BRANCH AND BOUND ALGORITHM WITH APPLICATIONS TO ROBUST STABILITY

LALEH RAVANBOD†, DOMINIKUS NOLL‡, AND PIERRE APKARIAN*

Abstract. We discuss a branch and bound algorithm for global optimization of NP-hard problems related to robust stability. This includes computing the distance to instability of a system with uncertain parameters, computing the minimum stability degree of a system over a given set of uncertain parameters, and computing the worst case H_∞ norm over a given parameter range. The success of our method hinges (i) on the use of an efficient local optimization technique to compute lower bounds fast and reliably, (ii) a method with reduced conservatism to compute upper bounds, and (iii) the way these elements are favorably combined in the algorithm.

Keywords. Branch and bound · dynamical system · parametric robustness · stability margin · minimum stability degree · nonsmooth optimization · frequency decomposition

1. Introduction

Physical system models are invariably corrupted by various types of uncertainty. Real parametric uncertainty includes phenomena like poorly identified or incompletely known physical parameters such as masses, inertia, aerodynamic coefficients, data in economics models, but also time-varying parameters due to aging, degrading component variations, thermal effects, and much else. Whether or not stability and performance of a system can be maintained despite such a mismatch between the mathematical model and reality is a key concern of modern design. Quantifying the degree of robustness of a systems is therefore of importance, but generally leads to challenging global optimization problems.

Here we are interested in the analysis of stability and performance of a linear time-invariant dynamical system in the presence of real uncertain parameters. Typical quantities which allow to assess the robustness of such a system include (a) its stability margin or *distance to instability*, (b) its worst-case spectral abscissa or *minimum stability degree* over a given range of uncertain parameters, and (c) its *worst case H_∞ performance* over that range. Unfortunately computation of these quantities is NP-hard, cf. [21, 11, 27], and this makes the use of good heuristic methods mandatory. These heuristics may then be combined with branch and bound to obtain global robustness certificates.

The importance of parametric robust stability and performance was recognized in the late 1970s and 1980s. The principled mathematical tool proposed to describe these properties is the μ-singular value [29], which originated from work by Safonov [24] and Doyle [12]. Our present approach still owes much to this classical line as we use μ-upper bounds according to Fan *et al.* [14] and Graham *et al.* [13] to obtain conservative stability tests for evaluation, see Section 5.

Branch and bound methods for the computation of the minimum stability degree of a system have previously been presented by De Gaston and Safonov [16], Balakrishnan and Boyd [4], Balakrishnan *et al.* [5], and Sideris and Peña [26]. Related branch and

Date: Received: date / Accepted: date.
†Institut de Mathématiques, Université de Toulouse, France lalehravanbod@yahoo.fr.
‡Institut de Mathématiques, Université de Toulouse, France dominikus.noll@math.univ-toulouse.fr.
* ONERA, Control System Department, Toulouse, France Pierre.Apkarian@onera.fr.

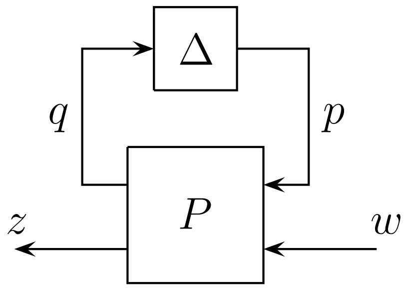

FIGURE 1. Robust system interconnection

bound approaches in robust stability are Bemporad *et al.* [8], Romero-Monsivais *et al.* [23], or Sakizlis *et al.* [25], see also [22]. Here we discuss a novel branch and bound algorithm, which has three key ingredients leading to a significantly better performance. Firstly, we use a highly dedicated local optimizer, based on a trust-region strategy, which computes excellent lower bounds reliably and speedily. Secondly, we present pruning tests that avoid the explicit computations of upper bounds, which leads to an improvement in speed. And thirdly, we present a new way to include frequency information in the setup, which has the effect that when branching occurs due to failure of pruning, it is at least possible to reduce the frequency band on which robust stability will have to be certified at the next level of the arborescence. Experiments with a test bench of 116 challenging examples indicate that our method is a promising tool for engineering control practice.

The structure of the paper is as follows. In section 2 we explain the setup and introduce the three quantities used to assess parametric robustness. In section 3 we develop the branch and bound algorithm for the worst-case spectral abscissa. Finite termination of the method is proved in section 4. Evaluation procedures which do not require the explicit computation of upper bounds are discussed in section 5. In section 6 we show how exploiting frequency information for subproblems can reduce the number of bisections in the branch and bound algorithm. Experiments with the minimum stability degree are reported in section 7. Branch and bound for computing the maximum H_∞-performance and the distance to instability are discussed in sections 8 and 10, while corresponding experimental results are reported in sections 9 and 11.

2. PROBLEM SETTING

Consider a Linear Fractional Transform (LFT) plant with real parametric uncertainties $\mathcal{F}_u(P,\Delta)$ as in Figure 1, where

$$P(s):\left\{\begin{array}{rclclcl} \dot{x} &=& Ax &+& Bp &+& B_w w\\ q &=& Cx &+& Dp &+& D_{qw} w\\ z &=& C_z x &+& D_{zp} p &+& D_{zw} w\end{array}\right. \tag{1}$$

and $x\in\mathbb{R}^n$ is the state, $w\in\mathbb{R}^{n_2}$ a vector of exogenous inputs, and $z\in\mathbb{R}^{n_1}$ a vector of regulated outputs. The uncertainty channel is defined as

$$p=\Delta q, \tag{2}$$

where the uncertain matrix Δ has the block-diagonal form

$$\Delta=\operatorname{diag}\left[\delta_1 I_{r_1},\ldots,\delta_m I_{r_m}\right], \tag{3}$$

with $\delta_1,\ldots,\delta_m$ representing real uncertain parameters, and r_i giving the number of repetitions of δ_i. Here we assume without loss that $\delta=0\in\boldsymbol{\Delta}$ represents the nominal parameter value, and we consider $\delta\in\boldsymbol{\Delta}$ in one-to-one correspondence with the matrix Δ in (3). For practical applications it is generally sufficient to consider the case $\boldsymbol{\Delta}=[-1,1]^m$. Note that every system with real-rational uncertain parameters can be represented by such an LFT.

To analyze the performance of (1) in the presence of the uncertain $\delta \in \mathbb{R}^m$ we compute the worst-case H_∞-performance

$$h^* = \max\{\|T_{wz}(\delta)\|_\infty : \delta \in \mathbf{\Delta}\}, \tag{4}$$

where $\|\cdot\|_\infty$ is the H_∞-norm, and where $T_{wz}(s,\delta) = \mathcal{F}_u(P(s),\Delta)$ is obtained by closing the loop between (1) and $p = \Delta q$ with (3) in Figure 1. The solution $\delta^* \in \mathbf{\Delta}$ of (4) represents a worst possible choice of the parameters $\delta \in \mathbf{\Delta}$, which is an important element in analyzing performance and robustness of the system.

Our second criterion is similar in nature, as it allows to verify whether the uncertain system (1) is robustly stable over a given parameter range $\mathbf{\Delta}$. This can be tested by maximizing the spectral abscissa of the system A-matrix over the parameter range

$$\alpha^* = \max\{\alpha(A(\delta)) : \delta \in \mathbf{\Delta}\}, \tag{5}$$

where $A(\delta) = A + B\Delta(I - D\Delta)^{-1}C$, and where the spectral abscissa of a square matrix A is defined as $\alpha(A) = \max\{\operatorname{Re}\lambda : \lambda \text{ eigenvalue of } A\}$. Since A is stable if and only if $\alpha(A) < 0$, robust stability of (1) over $\mathbf{\Delta}$ is certified as soon as $\alpha^* < 0$, while a destabilizing $\delta^* \in \mathbf{\Delta}$ is found as soon as $\alpha^* \geq 0$.

Note however that a decision in favor of robust stability over $\mathbf{\Delta}$ based on $\alpha^* < 0$ is only valid when the global maximum over $\mathbf{\Delta}$ is computed. This renders (5) a difficult problem, and it is in fact known that solving (5) globally is NP-hard. In [21] Poljak and Rohn have shown that for a gives set of matrices $A_0, \dots, A_k$, deciding whether $A_0 + r_1A_1 + \cdots + r_kA_k$ is stable for all $r_i \in [0,1]$ is NP-hard, and Braatz *et al.* [11] show that deciding whether a system with real (or mixed or complex) uncertainties is robustly stable over a range $\mathbf{\Delta} = [-1,1]^m$ is harder than globally solving a nonconvex quadratic programming problem, hence is NP-hard. For additional information on NP-hardness in control see also Toker and Özaby [27], or Blondel and Tsitsiklis [10], Blondel *et al.* [9].

Our third analysis problem is related to the previous ones and concerns computation of the distance to instability. Assuming $A(\delta)$ stable at the nominal value $\delta = 0$, we ask for the largest variation in the parameter δ under which the system remains stable. This leads to

$$d^* = \max\{d : A(\delta) \text{ stable for all } |\delta|_\infty < d\}, \tag{6}$$

where $|\delta|_\infty = \max\{|\delta_1|, \dots, |\delta_m|\}$ is the maximum norm. This quantity is also known as the stability margin, or as the radius of stability of (1). A formulation of (6) which does not require $A(0)$ to be stable is

$$d^* = \min\{|\delta|_\infty : A(\delta) \text{ unstable}\},$$

and this still works when $A(\delta)$ is stable for all δ, because then $d^* = \min\emptyset = +\infty$. This latter version can be given the form of a constrained optimization program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leqslant \delta_i \leqslant t, i = 1, \dots, m \\ & -\alpha(A(\delta)) \leqslant 0 \end{array} \tag{7}$$

with decision variable $(t,\delta) \in \mathbb{R}^{1+m}$.

3. Worst case spectral abscissa

In this section we consider computation of the global maximum

$$\alpha^* = \max_{\delta \in [-1,1]^m} \alpha\left(A(\delta)\right), \tag{8}$$

where $A(\delta) = A + B\Delta(I - D\Delta)^{-1}C$, and where Δ is in one-to-one correspondence with δ via (3). Robust stability of (1) over $[-1,1]^m$ is guaranteed as soon as $\alpha^* < 0$, and in that case $-\alpha^*$ is also called the minimum stability degree of (1) over $[-1,1]^m$, see [16, 5]. Namely, solutions of $\dot{x} = A(\delta)x$ then decay at least as fast as $e^{\alpha^* t}$ for all $\delta \in [-1,1]^m$. Here we develop a branch and bound algorithm for (5), whose elements are discussed in detail in the following subsections.

3.1. **Global lower bound.** We use a local optimization technique **Trust** based on a non-smooth bundling trust-region algorithm [1] to compute a locally optimal solution $\underline{\alpha}$ of (5), which gives a lower bound $\underline{\alpha} \leqslant \alpha^*$ for (5). More details on **Trust** are given in section 3.5.

Suppose the current best lower bound $\underline{\alpha}$ is attained at $\underline{\delta} \in [-1,1]^m$. Then $\underline{\delta}$ is our candidate for the global optimum, and following standard terminology, we call $\underline{\delta}$ the *incumbent*.

3.2. **Subboxes and subproblems.** The branch-and-bound method on $[-1,1]^m$ uses subproblems of (8) specified by subboxes of the form $\boldsymbol{\Delta} = \prod_{i=1}^m [a_i, b_i]$, where $-1 \leqslant a_i < b_i \leqslant 1$. The subproblem associated with $\boldsymbol{\Delta}$ is

$$\alpha^*(\boldsymbol{\Delta}) := \max_{\delta \in \boldsymbol{\Delta}} \alpha\left(A(\delta)\right). \tag{9}$$

Two subboxes $\boldsymbol{\Delta}, \boldsymbol{\Delta}'$ are non-overlapping if they are disjoint or have only boundary points in common. During the algorithm we maintain a list $\mathscr{L} = \{\boldsymbol{\Delta}_1, \dots, \boldsymbol{\Delta}_N\}$ of pairwise non-overlapping subboxes of $[-1,1]^m$, which we call the *list of doables*. Every such box represents a subproblem (9), which remains to be evaluated. The algorithm stops when the list of doables $\mathscr{L}$ is empty.

3.3. **Evaluating a subproblem.** A function $\overline{\alpha}(\cdot)$ defined on subboxes $\boldsymbol{\Delta} \subset [-1,1]^m$ is called an upper bound if it satisfies

$$\alpha^*(\boldsymbol{\Delta}) = \max_{\delta \in \boldsymbol{\Delta}} \alpha\left(A(\delta)\right) \leqslant \overline{\alpha}(\boldsymbol{\Delta}). \tag{10}$$

Any useful upper bound gets better as the boxes get smaller in the sense that

$$\lim_{\operatorname{diam}(\boldsymbol{\Delta}) \to 0} \overline{\alpha}(\boldsymbol{\Delta}) - \alpha^*(\boldsymbol{\Delta}) = 0. \tag{11}$$

Following standard terminology one *evaluates* a subproblem $\boldsymbol{\Delta} \in \mathscr{L}$ by computing its upper bound $\overline{\alpha}(\boldsymbol{\Delta})$. One then makes the following decision

$$\begin{cases} \text{if } \overline{\alpha}(\boldsymbol{\Delta}) \leqslant \underline{\alpha} \text{ then} & \texttt{pruning } \boldsymbol{\Delta} \\ \text{otherwise} & \texttt{not\_pruning} \end{cases} \tag{12}$$

Following standard terminology, the term `pruning` refers to the fact that one can remove $\boldsymbol{\Delta}$ from the list $\mathscr{L}$ without successors, because $\alpha^*(\boldsymbol{\Delta}) \leqslant \overline{\alpha}(\boldsymbol{\Delta}) \leqslant \underline{\alpha}$, so that $\boldsymbol{\Delta}$ cannot contain any solution better than the present incumbent. So here the list of doables shrinks. On the other hand, in case of the decision `not_pruning` one has to replace $\boldsymbol{\Delta}$ by two new boxes in the list $\mathscr{L}$, which makes $\mathscr{L}$ grow by one.

