Applied Interval Computation: A New Approach for Time-Delays Systems Analysis

Michael Di Loreto[1], Massa Dao[2], Luc Jaulin[3], Jean-François Lafay[1], and Jean Jacques Loiseau[1]

[1] IRCCyN, Institut de Recherche en Communications et Cybernétique de Nantes,
UMR CNRS 6597, 44321 Nantes Cedex 3, France
Michael.Di-Loreto@irccyn.ec-nantes.fr,
Jean-Francois.Lafay@irccyn.ec-nantes.fr,
Jean-Jacques.Loiseau@irccyn.ec-nantes.fr

[2] LISA, CNRS-FRE-2656, Université d'Angers, 62 Avenue Notre Dame du Lac,
49000 Angers, France
dao@istia.univ-angers.fr

[3] Laboratoire E3I2, ENSIETA, 2 rue François Verny, 29806 Brest Cedex 9, France
jaulinlu@ensieta.fr

Summary. This paper deals with interval analysis applied to linear time-delays systems. With basic examples, we describe some applications to solve various control problems, and to show that interval computation is an effective tool for time-delays systems analysis.

Keywords: Interval analysis, set inversion, constraint propagation, subpaving, neutral and retarded time-delays systems, quasipolynomial, robust stability, stabilization, frequency domain analysis, disturbance attenuation, model tracking.

1 Introduction

Time-delays systems are dead-time or aftereffect systems, hereditary systems, or systems governed by differential-difference equations, and are described by functional differential equations [2], [10], [11], [17], [26].

The analysis of time-delays systems has attracted much interest in the literature over this half century, especially in the last decade. A recurring subject of research is the stability or robust stability, and has undergone a notable development both conceptually and computationally (see *e.g.* [4], [9], [14], [15], [23], [26], [29], and references therein). Using different theoretical approaches, numerical methods and algorithms obtained are generally semi-analytic, with sometimes difficulties of implementation.

Another recurring subject of research is around optimal control, in particular H_∞ control, with a conceptual tools development adapted to time-delays systems and an extension of existing results for linear systems [8], [16], [19], [24].

Interval analysis has been a very active field in scientific computation for the last 20 years [7], [13], [20], and [25]. Interval computation leads naturally to

J. Chiasson and J.J. Loiseau (Eds.): Appl. of Time Delay Systems, LNCIS 352, pp. 175–197, 2007.
springerlink.com

numerous applications in varied fields, as applied and numerical mathematics, data processing, control systems, robotics or estimation theory [13], [21], [31].

A fundamental advantage of interval analysis is that it gives guaranteed results to a well posed problem. A small number of key concepts are at the core of interval computation and its implementation.

Briefly, consider a box $[\mathbf{x}]$ of $\mathbb{R}^n$, $n \in \mathbb{N}$, a function f from $\mathbb{R}^n$ to $\mathbb{R}$, and a subset $\mathbb{S}$ of $\mathbb{R}^n$ defined by a series of constraints. Three fundamental operations can be implemented by interval analysis. The first one is the notion of *inclusion function*, *i.e.* computing an interval that contains the image of $[\mathbf{x}]$ by f. The second operation is the *inclusion test*, *i.e.* testing when $[\mathbf{x}]$ belongs to $\mathbb{S}$, or more precisely whether $[\mathbf{x}] \subset \mathbb{S}$ or whether $[\mathbf{x}] \cap \mathbb{S} = \emptyset$. The third notion introduced is the *contraction*, *i.e.* the substitution of $[\mathbf{x}]$ by a smaller box $[\mathbf{z}] \subset [\mathbf{x}]$ such that $[\mathbf{z}] \cap \mathbb{S} = [\mathbf{x}] \cap \mathbb{S}$. If $\mathbb{S}$ defines the feasibility set for the solution of some problem, and if $[\mathbf{z}]$ turns out to be empty, then $[\mathbf{x}]$ can be eliminated from the list of boxes that may contain this solution. When no conclusion can be reached about a given box, we can do a *bisection* to obtain subboxes, and each of them can also be studied in turn. These key concepts allow to solve complex problems, with guaranteed and global solutions. All these concepts were inserted in the solver Proj2D[1]. We will see in section 3 that interval computation constitute a whole of adequate tools to analyze some fundamental properties of time-delays systems.

This paper is organized as follows. Section 2 is devoted to interval analysis. In section 3, we apply interval computation to solve some control problems for time-delays systems, like robust stability, robust stabilization, disturbance attenuation or approximtive model tracking. Illustrative examples are done troughout the paper.

2 Interval Computation

In this section, we carry out a short recall on interval computation. We start by presenting some basic concepts and definitions; After that, we analyze the contraction operation and the constraint propagation, for finally describing the set inversion algorithm, which is commonly used in control problems.

2.1 Preliminaries

Denote $\mathbb{R}$ the field of real numbers.

Definition 2.1. *[20] A real interval* $[\mathbf{x}_0]$ *is a connected subset of* $\mathbb{R}$. *The lower (upper) bound of an interval* $[\mathbf{x}_0]$ *is denoted by* $\underline{\mathbf{x}}_0$ ($\overline{\mathbf{x}}_0$ *respectively*).

The width of any non-empty interval $[\mathbf{x}_0]$ *is* $w([\mathbf{x}_0]) \doteq \overline{\mathbf{x}}_0 - \underline{\mathbf{x}}_0$.

The classical set-theoretic operations (union, intersection, cartesian product, ...) can be applied to intervals [20]. In the same manner, the four classical operations

[1] Available at http://www.istia.univ-angers.fr/∼dao/Proj2DV3.zip

of real arithmetic, namely addition (+), substraction (−), multiplication (∗) and division (÷) can be extended to intervals. For any such binary operator, denoted by ($\diamond$), performing the operation associated with $\diamond$ on the intervals $[\mathbf{x}_0]$ and $[\mathbf{y}_0]$ means computing

$$[\mathbf{x}_0] \diamond [\mathbf{y}_0] = [\{x \diamond y \in \mathbb{R} \,|\, x \in [\mathbf{x}_0],\, y \in [\mathbf{y}_0]\}]\,, \tag{1}$$

where $[\mathbb{A}]$ is the smallest interval that contains the set $\mathbb{A}$. For example,

$$\begin{array}{l} [\mathbf{x}_0] + [\mathbf{y}_0] = [\underline{\mathbf{x}}_0 + \underline{\mathbf{y}}_0, \overline{\mathbf{x}}_0 + \overline{\mathbf{y}}_0] \\ [\mathbf{x}_0] - [\mathbf{y}_0] = [\underline{\mathbf{x}}_0 - \overline{\mathbf{y}}_0, \overline{\mathbf{x}}_0 - \underline{\mathbf{y}}_0] \end{array}.$$

Elementary functions such as exp, log, tan, sin, cos, ... can be defined for interval computation. If f_0 is a function from $\mathbb{R}$ to $\mathbb{R}$, then its interval counterpart $[\mathbf{f}_0]$ is defined by

$$[\mathbf{f}_0]([\mathbf{x}_0]) \doteq [\{f_0(x) \,|\, x \in [\mathbf{x}_0]\}]. \tag{2}$$

These basic notions can be extended to the multivariable case [13], [20], [22].

Definition 2.2. *A real interval vector (or box)* $[\mathbf{x}]$ *is a subset of* $\mathbb{R}^n$ *which is defined by the Cartesian product of n closed intervals. It will be written as*

$$[\mathbf{x}] = [\mathbf{x}_1] \times \ldots \times [\mathbf{x}_n],\ \textit{with}\ [\mathbf{x}_i] = [\underline{\mathbf{x}}_i, \overline{\mathbf{x}}_i],\ \textit{for}\ i = 1, \ldots, n. \tag{3}$$

Its ith *interval component* $[\mathbf{x}_i]$ *is the projection of* $[\mathbf{x}]$ *onto the* ith *axis.*

The lower bound $\underline{\mathbf{x}}$ *of a box* $[\mathbf{x}]$ *is the punctual vector consisting of the lower bounds of its interval components* $\underline{\mathbf{x}} \doteq (\underline{\mathbf{x}}_1\ \ldots\ \underline{\mathbf{x}}_n)^T$*. Similarly, the upper bound* $\overline{\mathbf{x}}$ *of a box* $[\mathbf{x}]$ *is the punctual vector* $\overline{\mathbf{x}} \doteq (\overline{\mathbf{x}}_1\ \ldots\ \overline{\mathbf{x}}_n)^T$*.*

The width of the box $[\mathbf{x}] = ([\mathbf{x}_1]\ \ldots\ [\mathbf{x}_n])^T$ *is* $w([\mathbf{x}]) \doteq \max\limits_{1 \le i \le n} w([\mathbf{x}_i])$*.*

The set of all n-dimensional boxes will be denoted by $\mathbb{IR}^n$. The concept of inclusion function is fundamental for interval arithmetic.

Definition 2.3. *[20] Consider a function* $f : \mathbb{R}^n \to \mathbb{R}^m$*. The interval function* $[\mathbf{f}]$ *from* $\mathbb{IR}^n$ *to* $\mathbb{IR}^m$ *is an inclusion function for* f *if*

$$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ f([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]). \tag{4}$$

One of the purposes of interval computation is to provide, for a large class of functions f, inclusion functions that can be evaluated reasonably quickly and such that $[\mathbf{f}]([\mathbf{x}])$ is not too large.

