A TOOLBOX FOR COMPUTING THE STABILITY MARGIN OF UNCERTAIN SYSTEMS

Aurelio Piazzi[†], **Antonio Visioli**[*]

[†]Dipartimento di Ingegneria dell'Informazione, University of Parma (Italy), e-mail: aurelio@ce.unipr.it
[*]Dipartimento di Elettronica per l'Automazione, University of Brescia (Italy), email: visioli@ing.unibs.it

Keywords: robust stability, linear systems, interval algorithms, software package.

Abstract

A Matlab toolbox implemented for the computation of the stability margin of uncertain systems affected by structured perturbations is presented. The linear time-invariant systems that can be considered depend on continuously and nonlinearly by uncertain parameters. The proposed algorithm computes the stability margin as the maximal l_∞ domain in the parameter space compatible with stability. At the core of this algorithm there is an interval procedure to check positivity on an annular domain centered on the nominal parameter vector. An example of the use of the toolbox is included.

1 Introduction

The dissemination and the adoption of complex procedures for the analysis and design of control systems is greatly facilitated by their implementation in suitable software routines, which renders them transparent to the user. In this way, a deep understanding of these methodologies is not required in order to use them and therefore, provided that their scope is clear, they can be adopted as powerful tools by a large number of academicians and industrial practitioners.

Based on these considerations, in this paper we present a toolbox, which runs in the Matlab environment, for the determination of the stability margin of a linear system affected by parametric uncertainties.

Consider linear time-invariant systems whose stability is investigated by examining its characteristic polynomial $Q(s;p)$:

$$Q(s;p) = s^n + a_1(p)s^{n-1} + a_2(p)s^{n-2} + \ldots + a_{n-1}(p)s + a_n(p).$$

Here $p := [p_1, p_2, \ldots, p_q]^T \in \mathbb{R}^q$ is the uncertain parameter vector and $a_i(p)$, $i = 1, 2, \ldots, n$ are known nonlinear continuous coefficients depending on the uncertain parameters. Given the nominal parameter vector p^o and a weighted l_∞ norm in the parameter space ($||p||_\infty^w = \max_{i=1,\ldots,q}\{|p_i|/w_i\}$, $w_i > 0$), define the l_∞ domain $\mathcal{B}(\rho)$, centered on p^o, as

$$\mathcal{B}(\rho) := \{p \in \mathbb{R}^q : ||p - p^o||_\infty^w \leq \rho\}.$$

Then the stability margin ρ^* of the uncertain system is defined as $\rho^* := \sup \mathcal{M}$ with $\mathcal{M} := \{\rho \in \mathbb{R} : Q(s;p) \text{ is stable } \forall p \in \mathcal{B}(\rho)\}$.

In practical terms, the knowledge of ρ^* gives the meaningful information on how far the uncertain parameters can deviate from the nominal ones without suffering a loss of system stability. Since the importance of this kind of robust stability analysis, which can be also useful in the control system design context, the problem of ρ^* computation has been addressed by many researchers (see e.g. [1, 2, 3, 4]). However, in order to allow any nonlinear dependence on the uncertain parameters, which can include polynomial as well as transcendental ones, these algorithms are not sufficient. On the contrary, the use of interval arithmetic has been proved to be useful in this context [5].

In this paper we implement an interval arithmetic based algorithm which computes arbitrarily good lower and upper bounds of ρ^* and whose global convergence is assured under mild technical restrictions. At the core of this algorithm there is an ad hoc interval procedure to check the positivity of $a_n(p)$ and/or $H_{n-1}(p)$ (the Hurwitz determinant of order $n-1$) on an l_∞ annular domain centered on the nominal parameter vector p^o.

2 Theory

Denote with $\mathcal{D}$ any compact and convex set of $\mathbb{R}^q$ containing p^o. A fundamental result which is instrumental to our development is the following.

Theorem 1 *(Frazer and Duncan, 1929) $Q(s;p)$ is (Hurwitz) stable for all $p \in \mathcal{D}$ if and only if:*

a) $Q(s;p^o)$ is (Hurwitz) stable;

b) $a_n(p) > 0 \; \forall p \in \mathcal{D}$;

c) $H_{n-1}(p) > 0 \; \forall p \in \mathcal{D}$.

This theorem was first presented, in slightly different terms, by Frazer and Duncan [7] who derived it by means of a resultant-based variable elimination starting from boundary crossing conditions (see [8]).

This can be directly accomplished through the formula of Orlando [9, 10]. Indeed, having denoted with $z_i(p)$, $i = 1, \ldots, n$ the roots of $Q(s;p)$, we have that $a_n(p) = (-1)^n \prod_{i=1}^n z_i(p)$ and $H_{n-1}(p) = (-1)^{\frac{n(n-1)}{2}} \prod_{i<k}^{1,\ldots,n} (z_i(p) + z_k(p))$, the latter being Orlando's formula. Hence, the necessity part of Theorem 1 stems from stability definition, i.e. $\mathrm{Re}[z_i(p)] < 0$, $i = 1, \ldots, n$. On the other hand, the sufficiency can be proved by a continuity argument. The role of Orlando's formula in robustness has been shown by many researchers, Vicino *et al.* [4], Ackermann [8], etc.

