Branch and Bound Algorithm for Computing the Minimum Stability Degree of Parameter-dependent Linear Systems*

V. Balakrishnan†, S. Boyd†and S. Balemi‡

November 18, 1991

Abstract

We consider linear systems with unspecified parameters that lie between given upper and lower bounds. Except for a few special cases, the computation of many quantities of interest for such systems can be performed only through an exhaustive search in parameter space. We present a general branch and bound algorithm that implements this search in a systematic manner and apply it to computing the minimum stability degree.

1 Introduction

1.1 Notation

$\mathbf{R}$ ($\mathbf{C}$) denotes the set of real (complex) numbers. For $c \in \mathbf{C}$, Re c is the real part of c. The set of $n \times n$ matrices with real (complex) entries is denoted $\mathbf{R}^{n\times n}$ ($\mathbf{C}^{n\times n}$). P^T stands for the transpose of P, and P^*, the complex conjugate transpose. I denotes the identity matrix, with size determined from context. For a matrix $P \in \mathbf{R}^{n\times n}$ (or $\mathbf{C}^{n\times n}$), $\lambda_i(P)$, $1 \leq i \leq n$ denotes the ith eigenvalue of P (with no particular ordering). $\sigma_{\max}(P)$ denotes the maximum singular value (or spectral norm) of P, defined as

$$\sigma_{\max}(P) = \max_{1\leq i\leq n} \sqrt{\lambda_i(P^*P)}.$$

*This work was supported in part by the Air Force Office of Scientific Research under contract AFOSR 89-0228A and by the National Science Foundation under grant ECS 85-52465-A4.

†Information Systems laboratory, Department of Electrical Engineering, Stanford University, Stanford CA 94305.

‡Institut für Automatik, ETH-Zentrum, 8092 Zurich.

$\mathrm{SD}(P)$ denotes the *stability degree* of $P \in \mathbf{R}^{n\times n}$, defined as

$$\mathrm{SD}(P) = -\max_{1\le i\le n} \mathrm{Re}\ \lambda_i(P).$$

P is *stable* if $\mathrm{SD}(P) > 0$, *unstable* otherwise.

The stability degree determines the slowest decay rate of any solution of $\dot{x} = Px$:

$$\mathrm{SD}(P) = \inf_{x_0\in\mathbf{R}^n} \left\{ \liminf_{t\to\infty} \frac{-\log\|x(t)\|}{t} \;\middle|\; \dot{x} = Px,\ x(0) = x_0 \right\}.$$

Thus, P is stable if and only if all solutions of $\dot{x} = Px$ decay to zero as $t \to \infty$.

1.2 A Standard Form for a Parameter-Dependent Linear System

We consider the family of linear time-invariant systems described by

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad x(0) = x_0, \\ y &= Cx + Du, \\ u &= \Delta y, \end{aligned} \tag{1}$$

where $x(t) \in \mathbf{R}^n$, $u(t), y(t) \in \mathbf{R}^p$, and A, B, C and D are real matrices of appropriate sizes. Δ is a diagonal *perturbation matrix*. In the sequel, we will assume that Δ is parametrized by a vector of parameters $q = [q_1, q_2, \ldots, q_m]$, and is given by

$$\Delta = \mathrm{diag}(q_1 I_1, q_2 I_2, \ldots, q_m I_m), \tag{2}$$

where I_i is an identity matrix of size p_i. Of course, $\sum_i^m p_i = p$. We will also assume that q lies in a rectangle $\mathcal{Q}_{\mathrm{init}} = [l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_m, u_m]$. A block diagram of the above family of linear systems is given in figure 1.

For future reference, we define

$$H(s) = C(sI - A)^{-1}B + D,$$

which is the transfer matrix of the system from u to y. We will assume in the sequel that the realization $\{A, B, C, D\}$ is minimal.

We may now write down a state-space realization for the closed-loop system in figure 1:

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

for all Δ such that $(I - D\Delta)$ is invertible. We will use $\mathcal{A}(q)$ to denote the closed-loop system matrix, that is

$$\mathcal{A}(q) = (A + B\Delta(I - D\Delta)^{-1}C). \tag{3}$$

Note that the entries of $\mathcal{A}(q)$ are *rational functions* of the components of the parameter vector q. Conversely, given any $\mathbf{R}^{n\times n}$-valued function $\mathcal{A}(q)$ that has no singularities at $q = 0$, we can find A, B, C, D and Δ such that equation (3) holds, *i.e.*, we can cast the system $\dot{x} = \mathcal{A}(q)x$ in the standard form.

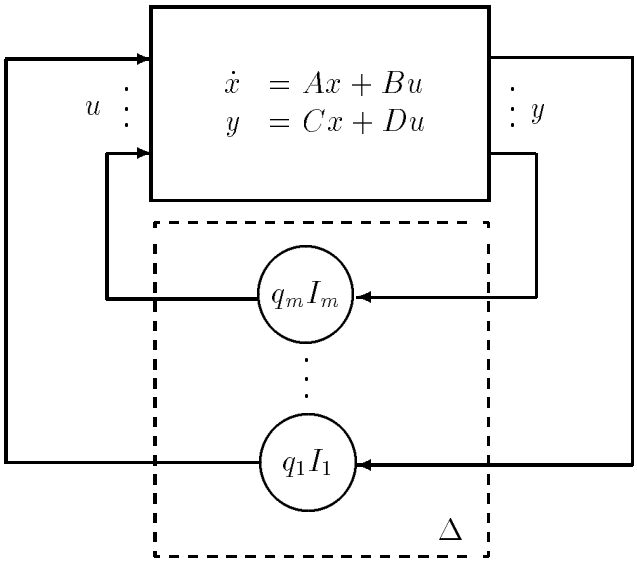

Figure 1: The standard form.

1.3 Some Important Questions

1.3.1 Well-posedness

Does the feedback system (1) make sense for all $q \in \mathcal{Q}_{\text{init}}$, that is, do we have

$$\det(I - D\Delta) \neq 0 \quad \text{for all} \quad q \in \mathcal{Q}_{\text{init}}, \tag{4}$$

or equivalently, does the rational function $\mathcal{A}(q)$ have no singularities in the rectangle $\mathcal{Q}_{\text{init}}$? If (4) holds we say that the system is *well-posed.*

1.3.2 Robust stability

If the feedback system (1) is well-posed, we can ask whether it is *robustly stable,* that is, whether we have

$$\mathcal{A}(q) \text{ is stable for all } q \in \mathcal{Q}_{\text{init}}. \tag{5}$$

The robust stability question can be adapted in several ways to form a *quantitative measure* of stability robustness. We now describe two of these measures.

1.3.3 Stability robustness margin

The *stability margin* SM of the system (1) is the largest factor by which the rectangle $\mathcal{Q}_{\text{init}}$ can be scaled about its center, while still guaranteeing well-posedness and robust stability. That is,

$$\text{SM}(\mathcal{A}, \mathcal{Q}_{\text{init}}) = \sup\{\gamma : \mathcal{A}(\gamma(q - q_0) + q_0) \text{ is well-posed and stable for all } q \in \mathcal{Q}_{\text{init}}\},$$

where q_0 is the center of the rectangle $\mathcal{Q}_{\text{init}}$. If the stability margin is much larger than one, we conclude that the uncertain system is not only robustly stable, but is "far away" from instability, in the sense that much larger parameter variations are needed to destabilize the system. Conversely if the stability margin is much smaller than one, we conclude that the system is not robustly stable, and indeed there are parameters near the center of the rectangle that result in an unstable system.

1.3.4 Minimum stability degree

If the parameter-dependent system (1) is well-posed, we define its *minimum stability degree* (MSD) as

$$\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}}) = \inf_{q \in \mathcal{Q}_{\text{init}}} \text{SD}\mathcal{A}(q).$$

Of course, the parameter-dependent system (1) is robustly stable if and only if its minimum stability degree is positive. Moreover, the MSD gives a guaranteed rate of decay of the solutions $x(t)$ of the state equations: for every value of the parameter vector q, the solutions $x(t)$ decay no slower than $e^{-(\text{MSD}(\mathcal{A},\mathcal{Q}_{\text{init}}))t}$. In fact,

$$\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}}) = \sup \left\{ \alpha : \lim_{t\to\infty} x(t)e^{\alpha t} = 0 \text{ whenever } \dot{x} = \mathcal{A}(q)x, q \in \mathcal{Q}_{\text{init}} \right\}.$$

Equivalently,

$$\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}}) = \inf_{x_0 \in \mathbf{R}^n,\ q \in \mathcal{Q}_{\text{init}}} \left\{ \liminf_{t\to\infty} \frac{-\log \|x(t)\|}{t} \;\middle|\; \dot{x} = \mathcal{A}(q)x,\ x(0) = x_0 \right\}.$$

1.4 Remarks

We note that the stability margin and minimum stability degree are not equivalent measures of stability robustness. Consider for example

$$\dot{x} = \begin{bmatrix} -\epsilon & q \\ -q & -\epsilon \end{bmatrix} x,$$

where ϵ is positive and small. No matter what interval the parameter q lies in, the stability margin is $+\infty$, and the minimum stability degree is ϵ. Thus, for ϵ small, this system has a large (indeed, infinite) SM but a small MSD.

Conversely consider

$$\dot{x} = -(1 + \epsilon - q)^{-1}x,$$

where ϵ is small and positive and $0 \le q \le 1$. For this system, the SM is $1 + 2\epsilon$, indicating that the parameter dependent system is "just barely" robustly stable. On the other hand, the MSD is $1/(1 + \epsilon)$. For small ϵ, therefore, all solutions decay not much slower than e^{-t}, which suggests the system is quite robust.

We also note that the MSD is a continuous function of the input data (A, B, C, D, l_i, u_i), whereas the SM is not [BKST89, BKST90].

1.5 Some Approaches

The questions described above have been extensively studied, and for some special cases, efficient methods are known. For systems with a single uncertain parameter, for example, the Evan's root locus can be used to ascertain robust stability or determine the stability margin or minimum stability degree [FPE86]. Less trivially, Kharitonov's theorem [Kha78, Bar84] gives a very efficient method for determining robust stability for the special case when the coefficients of the characteristic polynomial of $\mathcal{A}(q)$ are just the uncertain parameters q_i. Kharitonov's theorem has been extended to cover the case in which the characteristic polynomial is an affine function of q [BHL89, FB89].

In [ABJ75], Anderson *et al* observed that the robust stability question is *decidable,* which means that by evaluating a finite number of polynomial functions of the input data (the entries of A, B, C, D, and the l_i, u_i), we can determine whether the system is robustly stable. We can think of these *decision procedures* as generalizations of Routh's procedure for determining stability of the characteristic polynomial of a fixed matrix. It turns out, however, that these decision procedures involve an extraordinarily large number of polynomials, even for small systems with few parameters. Moreover the number of polynomials that need to be checked grows very rapidly (more than exponentially) with system size and number of parameters.

Many methods for assessing robust stability of parameter-dependent linear systems fall into two categories—those that *underestimate* robustness and those that *overestimate* robustness.

1.5.1 Pessimistic Methods

Pessimistic or conservative methods for robustness analysis underestimate robustness. These methods are usually based on some analytical result that describes sufficient (but not necessary) conditions for robust stability, for example, a small gain theorem, circle theorem, or Lyapunov theorem.