Property 2.4. *[20] An inclusion function* $[\mathbf{f}]$ *for* f *is* thin *if, for any punctual interval vector* $[\mathbf{x}] = x$, $[\mathbf{f}](x) = f(x)$*.*

The inclusion function $[\mathbf{f}]$ *is* minimal *if for any* $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ *is the smallest box that contains* $f([\mathbf{x}])$*. The minimal inclusion function for* f *is unique.*

To build an inclusion function for a function $f : \mathbb{R}^n \to \mathbb{R}$, we can apply the following theorem.

Theorem 2.5. *[20], [22] Consider a function*

$$f \,:\, \begin{cases} \mathbb{R}^n \to \mathbb{R} \\ (x_1, \ldots, x_n) \mapsto f(x_1, \ldots, x_n) \end{cases} \tag{5}$$

A thin inclusion function $[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}$ *for* f *is obtained by replacing each real variable* x_i *by an interval variable* $[\mathbf{x}_i]$ *and each operator or elementary function by its interval counterpart. This function is called the natural inclusion function of* f.

However, natural inclusion functions are not minimal in general [13], [22].

Example 2.6. *Consider the real function* $f : \mathbb{R}^2 \to \mathbb{R}$ *defined by*

$$f(x_1, x_2) = \frac{x_2}{x_1 + x_2} + \sin(x_1)\cos(x_1), \text{ with } x_1 \in [-1, 2] \text{ and } x_2 \in [3, 5]. \tag{6}$$

The natural inclusion function $[\mathbf{f}]_1$ *for* f *is obtained by replacing each real variable by an interval variable, and each real operation by its interval counterpart, i.e.*

$$[\mathbf{f}]_1([\mathbf{x}_1], [\mathbf{x}_2]) = \frac{[\mathbf{x}_2]}{[\mathbf{x}_1] + [\mathbf{x}_2]} + \sin([\mathbf{x}_1])\cos([\mathbf{x}_1]).$$

We have $[\mathbf{f}]_1([-1, 2], [3, 5]) = \frac{[3,5]}{[-1,2]+[3,5]} + \sin([-1, 2])\cos([-1, 2]) = [-0.42, 3.5]$. *A second interval extension* $[\mathbf{f}]_2$ *can be obtained rewriting* f *such that the variables appear at least twice:*

$$[\mathbf{f}]_2([\mathbf{x}_1], [\mathbf{x}_2]) = \frac{1}{1 + [\mathbf{x}_1]/[\mathbf{x}_2]} + \frac{\sin(2\,[\mathbf{x}_1])}{2}.$$

We obtain $[\mathbf{f}]_2([-1, 2], [3, 5]) = \frac{1}{1+[-1,2]/[3,5]} + \frac{\sin([-2,4])}{2} = [0.1, 2]$. *Evidently,* $[\mathbf{f}]_1$ *and* $[\mathbf{f}]_2$ *are both interval extensions of* f. *However,* $[\mathbf{f}]_2$ *is more accurate than* $[\mathbf{f}]_1$, *which suffers from the dependency effect. The interval computed by* $[\mathbf{f}]_2$ *is minimal, and thus equal to the image set* $f([-1, 2], [3, 5])$.

As seen, intervals and boxes form an attractive class of wrappers. However, these wrappers are not enough general to describe all types of sets under interest, which are of course not restricted to intervals and boxes, and include for instance unions of disconnected subsets.

The idea is to introduce the notion of subpaving, useful for the generalization and the implementation of set computation [13], [20].

A subpaving of a box $[\mathbf{x}] \subset \mathbb{R}^n$ is an union of non-overlapping subboxes of $[\mathbf{x}]$ with non zero width. Subpavings can also be employed to approximate compact sets in a guaranteed way. Thus, for any full compact set $\mathbb{X}$, it is possible to find two finite subpavings $\underline{\mathbb{X}}$ and $\overline{\mathbb{X}}$ such that $\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}}$. For interval computation, the notion of subpaving plays a fundamental role, as described below with the bisection operation.

Definition 2.7. *[13] Consider the box* $[\mathbf{x}] = [\mathbf{x}_1] \times \ldots \times [\mathbf{x}_n]$, *and take the index* j *of its first component of maximum width, i.e.*

$$j = \min\{i \mid w([\mathbf{x}_i]) = w([\mathbf{x}])\} \tag{7}$$

The bisection of the box $[\mathbf{x}]$ *is the operation which generates two boxes* $L[\mathbf{x}]$ *and* $R[\mathbf{x}]$*, defined by*

$$\begin{cases} L[\mathbf{x}] \doteq [\mathbf{x}_1] \times \ldots \times [\underline{\mathbf{x}}_i, m([\mathbf{x}_i])] \times \ldots \times [\mathbf{x}_n] \\ R[\mathbf{x}] \doteq [\mathbf{x}_1] \times \ldots \times [m([\mathbf{x}_i]), \overline{\mathbf{x}}_i] \times \ldots \times [\mathbf{x}_n] \end{cases}, \tag{8}$$

where $m([\mathbf{x}_i]) = \frac{\overline{\mathbf{x}}_i + \underline{\mathbf{x}}_i}{2}$ *is the midpoint of* $[\mathbf{x}_i]$*.* $L[\mathbf{x}]$ *is the left child of* $[\mathbf{x}]$*, and* $R[\mathbf{x}]$ *is the right child of* $[\mathbf{x}]$*.*

L and R may be viewed as operators from $\mathbb{IR}^n$ to $\mathbb{IR}^n$. The two boxes $L[\mathbf{x}]$ and $R[\mathbf{x}]$ are siblings. A subpaving of $[\mathbf{x}]$ is regular if each of its boxes can be obtained from $[\mathbf{x}]$ by a finite succession of bisections and selections (see [13] and references therein).

2.2 Constraint Propagation

In this section, we present the concepts of constraint propagation and contractors [3], [5], [7], [13].

Consider n_f relations or constraints, with n_x variables $x_i \in \mathbb{R}$, $i = 1, \ldots, n_\mathrm{x}$, of the form

$$f_j(x_1, \ldots, x_{n_\mathrm{x}}) = 0, \; j = 1, \ldots, n_\mathrm{f}. \tag{9}$$

Each variable x_i is known to belong to an interval (or a union of intervals) $[\mathbf{x}_i]$. Define the vector

$$x = (x_1, \ldots, x_{n_\mathrm{x}})^T$$

and the prior domain $[\mathbf{x}]$ for x as $[\mathbf{x}] = [\mathbf{x}_1] \times \ldots \times [\mathbf{x}_{n_\mathrm{x}}]$. Let f be the function whose coordinate functions are the f_js. Equation (9) can be written in the form $f(x) = 0$. This corresponds to a constraint satisfaction problem (CSP) $\mathcal{P}$, which can be formulated as

$$\mathcal{P}: \; (f(x) = 0, \; x \in [\mathbf{x}]). \tag{10}$$

The solution set of $\mathcal{P}$ is $\mathbb{S} = \{x \in [\mathbf{x}] \mid f(x) = 0\}$. Such CSPs may involve equality and inequality constraints. Contracting $\mathcal{P}$ means replacing $[\mathbf{x}]$ by a smaller domain $[\mathbf{x}']$ such that the solution set $\mathbb{S}$ remains unchanged, *i.e.* $\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$. There exists a unique optimal contraction of $\mathcal{P}$, which corresponds to replacing $[\mathbf{x}]$ by the smallest box that contains $\mathbb{S}$. A contractor for $\mathcal{P}$ is any operator that can be used to contract it.

Numerous basic contractors exist. Some of them are interval counterparts of classical point algorithms like Gauss elimination, Gauss-Seidel and Newton algorithms [5], [13], [18]. We describe here only the contractors based on constraint propagation, contractors which are implemented in the solver Proj2D.

These contractors permit to contract the domains of the CSP $\mathcal{P}$ by taking into account any one of the n_f constraints in isolation, say $f_j(x_1, \ldots, x_{n_\mathrm{x}}) = 0$. Assume that each constraint has the form $f_j(x_1, \ldots, x_{n_\mathrm{x}}) = 0$, where f_j can be decomposed into a sequence of operations involving elementary operators and functions like (+, −, ∗, ÷, sin, cos, ...). It is then possible to decompose this constraint into primitive constraints. Roughly speaking, a primitive constraint is a constraint involving a single operator or a single function. A method for contracting $\mathcal{P}$ with respect to a constraint is to contract each of the primitive constraints until the contractors become inefficient. This is the principle of constraint propagation [7], [13].

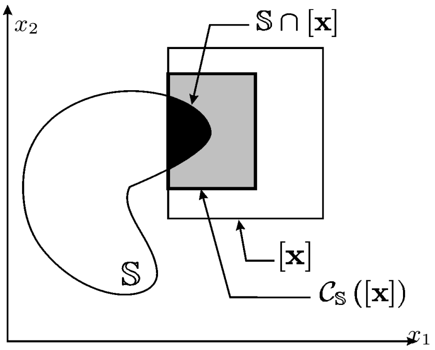
x_2
$\mathbb{S} \cap [\mathbf{x}]$
$\mathbb{S}$
$[\mathbf{x}]$
$\mathcal{C}_{\mathbb{S}}([\mathbf{x}])$
x_1

Fig. 1. Contraction of the box $[\mathbf{x}] = [\mathbf{x}_1] \times [\mathbf{x}_2]$ for the set $\mathbb{S}$, with $x_1 \in [\mathbf{x}_1]$ and $x_2 \in [\mathbf{x}_2]$

Definition 2.8. *[13] Let $\mathbb{S}$ be a set of $\mathbb{R}^n$. The operator $\mathcal{C}_{\mathbb{S}} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for $\mathbb{S}$ if it satisfies*

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}_{\mathbb{S}}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance)}, \\ [\mathbf{x}] \cap \mathbb{S} \subset \mathcal{C}_{\mathbb{S}}([\mathbf{x}]) & \text{(correctness)}. \end{cases} \tag{11}$$

A contractor is minimal if $[\mathbf{x}] \cap \mathbb{S} = \mathcal{C}_{\mathbb{S}}([\mathbf{x}])$.

