

Branch and bound algorithm for the robustness analysis of uncertain systems [★]

L. Ravanbod^{*} D. Noll^{*} P. Apkarian^{**}

^{*} *Université de Toulouse, Institut de Mathématiques, Toulouse, France*
lalehravanbod@yahoo.fr, dominikus.noll@math.univ-toulouse.fr

^{**} *ONERA, Toulouse, France Pierre.Apkarian@onera.fr*

Abstract: Computing the worst-case spectral abscissa of a system with uncertain parameters allows to decide whether it is robustly stable in a given parameter range. Since this problem is NP-hard, we use a heuristic local optimization method based on a bundle trust-region strategy to compute good lower bounds. Then we employ branch-and-bound to certify the global maximum. A specific frequency sweeping technique is used to accelerate the global optimization.

Keywords: Spectral abscissa · minimum stability degree · frequency sweep · bundle trust-region

1. INTRODUCTION

We consider a parameter-dependent linear time-invariant system

$$\begin{aligned} \dot{x} &= Ax + Bp \\ q &= Cx + Dp \\ p &= \Delta q \end{aligned} \quad (1)$$

with $x(t) \in \mathbb{R}^n$, $p(t), q(t) \in \mathbb{R}^r$, where A, B, C, D are real matrices of appropriate sizes, and where Δ is an $r \times r$ diagonal matrix of the form

$$\Delta = \text{diag}[\delta_1 I_{r_1}, \dots, \delta_m I_{r_m}] \quad (2)$$

with I_{r_i} an identity matrix of size r_i and $r = r_1 + \dots + r_m$. Assuming that the matrix A is stable, we ask whether the system (1) remains stable for all choices $\delta \in [-1, 1]^m$ of the uncertain real parameters. If we consider the matrices Δ in (2) in one-to-one correspondence with $\delta \in \mathbb{R}^m$, then this amounts to checking whether

$$\dot{x} = (A + B\Delta(I - D\Delta)^{-1}C)x \quad (3)$$

is stable for every $\delta \in [-1, 1]^m$. We assume throughout that $I - D\Delta$ is invertible for every $\delta \in [-1, 1]^m$, i.e., that (1) is robustly well-posed over $[-1, 1]^m$. All rational parameter variations in a nominal system $\dot{x} = Ax$ can be represented via a suitable LFT of the form (1).

Recall that the spectral abscissa of a square matrix A is $\alpha(A) = \max\{\text{Re}(\lambda) : \lambda \text{ eigenvalue of } A\}$, and that stability of A is equivalent to $\alpha(A) < 0$. The problem of robust stability of (1) over $\delta \in [-1, 1]^m$ can therefore be addressed by the optimization program

$$\alpha^* = \max_{\delta \in [-1, 1]^m} \alpha(A(\delta)), \quad (4)$$

where $A(\delta) = A + B\Delta(I - D\Delta)^{-1}C$. As soon as the global optimum satisfies $\alpha^* < 0$, the system (1) is certified robustly stable over $\delta \in [-1, 1]^m$, while a solution δ^* of (4) with $\alpha^* \geq 0$ gives a destabilizing choice of the uncertain parameters, which may represent valuable information for parametric robust synthesis, see Apkarian et al. (2014).

[★] The authors acknowledge financial support by Fondation d'Entreprise EADS (F-EADS) and Fondation de Recherche pour l'Aéronautique et l'Espace (FNRAE).

When $\alpha^* < 0$ every solution $x(t)$ of (3) decays at least as fast as $e^{\alpha^* t}$, in which case $-\alpha^* > 0$ is also known as the *minimum stability degree* of (1).

2. BRANCH AND BOUND STRATEGY

In this section we present the main ingredients of our branch and bound algorithm for (4). The differences with earlier work by Gaston et al. (1988), Sideris et al. (1989), and Balakrishnan et al. (1991) are (i) the use of a sophisticated local solver which gives an improved lower bound, (ii) an evaluation procedure which avoids computing explicit upper bounds, and (iii) a new element which integrates frequency information in the setup. We will explain these improvements as we go.

2.1 Basic setup

For every subbox $\Delta = \prod_{i=1}^m [a_i, b_i]$ of $[-1, 1]^m$ with $-1 \leq a_i < b_i \leq 1$ we consider the subproblem

$$\alpha^*(\Delta) = \max_{\delta \in \Delta} \alpha(A(\delta)) \quad (5)$$

of (4) associated with Δ . During the algorithm we maintain a finite list \mathcal{L} of subproblems specified by pairwise non-overlapping subboxes, called the *list of doables*. The algorithm stops as soon as the list \mathcal{L} has been worked off. The list is initialized with the box $[-1, 1]^m$. When a box $\Delta \in \mathcal{L}$ comes up for evaluation, we call a decision procedure \mathcal{P} , called a *pruning test*, which decides whether or not Δ can be pruned. When pruned, Δ simply disappears from the list \mathcal{L} . When \mathcal{P} decides that pruning is not possible, then Δ is divided into two successor boxes Δ' , Δ'' of half volume, Δ is removed from the list and Δ' , Δ'' are added, so that \mathcal{L} grows by one. Usually we cut the box in two halves along a longest edge.

