European Journal of Control (1997) 3:68–80

European
Journal of
Control

Guaranteed Analysis and Optimisation of Parametric Systems with Application to their Stability Degree

O. Didrit[1], L. Jaulin[2] and E. Walter[1]

[1]Laboratoire des Signaux et Systèmes, CNRS-Supélec, Plateau de Moulon, Gif-sur-Yvette, France; [2]Équipe Signaux, Systèmes et Applications, Faculté des Sciences, Université d'Angers, Angers, France

Two splitting-domain algorithms using interval analysis to analyse and optimise a scalar criterion that depends non-linearly on a parameter vector are presented. Their results are guaranteed, and provided with an estimation of their degree of approximation. As an example of application, the stability degree of linear time-invariant parametric systems is considered. It is thus possible to characterise isodegrees in the parameter space and to compute a set guaranteed to contain all values of the parameters that maximise (or minimise) the stability degree as well as an interval guaranteed to contain its optimal value. Several examples of the literature illustrate the efficiency of the method.

Keywords: Branch-and-bound algorithms; Global optimisation; Interval analysis; Splitting-domain algorithms; Stability degree

1. Introduction

In control, as well as in most pure and applied sciences, one is often interested in analysing the effect of the value of some parameter vector $\boldsymbol{p}$ on some real scalar criterion j. The vector $\boldsymbol{p}$ may for instance correspond to the parameters of a model to be estimated at best from experimental data or to those of a controller to be tuned optimally. Except in very special cases, there is no explicit formula for the computation of the global optimisers, i.e., of the values of $\boldsymbol{p}$ that optimise j. Most often, iterative local procedures aimed at improving the value of the criterion are started from some heuristically chosen initial point. No guarantee of convergence to *any* global optimiser can then be given, and global optimisers may be overlooked. It seems therefore particularly important to develop techniques that allow a guaranteed, even if approximate, localisation of *all* global optimisers of j. Interval analysis [20] is a particularly promising tool in this respect, as evidenced for instance by Hansen [11]. Even when they can be computed, the global optimisers of j are often not sufficient, however, and one would like in addition to characterise the set of all parameter vectors such that the value of the criterion remains acceptable. This analysis of a region of the parameter space should also be conducted in a guaranteed way.

We have chosen to illustrate the approach to be presented in the case where the criterion is the stability degree of a parameter-dependent linear system (see, for example, [17,23]), but many other applications can obviously be found in identification and control.

The paper is organised as follows. Section 2 summarises the very few notions of interval analysis needed and introduces the notation. Section 3 proposes two algorithms based on interval analysis for solving analysis and optimisation problems in a guaranteed way. The first one, Isocrit, characterises isocriteria in the parameter space for prespecified levels of j. The second one, Opticrit, is a branch-and-bound algorithm for global optimisation, slightly more sophisticated than Moore and Skelboe's algorithm (see [25]) and which can be seen as a simplification of Hansen's algorithm [14]. Section 4 applies these two algorithms to the stability degree of para-

Correspondence and offprint requests to: E. Walter, Laboratoire des Signaux et Systèmes, CNRS-Supelec, Plateau de Moulon, 91192 Gif-sur-Yvette, France. E-mail: walter@lss.supelec.fr.

Received 4 March 1995; Accepted in revised form 19 December 1996
Recommended by I. Postlethwaite and P. M Frank

metric linear systems. All the examples treated are taken from the literature, to facilitate comparison of results. Only examples where the characteristic polynomial of the system is non-linear in the parameters $\mathbf{p}$ are considered. This rules out methods based on Kharitonov's theorem [14]. It will be seen that ISOCRIT and OPTICRIT compare favourably on these examples with the algorithms previously used to treat them in the literature, in terms of the information provided as well as of computational complexity.

2. Interval Analysis

A *box*, or *vector interval*, $\mathbf{X}$ of $\mathbb{R}^n$ is the cartesian product of n real intervals:

$$\mathbf{X} = [x_1^-, x_1^+] \times \ldots \times [x_n^-, x_n^+] = X_1 \times \ldots \times X_n$$

A *subpaving* of $\mathbb{R}^n$ is the union of a finite set of non-overlapping boxes of $\mathbb{R}^n$. The set of all boxes of $\mathbb{R}^n$ will be denoted by $\mathbb{IR}^n$. All functions $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^p$ to be considered in what follows are assumed to be continuous. The function $\mathbf{F}: \mathbb{IR}^n \to \mathbb{IR}^p$ will be an *inclusion function* of $\mathbf{f}$ if and only if it satisfies $\mathbf{f}(\mathbf{X}) \subset \mathbf{F}(\mathbf{X})$ for any $\mathbf{X}$ of $\mathbb{IR}^n$. It will be *inclusion monotonic* if $\mathbf{X} \subset \mathbf{Y} \Rightarrow \mathbf{F}(\mathbf{X}) \subset \mathbf{F}(\mathbf{Y})$ and *convergent* if $w(\mathbf{X}) \to 0 \Rightarrow w(\mathbf{F}(\mathbf{X})) \to 0$, where $w(\mathbf{X})$ is the *width* of the box $\mathbf{X}$, i.e., the length of its largest side(s). If the effect of rounding is neglected, the computation of an inclusion monotonic and convergent inclusion function associated with any continuous function defined by an explicit formal expression (or finite program) is very simple, and routinely performed by commercially available languages such as C-XSC (see, for example, [16]). Any standard operator or function of real arithmetic can be extended to interval arithmetic in a natural way. For example, if $X = [x^-, x^+]$ and $Y = [y^-, y^+]$ are two intervals, the intervals resulting from their sum, product and exponentiaton can be written as

$$\begin{aligned} X + Y &= [x^- + y^-, x^+ + y^+] \\ XY &= [\min(x^-y^-, x^-y^+, x^+y^-, x^+y^+), \\ &\qquad \max(x^-y^-, x^-y^+, x^+y^-, x^+y^+)] \\ \exp(X) &= [\exp(x^-), \exp(x^+)] \end{aligned}$$

For any function obtained by the composition of elementary operators such as $+$, $-$, $\times$, $\div$, sin, cos, exp, …, it is easy to build an inclusion function by replacing these operators by their interval counterpart. The resulting inclusion function is called a natural inclusion function.

Example 2.1. An inclusion function for the scalar function $f(x, y) = y \cdot \exp(x) + x$ is given by $F(X, Y) = Y \exp(X) + X$. For $X = [-1, 3]$ and $Y = [0, 1]$, one then obtains

$$\begin{aligned} F([-1,3],[0,1]) &= [0,1] \cdot \exp([-1,3]) + [-1,3] \\ &= [0,1] \cdot [e^{-1}, e^3] + [-1,3] \\ &= [0, e^3] + [-1,3] = [-1, 3 + e^3] \end{aligned}$$

In this particular example, $F([-1,3],[0,1]) = f([-1,3],[0,1])$, i.e., the lower and upper bounds of $F([-1,3],[0,1])$ provide respectively the minimum and maximum values of f over $[-1,3] \times [0,1]$. However, this does not hold in the general case, because a natural inclusion function is usually pessimistic. Moreover, in practice, guaranteed results are only obtained if computations are performed with outward rounding in order to take the numerical errors introduced by the computer into account. As a consequence, the image of a vector by an inclusion function is then usually a box with non-zero width.