When a pessimistic method is applied to the robust stability problem, it can yield the answer "yes" (and indeed, in this case it supplies a proof or "certificate" supporting its answer) or "maybe", which means that the sufficient conditions were not met. Pessimistic methods can be used to give lower bounds on the SM or the MSD.

1.5.2 Optimistic Methods

Optimistic methods, on the other hand, overestimate robustness, often by restricting attention to a large but finite subset of $\mathcal{Q}_{\text{init}}$. One example is Monte Carlo methods: the minimum stability degree of a system is approximated by the smallest stability degree of $\mathcal{A}(q)$ over many values of q drawn from some distribution, often uniform, on $\mathcal{Q}_{\text{init}}$. Another class of optimistic methods uses (local) optimization to search for the "worst" parameter, that is, we find a local minimum of the function $\text{SD}(\mathcal{A}(q))$ over the rectangle $\mathcal{Q}_{\text{init}}$.

When an optimistic method is applied to the robust stability problem, it can yield the answer "no" or "maybe"; in the former case, it also provides a proof that robust stability does not hold, *i.e.*, a parameter value that results in an unstable system. Optimistic methods can be used to give upper bounds on the SM or the MSD.

1.5.3 Our Approach

We describe an approach that uses a pessimistic method to establish the robust stability of the *shifted* system $\dot{x} = (\mathcal{A}(q) + \alpha I)x$ for some α; this value of α then serves as a *lower* bound for $\mathrm{MSD}(\mathcal{A}, \mathcal{Q}_{\mathrm{init}})$. An *upper* bound for $\mathrm{MSD}(\mathcal{A}, \mathcal{Q}_{\mathrm{init}})$ is obtained by one of the optimistic methods above. However, these bounds may be unsatisfactory, in which case, a branch and bound technique [LW66, Bal68] is used to systematically improve the bounds. At each stage of the algorithm, guaranteed upper and lower bounds are available for $\mathrm{MSD}(\mathcal{A}, \mathcal{Q}_{\mathrm{init}})$.

The use of branch and bound algorithms for robustness analysis is not new. De Gaston and Safonov [GS88] use a branch and bound algorithm for computing the SM for systems with scalar uncertainties, *i.e.* , $p_i = 1$ in our notation (though they do not explicitly mention the term "branch and bound"). In [Sd86], this algorithm is extended to the case when the parameters may be interrelated (see also [SP89]). In [CEYB91], Chang *et al.* describe a similar branch and bound algorithm for computing the real structured singular value and the real multivariable stability margin. Vicino *et al.* [VTM90] use a branch and bound algorithm with geometric programming ideas to compute the SM. Demarco *et al.* [DBB90] use a branch and bound algorithm to study stability problems arising in power systems. Our algorithm is closer to those described in [DBB90], [GS88] and [SP89].

Before we describe the branch and bound algorithm, we make an important remark on the complexity of MSD computation. It is now known that the fundamental question of well-posedness (*cf.* equation (4)) is NP-hard in general [Roh89, Roh90, RP92, Dem92]. Roughly speaking, this means that for all known algorithms, the number of computations required to establish well-posedness, in the worst case, increases more than polynomially with the problem size m (which is the size of D and Δ). This makes it likely that *any* algorithm that computes the MSD to within some fixed accuracy also performs, in the worst case, computations that increase more than polynomially with the problem size. This is especially interesting in light of the fact that the number of branch and bound algorithm iterations, in the worst case, increases exponentially with m for a given accuracy (see the appendix). Thus it is likely that no algorithm would perform substantially better than a branch and bound algorithm on MSD computation.

In the following section, we describe the basic branch and bound algorithm in detail; we then use it to compute the MSD in subsequent sections.

2 The branch and bound algorithm

The branch and bound algorithm [LW66, Bal68] finds the (global) minimum of a function $f : \mathbf{R}^m \to \mathbf{R}$ over an m-dimensional rectangle $\mathcal{Q}_{\mathrm{init}}$.

For a rectangle $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ we define

$$\Phi_{\min}(\mathcal{Q}) = \min_{q \in \mathcal{Q}} f(q).$$

Then, the algorithm computes $\Phi_{\min}(\mathcal{Q}_{\text{init}})$ to within an absolute accuracy of $\epsilon > 0$, using two functions $\Phi_{\text{lb}}(\mathcal{Q})$ and $\Phi_{\text{ub}}(\mathcal{Q})$ defined over the set of rectangles contained in $\mathcal{Q}_{\text{init}}$, $\{\mathcal{Q} : \mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}\}$ (which, presumably, are easier to compute than $\Phi_{\min}(\mathcal{Q})$). These two functions must satisfy the following conditions.

(R1) $$\Phi_{\text{lb}}(\mathcal{Q}) \leq \Phi_{\min}(\mathcal{Q}) \leq \Phi_{\text{ub}}(\mathcal{Q}).$$

Thus, the functions Φ_{lb} and Φ_{ub} compute a lower and upper bound on $\Phi_{\min}(\mathcal{Q})$, respectively.

(R2) As the maximum half-length of the sides of $\mathcal{Q}$, denoted by size($\mathcal{Q}$), goes to zero, the difference between upper and lower bounds *uniformly* converges to zero, *i.e.,*

$$\forall\, \epsilon > 0\ \exists\, \delta > 0\ \forall\, \mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}\ \text{size}(\mathcal{Q}) \leq \delta \Longrightarrow \Phi_{\text{ub}}(\mathcal{Q}) - \Phi_{\text{lb}}(\mathcal{Q}) \leq \epsilon.$$

Roughly speaking, then, the bounds Φ_{lb} and Φ_{ub} become sharper as the rectangle shrinks to a point.

We now describe the algorithm. We start by computing $\Phi_{\text{lb}}(\mathcal{Q}_{\text{init}})$ and $\Phi_{\text{ub}}(\mathcal{Q}_{\text{init}})$. If $\Phi_{\text{ub}}(\mathcal{Q}_{\text{init}}) - \Phi_{\text{lb}}(\mathcal{Q}_{\text{init}}) \leq \epsilon$, the algorithm terminates. Otherwise we partition $\mathcal{Q}_{\text{init}}$ as a union of sub-rectangles as $\mathcal{Q}_{\text{init}} = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \ldots \cup \mathcal{Q}_N$, and compute $\Phi_{\text{lb}}(\mathcal{Q}_i)$ and $\Phi_{\text{ub}}(\mathcal{Q}_i)$, $i = 1, 2, ..., N$. Then

$$\min_{1 \leq i \leq N} \Phi_{\text{lb}}(\mathcal{Q}_i) \leq \Phi_{\min}(\mathcal{Q}_{\text{init}}) \leq \min_{1 \leq i \leq N} \Phi_{\text{ub}}(\mathcal{Q}_i),$$

so we have new bounds on $\Phi_{\min}(\mathcal{Q}_{\text{init}})$. If the difference between the new bounds is less than or equal to ϵ, the algorithm terminates. Otherwise, the partition of $\mathcal{Q}_{\text{init}}$ is further refined and the bounds updated.

If a partition $\mathcal{Q}_{\text{init}} = \cup_{i=1}^{N} \mathcal{Q}_i$ satisfies size($\mathcal{Q}_i$) $\leq \delta, i = 1, 2, ..., N$, then by condition (R2) above,

$$\min_{1 \leq i \leq N} \Phi_{\text{ub}}(\mathcal{Q}_i) - \min_{1 \leq i \leq N} \Phi_{\text{lb}}(\mathcal{Q}_i) \leq \epsilon;$$

thus a "δ-grid" ensures that $\Phi_{\min}(\mathcal{Q}_{\text{init}})$ is determined to within an absolute accuracy of ϵ. However, for the "δ-grid", the number of rectangles forming the partition (and therefore the number of upper and lower bound calculations) grows exponentially with $1/\delta$. The branch and bound algorithm applies a heuristic rule for partitioning $\mathcal{Q}_{\text{init}}$, which in most cases leads to a reduction of the number of calculations compared to the δ-grid. The heuristic is this: Given any partition $\mathcal{Q}_{\text{init}} = \cup_{i=1}^{N} \mathcal{Q}_i$ that is to be refined, pick a rectangle $\mathcal{Q}$ from the partition such that $\Phi_{\text{lb}}(\mathcal{Q}) = \min_{1 \leq i \leq N} \Phi_{\text{lb}}(\mathcal{Q}_i)$, and split it into two halves. The rationale behind this rule is that since we are trying to find the minimum of a function, we should concentrate on the "most promising" rectangle.

The general branch and bound algorithm

In the following description, k stands for the iteration index. $\mathcal{L}_k$ denotes the list of rectangles, L_k the lower bound and U_k the upper bound for $\Phi_{\min}(\mathcal{Q}_{\text{init}})$, at the end of k iterations.

2.0.4 The Algorithm

$k = 0$;
$\mathcal{L}_0 = \{\mathcal{Q}_{\text{init}}\}$;
$L_0 = \Phi_{\text{lb}}(\mathcal{Q}_{\text{init}})$;
$U_0 = \Phi_{\text{ub}}(\mathcal{Q}_{\text{init}})$;
while $U_k - L_k > \epsilon$, {
    pick $\mathcal{Q} \in \mathcal{L}_k$ *such that* $\Phi_{\text{lb}}(\mathcal{Q}) = L_k$;
    split $\mathcal{Q}$ *along one of its longest edges into* $\mathcal{Q}_I$ *and* $\mathcal{Q}_{II}$;
    $\mathcal{L}_{k+1} := (\mathcal{L}_k - \{\mathcal{Q}\}) \cup \{\mathcal{Q}_I, \mathcal{Q}_{II}\}$;
    $L_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{lb}}(\mathcal{Q})$;
    $U_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{ub}}(\mathcal{Q})$;
    $k = k + 1$;
}

The requirement that we split the chosen rectangle along a longest edge may seem mysterious at this point. This splitting rule controls the condition number of the rectangles in the partition; see the appendix.

At the end of k iterations, U_k and L_k are upper and lower bounds respectively for $\Phi_{\min}(\mathcal{Q}_{\text{init}})$. We prove in the appendix that since the bounds $\Phi_{\text{lb}}(\mathcal{Q})$ and $\Phi_{\text{ub}}(\mathcal{Q})$ satisfy condition (R2), $U_k - L_k$ is guaranteed to converge to zero, and therefore the branch and bound algorithm always terminates in finite number of steps.

It is clear that in the branching process described above, the number of rectangles grows with the number of iterations N. Thus, as iterations proceed, the number of rectangles might grow to be unmanageably large. However, under certain conditions, we may eliminate some rectangles from consideration; they may be *pruned* since $\Phi_{\min}(\mathcal{Q}_{\text{init}})$ cannot be achieved in them. This is done as follows.

Eliminate from list $\mathcal{L}_k$, *the rectangles* $\mathcal{Q} \in \mathcal{L}_k$ *that satisfy*

$$\Phi_{\text{lb}}(\mathcal{Q}) > U_k.$$

If a rectangle $\mathcal{Q} \in \mathcal{L}_k$ satisfies this condition, then $q \in \mathcal{Q} \Rightarrow f(q) > U_k$; however the minimum of $f(q)$ over $\mathcal{Q}_{\text{init}}$ is *guaranteed* to be less then U_k, and therefore cannot be found in $\mathcal{Q}$.