To get these new boxes we divide $\boldsymbol{\Delta}$ into two subboxes of half volume by cutting a longest edge in two. This gives $\boldsymbol{\Delta}'$, $\boldsymbol{\Delta}''$, which we add to $\mathscr{L}$. More explicitly, suppose $\boldsymbol{\Delta} = \prod_{i=1}^m [a_i, b_i]$, then choose a coordinate i_0 with the largest $b_{i_0} - a_{i_0}$ and take the obvious

$$\boldsymbol{\Delta}' = [a_1, b_1] \times \cdots \times [a_{i_0}, \tfrac{a_{i_0}+b_{i_0}}{2}] \times \cdots \times [a_m, b_m],$$

$$\boldsymbol{\Delta}'' = [a_1, b_1] \times \cdots \times [\tfrac{a_{i_0}+b_{i_0}}{2}, b_{i_0}] \times \cdots \times [a_m, b_m].$$

When operating the algorithm, it is reasonable to allow tolerances in the decision, that is, one might replace (12) by the following test

$$\mathscr{P}_{\text{ub}} : \begin{cases} \text{if } \overline{\alpha}(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon \text{ then} & \texttt{pruning } \mathbf{\Delta} \\ \text{otherwise} & \texttt{not\_pruning} \end{cases} \tag{13}$$

where $\epsilon \geqslant 0$ is some level of tolerance.

3.4. **Evaluation without upper bound.** We discuss a variant of evaluation, where it is not necessary to know a numerical value of $\overline{\alpha}(\mathbf{\Delta})$. To reach the decision (12) is would suffice to know whether $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha}$ or not, and similarly in (13). Since this decision has to be conservative, we propose the the following

Definition 1. A decision procedure $\mathscr{P}$ which, given a box $\mathbf{\Delta}$ and a reference value $\underline{\alpha}$ on input, and being allowed a tolerance level $\epsilon \geqslant 0$, issues a decision between `pruning` $\mathbf{\Delta}$ and `not_pruning`, is called a pruning test if the decision `pruning` $\mathbf{\Delta}$ is only issued when it is certified that $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon$. □

In other words, the decision $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, 0) =$ `pruning` gives a certificate that the box $\mathbf{\Delta}$ contains no value better than $\underline{\alpha}$, and can therefore be pruned when $\underline{\alpha}$ is a lower bound of α^*. Similarly, $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) =$ `pruning` certifies that $\mathbf{\Delta}$ cannot improve over $\underline{\alpha} + \epsilon$.

Lemma 1. *The test $\mathscr{P}_{\text{ub}}$ in (13) based on the upper bound $\overline{\alpha}(\mathbf{\Delta})$ is a pruning test.*

Proof. Indeed, since pruning occurs when $\overline{\alpha}(\mathbf{\Delta}) \leqslant \underline{\alpha}+\epsilon$, and since an upper bound satisfies $\alpha^*(\mathbf{\Delta}) \leqslant \overline{\alpha}(\mathbf{\Delta})$, the decision $\mathscr{P}_{\text{ub}}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) =$ `pruning` certifies that $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon$. □

Every upper bound gives rise to a pruning test, and inversely, a pruning test could be used to define an upper bound by way of bisections.

Proposition 1. *Suppose $\mathscr{P}$ is a pruning test and let $\epsilon \geqslant 0$. Define $\overline{\alpha}(\cdot)$ by*

$$\overline{\alpha}(\mathbf{\Delta}) = \sup\{\alpha \in \mathbb{R} : \mathscr{P}(\mathbf{\Delta}, \alpha, \epsilon) = \texttt{not\_pruning}\}.$$

Then $\overline{\alpha}(\cdot)$ is an ϵ-upper bound, i.e., $\alpha^*(\mathbf{\Delta}) \leqslant \overline{\alpha}(\mathbf{\Delta}) + \epsilon$.

Proof. Suppose the estimate is wrong, then one can choose α such that $\overline{\alpha}(\mathbf{\Delta}) < \alpha < \alpha^*(\mathbf{\Delta}) - \epsilon$. Due to the first inequality $\mathbf{\Delta}$ is pruned by $\mathscr{P}$ when given the value α on entry. But since $\mathscr{P}$ is a pruning test, this implies $\alpha^*(\mathbf{\Delta}) \leqslant \alpha + \epsilon$, a contradiction. □

This result indicates why it may in general be too costly to compute an explicit upper bound, and our present approach avoids this. Several pruning procedures without explicit computation of an upper bound will be discussed in section 5.

3.5. **Local trust-region optimization.** In this section we briefly discuss the local trust-region method **Trust**, which is the key element in our branch and bound approach for (5), because it gives excellent lower bounds $\underline{\alpha}$ very speedily.

To understand the context, recall that Balakrishnan *et al.* [5] use the following very simple lower bound. For every box $\mathbf{\Delta} \in \mathscr{L}$ they evaluate $\alpha\left(A(\delta_{\mathbf{\Delta}})\right)$ at the center $\delta_{\mathbf{\Delta}}$ of $\mathbf{\Delta}$, and take the maximum of these values as lower bound:

$$\alpha_{\text{lb}}^{(n)} = \max_{k \leqslant n} \max_{\mathbf{\Delta} \in \mathscr{L}_k} \alpha\left(A(\delta_{\mathbf{\Delta}})\right) \leqslant \alpha^*$$

where at iteration n maximization is over all instances $\mathscr{L}_1 \cup \cdots \cup \mathscr{L}_n$ of the list up to the present instance $\mathscr{L}_n$. This bound is updated fairly often, as its quality is poor at the beginning. The fact that $\underline{\alpha}$ computed by **Trust** is much more accurate gives our method an advantage for pruning. For a detailed analysis of the trust-region method we refer to [1, 3, 20, 19, 2]. For the current analysis it suffices to know that if **Trust** is started at an initial guess δ_0, then it finds a locally optimal solution δ^* such that $\alpha(A(\delta^*)) \geqslant \alpha(A(\delta_0))$.

The best value so achieved is chosen as lower bound $\underline{\alpha}$, and the monotony property is exploited in the branch and bound algorithm. With that it is easy to improve on $\underline{\alpha}$ by starting **Trust** at the center of the box $\mathbf{\Delta}$.

3.6. **Ranking boxes $\mathbf{\Delta} \in \mathscr{L}$.** As soon as two new boxes $\mathbf{\Delta}', \mathbf{\Delta}''$ enter the list $\mathscr{L}$, replacing an evaluated box $\mathbf{\Delta}$ which could not be pruned, we pass old information already known about $\mathbf{\Delta}$, and new information obtained during its evaluation process, on to the new boxes. This includes of course size and depth in the tree, but also the position $\underline{\delta}(\mathbf{\Delta}) \in \mathbf{\Delta}$ of the local minimizer obtained when running **Trust** inside the box $\mathbf{\Delta}$. For instance, $\underline{\delta}(\mathbf{\Delta}) \in \mathbf{\Delta}' \setminus \mathbf{\Delta}''$ is a strong indicator that $\mathbf{\Delta}''$ might be easier to prune than $\mathbf{\Delta}'$. Nearness $\underline{\alpha}(\mathbf{\Delta}) \approx \underline{\alpha}$ indicates that the box $\mathbf{\Delta}'$ might in the end still give an improvement of the lower bound, while $\underline{\alpha}(\mathbf{\Delta}) \ll \underline{\alpha}$ might indicate that both boxes $\mathbf{\Delta}', \mathbf{\Delta}''$ should be good for pruning.

In addition to these rather standard informations, we shall also store additional frequency information, which we shall discuss in section 6.

Algorithm 1. Branch and bound to compute α^* in program (5).

Parameters: Pruning tolerance $\epsilon \geqslant 0$.
Subroutines: **Trust** for local optimization, $\mathscr{P}$ for pruning, $\mathscr{R}$ for ranking.

Lower bound. Call local solver **Trust** to compute lower bound $\underline{\alpha}$ of α^*. Let $\underline{\delta} \in [-1,1]^m$ be a local maximum called the incumbent.
Initialize list. Put $\mathscr{L} = \{[-1,1]^m\}$.
while $\mathscr{L} \neq \emptyset$ **do**
  Choose element $\mathbf{\Delta} \in \mathscr{L}$ ranked first for evaluation.
  Call pruning test $\mathscr{P}$ for $\mathbf{\Delta}$ with tolerance ϵ.
  if $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) = \texttt{pruning}$ **then**
    Remove $\mathbf{\Delta}$ from $\mathscr{L}$
  else
    Remove $\mathbf{\Delta}$ and replace it by two successors $\mathbf{\Delta}'$, $\mathbf{\Delta}''$ in $\mathscr{L}$
    Call local solver within $\mathbf{\Delta}$. Update lower bound and incumbent.
  end if
  Call $\mathscr{R}$ to update ranking of $\mathscr{L}$
end while
Return $\underline{d}$ and $\underline{\delta}$.

---

4. Convergence and finite termination

In order to assure convergence or finite termination of algorithm 1, we have to assure that our conservative pruning test gets more and more accurate as the boxes $\mathbf{\Delta}$ get smaller. The following definition is helpful.

Definition 2. Let $\mathscr{P}$ be a pruning test. We say that $\mathscr{P}$ is weakly consistent if for every $\epsilon > 0$ there exists $\eta > 0$ such that whenever the diameter of the box $\mathbf{\Delta}$ is smaller than η and $\alpha \geqslant \alpha^*$ is used as reference value with tolerance ϵ, then $\mathbf{\Delta}$ is pruned. We say that $\mathscr{P}$ is consistent if the above is even true for every $\alpha \geqslant \alpha^*(\mathbf{\Delta})$. □

For short weak consistency says that for every $\epsilon > 0$ there exists $\eta > 0$ such that

$$\mathscr{P}(\mathbf{\Delta}, \alpha^* + a, \epsilon) = \texttt{pruning} \text{ whenever } \operatorname{diam}(\mathbf{\Delta}) < \eta \text{ and } a \geqslant 0.$$

For consistency the short form is: for every $\epsilon > 0$ there exists $\eta > 0$ such that

$$\mathscr{P}(\mathbf{\Delta}, \alpha^*(\mathbf{\Delta}) + a, \epsilon) = \texttt{pruning} \text{ whenever } \operatorname{diam}(\mathbf{\Delta}) < \eta \text{ and } a \geqslant 0.$$

Remark 1. Let us convince ourselves that the pruning test $\mathscr{P}_{\rm ub}$ based on an upper bound $\overline{\alpha}(\mathbf{\Delta})$ is consistent as soon as (11) holds. Indeed, from (11), given $\epsilon > 0$ we can find $\eta > 0$ such that for all boxes of diameter less than η we have $\overline{\alpha}(\mathbf{\Delta}) \leqslant \alpha^*(\mathbf{\Delta}) + \epsilon$. Therefore, if we choose a level $\alpha \geqslant \alpha^*(\mathbf{\Delta})$, then $\alpha + \epsilon \geqslant \alpha^*(\mathbf{\Delta}) + \epsilon \geqslant \overline{\alpha}(\mathbf{\Delta})$ in the test, hence $\mathbf{\Delta}$ is pruned.

4.1. **Convergence for consistent $\mathscr{P}$.** In the case of a consistent pruning test finiteness of the algorithm can be guaranteed.

Theorem 1. *Suppose the pruning test $\mathscr{P}$ is consistent. Fix $\epsilon > 0$ and run the branch and bound algorithm based on $\mathscr{P}$ and tolerance ϵ. Then the algorithm stops after a finite number of steps with an empty list $\mathscr{L}$, and it provides an incumbent and lower bound $\underline{\alpha}$ such that $\underline{\alpha} \geqslant \alpha^* - \epsilon$.*

Proof. 1) Since $\delta \mapsto \alpha\left(A(\delta)\right)$ is a continuous function of $\delta \in \mathbb{R}^m$, it is uniformly continuous on $[-1,1]^m$, hence there exists $\eta > 0$ such that for all $\delta, \delta' \in [-1,1]^m$, $|\delta - \delta'| < \eta$ implies $|\alpha(A(\delta)) - \alpha(A(\delta'))| < \frac{1}{2}\epsilon$. Therefore, as soon as a box $\mathbf{\Delta}$ with diameter $< \eta$ is evaluated, **Trust** finds a value $\underline{\alpha}(\mathbf{\Delta})$ which is $\frac{1}{2}\epsilon$-close to $\alpha^*(\mathbf{\Delta})$. Since $\underline{\alpha}$ is updated when an improvement is found, we know for such an evaluated box that $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha} + \frac{1}{2}\epsilon$.

2) Using consistency of $\mathscr{P}$, by further reducing the η found in 1), we can assume that the decision $\mathscr{P}(\mathbf{\Delta}, \alpha^*(\mathbf{\Delta}) + a, \frac{1}{2}\epsilon) = \texttt{pruning}$ occurs for every $\mathbf{\Delta}$ with diameter $< \eta$ and every $a \geqslant 0$.

3) Let us now show that if the algorithm ends after a finite number of steps, then we must have $\underline{\alpha} \geqslant \alpha^* - \epsilon$. Indeed, since in the end the list is empty, there must have been at some counter n a box $\mathbf{\Delta}^* \in \mathscr{L}$ which contains the global maximum δ^* and which was pruned based on the current lower bound $\underline{\alpha}^{(n)}$, which was no better than the final value of the lower bound $\underline{\alpha}$. But since we are using a pruning test, we know from definition 1 that $\alpha^* = \alpha^*(\mathbf{\Delta}^*) \leqslant \underline{\alpha}^{(n)} + \epsilon \leqslant \underline{\alpha} + \epsilon$ was satisfied when pruning occurred.