Stability of the nominal system – $Q(s;p^o)$ is stable – is assumed throughout. As a consequence $\mathcal{M}$ is not empty and it can be rewritten, by virtue of Theorem 1, as $\mathcal{M} = \{\rho : a_n(p) > 0 \; , \; H_{n-1}(p) > 0 \text{ for all } p \in \mathcal{B}(\rho)\}$. This suggests to compute the stability margin by solving the following optimization problem:

$$\rho^* = \sup \rho \tag{1}$$

subject to

$$a_n(p) > 0 \ \forall p \in \mathcal{B}(\rho) \tag{2}$$

$$H_{n-1}(p) > 0 \ \forall p \in \mathcal{B}(\rho) \tag{3}$$

This nonstandard semi-infinite optimization problem will be solved exactly by the positivity-based algorithm presented in the next section.
Define $\mathcal{M}_a := \{\rho \in \mathrm{R} : a_n(p) > 0 \ \forall p \in \mathcal{B}(\rho)\}$ and $\mathcal{M}_h := \{\rho \in \mathrm{R} : H_{n-1}(p) > 0 \, \forall p \in \mathcal{B}(\rho)\}$. Taking into account the continuity property of $a_n(p)$ and $H_{n-1}(p)$ it follows that $\mathcal{M}_a = [0, \rho_a^*)$ and $\mathcal{M}_h = [0, \rho_h^*)$ with $\rho_a^* = \sup \mathcal{M}_a$ and $\rho_h^* = \sup \mathcal{M}_h$.

Property 1 *(Vicino, Tesi and Milanese, 1990)*

$$\rho^* = \min\{\rho_a^*, \rho_h^*\}. \tag{4}$$

Relation (4) was first presented in [4] using a signomial formulation. The formal proof of this somewhat intuitive relation is omitted for brevity [11].

Property 2 *The following two statements hold:*

1. *If for a given real ρ_{al} it is verified that $a_n(p) > 0 \ \forall p \in \mathcal{B}(\rho_{al})$ then $\rho_{al} < \rho_a^*$, i.e. ρ_{al} is a strictly lower bound of ρ_a^*.*
2. *Let be given a point $p_c \in \mathbb{R}^q$ such that $a_n(p_c) \leq 0$. Then, having defined $\rho_{au} := ||p_c - p^o||_\infty^w$, it follows that $\rho_{au} \geq \rho_a^*$, i.e. ρ_{au} is an upper bound of ρ_a^*.*

Proof: Omitted for brevity [11].
Remark 1: A property, which is perfectly analogous to Property 2, holds in relation to set $\mathcal{M}_h$. Indeed Property 2 still holds if we substitute respectively $a_n(p)$, ρ_{al} and ρ_a^* with $H_{n-1}(p)$, ρ_{hl} and ρ_h^*.

Property 3 *Let be given real values ρ_{al}, ρ_{hl}, ρ_{au} and ρ_{hu} satisfying*

$$\rho_{al} < \rho_a^* \ , \ \rho_{hl} < \rho_h^* \tag{5}$$

and

$$\rho_{au} \geq \rho_a^* \ , \ \rho_{hu} \geq \rho_h^* \tag{6}$$

Then it follows that

$$\min\{\rho_{al}, \rho_{hl}\} < \rho^* \quad \text{and} \quad \rho^* \leq \min\{\rho_{au}, \rho_{hu}\} \tag{7}$$

Proof: Omitted for brevity [11].
The above Property 3 specifies how to construct lower and upper bounds of ρ^* from the knowledge of lower and upper bounds of ρ_a^* and ρ_h^*. This result together with Property 2 constitutes the theoretic basis of the positivity-based algorithm following in the next section.

