Solving polynomial static output feedback problems with PENBMI

Didier Henrion, Johan Löfberg, Michal Kočvara, Michael Stingl

Abstract— An algebraic formulation is proposed for the static output feedback (SOF) problem: the Hermite stability criterion is applied on the closed-loop characteristic polynomial, resulting in a non-convex bilinear matrix inequality (BMI) optimization problem for SIMO or MISO systems. As a result, the BMI problem is formulated directly in the controller parameters, without additional Lyapunov variables. The publicly available solver PENBMI 2.0 interfaced with YALMIP 3.0 is then applied to solve benchmark examples. Implementation and numerical aspects are widely discussed.

Index Terms— Static output feedback, BMI optimization, polynomials.

I. INTRODUCTION

Even though several relevant control problems boil down to solving convex linear matrix inequalities (LMI) –see [4] for a long list – there are still fundamental problems for which no convex LMI formulation has been found. The most fundamental of these problems is perhaps static output feedback (SOF) stabilization: given a triplet of state-space matrices A,B,C of suitable dimensions, find a matrix K such that the eigenvalues of matrix $A + BKC$ are all in a given region of the complex plane, say the open left half-plane.

No LMI formulation is known for the SOF stabilization problem, but a straightforward application of Lyapunov's stability theory leads to a bilinear matrix inequality (BMI) formulation. The BMI formulation of control problems was made popular in the mid 1990s [6]; at that time there were, however, no computational methods for solving non-convex BMIs, in contrast with convex LMIs for which powerful interior-point algorithms were available.

One decade later, this unsatisfactory state in BMI solvers is almost unchanged. Several researchers have tried to apply global or nonlinear optimization techniques to BMI problems, with moderate success so far. To our knowledge, PENBMI [10] is the first available general-purpose solver for BMIs. The algorithm, described in [9], is based on the augmented Lagrangian method. It can be viewed as a generalization to nonlinear semidefinite problems of the

D. Henrion is with LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France. He is also with the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic.

J. Löfberg is with the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Physikstrasse 3, ETL I22, 8092 Zürich, Switzerland.

M. Kočvara is with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Prague, Czech Republic. He is also with the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic.

M. Stingl is with the Institute of Applied Mathematics, University of Erlangen, Martensstrasse 3, 91058 Erlangen, Germany.

penalty-barrier-multiplier method originally introduced in [3] for convex optimization. Convergence to a critical point satisfying first order KKT optimality conditions is guaranteed. The solver PENBMI is fully integrated within the Matlab environment through version 3.0 of the YALMIP interface [13].

When following a state-space approach, the SOF problem can be formulated as the BMI

$$(A + BKC)^\star P + P(A + BKC) \prec 0, \quad P = P^\star \succ 0$$

in decision variables K and P where $\prec 0$ and $\succ 0$ stand for negative and positive definite, respectively, and the star denotes the conjugate transpose. If n, m, p denote the state, input, and output dimensions respectively, we see that SOF matrix K (the actual problem unknown) contains nm scalar entries, whereas Lyapunov matrix P (instrumental to ensuring stability) contains $n(n+1)/2$ scalar entries. When n is significantly larger than mp, the large number of resulting Lyapunov variables may be computationally prohibitive.

A first contribution of this paper is to propose an alternative BMI formulation of the SOF problem featuring entries of matrix K only. In order to get rid of the Lyapunov variables, we focus on a polynomial formulation of the SOF problem, applying the Hermite stability criterion on the closed-loop characteristic polynomial, in the spirit of [7]. The resulting matrix inequality constraint is bilinear[1] (BMI) when $m = 1$ (SIMO systems) or $p = 1$ (MISO systems).

A second contribution of this paper is in reporting numerical examples showing that PENBMI can indeed prove useful in solving non-trivial SOF problems formulated in this polynomial setting. The problems are extracted from the publicly available benchmark collection COMPl$_e$ib [11].