4) Let us next observe that if there exists an iteration index $n_0 \in \mathbb{N}$ and $\eta > 0$ such that for counters $n \geqslant n_0$ all boxes with diameter less than η are automatically pruned, then the algorithm terminates finitely. Indeed, after a finite number of steps we are running out of boxes larger than η.

5) Let us now assume that the algorithm does not halt and create a sequence $\underline{\alpha}^{(n)}$ of lower bounds with $\underline{\alpha}^{(1)} \leqslant \underline{\alpha}^{(2)} \leqslant \cdots \to \underline{\alpha}$. Then by 4) there must exist boxes $\mathbf{\Delta}_k$ with arbitrarily small diameters $\leqslant \eta_k \to 0$, which are evaluated but not pruned. For those boxes, $\limsup \alpha^*(\mathbf{\Delta}_k) \leqslant \underline{\alpha} + \frac{1}{2}\epsilon$. Let us assume that $\mathbf{\Delta}_k$ occurs at counter n_k. Then for k large enough, $\alpha^*(\mathbf{\Delta}_k) < \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon$, hence $\mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)}, \epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon, \frac{1}{2}\epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \alpha^*(\mathbf{\Delta}_k) + a_k, \frac{1}{2}\epsilon) = \texttt{pruning}$ due to part 2), where $a_k = \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon - \alpha^*(\mathbf{\Delta}_k) \geqslant 0$. This is a contradiction, because $\mathbf{\Delta}_k$ was not pruned. □

4.2. **Convergence for weakly consistent $\mathscr{P}$.** The following result uses a similar argument. We say that the algorithm follows the width-first rule if

$$\max\{\operatorname{diam}(\mathbf{\Delta}) : \mathbf{\Delta} \in \mathscr{L}_n\} \to 0 \quad (n \to \infty), \tag{14}$$

with $\mathscr{L}_n$ denoting the list at the nth iteration of the algorithm. This very tolerant rule says only that we are not allowed to drive the binary tree representing subproblems in the list to arbitrary hight while leaving large boxes in the bottom unevaluated, as we would do in a depth-first scenario.

Theorem 2. *Suppose $\mathscr{P}$ is weakly consistent, $\epsilon > 0$ is fixed and used as tolerance for pruning. Suppose the width-first rule is applied. Then the algorithm halts after a finite number of steps with a lower bound $\underline{\alpha} \geqslant \alpha^* - \epsilon$.*

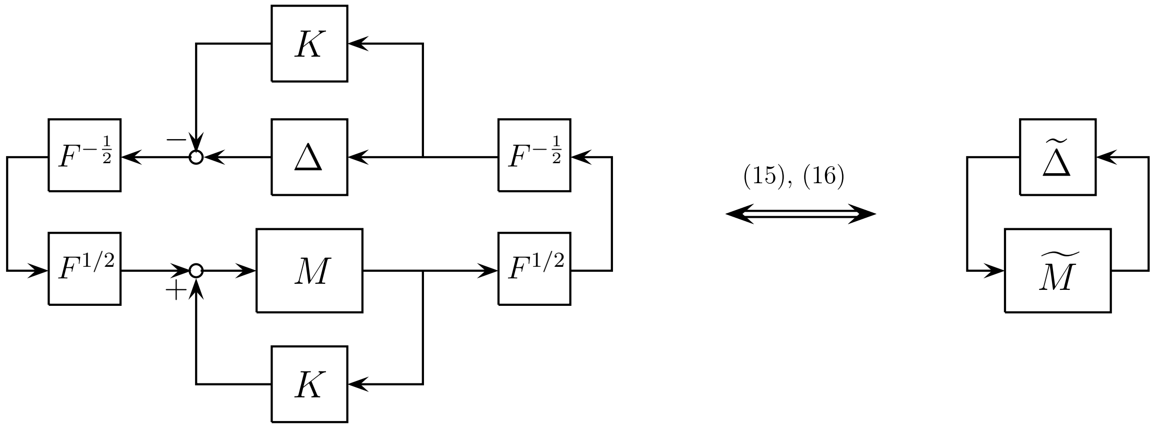

FIGURE 2. On the left $\Delta \leftrightarrow \delta \in \mathbf{\Delta} = \prod_{i=1}^{m}[a_i, b_i]$, on the right, $\widetilde{\Delta} \leftrightarrow \widetilde{\delta} \in \tilde{\mathbf{\Delta}} = [-1, 1]^m$. The two loops are equivalent under (15), (16). Note that $M(s) = C(sI - A)^{-1}B + D$, and $\widetilde{M}(s) = \widetilde{C}(sI - \widetilde{A})^{-1}\widetilde{B} + \widetilde{D}$.

Proof. Assuming that the algorithm does not come to halt, the width-first rule assures that $\underline{\alpha}^{(n)} \to \alpha^*$. Namely, if we fix a global maximum δ^* and call $\mathbf{\Delta}_n^*$ the box in $\mathscr{L}_n$ which contains δ^*, then every now and then this box will have to be evaluated due to the width-first rule, and then its diameter will eventually have to go down. In other words, we will improve the lower bound by evaluating these boxes, and therefore $\underline{\alpha}^{(n)}$ will get closer to α^* and if the algorithm turns infinitely, it will converge to α^*. But then we can modify part 5) of the proof of Theorem 1 such that $\mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)}, \epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon, \frac{1}{2}\epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \alpha^* + a_k, \frac{1}{2}\epsilon) =$ `pruning`, where $a_k = \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon - \alpha^* \geq 0$, so that the same contradiction occurs, now using only weak consistency. $\square$

Remark 2. Using the same width-first rule, and reducing the tolerance to $\epsilon = 0$, we can still prove convergence of the branch and bound method to the global maximum.

Remark 3. The width-first rule can be replaced by the following weaker assumption. If $\mathscr{L}_n$ is the list at iteration n, and $\mathbf{\Delta}_n \in \mathscr{L}_n$ is the element chosen for evaluation, then it must be guaranteed that the current best lower bound $\underline{\alpha}^{(n)}$ based on the history $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_n$ satisfies $\underline{\alpha}^{(n)} \to \alpha^*$. This is an even weaker hypothesis than (14).

5. Evaluation procedures

In this section we examine evaluation procedures without upper bounds in more detail. Following [5] we first apply a loop transformation so that the system (A, B, C, D) with uncertainty $\delta \in \mathbf{\Delta}$ is transformed to a system $(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D})$ which has its uncertainty $\widetilde{\delta}$ with the same structure (3) in $[-1, 1]^m$, (see Figure 2). Assuming $\mathbf{\Delta} = \prod_{i=1}^{m}[a_i, b_i]$, and putting

$$\begin{aligned} K &= \tfrac{1}{2}\mathrm{diag}\left[(a_1 + b_1)I_{r_1}, \ldots, (a_m + b_m)I_{r_m}\right], \\ F &= \tfrac{1}{2}\mathrm{diag}\left[(b_1 - a_1)I_{r_1}, \ldots, (b_m - a_m)I_{r_m}\right], \end{aligned} \tag{15}$$

we define

$$\begin{aligned} \widetilde{A} &= A + B(I - KD)^{-1}KC, & \widetilde{B} &= B(I - KD)^{-1}F^{1/2}, \\ \widetilde{C} &= F^{1/2}(I - DK)^{-1}C, & \widetilde{D} &= F^{1/2}D(I - KD)^{-1}F^{1/2}. \end{aligned} \tag{16}$$

Then we have the following

Lemma 2. (Balakrishnan *et al.* [5]). *For any $\underline{\alpha} \in \mathbb{R}$ the following are equivalent:*

i. $\alpha\left(A + B\Delta(I-D\Delta)^{-1}C\right) < \underline{\alpha}$ *for every* $\delta \in \mathbf{\Delta} = \prod_{i=1}^m [a_i, b_i]$;
ii. $\alpha\left(\widetilde{A} + \widetilde{B}\widetilde{\Delta}(I-\widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C}\right) < \underline{\alpha}$ *for every* $\widetilde{\delta} \in [-1,1]^m$.

Here $\widetilde{\Delta} \leftrightarrow \widetilde{\delta}$, $\Delta \leftrightarrow \delta$ *via* (3). *Moreover, the uncertainties* $\Delta, \widetilde{\Delta}$ *are in one-to-one correspondence via*

$$\widetilde{\Delta} = F^{-1/2}(\Delta - K)F^{-1/2}, \quad \Delta = K + F^{1/2}\widetilde{\Delta}F^{1/2}. \tag{17}$$

For the proof see [5]. □

5.1. **Evaluations based on μ-upper bounds.** Now we can present a first evaluation based on the following test:

$$\mathscr{P}_{\widetilde{\mu}} : \begin{cases} \text{if } \widetilde{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}) := \|(\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})\|_\infty < 1 \text{ then} & \texttt{pruning } \mathbf{\Delta} \\ \text{else} & \texttt{not\_pruning} \end{cases} \tag{18}$$

In fact, the H_∞-norm here is an overestimate of the μ-singular value of the shifted system $\widetilde{M}_{\underline{\alpha}} = (\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})$, hence a value $\widetilde{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}) < 1$ gives $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}) < 1$, hence gives a certificate of parametric robustness of $\widetilde{A}(\widetilde{\delta}) - \underline{\alpha} I = \widetilde{A} - \underline{\alpha} I + \widetilde{B}\widetilde{\Delta}(I - \widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C}$ over $\widetilde{\delta} \in [-1,1]^m$, hence of $A(\delta) - \underline{\alpha} I$ over the box $\delta \in \mathbf{\Delta}$ by Lemma 2, and this assures $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha}$, so that $\mathbf{\Delta}$ can be pruned. The version with tolerance $\epsilon > 0$ is as follows. We prune as soon as $\widetilde{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}+\epsilon}) < 1$.

A second less conservative stability test could be based on the μ-upper bound proposed in [14, 15], see also [7], defined as

$$\widehat{\mu}_{\mathbf{\Delta}}(M) = \inf_{\mathcal{G}\Delta = \Delta\mathcal{G}, \mathcal{D}\Delta = \Delta\mathcal{D}} \sup_{\omega} \inf\left\{\beta \geqslant 0 : M(j\omega)^H \mathcal{D} M(j\omega) + j\left(\mathcal{G}M(j\omega) - M(j\omega)^H \mathcal{G}\right) - \beta^2 \mathcal{D} \preceq 0\right\}. \tag{19}$$

The pruning test based on (19) is then

$$\mathscr{P}_{\widehat{\mu}} : \begin{cases} \text{if } \widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}) < 1 \text{ then} & \texttt{pruning } \mathbf{\Delta} \\ \text{else} & \texttt{not\_pruning} \end{cases} \tag{20}$$

where $\widetilde{M}_{\underline{\alpha}} = (\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})$ is again the shifted system obtained from the box $\mathbf{\Delta}$ as in (15), (16). Again we obtain a version with tolerance if we prune as soon as $\widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}+\epsilon}) < 1$. We conclude with the following observation.

Lemma 3. *The procedures* $\mathscr{P}_{\widetilde{\mu}}, \mathscr{P}_{\widehat{\mu}}$ *are pruning tests. If* $\underline{\alpha}$ *in* (18), (20) *is replaced by* $\underline{\alpha} + \epsilon$, *then they are pruning tests with tolerance level* $\epsilon \geq 0$.

Proof. Indeed, as we have seen above, the decision `pruning` $\mathbf{\Delta}$ in (18) is only issued when $\alpha(A + B\Delta(I-D\Delta)^{-1}C) < \underline{\alpha}$ for every $\delta \in \mathbf{\Delta}$. In other words $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha}$, so that $\mathbf{\Delta}$ may indeed be pruned. The case with tolerance ϵ is similar. □

Proposition 2. *Suppose* (1) *is nominally stable and well-posed over* $[-1,1]^m$. *Then the pruning tests* $\mathscr{P}_{\widetilde{\mu}}, \mathscr{P}_{\widehat{\mu}}$ *are consistent.*

Proof. Since $\mu_{\mathbf{\Delta}} \leqslant \widehat{\mu}_{\mathbf{\Delta}} \leqslant \widetilde{\mu}_{\mathbf{\Delta}}$, it suffices to consider the test $\mathscr{P}_{\widetilde{\mu}}$ based on the μ-upper bound $\mu_{\mathbf{\Delta}} \leqslant \widetilde{\mu}_{\mathbf{\Delta}}$. We have to show that there exists $\eta > 0$ such that when $\operatorname{diam}(\mathbf{\Delta}) < \eta$, $\alpha \geqslant \alpha^*(\mathbf{\Delta})$, and $\epsilon > 0$, then $\mathscr{P}_{\widetilde{\mu}}(\mathbf{\Delta}, \alpha, \epsilon) = \texttt{pruning}$. By (25) the latter means $\widetilde{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\alpha+\epsilon}) < 1$, where $\widetilde{M}_\alpha = (\widetilde{A} - \alpha I, \widetilde{B}, \widetilde{C}, \widetilde{D})$ is the shifted system (15), (16) for $\mathbf{\Delta}$.

