Mixed $H_2/H_\infty$ control via nonsmooth optimization

Pierre Apkarian, Dominikus Noll and Aude Rondepierre

Abstract—We present a new approach to mixed $H_2/H_\infty$ output feedback control synthesis. Our method uses nonsmooth mathematical programming techniques to compute locally optimal $H_2/H_\infty$-controllers, which may have a predefined structure. We prove global convergence of our method and present numerical tests to validate it numerically.

Index Terms—$H_2/H_\infty$ synthesis, multi-objective control, robustness and performance, nonsmooth optimization, bundle methods.

I. INTRODUCTION

Mixed $H_2/H_\infty$ synthesis is a prominent example of a multi-objective design problem, where the feedback controller has to respond favorably to several performance specifications. Typically in $H_2/H_\infty$ synthesis, the $H_\infty$ channel is used to enhance the robustness of the design, whereas the $H_2$ channel guarantees good performance of the system.

The interest in $H_2/H_\infty$ synthesis was originally risen by three publications [9], [10], [12]. The proposed numerical methods are based on coupled Riccati equations in tandem with homotopy methods, but the numerical success of these strategies remains to be established. With the rise of LMIs in the later 1990s, different strategies which convexify the problem became increasingly popular. The price to pay for convexifying is either a conservative conservatism, or that controllers have large state dimension [13], [11].

In [21], [22], [23] Scherer developed LMI formations for $H_2/H_\infty$ synthesis for full-order controllers, and reduced the problem to solving LMIs in tandem with nonlinear algebraic equalities. In this form, $H_2/H_\infty$ problems could in principle be solved via nonlinear semidefinite programming techniques. Alas, due to the presence of Lyapunov variables, whose number grows quadratically with the system size, BMI and LMI programs quickly lead to problem sizes where existing numerical methods fail.

Following [1], [2], [3], [4], [6], we address $H_2/H_\infty$ synthesis by a new strategy which avoids the use of Lyapunov variables. This leads to a nonsmooth and semi-infinite optimization program, which we solve with a spectral bundle method, inspired by [17], [18] and [1], [3].

This paper is organized as follows: the problem setting is given in Section II. The algorithm and its rationale are presented in Section III. Global convergence is addressed in Section IV. Implementation and numerical tests are discussed in Sections V and VI.

P. Apkarian is with ONERA - 2, avenue Edouard Belin, 31055 Toulouse, France - and Université Paul Sabatier, Toulouse, France

D. Noll is with Université Paul Sabatier, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France.

A. Rondepierre is with the Institut National des Sciences Appliquées, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France.
III. Nonsmooth Algorithm

In this section, we present our main result, a nonsmooth and nonconvex optimization method for program (3).

A. Progress function and Optimality conditions

Following an idea in [19], we address program (3) by introducing the progress function (see also [6], [5]):

\[
F(y,x) = \max \left\{ f(y) - f(x) - \mu [g(x) - y^2]^+ \right\}; \quad (4)
\]

where \( \mu > 0 \) is a fixed parameter. All we need to know about \( f \) is that it is of class \( C^2 \), while \( g \) is assumed of the form

\[
g(x) = \max_{\omega \in [0,\infty]} g(x, \omega) = \max_{\omega \in [0,\infty]} \lambda_1(G(x, \omega))
\]

with \( G : \mathbb{R}^n \times [0,\infty) \rightarrow \mathbb{R}^n \) of class \( C^2 \) in \( x \in \mathbb{R}^n \), jointly continuous in \((x, \omega)\). We then prove (see [5, Lemma 5.1]):

**Lemma 1:**
1) If \( x^* \in \mathbb{R}^n \) is a local minimum of (3), then \( x^* \) is also a local minimum of \( F(\cdot, x^*) \) and: \( 0 \in \partial F(x^*, x^*) \).

2) If \( x \) satisfies the F. John necessary optimality conditions for program (3), then \( 0 \in \partial F(x, x) \).