2.2 Lower bound

We use a local optimization method based on a bundle trust-region strategy Apkarian et al. (2015a,b) to compute a lower bound $\underline{\alpha} \leq \alpha^*$ of the global optimum. Suppose the

local optimum is attained at $\underline{\delta} \in [-1, 1]^m$, then $\underline{\delta}$ is our candidate for the solution, called the *incumbent*. Since the local solver is fast, it is re-started within Δ whenever a new subproblem $\Delta \in \mathcal{L}$ is evaluated. This may lead to an improved lower bound and incumbent. The information provided within Δ is also used to rank the boxes in the list of doables \mathcal{L} . A detailed description of the local solver can be found in Apkarian et al. (2015a,b). For the current analysis it is enough to know that when the algorithm is started with initial guess $\delta_0 \in \Delta$, then it always ends with a local maximum $\underline{\delta} \in \Delta$ satisfying $\alpha(A(\underline{\delta})) \geq \alpha(A(\delta_0))$.

2.3 Pruning test

Following standard terminology, a function $\bar{\alpha}(\cdot)$, defined on boxes Δ , is called an upper bound if $\alpha^*(\Delta) \leq \bar{\alpha}(\Delta)$. Given the current lower bound $\underline{\alpha}$ and a tolerance $\epsilon \geq 0$, the standard pruning test with upper bound $\bar{\alpha}(\cdot)$ is

$$\mathcal{P}_{\text{ub}} : \begin{cases} \text{if } \bar{\alpha}(\Delta) \leq \underline{\alpha} + \epsilon \text{ then } & \text{pruning } \Delta \\ \text{otherwise} & \text{not_pruning} \end{cases} \quad (6)$$

The idea is that the decision **pruning** is only issued when $\alpha^*(\Delta) \leq \bar{\alpha}(\Delta) \leq \underline{\alpha} + \epsilon$, in which case the present incumbent cannot be further improved within the tolerance ϵ by investigating subboxes of Δ . Hence Δ can be eliminated.

It turns out that in order to reach the decision (6) it is not necessary to compute an upper bound. Any method which allows to certify that $\alpha^*(\Delta) \leq \underline{\alpha} + \epsilon$ will be sufficient to reach the same decision. This is captured by the following

Definition 1. A decision procedure \mathcal{P} which, given a box Δ and a reference value $\underline{\alpha}$ on entry, and being allowed a tolerance $\epsilon \geq 0$, issues a decision between **pruning** Δ and **not_pruning** is called a *pruning test* if the decision **pruning** Δ is only issued when it is certified that $\alpha^*(\Delta) \leq \underline{\alpha} + \epsilon$. \square

We shall use the shorthand $\mathcal{P}(\Delta, \underline{\alpha}, \epsilon) = \text{pruning}$, respectively, $\mathcal{P}(\Delta, \underline{\alpha}, \epsilon) = \text{not_pruning}$. In order to succeed, a pruning test \mathcal{P} has to satisfy the following property:

Definition 2. A pruning test \mathcal{P} is *consistent* if for every $\epsilon > 0$ there exists $\eta > 0$ such that for every box Δ with diameter $< \eta$ and for every $\alpha \geq 0$ the decision $\mathcal{P}(\Delta, \alpha^*(\Delta) + \alpha, \epsilon) = \text{pruning}$ is made. \square

The explanation is that sufficiently small boxes will get pruned when the global lower bound $\underline{\alpha}$ is better than their value $\alpha^*(\Delta)$ within the allowed level of tolerance ϵ . For the classical pruning test (6) consistency amounts to requiring

$$\lim_{\Delta \rightarrow 0} \alpha^*(\Delta) - \bar{\alpha}(\Delta) = 0.$$

In section 3 we shall present several consistent pruning tests, which do not require computing an upper bound $\bar{\alpha}(\cdot)$. This leads to an advantage in speed.

2.4 Presentation of the algorithm

In this section we present the algorithm by way of the pseudo-code given below. The principal property of the algorithm can be summarized by the following

Theorem 1. Suppose algorithm 1 is operated with a consistent pruning test \mathcal{P} and tolerance level $\epsilon > 0$. Then it terminates with an empty list \mathcal{L} after a finite number of steps, and on exit the returned lower bound $\underline{\alpha}$ satisfies

Algorithm 1. Branch and bound for program (4).

```

1: Lower bound. Call local solver to compute lower
   bound  $\underline{\alpha}$ . Initialize list  $\mathcal{L} = \{[-1, 1]^m\}$ .
2: while  $\mathcal{L} \neq \emptyset$  do
3:   Choose first element  $\Delta \in \mathcal{L}$  for evaluation
4:   Call local solver in  $\Delta$  to update lower bound  $\underline{\alpha}$ .
5:   Call pruning test  $\mathcal{P}$ .
6:   if  $\mathcal{P}(\Delta, \underline{\alpha}, \epsilon) = \text{pruning}$  then
7:     Remove  $\Delta$  from  $\mathcal{L}$ 
8:   else
9:     Remove  $\Delta$  and replace it by two successors
10:     $\Delta', \Delta''$  in  $\mathcal{L}$ 
11:   end if
12:   Update ordering of  $\mathcal{L}$ 
13: end while
14: Return  $\underline{\delta}$  and  $\underline{\alpha}$ .
```

$\alpha^* \leq \underline{\alpha} + \epsilon$. In particular, if $\underline{\alpha} + \epsilon < 0$, then a robust stability certificate for (1) is obtained.

