Stability analysis of incommensurate elementary fractional systems using interval arithmetics

Rachid Malti, Milan R Rapaić, Vukan Turkulov

Stability analysis of incommensurate elementary fractional systems using interval arithmetics

Rachid MALTI
Univ. Bordeaux – IMS UMR 5218 CNRS, France
Email: firstname.lastname@ims-bordeaux.fr

Milan R. RAPAIĆ, Vukan TURKULOV
Faculty of Technical Sciences, University of Novi Sad, Serbia
Email: rapaja@uns.ac.rs, vukan_turkulov@uns.ac.rs

***Abstract*—The objective of this paper is to present a new method for stability analysis of fractional systems with two differentiation orders, based on interval arithmetics. It allows determining stability/instability regions in the parametric space. Hence, all transfer functions which parameters belong to the same stability region have the same stability property.**

I. Introduction

Fractional systems has been attracting a lot of interest during the last two decades in different fields of engineering and science, since the seminal work by Oldham and Spanier [1], [2] for modeling diffusive phenomena.

Due to its simplicity, the most used criterion for testing stability of fractional systems is Matignon's theorem [3]. It allows deciding whether a system is stable by locating its s^ν-poles. It generalizes the classical Routh-Hurwitz criterion for rational systems. However, Matignon's theorem applies only for stability checking of commensurate fractional systems. When the system is incommensurate, some other criteria, mainly based on Cauchy's principal theorem [4] or its derivatives such as the Nyquist theorem [5] are used. However, these methods are quite difficult to implement in practice.

Fractional systems are frequently described by elementary transfer functions of the first kind,

$$G_1(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^\nu + 1}, \tag{1}$$

and/or the second kind,

$$G_2(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^{2\nu} + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}, \tag{2}$$

where $\nu \in [0, 2]$. Hence, it is useful to deduce standard properties of these models, such as stability, resonance, and root locus. Stability conditions, deduced from Matignon's theorem, provided $\omega_0 > 0$, are respectively given by [6]

$$0 < \nu < 2 \tag{3}$$

$$0 < \nu < 2 \quad \text{and} \quad \zeta > -\cos\left(\nu\frac{\pi}{2}\right). \tag{4}$$

These elementary commensurate transfer functions were extended in [7], to a third kind of elementary system:

$$G_3(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^{\nu+1} + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}, \tag{5}$$

which is incommensurate for all $\nu \in \mathbb{R}/Q$.

Further in [8], the following second order fractional transfer function is studied (the highest order is set to 2), due to its interest in modeling viscoelastic and visco-inertial materials

$$G_4(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}. \tag{6}$$

Stability analysis of incommensurate transfer functions such as (5) and (6) is not an easy task, as it requires using criteria difficult to implement, mainly based on Cauchy's argument principle.

The objective of this paper is to study stability of fractional transfer functions described by

$$F(s, \boldsymbol{\alpha}) = \frac{1}{\left(\frac{s}{\omega_0}\right)^{\alpha_2} + 2\zeta\left(\frac{s}{\omega_0}\right)^{\alpha_1} + 1}, \tag{7}$$

with $\omega_0 > 0$. Such transfer functions generalize the previous ones: (1) by setting α_2 to 0, (2) by setting $\alpha_2 = 2\alpha_1$, (5) by setting $\alpha_2 = \alpha_1 + 1$, and (6) by setting $\alpha_2 = 2$.

Stability domain is computed by determining **Stability Crossing Sets (SCS)** between stability and instability regions in the parametric space[1]. Hence, if the system is stable at a given parametric point, it is also stable in the whole region delimited by the SCS. The only parameters influencing stability in (7) are α_1, α_2, and ζ, since $\omega_0 > 0$. Although, the method can be applied to any of these parameters, the following restriction is set in this paper

$$\zeta = 1, \tag{8}$$

for representation purposes only, so that the SCS can be represented in the (α_1, α_2)-plane.

[1]More precisely the SCS allow delimiting regions based on the number of unstable poles. When this number equals zero, one gets the stability region.

The proposed method is based on interval arithmetics, introduced in the next sub-section. Then the problem is formulated in section II and solved in section III. Results of applying the proposed algorithm on (7) are presented in section IV, before concluding.