II. PROBLEM STATEMENT

Consider the linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

of order n with m inputs and p outputs, that we want to stabilize by static output feedback

$$u = Ky.$$

In other words, given matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, we want to find matrix $K \in \mathbb{R}^{m \times p}$ such that the

[1]Strictly speaking, the matrix constraint is quadratic, not bilinear. By a slight abuse of terminology and to avoid introducing new acronyms, we use the term BMI to refer to these quadratic matrix inequalities.

eigenvalues of closed-loop matrix $A + BKC$ all belong to a region

$$\mathcal{D} = \{s \in \mathbb{C} : d_0 + d_1(s + s^\star) + d_2 s s^\star < 0\}$$

of the complex plane, where $d_0, d_1, d_2 \in \mathbb{R}$ are given scalars. Typical choices are $d_0 = d_2 = 0$, $d_1 = 1$ for the left half-plane (continuous-time stability) and $d_2 = -d_0 = 1$, $d_1 = 0$ for the unit disk (discrete-time stability).

Problem SOF: *Given matrices A, B, C, find matrix K such that eigenvalues of matrix $A + BKC$ all belong to given stability region $\mathcal{D}$.*

III. PMI FORMULATION

A. Characteristic polynomial

Let $k \in \mathbb{R}^{mp}$ be the vector obtained by stacking the columns of matrix K. Define

$$q(s, k) = \det(sI - A - BKC) = \sum_{i=0}^{n} q_i(k) s^i \quad (1)$$

as the characteristic polynomial of matrix $A + BKC$. Coefficients of increasing powers of indeterminate s in polynomial $q(s, k)$ are multivariate polynomials in k, i.e.

$$q_i(k) = \sum_{\alpha} q_{i\alpha} k^\alpha \quad (2)$$

where $\alpha \in \mathbb{N}^{mp}$ describes all monomial powers.

B. Hermite stability criterion

The roots of polynomial $q(s, k)$ belong to stability region $\mathcal{D}$ if and only if

$$H(k) = \sum_{i=0}^{n} \sum_{j=0}^{n} q_i(k) q_j(k) H_{ij} \succ 0$$

where $H(k) = H^\star(k) \in \mathbb{R}^{n \times n}$ is the Hermite matrix of $q(s, k)$. Coefficients $H_{ij} = H_{ij}^\star \in \mathbb{R}^{n \times n}$ depend on the stability region $\mathcal{D}$ only, see [12] or [8].

C. Polynomial matrix inequality

Hermite matrix $H(k)$ depends polynomially on vector k, hence the equivalent notation

$$H(k) = \sum_{\alpha} H_\alpha k^\alpha \succ 0 \quad (3)$$

where matrices $H_\alpha = H_\alpha^\star \in \mathbb{R}^{n \times n}$ are obtained by combining matrices H_{ij}, and $\alpha \in \mathbb{N}^{mp}$ describes all monomial powers.

Lemma 3.1: Problem SOF is solved if and only if matrix K solves the PMI (3).

Corollary 3.2: Note that if $\operatorname{rank} B = 1$ (single input) or $\operatorname{rank} C = 1$ (single output) then $\deg q_i(k) = 1$ and hence $\deg H(k) = 2$, meaning that PMI (3) in Lemma 3.1 is a BMI.

IV. NUMERICAL ASPECTS

A. Computing the characteristic polynomial

In order to build up characteristic polynomial $q(s, k)$ we need to evaluate coefficients $q_{i\alpha}$ of the determinant of the first degree multivariate polynomial matrix $sI - A - BKC$.

One possible way could be to transform the multivariate polynomial matrix into some triangular form so that computation of its determinant would be reduced to a sequence of routine multivariate polynomial multiplications. We have chosen another strategy avoiding computationally costly symbolic computations. We proceed numerically by interpolation: coefficients of $q(s, k)$ are determined by solving a linear system of equation built on a truncated multivariate Vandermonde matrix.

Characteristic polynomial $q(s, k)$ is expressed in (1) in the standard multivariate power monomial basis with indeterminates $s, k_1, \ldots, k_{mp}$. In order to avoid introducing ill-conditioning into the numerical problem, we choose complex interpolation points uniformly distributed along the unit circle and we proceed by oversampling. The rectangular linear system of equations is over-determined, but with unitary vectors. As a result, only unitary matrix multiplication is used for solving the system and retrieving polynomial coefficients.

