

# Application du calcul par intervalles aux systèmes à retards

Massa DAO<sup>1</sup>, Michaël DI LORETO<sup>2</sup>, Luc JAULIN<sup>1</sup>, Jean-François LAFAY<sup>2</sup> et  
Jean Jacques LOISEAU<sup>2</sup>

<sup>1</sup>LISA, CNRS-FRE-2656,  
Université d'Angers, 62 Avenue Notre Dame du Lac, 49000 Angers, France

dao@istia.univ-angers.fr, jaulin@univ-angers.fr

<sup>2</sup>IRCCyN, CNRS-UMR-6597  
1 rue de la Noë, BP 92101, 44321 Nantes cedex 03, France

{michael.di-loreto, jean-francois.lafay, jean-jacques.loiseau}@irccyn.ec-nantes.fr

*Résumé*— Le but de ce papier est de montrer, à travers quelques exemples issus de la littérature, que le calcul par intervalles combiné aux techniques de propagation de contraintes est un outil particulièrement bien adapté à l'étude des systèmes à retards.

*Mots-clés*— Analyse par intervalles, inversion ensembliste, propagation de contraintes, systèmes à retards, norme  $H_\infty$ , stabilité robuste, systèmes à retards de type neutre, quasipolynômes.

## I. INTRODUCTION

On propose dans ce papier d'appliquer les méthodes du calcul par intervalles ([2], [13], [14] et [9]) aux systèmes à retards, afin de caractériser leurs propriétés fondamentales, et d'analyser la stabilité robuste, la norme  $H_\infty$  de leurs fonctions de transfert ou encore le lieu des racines d'un quasipolynôme.

Les caractérisations de ces propriétés fondamentales ont été largement étudiées dans cette dernière décennie, avec l'obtention de nombreuses méthodes numériques et algorithmiques afin de résoudre des problèmes classiques en Automatique ([16], [7]), tels que la stabilisation robuste ou encore le rejet de perturbation. Cependant, de nombreuses méthodes proposées ne s'appliquent que pour une classe particulière de systèmes à retards, et ces méthodes présentent en général des difficultés de mise en oeuvre. Dans ce sens, le calcul par intervalles permet d'apporter une solution globale et garantie à certains problèmes numériques. Les problématiques liées à ces systèmes sont un sujet de recherche actuel (voir par exemple [8], [11], [19] et [21]).

Les systèmes à retards ont une dynamique dont l'évolution à un instant donné dépend de son comportement passé sur une période finie.

Dans le cas linéaire, l'étude de leur dynamique est basée sur celle de leur équation caractéristique, qui est une équation algébrique transcendante, typiquement un quasipolynôme ([1], [18]). Il existe de nombreux résultats concernant le test de la stabilité des quasipolynômes ou de certaines fa-

milles de quasipolynômes (voir par exemple [16], [17] et [7] et leurs références). Beaucoup de ces travaux concernent la stabilité robuste en fonction d'incertitudes paramétriques sur les coefficients de quasipolynômes. Un problème classique consiste par exemple à déterminer les intervalles  $[a_{ik}, \bar{a}_{ik}] \subset \mathbb{R}$  de variation des coefficients  $a_{ik}$  (pour  $i = 0, \dots, n$  et  $k = 0, \dots, m$ ) tels que la stabilité du quasipolynôme  $q(s) = \sum_{i=0}^n \sum_{k=0}^m a_{ik} s^i e^{k\theta s}$  soit conservée à l'intérieur du pavé correspondant, avec un retard  $\theta$  constant (voir par exemple [10], [19]). Une autre question fréquemment abordée est la stabilité robuste en fonction des variations du retard dans un intervalle donné [17].

La stabilité d'un quasipolynôme  $q(s)$  revient à l'existence d'un réel  $\sigma > 0$  tel que toutes les racines  $r_j$  de  $q(s)$  soient à partie réelle strictement inférieure à  $-\sigma$ . Ainsi, tester la stabilité pour un système à retards revient à vérifier l'absence de racines à partie réelle positive de l'équation caractéristique du système. Une variante consiste à vérifier que le transfert du système est borné à la droite de cette verticale d'abscisse  $-\sigma$ . Ces tests sont en général semi-analytiques, et leur mise en oeuvre numérique peut être assez complexe ([17], [7]).

Les algorithmes de calcul par intervalles, qui ont été développés depuis quelques années (voir par exemple [9], [13], [14] et [15]), sont bien adaptés aux problèmes rencontrés, comme par exemple tester l'absence garantie de zéros de quasipolynômes ou montrer qu'une fonction de transfert transcendante est bornée, principalement quand les paramètres du système sont soumis à des incertitudes.

On se propose d'illustrer sur quelques exemples de systèmes à retards cette application des algorithmes par intervalles. Ce document est structuré de la manière suivante; dans le paragraphe II, on effectue un bref rappel du calcul par intervalles, pour décrire dans le paragraphe III l'algorithme d'inversion ensembliste. Puis, le paragraphe IV est consacré au tracé du gain d'une fonction de transfert d'un système à retard de type neutre, en comparant notam-

ment les résultats obtenus avec Matlab et Proj2D<sup>1</sup>, un solveur utilisant le calcul par intervalles et la propagation de contraintes. Enfin, dans le paragraphe V, on traite quelques exemples de test de stabilité robuste de systèmes à retard.

## II. PAVÉS ET CONTRACTEURS

Dans cette partie, on définit quelques notions sur les pavés et les contracteurs, à la base du calcul par intervalles ([13], [14], [2] et [9]). Ces notions seront utilisées pour décrire l'algorithme d'inversion ensembliste.