Although it was originally developed to quantify numerical errors introduced by computers, interval analysis has become a very interesting tool to solve difficult problems like non-linear systems of equations or non-linear optimisation in a global and guaranteed way [11,20]. Most of the resulting algorithms are members of the splitting-domain family : they sequentially split some prior box of interest into sub-boxes, eliminate unfeasible boxes using interval arithmetic, and finally isolate all solutions of the problem into a union of solution boxes. For example, as in Section 4, one may be interested in the characterisation of the set over which a function $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^p$ is componentwise strictly positive. Consider a box $\mathbf{P}$ of $\mathbb{IR}^n$, and let F_i be the ith component of an inclusion function for $\mathbf{f}$, so that

$$F_i(\mathbf{P}) = [f_i^-(\mathbf{P}), f_i^+(\mathbf{P})] \quad i = 1, \ldots, p$$

Then,

- if $f_i^-(\mathbf{P}) > 0\ \forall i \in \{1, \ldots, p\}$, then $\mathbf{f}$ is strictly positive on $\mathbf{P}$, noted $\mathbf{f}(\mathbf{P}) > \mathbf{0}$;
- if $\exists i \in \{1, \ldots, p\}$ such that $f_i^+(\mathbf{P}) < 0$, then $\mathbf{f}$ is never positive on $\mathbf{P}$, noted $\mathbf{f}(\mathbf{P}) \not> \mathbf{0}$;
- if neither $\mathbf{f}(\mathbf{P}) > \mathbf{0}$ nor $\mathbf{f}(\mathbf{P}) \not> 0$, then nothing can in general be said about the sign of $\mathbf{f}$ on $\mathbf{P}$, since the inclusion function is usually pessimistic.

In the context of global optimisation, classical interval algorithms belong to the branch-and-bound family, a special class of splitting-domain algorithms. The main difficulty of classical non-interval algorithms of this family is to compute lower and upper bounds for the criterion to be optimised for any subdomains of the prior domain of interest. Interval analysis just provides a natural way to obtain these bounds for any box via the notion of inclusion function. Consequently, the main contribution of interval techniques to global optimisation is not the well-known concept of branch-and-bound but a simple and natural tool to compute the necessary bounds. The objective of this paper is therefore to show how interval analysis, an innovative branch of numerical analysis almost unknown in the field of automatic control theory, can be put at work to solve important problems much more efficiently than non-interval splitting-domain techniques.

3. Analysis and Optimisation of a Scalar Criterion

The two problems to be considered in this section are the characterisation of isocriteria in the parametric space associated with given levels $j_1, j_2, \ldots$, and the computation of a guaranteed approximation of the set of all global optimisers of $j(\boldsymbol{p})$ and of the associated optimal value of the criterion. The parameter vector $\boldsymbol{p}$ is assumed to belong to some given prior box $\boldsymbol{P}_0$.

3.1. Characterisation of Isocriteria

The isocriterion associated with a given level $j_i, 1 \leq i \leq m$, is the set of all values of $\boldsymbol{p}$ such that $j(\boldsymbol{p}) = j_i$. Let $J(.)$ be an inclusion function for the criterion $j(.)$. For any given box $\boldsymbol{P}$, $J(\boldsymbol{P})$ provides a lower bound $j^-(\boldsymbol{P})$ and an upper bound $j^+(\boldsymbol{P})$ for the value of the criterion over $\boldsymbol{P}$. Two cases must be considered:

(i) if there is no i, $1 \leq i \leq m$, such that $j_i \in J(\boldsymbol{P})$, then $\boldsymbol{P}$ is guaranteed not to contain any point of any of the desired isocriteria; $\boldsymbol{P}$ is said to be *unsuitable*;

(ii) if there is at least one i, $1 \leq i \leq m$, such that $j_i \in J(\boldsymbol{P})$, then $\boldsymbol{P}$ may contain points of the associated isocriteria, in which case it will be said to be *indeterminate*; unless $w(\boldsymbol{P})$ is smaller than a tolerance ϵ_p, it will be bisected into two sub-boxes, on which the process will be iterated.

Case (ii) takes place either when $\boldsymbol{P}$ intersects at least one isocriterion j_i, or when pessimism of the inclusion function prevents decision. Iterations end with a partition of the parameter space in two sets, namely that of all unsuitable boxes, guaranteed not to contain any point belonging to any of the isocriteria j_i, and that of all indeterminate boxes, with width smaller than ϵ_p, which accumulate on the isocriteria $j_i, i = 1, \ldots, m$. The thickness of the associated uncertainty layers, guaranteed to contain all the isocriteria of interest in $\boldsymbol{P}_0$, depends on the required tolerance ϵ_p and on the pessimism of J, related to the computational complexity of j.

Isocrit

Isocrit characterises all isocriteria associated with given levels over a prior box of interest, up to a given tolerance level.

Inputs

$J(.)$	inclusion function for the criterion,
$j_1, \ldots, j_m$	levels of the isocriteria to be characterised,
$\boldsymbol{P}_0$	prior box of interest in the parameter space,
ϵ_p	tolerance on the width of any indeterminate box.

Initialisation

$\mathcal{L} := \{\boldsymbol{P}_0\}$	stack of all boxes still to be studied,
$\mathcal{L}_{\text{uns}} := \{\emptyset\}$	set of all boxes that have been proved unsuitable,
$\mathcal{L}_{\text{ind}} := \{\emptyset\}$	set of all indeterminate boxes with width smaller than ϵ_p.

Iteration

Step 1	If $\mathcal{L}$ is empty, then END, else extract its first element $\boldsymbol{P}$.
Step 2	If there is no $i, 1 \leq i \leq m$, such that $j_i \in J(\boldsymbol{P})$, then put $\boldsymbol{P}$ into $\mathcal{L}_{\text{uns}}$, else if $w(\boldsymbol{P}) \leq \epsilon_p$ then put $\boldsymbol{P}$ into $\mathcal{L}_{\text{ind}}$, else bisect $\boldsymbol{P}$ and put the two resulting boxes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ into $\mathcal{L}$.
Step 3	Go to Step 1.

Outputs

Sets $\mathcal{L}_{\text{uns}}$ and $\mathcal{L}_{\text{ind}}$.

Remark 3.1. $\mathcal{L}$ is organised as a *stack*, i.e., on a first-in-last-out basis. This makes it possible to treat the boxes in depth first, which does not change the total number of iterations but decreases the memory requirements.

Remark 3.2. After a sufficient number of bisections, the resulting sub-boxes will all be either unsuitable or indeterminate with a width smaller than the tolerance ϵ_p. The final size of the list of these indeterminate boxes depends on the precision ϵ_p chosen and on the computational complexity of the criterion. It is often worthwhile to use elaborate inclusion functions (see, for example, [24]) to decrease the size of the list of indeterminate boxes.

3.2. Optimisation of the Criterion

The second issue addressed in this paper is the computation of a guaranteed approximation of the set of all global optimisers of $j(\boldsymbol{p})$, and of its associated optimal value. In what follows, the algorithm is presented for the maximisation case. It is of course trivial to modify it so as to handle minimisations (e.g., by changing the sign of the criterion).

As already pointed out in Section 2, classical interval optimisation algorithms are based on the branch-and-bound approach. This technique is characterised by fixed structure and rules. The two principal rules are sequential splitting of a prior domain of interest into sub-boxes and computation of lower and upper bounds of the criterion over any encountered sub-boxes. The contribution of interval analysis is to provide a simple way to compute these lower and upper bounds for the criterion over any box. If one knows an inclusion function J for the criterion, then, by definition, it satisfies for any box $\boldsymbol{P}$

$$J(\boldsymbol{P}) = [j^-(\boldsymbol{P}), j^+(\boldsymbol{P})] \supset \{j(\boldsymbol{p}), \boldsymbol{p} \in \boldsymbol{P}\},$$

so that

$$j^-(\boldsymbol{P}) \leq \min_{\boldsymbol{p}\in\boldsymbol{P}} j(\boldsymbol{p}) \leq \max_{\boldsymbol{p}\in\boldsymbol{P}} j(\boldsymbol{p}) \leq j^+(\boldsymbol{P}).$$

If the inclusion function is moreover convergent, then $j^-(\boldsymbol{P})$ and $j^+(\boldsymbol{P})$ readily provide two functions that satisfy the convergence conditions of a branch-and-bound algorithm. When the criterion is given explicitly using elementary operators and functions, these functions are obtained in a very simple way using natural interval extensions presented in Section 2. As will be seen in Section 4, a convergent inclusion function can be built even when no explicit formula is available for the corresponding real function.