For the proof we may assume $\alpha = \alpha^*(\mathbf{\Delta})$. Now since (1) is well-posed over $[-1,1]^m$ and $\widetilde{A} - (\alpha+\epsilon)I$ is stable, the structured singular value of $\widetilde{M}_{\alpha+\epsilon}$ may be expressed as

$\mu_{\boldsymbol{\Delta}}(\widetilde{M}_{\alpha+\epsilon}) = \sup\{1/\overline{\sigma}(\widetilde{\Delta}) : \widetilde{A} - (\alpha+\epsilon)I + \widetilde{B}\widetilde{\Delta}(I - \widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C} \text{ unstable}, \widetilde{\delta} \in [-1,1]^m\}$. This follows from the fact that the closed-loop characteristic polynomial can be written as

$$|sI - (\widetilde{A}(\widetilde{\delta}) - (\alpha+\epsilon)I)| = |sI - (\widetilde{A} - (\alpha+\epsilon)I||I - \widetilde{D}\widetilde{\Delta}|^{-1}|I - \widetilde{M}_{\alpha+\epsilon}(s)\widetilde{\Delta}|,$$

where the first and second term on the right are non-zero. Similarly, the upper bound $\widetilde{\mu}_{\boldsymbol{\Delta}}$ may be expressed as

$$\widetilde{\mu}_{\boldsymbol{\Delta}}(\widetilde{M}_{\alpha+\epsilon}) = \sup\{1/\overline{\sigma}(\widetilde{\Xi}) : \widetilde{A} - (\alpha+\epsilon)I + \widetilde{B}\widetilde{\Xi}(I - \widetilde{D}\widetilde{\Xi})^{-1}\widetilde{C} \text{ unstable}, \widetilde{\Xi} \text{ complex}\},$$

where $\text{size}(\Delta) = \text{size}(\Xi)$, but where Ξ is unstructured. Now suppose the statement is incorrect, then there exist boxes $\boldsymbol{\Delta}$ of arbitrarily small diameter such that $\widetilde{\mu}_{\boldsymbol{\Delta}}(\widetilde{M}_{\alpha+\epsilon}) > 1$ for the corresponding shifted system. Pick a complex $\widetilde{\Xi}$ such that $1/\overline{\sigma}(\widetilde{\Xi}) > 1$ and $\widetilde{A} - (\alpha+\epsilon)I + \widetilde{B}\widetilde{\Xi}(I - \widetilde{D}\widetilde{\Xi})^{-1}\widetilde{C}$ unstable. Applying the inverse loop transformation shows that there exists a complex Ξ such that $A - (\alpha+\epsilon)I + B\Xi(I - D\Xi)^{-1}C$ is unstable and $\Xi = K + F^{1/2}\widetilde{\Xi}F^{1/2}$. Then $\mathcal{A} := A - (\alpha+\epsilon)I + B\Xi(I - D\Xi)^{-1}C$ is also unstable.

Now pick $\widetilde{\delta} \in [-1,1]^m$ such that $\alpha(\widetilde{A} + \widetilde{B}\widetilde{\Delta}(I - \widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C}) = \alpha = \alpha^*(\boldsymbol{\Delta})$, and let $\Delta = K + F^{1/2}\widetilde{\Delta}F^{1/2}$ be its inverse under the loop transform. Then $\delta \in \boldsymbol{\Delta}$ by construction, and $\alpha(A + B\Delta(I - D\Delta)^{-1}C) = \alpha$. Now we may decompose the unstable matrix $\mathcal{A}$ as follows

$$\begin{aligned}\mathcal{A} &= \left[A + B\Delta(I - D\Delta)^{-1}C - \alpha I - \epsilon I\right] + B\Xi(I - D\Xi)^{-1}C - B\Delta(I - D\Delta)^{-1}C \\ &= \mathcal{A}(\Delta) + \mathcal{B}(\Xi, \Delta)\end{aligned}$$

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

5.2. **LMI-based evaluation.** The test (13) is easy to compute, but rather conservative. Test (20) is less conservative but computationally more demanding. The following test offers an alternative which also reduces conservatism over (18). We use the following

Lemma 4. (Graham *et al.* [13]). *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*

$$\begin{bmatrix} \widetilde{C}^H Z_1 \widetilde{C} & \widetilde{C}^H Z_1 \widetilde{D} - j\widetilde{C}^H Y_1 \\ * & \widetilde{D}^H Z_1 \widetilde{D} - Z_1 + jY_1\widetilde{D} + * \end{bmatrix} + \begin{bmatrix} F \\ G \end{bmatrix} \begin{bmatrix}(-\widetilde{A} + \underline{\alpha} I) & -\widetilde{B}\end{bmatrix} + * \prec 0 \tag{21}$$

$$\begin{bmatrix} \widetilde{C}^H Z_2 \widetilde{C} & \widetilde{C}^H Z_2 \widetilde{D} - j\widetilde{C}^H Y_2 \\ * & \widetilde{D}^H Z_2 \widetilde{D} - Z_2 + jY_2\widetilde{D} + * \end{bmatrix} + \begin{bmatrix} F \\ G \end{bmatrix} \begin{bmatrix}(j\omega_0 I - \widetilde{A} + \underline{\alpha} I) & -\widetilde{B}\end{bmatrix} + * \prec 0 \tag{22}$$

and there exist Hermitian matrices $Z_3, Z_4 \succ 0$, Hermitian matrices Y_3, Y_4, all commuting with the Δ, and complex matrices $\widetilde{F}, \widetilde{G}$ such that

$$\begin{bmatrix} \widetilde{C}^H Z_3 \widetilde{C} & \widetilde{C}^H Z_3 \widetilde{D} - j\widetilde{C}^H Y_3 \\ * & \widetilde{D}^H Z_3 \widetilde{D} - Z_3 + jY_3\widetilde{D} + * \end{bmatrix} + \begin{bmatrix} \widetilde{F} \\ \widetilde{G} \end{bmatrix} [jI \;\; 0] + * \prec 0 \tag{23}$$

$$\begin{bmatrix} \widetilde{C}^H Z_4 \widetilde{C} & \widetilde{C}^H Z_4 \widetilde{D} - j\widetilde{C}^H Y_4 \\ * & \widetilde{D}^H Z_4 \widetilde{D} - Z_4 + jY_4\widetilde{D} + * \end{bmatrix} + \begin{bmatrix} \widetilde{F} \\ \widetilde{G} \end{bmatrix} \begin{bmatrix}(jI - \omega_0^{-1}\widetilde{A} + \omega_0^{-1}\underline{\alpha} I) & -\omega_0^{-1}\widetilde{B}\end{bmatrix} + * \prec 0. \tag{24}$$

Then $\alpha^*(\boldsymbol{\Delta}) < \underline{\alpha}$.

Proof. Indeed, if the LMIs (21), (22) are satisfied for Z_1, Z_2, Y_1, Y_2, F, G, then by [13, Theorem 1] (applied with $\eta = 1$, $\omega_1 = 0, \omega_2 = \omega_0$) we have $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega)) \leqslant 1$ for every $\omega \in [0, \omega_0]$, where $\widetilde{M}_{\underline{\alpha}} = (\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})$. Similarly, if the LMIs (23), (24) are satisfied, then by [13, Theorem 2] (applied with $\eta = 1$, $\gamma_1 = 0$, $\gamma_2 = 1/\omega_0$) we have $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega)) \leqslant 1$ for every $\omega \in [\omega_0, \infty]$. Combining both, we have $\mu_{\mathbf{\Delta}} \leqslant 1$, which gives robust stability of $(\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})$ over $[-1, 1]^m$. Hence

$$\alpha\left(\widetilde{A} - \underline{\alpha} I + \widetilde{B}\widetilde{\Delta}(I - \widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C}\right) < 0$$

for every $\widetilde{\delta} \in [-1, 1]^m$, and therefore $\alpha\left(\widetilde{A} + \widetilde{B}\widetilde{\Delta}(I - \widetilde{D}\widetilde{\Delta})^{-1}\widetilde{C}\right) < \underline{\alpha}$ for every $\widetilde{\delta} \in [-1, 1]^m$. Hence by Lemma 2, $\alpha\left(A + B\Delta(I - D\Delta)^{-1}C\right) < \underline{\alpha}$ for every $\delta \in \mathbf{\Delta}$. $\square$

Note that (21), (22) are coupled through F, G, and (23), (24) are coupled through $\widetilde{F}, \widetilde{G}$, but both blocks can be checked independently. In particular, if the first one fails, then we do not have to check the second one in order to reach our decision:

$$\mathscr{P}_{\mathrm{LMI}} : \begin{cases} \text{if (21) – (24) hold for } \mathbf{\Delta} \text{ and } \underline{\alpha} \text{ then} & \texttt{pruning } \mathbf{\Delta} \\ \text{otherwise} & \texttt{not\_pruning} \end{cases} \tag{25}$$

We conclude with the following immediate consequence of Lemma 4.

Lemma 5. *The procedure $\mathscr{P}_{\mathrm{LMI}}$ is a consistent pruning test.*

Proof. Since $\mathscr{P}_{\mathrm{LMI}}$ is less conservative than $\mathscr{P}_{\widetilde{\mu}}$, the result follows from the corresponding properties of $\mathscr{P}_{\widetilde{\mu}}$. $\square$

6. Frequency information

A particularity of the three pruning tests (18), (20), (25) is that they may provide useful information even when the decision is $\mathscr{P} = \texttt{not\_pruning}$.

Lemma 6. *Let $\mathbf{\Delta}$ be a subbox of $[-1, 1]^m$. Let $M = (A, B, C, D)$, $\widetilde{M}_{\underline{\alpha}} = (\widetilde{A} - \underline{\alpha} I, \widetilde{B}, \widetilde{C}, \widetilde{D})$ the shifted system* (15), (16) *for $\mathbf{\Delta}$, and suppose $\underline{\alpha} \geqslant 0$. Let ω_0 be a frequency such that $\mu_{\mathbf{\Delta}}(M(j\omega_0)) < 1$. Then also $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega_0)) < 1$.*

Proof. Suppose $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega_0)) \geqslant 1$, then also $\mu_{\mathbf{\Delta}}(\widetilde{M}(j\omega_0)) \geqslant 1$, because $\underline{\alpha} \geqslant 0$ and $\mu_{\mathbf{\Delta}}$ is decreasing with respect to $\underline{\alpha}$. Hence there exists a structured perturbation $\widetilde{\Delta}$ as in (3) such that $I - \widetilde{M}(j\omega_0)\widetilde{\Delta}$ is singular and $1/\overline{\sigma}(\widetilde{\Delta}) \geqslant 1$. Now we have two cases. Suppose first that $I - M(j\omega_0)K$ is singular. Then it suffices to observe that K is itself a Δ with $\delta \in \mathbf{\Delta}$, namely the midpoint $\delta_{\mathbf{\Delta}}$ of $\mathbf{\Delta}$. That implies $\mu_{\mathbf{\Delta}}(M(j\omega_0)) \geqslant 1/\overline{\sigma}(K) \geqslant 1$ due to $-1 \leqslant a_i < b_i \leqslant 1$.

The second case is when $I - M(j\omega_0)K$ is regular. In that case we may write $\widetilde{M}(j\omega_0) = F^{1/2}(I - M(j\omega_0)K)^{-1}M(j\omega_0)F^{1/2}$ and $\widetilde{\Delta} = F^{-1/2}(\Delta - K)F^{-1/2}$, hence

$$\begin{aligned} I - \widetilde{M}(j\omega_0)\widetilde{\Delta} &= I - F^{1/2}(I - M(j\omega_0)K)^{-1}M(j\omega_0)(\Delta - K)F^{-1/2} \\ &= F^{1/2}\left[I - (I - M(j\omega_0)K)^{-1}M(j\omega_0)(\Delta - K)\right]F^{-1/2} \\ &= F^{1/2}(I - M(j\omega_0)K)^{-1}\left[I - M(j\omega_0)K - M(j\omega_0)(\Delta - K)\right]F^{-1/2} \\ &= F^{1/2}(I - M(j\omega_0)K)^{-1}(I - M(j\omega_0)\Delta)F^{-1/2}. \end{aligned}$$

Therefore $I - M(j\omega_0)\Delta$ is singular. But $\overline{\sigma}(\widetilde{\Delta}) \leqslant 1$, hence $\widetilde{\delta} \in [-1, 1]^m$, hence $\delta \in \mathbf{\Delta}$, hence $\overline{\sigma}(\Delta) \leqslant \max\{|a_i|, |b_i|\} \leqslant 1$. That implies $\mu_{\mathbf{\Delta}}(M(j\omega_0)) \geqslant 1/\overline{\sigma}(\Delta) \geqslant 1$. $\square$

Remark 4. Obviously, the same holds for any box $\mathbf{\Delta}$ with $\underline{\alpha}$ and any of its subboxes $\mathbf{\Delta}'$ with $\underline{\alpha}' \geqslant \underline{\alpha}$. In other words, if $\widetilde{M}_{\underline{\alpha}}$ is the shifted system for $\mathbf{\Delta}$ with $\underline{\alpha}$, and $\widetilde{M}_{\underline{\alpha}'}$ the shifted system for $\mathbf{\Delta}'$ with $\underline{\alpha}' \geqslant \underline{\alpha}$, then $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega_0)) < 1$ implies $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}'}(j\omega_0)) < 1$.

This result allows the following improvement of the evaluations based on any of the μ-upper bounds. We represent it for $\widehat{\mu}_{\mathbf{\Delta}}$. Suppose we evaluate $\mathbf{\Delta} \in \mathscr{L}$ at the current lower bound $\underline{\alpha}$ found by **Trust**. Suppose the stability test $\widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}) \overset{?}{<} 1$ delivers the decision `not_pruning`, so that $\mathbf{\Delta}$ will have to be divided into two successor boxes $\mathbf{\Delta}'$, $\mathbf{\Delta}''$, which enter the list $\mathscr{L}$. Suppose however that we have a partial stability result in the sense that $\mu_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega)) \leqslant \widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\underline{\alpha}}(j\omega)) < 1$ say for all $\omega \in [0, \omega^\flat] \cup [\omega^\sharp, \infty]$, where $[0, \omega^\flat]$ is a low frequency band, $[\omega^\sharp, \infty]$ a high frequency band ($\omega^\flat < \omega^\sharp$). Then we store this information along with $\mathbf{\Delta}', \mathbf{\Delta}'' \in \mathscr{L}$, so that when these boxes turn up for evaluation, it will be clear that the stability test can be limited to the frequency band $[\omega^\flat, \omega^\sharp]$. Note that the tests (13), (18), (20) can all be restricted to frequency bands, which increases the chances for pruning.