Proof. 1) Let $\underline{\alpha}^{(n)}$ be the best lower bound found after iteration n . Then $\underline{\alpha}^{(n)} \leq \underline{\alpha}^{(n+1)} \rightarrow \underline{\alpha} \leq \alpha^*$. Now suppose first the algorithm ends finitely at iterate n , then at some stage $k \leq n$ a box Δ containing a global maximum δ^* has been pruned. This box satisfies $\alpha^*(\Delta) = \alpha^*$, and since the pruning test was based on $\underline{\alpha}^{(k)}$, we have $\alpha^* = \alpha^*(\Delta) \leq \underline{\alpha}^{(k)} + \epsilon \leq \underline{\alpha} + \epsilon$. That gives the estimate claimed in the statement.

2) It obviously suffices to show that there exists $\eta > 0$ and an iteration counter n_0 such that for all counters $n \geq n_0$ boxes with $\text{diam}(\Delta) < \eta$ are automatically pruned when evaluated.

3) Since α is a continuous function, $\delta \mapsto \alpha(A(\delta))$ is uniformly continuous on $[-1, 1]^m$ by the hypothesis of robust well-posedness, hence there exists $\eta > 0$ such that for all boxes with $\text{diam}(\Delta) < \eta$ and all $\delta, \delta' \in \Delta$ we have $|\alpha(A(\delta)) - \alpha(A(\delta'))| < \frac{1}{2}\epsilon$. That means as soon as a box with $\text{diam}(\Delta) < \eta$ is evaluated, the local optimizer finds a value $\underline{\alpha}(\Delta)$ such that $|\alpha^*(\Delta) - \underline{\alpha}(\Delta)| < \frac{1}{2}\epsilon$. If this evaluation occurs at iteration n , then $\alpha^*(\Delta) \leq \underline{\alpha}(\Delta) + \frac{1}{2}\epsilon \leq \underline{\alpha}^{(n)} + \frac{1}{2}\epsilon \leq \underline{\alpha} + \frac{1}{2}\epsilon$, because the lower bound is regularly updated.

4) Using consistency of \mathcal{P} , by reducing η found in 3), we can further assume that $\mathcal{P}(\Delta, \alpha^*(\Delta) + \alpha, \frac{1}{2}\epsilon) = \text{pruning}$ for every $\alpha \geq 0$ and every box with $\text{diam}(\Delta) < \eta$.

5) Now assume the algorithm does not terminate. Then there exist boxes Δ_k of diameter $\leq \eta_k \rightarrow 0$, $\eta_k < \eta$, which are evaluated at counter n_k , but not pruned. Then $\alpha^*(\Delta_k) \leq \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon$, hence $\mathcal{P}(\Delta_k, \underline{\alpha}^{(n_k)}, \epsilon) = \mathcal{P}(\Delta_k, \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon, \frac{1}{2}\epsilon) = \mathcal{P}(\Delta_k, \alpha^*(\Delta_k) + a_k, \frac{1}{2}\epsilon) = \text{pruning}$ by consistency, where $a_k = \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon - \alpha^*(\Delta_k) \geq 0$. This contradicts the assumption that Δ_k was not pruned and completes the proof. \square

3. CENTRALIZING LOOP TRANSFORMATION

In order to prepare our pruning tests we follow Balakrishnan et al. (1991) and apply a loop transformation to the system (A, B, C, D) with uncertainty $\delta \in \Delta$ such that the transformed system $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ has its uncer-

tainty $\tilde{\delta}$ with the same structure (2) in $[-1, 1]^m$. Assuming $\mathbf{\Delta} = \prod_{i=1}^m [a_i, b_i]$, and putting

$$\begin{aligned} K &= \frac{1}{2} \text{diag} [(a_1 + b_1)I_{r_1}, \dots, (a_m + b_m)I_{r_m}], \\ F &= \frac{1}{2} \text{diag} [(b_1 - a_1)I_{r_1}, \dots, (b_m - a_m)I_{r_m}], \end{aligned} \quad (7)$$

we define

$$\begin{aligned} \tilde{A} &= A + B(I - KD)^{-1}KC, \quad \tilde{B} = B(I - KD)^{-1}F^{1/2}, \\ \tilde{C} &= F^{1/2}(I - DK)^{-1}C, \quad \tilde{D} = F^{1/2}D(I - KD)^{-1}F^{1/2}. \end{aligned} \quad (8)$$

This is indeed what is required because we have the following

Lemma 2. (Balakrishnan et al. (1991)). Let $\mathbf{\Delta} = \prod_{i=1}^m [a_i, b_i]$ and $\underline{\alpha} \in \mathbb{R}$, then the following are equivalent:

- i. $\alpha(A + B\Delta(I - D\Delta)^{-1}C) < \underline{\alpha}$ for every $\delta \in \mathbf{\Delta}$;
- ii. $\alpha(\tilde{A} + \tilde{B}\tilde{\Delta}(I - \tilde{D}\tilde{\Delta})^{-1}\tilde{C}) < \underline{\alpha}$ for every $\tilde{\delta} \in [-1, 1]^m$.