3) Conversely, suppose \( 0 \in \partial F(x, x) \) for some \( x \in \mathbb{R}^n \).

(i) Either \( g(x) \geq y^2 \), then \( x \) is a critical point of \( g \), called a critical point of constraint violation.

(ii) Or \( g(x) \leq y^2 \), then \( x \) satisfies the F. John necessary optimality conditions for program (3) and either

(a) \( x \) is a Karush-Kuhn-Tucker (KKT) point of (3),

(b) or \( x \) fails to be a KKT-point of (3). This could only happen when \( g(x) = y^2 \) and \( 0 \in \partial g(x) \).

Lemma 1 shows why we should search for points \( x \) satisfying \( 0 \in \partial F(x, x) \). It also indicates that minimizing \( F \) leads to so-called phase I/phase II methods (see [19, section 2.6]). Namely, as long as iterates stay infeasible, the right hand term in \( F \) is dominant, so reducing \( F \) reduces constraint violation. This is the phase I. Once a feasible iterate has been found, phase I terminates successfully and iterates henceforth stay feasible. Then phase II begins and \( F \) is optimized.

B. First-order local model

In this section, we introduce a convex local model for \( F \). What we call a model of \( F \) in a neighborhood of \( x \) could be thought of as a nonsmooth analogue of the Taylor expansion. We obtain an approximation of \( F \) around \( x \) by:

\[
\phi(y,x) = \max \left\{ f'(x)(y-x) - \mu [g(x) - y^2]^+ \right\};
\]

\[
\phi_2(y,x) = \max \left\{ (G(x, \omega) + G'(x, \omega)(y-x)) \right\} \quad (5)
\]

where \( \phi_2(\cdot,x) \) is an approximation of \( g \) in a neighborhood of \( x \) obtained by linearizing the operator \( y \mapsto G(x, \omega) \) around \( x \):

\[
\phi_2(y,x) = \max_{\omega \in [0,\infty]} \lambda_1(G(x, \omega) + G'(x, \omega)(y-x)) \quad (5)
\]

The derivative \( G'(x, \omega) \) refers to the variable \( x \). Notice that \( \phi_2(\cdot,x) \) and \( \phi(\cdot,x) \) are convex models satisfying \( \phi_2(x,x) = g(x), \phi(x,x) = F(x,x) = 0 \) and \( \partial \phi(x,x) \subset \partial F(x,x) \). We can also prove that \( \phi(y,x) \) is close to \( F(y,x) \) for \( y \) close to \( x \):

\[
\text{Lemma 2 ([5, lemma 5.1]):} \quad \text{Let} \ B \subset \mathbb{R}^n \text{be a bounded set. Then there exists} \ L > 0 \text{such that for all} \ x, y \in B:\
\]

\[
|g(y) - \phi(y,x)| \leq L|y-x|^2, \quad |F(y,x) - \phi(y,x)| \leq L|y-x|^2.
\]

It is convenient to represent the local model (5) differently. Let us introduce

\[
a_0 = -\mu [g(x) - y^2]^+ \in \mathbb{R}, \quad s_0 = f'(x) \in \mathbb{R}^n,
\]

\[
\begin{align*}
\{a(\omega, Z) &= Z \cdot g(x, \omega) - y^2 - [g(x) - y^2]^+ \in \mathbb{R}, \\
\{s(\omega, Z) &= G'(x, \omega)^T Z \in \mathbb{R}^n,
\end{align*}
\]

where dependence on the point \( x \) is suppressed for convenience. Then the local model \( \phi(\cdot,x) \) may be written as the envelope of cutting planes:

\[
\phi(y,x) = \max \left\{ a + s^T (y-x) : (a,s) \in \mathcal{G} \right\} \quad (6)
\]

\[
\mathcal{G} = \{a(\omega, Z), s(\omega, Z) : \omega \in [0,\infty], Z \in \mathbb{B}_m \}
\]

Any element \( (a,s) \) in \( \mathcal{G} \) defines an affine minorant: \( y \mapsto a + s^T (y-x) \) of our model \( \phi(\cdot,x) \). The advantage of (6) over (5) is that elements \( (a,s) \) of \( \mathcal{G} \) are easier to store than elements \( (\omega,Z) \in [0,\infty] \times \mathbb{B}_m \). Also, as we shall see next, it is more convenient to construct approximations of \( \phi \).

