij})). \qquad (6)$$

The following holds for matrices of the same dimension:

$$\operatorname{vec} A \circ \operatorname{vec} B = \operatorname{diag}(\operatorname{vec} A)\operatorname{vec} B, \qquad (7)$$

where the operator diag forms a diagonal matrix with $\operatorname{vec} A$ on the main diagonal.

2 NONLINEAR SPECTRAL SDP METHOD

In this chapter we present and discuss our method to solve BMI-constrained optimization problems. Section 2.1 gives the general out-set, while the subsequent sections discuss theoretical and implementational aspects of the method.

2.1 General outline

The method we are about to present applies to more general classes of objective functions than (1). We shall consider matrix inequality constrained programs of the form

$$\begin{array}{ll} \text{minimize} & f(x),\ x \in \mathbb{R}^n \\ \text{subject to} & \mathcal{F}(x) \preceq 0, \end{array} \qquad (8)$$

where f is a class $\mathcal{C}^2$ function, $\mathcal{F} : \mathbb{R}^n \to \mathbb{S}^m$ a class $\mathcal{C}^2$ operator. We will use the symbol $\mathcal{B}$ whenever we wish to specialize to bilinear operators (2) or to the form (1), and we use $\mathcal{A}$ for affine operators.

Our method is a penalty/barrier multiplier algorithm as proposed in (Zibulevsky, 1996; Mosheyev and Zibulevsky, 2000). According to that terminology, a spectral penalty (SP) function for (1) is defined as

$$F(x, p) = f(x) + \operatorname{Tr}\Phi_p(\mathcal{F}(x)), \qquad (9)$$

where $\varphi_p : \mathbb{R} \to \mathbb{R}$ is a parametrized family of scalar functions, which generates a family of matrix-valued operators $\Phi_p : \mathbb{S}^m \to \mathbb{S}^m$ upon defining:

$$\Phi_p(X) := S\operatorname{diag}\left[\varphi_p\left(\lambda_i(X)\right)\right] S^T. \qquad (10)$$

Here $\lambda_i(X)$ stands for the ith eigenvalue of $X \in \mathbb{S}^m$ and S is an orthonormal matrix of associated eigenvectors. An alternative expression for (9) using (10) is:

$$F(x,p) = f(x) + \sum_{i=1}^{m} \varphi_p(\lambda_i(\mathcal{F}(x)) \,. \qquad (11)$$

As φ_p is chosen strictly increasing and satisfies $\varphi_p(0) = 0$, each of the following programs (parametrized through $p > 0$)

$$\begin{array}{ll} \text{minimize} & f(x),\, x \in \mathbb{R}^n \\ \text{subject to} & \Phi_p(\mathcal{F}(x)) \preceq 0 \end{array} \qquad (12)$$

is equivalent to (8). Thus, $F(x,p)$ may be understood as a penalty function for (8). Forcing $p \to 0$, we expect the solutions to the unconstrained program $\min_x F(x,p)$ to converge to a solution of (8).

It is well-known that pure penalty methods run into numerical difficulties as soon as penalty constants get large. Similarly, using pure SP functions as in (9) would lead to ill-conditioning for small $p > 0$. The epoch-making idea of Hestenes (Hestenes, 1969) and Powell (Powell, 1969), known as the augmented Lagrangian approach, was to avoid this phenomenon by including a linear term carrying a Lagrange multiplier estimate into the objective. In the present context, we follow the same line, but incorporate Lagrange multiplier information by a nonlinear term. We define the augmented Lagrangian function associated with the matrix inequality constraints in (1) as

$$\begin{aligned} L(x,V,p) &= f(x) + \operatorname{Tr}\Phi_p(V^T\mathcal{F}(x)V)\,, \quad (13) \\ &= f(x) + \sum_{i=1}^{m} \varphi_p(\lambda_i(V^T\mathcal{F}(x)V)). \end{aligned}$$

In this expression, the matrix variable V has the same dimension as $\mathcal{F}(x) \in \mathbb{S}^m$ and serves as a factor of the Lagrange multiplier variable $U \in \mathbb{S}^m$, $U = VV^T$ (for instance, a Cholesky factor). This has the immediate advantage of maintaining non-negativity of $U \succeq 0$. In contrast with classical augmented Lagrangians, however, the Lagrange multiplier U is not involved linearly in (13). We nevertheless reserve the name of an augmented Lagrangian for $L(x,V,p)$, as its properties resemble those of the classical augmented Lagrangian. Surprisingly, the non-linear dependence of $L(x,V,p)$ on V is not at all troublesome and a suitable first-order update formula $V \to V^+$, generalizing the classical one, will be readily derived in section 3.2. We will even see some genuine advantages of (13) over the case of a linear multiplier U.

Let us mention that the convergence theory for an augmented Lagrangian method like the present one splits into a local and a global branch. Global theory gives weak convergence of the method towards critical points from arbitrary starting points x, if necessary by driving $p \to 0$. An important complement is provided by local theory, which shows that as soon as the iterates reach a neighbourhood of attraction of one of those critical points predicted by global theory, and if this critical point happens to be a local minimum satisfying the sufficient second order optimality conditions, then the sequence will stay in this neighbourhood, converge to the minimum in question, and the user will not have to push the parameter p below a certain threshold $\bar{p} > 0$. Proofs for global and local convergence of the AL exist in traditional nonlinear programming (see for instance (Conn et al., 1991; Conn et al., 1996; Conn et al., 1993b; Conn et al., 1993a; Bertsekas, 1982)). Convergence theory for matrix inequality constraints is still a somewhat unexplored field. A global convergence result which could be adapted to the present context is given in (Noll et al., 2002). Local theory for the present approach covering a large class of penalty functions φ_p will be presented in a forthcoming article. Notice that in the convex case a convergence result has been published in (Zibulevsky, 1996). It is based on Rockafellar's idea relating the AL method to a proximal point algorithm. Our present testing of nonconvex programs (1) shows a similar picture. Even without convexity the method converges most of the time and the penalty parameter is in the end stably away from 0, $p \geq \bar{p}$.

Schematically, the augmented Lagrangian technique is as follows:

Spectral augmented Lagrangian algorithm

1. Initial phase. Set constants $\gamma > 0, \rho < 1$. Initialize the algorithm with x_0, V_0 and a penalty parameter $p_0 > 0$.

2. Optimization phase. For fixed V_j and p_j solve the unconstrained subproblem

$$\text{minimize}_{x \in \mathbb{R}^n} \quad L(x, V_j, p_j) \qquad (14)$$

Let x_{j+1} be the solution. Use the previous iterate x_j as a starting value for the inner optimization.

3. Update penalty and multiplier. Apply first-order rule to estimate Lagrange multiplier :

$$\begin{aligned} V_{j+1}V_{j+1}^T = V_j S \left[\operatorname{diag}\varphi_p'\left(\lambda_i\left(V_j^T\right.\right.\right. \\ \left.\left.\left.\cdot\mathcal{F}(x_{j+1})V_j\right)\right)\right] S^T V_j^T, \quad (15) \end{aligned}$$

where S diagonalizes $V_j^T\mathcal{F}(x_{j+1})V_j$.
Update the penalty parameter using :

$$p_{j+1} = \begin{cases} \rho p_j, & \text{if } \lambda_{\max}(0,\mathcal{F}(x_{j+1})) \\ & \quad > \gamma\, \lambda_{\max}(0,\mathcal{F}(x_j)) \\ p_j, & \text{else} \end{cases} \qquad (16)$$

Increase j and go back to step 2.