Here $\tilde{\Delta} \leftrightarrow \tilde{\delta}$, $\Delta \leftrightarrow \delta$ via (2). Moreover, the uncertainties $\Delta, \tilde{\Delta}$ are in one-to-one correspondence via

$$\tilde{\Delta} = F^{-1/2}(\Delta - K)F^{-1/2}, \quad \Delta = K + F^{1/2}\tilde{\Delta}F^{1/2}. \quad (9)$$

□

4. PRUNING VIA μ -UPPER BOUNDS

Due to the centralization in Lemma 2 we can now use overestimates of the structured singular value $\mu_{\mathbf{\Delta}}$ of Safonov (1980) and Doyle (1982) to define pruning tests. We introduce the notation $M_{\alpha}(s) = C(sI - (A - \alpha I))^{-1}B + D$, and similarly $\tilde{M}_{\alpha}(s) = \tilde{C}(sI - (\tilde{A} - \alpha I))^{-1}\tilde{B} + \tilde{C}$ for the shifted system (7), (8). If $\tilde{A} - \alpha I$ is stable, and since (1) is robustly well-posed over $[-1, 1]^m$, we can write the structured singular value as

$$\mu_{\mathbf{\Delta}}(\tilde{M}_{\alpha}) = \sup \left\{ \bar{\sigma}(\tilde{\Delta})^{-1} : \tilde{A} - \alpha I + \tilde{B}\tilde{\Delta}(I - \tilde{D}\tilde{\Delta})^{-1}\tilde{C} \text{ unstable, } \tilde{\Delta} \leftrightarrow \tilde{\delta} \text{ via (2)} \right\}. \quad (10)$$

Replacing uncertainties $\tilde{\Delta}$ as in (2) by unstructured perturbations $\tilde{\Xi}$ of the same size leads to a first rather conservative μ -upper bound $\tilde{\mu}_{\mathbf{\Delta}}(M) = \sup\{\bar{\sigma}(\tilde{\Xi})^{-1} : |I - M(s)\tilde{\Xi}| = 0\}$, which we can express as

$$\begin{aligned} \tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha}) &= \\ \sup \left\{ \bar{\sigma}(\tilde{\Xi})^{-1} : \tilde{A} - \alpha I + \tilde{B}\tilde{\Xi}(I - \tilde{D}\tilde{\Xi})^{-1}\tilde{C} \text{ unstable} \right\}, \end{aligned} \quad (11)$$

and which has the advantage that it can be evaluated fast as $\tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha}) = \|(\tilde{A} - \alpha I, \tilde{B}, \tilde{C}, \tilde{D})\|_{\infty}$, where $\|\cdot\|_{\infty}$ is the H_{∞} -norm. This leads to the following pruning test.

$$\mathcal{P}_{\tilde{\mu}} : \begin{cases} \text{if } \tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha+\epsilon}) < 1 \text{ then} & \text{pruning } \mathbf{\Delta} \\ \text{otherwise} & \text{not_pruning} \end{cases} \quad (12)$$

We consider the less conservative μ -upper bound of Fan et al. (1991),

$$\begin{aligned} \hat{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha}) &= \inf_{G\Delta = \Delta G, D\Delta = \Delta D} \sup_{\omega} \\ \inf \left\{ \beta \geq 0 : \tilde{M}_{\alpha}^H D \tilde{M}_{\alpha} + j(G\tilde{M}_{\alpha} - \tilde{M}_{\alpha}^H G) - \beta^2 D \preceq 0 \right\}. \end{aligned}$$

This bound satisfies $\mu_{\mathbf{\Delta}} \leq \hat{\mu}_{\mathbf{\Delta}} \leq \tilde{\mu}_{\mathbf{\Delta}}$, and the corresponding pruning test is

$$\mathcal{P}_{\hat{\mu}} : \begin{cases} \text{if } \hat{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha+\epsilon}) < 1 \text{ then} & \text{pruning } \mathbf{\Delta} \\ \text{otherwise} & \text{not_pruning} \end{cases} \quad (13)$$

Proposition 3. Suppose (1) is nominally stable over $[-1, 1]^m$. Then $\mathcal{P}_{\tilde{\mu}}$ and $\mathcal{P}_{\hat{\mu}}$ are consistent pruning test.

