BRANCH AND BOUND ALGORITHM FOR GLOBAL OPTIMIZATION IN CONTROL THEORY *

S. Balemi
Automatic Control Laboratory, ETH-Zentrum, 8092 Zürich, Switzerland

V. Balakrishnan
Information Systems Laboratory, Stanford University, Stanford, CA 94305, USA

Abstract. The computation of several quantities of interest in the analysis and design of parameter-dependent linear systems can be posed as a nonconvex optimization problem. We present a branch and bound algorithm that solves such optimization problems. The algorithm is worst-case combinatoric, but often performs well.
We demonstrate the algorithm with the computation of the maximum and minimum RMS-gain of a discrete-time linear system over a set of parameters. The first problem might correspond the the worst-case analysis of an uncertain system, whereas the second problem might be regarded as the design of a parametric controller. Finally, we present the 'hybrid' problem, where the RMS-gain is maximized over a set of parameters and minimized over another set of parameters.

Keywords. Global optimization, Branch-and-Bound algorithm, Worst-case analysis, Robustness analysis, Parametric design, Minimax problems, Discrete-time $\mathbf{H}_\infty$-norm.

INTRODUCTION

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

$$\begin{aligned}
x(k+1) &= Ax(k) + B_u u(k) + B_w w(k), \\
y(k) &= C_y x(k) + D_{yu} u(k) + D_{yw} w(k), \\
z(k) &= C_z x(k) + D_{zu} u(k) + D_{zw} w(k), \\
u(k) &= \Delta\, y(k),
\end{aligned} \tag{1}$$

with $x(0) = x_0$, where $x(k) \in \mathbb{R}^n$, $w(k) \in \mathbb{R}^{n_i}$, $z(k) \in \mathbb{R}^{n_o}$, $u(k), y(k) \in \mathbb{R}^p$, and A, B_u, B_w, C_y, C_z, D_{yu}, D_{yw}, D_{zu} and D_{zw} are real matrices of appropriate sizes. Δ is a diagonal matrix parametrized by a vector of parameters $q = [q_1, q_2, \ldots, q_m]$, and given by the expression

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

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

$$\begin{aligned}
P_{yu} &= C_y(zI - A)^{-1} B_u + D_{yu} \\
P_{yw} &= C_y(zI - A)^{-1} B_w + D_{yw} \\
P_{zu} &= C_z(zI - A)^{-1} B_u + D_{zu} \\
P_{zw} &= C_z(zI - A)^{-1} B_w + D_{zw}.
\end{aligned}$$

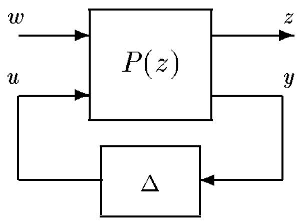

Fig. 1. System in standard form.

We may now write down an expression for the closed-loop transfer matrix from w to z:

$$P_{\mathrm{cl}}(q) = P_{zw} + P_{zu}\Delta(I - P_{yu}\Delta)^{-1} P_{yw}.$$

There are several quantities of interest associated with parameter-dependent systems such as the above. For instance, for continuous-time parameter-dependent systems, there are the stability margin (see De Gaston et al. 1988), the minimum stability degree (Balakrishnan et al. 1991), the $\mathbf{H}_\infty$-norm (Balemi et al. 1991) *etc.* We refer the reader to Balakrishnan and Boyd (1992) for the computation of these and other stability measures for continuous-time linear systems in a unified set-up using a branch and bound algorithm.

In the sequel, we will concern ourselves with the study of the RMS gain between $w(k)$ and $z(k)$ in Figure 1. We will first consider the computation of the maximum $\mathbf{H}_\infty$-norm ($\mathcal{H}_{\max}$) of the system (1), defined as

$$\begin{aligned}
\mathcal{H}_{\max}(\mathcal{Q}_{\mathrm{init}}) &= \max_{q \in \mathcal{Q}_{\mathrm{init}}} \max_{w \neq 0} \frac{\|z\|_{\mathrm{RMS}}}{\|w\|_{\mathrm{RMS}}} \\
&= \max_{q \in \mathcal{Q}_{\mathrm{init}}} \|P_{\mathrm{cl}}(q)\|_\infty,
\end{aligned}$$

*Some of this material is based on results already published by Balakrishnan et al. (1991) and Balemi et al. (1991)

where $\|w\|_{\rm RMS}$ of the vector signal w refers to the RMS-value

$$\|w\|_{\rm RMS} = \lim_{K\to\infty} \frac{1}{K}\sqrt{\sum_{k=0}^{K}(w(k)'w(k))}$$

and where $\|\cdot\|_\infty$ refers to the discrete-time $\mathbf{H}_\infty$-norm:

$$\|G\|_\infty = \sup_{0\le\omega\le\pi} \sigma_{\max}(G(e^{j\omega}))$$

($\sigma_{\max}(M)$ is the maximum singular value of M). $\mathcal{H}_{\max}$ is just the *worst-case root mean square gain* (RMS-gain) of the system between the input $w(k)$ and the output $z(k)$.

