In "Control of Uncertain Dynamic Systems," CRC Press, 1991.

Computing the Minimum Stability Degree of Parameter-dependent Linear Systems *

V. Balakrishnan, S. Boyd and S. Balemi †
Information Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford CA 94305, USA.

ABSTRACT

For linear systems with unspecified parameters that lie between given upper and lower bounds, we present a branch and bound algorithm that computes 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 $\mathbf{C}^{n\times n}$), $\lambda_i(P)$, $1 \le i \le n$ 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

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

*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.
A more detailed version of the paper has been submitted to the International Journal of Robust and Nonlinear Control [1].

†Currently with the Institut für Automatik, ETH-Zentrum, 8092 Zürich, Switzerland.

$\mathrm{SD}(P)$ 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 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.

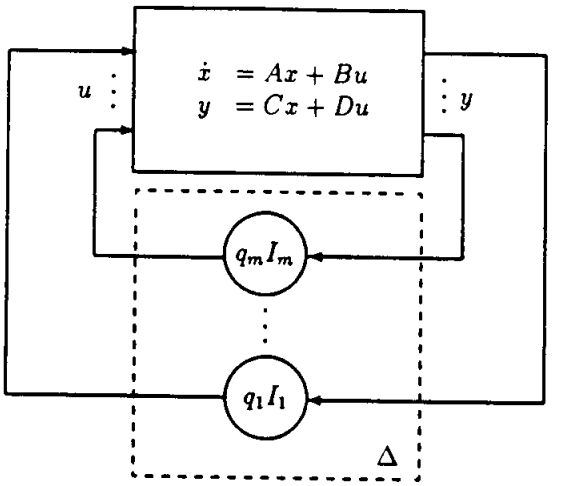


Figure 1: The standard form.

1.3. Some Important Questions

For uncertain systems in the standard form, there arise several important questions:

- 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.*

- 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.

- *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,

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

  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.

- *Minimum stability degree*

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

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

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 $e^{-(MSD(\mathcal{A}, \mathcal{Q}_{\mathrm{init}}))t}$. In fact,

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

Equivalently,

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

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 [2, 3]. Continuity of the MSD with respect to the input data is an immediate consequence of the fact that the eigenvalues of the matrix $(A + B\Delta(I - D\Delta)^{-1}C)$ are continuous functions of A, B, C, D and Δ, as long

as $\det(I - D\Delta)$ is bounded away from zero. In contrast, what follows shows that the SM is not, in general, a continuous function of the input data.

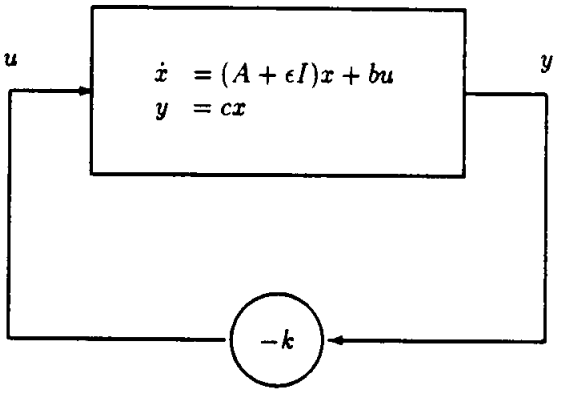
Our example consists of a family of single-input-single-output linear time invariant systems

$$\dot{x} = (A + \epsilon I - kbc)x,$$

where ϵ and k are parameters and A, b and c are constant matrices with

$$c(sI - A)^{-1}b = \frac{((s+0.8)^2 + 17.8929)\,((s+1.2)^2 + 16)}{((s+1)^2 + 17.8929)\,((s+1)^2 + 16)\,(s-4)(s-3)(s-2)}.$$

The following block diagram shows the setup.



We will regard k as the uncertain parameter and ϵ is the "state-space parameter" or the "input datum". Then for fixed ϵ, the robust stability margin SM is just the half-length of the largest interval of k centered at the nominal value k_0 $(= 100)$, for which the system is stable. For this system, it turns out that the robust stability margin with respect to k is not a continuous function of ϵ. Figure 2 shows the SM as a function of ϵ.

