QUIMPER : un langage de programmation pour le calcul ensembliste ; Application à l'automatique

Luc Jaulin et Gilles Chabert

DTN, ENSIETA,
2 rue François Verny, 29806 Brest Cédex 09
jaulinlu@ensieta.fr
chabergi@ensieta.fr

Mots clefs: analyse par intervalles, calcul ensembliste, commande robuste, estimation ensembliste, méthodes garanties, propagation de contraintes.

Résumé : QUIMPER est un langage de programmation dédié au calcul ensembliste. Il permet de programmer une grande classe d'algorithmes utilisant le calcul par intervalles avec une grande facilité. Après avoir introduit brièvement les notions fondamentales sur les contracteurs, cet article présente le langage QUIMPER et illustre son utilisation pour la résolution de problèmes ensemblistes issus de l'automatique.

1 Introduction

Le calcul par intervalles combiné à la propagation de contraintes [5], [4], [9], permet la résolution rapide d'une grande classe de problèmes ensemblistes. Les algorithmes qui en résultent font appel à des concepts relativement abstraits (comme les *contracteurs*) et sont difficiles à implémenter avec des langages classiques (comme le PASCAL ou le C++). QUIMPER [3] est un langage qui permet la programmation d'algorithmes ensemblistes avec une grande rapidité. Il est gratuit et peut être téléchargé librement sur `http://www.ibex-lib.org/`. Bien sûr, il est toujours nécessaire de maitriser les concepts abstraits issus du calcul ensembliste, mais il est désormais possible de se passer d'une grosse part de détails techniques liés à l'implémentation.

QUIMPER un langage interprété de haut niveau qui utilise principalement la notion de contracteur. En bref, on peut dire que QUIMPER tente d'être au calcul par intervalles ce que MATLAB est au calcul matriciel. Ce langage a été programmé en C++ à l'aide de la

librairie IBEX [2]. Le but de cet article est de montrer la résolution avec QUIMPER de quelques exemples issus de l'automatique. Cela permettra au lecteur de comprendre par l'exemple la philosophie du langage et comment programmer sous QUIMPER. Attention, QUIMPER n'est pas un solveur intervalle (comme c'est le cas pour REALPAVER [7]) censé résoudre une classe de problèmes bien particulière, mais un langage de programmation qui demande donc une certaine expérience avant de pouvoir résoudre un problème ensembliste bien précis.

Le paragraphe 2 rappelle la notion de contracteur, les opérations élémentaires sur les contracteurs et les propriétés qui en résultent. Dans les sections qui suivront, des exemples de la littérature seront traités. Le paragraphe 3 s'intéresse à un problème d'estimation à erreurs bornées sur le cas d'un circuit électrique. Un problème d'estimation dans le cas où les variables indépendantes sont incertaines sera considéré dans le paragraphe 4. Un exemple de preuve de stabilité robuste pour un système linéaire incertain sera traité dans le paragraphe 5. Enfin, le paragraphe 6 conclura cet article.

2 Contracteur

Un intervalle est un ensemble compact et connexe de $\mathbb{R}$. Un pavé $[\mathbf{x}]$ de $\mathbb{R}^n$ est un produit cartésien de n pavés. Notons $\mathbb{IR}^n$ l'ensemble des pavés de $\mathbb{R}^n$. Un pavé singleton formé de l'unique élément $\mathbf{x}$ sera noté sous la forme $\{\mathbf{x}\}$ et parfois même, par abus de langage, tout simplement $\mathbf{x}$. L'opérateur $\mathcal{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ est un *contracteur* si

$$\begin{array}{lll} \text{(i)} & \forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance)} \\ \text{(ii)} & (\mathbf{x} \in [\mathbf{x}], \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}) \Rightarrow \mathbf{x} \in \mathcal{C}([\mathbf{x}]) & \text{(consistance)} \\ \text{(iii)} & \mathcal{C}(\{\mathbf{x}\}) = \emptyset \Leftrightarrow (\exists \varepsilon > 0, \forall [\mathbf{x}] \subset B(\mathbf{x}, \varepsilon), \mathcal{C}([\mathbf{x}]) = \emptyset) & \text{(convergence)} \end{array}$$