Should the stability test for say the box $\mathbf{\Delta}'$ fail again, even though now restricted to $[\omega^\flat, \omega^\sharp]$, we may at least succeed to improve on the cutoff frequencies $\omega^\flat, \omega^\sharp$ in the sense that $\omega^\flat \to \omega^\flat + \Delta\omega^\flat$ gets bigger, $\omega^\sharp \to \omega^\sharp - \Delta\omega^\sharp$ gets smaller. In consequence the successors of $\mathbf{\Delta}'$ get an even smaller frequency band on which $\mu_{\mathbf{\Delta}} < 1$ has to be checked. Note also that if the lower bound $\underline{\alpha}$ changes between two such stages, this is no hindrance due to lemma 6, as the value increases, so that the information is not lost. For the LMI-based pruning test, the banded version takes the following form

$$\mathscr{P}_{\mathrm{LMI}}(\omega^\flat, \omega^\sharp) : \begin{cases} \text{if (27)-(28) hold} & \texttt{pruning } \mathbf{\Delta} \\ \text{otherwise} & \texttt{not\_pruning} \end{cases} \tag{26}$$

where

$$\begin{bmatrix} \widetilde{C}^H Z_1 \widetilde{C} & \widetilde{C}^H Z_1 \widetilde{D} - j\widetilde{C}^H Y_1 \\ * & \widetilde{D}^H Z_1 \widetilde{D} - Z_1 + jY_1 \widetilde{D} + * \end{bmatrix} + \begin{bmatrix} F \\ G \end{bmatrix} \begin{bmatrix} (j\omega^\flat I - \widetilde{A} + \underline{\alpha} I) & -\widetilde{B} \end{bmatrix} + * \prec 0 \tag{27}$$

$$\begin{bmatrix} \widetilde{C}^H Z_2 \widetilde{C} & \widetilde{C}^H Z_2 \widetilde{D} - j\widetilde{C}^H Y_2 \\ * & \widetilde{D}^H Z_2 \widetilde{D} - Z_2 + jY_2 \widetilde{D} + * \end{bmatrix} + \begin{bmatrix} F \\ G \end{bmatrix} \begin{bmatrix} (j\omega^\sharp I - \widetilde{A} + \underline{\alpha} I) & -\widetilde{B} \end{bmatrix} + * \prec 0 \tag{28}$$

Remark 5. In order to compute the low and high frequency bands $[0, \omega^\flat]$ and $[\omega^\sharp, \infty]$ it is helpful to start the search using information obtained from the local solver. Suppose the best lower bound is attained at $\underline{\alpha} = \operatorname{Re} \underline{\lambda}$ for $\underline{\lambda} = \underline{\alpha} + j\underline{\omega}$, then we search for $\omega^\flat$ in the range $[0, 0.9\,\underline{\omega}]$, and for $\omega^\sharp$ in the range $[2\,\underline{\omega}, \infty]$.

7. Experiments for minimum stability degree

In this section we present the results achieved by our branch and bound algorithm for program (5) applied to the 32 benchmarks of Table 1. The tests were realized using Matlab R2014b and on a 64-bit PC with 2.70GHz dual-core and 16, 0 Go RAM.

In Table 1 column n shows the number of states in (1), while the columns named structure allows to retrieve the uncertain structure $[r_1, \ldots, r_m]$ of (3). 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 $[1^3 2^4 1^4] = [1\,1\,1\,2\,2\,2\,2\,1\,1\,1\,1] = [r_1 \ldots r_{11}]$ in benchmark Hard-Disk-Driver 4. The number of decision variables in (21)-(24) and (27)-(28) is given in column n_{dec}.

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

TABLE 1. Benchmarks from [3] used for (5) and (6).

7.1. **Test with μ-upper bound pruning.** In Table 2, the third column $\underline{\alpha}$ gives the best (final) lower bound achieved by the local solver during branch and bound, where the algorithm uses $\mathscr{P} = \mathscr{P}_{\widehat{\mu}}$, while ranking $\mathscr{R}$ pushes those $\mathbf{\Delta} \in \mathscr{L}$ towards the end of the list, in which a $\underline{\delta}$ associated with the current $\underline{\alpha}$ occurs.

Column $\overline{\alpha}$ corresponds to the value $\overline{\alpha} = \underline{\alpha} + \epsilon = \underline{\alpha} + |\underline{\alpha}| \cdot tol$, where ϵ is scaled to the initial value $\underline{\alpha}$ such that the relative error is fixed to tol. On exit the branch and bound algorithm believes that the global maximum is $\underline{\alpha}$, and certifies that the true global maximum α^* lies between $\underline{\alpha}$ and $\overline{\alpha} = \underline{\alpha} + |\underline{\alpha}| \cdot tol = \underline{\alpha} + \epsilon$. The branch and bound algorithm converged in t^* seconds CPU.

Branch and bound was initialized by the value $\underline{\alpha}$ of column 3, computed in $\underline{t}$ seconds CPU, where the local solver was run as stand-alone. For comparison we also tested the algorithm with $\mathscr{P} = \mathscr{P}_{\widetilde{\mu}}$. This corresponds to an improved version of the method of [4]. Here the CPU is exceedingly long due to the strong conservatism of the pruning test, which has the effect that only very tiny boxes are pruned. We do not report the CPUs here.

7.2. **Test with frequency sweep.** Branch and bound was then tested with the frequency sweep $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon, \omega^\flat, \omega^\sharp)$ as in (26). In the case of benchmarks 20-22, the LMI solver failed in the computation of $\omega^\flat, \omega^\sharp$ due to the large number of decision variables (see Table 1, column n_{dec}). In the remaining cases, the search for $\omega^\flat$ and $\omega^\sharp$ turned out time consuming. For example, for the first benchmark, computing these frequencies for $\mathbf{\Delta} = [-1, 1]^5$ takes 20.22 respectively 204.87 seconds, leading to $\omega^\flat = 0.05$ and $\omega^\sharp = 0.1$. Subsequently, both $\mathbf{\Delta}' = \begin{bmatrix} [-1\ 0] \\ [-1\ 1]^4 \end{bmatrix}$ and $\mathbf{\Delta}'' = \begin{bmatrix} [0\ 1] \\ [-1\ 1]^4 \end{bmatrix}$ are pruned rapidly, because $\mathscr{P}(\mathbf{\Delta}', \underline{\alpha}, \epsilon, 0.05, 0.1) = \texttt{pruning}$ and $\mathscr{P}(\mathbf{\Delta}'', \underline{\alpha}, \epsilon, 0.05, 0.1) = \texttt{pruning}$. 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.

In the last test we run the algorithm with $\mathscr{P} = \mathscr{P}_{\widetilde{\mu}}$ using a frequency sweep. The frequencies $\omega^\flat$ and $\omega^\sharp$ are computed by bisection and evaluation of the H_∞-norm on the low- and high-frequency bands. We observed that evaluation of $\omega^\flat$ and $\omega^\sharp$ was considerably

$\sharp$	$\overline{\alpha}$	$\underline{\alpha}$	tol	t^*	$\underline{t}$	α_{ZM}	t_{ZM}	α_{LMI}	t_{LMI}
1	-1.2e-7	-1.2e-7	0.01	2.03	0.19	-1.2e-7	32.70	x	x
2	-1.7e-7	-1.7e-7	0.01	0.73	0.04	-1.7e-7	33.00	x	x
3	1.88e-2	1.86e-2	0.01	588.45	0.23	1.86e-2	90.25	x	x
4	-5e-7	-1e-6	0.50	53019.7	0.39	-1e-6	39.63	x	x
5	-9.9e-7	-1e-6	0.01	0.71	0.08	-1e-6	39.63	x	x
6	-9.9e-4	-1e-3	0.01	1.68	0.02	-1e-3	20.53	x	x
7	-9.9e-4	-1e-3	0.05	1.79	0.02	-1e-3	20.74	x	x
8	-4.9e-3	-1.65e-2	0.7	3.30e5	0.04	-1.65e-2	42.29	x	x
9	1.02e-2	8.9e-3	0.15	669.84	0.10	8.9e-3	159.91	x	x
10	-7.5e-7	-7.5e-7	0.01	4.12	0.29	-7.5e-7	73.86	x	x
11	-1e-7	-1e-7	0.01	5.35	0.29	-1e-7	74.63	x	x
12	-6e-6	-6e-6	0.01	0.67	0.02	-6e-6	26.03	x	x
13	-6e-6	-6e-6	0.01	0.25	0.02	-6e-6	26.20	x	x
14	2.72e2	2.66e2	0.02	3578.4	0.20	2.66e2	1252.5	x	x
15	-1.57	-1.60	0.02	1.52	0.06	-1.6026	80.40	x	x
16	-2.97e-1	-3e-1	0.01	0.57	0.04	-3e-1	51.41	x	x
17	-2.97e-1	-3e-1	0.01	0.72	0.02	-3e-1	50.95	x	x
18*	-5.3e-3	-5.4e-3	0.01	6.3	0.01	-5.4e-3	31.59	x	x
19	-3.65e-2	-3.68e-2	0.01	5.0	0.01	-3.68e-2	16.94	x	x
20	2.29e1	2.26e1	0.01	39.43	0.07	2.26e1	104.18	x	x
21	-4.9e-1	-5e-1	0.01	17.32	0.07	-5e-1	51.78	x	x
22	-4.9e-1	-5e-1	0.01	22.55	0.07	-5e-1	52.24	x	x
23+	-1.46e-2	-1.48e-2	0.01	0.1	0.06	-1.48e-2	7.05	x	x
24+	-1.46e-2	-1.48e-2	0.01	0.1	0.02	-1.48e-2	6.89	x	x
25*	-2.47e-1	-2.5e-1	0.01	0.24	0.01	-2.5e-1	12.83	-2.47e-1	0.25
26*	-2.42e-1	-2.5e-1	0.03	0.22	0.01	-2.5e-1	12.90	-2.47e-1	0.26
27	4.2e-5	3.9e-5	0.05	1557.7	0.02	3.9e-5	44.02	x	x
28	-2.55e-2	-2.69e-2	0.05	0.45	0.02	-2.69e-2	26.02	x	x
29	-2.65e-2	-2.69e-2	0.01	0.45	0.02	-2.69e-2	26.08	x	x
30+	2.04e-1	2.02e-1	0.02	0.24	0.01	2.02e-1	8.30	x	x
31+	-9.9e-2	-1e-1	0.01	0.16	0.01	-1e-1	6.91	x	x
32+	-9.9e-2	-1e-1	0.01	0.2	0.01	-1e-1	6.94	x	x

TABLE 2. The most rapid results obtained with branch and bound Algorithm 1 and its frequency sweep based versions for the benchmarks of Table 1. (*) signifies by frequency sweep version of $\mathscr{P}_{\tilde{\mu}}$. (+) signifies by frequency sweep version of $\mathscr{P}_{\tilde{\mu}}$ and $\mathscr{P}_{\text{LMI}}$ (mixed approach). Others are obtained using $\mathscr{P}_{\widehat{\mu}}$.

faster than with the LMI method, but $\widetilde{\mu}_{\mathbf{\Delta}}$ computed on $[\omega^\flat, \omega^\sharp]$ remained very conservative, so that pruning occurred only for tiny boxes. Except for benchmarks 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 $\omega^\flat$ and $\omega^\sharp$ by the H_∞-norm method reduced t^* considerably. We refer to this as the *mixed approach.*

Figure 3 illustrates different stages of the algorithm for benchmark $\sharp = 18$, where the pruning test $\mathscr{P} = \mathscr{P}_{\widehat{\mu}}$ is used. Dark gray squares are pruned, light gray squares are further divided. The global minimum is attained at $\delta_{wc} = (-1, 1)$ indicated by the star, and found by the local solver. The algorithm converges in 21.3 seconds. Figure 4 shows

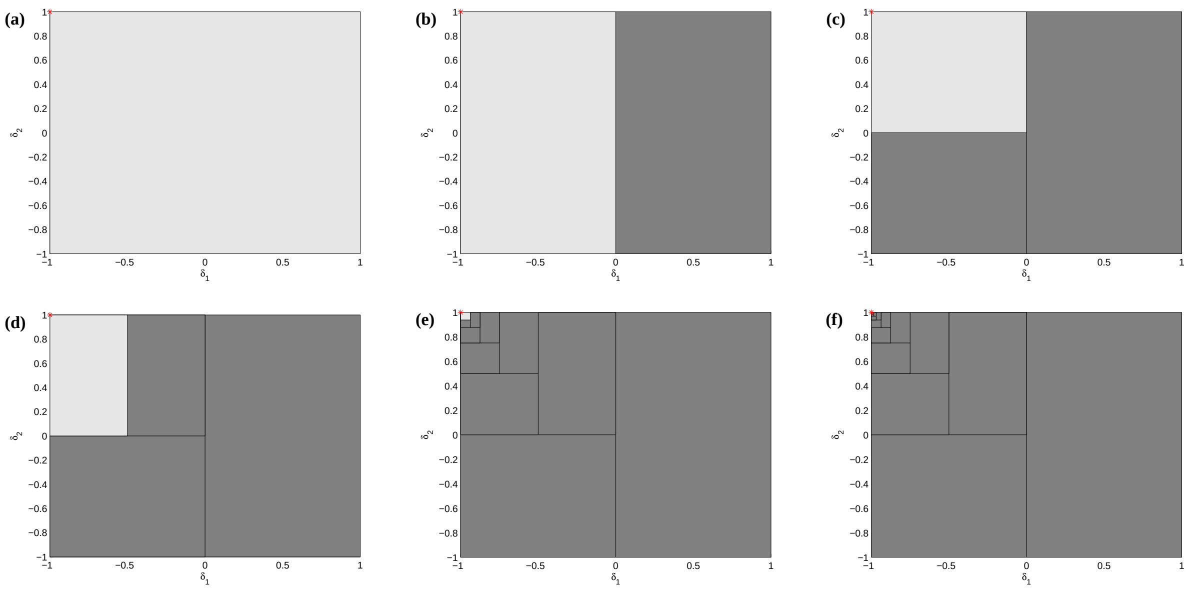

FIGURE 3. Pruning for benchmark 18 using $\mathscr{P}_{\widehat{\mu}}$. Dark squares are pruned, light squares are divided. The method converges in 21.3 seconds.

