Mixed LMI/Randomized Methods for Static Output Feedback Control Design*

Denis Arzelier†, Elena N. Gryazina‡, Dimitri Peaucelle† and Boris T. Polyak‡

September 24, 2009

Keywords: Randomized algorithms, H_2 and H_∞ Optimal Control, Static Output Feedback Stabilization

Abstract

This paper addresses the problem of stabilization of LTI systems via static output feedback (sof) and optimal H_2 and H_∞ sof control. Various algorithms based on the same mixed LMI/randomized approach are defined for the computation of sets of stabilizing sof and optimal H_2 and H_∞ sof control. The main idea is to combine a particular relaxed LMI parameterization of stabilizing sof with high efficiency of Hit-and-Run method for generating random points in a given domain. Their respective efficiency is compared on several examples of the COMPl$_e$ib library. Dual of the approach proposed in [35], the mixed LMI/randomized algorithms proposed here appear to be much more effective. Indeed, obtained results are almost comparable to those obtained by HIFOO package which is considered to be the most effective tool for optimal H_∞ sof control. Finally, the paper additionally provides an extensive evaluation of the different instances proposed in the COMPl$_e$ib library in terms of stabilization and optimal H_2/H_∞ sof control as well.

1 Introduction

One of the most challenging open problems in control theory is the synthesis of static output-feedback (SOF) controllers that meet desired performances and/or robustness specifications [38]. The static or reduced fixed-order dynamic output feedback problem is therefore always an active research area in the control literature.

In the recent years, many attempts have been made to give efficient numerical procedures to solve related problems, [1, 36, 25, 14, 12, 15, 24, 16]. In [11], a numerical comparison was performed and classification into three categories (nonlinear programming, parametric optimization and convex programming approaches) was proposed. Even if these three classes may overlap, it gives a clear picture of the situation at that moment. Since then, new developments have been witnessed and the literature has been enriched by numerous contributions on the so-called nonlinear programming approach [26, 5, 10, 9, 3, 2] while pure convex programming methods or parametric optimization methods (which could be merged in a unique class) were scarcely still considered [31, 35]. The main reason mainly relies on the use of new powerful numerical tools based on nonsmooth optimization for the solution of static output feedback stabilization problems or static output feedback design with closed-loop performance guarantees [10, 19, 20]. The algorithms based on these techniques may be considered as the most numerically efficient ones at the moment as reported in different dedicated publications [4, 21, 18].

The first objective of this paper is not to contest the present superiority of existing packages but rather to show that there is still room for alternative useful options on the specific issues mentioned above. In [32], a first step for designing sets of stabilizing static output feedbacks was proposed which is dual of the one proposed later in [35] and [34]. The effective construction of samples in these ellipsoidal sets relied on usual deterministic local optimization algorithm. Recently, new randomized algorithms have been designed for the synthesis of feedback controllers in different contexts [17], [33], [13]. Very efficient from a numerical point of view, an initial guess is generally needed to adequately build the set of admissible controllers. The idea is therefore to combine these two approaches into a mixed LMI/randomized algorithm to tackle the problem of static output feedback stabilization.

If stabilization is a mandatory requirement when designing control systems, it is not sufficient for many applications. It is therefore essential to provide the designer with adequate methods for the computation of feedback

*This work was supported by CNRS and RFBR via PICS No. 4281.

†D. Arzelier and D. Peaucelle are with CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France. They are also with Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France. `arzelier,peaucelle@laas.fr`

‡E.N. Gryazina and B.T. Polyak are with the Institute of Control Science, Russian Academy of Sciences, Moscow, Russia, `fboris,gryazinag@ ipu.ru`

laws with guaranteed closed-loop performance. H_2 and H_∞ norms are now widely used in the control community as complementary effective measures of optimal average performance for the former and robustness for the latter The problems of H_2 and H_∞ optimal static output feedback stabilization are therefore addressed and mixed LMI/randomized algorithms based on a coordinate-descent cross-decomposition scheme allows to get suboptimal static output feedback gains for these two problems.

In the final section, different variations of the basic mixed LMI/randomized algorithm are analyzed on a large part of the benchmarks of the COMPl$_e$ib library [22], [28], [29].

1.1 Notations

For conciseness reasons, some abbreviations are used. $sym(A) = A + A'$. $[\star]'BA = A'BA$ and

$$\begin{bmatrix} A & B \\ \star & C \end{bmatrix} = \begin{bmatrix} A & B \\ B' & C \end{bmatrix} \tag{1}$$

$\mathbb{C}$ is the field of real numbers and $\mathbb{C}$ is the field of complex numbers while $\mathbb{C}^-$ is the open subset of complex numbers with strictly negative real part.

For symmetric matrices, $\succ$ $(\succeq)$ is the Löwner partial order ($A \succ (\succeq) B$ if and only if $A - B$ is positive (semi) definite) defined in the cone of positive (semi)definite matrices $\mathbb{S}^{+*}$ ($\mathbb{S}^{+}$). $\mathrm{trace}(A)$ is the sum of diagonal elements of a (square) matrix. $\Lambda(A)$ is the spectrum of the matrix A (the set of all eigenvalues of the matrix A). $\mathbb{R}^{m\times n}$ is the linear space of rectangular $m \times n$ matrices with real entries and equipped with the inner product defined as the usual inner product of the vectors representing the matrices $< A, B >= \mathrm{trace}(A'B)$.

1.2 Statement of stabilization and performance problems

Let the state-space model of the system be given by its minimal state-space realization:

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + B_1 w(t) + Bu(t) \\ z(t) & = & C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) & = & Cx(t) + D_{21} w(t) \end{array} \qquad P(s) := \left[\begin{array}{c|cc} A & B_1 & B \\ \hline C_1 & D_{11} & D_{12} \\ C & D_{21} & \mathbf{0} \end{array}\right] \tag{2}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control vector, $y \in \mathbb{R}^r$ is the output vector, $w \in \mathbb{R}^{m_2}$ is the disturbance vector and $z \in \mathbb{R}^{r_2}$ is the controlled output vector. All matrices are assumed to be of appropriate dimensions and it is assumed throughout the paper that $\mathrm{rank}(B) = m$ and $\mathrm{rank}(C) = r$.

The model (2) is stabilizable by static output-feedback (sof) if there exists a gain matrix $K_{sof} \in \mathbb{R}^{m\times r}$ such that the closed-loop matrix $A + BK_{sof}C$ is asymptotically stable (a Hurwitz matrix). Moreover, the system is stabilizable by state-feedback (sf) if there exists a gain matrix $K \in \mathbb{R}^{n\times r}$ such that the matrix $A + BK_{sf}$ is Hurwitz. This last property is a special case of the former and corresponds to full state information output-feedback ($C = \mathbf{1}$) and has found a tractable solution through convex optimization and LMI formalism [8]. Let us define the set of stabilizing state-feedback matrices:

$$\mathcal{K}_{sf} = \left\{ K_{sf} \in \mathbb{R}^{m\times n} \ : \ \Lambda(A + BK_{sf}) \in \mathbb{C}^- \right\} \tag{3}$$

and the set of stabilizing static output-feedback matrices:

$$\mathcal{K}_{sof} = \left\{ K_{sof} \in \mathbb{R}^{m\times r} \ : \ \Lambda(A + BK_{sof}C) \in \mathbb{C}^- \right\} \tag{4}$$

The first problem considered in this paper is to build non trivial sets of stabilizing sof for (2).

Problem 1 *sof stabilization*
Given the model (2), build a non trivial subset $\mathcal{K}_{sof}^{n_{sof}} \subset \mathcal{K}_{sof}$ *of* $n_{sof} \geq 1$ *instances.*

In general, when evaluating algorithms for sof design, one important issue is not only to find stabilizing sof but rather to find adequate candidates for optimizing some prespecified performance. For $K_{sof} \in \mathcal{K}_{of}$ and $K_{sf} \in \mathcal{K}_{sf}$, consider the following transfer functions:

$$\begin{aligned} T(s, K_{sof}) &= (C_2 + D_2 K_{sof} C)(s\mathbf{1} - (A + BK_{sof}C))^{-1} B_2 \\ T_s(s, K_{sf}) &= (C_2 + D_2 K_{sf})(s\mathbf{1} - (A + BK_{sf}))^{-1} B_2 \end{aligned} \tag{5}$$

Let $\|T(s, K_*)\|_2$ and $\|T(s, K_*)\|_\infty$ be respectively the H_2 norm and the H_∞ norm of the transfer matrix $T(s, K_*)$. The generic performance synthesis problem to be addressed is stated as follows.

Problem 2 *optimal sof control*

Find $K^*_{sof} \in \mathcal{K}_{sof}$ such that $\|T(s, K_{sof})\|_*$ where $* = 2$ or ∞ is minimum.

$$K^*_{sof} = \arg \left\{ \min_{K_{sof} \in \mathcal{K}_{sof}} \|T(s, K_{sof})\|_* \right\} \tag{6}$$

This problem which has been dealt with for the first time in [30] in the context of H_2 optimal sof control, is known to be a difficult one for which no convex formulation exists yet for $* = 2$ or ∞. The problem is supposed to have many local minima and it is believed that any reformulation will also exhibit local minima. To overcome this particular problem, many approaches use heuristics to make the optimization problem more tractable. In [31], an efficient numerical cross-decomposition procedure based on a new parameterization of $\mathcal{K}_{sof}$ has given promising results in the H_2 case. It is proposed to build on these previous results by adding a randomized step that will enforced the obtained results. Note that in [31], the H_2 optimal sof control problem was addressed but $\mathcal{H}_\infty$ optimal sof control may be considered in the same framework. In the sequel, when considering the H_2 sof optimal control problem, it is assumed that $D_{11} = \mathbf{0}$ and $D_{21} = \mathbf{0}$.

2 Construction of stabilizing sof sets via a mixed LMI/randomized approach

2.1 Parameterization of $\mathcal{K}_{sof}$

Two different mixed LMI/randomized algorithms may be defined. They both rely on the following parameterization of stabilizing sof matrices that has been first proposed in [31]. Let us define the following notation:

$$M(P) = \begin{bmatrix} A'P + PA & PB \\ B'P & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ A & B \end{bmatrix}' \begin{bmatrix} \mathbf{0} & \mathbf{0} & P \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ P & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ A & B \end{bmatrix} \tag{7}$$

A necessary and sufficient condition for the existence of a stabilizing sof for (2) is given in the following theorem.

Theorem 1 *[31]*

$\exists\ K_{sof} \in \mathcal{K}_{sof}$ for the model (2) if and only if there exist a stabilizing sf matrix $K_{sf} \in \mathcal{K}_{sf}$, a matrix $P \in \mathbb{S}^{+*}$ and matrices $F \in \mathbb{R}^{m \times m}$, $Z \in \mathbb{R}^{m \times r}$ solutions of the matrix inequality (8):

$$L(P, K_{sf}, Z, F) = M(P) + \mathrm{sym}\left(\begin{bmatrix} K'_{sf} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} ZC & F \end{bmatrix} \right) \prec \mathbf{0} \tag{8}$$

Moreover, $K_{sof} = -F^{-1}Z \in \mathcal{K}_{sof}$.