The Moore–Skelboe algorithm seems to be the simplest interval technique for global optimisation, since it only couples the branch-and-bound structure with the notion of an inclusion function. It is, however, known to be rather inefficient, especially when the evaluation of the criterion is complicated. This is due to the fact that the inclusion function J then becomes very pessimistic for large boxes, so that many bisections are necessary before bounds can be compared with sufficient accuracy.

This is why, as usual in optimisation algorithms based on interval analysis [11], guaranteed evaluations of the criterion at some points will be used regularly in the algorithm presented below in order to increase the speed of convergence. For this purpose, the value $J(\boldsymbol{P})$ of the criterion over some current box $\boldsymbol{P}$ is compared to the best available lower bound for the criterion over the centres of all boxes encountered previously. Let j_c^- be this best lower bound. If $j^+(\boldsymbol{P}) < j_c^-$, then $\boldsymbol{P}$ is eliminated, else $\boldsymbol{P}$ is kept for further bisection. If the lower bound of the criterion at the centre $\boldsymbol{p}_c$ of $\boldsymbol{P}$ turns out to be larger than the current value of j_c^-, then j_c^- is updated. The choice of the centre of $\boldsymbol{P}$ as point argument for the evaluation is arbitrary, and a local optimisation procedure could be used to improve it. Numerical experimentation tends so far to indicate that the amelioration is marginal.

OPTICRIT

OPTICRIT can be seen as the core of Hansen's algorithm [11], when all steps using inclusion functions for the gradient and Hessian of j have been eliminated. We have chosen this approach because these derivative functions are not always easy to obtain, and the optimisation of the stability degree will provide a case in point. For cases where efficient inclusion functions are readily available, the use of Hansen's algorithm should provide better performances. OPTICRIT computes a set of boxes $\mathcal{L}$, the union of which is guaranteed to contain all global optimisers in the prior box of interest $\boldsymbol{P}_0$, and the lower bound j_c^-. From these quantities, it will then be easy to bracket the maximal value j_{max} of the criterion $j(\boldsymbol{p})$ over $\boldsymbol{P}_0$. In the description of the algorithm, $\boldsymbol{p}_{ic}$ stands for the centre of the box $\boldsymbol{P}_i$.

Inputs

$J(.)$	inclusion function for the criterion to be *maximised*,
$\boldsymbol{P}_0$	prior box of interest in the parameter space,
ϵ_p	tolerance on the width of any indeterminate box.

Initialisation

$j_c^- := j^-(\boldsymbol{p}_{0c})$	initial best lower bound for the value of j_{max},
$\mathcal{L} := \{\boldsymbol{P}_0\}$	

Iteration

Step 1	If all elements of $\mathcal{L}$ have a width smaller than ϵ_p, then END. Else extract the first box $\boldsymbol{P}$ with a width larger than ϵ_p from $\mathcal{L}$.
Step 2	Bisect $\boldsymbol{P}$ into $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.

Step 3 If $j^+(\boldsymbol{P}_1) \geq j_c^-$, then put $\boldsymbol{P}_1$ into $\mathcal{L}$, $j_c^- := \max(j_c^-, j^-(\boldsymbol{p}_{1c}))$.
Step 4 If $j^+(\boldsymbol{P}_2) \geq j_c^-$, then put $\boldsymbol{P}_2$ into $\mathcal{L}$, $j_c^- := \max(j_c^-, j^-(\boldsymbol{p}_{2c}))$.
Step 5 If j_c^- has been improved at Step 3 or 4, then eliminate all boxes $\boldsymbol{Q}$ such that $j^+(\boldsymbol{Q}) < j_c^-$ from $\mathcal{L}$. Go to Step 1.

Outputs
List $\mathcal{L}$ and j_c^-.

Remark 3.3. The ordering of the boxes in $\mathcal{L}$ has a considerable influence on the efficiency of the algorithm. Numerical experimentation has shown that a very good policy, in terms of the number of boxes generated, consists in sorting the boxes in decreasing order with respect firstly to their upper bound $j^+(\boldsymbol{P})$, secondly to their lower bound $j^-(\boldsymbol{P})$ (for equal upper bounds), and lastly to the lower bound computed in their centre (for equal upper and lower bounds). This ordering is similar to those advocated by Hansen [11], and Balakhrisnan et al. [3]. The box to be extracted from $\mathcal{L}$ on Step 1 is then the one that looks the most promising. With this policy, the algorithm avoids bisecting less suitable boxes which will later be eliminated very simply at Step 5.

Remark 3.4. For any given tolerance ϵ_p, OPTICRIT will stop after a finite number of steps. Moreover, since the lower bound j_c^- is computed using outward rounding interval arithmetic, the results are guaranteed. Note that the number of boxes in $\mathcal{L}$ may differ from the number of global optimisers, since there may be several optimisers in a given box or some boxes may contain no optimiser because of the pessimistic nature of the inclusion function.

From the outputs of OPTICRIT, it is easy to bracket the optimal value $j_{\max}$ of the criterion as:

$$j_c^- \leq j_{max} \leq \max_{\boldsymbol{P}\in\mathcal{L}} j^+(\boldsymbol{P}).$$

4. Application to the Degree of Stability

The last decade has seen an explosion in the number of publications about robust stability and performances of uncertain systems with structured parametric perturbations. For reviews of the main concepts, methods and results in this domain, see, for example, [5,12,19,22,27].

The stability and many dynamical properties of linear time-invariant finite-dimensional systems are conveniently studied via the computation of their characteristic polynomial. Special methods are required when this characteristic polynomial depends on some parameter vector $\boldsymbol{p}$ (see [9] for an early paper on this subject). The seminal theorem of Kharitonov [14] on the stability of interval polynomials has been followed by an impressive series of generalisations and applications (see the survey [6]) under the generic name of extreme-point results. All these methods tend to determine a set T of deterministic polynomials, as simple as possible, such that stability over T implies stability over the whole region of uncertainty for $\boldsymbol{p}$. In the case of interval polynomials, it is well known that T consists of four polynomials. However, in spite of this remarkable progress, extreme-point results still apply to a very limited class of problems. Most non-linearly parametrised polynomials cannot be handled. In the multilinear case, the set T can only be reduced to extremal manifolds which remain to be studied [7], e.g., with methods similar to that proposed here.

If one excludes systematic exploration over a grid and random scanning, which are both computer intensive and produce no guaranteed results, two main classes of approaches are available to deal with non-linearly parametrised polynomials $\mathcal{A}_{\boldsymbol{p}}(s)$.

The first one is the so-called *parameter space method* [2], which uses the properties of $\mathcal{A}_{\boldsymbol{p}}(j\omega)$ to obtain an explicit graphic determination of the boundaries of the stability region with respect to two components of the parameter vector. If $\dim \boldsymbol{p} > 2$, gridding must be used to make the method applicable, and the guaranteed nature of the result is lost. Recently, Kaminski and Djaferis [13] proposed another method also based on the study of $\mathcal{A}_{\boldsymbol{p}}(j\omega)$. In both cases, the preparation of the computations to be performed may involve considerable algebraic manipulations.

The second class, to which the algorithms presented in this paper belong, consists of *splitting domain algorithms*, where the prior region of interest is cut into subdomains so as to facilitate exploration. With this approach, remarkable results have been achieved on the computation of stability radii [8,26]. More recently, Walter and Jaulin [29] have used a set-inversion algorithm based on interval analysis to characterise the stability domain for arbitrary parametrisations. Interval analysis has also been used to obtain guaranteed results about the stability and performances of uncertain models by Kolev [18] and Fioro et al. [10].

The criterion $j(\boldsymbol{p})$ to be considered here is the *stability degree*, originally defined to quantify the quality of the transient response [23]. This criterion has already been used by Balakrishnan et al. [3] to compute the minimum stability degree of parameter-dependent linear systems with a branch-and-bound algorithm using the H_∞ norm, and we shall compare our results with theirs in Section 4.