Though pruning is not necessary for the algorithm to work, it does reduce storage requirements. We will see in the examples we present that the algorithm often quickly prunes a large portion $\mathcal{Q}_{\text{init}}$, and works with only a small remaining subset.

3 Computation of the MSD

3.1 Computation of Upper and Lower Bounds for the MSD

With the system in the standard form, we now consider the problem of computing upper and lower bounds for the MSD. Following the notation used to describe the branch and bound algorithm, we have $f(q) = \mathrm{SD}(\mathcal{A}(q))$ and $\Phi_{\min}(\mathcal{Q}) = \mathrm{MSD}(\mathcal{A}, \mathcal{Q})$. We now need to compute a lower bound $\Phi_{\mathrm{lb}}(\mathcal{Q})$ and an upper bound $\Phi_{\mathrm{ub}}(\mathcal{Q})$ for $\mathrm{MSD}(\mathcal{A}, \mathcal{Q})$.

For simplicity, we first consider the case where $\mathcal{Q}$ is the cube $\mathcal{U} = [-1, 1]^m$. We then demonstrate how the problem of computation of the bounds for a general rectangle $\mathcal{Q}$ can be transformed into the simpler problem where $\mathcal{Q} = \mathcal{U}$.

3.1.1 Bounds for an m-dimensional cube $\mathcal{U}$

A simple upper bound on the MSD over the cube $\mathcal{U}$ is just the stability degree of the system evaluated at the *midpoint* of the cube. Thus:

$$\Phi_{\mathrm{ub}}(\mathcal{U}) = \mathrm{SD}(\mathcal{A}(0)) = \mathrm{SD}(A). \tag{6}$$

Computation of the lower bound is a little more involved; it is based on the application of the *small gain theorem* (SGT) [DV75]. SGT states that the system in figure 1 is well-posed and robustly stable (with $\mathcal{Q} = \mathcal{U}$) if $\|H\|_\infty < 1$, where

$$\|H\|_\infty = \sup_{\mathrm{Re}\ s>0} \sigma_{\max}(H(s))$$

is the $\mathbf{H}_\infty$ norm of the transfer matrix H. Thus, we have

$$\|H\|_\infty < 1 \implies \mathcal{A} \text{ is well-posed over } \mathcal{U} \text{ and } \mathrm{MSD}(\mathcal{A},\mathcal{U}) > 0.$$

To derive a better lower bound on $\mathrm{MSD}(\mathcal{A},\mathcal{U})$, we consider the exponentially time-weighted system

$$\begin{aligned} \dot{z} &= (A + \alpha I)z + Bu, \quad z(0) = x_0, \\ y &= Cz + Du, \\ u &= \Delta y. \end{aligned} \tag{7}$$

Note that the solutions of equations (7) and (1) are simply related by $z(t) = e^{\alpha t}x(t)$. Therefore,

$$\mathrm{MSD}(\mathcal{A} + \alpha I,\mathcal{U}) = \mathrm{MSD}(\mathcal{A},\mathcal{U}) - \alpha.$$

Thus we have

$$\mathrm{MSD}(\mathcal{A},\mathcal{U}) \geq \alpha, \quad \text{whenever } \|H\|_{\infty,\alpha} < 1,$$

where

$$\|H\|_{\infty,\alpha} = \sup_{\mathrm{Re}\ s>-\alpha} \sigma_{\max}(H(s))$$

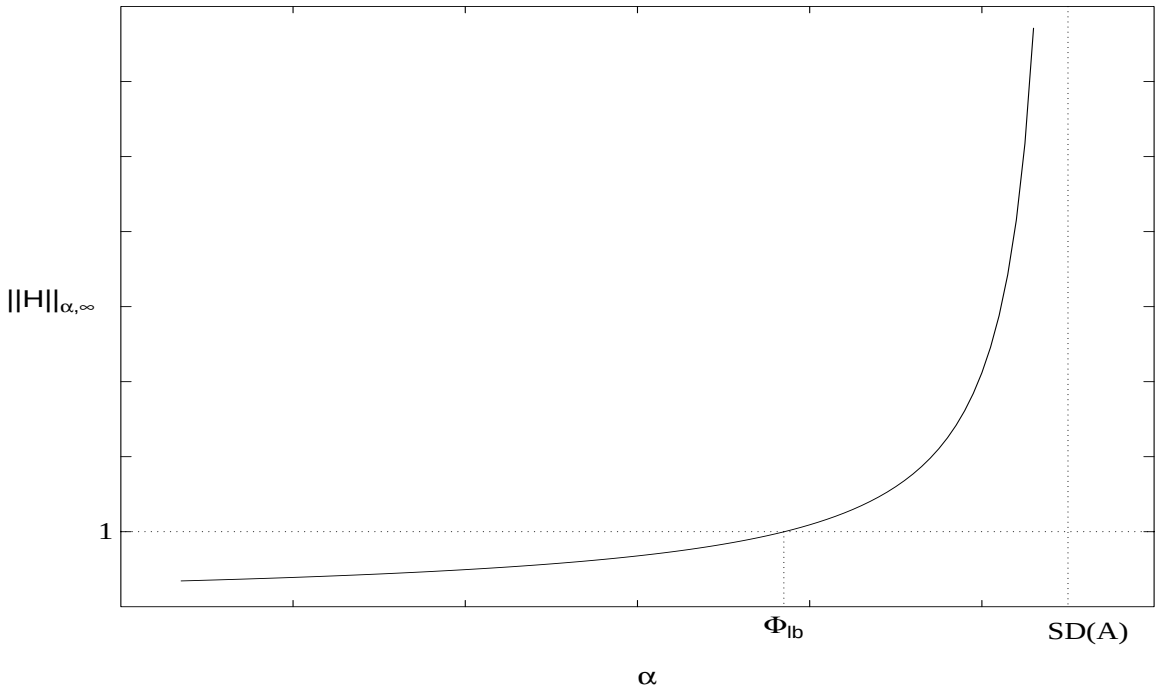

Figure 2: When $\sigma_{\max}(D) < 1$, a bisection method can be used to compute $\Phi_{\rm lb}$.

is the α-shifted $\mathbf{H}_\infty$ norm of H [BB91b]. Therefore, we define $\Phi_{\rm lb}(\mathcal{U})$ as

$$\Phi_{\rm lb}(\mathcal{U}) = \inf \{\alpha : \|H\|_{\infty,\alpha} \geq 1\} . \tag{8}$$

(Note that if $\|H\|_{\infty,\alpha} \geq 1$ for all α, then $\Phi_{\rm lb}(\mathcal{U}) = -\infty$.)

We now show how to compute $\Phi_{\rm lb}(\mathcal{U})$. We first observe that:

- $\|H\|_{\infty,\alpha}$ is a nondecreasing function of α.
- $\|H\|_{\infty,\alpha} = \infty$ for $\alpha \geq \mathrm{SD}(\mathcal{A})$.
- $\|H\|_{\infty,\alpha} \to \sigma_{\max}(D)$ as $\alpha \to -\infty$.

Obviously, $\Phi_{\rm lb}(\mathcal{U}) = -\infty$ if and only if $\sigma_{\max}(D) \geq 1$, in which case SGT cannot even establish well-posedness. However, $\sigma_{\max}(D) < 1$ ensures that $(I - D\Delta)$ is invertible for all $q \in \mathcal{U}$, and the situation shown in figure 2 obtains.

In [BBK89, BB90], it is shown that provided $\sigma_{\max}(D) < 1$ and $\alpha < \mathrm{SD}(A)$,

$$\|H\|_{\infty,\alpha} < 1 \qquad \Longleftrightarrow$$

$$M_\alpha = \begin{bmatrix} A + \alpha I + BR^{-1}D^TC & -BR^{-1}B^T \\ C^TS^{-1}C & -A^T - \alpha I - C^TDR^{-1}B^T \end{bmatrix} \text{ has no imaginary eigenvalues,}$$

where $R = (I - D^TD)$ and $S = (I - DD^T)$. Therefore, we may compute $\Phi_{\rm lb}(\mathcal{U})$ via a bisection on α, by checking whether M_α has any imaginary eigenvalues.

We note that the above procedure for computing $\Phi_{\rm lb}(\mathcal{U})$ is an application of the "shifted circle criterion" (Anderson and Moore [AM69]).

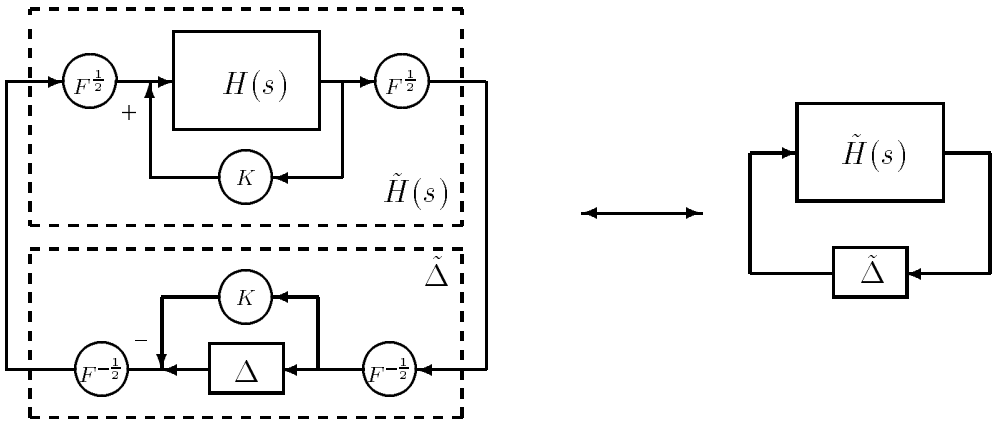

Figure 3: Loop Transformation.

3.1.2 Normalization of the parameter rectangle Q

We demonstrate how, given a general rectangle $\mathcal{Q}$, we may perform a loop transformation so that the transformed system has perturbations that lie in $[-1,1]^m$, so that then we may directly apply the results of the previous subsection. Figure 3 demonstrates the loop transformation, where the symbols $\tilde{H}(s)$ and $\tilde{\Delta}$ refer to the "new" system and the normalized perturbation. (See [DV75] for a complete discussion of loop transformations.)

The loop transformation can be interpreted as translating $\mathcal{Q}$ to the origin, and then scaling it to the hypercube $[-1,1]^m$.

$$K = \mathrm{diag}(\frac{u_1+l_1}{2}I_1, \frac{u_2+l_2}{2}I_2, \ldots, \frac{u_m+l_m}{2}I_m),$$
$$F = \mathrm{diag}(\frac{u_1-l_1}{2}I_1, \frac{u_2-l_2}{2}I_2, \ldots, \frac{u_m-l_m}{2}I_m)$$

are the *offset* and *scaling* respectively that accomplish this.