The discontinuity of SM as a function of ϵ is evident. The reason for the discontinuity is immediately obvious from the root locus with k of the system shown in figure 3 (with $\epsilon = 0$).

It is clear from the root locus plot that k is not a (single-valued) function of the real part of the least damped eigenvalue. Now, as ϵ increases, the locus shifts to the right and the value of k for which the locus crosses into the right half plane decreases continuously for a while. When $\epsilon \approx 0.34$, the value of k for which the locus crosses into the right half plane becomes non-unique and causes the SM to have a discontinuity.

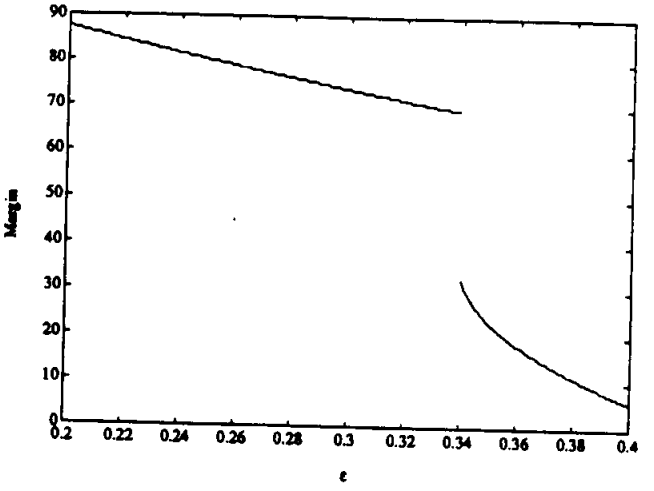


Figure 2: The stability margin of the system as function of ϵ.

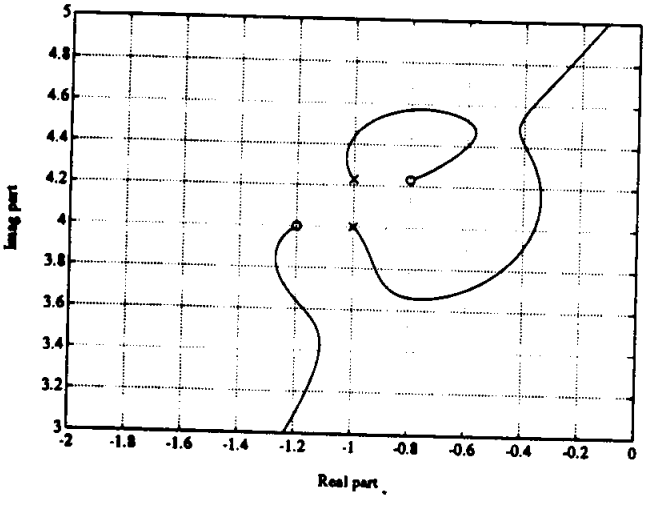


Figure 3: A portion of the root locus of the system showing the eigenvalues with positive imaginary part.

1.5. Some Approaches

Many of 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 [4]. Less trivially, Kharitonov's theorem [5, 6] 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 [7, 8].

In [9], 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.

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.

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 $\mathcal{Q}_{\text{init}}$.

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 $\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}})$. An *upper* bound for $\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}})$ is obtained by one of the optimistic methods above. However, these bounds may be unsatisfactory, in which case, a branch and bound technique [10, 11] is used to systematically improve the bounds. At each stage of the algorithm, guaranteed upper and lower bounds are available for $\text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}})$.

The use of branch and bound algorithms for robustness analysis is not new. De Gaston and Safonov [12] use a branch and bound algorithm for computing the SM for systems with uncorrelated uncertain parameters (though they do not explicitly mention the term "branch and bound"). Sideris and Peña [13] extend this algorithm to the case when the parameters are real and may be correlated. In [14], 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.* [15] use a branch and bound algorithm with geometric programming ideas to compute the SM. Demarco et al. [16] use a branch and bound algorithm to study stability problems arising in power systems. Our algorithm is closer to those described in [16], [12] and [13].

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 [10, 11] finds the (global) minimum of a function $f : \mathbf{R}^m \to \mathbf{R}$ over an m-dimensional rectangle $\mathcal{Q}_{\text{init}}$.