Proof. 1) In view of $\mu_{\mathbf{\Delta}} \leq \hat{\mu}_{\mathbf{\Delta}} \leq \tilde{\mu}_{\mathbf{\Delta}}$ the decision **pruning** is only issued when $\mu_{\mathbf{\Delta}}(\tilde{M}_{\alpha+\epsilon}) < 1$. That means $\tilde{M}_{\alpha+\epsilon}$ is robustly stable over $[-1, 1]^m$, hence $\alpha(\tilde{A} - (\alpha + \epsilon)I + \tilde{B}\tilde{\Delta}(I - \tilde{D}\tilde{\Delta})^{-1}\tilde{C}) < 0$ for every $\tilde{\delta} \in [-1, 1]^m$, hence $\alpha(\tilde{A} + \tilde{B}\tilde{\Delta}(I - \tilde{D}\tilde{\Delta})^{-1}\tilde{C}) < \alpha + \epsilon$ for every $\tilde{\delta} \in [-1, 1]^m$, hence by Lemma 2 $\alpha(A + B\Delta(I - D\Delta)^{-1}C) < \alpha + \epsilon$ for every $\delta \in \mathbf{\Delta}$, hence $\alpha^*(\mathbf{\Delta}) \leq \alpha + \epsilon$. Hence $\mathcal{P}_{\tilde{\mu}}$ is a pruning test, and so is $\mathcal{P}_{\hat{\mu}}$.

2) It suffices to prove consistency of $\mathcal{P}_{\tilde{\mu}}$. We have to find $\eta > 0$ such that when $\text{diam}(\mathbf{\Delta}) < \eta$, $\alpha \geq 0$, then $\mathcal{P}_{\tilde{\mu}}(\mathbf{\Delta}, \alpha^*, \epsilon) = \text{pruning}$, where $\alpha^* := \alpha^*(\mathbf{\Delta}) + \alpha$. By (12) the latter means $\tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha^*+\epsilon}) < 1$, where $\tilde{M}_{\alpha} = (\tilde{A} - \alpha I, \tilde{B}, \tilde{C}, \tilde{D})$ is the shifted system (7), (8) for $\mathbf{\Delta}$ and α . For the proof we may assume $\alpha^* = \alpha^*(\mathbf{\Delta})$. Now the structured singular value of $\tilde{M}_{\alpha^*+\epsilon}$ may be expressed as (10), while the upper bound $\tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha^*+\epsilon})$ may be expressed as (11). Suppose the statement is incorrect, then there exist boxes $\mathbf{\Delta}$ of arbitrarily small diameter such that $\tilde{\mu}_{\mathbf{\Delta}}(\tilde{M}_{\alpha^*+\epsilon}) > 1$. Pick a complex $\tilde{\Xi}$ such that $1/\bar{\sigma}(\tilde{\Xi}) > 1$ and $\tilde{A} - (\alpha^* + \epsilon)I + \tilde{B}\tilde{\Xi}(I - \tilde{D}\tilde{\Xi})^{-1}\tilde{C}$ unstable. Applying the inverse loop transformation (9) shows that there exists a complex Ξ such that $A - (\alpha^* + \epsilon)I + B\Xi(I - D\Xi)^{-1}C$ is unstable and $\Xi = K + F^{1/2}\tilde{\Xi}F^{1/2}$. Then $\mathcal{A} := A - (\alpha^* + \epsilon)I + B\Xi(I - D\Xi)^{-1}C$ is also unstable. Now pick $\tilde{\delta} \in [-1, 1]^m$ such that $\alpha(\tilde{A} + \tilde{B}\tilde{\Delta}(I - \tilde{D}\tilde{\Delta})^{-1}\tilde{C}) = \alpha^* = \alpha^*(\mathbf{\Delta})$, and let $\Delta = K + F^{1/2}\tilde{\Delta}F^{1/2}$ be its inverse transform. Then $\delta \in \mathbf{\Delta}$ by construction, and $\alpha(A + B\Delta(I - D\Delta)^{-1}C) = \alpha^*$. Now we may decompose the unstable matrix \mathcal{A} as

$$\mathcal{A} = \mathcal{A}(\Delta) + \mathcal{B}(\Xi, \Delta)$$

where $\mathcal{A}(\Delta) := [A + B\Delta(I - D\Delta)^{-1}C - \alpha^*I - \epsilon I]$ satisfies $\alpha(\mathcal{A}(\Delta)) = -\epsilon$, and $\mathcal{B}(\Xi, \Delta) = B\Xi(I - D\Xi)^{-1}C - B\Delta(I - D\Delta)^{-1}C$. Since this works for arbitrarily small boxes $\mathbf{\Delta}$, we can pick $K, F \rightarrow 0$ such that $\mathcal{B}(\Xi, \Delta) \rightarrow 0$, $\alpha \rightarrow \alpha_{\infty}$ for some limit, and $\mathcal{A}(\Delta) \rightarrow A - \alpha_{\infty}I - \epsilon I$. Then $\mathcal{A} \rightarrow A - \alpha_{\infty}I - \epsilon I$, which has spectral abscissa = $-\epsilon$. This contradicts the fact that each \mathcal{A} is unstable. □

5. LMI-BASED PRUNING TEST

The test (12) is easy to compute, but rather conservative. Test (13) is less conservative but computationally more demanding. The following test is yet another alternative with reduced conservatism, now based on LMIs. We use the following

Lemma 4. (Graham et al. (2006)). Fix $0 < \omega_0 < \infty$. Suppose there exist Hermitian matrices $Z_1, Z_2 \succ 0$ commuting with the Δ , Hermitian matrices Y_1, Y_2 commuting with the Δ , and complex matrices F, G such that

7. EXPERIMENTS

In this section we present the results achieved by our branch and bound algorithms on 32 benchmarks given in Table 1. The tests were realized using MatlabR2014b and on a 64-bit PC with 2.70GHZ dual-core and 16, 0 Go RAM.