On the other hand, equation (1) might correspond to a system with a parametric controller, with q containing the design parameters. Then, it is of interest to find the parameters that minimize the discrete $\mathbf{H}_\infty$-norm, *i.e.*

$$\begin{aligned}\mathcal{H}_{\min}(\mathcal{Q}_{\rm init}) &= \min_{q\in\mathcal{Q}_{\rm init}} \max_{w\neq 0} \frac{\|z\|_{\rm RMS}}{\|w\|_{\rm RMS}} \\ &= \min_{q\in\mathcal{Q}_{\rm init}} \|P_{\rm cl}(q)\|_\infty,\end{aligned}$$

Finally, if Δ contains both uncertainties *and* design parameters, the so-called *minmax* problem arises. Here, we seek the choice of design parameters that minimizes the $\mathcal{H}_{\max}$ over the uncertain parameters. More precisely, let the first m_1 parameters be design parameters and remaining m_2 parameters be uncertainties ($m_1 + m_2 = m$). For convenience, let us rename the m_1 design parameters as $\underline{q} = [\underline{q}_1, \underline{q}_2, \ldots, \underline{q}_{m_1}]$ and the m_2 uncertain parameters as $\overline{q} = [\overline{q}_1, \overline{q}_2, \ldots, \overline{q}_{m_2}]$. Let

$$\begin{aligned}\underline{\mathcal{Q}}_{\rm init} &= [l_1, u_1]\times[l_2, u_2]\times\cdots\times[l_{m_1}, u_{m_1}], \\ \overline{\mathcal{Q}}_{\rm init} &= [l_{m_1+1}, u_{m_1+1}]\times\cdots\times[l_m, u_m].\end{aligned}$$

Then the minimax problem is the computation of

$$\begin{aligned}&\mathcal{H}_{\rm minmax}(\underline{\mathcal{Q}}_{\rm init}, \overline{\mathcal{Q}}_{\rm init}) \\ &\quad = \min_{\underline{q}\in\underline{\mathcal{Q}}_{\rm init}} \max_{\overline{q}\in\overline{\mathcal{Q}}_{\rm init}} \left\{\max_{w\neq 0} \frac{\|z\|_{RMS}}{\|w\|_{RMS}}\right\} \\ &\quad = \min_{\underline{q}\in\underline{\mathcal{Q}}_{\rm init}} \max_{\overline{q}\in\overline{\mathcal{Q}}_{\rm init}} \|P_{\rm cl}(q)\|_\infty.\end{aligned}$$

There exist no methods that compute any of the three quantities above exactly; however, there are several methods that provide good upper and lower bounds for $\mathcal{H}_{\max}$ and $\mathcal{H}_{\min}$. For example, lower (upper) bounds for $\mathcal{H}_{\max}$ ($\mathcal{H}_{\min}$) are provided by Monte Carlo methods where $\mathcal{H}_{\max}$ ($\mathcal{H}_{\min}$) is approximated by the largest (smallest) value of $\|P_{\rm cl}(q)\|_\infty$ over many values of q drawn according to some distribution. Another class of methods that yields lower (upper) bounds for $\mathcal{H}_{\max}$ ($\mathcal{H}_{\min}$) are *local optimization methods.* Here a local search is made for the parameter that finds a local maximum (minimum) of $\|P_{\rm cl}(q)\|_\infty$. On the other hand, upper (lower) bounds for $\mathcal{H}_{\max}$ ($\mathcal{H}_{\min}$) are provided by conservative methods. These are usually based on some analytical result, such as a small gain theorem, or a Lyapunov theorem.

In this paper, we employ an approach where we first compute upper and lower bounds for $\mathcal{H}_{\max}(\mathcal{Q}_{\rm init})$, $\mathcal{H}_{\min}(\mathcal{Q}_{\rm init})$ or $\mathcal{H}_{\rm minmax}(\underline{\mathcal{Q}}_{\rm init}, \overline{\mathcal{Q}}_{\rm init})$ using some of the methods described above; if these bounds are not satisfactory, that is, if they are not close enough, a branch and bound technique is used to systematically refine the bounds. At each stage of the algorithm, upper and lower bounds are maintained for $\mathcal{H}_{\max}(\mathcal{Q}_{\rm init})$, $\mathcal{H}_{\min}(\mathcal{Q}_{\rm init})$ or $\mathcal{H}_{\rm minmax}(\underline{\mathcal{Q}}_{\rm init}, \overline{\mathcal{Q}}_{\rm init})$. The branch and bound technique used in this paper is described in detail by Balakrishnan et al. (1991), where it is used to compute the minimum stability degree for a parameter-dependent continuous-time system. For details about applying the branch and bound algorithm to various simple minimization or maximization problems in the analysis and design of parameter-dependent linear systems, see Balakrishnan and Boyd (1992).

In the following section, we briefly describe the basic branch and bound algorithm and its extension ot the minmax case; we then use it to compute $\mathcal{H}_{\max}$, $\mathcal{H}_{\min}$ and $\mathcal{H}_{\rm minmax}$ in subsequent sections.