In our implementation, following the recommendation in (Mosheyev and Zibulevsky, 2000), we have used the log-quadratic penalty function $\varphi_p(t) = p\varphi_1(t/p)$ where

$$\varphi_1(t) = \begin{cases} t + \frac{1}{2}t^2 & \text{if} \quad t \geq -\frac{1}{2} \\ -\frac{1}{4}\log(-2t) - \frac{3}{8} & \text{if} \quad t < -\frac{1}{2}, \end{cases} \tag{17}$$

but other choices could be used (see for instance (Zibulevsky, 1996) for an extended list).

The multiplier update formula (15) requires the full machinery of differentiability of the spectral function $\operatorname{Tr}\Phi_p$ and will be derived in section 3.2. Algorithmic aspects of the subproblem in step 2 will be discussed in section 3.3. We start by analyzing the idea of the spectral AL-method.

2.2 The mechanism of the algorithm

In order to understand the rationale behind the AL-algorithm, it may instructive to consider classical nonlinear programming, which is a special case of the scheme if the values of the operator $\mathcal{F}$ are diagonal matrices: $\mathcal{F}(x) = \operatorname{diag}[c_1(x), \ldots, c_m(x)]$. Then the Lagrange multiplier U and its factor V may also be restricted to the space of diagonal matrices, $U = \operatorname{diag} u_i$, $V = \operatorname{diag} v_i$, and we recover the situation of classical polyhedral constraints. Switching to a more convenient notation, the problem becomes

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c_j(x) \leq 0, \; j = 1, \ldots, m \end{array}$$

With $u_i = v_i^2$, we obtain the analogue of the augmented Lagrangian (13):

$$L(x, v, p) \; = \; f(x) \; + \; \sum_{i=1}^{m} p\varphi_1(v_i^2 c_i(x)/p).$$

Here we use (17) or another choice from the list in (Zibulevsky, 1996). Computing derivatives is easy here and we obtain

$$\begin{aligned} \nabla L(x, v, p) &= \nabla f(x) \\ &\quad + \sum_{i=1}^{m} \varphi_1'(v_i^2 c_i(x)/p) v_i^2 \nabla c_i(x). \end{aligned}$$

If we compare with the Lagrangian :
$L(x, u) = f(x) + u^T c(x)$,
and its gradient :
$\nabla L(x, u) = \nabla f(x) + \sum_{i=1}^{m} u_i \nabla c_i(x)$,
the following update formula readily appears:

$$u_i^+ = v_i^{+2} = v_i^2 \varphi_1'(v_i^2 c_i(x^+)/p),$$

the scalar analogue of (15). Suppose now that φ_1 has the form (17). Then for sufficiently small $v_i^2 c_i(x)$ we expect $v_i^2 c_i(x)/p > -\frac{1}{2}$, so

$$\begin{aligned} p\,\varphi_1(v_i^2 c_i(x)/p) &= p\big(v_i^2 c_i(x)/p \\ &\qquad + \frac{1}{2} v_i^4 c_i(x)^2/p^2\big) \\ &= u_i c_i(x) + \frac{1}{2}(v_i^4/p) c_i(x)^2. \end{aligned}$$

If we remember the form of the classical augmented Lagrangian for inequality constraints (cf. (Bertsekas, 1982))

$$\begin{aligned} \mathcal{L}(x, u, \mu) &= f(x) \\ &\quad + \frac{\mu}{2} \sum_{i=1}^{m} \left(\max\left[0, u_i + c_i(x)/\mu\right]^2 - u_i^2 \right), \end{aligned}$$

we can see that the term v_i^4/p takes the place of the penalty parameter $1/\mu$ as long as the $v_i's$ remain bounded. This suggests that the method should behave similarly to the classical augmented Lagrangian method in a neighbourhood of attraction of a local minimum. In consequence, close to a minimum the updating strategy for p should resemble that for the classical parameter μ. In contrast, the algorithm may perform very differently when iterates are far away from local minima. Here the smoothness of the functions φ_1 may play favorably. We propose the update (16) for the penalty parameter, but other possibilities exist, see for instance Conn et al., and need to be tested and compared in the case of matrix inequality constraints.

3 DERIVATIVES OF SP FUNCTIONS

In this section we obtain formulas for the first- and second-order derivatives of the augmented Lagrangian function (13). This part is based on the differentiability theory for spectral functions developed by Lewis et al. (Lewis, 1996; Lewis, 2001; Lewis and S.Sendov, 2002). To begin with, recall the following definition from (Lewis and S.Sendov, 2002).

Definition 3.1 *Let $\lambda : \mathbb{S}^m \to \mathbb{R}^m$ denote the eigenvalue map $\lambda(X) := (\lambda_1(X), \ldots, \lambda_m(X))$. For a symmetric function $\psi : \mathbb{R}^m \to \mathbb{R}$, the spectral function Ψ associated with ψ is defined as $\Psi : \mathbb{S}^m \to \mathbb{R}$, $\Psi = \psi \circ \lambda$.*

First order differentiability theory for spectral functions is covered by (Lewis, 1996). We will need the following result from (Lewis, 1996):

Lemma 3.2 *Let ψ a symmetric function defined on an open subset D of $\mathbb{R}^n$. Let $\Psi = \psi \circ \lambda$ be the spectral function associated with ψ. Let $X \in \mathbb{S}^m$ and suppose*

$\lambda(X) \in D$. *Then* Ψ *is differentiable at* X *if and only if* ψ *is differentiable at* $\lambda(X)$. *In that case we have the formula:*

$$\nabla\Psi(X) = \nabla(\psi\circ\lambda)(X) = S\,\mathrm{diag}\,\nabla\psi\,(\lambda(X))\,S^T$$

for every orthogonal matrix S *satisfying*
$X = S\,\mathrm{diag}\lambda(X)\,S^T$. □

The gradient $\nabla\Psi(X)$ is a (dual) element of $\mathbb{S}^m$, which we distinguish from the differential, $d\Psi(X)$. The latter acts as a linear form on tangent vectors $dX \in \mathbb{S}^m$ as

$$\begin{aligned} d\Psi(X)[dX] &= \nabla\Psi(X)\bullet dX \\ &= \mathrm{Tr}\,\left(S\,\mathrm{diag}\,\nabla\psi\,(\lambda(X))S^T\,dX\right). \end{aligned}$$

It is now straightforward to compute the first derivative of a composite spectral functions $\Psi\circ\mathcal{F}$, where $\mathcal{F}:\mathbb{R}^n \to \mathbb{S}^m$ is sufficiently smooth. We obtain

$$\nabla\left(\Psi\circ\mathcal{F}\right)(x) = d\mathcal{F}(x)^*\left[\nabla\Psi\left(\mathcal{F}(x)\right)\right] \in \mathbb{R}^n, \quad (18)$$

where $d\mathcal{F}(x)$ is a linear operator mapping $\mathbb{R}^n \to \mathbb{S}^m$, and $d\mathcal{F}(x)^*$ its adjoint, mapping $\mathbb{S}^m \to \mathbb{R}^n$. With the standing notation

$$F := \mathcal{F}(x), \quad F_i := \frac{\partial\mathcal{F}(x)}{\partial x_i} \in \mathbb{S}^m,$$

we obtain the representations

$$\begin{aligned} d\mathcal{F}(x)[\delta x] &= \sum_{i=1}^n F_i\delta x_i, \\ d\mathcal{F}(x)^*[dX] &= \left(F_1\bullet dX,\ldots,F_n\bullet dX\right). \end{aligned}$$

Then the ith element of the gradient is

$$\begin{aligned} &\left(\nabla\left(\Psi\circ\mathcal{F}\right)(x)\right)_i \\ &\quad = \mathrm{Tr}\,\left(F_i S \mathrm{diag}\nabla\psi\left(\lambda(F)\right)S^T\right). \quad (19) \end{aligned}$$

Observe that $F_i = A_i + \sum_{j<i} B_{ji}x_j + \sum_{i<j} B_{ij}x_j$ if $\mathcal{B}$ is of the form (2), $F_i = A_i$ for affine operators $\mathcal{A}$.