#	Benchmark	n	Structure	n_{dec}
1	Beam3	11	$1^3 3^1 1^1$	338
2	Beam4	11	$1^3 3^1 1^1$	338
3	Dashpot system1	17	1^6	534
4	Dashpot system2	17	1^6	534
5	Dashpot system3	17	1^6	534
6	DC motor3	7	$1^1 2^2$	162
7	DC motor4	7	$1^1 2^2$	162
8	DVD driver2	10	$1^1 3^3 1^1 3^1$	542
9	Four-disk system3	16	$1^1 3^5 1^4$	1112
10	Four-disk system4	16	$1^1 3^5 1^4$	1112
11	Four-disk system5	16	$1^1 3^5 1^4$	1112
12	Four-tank system3	12	1^4	268
13	Four-tank system4	12	1^4	268
14	Hard disk driver3	22	$1^3 2^4 1^4$	1258
15	Hard disk driver4	22	$1^3 2^4 1^4$	1258
16	Hydraulic servo3	9	1^9	266
17	Hydraulic servo4	9	1^9	266
18	Mass-spring3	8	1^2	112
19	Mass-spring4	8	1^2	112
20	Missile3	35	$1^3 6^3$	3174
21	Missile4	35	$1^3 6^3$	3174
22	Missile5	35	$1^3 6^3$	3174
23	Filter3	8	1^1	92
24	Filter4	8	1^1	92
25	Filter-Kim3	3	1^2	32
26	Filter-Kim4	3	1^2	32
27	Satellite3	11	$1^1 6^1 1^1$	460
28	Satellite4	11	$1^1 6^1 1^1$	460
29	Satellite5	11	$1^1 6^1 1^1$	460
30	Mass-spring-damper3	13	1^1	212
31	Mass-spring-damper4	13	1^1	212
32	Mass-spring-damper5	13	1^1	212

Table 1. Benchmarks; Apkarian et al. (2015b).

In Table 1 column n shows the number of states in (1), while column structure allows to retrieve the uncertain structure $[r_1, \dots, r_m]$. For instance $[1^3 3^1 1^1] = [1 \ 1 \ 1 \ 3 \ 1] = [r_1 \ r_2 \ r_3 \ r_4 \ r_5]$ in benchmark Beam 3, and $[1^3 2^4 1^4] = [1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 1] = [r_1 \dots r_{11}]$ in benchmark Hard-Disk-Driver 4. The number of decision variables in (14)-(17) and (20)-(21) is given in column n_{dec} .

In Table 2, column $\underline{\alpha}$ gives the best lower bound achieved by the local solver during B&B with $\mathcal{P} = \mathcal{P}_{\bar{\mu}}$, where ranking pushes those Δ to the end of the list, in which a $\underline{\delta}$ realizing the current $\underline{\alpha}$ occurs. Column $\bar{\alpha}$ gives the value $\bar{\alpha} = \underline{\alpha} + \epsilon = \underline{\alpha} + |\underline{\alpha}| \cdot tol$, where ϵ is scaled to $\underline{\alpha}$ such that the relative error is $tol \approx 0.01$. On exit the algorithm believes that the global maximum is $\underline{\alpha}$, and certifies that the true global maximum α^* lies between $\underline{\alpha}$ and $\bar{\alpha} = \underline{\alpha} + |\underline{\alpha}| \cdot tol = \underline{\alpha} + \epsilon$. The CPU times are t if the local solver is run as stand-alone to achieve the value $\underline{\alpha}$, and t^* for the branch-and-bound solver to achieve $\alpha^* \in [\underline{\alpha}, \bar{\alpha}]$.

We also tested the algorithm with $\mathcal{P} = \mathcal{P}_{\bar{\mu}}$. This corresponds to an improved version of Balakrishnan et al. (1991). Here the CPU is exceedingly long due to the strong