Introduction to interval arithmetics

Interval analysis was initially introduced by Moore [9]. An interval $[x] = [\underline{x}, \overline{x}]$ is a closed, bounded, and connected set of real numbers. The set of all intervals is denoted by $\mathbb{IR}$. Real operations are extended to intervals as follows. Given $[x] \in \mathbb{IR}$ and $[y] \in \mathbb{IR}$:

$$[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \tag{9}$$

$$[x] - [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \tag{10}$$

$$[x] \times [y] = [\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})] \tag{11}$$

$$[x]/[y] = \begin{cases} [x] \times \left[\frac{1}{\overline{y}}, \frac{1}{\underline{y}}\right], & \text{if } 0 \notin [y] \\ (-\infty, \infty), & \text{if } 0 \in [y]. \end{cases} \tag{12}$$

Interval arithmetics do not define an algebra because $(\mathbb{IR}, +)$ is not a group. Indeed, elements of $\mathbb{IR}$ do not have an inverse. Take for instance $A = [-1, 1] \in \mathbb{IR}$, then $A + (-A) = [-2, 2]$ is not equal to the degenerated interval $[0] = [0, 0] = \{0\}$. Either, $(\mathbb{IR}, +, *)$ is not a ring etc. Additionally, arithmetic operations on intervals introduce often pessimism because the result of each operation must be included in an interval.

II. Problem formulation

Stability of fractional transfer functions described by

$$F(s, \boldsymbol{\alpha}) = \frac{1}{s^{\alpha_2} + 2s^{\alpha_1} + 1}, \tag{13}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) \in \mathcal{A}_1 \times \mathcal{A}_2 \subset \mathbb{R}_+^2$, $\mathcal{A}_1$ and $\mathcal{A}_2$ define the searching domains, is considered in this paper. It can be analyzed by checking the position of the zeros of the characteristic function

$$f(s, \boldsymbol{\alpha}) = s^{\alpha_2} + 2s^{\alpha_1} + 1 \tag{14}$$

$$f(\rho e^{j\theta}, \boldsymbol{\alpha}) = \rho^{\alpha_2} \cos(\theta\alpha_2) + 2\rho^{\alpha_1} \cos(\theta\alpha_1) + 1 + \mathrm{j}\left(\rho^{\alpha_2} \sin(\theta\alpha_2) + 2\rho^{\alpha_1} \sin(\theta\alpha_1)\right) \tag{15}$$

More precisely, the problem can be formulated in two different ways in the parametric space $\boldsymbol{\alpha} \in \mathcal{A}_1 \times \mathcal{A}_2$.

(P1) Finding stability and instability regions. In this case, the objective is to check whether, for positive $\boldsymbol{\alpha} \in \mathcal{A}_1 \times \mathcal{A}_2 \subset \mathbb{R}_+^2$, $f(\rho e^{j\theta}, \boldsymbol{\alpha})$ has zeros in the right half complex plane including the imaginary axis. However, due to the symmetry of complex conjugate zeros, the searching domain can be restreint to the first quadrant of the complex s-plane: $(\rho, \theta) \in \mathbb{R}_+ \times \left[0, \frac{\pi}{2}\right]$.

(P2) Finding the SCS between stability and the instability regions. In this case, the searching domain in the complex plane is restraint to $(\rho, \theta) \in \mathbb{R}_+ \times \{\frac{\pi}{2}\}$. Hence, only values of $\boldsymbol{\alpha} \in \mathcal{A}_1 \times \mathcal{A}_2$, for which the poles are crossing the imaginary axis towards the instability region are searched for.

Hence, the problems (P1) and (P2) can be formulated as finding the set of all feasible parameters

$$\boldsymbol{\theta} = (\rho, \theta, \alpha_1, \alpha_2)^T \in \Omega = (\mathbb{R}_+ \times \Theta \times \mathcal{A}_1 \times \mathcal{A}_2), \tag{16}$$

where $\Theta = \left[0, \frac{\pi}{2}\right]$ for (P1) and $\Theta = \left\{\frac{\pi}{2}\right\}$ for (P2), satisfying

$$\begin{cases} \mathscr{R}e\{f(\boldsymbol{\theta})\} = 0 \\ \quad \text{and} \\ \mathscr{I}m\{f(\boldsymbol{\theta})\} = 0 \end{cases} \tag{17}$$

If $\exists \boldsymbol{\theta} = (\rho, \theta, \alpha_1, \alpha_2)^T \in \Omega$ such that $f(\rho e^{j\theta}, \boldsymbol{\alpha}) = 0$, then, for the problem (P1), the characteristic function has zeros in the closed right half complex plane and, for the problem (P2), on the imaginary axis, which allows determining the SCS.