3 The positivity-based algorithm

Nomenclature and Definitions 1
ρ_l: strictly lower bound of ρ^*: $\rho_l < \rho^*$
ρ_u: upper bound of ρ^*: $\rho_u \geq \rho^*$
$\rho_a^* := \sup\{\rho \in \mathbb{R} : a_n(p) > 0 \, \forall p \in \mathcal{B}(\rho)\}$
ρ_{al}: strictly lower bound of ρ_a^*: $\rho_{al} < \rho_a^*$
ρ_{au}: upper bound of ρ_a^*: $\rho_{au} \geq \rho_a^*$
$\rho_h^* := \sup\{\rho \in \mathbb{R} : H_{n-1}(p) > 0 \, \forall p \in \mathcal{B}(\rho)\}$
ρ_{hl}: strictly lower bound of ρ_h^*: $\rho_{hl} < \rho_h^*$
ρ_{hu}: upper bound of ρ_h^*: $\rho_{hu} \geq \rho_h^*$
$\mathcal{A}(\rho_1, \rho_2) := \{p \in \mathbb{R}^q : ||p - p^o||_\infty^w \geq \rho_1 \text{ and } ||p - p^o||_\infty^w \leq \rho_2\}$ with $\rho_2 > \rho_1 \geq 0$ (l_∞ annular domain centered on p^o)
ϵ: given required precision to compute ρ^* ($\epsilon > 0$)
ϵ_p: numerical threshold to be used with the IPTEST procedure ($\epsilon_p > 0$)
ϵ_z: numerical threshold to be used with the ZSEARCH procedure ($\epsilon_z > 0$)
$f(p)$: real continuous function defined over the uncertain parameter space for which $f(p^o) > 0$ (it can be $a_n(p)$ or $H_{n-1}(p)$) $\rho_f^* := \sup\{\rho \in \mathbb{R} : f(p) > 0 \, \forall p \in \mathcal{B}(\rho)\}$
ρ_{fl}: strictly lower bound of ρ_f^*: $\rho_{fl} < \rho_f^*$
ρ_{fu}: upper bound of ρ_f^*: $\rho_{fu} \geq \rho_f^*$
$f^*(\mathcal{D})$: global minimum of function $f(p)$ over any compact domain $\mathcal{D} \subseteq \mathbb{R}^q$
$\mathcal{P}_a := \{p \in \mathbb{R}^q : a_n(p) > 0\}$, positive region of the parameter space relative to $a_n(p)$
$\mathcal{P}_h := \{p \in \mathbb{R}^q : H_{n-1}(p) > 0\}$, positive region of the parameter space relative to $H_{n-1}(p)$
$\mathcal{P}_f := \{p \in \mathbb{R}^q : f(p) > 0\}$, positive region of the parameter space relative to $f(p)$
$\partial\mathcal{P}$:= boundary of (any) set $\mathcal{P} \subseteq \mathbb{R}^q$
$\mathcal{S}(p; r) := \{p' \in \mathrm{R}^q : ||p' - p|| \leq r\}$ ($r > 0$), sphere of radius r and center p ($|| \cdot ||$ denote any norm in $\mathbb{R}^q$).
The positivity-based stability margin computation algorithm herein presented, denoted shortly as algorithm PBSMC, is composed of three parts, denoted as *Phase* I, *Phase* II and *Phase* III. The input-output description of the overall algorithm is the following.

Input of algorithm PBSMC: $a_n(p)$, $H_{n-1}(p)$, thresholds ϵ_p, ϵ_z and precision ϵ.

Output of algorithm PBSMC: ρ_l and ρ_u satisfying $\rho_u - \rho_l \leq \epsilon$.

The variables ϵ_p, ϵ_z and ϵ has to be considered as global variables so that they are available to any phase or algorithm fragment.
The aim of the following Phase I is to determine a positive lower bound of ρ^*, i.e. $\rho_l > 0$.

Phase I

1. $\rho_{al} := 0$, $\rho_{au} := +\infty$.
2. $\rho_{hl} := 0$, $\rho_{hu} := +\infty$.
3. $\rho_a := 1$, $\rho_h := 1$.
4. Apply procedure LBIMPROVEMENT with arguments $a_n(p)$, ρ_{al}, ρ_a to obtain $\rho_{al} > 0$ and possibly a finite ρ_{au}.
5. Apply procedure LBIMPROVEMENT with arguments $H_{n-1}(p)$, ρ_{hl}, ρ_h to obtain $\rho_{hl} > 0$ and possibly a finite ρ_{hu}.
6. $\rho_l := \min\{\rho_{al}, \rho_{hl}\}$, $\rho_u := \min\{\rho_{au}, \rho_{hu}\}$.
7. End.

The role of procedure LBIMPROVEMENT, applied twice at steps 4 and 5, is the improvement of the current lower bounds of ρ_a^* and ρ_h^*. It follows a formal description of this procedure.

Input of LBIMPROVEMENT: $f(p)$, ρ_1 and ρ_2 such that $\rho_1 < \rho_f^*$ and $\rho_2 > \rho_1 \geq 0$.

Output of LBIMPROVEMENT: ρ_{fl} such that $\rho_1 < \rho_{fl} < \rho_f^*$, and if possible a ρ_{fu} such that $\rho_{fu} \geq \rho_f^*$.

Procedure LBIMPROVEMENT

1. flagub := ”false”.
2. Apply procedure IPTEST to $f(p)$ over $\mathcal{A}(\rho_1, \rho_2)$ and obtain ξ_f.
3. In case $\xi_f = -1$ then $\rho_2 := (\rho_1 + \rho_2)/2$, flagub := “true” and go to 2.
4. In case $\xi_f = 0$ then $\epsilon_p := \epsilon_p/2$, $\rho_2 := (\rho_1 + \rho_2)/2$ and go to 2.
5. In case $\xi_f = +1$ then $\rho_{fl} := \rho_2$.
6. If flagub = “true” then apply procedure ZSEARCH to obtain ρ_{fu}.
7. End.