### A. Définitions

*Définition II.1:* Un pavé ou vecteur d'intervalles  $[\mathbf{x}]$  de  $\mathbb{R}^n$  est le produit cartésien de  $n$  intervalles réels. On note  $\underline{\mathbf{x}}$  et  $\bar{\mathbf{x}}$  les bornes respectivement inférieures et supérieures de  $[\mathbf{x}]$ . L'intervalle  $[\mathbf{x}_i] = [\underline{\mathbf{x}}_i, \bar{\mathbf{x}}_i]$  correspond à la  $i^{\text{ème}}$  composante de  $[\mathbf{x}]$ . La longueur de  $[\mathbf{x}]$  est  $w([\mathbf{x}]) = \max_{i=1,2,\dots,n} (w([\mathbf{x}_i]))$ , avec  $w([\mathbf{x}_i]) = \underline{\mathbf{x}}_i - \bar{\mathbf{x}}_i$ .

*Définition II.2:* La bisection d'un pavé  $[\mathbf{x}]$  est une opération qui génère les pavés  $L[\mathbf{x}]$  et  $R[\mathbf{x}]$  tels que

$$\begin{cases} L[\mathbf{x}] & \doteq [\mathbf{x}_1] \times \dots \times [\underline{\mathbf{x}}_i, \mathbf{x}_{ic}] \times \dots \times [\mathbf{x}_n] \\ R[\mathbf{x}] & \doteq [\mathbf{x}_1] \times \dots \times [\mathbf{x}_{ic}, \bar{\mathbf{x}}_i] \times \dots \times [\mathbf{x}_n] \end{cases} \quad (1)$$

où  $\mathbf{x}_{ic} = \frac{\bar{\mathbf{x}}_i + \underline{\mathbf{x}}_i}{2}$  et  $i = \min\{j \mid w([\mathbf{x}]) = w([\mathbf{x}_j])\}$

*Définition II.3:* Soient  $[\mathbf{x}]$  un pavé de  $\mathbb{R}^n$  et  $\mathbb{S}$  un ensemble défini par des inégalités non linéaires. Un contracteur pour  $\mathbb{S}$  est l'opérateur

$$\mathcal{C}_{\mathbb{S}}: \mathbb{I}\mathbb{R}^n \longrightarrow \mathbb{I}\mathbb{R}^n \\ [\mathbf{x}] \longmapsto \mathcal{C}_{\mathbb{S}}([\mathbf{x}]) \supset [[\mathbf{x}] \cap \mathbb{S}] \quad , \quad (2)$$

où  $\mathbb{I}\mathbb{R}^n$  est l'ensemble des pavés de  $\mathbb{R}^n$  et  $[\mathbf{A}]$  est le plus petit pavé contenant l'ensemble  $\mathbf{A}$ . Une illustration de la contraction de  $[\mathbf{x}]$  par rapport à  $\mathbb{S}$  est donnée par la figure ci-dessous.

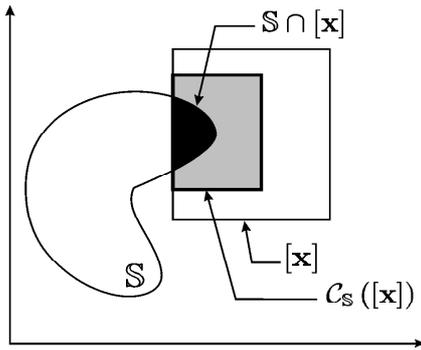


Figure 1. Contraction d'un pavé  $[\mathbf{x}]$  par rapport à  $\mathbb{S}$ .

Un contracteur optimal ou idéal pour  $\mathbb{S}$  est un contracteur  $\mathcal{C}_{\mathbb{S}}^*$  tel que  $\mathcal{C}_{\mathbb{S}}^*([\mathbf{x}]) = [[\mathbf{x}] \cap \mathbb{S}]$ . Il n'y a pas de méthode exacte qui permette de déterminer le contracteur optimal pour tout ensemble  $\mathbb{S}$  défini par des inégalités. Une approche désormais classique et efficace est celle induite par la propagation de contraintes ([5], [4]). Cette méthode, que l'on va maintenant présenter, sert à la réalisation de nos contracteurs.

<sup>1</sup>disponible sur <http://www.istia.univ-angers.fr/~dao/Proj2DV3.zip>

### B. Propagation de contraintes sur les intervalles

Le principe de la propagation des contraintes repose sur l'arithmétique des intervalles. Par exemple, la somme et la soustraction de 2 intervalles se définissent naturellement par

$$\begin{aligned} [\mathbf{x}] + [\mathbf{y}] &= [\underline{\mathbf{x}} + \underline{\mathbf{y}}, \bar{\mathbf{x}} + \bar{\mathbf{y}}] \\ [\mathbf{x}] - [\mathbf{y}] &= [\underline{\mathbf{x}} - \bar{\mathbf{y}}, \bar{\mathbf{x}} - \underline{\mathbf{y}}] \end{aligned}$$

Les autres opérateurs ( $*$ ,  $\div$ ) et les fonctions élémentaires ( $\sin$ ,  $\cos$ ,  $\tan$ , ...) peuvent être aussi définis (voir [15]). Une telle arithmétique permet alors de contracter les pavés relativement à un ensemble  $\mathbb{S}$  défini par des inégalités. La propagation des contraintes est décrite à travers l'exemple suivant.