Proof

Note that the sof stabilizability of (2) ($\exists\ K_{sof} \in \mathcal{K}_{sof}$) is equivalent to the existence of a matrix $K_{sof} \in \mathbb{R}^{m \times r}$ and a matrix $P \in \mathbb{S}^{+*}$ solutions to the following Lyapunov inequality:

$$(A + BK_{sof}C)'P + P(A + BK_{sof}C) = \begin{bmatrix} \mathbf{1} & C'K'_{sof} \end{bmatrix} M(P) \begin{bmatrix} \mathbf{1} \\ K_{sof}C \end{bmatrix} \prec \mathbf{0} \tag{9}$$

Applying elimination lemma [37], this is also equivalent to the existence of matrices $K_{sof} \in \mathbb{R}^{m \times r}$, $F_1 \in \mathbb{R}^{r \times m}$, $F_2 \in \mathbb{R}^{m \times m}$ and a matrix $P \in \mathbb{S}^{+*}$ solutions to the following matrix inequality:

$$M(P) + \mathrm{sym}\left(\begin{bmatrix} C'K'_{sof} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} F_1 & F_2 \end{bmatrix} \right) \prec \mathbf{0} \tag{10}$$

The matrix F_2 is always invertible since the block $(2, 2)$ reads $F_2 + F_2' \succ \mathbf{0}$. Factorizing F_2 in 10 leads to the result. ■

Even if the existence condition is expressed in terms of solution to a Bilinear Matrix Inequality (BMI), this parameterization has some interesting characteristics that will be useful for defining efficient numerical procedure for sof synthesis. First, thanks to the introduction of additional variables, there is a decoupling between the computation of $K_{sof} \in \mathcal{K}_{sof}$ and the Lyapunov certificate P. Moreover, the matrix K_{sf} at the origin of the non convexity of condition (8) must not be arbitrarily chosen but it has to be a stabilizing sf for (2), i.e. $K_{sf} \in \mathcal{K}_{sf}$. It is well-known that the computation of a stabilizing sf for (2) is a convex problem that may solved via the following LMI optimization problem [8]:

$$\begin{array}{ll} \min\limits_{X,R} & \text{trace}(X) \\ \text{under} & \text{trace(X)} > \alpha \\ & X \succ \mathbf{0} \\ & AX + XA' + BR + R'B' \prec \mathbf{0} \end{array} \tag{11}$$

for some $\alpha > 0$ and where $K_{sf} = RX^{-1} \in \mathcal{K}_{sf}$.

For a given $K_{sf} \in \mathcal{K}_{sf}$, the LMI convex set $\mathcal{L}_{sof}^{K_{sf}}$ defined by:

$$\mathcal{L}_{sof}^{K_{sf}} = \left\{(P, Z, F) \in \mathbb{S}^{+*} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{m\times m} \ : \ \exists\, K_{sf} \in \mathcal{K}_{sf} \mid L(P, K_{sf}, Z, F) \prec \mathbf{0}\right\} \tag{12}$$

is a convex parameterization that approximates the set of all stabilizing sof $\mathcal{K}_{sof}$. Note that this convex approximation may be empty ($\mathcal{L}_{sof}^{K_{sf}} = \emptyset$) even if $\mathcal{K}_{sof} \neq \emptyset$ for a given $K_{sf} \in \mathcal{K}_{sf}$. However, a complete parameterization of $\mathcal{K}_{sof}$ is obtained when K_{sf} covers the whole continuum of $\mathcal{K}_{sf}$.

$$\mathcal{K}_{sof} = \left\{ \bigcup_{i}^{\infty} \mathcal{L}_{sof}^{K_{sf}^{i}} \mid K_{sf}^{i} \in \mathcal{K}_{sf} \right\} \tag{13}$$

In fact, the problem of finding a stabilizing sof amounts to find a triplet composed of $(K_{sof}, K_{sf}, P) \in \mathcal{K}_{sof} \times \mathcal{K}_{sf} \times \mathbb{S}^{+*}$ verifying (8), meaning that a common Lyapunov certificate P has to be found for K_{sf} and K_{sof}. This last interpretation is reminiscent of a necessary and sufficient condition of sof stabilizability proposed in [7].

Remark 1 *:*

Note that the parameterization (8) of $\mathcal{K}_{sof}$ has an equivalent one for $\mathcal{K}_{sf}$ since:

$$L(P, K_{sf}, Z, F) = L(P, K_{sof}, H, F) = M(P) + \text{sym}\left(\begin{bmatrix} C'K_{sof}' \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} H' & F' \end{bmatrix}\right) \prec \mathbf{0} \tag{14}$$

where $K_{sf} = -F^{-T}H' \in \mathcal{K}_{sf}$. Given $K_{sof} \in \mathcal{K}_{sof}$, one can easily get a convex parameterization approximating the set of all stabilizing sf. This seems of no consequence since the complete set $\mathcal{K}_{sf}$ may be parameterized by the LMI convex set defined in (11). Nevertheless, this alternate parameterization will appear useful in the sequel when looking for sets of sof of for optimal H_2 or H_∞ sof.

For a given $K_{sof} \in \mathcal{K}_{sof}$, the LMI convex set $\mathcal{L}_{sf}^{K_{sof}}$ defined by:

$$\mathcal{L}_{sf}^{K_{sof}} = \left\{(P, H, F) \in \mathbb{S}^{+*} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times m} \ : \ \exists\, K_{sof} \in \mathcal{K}_{sof} \mid L(P, K_{sof}, H, F) \prec \mathbf{0}\right\} \tag{15}$$

is a convex parameterization that approximates the set of all stabilizing sf $\mathcal{K}_{sf}$.

2.2 Hit-and-Run design techniques for stabilization

Hit-and-Run for the stability set for matrices.

We apply HR for generating the sets of stabilizing state-feedback matrices K_{sf} (3) and static output-feedback matrices K_{sof} (4). The approach is the same for both cases. We describe it for static output-feedback matrices (for state-feedback matrices take $C = I$, $K_{sf} \in \mathbb{R}^{m\times n}$). Suppose matrices A, B, C are given and $K \in \mathbb{R}^{m\times r}$ is a variable, K^0 belongs to the bounded set of stabilizing gains:

$$\mathcal{K}_{sof} = \{K : A + BKC \text{ is Hurwitz.}\} \tag{16}$$

The structure of this set is analyzed in [17]. It can be nonconvex and can consist of many disjoint domains.

In every step of the HR algorithm we generate matrix $D = Y/||Y||, Y = \texttt{randn(m,r)}$ which is uniformly distributed on the unit sphere in the space of matrices equipped with Frobenius norm. Matrix D is a random direction in the space of $m \times r$-matrices. We call *boundary oracle* an algorithm which provides $L = \{t \in R : K^0 + tD \in \mathcal{K}_{sof}\}$. We denote

$$A + B(K^0 + tD)C = F + tG,$$
$$F = A + BK^0C, \quad G = BDC$$

for a matrix $K^0 \in \mathcal{K}_{sof}$, then $L = \{t : F + tG \text{ is Hurwitz}\}$. In the simplest case when $\mathcal{K}_{sof}$ is convex, this set is the interval $(-\underline{t}, \bar{t})$ where $\bar{t} = \sup\{t : K^0 + tD \in \mathcal{K}_{sof}\}, \underline{t} = \sup\{t : K^0 - tD \in \mathcal{K}_{sof}\}$. In more general situations boundary oracle provides all intersections of the straight line $K^0 + tD, -\infty < t < +\infty$ with $\mathcal{K}_{sof}$. L consists of finite number of intervals, the algorithm for calculating their end points is presented in [17], Section 4. However sometimes "brute force" approach is more simple. Introduce $f(t) = \max \Re \operatorname{eig}(F + tG)$, then the end points of the intervals are solutions of the equation $f(t) = 0$ and can be found by use of standard 1D equation solvers (such as command `fsolve` in Matlab).

HR method works as follows.

1. Find a starting point $K^0 \in \mathcal{K}_{sof}$; $i = 0$.
2. At the point $K^i \in \mathcal{K}_{sof}$ generate a random direction $D^i \in \mathbb{R}^{m \times r}$ uniformly distributed on the unit sphere.
3. Apply boundary oracle procedure, i.e., define the set
$$L_i = \{t \in \mathbb{R} : K^i + tD^i \in \mathcal{K}_{sof}\}.$$
4. Generate a point t_i uniformly distributed in L_i (we recall that L_i is, in general, a finite set of intervals), and compute a new point
$$K^{i+1} = K^i + t_i D^i.$$
5. Go to step 2 and increase i.

The simplest theoretical result on the behavior of HR method states that if $\mathcal{K}$ does not contain lower dimensional parts, then the method achieves the neighborhood of any point of $\mathcal{K}$ with nonzero probability and asymptotically the distribution of points k_i tends to uniform one. The rage of convergence strongly depends on geometry of $\mathcal{K}$ and its dimension.

2.3 Two LMI/randomized algorithms for sof stabilization

Two algorithms using complementary advantages of degrees of freedom offered by parameterization (8) and Hit-and-Run numerical efficiency are built in order to generate non trivial sets of stabilizing sof. The first algorithm uses Hit-and-Run only for generating a large subset $\mathcal{K}_{sf}^{n_{sf}}$ of $\mathcal{K}_{sf}$ that will serve for checking if $\mathcal{L}_{sof}^{K_{sf}}$ is empty.

Algorithm 1 *:*

1- Compute a stabilizing sf $K_{sf}^0 = RX^{-1} \in \mathcal{K}_{sf}$ via the solution of the LMI problem (11):

2- From K_{sf}^0, generate a set $\mathcal{K}_{sf}^{n_{sf}} \subset \mathcal{K}_{sf}$ which is the collection of n_{sf} samples of stabilizing sf matrices via H.R. ;

3- Compute a set $\mathcal{K}_{sof}^{n_{sof}^1} \subset \mathcal{K}_{sof}$ which is the collection of n_{sof}^1 samples of stabilizing sof matrices:

$\forall\ K_{sf}^i \in \mathcal{K}_{sf}^{n_{sf}}$, if the LMI set $\mathcal{L}_{sof}^{K_{sf}^i} \neq \emptyset$, add the solution $K_i = -F_i^{-1} Z_i$ to the collection set $\mathcal{K}_{sof}^{n_{sof}^1}$.

Remark 2 *:*

(11) is not the only way to initialize algorithm 1. K_{sf}^0 may also be computed as the LMI solution of the H_2 or H_∞ state-feedback problems (see [6] for more details).

Step 3 is performed by testing the realizability of the LMI set $\mathcal{L}_{sof}^{K_{sf}}$ for each instance $K_{sf} \in \mathcal{K}_{sf}^{n_{sf}}$ implying the solution of the homogeneous LMIs (10) with respect to the decision variables (P, Z, F). To avoid numerical problems, this step must be done by solving the following semidefinite programming problem:

$$\begin{array}{ll} \min\limits_{P,Z,F} & \text{trace}(P) \\ \text{under} & \text{trace(P)} > \alpha \\ & P \succ \mathbf{0} \\ & M(P) + \text{sym}\left(\begin{bmatrix} K'_{sf} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} ZC & F \end{bmatrix} \right) \prec \mathbf{0} \end{array} \tag{17}$$

for some $\alpha > 0$.

The interest of running this algorithm is twofold. First, it will be a good way to evaluate the conservatism of the convex approximation of $\mathcal{K}_{sof}$ induced by parameterization (8). Indeed, the percentage n^1_{sof}/n_{sf} may be defined as a quantitative measure of both the conservatism of the convex approximation as well as the difficulty to stabilize the plant via sof. Secondly, this algorithm will be used for H_2 optimal synthesis via sof in two different ways as will be explained in section 3.