We are now ready for the first derivative of the augmented Lagrangian function $L(x,V,p)$. We consider a special class of symmetric functions $\psi:\mathbb{R}^n \to \mathbb{R}$,

$$\psi(x) = \sum_{i=1}^n \varphi(x_i), \quad (20)$$

generated by a single scalar function $\varphi:\mathbb{R}\to\mathbb{R}$. Here the spectral function $\Psi = \psi\circ\lambda$ is of the form $\Psi = \mathrm{Tr}\ \Phi$ with $\Phi:\mathbb{S}^m \to \mathbb{S}^m$ the matrix-valued operator

$$\Phi(X) = S\,\mathrm{diag}\,\varphi\left(\lambda(X)\right)\,S^T,$$

S being an orthogonal matrix diagonalizing X. Using φ_p, ψ_p and Ψ_p instead of φ, ψ and Ψ, the penalty term in (13) takes the form

$$\begin{aligned} \Psi_p(V^T\mathcal{F}(x)V)) &= \psi_p\circ\lambda(V^T\mathcal{F}(x)V)) \\ &= \sum_{i=1}^m \varphi_p(\lambda_i(V^T\mathcal{F}(x)V)) \\ &= \mathrm{Tr}\,\Phi_p(V^T\mathcal{F}(x)V), \end{aligned}$$

where Φ_p is given in (10). Now we apply Lemma 3.2, respectively formula (18) or even (19) to $\Psi_p = \mathrm{Tr}\ \Phi_p$ and the operator $\widetilde{\mathcal{F}}(x) = V^T\mathcal{F}(x)V$, where we observe *en passant* that

$$\begin{aligned} d\widetilde{\mathcal{F}}(x)^*[X] &= d\left(\left\{V^T\mathcal{F}V\right\}(x)\right)^*[X] \\ &= d\mathcal{F}(x)^*[VXV^T], \end{aligned}$$

or put differently, $\widetilde{F}_i = V^T F_i V$. This gives the following

Proposition 3.3 *With the above notations the gradient of* $L(x,V,p)$ *is given as*

$$\begin{aligned} &\frac{\partial}{\partial x_i}L(x,V,p) = \frac{\partial}{\partial x_i}f(x) \\ &\quad + \mathrm{Tr}\,\left(F_i\,V\,S\,\mathrm{diag}\left[\varphi_p'(\lambda_i(V^TFV))\right]S^T\,V^T\right). \end{aligned}$$

In vector notation

$$\begin{aligned} &\nabla L(x,V,p) = \nabla f(x) + \left[\mathrm{vec}\,(F_1),\ldots,\mathrm{vec}\,(F_n)\right]^T \\ &\quad\cdot\ \mathrm{vec}\,\left(VS\,\mathrm{diag}\left[\varphi_p'(\lambda_i(V^TFV))\right]S^TV^T\right), \end{aligned}$$

where S *is orthogonal and diagonalizes* V^TFV. □

3.1 Second derivatives

The second order theory of spectral function is more involved and presented in (Lewis, 2001; Lewis and S.Sendov, 2002). We require the following central observation.

Lemma 3.4 *With the same notations as above,* $\Psi = \psi\circ\lambda$ *is twice differentiable at* $X \in \mathbb{S}^m$ *if and only if* ψ *is twice differentiable at* $\lambda(X) \in D$. □

We specialize again to the class of spectral functions $\Psi = \mathrm{Tr}\ \Phi$ generated by a single scalar function φ, see (20). Then we have the following result imported from (Shapiro, 2002):

Lemma 3.5 *Let* $\varphi:\mathbb{R}\to\mathbb{R}$ *be a function on the real line, and let* ψ *be the symmetric function on* $\mathbb{R}^n$ *generated by (20) above. Let* $\Psi = \mathrm{Tr}\ \Phi$ *denote the spectral function associated with* ψ. *Suppose* φ *is twice differentiable on an open interval* I. *Then* Ψ *is twice differentiable at* X *whenever* $\lambda_i = \lambda_i(X) \in I$ *for every* i. *Moreover, if we define the matrix* $\varphi'^{[1]} = \varphi'^{[1]}(X) \in \mathbb{S}^m$ *as:*

$$\varphi'^{[1]}_{ij} := \begin{cases} \frac{\varphi'(\lambda_i)-\varphi'(\lambda_j)}{\lambda_i-\lambda_j} & \textit{if} \quad \lambda_i \neq \lambda_j \\ \varphi''(\lambda_i) & \textit{if} \quad \lambda_i = \lambda_j \end{cases},$$

then the following formula defines the second order derivative of Ψ *at* X

$$d^2\Psi(X))[dX,dX] = \mathrm{Tr}\,(S^TdXS\varphi'^{[1]}\circ\{S^TdXS\}),$$

where S *is orthogonal and diagonalizes* X. □

In order to obtain second derivatives of composite functions, $\Psi \circ \mathcal{F}$, we will require the chain rule, which for the second differential forms $d^2\Psi(X)[dX, dX]$ of Ψ and $d^2\mathcal{F}(x)[\delta x, \delta x]$ of $\mathcal{F}$ amounts to:

$$\begin{aligned} d^2\left(\Psi \circ \mathcal{F}\right)(x)[\delta x, \delta x] \\ = d\Psi\left(\mathcal{F}(x)\right)\left[d^2\mathcal{F}(x)[\delta x, \delta x]\right] \\ +\ d^2\Psi\left(\mathcal{F}(x)\right)\left[d\mathcal{F}(x)[\delta x], d\mathcal{F}(x)[\delta x]\right]. \quad (21) \end{aligned}$$

For bilinear $\mathcal{B}$, the second order form $d^2\mathcal{B}(x)$ is of course independent of x and given as

$$d^2\mathcal{B}(x)[\delta x, \delta x] \ = \ \sum_{i<j} \delta x_i \delta x_j B_{ij}.$$

In particular, $d^2\mathcal{A} = 0$ for affine operators $\mathcal{A}$. During the following we shall use these results to obtain second derivatives of the augmented Lagrangian function $L(x, V, p)$.