*Exemple II.1:* Soit  $\mathbb{S}$  un ensemble défini par

$$\mathbb{S} = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_3 = x_1 + x_2\} \quad (3)$$

et le pavé  $[\mathbf{x}] = [\mathbf{x}_1] \times [\mathbf{x}_2] \times [\mathbf{x}_3]$ , avec  $[\mathbf{x}_1] = [-1, 2]$ ,  $[\mathbf{x}_2] = [0, 3]$  et  $[\mathbf{x}_3] = [4, 8]$ . Ainsi, pour  $(x_1, x_2, x_3) \in [\mathbf{x}]$ , on obtient

$$\begin{aligned} x_1 &\in [\mathbf{x}_1] \cap ([\mathbf{x}_3] - [\mathbf{x}_2]) = [1, 2] \\ x_2 &\in [\mathbf{x}_2] \cap ([\mathbf{x}_3] - [\mathbf{x}_1]) = [2, 3] \\ x_3 &\in [\mathbf{x}_3] \cap ([\mathbf{x}_1] + [\mathbf{x}_2]) = [4, 5] \end{aligned} \quad (4)$$

Le pavé obtenu après contraction de  $[\mathbf{x}]$  suivant  $\mathbb{S}$  est (voir [9]) :

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

## III. ALGORITHME D'INVERSION ENSEMBLISTE (SIVIA)

Un algorithme d'inversion ensembliste permet d'approximer l'ensemble  $\mathbb{X}$  décrit par

$$\mathbb{X} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\} = [\mathbf{x}] \cap \mathbf{f}^{-1}([\mathbf{y}]), \quad (5)$$

où  $\mathbf{f}$  est une fonction continue de  $\mathbb{R}^n$  dans  $\mathbb{R}^m$  (voir [15]),  $[\mathbf{y}]$  un pavé de  $\mathbb{R}^m$  et  $[\mathbf{x}]$  un pavé quelconque de  $\mathbb{R}^n$ . Cette approximation sera faite par un sous-pavage, c'est à dire par une union de pavés disjoints.

L'algorithme d'inversion ensembliste SIVIA (Set Inverter Via Interval Analysis) est de type arborescent (*branch and bound*). Cet algorithme est décrit dans [9]. On se contente ici d'en donner une version récursive. Cet algorithme suppose que nous avons à notre disposition un contracteur  $\mathcal{C}_{\mathbb{X}}$  pour  $\mathbb{X}$  et une liste de pavés  $\mathcal{L}$  initialement vide. Le premier appel de l'algorithme se fait par  $\text{SIVIA}([\mathbf{x}])$ .

SIVIA (Entrée : $[\mathbf{x}]$ )	
<b>1</b>	Si $(w([\mathbf{x}]) \leq \varepsilon)$ : $\mathcal{L} \leftarrow \mathcal{L} \cup \{[\mathbf{x}]\}$ , aller à <b>6</b> ;
<b>2</b>	$[\mathbf{x}] \leftarrow \mathcal{C}_{\mathbb{X}}([\mathbf{x}])$ ;
<b>3</b>	Si $([\mathbf{x}] = \emptyset)$ , aller à <b>6</b> ;
<b>4</b>	Bisection : $([\mathbf{x}]) \rightarrow (L[\mathbf{x}], R[\mathbf{x}])$ ;
<b>5</b>	SIVIA( $L[\mathbf{x}]$ ) ; SIVIA( $R[\mathbf{x}]$ ) ;
<b>6</b>	Fin.

Tableau 1. Version récursive de la routine SIVIA.

Cette routine génère un sous-pavage contenant l'ensemble  $\mathbb{X}$ , et  $\varepsilon$  est un paramètre de précision qui limite la bisection de  $[\mathbf{x}]$ . Cet algorithme a été inséré dans le solveur Proj2D. L'adaptation de cet algorithme et de la propagation de contraintes par intervalles aux problèmes liés aux systèmes

à retards se révèle particulièrement efficace. Par exemple, pour le calcul des zéros d'un quasipolynôme  $q(s)$ ,  $s \in \mathbb{C}$ , on aura

$$q(s) = 0 \Leftrightarrow \begin{cases} \operatorname{Re} q(s) = 0 \\ \operatorname{Im} q(s) = 0 \end{cases},$$

avec  $s = x + iy$ ,  $x \in [\underline{x}, \bar{x}]$  et  $y \in [\underline{y}, \bar{y}]$ . De même, tester  $|H(s)| < M$  pour  $s \in \mathbb{C}^+$ ,  $M \in \mathbb{R}^+$ , revient à écrire  $|H(s)| \in [0, M]$ , avec  $s = x + iy$ ,  $x \in [0, \bar{x}]$  et  $y \in [\underline{y}, \bar{y}]$ . On se propose d'illustrer cette application sur quelques exemples de systèmes à retard.

#### IV. ETUDE DU GAIN D'UN SYSTÈME À RETARD

Ce paragraphe est consacré à l'application du calcul par intervalles au tracé du gain d'un système à retard de type neutre. Le but de cet exemple est de montrer l'intérêt de l'algorithme SIVIA dans le cadre du tracé garanti d'un graphe d'une fonction. On utilise Matlab et Proj2D afin de comparer les résultats. Un lieu des pôles dans le plan complexe complète ce paragraphe.