Procedure LBIMPROVEMENT uses, at step 2, the IPTEST procedure to check the positivity of $f(p)$ over the annular domain $\mathcal{A}(\rho_1, \rho_2)$. This positivity test, for which a suitable interval application is proposed in Section 4, has to satisfy this input-output definition.

Input of IPTEST: $f(p)$ and $\mathcal{A}(\rho_1, \rho_2)$.

Output of IPTEST: an integer $\xi_f \in \{-1, 0, +1\}$ satisfying these statements:

a) if $\xi_f = +1$ then it has been proved that $f(p) > 0 \ \forall p \in \mathcal{A}(\rho_1, \rho_2)$ (11a);

b) if $\xi_f = -1$ then it has been found a point $p_c \in \mathcal{A}(\rho_1, \rho_2)$ such that $f(p_c) \leq 0$ (11b);

c) if $\xi_f = 0$ then it has been proved that $|f^*(\mathcal{A}(\rho_1, \rho_2))| < \epsilon_p$. In case $\xi_f = -1$ the procedure output includes the point $p_c \in \mathcal{A}(\rho_1, \rho_2)$ and $f_c := f(p_c)$ (11c).

At step 6 of procedure LBIMPROVEMENT if the logical variable flagub is set to “true” we could define directly ρ_{fu} as $||p_c - p^o||_\infty^w$ since $||p_c - p^o||_\infty^w \geq \rho_f^*$. In order to obtain a better upper bound of ρ_f^*, and in such a way to speed up the convergence of the overall algorithm, it is applied the procedure ZSEARCH which performs a zero search on the segment line connecting p^o with p_c: $p(\alpha) := (1-\alpha)p^o + \alpha p_c$, $\alpha \in [0, 1]$. Indeed considering that $f(p^o) > 0$ and $f(p_c) \leq 0$ and taking into account the continuity of $f(p)$ it is possible to determine, given a small positive threshold ϵ_z, a point of the segment $p(\alpha')$ for which $f(p(\alpha')) \leq 0$ and $f(p(\alpha')) + \epsilon_z > 0$. Therefore, in general, $\rho_{fu} := ||p(\alpha') - p^o||_\infty^w \leq ||p_c - p^o||_\infty^w$. This can be accomplished by means of a simple bisection method. The input-output definition of procedure ZSEARCH is the following.

Input of ZSEARCH: $f(p)$, p_c, f_c with $f_c = f(p_c) \leq 0$.

Output of ZSEARCH: a ρ_{fu} such that $\rho_{fu} \leq ||p_c - p^o||_\infty^w$ and for which there exists $p' \ni \rho_{fu} = ||p' - p^o||_\infty^w$, $f(p') \leq 0$ and $f(p') + \epsilon_z > 0$.

The aim of the following Phase II is to determine a finite upper bound of ρ^*.

Phase II

1. If $\rho_u < +\infty$ then terminate Phase II.
2. $\rho := \max\{\rho_{al}, \rho_{hl}\}$.
3. $\rho := 2\rho$.
4. Apply procedure IPTEST to $a_n(p)$ over $\mathcal{A}(\rho_{al}, \rho)$ and obtain ξ_a.
5. In case $\xi_a = -1$ apply procedure ZSEARCH to determine ρ_{au} and terminate Phase II.
6. In case $\xi_a = 0$ set $\epsilon_p := \epsilon_p/2$ and apply procedure IPTEST to $H_{n-1}(p)$ over $\mathcal{A}(\rho_{hl}, \rho)$ and obtain ξ_h.
   - 6.1. In case $\xi_h = -1$ apply procedure ZSEARCH to determine ρ_{hu} and terminate Phase II.
   - 6.2. In case $\xi_h = 0$ set $\epsilon_p := \epsilon_p/2$ and go to 3.
   - 6.3. In case $\xi_h = +1$ set $\rho_{hl} := \rho$ and go to 3.
7. In case $\xi_a = +1$ set $\rho_{al} := \rho$ and apply procedure IPTEST to $H_{n-1}(p)$ over $\mathcal{A}(\rho_{hl}, \rho)$ and obtain ξ_h.
   - 7.1. In case $\xi_h = -1$ apply procedure ZSEARCH to determine ρ_{hu} and set $\rho_l := \rho_{hl}$, $\rho_u := \rho_{hu}$; then apply procedure FINALP, with arguments $H_{n-1}(p)$, ρ_l, ρ_u, and terminate the algorithm (Phase III has not to be activated).
   - 7.2. In case $\xi_h = 0$ set $\epsilon_p := \epsilon_p/2$ and go to 3.
   - 7.3. In case $\xi_h = +1$ set $\rho_{hl} := \rho$ and go to 3.
8. End.