The idea is better illustrated with a simple example. Consider a two-variable polynomial

$$q(x_1, x_2) = q_{00} + q_{01} x_2 + q_{12} x_1 x_2^2.$$

We would like to interpolate coefficients q_{00}, q_{01} and q_{12} from values taken by $q(x_1, x_2)$ at given points x. Our polynomial is of degree 1 in variable x_1 and degree 2 in variable x_2, so we use 2 samples x_{10}, x_{11} for x_1, and 3 samples x_{20}, x_{21}, x_{22} for x_2. Let

$$V_1 = \left[\begin{array}{cc} x_{10}^0 & x_{10}^1 \\ x_{11}^0 & x_{11}^1 \end{array} \right], \quad V_2 = \left[\begin{array}{ccc} x_{20}^0 & x_{20}^1 & x_{20}^2 \\ x_{21}^0 & x_{21}^1 & x_{21}^2 \\ x_{22}^0 & x_{22}^1 & x_{22}^2 \end{array} \right]$$

be the corresponding Vandermonde matrices, and let

$$V = V_2 \otimes V_1$$

be the 6-by-6 Vandermonde obtained by sampling all possible second degree monomials, where $\otimes$ denotes the Kronecker product. Removing columns 2, 4 and 5 in V we obtain the over-determined consistent linear system of equations

$$Aq = \left[\begin{array}{ccc} x_{10}^0 x_{20}^0 & x_{10}^0 x_{20}^1 & x_{10}^1 x_{20}^2 \\ x_{11}^0 x_{20}^0 & x_{11}^0 x_{20}^1 & x_{11}^1 x_{20}^2 \\ x_{10}^0 x_{21}^0 & x_{10}^0 x_{21}^1 & x_{10}^1 x_{21}^2 \\ x_{11}^0 x_{21}^0 & x_{11}^0 x_{21}^1 & x_{11}^1 x_{21}^2 \\ x_{10}^0 x_{22}^0 & x_{10}^0 x_{22}^1 & x_{10}^1 x_{22}^2 \\ x_{11}^0 x_{22}^0 & x_{11}^0 x_{22}^1 & x_{11}^1 x_{22}^2 \end{array} \right] \left[\begin{array}{c} q_{00} \\ q_{01} \\ q_{12} \end{array} \right]$$

$$= \left[\begin{array}{c} q(x_{10}, x_{20}) \\ q(x_{11}, x_{20}) \\ q(x_{10}, x_{21}) \\ q(x_{11}, x_{21}) \\ q(x_{10}, x_{22}) \\ q(x_{11}, x_{22}) \end{array} \right] = b$$

linking the truncated Vandermonde matrix A, the vector q of coefficients and the vector b of values taken at the interpolation points.

If we choose complex samples $x_{10} = e^{i0}$, $x_{11} = e^{i\pi}$, and $x_{20} = e^{i0}$, $x_{21} = e^{i2\pi/3}$, $x_{22} = e^{i4\pi/3}$ uniformly along the unit disk, then normalized Vandermonde matrices $(\sqrt{2}/2)V_1$ and $(\sqrt{3}/3)V_2$ are unitary. Consequently, columns in matrix $(\sqrt{6}/6)A$ are orthonormal, so that

$$q = \frac{1}{6}A^\star b$$

and no matrix inversion is required to solve the interpolation problem.