où $B(\mathbf{x}, \varepsilon)$ désigne la boule de centre $\mathbf{x}$ et de rayon ε. Un pavé est dit *insensible* au contracteur C si $\mathcal{C}([\mathbf{x}]) = [\mathbf{x}]$. La propriété (i) nous assure que par l'action d'un contacteur, un pavé ne peut que se contracter. La propriété (ii) nous dit que tout pavé gardera, après contraction, tous ses points $\mathbf{x}$ insensibles à ce même contracteur. Enfin, (iii) nous assure entre autre l'ensemble des $\mathbf{x}$ sensibles à $\mathcal{C}$ est un ensemble ouvert, et donc que l'ensemble des $\mathbf{x}$ insensibles est fermé. L'*ensemble* (ou contrainte) associé à un contracteur $\mathcal{C}$ est l'ensemble formé par l'union des singletons insensibles à $\mathcal{C}$, c'est-à-dire,

$$\text{set}(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}\}. \tag{1}$$

Un contracteur peut être vu comme un moyen de représenter un sous-ensemble de $\mathbb{R}^n$. Il correspond à un algorithme implémentable en machine et contient toute l'information sur l'ensemble qu'il représente. Contrairement à ce même ensemble, il est très facile à manipuler. Les manupulations sur les ensembles (intersections, union, inversion, ...), se feront donc, sous QUIMPER, à travers leur contracteur.

Soit $\mathbb{S}$ un ensemble fermé défini par des égalités ou des inégalités, le calcul par intervalles propose un ensemble de méthodes qui permettent la construction de contracteurs efficaces $\mathcal{C}$ (c'est-à-dire polynomial en n) associés à $\mathbb{S}$, c'est-à-dire tels que $\text{set}(\mathcal{C}) = \mathbb{S}$. Nous avons les définitions suivantes

$\mathcal{C}$ est *monotone* si	$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}])$
$\mathcal{C}$ est *minimal* si	$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ \mathcal{C}([\mathbf{x}]) = [[\mathbf{x}] \cap \text{set}\,(C)]$
$\mathcal{C}$ est *idempotent* si	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}\left(\mathcal{C}([\mathbf{x}])\right) = \mathcal{C}([\mathbf{x}]),$

(2)

où $[\mathbb{A}]$ représente le plus petit pavé contenant $\mathbb{A}$. Un contracteur vérifie la propriété dite de *complétude* :

$$\mathcal{C}([\mathbf{x}]) \supset [\mathbf{x}] \cap \text{set}\,(\mathcal{C})\,. \tag{3}$$

Tous les contracteurs minimaux sont idempotents. L'ensemble $\text{set}(\mathcal{C})$ est toujours un ensemble fermé de $\mathbb{R}^n$. Cette propriété est une conséquence de la propriété de convergence.

Exemple 1. Un *contracteur de précision* est défini par

$$\mathcal{C}_\varepsilon\,([\mathbf{x}]) \begin{cases} = [\mathbf{x}] & \text{si } w\,([\mathbf{x}]) > \varepsilon \\ = \emptyset & \text{sinon} \end{cases}$$

où $\varepsilon > 0$. Ce contracteur est monotone et idempotent. Mais, il n'est pas minimal. De plus $\text{set}(\mathcal{C}_\varepsilon) = \emptyset$. Notons que si nous avions $\varepsilon = 0$, alors l'opérateur $\mathcal{C}_\varepsilon$ ne serait plus un contrcateur car il ne vérifirait plus la propriété de convergence.

On définit les opérations suivantes

intersection	$(\mathcal{C}_1 \cap \mathcal{C}_2)\,([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1\,([\mathbf{x}]) \cap \mathcal{C}_2\,([\mathbf{x}])$
union	$(\mathcal{C}_1 \sqcup \mathcal{C}_2)\,([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1\,([\mathbf{x}]) \sqcup \mathcal{C}_2\,([\mathbf{x}])$
composition	$(\mathcal{C}_1 \circ \mathcal{C}_2)\,([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1\,(\mathcal{C}_2\,([\mathbf{x}]))$
répétition	$\mathcal{C}_1^\infty \stackrel{\text{def}}{=} \mathcal{C}_1 \circ \mathcal{C}_1 \circ \mathcal{C}_1 \circ \ldots$
intersection répétée	$\mathcal{C}_1 \sqcap \mathcal{C}_2 = (\mathcal{C}_1 \cap \mathcal{C}_2)^\infty\,.$

(4)

où $\sqcup$ désigne le plus petit pavé contenant l'union. On peut noter que la composition n'est pas commutative (i.e., $\mathcal{C}_1 \circ \mathcal{C}_2 \neq \mathcal{C}_2 \circ \mathcal{C}_1$) contrairement à tous les autres $\cap, \sqcup, \sqcap$.