Characteristic polynomials can be used to study the properties of a very large class of systems (SISO or MIMO, continuous- or discrete-time, described by state equation or transfer function). To simplify notation and exposition, only continuous-time systems will be considered, but transposition to discrete time is trivial. For continuous-time systems, the stability degree is the opposite of the largest real part of the roots of the characteristic polynomial. Consider a linear time-invariant parameter-dependent system $\mathcal{M}(\boldsymbol{p})$ and its characteristic polynomial

$$\mathcal{A}_{\boldsymbol{p}}(s) = s^n + a_{n-1}(\boldsymbol{p})s^{n-1} + \ldots + a_1(\boldsymbol{p})s + a_0(\boldsymbol{p}),$$

where each coefficient a_i may depend non-linearly on the parameter vector $\boldsymbol{p}$. Again to simplify exposition, the degree n of $\mathcal{A}_{\boldsymbol{p}}(s)$ is supposed not to vary with $\boldsymbol{p}$, so that $\mathcal{A}_{\boldsymbol{p}}(s)$ can arbitrarily be taken monic ($a_n = 1$).

Definition 4.1. The system $\mathcal{M}(\boldsymbol{p})$ is δ-stable if the largest real part of the roots of $\mathcal{A}_{\boldsymbol{p}}(s)$ is lower than $-\delta$.

The stability degree $j(\boldsymbol{p})$ of $\mathcal{M}(\boldsymbol{p})$ is then the largest value of δ, noted $\delta_M(\boldsymbol{p})$, for which $\mathcal{M}(\boldsymbol{p})$ is δ-stable. δ-stability implies that the transient response of the system converges more quickly than $\exp(-\delta t)$. It is a special case of Γ-stability, and the results presented here could be extended to other cases for which a criterion could be made explicit [28]. δ-stability is equivalent to the asymptotic stability of the shifted polynomial

$$\begin{aligned}\mathcal{B}_{\boldsymbol{p},\delta}(s) &= \mathcal{A}_{\boldsymbol{p}}(s-\delta)\\ &= s^n + b_{n-1}(\boldsymbol{p},\delta)s^{n-1} + \ldots + b_1(\boldsymbol{p},\delta)s + b_0(\boldsymbol{p},\delta)\end{aligned}$$

so that it will be easy to evaluate the criterion $j(\boldsymbol{p})$ via the computation of the Routh table of the polynomial $\mathcal{B}_{\boldsymbol{p},\delta}(s)$. Since this polynomial is monic, the first entry of the first column of the Routh table is equal to 1. Hence the following definition.

Definition 4.2. The Routh function of a polynomial, noted $\boldsymbol{r}$, is the vector whose ith component is the $(i+1)$th element of the first column of the Routh table of the polynomial, with $1 \le i \le n$.

If $\boldsymbol{r}(\boldsymbol{p},\delta)$ is the Routh function of $\mathcal{B}_{\boldsymbol{p},\delta}(s)$, then

$$\mathcal{M}(\boldsymbol{p}) \textit{ is } \delta\textit{-stable} \Leftrightarrow \boldsymbol{r}(\boldsymbol{p},\delta) > \mathbf{0}$$

where the inequality is to be understood componentwise. The criterion $j(\boldsymbol{p})$ can thus be computed as the largest value of δ such that $\boldsymbol{r}(\boldsymbol{p},\delta) > \mathbf{0}$. An example is now considered to illustrate the computation of the Routh function and of an inclusion function for it, that will be used to apply Isocrit and Opticrit.

Example 4.1. Consider the characteristic polynomial

$$\mathcal{A}_{\boldsymbol{p}}(s) = s^3 + s^2 + (p_1^2 + p_2^2 + 1)s + 1$$

that depends on the two-dimensional parameter vector $\boldsymbol{p} = (p_1, p_2)^T$. For any real δ, the shifted polynomial is defined by

$$\begin{aligned}\mathcal{B}_{\boldsymbol{p},\delta}(s) &= \mathcal{A}_{\boldsymbol{p}}(s-\delta)\\ &= s^3 + (-3\delta + 1)s^2\\ &\quad + (3\delta^2 - 2\delta + 1 + p_2^2 + p_1^2)s - \delta^3 + \delta^2\\ &\quad - (1 + p_2^2 + p_1^2)\delta + 1\end{aligned}$$

for which the Routh table can easily be constructed so that the Routh function of $\mathcal{A}_{\boldsymbol{p}}(s)$ is given by

$$\boldsymbol{r}(\boldsymbol{p},\delta) = \begin{pmatrix} -3\delta + 1 \\ -8\delta^3 + 8\delta^2 - (2p_1^2 + 2p_2^2 + 4)\delta + p_2^2 + p_1^2 \\ -\delta^3 + \delta^2 - (1 + p_2^2 + p_1^2)\delta + 1 \end{pmatrix}$$

Each component of $\boldsymbol{r}(\boldsymbol{p},\delta)$ is a function of $\boldsymbol{p}$ and δ. Therefore, if one firstly considers $\boldsymbol{r}$ as a function of only $\boldsymbol{p}$, an inclusion function of $\boldsymbol{r}$ when $\boldsymbol{p}$ belongs to a box $\boldsymbol{P} = (P_1, P_2)^T$ is directly obtained using natural interval extensions

$$\boldsymbol{R}(\boldsymbol{P},\delta) = \begin{pmatrix} -3\delta + 1 \\ -8\delta^3 + 8\delta^2 - (2P_1^2 + 2P_2^2 + 4)\delta + P_2^2 + P_1^2 \\ -\delta^3 + \delta^2 - (1 + P_2^2 + P_1^2)\delta + 1 \end{pmatrix}$$

As a consequence, one can deduce global properties for the sign of the Routh function when the parameters describe intervals. For example, for $P_1 = [0,1]$ and $P_2 = [1,2]$, we have for any value of δ using natural interval extensions defined in Section 2

$$\boldsymbol{R}(\boldsymbol{P},\delta) = \begin{pmatrix} -3\delta + 1 \\ -8\delta^3 + 8\delta^2 - [6,14]\delta + [1,5] \\ -\delta^3 + \delta^2 - [2,6]\delta + 1 \end{pmatrix}$$

Then, if δ now takes fixed real values, we obtain $\boldsymbol{R}(\boldsymbol{P},0) = (1,[1,5],1)^{\mathrm{T}}$, which shows that $\boldsymbol{r}$ is strictly positive for $\delta = 0$ and for any values of p_1 and p_2 respectively in $[0,1]$ and $[1,2]$. This implies that the polynomial $\mathcal{A}_{\boldsymbol{p}}$ is stable for all these values of the parameters. On the other hand, $\boldsymbol{R}(\boldsymbol{P},1) = (-2,[-13,-1],[-5,-1])^T$ shows that $\mathcal{A}_{\boldsymbol{p}}$ is never 1-stable for any value of the parameters in these intervals.

4.1. Characterisation of Isodegrees

The following notions will be used to adapt Isocrit to the computation of isodegrees of stability associated to given positive levels $\delta_1 > \delta_2 > \ldots > \delta_m$.

Definition 4.3. The ith layer of stability, $1 \le i \le m$, is defined by

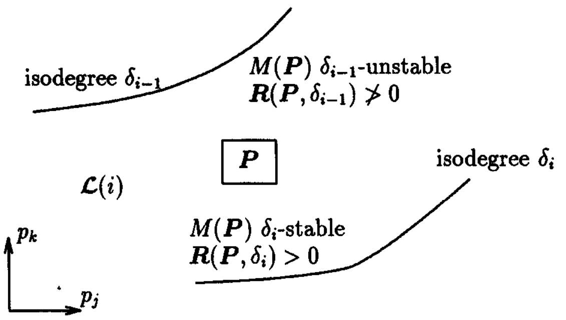

Fig. 1. Layers of stability and associated isodegrees.

$$\begin{aligned}\mathcal{L}(i) &= \{\boldsymbol{p} \text{ such that } \mathcal{M}(\boldsymbol{p}) \text{ is } \delta_i\text{-stable and } \delta_{i-1}\text{-unstable}\} \\ &= \{\boldsymbol{p} \text{ such that } \boldsymbol{r}(\boldsymbol{p}, \delta_i) > \boldsymbol{0} \text{ and } \boldsymbol{r}(\boldsymbol{p}, \delta_{i-1}) \not> \boldsymbol{0}\}\end{aligned}$$

As shown by Fig. 1, the layer of stability $\mathcal{L}(i)$ is between the isodegrees δ_i and δ_{i-1}.