C. Working model and tangent program

In our algorithm to be designed, we do not work directly with the convex model \( \phi \), but with an approximation \( \phi_k \), referred to as the working model, which is updated at each iteration \( k \), and which is easier to manage than \( \phi \).

Let \( x \in \mathbb{R}^n \) be the current iterate. We call \( \phi_k(\cdot,x) \) a working model of \( F(\cdot,x) \) if it is a convex function satisfying:

- \( \phi_k(x,x) = \phi(x,x) \) and \( \partial \phi_k(x,x) \subset \partial \phi(x,x) \).
- \( \phi_k(x,y) \leq \phi(x,y), \forall y \in \mathbb{R}^n \).

Using (6), it will be of the form:

\[
\phi_k(y,x) = \max \left\{ a + s^T (y-x) : (a,s) \in \mathcal{G}_k \right\} \quad (7)
\]

where \( \mathcal{G}_k \subset \mathcal{G} \). Suppose now that a working model \( \phi_k(\cdot,x) \) at counter \( k \) has been decided on. A new trial step is computed via the tangent program:

\[
\min_{y \in \mathbb{R}^n} \phi_k(y,x) + \frac{\delta_k}{2} \|y-x\|^2 \quad (8)
\]

where \( \delta_k > 0 \) is the so-called proximity control parameter, which is specified anew at each step (see section IV).

Let \( y^{k+1} \) be a local solution of (8) in the sense that:

\[
0 \in \partial \phi_k(y^{k+1}, x) + \delta_k(y^{k+1} - x) \quad (9)
\]

What happens if the solution of the program (8) is \( y^{k+1} = x \)?

**Lemma 3:** Suppose \( y^{k+1} = x \). Then \( 0 \in \partial F(x,x) \).

Indeed, if \( y^{k+1} = x \) then condition (9) becomes \( 0 \in \partial F(x,x) \), which implies \( 0 \in \partial F(x,x) \) by the property \( \partial \phi_k(x,x) \subset \partial F(x,x) \) of a working model.

From now on, assume \( 0 \notin \partial F(x,x) \). Then \( 0 \notin \partial \phi_k(x,x) \), and \( \phi_k(y^{k+1}, x) < \phi(x,x) = 0 \), so that the solution \( y^{k+1} \) of (8) is predicting a decrease of the value of the progress function (4) at \( y^{k+1} \). This gives \( y^{k+1} \) the option to improve over the current iterate \( x \) and become the new iterate \( x^+ \).
According to standard terminology, when \( y^{k+1} \) is accepted as the new iterate \( x^+ \), it is called a serious step, while trial points \( y^{k+1} \) are rejected, are called null steps.

Let \( y^{k+1} \) be a null step. How to build a new \( \phi_{k+1}(\cdot, x) \)?

The first element is to guarantee exactness:

\[
\begin{align*}
\phi_{k+1}(x, x) &= \phi(x, x) \quad \text{[} F(x, x) = 0 \text{]} \\
\partial_1 \phi_{k+1}(x, x) &\subset \partial_1 \phi(x, x) \subset \partial_1 F(x, x). \\
\end{align*}
\]

(10)