Suppose that when evaluating a given polynomial $q(x_1, x_2)$ at these points we obtain

$$b = \begin{bmatrix} 6 \\ 0 \\ -1.5000 - i0.8660 \\ 1.5000 + i4.3301 \\ -1.5000 + i0.8660 \\ 1.5000 - i4.3301 \end{bmatrix}.$$

Its coefficients are then given by

$$q = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -0.5000 + i0.8660 & -0.5000 - i0.8660 \\ 1 & -0.5000 + i0.8660 & 0.5000 + i0.8660 \\ 1 & -0.5000 - i0.8660 & -0.5000 + i0.8660 \\ 1 & -0.5000 - i0.8660 & 0.5000 - i0.8660 \end{bmatrix}^\star b$$

$$= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

hence

$$q(x_1, x_2) = 1 + 2x_2 + 3x_1 x_2^2.$$

B. Building up the Hermite matrix

Once coefficients $q_i(k)$ of the characteristic polynomial are given, we need only coefficients H_{ij} of the Hermite matrix in order to build matrix inequality (3). These matrix coefficients depend only on the stability region $\mathcal{D}$. They are computed numerically by solving a simple linear system of equations, as shown in [8].

Note also that when $\mathcal{D}$ is the left half-plane ($d_0 = d_2 = 0$, $d_1 = 1$) the Hermite matrix has a special structure: by permuting odd and even rows and columns we obtain a block diagonal matrix with two blocks of half size. This block structure can be exploited when solving the BMI.

As an illustrative example, consider problem NN1 in [11]:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 13 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 5 & -1 \\ -1 & -1 & 0 \end{bmatrix}$$

with

$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}.$$

The characteristic polynomial is

$$\begin{aligned} q(s, k) &= \det(sI - A - BKC) \\ &= k_2 + (-13 - 5k_1 + k_2)s + k_1 s^2 + s^3 \\ &= q_0(k) + q_1(k)s + q_2(k)s^2 + q_3(k)s^3 \end{aligned}$$

The Hermite matrix corresponding to the left half-plane $\mathcal{D} = \{s \in \mathbb{C} : s + s^\star < 0\}$, after permutation of odd and even rows and columns, is given by

$$\begin{aligned} H(k) &= \begin{bmatrix} q_0 q_1 & q_0 q_3 & 0 \\ q_0 q_3 & q_2 q_3 & 0 \\ 0 & 0 & q_1 q_2 - q_0 q_3 \end{bmatrix} \\ &= \begin{bmatrix} -13k_2 - 5k_1 k_2 + k_2^2 & k_2 & 0 \\ k_2 & k_1 & 0 \\ 0 & 0 & -13k_1 - k_2 - 5k_1^2 + k_1 k_2 \end{bmatrix}. \end{aligned}$$

It is positive definite if and only if $k_1 > 0$, $-13 - 5k_1 + k_2 > 0$ and $-13k_1 - k_2 - 5k_1^2 + k_1 k_2 > 0$, which corresponds to the interior of a hyperbolic branch in the positive orthant, see Figure 1.
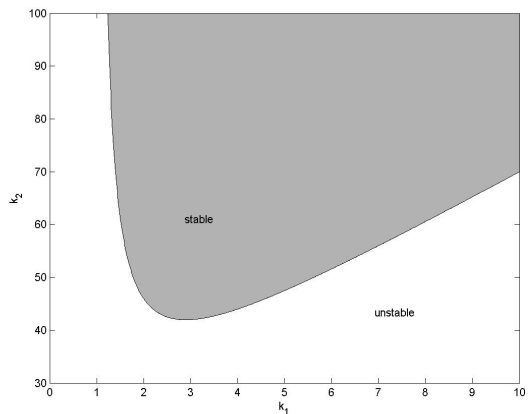

Fig. 1. SOF example NN1. Region of stabilizing gains (in gray).

C. Convexity and non-convexity

Notice that the stability region in the parameter space k_1, k_2 is convex for the example of section IV-B, see Figure 1. Using the classification established in [2], it is actually the convex branch of a hyperbola that can equivalently be described by the LMI

$$\begin{bmatrix} -1 + k_1 & 1 \\ 1 & -1 - \frac{5}{18}k_1 + \frac{1}{18}k_2 \end{bmatrix} \succ 0$$

which was not apparent by inspecting the original Hermite BMI. In other words, in this particular case, the SOF problem boils down to solving a convex LMI problem in the parameter space.

From these observations, it makes sense to apply a BMI solver which can exploit convexity of the optimization space. The algorithm implemented in PENBMI, as an extension of an algorithm originally developed for convex optimization, has this important feature.