It is now easily verified that $\tilde{\Delta}$ has the form $\mathrm{diag}(\tilde{q_1}I_1, \tilde{q_2}I_2, \ldots, \tilde{q_m}I_m)$, where $\tilde{q}$ lies in the m-dimensional cube $[-1,1]^m$. It is also easily verified that a state-space representation of the loop-transformed system $\tilde{H}(s)$ is given by $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\}$, where

$$\begin{aligned} \tilde{A} &= A + B(I-KD)^{-1}KC, \quad & \tilde{B} &= B(I-KD)^{-1}F^{1/2}, \\ \tilde{C} &= F^{1/2}(I-DK)^{-1}C, & \tilde{D} &= F^{1/2}D(I-KD)^{-1}F^{1/2}. \end{aligned} \tag{9}$$

Performing this loop transformation immediately checks the well-posedness of the closed-loop system in figure 1 with $\Delta = K$: the system is well-posed for $\Delta = K$ if and only if $(I-KD)$ is invertible. We can also check how close the closed-loop system in figure 1 is to being ill-posed for $\Delta = K$ by checking the condition number of the matrix $(I-KD)$.

We finally summarize the computation of the lower bound $\Phi_{\mathrm{lb}}(\mathcal{Q})$.

1. Compute $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ and $\tilde{D}$ according to equation (9).

2. Check that $\tilde{R} = (I - \tilde{D}^T\tilde{D}) > 0$ and $\tilde{S} = (I - \tilde{D}\tilde{D}^T) > 0$. If either fails to hold, then our lower bound on the MSD is $-\infty$, *i.e.*, we cannot even be sure that the system is well-posed over $\mathcal{Q}$. Otherwise, we have established well-posedness of the feedback system in figure 1 for all $q \in \mathcal{Q}$.

3. If the feedback system is well-posed, then compute the lower bound as

$$\Phi_{\rm lb}(\mathcal{Q}) = \inf \left\{ \alpha : \begin{bmatrix} \tilde{A} + \alpha I - \tilde{B}\tilde{R}^{-1}\tilde{D}^T\tilde{C} & -\tilde{B}\tilde{R}^{-1}\tilde{B}^T \\ \tilde{C}^T\tilde{S}^{-1}\tilde{C} & -\tilde{A}^T - \alpha I + \tilde{C}^T\tilde{D}\tilde{R}^{-1}\tilde{B}^T \end{bmatrix} \text{ has imaginary eigenvalues} \right\}.$$

We show in the appendix that the bounds that we have derived above satisfy the second requirement (R2) listed at beginning of section 2, *i.e.*, that the difference between the two bounds converges uniformly to zero as the size of the parameter region goes to zero.

3.2 Remarks

The branch and bound algorithm outlined in §2 may now be directly used to compute the MSD. We observe the following:

- The algorithm first tries to establish well-posedness, and then goes on to compute the MSD. To see this, we note that during the kth iteration, the branch and bound algorithm splits a rectangle $\mathcal{Q}$ which satisfies $\Phi_{\rm lb}(\mathcal{Q}) = L_k$, where L_k is the lower bound on the MSD. Therefore, if $L_k = -\infty$, the rectangles which are split are those over which the algorithm has been unable to establish well-posedness. And the algorithm continues to concentrate on such rectangles until it either establishes well-posedness through determining a lower bound for the MSD that is greater than $-\infty$ or finds a parameter value $q_{\rm ill\_posed}$ such that the feedback system is not well-posed for $q = q_{\rm ill\_posed}$.

- If $\Phi_{\rm lb}(\mathcal{Q}) > -\infty$, the algorithm also provides a "certificate" that *proves* that $\Phi_{\rm lb}(\mathcal{Q})$ is a lower bound: it is shown in [BBK89] (see also [HB91, Wil73]) how to construct a quadratic positive definite Lyapunov function $V(x)$ that satisfies

$$\dot{V}(x) \leq -2\Phi_{\rm lb}(\mathcal{Q})V(x) \text{ for all } x \in \mathbf{R}^n,\ q \in \mathcal{Q}.$$

  This proves that $\mathrm{SD}(\mathcal{A}(q)) \geq \Phi_{\rm lb}(\mathcal{Q})$ for all $q \in \mathcal{Q}$. Thus the algorithm proves every lower bound on the MSD by "paving" the parameter space with quadratic Lyapunov functions.

- The algorithm provides a "bad" parameter value on exit: the parameter vector $q_{\rm bad}$ which achieves the upper bound on the MSD satisfies

$$\mathrm{SD}(\mathcal{A}(q_{\rm bad})) - \mathrm{MSD}(\mathcal{A}, \mathcal{Q}_{\rm init}) < \epsilon.$$

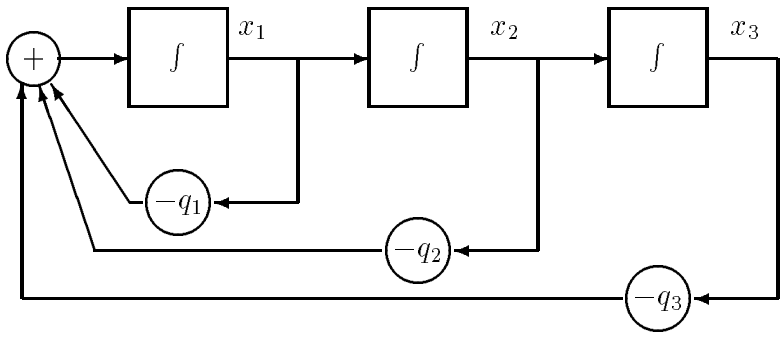

Figure 4: Family of systems whose characteristic polynomials lie in a rectangle.

4 Some examples

4.1 A rectangle of polynomials

We first consider a simple problem for which an analytical result allows us to compute the MSD exactly. We consider the system shown below in figure 4, with parameter ranges

$$2 \leq q_1 \leq 3, \qquad 3 \leq q_2 \leq 5, \qquad -1 \leq q_3 \leq 1.$$

Kharitonov's theorem [Kha78, Bar84] can be used to determine robust stability of this system. An extension due to Petersen [Pet90] shows that the MSD of this system is the minimum of the stability degrees of the eight systems where the parameters assume their extreme values, and is -0.2757.

Figure 5 shows the convergence of the upper and lower bounds, the total percentage pruned volume, that is, the total volume of the pruned rectangles divided by the volume of the original rectangle, and the total number of active rectangles, that is, the number of rectangles at the end of each iteration. We observe that after 50 iterations, 37 rectangles have been pruned corresponding to about 30% of $\mathcal{Q}_{\text{init}}$ in volume. Moreover, for any parameter vector lying in the pruned region, we have $\text{SD}(\mathcal{A}(q)) > U_{50} = -0.239$ (recall that U_k is the upper bound after k iterations). The algorithm takes about 215 iterations to return MSD $= -0.275$ to within an absolute accuracy of 0.001.

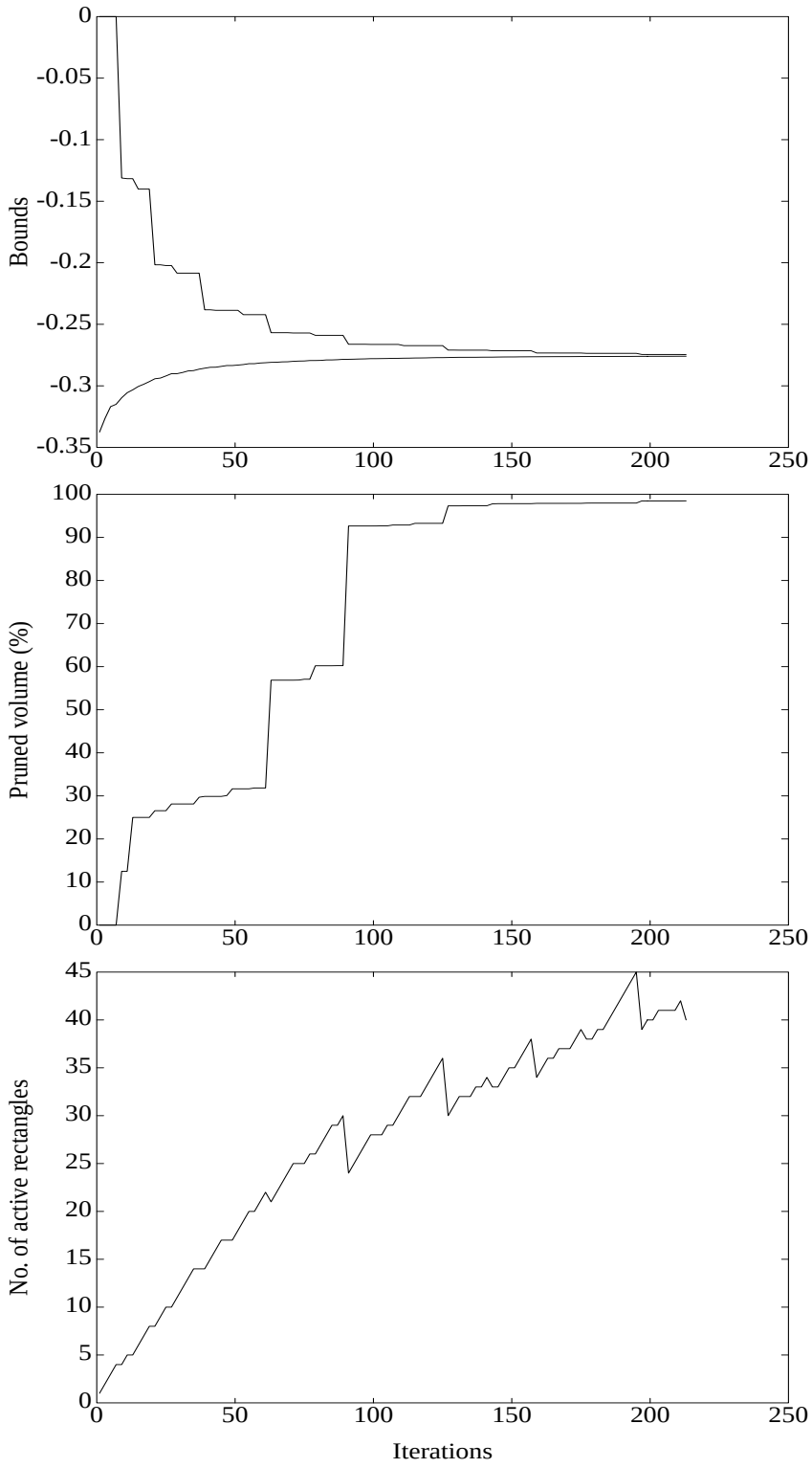

Figure 5: A rectangle of polynomials.

4.2 An interval matrix problem

Many researchers have considered questions of robust stability of "interval matrices," *i.e.,* matrices whose entries lie in intervals [Hei84, MK87, Xin87, Bia83, BH84, BFS88, DPA+88]. However there are no known analytical results that allow the computation of the MSD (or indeed, the determination of robust stability) for such systems, except in special cases.

We consider the family of 3×3 matrices lying in the "interval"

$$\begin{bmatrix} -1 & [1,4] & [0.5,1] \\ 0 & -2 & [2,3] \\ [-6,-3] & 1 & [-4,-3] \end{bmatrix}.$$

This can be expressed in our framework with:

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\Delta = \operatorname{diag}(q_1, q_2, q_3, q_4, q_5)$, with $1 \le q_1 \le 4$, $0.5 \le q_2 \le 1$, $2 \le q_3 \le 3$, $-6 \le q_4 \le -3$, $-4 \le q_5 \le -3$. The reader can easily construct the associated block diagram of the standard form.