Remark 3 *:*
If K is a stabilizing sof a priori *known, then the set $\mathcal{L}^{KC}_{sof} \neq \emptyset$ since the choice $Z = -FK$ will lead to the existence of $P \in \mathbb{S}^{+*}$ and $F \in \mathbb{R}^{m\times m}$ such that:*

$$M(P) + \mathrm{sym}\left(\begin{bmatrix} C'K' \\ -\mathbf{1} \end{bmatrix}\begin{bmatrix} -FKC & F \end{bmatrix}\right) = M(P) + \mathrm{sym}\left(\begin{bmatrix} C'K' \\ -\mathbf{1} \end{bmatrix}[-F]\begin{bmatrix} KC & -\mathbf{1} \end{bmatrix}\right) \prec \mathbf{0} \quad (18)$$

Applying elimination lemma, this last condition is equivalent to the existence of a matrix $P \in \mathbb{S}^{+*}$ such that:

$$(A + BKC)'P + P(A + BKC) \prec \mathbf{0} \quad (19)$$

which is obviously true since $K \in \mathcal{K}_{sof}$

A much more numerically efficient mixed LMI/randomized algorithm generating sets of stabilizing sof may be deduced from the previous one. It mainly avoids to run the computational burden of checking if LMI set $\mathcal{L}^{K^i_{sf}}_{sof}$ is empty for every instance $\forall\ K^i_{sf} \in \mathcal{K}^{n_{sf}}_{sf}$ but starts a hit and run generation of $\mathcal{K}^{n^2_{sof}}_{sof}$ as soon as an initial $K_{sof} \in \mathcal{K}_{sof}$ is found by LMI step. It proves to be much more efficient in practice to generate sets of stabilizing sof ion almost every studied cases of the COMPl$_e$ib library.

Algorithm 2 *:*

1- *Compute a stabilizing state-feedback $K^0_{sf} = RX^{-1} \in \mathcal{K}_{sf}$ via the solution of the LMI problem (11) ;*

2- *From K^0_{sf}, generate a set $\mathcal{K}^{n_{sf}}_{sf} \subset \mathcal{K}_{sf}$ which is the collection of n_{sf} samples of stabilizing sf matrices via H.R. ;*

3- *Find an initial stabilizing sof $K^0_{sof} = -F_0^{-1}Z_0$: Check every $K^i_{sf} \in \mathcal{K}^{n_{sf}}_{sf}$ until $\mathcal{L}^{K^i_{sf}}_{sof} \neq \emptyset$;*

3- *From K^0_{sof}, compute a set $\mathcal{K}^{n^2_{sof}}_{sof} \subset \mathcal{K}_{sof}$ which is the collection of n^2_{sof} samples of stabilizing sof matrices via H.R.*

2.4 Comments on results from the COMPl$_e$ib library

The COMPl$_e$ib library is composed of different LTI models (2) ranging from purely academic problems to more realistic industrial examples. The underlying systems that are already open-loop asymptotically stable have not been considered here for obvious reasons. With this last restriction, 53 different models mainly classified in six classes have been tested.

- Aerospace models: Aircraft models (AC), helicopter models (HE), jet engine models (JE)
- Reactor models (REA)
- Decentralized interconnected systems (DIS)
- Academic tests problems (NN)
- Various applications: Wind energy conversion model (WEC), binary distillation towers (BDT) and terrain following models (TF), string of high-speed vehicles (IH), strings (CSE), piezoelectric bimorph actuator (PAS),
- Second order models: A tuned mass damper (TMD), a flexible satellite (FS)
- 2D heat flow models (HF2D)

For precise details concerning each single example and benchmark, the interested reader may read [28] and [27].

- First note that it has been possible to stabilize every model except `AC10`. In the first case, the HIFOO package which may be considered as one of the most effective tool for sof stabilization and for optimal H_∞ sof control is not able either to stabilize it.
- A quick look at the reference [34] clearly shows that algorithms 1 and 2 gives far better results than this last reference. In [34], the proposed algorithm was unable to stabilize plants `AC10`, `NN1`, `NN5`, `NN5`, `NN7`, `NN10` and `NN12`.
- Hard to stabilize examples may be identified as the ones for which less than 5% of the initializing state-feedback succeed in finding a stabilizing sof. `AC9`, `AC13`, `AC14`, `AC18`, `HE3`, `JE3`, `BDT2`, `TF3`, `NN9`, `NN12`, `NN14` are such plants for which n^1_{sof} obtained with algorithm 1 is rather low. This is mainly due to the bad conditioning of numerical operations (matrix inversion) and LMI optimization rather than a failure of the method in itself. This is confirmed by the reference [29] where similar failures of SDP solvers were already noticed for some of the previous examples (`AC14`, `AC18`, `JE3`) for the problems of H_2 or H_∞ state feedback optimal control which are known to have a convex formulation. Moreover, the initialization $K_{sf} = KC$ where $K \in \mathcal{K}_{sof}$ is known *a priori*, does not perform well for all these examples, demonstrating that the plant matrices are poorly conditioned (see remark 3).
- It is not a surprise to note that $n^1_{sof} \leq n^2_{sof}$ since if the initialization step succeeds $n^2_{sof} = 1000$ while $\max(n^1_{sof}) = 1000$. More surprising is the easiness to get n^2_{sof} stabilizing sof for almost all examples and considering also that the complete run is really fast in general. Size seems to be a limiting factor for LMI step but not for Hit-and-Run step.

Figure 1 shows the population stabilizing gains with the boundary for example `AC7`. It is interesting to note that the exact shape of the set of stabilizing sof is easily obtained with the mixed LMI/randomized algorithm 1 (see [22] for comparison). Similar results have been obtained for all two-parameters examples of the database COMPl$_e$ib, labelled `AC4`, `NN1`, `NN5` (see below), `NN17` and `HE1`.

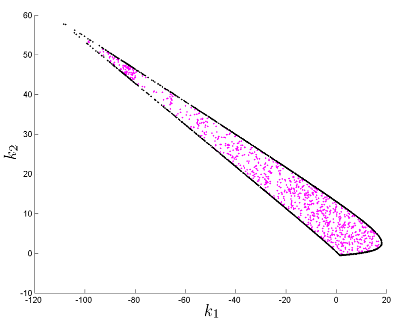

Figure 1: Set of stabilizing sof generated with LMI/Randomized algorithm 2 for AC7 example

Figures 2 and 3 show a comparison of the populations respectively obtained via algorithms 1 and 2 for benchmark `NN5`.

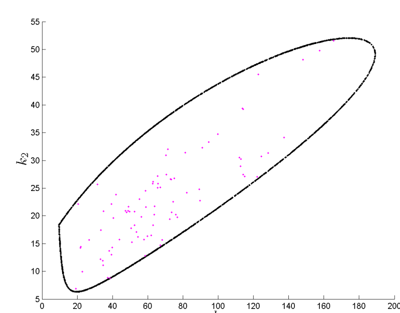

Figure 2: Set of stabilizing sof generated with LMI/Randomized algorithm 1 for NN5 example

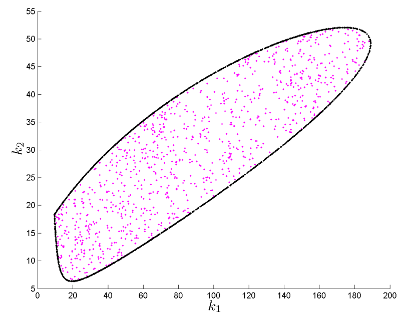

Figure 3: Set of stabilizing sof generated with LMI/Randomized algorithm 2 for NN5 example

3 H_2 optimal synthesis via sof

An obvious and easy way to compute an H_2 suboptimal sof solution for (2) is to pick the best element (with respect to the closed-loop H_2 norm) in the sets $\mathcal{K}_{sof}^{n_{sof}^1}$ and $\mathcal{K}_{sof}^{n_{sof}^2}$ computed by algorithms 1 and 2. This is an alternative to the usual gridding approach for the computation of suboptimal controllers. In general, this approach leads to a rather crude and possibly very conservative approximation of the global solution. Indeed, the genuine H_2 sof optimal control problem (6) is replaced by the following relaxation:

$$K_{sof}^{sub_*} = \arg\left\{ \min_{K_{sof} \in \mathcal{K}_{sof}^{n_{sof}^*}} \quad \|T(s, K_{sof})\|_2 \right\} \tag{20}$$

Corresponding results for COMPl$_e$ib are labelled as H_2 s_1 for algorithm 1 and H_2 s_1 for algorithm 2 in table 1, 2, 3, 4. In the following, different alternative iterative optimization procedures for H_2 optimal sof control are proposed.

3.1 Parameterizations for H_2 optimal control

First, the parameterization of theorem 1 is extended to give a solution to the problem (6). Let us define:

$$N(P_2) = M(P_2) + \begin{bmatrix} C_1 & D_{12} \end{bmatrix}' \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \tag{21}$$

Theorem 2 *:*
The H_2 optimal sof for (2) is given by $K_{sof_2}^* = -F^{-1*}Z^*$ where the triplet $(P_2^*, Z^*, F^*) \in \mathbb{S}^{+*} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{m\times m}$ is the global optimal solution of the non convex optimization problem (22):

$$\begin{array}{ll} \min\limits_{P_2, Z, F, K_{sf_2}} & \mathrm{trace}(B_1' P_2 B_1) \\ \text{under} & P_2 \succ \mathbf{0} \\ & L_2(P_2, K_{sf_2}, Z, F) = N(P_2) + sym\left(\begin{bmatrix} K_{sf_2}' \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} ZC & F \end{bmatrix} \right) \prec \mathbf{0} \end{array} \tag{22}$$

and $K_{sf_2} \in \mathcal{K}_{sf}$.

When dealing with the simpler case of H_2 sf optimal control (problem (22) for which $C = \mathbf{1}$) , the exact H_2 sf optimal control may be computed via the solution of the following convex LMI optimization problem:

$$\begin{array}{ll} \min\limits_{X_2, R, T} & \mathrm{trace}(T) \\ \text{under} & X_2 \succ \mathbf{0} \\ & \begin{bmatrix} -T & C_1 X_2 + D_{12} R \\ X_2 C_1' + R' D_{12}' & -X_2 \end{bmatrix} \prec \mathbf{0} \\ & A X_2 + X_2 A' + BR + R'B' + B_1' B_1 \prec \mathbf{0} \end{array} \tag{23}$$

In [31], parameterization of theorem 2 is used to derive a coordinate-descent cross-decomposition algorithm allowing to compute a H_2 sub-optimal sof. This algorithm is recalled for sake of clarity.

Algorithm 3 *:*

1. *(Initialization step - k=1), choose a stabilizing sf gain* $K^0_{sf} \in \mathcal{K}_{sf}$.
2. *(Step k - first part), for this choice of* $K^k_{sf} = K^0_{sf}$*, solve the following LMI relaxation of minimization problem (22):*

$$\begin{array}{rcll} \Gamma^2_{k,1} & = & \min\limits_{P_2,Z,F} & \mathrm{trace}(B_1'P_2B_1) \\ & & \text{under} & P_2 \succ \mathbf{0} \\ & & & L_2(P_2,K^0_{sf_2},Z,F) = N(P_2) + sym\left(\left[\begin{array}{c} K^{0'}_{sf_2} \\ -\mathbf{1} \end{array}\right]\left[\begin{array}{cc} ZC & F \end{array}\right]\right) \prec \mathbf{0} \end{array} \tag{24}$$

At the optimum, freeze $Z = Z_k$ *and* $F = F_k$.