Both of these problems can be formulated as a Constraint Satisfaction Problem $\mathcal{CSP}$[2]

$$\mathcal{CSP}: \begin{cases} \mathscr{R}e\{f(\boldsymbol{\theta})\} = 0 \\ \mathscr{I}m\{f(\boldsymbol{\theta})\} = 0 \\ 0 < \rho < \mathcal{R}, \quad \theta \in \Theta, \\ \alpha_1 \in \mathcal{A}_1, \ \alpha_2 \in \mathcal{A}_2 \end{cases} \tag{18}$$

where $\mathcal{R}$ is ∞ in theory and is finite in practice for evident implementation reasons. The solution set $\mathbb{S}$ for the problem (18) is rewritten as:

$$\mathbb{S} = \left\{\boldsymbol{\theta} \in \Omega \mid \mathscr{R}e\{f(\boldsymbol{\theta})\} \subset [0] \text{ and } \mathscr{I}m\{f(\boldsymbol{\theta})\} \subset [0]\right\}. \tag{19}$$

The characterization of the whole set $\mathbb{S}$ can be formulated as a set inversion problem:

$$\mathbb{S} = f^{-1}([0]) \cap \Omega, \tag{20}$$

and solved by guaranteed methods.

Hence, the following algorithm can be applied for solving the problem.

Algorithm Stability analysis

1) Find the number of right-half zeros of $f(\rho e^{j\theta}, \boldsymbol{\alpha})$ at a given point, $\boldsymbol{\alpha}_0$. An integer value could be a good choice.

[2]Usually a $\mathcal{CSP}$ is formulated using inequalities

$$\mathcal{CSP}: \begin{cases} \underline{x} \le \mathscr{R}e\{f(\boldsymbol{\theta})\} \le \overline{x} \\ \underline{y} \le \mathscr{I}m\{f(\boldsymbol{\theta})\} \le \overline{y} \\ 0 < \rho < \mathcal{R}, \quad \theta \in \Theta, \\ \alpha_1 \in \mathcal{A}_1, \ \alpha_2 \in \mathcal{A}_2, \end{cases}$$

where $\underline{x}, \overline{x}, \underline{y}, \overline{y}$ can also be set to small enough values $-\epsilon, \epsilon, -\epsilon, \epsilon$; as in the fourth initialization of the example in section IV-D.

2) Initialise $\mathcal{R}$, the maximum radius value of ρ, which defines the searching domain in the right half complex plane for the (P1) problem, and the maximum interval along the imaginary axis for the (P2) problem.
3) Solve the $\mathcal{CSP}$ defined in (18) to find the SCS or the stability/instability region around the $\boldsymbol{\alpha}_0$ point.
4) Check *a posteriori* that the possible location of the poles is strictly inside the searching domain defined by $\mathcal{R}$. Otherwise, if the possible location of poles touches the borders, choose a bigger $\mathcal{R}$ and repeat steps 3 and 4.

III. Solving the $\mathcal{CSP}$

A. Contractors

The $\mathcal{CSP}$ (18) is solved by a contractor $\mathcal{C}$, which is an operator which permits to reduce the domain $[\boldsymbol{\theta}]$ without any bisection. Hence, contracting the box $[\boldsymbol{\theta}]$ means replacing it by a smaller box $[\boldsymbol{\theta}]^*$ such that the solution set $\mathbb{S}$ remains unchanged, i.e. $\mathbb{S} \subset [\boldsymbol{\theta}]^* \subset [\boldsymbol{\theta}]$ [10]. There exists different types of contractors depending on whether the system to be solved is linear or not.

In our study, a non linear type contractor named *forward-backward contractor* is used to reduce the initial searching space. The basic idea when implementing this contractor is to decompose a principal constraint into primitive constraints. Each primitive constraint involves elementary operators and functions such as $\{+, -, \times, /, \exp, \log, \ldots\}$. The next example illustrates how a given constraint is used to contract a domain.