Considérons le système à retard de type neutre dont la fonction de transfert est

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{(s+1)(s(1-e^{-s})+1)} \quad (6)$$

Sur l'axe imaginaire, le gain de cette fonction de transfert s'écrit  $G(\omega) = |H(j\omega)|$ , avec

$$G(\omega) = \frac{1}{A} \frac{1}{\sqrt{(1-\omega \sin(\omega))^2 + \omega^2(1-\cos(\omega))^2}} \quad (7)$$

où  $A = \sqrt{1+\omega^2}$ . Le graphe du gain en fréquence  $h = G(\omega)$  est donné par l'ensemble

$$\mathbb{S} = \{(\omega, h) \in [\omega] \times [h] \mid G(\omega) - h = 0\} \quad (8)$$

où  $[h]$  et  $[\omega]$  sont 2 intervalles supposés contenir respectivement les valeurs *a priori* possibles du gain (7) et celles des fréquences pour le tracé du gain.

##### A. Tracé du graphe sous Matlab

A l'aide de Matlab, on effectue le tracé des graphes par interpolation et lissage. La technique consiste à relier les éléments d'un nuage de points par des segments de droite ou des tronçons de courbe. L'inconvénient de cette méthode réside dans le risque de ne pas détecter la présence de pics de faible largeur. Ce problème reste présent quel que soit le pas d'échantillonnage choisi. En effet, une brève analyse de (6) montre que les pôles du système se situent d'une part en  $-1$  pour le pôle simple, et pour la partie neutre, admettent l'axe imaginaire comme direction asymptotique, avec comme approximation  $s_k = j(2k\pi + \frac{1}{2k\pi}) + o(\frac{1}{k^2})$ ,  $k \in \mathbb{Z}^*$ , et  $H(s_k) = -2 - \frac{2j}{3k\pi} + o(\frac{1}{k^2})$  (voir [18] et [1]). Par conséquent, dès que  $k$  augmente en module (pour  $|k| \geq 3$ ), (7) présente des pics régulièrement espacés de  $2\pi$ , dont la largeur est décroissante en fonction de  $k$ , et dont l'amplitude est constante.

On risque donc d'obtenir, pour des fréquences non nulles, des amplitudes erronées de ces pics, voir même l'inexistence de certains pics par suite de l'échantillonnage réalisé sous

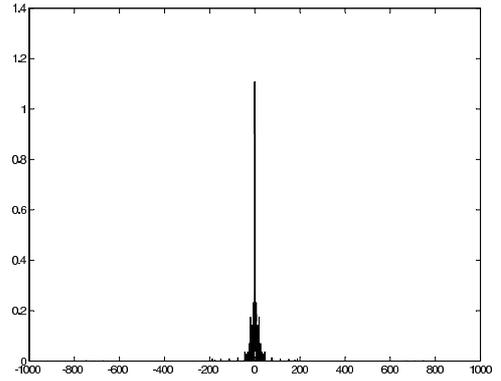


Figure 2. Gain  $h = G(\omega)$  sous Matlab, avec un pas d'échantillonnage de  $\Delta\omega_1 = 0.1$  Hz, pour  $\omega \in [-1000, 1000]$ .

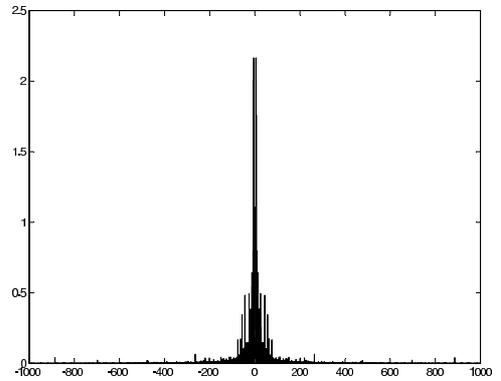


Figure 3. Gain  $h = G(\omega)$  sous Matlab, avec un pas d'échantillonnage de  $\Delta\omega_2 = 0.001$  Hz, pour  $\omega \in [-1000, 1000]$ .

Matlab. Ce phénomène s'accroît lorsque la fréquence augmente, car la largeur de ces pics est de plus en plus faible. On illustre ce problème avec 2 tracés de  $h = G(\omega)$  pour 2 pas d'échantillonnage différents, respectivement  $\Delta\omega_1 = 0.1$  Hz et  $\Delta\omega_2 = 0.001$  Hz, avec  $\omega \in [-1000, 1000]$ .

Sur les figures 2 et 3, très peu de pics sont présents dans le graphe du gain, et cela malgré un pas d'échantillonnage très petit. Les amplitudes des pics obtenus sont par ailleurs différentes, et n'ont aucune signification. La question que l'on peut alors se poser est la suivante : lequel de ces 2 graphes est conforme à la courbe  $h = G(\omega)$  ?

##### B. Tracé du graphe sur Proj2D

Le solveur Proj2D permet, grâce à SIVIA, d'approximer par des pavés tous les points d'un graphe, et donc de pallier à l'inconvénient mis en évidence dans le paragraphe IV-A. Avec ces informations, les valeurs prises par le gain seront bornées, et ceci dans un intervalle de fréquence initial choisi. A titre de comparaison avec ce qui a été fait précédemment, on réalise sur Proj2D le tracé de plusieurs graphes dans des domaines de fréquence différents. Les zones grises renferment de façon garantie tous les points possibles du graphe  $h = G(\omega)$ . Les figures 4 et 5 permettent de vérifier que les pics, de largeur décroissante en fonction de la fréquence, ont tous une même amplitude de

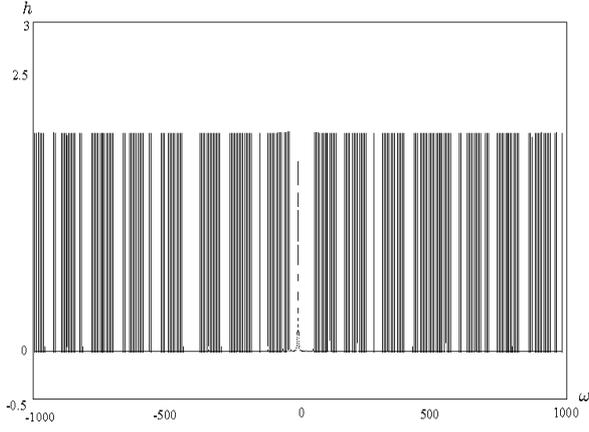


Figure 4. Gain  $h = G(\omega)$  sur l'intervalle fréquentiel  $[-1000, 1000]$  sous Proj2D, avec  $h \in [-0.5, 3]$ .