Remark 2: If Phase I has also determined a finite ρ_{au} or ρ_{hu} then Phase II is not necessary (see step 1 of Phase II).

At step 7 of Phase II if $\xi_h = -1$ this proves that the critical constraint of problem (1) is that relative to $H_{n-1}(p)$. Hence $\rho^* = \rho_h^*$ and it is not necessary to proceed with Phase III. Indeed it suffices to activate the procedure FINALP to improve ρ_l and ρ_u until the required precision is reached.

Input of FINALP: $f(p)$, ρ_{fl} and ρ_{fu}.

Output of FINALP: new values of ρ_{fl} and ρ_{fu} satisfying $(\rho_{fu} - \rho_{fl}) \leq \epsilon$.

Procedure FINALP

1. If $(\rho_{fu} - \rho_{fl}) \leq \epsilon$ then terminate.
2. $\rho := (\rho_{fl} + \rho_{fu})/2$.

3. Apply procedure IPTEST to $f(p)$ over $\mathcal{A}(\rho_{fl}, \rho)$ and obtain ξ_f.
4. In case $\xi_f = -1$ apply procedure ZSEARCH to determine ρ_{fu} and go to 1.
5. In case $\xi_f = 0$ set $\epsilon_p := \epsilon_p/2$ and $\rho := (\rho_{fl} + \rho)/2$; apply procedure LBIMPROVEMENT with arguments $f(p)$, ρ_{fl} and ρ to improve ρ_{fl} and possibly ρ_{fu}. Go to 1.
6. In case $\xi_f = +1$ set $\rho_{fl} := \rho$ and go to 1.
7. End.

It follows the description of the last Phase III which computes ρ_l and ρ_u with required precision.

Phase III

1. $\rho_u := \min\{\rho_{au}, \rho_{hu}\}$.
2. $\rho_l := \min\{\rho_{al}, \rho_{hl}\}$.
3. If $(\rho_u - \rho_l) \leq \epsilon$ then terminate.
4. $\rho := (\rho_l + \rho_u)/2$.
5. If $(\rho_{al} \geq \rho_{hl})$ then go to 11.
6. If $(\rho_{au} \leq \rho_{hl})$ then apply procedure FINALP with arguments $a_n(p)$, ρ_l, ρ_u and terminate.
7. Apply procedure IPTEST to $a_n(p)$ over $\mathcal{A}(\rho_{al}, \rho)$ and obtain ξ_a.
8. In case $\xi_a = -1$ apply procedure ZSEARCH to update ρ_{au} and set $\rho_u := \rho_{au}$.
   8.1. If $(\rho_{au} \leq \rho_{hl})$ then apply procedure FINALP with arguments $a_n(p)$, ρ_l, ρ_u and terminate, else go to 3.
9. In case $\xi_a = 0$ set $\epsilon_p := \epsilon_p/2$ and $\rho := (\rho_l + \rho)/2$; apply also procedure LBIMPROVEMENT with arguments $a_n(p)$, ρ_{al}, ρ to improve ρ_{al} and possibly ρ_{au}.
   9.1. If the application of LBIMPROVEMENT has also improved ρ_{au} then $\rho_u := \rho_{au}$. Go to 2.
10. In case $\xi_a = +1$ set $\rho_{al} := \rho$ and go to 2.
11. If $(\rho_{hu} \leq \rho_{al})$ then apply procedure FINALP with arguments $H_{n-1}(p)$, ρ_l, ρ_u and terminate.
12. Apply procedure IPTEST to $H_{n-1}(p)$ over $\mathcal{A}(\rho_{hl}, \rho)$ and obtain ξ_h.
13. In case $\xi_h = -1$ apply procedure ZSEARCH to update ρ_{hu} and set $\rho_u := \rho_{hu}$.
   13.1. If $(\rho_{hu} \leq \rho_{al})$ then apply procedure FINALP with arguments $H_{n-1}(p)$, ρ_l, ρ_u and terminate, else go to 3.
14. In case $\xi_h = 0$ set $\epsilon_p := \epsilon_p/2$ and $\rho := (\rho_l + \rho)/2$; apply also procedure LBIMPROVEMENT with arguments $H_{n-1}(p)$, ρ_{hl}, ρ to improve ρ_{hl} and possibly ρ_{hu}.
   14.1. If the application of LBIMPROVEMENT has also improved ρ_{hu} then $\rho_u := \rho_{hu}$. Go to 2.
15. In case $\xi_h = +1$ set $\rho_{hl} := \rho$ and go to 2.
16. End.

The technical assumptions adopted for functions $a_n(p)$ and $H_{n-1}(p)$ which are not restrictive from a control engineering viewpoint, are introduced with the following definition.

Definition 1 *The positive region $\mathcal{P}_f$ is said to be not degenerate if one of the following conditions holds:*

i) $\partial\mathcal{P}_f$ is empty.
ii) For every $p \in \partial\mathcal{P}_f$ and for every sphere $\mathcal{S}(p; r)$ there exists a point $\tilde{p} \in \mathcal{S}(p; r)$ such that $f(\tilde{p}) < 0$.