1) Example: Consider the constraint:

$$\begin{cases} f(\mathbf{x}) = x_3 - x_2 x_1 = 0, \\ x_1 \in [2, 10],\ x_2 \in [1, 10],\ x_3 \in [1, 5], \end{cases} \tag{21}$$

which can be rewritten as: $x_3 = x_2 x_1$. The forward interval constraint propagation removes all inconsistent values from $[x_3]$ as follows:

$$[x_3] = ([x_1] \times [x_2]) \cap [x_3] = [2, 5].$$

Then, the backward interval constraint propagation removes all inconsistent values from x_1 and x_2 as follows:

$$[x_1] = ([x_3]/[x_2]) \cap [x_1] = [2, 5],$$

$$[x_2] = ([x_3]/[x_1]) \cap [x_2] = [1, 5/2].$$

After a forward and a backward propagation, the contracted box is $[\mathbf{x}] = \big([2, 5], [1, 5/2], [2, 5]\big)^T$ which contains the solution of the $\mathcal{CSP}$.

```
function [x]=Comb_Contractor_Red(x)
 global nb_siv;
 xx = x;
 rho = x(1); theta = x(2);
 alpha1 = x(3); alpha2 = x(4);
%Forward
 x1 = rho^(2*alpha2);
 x2 = 2*rho^alpha2;
 x3 = theta*alpha2;
 x4 = cos(x3);
 x5 = x2*x4;
 x6 = x1 + x5 + 1;
 x7 = 4*rho^(2*alpha1);
%Backward
 x7    = intersect(x6, x7);
 alpha1= intersect(alpha1,1/2*log(x7/4)/log(rho));
 rho   = intersect(rho, (x7/4)^(1/(2*alpha1)));
 x6    = intersect(x6, x7);
 x5    = intersect(x5, x6 - x1 - 1);
 x2    = intersect(x2, x5/x4);
 alpha2= intersect(alpha2, log(x2/2)/log(rho));
 rho   = intersect(rho, (x2/2)^(1/alpha2));
 x1    = intersect(x1, x6 - x5 - 1);
 rho   = intersect(rho, x1^(1/(2*alpha2)));
 alpha2= intersect(alpha2, log(x1)/(2*log(rho)));

 x = [rho, theta, alpha1, alpha2];
 if any(isnan(x))
     x=xx;
 end
end
```

Fig. 1. The implementation of the combined contractor (24) using the IntLab toolbox [11] under Matlab.

2) Implementing the forward-backward contractor on the system under study: A first contractor could be implemented, after the real part of f:

$$\mathscr{R}e\{f(\rho e^{j\theta}, \boldsymbol{\alpha})\}=0 \Leftrightarrow \rho^{\alpha_2} \cos(\theta\alpha_2)+1=-2\rho^{\alpha_1} \cos(\theta\alpha_1) \tag{22}$$

A second one could also be implemented, after the imaginary part of f:

$$\mathscr{I}m\{f(\rho e^{j\theta}, \boldsymbol{\alpha})\}=0 \Leftrightarrow \rho^{\alpha_2} \sin(\theta\alpha_2)=-2\rho^{\alpha_1} \sin(\theta\alpha_1) \tag{23}$$

However, handling sin and cos functions in each contractor is not an easy task because asin and acos functions return angles in their principal determination, i.e. between 0 and π for the acos, and between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ for the asin. In that case, care must be taken to set back the angles to the correct determination. Another alternative, is to combine (22) and (23) to obtain another contractor with less sin and cos functions. Such a contractor, named *combined contractor*, is obtained by squaring both equations and summing them up

$$\rho^{2\alpha_2} + 2\rho^{\alpha_2} \cos(\theta\alpha_2) + 1 = 4\rho^{2\alpha_1}. \tag{24}$$

A single cos function remains in (24) instead of two in the previous two contractors, which is easier to handle. This contractor is implemented in Fig.1, using the IntLab toolbox [11] under Matlab.

In some cases the contractor cannot reduce enough the parameters domain. In such cases, bisection of the variable vector $\boldsymbol{\theta}$ is necessary. The algorithm SIVIA [12], which is described in the following section is based on the association of contractors and splitting.