3. *(Step k - second part), for this choice of* Z_k *and* F_k*, solve the following LMI relaxation of minimization problem (22):*

$$\begin{array}{rcll} \Gamma^2_{k,2} & = & \min\limits_{P_2,K_{sf}} & \mathrm{trace}(B_1'P_2B_1) \\ & & \text{under} & P_2 \succ \mathbf{0} \\ & & & L_2(P_2,K_{sf_2},Z_k,F_k) = N(P_2) + sym\left(\left[\begin{array}{c} K'_{sf_2} \\ -\mathbf{1} \end{array}\right]\left[\begin{array}{cc} Z_kC & F_k \end{array}\right]\right) \prec \mathbf{0} \end{array} \tag{25}$$

At the optimum, freeze K_{sf}.

4. *(Termination step), if* $\Gamma_{k,1} - \Gamma_{k,2} < \epsilon$*, then stop,* $K_{sof} = -F_k^{-1}Z_k$*, otherwise* $k \leftarrow k+1$ *and go to step 2.*

Note that this algorithm always generates a non increasing sequence of H_2 sub-optimal costs:

$$\cdots \geq \Gamma_{k-1,2} \geq \Gamma_{k,1} \geq \Gamma_{k,2} \geq \cdots$$

Remark 4 *:*
The matrix K_{sf_2} *in (22) must belong to the set:*

$$\left\{K_{sf} \in \mathcal{K}_{sf} \;:\; \exists\, P_2 \in \mathbb{S}^{+*} \mid (A+BK_{sf})'P_2 + P_2(A+BK_{sf}) + (C_1+D_{12}K_{sf})'(C_1+D_{12}K_{sf}) \prec \mathbf{0}\right\} \tag{26}$$

This is not restrictive since for $K_{sf} \in \mathcal{K}_{sf}$*, it is always possible to find a matrix* $P \succ \mathbf{0}$ *such that:*

$$(A+BK_{sf})'P + P(A+BK_{sf}) + (C_1+D_{12}K_{sf})'(C_1+D_{12}K_{sf}) \prec \mathbf{0} \tag{27}$$

by choosing a matrix $P \succ W_o$ *where* W_o *is the observability grammian defined as the solution of the Lyapunov equation:*

$$(A+BK_{sf})'W_o + W_o(A+BK_{sf}) + (C_1+D_{12}K_{sf})'(C_1+D_{12}K_{sf}) = \mathbf{0} \tag{28}$$

For a given $K_{sf_2} \in \mathcal{K}_{sf}$, the LMI convex set $\mathcal{L}^{K_{sf_2}}_{sof}$ may be defined as:

$$\mathcal{L}^{K_{sf_2}}_{sof} = \left\{(P_2,Z,F) \in \mathbb{S}^{+*} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{m\times m} \;:\; \exists\, K_{sf_2} \in \mathcal{K}_{sf} \mid L_2(P_2,K_{sf_2},Z,F) \prec \mathbf{0}\right\} \tag{29}$$

This algorithm generally works well in practice and allows to get suboptimal H_2 sof in a moderate cpu time when the initializing $K^0_{sf_2}$ is adequately chosen, i.e. $\mathcal{L}^{K^0_{sf_2}}_{sof} \neq \emptyset$. As may be seen in tables 1, 2, 3, 4, this is not always the case and one has to resort to more efficient alternatives. Following remark 1 and noting that a set of initial stabilizing sof may be computed via algorithms 1 and 2, an alternate parameterization may be proposed.

$$L_2(P_2,K_{sof},H,F) = N(P_2) + \mathrm{sym}\left(\left[\begin{array}{c} C'K'_{sof} \\ -\mathbf{1} \end{array}\right]\left[\begin{array}{cc} H' & F' \end{array}\right]\right) \prec \mathbf{0} \tag{30}$$

where $K_{sf} = -F^{-T}H' \in \mathcal{K}_{sf}$. It leads to an alternate coordinate-descent cross-decomposition algorithm.

Algorithm 4 *:*

1. *(Initialization step - k=1), choose a stabilizing sof gain* $K_{sof} \in \mathcal{K}_{sof}^{n^1_{sof}}$ *or* $K_{sof} \in \mathcal{K}_{sof}^{n^2_{sof}}$.

2. *(Step k - first part), for this choice of* $K^k_{sof} = K^0_{sof}$*, solve the following LMI relaxation of minimization problem (22):*

$$
\begin{array}{rcll}
\Gamma^2_{k,1} & = & \min\limits_{P_2, Z, F} & \mathrm{trace}(B_1' P_2 B_1) \\
 & & \text{under} & P_2 \succ \mathbf{0} \\
 & & & L_2(P_2, K^k_{sof}, H, F) = N(P_2) + \mathrm{sym}\left(\begin{bmatrix} C' K^{k'}_{sof} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} H' & F' \end{bmatrix} \right) \prec \mathbf{0}
\end{array} \tag{31}
$$

At the optimum, freeze $H = H_k$ *and* $F = F_k$.

3. *(Step k - second part), for this choice of* H_k *and* F_k*, solve the following LMI relaxation of minimization problem (22):*

$$
\begin{array}{rcll}
\Gamma^2_{k,2} & = & \min\limits_{P_2, K_{sof}} & \mathrm{trace}(B_1' P_2 B_1) \\
 & & \text{under} & P_2 \succ \mathbf{0} \\
 & & & L_2(P_2, K_{sof}, H_k, F_k) = N(P_2) + \mathrm{sym}\left(\begin{bmatrix} C' K'_{sof} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} H'_k & F'_k \end{bmatrix} \right) \prec \mathbf{0}
\end{array} \tag{32}
$$

At the optimum, freeze $K_{sof} = K^k_{sof}$.

4. *(Termination step), if* $\Gamma_{k,1} - \Gamma_{k,2} < \epsilon$*, then stop,* $K_{sof} = K^k_{sof}$*, otherwise* $k \leftarrow k+1$ *and go to step 2.*

For a given $K_{sof} \in \mathcal{K}_{sof}$, the LMI convex set $\mathcal{L}_{sf}^{K_{sof}}$ is defined by:

$$
\mathcal{L}_{sf}^{K_{sof}} = \left\{ (P, H, F) \in \mathbb{S}^{+*} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{m \times m} \ : \ \exists\, K_{sof} \in \mathcal{K}_{sof} \mid L_2(P_2, K_{sof}, H, F) \prec \mathbf{0} \right\} \tag{33}
$$

Once again, algorithm 4 generally works well in practice except when $\mathcal{L}_{sf}^{K^0_{sof}} = \emptyset$. The idea is to use potentialities of randomized algorithms to generate non trivial sets of stabilizing sf for algorithm 3 and sets of stabilizing sof for algorithm 4 in which it will be easier to find at least one instance K^0_{sf} (resp. K^0_{sof}) such that $\mathcal{L}_{sof}^{K^0_{sf}} \neq \emptyset$ (resp. $\mathcal{L}_{sf}^{K^0_{sof}} \neq \emptyset$).

3.2 Mixed LMI/randomized algorithms for optimal H_2 sof control

The first algorithm is built upon the parameterization of theorem 2, coordinate-descent algorithm 3 and on the construction of a sufficiently representative subset of stabilizing sf $\mathcal{K}_{sf_2}^{n_{sf}}$.

Algorithm 5 *:*

1- *Compute a* H_2 *optimal sf* $K^0_{sf_2} = R^{-1} X_2 \in \mathcal{K}_{sf}$ *via the solution of the LMI problem (23):*

2- *From* $K^0_{sf_2}$*, generate a set* $\mathcal{K}_{sf_2}^{n_{sf}} \subset \mathcal{K}_{sf}$ *which is the collection of* n_{sf} *samples of stabilizing sf matrices via H.R. ;*

3- *Compute a set* $\mathcal{K}_{sof_2}^{n_{sof_2}} \subset \mathcal{K}_{sof}$ *which is the collection of* n_{sof_2} H_2 *suboptimal samples of stabilizing sof matrices:*

$\forall\, K^i_{sf} \in \mathcal{K}_{sf_2}^{n_{sf}}$*, if the LMI set* $\mathcal{L}_{sof}^{K^i_{sf_2}} \neq \emptyset$*, run the coordinate descent algorithm 3 and add the solution* $K_i = -F_i^{-1} Z_i$ *to the collection set* $\mathcal{K}_{sof_2}^{n_{sof_2}}$.

4- *Compute the best* H_2 *solution* $K^*_2 \in \mathcal{K}_{sof_2}^{n_{sof}}$ *out of* n_{sof_2} *choices.*

Note that an alternative to the first step for the initialization of algorithm 5 could be to compute a stabilizing sf via the solution of (11) or (36) defined in the section dedicated to optimal H_∞ sof control.

The natural alternative to the previous algorithm 5 is deduced from parameterization (30) and coordinate-descent algorithm 4.

Algorithm 6 *:*

1- *Compute a set $\mathcal{K}_{sof}^{n_{sof}^*}$ of stabilizing sof via algorithms 1 or 2.*

2- *Compute a set $\mathcal{K}_{sof_2}^{n_{sof_2}} \subset \mathcal{K}_{sof}$ which is the collection of n_{sof_2} H_2 suboptimal samples of stabilizing sof matrices:*

$\forall\ K_{sof}^i \in \mathcal{K}_{sof_2}^{n_{sof}^1}$ or $\mathcal{K}_{sof_2}^{n_{sof}^2}$, if the LMI set $\mathcal{L}_{sf}^{K_{sof_2}^i} \neq \emptyset$, run the coordinate descent algorithm 4 and add the solution $K_i = -F_i^{-1} Z_i$ to the collection set $\mathcal{K}_{sof_2}^{n_{sof_2}}$.

4- *Compute the best H_2 solution $K_2^* \in \mathcal{K}_{sof_2}^{n_{sof}}$ out of n_{sof_2} choices.*

This algorithm may be run in two different ways depending upon the choice of the initial stabilizing sof sets $\mathcal{K}_{sof}^{n_{sof}^1}$ or $\mathcal{K}_{sof}^{n_{sof}^2}$.

3.3 Comments on results from COMPl$_e$ib library for optimal H_2 sof control

First, note that algorithm 4 has been initialized with a stabilizing sof computed via the solver HIFOO [18]. Note also that the H_2 performance presented in the tables are the actual H_2 performance computed for the closed-loop system.

Figures 4 and 5 illustrate how algorithm 5 generates a path to a suboptimal solution in the set of admissible sof in the case of example `NN5`.