#	$\bar{\alpha}$	$\underline{\alpha}$	tol	t^*
1	$-1.23e-7$	$-1.23e-7$	0.01	2.03
2	$-1.75e-7$	$-1.74e-7$	0.01	0.73
3	0.0188	0.0186	0.01	588.45
4		$-1.0e-6$		
5	$-0.99e-6$	$-1.0e-6$	0.01	0.71
6	$-0.99e-3$	$-1e-3$	0.01	1.68
7	$-0.99e-3$	$-1e-3$	0.05	1.79
8		-0.0165		
9	0.0102	0.0089	0.15	669.84
10	$-7.5e-7$	$-7.5e-7$	0.01	4.12
11	$-1e-7$	$-1e-7$	0.01	5.35
12	$-6.0e-6$	$-6.0e-6$	0.01	0.67
13	$-6.0e-6$	$-6.0e-6$	0.01	0.25
14	272.03	266.7	0.02	3578.4
15	-1.57	-1.6026	0.02	1.52
16	-0.297	-0.3000	0.01	0.57
17	-0.297	-0.3000	0.01	0.72
18	-0.0053	-0.0054	0.01	21.3
19	-0.0365	-0.0368	0.01	5.0
20	22.86	22.6302	0.01	39.43
21	-0.49	-0.5000	0.01	17.32
22	-0.49	-0.5000	0.01	22.55
23	-0.0146	-0.0148	0.01	0.1
24	-0.0146	-0.0148	0.01	0.1
25	-0.247	-0.2500	0.01	13
26	-0.242	-0.2500	0.03	13.2
27	$4.2e-5$	$3.9e-5$	0.05	1557.7
28	-0.0255	-0.0269	0.05	0.45
29	-0.0265	-0.0269	0.01	0.45
30	0.2042	0.2022	0.02	0.24
31	-0.099	-0.1000	0.01	0.16
32	-0.099	-0.1000	0.01	0.2

Table 2. Results obtained with Branch-and-Bound algorithm I when $\mathcal{P} = \mathcal{P}_{\bar{\mu}}$

conservatism of the pruning test, with the effect that only very tiny boxes are pruned. We do not report the CPUs.

The algorithm was then tested with the frequency sweep $\mathcal{P}(\Delta, \underline{\alpha}, \epsilon, \omega^b, \omega^\sharp)$ in (19). For cases 20-22 the LMI solver failed to compute ω^b, ω^\sharp due to the large number of decision variables (see Table 1, column n_{dec}). In the other cases, the search for ω^b and ω^\sharp turned out time consuming. For example, in benchmark 1 computing these frequencies for $\Delta = [-1, 1]^5$ takes 20.22 respectively 204.87 seconds, leading to $\omega^b = 0.05$ and $\omega^\sharp = 0.1$.

Subsequently, $\Delta' = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}^4$, $\Delta'' = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^4$ are pruned rapidly, because $\mathcal{P}(\Delta', \underline{\alpha}, \epsilon, 0.05, 0.1) = \text{pruning}$ and $\mathcal{P}(\Delta'', \underline{\alpha}, \epsilon, 0.05, 0.1) = \text{pruning}$. The final t^* for the first two benchmarks are 228.43 and 234.2 instead of 2.03 and 0.73 seconds reported in Table 2.

The last test was algorithm 1 with $\mathcal{P} = \mathcal{P}_{\bar{\mu}}$ and frequency sweep. Frequencies ω^b and ω^\sharp were computed by bisection and evaluation of the H_∞ -norm on the low- and high-frequency bands. We observed that evaluation of ω^b and ω^\sharp was considerably faster than with the LMI method, but $\bar{\mu}$ computed on $[\omega^b, \omega^\sharp]$ remained very conservative, so that pruning occurred only for tiny boxes. Except for cases 23-24 and 30-32, which have a very simple uncertain structure, t^* was extremely large and we do not report the result here. In contrast, we observed that pruning by

the LMI method, and evaluation of ω^\flat and ω^\sharp by the H_∞ -norm method reduced t^* considerably. For the first two benchmarks, t^* improved to 45.8 and 42.9 seconds.

We also tested two alternative global optimizers, the methods of Zheng et al. (1995), and Lasserre (2001), Henrion et al. (2004). In table 3 the Zheng-method computes α_{ZM} in t_{ZM} , Lasserre's method α_{LMI} in t_{LMI} seconds, and both are compared to $\underline{\alpha}$ obtained in \underline{t} seconds when the local solver was used as stand-alone. This value $\underline{\alpha}$ is used to initialize the B&B.

Algorithm 2 . Zheng-method for $\alpha^* = \max_{x \in \Delta} f(x)$

- 1: **Initialize.** Choose initial $\alpha < \alpha^*$.
- 2: **Loop.** Compute $\alpha^+ = \int_{[f \geq \alpha]} f(x) d\mu(x) / \mu[f \geq \alpha]$.
- 3: **Stopping.** If progress of α^+ over α is marginal, stop, otherwise $\alpha \rightarrow \alpha^+$ and goto step 2.