To arrange this, pick an element \( s \in \partial \phi(x, x) \subset \partial F(x, x) \) and assure that: \( \phi_{k+1}(y, x) \geq \phi(x, x) + s^T(y-x) \) for all \( y \in \mathbb{R}^n \), which is nothing else but adding active elements \((a, x) \in \mathcal{G}\) in the next set \( \mathcal{G}_{k+1} \). More precisely, depending on which branch is active in \( \phi(\cdot, x) \), we assure that:

\[
\phi_{k+1}(y, x) \geq \max \left\{ a_0 + s_0^T(y-x); \quad a_0(a_0, Z_0) + s(a_0, Z_0)^T(y-x) \right\}
\]

where \( a_0 \) is any of the active frequencies at \( x \), and where \( Z_0 := e_0 \varepsilon_0 \in \mathbb{B}_m \) is computed by choosing a normalized eigenvector \( e_0 \) associated with the maximum eigenvalue \( g(x) = A_1(G(x, a_0)) \) of \( G(x, a_0) \). In tandem with \( \phi(\cdot, x) \geq \phi_{k+1}(\cdot, x) \), we then ensure exactness:

**Lemma 4:** If \( (a_0, s_0) \in \mathcal{G}_{k+1} \) and \( (a(a_0, Z_0), s(a_0, Z_0)) \in \mathcal{G}_{k+1} \), then \( \phi_{k+1}(y, x) = \max \{ a + s^T(y-x) : (a,x) \in \mathcal{G}_{k+1} \} \) satisfies the exactness properties (10).

**Remark 1:** In practice it is useful to enrich the set \( \mathcal{G}_{k+1} \) so that it contains the subdifferential \( \partial F(x, x) \) at \( x \). This can be arranged in those cases where \( \Omega(x) \), the set of active frequencies, is finite, by adding every active elements.

Notice that the exactness plane is kept in the model \( \phi(\cdot, x) \) at all \( k \). To make \( \phi_{k+1}(\cdot, x) \) better than \( \phi_k(\cdot, x) \), we need two more elements, referred as cutting planes and aggregation.

Let us first look at the cutting plane generation. Suppose the solution \( y^{k+1} \) of the tangent program (8) based on the latest model \( \phi_k(\cdot, x) \) is a null step. Pick \( s_{k+1} = \partial \phi(y^{k+1}, x) \).

By convexity of \( \phi(\cdot, x) \):

\[
\phi(y, x) \geq \phi(y^{k+1}, x) + s_{k+1}^T(y-y^{k+1})
\]

for all \( y \in \mathbb{R}^n \). Putting \( a_{k+1} = \phi(y^{k+1}, x) + s_{k+1}^T(x-x^{k+1}) \), this means that \( y \mapsto a_{k+1} + s_{k+1}^T(y-x^{k+1}) \) is affine support function to \( \phi(\cdot, x) \) at \( y^{k+1} \). This affine function is called the cutting plane. Including the cutting plane in the new model \( \phi_{k+1}(\cdot, x) \), we then cut away the unsuccessful trial step \( y^{k+1} \):

**Lemma 5:** If we keep \( (a_{k+1}, s_{k+1}) \in \mathcal{G}_{k+1} \), then:

\[
\phi_{k+1}(y^{k+1}, x) = \phi(y^{k+1}, x) \quad \text{and} \quad s_{k+1} = \partial \phi_{k+1}(y^{k+1}, x).
\]

**Remark 2:** Suppose the right hand branch of (5) is active at \( y^{k+1} \), and let \( a_{k+1} \in [0, \infty) \) and \( Z_{k+1} \in \mathcal{G} \) be one of the pairs where the maximum (5) is attained. Then:

\[
a_{k+1} = a(a_{k+1}, Z_{k+1}) \quad \text{and} \quad s_{k+1} = s(a_{k+1}, Z_{k+1}).
\]

**Remark 3:** If the right hand branch in (5) is not active, it suffices to have \( (a_0, s_0) \in \mathcal{G}_{k+1} \), which is ensured by exactness anyway. No action on cutting planes is then required.

We need yet another process to improve the model \( \phi_{k+1} \), referred to as aggregation. The basic idea is to recycle some of the information stored in \( \phi_k \) for the new working model.