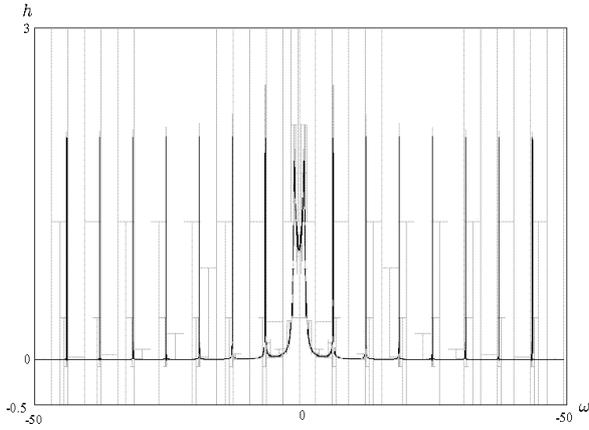


Figure 5. Gain  $h = G(\omega)$  sur l'intervalle fréquentiel  $[-50, 50]$  sous Proj2D, avec  $h \in [-0.5, 3]$ .

manière asymptotique, et qu'ils sont régulièrement espacés de  $2\pi$ . Evidemment, en utilisant Matlab, le résultat, malgré un pas d'échantillonnage choisi aussi petit que possible, ne sera jamais compatible avec celui du calcul par intervalles, par suite de sa structure.

De plus, l'analyse par intervalle permet, en vue des résultats obtenus, de déterminer la norme  $H_\infty$  du transfert (6). En effet, le gain maximal vaut environ  $\gamma \simeq 2$ , et ce dernier correspond à la norme  $H_\infty$  du transfert si celui-ci est stable (voir [6]).

### C. Recherche de pôles dans le plan complexe

Dans ce paragraphe, on s'intéresse à la localisation des pôles du système (6). Pour cela, on utilise Proj2D afin de résoudre l'équation caractéristique de type neutre (9) dans un domaine borné du plan complexe.

$$(s+1)(s(1-e^{-s})+1) = 0, \quad s \in \mathbb{C} \quad (9)$$

En notant une racine de (9) sous la forme  $r_k = (\text{Re } r_k, \text{Im } r_k)$ , l'ensemble  $\mathbb{S}$  des pôles du système s'écrit sous la forme  $\mathbb{S} = \{(-1, 0) \cup \mathbb{S}_n\}$ , avec  $\mathbb{S}_n$  l'ensemble des racines du terme neutre  $s(1-e^{-s})+1 = 0$ . En particulier, on a

$$\mathbb{S}_n = \{(x, y) \in \mathbb{R}^2 \mid \mathbf{f}(x, y) = \mathbf{0}\}, \text{ avec}$$

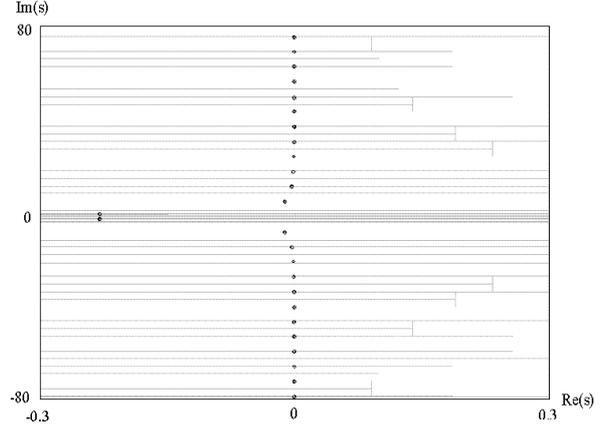


Figure 6. Lieu de l'ensemble  $\mathbb{S}_n$  des pôles du système (6), avec  $(x, y) \in [-0.3, 0.3] \times [-80, 80]$ .

$$\mathbf{f}(x, y) = \begin{pmatrix} x - (x \cos y + y \sin y) e^{-x} + 1 \\ y + (x \sin y - y \cos y) e^{-x} \end{pmatrix}^\top.$$

En termes d'inversion ensembliste, on a  $\mathbb{S}_n = \mathbf{f}^{-1}(\mathbf{0})$ . Sur la figure 6, les pavés noirs indiquent l'emplacement des pôles dans le plan complexe. On vérifie de cette manière la direction asymptotique des pôles du terme neutre qui est l'axe imaginaire.

### V. STABILITÉ ROBUSTE DE SYSTÈMES À RETARDS

Dans ce paragraphe, on présente quelques tests de stabilité robuste appliqués à des systèmes à retard. La caractérisation de la stabilité nécessite le calcul des racines des équations caractéristiques des systèmes, qui sont des quasipolynômes en la variable  $s$ , en se limitant notamment au demi-plan complexe droit fermé.