THE BRANCH AND BOUND ALGORITHMS

The first branch and bound algorithm we present finds the maximum of a function $f : \mathbb{R}^m \to \mathbb{R}$ over an m-dimensional rectangle $\mathcal{Q}_{\rm init}$ (the subscript "init" stands for *initial* rectangle).

Branch and Bound Algorithm for Maximization and Minimization

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

$$\Phi_{\max}(\mathcal{Q}) = \max_{q\in\mathcal{Q}} f(q).$$

Then, the algorithm computes $\Phi_{\max}(\mathcal{Q}_{\rm init})$ to within an absolute accuracy of $\epsilon > 0$. The algorithm uses two functions $\Phi_{\rm lb}(\mathcal{Q})$ and $\Phi_{\rm ub}(\mathcal{Q})$ defined over $\{\mathcal{Q} : \mathcal{Q} \subseteq \mathcal{Q}_{\rm init}\}$ which are easier to compute than $\Phi_{\max}(\mathcal{Q})$. These two functions must satisfy the two following conditions:

(R1) $$\Phi_{\rm lb}(\mathcal{Q}) \le \Phi_{\max}(\mathcal{Q}) \le \Phi_{\rm ub}(\mathcal{Q}) \quad .$$

(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.,*

$$\begin{aligned}&\forall\ \epsilon > 0\ \exists\ \delta > 0\ \forall\ \mathcal{Q} \subseteq \mathcal{Q}_{\rm init}\ \mathrm{size}(\mathcal{Q}) \le \delta \\ &\qquad\Longrightarrow \Phi_{\rm ub}(\mathcal{Q}) - \Phi_{\rm lb}(\mathcal{Q}) \le \epsilon.\end{aligned}$$

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

We describe the algorithm briefly (for a detailed description as well as for a discussion of convergence issues, see Balakrishnan et al. (1991)). In what follows,

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 $\Phi_{\max}(\mathcal{Q}_{\text{init}})$, 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* $\Phi_{\text{ub}}(\mathcal{Q}) = U_k$;
  split $\mathcal{Q}$ *into* $\mathcal{Q}_I$ *and* $\mathcal{Q}_{II}$
  $\mathcal{L}_{k+1} := (\mathcal{L}_k - \{\mathcal{Q}\}) \cup \{\mathcal{Q}_I, \mathcal{Q}_{II}\}$;
  $L_{k+1} := \max_{\mathcal{Q}\in\mathcal{L}_{k+1}} \Phi_{\text{lb}}(\mathcal{Q})$;
  $U_{k+1} := \max_{\mathcal{Q}\in\mathcal{L}_{k+1}} \Phi_{\text{ub}}(\mathcal{Q})$;
  $k = k + 1$;
}

At the end of k iterations, U_k and L_k are upper and lower bounds respectively for $\Phi_{\max}(\mathcal{Q}_{\text{init}})$. Since $\Phi_{\text{lb}}(\mathcal{Q})$ and $\Phi_{\text{ub}}(\mathcal{Q})$ satisfy condition (R2), $U_k - L_k$ is guaranteed to converge to zero.

The algorithm, can be used to compute the minimum of a function f simply by computing the maximum of $-f$.

In order to compute the desired quantities $\mathcal{H}_{\max}$ and $\mathcal{H}_{\min}$ of systems described by equations (1), the task that remains is the computation of appropriate upper and lower bounds for $\mathcal{H}_{\max}$ and $-\mathcal{H}_{\min}$.

Branch and Bound Algorithm for minmax Problems

We now present an extension of the branch and bound algorithm of the previous sections which minimizes over a set of parameters, the maximum of the function over another set of parameters. More precisely, for a function $g(\underline{q}, \overline{q})$ we seek

$$\Psi_{\text{minmax}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) = \min_{\underline{q}\in\underline{\mathcal{Q}}} \max_{\overline{q}\in\overline{\mathcal{Q}}} g(\underline{q}, \overline{q}).$$

The extended branch and bound algorithm needs two functions $\Psi_{\text{lb}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}})$ and $\Psi_{\text{ub}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}})$ defined over $\underline{\mathcal{Q}} \subseteq \underline{\mathcal{Q}}_{\text{init}}$, $\overline{\mathcal{Q}} \subseteq \overline{\mathcal{Q}}_{\text{init}}$ which are easier to compute than $\Psi_{\text{minmax}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}})$. These two functions must satisfy the two following conditions:

(R3) $\Psi_{\text{lb}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) \le \Psi_{\text{minmax}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) \le \Psi_{\text{ub}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}})$.