Suppose that the solution \( y^{k+1} \) of the old tangent program (8) based on \( \phi_k(\cdot, x) \) is a null step. By the optimality condition we have \( 0 \in \partial \phi(y^{k+1}, x) + \partial \phi(y^{k+1}, x) \) i.e.:

\[
s_k^{k+1} := \partial \phi(y^{k+1}, x) \in \partial \phi(y^{k+1}, x).
\]

(11)

We put: \( a_{k+1} = \phi_k(y^{k+1}, x) + s_k^{k+1} - x^{k-1} \) by construction: \( y \mapsto a_{k+1} + s_k^{k+1} \) (y-x) is an affine support function to \( \phi_k(\cdot, x) \) at \( y^{k+1} \), called the aggregate plane.

**Lemma 6:** Keeping the aggregate element \( (a_{k+1}^{k+1}, s_k^{k+1}) \) in the new \( \mathcal{G}_{k+1} \) ensures:

\[
\phi_{k+1}(y^{k+1}, x) = \phi_k(y^{k+1}, x).
\]

**D. The algorithm**

In this section we present the nonsmooth spectral bundle algorithm for program (3).

**Algorithm 1. Proximity control algorithm for (3)**

**Parameters:** \( 0 < \gamma < \bar{\gamma} < \Gamma < 1 \) (e.g. \( \gamma = 0.25 \) and \( \Gamma = 0.75 \))

1. **Initialize outer loop.** Choose initial guess \( x^1 \). Put \( j = 1 \).

2. **Stopping test.** At outer loop counter \( j \), stop at the current iterate \( x^j \) if \( 0 \in \partial \phi(x^j, x^j) \). Otherwise goto inner loop.

3. **Initialize inner loop.** Put inner counter \( k = 1 \) and \( \delta_k > 0 \). If memory element for \( \delta \) is available, use it to initialize \( \delta_k \). Choose initial convex model \( \phi_k(\cdot, x^j) \).

4. **Tangent program.** At inner loop counter \( k \), solve tangent program:

\[
\min \phi_k(y, x) + \frac{\delta_k}{2} ||y-x^j||^2,
\]

Solution is the new trial step \( y^{k+1} \).

5. **Test of progress.** Check whether:

\[
\rho_k = F(x^j, x^j) - F(y^{k+1}, x^j) - \phi_k(y^{k+1}, x^j) \geq \gamma
\]

If this is the case, put \( x^{k+1} = y^{k+1} \) (serious step), quit inner loop and goto step 8. On the other hand, if this is not the case (null step) continue inner loop with step 6.

6. **Update proximity control.** Compute control parameter

\[
\tilde{\rho} = \frac{F(x^j, x^j) - \phi_k(y^{k+1}, x^j)}{F(x^j, x^j) - \phi_k(y^{k+1}, x^j)}
\]

Then put:

\[
\delta_{k+1} = \begin{cases} \delta_k, & \text{if } \tilde{\rho} < \bar{\gamma} \\ 2\delta_k, & \text{if } \tilde{\rho} \geq \bar{\gamma} \end{cases}
\]

7. **Update working model.** Build new convex working model \( \phi_{k+1}(\cdot, x^j) \) by respecting the three rules (exactness, cutting plane, aggregation) based on null step \( y^{k+1} \).

Then increase inner counter \( k \) and go back to step 4.

8. **Update memory element.** Compute new memory element \( \delta^+ \) as:

\[
\delta^+ = \begin{cases} \frac{\delta_k}{2}, & \text{if } \rho_k > \Gamma \\ \delta_k, & \text{otherwise} \end{cases}
\]

Increase outer loop counter \( j \), and go back to step 2.
IV. MANAGEMENT OF THE PROXIMITY PARAMETER

How to decide whether the trial step $x_{k+1}$ is accepted or not? We introduce two constants $0 < \gamma < \Gamma < 1$ and compute at step 5 the parameter $\rho_k$, which compares the current model $\phi_k(\cdot, x)$ to the truth $F(\cdot, x^*)$ at $x_{k+1}$. Good agreement would give $\rho_k \approx 1$, but we accept $x_{k+1} = x^*$ already when $\rho_k \geq \gamma$. The trial step $x_{k+1}$ is called good if $\rho_k \geq \Gamma$, not bad if $\gamma \leq \rho_k < \Gamma$, and bad if $\rho_k < \gamma$. Our strategy is to accept the trial step if $\rho_k \geq \gamma$; i.e. if the step is not bad, and $x^* = x_{k+1}$ becomes the next iterate. In case of a good step, the model $\phi_k$ seems reliable, so we can relax proximity control in the next sweep as done in step 8.