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 $\{\mathcal{Q} : \mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}\}$ (which, presumably, are easier to compute than $\Phi_{\min}(\mathcal{Q})$). These two functions 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 $\text{size}(\mathcal{Q})$, 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 $\Phi_{\text{ub}}(\mathcal{Q}_i)$, $i = 1, 2, \ldots, N$. 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 $\text{size}(\mathcal{Q}_i) \leq \delta, i = 1, 2, \ldots, N$, 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 $\mathcal{Q}_i$ from the partition such that $\Phi_{\text{lb}}(\mathcal{Q}_i) = \min_{1 \le i \le N} \Phi_{\text{lb}}(\mathcal{Q}_i)$, 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.

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* $\mathcal{Q}_{II};$
  $\mathcal{L}_{k+1} := (\mathcal{L}_k - \{\mathcal{Q}\}) \cup \{\mathcal{Q}_I, \mathcal{Q}_{II}\};$
  $L_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{lb}}(\mathcal{Q});$
  $U_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{ub}}(\mathcal{Q});$
  $k = k + 1;$
}

At the end of k iterations, U_k and L_k are upper and lower bounds respectively for $\Phi_{\min}(\mathcal{Q}_{\text{init}})$. 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. We refer the reader to [1] for a proof of this fact.

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 the computation and 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 $f(q) = \text{SD}(\mathcal{A}(q))$ and $\Phi_{\text{min}}(\mathcal{Q}) = \text{MSD}(\mathcal{A}, \mathcal{Q})$. We now need to compute a lower bound $\Phi_{\text{lb}}(\mathcal{Q})$ and an upper bound $\Phi_{\text{ub}}(\mathcal{Q})$ for $\text{MSD}(\mathcal{A}, \mathcal{Q})$.

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:

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

Computation of the lower bound is a little more involved; it is based on the application of the *small gain theorem* (SGT) [17]. SGT states that the system in figure 1 is well-posed and robustly stable (with $\mathcal{Q} = \mathcal{U}$) if $\|H\|_\infty < 1$, where

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

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

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

To derive a better lower bound on $\text{MSD}(\mathcal{A},\mathcal{U})$, 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,

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

Thus we have

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

where

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

is the α-shifted $\mathbf{H}_\infty$ norm of H [18]. Therefore, we define $\Phi_{\mathrm{lb}}(\mathcal{U})$ as

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

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

We now show how to compute $\Phi_{\mathrm{lb}}(\mathcal{U})$. 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, $\Phi_{\mathrm{lb}}(\mathcal{U}) = -\infty$ 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 4 obtains.

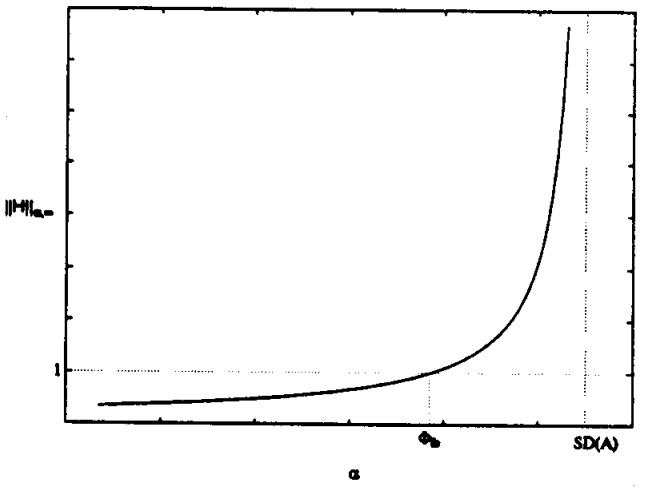


Figure 4: When $\sigma_{\max}(D) < 1$, a bisection method can be used to compute Φ_{lb}.

In [19, 20], it is shown that provided $\sigma_{\max}(D) < 1$ and $\alpha < \mathrm{SD}(A)$,

$\|H\|_{\infty,\alpha} < 1$ if and only if

$$M_\alpha = \begin{bmatrix} A + \alpha I + BR^{-1}D^TC & -BR^{-1}B^T \\ C^TS^{-1}C & -A^T - \alpha I - C^TDR^{-1}B^T \end{bmatrix}$$

has no imaginary eigenvalues,

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

We note that the above procedure for computing $\Phi_{\text{lb}}(\mathcal{U})$ is an application of the "shifted circle criterion" (Anderson and Moore [21]).