Figure 6 gives a plot of the upper and lower bounds on the MSD, the pruned volume percentage and the number of active rectangles, versus iterations. We note that unlike the the previous example, the algorithm does not prune any volume at all until about 100 iterations. By about 1000 iterations, 82% of the volume of $\mathcal{Q}_{\text{init}}$ has been pruned and the difference between the upper and lower bounds on the MSD is down to 0.0155. The algorithm takes about a thousand more iterations to return an MSD of -0.148 to within an accuracy of 0.001.

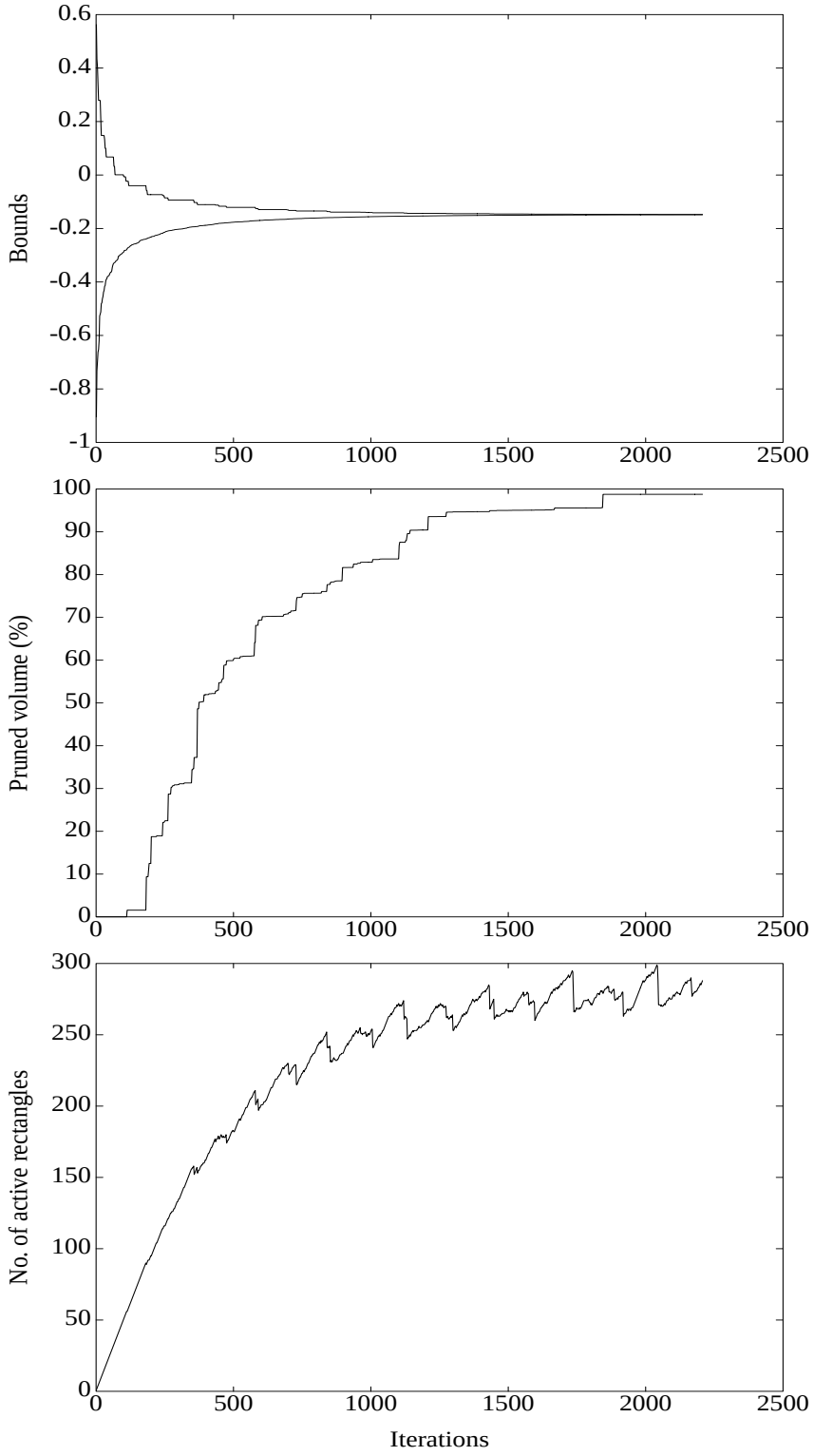

Figure 6: An interval matrix problem.

4.3 Matrices with entries that are rational functions of the parameters

We consider a family of systems described by

$$\dot{x} = \begin{bmatrix} \dfrac{q_2}{1+q_2} & 2 \\ \dfrac{q_2}{1+q_1} & \dfrac{q_1}{1+q_2^2} \end{bmatrix} x,$$

where $1 \leq q_1 \leq 2$ and $0 \leq q_2 \leq 0.5$.

The block diagram shown in figure 7 shows that this family can be described in our setup as follows.

$$A = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & -1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

and $\Delta = \text{ diag}(q_1, q_1, q_2, q_2, q_2, q_2)$.

Figure 8 shows the partition of rectangles at three different points during the course of the algorithm. At the end of 200 iterations, the algorithm guarantees that MSD cannot be achieved outside the regions shaded black. From the figure, it seems likely that the minimum is achieved at the lower right hand corner of the parameter region. However, our branch and bound algorithm eventually shows that this is not the case (it finds that the stability degree of the system is *lower* than at any other vertex for the point $q_1 = 2.00$, $q_2 = 0.0913$). This example demonstrates that there is no guarantee that a worst parameter lies at a vertex of $\mathcal{Q}_{\text{init}}$ (in fact, in general, there is no guarantee that it lies even on the boundary; it is quite easy to construct examples that show this).

Figure 9 shows the performance of the algorithm on this problem. It is interesting to note that the algorithm, once again, prunes a large fraction of $\mathcal{Q}_{\text{init}}$ very quickly, and spends the rest of the time refining the bounds on the remaining volume in parameter space. The algorithm returns an MSD of -2.015 after about 700 iterations to within an absolute accuracy of 0.001.

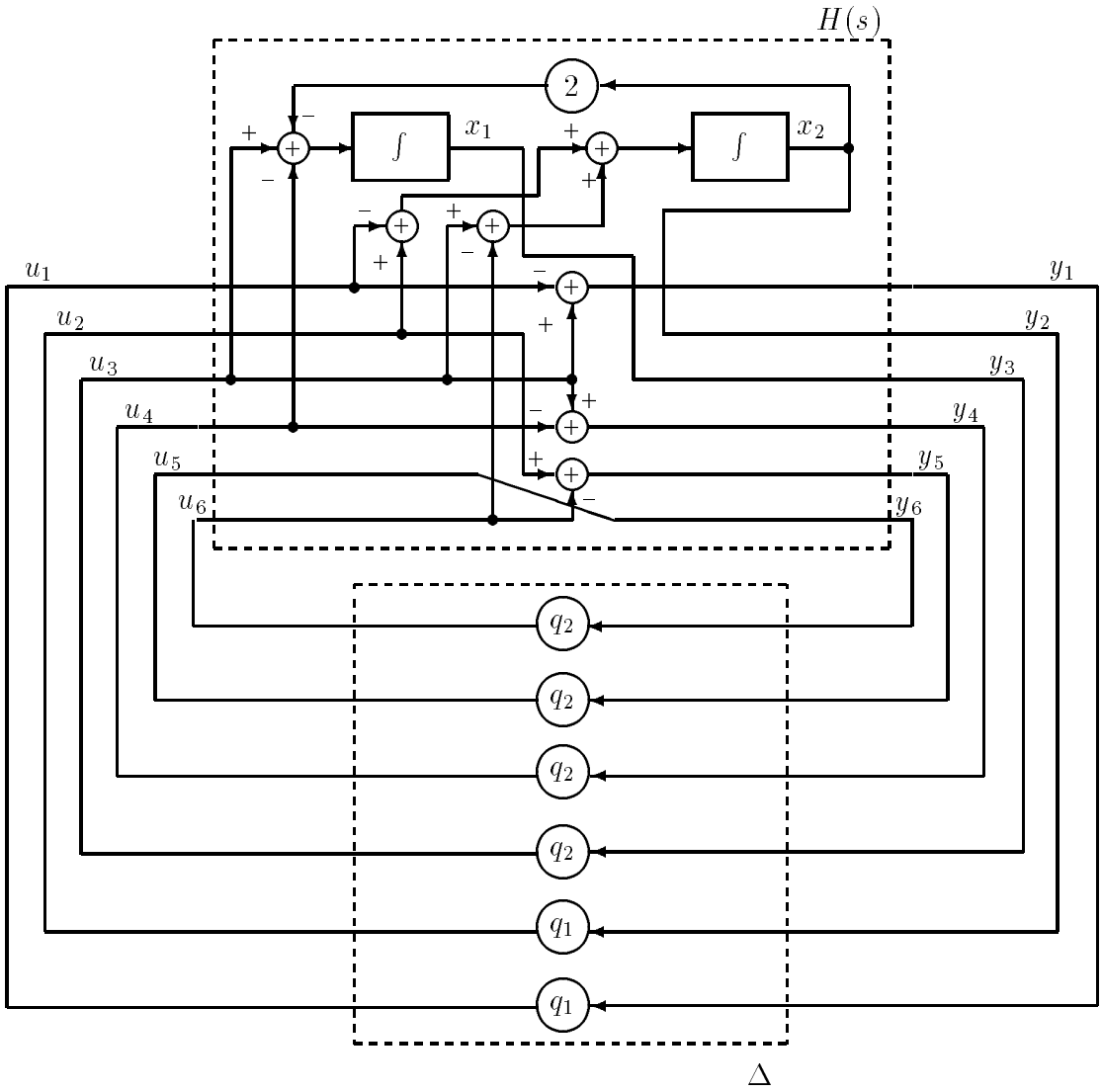

Figure 7: Block diagram of the standard form for a matrix whose entries are a rational function of the parameters.

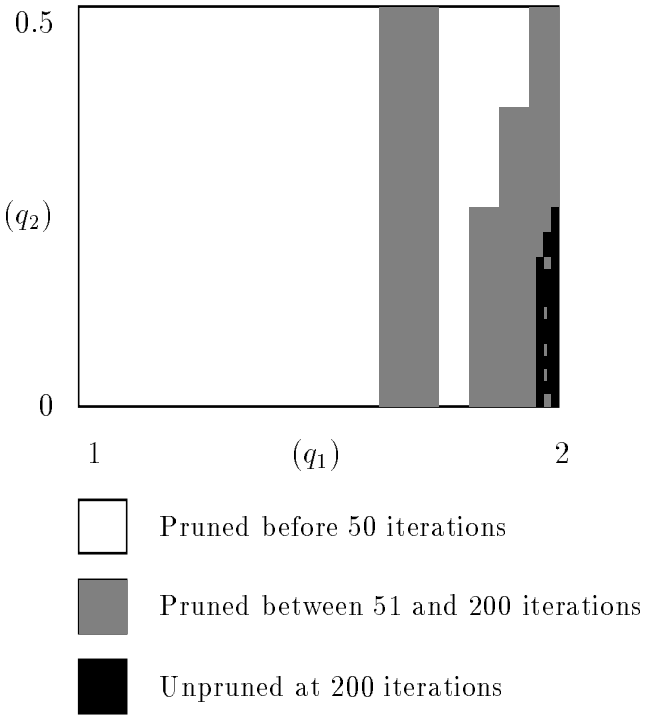

Figure 8: The partitioned parameter region at various stages of the algorithm.