If none of the above conditions is satisfied then $\mathcal{P}_f$ is said to be degenerate.

Remark 3: Note that $\partial\mathcal{P}_f$ is empty if and only if $\mathcal{P}_f = \mathbb{R}^q$. The assumed continuity of function $f(p)$ implies also that for every $p \in \partial\mathcal{P}_f$ then $f(p) = 0$.

Lemma 1 *Procedure LBIMPROVEMENT converges with certainty, satisfying the exposed input-output definition, for any positive values of ϵ_p and ϵ_z.*

Lemma 2 *Assume that $\mathcal{P}_f$ is not degenerate. Then procedure FINALP converges with certainty, satisfying the exposed input-output definition, for any positive values of ϵ_p and ϵ_z.*

Theorem 2 *Assume that $\mathcal{P}_a$ and $\mathcal{P}_h$ are not degenerate and $\rho^* < +\infty$. Then algorithm PBSMC converges with certainty, satisfying the exposed input-output definition, for every positive values of ϵ_p and ϵ_z.*

Proofs of the above lemmas and Theorem 2 are reported in [11].

4 The interval positivity procedure IPTEST

Nomenclature and Definitions 2

- "box" of $\mathbb{R}^q$ a finite multidimensional interval which can be defined as $[p_1^-, p_1^+] \times [p_2^-, p_2^+] \times \ldots \times [p_q^-, p_q^+] \subseteq \mathbb{R}^q$ with $-\infty < p_i^- \leq p_i^+ < +\infty$, $i = 1, 2, \ldots, q$.
- mid($\mathcal{B}$) $:= ((p_1^- + p_1^+)/2, (p_2^- + p_2^+)/2, \ldots, (p_q^- + p_q^+)/2) \in \mathbb{R}^q$, "midpoint" of box $\mathcal{B}$.
- lb($f, \mathcal{B}$) lower bound of the global minimum $f^*(\mathcal{B})$ computed as lower endpoint of an "inclusion function" (associated to $f(p)$) evaluated at $\mathcal{B}$ (see [14]).

The input-output definition of procedure IPTEST satisfying statements (11) poses the problem of devising a suitable algorithmic method which has to deal with the special l_∞ annular domain $\mathcal{A}(\rho_1, \rho_2)$.
A possible, but difficult, choice is that of adapting the methods of deterministic global optimization, for example those described by Horst and Tuy [12], for solving with guaranteed precision a nonconvex minimization problem over an l_∞ annular

domain.

In this paper we propose, as a more viable choice, to adapt — for the special annular domain to be used — the interval positivity test over a box of $\mathbb{R}^q$ presented in [13] (a suitable introduction to interval analysis techniques can be found in Moore's book [6]). Indeed it is always possible to decompose $\mathcal{A}(\rho_1, \rho_2)$ in $2q$ boxes of $\mathcal{R}^q$. Consider for example the bidimensional case $q = 2$. Then $\mathcal{A}(\rho_1, \rho_2) = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3 \cup \mathcal{B}_4$ with

$$\begin{aligned}
\mathcal{B}_1 = &\quad [p_1^o - w_1\rho_2, p_1^o - w_1\rho_1] \times [p_2^o - w_2\rho_1, p_2^o + w_2\rho_1],\\
\mathcal{B}_2 = &\quad [p_1^o + w_1\rho_1, p_1^o + w_1\rho_2] \times [p_2^o - w_2\rho_1, p_2^o + w_2\rho_1],\\
\mathcal{B}_3 = &\quad [p_1^o - w_1\rho_2, p_1^o + w_1\rho_2] \times [p_2^o - w_2\rho_2, p_2^o - w_2\rho_1],\\
\mathcal{B}_4 = &\quad [p_1^o - w_1\rho_2, p_1^o + w_1\rho_2] \times [p_2^o + w_2\rho_1, p_2^o + w_2\rho_2].
\end{aligned} \tag{8}$$

Formulas (8) can be easily generalized for the q-dimensional case.

The interval positivity test to be used with algorithm PBSMC is the following.