(R4) As the maximum half-length of the sides of $\overline{\mathcal{Q}}$ and $\underline{\mathcal{Q}}$ denoted by $\text{size}(\overline{\mathcal{Q}})$ and $\text{size}(\underline{\mathcal{Q}})$ respectively go to zero, the difference between upper and lower bounds *uniformly* converges to zero, *i.e.*,

$$\begin{aligned}
&\forall\, \epsilon > 0\ \exists\, \delta > 0 \ \text{ such that}\\
&\forall\, \underline{\mathcal{Q}} \subseteq \underline{\mathcal{Q}}_{\text{init}} \text{ and } \overline{\mathcal{Q}} \subseteq \overline{\mathcal{Q}}_{\text{init}},\\
&\text{size}(\underline{\mathcal{Q}}) \le \delta \text{ and } \text{size}(\overline{\mathcal{Q}}) \le \delta\\
&\quad \Longrightarrow \Psi_{\text{ub}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) - \Psi_{\text{lb}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) \le \epsilon.
\end{aligned}$$

As with the simpler branch and bound algorithm for maximization or minimization, the algorithm starts by computing $\Psi_{\text{lb}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}})$ and $\Psi_{\text{ub}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}})$. If the difference $\Psi_{\text{lb}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}}) - \Psi_{\text{ub}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}}) \le \epsilon$, the algorithm terminates. Otherwise $\underline{\mathcal{Q}}_{\text{init}}$ is partitioned as a union of subrectangles as $\underline{\mathcal{Q}}_{\text{init}} = \underline{\mathcal{Q}}_1 \cup \underline{\mathcal{Q}}_2 \cup \cdots \cup \underline{\mathcal{Q}}_N$, and $\Psi_{\text{lb}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{\text{init}})$ and $\Psi_{\text{ub}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{\text{init}})$, $i = 1, 2, \cdots, N$ are computed. Then

$$\begin{aligned}
&\min_{1\le i\le N} \Psi_{\text{lb}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{\text{init}})\\
&\le \Psi_{\text{minmax}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}}) \le\\
&\min_{1\le i\le N} \Psi_{\text{ub}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{\text{init}}),
\end{aligned}$$

If the difference between these two bounds is small enough, the algorithm terminates. Otherwise any of the subrectangles $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{\text{init}}$ is partitioned into smaller subrectangles as $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{\text{init}} = \underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{i1} \cup \underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{i2} \cup \cdots \cup \underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{iM_i}$, and $\Psi_{\text{lb}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij})$ and $\Psi_{\text{ub}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij})$ are computed. Then

$$\begin{aligned}
&\min_{1\le i\le N} \left\{\max_{1\le j\le M_i} \Psi_{\text{lb}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij})\right\}\\
&\le \Psi_{\text{minmax}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}}) \le\\
&\min_{1\le i\le N} \left\{\max_{1\le j\le M_i} \Psi_{\text{ub}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij})\right\}
\end{aligned}$$

Once more, if the difference between the new bounds is less than or equal to ϵ, the algorithm terminates. Otherwise either $\underline{\mathcal{Q}}_{\text{init}}$ is partitioned into smaller rectangles, or a subrectangle $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{\text{init}}$ is partitioned into smaller subrectangles: in both cases the bounds may be updated. It is also possible to prune those rectangles over which we can establish that $\Psi_{\text{minmax}}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}})$ cannot be achieved.

The general branch and bound algorithm for minmax problems. In the following description, k stands for the iteration index. $\mathcal{L}_k$ denotes a list of N_k rectangle lists. Every rectangle list corresponds to a member $\underline{\mathcal{Q}}_i$ of a partition of $\underline{\mathcal{Q}}_{\text{init}}$ and is therefore denoted by $\ell(\underline{\mathcal{Q}}_i)$. Every subrectangle in $\ell(\underline{\mathcal{Q}}_i)$ is of the form $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}$, with $\overline{\mathcal{Q}}_{ij} \subseteq \overline{\mathcal{Q}}_{\text{init}}$. $M_{i,k}$ stands for the number of subrectangles in the ith list $\ell(\underline{\mathcal{Q}}_i)$ at the end of k iterations. In other words, we have a two-dimensional list of rectangles, first partitioned along the minimizing parameters to yield the rectangle lists, and each of these lists further partitioned along the maximizing parameters. L_k and U_k are lower and upper bounds respectively for $\Psi_{\max}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}})$ at the end of the k-th iteration. Let

$$\begin{aligned}
l_k(\underline{\mathcal{Q}}_i) &= \max_{1\le j\le M_k(i)} \Psi_{\text{lb}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij})\\
u_k(\underline{\mathcal{Q}}_i) &= \max_{1\le j\le M_k(i)} \Psi_{\text{ub}}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij}).
\end{aligned}$$

l_k and u_k are lower and upper bounds for Ψ_{minmax} over $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{\text{init}}$.

The Algorithm.

$k = 0$;
$\ell(\underline{\mathcal{Q}}_{\text{init}}) = \{\underline{\mathcal{Q}}_{\text{init}} \times \overline{\mathcal{Q}}_{\text{init}}\}$;
$\mathcal{L}_0 = \{\ell(\underline{\mathcal{Q}}_{\text{init}})\}$;
$L_0 = \Psi_{\text{lb}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}})$;
$U_0 = \Psi_{\text{ub}}(\underline{\mathcal{Q}}_{\text{init}}, \overline{\mathcal{Q}}_{\text{init}})$;