Layers of stability are useful for the characterisation of isodegrees, because it is easier to test whether a parameter box $\boldsymbol{P}$ belongs to a given layer of stability (defined by explicit inequalities) than to check, as in Step 2 of ISOCRIT, that $\boldsymbol{P}$ does not intersect any isodegree.

$$\boldsymbol{P} \in \mathcal{L}(i) \Leftrightarrow \mathcal{M}(\boldsymbol{P}) \text{ is } \delta_i\text{-stable and } \delta_{i-1}\text{-unstable.}$$

A sufficient condition for $\boldsymbol{P}$ to belong to $\mathcal{L}(i)$ is therefore

$$\boldsymbol{R}(\boldsymbol{P}, \delta_i) > \boldsymbol{0} \text{ and } \boldsymbol{R}(\boldsymbol{P}, \delta_{i-1}) \not> \boldsymbol{0}$$

where $\boldsymbol{R}$ is an inclusion function of the Routh function.

It suffices then to modify Step 2 of ISOCRIT so that a box $\boldsymbol{P}$ is proved to be unsuitable if there exists i, $1 \leq i \leq m$, such that $\boldsymbol{P} \in \mathcal{L}(i)$. For this purpose, each box $\boldsymbol{P}$ appearing in the algorithm is assigned two indices, namely the index i_{last} of the last δ_i such that $\mathcal{A}_{\boldsymbol{P}}$ is proved not to be δ_i-stable, i.e., $\boldsymbol{R}(\boldsymbol{P}, \delta_{i_{\text{last}}}) \not> \boldsymbol{0}$, and the index i_{first} of the first δ_i such that $\mathcal{A}_{\boldsymbol{P}}$ is proved to be δ_i-stable, i.e., $\boldsymbol{R}(\boldsymbol{P}, \delta_{i_{\text{first}}}) > \boldsymbol{0}$.

If $i_{\text{first}} = i_{\text{last}} + 1$, then $\boldsymbol{R}(\boldsymbol{P}, \delta_{i_{\text{first}}}) > \boldsymbol{0}$ and $\boldsymbol{R}(\boldsymbol{P}, \delta_{i_{\text{first}}-1}) \not> \boldsymbol{0}$. Consequently $\boldsymbol{P}$ is proved to belong to the layer of stability $\mathcal{L}(i_{\text{first}})$ and is therefore unsuitable. Else, if $w(\boldsymbol{P}) > \epsilon_p$, then it is bisected into two sub-boxes. According to the property of monotonic inclusion of $\boldsymbol{R}$, each sub-box can be initially assigned the indices i_{last} and i_{first} of $\boldsymbol{P}$, expecting that the computation of $\boldsymbol{R}$ on these boxes will respectively increase and decrease these indices.

The following two examples have been treated with an ADA implementation of ISOCRIT including outward rounding on a personal computer with a Pentium 90 MHz processor.

Example 4.2. Consider the characteristic polynomial

$$\begin{aligned}\mathcal{A}_p(s) = {} & s^3 + (p_1 + p_2 + 1)s^2 + (p_1 + p_2 + 3)s \\ & + (1 + R^2 + 6p_1 + 6p_2 + 2p_1 p_2)\end{aligned}$$

It is associated with the benchmark example of Ackermann et al. [1], used to review methods for characterising domains of stability when the characteristic polynomial is non-linear in the parameters, so that Kharitonov's results do not apply. When p_1 and p_2 are positive, the unstability domain corresponds to a disk with radius R, so for $R = 0$ it becomes a singleton, which makes this example impossible to treat by random or grid scanning. ISOCRIT has been applied to the cases $R = 0.5$ and $R = 0$, for two values of the isodegree, namely $\delta_1 = 0.1$ and $\delta_2 = 0$. In all cases, the initial box of interest was chosen as $\boldsymbol{P}_0 = [-3, 7] \times [-3, 7]$, i.e., much larger than that considered in [1], and $\epsilon_p = 0.05$. For $\delta = 0$, a specific inclusion function was used, in which each parameter p_1 or p_2 appears at most once in each component of the Routh function. This made the inclusion function minimally pessimistic but could not be used in the general case $\delta \neq 0$.

Figure 2 presents the results obtained for $R = 0.5$ in 3.7 s. In that case, the instability region does not reduce to a singleton, and ISOCRIT has been able to construct boxes guaranteed to belong to it.

Figure 3 presents the results obtained for $R = 0$ in 3.4 s. It is of course no longer possible to obtain unstable boxes since the unstable region reduces to a point, but an uncertainty region that may contain unstable points has been detected. This region reduces to a single very small box.

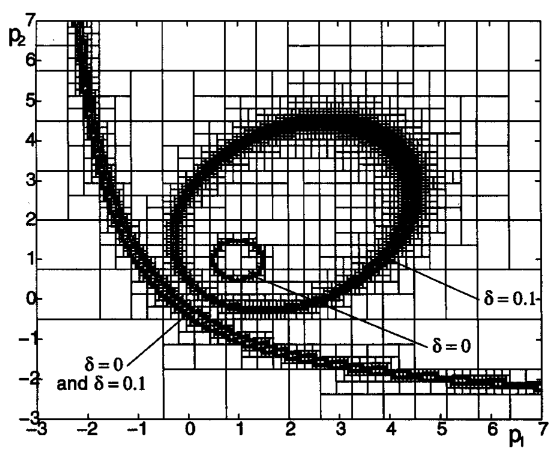

Fig. 2. Isodegrees in the (p_1, p_2) space for $R = 0.5$ in Example 4.2 (in black).

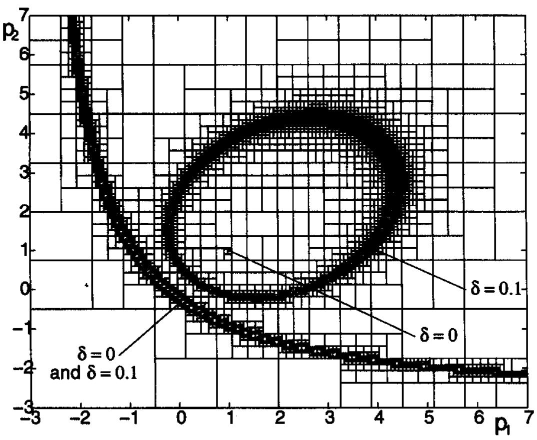

Fig. 3. Isodegrees in the (p_1, p_2) space for $R = 0$ in Example 4.2 (in black)

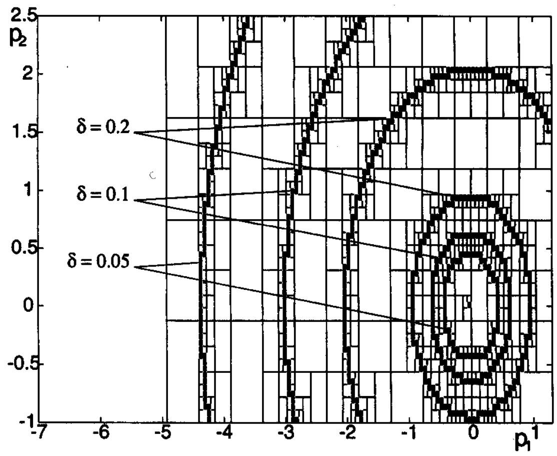

Fig. 4. Isodegrees in the (p_1, p_2) space for Example 4.3 (in black).

Note that this case has also been treated by Kiendl and Michalske [15] with a branch-and-bound algorithm but with a Lyapunov approach. Their method requires the characteristic polynomial to be linear in its parameters, which implied a change of variables not always possible.