Procedure IPTEST

1. $u := +\infty$.
2. Decompose $\mathcal{A}(\rho_1, \rho_2)$ into $2q$ boxes $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{2q}$.
3. For $i = 1, 2, \ldots, 2q$.
   - 3.1. If $f(\text{mid}(\mathcal{B}_i)) \leq 0$ then $p_c := \text{mid}(\mathcal{B}_i)$, $f_c := f(p_c)$, $\xi_f := -1$ and terminate.
   - 3.2. If $\text{lb}(f, \mathcal{B}_i) \leq 0$ then put pair $(\mathcal{B}_i, \text{lb}(f, \mathcal{B}_i))$ into *List* in such a way to preserve the nondecreasing ordering of the lower bounds and set $u := \min\{u, f(\text{mid}(\mathcal{B}_i))\}$.
   - 3.3. End of i-loop.
4. If *List* is empty set $\xi_f := +1$ and terminate.
5. $l :=$ the second member of the first element of *List*.
6. If $(u - \epsilon_p) < 0$ and $(l + \epsilon_p) > 0$ then $\xi_f := 0$ and terminate.
7. Bisect, thus getting boxes $\mathcal{D}_1$ and $\mathcal{D}_2$, the box of the first element of *List* on its maximal dimension.
8. If $f(\text{mid}(\mathcal{D}_1)) \leq 0$ then $p_c := \text{mid}(\mathcal{D}_1)$, $f_c := f(p_c)$, $\xi_f := -1$ and terminate.
9. Repeat for the box $\mathcal{D}_2$ the same action performed (for box $\mathcal{D}_1$) at step 8.
10. Discard the first element from *List*.
11. If $\text{lb}(f, \mathcal{D}_1) \leq 0$ then put pair $(\mathcal{D}_1, \text{lb}(f, \mathcal{D}_1))$ into *List* in such a way to preserve the nondecreasing ordering of the lower bounds and set $u := \min\{u, f(\text{mid}(\mathcal{D}_1))\}$.
12. Repeat for the box $\mathcal{D}_2$ the same action performed (for box $\mathcal{D}_1$) at step 11.
13. Go to 4.
14. End.

Remark 4: In Procedure IPTEST at any stage of iterations *List* = $\{(\mathcal{C}_1, l_1), (\mathcal{C}_2, l_2), \ldots, (\mathcal{C}_h, l_h)\}$ with boxes $\mathcal{C}_i \subseteq \mathcal{A}(\rho_1, \rho_2)$, $i = 1, 2, \ldots, h$ and $l_1 \leq l_2 \leq \ldots \leq l_h \leq 0$ ($l := l_1$).

Remark 5: Note that in case $\rho_1 = 0$ it is not necessary to decompose $\mathcal{A}(\rho_1, \rho_2)$. Indeed $\mathcal{A}(0, \rho_2) = \mathcal{B}(\rho_2)$ which is a box of $\mathbb{R}^q$. This permit to simplify, for the specific case, the instructions at step 3. Observe also that during Phase I the procedure IPTEST is always called with $\rho_1 = 0$.

Proof of convergence of procedure IPTEST according to statements (11) is omitted (see [13] and [15]).

5 The Matlab Toolbox

The proposed algorithm has been implemented in Matlab[1] and it relies on the excellent *b4m* toolbox for Matlab made by Zemke [17], in addition to the standard *Symbolic toolbox* of Matlab. The choice to use Matlab has been done in order to address the overall problem in an integrated environment, and therefore to give the chance to solve the stability margin problem in a broader control system design context, despite the resulting computational time is much higher than that required by using C/C++ language or any other compiled languages in which the code can be optimized.

The toolbox is based on the following main function:

```
function [ro_l,ro_u]=
stabilitymargin(Q,p_0,omega,epsilon,Baumann)
```

where `ro_l` and `ro_u` are the resulting strictly lower bound ρ_l and upper bound ρ_u of ρ^* respectively, `Q` is the symbolic expression of the characteristic polynomial, `p_0` is the array of the nominal values of the parameters, `omega` is the array of the weights associated with the parameters, `epsilon` is the value of the precision parameter ε and `Baumann` is a flag which allows to use the Baumann meanvalue form if it is fixed equal to one (otherwise the standard inclusion function is adopted). Note that the characteristic monic polynomial has to be expressed in symbolic form as an array whose elements are the coefficients of the polynomial in descending order and in which the uncertain parameters are denoted as `p1`, `p2`, etc. (see Section 6). The first coefficient, which has to be equal to one, must be omitted. Finally, note that, to keep the use of the toolbox as simple as possible, it has been fixed $\epsilon_p = 10^{-3}$ and $\epsilon_z = 10^{-3}$ and the values of these parameters cannot be changed by the user.

6 An illustrative example

In this section we describe the application of the function `stabilitymargin` to a robust stability example taken from the literature ([3]). Consider the following characteristic polynomial

$$Q(s;p) = s^4 + p_1^3 p_2 s^3 + p_1^2 p_2^2 p_3 s^2 + p_1 p_2^3 p_3^2 s + p_3^3 \tag{9}$$

[1]The toolbox is available upon request to the authors

where p_1, p_2, and p_3 are the uncertain parameters. The nominal system takes the values $p_1^o = 1.4$, $p_2^o = 1.5$, and $p_3^o = 0.8$; $Q(s; p^o)$ is stable. To perform the robust stability analysis we chose $w_1 = 0.25$, $w_2 = 0.20$, and $w_3 = 0.20$ as the l_∞ norm weights. From (9) it can be derived (note that this computation is not required by the user):

$$\begin{aligned} a_4(p) &= p_3^3 \\ H_3(p) &= \det \begin{bmatrix} a_1(p) & a_3(p) & 0 \\ 1 & a_2(p) & a_4(p) \\ 0 & a_1(p) & a_3(p) \end{bmatrix} = \\ & p_1^6 p_2^6 p_3^3 - p_1^6 p_2^2 p_3^3 - p_1^2 p_2^6 p_3^4. \end{aligned}$$