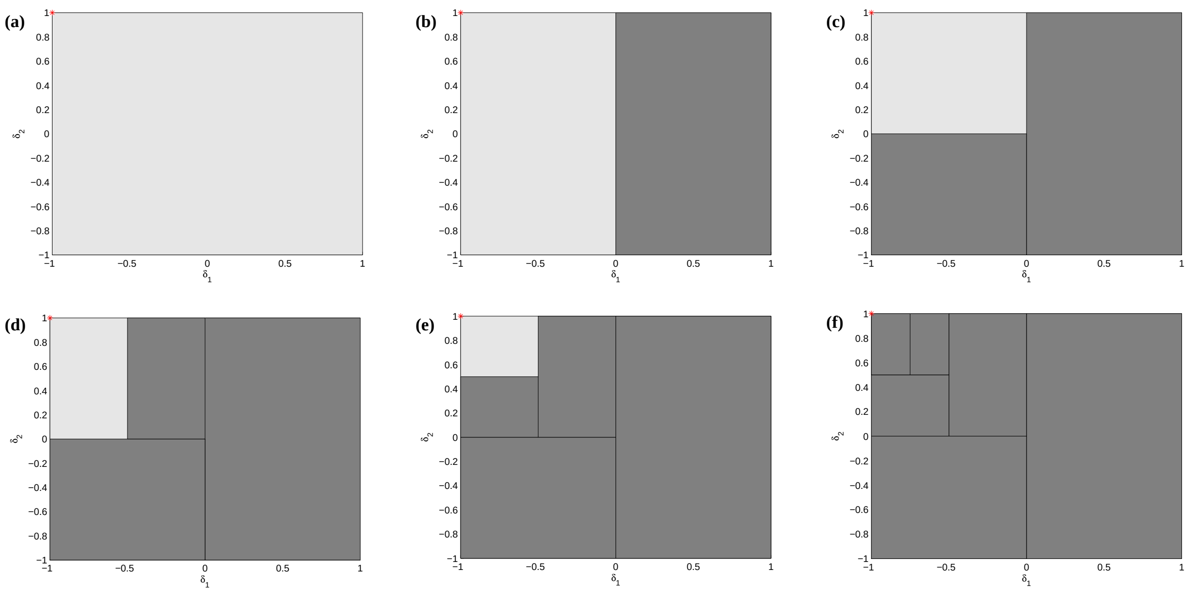

FIGURE 4. Pruning for benchmark 18 by frequency sweep (mixed approach). The algorithm converges after 6.3 seconds.

the corresponding steps based on the mixed approach. Here convergence occurred after 6.3 seconds.

For the first two benchmarks, t^* improved to 45.8 and 42.9 seconds. In our numerical tests, the mixed approach turned out fastest for $n_{dec} \leq 112$, except for very simple structured uncertainty (i.e. **Struct.**=1[1]), where the frequency sweep $\mathscr{P}_{\widetilde{\mu}}$ was the best.

7.3. **Test with Zheng method and polynomial optimization.** We also tested two alternative global optimizers, the methods of Zheng et al. [28] which is a probabilistic optimizer shown as Algorithm 2, and Lasserre's method [18].

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

1: **Initialize.** Choose initial $\alpha < \alpha^*$.
2: **Loop.** Compute
$$\alpha^+ = \frac{\int_{[f \geqslant \alpha]} f(x)\, d\mu(x)}{\mu[f \geqslant \alpha]}.$$
3: **Stopping.** If progress of α^+ over α is marginal stop, otherwise $\alpha \to \alpha^+$ and loop on with step 2.

Lasserre's method solves the problem (8) by a hierarchy of LMIs. Following Henrion et al. [17], robust stability of $A(\delta) - \overline{\alpha} I$ over $\mathbf{\Delta}$ is certified when the value of the following polynomial optimization problem is > 0:

$$\begin{array}{ll} \text{minimize} & \det(H(\delta)) \\ \text{subject to} & \delta \in \mathbf{\Delta} \end{array}$$

where $H(\delta)$ is the so-called Hermite-matrix of $A(\delta) - \overline{\alpha} I$. The method uses GloptiPoly, and Maple 14 to compute this determinant.

In table 2 the results α_{ZM} of the Zheng-method computed in t_{ZM} seconds and α_{LMI} of Lasserre's method are also reported. α_{LMI} improves over $\underline{\alpha}$ in cases 25 and 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 can not be taken into account, and the necessity to compute the determinant formally. For instance, for the benchmark 6 (DC motor 3), Maple produces 75 pages output for the determinant. In all other aspects the method is very promising.

Remark 6. For the benchmarks in Table 2 the branch and bound method certifies the lower bounds $\underline{\alpha}$ found by the local trust-region solver within a very mild tolerance ϵ. The results are in almost perfect agreement with α_{ZM} found by the probabilistic Zheng method (algorithm 2).

The CPUs of the local solver are orders of magnitude faster than those of the global techniques. The performance of the branch and bound technique hinges in large parts on the severeness of conservatism of the μ-upper bound pruning test. A large number of repetitions r_i in (3) usually leads to strong conservatism, and in consequence, to slow convergence.

8. Worst-case H_∞-performance

A branch and bound algorithm for problem (4) can be organized in much the same way as for (5) by using a similar pruning test. Analogous μ-upper bounds can be based e.g. on [29, Theorem 11.9], where parametric robust H_∞-performance is identified as a special case of parametric robust stability with regard to a suitably augmented system, see (1).

Theorem 3. *Let* $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ *be a stable, real-rational transfer function with* $n_1 + m_1$ *inputs and* $n_2 + m_2$ *outputs. Fix* $\beta > 0$. *For every* $\Delta \in \mathbb{C}^{m_1 \times m_2}$ *with* $\overline{\sigma}(\Delta) < 1/\beta$, *the linear fractional transform* $\mathcal{F}_u(\Delta, P) = P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12}$ *is well-posed, internally stable, and satisfies* $\|\mathcal{F}_u(P, \Delta)\|_\infty \leqslant \beta$ *if and only if*

$$\sup_{\omega \in \mathbb{R}} \mu_{\mathbf{\Delta}_e}(P(j\omega)) \leqslant \beta,$$

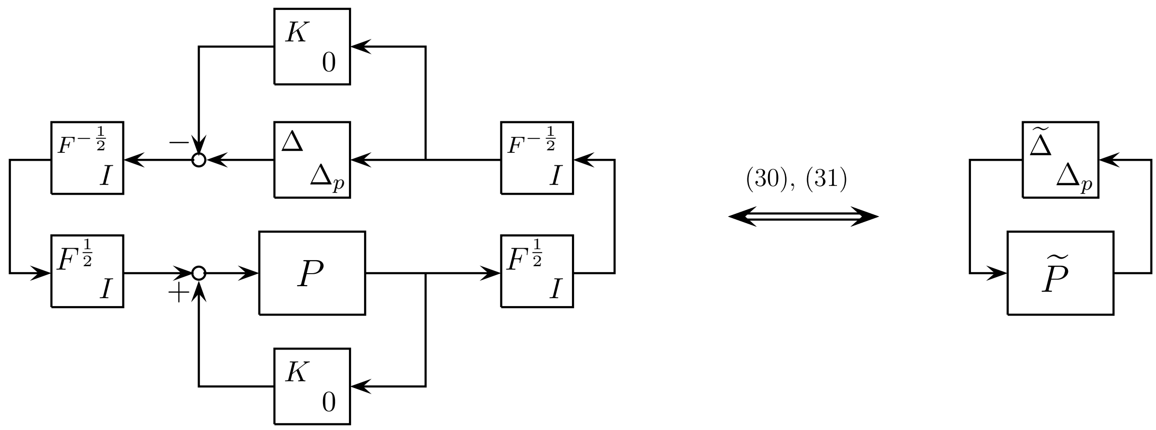

FIGURE 5. On the left $\Delta \leftrightarrow \delta \in \mathbf{\Delta} = \prod_{i=1}^m [a_i, b_i]$, on the right $\widetilde{\Delta} \leftrightarrow \widetilde{\delta} \in \tilde{\mathbf{\Delta}} = [-1,\, 1]^m$, while $\Delta_p \in \mathbb{C}^{n_1 \times n_2}$ is invariant. The two loops are equivalent under (30), (31). Note that $P(s) = C_0(sI - A_0)^{-1}B_0 + D_0$, and $\widetilde{P}(s) = \widetilde{C}_0(sI - \widetilde{A}_0)^{-1}\widetilde{B}_0 + \widetilde{D}_0$.

where

$$\mathbf{\Delta}_e := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_{\mathrm{p}} \end{bmatrix} : \Delta \in \mathbb{C}^{m_1 \times m_2}, \Delta_p \in \mathbb{C}^{n_1 \times n_2} \right\}.$$

□

For the proof see [29]. In the following we use the last theorem and also the following result, which is proved in [29]:

Lemma 7.

$$\max_{\overline{\sigma}(\Delta) \leqslant 1} \|\mathcal{F}_u(P, \Delta)\|_\infty \leq \gamma \;\Leftrightarrow\; \max_{\omega \in [0,\infty]} \mu_{\mathbf{\Delta}_e}\left(\begin{bmatrix} P_{11}(j\omega) & P_{12}(j\omega) \\ P_{21}(j\omega)/\gamma & P_{22}(j\omega)/\gamma \end{bmatrix} \right) \leqslant 1. \tag{29}$$

□

Based on these results, we now propose a variant of the branch and bound algorithm 1 for program (4). The algorithm is initialized with $\gamma = \underline{h}(1 + tol)$, where $0 < tol \ll 1$ is the desired precision. A box $\mathbf{\Delta}$ is pruned if

$$\widehat{\mu}_{\mathbf{\Delta}_e}\left(\begin{bmatrix} \widetilde{P}_{11} & \widetilde{P}_{12} \\ \widetilde{P}_{21}/\gamma & \widetilde{P}_{22}/\gamma \end{bmatrix} \right) < 1.$$

If $\mathbf{\Delta}$ cannot be pruned, it is bisected and **Trust** is called to find a new local worst-case gain and its associated worst-case $\delta \leftrightarrow \Delta$. The final result is denoted by $\overline{h}$, computed in $\overline{t}$ seconds. According to the choice of $tol > 0$ we have $\underline{h} \leqslant h^* \leqslant \overline{h} = \underline{h}(1 + tol)$ on exit, so that for small tol the value h^* is tightly bound.

Remark 7. Due to $\Delta_p \in \mathbb{C}^{n_1 \times n_2}$, which does not depend on γ, the loop transformation (15), (16) is modified to

$$\begin{aligned} K_0 &= \mathrm{diag}(K, 0_{n_1 \times n_2}), \\ F_1 &= \mathrm{diag}(F, I_{n_1}), \\ F_2 &= \mathrm{diag}(F, I_{n_2}), \end{aligned} \tag{30}$$

and

$$\begin{aligned} \widetilde{A}_0 &= A_0 + B_0(I - K_0D_0)^{-1}K_0C_0, & \widetilde{B}_0 &= B_0(I - KD_0)^{-1}F_2^{1/2}, \\ \widetilde{C}_0 &= F_1^{1/2}(I - D_0K_0)^{-1}C_0, & \widetilde{D}_0 &= F_1^{1/2}D_0(I - K_0D_0)^{-1}F_2^{1/2}, \end{aligned} \tag{31}$$

where

$$A_0 = A,\ B_0 = \begin{bmatrix} B & B_w \end{bmatrix},\ C_0 = \begin{bmatrix} C \\ C_z \end{bmatrix},\ D_0 = \begin{bmatrix} D & D_{qw} \\ D_{zp} & D_{zw} \end{bmatrix},$$

and where all the involved matrix are given in (1), (15), (16). By inspecting Figure 5 one easily proves that

$$\operatorname{diag}(\widetilde{\Delta}, \Delta_p) = F_1^{-1/2}(\operatorname{diag}(\Delta, \Delta_p) - K_0)F_2^{-1/2}.$$

9. Experiments with worst-case H_∞

♯	**Benchmark**	n	**Structure**	$\underline{h}$	$\overline{h}$	$\overline{h}_{wc}$	tol	$\underline{t}$	$\overline{t}$
1	Beam1	11	$1^3 3^1 1^1$	1.71	1.74	1.70	0.02	0.4	412.9
2	Beam2	11	$1^3 3^1 1^1$	1.29	1.32	1.29	0.02	0.3	199.2
3	DC motor1	7	$1^1 2^2$	0.72	0.73	0.72	0.01	0.2	2.2
4	DC motor2	7	$1^1 2^2$	0.50	0.50	0.50	0.01	0.1	0.18
5	DVD driver1	10	$1^1 3^3 1^1 3^1$	45.45	47.72	45.46	0.05	0.2	42412.3
6	Four-disk system1	16	$1^1 3^5 1^4$	4.56	4.60	3.5	0.01	0.6	1.6
7	Four-disk system2	16	$1^1 3^5 1^4$	0.68	0.69	0.69	0.01	0.4	22.74
8	Four-tank system1	12	1^4	5.60	5.65	5.6	0.01	0.5	2.3
9	Four-tank system2	12	1^4	5.57	5.62	5.6	0.01	0.4	13.1
10	Hard disk driver1	22	$1^3 2^4 1^4$	7526.6	failed	Inf	-	2.1	-
11	Hard disk driver2	22	$1^3 2^4 1^4$	30e-3	31e-3	30e-3	0.03	0.2	784.4
12	Hydraulic servo1	9	1^9	1.17	1.20	1.17	0.03	0.3	31.3
13	Hydraulic servo2	9	1^9	0.70	0.72	0.70	0.03	0.3	1814.1
14	Mass-spring1	8	1^2	6.19	6.25	3.71	0.01	0.3	8.2
15	Mass-spring2	8	1^2	6.84	7.05	7.16	0.03	0.9	1069.9
16	Missile1	35	$1^3 6^3$	5.15	5.3	5.12	0.03	0.4	107.9
17	Missile2	35	$1^3 6^3$	1.82	1.92	1.83	0.05	0.2	168.5
18	Filter1	8	1^1	4.86	4.91	4.86	0.01	0.4	3.3
19	Filter2	8	1^1	2.64	2.66	2.63	0.01	0.3	3.2
20	Filter-Kim1	3	1^2	2.96	2.99	2.95	0.01	0.2	2.3
21	Filter-Kim2	3	1^2	2.79	2.82	2.79	0.01	0.1	1.2
22	Satellite1	11	$1^1 6^1 1^1$	0.16	0.17	0.16	0.01	0.4	99.9
23	Satellite2	11	$1^1 6^1 1^1$	0.15	0.16	0.15	0.03	0.2	803.4
24	Mass-spring-damper1	13	1^1	8.85	8.93	7.63	0.01	0.2	3.9
25	Mass-spring-damper2	13	1^1	1.65	1.66	1.65	0.01	0.1	4.63

Table 3. Benchmark problems for (4) and some results.