We give here a useful theorem for a contractor's construction based on the constraint propagation.

Theorem 2.9. *[5], [7] Let $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_f}$ a constraint function. Consider the solution set $\mathbb{S}$ in (10) of vectors x that verify $f(x) = 0$. Suppose there exist some functions g_i, $i = 1, \ldots, n_x$, such that*

$$f(x) = 0 \Longleftrightarrow x_i = g_i({}^i x), \ \forall i \in \{1, \ldots, n_x\}, \tag{12}$$

where ${}^i x = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_x})^T$. Denote $[\mathbf{g}_i]$ an inclusion function for g_i, $i = 1, \ldots, n_x$. A contractor for the set $\mathbb{S}$ is given by

$$\mathcal{C}_{\mathbb{S}}([\mathbf{x}_i]) = [\mathbf{x}_i] \cap [\mathbf{g}_i]([{}^i\mathbf{x}]), \ \forall i \in \{1, \ldots, n_x\}, \tag{13}$$

with $[{}^i\mathbf{x}] = ([\mathbf{x}_1], \ldots, [\mathbf{x}_{i-1}], [\mathbf{x}_{i+1}], \ldots, [\mathbf{x}_{n_x}])^T$. Furthermore, if g_i is continuous and $[\mathbf{g}_i]$ is minimal, then the contractor defined in (13) is minimal.

Example 2.10. *Let $\mathbb{S}$ be the set defined by*

$$\mathbb{S} = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \,|\, x_3 = x_1 + x_2\}, \tag{14}$$

and the box $[\mathbf{x}] = [\mathbf{x}_1] \times [\mathbf{x}_2] \times [\mathbf{x}_3]$*, with* $[\mathbf{x}_1] = [-1, 2]$*,* $[\mathbf{x}_2] = [0, 3]$ *and* $[\mathbf{x}_3] = [4, 8]$*. For* $(x_1, x_2, x_3) \in [\mathbf{x}]$*, we obtain by applying Theorem 2.9:*

$$\begin{array}{l} x_1 \in [\mathbf{x}_1] \cap ([\mathbf{x}_3] - [\mathbf{x}_2]) = [1, 2] \\ x_2 \in [\mathbf{x}_2] \cap ([\mathbf{x}_3] - [\mathbf{x}_1]) = [2, 3]\,. \\ x_3 \in [\mathbf{x}_3] \cap ([\mathbf{x}_1] + [\mathbf{x}_2]) = [4, 5] \end{array} \tag{15}$$

The box obtained after contraction of $[\mathbf{x}]$ *for* $\mathbb{S}$ *is:*

$$\mathcal{C}_{\mathbb{S}}([\mathbf{x}]) = [1, 2] \times [2, 3] \times [4, 5],$$

which is minimal [7].

2.3 Set Inversion Algorithm

In this section, we analyze the set computation implementation, and more particularly the set inversion algorithm which we will use to solve control problems with guaranteed solutions.

The set inversion operation is the computation of the reciprocal image of a regular subpaving. The approximation is realized by a subpaving with a fixed size to guarantee a desired precision. This set inversion is realized in the algorithm Sivia (Set Inverter Via Interval Analysis) we describe now [13], [20].

Consider a continuous function f from $\mathbb{R}^n$ to $\mathbb{R}^m$, $[\mathbf{y}]$ a box of $\mathbb{R}^m$ and $[\mathbf{x}]$ a box of $\mathbb{R}^n$. The set inversion algorithm Sivia allows to approximate with a subpaving the set $\mathbb{S}_{\mathrm{x}}$ described by

$$\mathbb{S}_{\mathrm{x}} = \{x \in [\mathbf{x}] \,|\, f(x) \in [\mathbf{y}]\} = [\mathbf{x}] \cap f^{-1}([\mathbf{y}]). \tag{16}$$

This approximation is realized with an inner and outer subpavings, respectively $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$, such that $\underline{\mathbb{S}} \subset \mathbb{S}_{\mathrm{x}} \subset \overline{\mathbb{S}}$. We give in Table 1 a recursive version of the set inversion algorithm for a set of equations. We suppose to have a contractor $\mathcal{C}_{\mathbb{S}_{\mathrm{x}}}$ for the set $\mathbb{S}_{\mathrm{x}}$, as described in Section 2.2. In the solver Proj2D, the contractor

Table 1. Algorithm Sivia for solving a set of constraints

```
Sivia(in: $[\mathbf{x}]$, $\mathcal{C}_{\mathbb{S}_{\mathrm{x}}}$, $\varepsilon$; inout: $\mathcal{L}$)
1 $[\mathbf{x}] := \mathcal{C}_{\mathbb{S}_{\mathrm{x}}}([\mathbf{x}])$;
2 if ($[\mathbf{x}] = \emptyset$) then return;
3 if ($w([\mathbf{x}]) < \varepsilon$) then
      $\mathcal{L} := \mathcal{L} \cup \{[\mathbf{x}]\}$; return;
4 bisection of $[\mathbf{x}]$ into $L([\mathbf{x}])$ and $R([\mathbf{x}])$;
5 Sivia($L([\mathbf{x}]), \mathcal{C}_{\mathbb{S}_{\mathrm{x}}}, \varepsilon, \mathcal{L}$); Sivia($R([\mathbf{x}]), \mathcal{C}_{\mathbb{S}_{\mathrm{x}}}, \varepsilon, \mathcal{L}$).
```

used in SIVIA is based on the constraint propagation. $\mathcal{L}$ is a boxes list, initialized as an empty list, and ε is a precision parameter.

The union of all boxes in the list $\mathcal{L}$ returned by SIVIA contains the set $\mathbb{S}_{\mathrm{x}}$. The subpaving $\Delta\mathbb{S}$ consisting of all boxes of $\overline{\mathbb{S}}$ that are not in $\underline{\mathbb{S}}$ is called the uncertainty layer. It is a regular subpaving, where all internal boxes have a width smaller than ε.

3 Control Applications

The aim of this section is to introduce the application of interval techniques presented in Section 2 to solve some control problems for time-delay systems.

Interval computation allows, with an another point of view, to solve control problems, with guaranteed solutions. All results presented in Section 3 were obtained with the solver Proj2D, that uses the algorithm SIVIA and the constraint propagation technique. This solver presents solutions of a problem in a graphical form, with a colored subpaving to distinguish boxes characteristics. To solve a problem of the form (16), we obtain three classes of boxes. The first one is a box solution, *i.e.* $\mathbb{X}_{\mathrm{r}} = \{x \in [\mathbf{x}] \,|\, \forall z \in [\mathbf{z}],\ f(x,z) \in [\mathbf{y}]\}$, and its complementary set $\mathbb{X}_{\mathrm{r}}^{\mathrm{c}} = \{x \in [\mathbf{x}] \,|\, \exists z \in [\mathbf{z}],\ f(x,z) \notin [\mathbf{y}]\}$. The second one is a no–solution box, *i.e.* $\mathbb{X}_{\mathrm{b}} = \{x \in [\mathbf{x}] \,|\, \forall z \in [\mathbf{z}],\ f(x,z) \notin [\mathbf{y}]\}$, and its complementary set $\mathbb{X}_{\mathrm{b}}^{\mathrm{c}} = \{x \in [\mathbf{x}] \,|\, \exists z \in [\mathbf{z}],\ f(x,z) \in [\mathbf{y}]\}$. Finally, the last one is the uncertainty layer where all its boxes have the same desired fixed size (see Section 2.3). This characterization is sufficient to solve numerous control problems, as we will describe in the next subsections.

3.1 Frequency-Domain Analysis

We present interval analysis based procedures to construct the well known frequency-domain plots, as Bode, Nyquist or Nichols diagrams. The proposed procedures can be used to construct the plots reliably and with a prescribed accuracy over a finite user specified frequency range.

For transfer functions having a rational form, procedures are available in Matlab or Scilab. However, these procedures have several limitations. In fact, the number of grid points required to obtain a specified accuracy is unknown, as well as the amount of error present for a given frequency response plot, *i.e.* no error estimates are available. These limitations show up particularly severely when the frequency responses exhibit single or multiple sharp peaks or dips, that often happens with time-delays systems.

Interval analysis allows to supply this limitation. Consider a transfer function $H(s)$ including time-delays. We denote by $|H(j\omega)|$ and $\angle H(j\omega)$, the magnitude and phase expressions respectively of $H(s)$ on the imaginary axis, where ω is the frequency variable.

Construct natural interval extensions g and a for $|H(j\omega)|$ and $\angle H(j\omega)$ respectively. The interval frequency range is denoted by Ω.