As in the previous section, we apply the results to $\Psi_p \ = \ \mathrm{Tr}\ \Phi_p$ generated by φ_p and to the operator $\widetilde{\mathcal{F}}(x) \ = \ V^T\mathcal{F}(x)V$. As we have already seen, for $\widetilde{\mathcal{F}}$ we have the formula

$$\begin{aligned} d\widetilde{\mathcal{F}}(x)[\delta x] \quad &= \quad \sum_{i=1}^{n} V^T F_i V \delta x_i \\ &= \quad V^T \sum_{i=1}^{n} F_i \delta x_i V \\ &= \quad V^T d\mathcal{F} V. \end{aligned}$$

Therefore, using Lemma 3.5, the second term in (21) becomes

$$\begin{aligned} & d^2\Psi_p(V^TFV)[V^Td\mathcal{F}\,V, V^Td\mathcal{F}\,V] \\ & = \mathrm{Tr}\ \left(S^TV^T\, d\mathcal{F}\, V\, S\, \varphi_p^{'[1]} \circ \{S^T\, V^T\, d\mathcal{F}\, V\, S\}\right), \end{aligned}$$

where S is orthogonal and diagonalizes V^TFV and $\varphi_p^{'[1]} = \varphi_p^{'[1]}(V^TFV)$. This may now be transformed to vector-matrix form using the operator vec and the relations (4) - (7).

$$\begin{aligned} d^2\Psi_p(V^TFV)[V^Td\mathcal{F}\,V, V^Td\mathcal{F}\,V] \\ = \mathrm{vec}^T\left((VS)^T d\mathcal{F}\,(VS)\right) \\ \cdot\, \mathrm{vec}\left(\varphi_p^{'[1]} \circ \{(VS)^T\, d\mathcal{F}\,(VS)\}\right) \\ = \mathrm{vec}^T(d\mathcal{F})\,\mathcal{K}_p\ \mathrm{vec}\,(d\mathcal{F})\,, \quad (22) \end{aligned}$$

where the matrix $\mathcal{K}_p$ is

$$\begin{aligned} \mathcal{K}_p = [(VS) \otimes (VS)]\mathrm{diag}\left\{\mathrm{vec}\,\varphi_p^{'[1]}\right\} \\ \cdot\,[(VS)^T \otimes (VS)^T]. \end{aligned}$$

Using $d\mathcal{F} = \sum_{i=1}^n F_i \delta x_i$, the last line in (22) gives

$$\begin{aligned} & d^2\Psi_p(V^TFV)[V^Td\mathcal{F}\,V, V^Td\mathcal{F}\,V] \\ & \quad = \delta x^T\left[\mathrm{vec}\,(F_1), \ldots, \mathrm{vec}\,(F_n)\right]^T \mathcal{K}_p \\ & \qquad \cdot\,\left[\mathrm{vec}\,(F_1), \ldots, \mathrm{vec}\,(F_n)\right]\,\delta x, \end{aligned}$$

so we are led to retain the symmetric $n \times n$ matrix

$$\begin{aligned} H_1 \ = \ & \left[\mathrm{vec}\,(F_1), \ldots, \mathrm{vec}\,(F_n)\right]^T \mathcal{K}_p \\ & \cdot\,\left[\mathrm{vec}\,(F_1), \ldots, \mathrm{vec}\,(F_n)\right]. \quad (23) \end{aligned}$$

Let us now consider the second part of the Hessian $\nabla^2L(x, V, p)$, coming from the term $d\Psi\left(\mathcal{F}(x)\right)\left[d^2\mathcal{F}(x)[\delta x, \delta x]\right]$ in (21). First observe that trivially

$$\begin{aligned} d^2\widetilde{\mathcal{F}}(x)[\delta x, \delta x] \quad &= \quad d^2\{V^T\mathcal{F}V\}(x)[\delta x, \delta x] \\ &= \quad V^Td^2\mathcal{F}(x)[\delta x, \delta x]\,V. \end{aligned}$$

Hence

$$\begin{aligned} & d\Psi_p(V^TFV)\left[V^Td^2\mathcal{F}(x)[\delta x, \delta x]V\right] \\ & \quad = \mathrm{Tr}\ \left(S\,\mathrm{diag}\,\varphi_p'\left(\lambda_i(V^TFV)\right)\,S^T\right. \\ & \qquad \left.\cdot V^Td^2\mathcal{F}(x)[\delta x, \delta x]V\right) \\ & \quad = \mathrm{Tr}\ \left(V\,S\,\mathrm{diag}\,\varphi_p'\left(\lambda_i(V^TFV)\right)\,S^TV^T\right. \\ & \qquad \left.\cdot d^2\mathcal{F}(x)[\delta x, \delta x]\right). \quad (24) \end{aligned}$$

Introducing matrices $F_{ij} = F_{ij}(x) \in \mathbb{S}^m$ to represent

$$d^2\mathcal{F}(x)[\delta x, \delta x] \ = \ \sum_{i,j=1}^{n} F_{ij}\delta x_i \delta x_j,$$

relation (24) may be written as

$$\begin{aligned} & d\Psi_p(V^TFV)\left[V^Td^2\mathcal{F}(x)[\delta x, \delta x]V\right] \\ & \quad = \sum_{i,j=1}^{n} \delta x_i \delta x_j \mathrm{Tr}\ \left(F_{ij}V\,S\,\mathrm{diag}\,\varphi_p'\left(\lambda_i(V^T\right.\right. \\ & \qquad \left.\left.\cdot FV)\right)\,S^TV^T\right). \end{aligned}$$

Hence the interest to define a second symmetric $n \times n$ matrix H_2 by

$$(H_2)_{ij} = \mathrm{Tr}\ \left(F_{ij}VS\mathrm{diag}\varphi_p'\left(\lambda_i(V^TFV)\right)S^TV^T\right), \quad (25)$$

where as before S is orthogonal and diagonalizes V^TFV. We summarize by the following

Proposition 3.6 *The Hessian of the augmented Lagrangian function $L(x, V, p)$ is of the form*

$$\nabla^2L(x, V, p) \ = \ \nabla^2f(x) + H_1 + H_2,$$

where H_1, given by (23), is positive semidefinite. H_2 is given by (25).

Proof. The only element which remains to be proved is non-negativity of H_1, which hinges on the non-negativity of $\mathcal{K}_p$, and hence on that of $\varphi_p^{'[1]}$. The latter is a consequence of the convexity of φ_p, hence the result. □

3.2 Multiplier update rule

The first-order multiplier update rule is derived similarly to the classical case of polyhedral constraints. For our analysis we will require the traditional Lagrangian of problem (8), which is

$$L(x, U) \;=\; f(x) + U \bullet \mathcal{F}(x). \qquad (26)$$

Suppose x^+ is the solution of problem (14) in step 2 of our algorithm. Let $U = VV^T$ the current Lagrange multiplier estimate. A new multiplier U^+ and therefore a new factor estimate V^+ with $U^+ = V^+V^{+T}$ is then obtained by equating the gradients of the augmented Lagrangian in (13) and the traditional Lagrangian of problem (1). Writing $L(x, U) = f(x) + \operatorname{Tr}(U\mathcal{F}(x)) = f(x) + \operatorname{Tr}(VV^T\mathcal{F}(x)) = f(x) + \operatorname{Tr}(V^T\mathcal{F}(x)V)$, and computing its gradient at x^+ and $U^+ = V^+V^{+T}$ implies

$$\begin{aligned}&\nabla L(x^+, U^+) = \nabla f(x^+)\\ &\qquad + [\operatorname{vec}(F_1), \ldots, \operatorname{vec}(F_n)]^T \operatorname{vec}(V^+V^{+T}).\end{aligned}$$