In table 3, $\underline{h}$ denotes the best lower bound of the worst-case gain found by **Trust** in $\underline{t}$ seconds CPU. Column $\overline{h}$ is the result obtained by branch and bound in $\overline{t}$ seconds CPU. Column $\overline{h}_{wc}$ is the worst-case gain upper bound obtained by the `wcgain` function of MATLAB [30]. As can be seen, with the exception of problems 6, 14 and 24, `wcgain` gives results close to those found by **Trust** and branch and bound. In order to achieve a compromise between the execution time of the branch and bound algorithm and its precision, we increased *tol* for example from 0.01 for the simple uncertainty structures to 0.05 for the more complex ones. Note that an uncertain structure (3) is the more complicated, the larger the numbers r_i of repetitions, as this renders the μ-based pruning tests sensibly more conservative.

10. Distance to instability

In this section we discuss a variant of the branch-and-bound algorithm for computation of the distance to instability (6), or *stability margin* d^*, of (1). If A is nominally stable, the margin could be defined as

$$d^* = \sup\{d : A(\delta) \text{ stable for all } |\delta|_\infty < d\}.$$

We work under the assumptions that A is nominally stable, that some $\delta_0 \in \mathbb{R}^m$ is known for which $A(\delta_0)$ is unstable, but that (1) is well-posed in the box $|\delta|_\infty \leqslant |\delta_0|_\infty$. In consequence (6) may also be represented as the minimization problem

$$d^* = \inf\{|\delta|_\infty : A(\delta) \text{ unstable}\}$$

and this can be written as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leqslant \delta_i \leqslant t, i = 1, \ldots, m \\ & \alpha(A(\delta)) \geqslant 0 \end{array} \tag{32}$$

with decision variable $(t, \delta) \in \mathbb{R}^{1+m}$. Due to the form (32) the problem is now a global minimization problem, which means that the terminology of the previous sections has to be adapted to this change of sign.

10.1. **Local solver.** We have once again our local solver **Trust** at our disposal, which computes a local minimum $\overline{d}$ of (32) that serves as an upper bound of the global minimum d^*, that is, $d^* \leqslant \overline{d}$. Note that **Trust** is in fact applied to the following penalized form of (32), which is amenable to the approach in [1, 3]:

$$\min\{\max\{t, k \max\{0, -\alpha(A(\delta))\}\} : (t, \delta) \in C\},$$

where the penalty constant $k > 0$ is suitably adapted, and where C represents the convex constraints in (32). In the branch and bound algorithm we re-evaluate $\overline{d}$ regularly on boxes representing subproblems, which occasionally leads to an improved upper bound. Running the local solver within $\mathbf{\Delta}$ also helps to rank subproblems in the list of doables. A parameter $\overline{\delta} \in [-\overline{d}, \overline{d}]^m$, where the current best value obtained by **Trust** is $\overline{d}$, is stored and named the incumbent.

10.2. **Subproblems.** Let $\overline{d}$ be the current best upper bound found by the local solver. Then subproblems are specified by axe-parallel subboxes of the box $[-\overline{d}, \overline{d}]^m$, and the current list of doables $\mathscr{L}$ contains finitely many subproblems. Every subproblem can be formally defined as

$$d^*(\mathbf{\Delta}) = \min\{|\delta|_\infty : \delta \in \mathbf{\Delta}, A(\delta) \text{ unstable}\}.$$

Suppose the current list is $\mathscr{L} = \{\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_r\}$, and suppose the local solver finds an improved upper bound $\overline{d}' < \overline{d}$ and incumbent $\overline{\delta}'$. Then we replace each element $\mathbf{\Delta}_i \in \mathscr{L}$ by $\mathbf{\Delta}_i' = \mathbf{\Delta}_i \cap [-\overline{d}', \overline{d}']^m$, because it is then no longer of interest to search for destabilizing δ's outside $[-\overline{d}', \overline{d}']^m$. We abbreviate this operation by $\mathscr{L} \to \mathscr{L}'$. It may occasionally reduce the length of the list, as some $\mathbf{\Delta}_i'$ may be empty, but its main benefit is that it makes the subproblems easier for pruning.

10.3. **Pruning test.** Let $\overline{d}$ be the current best upper bound, and let $\mathbf{\Delta}$ be a subbox of $[-\overline{d}, \overline{d}]^m$. As soon as it is known that $d^*(\mathbf{\Delta}) \geqslant \overline{d}$, it is not necessary to evaluate $\mathbf{\Delta}$, as no improvement over the current $\overline{d}$ could be found.

In order to provide a pruning test, we use once again μ-upper bounds. If the system (A, B, C, D) is robustly stable over $\mathbf{\Delta}$ and if a tolerance $\epsilon > 0$ is allowed, we know that

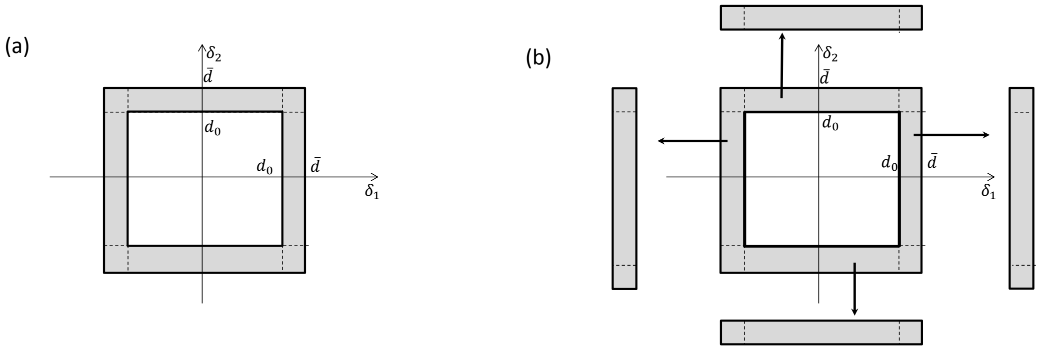

FIGURE 6. The hollow hyperbox $[-\overline{d}+\epsilon,\ \overline{d}-\epsilon]^m \setminus [-\underline{d}_0,\ \underline{d}_0]^m$ shown for $m=2$ on the left is covered by $2m=4$ partly overlapping subboxes.

$\mu_{\mathbf{\Delta}}^{-1} \leq \overline{d}-\epsilon$. This means that no $\delta \in \mathbf{\Delta}$ can improve over the incumbent $\overline{\delta}$ within that tolerance ϵ, and $\mathbf{\Delta}$ can therefore be pruned.

In order to decide, we apply the loop transformation (15), (16) to shift (A,B,C,D) with uncertainty on $\mathbf{\Delta}$ to the equivalent system $(\widetilde{A},\widetilde{B},\widetilde{C},\widetilde{D})$ with uncertainty $\widetilde{\delta} \in [-1,1]^m$. Then we check robust stability over $[-1,1]^m$ using any of the μ-upper bounds $\widehat{\mu}_{\mathbf{\Delta}}$, $\widetilde{\mu}_{\mathbf{\Delta}}$, or the LMI-variant, the justification coming again from Lemma 2. In other words, with the notation of section 3 we compute the test $\mathscr{P}(\mathbf{\Delta},0,0)$.

Naturally, we may as before store low and high frequency bands on which robust stability is already certified, so that the pruning test can be limited to the band $[\omega^\flat,\omega^\sharp]$. Here we write $\mathscr{P}(\mathbf{\Delta},0,0,\omega^\flat,\omega^\sharp)$. If a box in $\mathscr{L}$ is evaluated but cannot be pruned, then we divide it into two non-overlapping subboxes, where we cut one of the longest edges in two halves.

10.4. **Initialization of the list $\mathscr{L}$.** In contrast with section 5 we shall apply a different initialization. Suppose the list of doables $\mathscr{L}$ is first initialized with the sole box $[-\overline{d},\overline{d}]^m$. Let $\widetilde{M}_{\overline{d}-\epsilon}$ be the shifted system for this initial box with the tolerance $\epsilon > 0$ taken into account, then $\widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_{\overline{d}-\epsilon}) > 1$, because otherwise the procedure ends successfully after the first step. Now we define

$$d_0 = \sup\{d \geqslant 0 : \widehat{\mu}_{\mathbf{\Delta}}(\widetilde{M}_d) < 1\} \leqslant \overline{d}-\epsilon, \tag{33}$$

where $\widetilde{M}_d$ is the shifted system for the box $[-d,d]^m$. In other words, we apply a conservative test which assures robust stability of $A(\delta)$ over the smaller box $[-d_0,d_0]^m$. In order to certify $\overline{d}$ up to the tolerance ϵ, it remains to test robust stability over the region $R = [-\overline{d}+\epsilon,\overline{d}-\epsilon]^m \setminus [-d_0,d_0]^m$. As this set has a hole, it is not easily covered by axe-parallel boxes, in fact, one needs $2m$ such boxes, which are of the form $[-\overline{d}+\epsilon,-d_0] \times [-\overline{d}+\epsilon,\overline{d}-\epsilon]^{m-1}$ and $[d_0,\overline{d}-\epsilon] \times [-\overline{d}+\epsilon,\overline{d}-\epsilon]^{m-1}$, and similarly for the other coordinates. This means that the list of doables starts with $2m$ elements, and those are now partly overlapping. For a visualization of the two-dimensional case see Figure 6.

10.5. **Tolerances.** On exit the branch and bound algorithm certifies that $\underline{d} = \overline{d}-\epsilon < d^* \leqslant \overline{d}$. In order to obtain a scale-invariant procedure, we steer the method such that

$$\epsilon = \overline{d} \cdot tol,$$

where we want tol as small as possible and if possible, the same in each test. The actual values of tol we achieved are given in the forth column of Table 4.

10.6. **Running local solver at evaluation.** One could use the following procedure. Re-center $\mathbf{\Delta}$ into $\widetilde{\mathbf{\Delta}}$. Then maximize $\alpha(\widetilde{A}(\widetilde{\delta}))$ locally over $\widetilde{\mathbf{\Delta}}$ using **Trust**. If a value $\alpha \geqslant 0$ is found at $\widetilde{\delta} \in \widetilde{\mathbf{\Delta}}$, then the upper bound $\overline{d}$ is reduced as follows. Transform $\widetilde{\delta}$ back to $\delta \in \mathbf{\Delta}$, and put $\overline{d} = |\delta|_\infty$.

Algorithm 3. Branch and bound for d^* in program (6).

Parameters: Tolerance $\epsilon > 0$.
Subroutines: Local solver **Trust**, $\mathscr{P}$ for pruning, $\mathscr{R}$ for ranking.

Lower bound. Call local solver to compute upper bound $\overline{d}$ and incumbent $\overline{\delta}$. Compute d_0 according to (33). If $\overline{d} - d_0 \leqslant \epsilon$ stop. Otherwise continue.
Initialize list. Compute $2m$ boxes covering $[-\overline{d}+\epsilon, \overline{d}-\epsilon]^m \setminus [-d_0, d_0]^m$ and initialize list $\mathscr{L}$ with those. Call $\mathscr{R}$ to rank boxes in $\mathscr{L}$.
while $\mathscr{L} \neq \emptyset$ **do**
    Choose element $\mathbf{\Delta} \in \mathscr{L}$ ranked first for evaluation.
    Call pruning test $\mathscr{P}$ for $\mathbf{\Delta}$ with tolerance ϵ.
    if $\mathscr{P}(\mathbf{\Delta}, 0, 0) =$ `pruning` **then**
        Remove $\mathbf{\Delta}$ from $\mathscr{L}$
    else
        Remove $\mathbf{\Delta}$ and replace it by two successors $\mathbf{\Delta}'$, $\mathbf{\Delta}''$ in $\mathscr{L}$
        Call local solver within $\mathbf{\Delta}$. Update upper bound and incumbent and actualize list using operation $\mathscr{L} \to \mathscr{L}'$.
    end if
    Call $\mathscr{R}$ to update ranking of $\mathscr{L}$
end while
Return $\overline{d}$ and $\overline{\delta}$.

---

11. Experiments with distance to instability

In this section we report experiments on computation of the distance to instability. First our branch and bound method is evaluated, and then comparison is made with an alternative method from [6].