B. Set Inversion Via Interval Analysis (SIVIA)

This algorithm, proposed by [12], allows to obtain an inner $\underline{\mathbb{S}}$ and an outer $\overline{\mathbb{S}}$ enclosures of the solution set $\mathbb{S}$ (if it exists), such that:

$$\underline{\mathbb{S}} \subseteq \mathbb{S} \subseteq \overline{\mathbb{S}}. \tag{25}$$

SIVIA is a recursive algorithm based on partitioning of the parameter set into three regions: feasible, undeterminate and unfeasible. SIVIA uses an inclusion test $[t] : \mathbb{IR} \to \mathbb{N}$ which is a function allowing to prove if an interval $[\boldsymbol{\theta}]$ is feasible in which case it is added to the set $\underline{\mathbb{S}}$. Any undetermined region is bisected and tested again, unless its size $w([\boldsymbol{\theta}])$ is less than a precision parameter η tuned by the user and which ensures that the algorithm terminates after a finite number of iterations. The outer approximation is then computed as $\overline{\mathbb{S}} = \underline{\mathbb{S}} \cup \Delta\mathbb{S}$ where $\Delta\mathbb{S}$ is the union of all remaining undetermined boxes. Hence, the SIVIA algorithm is presented as follow:

Algorithm SIVIA (in: $[t], [\boldsymbol{\theta}], \eta$; out: $\underline{\mathbb{S}}, \overline{\mathbb{S}}$)

1) Option: Call contractor on $\boldsymbol{\theta}$.
2) If $[t]([\boldsymbol{\theta}]) = [0]$, return;
3) If $[t]([\boldsymbol{\theta}]) = [1]$, then $\underline{\mathbb{S}} := \underline{\mathbb{S}} \cup [\boldsymbol{\theta}]; \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\boldsymbol{\theta}]$, return;
4) If $w([\boldsymbol{\theta}]) \leq \eta, \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\boldsymbol{\theta}]$;
   Else bisect $[\boldsymbol{\theta}]$ into $[\boldsymbol{\theta}_1]$ and $[\boldsymbol{\theta}_2]$;
5) SIVIA (in: $[t], [\boldsymbol{\theta}_1], \eta$; out: $\underline{\mathbb{S}}, \overline{\mathbb{S}}$);
6) SIVIA (in: $[t], [\boldsymbol{\theta}_2], \eta$; out: $\underline{\mathbb{S}}, \overline{\mathbb{S}}$).

The option in line 1 allows either to call the contractor or not at each execution of the SIVIA algorithm which complexity is known to be exponential!

IV. Application to the system under study

The algorithm is applied to the characteristic function (14), using four different initializations. In the first three, the problem (P2) is considered and in the fourth, the problem (P1) is treated.

A. First initialization

The initial searching box and tolerance are respectively set to:

$$\boldsymbol{\theta} = (\rho, \theta, \alpha_1, \alpha_2)^T \in [0, 4] \times \left\{\frac{\pi}{2}\right\} \times [0, 3] \times [0, 4.5] \tag{26}$$

$$\eta = \text{diam}(\boldsymbol{\theta})/2^7 \tag{27}$$

where $\text{diam}(\boldsymbol{\theta})$ defines the length of each element of $(\boldsymbol{\theta})$.

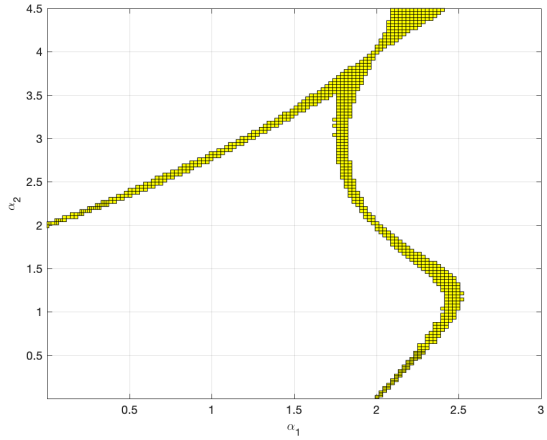

Fig. 2. First initialisation – Stability crossing sets obtained **without contractors**. Zeros of the characteristic function f which arguments equal $\frac{\pi}{2}$ are probably contained in the yellow boundary (outer enclosure $\overline{\mathbb{S}}$). The lower left region delimited by the yellow boundary represents the guaranteed stability region.

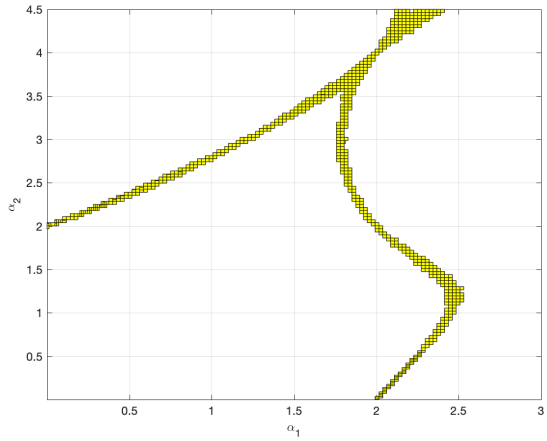

Fig. 3. The same as Fig.2, however **with contractors**.