Comparing with

$$\begin{aligned}&\nabla L(x^+, V^+, p)\\ &\quad = \nabla f(x^+) + [\operatorname{vec}(F_1), \ldots, \operatorname{vec}(F_n)]^T\\ &\quad \cdot \operatorname{vec}\left(V\,S\,\operatorname{diag}\varphi_p'(\lambda_i(V^T\mathcal{F}(x^+)V)S^TV^T\right)\end{aligned}$$

suggests the update rule

$$V^+V^{+T} = VS\,[\operatorname{diag}\varphi_p'(\lambda_i(V^T\mathcal{F}(x^+)V))]\,(VS)^T, \qquad (27)$$

where S is orthogonal and diagonalizes $V^T\mathcal{F}(x^+)V$. This was already presented in our algorithm. The argument suggests the equality to hold only modulo the null space of $[\operatorname{vec}(F_1), \ldots, \operatorname{vec}(F_n)]^T$, but the limiting case gives more information. Notice that equality

$$\nabla L(\bar{x}, \bar{U}) = \nabla L(\bar{x}, \bar{V}, p)$$

holds at a Karush-Kuhn-Tucker pair $(\bar{x}, \bar{U})$ and $\bar{U} = \bar{V}\bar{V}^T$ for every $p > 0$, a formula whose analogue in the classical case is well-known. That means we would (27) expect to hold at $x^+ = \bar{x}$, $V = V^+ = \bar{V}$. This is indeed the case since by complementarity, $\bar{V}^T\mathcal{F}(\bar{x})\bar{V} = 0$. With $\varphi_p'(0) = 1$ for every $p > 0$, the right hand side in (27) is $\bar{V}^TS^TS\bar{V} = \bar{V}^T\bar{V}$, hence equals the left hand side. We may therefore read (27) as some sort of fixed-point relation, which is satisfied in the limit $V = V^+$ if iterates converge. Notice, however, that convergence has to be proved by an extra argument, because (27) is not of contraction type.

Notice that by construction the right hand term in (27) is indeed positive definite, so V^+ could for instance be chosen as a Cholesky factor.

3.3 Solving the subproblem - implementational issues

Efficient minimization of $L(x, V_j, p_j)$ for fixed V_j, p_j is at the core of our approach and our implementation is based on a Newton trust region method, following the lines of Lin and Moré (Lin and More, 1998). As compared to Newton line search algorithms or other descent direction methods, trust regions can take better advantage of second-order information. This is witnessed by the fact that negative curvature in the tangent problem Hessian, frequently arising in BMI-minimization when iterates are still far away from any neighbourhood of local convergence, may be taken into account. Furthermore, trust region methods often (miraculously) find good local minima, leaving the bad ones behind. This additional benefit is in contrast with what line search methods achieve, and is explained to some extent by the fact that, at least over the horizon specified by the current trust region radius, the minimization in the tangent problem is a global one.

As part of our testing, and at an early stage of the development, we studied the behavior of our code on a class of LMI problems, where we compared line search and trust region approaches. Even in that situation, where line search is supposedly at its best due to convexity, the trust region variant appeared to be more robust, though often slower. This is significant when problems get ill-conditioned, which is often the case in automatic control applications. Then the Newton direction, giving descent in theory, often fails to do so due to ill-conditioning. In that event recourse to first-order methods has to be taken. This is where a trust region strategy continues to perform reasonably well.

As currently implemented, both the trust region and line search variant of our method apply a Cholesky factorization to the tangent problem Hessian. This serves either to build an efficient preconditioner, or to compute the Newton direction in the second case. When the Hessian is close to becoming indefinite, a modified Cholesky factorization is used ($\nabla^2 L + E = JJ^T$, with J a lower triangular matrix and E a shift matrix of minimal norm). This is a critical point of the algorithm, particularly for our hard problems, because the condition number of the shifted Hessian has to be reasonable to avoid large computational roundoff errors. However, trading a larger condition number may be often desirable, for example when the Hessian at the solution is ill-conditioned. We found that the revised modified Cholesky algorithm proposed by (Schnabel and Eskow, 1999) achieves this goal fairly well.

There are a few more features of our approach, which in our opinion merit special attention as they seem to be typical for automatic control applications.

A first issue is the magnitude of the decision variables, which may vary greatly within a given problem. In particular the range of the Lyapunov variables may show differences of several orders of magnitude, a phenomenon which is difficult to predict in advance, as the physical meaning of the Lyapunov variables is often lacking. Properly scaling these variables is therefore difficult in general.

A similar issue is to give prior bounds on the controller gains. This is mandatory because these values are to be useful in the hard-ware implementations of the theoretically computed controllers. Exceedingly large gains would bear the risk to amplify measurement noise and thwart the process. This may also be understood as a quest on the well-posedness of the initial control problem, which should include explicit gain bound constraints.

Incomplete knowledge about the numerical range of the different variables is in our opinion the source of most numerical difficulties and a method to balance data is extremely important.

4 NUMERICAL EXAMPLES

In this section we test our code on a variety of difficult applications in automatic control. The first part uses examples from static output feedback control. We then discuss applications like sparse linear constant output-feedback design, simultaneous state-feedback stabilization with limits on feedback gains and finally multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$-controller design. Here we import examples from (Hassibi et al., 1999). In all these cases our method achieves closed-loop stability, while performance indices like $\mathcal{H}_\infty$, $\mathcal{H}_2$ or mixed performances previously published in the literature are significantly improved. We always start our algorithm at $x_0 = 0$ in order to capture a realistic scenario, where no such prior information is available.

4.1 Static output feedback controller synthesis

We first recall a BMI characterization of the classical output feedback synthesis problem. To this end let $P(s)$ be an LTI (Linear Time Invariant) system with state-space equations:

$$P(s): \quad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{28}$$

where

- $x \in \mathbb{R}^{n_1}$ is the state vector,
- $u \in \mathbb{R}^{m_2}$ is the vector of control inputs,
- $w \in \mathbb{R}^{m_1}$ is a vector of exogenous inputs,
- $y \in \mathbb{R}^{p_2}$ is the vector of measurements,
- $z \in \mathbb{R}^{p_1}$ is the controlled or performance vector.
- $D_{22} = 0$ is assumed without loss of generality.

Let $T_{w,z}(s)$ denote the closed-loop transfer functions from w to z for some static output-feedback control law:

$$u = Ky\,. \tag{29}$$

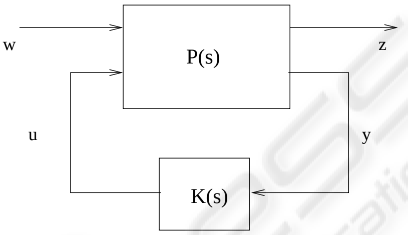

Figure 1: Output-feedback Linear Fractional Transform.