For a Bode diagram, we consider the set in (16) defined by

$$\mathbb{S}_{\mathrm{x}} = \{(\omega, g) \in \Omega \times [\mathbf{g}] \mid |H(j\omega)| - g = 0\} \tag{17}$$

for the magnitude plot, and

$$\mathbb{S}_{\mathrm{x}} = \{(\omega, a) \in \Omega \times [\mathbf{a}] \mid \angle H(j\omega) - a = 0\} \tag{18}$$

for the phase plot. By the set inversion algorithm (Section 2.3), it is enough to plot $20 \log10(g(\omega))$ for the magnitude, and $a(\omega)$ for the phase. The precision parameter ε in Sivia ensures the control of boxes width that include the exact frequency plot.

Evidently, this method can be applied without difficulty to the Nyquist and Nichols diagrams. Indeed, consider again the transfer function $H(s)$ of a time-delays systems. Decompose this last one in real and imaginary parts, as

$$H(j\omega) = \mathrm{Re}(H(j\omega)) + j\,\mathrm{Im}(H(j\omega)). \tag{19}$$

We note $H_{\mathrm{R}}(\omega) = \mathrm{Re}(H(j\omega))$ and $H_{\mathrm{I}}(\omega) = \mathrm{Im}(H(j\omega))$. Denote by h_{R} and h_{I} the natural interval extensions of $H_{\mathrm{R}}(\omega)$ and $H_{\mathrm{I}}(\omega)$ respectively. We solve with Sivia the problem

$$\mathbb{S}_{\mathrm{x}} = \{(\omega, h_{\mathrm{R}}, h_{\mathrm{I}}) \in \Omega \times [\mathbf{h}_{\mathrm{R}}] \times [\mathbf{h}_{\mathrm{I}}] \mid H_{\mathrm{R}}(\omega) - h_{\mathrm{R}} = 0 \text{ and } H_{\mathrm{I}}(\omega) - h_{\mathrm{I}} = 0\} \tag{20}$$

and we plot the results in the $(h_{\mathrm{R}}, h_{\mathrm{I}})$ plane to obtain the Nyquist diagram. For the Nichols diagram, we solve

$$\mathbb{S}_{\mathrm{x}} = \{(\omega, g, a) \in \Omega \times [\mathbf{g}] \times [\mathbf{a}] \mid |H(j\omega)| - g = 0 \text{ and } \angle H(j\omega) - a = 0\} \tag{21}$$

with the notations of (17) and (18), and the solution is reported in the (a, g) plane.

The main advantage of the plots described here is that the frequency diagram obtained is guaranteed, advantage we don't have with Matlab or Scilab. Furthermore, these plots have a numerical interest, as for example the determination of $\sup\limits_{\omega \in \mathbb{R}} |H(j\omega)|$.

Example 3.1. *Consider the system of transfer function*

$$H(s) = \mathrm{e}^{-s\tau} - 1, \tag{22}$$

with $\tau = 0.1$. *The Magnitude Bode diagram of (22) is reported on Figure 2, thanks to equation (17).*

Example 3.2. *Consider the system of transfer function*

$$H(s) = \frac{1 - \mathrm{e}^{1-s}}{s - 1}, \tag{23}$$

which is analytic for all $s \in \mathbb{C}$ *and corresponds to a distributed delay. The magnitude plot* $|H(j\omega)|$ *when* $\omega \in [-100, 100]$ *is reported on Figure 3.*

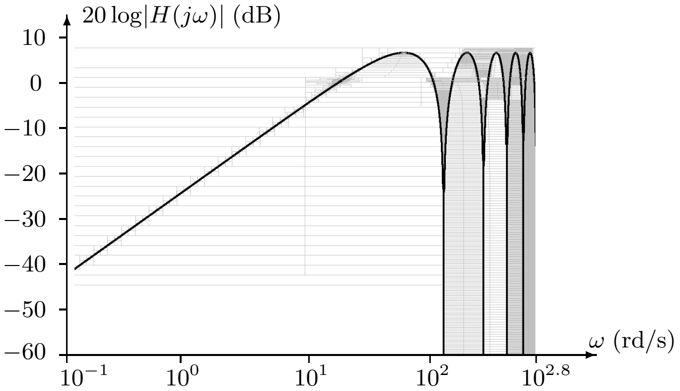

Fig. 2. Magnitude Bode diagram of $H(s)$ in (22)

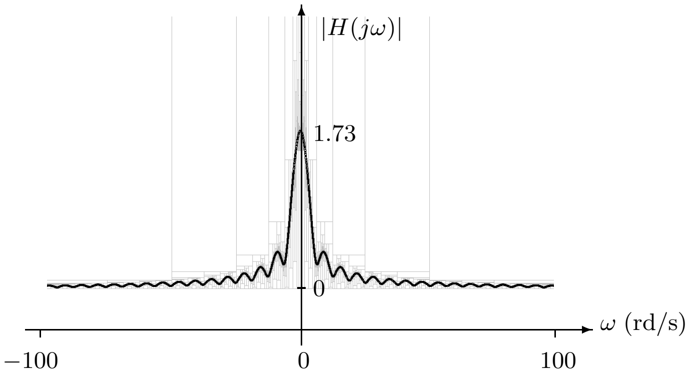

Fig. 3. Magnitude diagram $|H(j\omega)|$ for (23)

3.2 Robust Stability Analysis

The stability of time-delays systems is a problem of recurring interest in the last twenty years, thanks to the possibility to destabilize a system with the existence of a small delay.

In the literature, two classes of stability criteria for linear time-delays systems occur, according to their dependence with respect to the size of delays. The corresponding methods can be cast into two classes: frequency-domain and time-domain based methods. In the first one, we can include the approach based on the small gain theorem, two variables polynomials approach, or a generalized eigenvalues approach. In the second one, we can include the matrix measure approach, the Lyapunov stability approach combined with Lyapunov equations, Riccati equations or linear matrix inequalities, to apply techniques as the Lyapunov-Razumikhin function approach or the Lyapunov-Krasovkii functional approach. For further informations, the reader is referred to [10], [11], [26], and references therein.

A central rule of stability analysis is played by quasipolynomials associated with the characteristic equation of a time-delays systems. We distinguish two general classes of quasipolynomials, associated with retarded or neutral time-delays systems.

A retarded quasipolynomial can be written as

$$f(s) = a_0(s) + \sum_{k=1}^{m} a_k(s)\mathrm{e}^{-\tau_k s}, \tag{24}$$

where $\tau_0 = 0 < \tau_1 < \ldots < \tau_m$, and $a_k(s)$, for $k = 0$ to m, are real polynomials described by

$$\begin{aligned} a_0(s) &= s^n + \textstyle\sum\limits_{i=0}^{n-1} a_{0,i}s^i, \\ a_k(s) &= \textstyle\sum\limits_{i=0}^{n-1} a_{k,i}s^i,\ k = 1, \ldots, m. \end{aligned} \tag{25}$$

The corresponding time-delays systems are given by

$$x^{(n)}(t) + \sum_{i=0}^{n-1}\sum_{k=0}^{m} a_{k,i}x^{(i)}(t - \tau_k) = 0. \tag{26}$$

The quasipolynomial (24) is said to be stable if $f(s) \neq 0$, $\forall s \in \mathbb{C}_+ = \{s \,|\, \mathrm{Re}(s) \geq 0\}$. It is said to be stable independent of delay if this condition holds for all τ_k, $k = 1, \ldots, m$. A neutral time-delays system is governed by a functional differential equation of the form

$$x^{(n)}(t) + \sum_{k=1}^{m} a_{k,n}x^{(n)}(t - \tau_k) + \sum_{i=0}^{n-1}\sum_{k=0}^{m} a_{k,i}x^{(i)}(t - \tau_k) = 0, \tag{27}$$

with its characteristic equation

$$f(s) = s^n \left(1 + \sum_{k=1}^{m} a_{k,n}\mathrm{e}^{-s\tau_k}\right) + \sum_{i=0}^{n-1} a_{0,i}s^i + \sum_{k=1}^{m} a_k(s)\mathrm{e}^{-\tau_k s}, \tag{28}$$

where $a_k(s)$ are given in (25). The system (27) is said to be stable if there exists $\alpha > 0$ such that $f(s) \neq 0$ for all $s \in \mathbb{C}$ with $\mathrm{Re}(s) > -\alpha$. A large number of results is well developed for quasipolynomials analysis, with different levels of difficulty for their implementation. We can cite for instance [4], [9], [12], [28] or [30]. A difficulty issued from these results is for instance to characterize the robust stability of a given system for constant uncertain parameters and delays, which lie in known bounded intervals. Here, interval computation brings some new elements and responses. By interval computation, the localization of quasipolyomials roots in a compact set is reduced to an easy set inversion problem, solvable with the algorithm Sivia.

We shall focus attention on robust stability and robust control problems for uncertain systems that can be described by parametric models, the unknown parameters of which are assumed to lie between known finite bounds. We begin with the problem of roots localization of quasipolynomials.