3.1.2. Normalization of the Parameter Rectangle $\mathcal{Q}$

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 5 demonstrates the loop transformation, where the symbols $\tilde{H}(s)$ and $\tilde{\Delta}$ refer to the "loop-transformed" system and the normalized perturbation. (See [17] 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 $[-1,1]^m$.

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

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

It is now easily verified that $\tilde{\Delta}$ has the form $\text{diag}(\tilde{q}_1 I_1, \tilde{q}_2 I_2, \ldots, \tilde{q}_m I_m)$, 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

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

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

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

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

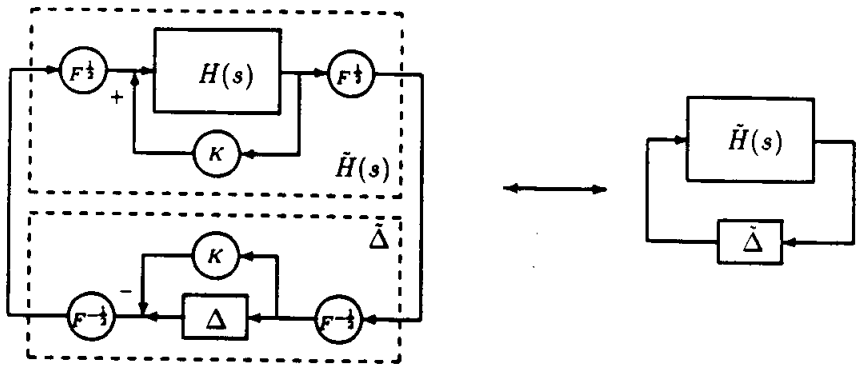


Figure 5: Loop Transformation.

We show in [1] 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 $\Phi_{\text{lb}}(\mathcal{Q}) = L_k$, 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 $q_{\text{ill\_posed}}$ such that the feedback system is not well-posed for $q = q_{\text{ill\_posed}}$.

- If $\Phi_{\text{lb}}(\mathcal{Q}) > -\infty$, the algorithm also provides a "certificate" that *proves* that $\Phi_{\text{lb}}(\mathcal{Q})$ is a lower bound: it is shown in [19] (see also [22, 23]) how to construct a quadratic positive definite Lyapunov function $V(x)$ that satisfies

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

  This proves that $\text{SD}(\mathcal{A}(q)) \geq \Phi_{\text{lb}}(\mathcal{Q})$ 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_{bad} such that $\text{SD}(\mathcal{A}(q_{\text{bad}}))$ equals the upper bound on the MSD on exit, satisfies

$$f(q_{\text{bad}}) - \text{MSD}(\mathcal{A}, \mathcal{Q}_{\text{init}}) < \epsilon.$$

4. AN EXAMPLE

We consider the problem of computing the MSD of a family of systems

$$\dot{x} = \begin{bmatrix} \frac{1}{a(q_1,q_2)} & 0 \\ 0 & \frac{1}{b(q_1,q_2)} \end{bmatrix} x,$$

where $q_1 \in [-4,0]$ and $q_2 \in [-4,4]$ and

$$\begin{aligned} a(q_1,q_2) &= (q_1+3.5)^2 + (q_2+1)^2 + 1/0.9 \\ b(q_1,q_2) &= q_1^4 + q_2^4 + 1. \end{aligned}$$

Obviously, the eigenvalues of $\mathcal{A}(q)$ are just $1/a(q_1,q_2)$ and $1/b(q_1,q_2)$; therefore the MSD is -1, and is achieved at $q_1 = 0$, $q_2 = 0$.

As the first step towards applying the branch and bound algorithm to this problem, we cast the system into the standard form of figure 1. This is a tedious, but straightforward exercise; it involves realizing the rational functions that make up the entries of $\mathcal{A}(q)$ as a block diagram involving only simple gain blocks. The A, B, C and D matrices for the standard form are not shown here.