Our aim is to compute K subject to the following constraints:

- *internal stability:* for $w = 0$ the state vector of the closed-loop system (28) and (29) tends to zero as time goes to infinity.
- *performance:* the $\mathcal{H}_\infty$ norm $\|T_{w,z}(s)\|_\infty$, respectively the $\mathcal{H}_2$ norm $\|T_{w,z}(s)\|_2$, is minimized where the closed-loop transfer $T_{w,z}(s)$ is described as :

$$\left\{ \begin{array}{l} \dot{x} = (A + B_2KC_2)x + (B_1 + B_2KD_{21})w \\ z = (C_1 + D_{12}KC_2)x + (D_{11} + D_{12}KD_{21})w. \end{array} \right.$$

Recall that in the $\mathcal{H}_2$ case, some feedthrough terms must be nonexistent in order for the $\mathcal{H}_2$ performance to be well defined. We have to assume $D_{11} = 0$, $D_{21} = 0$ for the plant in (28). This guaranteed, both static $\mathcal{H}_\infty$ and $\mathcal{H}_2$ indices are then transformed into a matrix inequality condition using the bounded real lemma (Anderson and Vongpanitlerd, 1973). This leads to well-know characterizations:

Proposition 4.1 *A stabilizing static output feedback controller K with $\mathcal{H}_\infty$ gain $\|T_{w,z}(s)\|_\infty \le \gamma$ exists provided there exists $X \in \mathbb{S}^{n_1}$ such that:*

$$\begin{bmatrix} (A+B_2KC_2)^T X + * & * & * \\ (B_1+B_2KD_{21})^T X & -\gamma I & * \\ (C_1+D_{12}KC_2) & (D_{11}+D_{12}KD_{21}) & -\gamma I \end{bmatrix} \prec 0 \tag{30}$$

$$x \succ 0. \tag{31}$$

Proposition 4.2 *A stabilizing static output feedback controller K with $\mathcal{H}_2$ performance $\|T_{w,z}(s)\|_2 \le \sqrt{\gamma}$ exists provided there exist $X, M \in \mathbb{S}^{n_1}$ such that:*

$$\begin{bmatrix} (A+B_2KC_2)^T X + * & * \\ (C_1+D_{12}KC_2) & -I \end{bmatrix} \prec 0 \qquad (32)$$

$$\begin{bmatrix} M & * \\ X\,(B_1+B_2KD_{21}) & X \end{bmatrix} \succ 0 \qquad (33)$$

$$\mathrm{Tr}\,(M) \leq \gamma. \qquad (34)$$

In order to convert these programs to the form (1), we have to replace strict inequalities $\prec 0$ by $\preceq -\epsilon$ for a suitable threshold. The choice of $\epsilon > 0$ poses no problem in practice.

4.1.1 Transport airplane (TA)

We consider the design of a static $\mathcal{H}_\infty$ controller for a transport airplane from (Gangsaas et al., 1986). Our algorithm computes a static controller K after solving 29 unconstrained minimization subproblems (14):

$$K_\infty = \begin{bmatrix} 0.69788 \\ -0.64050 \\ -0.83794 \\ 0.09769 \\ 1.57062 \end{bmatrix}^T .$$

The associated $\mathcal{H}_\infty$ performance is 2.22 and represents 30% improvement over the result in (Leibfritz, 1998).

4.1.2 VTOL helicopter (VH)

The state-space data of the VTOL helicopter are borrowed from (Keel et al., 1988). They are obtained by linearizing the helicopter rigid body dynamics at given flight conditions. This leads to a fourth-order model. Both $\mathcal{H}_\infty$ and $\mathcal{H}_2$ static controllers are computed with the spectral SDP method. The algorithm converges after solving 29 tangent problems (14):

$$K_\infty = \begin{bmatrix} 0.50750 \\ 10.00000 \end{bmatrix}.$$

The final $\mathcal{H}_\infty$ performance is $1.59e$-1, which is about 40% improvement over the performance obtained in (Leibfritz, 1998).

$$K_2 = \begin{bmatrix} 0.13105 \\ 5.95163 \end{bmatrix},$$

which gives a $\mathcal{H}_2$ performance of $9.54e$-2 after solving 28 subproblems.

4.1.3 Chemical reactor (CR)

A model description for the chemical reactor can be found in (Hung and MacFarlane, 1982). Both $\mathcal{H}_\infty$ and $\mathcal{H}_2$ static controllers are computed with the spectral SDP method, which requires solving respectively 28 and 41 subproblems. The resulting gains are:

$$K_\infty = \begin{bmatrix} -10.17039 & -31.54413 \\ -26.01021 & -100.00000 \end{bmatrix},$$

Table 1: Static output feedback synthesis. n and m give size of (1), iter counts instances of (14), cpu in seconds, index gives the corresponding $\mathcal{H}_\infty$ or $\mathcal{H}_2$ performance.

Problem	n	m	iter	cpu	index
TA ($\mathcal{H}_\infty$)	51	30	29	39	2.22
VH ($\mathcal{H}_\infty$)	13	12	29	0.3	$1.59e$-1
VH ($\mathcal{H}_2$)	16	13	28	0.5	$9.54e$-2
CR ($\mathcal{H}_\infty$)	15	18	28	1.3	1.17
CR ($\mathcal{H}_2$)	25	19	41	24.6	1.94
PA ($\mathcal{H}_\infty$)	19	13	30	13.5	$2.19e$-3

with $\mathcal{H}_\infty$-performance $\|T_{w,z}(s)\|_\infty = 1.17$.

$$K_2 = \begin{bmatrix} 0.35714 & -2.62418 \\ 2.58159 & 0.77642 \end{bmatrix},$$

with $\mathcal{H}_2$ performance $\|T_{w,z}(s)\|_2 = 1.94$.

4.1.4 Piezoelectric actuator (PA)

In this section, we consider the design of a static controller for a piezoelectric actuator system. A comprehensive presentation of this system can be found in (Chen, 1998). Our algorithm computes a static $\mathcal{H}_\infty$ controller after solving 30 instances of (14):

$$K_\infty = [0.09659 \quad -1.45023 \quad -100.00] .$$

The computed $\mathcal{H}_\infty$-performance $2.19e$-3 improves significantly over the value 0.785 given in (Leibfritz, 1998).

4.2 Miscellaneous examples

4.2.1 Sparse linear constant output-feedback design

In many applications it is of importance to design sparse controllers with a small number of nonzero entries. This is for instance the case when it is mandatory to limit the number of arithmetic operations in the control law, or when reducing the number of sensor/actuator devices is critical. In such a case one may attempt to synthesis a controller with an imposed sparsity structure. A more flexible idea is to let the problem find its own sparsity structure by using a ℓ_1-norm cost function. We solve the following (BMI) problem:

$$\begin{array}{ll} \text{min} & \sum_{\substack{1\le i\le n,\\ 1\le j\le n}} |K_{ij}|, \\ \text{s. t.} & P \succeq \epsilon I \\ & (A + BKC)^T P + P(A + BKC) \\ & \qquad \preceq -2\,\alpha P - \epsilon I \end{array}$$

where ϵ is the usual threshold to adapt strict inequalities to the form (1) or (8). This corresponds to designing a sparse linear constant output feedback control $u = Ky$ for the system $\dot{x} = Ax + Bu$, $y = Cx$ with an imposed decay rate of at least α in the closed-loop system.

Minimizing the ℓ_1-norm of the feedback gain is a good heuristic for obtaining sparse feedback gain matrices. This may be explained by the fact that the function $\phi(t) = |t|$ is the natural continuous interpolant of the counting function $c(t) = \lfloor t \rfloor$. Minimizing the function $\sum_{ij} \lfloor K_{ij} \rfloor$ or even counting the nonzero entries K_{ij} in K is what we would really like to do, but this is a discrete optimization problem subject to matrix inequality constraints.