What should we do when $\rho_k < \gamma$? The trial step $x_{k+1}$ is then rejected, and $\phi_k(\cdot, x)$ has to be improved. We introduce a new constant $\gamma < \widetilde{\gamma} < 1$ and a second parameter $\tilde{\rho}_k$, which now compares the current model $\phi_k(\cdot, x)$ to the model $\phi(\cdot, x)$ at $x_{k+1}$. If $\tilde{\rho}_k < \widetilde{\gamma}$, the model $\phi_k(\cdot, x)$ is too far from $\phi(\cdot, x)$; we then keep $\delta_k = \delta_k$ unchanged, being reluctant to increase the $\delta$-parameter prematurely, and continue to rely on cutting planes and aggregation, hoping that this will drive $\phi_k$ closer to $\phi$. Otherwise, if $\tilde{\rho}_k \geq \widetilde{\gamma}$, we consider that driving $\phi_k$ closer to $\phi$ alone will not be sufficient to make progress, simply because $\phi$ itself is too far from the true $F$. We then decide to tighten proximity control, by increasing $\delta_k^{k+1} = 2\delta_k$.

Remark 4: Notice that the parameters $\rho_k$ and $\tilde{\rho}_k$ in steps 5 and 6 are well defined because we only enter the inner loop when $0 \notin \partial F(x, x)$, in which case: $\phi_k(x_{k+1}, x) < \phi_k(x, x) = 0$.

We then can prove global convergence of our algorithm: 

**Theorem 1**: Assume:

- $f$ is weakly coercive on the level set $\{x \in \mathbb{R}^n : g(x) \leq \gamma^2\}$
  - i.e. if $x^i$ is a sequence of feasible iterates with $\|x^i\| \to \infty$, then $f(x^i)$ is not strictly monotonically decreasing.
- $g$ is weakly coercive.

Then every accumulation point $x$ of the sequence of serious steps $x^i$ generated by the algorithm satisfies $0 \in \partial F(x, x)$. A complete proof of Theorem 1 is detailed in [5].

V. IMPLEMENTATION

We implement algorithm 1 for structured mixed synthesis, and we use the enriched working models $\phi_k$ to speed up convergence. In each example, the algorithm is initialized by a closed-loop stabilizing $K^0$, which is computed using techniques described in [7].

A. Stopping criteria

We use the following numerical stopping criteria: we first check criticality $0 \in \partial F(x, x)$ by checking

$$\inf\{\|h\| : h \in \partial F(x, x)\} < \varepsilon_1.$$ 

Two additional tests are implemented to avoid pointless computational efforts during the final phase, where iterates make minor progress: we compare the progress of the local model around the current iterate and we evaluate the relative step length to the controller gains:

$$|F(x^+, x)| \leq \varepsilon_2 \quad \text{and} \quad |x^+ - x| \leq \varepsilon_3(1 + |x|). \quad (12)$$

For stopping, we require that either the first, or the second and third be satisfied.

B. Choice of the performance level $\gamma$

In all test examples we first compute (locally) optimal $H_2$ and $H_\infty$ controllers $K_2$ and $K_\infty$. It is now obvious that the performance level $\gamma$ in program (3) has to satisfy

$$||T_{\infty}(K_\infty)|| \leq \gamma \leq ||T_{\infty}(K_2)||.$$ \quad (13)

Indeed, the mixed $H_2/H_\infty$ problem (3) is infeasible for $\gamma < ||T_{\infty}(K_\infty)||$, while for $\gamma \geq ||T_{\infty}(K_2)||$ the optimal $H_2$ controller $K_2$ is also optimal for (3).