Example 4.3. The characteristic polynomial already studied in Example 4.1

$$\mathcal{A}_p(s) = s^3 + s^2 + (p_1^2 + p_2^2 + 1)s + 1$$

has been considered by Kokame and Mori [17]. It corresponds to a linear state-space system, where the autonomous equation is given by:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & -p_1 \\ 1 & 0 & -p_2 \\ p_1 & p_2 & 1 \end{bmatrix} \boldsymbol{x}$$

As for Ackerman's example with $R = 0$, it is easy to show with the Routh table that this system is stable everywhere but in $p_1 = p_2 = 0$. The isodegree for any given value of δ can be computed explicitly. It consists of a pair of inner and outer circles centred on the origin and whose radii respectively increase and decrease with respect to δ. Isocrit has been applied for the isodegrees $\delta = 0, 0.05, 0.1$, and 0.2. As in the initial paper, the initial box of interest is $\boldsymbol{P}_0 = [-7, 1.3] \times [-1, 2.5]$. Figure 4 clearly evidences the isodegrees, as obtained in 0.3 s. for $\epsilon_p = 0.1$.

Note that the uncertainty layer associated with $\delta = 0$ reduces in $\boldsymbol{P}_0$ to a single box enclosing the origin. The analytical simplicity of $\mathcal{A}_p$ has allowed Isocrit to quickly partition the initial box into layers of stability. Large boxes have been classified without being bisected. The results obtained are much more complete than with the algorithm proposed by Kokame and Mori, which only finds one point in the parameter space guaranteed to be δ-stable for a single value of δ.

4.2. Optimisation of the Stability Degree

Opticrit will now be used to find a subpaving containing all global optimisers of the stability degree of a parametric system. This requires the evaluation of an inclusion function of the stability degree over a box $\boldsymbol{P}$. (Here, the values of interest of the criterion are not fixed *a priori*, contrary to the case of the characterisation of isodegrees, and this is why stability layers can no longer be used.)

The stability degree $\delta_M(\boldsymbol{p})$ associated with a parameter vector $\boldsymbol{p}$ is unique. On the other hand, when $\boldsymbol{p}$ is only known to belong to a box $\boldsymbol{P}$, then it is associated with a set $\delta_M(\boldsymbol{P})$ of stability degrees, defined by

$$\delta_M(\boldsymbol{P}) = \{\delta_M(\boldsymbol{p}), \boldsymbol{p} \in \boldsymbol{P}\}$$

Since $\boldsymbol{P}$ is a box and δ_M is continuous with respect to $\boldsymbol{p}$, $\delta_M(\boldsymbol{P})$ is an interval

$$\delta_M(\boldsymbol{P}) = [\delta_{M\min}(\boldsymbol{P}), \delta_{M\max}(\boldsymbol{P})]$$

In contrast to the Routh function, no explicit expression exists in general for the stability degree $\delta_M(\boldsymbol{p})$ of a parameter vector $\boldsymbol{p}$, so no inclusion function can be directly obtained using natural interval extensions. Our objective is then to compute an inclusion function $\Delta_M(\boldsymbol{P})$ by approximating $\delta_M(\boldsymbol{P})$ by an interval guaranteed to contain it. For this purpose, the iterative algorithm Degbound is used to estimate a lower bound of $\delta_{M\min}(\boldsymbol{P})$ and an upper bound of $\delta_{M\max}(\boldsymbol{P})$. This algorithm uses the following property of the Routh function:

δ_M is the stability degree for $\boldsymbol{p}$

$\Leftrightarrow \boldsymbol{r}(\boldsymbol{p},\delta) > \boldsymbol{0}$ if $\delta < \delta_M$ and $\boldsymbol{r}(\boldsymbol{p},\delta) \not> \boldsymbol{0}$ if $\delta > \delta_M$

Degbound

Only the part of Degbound that computes a lower bound δ_M^- of $\delta_{M\min}$ on some box of interest $\boldsymbol{P}$ will be presented. The part computing an upper bound of $\delta_{M\max}$ is similar.

Inputs

- $\boldsymbol{P}$ box of interest,
- $\boldsymbol{R}$ inclusion function of the Routh function,
- m prior lower bound for $\delta_M^-(\boldsymbol{P})$,
- M prior upper bound for $\delta_M^-(\boldsymbol{P})$,
- ϵ_δ smallest feasible increment for δ.

Initialisation

- $progress := M - m$ initial step of the dichotomy,
- $\delta := m$ initial value for δ_M^-

Iteration

- Step 1 $progress := progress/2$.
- Step 2 If $\boldsymbol{R}(\boldsymbol{P}, \delta + progress) > \boldsymbol{0}$ then $\delta := \delta + progress$.
- Step 3 If $progress \leq \epsilon_\delta$ then END. Else go to Step 1.

Output

$\delta_M^-(\boldsymbol{P}) := \delta$

Remark 4.1. M and m, initial values respectively for the upper and lower bounds, are inputs of this procedure, to be provided by Opticrit. When Opticrit bisects a box $\boldsymbol{P}$ into $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, it must then call Degbound to evaluate $\Delta_M(\boldsymbol{P}_1)$ and $\Delta_M(\boldsymbol{P}_2)$, and possibly $\delta_M^-(\boldsymbol{p}_{1c})$ and $\delta_M^-(\boldsymbol{p}_{2c})$. Then, for each of both sub-boxes, $\delta_M^-(\boldsymbol{P})$ and $\delta_M^+(\boldsymbol{P})$ are used in place of m and M for these evaluations. For the initial box $\boldsymbol{P}_0$, the values of $|m|$ and $|M|$ are arbitrarily taken as very large ($m < 0, M > 0$).

Remark 4.2. The final precision on the value of $\delta_{M\min}(\boldsymbol{P})$ is generally not equal to ϵ_δ. The distance to the exact value depends on the inclusion function used for $\boldsymbol{r}$. However, it is known that the smaller the box $\boldsymbol{P}$ is, the sharper the estimation of $\boldsymbol{R}(\boldsymbol{P},\delta)$ will be, and the closer the precision on the value of $\delta_{M\min}(\boldsymbol{P})$ will be from ϵ_δ.

The algorithm Degbound can be seen as an interval dichotomy which uses the property of the inclusion functions for the stability degree and Routh function that is illustrated by Fig. 5.

Degbound finally provides an inclusion function for the stability degree

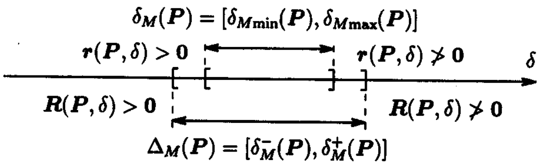

Fig. 5. Routh function and stability degree.

$$\Delta_M(\boldsymbol{P}) = [\delta_M^-(\boldsymbol{P}), \delta_M^+(\boldsymbol{P})] \supset [\delta_{M\min}(\boldsymbol{P}), \delta_{M\max}(\boldsymbol{P})]$$

In fact, only one bound, the lower for a maximisation, the upper for a minimisation, is required by Opticrit, but Degbound is also used to obtain sharp and guaranteed bounds for the value of the stability degree at the centre of the boxes. As a consequence, an iteration of Opticrit requires at least two executions of Degbound. This subroutine is based on a simple dichotomic structure and the computation of an inclusion function for a Routh function is straightforward. From the outputs of Opticrit, it is easy to bracket the maximal value $\delta_{\max}$ of the stability degree as

$$\delta_{Mc}^- \leq \delta_{\max} \leq \max_{\boldsymbol{P}\in\mathcal{L}} \delta_M^+(\boldsymbol{P})$$

The two following examples have been treated using an Ada implementation of Opticrit and Degbound including outward rounding, also on a personal computer with a Pentium 90 MHz processor.