The SIVIA algorithm is executed:

- without contractors (without step 1 in the algorithm). In this case the SIVIA function is called 13 543 times in 190 sec. The obtained outer enclosure $\overline{\mathbb{S}}$ is plotted in Fig.2.
- with the combined contractor (24) called at each step of the SIVIA algorithm (with step 1 in the algorithm). The SIVIA function is called 8 711 times in 296 sec. The obtained outer enclosure $\overline{\mathbb{S}}$ is plotted in Fig.3.

Moreover, the values at which the poles cross the imaginary axis correspond more or less exactly in both cased to the plot of Fig.4, which validates *a posteriori* that all the poles are inside the searching interval $\rho \in [0, 4]$. In case some poles were touching the limit $\mathcal{R} = 4$, it would have been necessary to choose a bigger $\mathcal{R}$.

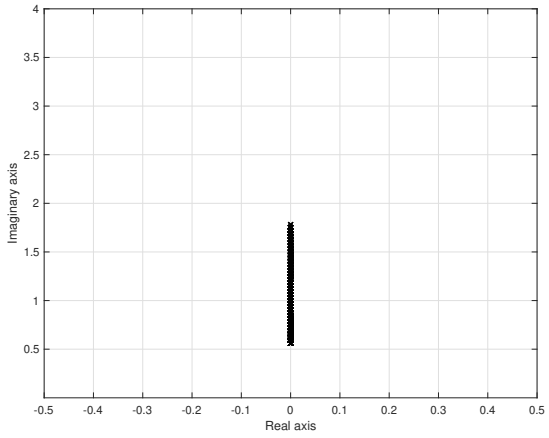

Fig. 4. First initialization – Zeros of the characteristic function f crossing the imaginary axis.

As a conclusion, regarding this first initialization, the algorithm using the combined contractor is a little bit more precise for the same tolerance factor η. Execution speeds of both algorithms are comparable. The former as compared to the latter converges in a bigger number of iterations, however quicker, because the latter calls the contractor at each SIVIA iteration.

B. Second initialization

Let's search for the SCS by enlarging the searching domain. The initial box is now set to:

$$\boldsymbol{\theta} = (\rho, \theta, \alpha_1, \alpha_2)^T \in [0,4] \times \left\{\frac{\pi}{2}\right\} \times [0,15] \times [0,20] \quad (28)$$

The tolerance is defined as in (27), however applied to the new definition of the initial searching box.

The SIVIA algorithm, without contractors, is called 73 037 times in 1 090 seconds. The obtained outer enclosure $\overline{\mathbb{S}}$ of the SCS is plotted in Figs 5, which indicates additionally the number of unstable poles, computed at integer values of the parametric points. Hence, all systems with parameters inside the different regions delimited by the SCS have the indicated number of poles. The intervals of poles look very much like the ones in Fig.4.

C. Third initialization

A major change is operated here. Instead of testing, the $\mathcal{CSP}$ defined in (18), a new $\mathcal{CSPN}$ is defined by enlarging the acceptable mapping of $f(\rho e^{j\theta}, \boldsymbol{\alpha})$ to a square of size ϵ instead of a single point (the origine).

$$\mathcal{CSPN}: \begin{cases} -\epsilon \leq \mathscr{R}e\{f(\rho e^{j\theta}, \boldsymbol{\alpha})\} \leq \epsilon, \\ -\epsilon \leq \mathscr{I}m\{f(\rho e^{j\theta}, \boldsymbol{\alpha})\} \leq \epsilon, \\ 0 < \rho < \infty, \quad \theta \in \Theta, \\ 0 < \alpha_1 < \infty, \ 0 < \alpha_2 < \infty, \\ \epsilon = 0.1 \end{cases} \quad (29)$$