Generally speaking, it would be interesting to design an algorithm detecting from the outset the hidden convexity of stability conditions in the parameter space, and to derive the corresponding LMI formulation when possible. See [5] for recent results on detecting convexity of polynomial matrix functions.

D. Strict feasibility and BMI optimization

In order to solve the strict BMI feasibility problem (3), we can solve the non-strict BMI optimization problem

$$\begin{array}{ll} \max & \lambda \\ \text{s.t.} & H(k) \succeq \lambda I. \end{array}$$

If $\lambda > 0$ in the above problem then k is a strictly feasible point for BMI (3).

In practice however the feasibility set of BMI (3) can be unbounded in some directions, see e.g. Figure 1, and hence λ can grow unreasonably large. Practice then reveals that the BMI optimization problem

$$\begin{array}{ll} \max & \lambda - \mu\|k\| \\ \text{s.t.} & H(k) \succeq \lambda I \end{array}$$

is more appropriate, where $\mu > 0$ is a parameter and $\|.\|$ is any suitable norm. Parameter μ allows to trade off between feasibility of the BMI and a moderate norm of SOF matrix K, which is generally desirable in practice, to avoid large feedback signals.

Returning to the numerical example of section IV-B, we see that strict feasibility of the BMI is essential, otherwise the point $k_1 = k_2 = 0$ is a trivial solution of the non-strict BMI. Note that this point does not even belong to the boundary of the feasible set !

Notice also that maximizing λ under the BMI constraint

$$\begin{bmatrix} -13k_2 - 5k_1k_2 + k_2^2 & k_2 & 0 \\ k_2 & k_1 & 0 \\ 0 & 0 & -13k_1 - k_2 - 5k_1^2 + k_1k_2 \end{bmatrix} \succeq \lambda I$$

is actually an unbounded problem. Indeed, with the choice $k_2 = 13k_1 + 10$ the Hermite matrix $H(k)$ becomes a monovariate polynomial matrix

$$H(k_1) = \begin{bmatrix} 65k_1 + 50k_1^2 & 13 + 10k_1 & 0 \\ 13 + 10k_1 & k_1 & 0 \\ 0 & 0 & -13 - 10k_1 + 5k_1^2 \end{bmatrix}$$

whose zeros (the roots of its determinant) are -1.3000, -0.8974 and 2.8974. For $k_1 > 2.8974$ matrix $H(k_1)$ is positive definite and its minimum eigenvalue can be made arbitrary large by choosing k_1 large enough.

On the other hand, a strictly feasible point with small Euclidean norm is $k_1 = 2.8845$, $k_2 = 41.9791$, but it lies very near the stability boundary. The resulting SOF controller is extremely fragile and a tiny perturbation on the open-loop system matrices A, B, C or on the SOF gain matrix K itself destabilizes the closed-loop system.

It is recommended to introduce parameter μ so as to trade off between these two extreme cases.

As an alternative option, one can introduce additional redundant constraints, such as sufficiently loose lower and upper bounds on the individual entries k_i (large SOF coefficients are not recommended for physical implementation reasons), or simple linear cuts derived from necessary stability conditions (e.g. all coefficients strictly positive for continuous-time stability, which excludes the origin for the above example). This option is not pursued here however.

V. NUMERICAL EXAMPLES

The numerical examples are processed with YALMIP 3.0 [13] and Matlab 7 running on a SunBlade 100 Unix workstation. We use version 2.0 of PENBMI [10] to solve the BMI problems. We set the PENBMI penalty parameter P0 by default to 0.001 (note that this is not the default YALMIP 3.0 setting). The SOF problems are extracted from Leibfritz's database [11]. Numerical values are provided with 5 significant digits.

A. Default tunings

Two tuning parameters which are central to a good performance of PENBMI are initial point K_0 (since it is a local solver) and weighting parameter μ (tradeoff between BMI feasibility and SOF feedback norm). For many BMI SOF problems we observe a good solver behavior with the default tuning $K_0 = 0$ and $\mu = 1$.