Figure 6 shows the level sets of $\mathrm{SD}(\mathcal{A}(q))$ over the parameter region. This is nothing but a contour plot of the negative of the maximum real part of the eigenvalues of $\mathcal{A}(q)$. There are two local minima for $\mathrm{SD}(\mathcal{A}(q))$, and it so happens that the minimum on the right is the global minimum. It is clear from the figure that a *local* optimization procedure with random starting points will converge to the "spurious" minimum on the left in about 75% of the cases. This includes the vertices of the parameter region; a local optimization starting at any of the vertices will converge to the peak on the left, which is not the global minimum. Of course, if the local optimization procedure were repeated several times with randomly chosen initial starting points, the procedure would probably converge to the global minimum at least once. However, there is no way of telling if the local method has converged to the global minimum; in other words, there are no guarantees.

On the other hand, figure 7 shows *guaranteed* upper and lower bounds for the MSD returned by the branch and bound algorithm. It is clear that the algorithm first attempts to establish well-posedness, *i.e.* , that the entries of $\mathcal{A}(q)$ are bounded over the range of values for q. Figure 7 also shows the fraction of the volume (or in this case, the area) of parameter space that has been eliminated from consideration by the algorithm, as a function of iterations. Finally, figure 7 shows the number of rectangles in the rectangle list, as a function of iterations.

Figure 8 shows the regions in parameter space that are still under consideration at various stages of the algorithm. The algorithm can *prove* that the MSD cannot be achieved outside these regions.

The algorithm returns a value of 1.00 for the MSD to within an absolute accuracy of 0.01 at the end of about 10000 iterations. The algorithm also returns a "bad" parameter value of $q1 = 0$ and $q2 = 0$.

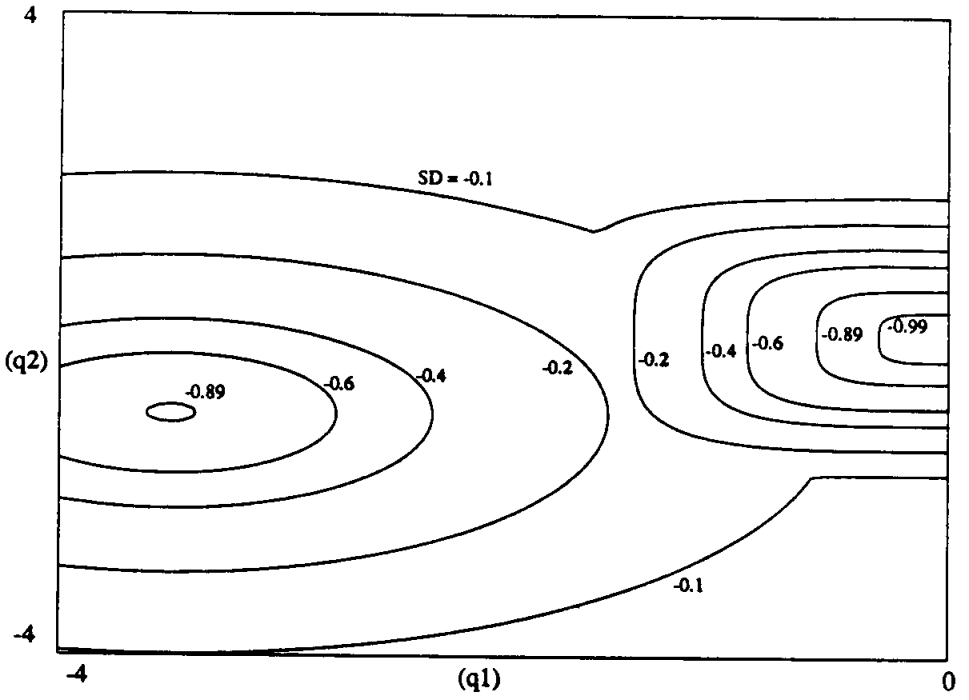


Figure 6: Level sets of $\mathrm{SD}(\mathcal{A}(q))$.

The reader might note with some concern that the algorithm takes 10000 iterations to compute the MSD which, as noted before, can be determined by mere inspection. We address this with two remarks. First, the example presented above has been carefully designed so that the MSD is a particularly simple function of the parameters, which, needless to say, will not be the case in general. Secondly, the bounds on the MSD that we have employed in the branch and bound algorithm are quite crude; we mention a few improvements to the bounds in the conclusion. We believe that such improvements will cut down the number of branch and bound iterations significantly.