while $U_k - L_k > \epsilon$ {
pick $\ell(\underline{\mathcal{Q}}_i) \in \mathcal{L}_k$ *such that* $l_k(\underline{\mathcal{Q}}_i) = L_k$;
pick $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij} \in \ell(\underline{\mathcal{Q}}_i)$ *such that* $\Psi_{\rm ub}(\underline{\mathcal{Q}}_i, \overline{\mathcal{Q}}_{ij}) = u_k(\underline{\mathcal{Q}}_i)$;
split $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}$ *along one of the edges of* $\overline{\mathcal{Q}}_{ij}$ *into* $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}^{'}$ *and* $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}^{''}$;
$\ell(\underline{\mathcal{Q}}_i) = (\ell(\underline{\mathcal{Q}}_i) - \underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}) \cup \{\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}^{'}, \underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij}^{''}\}$;
split all $\underline{\mathcal{Q}}_i \times \overline{\mathcal{Q}}_{ij} \in \ell(\underline{\mathcal{Q}}_i)$ *along one of the edges of* $\underline{\mathcal{Q}}_i$ *into* $\underline{\mathcal{Q}}_i^{'} \times \overline{\mathcal{Q}}_{ij}$ *and* $\underline{\mathcal{Q}}_i^{''} \times \overline{\mathcal{Q}}_{ij}$;
$\ell(\underline{\mathcal{Q}}_i^{'}) = \bigcup_j \left\{ \underline{\mathcal{Q}}_i^{'} \times \overline{\mathcal{Q}}_{ij} \right\}$;
$\ell(\underline{\mathcal{Q}}_i^{''}) = \bigcup_j \left\{ \underline{\mathcal{Q}}_i^{''} \times \overline{\mathcal{Q}}_{ij} \right\}$;
$\mathcal{L}_{k+1} := (\mathcal{L}_k - \ell(\underline{\mathcal{Q}}_i)) \cup \{\ell(\underline{\mathcal{Q}}_i^{'}), \ell(\underline{\mathcal{Q}}_i^{''})\}$;
$L_{k+1} := \min_{\ell(\underline{\mathcal{Q}}_i) \in \mathcal{L}_{k+1}} l_k(\underline{\mathcal{Q}}_i)$;
$U_{k+1} := \min_{\ell(\underline{\mathcal{Q}}_i) \in \mathcal{L}_{k+1}} u_k(\underline{\mathcal{Q}}_i)$;
$k = k + 1$;
}

Using simple bounds. We now show how we may obtain bounds $\Psi_{\rm lb}$ and $\Psi_{\rm ub}$ from the bounds for the simple minimization or maximization of a function. The conditions under which these bounds can be used are stated in the following proposition.

Proposition 1: *Given any* $\underline{\mathcal{Q}}$ *and* $\overline{\mathcal{Q}}$ *let*

$$\begin{aligned} \Phi_{\rm lb}(\underline{\mathcal{Q}}, \overline{q}_o) &\leq \min_{\underline{q} \in \underline{\mathcal{Q}}} g(\underline{q}, \overline{q}_o) \\ \Phi_{\rm ub}(\underline{q}_o, \overline{\mathcal{Q}}) &\geq \max_{\overline{q} \in \overline{\mathcal{Q}}} g(\underline{q}_o, \overline{q}) \end{aligned}$$

with $\underline{q}_o$ *and* $\overline{q}_o$ *being any point in* $\underline{\mathcal{Q}}$ *and* $\overline{\mathcal{Q}}$ *respectively.* $\Phi_{\rm lb}$ *and* $\Phi_{\rm ub}$ *are lower and upper bounds for simple minimization and maximization problems respectively. Then*

$$\begin{aligned} \Psi_{\rm lb}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) &= \Phi_{\rm lb}(\underline{\mathcal{Q}}, \overline{q}_o) \\ \Psi_{\rm ub}(\underline{\mathcal{Q}}, \overline{\mathcal{Q}}) &= \Phi_{\rm ub}(\underline{q}_o, \overline{\mathcal{Q}}) \end{aligned} \tag{3}$$

are bounds for $\Psi_{\rm minmax}$ *satisfying* (R3). *Moreover they satisfy* (R4) *if*

1. $g(\underline{q}, \overline{q})$ *is continuous in* $\{(\underline{q}, \overline{q}) : \underline{q} \in \underline{\mathcal{Q}},\ \overline{q} \in \overline{\mathcal{Q}}\}$.
2. $\Phi_{\rm lb}(\underline{\mathcal{Q}}, \overline{q})$ *and* $\min_{\underline{q} \in \underline{\mathcal{Q}}} g(\underline{\mathcal{Q}}, \overline{q})$ *satisfy* (R2) $\forall \overline{q} \in \overline{\mathcal{Q}}$.
3. $\Phi_{\rm ub}(\underline{q}, \overline{\mathcal{Q}})$ *and* $\max_{\overline{q} \in \overline{\mathcal{Q}}} g(\underline{q}, \overline{\mathcal{Q}})$ *satisfy* (R2) $\forall \underline{q} \in \underline{\mathcal{Q}}$.