#### A. Un système à retard de type retardé

On considère le système à retard de type retardé (voir [16]) :

$$\begin{cases} \dot{x}(t) = -ax(t) - bx(t - \tau) \\ (a, b, \tau) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \end{cases} \quad (10)$$

avec une condition initiale appropriée. L'équation caractéristique du système (10) est donnée par :

$$s + a + be^{-s\tau} = 0 \quad (11)$$

qui est une équation transcendantale, avec en général une infinité de solutions. Son analyse peut se faire par exemple par la méthode de  $\mathcal{D}$ -subdivision ([3], [17], [20]), dans l'espace paramétrique  $(a, b)$ .

Dans [16] et [17], on introduit les espaces suivants :

$$\begin{aligned} S_{w, \infty} &= \{(a, b) : (10) \text{ asymptotiquement stable, } \forall \tau \in \mathbb{R}^+\}, \\ S_{s, \infty} &= \{(a, b) \in S_{w, \infty} : (11) \text{ n'a pas de racine } s = j\omega \\ &\quad \text{quand } \tau \rightarrow \infty\}, \\ S_\tau &= \{(a, b) : \exists \tau^* \in [0, +\infty[ \mid (10) \text{ asymptotiquement} \\ &\quad \text{stable, } \forall \tau \in [0, \tau^*[, (10) \text{ instable si } \tau \geq \tau^*\}. \end{aligned}$$

Ces trois espaces permettent de caractériser la stabilité asymptotique du système (10).

*Théorème V.1:* [17] Soit le système décrit par (10). Alors on a :

1.  $S_{s, \infty} = \{(a, b) : a > |b|\}$
2.  $S_{w, \infty} = \{(a, b) : a \geq |b|, a + b > 0\}$
3.  $S_\tau = \{(a, b) : b > |a|\}$

De plus, si  $(a, b) \in S_\tau$ , alors le système est asymptotiquement stable pour n'importe quel retard  $\tau$ ,  $0 \leq \tau < \tau^*$ , avec

$$\tau^* = \frac{\arccos(-\frac{a}{b})}{\sqrt{b^2 - a^2}} \quad (12)$$

Le théorème V.1 permet de représenter dans l'espace paramétrique  $(a, b)$  les zones de stabilité asymptotique. Les espaces  $S_{s,\infty}$  et  $S_{w,\infty}$  représentent les zones de stabilité,  $\forall \tau \in \mathbb{R}^+$  (hormis la distinction lorsque le retard tend vers  $+\infty$ ). L'espace  $S_\tau$  représente la zone de stabilité du système pour un retard  $\tau$  tel que  $0 \leq \tau < \tau^*$ , qui est délimité par la droite  $b = a$  et par la fonction implicite qui dépend du retard  $\tau$  donnée par (12).

Dans le plan  $(a, b)$ , on a donc la figure 7.

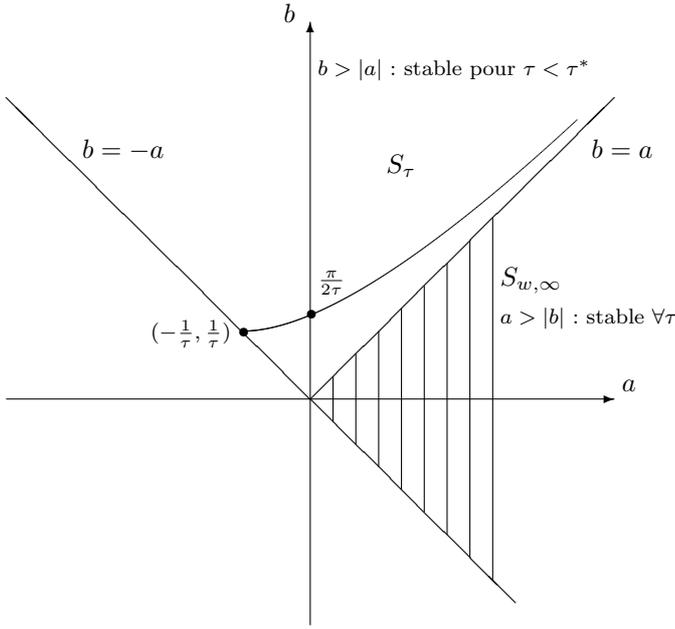


Figure 7. Zones de stabilité-instabilité du système à retard (10) dans le plan  $(a, b)$ .

Il est donc intéressant d'appliquer la méthode par intervalles pour analyser les différentes zones de stabilité décrite sur la figure 7. Pour se faire, il suffit de déterminer l'absence de solutions à l'équation caractéristique (11) dans le demi-plan droit complexe. On peut ainsi valider les résultats théoriques issus de la  $\mathcal{D}$ -subdivision.

On prend donc le système (10) à paramètres et retard incertains, de sorte que  $a \in [a_1, a_2]$ ,  $b \in [b_1, b_2]$  et  $\tau \in [\tau_1, \tau_2]$ . On sait alors que la stabilité de (10) équivaut à tester l'absence de racines à partie réelle positive de (11) (voir par exemple [1] et [10]).

Considérons donc les 3 cas suivants :

Cas 1.  $[a_1, a_2] \times [b_1, b_2] \times [\tau_1, \tau_2] = [-1, 1] \times [2, 3] \times [0, 0.4]$ ,

Cas 2.  $[a_1, a_2] \times [b_1, b_2] \times [\tau_1, \tau_2] = [-1, 1] \times [2, 3] \times [0, 0.5]$ ,

Cas 3.  $[a_1, a_2] \times [b_1, b_2] \times [\tau_1, \tau_2] = [0.5, 300] \times [-0.4, 0.4] \times [0, 0.5]$ .