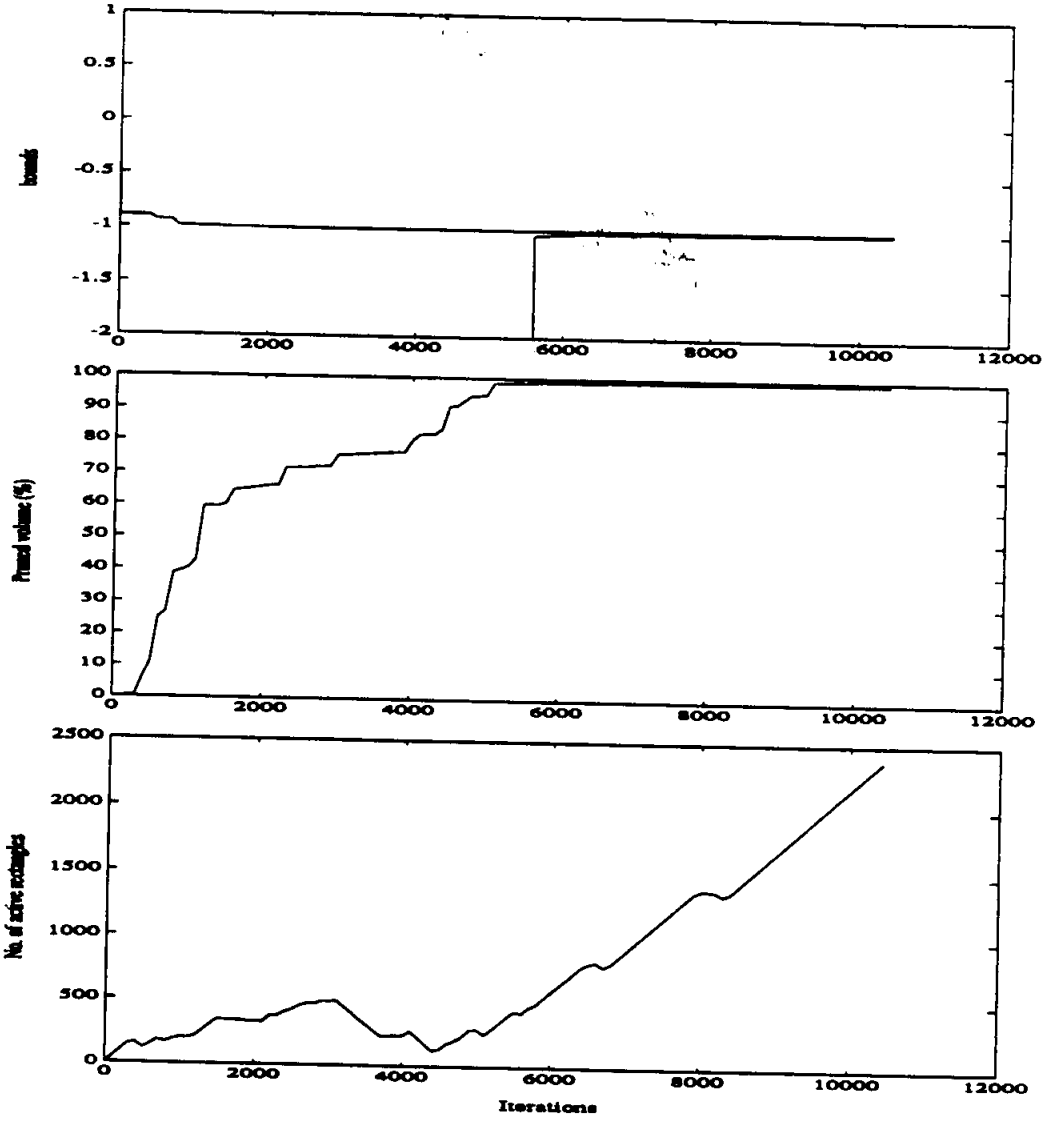


Figure 7: Branch and bound algorithm on the example problem.