Problem 3.1. *Consider a retarded or a neutral time-delays system of the form (26) or (27) with $f(s)$ its characteristic equation, and a given box $\mathbb{X}$ of $\mathbb{C}$. We want to solve $f(s) = 0$, for $s \in \mathbb{X}$.*

Writing $s = x + jy$, $(x, y) \in \mathrm{IR}^2$, the set $\mathbb{X}$ is decomposed as a Cartesian product of real intervals $\mathbb{X} = [\mathbf{x}] \times [\mathbf{y}]$, with $x \in [\mathbf{x}]$ and $y \in [\mathbf{y}]$. The Problem 3.1 is equivalent to the set inversion problem

$$\mathbb{S} = \{(x, y) \in [\mathbf{x}] \times [\mathbf{y}] \mid f(x + jy) = 0\} = ([\mathbf{x}] \times [\mathbf{y}]) \cap f^{-1}(0), \tag{29}$$

that can be performed by Sivia, described in Section 2.3. Note that results obtained in (29) are guaranteed, so that we are ensured of the absence or presence of quasipolynomials roots in the box $[\mathbf{x}] \times [\mathbf{y}]$.

A direct application of the Problem 3.1 is the characterization of stability of a retarded quasipolynomial with known and constant parameters. In fact, for retarded time-delays systems, we can compute a positive born $R < \infty$ such that all unstable roots of the characteristic equation lie in the box $[0, R] \times [-R, R]$ [27]. We are also able to calculate all the unstable roots with the solutions of Problem 3.1.

For neutral systems, the conclusion is less obvious. The presence of zeros asymptotic directions of (28) required non-bounded search boxes, and an estimation of a larger born for the module of unstable zeros is not always realizable. However, interval computation allows to give some important and guaranteed indications.

For a robust stability analysis of time-delays systems, we can apply a similar reasoning. Consider a system of characteristic equation (24) or (28), *i.e.* of a general form

$$g(s, q, \tau) = \sum_{i=0}^{n} \sum_{k=0}^{m} q_{ik} s^i \mathrm{e}^{-\tau_k s}, \tag{30}$$

with $q = (q_{ik}) \in \mathrm{IR}^{(n+1)\times(m+1)}$, $\tau = (\tau_0, \ldots, \tau_m)^T$, and $\tau_0 = 0 < \ldots < \tau_m$. The coefficients q_{ik} and delays τ_k are constant but uncertain. They are supposed to lie in closed intervals with known finite bounds:

$$\begin{cases} q_{ik} \in [\underline{\mathbf{q}}_{ik}, \overline{\mathbf{q}}_{ik}] = [\mathbf{q}_{ik}], \text{ for } i = 0, \ldots, n \text{ and } k = 0, \ldots, m, \\ \tau_k \in [\underline{\mathbf{d}}_k, \overline{\mathbf{d}}_k] = [\mathbf{d}_k], \quad \text{ for } k = 0, \ldots, m. \end{cases}$$

with $[\mathbf{d}_k] \subset \mathrm{IR}_+$, for $k = 0, \ldots, m$. Denote

$$\begin{cases} [\mathbf{q}] = \{[\mathbf{q}_{ik}], \text{ for } i = 0, \ldots, n \text{ and } k = 0, \ldots, m\} \\ [\mathbf{d}] = \{[\mathbf{d}_k], \text{ for } k = 0, \ldots, m\} \end{cases}, \tag{31}$$

the vectors of the parameters and the delays uncertainties intervals respectively. The quasipolynomials family

$$\mathcal{G} = \{g(s,q,\tau) \,|\, q \in [\mathbf{q}],\ \tau \in [\mathbf{d}]\},\ s \in \mathbb{C}, \tag{32}$$

is said to be robustly stable if for all $q \in [\mathbf{q}]$ and $\tau \in [\mathbf{d}]$,

$$g(s,q,\tau) \neq 0,\ \forall s \in \mathbb{C}_{+}. \tag{33}$$

It is robustly stable independent of delays if (33) holds for all $\tau \in \mathbb{R}_{+}^{n+1}$.

Problem 3.2. *Consider a time-delays system of characteristic equation of the form (30). We want to characterize robust stability of quasipolynomials family $\mathcal{G}$ in (32), using interval computation and property (33).*

To solve Problem 3.2, we use the set inversion algorithm applied to the set $\mathbb{S}$

$$\mathbb{S} = \{(s,q,\tau) \in [\mathbf{s}] \times [\mathbf{q}] \times [\mathbf{d}] \,|\, g(s,q,\tau) = 0\} = ([\mathbf{s}] \times [\mathbf{q}] \times [\mathbf{d}]) \cap g^{-1}(0), \tag{34}$$

where $[\mathbf{s}]$ is an interval variation of $s \in \mathbb{C}$. In practice, we will decompose in real and imaginary parts $s = x + jy$ to obtain $[\mathbf{s}] = [\mathbf{x}] \times [\mathbf{y}]$, with $[\mathbf{x}]$ and $[\mathbf{y}]$ real intervals, and we can test the absence of solutions in regions of the right half complex plane.

For retarded time-delays systems, the solution obtained for Problem 3.2 is a proof of robust stability, thanks to the existence of a finite larger bound of unstable roots modules of (24).

Problem 3.2 applied to neutral time-delays systems does not allow, without any other assumption, a conclusion on robust stability, but it provides significant indications.

Finally, note that the solution of Problem 3.2 can be projected onto a parametric plane, where only the values of coefficients $q \in [\mathbf{q}]$ and delays $\tau \in [\mathbf{d}]$ are reported. Then, we can analyze parametric regions for which the robust stability is ensured, and those for which we loose this robust property. This kind of plot brings an invaluable help for dynamics analysis.

An another interesting problem is the stabilization or robust stabilization of time-delays systems. Here, interval computation presents two limits. The first one is the restricted number of parameters, to avoid significant computing times. The second one is the necessity to choose a feedback with a predefined structure. The idea is in fact to reduce the problem of (robust) stabilization to a (robust) stability problem, treated with Problems 3.1 and 3.2, with some additional quasipolynomial coefficients to be determined which depend on the feedback structure.

Consider a time-delays system (Σ), with input u and output x. No assumption is made on the delays localization. Denote by $\hat{u}(s)$ and $\hat{x}(s)$ the Laplace transforms of u and x respectively, and by $H(s) = \frac{\hat{x}(s)}{\hat{u}(s)}$ the transfer of (Σ). Finally, denote by $k(s)$ a stabilizing feedback for Σ such that $\hat{u}(s) = k(s)\hat{x}(s)$. Interval computation allows to choose simple predefined structures for $k(s)$, as for example proportional, proportional-integral or proportional-integral-derivative controllers, or generalized feedbacks which take into account delayed state, and eventually delayed state derivatives or integrals, as for example

$$k(s) = \sum_{i=0}^{h} \sum_{l=0}^{r} k_{il} s^{i-p} \mathrm{e}^{-s\tau_l}, \tag{35}$$

with $(p, h, r) \in \mathbb{N}^3$, $k_{il} \in \mathbb{R}$ (with $r \leq m$ and $p \leq n$ for a system (Σ) of the form (30)). In practice, since the number of parameters is restricted, we will consider controllers with a maximum of 2 or 3 coefficients parameters k_{il}. The expression of $k(s)$ in (35) is not enough general; the choice of feedbacks structure is directly related to systems dynamics. The predefined structure of $k(s)$ is then to adapt to the considered problem.

Problem 3.3. *Consider an unstable time-delays system with transfer function $H(s)$, and a feedback $k(s)$ with unknown coefficients. How to ensure stability in closed loop by the choice of coefficients of $k(s)$?*

To answer Problem 3.3, note that in closed loop, the characteristic equation is of the form (30), where coefficients q_{ik} depend on the controllers coefficients k_{il} in (35). Then, in closed loop, the characteristic equation is given by a quasipolynomial of the form

$$g(s, \mathbf{k}) = \sum_{i} \sum_{l} q_{il}(\mathbf{k}) s^i \mathrm{e}^{-s\tau_l} \tag{36}$$

where $\mathbf{k}$ is the coefficients vector of the feedback $k(s)$. We are reduced to solve

$$\mathbb{S} = \{(s, \mathbf{k}) \in [\mathbf{s}] \times [\mathbf{k}] \,|\, g(s, \mathbf{k}) = 0,\ \mathrm{Re}(s) < 0\}, \tag{37}$$

where $[\mathbf{k}]$ is an admissible values interval for $\mathbf{k}$. Applying algorithm Sivia, we obtain the guaranteed results, *i.e.* the values $\mathbf{k} \in [\mathbf{k}]$ of the feedback coefficients such that the stability is guaranteed in closed loop, at least for retarded time-delays systems. For neutral time-delays systems, we can obtain only indications, that we can verify in a second time.

A more complex problem is the robust stabilization by feedback. For this problem, we take notations of Problems 3.1 and 3.2.

Problem 3.4. *Consider a time-delays system, with uncertain and constant parameters, which lie in closed intervals with known bounds. With an appropriate feedback to determine, we want to ensure the robust stability in closed loop.*

In closed loop, the characteristic equation becomes

$$g(s, q, \tau, \mathbf{k}) = \sum_{i} \sum_{l} q_{il}(\mathbf{k}) s^i \mathrm{e}^{-s\tau_l}, \tag{38}$$

where (q, d) are defined in (31), and $\mathbf{k}$ in (36). The Problem 3.4 is reduced to the set inversion problem

$$\mathbb{S} = \{(s, q, \tau, \mathbf{k}) \in [\mathbf{s}] \times [\mathbf{q}] \times [\mathbf{d}] \times [\mathbf{k}] \,|\, g(s, q, \tau, \mathbf{k}) = 0,\ \mathrm{Re}(s) < 0\}, \tag{39}$$

where solutions given by Sivia ensure the closed loop stability of the quasipolynomials family (38), at least for retarded systems.