There are many more applications to this heuristic. Finding sparse feedback gain matrices is also a good way of solving the actuator/sensor placement or controller topology design problems. In our example the method works nicely and a sparsity pattern is apparent. If small but nonzero values occur, one may set these values to zero if below a certain threshold, derive a sparsity pattern, and attempt a structured design with that pattern imposed. This leads to another class of BMI problems.

In our example, the system is initially unstable with rate of $\alpha_0 = -0.2832$. This value is obtained by computing the smallest negative real part of the eigenvalues of A (data available from (Hassibi et al., 1999)).

In a first test similar to the one in (Hassibi et al., 1999), we seek a sparse K so that the decay rate of the closed-loop system is not less than 0.35. We obtain the following gain matrix:

$$K_\infty = \begin{bmatrix} 0.2278 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.3269 & 0.0000 & 0.0000 \end{bmatrix},$$

with only two non-zeros elements. The path-following method presented in (Hassibi et al., 1999) gives 3 non-zeros elements. The spectral method gives once again a significant improvement, while maintaining the closed-loop stability.

If now in a second test a higher decay rate ($\alpha \geq 0.5$) (and so a better damped closed-loop system), is required, our method computes

$$K_\infty = \begin{bmatrix} 0.1072 & 0.0000 & 0.0000 & 0.0000 \\ -0.0224 & 0.0000 & 0.0000 & 0.0000 \\ 0.3547 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0186 & 0.1502 & 0.0000 & 0.0000 \end{bmatrix},$$

with 5 non-zeros entries. As expected, the sparsity decreases when α increases.

4.2.2 Simultaneous state-feedback stabilization with limits on feedback gains

One attempts here to stabilize three different linear systems using a common linear constant state-feedback law with limits on the feedback gains. Specifically, suppose that:

$$\dot{x} = A_k x + B_k u,\ u = Kx,\ k = 1, 2, 3.$$

The goal is to compute K such that $|K_{ij}| \leq K_{ij,max}$ and the three closed-loop systems:

$$\dot{x} = (A_k + B_k K)\,x\,, k = 1, 2, 3\,,$$

are stable. Under these constraints the worst decay-rate of the closed-loop systems is maximized.

Proposition 4.3 *The feedback gain K stabilizes the above systems simultaneously if and only if there exist $P_k \succ 0$, $\alpha_k > 0$, $k = 1, 2, 3$ such that*

$$(A_k + B_k K)^T P_k + P_k(A_k + B_k K) \prec -2\alpha_k P_k, \\ k = 1, 2, 3.$$

Introducing our usual threshold parameter $\epsilon > 0$, we have to check whether the value of the following BMI program is positive:

$$\begin{array}{ll} \text{maximize} & \min_{k=1,2,3} \alpha_k \\ \text{subject to} & |K_{ij}| \leq K_{ij,max}\,, \\ & (A_k + B_k K)^T P_k + P_k(A_k + B_k K) \\ & \qquad \preceq -2\,\alpha_k P_k - \epsilon I \\ & P_k \succeq \epsilon I, k = 1, 2, 3. \end{array}$$

By looking in (Hassibi et al., 1999) and computing the corresponding eigenvalues, the reader will notice that all three systems are unstable. We chose, as proposed by Hassibi, How and Boyd, $K_{ij,max} = 50$. Actually this optimization problem turned out one of the hardest for our algorithm. We obtain a solution after 68 unconstrained minimization subproblems, in 7.3 seconds. Actually a restart from a stabilizing point was done here, which explains the unexpectedly high number ($68 = 32 + 36$) of subproblems iterations.

On the other hand, this experiment gave the best improvement among our present tests. The worst decay-rate was increased to $\alpha = 3.42$, an improvement of more than 200% over the value 1.05 obtained in (Hassibi et al., 1999). Notice that none of the gain bounds was active at the optimum:

$$K_\infty = \begin{bmatrix} -42.4186 & -38.6457 & -17.4347 \\ 37.4186 & -15.9380 & -33.6942 \end{bmatrix}.$$

Table 2: Miscellaneaous examples.

Problem	n	m	iter	cpu
Sparse design 1	56	51	29	2.9
Sparse design 2	56	51	19	7.3
Simult. stab.	28	34	68	7.3
$\mathcal{H}_2/\mathcal{H}_\infty$	17	27	25	1.1

4.2.3 $\mathcal{H}_2/\mathcal{H}_\infty$ controller design

Mixed $\mathcal{H}_2/\mathcal{H}_\infty$-performance indices represent one of those prominent cases where the projection Lemma fails and direct BMI-techniques are required. Here we consider the case where a static state-feedback controller $u = Kx$ for the open loop system

$$\begin{aligned} \dot{x} &= Ax + B_1 u + B_2 w \\ z_1 &= C_1 x + D_{11} u \\ z_2 &= C_2 x + D_{21} u \end{aligned}$$

is sought for. The purpose is to find K such that the $\mathcal{H}_2$ norm $w \rightarrow z_2$ is minimized, while the $\mathcal{H}_\infty$-norm $w \rightarrow z_1$ is fixed below a performance level γ. After introducing our usual threshold $\epsilon > 0$, the following matrix inequality optimization problem arises:

$$\begin{aligned} &\text{minimize} \quad \eta^2 \\ &\text{subject to} \end{aligned}$$

$$\begin{bmatrix} \begin{pmatrix} (A + B_1 K)^T P_1 \\ +P_1(A + B_1 K) \\ +(C_1 + D_{11} K)^T (C_1 + D_{11} K) \end{pmatrix} & * \\ {B_2}^T P_1 & -\gamma^2 I \end{bmatrix} \preceq -\epsilon I$$

$$\begin{bmatrix} (A + B_1 K)^T P_2 + P_2(A + B_1 K) & * \\ B_2^T P_2 & -I \end{bmatrix} \preceq -\epsilon I$$

$$\begin{bmatrix} P_2 & * \\ C_2 & Z \end{bmatrix} \succeq \epsilon I, \ \mathrm{Tr}\,(Z) \leq \eta^2, \ P_1 \succeq \epsilon I, \ P_2 \succeq \epsilon I.$$

The first block contains multilinear terms, which may be converted to BMI-from using a Schur complement. With $\gamma = 2$, the algorithm needs 25 instances of (14) to compute the compensator:

$$K_{2,\infty} = [1.1500 \quad -0.4685 \quad -1.1500],$$

with corresponding $\mathcal{H}_2$-norm 0.8384. In this case the improvement over the result in (Hassibi et al., 1999) is negligible, which we interpret in the sense that this result was already close to the global optimum. (Nota bene, one cannot have any guarantee as to the truth of this statement. Indeed, in this example we found that many other local minima could be computed).

5 CONCLUSION

A spectral penalty augmented Lagrangian method for matrix inequality constrained nonlinear optimization programs was presented and applied to a number of small to medium-size test problems in automatic control.