For example, problem AC7 ($n = 9$, $mp = 2$) is solved after 36 iterations (CPU time = 1.13s). The resulting SOF is

$$K = \begin{bmatrix} 1.3340 & -0.44245 \end{bmatrix}$$

and $\lambda = 104.29$.

Problem AC8 ($n = 9$, $mp = 5$) is solved after 10 iterations (CPU time = 0.23s). The resulting SOF is

$$K = \begin{bmatrix} 0.38150 & -0.051265 & -0.47018 & -0.0083111 & -0.45831 \end{bmatrix}$$

and $\lambda = 141.78$.

Problem REA3 ($n = 12$, $mp = 3$) is solved after 8 iterations (CPU time = 0.02s). The resulting SOF

$$K = \begin{bmatrix} 1.3304 \cdot 10^{-15} & -6.9350 \cdot 10^{-8} & -2.1498 \cdot 10^{-6} \end{bmatrix}$$

has very little norm, indicating that PENBMI found a feasible point very close to the initial point. Note that this feedback matrix may be sensitive to modeling or round-off error. We return to this example in the next section.

B. Tradeoff between feasibility and SOF matrix norm

For default tunings on problem PAS ($n = 5$, $mp = 3$) PENBMI after 9 iterations (CPU time = 0.03s) returns an almost zero feedback matrix K with $\lambda = -1.5429 \cdot 10^{-11}$ slightly negative. As a result, one closed-loop pole of matrix $A + BKC$ is located at $3.7999 \cdot 10^{-10}$, in the right half-plane. This is a typical behavior when PENBMI is not able to find a strictly stabilizing feedback.

In order to ensure positivity of λ and hence strict feasibility of the BMI, we choose $\mu = 10^{-4}$ as a new weighting parameter. After 17 iterations (CPU time = 0.08s), PENBMI returns the SOF

$$K = \begin{bmatrix} -2.5775 \cdot 10^{-4} & -58.350 & -37.751 \end{bmatrix}$$

and $\lambda = 146.58$, ensuring closed-loop stability.

Returning to example `REA3`, the setting $\mu = 10^{-4}$ results after 28 iterations (CPU time = 0.76s) in an SOF

$$K = \begin{bmatrix} -1.3964 \cdot 10^{-6} & -3684.6 & -6893.1 \end{bmatrix}$$

of significant norm when compared with the result of the previous section obtained for $\mu = 1$. It is likely that this feedback is more robust, although checking this would require appropriate tools which are out of the scope of this paper.

C. Singularity of the Hermite matrix

Consider problem `NN17` ($n = 3$, $mp = 2$). With default tuning $\mu = 1$, PENBMI after 19 iterations (CPU time = 0.09s) returns the feedback

$$K = \begin{bmatrix} -0.26682 \\ 0.14816 \end{bmatrix}$$

and $\lambda = -3.0558 \cdot 10^{-12}$. Note that contrary to the above example `PAS`, the norm of K is not close to zero. However, since λ is slightly negative, one can expect that stability is not achieved. Indeed, closed-loop poles are located at -2.2668 (stable), -1.0817 (stable) and 1.0817 (unstable). Inspection of the resulting 3×3 Hermite matrix $H(k)$ reveals eigenvalues at 10.7402, $1.470 \cdot 10^{-11}$ and $-2.980 \cdot 10^{-11}$. In words, the Hermite matrix is almost singular, and not positive definite. Singularity of the Hermite matrix is related with location of a root along the stability boundary, but also with symmetry of the spectrum with respect to this boundary (here the imaginary axis), see [12] for more details. This is also a typical behavior of PENBMI when failing to find a strictly stabilizing point.

With the choice $\mu = 10^{-3}$, PENBMI solves this problem in 16 iterations (CPU time = 0.12s) and returns the stabilizing feedback

$$K = \begin{bmatrix} -99.363 \\ 3000.2 \end{bmatrix}.$$

D. Choice of initial point

Since PENBMI is a local optimization solver, it can be sensitive to the choice of the initial point. Since we have generally no guess on the approximate location of a feasible point for BMI (3), in most of the numerical examples we choose the origin as the initial point. However, this is not always an appropriate choice, as illustrated below.