Example 3.5. *Let the retarded time-delay system [23], [26],*

$$\dot{x}(t) = -ax(t) - bx(t - \tau) \tag{40}$$

with $(a, b, \tau) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+$ *constant uncertain parameters, which lie in* $[-1, 1] \times [2, 3] \times [0, 0.5]$ *respectively. Its characteristic equation is* $s + a + b\mathrm{e}^{-s\tau} = 0$. *We verify with interval methods if this system is robustly stable. We report solutions in the parametric plane* (a, b) *on the Figure 4. The white region ensures robust stability, for all* $\tau \in [0, 0.5]$. *The grey region does not guarantee robust stability, i.e. in each grey box, there exists at least one value of* (a, b, τ) *such that (40) becomes unstable. We find again the well known results on the stability of (40).*

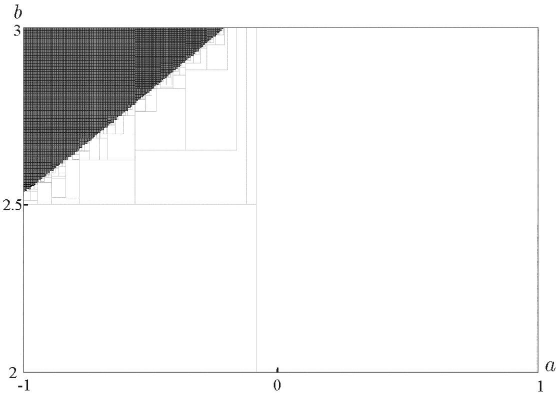

Fig. 4. Robust stable or unstable regions in the parametric plane (a, b) of (40)

Example 3.6. *Consider the system, with an appropriate initialization, described by*

$$x(t) = \frac{3}{4}x(t-1) - \frac{3}{4}x(t-\tau), \tag{41}$$

with its associated characteristic equation $f(s) = 1 - \frac{3}{4}\mathrm{e}^{-s} - \frac{3}{4}\mathrm{e}^{-s\tau} = 0$. *If we take* $\tau = 2$, *the solutions of this equation are stable, since denoting* $\lambda = \mathrm{e}^{s}$, *we have two solutions in* λ *which are* $\lambda_{1,2} = \frac{3}{8} \pm j\frac{\sqrt{39}}{8}$, *and* $|\lambda_{1,2}| < 1$. *Now, taking the delay* τ *in* $[\mathbf{d}] = [2, 3]$, *the system (41) becomes unstable, as shown in Figure 5, where the roots localization of the characteristic equation (41) is reported. For more precisions on this example and the loss of stability, see [11].*

Suppose now that we want to stabilize (41), i.e.

$$x(t) + u(t) = \frac{3}{4}x(t-1) - \frac{3}{4}x(t-\tau), \tag{42}$$

with $u(t)$ *the control variable and* $\tau \in [2, 3]$. *We want to stabilize (42) with a control law of the form*

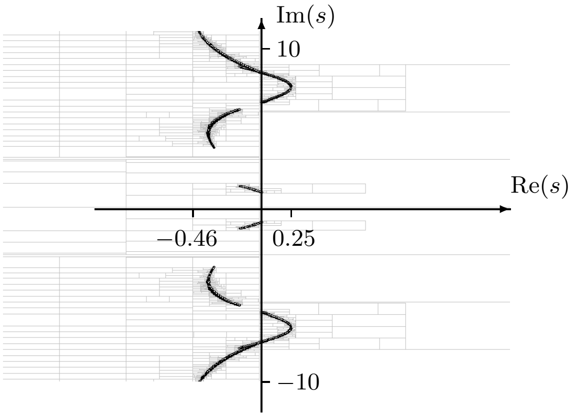

Fig. 5. Roots localization of the characteristic equation of (41), for $\tau \in [2, 3]$

$$u(t) = k_1 x(t) + k_2 x(t-1), \tag{43}$$

with $(k_1, k_2) \in [-5, 5] \times [-3, 3]$ *parameters to be determined (Problem 3.4). Applying the algorithm* Sivia, *we guarantee the absence of roots with positive real part of the closed loop characteristic equation. In the parametric plane* (k_1, k_2)*, we obtain Figure 6. The white zone is a stable zone of (42) and (43), for all* $\tau \in [2, 3]$*. The dark-grey zone is a non robust stable zone, i.e. in each boxes, there exists at least one value of* (k_1, k_2, τ) *such that the closed loop system (42) and (43) is unstable.*

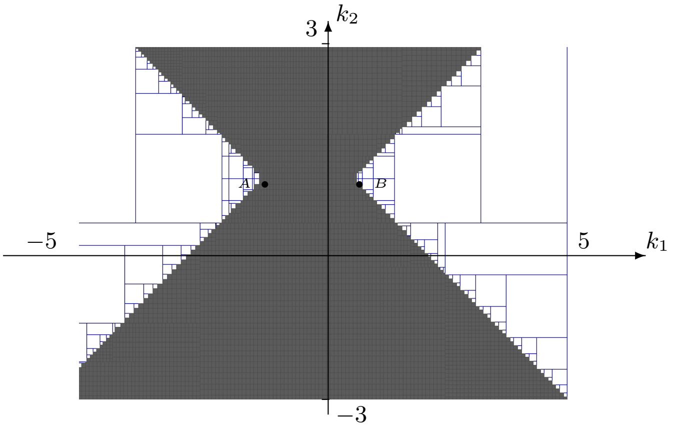

Fig. 6. Parametric regions (k_1, k_2) which ensure robust stability (white zone) of (42) and (43) in closed loop, for $\tau \in [2, 3]$, with $A = (-1.75, 0.75)$ and $B = (-0.25, 0.75)$

3.3 Other Control Problems

We are interested in this section by some other important control problems: the disturbance attenuation problem and the approximative tracking model for time-delays systems.

We choose these two control problems to show the potentiality of interval methods. The objective of this section is to pose simple problems, without establishing theoretical links with existing methods, as for instance H_∞-control. For these methods, the reader is referred to [16], [19], [26], and in references therein.

Consider a time-delays system with transfer function $H(s)$, and the control loop in Figure 7. Denote by u the control law, x the output, w a disturbance acting on u, r a reference trajectory and e the tracking error. The Laplace transforms of these signals are noted $(\hat{\cdot})(s)$.

Denote by $\mathbf{k}$ the set of all parameters of the feedback $k(s)$ to be determined.

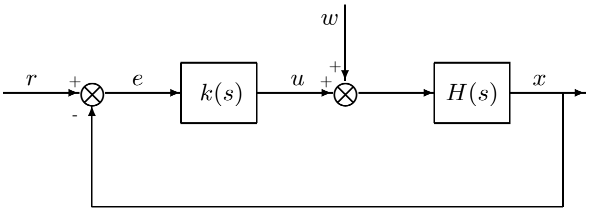

Fig. 7. Control loop of a time-delays system $H(s)$ with a feedback $k(s)$ to be determined

We have

$$\begin{array}{lll} S(s,\mathbf{k}) &= \frac{\hat{e}(s)}{\hat{r}(s)} &= \frac{1}{1+H(s)k(s)} \\ T(s,\mathbf{k}) &= \frac{\hat{x}(s)}{\hat{r}(s)} &= \frac{H(s)k(s)}{1+H(s)k(s)} \\ T_{wx}(s,\mathbf{k}) &= \frac{\hat{x}(s)}{\hat{w}(s)} &= \frac{H(s)}{1+H(s)k(s)} \end{array}. \qquad (44)$$

A performance specification can be expressed succinctly by $\|S(s,\mathbf{k})\|_\infty \leq \varepsilon$, or in a more generally form as $\|S(s,\mathbf{k})W_1(s)\|_\infty \leq 1$, where $W_1(s)$ is a weighting function whose magnitude is frequency dependent. A similar reasoning allows to establish inequalities on the transfer $T_{wx}(s,\mathbf{k})$ and $T(s,\mathbf{k})$, with direct applications, respectively to an attenuation disturbance problem and a robust stabilization problem. Furthermore, we have the property of internal stability if all transfer functions in (44) are stable (if others disturbances actuate in the closed loop, all internal transfers must be stable). We solve these frequency inequalities using interval computation.

Problem 3.7. *Let $T_{wx}(s,\mathbf{k})$ be given in (44). We want to find the set parameters $\mathbf{k}$ of $k(s)$ such that*

$$\forall\omega \in \Omega,\ |T_{wx}(j\omega,\mathbf{k})| \leq \frac{1}{|W(j\omega)|},\ \textit{and}\ T_{wx}(s,\mathbf{k})\ \textit{be stable}, \qquad (45)$$

with $\Omega \subset \mathrm{I\!R}$ a given finite frequency range and $W(s)$ a weighting function.

For example, we can take $W(j\omega) = \frac{1}{\varepsilon}$, $\forall \omega \in \Omega$, with $\varepsilon > 0$ a predefined attenuation parameter. For time-delays systems, as for systems without delays, this condition is often too restrictive [8]. A variable weighting function $W(s)$ allows to attenuate disturbance effects in function of frequency values.