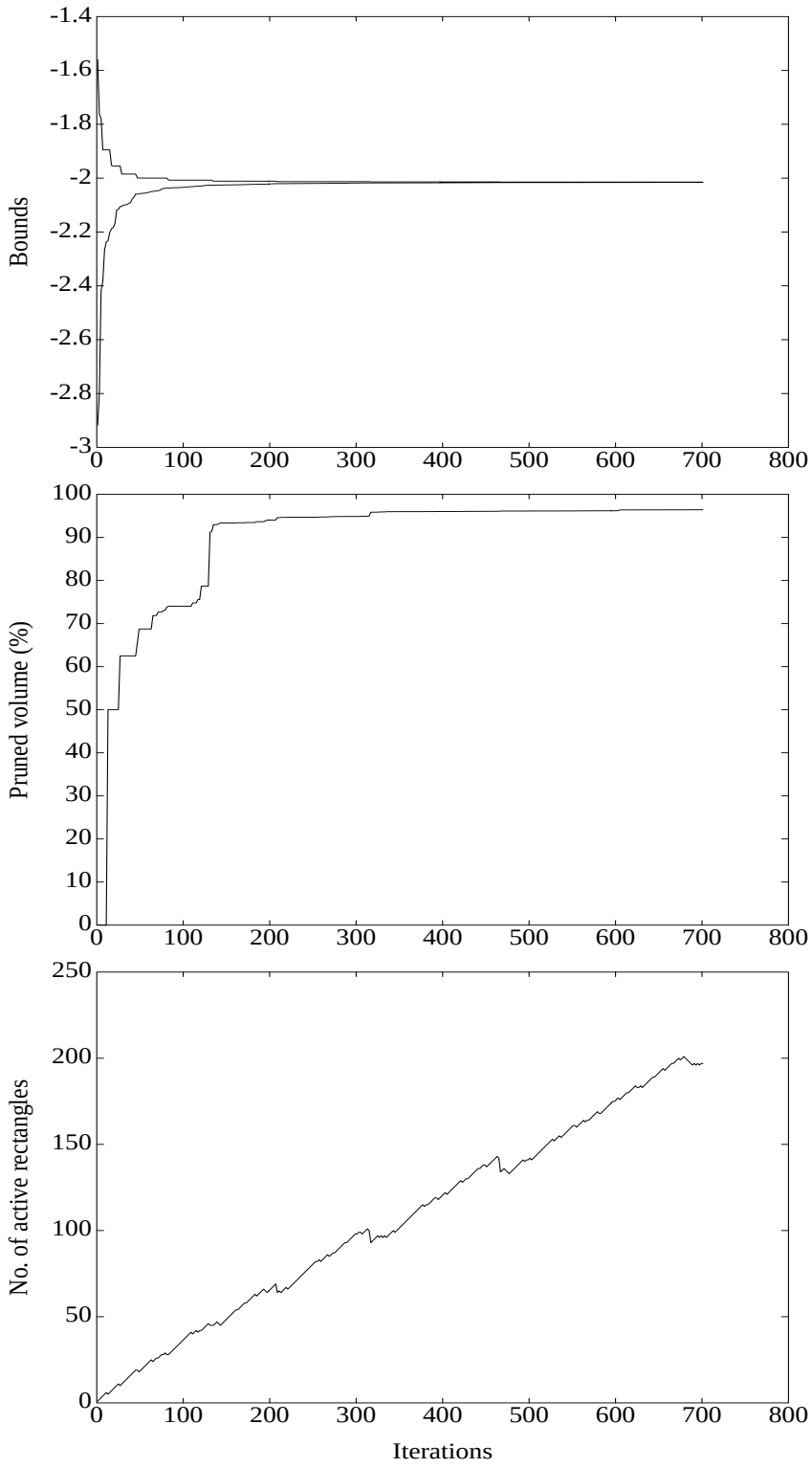

Figure 9: Matrices with entries that are rational functions of the parameters.

4.4 A mechanical example

We consider the system described by

$$M\ddot{x} + D\dot{x} + Kx = 0, \tag{10}$$

where $x(t) \in \mathbf{R}^n$ and $M = M_0 + \delta_1 M_1$, $D = D_0 + \delta_2 D_1$, $K = K_0 + \delta_3 K_1$ and $|\delta_i| \leq 1$, $i = 1, 2, 3$. Thus $\mathcal{Q}_{\text{init}} = [-1, 1]^3$. This system might represent a mechanical system with mass matrix M, damping matrix D and stiffness matrix K, which depend on the parameters δ_1, δ_2 and δ_3 respectively.

Of course, there exist many analytical results for such systems. For example, a sufficient condition for robust stability of the system over $[-1, 1]^3$ is that M, D and K are all positive definite over $[-1, 1]^3$, which in turn, may be checked via three generalized eigenvalue problems.

We will demonstrate the branch and bound algorithm on this problem. Our example uses the matrices

$$M_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}, \quad D_0 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}, \quad K_0 = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix},$$

$M_1 = m_1 m_1^T$, $D_1 = d_1 d_1^T$ and $K_1 = k_1 k_1^T$, where

$$m_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & 0 \end{bmatrix}, \quad d_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad k_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

In particular, $\text{rank}(M_1) = 2$, and D_1 and K_1 are rank one matrices, so that the parameters induce rank two variations in the M matrix and rank one variations in the D and K matrices.

Figure 10 shows that this problem can be cast into our framework as follows.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -8 & -2 & 4 & -6 & 1 & 3 \\ -1 & -1 & 0 & 0 & -3 & -1 \\ 4 & 1 & -4 & 3 & -1 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & -2 & -1 & 2 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix},$$

$$C = \begin{bmatrix} 7 & 1 & -4 & 6 & -4 & -4 \\ -7 & -1 & 4 & -6 & 4 & 4 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -3 & 3 & 2 & -1 \\ 3 & -3 & -2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note the $\Delta = \text{diag}(\delta_1, \delta_1, \delta_2, \delta_3)$, so that the first parameter δ_1 has two channels, corresponding to $\text{rank}(M_1) = 2$.

Figure 11 shows the performance of the algorithm. The algorithm returns an MSD of 0.177 after 250 iterations.

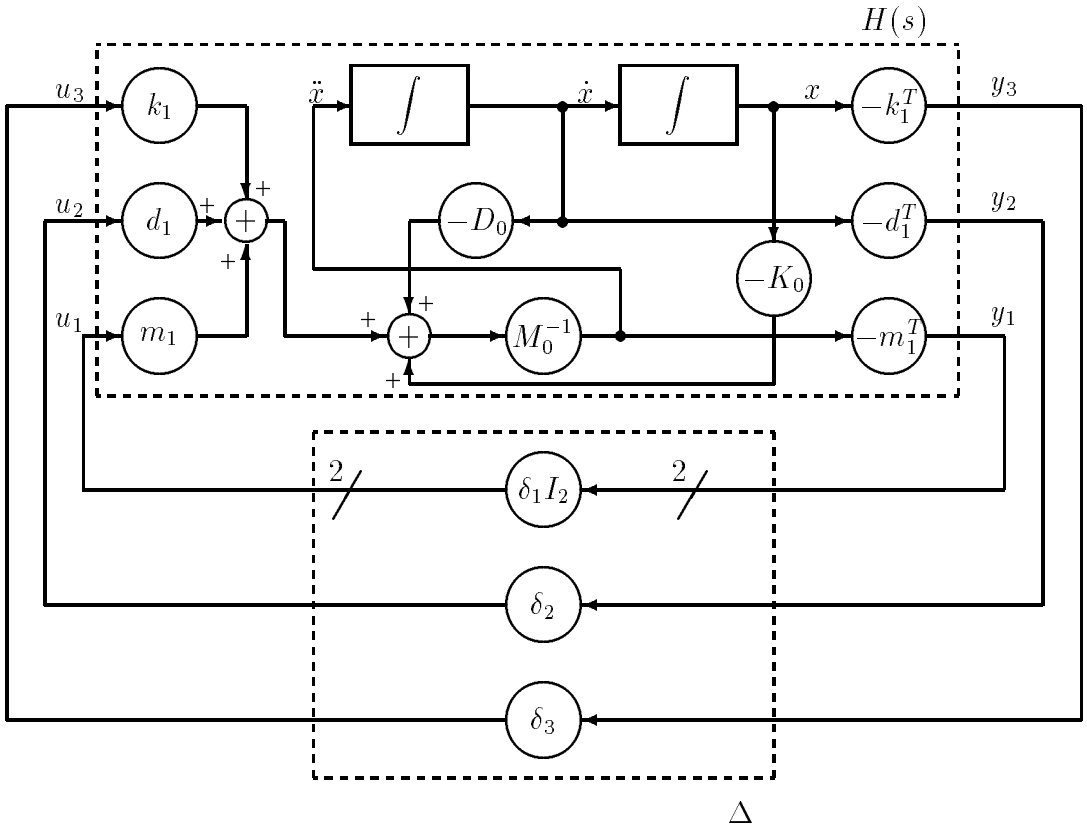

Figure 10: Block diagram of the standard form for the mechanical system.