Consider problem `NN1` ($n = 3$, $mp = 2$) already studied in section IV-B. With $\mu = 10^{-4}$ and the initial condition $K_0 = [0\,0]$, PENBMI after 46 iterations (CPU time = 0.52s) returns a slightly negative value of λ and a feedback K which is not stabilizing. With initial condition e.g. $K_0 = [0\,30]$, PENBMI after 10 iterations (CPU time = 0.04s) converges to a stabilizing SOF

$$K = [343.60\ 1828.5].$$

Note that as shown in section IV-C in this case the feasibility region is convex. Despite convexity on the underlying optimization problem, its BMI formulation however renders PENBMI sensitive to the initial condition.
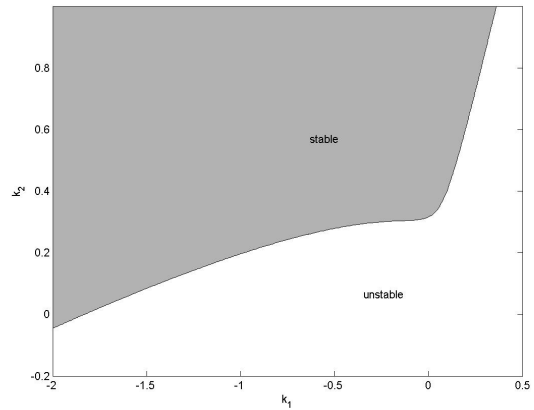

Fig. 2.   SOF example `HE1`. Region of stabilizing gains (in gray).

Consider now problem `HE1` ($n = 4$, $mp = 2$). Default tunings do not allow PENBMI to find out a stabilizing SOF, the resulting Hermite matrix being singular. However, with the choice $K_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\star$ which is not a stabilizing SOF, PENBMI converges after 10 iterations (CPU time = 0.03s) to the stabilizing SOF

$$K = \begin{bmatrix} -3.3208 \\ 7.6991 \end{bmatrix}$$

for which $\lambda = 138.91$. Note that for the choice $K_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\star$, a stabilizing SOF, PENBMI converges to the same stabilizing SOF as above. The non-convex feasible set for this example is represented on Figure 2.

It would be welcome to characterize the basin of attraction for which PENBMI converges to a feasible point. The polynomial formulation allows to carry out 2D graphical experiments in the case $mp = 2$ since the optimization is over the SOF coefficients only.

E. Ill-conditioning and scaling

Finally, for some of the SOF problems we faced numerical problems that are certainly related with ill-conditioning or at least bad data scaling.

Problems `NN6` and `NN7`, both with $n = 9$ and $mp = 4$, produce a Hermite matrix which is ill-conditioned around the origin (ratio of maximum and minimum singular values around 10^{-17}). Note that this has nothing to do with the way characteristic polynomial coefficients are computed, since the truncated Vandermonde matrix is unitary, see section IV-A. We suspect that ill-conditioning here is related with the choice of the monomial basis 1, s, $s^2 \ldots$ used to represent the characteristic polynomial. See the conclusion for more comments on this particular point.

Ill-conditioning and bad data scaling are however not only related with the polynomial formulation. For problem `PAS` ($n = 5$, $mp = 3$) the B matrix has Euclidean norm $1.5548 \cdot 10^{-2}$. PENBMI fails to find a stabilizing SOF for this problem. However, by solving the SOF problem of the scaled

triplet $(A, 1000B, 1000C)$ and with $\mu = 10^{-8}$, PENBMI after 7 iterations (CPU time = 0.05s) returns the SOF

$$K = [-0.16422 \ -6266.9 \ -0.38369]$$

stabilizing the original triplet (A, B, C). The development of appropriate data scaling, or pre-conditioning policies for SOF problems (in state-space or polynomial setting) is also an interesting subject of research.