COMPUTATION OF $\mathcal{H}_{\max}$

Recall that our first objective is to compute

$$\mathcal{H}_{\max}(\mathcal{Q}_{\rm init}) = \max_{q \in \mathcal{Q}_{\rm init}} \|P_{\rm cl}(q)\|_\infty.$$

Then, following the notation used to describe the branch and bound algorithm, we have $f(q) = \|P_{\rm cl}(q)\|_\infty$ and $\Phi_{\max}(\mathcal{Q}) = \mathcal{H}_{\max}(\mathcal{Q})$. The task that remains before the branch and bound algorithm can be applied to this problem is the computation of a lower bound $\Phi_{\rm lb}(\mathcal{Q})$ and an upper bound $\Phi_{\rm ub}(\mathcal{Q})$ for $\mathcal{H}_{\max}$.

Given any parameter rectangle, we may first apply a loop transformation so that we have $\mathcal{Q} = [-1, 1]^m$. Therefore, we will consider only the case $\mathcal{Q} = [-1, 1]^m$. Note that from equation (2), we have $\|\Delta\|_\infty = 1$.

A simple lower bound for $\mathcal{H}_{\max}(\mathcal{Q})$ is just the discrete-time $\mathbf{H}_\infty$-norm of the closed-loop system with the parameter vector set to the midpoint of the parameter region $\mathcal{Q}$:

$$\Phi_{\rm lb}(\mathcal{Q}) = \|P_{\rm cl}(0)\|_\infty = \|P_{zw}\|_\infty. \tag{4}$$

We now describe a simple scheme for computing an upper bound that is based on a small gain based robust stability condition due to Doyle (1982), Safonov (1984) and Balemi et al. (1991).

We define

$$P_\beta = \begin{bmatrix} \dfrac{P_{zw}}{\beta} & \dfrac{P_{zu}}{\sqrt{\beta}} \\ \dfrac{P_{yw}}{\sqrt{\beta}} & P_{yu} \end{bmatrix}, \tag{5}$$

where $\beta > 0$. Then

$$\|P_\beta\|_\infty < 1 \Longrightarrow \sup_{\|\Delta\|_\infty \leq 1} \left\| \left[P_{zw} + P_{zu}\Delta(I - P_{yu}\Delta)^{-1} P_{yw} \right] \right\|_\infty < \beta.$$

Our upper bound is:

$$\Phi_{\rm ub}(\mathcal{Q}) = \inf\left\{\beta : \|P_\beta\|_\infty < 1\right\}, \tag{6}$$

with the convention that the infimum of a function over the empty set is infinity.

The condition in (6) is readily checked by forming an appropriate Hamiltonian matrix and checking its eigenvalues as shown for a similar problem by Boyd et al. (1989). Thus a simple bisection can be used to compute $\Phi_{\rm ub}$.

Of course, more sophisticated bounds can be used. A local optimization procedure can be used to search for a (locally) worst parameter value, which would give a good lower bound. The upper bound can be vastly improved by scaling (see Doyle (1982), Safonov (1982)) or other techniques for approximating the structured singular value (see Fan and Tits 1986).

Bounds for a general rectangle Q. In order to normalize the parameter rectangle Q to be the unit cube $[-1, 1]^m$, we perform a loop transformation of system $P(z)$. Figure 2 demonstrates the loop transformation, where the symbols $\tilde{P}(z)$ and $\tilde{\Delta}$ refer to the loop-transformed system and the normalized perturbation. Let

$$\begin{aligned} K &= \operatorname{diag}(\tfrac{u_1+l_1}{2} I_1, \tfrac{u_2+l_2}{2} I_2, \ldots, \tfrac{u_m+l_m}{2} I_m), \\ F &= \operatorname{diag}(\tfrac{u_1-l_1}{2} I_1, \tfrac{u_2-l_2}{2} I_2, \ldots, \tfrac{u_m-l_m}{2} I_m). \end{aligned}$$

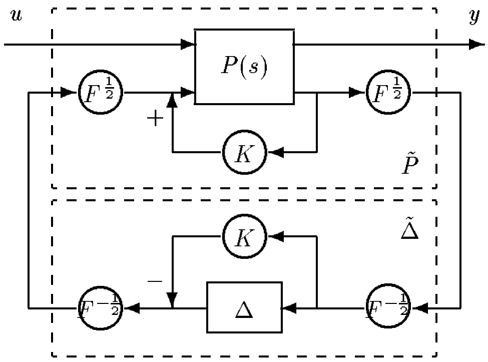

Fig. 2. Loop Transformation.

K is the *offset* corresponding to the rectangle $\mathcal{Q}$, and F, the *scaling*. It is now easily verified that $\tilde{\Delta}$ has the form $\mathrm{diag}(\tilde{q_1}I_1, \tilde{q_2}I_2, ..., \tilde{q_m}I_m)$, where $\tilde{q}$ lies in the m-dimensional unit cube.