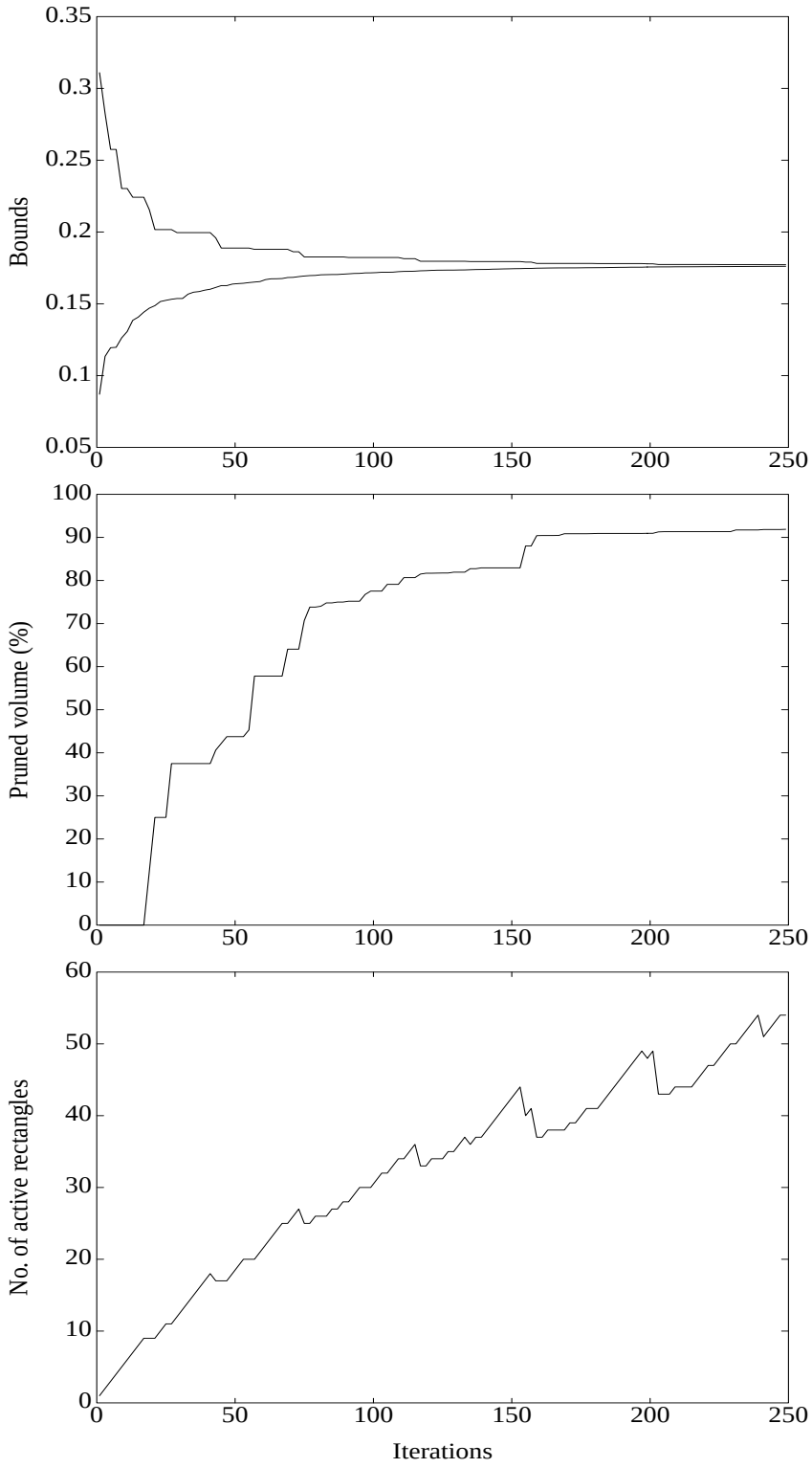

Figure 11: A mechanical system.

5 Conclusions

We have described a simple branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. The algorithm maintains provable upper and lower bounds on the MSD as it proceeds: it "paves" the parameter space with quadratic positive definite Lyapunov functions that prove the lower bound, and gives a parameter value that achieves the upper bound. The upper and lower bounds are guaranteed to converge to the MSD. As iterations progress, the algorithm prunes regions of parameter space, eliminating the possibility of the MSD being achieved in these regions. Thus, the algorithm may be terminated at any stage giving useful information about the MSD. The algorithm often performs well, but, in the worst case, effectively grids the parameter space in which case the computational effort will increase exponentially with the number of parameters.

There are some obvious ways in which the algorithm may be improved. The upper bound computation may be improved through a local optimization or line search (see, for example, [Lue84]). The lower bound computation can be improved via scaling the transfer matrix H so as to reduce its $\mathbf{H}_\infty$ norm [Gha90, Saf86, Doy82]. At present, the best lower bound available is from [FTD91], which requires the solution of a convex optimization problem at each frequency. While these improvements can substantially reduce computation times, they do not alter the worst-case combinatorial nature of the algorithm, as far as we know.

The branch and bound algorithm may be readily applied towards the computation of many other quantities of interest for linear systems with parameters (see [BB91a, BB92]).

Appendix

In the following, we show that the branch and bound algorithm converges in a finite number of steps, provided the bound functions $\Phi_{\rm lb}(\cdot)$ and $\Phi_{\rm ub}(\cdot)$ satisfy conditions (R1) and (R2) listed at the beginning of §2. We then show that the bounds for the MSD satisfy these conditions.

An upper bound on the number of branch and bound iterations

The derivation of an upper bound on the number of iterations of the branch and bound algorithm involves the following steps. We first show that after a large number of iterations k, the partition $\mathcal{L}_k$ must contain a rectangle of small volume. We then show that this rectangle has a small size, and this in turn implies that $U_k - L_k$ is small.

First, we observe that the number of rectangles in the partition $\mathcal{L}_k$ is just k (without pruning, which in any case does not affect the number of iterations). The total volume of these rectangles is $\mathrm{vol}(\mathcal{Q}_{\rm init})$, and therefore

$$\min_{\mathcal{Q}\in\mathcal{L}_k} \mathrm{vol}(\mathcal{Q}) \leq \frac{\mathrm{vol}(\mathcal{Q}_{\rm init})}{k}. \tag{11}$$

Thus, after a large number of iterations, at least one rectangle in the partition has small volume.

Next, we show that small volume implies small size for a rectangle in any partition. We define the *condition number* of a rectangle $\mathcal{Q} = \prod_i [l_i, u_i]$ as

$$\mathrm{cond}(\mathcal{Q}) = \frac{\max_i (u_i - l_i)}{\min_i (u_i - l_i)}.$$

We then observe that our splitting rule, which requires that we split rectangles along a longest edge, results in a uniform bound on the condition number of rectangles in our partition.

Lemma 1 *For any k and any rectangle $\mathcal{Q} \in \mathcal{L}_k$,*

$$\mathrm{cond}(\mathcal{Q}) \leq \max\{\mathrm{cond}(\mathcal{Q}_{\mathrm{init}}), 2\}. \tag{12}$$

Proof

It is enough to show that when a rectangle $\mathcal{Q}$ is split into rectangles $\mathcal{Q}_1$ and $\mathcal{Q}_2$,

$$\mathrm{cond}(\mathcal{Q}_1) \leq \max\{\mathrm{cond}(\mathcal{Q}), 2\}, \qquad \mathrm{cond}(\mathcal{Q}_2) \leq \max\{\mathrm{cond}(\mathcal{Q}), 2\}.$$

Let $\nu_{\max}$ be the maximum edge length of $\mathcal{Q}$, and $\nu_{\min}$, the minimum. Then $\mathrm{cond}(\mathcal{Q}) = \nu_{\max}/\nu_{\min}$. When $\mathcal{Q}$ is split into $\mathcal{Q}_1$ and $\mathcal{Q}_2$, our splitting rule requires that $\mathcal{Q}$ be split along an edge of length $\nu_{\max}$. Thus, the maximum edge length of $\mathcal{Q}_1$ or $\mathcal{Q}_2$ can be no larger than $\nu_{\max}$. Their minimum edge length could be no smaller than the minimum of $\nu_{\max}/2$ and $\nu_{\min}$, and the result follows. ■

We note that there are other splitting rules that also result in a uniform bound on the condition number of the rectangles in any partition generated. One such rule is to cycle through the index on which we split the rectangle. If $\mathcal{Q}$ was formed by splitting its parent along the ith coordinate, then when we split $\mathcal{Q}$, we split it along the $(i+1)$modulo m coordinate.

We can bound the size of a rectangle $\mathcal{Q}$ in terms of its volume and condition number, since

$$\begin{aligned}
\mathrm{vol}(\mathcal{Q}) &= \prod_i (u_i - l_i) \\
&\geq \max_i (u_i - l_i) \left(\min_i (u_i - l_i) \right)^{m-1} \\
&= \frac{(2\,\mathrm{size}(\mathcal{Q}))^m}{\mathrm{cond}(\mathcal{Q})^{m-1}} \\
&\geq \left(\frac{2\,\mathrm{size}(\mathcal{Q})}{\mathrm{cond}(\mathcal{Q})} \right)^m .
\end{aligned}$$

Thus,

$$\text{size}(\mathcal{Q}) \leq \frac{1}{2}\,\text{cond}(\mathcal{Q})\text{vol}(\mathcal{Q})^{1/m}. \tag{13}$$

Combining equations (11), (12) and (13) we get

$$\min_{\mathcal{Q}\in\mathcal{L}_k} \text{size}(\mathcal{Q}) \leq \frac{1}{2}\ \max\{\text{cond}(\mathcal{Q}_{\text{init}}), 2\}\left(\frac{\text{vol}(\mathcal{Q}_{\text{init}})}{k}\right)^{1/m}. \tag{14}$$

Thus, for large k, the partition $\mathcal{L}_k$ must contain a rectangle of small size.

Finally, we show that if a partition has a rectangle of small size, the upper and lower bounds cannot be too far apart. More precisely, we show that given some $\epsilon > 0$, there is some N such that $U_N - L_N \leq \epsilon$ for some $N \leq k$.

First, let δ be small enough such that if $\text{size}(\mathcal{Q}) \leq 2\delta$ then $\Phi_{\text{ub}}(\mathcal{Q}) - \Phi_{\text{lb}}(\mathcal{Q}) \leq \epsilon$ (recall requirement (R2) at the beginning of §2). Let k be large enough such that

$$\max\{\text{cond}(\mathcal{Q}_{\text{init}}), 2\}\left(\frac{\text{vol}(\mathcal{Q}_{\text{init}})}{k}\right)^{1/m} \leq 2\,\delta. \tag{15}$$

Then from equation (14), some $\mathcal{Q} \in \mathcal{L}_k$ satisfies $\text{size}(\mathcal{Q}) \leq \delta$. Then the rectangle $\tilde{\mathcal{Q}}$, one of whose halves is $\mathcal{Q}$, must satisfy $\text{size}(\tilde{\mathcal{Q}}) \leq 2\delta$, and therefore

$$\Phi_{\text{ub}}(\tilde{\mathcal{Q}}) - \Phi_{\text{lb}}(\tilde{\mathcal{Q}}) \leq \epsilon.$$

However, since $\tilde{\mathcal{Q}}$ was split at some previous iteration, it must have satisfied $\Phi_{\text{lb}}(\tilde{\mathcal{Q}}) = L_N$ for some $N \leq k$. Thus

$$U_N - L_N \leq \Phi_{\text{ub}}(\tilde{\mathcal{Q}}) - L_N \leq \epsilon,$$

or we have an upper bound on the number of branch and bound iterations.

Convergence of the bounds for the MSD

We now show that the bounds that we have derived for the MSD satisfy condition (R2) of §2. Indeed, we will show that there exist positive real numbers M, δ and ν such that for every rectangle $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ such that $\text{size}(\mathcal{Q}) < \delta$,

$$\Phi_{\text{ub}}(\mathcal{Q}) - \Phi_{\text{lb}}(\mathcal{Q}) < M\ \text{size}(\mathcal{Q})^{\nu}. \tag{16}$$

We assume that the system is well-posed.

Recall that given a rectangle $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, the lower bound $\Phi_{\text{lb}}(\mathcal{Q})$ is

$$\Phi_{\text{lb}}(\mathcal{Q}) = \inf\left\{\alpha : \|\tilde{H}\|_{\infty,\alpha} \geq 1\right\},$$

where $\tilde{H}(s)$ has a state space representation $\{\tilde{A},\tilde{B},\tilde{C},\tilde{D}\}$ and

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

The upper bound $\Phi_{\rm ub}(\mathcal{Q})$ is just $\mathrm{SD}(\tilde{A})$. Note that $\sigma_{\max}(F) = \mathrm{size}(\mathcal{Q})$.

Now, let $\lambda_1, \lambda_2, \ldots, \lambda_l$, be the distinct eigenvalues of $\tilde{A}$, with $\mathrm{SD}(\tilde{A}) = \mathrm{Re}\ -\lambda_1 = \alpha_0$ (say). Then, we may write down a residue expansion for the transfer matrix $\tilde{H}(s)$ as

$$\tilde{H}(s) = F^{1/2} \underbrace{\left(\sum_{i=1}^{l} \sum_{j=1}^{r_i} \frac{R_{ij}}{(s-\lambda_i)^j} + \hat{D} \right)}_{\hat{H}(s)} F^{1/2}. \tag{17}$$

The residue expansion above depends on the offset K of $\mathcal{Q}$; we do not show this dependence explicitly.