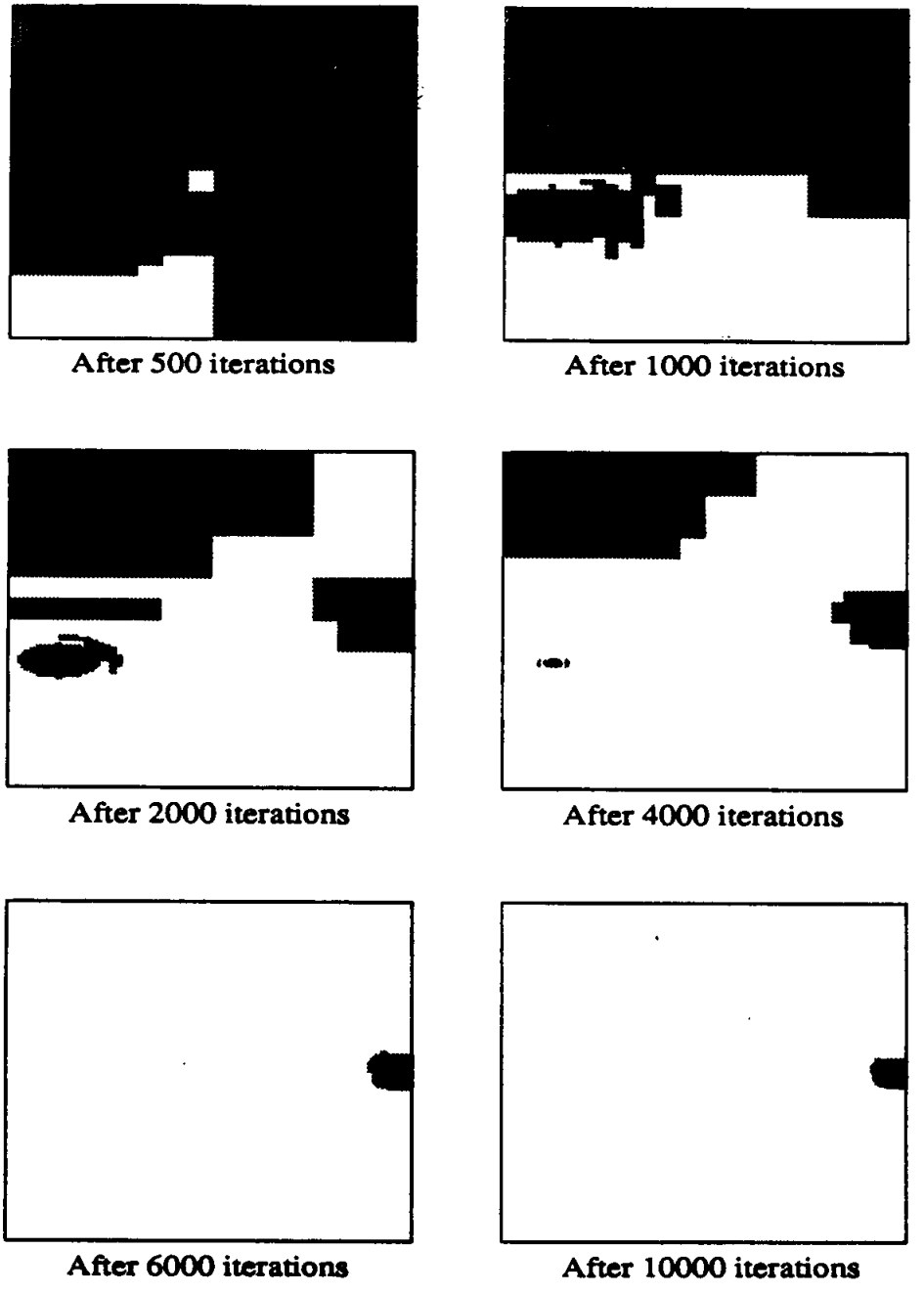


Figure 8: The unpruned parameter region at various stages of the algorithm.

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, [24]). The lower bound computation can be improved via scaling the transfer matrix H so as to reduce its $\mathbf{H}_\infty$-norm [25, 26, 27]. 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.

REFERENCES

[1] V. Balakrishnan, S. Boyd, and S. Balemi, Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems, *submitted, International Journal of Nonlinear and Robust Control*, 1991.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

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

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.

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

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

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

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

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

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

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