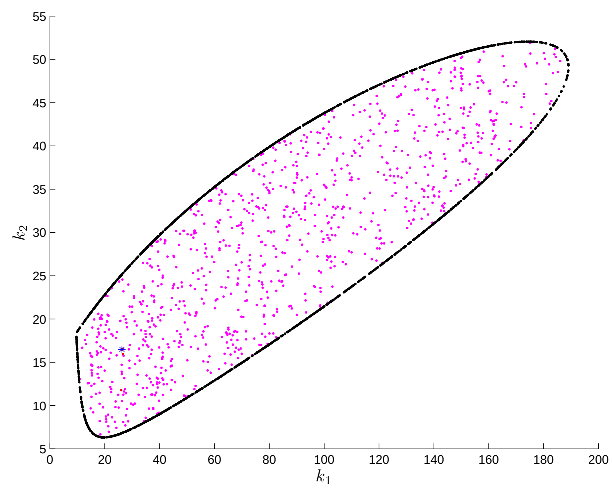

Figure 4: Suboptimal H_2 sof (algorithm 5) for NN5 example

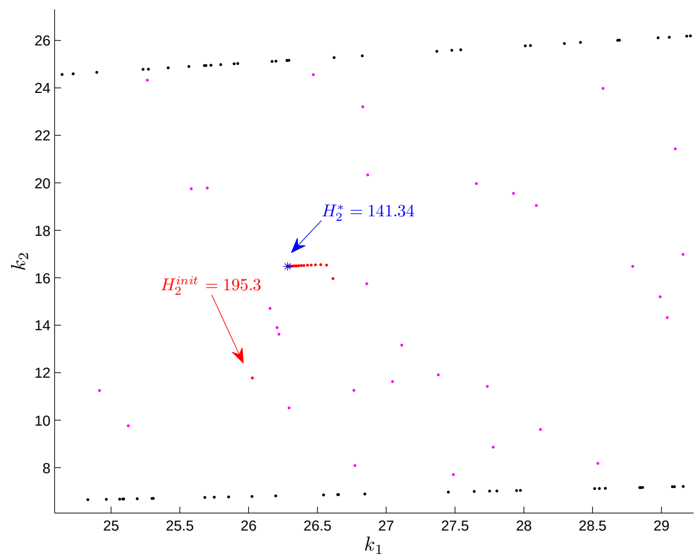

Figure 5: Zoom of the subobtimal path (algorithm 5) for NN5 example

- First of all, it is interesting to note that plants DIS2, NN2, HF2D10, HF2D11, HF2D18 are not really discriminant benchmarks for the optimal H_2 sof control problem. Except for very crude approximations given by picking the best element in the sets $\mathcal{K}_{sof}^{n_{sf}^1}$ or $\mathcal{K}_{sof}^{n_{sf}^2}$, all algorithms give an identical result. For instance, NN2 is a classical academic example for which the exact solution is known and may be obtained easily via analytic derivation [30]. All algorithms are able to get the optimal global solution which is somehow reassuring.

- Since $n_{sof}^1 \leq n_{sof}^2$, it could be expected that $\|T(s, K_{sof}^{sub_2})\|_2 = H_2\ s_2 \leq \|T(s, K_{sof}^{sub_1})\|_2 = H_2\ s_1$. As can be seen in tables of results 1, 2, 3, 4, this is far from being always the case (see instances AC11, AC18, HE4, DIS4, REA1, REA2, REA3, PAS). This point may be mainly explained by the random nature of the discrete sets $\mathcal{K}_{sof}^{n_{sof}^1}$ and $\mathcal{K}_{sof}^{n_{sof}^1}$. Moreover, these mixed results shows the necessity to perform an additional iterative LMI optimization step. Compared to the best result among all possible approaches, this gridding-type method gives rather crude suboptimal approximations with a large number of generated points even on less challenging instances such the ones mentioned above.

- The third remark is that less complex (from a numerical point of view) algorithms 3 and 4 may be an option to get a first fast approximation of a possible guaranteed performance for benchmarks that are not too hard to stabilize. As could be expected 3 and 4 fail on more demanding examples: AC5, AC13, AC18, JE2, PAS, NN1, NN5, NN6, NN7, NN12 and NN17 for algorithm 3 and AC1, AC11, AC13, DIS4, JE2, PAS, FS for algorithm 4. At first, it may be surprising that this last algorithm is not really competitive. This may be explained by the initialization which does not really take advantage of the parameterization leading to an unfeasible second step for several examples. Though, it should be noted that pure coordinate-descent algorithms cannot compete in general at the exception of "easy benchmarks" and instance AC18.

- Obviously, algorithm 5 gives the best results except for the very specific instance AC18.

- Results obtained for example NN6 should be compared to the one in reference [23] where a suboptimal cost is found to be 217 10^3 where here the worst cost is given by 12398 while the best is 1350.87.

4 H_∞ optimal synthesis via sof

This section follows the same lines as the previous section 3. Consequently, only an abridged version giving the main differences are presented.

4.1 Parameterizations and algorithms for H_∞ optimal control

First, the parameterization of theorem 1 is extended to give a solution to the problem (6) in the optimal H_∞ sof control case. Let us define:

$$N_\infty(P_\infty) = [\star]' \begin{bmatrix} \mathbf{0} & P_\infty \\ \star & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ A & B_\infty & B \end{bmatrix} + [\star]' \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\gamma^2\mathbf{1} \end{bmatrix} \begin{bmatrix} C_\infty & D_\infty & D_{\infty u} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \quad (34)$$

Theorem 3 *:*

The H_∞ optimal sof for (2) is given by $K^*_{sof_\infty} = -F^{-1*}Z^*$ where the triplet $(P^*_\infty, Z^*, F^*) \in \mathbb{S}^{+*} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{m\times m}$ is the global optimal solution of the non convex optimization problem (35):

$$\begin{array}{ll} \min\limits_{P_\infty, Z, F, K_{sf_\infty}} & \gamma^2 \\ \text{under} & P_\infty \succ \mathbf{0} \\ & N_\infty(P_\infty, K_{sf_\infty}, Z, F) + sym\left(\begin{bmatrix} K'_{sf_\infty} \\ K'_{w\infty} \\ -\mathbf{1} \end{bmatrix} \begin{bmatrix} ZC & ZD_{21} & F \end{bmatrix} \right) \prec \mathbf{0} \end{array} \quad (35)$$

with $K_{sf_\infty} \in \mathcal{K}_{sf}$ and $K_{fi_\infty} = \begin{bmatrix} K_{sf_\infty} & K_w \end{bmatrix}$ is feasible for the H_∞ optimal full-information problem.

It is recalled that the H_∞ optimal full-information problem is defined as problem (6) for which the measured output is given by $y = \begin{bmatrix} x \\ w \end{bmatrix}$ i.e. $C = \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix}'$ and $D_{21} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}'$ in (2) [39]. A convex solution for this last problem may be easily formulated:

$$\begin{array}{ll} \min\limits_{X_\infty, R, K_w} & \gamma^2 \\ \text{under} & X_\infty \succ \mathbf{0} \\ & \begin{bmatrix} AX_\infty + X_\infty A' + BR + R'B' & B_1 + BK_w & (C_1 + D_{12}R)' \\ (B_1 + BK_w)' & -\gamma^2\mathbf{1} & (D_{11} + D_{12}K_w)' \\ (C_1 + D_{12}R) & (D_{11} + D_{12}K_w) & -\mathbf{1} \end{bmatrix} \prec \mathbf{0} \end{array} \quad (36)$$

where $K_{fi} = \begin{bmatrix} RX\infty^{-1} & K_w \end{bmatrix}$.

A coordinate-descent algorithm very similar to algorithm 3 may be designed. The only difference is that the initialization is made by using the complete solution K_{fi} of the H_∞ optimal full-information problem given by (36). As the basic principles of this algorithm have been dealt with in the previous section, it is not recalled here.

As for the H_2 case, an alternate parameterization may be used when $K_{sof} \in \mathcal{K}_{sof}$ is given.

$$L_\infty(P_\infty, K_{sof}, K_1, K_2, F) = N(P_\infty) + sym\left(\begin{bmatrix} K_1 \\ K_2 \\ F \end{bmatrix} \begin{bmatrix} K_{sof}C & K_{sof}D_{21} & -\mathbf{1} \end{bmatrix} \right) \prec \mathbf{0} \quad (37)$$

It leads to an alternate coordinate-descent cross-decomposition algorithm similar to algorithm 3.

In addition to these two algorithms, two mixed LMI/randomized algorithms for optimal H_∞ sof control are built along the lines of subsection 3.2.

4.2 Comments on results from COMPl$_e$ib library for optimal H_∞ sof control

For the case of optimal H_∞ sof control, HIFOO [10, 19, 20, 18] may be considered as the most numerically efficient package at the moment as reported in different dedicated publications. It has been run 10 times on every tested example. It is recalled that the default options of this package uses 3 random initialization when searching for a solution. The obtained result will give the reference for the H_∞ norms computed with our algorithms. Moreover, algorithm 4 is initialized with a stabilizing sof found by HIFOO. In the presented results, no computation time has been exhibited but it is important to mention that HIFOO is much more efficient than any other tested LMI/randomized algorithm. This is mainly due to the LMI optimizations steps that still remain too demanding for large scale benchmarks.

- In every case, when they succeed, iterative mixed LMI/randomized algorithms (particularly algorithm 5) outperforms results given in [34]. In many cases, algorithm 5 is close to the solution given by HIFOO and even better in some cases.

- CSE2 is a demanding problem in terms of computation time due to the large number of variables involved in the optimization but appears to be far less challenging with respect to the optimal H_∞ sof control problem. Every algorithm leads to an identical result (except for algorithm 4 which probably needs more iterations to converge to the optimum).

- None of the presented LMI/randomized algorithms works fine on example JE2. Note that the performance channels have been artificially selected in COMPl$_e$ib. Note also that the initialization $K_{sf} = K_{sof}C$ where $K_{sof} \in \mathcal{K}_{sof}^{n^*_{sof}}$ fails wheras it should give a solution. Different tested LMI solvers (SeDuMi, SDPT3, SDPA) display a bad numerical behavior leading to a diagnosis of infeasibility while a simple solution may be constructed (see remark 3).

- Algorithm 5 (and algorithm 3) does not give any result on examples AC14, HF2D10, HF2D11, HF2D14, HF2D15, HF2D16, HF2D17 but may be considered as the more interesting one to get results close to HIFOO and better in some few cases. One of the reason of this relative failure is probably due to the construction of random full information static gains *a priori* structured as:

$$K_{fi\infty} = \begin{bmatrix} K_{sf_\infty} & \mathbf{0} \end{bmatrix} \quad (38)$$

where K_{sf_∞} is randomly chosen in $\mathcal{K}_{sf}$ via Hit-and-Run.

5 Conclusions

In this article, two different mixed LMI/Hit-and-run algorithms have been proposed for the stabilization of LTI systems via static output feedback. Evaluation of these algorithms on the 53 benchmarks that are not open-loop stable shows that both may be considered as effective tools for the construction of non trivial sets of stabilizing sof. These examples exhibit relatively different structural features and different size of the state-space realization as well. Due to the relative limitations of SDP solvers in terms of computation time for large scale LTI systems, these approaches cannot be considered as competitive packages with respect to the last dedicated packages based on nonsmooth optimization [4, 18]. Nevertheless, we strongly believe that a promising new perspective is possible in the continuity of the seminal work of [13].

6 Ackowledgements

The authors gratefully acknowledge the financial contribution of CNRS and RFBR via PICS No. 4281. One of the authors (D.A.) would like to thank Michael Overton and the Courant Institute of Mathematical Sciences of NYU, New York City, New York, USA, for hospitality, where this work was done. Support for this work was provided in part by the grant DMS-0714321 from the U.S. National Science Foundation. He would also like to thank Didier Henrion for helpful comments on early versions of this draft.