Example 4.4 Consider again the characteristic polynomial of Example 4.3,

$$\mathcal{A}_{\boldsymbol{p}}(s) = s^3 + s^2 + (p_1^2 + p_2^2 + 1)s + 1$$

The two circles associated with the isodegree δ move towards each other when δ increases, until they merge at $\delta = 1/3$, which corresponds to the maximal stability degree achievable. Opticrit is applied on the same prior box of interest as previously, with $\epsilon_p = 0.05$ and $\epsilon_\delta = 0.001$. In accordance with these theoretical results, it brackets the optimal value for δ between 0.3333 and 0.3339 in 588 iterations. Moreover, 2.5 s. suffice to bracket the set of all values of the parameters that achieve this optimal stability degree (Fig. 6), again in accordance with the theoretical results.

Example 4.5. Consider a family of state-space systems described by:

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

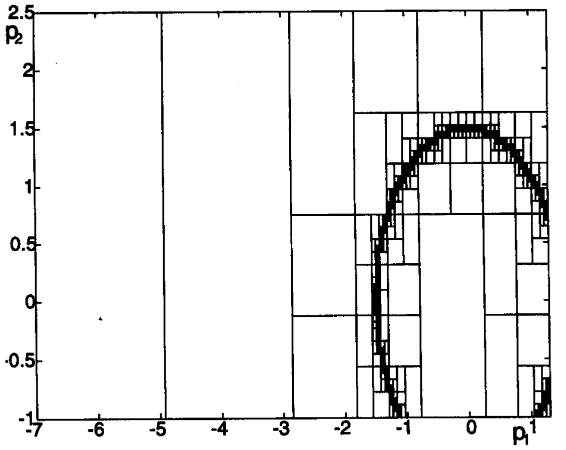

Fig. 6. Subpaving containing all global maximisers of the stability degree in the (p_1, p_2) space for Example 4.4 (in black).

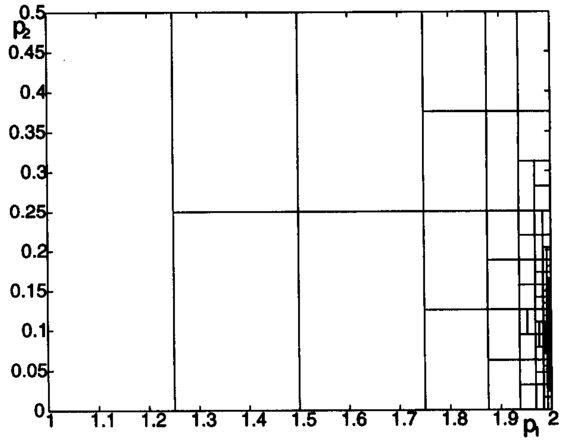

Fig. 7. Subpaving containing all global minimisers of the stability degree in the (p_1, p_2) space for Example 4.5 (in black).

where $1 \leq q_1 \leq 2$ and $0 \leq q_2 \leq 0.5$. The problem of interest here is to *minimise* the stability degree, in order to find the worse values of the uncertain parameters.

This example have been studied among others by Balakrishnan et al. [3] also using a branch-and-bound structure but with totally different arguments to compute lower and upper bounds of the stability degree over a box. Their method requires a transformation of the system description into a block diagram suitable for treatment with the H_∞ approach. The computation of this standard form is known to be straightforward but tedious and conservative in most cases of non-linear structure. Moreover, an overparametrisation is necessary in general so that the dimension of the problem notably increases. For example, the considered two-parameter system leads to a six-parameter formulation. Since computations of any branch-and-bound algorithms are known to increase, in the worst case, exponentially with the number of parameters, such an overparametrisation is really a handicap. Note that, as with many robust stability problems [21], the computation of the optimum stability degree is known to be an NP–hard problem, so that, as stressed by Balakrishnan et al. [3], no algorithm would perform substantially better than a branch-and-bound algorithm on such a computation.

With the tolerance parameters ϵ_δ and ϵ_p equal to 0.001, the following bracketing is obtained

$$-2.01623 \leq \delta_{min} \leq -2.01451$$

after the exploration of 247 sub-boxes (Fig. 7), and the same number of iterations.

The same performance measures as in [3] have been used, namely the bounds on the value of the criterion (Fig. 8), the percentage of pruned volume (Fig. 9) and the number of active rectangles (Fig. 10). The results obtained by programs developed by Balakrishnan et al. in Matlab during the same number of iterations have been superimposed to facilitate comparison. After 247 iterations, OPTICRIT reached a precision of 1.72×10^{-3} (9.28×10^{-3} in [3]), pruned 99.98% of the initial volume (94.18% in [3]), and left 91 active boxes at the end (78 in [3] but with a larger total volume). The programs of Balakrishnan et al. need 653 iterations to achieve a comparable precision, 3570 iterations to prune a comparable volume.

The technique used in [3] to compute the bounds for the branch-and-bound algorithm requires, as for OPTICRIT, an evaluation of the stability degree at the centre of the box and a dichotomy (real in [3], interval and thus guaranteed in OPTICRIT). The complexity of an iteration of both approaches is therefore comparable. The main difference comes from the theoretical arguments considered for the computation of the bounds for the stability degree over a box since our approach only needs the computation of the Routh table of the characteristic polynomial. Given the NP–hardness of the problem, the efficiency of an iteration is especially important in order to delay the explosion of complexity with respect to the number of parameters. In this respect, the gain obtained via our method is even more apparent in the last example.

Example 4.6. Consider the following system, also studied by Balakrishnan et al. [4]

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

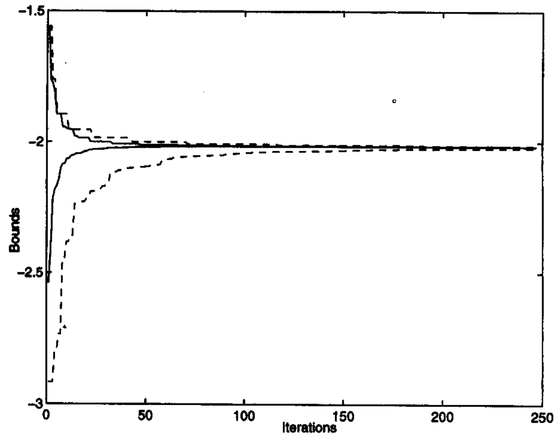

Fig. 8. Lower and upper bounds for $\delta_{\min}$ versus iterations in Example 4.5 (solid line: OPTICRIT, dashed line: branch-and-bound algorithm in [3]).

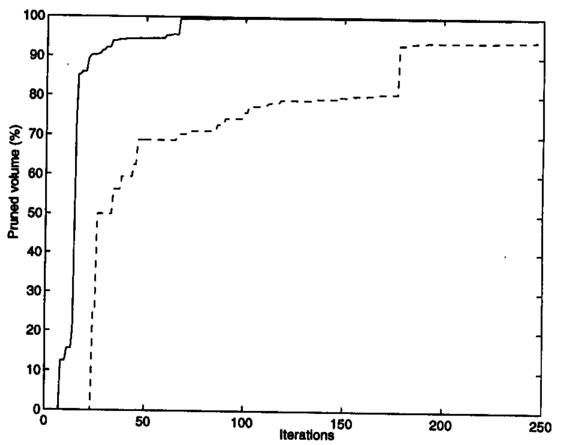

Fig. 9. Pruned volume versus iterations in Example 4.5 (solid line: OPTICRIT, dashed line: branch-and-bound algorithm in [3]).

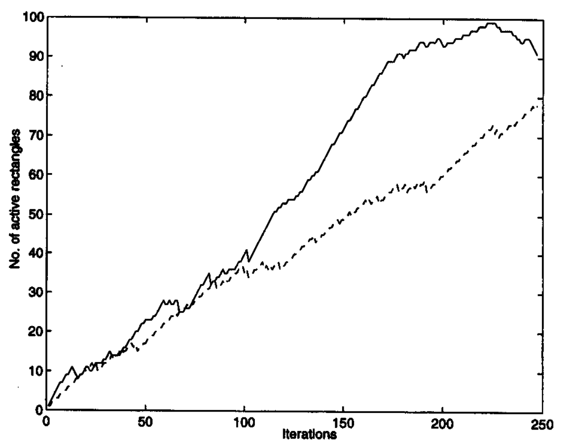

Fig. 10. Number of active boxes versus iterations in Examle 4.5 (solid line: OPTICRIT, dashed line: branch-and-bound algorithm in [3]).