In terms of interval computation, we suppose that $\mathbf{k}$ lie in an acceptable known box $[\mathbf{k}]$, and we are reduced to solve the set inversion problem

$$\mathbb{S} = \left\{\mathbf{k} \in [\mathbf{k}] \,\middle|\, \forall \omega \in \Omega,\ |T_{wx}(j\omega, \mathbf{k}) W(j\omega)| \leq 1,\ \text{with stability}\right\}. \tag{46}$$

The solution of Problem 3.7 is given by the algorithm Sivia, and we will choose the coefficients $\mathbf{k}$ of $k(s)$ which guarantee the disturbance attenuation Problem 3.7. The stability is verified in Section 3.2.

With a similar reasoning, we can ensure a disturbance attenuation for an uncertain plant $H(s)$, whose constant uncertain coefficients lie in given bounded intervals.

An interesting point, directly related to an optimal disturbance attenuation, is to find $\mathbf{k_o} \in [\mathbf{k}]$, if it exists, such that

$$\sup_{\omega \in \Omega} |T_{wx}(j\omega, \mathbf{k_o})| = \min_{\mathbf{k} \in [\mathbf{k}]} \sup_{\omega \in \Omega} |T_{wx}(j\omega, \mathbf{k})|,\ \textit{and } T_{wx}(s, \mathbf{k_o}) \textit{ be stable.} \tag{47}$$

This kind of problem can be solved with interval methods, as described in Example 3.9.

An another basic problem, although similar to the previous one, is the approximative tracking model.

Problem 3.8. *Let $H(s)$ be a given stable plant, and $H_M(s)$ a stable model transfer function for $H(s)$. The approximate tracking problem is to solve, with the choice of a stable feedback $k(s)$, the inequality*

$$\forall \omega \in \Omega,\ |H_M(j\omega) - H(j\omega)k(j\omega)| \leq \frac{1}{|W(j\omega)|}, \tag{48}$$

with $\Omega \subset \mathrm{I\!R}$ a given finite frequency range and $W(s)$ a given weighting function.

Problem 3.8 is written in a similar form of Problem 3.7, *i.e.*

$$\mathbb{S} = \left\{\mathbf{k} \in [\mathbf{k}] \,\middle|\, \forall \omega \in \Omega,\ |(H_M(j\omega) - H(j\omega)k(j\omega))W(j\omega)| \leq 1\right\}, \tag{49}$$

with $k(s)$ be stable. A robust approximate tracking model can be defined and solved with interval methods for uncertain plants. Only the number of parameters to be determined is increased, and the methodology is the same that the previous one.

Example 3.9. *Let a transfer function between a disturbance $w(t)$ and an output $x(t)$:*

$$H(s) = \frac{\hat{x}(s)}{\hat{w}(s)} = \frac{1}{s + a\mathrm{e}^{-s\tau} + b}, \tag{50}$$

with $\tau = 1$, $a = b = 1$. The transfer $H(s)$ is stable (see Section 3.2).

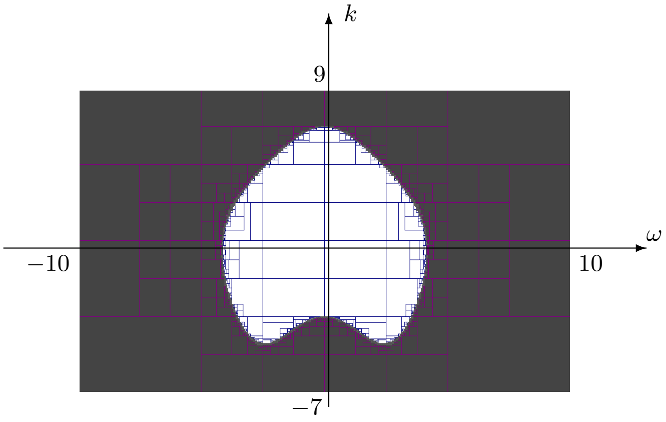

Fig. 8. Set solution $k \in [\mathbf{k}]$ of Example 3.9. Frequencies ω are reported in x-coordinates, and coefficients k in y-coordinates. The size of the white central zone is almost $[-4.1, 4.1] \times [-4.5, 7]$.

We take a feedback $k(s)$ of proportional type, i.e. $\hat{u}(s) = k\hat{x}(s)$, where k is a coefficient to be determined. We want to guarantee

$$\forall \omega \in \Omega,\ |T_{wx}(j\omega, k)| \leq \varepsilon,\ \textit{and}\ T_{wx}(s,k)\ \textit{be stable},$$

where $\Omega = [-1000, 1000]$, $\varepsilon = 0.2$, and $T_{wx}(s,k)$ is given by

$$T_{wx}(s,k) = \frac{1}{s + a\mathrm{e}^{-s\tau} + b - k} \tag{51}$$

For $k \in [\mathbf{k}] = [-7, 9]$, we solve the Problem 3.7 of set inversion by Sivia, *to obtain the set solution $k \in [\mathbf{k}]$ reported on Figure 8, in function of $\omega \in \Omega$. The white central zone is a no-solution zone, i.e. for a given $k \in [-4.5, 7]$, $\forall \omega \in [-4.1, 4.1]$, $|T_{wx}(j\omega, k)| > \varepsilon$. In the dark-grey zone, the inequality $|T_{wx}(j\omega, k)| \leq \varepsilon$ holds. Then, if we take $k \in [-4.5, 7]$, the norm constraint is not satisfied, and a more complex feedback must be choosen.*

Solutions $k \in [\mathbf{k}]$ are also included in $[-7, -4.5] \cup [7, 9]$. The stability analysis in closed loop implies that $k < -2$, i.e. the set solution is $[-7, -4.5]$.

Take for example $k = -5$. The transfer function (51) is stable, and a Bode magnitude plot is reported on Figure 9. We verify that

$$\sup_{\omega \in \mathrm{I\!R}} (20\log10|T_{wx}(j\omega)|) = -14 < 20\log10(\varepsilon) = -13.98$$

A similar analysis can be done with uncertain constant parameters (a, b, τ).

Consider now the problem of optimal attenuation, i.e. of finding $k_o \in [\mathbf{k}]$ such that

$$\sup_{\omega \in \Omega} |T_{wx}(j\omega, k_o)| = \min_{k \in [\mathbf{k}]} \sup_{\omega \in \Omega} |T_{wx}(j\omega, k)|,\ \textit{and}\ T_{wx}(s, k_o)\ \textit{be stable}. \tag{52}$$

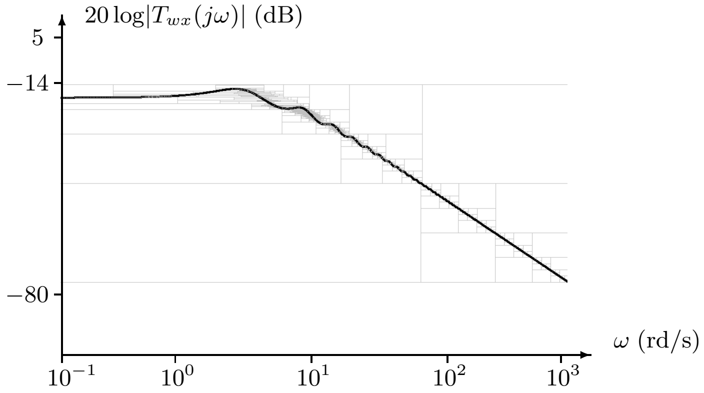

Fig. 9. Bode magnitude plot of (51), with $k = -5$

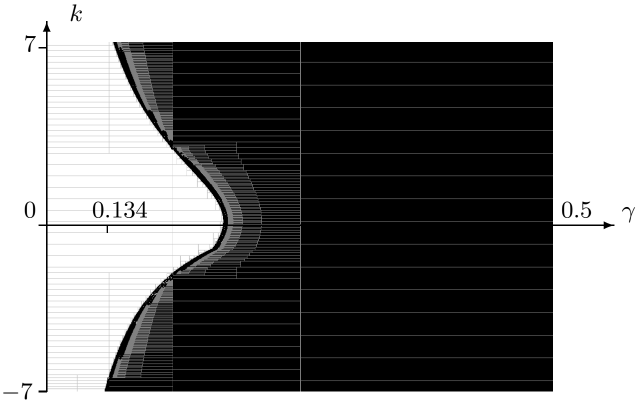

Fig. 10. Set solution (γ, k) of (53)

To solve this optimization problem, we use SIVIA *to analyze the set*

$$\mathbb{S} = \{(k, \gamma) \in [\mathbf{k}] \times \Upsilon \,|\, \forall \omega \in \Omega,\ |T_{wx}(j\omega, k)| \leq \gamma\},$$