A Tables resuming obtained numerical results for H_2 optimal sof control

The following notations are used in tables 1, 2, 3, 4.

- Name* means that performance channels are artificially built up in COMPL$_e$ib for example Name
- OLS stands for Open-Loops Stability
- OLMS stands for open-loop marginal stable (a unique eigenvalue at 0 or multiple eigenvalue with 0 real part but scalar associated Jordan blocks)
- OLNS stands for open-loop non stable `max(real(eig(A)))>0`
- n^1_{sof}/n_{sf} stands for the percentage of stabilizing static output feedback found by the algorithm 1
- H_2 $s_1|s_2$ stands for the best H_2 norm of the computed SOF samples with algorithm 1 and 2
- H_2 Alg. i stands for the best H_2 norm computed via algorithm i (default =30 iterations)

- $number$* stands for a non stationary value reached by the criterion at the end of iterations (i.e. the sof matrix and the actual H_2 cost could be significantly improved by increasing the number of iterations of the coordinate-descent algorithm.

- Inf is used when the closed-loop feedthrough matrix is not zero due to $D_{11} \neq \mathbf{0}$ or $D_{12} \neq \mathbf{0}$ and $D_{12} \neq \mathbf{0}$

- $number$# stands for a value for which the coordinate-descent algorithm failed at step k second part

- $*$ stands for a complete failure of the algorithm

- pstl is the abbreviation for problem size too large

- Alg. 6.1 stands for algorithm 6 initialized with $K_{sof} \in \mathcal{K}_{sof}^{n_{sof}^1}$

- Alg. 6.2 stands for algorithm 6 initialized with $K_{sof} \in \mathcal{K}_{sof}^{n_{sof}^2}$.

Ex.	n_x	n_u	n_y	OLS	n^1_{sof}/n_{sf}	H_2 Alg. 3	H_2 Alg. 4	H_2 $s_1\|s_2$	H_2 Alg. 5	H_2 Alg. 6.1	H_2 Alg. 6.2
AC1	5	3	3	OLMS	240/1000	0.0331	1472.1#	6.3|0.1533	$2.5\ 10^{-6}$*	$8.7\ 10^{-7}$	$3.4\ 10^{-7}$
AC2	5	3	3	OLMS	334/1000	0.0784*	2.77	27.1|0.2321	0.0505*	0.0504	0.0503
AC5*	4	2	2	OLNS	283/1000	*	1466.7	1481.9|1500.4	1466.7	1466.7	1466.7
AC9	10	4	5	OLNS	34/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
AC10	55	2	2	OLNS	*	*	*	*	*	*	*
AC11*	5	2	4	OLNS	996/1000	4.4493*	2728.7#	7|9.5524	3.9424*	3.9459*	4.2578*
AC12	4	3	4	OLNS	997/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
AC13*	28	3	4	OLNS	39/1000	*	3934#	172.5458|978.0304	132.4094*	132.4679*	132.48*
AC14	40	3	4	OLNS	13/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
AC18	10	2	2	OLNS	17/1000	*	21.7774*	25.3|43.5529	22.5732	19.742*	23.6158*
HE1	4	2	1	OLNS	93/1000	0.0955	0.2177*	0.1|0.1671	0.0954	0.0954	0.0958*
HE3	8	4	6	OLNS	12/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
HE4	8	4	6	OLNS	1000/1000	24.0053	27.0403*	499.5338|1526	21.72*	499.5338#	1526#
HE5	4	2	2	OLNS	13/1000	Inf	Inf	Inf |Inf	Inf	Inf	Inf
HE6	20	4	6	OLNS	321/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
HE7	20	4	6	OLNS	327/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
DIS2*	3	2	2	OLNS	842/1000	1.416	1.416	22.4|21.5901	1.416	1.416	1.416
DIS4*	6	4	6	OLNS	1000/1000	1.6924	30663#	262.9|327.8694	1.6924	1.6924	17.25*
DIS5	4	2	2	OLNS	725/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf

Table 1: Numerical results for the COMPl$_e$ib library and H_2 optimal sof control

Ex.	n_x	n_u	n_y	OLS	n^1_{sof}/n_{sf}	H_2 Alg. 3	H_2 Alg. 4	H_2 $s_1 \mid s_2$	H_2 Alg. 5	H_2 Alg. 6.1	H_2 Alg. 6.2
JE2*	21	3	3	OLMS	94/1000	*	6330.5#	1200.24|1128	1014.25	1170.75	1128#
JE3	24	3	6	OLMS	44/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
REA1*	4	2	3	OLNS	999/1000	1.8204	1.9454*	3.3|3.3571	1.8204	1.8204	1.8928*
REA2*	4	2	2	OLNS	554/1000	1.8615	1.9220*	4.6|21.0951	1.8615	1.8618	1.9404*
REA3*	12	1	3	OLNS	965/1000	12.0873	12.4981*	49.97|51.3185	12.0873	12.3960*	18.4358*
WEC1*	10	3	4	OLNS	775/1000	7.5050*	9.1176*	93.7|22.7250	7.3611	8.3272*	7.6922*
BDT2	82	4	4	OLMS	43/1000	pstl	pstl	1.6|1.3426	pstl	pstl	pstl
IH	21	11	10	OLMS	63/1000	2.1144*	2.6772*	23.2|8.7342	1.6617*	2.3211*	2.8052*
CSE2	60	2	30	OLNS	10/10	0.0089	0.0089	0.0126|0.0238	0.0089	0.0089	0.0089
PAS	5	1	3	OLMS	236/1000	*	1331500#	0.1629|2.2183	0.0092	0.0256	2.2183#

Table 2: Numerical results for the COMPl$_e$ib library and H_2 optimal sof control

<table>
<tr><th>Ex.</th><th>n_x</th><th>n_u</th><th>n_y</th><th>OLS</th><th>n^1_{sof}/n_{sf}</th><th>H_2 Alg. 3</th><th>H_2 Alg. 4</th><th>H_2 $s_1|s_2$</th><th>H_2 Alg. 5</th><th>H_2 Alg. 6.1</th><th>H_2 Alg. 6.2</th></tr>
<tr><td>TF1</td><td>7</td><td>2</td><td>4</td><td>OLMS</td><td>81/1000</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>TF2</td><td>7</td><td>2</td><td>3</td><td>OLMS</td><td>189/1000</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>TF3</td><td>7</td><td>2</td><td>3</td><td>OLMS</td><td>6/1000</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>NN1*</td><td>3</td><td>1</td><td>2</td><td>OLNS</td><td>629/1000</td><td>*</td><td>41.8350*</td><td>75.7|49.6241</td><td>41.8348</td><td>41.8348</td><td>41.8348</td></tr>
<tr><td>NN2</td><td>2</td><td>1</td><td>1</td><td>OLMS</td><td>1000/1000</td><td>1.5651</td><td>1.5651</td><td>1.85| 1.9193</td><td>1.5651</td><td>1.5651</td><td>1.561</td></tr>
<tr><td>NN5*</td><td>7</td><td>1</td><td>2</td><td>OLNS</td><td>84/1000</td><td>*</td><td>141.455</td><td>275.6|141.6009</td><td>141.4555</td><td>141.4555</td><td>141.4555</td></tr>
<tr><td>NN6</td><td>9</td><td>1</td><td>4</td><td>OLNS</td><td>983/1000</td><td>*</td><td>1726.5*</td><td>12398|3000.2</td><td>1350.87</td><td>11763*</td><td>1563.61*</td></tr>
<tr><td>NN7</td><td>9</td><td>1</td><td>4</td><td>OLNS</td><td>620/1000</td><td>*</td><td>133.2431*</td><td>187.9|140.1334</td><td>133.2422</td><td>133.2422</td><td>133.2422</td></tr>
<tr><td>NN9</td><td>5</td><td>3</td><td>2</td><td>OLNS</td><td>7/1000</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>NN12*</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>28/1000</td><td>*</td><td>20.1532*</td><td>19.5595|33.124</td><td>18.9256*</td><td>19.2346*</td><td>19.1463*</td></tr>
<tr><td>NN13</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>77/1000</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>NN14</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>44/1000</td><td>Inf</td><td>Inf</td><td>Inf</td><td>Inf|Inf</td><td>Inf</td><td>Inf</td></tr>
<tr><td>NN15</td><td>3</td><td>2</td><td>2</td><td>OLMS</td><td>821/1000</td><td>0.0485</td><td>3.5881</td><td>2.2074|0.3038</td><td>0.0485</td><td>0.0485</td><td>0.0485</td></tr>
<tr><td>NN16</td><td>8</td><td>4</td><td>4</td><td>OLMS</td><td>61/1000</td><td>0.3164*</td><td>0.3074*</td><td>1.8111|5.1376</td><td>0.2994*</td><td>0.2982*</td><td>0.3078*</td></tr>
<tr><td>NN17</td><td>3</td><td>2</td><td>1</td><td>OLNS</td><td>125/1000</td><td>*</td><td>9.4559</td><td>134.1|13.3228</td><td>9.4559</td><td>9.4559</td><td>9.4559</td></tr>
</table>

Table 3: Numerical results for the COMPl$_e$ib library and H_2 optimal sof control

Ex.	n_x	n_u	n_y	OLS	n^1_{sof}/n_{sf}	H_2 Alg. 3	H_2 Alg. 4	H_2 $s_1\|s_2$	H_2 Alg. 5	H_2 Alg. 6.1	H_2 Alg. 6.2
HF2D10	5	2	3	OLNS	991/1000	71248.7	71250*	277980|477510	71248.7	71248.7	71250*
HF2D11	5	2	3	OLNS	993/1000	85148.6	85148	29888000|164540	85148.6	85148.6	85148
HF2D14	5	2	4	OLNS	1000/1000	373561	13052693	3234600|1868300	373561	373561	1803800*
HF2D15	5	2	4	OLNS	1000/1000	297135	297135	29647000|601330	297135	297135	554819*
HF2D16	5	2	4	OLNS	998/1000	286492	1572399	6228900|2089000	285340	285340	1140483*
HF2D17	5	2	4	OLNS	1000/1000	375601	2064883	11874000|3226700	375600	375600	498873*
HF2D18	5	2	2	OLNS	755/1000	27.8436	27.8487*	46.8|28.6525	27.8436	27.8436	27.850*
TMD	6	2	4	OLNS	654/1000	Inf	Inf	Inf|Inf	Inf	Inf	Inf
FS*	5	1	3	OLNS	977/1000	16851	596328#	35532|30391	16851	16851	24849

Table 4: Numerical results for the COMPl$_e$ib library and H_2 optimal sof control

B Tables resuming obtained numerical results for H_∞ optimal sof control

The following notations are used in tables 5, 6, 7, 8.