#	$\underline{\alpha}$	α_{ZM}	α_{LMI}	\underline{t}	t_{ZM}	t_{LMI}
1	-1.2e-7	-1.2e-7	x	0.19	32.70	x
2	-1.7e-7	-1.7e-7	x	0.04	33.00	x
3	0.0186	0.0185	x	0.23	90.25	x
4	-1.0e-6	-1.0e-6	x	0.39	39.63	x
5	-1.0e-6	-1.0e-6	x	0.08	39.63	x
6	-0.0010	-0.0010	x	0.02	20.63	x
7	-0.0010	-0.0010	x	0.02	20.74	x
8	-0.0165	-0.0165	x	0.04	49.29	x
9	0.0089	0.0089	x	0.10	159.91	x
10	-7.5e-7	-7.5e-7	x	0.29	73.86	x
11	-1.0e-7	-1.0e-7	x	0.29	74.63	x
12	-6.0e-6	-6.0e-6	x	0.02	26.03	x
13	-6.0e-6	-6.0e-6	x	0.02	26.2	x
14	266.7	-1.60	x	0.2	1252.5	x
15	-1.6026	-1.6026	x	0.06	80.4	x
16	-0.3000	-0.3000	x	0.04	51.41	x
17	-0.3000	-0.3000	x	0.02	50.95	x
18	-0.0054	-0.0054	x	0.01	31.59	x
19	-0.0368	-0.0370	x	0.01	16.94	x
20	22.6302	22.1682	x	0.07	104.18	x
21	-0.5000	-0.5000	x	0.07	51.78	x
22	-0.5000	-0.5000	x	0.07	52.24	x
23	-0.0148	-0.0148	x	0.06	7.05	x
24	-0.0148	-0.0148	x	0.02	6.89	x
25	-0.2500	-0.2500	-0.2475	0.01	12.83	0.25
26	-0.2500	-0.2500	-0.2475	0.01	12.9	0.26
27	3.9e-5	3.9e-5	x	0.02	44.02	x
28	-0.0269	-0.0269	x	0.02	26.02	x
29	-0.0268	-0.0268	x	0.02	26.08	x
30	0.2022	0.2022	x	0.01	8.30	x
31	-0.1000	-0.1000	x	0.01	6.91	x
32	-0.1000	-0.1000	x	0.01	6.94	x

Table 3. Comparison of global solvers.

Lasserre's method solves the BMI problem (4) by a hierarchy of LMIs. Following Henrion et al. (2004), robust stability of $M_{\underline{\alpha}}$ over Δ is certified when the value of the following polynomial optimization problem is > 0 :

$$\begin{aligned} & \text{minimize } \det(H(\delta)) \\ & \text{subject to } \delta \in \Delta \end{aligned} \quad (22)$$

where $H(\delta)$ is the so-called Hermite-matrix of $M_{\underline{\alpha}}$. The method uses GloptiPoly, and Maple 14 to compute $\det(H(\delta))$. In table 3 α_{LMI} improves over $\underline{\alpha}$ in cases 25,

26, but in the remaining cases no certificate could be obtained even when feasibility of the SDP-solver SeDuMi was enlarged to 10^3 , and a large number of LMIs was considered. The bottleneck of Lasserre's method is slow convergence of the LMI approximation, that lower bounds cannot be taken into account, and the necessity to compute $\det(H(\delta))$ formally. For instance, for $n = 7$ Maple produces 75 pages output for $\det(H(\delta))$.

8. CONCLUSION

We presented a branch and bound algorithm to compute the worst-case spectral abscissa of a system with uncertain parameters. A bundle trust-region solver was used to compute good lower bounds, and a frequency sweeping technique increased the efficiency of the pruning test, which gives our method a decisive advantage over previous approaches such as Gaston et al. (1988), Sideris et al. (1989), or Balakrishnan et al. (1991). The method was tested on a bench of 32 systems with up to 35 states, 4 uncertain parameters, and 4 repetitions. The results were matched with those of two alternative global solvers.

REFERENCES

- P. Apkarian, M.N. Dao, D. Noll. Parametric robust structured control design. *IEEE Trans. Aut. Con.* 2015.
- P. Apkarian, D. Noll, L. Ravanbod. Nonsmooth bundle trust-region algorithm with applications to robust stability. Submitted 2015.
- P. Apkarian, D. Noll, L. Ravanbod. Computing the structured distance to instability. *Proc. SIAM Conf. Contr. Appl.*, Paris, 2015.
- P. Apkarian, D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Automat. Control* 51(1): 2006, 71-86.
- V. Balakrishnan, S. Boyd, and S. Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *Int. J. Robust Nonlinear Control*, 1(4):1991,295-317.
- J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proc. D* 129(6):1982,242-250.
- M.K.H. Fan, A.L. Tits, J.C. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Aut. Con.* 36(1):1991,25-38.
- R. E. De Gaston, M.G. Safonov. Exact calculation of the multiloop stability margin. *IEEE TAC*, 33(2):1988,156-171.
- M.R. Graham, M.C. de Oliveira, R.A. de Callafon. A linear matrix inequality for robust stability analysis with frequency dependent multipliers. *Proc. 45th IEEE CDC*, San Diego, 2006, 5144-5149.
- D. Henrion, D. Arzelier, D. Peaucelle, J.-B. Lasserre. On parameter-dependent Lyapunov functions for robust stability of linear systems. *43rd IEEE CDC*, 2004.
- J.-B. Lasserre. Global optimization with polynomials and the problem of moments. *SIOPT* 11:2001,796-817.
- M.G. Safonov. Stability and robustness of multivariable feedback systems. MIT Press, Cambridge, 1980.
- A. Sideris, R. S. S. Peña. Fast computation of the multivariable stability margin for real interrelated uncertain parameters. *IEEE TAC*, 34(12):1989,1272-1276.
- Q. Zheng and D. Zhuang. Integral global minimization: algorithms, implementations, and numerical tests. *Journal of Global Optimization* 7:1995, 421 - 454.