Dans les cas 1 et 2, on se place dans la zone  $S_\tau$ , avec  $\tau^* \simeq 0.4352$ . Dans le dernier cas, on se place dans  $S_{w,\infty}$ . Sur la figure 8, on se place dans le cas 2, et la zone blanche garantit la stabilité du système malgré l'incertitude portant sur les paramètres et le retard. La zone gris-foncé correspond quant à elle à une zone « potentiellement instable », c'est à dire pour laquelle il existe au moins une valeur de

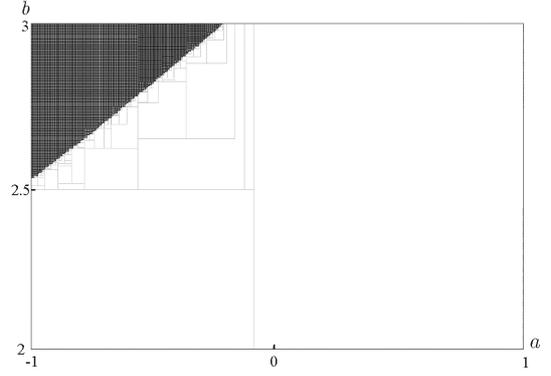


Figure 8. Cas 2, zones de stabilité dans le plan  $(a, b)$  du système (10) à paramètres et retard incertains. La zone gris-foncé est une zone d'instabilité dépendant du retard, et la zone blanche garantit la stabilité robuste du système.

$\tau \in [\tau_1, \tau_2]$  tel que le système (10) soit instable. En réalité, c'est la zone correspondant à  $\tau \in [\tau^*, \tau_2]$  qui crée cette instabilité (voir le théorème V-1). Au contraire, dans les cas 1 et 3, le système est stable, quel que soient  $a, b$  et  $\tau$  dans leurs respectifs intervalles d'incertitude, et aucune présence de zone instable ou « potentiellement instable » n'est détectée.

### B. Stabilité d'un système à retard de type neutre

De même que précédemment, on s'intéresse dans ce paragraphe à la stabilité robuste d'un système à retard de type neutre ([7], [17]).

Soit le système

$$\begin{cases} \dot{x}(t) - d\dot{x}(t - \tau) = -ax(t) - bx(t - \tau) \\ (a, b, d, \tau) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \end{cases} \quad (13)$$

avec une condition initiale appropriée. L'équation caractéristique de (13) est :

$$s(1 - de^{-s\tau}) + a + be^{-s\tau} = 0 \quad (14)$$

Si  $|d| > 1$ , alors (14) admet une infinité de racines à parties réelles positives. On a :

*Théorème V.2:* [17] Soit le système décrit par (13), et on suppose  $|d| < 1$ . Alors :

1.  $S_{w,\infty} = \{(a, b, d) : a \geq |b|, a + b > 0\}$
2.  $S_\tau = \{(a, b, d) : b > |a|\}$

De plus, si  $(a, b, d) \in S_\tau$ , alors le système est asymptotiquement stable pour n'importe quel retard  $\tau$ ,  $0 \leq \tau < \tau^*$ , avec

$$\tau^* = \sqrt{\frac{1 - d^2}{b^2 - a^2}} \arcsin \left( \frac{\sqrt{(b^2 - a^2)(1 - d^2)}}{b - ad} \right) \quad (15)$$

On retrouve donc pour ce système neutre une zone de stabilité tout à fait similaire à celle obtenue dans le cas retardé (voir paragraphe V-A). Par rapport à la figure 7, une des principales modifications est le changement du point critique  $(-\frac{1}{\tau}, \frac{1}{\tau})$  en  $(-\frac{1-d}{\tau}, \frac{1-d}{\tau})$ . Le calcul par intervalles se montre alors fort utile pour caractériser la stabilité robuste de (13), principalement lorsque l'on se place dans la zone paramétrique  $S_\tau(a, b, d)$ , où  $(a, b, d) \in [a_1, a_2] \times [b_1, b_2] \times [d_1, d_2]$ . Les résultats obtenus restent similaires à ceux du paragraphe V-A.

### C. Un cas critique

Soit le système instable

$$\dot{x}(t) = x(t) + u(t-1) \quad (16)$$

On cherche à le stabiliser par une commande du type

$$u(t) = \alpha x(t) + \beta x(t-1) \quad (17)$$

avec  $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}$ . Une telle loi de commande appliquée à (16) est stabilisante si par exemple nous prenons comme paramètres  $\alpha = -1.5$  et  $\beta = 0.4$ . Cependant, elle est stabilisante seulement si les paramètres  $(\alpha, \beta)$  sont soumis à des variations inférieures en valeur absolue à 0.2. Par appli-

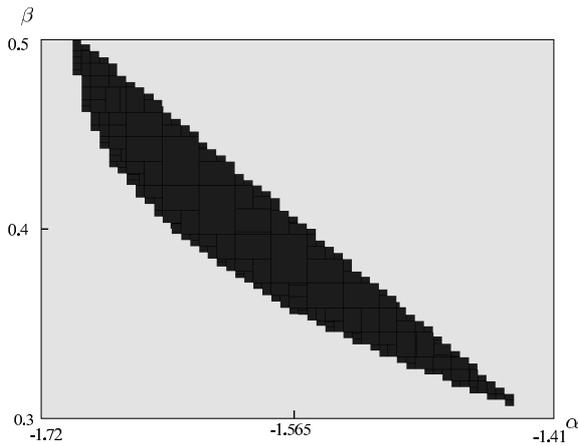


Figure 9. Zone de stabilité du système (16) en boucle fermée dans le plan  $(\alpha, \beta)$ ; la loi de commande stabilisante est déterminée par (17). La zone gris-foncée représente la zone de stabilité garantie du système en boucle fermée. Elle s'étend environ de  $\alpha \in [-1.72, -1.41]$  et  $\beta \in [0.3, 0.5]$ . La zone gris-clair représente celle d'instabilité potentielle.