The algorithm performs robustly if parameters are carefully tuned. This refers to the choice of the initial penalty value, p, the control of the condition number of the tangent problem Hessian $\nabla^2 L(x, V, p)$ in (14), and to the stopping tests (complementarity, stationarity), where we followed the lines of (Henrion et al., 2003). For the updates $p \rightarrow p^+$ and $V \rightarrow V^+$ we found that it is vital to avoid drastic changes, since the new subproblem (14) may be much more difficult to solve. This is particularly so for the nonconvex BMI case.

We remind the reader that similar to our previous approaches, the proposed algorithm is a local optimization method, which gives no certificate as to finding the global optimal solution of the control problem. If fact, such a method may in principle even fail completely by converging to a local minimum of constraint violation. For all that, our general experience, confirmed in the present testing, is that this type of failure rarely occurs. We consider the local approach largely superior for instance to a very typical class of methods in automatic control, where hard problems are solved by a tower of LMI problems with growing size, N, where a solution is guaranteed as $N \rightarrow \infty$. Such a situation is often useless, since the size of the LMI problem needed to solve the underlying hard problem is beyond reach, and since a prior estimate is not available or too pessimistic. The major advantage of local methods is that the subproblems, on whose succession the problem rests, all have the same dimension.

We finally point out that the local convergence of the algorithm will be proved in a forthcoming article.

REFERENCES

Anderson, B. and Vongpanitlerd, S. (1973). *Network analysis and synthesis: a modern systems theory approach*. Prentice-Hall.

Apkarian, P., Noll, D., Thevenet, J. B., and Tuan, H. D. (2003a). A Spectral Quadratic-SDP Method with Applications to Fixed-Order $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Synthesis. *submitted.* Rapport Interne , MIP, UMR 5640, Maths. Dept. - Paul Sabatier University.

Apkarian, P., Noll, D., and Tuan, H. D. (2002). Fixed-order $\mathcal{H}_\infty$ control design via an augmented Lagrangian method . *Rapport Interne 02-13, MIP, UMR 5640, Maths. Dept. - Paul Sabatier University* . `http://mip.ups-tlse.fr/publi/publi.html`.

Apkarian, P., Noll, D., and Tuan, H. D. (2003b). Fixed-order $\mathcal{H}_\infty$ control design via an augmented Lagrangian method . *Int. J. Robust and Nonlinear Control*, 13(12):1137–1148.

Ben-Tal, A. and Zibulevsky, M. (1997). Penalty/barrier multiplier methods for convex programming problems. *SIAM J. on Optimization*, 7:347–366.

Bertsekas, D. P. (1982.). *Constrained optimization and Lagrange multiplier methods*. Academic Press, London.

Chen, B. M. (1998). *H_∞ Control and Its Applications*, volume 235 of *Lectures Notes in Control and Information Sciences*. Springer Verlag, New York, Heidelberg, Berlin.

Conn, A. R., Gould, N., and Toint, P. L. (1991). A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.*, 28(2):545 – 572.

Conn, A. R., Gould, N. I. M., Sartenaer, A., and Toint, P. L. (1993a). Global Convergence of two Augmented Lagrangian Algorithms for Optimization with a Combination of General Equality and Linear Constraints. Technical Report TR/PA/93/26, CERFACS, Toulouse, France.

Conn, A. R., Gould, N. I. M., Sartenaer, A., and Toint, P. L. (1993b). Local convergence properties of two augmented Lagrangian algorithms for optimization with a combination of general equality and linear constraints. Technical Report TR/PA/93/27, CERFACS, Toulouse, France.

Conn, A. R., Gould, N. I. M., Sartenaer, A., and Toint, P. L. (1996). Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. *SIAM J. on Optimization*, 6(3):674 – 703.

Fares, B., Apkarian, P., and Noll, D. (2000). An Augmented Lagrangian Method for a Class of LMI-Constrained Problems in Robust Control Theory. In *Proc. American Control Conf.*, pages 3702–3705, Chicago, Illinois.

Fares, B., Apkarian, P., and Noll, D. (2001). An Augmented Lagrangian Method for a Class of LMI-Constrained Problems in Robust Control Theory. *Int. J. Control*, 74(4):348–360.

Fares, B., Noll, D., and Apkarian, P. (2002). Robust Control via Sequential Semidefinite Programming. *SIAM J. on Control and Optimization*, 40(6):1791–1820.

Gangsaas, D., Bruce, K., Blight, J., and Ly, U.-L. (1986). Application of modern synthesis to aircraft control: Three case studies. *IEEE Trans. Aut. Control*, AC-31(11):995–1014.

Hassibi, A., How, J., and Boyd, S. (1999). A pathfollowing method for solving bmi problems in control. In *Proc. American Control Conf.*, pages 1385–1389.

Henrion, D., M.Kocvara, and Stingl, M. (2003). Solving simultaneous stabilization BMI problems with PENNON. In *IFIP Conference on System Modeling and Optimization*, volume 7, Sophia Antipolis, France.

Hestenes, M. R. (1969). Multiplier and gradient method. *J. Optim. Theory Appl.*, 4:303 – 320.

Hung, Y. S. and MacFarlane, A. G. J. (1982). *Multivariable feedback: A classical approach*. Lectures Notes in Control and Information Sciences. Springer Verlag, New York, Heidelberg, Berlin.

Keel, L. H., Bhattacharyya, S. P., and Howze, J. W. (1988). Robust control with structured perturbations. *IEEE Trans. Aut. Control*, 36:68–77.

Kocvara, M. and Stingl, M. (2003). A Code for Convex Nonlinear and Semidefinite Programming. *Optimization Methods and Software*, 18(3):317–333.

Leibfritz, F. (1998). Computational design of stabilizing static output feedback controller. Rapports 1 et 2 Mathematik/Informatik, Forschungsbericht 99-02, Universitt Trier.

Lewis, A. (1996). Derivatives of spectral functions. *Mathematics of Operations Research*, 21:576–588.

Lewis, A. (2001). Twice differentiable spectral functions. *SIAM J. on Matrix Analysis and Applications*, 23:368–386.

Lewis, A. and S.Sendov, H. (2002). Quadratic expansions of spectral functions. *Linear Algebra and Appl.*, 340:97–121.

Lin, C. and More, J. (1998). Newton's method for large bound–constrained optimization problems. Technical Report ANL/MCS-P724–0898, Mathematics and Computer Sciences Division, Argonne National Laboratory.

Mosheyev, L. and Zibulevsky, M. (2000). Penalty/barrier multiplier algorithm for semidefinite programming. *Optimization Methods and Software*, 13(4):235–261.

Noll, D., Torki, M., and Apkarian, P. (2002). Partially Augmented Lagrangian Method for Matrix Inequality Constraints. *submitted*. Rapport Interne , MIP, UMR 5640, Maths. Dept. - Paul Sabatier University.

Powell, M. J. D. (1969). A method for nonlinear constraints in minimization problem. In Fletcher, R., editor, *Optimization*. Academic Press, London, New York.

Schnabel, R. B. and Eskow, E. (1999). A revised modified cholesky factorization algorithm. *SIAM J. on Optimization*, 9(4):1135–1148.

Shapiro, A. (2002). On differentiability of symmetric matrix valued functions. *School of Industrial and Systems Engineering, Georgia Institute of Technology*. Preprint.

Zibulevsky, M. (1996). *Penalty/Barrier Multiplier Methods for Large-Scale Nonlinear and Semidefinite Programming*. Ph. D. Thesis, Technion Isral Institute of Technology.