Table I reports the problem dimensions $n_x, n_y, n_u$, the synthesized controller orders $n_K$, $||T_{\infty}(K_\infty)||$, and $||T_{\infty}(K_2)||$, which are the bounds in (13) and $||T_{\infty}(K_2)||_2$ which is a lower bound on the optimal value $||T_{\infty}(K)||_2$ of (3).

### VI. NUMERICAL EXPERIMENTS

In this section we test our nonsmooth algorithm on a variety of $H_2/H_\infty$ synthesis problems from the literature. We especially study the benefit of the penalty term $\mu\|g(x) - \gamma^2\|$ in our progress function and the choice of the parameter $\Gamma$.

A. Four disks

The four disks model is originally described in [8] and has previously been studied to evaluate reduced-order design methods. The open loop plant is of order $n_x = 8$ and has two stable poles. We first focus on mixed $H_2/H_\infty$ synthesis of full order controllers in order to compare our nonsmooth algorithm to the original Riccati equation approach in [12]. Results are presented in Table II for full-order synthesis, and in Table III for reduced-order synthesis.

As shown in Table II, our method significantly improves the older results in [12] based on coupled Riccati equations. This highlights the reduction of conservatism of our approach compared to Riccati and LMI methods.

B. Choice of the penalty parameter $\mu$

In [20] Sagastizabal and Solodov use a different progress function, referred to as an improvement function, which does not feature the penalty term $\mu\|g(x) - \gamma^2\|$. Since this term equals 0 in phase II, both criteria lead to the same steps in phase II, and differences could only occur in phase I.
To observe the impact of the choice of $\mu$, we test the algorithm on the four disks problem for four different values of $\mu$, including the case $\mu = 0$ to compare with the improvement function of [20]. Results are given on Figure 1 and Table IV.

As shown in Table IV, using the improvement function proposed in [20], the algorithm fails to reach a feasible point. This could be explain by the fact that with $\mu = 0$, every step has to be a descent step for both the objective $f$ and the constraint $g$, and the algorithm gets then trapped as soon as it reaches a local minimum of either $f$ or $g$. In our approach ($\mu > 0$), when reducing constraint violation, a slight increase in $f$ not exceeding $\mu [g(x) - \gamma^2]_+$ is granted. This helps the algorithm in not being trapped at infeasible local minima of $f$ alone. Naturally, the difficulty of local minima of $g$ alone remains with both criteria.

Among the choices $\mu > 0$ we noticed that when $\mu$ is not too small, the number of iterations to reach a feasible point decreases as $\mu$ increases. However choosing $\mu$ too large as shown in Table IV, does not give the best results either. Nothing decisive can be proposed to date, but we notice that choosing $\mu$ of the same order of magnitude as $F(y^{k+1}, x)$ with $\mu = 0$, gave so far the best results in practice.

C. Choice of $\Gamma$

The last issue we address is the choice of $\Gamma$, which is crucial because step 5 is the only place in the algorithm where the proximity parameter $\delta_k$ can be reduced. Too large a $\Gamma$ gives few reductions of $\delta_k$, and since the latter is often increased during the inner loop, this bears the risk of exceedingly large $\delta_k$, causing the algorithm to stop.

To illustrate this, we have run the four disks example for three different values $\Gamma \in \{0.4, 0.6, 0.8\}$. The results are illustrated in Figure 2. We observe that the number of iterations increases with the values of $\Gamma$. The best numerical results were obtained for $\Gamma = 0.6$, and this is the value we retained for all the numerical tests. At least over a certain range one can say that the larger $\Gamma$, the smaller the steps accepted as serious steps $x \rightarrow x^+$, and the more outer iterations are needed to reach the same $H_2$ performance.

D. COMPl_sib examples

The models are from the COMPl_sib collection [15]: distillation tower 'BDT2', heat flow in a thin rod 'HF1' and cable mass model 'CM4'. They are originally designed for $H_\infty$ synthesis, so an $H_2$ channel is added as suggested by F. Leibfritz [14], [15] by choosing $B_2 = B_\infty$ and $D_2 = 0$.