11.1. **Results for algorithm 3.** Table 4 reports the results of the branch and bound algorithm 3 for program (6) using the 32 examples of the test bench of Table 1. On exit, the branch and bound algorithm believes that the correct distance to instability is $d^* = \overline{d}$, and certifies that d^* satisfies $\underline{d} \leqslant d^* \leqslant \overline{d}$. The result is achieved in $\underline{t}$ seconds CPU. The local solver needs $\overline{t}$ seconds CPU to compute a first upper bound $\overline{d}$ in stand-alone mode, and this result is scarcely improved at the end of branch and bound, reported in column 2.

We have also tested the variant with frequency sweep. In test cases 6 and 8 it was not possible to compute $\omega^\sharp$, as even frequencies $\omega > 10^{20}$ did not lead to a stable high pass band. In all other cases the search for $\omega^\flat, \omega^\sharp$ was rapidly successful when the method of remark 5 was used in the following adapted form. If $\alpha(A(\overline{\delta})) = \operatorname{Re}\overline{\lambda}$ with $\overline{\lambda} = \overline{\alpha} + j\overline{\omega}$, then we search for $\omega^\flat$ in $[0, 0.9\,\overline{\omega}]$, and for $\omega^\sharp$ in $[2\,\overline{\omega}, 20\,\overline{\omega}]$. In several cases (6,7,8,12,13,16,17,25,26,28,29) the value $\overline{\omega} = 0$ was obtained, in which event $\omega^\flat = 0$ was chosen.

In a number of cases the algorithm turns much faster with the frequency sweep method to reach the same precision. In studies 1 and 20 the time was reduced by 50%. In study 3 the basic method had not even achieved 20% of its work, when the frequency based method already touched base.

$\sharp$	$\overline{d}$	$\underline{d}$	tol	$\overline{t}$	$\underline{t}$
1	3.7517	3.6767	0.020	0.27	540.10
2	4.3509	4.2639	0.020	0.18	855.60
3	0.1398	0.1370	0.020	0.36	7994.10
4	1.0329	0.7230	0.300	0.35	27471.40
5	1.2868	1.2740	0.010	0.30	118.75
6	3.3333	3.2667	0.002	0.28	212.8
7	3.3333	3.3008	0.010	0.29	42.26
8	3.0303	0.6061	0.80	0.90	7927.36
9	0.7484	0.7334	0.020	0.78	51201.10
10	1.2087	1.1845	0.020	0.98	18464.30
11	1.8597	1.5808	0.15	1.15	38125.24
12	6.6680	6.6346	0.005	0.18	25.67
13	6.6711	6.6378	0.005	0.16	35.14
14	0.8447	0.8194	0.030	1.06	712.1
15	1.4571	1.4134	0.030	1.15	2859.60
16	1.5000	1.47	0.020	0.36	2933.7
17	1.5000	1.4550	0.030	0.35	7140.50
18	1.0816	1.0762	0.005	0.08	17.83
19	1.3813	1.3122	0.05	0.54	487.48
20	0.2610	0.2580	0.005	1.26	448.89
21	1.5146	1.5070	0.005	1.29	515.61
22	3.1663	2.53	0.2	1.34	34087.6
23	1.3333	1.3177	0.012	0.24	13.29
24	1.3333	1.3177	0.012	0.13	12.96
25	3.3333	3.3073	0.008	0.06	55.06
26	3.3333	3.3301	0.001	0.15	1.50
27	0.9994	0.9867	0.013	0.41	2.99
28	2.4432	2.4188	0.01	0.59	53.2
29	2.4432	2.4188	0.01	0.52	54.7
30	0.5815	0.5741	0.013	0.22	28.72
31	1.2309	1.2153	0.013	0.19	17.18
32	1.3718	1.3557	0.012	0.25	16.76

TABLE 4. The results with Algorithm 3 for the benchmarks of Table 1.

Remark 8. Posterior inspection of those cases where locating a robustly stable high frequency band $[\omega^\sharp, \infty]$ failed indicates that this is *never* due to an active frequency $\omega = \infty$, contrary to the case of an active $\omega = 0$, which does indeed occur. A practical solution for these cases is to run the entire method on a band $[0, \omega^\sharp]$ only, where $\omega^\sharp$ is fixed throughout. In consequence the final certificate is then only obtained on $[0, \omega^\sharp]$.

For benchmarks 4,9,11 and 22 the execution time of branch and bound was excessively large even with a relatively large value *tol*. Not surprisingly CPUs are very sensitive to the choice of the tolerance *tol*. For instance, in test case 9 slightly relaxing the tolerance from $tol = 0.02$ to $tol = 0.05$ reduced execution time from $\underline{t} = 51201.1$ seconds to $\underline{t} = 1981.3$ seconds.

11.2. **Comparative results.** A new benchmark from our colleagues [6], available in [31], is also considered to illustrate the efficiency of Algorithm 3. The results are shown

♯	**Benchmark**	n	**Structure**	$\overline{d}$	$d_{\mathrm{F}}/\overline{d}$	$\underline{d}/\overline{d}$	$\overline{t}$	$\underline{t}$
33	Academic example	5	1^1	0.79	1	1	0.15	21.5
34	Academic example	4	1^3	3.41	1	0.98	0.13	8.8
35	Academic example	4	2^2	0.58	1	0.98	0.15	22.6
36	Inverted pendulum	4	1^3	0.84	1	0.98	0.22	1756.6
37	DC motor	4	$1^2\ 2^1\ 1^1$	1.25	1	0.98	0.19	137.5
38	Bus steering system	9	$2^1\ 3^1$	1.32	0.99	0.98	0.37	17.3
39	Satellite	9	$2^1\ 1^2$	1.01	0.99	0.98	0.3	1782.1
40	Bank-to-turn missile	6	1^4	0.60	0.99	0.98	0.17	2.9
41	Aeronautical vehicle	8	1^4	0.61	0.99	0.98	0.19	79.8
42	Four-tank system	10	1^4	6.67	0.99	0.99	0.27	4.7
43	Re-entry vehicle	6	$3^1\ 2^1\ 3^1$	6.20	1	0.99	0.44	7.0
44	Missile	14	1^4	7.99	1	0.99	0.25	3.6
45	Cassini spacecraft	17	1^4	0.06	1	0.97	0.13	90126.3
45	Mass-spring-damper	7	1^6	1.17	1	0.99	0.17	5.5
46	Spark ignition engine	4	1^7	1.22	0.99	0.98	0.41	9.4
47	Hydraulic servo system	8	1^8	1.50	0.99	0.98	0.41	689.4
48	Academic example	41	$2^1\ 1^3$	1.18	0.99	0.99	0.57	6.0
49	Drive-by-wire vehicle	4	$1^2\ 2^7$	1	0.99	0.77	0.96	31.5
50	Re-entry vehicle	7	$1^3\ 6^1\ 4^1$	1.02	0.98	0.97	0.42	152.4
51	Space shuttle	34	1^9	0.79	0.99	0.99	0.8	4 .1
52	Rigid aircraft	9	1^{14}	5.42	1	0.99	0.54	27042.7
53	Fighter aircraft	10	$3^1\ 15^1\ 1^6\ 2^1\ 1^1$	0.59	0.99	0.90	1.31	2860.3
54	Flexible aircraft	46	1^{20}	0.22	0.99	0.94	1.26	57.1
55	Telescope mockup	70	1^{20}	0.02	0.99	0.92	1.37	1385.1
56	Hard disk drive	29	$1^8\ 2^4\ 1^{11}$	0.82	1	0.95	2.87	358.5
57	Launcher	30	$1^2\ 2^2\ 1^2\ 3^1\ 6^1\ 1^{12}\ 2^8$	1.16	0.99	0.88	4.08	103.5
58	Helicopter	12	30^4	0.08	0.99	0.99	0.85	163.7
59	Biochemical network	7	39^{13}	1.4e-3	1	failed	36.76	-

TABLE 5. Benchmarks of [6, 31] and different estimations of d^*.

in Table 5. In the test cases 49 and 57 where the value of $\underline{d}/\overline{d}$ is relatively small, we could not implement the second step. In these cases numerical errors due to excessively large conditioning number of $I - KD$ and $I - DK$ (see (16)) occurred. The failure of case 59 is again due to numerical problem. This is explained because the uncertainty structure is very complex **Structure**$=39^{13}$. In table 5, the estimation of d^* found by the method of [6] is termed d_F.

12. CONCLUSION

We have presented a branch and bound scheme to compute typical quantities in the robustness analysis of dynamical systems with real uncertain parameters. Since eigenvalue functions of non-symmetric matrices do not easily lend themselves to interval arithmetics, we have developed problem-specific pruning tests, which avoid the explicit computation of upper bounds. A novel method to include frequency domain information in the pruning test often leads to an additional gain of speed. The method was tested on a bench of 116 challenging problems featuring systems with up to 70 states, up to 28 uncertain parameters, and with up to 20 repetitions. Heuristic bounds were computed speedily and reliably using a non-smooth local optimization method based on a bundle trust-region method from [1]. In all cases the branch and bound algorithm succeeded in certifying the bound delivered by the local solver, with CPUs ranging from seconds to days in some difficult cases.

Future work should extend the present approach to complex uncertainties, and use these methods in controller synthesis. Speeding up the pruning test remains a principal issue of the branch and bound approach. It may also be worthwhile to investigate whether interval arithmetic techniques can be extended to address eigenvalue functions like α of the H_∞-norm.

References

[1] P. Apkarian, D. Noll, L. Ravanbod. Nonsmooth bundle trust-region algorithm with applications to robust stability. *Set-Valued and Variational Analysis,* 24(1):2016,115-148.

[2] P. Apkarian, D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Automat. Control*, 51(1): 2006, 71-86.

[3] P. Apkarian, D. Noll, L. Ravanbod. Computing the structured distance to instability. *Proc. SIAM Conf. Control. Appl.,* Paris, 2015. 423–430. http ://dx.doi.org/10.1137/1.9781611974072.58

[4] V. Balakrishnan, S. Boyd. Global optimization in control system analysis and design. In C.T. Leonides, editor, *Control and Dynamic Systems: Advances in Theory and Applications*, vol. 53. Academic Press, New York, New York, 1992.

[5] V. Balakrishnan, S. Boyd, and S. Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *Int. J. of Robust and Nonlinear Control,* 1(4):1991,295-317.

[6] A. Fabrizi, C. Roos, J.M. Biannic. A detailed comparative analysis of lower bound algorithms. *Proc. European Control Conference* 2014, Strasbourg, France.

[7] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith. μ-Analysis and synthesis toolbox: User's Guide. The MathWorks, Inc., 1991.

[8] A. Bemporad, D. Mignone, M. Morari. An efficient branch and bound algorithm for state estimation and control of hybrid systems. *Proc. Europ. Control Conf.* 1999, 557 – 562.

[9] V. Blondel, C. T. Abdallah, and G. L. Heileman. Complexity issues and decision methods in control systems. *J. Symbolic Comp.*, 11:1995, 1–12.

[10] V. Blondel and J.N. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM J. Control Optim.*, 35(6):1997, 2118-2127.

[11] R. D. Braatz and P. M. Young and J. C. Doyle and M. Morari. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, 39(5):1994, 1000–1002.

[12] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proc. D (Control Theory Appl.)* 129(6):1982,242-250.

[13] 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.

[14] M.K.H. Fan, A.L. Tits, J.C. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Autom. Contr.* 36(3):1991,25-38.

[15] M.K.H. Fan. User's guide to MUSOL2: a package for computing the structured singular value or its upper bound.

[16] R.R.E. De Gaston, M.G. Safonov. Exact calculation of there multiloop stability margin. *IEEE Trans. Autom. Control.* 33(2):1988,156-171.

[17] D. Henrion, D. Arzelier, D. Peaucelle, J.-B. Lasserre. On parameter-dependent Lyapunov functions for for robust stability of linear systems. *43rd IEEE Conference on Decision and Control*, December 14-17, 2004 Atlantis, Paradise Island, Bahamas

[18] J.-B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* 11:2001, 796-817.

[19] D. Noll. Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality. *J. Optim. Theory Appl.*, 160(2):2014, 553-572.

[20] D. Noll, O. Prot, A. Rondepierre. A proximity control algorithm to minimize nonsmooth and non-convex functions. *Pacific J. Optim.*, 4(3):2008, 571-604.

[21] S. Poljak and J. Rohn. Checking robust nonsingularity is NP-hard. *Math. Control Signals Sys.*, 6:1993, 1–9.

[22] L. Ravanbod, D. Noll, P. Apkarian. Branch and bound algorithm for the robustness analysis of uncertain systems. *Proc. 16th IFAC Workshop on Control Applications of Optimization (CAO' 2015)*, october 6 - 9, 2015, Garmisch-Partenkirchen, 85 – 90.

[23] H. Romero-Monsivais, E. Rodriguez-Tello, G. Ramirez. A new branch and bound algorithm for the cyclic bandwidth problem. *Lect. Notes in Artificial Intell.* 7630:2012, 139 – 150.

[24] M.G. Safonov. Stability and robustness of multivariable feedback systems. MIT Press, Cambridge, 1980.

[25] V. Sakizlis, M.P. Kakalis, V. Dua, D. Perkins, E.N. Pistikopoulos. Design of robust model-based controllers via parametric programming. *Automatica,* 40(2):2004, 189-201.

[26] A. Sideris, R.S.S. Peña. Fast computation of the multivariable stability margin for real inter- related uncertain parameters. *IEEE Trans. Aut. Control*, 34(12):1989,1272-1276.

[27] O. Toker, H. Özbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. *Proc. ACC,* Seattle, 1995, 2525-2526.

[28] Q. Zheng and D. Zhuang. Integral global minimization: algorithms, implementations, and numerical tests. *Journal of Global Optimization*, 7:1995, 421 – 454.

[29] K. Zhou, K. Glover, J.C. Doyle. Robust and Optimal Control. Prentice Hall, 1996.

[30] Robust Control Toolbox 5.0. MathWorks, Natick, MA, USA, Sept 2013.

[31] SMAC Toolbox, ONERA 2012-15, `http://w3.onera.fr/smac`