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

$$a(q_1, q_2) = (q_1 + 3.5)^2 + (q_2 + 1)^2 + \frac{1}{0.9}$$
$$b(q_1, q_2) = q_1^4 + q_2^4 + 1$$

It can easily be shown that the minimum stability degree is -1, achieved at $q_1 = 0$, $q_2 = 0$. OPTICRIT has been applied with the tolerance parameters ϵ_δ and ϵ_p equal to 0.01. After 20 iterations, the following bracketing is obtained

$$-1.0048866272 \leq \delta_{\min} \leq -0.9999999781$$

and more than 99% of the initial volume is pruned after 1128 iterations. Similar values are achieved after more than 10000 iterations in [4].

5. Conclusions

This paper has presented two splitting-domain algorithms using interval analysis, respectively to analyse and optimise a scalar criterion that depends non-linearly on a parameter vector. Even trigonometric or exponential parametrisation can be handled. The results are guaranteed, i.e., take the numerical errors introduced by the computer into account, and provided with an estimation of their degree of approximation. As an example of application, these algorithms have been used on the stability degree of linear time-invariant parametric systems. It is thus possible to characterise isodegrees and to compute a set of boxes guaranteed to contain all values of the parameters that maximise (or minimise) the stability degree, as well as an interval guaranteed to contain its extremal value. The computation of an inclusion function for the stability degree was made possible by a simple algorithm. Several examples have illustrated the efficiency of the method, compared to those of the reference papers.

All the examples treated have two parameters in order to allow a vizualisation of the results. Although our algorithms can obviously be used for any number of parameters, their efficiency will drastically decrease with more than, say, six parameters. Even if this limitation is known as the main drawback of any branch-and-bound type of algorithm, it must be noticed that robust stability problems, such as optimum stability degree computation, are known to be NP–hard so that polynomial complexity cannot be achieved. Characteristic polynomials of relatively high order can however be considered, provided that they are (possibly non-linear) functions of a few uncertain physical parameters. In practice, the large number of parameters in interval polynomials or matrices often comes from the fact that physical parameters have been discarded, to get rid of non-linearities that could not be handled.

ISOCRIT and OPTICRIT can be used on many problems other than the stability degree of continuous-time systems. In particular, the stability degree of discrete-time systems can be treated, as well as other measures of robust performance such as stability margins. Moreover, it would be easy to take inequality constraints on the system performances or parameters into account. Finally, the method extends without any modification to the characterisation of isocriteria or optimisation of parameters in the context of estimation.

An interesting extension of the methodology presented here would be to consider problems involving two families of uncertain parameters, one associated with the process and the other with its controller. One could then look for the values of the parameters of the controller that maximise the stability degree for the worst possible value of the process parameters. Replacing process parameters by intervals in ISOCRIT and OPTICRIT would be possible (and easy in ADA) but not very efficient, because inclusion functions would then be evaluated at a box of the process parameter space that could not be split to improve the precision. One should rather combine two splitting algorithms applied to the controller and process parameters respectively.

Acknowledgements

The authors wish to thank Michel Petitot for very helpful discussions concerning the ADA implementation of interval computations.

References

1. Ackermann J, Hu HZ, Kaesbauer D. Robustness analysis: a case study. IEEE Trans on Automat Contr 1990; AC-35:352–356
2. Ackermann J. Uncertainty structures and robust stability analysis. Proc ECC 91 Europ Contr Conf Grenoble, France. 1991. 2318–2327
3. Balakrishnan V, Boyd S, Balemi S. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. Int J Robust Nonlinear Contr 1991; 1:295–317
4. Balakrishnan V, Boyd S, Balemi S. Computing the minimum stability degree of parameter-dependent linear systems. In: Bhattacharyya SP, Keel LH (eds). Control of uncertain dynamic systems. CRC Press, Boca Raton. 1991. 359–378
5. Barmish BR. New tools for robustness analysis. Proc. 27th IEEE Conf Decision Contr Austin, Texas. 1988. 1–6.
6. Barmish BR, Kang HJ. A survey of extreme point resuls for robustness of control systems. Automatica 1993; 29:13–35
7. Chapellat H, Keel LH, Bhattacharyya SP. Extremal robustness properties of multilinear interval systems. Automatica 1994; 30: 1037–1042
8. De Gaston RRE, Safonov MG. Exact calculation of the multiloop stability margin. IEEE Trans on Automat Contr 1988; AC-33:156–170
9. Faedo S. Un nuovo problema di stabilita per le equazioni algebriche a coefficienti reali. Annali de Scuola Normale Superiore di Pisa 1953; SFM-7:53–63
10. Fioro G, Malan S, Milanese M, Taragna M. Robust performance design of fixed structure controllers for systems with uncertain parameters. Proc 32nd IEEE Conf Decision Contr San Antonio, Texas. 1993. 3029–3031.
11. Hansen ER, Global optimization using interval analysis. Marcel Dekker, New York 1992
12. Jury EI, Robustness of discrete systems: a review. Proc 11th IFAC Congress, Tallin, vol. 5. 1990. 184–189
13. Kaminski RD, Djaferis TE. A novel approach to the analysis and synthesis of controllers for parametrically uncertain systems. IEEE Trans on Automat Contr 1994; AC-39:874–876
14. Kharitonov VL. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. Differential'nye Uravenyia 1978; 14:1483–1485
15. Kiendl H, Michalske A. Robustness analysis of linear control systems with uncertain parameters by the method of convex decomposition. In: Robustness of dynamic systems with parameter uncertainties. Birkhaüser, Basel. 1992. 180–198
16. Klate R, Kulish UW, Wiethoff A, Lawo C, Rauch M. C-XSC: A C++ class library for extended scientific computing. Springer, Heidelberg 1993
17. Kokame H, Mori T. A branch and bound algorithm to check the stability of a polytope of matrices. In: Robustness of dynamic systems with parameter uncertainties. Birkhaüser Basel 1992. 125-137
18. Kolev L. An interval first-order method for robustness anlaysis. Int symp on circuits and systems. Chicago. 1993. 2522–2524
19. Mansour M, Balemi S, Truöl W. Robustness of dynamic systems with parameter uncertainties. Birkaüser Monte Verita 1992
20. Moore RE, Methods and applications of interval analysis. SIAM Philadelphia 1979
21. Nemirovskii A. Several NP-hard problems arising in robust stability analysis. Math Contr Signals Syst 1994; 6:99–105
22. Polis MP, Olbrot AW, Fu M. An overview of recent results on the parametric approach to robust stability. Proc 28th IEEE Conf Decision Contr Tampa, Florida. 1989. 23–29
23. Popov EP. The dynamics of automatic control systems. Pergamon Press Oxford 1962
24. Ratschek H, Rokne J. Computer methods for the range of functions. Ellis Horwood-Halstead Press, New York 1984
25. Ratschek H, Rokne J. New Computer methods for global optimisation. Ellis Horwood-Wiley, New York 1988
26. Sideris A, Peña RSS. Fast computation of the multiloop stability margin for real inter-related uncertain

parameters. IEEE Trans on Automat Contr 1989; AC-34:1272–1276

27. Siljak DD, Parameter space methods for robust control design: a guided tour. IEEE Trans on Automat Contr 1983; AC-34:674–688
28. Sondergeld K-P. A generalization of the Routh–Hurwitz stability criterion and an application to a problem in robust controller design. IEEE Trans on Automat Contr 1983; AC-28:965–970
29. Walter E, Jaulin L. Guaranteed characterisation of stability domains via set inversion. IEEE Trans on Automat Contr 1994; AC-39: 886–889

Notation

Lower case letters (x, δ)	Scalars
Upper case letters (X, Δ)	Scalar intervals
Bold lower case letters $(\mathbf{x})$	Vectors
Bold upper case letters $(\mathbf{X})$	Vector intervals, or boxes