In each example, we first choose the $H_\infty$ performance level $\gamma$ larger than $\|T_\infty(K_2)\|_\infty$. In doing this we have to obtain an estimate of the optimal $H_2$ performance $\|T_2(K_2)\|_2$ given in Table I. Numerical results are in Tables V and VI.

VII. CONCLUSION

$H_2/H_\infty$ controller synthesis is a practically important problem for which successful numerical methods are lacking. We propose a strategy based on local optimization, which comes
with a local optimality certificate, but has the benefit to work in practice. The problem being nonconvex, nonsmooth and semi-infinite, we develop nonsmooth programming techniques suited for the $H_2/H_\infty$ problem and other programs of a similar structure. The current work expands on our previous results in [6], proposing a different algorithm based on bundle techniques, and improving again existing results.

REFERENCES


![Graph](image)

Fig. 2. Full-order $H_2/H_\infty$ synthesis for the four disks pb: the $H_2$ norm vs the number of serious steps for three different values of $\Gamma \in \{0.4, 0.6, 0.8\}$.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Four Disks [27]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(n_1, n_2, n_3)$</td>
<td>(8, 1, 1)</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.6</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.5</td>
</tr>
<tr>
<td>serious steps</td>
<td>0.4</td>
</tr>
<tr>
<td>$|T_2(K)^{(1)}|_2$</td>
<td>0.2062</td>
</tr>
<tr>
<td>$|T_\infty(K)^{(1)}|_\infty$</td>
<td>0.6000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pb</th>
<th>$(n_1, n_2, n_3)$</th>
<th>$n_\kappa$</th>
<th>$\gamma$</th>
<th>Serious steps</th>
<th>$|T_2(K)|_2$</th>
<th>$|T_\infty(K)|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDT2</td>
<td>(82, 4, 4)</td>
<td>10</td>
<td>10</td>
<td>148</td>
<td>8.0402e-01</td>
<td>1.058</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8</td>
<td>324</td>
<td>7.6480e-01</td>
<td>1.144</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8</td>
<td>404</td>
<td>7.7146e-01</td>
<td>0.800</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41</td>
<td>0.8</td>
<td>115</td>
<td>7.8882e-01</td>
<td>0.800</td>
</tr>
<tr>
<td>HF1</td>
<td>(130, 1, 2)</td>
<td>0</td>
<td>10</td>
<td>7</td>
<td>5.8193e-02</td>
<td>4.696e-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.45</td>
<td>16</td>
<td>5.8795e-02</td>
<td>4.399e-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.45</td>
<td>33</td>
<td>5.8706e-02</td>
<td>4.500e-01</td>
</tr>
<tr>
<td>CM4</td>
<td>(240, 1, 2)</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>9.2645e-01</td>
<td>1.655</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>1</td>
<td>15</td>
<td>9.8438e-01</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>1</td>
<td>41</td>
<td>9.4035e-01</td>
<td>1.000</td>
</tr>
</tbody>
</table>

TABLE V

RESULTS OF MIXED $H_2/H_\infty$ SYNTHESIS FOR TEST EXAMPLES FROM COMPLi/cb - CRITICALITY IS SATISFIED FOR EACH TEST

<table>
<thead>
<tr>
<th>Problem</th>
<th>$\gamma$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDT2</td>
<td>10</td>
<td>$-0.6186$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.6357$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.07527$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.6731$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.9207$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.7452$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.7119$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0887$</td>
</tr>
<tr>
<td>HF1</td>
<td>10</td>
<td>$-0.1002$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.2521$</td>
</tr>
<tr>
<td>CM4</td>
<td>10</td>
<td>$-0.5448$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.5146$</td>
</tr>
</tbody>
</table>

TABLE VI

STATIC $H_2/H_\infty$ CONTROLLERS FOR EXAMPLES FROM COMPLi/cb