VI. CONCLUSION

In this paper SOF problems from the COMPl$_e$ib library were formulated in a polynomial setting and solved with the publicly available PENBMI solver. The user controls two basic tuning parameters: the initial feedback estimate K_0 and the weighting parameter μ trading off between feasibility of the BMI and norm of the feedback. Generally, default tunings ($K_0 = 0$, $\mu = 1$) suffice, but for some problems it may be necessary to decrease μ and/or to try different initial conditions K_0. Other aspects specific to the local convergence nature of the optimization algorithm were touched on via numerical examples.

Characteristic polynomial $q(s, k)$ is expressed in (1) in the standard multivariate monomial basis with indeterminates $s, k_1, \ldots, k_{mp}$. While this basis is most convenient for notational purposes, it is well-known that its numerical conditioning is not optimal. Orthogonal polynomial bases such as Chebyshev or Legendre polynomials are certainly more appropriate in the interpolation scheme of Section IV-A. The use of alternative polynomial bases in this context is, in our opinion, an interesting subject of further research.

As it is described in this paper, the approach is restricted to SIMO or MISO SOF problems, otherwise the Hermite stability criterion results in a polynomial matrix inequality (PMI) in the feedback matrix entries. In order to deal with MIMO SOF problems, one can try to obtain a matrix polynomial version of the Hermite criterion, see [1] for an early and only partially successful attempt. By introducing lifting variables, PMI can be rewritten as BMI problems with explicit additional equality constraints that should be handled by next versions of PENBMI. Another way out could be to extend PENBMI to cope with general PMI problems, which can be done without major theoretical or technical restriction. The PMI formulation is already fully covered in version 3.0 of the YALMIP software.

REFERENCES

[1] B. D. O. Anderson, R. R. Bitmead. Stability of matrix polynomials. Int. J. Control, 26(2):235–247, 1977.

[2] A. Ben-Tal, A. Nemirovskii. Lectures on modern convex optimization: analysis, algorithms and engineering applications. SIAM, Philadelphia, PA, 2001.

[3] A. Ben-Tal, M. Zibulevsky. Penalty-barrier multiplier methods for convex programming problems. SIAM J. Optim. 7:347–366, 1997.

[4] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan. Linear matrix inequalities in system and control theory. SIAM, Philadelphia, 1994.

[5] J. F. Camino, J. W. Helton, R. E. Skelton, J. Ye. Matrix inequalities: a symbolic procedure to determine convexity automatically. Int. Equ. Oper. Theory, 46(4):399-454, 2003.

[6] K. C. Goh, M. G. Safonov, J. H. Ly. Robust synthesis via bilinear matrix inequality. Int. J. Robust and Nonlinear Control, 6(9/10):1079–1095, 1996.

[7] D. Henrion, S. Tarbouriech, M. Šebek. Rank-one LMI approach to simultaneous stabilization of linear systems. Syst. Control Letters, 38(2):79–89, 1999.

[8] D. Henrion, D. Peaucelle, D. Arzelier, M. Šebek. Ellipsoidal approximation of the stability domain of a polynomial. IEEE Trans. Autom. Control, 48(12):2255–2259, 2003.

[9] M. Kočvara, M. Stingl. PENNON – a code for convex nonlinear and semidefinite programming. Opt. Methods and Software, 18(3):317–333, 2003.

[10] M. Kočvara, M. Stingl. PENBMI, Version 2.0, 2004. See www.penopt.com for a free developer version.

[11] F. Leibfritz. COMPl$_e$ib:: constrained matrix optimization problem library – a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Research report, Department of Mathematics, University of Trier, Germany, 2003. See www.mathematik.uni-trier.de/~leibfritz.

[12] H. Lev-Ari, Y. Bistritz, T. Kailath. Generalized Bezoutians and families of efficient zero-location procedures. IEEE Trans. Circ. Syst. 38(2)170–186, 1991.

[13] J. Löfberg. YALMIP: a toolbox for modeling and optimization in Matlab. Proc. IEEE Symposium on Computer-Aided Control System Design, Taipei, Taiwan, 2004. See control.ee.ethz.ch/~joloef.