The function `stabilitymargin` has been adopted by previously running the following commands:

```
syms p1 p2 p3
Q=[p1^3*p2 p1^2*p2^2*p3 p1*p2^3* p3^2
p3^3]
p_0=[1.4 1.5 0.8]; omega=[0.25 0.20 0.20]
epsilon=0.01
Baumann=0
```

The result `ro_l=1.086` and `ro_u=1.093` has been obtained in 73.5 s. Instead, by applying the same function with `Baumann=1`, the result `ro_l=1.089` and `ro_u=1.097` is achieved in 76.1 s.

7 Conclusions

In this paper a Matlab toolbox, in which an algorithm for computing without conservativeness the stability margin ρ^* of linear time-invariant systems depending on uncertain parameters is implemented, has been presented. The algorithm uses the positivity of functions $a_n(p)$ and $H_{n-1}(p)$ over annular domains centered on p^o to develop a branch and bound strategy which permits efficiently to discard from computations, as soon as possible, one of the two functions. The task of testing positivity over the annular domains of the parameter space is committed to an interval procedure. In such a way it is possible to perform a robust stability analysis for systems whose characteristic polynomials depend on uncertain parameters through general nonlinear functions. In practice, the coefficients $a_i(p)$ can be any continuous functions. The presented computational result show that the function `stabilitymargin` implemented is very effective with small problems, despite the computational complexity of the stability margin computation does not permit to claim the same effectiveness when analyzing moderate or large robust stability problems. Thus, it can be a useful tool in a general computer-aided control system design task.

References

[1] R. R. E. De Gaston and M. G. Safonov, "Exact calculation of the multiloop stability margin", *IEEE Transactions on Automatic Control*, Vol. 33, No. 2, pp. 156–171, 1988.

[2] L. A. Zadeh and C. A. Desoer, *Linear System Theory.* New York (U.S.A.): McGraw-Hill, 1963.

[3] A. Sideris and R. S. Sanchez Pena, "Fast computation of the multivariable stability margin for real interrelated uncertain parameters", *IEEE Trans. on Aut. Cont.*, Vol. 34, pp. 1272-1276, 1989.

[4] A. Vicino, A. Tesi and M. Milanese, "Computation of nonconservative stability perturbation bounds for systems with nonlinearly correlated uncertainties," *IEEE Transactions on Automatic Control*, Vol. 35, No. 7, pp. 835–841, 1990.

[5] S. Malan, M. Milanese and M. Milanese, "Robust analysis and design of control systems using interval arithmetic", *Automatica*, Vol. 33, No. 7, pp. 1363–1372, 1997.

[6] R. E. Moore, *Methods and Applications of Interval Analysis*, Philadelphia: SIAM Press, 1979.

[7] R. A. Frazer and W. J. Duncan, "On the criteria for the stability of small motions," *Proc. Royal Society*, Vol. 124, pp. 642–654, 1929.

[8] J. Ackermann, "Uncertainty structures and robust stability analysis," ed. Hermes, *Proceedings of the first European Control Conference*, Vol. 3, pp. 2318–2327, 1991.

[9] L. Orlando, "Sul problema di Hurwitz relativo alle parti reali delle radici di un'equazione algebrica," *Mathematische Annalen*, Vol. 71, pp. 233–245, 1911.

[10] F.R. Gantmacher, *The Theory of Matrices*, New York: Chelsea Publishing Company, 1959.

[11] A. Piazzi, A. Visioli, "A positivity-based algorithm for computing the stability margin of uncertain systems," technical report of *Dipartimento di Ingegneria dell'Informazione*, University of Parma (Italy), 2002.

[12] R. Horst and H. Tuy, *Global Optimization*, Heildelberg (Germany): Springer-Verlag, 1990.

[13] A. Piazzi and G. Marro, "On computing the robust decay rate of uncertain systems", *Proceedings of the IFAC Symp. on Robust Control Design*, Rio de Janerio, pp.46–51, 1994.

[14] H. Ratschek and J. Rokne, *New computer methods for global optimization*, Chichester (GB): Ellis Horwood Limited, 1988.

[15] A. Piazzi and G. Marro, "Robust stability using interval analysis", *International Journal of Systems Science*, Vol. 27, No. 12, pp. 1381–1390, 1996.

[16] E. Baumann, "Globale Optimierung stetig differenzierbarer Funktionen einen Variablen," *Freiburger Intervall-Berichte 86/6*, Institut fur Angewandte Mathematik, University of Freiburg (Germany) 1986.

[17] J. Zemke, "b4m", *Technische Universität Hamburg-Harburg (Germany).*