COMPUTATION OF $\mathcal{H}_{\min}$

Our objective is now to compute

$$\mathcal{H}_{\min}(\mathcal{Q}_{\mathrm{init}}) = \min_{q \in \mathcal{Q}_{\mathrm{init}}} \|P_{\mathrm{cl}}(q)\|_\infty.$$

Then, following the notation used to describe the branch and bound algorithm, we have $f(q) = -\|P_{\mathrm{cl}}(q)\|_\infty$ and $\Phi_{\max}(\mathcal{Q}) = -\mathcal{H}_{\min}(\mathcal{Q})$. In order to determine a lower bound $\Phi_{\mathrm{ub}}(\mathcal{Q})$ and an upper bound $\Phi_{\mathrm{lb}}(\mathcal{Q})$ for $\Phi_{\max}(\mathcal{Q})$, we again perform a loop transformation so to have $\mathcal{Q} = [-1, 1]^m$.

A simple lower bound for $-\mathcal{H}_{\min}(\mathcal{Q})$ is just the negative of the lower bound for $\mathcal{H}_{\max}(\mathcal{Q})$, *i.e.*

$$\Phi_{\mathrm{lb}}(\mathcal{Q}) = -\|P_{\mathrm{cl}}(0)\|_\infty = -\|P_{zw}\|_\infty. \tag{7}$$

The upper bound is based on the simple norm inequality

$$\|P_{\mathrm{cl}}(q)\|_\infty \geq \|P_{zw}\|_\infty - \frac{\|P_{zu}\|_\infty \|P_{yw}\|_\infty}{1 - \|P_{yu}\|_\infty} = \hat{\Phi}_{ub}$$

and is

$$\Phi_{\mathrm{ub}}(\mathcal{Q}) = \min\{0, \ -\hat{\Phi}_{ub}\} \tag{8}$$

for $\|P_{yu}\|_\infty < 1$, otherwise $\Phi_{\mathrm{ub}}(\mathcal{Q}) = 0$. The result of the branch and bound algorithm for these bounds then yields the negative of the desired quantity $\mathcal{H}_{\min}$.

Again, more sophisticated bounds than the ones preposed above can be used. Tighter bounds will lead to fewer branch and bound iterations; however, the computational cost associated with each iteration will increase. Thus there is a natural trade-off between the tightness of the bounds and their computational (in)efficiency.

COMPUTATION OF $\mathcal{H}_{\mathrm{minmax}}$

The bounds for $\mathcal{H}_{\mathrm{minmax}}$ can be derived from the bounds presented previously. Let us denote by $u = [\underline{u}, \ \overline{u}]'$ and $y = [\underline{y}, \ \overline{y}]'$ the partition of u and y corresponding to the design parameters and the uncertain parameters respectively, and by $P_{\underline{u}\underline{y}} \ldots P_{\overline{y}\,\overline{u}}$ the respective transfer functions.

We will use $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$ to denote unit cubes of the same dimensions as $\underline{\mathcal{Q}}$ and $\overline{\mathcal{Q}}$. Note that the loop transformation presented before allows us to assume, without loss of generality, that the bounds are now computed over $\underline{\mathcal{Q}} = \underline{\mathcal{U}}$ and $\overline{\mathcal{Q}} = \overline{\mathcal{U}}$.

A lower bound for $\mathcal{H}_{\mathrm{minmax}}(\underline{\mathcal{U}}, \overline{\mathcal{U}})$ can be obtained by using a lower bound for $\mathcal{H}_{\min}$, in particular the negative of the upper bound for $-\mathcal{H}_{\min}$ from equation (8). From equation (3) of the previous section, we can choose $\overline{q}_o = 0$, that is the center of the box $\overline{\mathcal{U}}$, and compute directly from equation (8)

$$\Psi_{lb}(\underline{\mathcal{U}}, \overline{\mathcal{U}}) = \max\left\{ \|P_{zw}\|_\infty - \frac{\|P_{z\underline{u}}\|_\infty \|P_{\underline{y}w}\|_\infty}{1 - \|P_{\underline{y}\underline{u}}\|_\infty}, \ 0 \right\} \tag{9}$$

if $\|P_{\underline{y}\underline{u}}\|_\infty < 1$, $\Psi_{lb}(\underline{\mathcal{U}}, \overline{\mathcal{U}}) = 0$ otherwise. An upper bound for $\mathcal{H}_{\mathrm{minmax}}(\underline{\mathcal{U}}, \overline{\mathcal{U}})$ can be obtained by using the upper bound for $\mathcal{H}_{\max}$ of equation (6). From equation (3) of the previous section we can choose $\underline{q}_o = 0$ that is the center of the box $\underline{\mathcal{U}}$, and compute directly from equations (6)

$$\Psi_{ub}(\underline{\mathcal{U}}, \overline{\mathcal{U}}) = \inf\left\{ \beta : \left\| \begin{bmatrix} \frac{P_{zw}}{\beta} & \frac{P_{z\overline{u}}}{\sqrt{\beta}} \\ \frac{P_{\overline{y}w}}{\sqrt{\beta}} & P_{\overline{y}\,\overline{u}} \end{bmatrix} \right\|_\infty < 1 \right\}. \tag{10}$$

We can now apply the branch and bound algorithm together with these bounds to compute $\mathcal{H}_{\mathrm{minmax}}$.