cation au calcul par intervalles, on détermine facilement, dans le plan paramétrique  $(\alpha, \beta)$  la zone des valeurs de ces paramètres garantissant la stabilité du système (16) en boucle fermée par retour d'état avec (17). Sur la figure 9, les pavés gris-foncé caractérisent une zone de stabilité garantie du système en boucle fermée. Ces résultats sont obtenus pour des intervalles d'incertitude paramétrique fixés ( $\alpha \in [-1.72, -1.41]$  et  $\beta \in [0.3, 0.5]$ ) et pour une précision  $\varepsilon = 0.05$ .

## VI. CONCLUSION

On a traité dans ce papier quelques exemples d'application de l'analyse par intervalles et de la propagation de contraintes par intervalles à la localisation des racines d'un quasipolynôme, à la vérification de la stabilité, au calcul d'une norme  $H_\infty$  pour un système donné, et à la vérification de la stabilité robuste d'une famille de quasipolynômes dépendant de paramètres et d'un retard incertains dans des intervalles donnés. Les exemples incluent des systèmes à retard de type retardé et d'autres de type neutre.

Les exemples choisis sont volontairement simples, cette démarche étant originale dans le domaine des systèmes à retards. Ce sont des exemples pour lesquels il existe des méthodes analytiques d'étude, qui nous ont permis de valider les résultats obtenus avec Proj2D. L'exemple du paragraphe V-C est un peu plus complexe, et montre que l'ana-

lyse par intervalles s'applique indépendamment de l'existence d'une solution analytique. Il illustre aussi le fait que la présentation graphique des résultats de calcul proposée par Proj2D se prête particulièrement bien à la conception de systèmes de commande et à l'évaluation de leur robustesse.

## RÉFÉRENCES

- [1] R. Bellman et K. L. Cooke, *Differential-difference equations*, Academic Press, New York, 1963.
- [2] F. Benhamou, F. Goualard, L. Granvilliers et J. F. Puget, « Revising Hull and Box Consistency », *Proc. of the International Conference on Logic Programming*, Las Cruces, NM, 1999.
- [3] D. Brethé, *Stabilité et stabilisabilité des systèmes à retards*, Rapport de Diplôme d'Études Approfondies, Ecole Centrale de Nantes-Université de Nantes, Nantes, 1994.
- [4] J. G. Cleary, « Logical arithmetic », *Future Computing Systems*, vol. 2, n. 2, pp. 125-149, 1987.
- [5] E. Davis, « Constraint propagation with interval labels », *Artificial Intelligence*, vol. 32, n. 3, pp. 281-331, 1987.
- [6] C. A. Desoer et M. Vidyasagar, *Feedback Systems : Input-Output Properties*, Academic Press, New York, 1975.
- [7] K. Gu, V. L. Kharitonov et J. Chen, *Stability of time-delay systems*, Birkhauser, Boston, 2002.
- [8] N. Hohenbichler et J. Ackermann, « Computing stable regions in parameter spaces for a class of quasipolynomials », *4th IFAC Workshop TDS*, Rocquencourt, France, 2003.
- [9] L. Jaulin, M. Kieffer, O. Didrit et E. Walter, *Applied interval analysis*, Springer, London, 2001.
- [10] V. L. Kharitonov et A. P. Zhabko, « Robust Stability of Time-Delay Systems », *IEEE Trans. Aut. Control*, vol. 39, n. 12, pp. 2388-2397, 1994.
- [11] X. Li, C. E. de Souza et A. Trofino, « Delay-dependent robust stabilization of uncertain linear state-delayed systems via static output feedback », *Proc. IFAC Workshop LTDS*, Grenoble, France, 1998.
- [12] J. J. Loiseau, « Algebraic tools for the control and stabilization of time-delay systems », *Proc. IFAC Workshop LTDS*, Grenoble, France, 1998.
- [13] R. E. Moore, *Interval Analysis*, Prentice Hall, Englewood Cliffs, 1966.
- [14] R. E. Moore, « Methods and Applications of Interval Analysis », *SIAM*, Philadelphia, PA, 1979.
- [15] R. E. Moore et H. Ratschek, « Inclusion function and global optimization II », *Mathematical Programming*, vol. 41, n. 3, pp. 341-356, 1988.
- [16] S. I. Niculescu, *Sur la stabilité et la stabilisation des systèmes linéaires à états retardés*, Thèse de Doctorat, INPG, Grenoble, 1996.
- [17] S. I. Niculescu, *Delay Effects on Stability : A Robust Control Approach*, Springer, New York, 2001.
- [18] L. S. Pontryagin, « On the zeros of some elementary transcendental functions », *Izvestiya Akademii Nank SSSR*, vol. 6, pp. 115-131, 1942.
- [19] J. Santos, S. Mondié et V. L. Kharitonov, « Robust stability of time-delay systems and the finite inclusions theorem », *4th IFAC Workshop TDS*, Rocquencourt, France, 2003.
- [20] G. M. Schoen et H. P. Geering, « Stability condition for a delay differential system », *Int. J. Control*, vol. 58, n. 1, pp. 247-252, 1993.
- [21] T. Vyhlídal et P. Zitek, « Quasipolynomial mapping based root-finder for analysis of time delay systems », *4th IFAC Workshop TDS*, Rocquencourt, France, 2003.