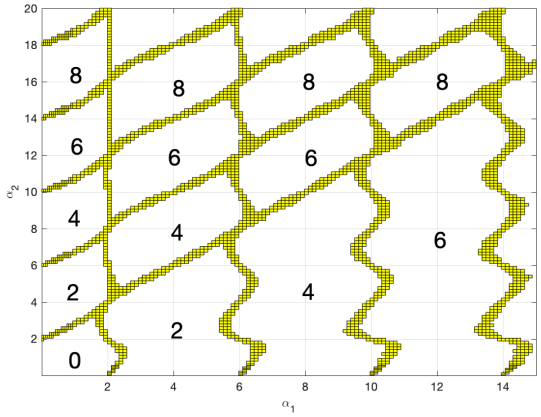

Fig. 5. Second initialization – Stability Crossing Sets (wider intervals as compared to Figs.2 and 3). The number of unstable poles is indicated in each region.

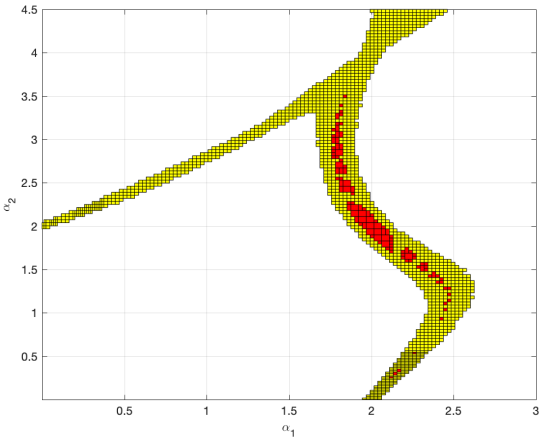

Fig. 6. Third initialization – Inner $\underline{\mathbb{S}}$ (in red), and Outer $\overline{\mathbb{S}}$ (in yellow) enclosures of the $\mathcal{CSPN}$ defined in (29)

Hence, instead of searching for the zeros of the characteristic function f in (14), the algorithm searches for intervals $[\boldsymbol{\theta}]$ that are mapped according to f inside a square of length ϵ. This is the usual way $\mathcal{CSP}$s are formulated. The same parameters and tolerance are chosen as in the first initialization in (26) and (27).

The results, obtained without contractors in 27 795 iterations and 443 sec, are plotted in Fig.6, where red and yellow parts indicate the inner and the outer enclosures $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ of (25).

It turns out not to be interesting to consider the $\mathcal{CSPN}$ (29) instead of the initial $\mathcal{CSP}$ (18), because it widens the feasible solution set as the square defining the admissible mapping gets wider.

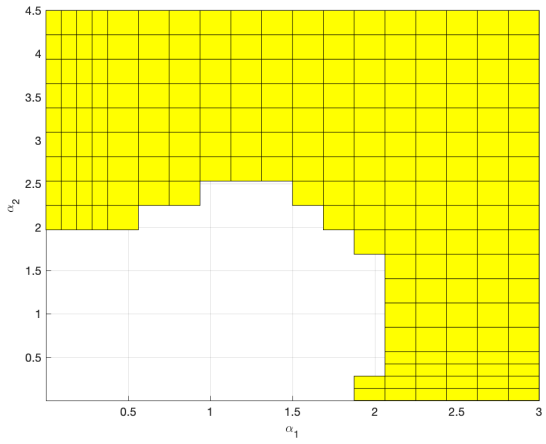

Fig. 7. Fourth initialization – Guaranteed stability in white and possible instability (outer enclosure $\overline{\mathbb{S}}$) in yellow

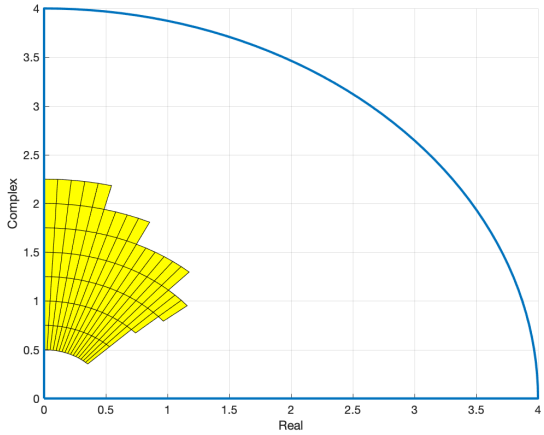

Fig. 8. Fourth initialization – Possible root location in yellow, searching domain boundary in blue