AN EXAMPLE

Consider a discrete-time plant $P(z)$ described by the state-space matrices

$$\begin{aligned} A &= \begin{bmatrix} 0.5 & 1 \\ 0 & -0.5 \end{bmatrix}; \\ B &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix}; \\ D &= \begin{bmatrix} 0 \end{bmatrix}; \end{aligned}$$

with a constant feedback as shown in Figure 3.

Design. Our first objective is to find the value of the parameter δ in [0, 0.5] that minimizes the RMS-gain from r to y. This parameter can be regarded a knob which we use to tune the system. The algorithm gives after 37 iterations a value for $\mathcal{H}_{\min}$ of 1.0 for $\delta = 0.25$. The resulting plot can be seen in Figure 4.

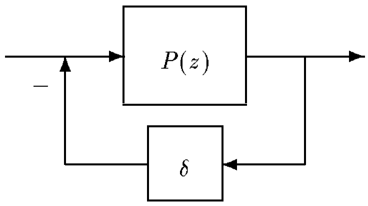

Fig. 3. Discrete-time plant with constant output feedback.

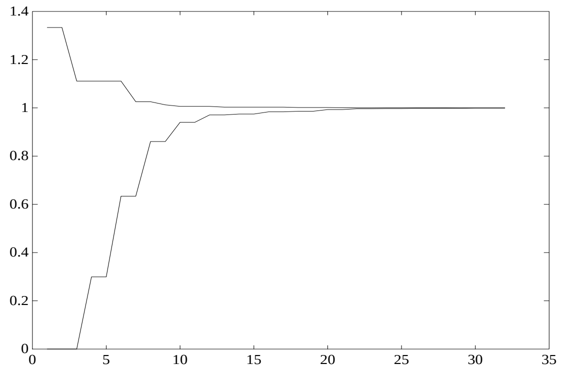

Fig. 4. Bounds on $\mathcal{H}_{\min}$ for a parameter δ in the interval $[0, 0.5]$.

Analysis. Next, we consider the above system with δ fixed at .25. We assume that the system matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & -0.5 \end{bmatrix}$$

depends on the two parameters $a_{11} \in [0.4,\ 0.6]$ and $a_{12} \in [0.9,\ 1.2]$. The parameters now have the interpretation of uncertainties.

The algorithm terminates after 45 iterations yielding a worst-case gain $\mathcal{H}_{\max} = 1.35$ for parameter values $a_{11} = 0.4$ and $a_{12} = 1.2$. We can see the result of the computation in Figure 5.

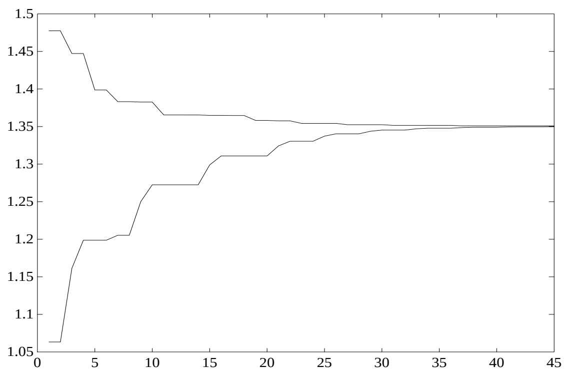

Fig. 5. Bounds on $\mathcal{H}_{\max}$ for the uncertain system and $\delta = 0.25$.

Finally, the minmax computation yields a value of 1.34 for parameters $\delta = .245$, $a_{11} = .4$ and $a_{12} = 1.2$. It is likely that local optimization methods would find this set of parameters fairly quickly. However, unlike our algorithm, local methods have no way of guaranteeing global optimality.

CONCLUSIONS

We have presented branch and algorithms, based on which we may compute the maximum, minimum and minimax discrete-time $\mathbf{H}_\infty$-norm for parameter-dependent discrete-time linear systems. We have also illustrated our algorithm with an example. The algorithm may be used to compute several other interesting quantities for parameter-dependent systems as well.

It can be proven rigorously that the branch and bound algorithm terminates after a finite number of iterations; moreover, it is possible to derive bounds on the number of branch and bound iterations.

REFERENCES

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

Balakrishnan, V. and S. Boyd. Global optimization in control system analysis and design. Leondes, C., editor, *Advances in Control Systems.* Academic Press New York, New York (1992).

Balemi, S., S. Boyd and V. Balakrishnan (1991). Computation of the Maximum $\mathbf{H}_\infty$-norm of Parameter-dependent Linear Systems by a Branch and Bound Algorithm. *Proceedings of MTNS-91.* 305–310 Kobe, Japan.

Boyd, S., V. Balakrishnan and P. Kabamba (1989). 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.

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

Fan, M. K. and A. L. Tits (1986). Characterization and efficient computation of the structured singular value. *IEEE Trans. Aut. Control* AC-31(8), 734–743.

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

Safonov, M. G. (1982). Stability margins of diagonally perturbed multivariable feedback systems. *IEE Proc.* 129-D, 251–256.

Safonov, M. G. (1984). Exact calculation of the multivariable structured-singular-value stability margin. *Proc. IEEE Conf. on Decision and Control.* 1224–1225 Las Vegas, NV.