- `Name*` means that performance channels are artificially built up in COMPL$_e$ib for example `Name`
- OLS stands for Open-Loops Stability
- OLMS stands for open-loop marginal stable (a unique eigenvalue at 0 or multiple eigenvalue with 0 real part but scalar associated Jordan blocks)
- OLNS stands for open-loop non stable `max(real(eig(A)))>0`
- n^1_{sof}/n_{sf} stands for the percentage of stabilizing static output feedback found by the algorithm 1
- H_∞ $s_1|s_2$ stands for the best H_∞ norm of the computed SOF samples with algorithm 1 and 2
- H_∞ Alg. i stands for the best H_∞ norm computed via algorithm i (default =30 iterations)
- $number$* stands for a non stationary value reached by the criterion at the end of iterations (i.e. the sof matrix and the actual H_∞ cost could be significantly improved by increasing the number of iterations of the coordinate-descent algorithm.
- $number$# stands for a value for which the coordinate-descent algorithm failed at step k second part
- $*$ stands for a complete failure of the algorithm
- pstl is the abbreviation for problem size too large
- nk is the abbreviation for not known
- Alg. 6.1 stands for algorithm 6 initialized with $K_{sof} \in \mathcal{K}_{sof}^{n^1_{sof}}$
- Alg. 6.2 stands for algorithm 6 initialized with $K_{sof} \in \mathcal{K}_{sof}^{n^2_{sof}}$.

<table>
<tr><th>Ex.</th><th>n_x</th><th>n_u</th><th>n_y</th><th>OLS</th><th>n_{sof}^1/n_{sf}</th><th>H_∞ Alg. 3</th><th>H_∞ Alg. 4</th><th>H_∞ $s_1|s_2$</th><th>H_∞ Alg. 5</th><th>H_∞ Alg. 6.1</th><th>H_∞ Alg. 6.2</th><th>H_∞ [18]</th><th>H_∞ [34]</th></tr>
<tr><td>AC1</td><td>5</td><td>3</td><td>3</td><td>OLMS</td><td>240/1000</td><td>0.05*</td><td>4.78</td><td>15.9|0.1829</td><td>1.7610^{-6}</td><td>0.0162*</td><td>0.0688*</td><td>$4.137\ 10^{-7}$</td><td>5.08</td></tr>
<tr><td>AC2</td><td>5</td><td>3</td><td>3</td><td>OLMS</td><td>334/1000</td><td>0.1385*</td><td>0.4341</td><td>41.8|0.3510</td><td>0.1115</td><td>0.122*</td><td>0.1234*</td><td>0.1115</td><td>5.08</td></tr>
<tr><td>AC5*</td><td>4</td><td>2</td><td>2</td><td>OLNS</td><td>283/1000</td><td>721.4225</td><td>831.36</td><td>717.0052|760.94</td><td>661.7</td><td>676.2</td><td>673.8391</td><td>669.56</td><td>2930</td></tr>
<tr><td>AC9</td><td>10</td><td>4</td><td>5</td><td>OLNS</td><td>34/1000</td><td>1.0162</td><td>13.50*</td><td>51916|59.27</td><td>1.0061*</td><td>2.77*</td><td>1.8726*</td><td>1.0029</td><td>19.3</td></tr>
<tr><td>AC10</td><td>55</td><td>2</td><td>2</td><td>OLNS</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr>
<tr><td>AC11*</td><td>5</td><td>2</td><td>4</td><td>OLNS</td><td>996/1000</td><td>3.0179</td><td>36.70#</td><td>6.5|7.97</td><td>2.8375*</td><td>2.859*</td><td>2.9166*</td><td>2.8335</td><td>4.49</td></tr>
<tr><td>AC12</td><td>4</td><td>3</td><td>4</td><td>OLNS</td><td>997/1000</td><td>0.3238*</td><td>2350.53#</td><td>5.8022|41550</td><td>0.3165</td><td>1.937</td><td>41550#</td><td>0.3120</td><td>2.15</td></tr>
<tr><td>AC13*</td><td>28</td><td>3</td><td>4</td><td>OLNS</td><td>39/1000</td><td>*</td><td>1893.43#</td><td>395.0404|805.63</td><td>429.8111</td><td>395.0404#</td><td>805.63#</td><td>163.33</td><td>nk</td></tr>
<tr><td>AC14</td><td>40</td><td>3</td><td>4</td><td>OLNS</td><td>13/1000</td><td>*</td><td>319.31#</td><td>1111.2|43384</td><td>*</td><td>1011.6</td><td>43384#</td><td>101.7203</td><td>nk</td></tr>
<tr><td>AC18</td><td>10</td><td>2</td><td>2</td><td>OLNS</td><td>17/1000</td><td>*</td><td>17.07#</td><td>16.2|67.78</td><td>10.6214</td><td>11.47*</td><td>67.78#</td><td>12.6282</td><td>8000</td></tr>
<tr><td>HE1</td><td>4</td><td>2</td><td>1</td><td>OLNS</td><td>93/1000</td><td>0.1539</td><td>0.309*</td><td>0.5|0.1536</td><td>0.1538*</td><td>0.1555*</td><td>0.1536</td><td>0.1539</td><td>nk</td></tr>
<tr><td>HE3</td><td>8</td><td>4</td><td>6</td><td>OLNS</td><td>12/1000</td><td>*</td><td>1.045*</td><td>33641|2526.9</td><td>0.8291*</td><td>8174.8</td><td>1.4949*</td><td>0.8061</td><td>nk</td></tr>
<tr><td>HE4</td><td>8</td><td>4</td><td>6</td><td>OLNS</td><td>1000/1000</td><td>*</td><td>47.78</td><td>100.7373|1154</td><td>22.8282</td><td>88.03</td><td>1154#</td><td>22.8282</td><td>nk</td></tr>
<tr><td>HE5</td><td>4</td><td>2</td><td>2</td><td>OLNS</td><td>13/1000</td><td>*</td><td>707.76#</td><td>66.5047|61.98</td><td>17.6061*</td><td>39.9244</td><td>37.8893</td><td>8.8952</td><td>nk</td></tr>
<tr><td>HE6</td><td>20</td><td>4</td><td>6</td><td>OLNS</td><td>321/1000</td><td>496.4097</td><td>36935#</td><td>26494|1448.8</td><td>401.7698</td><td>416.47</td><td>1448.8#</td><td>192.3445</td><td>nk</td></tr>
<tr><td>HE7</td><td>20</td><td>4</td><td>6</td><td>OLNS</td><td>327/1000</td><td>493.2068</td><td>380.22*</td><td>160450|3068.5</td><td>477.7</td><td>353.9425</td><td>3068.5#</td><td>192.3885</td><td>nk</td></tr>
<tr><td>DIS2*</td><td>3</td><td>2</td><td>2</td><td>OLNS</td><td>842/1000</td><td>1.2507*</td><td>1.1050*</td><td>1.9|1.5152</td><td>1.0232*</td><td>1.0244*</td><td>1.0251*</td><td>1.0412</td><td>nk</td></tr>
<tr><td>DIS4*</td><td>6</td><td>4</td><td>6</td><td>OLNS</td><td>1000/1000</td><td>1.0368</td><td>42372#</td><td>26.2|2.99</td><td>0.7672*</td><td>0.7404*</td><td>0.7667*</td><td>0.7394</td><td>nk</td></tr>
<tr><td>DIS5</td><td>4</td><td>2</td><td>2</td><td>OLNS</td><td>725/1000</td><td>*</td><td>1032.5*</td><td>17806|10569</td><td>1030.82*</td><td>1031.5*</td><td>1031.58</td><td>1035.5</td><td>nk</td></tr>
</table>

Table 5: Numerical results for the COMPl$_e$ib library and H_∞ optimal sof control

Ex.	n_x	n_u	n_y	OLS	n^1_{sof}/n_{sf}	H_∞ Alg. 3	H_∞ Alg. 4	H_∞ $s_1\|s_2$	H_∞ Alg. 5	H_∞ Alg. 6.1	H_∞ Alg. 6.2	H_∞ [18]
JE2*	21	3	3	OLMS	94/1000	*	365.09#	368.88|366.62	*	368.88#	366.6#	183.3512
JE3	24	3	6	OLMS	44/1000	119.2782	22.178#	18.2705|32.2356	9.194	18.2705#	32.2356#	5.0963
REA1*	4	2	3	OLNS	999/1000	0.9624*	25.69#	3.5|0.8972	0.8661*	0.8707*	0.8748*	0.8694
REA2*	4	2	2	OLNS	554/1000	1.1590*	1.2326*	2.7|1.2274	1.1482*	1.1484*	1.15*	1.1492
REA3*	12	1	3	OLNS	965/1000	74.2513	2209.5#	74.4|76.69	74.2513	74.2513	74.71	74.2513
WEC1*	10	3	4	OLNS	775/1000	4.5780	142.8#	65.2|14.18	4.1055*	4.369*	9.43*	4.0502
BDT2	82	4	4	OLMS	43/1000	pstl	pstl	3.5|2.90	pstl	pstl	pstl	0.6471
IH	21	11	10	OLMS	63/1000	2.9276	5.4339*	57.1|11.60	2.4252*	3.2850*	4.54*	0.6521
CSE2	60	2	30	OLNS	10/10	0.02	0.1339*	0.023|0.02	0.02	0.02	0.02	0.0201
PAS	5	1	3	OLMS	236/1000	*	86058918455#	0.1935|5.7375	18.28	0.0087	0.00966	32.2258

Table 6: Numerical results for the COMPl$_e$ib library and H_∞ optimal sof control

<table>
<tr><th>Ex.</th><th>n_x</th><th>n_u</th><th>n_y</th><th>OLS</th><th>n^1_{sof}/n_{sf}</th><th>H_∞ Alg. 3</th><th>H_∞ Alg. 4</th><th>H_∞ $s_1|s_2$</th><th>H_∞ Alg. 5</th><th>H_∞ Alg. 6.1</th><th>H_∞ Alg. 6.2</th><th>H_∞ [18]</th></tr>
<tr><td>TF1</td><td>7</td><td>2</td><td>4</td><td>OLMS</td><td>81/1000</td><td>1.3785</td><td>2.4481*</td><td>275.4|1.6808</td><td>0.3799*</td><td>0.4462*</td><td>0.392*</td><td>0.3736</td></tr>
<tr><td>TF2</td><td>7</td><td>2</td><td>3</td><td>OLMS</td><td>189/1000</td><td>5200</td><td>5200</td><td>5200|5200</td><td>5200</td><td>5200</td><td>5200</td><td>5200</td></tr>
<tr><td>TF3</td><td>7</td><td>2</td><td>3</td><td>OLMS</td><td>6/1000</td><td>0.3287</td><td>4101#</td><td>8.1803|17.71</td><td>0.3264</td><td>0.7051*</td><td>0.7240*</td><td>0.4567</td></tr>
<tr><td>NN1*</td><td>3</td><td>1</td><td>2</td><td>OLNS</td><td>629/1000</td><td>*</td><td>14.02*</td><td>51.5|13.97</td><td>13.4580*</td><td>13.9534*</td><td>13.5955*</td><td>13.9089</td></tr>
<tr><td>NN2</td><td>2</td><td>1</td><td>1</td><td>OLMS</td><td>1000/1000</td><td>2.2209</td><td>2.2209*</td><td>2.84|4.8691</td><td>2.2050*</td><td>2.2209</td><td>2.2209*</td><td>2.2216</td></tr>
<tr><td>NN5*</td><td>7</td><td>1</td><td>2</td><td>OLNS</td><td>84/1000</td><td>*</td><td>266.5455</td><td>1238|268.7</td><td>266.5445</td><td>266.5455</td><td>266.5455</td><td>266.54</td></tr>
<tr><td>NN6</td><td>9</td><td>1</td><td>4</td><td>OLNS</td><td>983/1000</td><td>9648.8</td><td>7408.73#</td><td>6793|6898</td><td></td><td>5602</td><td>6782</td><td>5602</td></tr>
<tr><td>NN7</td><td>9</td><td>1</td><td>4</td><td>OLNS</td><td>620/1000</td><td>74.0372</td><td>74.03*</td><td>114|101.90</td><td>74.0372</td><td>74.0372</td><td>74.0372</td><td>74.0757</td></tr>
<tr><td>NN9</td><td>5</td><td>3</td><td>2</td><td>OLNS</td><td>7/1000</td><td>*</td><td>3220#</td><td>178.7001|56.37</td><td>32.392*</td><td>178.7001#</td><td>31.03*</td><td>28.6633</td></tr>
<tr><td>NN12*</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>28/1000</td><td>*</td><td>17.34*</td><td>18.5436|21.365</td><td>16.3116*</td><td>16.6343*</td><td>16.72*</td><td>16.3925</td></tr>
<tr><td>NN13</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>77/1000</td><td>14.0909</td><td>5086</td><td>115.7|26.5693</td><td>14.0579</td><td>14.5894*</td><td>21.12*</td><td>14.0589</td></tr>
<tr><td>NN14</td><td>6</td><td>2</td><td>2</td><td>OLNS</td><td>44/1000</td><td></td><td>17.4757</td><td>48173# 210.2|29.21</td><td>17.4757*</td><td>17.4763*</td><td>22.8649*</td><td>17.4778</td></tr>
<tr><td>NN15</td><td>3</td><td>2</td><td>2</td><td>OLMS</td><td>821/1000</td><td>16.0593</td><td>0.5368#</td><td>0.7177|0.0997</td><td></td><td>0.0981</td><td>0.0980*</td><td>0.0982</td></tr>
<tr><td>NN16</td><td>8</td><td>4</td><td>4</td><td>OLMS</td><td>61/1000</td><td>0.9661</td><td>1.2651*</td><td>2.4484|15.9</td><td>0.9592*</td><td>0.9862*</td><td>1.4532*</td><td>0.9556</td></tr>
<tr><td>NN17</td><td>3</td><td>2</td><td>1</td><td>OLNS</td><td>125/1000</td><td>11.2219</td><td>11.2182</td><td>24.4|11.6287</td><td>11.2182</td><td>11.2182</td><td>11.2182</td><td>11.2182</td></tr>
</table>