D. Fourth initialization

In this part, the problem (P1) is solved. Hence, instead of looking for the stability crossing sets, let's look for all the zeros of $f(\rho e^{j\theta}, \boldsymbol{\alpha})$ in the first quadrant. Consider the $\mathcal{CSP}$ in (18), and the following searching box:

$$\boldsymbol{\theta} = (\rho, \theta, \alpha_1, \alpha_2)^T \in [0,4] \times \left[0, \frac{\pi}{2}\right] \times [0,3] \times [0,4.5] \quad (30)$$

When setting the tolerance to (27), the algorithm is stopped after an hour because of convergence issues. Then, the tolerance is augmented to:

$$\eta = \text{diam}(\boldsymbol{\theta})/2^4$$

The algorithm converges in 12 525 iterations in 194 sec. The obtained outer enclosure $\overline{\mathbb{S}}$ is plotted in Fig.7.

Apparently, the root-searching-domain in the first quadrant, is validated *a posteriori* in Fig.8: all the poles of the first quadrant are inside the searching domain, defined by $\rho \in [0,4]$, when $(\alpha_1, \alpha_2) \in \times[0,3] \times [0,4.5]$.

Higher precision is definitely required to find out a better sketch of the stability region (in white).

However, this problem appears to be ill-posed as the $\mathcal{CSP}$ (18) evaluated for interval values of $[\boldsymbol{\theta}]$, can never be satisfied. A mapping of $[\boldsymbol{\theta}]$ with f is an interval that can never be a subset of $\{0\}$. Hence, in the instability region, the algorithm will keep bisecting, until reaching the precision η. It turns out that the time complexity of the SIVIA algorithm is higher than a brute-force search on boxes of elementary sizes η.

V. Conclusions

This paper proposes an algorithm, based on interval arithmetics, for stability analysis of an elementary transfer function (7). Guaranteed stability region is determined in the parametric space. Two problems have been formulated. It turns out that the problem of finding the parametric region for which the system is unstable is ill-posed because the bisection algorithm has a time-complexity worse than a brute-force search. However, the problem of finding stability crossing sets turns out to be very interesting, as it allows finding with a reasonable complexity, the stability crossing sets and hence deducing the whole stability region.

References

[1] K. Oldham and J. Spanier, "The replacement of Fick's laws by a formulation involving semi-differentiation," *Electro-anal. Chem. Interfacial Electrochem*, vol. 26, pp. 331–341, 1970.

[2] ——, *The fractionnal calculus - Theory and Applications of Differentiation and Integration to Arbitrary Order*. Academic Press, New-York and London, 1974.

[3] D. Matignon, "Stability properties for generalized fractional differential systems," *ESAIM proceedings - Systèmes Différentiels Fractionnaires - Modèles, Méthodes et Applications*, vol. 5, 1998.

[4] C. Hwang and Y.-C. Cheng, "A numerical algorithm for stability testing of fractional delay systems," *Automatica*, vol. 42, no. 5, pp. 825 – 831, 2006.

[5] J. Trigeassou, A. Benchellal, N. Maamri, and T. Poinot, "A frequency approach to the stability of fractional differential equations," *Transactions on Systems, Signals & Devices (TSSD)*, vol. 4, no. 1, pp. 1 – 25, 2009.

[6] R. Malti, X. Moreau, F. Khemane, and A. Oustaloup, "Stability and resonance conditions of elementary fractional transfer functions," *Automatica*, vol. 47, no. 11, pp. 2462–2467, 2011.

[7] A. Ben Hmed, M. Amairi, and M. Aoun, "Stability and resonance conditions of the non-commensurate elementary fractional transfer functions of the second kind," *Communications in Nonlinear Science and Numerical Simulation*, vol. 22, pp. 842–865, 2015.

[8] E. Ivanova, X. Moreau, and R. Malti, "Stability and resonance conditions of second-order fractional systems," *Journal of Vibration and Control*, pp. 1–15, 2016. [Online]. Available: 10.1177/1077546316654790

[9] R. Moore, *Interval analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[10] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis*. London: Springer-Verlag, 2001.

[11] S. Rump, *Developments in reliable computing*. Kluwer academic publisher, 1999, ch. IntLab – Interval laboratory, pp. 77–104.

[12] L. Jaulin and E. Walter, "Set inversion via interval analysis for nonlinear bounded-error estimation," *Automatica*, vol. 29, no. 4, pp. 1053–1064, 1993.