Solutions of this problem are given in Figure 10, in the plane (γ, k), with $\gamma \in \Upsilon = [0, 0.5]$ and $k \in [-7, 7]$. The white zone (γ, k) is a no-solution zone, i.e. exists $\omega \in \Omega$ such that $|T_{wx}(j\omega, k)| > \gamma$. The black zone is a solution zone, i.e. $\forall \omega \in \Omega$, $|T_{wx}(j\omega, k)| \leq \gamma$. Moreover, on Figure 10, we can determine k_o in (52). In fact, it corresponds to

$$k_o = \min_{\gamma \in \Upsilon} \{k \,|\, \forall \omega \in \Omega,\ |T_{wx}(j\omega, k)| \leq \gamma\},$$

that is in our case $k_o = -7$. The optimal value of disturbance attenuation is

$$\sup_{\omega \in \Omega} |T_{wx}(j\omega, k_o)| = 0.134.$$

Example 3.10. *Let* $H(s) = \frac{e^{-s}}{s+s_0}$ *a uncertain plant with* $s_0 \in [0.5, 1.5]$, $H_M(s) = \frac{e^{-s}}{s+2}$ *a model transfer function for* $H(s)$. *We want to ensure a robust approximative model tracking with a controller* $k(s)$ *of the form* $k(s) = \frac{p(s+q)}{s+2}$, *such that*

$$\forall \omega \in \Omega = [-1000, 1000],\ |E(j\omega, \mathbf{k})| = |H_M(j\omega) - H(j\omega)k(j\omega)| \leq 0.2, \qquad (53)$$

for $s_0 \in [0.5, 1.5]$ *and* $\mathbf{k} = (p, q) \in [-10, 10] \times [-10, 10]$ *which are the parameters to be determined.*

We are analyzing a problem of type 3.6. The solutions plot is reported in the parametric plane (p, q) *on Figure 11. The grey zone is the solution set of* (p, q) *such that* $\forall(\omega, s_o) \in \Omega \times [0.5, 1.5]$, $|E(j\omega, \mathbf{k})| \leq 0.2$. *The white zone is the*

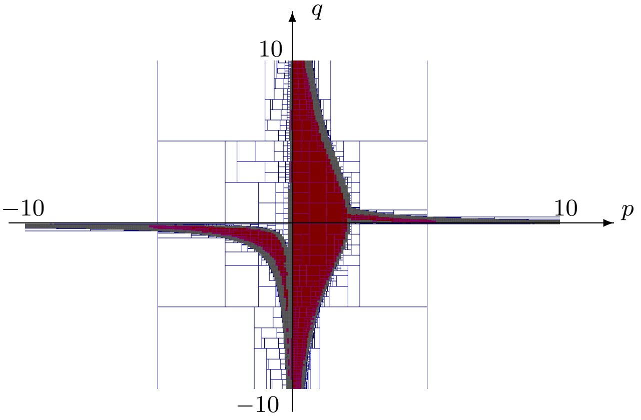

Fig. 11. Set solution (p, q) of (53). The grey zone is the solution set

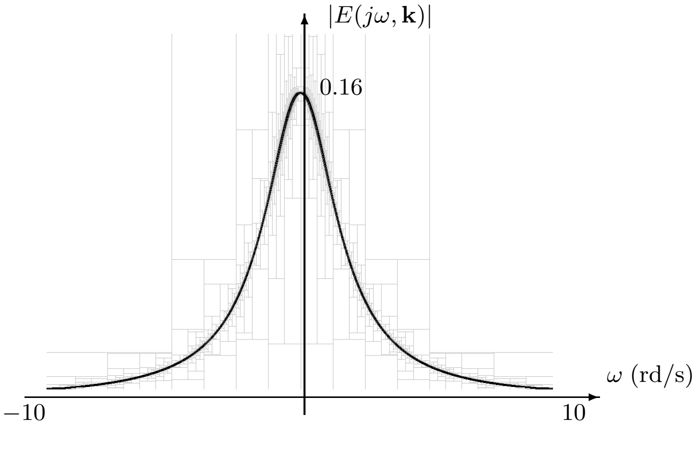

Fig. 12. Magnitude plot of $|E(j\omega, \mathbf{k})|$, for $s_o = 1.5$, and $\mathbf{k} = (p, q) = (1, 1)$

no-solution set of (p, q) *such that* $\exists(\omega, s_o) \in \Omega \times [0.5, 1.5]$ *with* $|E(j\omega, \mathbf{k})| > 0.2$*. For example, taking* $s_0 = 1.5$*,* $p = 1$ *and* $q = 1$*, we are in the grey zone. A plot of the magnitude* $|E(j\omega, \mathbf{k})|$ *with respect to* ω *is reported on Figure 12. We verify that* $\sup_{\omega \in \Omega} |E(j\omega, \mathbf{k})| = 0.16 < 0.2$*. A choice of* $\mathbf{k}$ *can be made to ensure a minimal tracking error, as seen in the previous example.*

4 Conclusion

In this paper, we apply interval computation to time-delays systems, to solve some control problems, as robust stability, stabilization, or disturbance attenuation by feedback. Basic illustrative examples are reported, to clarify interval methods.

In spite of a limit on the parameters number, interval computation allows to obtain guaranteed solutions for a large number of control problems, and that in an original way for time-delays systems. Graphical solutions allow an easy interpretation of physical phenomena.

References

1. Barmish B. R., Hollot C. V., Kraus F. J., Tempo R. (1992) Extreme Points Results for Robust Stabilization of Interval Plants with First Order Compensators. IEEE Transactions on Automatic Control 37:707–714
2. Bellman R., Cooke K. L. (1963) Differential–difference equations, Academic Press, New York
3. Benhamou F., Goualard F., Granvilliers L., Puget J. F. (1999) Revising Hull and Box Consistency. In Proc. Int. Conf. Logic Programming, Las Cruces, New Mexico
4. Chen J., Niculescu S. I. (2004) Robust Stability of Quasipolynomials: Vertex–Type Tests and Extensions. In Proc. ACC, pp. 4159–4164, Boston, Massachussets
5. Cleary J. G. (1987) Logical arithmetic, Future Computing Systems 2:125–149
6. Dao M., Di Loreto M., Jaulin L., Lafay J. F., Loiseau J. J. (2004) Application du calcul par intervalles aux systèmes à retards. In Proc. CIFA, Douz, Tunisia
7. Davis E. (1987) Constraint propagation with interval labels. Artificial Intelligence 32:281–331
8. Doyle J., Francis B., Tannenbaum A. (1992) Feedback Control Theory, Macmillan, Houndmills
9. Engelborghs K., Roose D. (2002) On stability of LMS–methods and characteristic roots of delay differential equations. SIAM J. Numerical Analysis 40:629–650
10. Gu K., Kharitonov V. L., Chen J. (2002) Stability of time–delay systems. Birkhauser, Boston
11. Hale J. K., Verduyn Lunel S. M. (1993) Introduction to Functional Differential Equations. Springer–Verlag, Berlin Heidelberg New York
12. Hohenbichler N., Ackermann J. (2003) Computing stable regions in parameter spaces for a class of quasipolynomials. In Proc. 4th IFAC TDS, Rocquencourt, France
13. Jaulin L., Kieffer M., Didrit O., Walter E. (2001) Applied interval analysis, Springer, Berlin Heidelberg New York

14. Kamen E. W. (1982) Linear systems with commensurate time delays: Stability and stabilization independent of delay. IEEE Transactions on Automatic Control 27:367–375
15. Kharitonov V. L, Zhabko A. (1994) Robust Stability for Time–Delay Systems. IEEE Transactions on Automatic Control 39:2388–2397
16. Kojima A., Ishijima S. (2001) H_∞ control for preview and delayed strategies. In Proc. 40th IEEE Conf. Decision Control, pp. 991–996, Orlando, Florida
17. Kolmanovskii V., Myshkis A. (1992) Applied theory of functional differential equations, Kluwer Academic Publishers, Dordrecht
18. Lhomme O. (1993) Consistency techniques for numeric CSPs. Proc. Int. Conf. Artificial Intelligence, pp. 232–238, Chambéry, France
19. Meinsma G., Zwart H. (2000) On H_∞ control for dead–time systems. IEEE Transactions on Automatic Control 45:272–285
20. Moore R. E. (1966) Interval Analysis. Prentice–Hall, Englewood Cliffs
21. Moore R. E. (1979) Methods and Applications of Interval Analysis. SIAM, Philadelphia
22. Moore R. E., Ratschek H. (1988) Inclusion function and global optimization. Mathematical Programing 41(3):341–356
23. Mori T., Kokame H. (1989) Stability of $\dot{x}(t) = Ax(t)+Bx(t-\tau)$. IEEE Transactions on Automatic Control 34:460–462
24. Nagpal K., Ravi R. (1997) H_∞ control and estimation problems with delayed measurements: State space solutions. SIAM J. Control Optimization 35:1217–1243
25. Neumaier A. (1990) Interval Methods for Systems of Equations. Cambridge University Press, Cambridge
26. Niculescu S. I. (2001) Delay Effects on Stability: A Robust Control Approach. Springer, Berlin Heidelberg New York
27. Santos J., Mondié S., Kharitonov V. (2003) Robust stability of time delay systems and the finite inclusions theorem. In Proc. 4th IFAC TDS, Rocquencourt, France
28. Tzypkin Y. Z., Fu M. (1993) Robust stability of time–delay systems with an uncertain time–delay constant. International J. Control 57:865–879
29. Verriest E. I., Fan M., Kullstam J. (1993) Frequency domain robust stability criteria for linear delay systems. In Proc. 32nd IEEE Conf. Decision Control, pp. 3473–3478, San Antonio, Texas
30. Vyhlídal T., Zítek P. (2003) Quasipolynomial mapping based rootfinder for analysis of time delay systems. In Proc. 4th IFAC TDS, Rocquencourt, France
31. Walter E., Jaulin L. (1994) Guaranteed characterization of stability domains via set inversion. IEEE Transactions on Automatic Control 39:886–889