Table 7: Numerical results for the COMPl$_e$ib library and H_∞ optimal sof control

Ex.	n_x	n_u	n_y	OLS	n^1_{sof}/n_{sf}	H_∞ Alg. 3	H_∞ Alg. 4	H_∞ $s_1 \mid s_2$	H_∞ Alg. 5	H_∞ Alg. 6.1	H_∞ Alg. 6.2	H_∞ [18]
HF2D10	5	2	3	OLNS	991/1000	*	491918#	138240$\mid$82314	*	82824	82314#	79853
HF2D11	5	2	3	OLNS	993/1000	*	104091#	124220$\mid$78843	*	78248	78843#	7719
HF2D14	5	2	4	OLNS	1000/1000	*	2565087#	751400$\mid$557008	*	561600	557008#	53156
HF2D15	5	2	4	OLNS	1000/1000	*	505064#	622030$\mid$202610	*	187640	202610#	17521
HF2D16	5	2	4	OLNS	998/1000	*	9075335746#	273090$\mid$465790	*	458560	465790#	44432
HF2D17	5	2	4	OLNS	1000/1000	*	347800#	544330$\mid$303380	*	301170	303380#	30024
HF2D18	5	2	2	OLNS	755/1000	187.99	187.09	187.4$\mid$183.4	156.39	154.9910	183.4#	124.7259
TMD	6	2	4	OLNS	654/1000	33.7305	2.8256*	106.6$\mid$103.4064	2.1622*	2.4703*	49.95*	2.5267
FS*	5	1	3	OLNS	977/1000	*	8584413#	415990$\mid$87160	*	91142	87160#	96925

Table 8: Numerical results for the COMPl$_e$ib library and H_∞ optimal sof control

References

[1] B.D.O. Anderson, N.K. Bose, and E.I. Jury. Output feedback stabilization and related problems - solutions via decision methods. *IEEE Trans. on Automat. Control*, 20(1), February 1975.

[2] P. Apkarian and D. Noll. Controller design via nonsmooth multi-directional search. *SIAM J. on Control and Optimization*, 44(6):1923–1949, 2006.

[3] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

[4] P. Apkarian, D. Noll, and A. Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. *SIAM J. on Control and Optimization*, 47(3):1516–1546, 2008.

[5] P. Apkarian, D. Noll, and H. Tuan. Fixed-order H_∞ control design via a partially augmented lagrangian method. *International Journal of Robust and Nonlinear Control*, 13(12):1137–1148, October 2003.

[6] D. Arzelier, E. Gryazina, D. Peaucelle, and Boris Polyak. Mixed lmi/hit-and-run alorithms for static ouput stabilization: H_2 and h_∞ performances. Technical Report XXX, LAAS-CNRS, Toulouse, October 2009. URL: `http://www.laas.fr/~arzelier/`.

[7] A. Astolfi and P. Colaneri. Hankel/Toeplitz matrices and the static output feedback stabilization problem. *Mathematics of Control, Signals, and Systems (MCSS)*, 17(4):231–268, 2005.

[8] J. Bernussou, J.C. Geromel, and P.L.D. Peres. A linear programing oriented procedure for quadratic stabilization of uncertain systems. *Systems & Control Letters*, 13:65–72, July 1989.

[9] J.V. Burke, D. Henrion, A.S. Lewis, and M.L Overton. Hifoo - a matlab package for fixed-order controller design and h-infinity optimization. In *IFAC Symposium On RObust CONtrol Design, ROCOND06*, Toulouse, France, June 2006.

[10] J.V. Burke, D. Henrion, A.S. Lewis, and M.L. Overton. Stabilization via nonsmooth, nonconvex optimization. *IEEE Trans. on Automat. Control*, 51:1760–1769, November 2006.

[11] M.C. de Oliveira and J.C. Geromel. Numerical comparison of output feedback design methods. In *American Control Conference*, Albuquerque, New Mexico, June 1997.

[12] L. El Ghaoui, F. Oustry, and M. AitRami. A cone complementarity linearization algorithm for static ouput-feedback and related problems. *IEEE Trans. on Automat. Control*, 42(8):1171–1176, August 1997.

[13] Y. Fujisaki, Y. Oishi, and R. Tempo. Mixed deterministic/randomized methods for fixed order controller design. *IEEE Trans. on Automat. Control*, 53(9):2033–2047, 2008.

[14] J.C. Geromel, P.L.D. Peres, and S.R. Souza. Convex analysis of output feedback control problems: Robust stability and performance. *IEEE Trans. on Automat. Control*, 41(7):997–1003, July 1996.

[15] J.C. Geromel, C.C. de Souza, and R.E. Skelton. Static output feedback controllers: Stability and convexity. *IEEE Trans. on Automat. Control*, 43(1):120–125, January 1998.

[16] K.M. Grigoriadis and E.B. Beran. *Advances in Linear Matrix Inequality Methods in Control*, chapter 13 : Alternating Projection Algorithms for Linear Matrix Inequalities Problems with Rank Constraints, pages 251–267. SIAM, Philadelphia, 2000. Edited by L. El Ghaoui and S.-I. Niculescu.

[17] E.N. Gryazina and B.T. Polyak. Stability domain in the parameter space: D-decomposition revisited. *Automatica*, 42(1):13–26, 2006.

[18] S. Gumussoy, D. Henrion, M. Millstone, and M.L. Overton. Multiobjective robust control with HIFOO 2.0. In *IFAC RObust CONtrol Design, ROCOND09*, Haifa, Israël, June 2009.

[19] S. Gumussoy, M. Millstone, and M.L. Overton. H_∞ strong stabilization via HIFOO, a package for fixed-order controller design. In *IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

[20] S. Gumussoy and M.L. Overton. Fixed-order H_∞ controller design via HIFOO, a specialized nonsmooth optimization package. In *IEEE American Control Conference*, pages 2750–2754, Seattle, USA, June 2008.

[21] D. Henrion. Solving static output feedback problems by direct search optimization. In *Proceedings of the joint IEEE CCA/CACSD/ISIC Conference*, Munich, Germany, October 2006.

[22] D. Henrion and M. Sebek. Plane geometry and convexity of polynomial stability regions. In *Proceedings of the International Symposium on Symbolic and Algebraic Computations (ISSAC)*, Hagenberg, Austria, July 2008.

[23] H.P. Horisberger and P.R. Bélanger. Solution of the optimal constant output feedback problem by conjugate gradients. *IEEE Transactions Automatic Control*, 19(4):434–435, 1974.

[24] T. Iwasaki. The dual iteration for fixed-order control. *IEEE Trans. on Automat. Control*, 44(4):783–788, April 1999.

[25] T. Iwasaki and R.E. Skelton. The XY-centring algorithm for the dual LMI problem: a new approach to fixed-order control design. *Int. J. Control*, 62(6):1257–1272, 1995.

[26] F. Leibfritz. An LMI-based algorithm for designing suboptimal static H_2/H_∞ output feedback controllers. *SIAM Journal on Control and Optimization*, 39(6):1711 – 1735, 2001.

[27] F. Leibfritz. COMPleib: COnstraint Matrix-optimization Problem library - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, University of Trier, Department of Mathematics, 2004. URL: `www.complib.de`.

[28] F. Leibfritz and W. Lipinski. Description of the benchmark examples in compleib 1.0. Technical report, University of Trier, Department of Mathematics, 2003. URL: `www.complib.de`.

[29] F. Leibfritz and W. Lipinski. COMPleib 1.0: COnstraint Matrix-optimization Problem - user manual and quick reference. Technical report, University of Trier, Department of Mathematics, 2004. URL: `www.complib.de`.

[30] W.S. Levine and M. Athans. On the determination of the optimal constant output feedback gains fro linear multivariable systems. *IEEE Trans. on Automat. Control*, 15(1), February 1970.

[31] D. Peaucelle and D. Arzelier. An efficient numerical solution for H_2 static output feedback synthesis. In *European Control Conference*, pages 3800–3805, Porto, Portugal, September 2001.

[32] D. Peaucelle and D. Arzelier. Ellipsoidal sets for resilient and robust static output-feedback. *IEEE Trans. on Automat. Control*, 50(6):899–904, June 2005.

[33] B.T. Polyak and E.N. Gryazina. Hit-and-run: new design technique for stabilization, robustness and optimization of linear systems. In *IFAC World Congress*, Seoul, July 2008.

[34] E. Prempain. A comparison of two algorithms for the static H_∞ loop shaping problem. In *IFAC World Congress*, Prague, July 2005.

[35] E. Prempain and I. Postlethwaite. Static h-infinity loop shaping control of a fly-by-wire helicopter. *Automatica*, 41(9), September 2005.

[36] M.A. Rotea and T. Iwasaki. An alternative to the D-K iteration? In *American Control Conference*, pages 53–57, Baltimore, June 1994.

[37] R.E. Skelton, T. Iwasaki, and K. Grigoriadis. *A Unified Approach to Linear Control Design*. Taylor and Francis series in Systems and Control, 1998.

[38] V.L. Syrmos, C.T. Abdallah, P. Dorato, and K. Grigoriadis. Static output feedback: A survey. *Automatica*, 33(2):125–137, 1997.

[39] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Opimal Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.