Then for any $\alpha < \alpha_0$ such that $\|\tilde{H}\|_{\infty,\alpha} \geq 1$, we have from equation (17),

$$\mathrm{size}(\mathcal{Q})\ \|\hat{H}\|_{\infty,\alpha} \geq 1,$$

which implies that

$$\mathrm{size}(\mathcal{Q})\ \left(\sum_{i=1}^{l} \sum_{j=1}^{r_i} \sigma_{\max}(R_{ij}) \left\| \frac{1}{(s-\lambda_i)^j} \right\|_{\infty,\alpha} + \sigma_{\max}(\hat{D}) \right) \geq 1.$$

Since $\mathrm{Re}\ \lambda_1 \geq \mathrm{Re}\ \lambda_i, \quad i = 1, 2, \ldots, l$ and $\alpha_0 = -\mathrm{Re}\ \lambda_1$, we have

$$0 > \alpha - \alpha_0 \geq \alpha + \mathrm{Re}\ \lambda_i, \quad i = 1, 2, ..., l,$$

and therefore

$$\left\| \frac{1}{(s-\lambda_i)^j} \right\|_{\infty,\alpha} = \left| \frac{1}{(-\mathrm{Re}\ \lambda_i - \alpha)^j} \right| \leq \frac{1}{(\alpha_0 - \alpha)^j}.$$

Thus,

$$\mathrm{size}(\mathcal{Q})\ \left(\sum_{i=1}^{l} \sum_{j=1}^{r_i} \frac{\sigma_{\max}(R_{ij})}{(\alpha_0 - \alpha)^j} + \sigma_{\max}(\hat{D}) \right) \geq 1. \tag{18}$$

If $\mathrm{size}(\mathcal{Q})$ is small enough such that

$$\mathrm{size}(\mathcal{Q})\ \left(\sum_{i,j} (\sigma_{\max}(R_{ij})) + \sigma_{\max}(\hat{D}) \right) < 1/2,$$

then from equation (18) we must have $(\alpha_0 - \alpha) < 1$ and therefore equation (18) yields

$$\mathrm{size}(\mathcal{Q}) \left(\sum_{i,j} (\sigma_{\max}(R_{ij})) + \sigma_{\max}(\hat{D}) \right) > (\alpha_0 - \alpha)^{\max_i r_i} \geq (\alpha_0 - \alpha)^n.$$

(Recall that $A \in \mathbf{R}^{n \times n}$.) Thus,

$$\mathrm{size}(\mathcal{Q})^{1/n} \left(\sigma_{\max}(\hat{D}) + \sum_{i,j} \sigma_{\max}(R_{ij}) \right)^{1/n} > (\alpha_0 - \alpha),$$

and therefore

$$\mathrm{size}(\mathcal{Q})^{1/n}\left(\sigma_{\max}(\hat{D})+\sum_{i,j}\sigma_{\max}(R_{ij})\right)^{1/n} > (\Phi_{\mathrm{ub}}(\mathcal{Q})-\Phi_{\mathrm{lb}}(\mathcal{Q})).$$

The above equation almost gives us the quantities that we seek in order to show that the bounds we have derived satisfy the requirement (16). However, the quantities R_{ij} depend on the offset K of the rectangle under consideration, while the requirement (16) should be satisfied uniformly, irrespective of the K. In order to show this, we define

$$M = \max_{\mathcal{Q}\subseteq\mathcal{Q}_{\mathrm{init}}}\left(\sigma_{\max}(\hat{D})+\sum_{i,j}\sigma_{\max}(R_{ij})\right)^{1/n}.$$

Note that $M<\infty$, as long as all the entries of $\mathcal{A}(q)$ are bounded for all $q\in\mathcal{Q}_{\mathrm{init}}$, which holds since the system is well-posed. This definition for M, along with $\nu=1/n$ yields

$$(\Phi_{\mathrm{ub}}(\mathcal{Q})-\Phi_{\mathrm{lb}}(\mathcal{Q})) < M\ \mathrm{size}(\mathcal{Q})^{\nu},$$

for every $\mathcal{Q}\subseteq\mathcal{Q}_{\mathrm{init}}$ such that $\mathrm{size}(\mathcal{Q})<1/(2M^n)$.

References

[ABJ75] B. D. Anderson, N. K. Bose, and E. I. Jury. Output feedback stabilization and related problems—Solution via decision methods. *IEEE Trans. Aut. Control*, AC-20:53–66, 1975.

[AM69] B. Anderson and J. B. Moore. Linear system optimization with prescribed degree of stability. *Proc. IEEE*, 116(12):2083–2087, December 1969.

[Bal68] E. Balas. A note on the branch-and-bound principle. *Operations Research*, 16:442–445, 1968.

[Bar84] B. R. Barmish. Invariance of the strict Hurwitz property for polynomials with perturbed coefficients. *IEEE Trans. Aut. Control*, AC-29:935–936, 1984.

[BB90] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. *Syst. Control Letters*, 15:1–7, 1990.

[BB91a] V. Balakrishnan and S. Boyd. Computation of the worst-case covariance for linear systems with uncertain parameters. In *Proc. IEEE Conf. on Decision and Control*, Brighton, U. K, December 1991.

[BB91b] S. Boyd and C. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, 1991.

[BB92] V. Balakrishnan and S. Boyd. Global optimization in control system analysis and design. In C.T. Leondes, editor, *Advances in Control Systems*. Academic Press, New York, New York, 1992.

[BBK89] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems. *Mathematics of Control, Signals, and Systems*, 2(3):207–219, 1989.

[BFS88] B. Ross Barmish, M. Fu, and S. Saleh. Stability of a polytope of matrices: Counterexamples. *IEEE Trans. Aut. Control*, 33(6):569–572, June 1988.

[BH84] B. R. Barmish and C. V. Hollot. Counterexamples to a recent result on the stability of interval matrices by S. Bialas. *Int. J. Control*, 39(5):1103–1104, 1984.

[BHL89] A. C. Bartlett, C. V. Hollot, and H. Lin. Root locations of an entire polytope of polynomials: it suffices to check the edges. *Mathematics of Control, Signals, and Systems*, 1(1):61–71, 1989.

[Bia83] S. Bialas. A necessary and sufficient condition for stability of interval matrices. *Int. J. Control*, 37(4):717–722, 1983.

[BKST89] B. R. Barmish, P. P. Khargonekar, Z. C. Shi, and R. Tempo. A pitfall in some of the robust stability literature. In *Proc. IEEE Conf. on Decision and Control*, pages 2273–2277, Tampa, Florida, December 1989.

[BKST90] B. R. Barmish, P. P. Khargonekar, Z. C. Shi, and R. Tempo. Robustness margin need not be a continuous function of the problem data. *Syst. Control Letters*, 15:91–98, Aug 1990.

[CEYB91] B. C. Chang, O. Ekdal, H. H. Yeh, and S. S. Banda. Computation of the real structured singular value via polytopic polynomials. *J. of Guidance*, 14(1):140–147, Jan-Feb 1991.

[DBB90] C. DeMarco, V. Balakrishnan, and S. Boyd. A branch and bound methodology for matrix polytope stability problems arising in power systems. In *Proc. IEEE Conf. on Decision and Control*, pages 3022–3027, Honolulu, Hawaii, December 1990.

[Dem92] J. Demmel. The componentwise distance to the nearest singular matrix. *SIAM J. on Matrix Analysis and Applications*, 1991-92. To appear.

[Doy82] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proc.*, 129-D(6):242–250, November 1982.

[DPA+88] S. Dasgupta, P. J. Parker, B. D. O. Anderson, F.J. Kraus, and M.Mansour. Frequency domain conditions for the robust stability of linear and nonlinear dynamical systems. In *Proc. American Control Conf.*, pages 1863–1868, Atlanta, Georgia, June 1988.

[DV75] C. A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties.* Academic Press, New York, 1975.

[FB89] M. Fu and B. R. Barmish. Polytopes of polynomials with zeros in a prescribed region. *IEEE Trans. Aut. Control*, 34(5):544–546, 1989.

[FPE86] G. F. Franklin, J. D. Powell, and A. Emami-Naeni. *Feedback Control of Dynamic Systems.* Addison-Wesley, 1986.

[FTD91] M. K. H. Fan, A. L. Tits, and J. C. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Aut. Control*, 36(1):25–38, Jan 1991.

[Gha90] Laurent El Ghaoui. *Robustness of Linear Systems against Parameter Variations.* PhD thesis, Stanford University, Stanford CA 94305, March 1990.

[GS88] R. R. E. De Gaston and M. G. Safonov. Exact calculation of the multiloop stability margin. *IEEE Trans. Aut. Control*, 33(2):156–171, 1988.

[HB91] W. M. Haddad and D. S. Bernstein. Explicit construction of quadratic Lyapunov functions for the small gain and positivity theorems. March 1991. Under preparation.

[Hei84] J. A. Heinen. Sufficient conditions for stability of interval matrices. *Int. J. Control*, 39(6):1323–1328, 1984.

[Kha78] V. L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differential'nye Uraveniya*, 14(11):1483–1485, 1978.

[Lue84] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Mass., 2nd edition, 1984.

[LW66] E. L. Lawler and D.E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14:699–719, 1966.

[MK87] T. Mori and H. Kokame. Convergence property of interval matrices and interval polynomials. *Int. J. Control*, 45(2):481–484, 1987.

[Pet90] I. R. Petersen. A new extension to Kharitonov's theorem. *IEEE Trans. Aut. Control*, 35(7):825–828, July 1990.

[Roh89] J. Rohn. Systems of linear interval equations. *Linear Algebra and its Applications*, 126:39–78, 1989.

[Roh90] J. Rohn. Nonsingularity under data rounding. *Linear Algebra and its Applications*, 139:171–174, 1990.

[RP92] J. Rohn and S. Poljak. Radius of nonsingularity. *Mathematics of Control, Signals, and Systems*, 1991-92. To appear.

[Saf86] M. G. Safonov. Optimal diagonal scaling for infinity-norm optimization. *Syst. Control Letters*, 7:257–260, 1986.

[Sd86] A. Sideris and R. deGaston. Multivariable stability margin calculation with uncertain correlated parameters. In *Proceedings of 25th Conference on Decision and Control*, pages 766–771. IEEE, December 1986.

[SP89] A. Sideris and R. S. S. Peña. Fast computation of the multivariable stability margin for real interrelated uncertain parameters. *IEEE Trans. Aut. Control*, 34(12):1272–1276, December 1989.

[VTM90] A. Vicino, A. Tesi, and M. Milanese. Computation of nonconservative stability perturbation bounds for systems with nonlinearly correlated uncertainties. *IEEE Trans. Aut. Control*, 35(7):835–841, 1990.

[Wil73] J. L. Willems. The circle criterion and quadratic Lyapunov functions for stability analysis. *IEEE Trans. Aut. Control*, 18:184–186, 1973.

[Xin87] L. X. Xin. Necessary and sufficient conditions for the stability of a class of interval matrices. *Int. J. Control*, 45(1):211–214, 1987.

List of Figures