On peut aisément vérifier que les opérateurs définis ci-dessus forment effectivement des contracteurs et qu'ils satisfont donc les propriétés de contractance, de consistance et de convergence. Notons aussi que, du fait que le complémentaire d'un ensemble fermé ne soit pas fermé (sauf exception), la notion de contracteur complémentaire n'a pas de sens. On définit l'inclusion entre contracteurs comme suit

$$\mathcal{C}_1 \subset \mathcal{C}_2 \Leftrightarrow \forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_1 ([\mathbf{x}]) \subset \mathcal{C}_2 ([\mathbf{x}]) . \tag{5}$$

On peut alors énoncer le résultat suivant.

Théorème: L'ensemble des contracteurs idempotents et monotones muni de l'inclusion forme un treillis complet. Ses deux loi internes (inf et sup) sont données par $\mathcal{C}_1 \sqcup \mathcal{C}_2$ et $\mathcal{C}_1 \sqcap \mathcal{C}_2$. Son plus petit élément $\mathcal{C}^{\perp}$ est le contracteur vide et son plus grand élément $\mathcal{C}^{\top}$ est l'identité :

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \ \mathcal{C}^{\perp} ([\mathbf{x}]) = \emptyset \text{ et } \mathcal{C}^{\top} ([\mathbf{x}]) = [\mathbf{x}]. \tag{6}$$

Malheureusement, ce treillis n'est pas distributif. Mais on a tout de même une propriété qu'on pourrait qualifier de sous-distributivité

$$(\mathcal{C}_1 \sqcap (\mathcal{C}_2 \sqcup \mathcal{C}_3)) \supset (\mathcal{C}_1 \sqcap \mathcal{C}_2) \sqcup (\mathcal{C}_1 \sqcap \mathcal{C}_3) . \tag{7}$$

Voici quelques propriétés sur les contracteurs, que nous ne démontrerons pas.

$$\begin{array}{l} \text{set} (\mathcal{C}_1 \sqcap \mathcal{C}_2) = \text{set} (\mathcal{C}_1 \cap \mathcal{C}_2) = \text{set} (\mathcal{C}_1) \cap \text{set} (\mathcal{C}_2) = \text{set} (\mathcal{C}_1 \circ \mathcal{C}_2) \\ \text{set} (\mathcal{C}_1 \sqcup \mathcal{C}_2) = \text{set} (\mathcal{C}_1) \cup \text{set} (\mathcal{C}_2) \\ \text{set} (\mathcal{C}_1^{\infty}) = \text{set} (\mathcal{C}_1) \\ \mathcal{C}_1 \circ \mathcal{C}_2 \subset \mathcal{C}_1 \sqcap \mathcal{C}_2 \text{ (si } \mathcal{C}_1 \text{ est monotone)} \\ \mathcal{C}_1 \sqcap (\mathcal{C}_2 \sqcup \mathcal{C}_3) \supset (\mathcal{C}_1 \sqcap \mathcal{C}_2) \sqcup (\mathcal{C}_1 \sqcap \mathcal{C}_3) . \end{array} \tag{8}$$

Soit un contracteur $\mathcal{C} ([\mathbf{x}], [\mathbf{y}])$, où ici $[\mathbf{x}] \in \mathbb{R}^n, [\mathbf{y}] \in \mathbb{R}^p$. On définit les deux opérateurs sur $[\mathbf{x}]$ suivants :

$$\left\{ \begin{array}{lcl} \mathcal{C}^{\cup [\mathbf{y}]} ([\mathbf{x}]) & = & \bigsqcup_{\mathbf{y} \in [\mathbf{y}]} \pi_{\mathbf{x}} (\mathcal{C} ([\mathbf{x}], \mathbf{y})) \\ \mathcal{C}^{\cap [\mathbf{y}]} ([\mathbf{x}]) & = & \bigcap_{\mathbf{y} \in [\mathbf{y}]} \pi_{\mathbf{x}} (\mathcal{C} ([\mathbf{x}], \mathbf{y})) \end{array} \right. \tag{9}$$

où l'opérateur $\pi_x ([\mathbf{x}], [\mathbf{y}])$ représente la projection $[\mathbf{x}]$ du pavé $[\mathbf{x}] \times [\mathbf{y}]$. Les deux opérateurs que l'on peut noter

$$\begin{array}{lcl} \mathcal{C} ([\mathbf{x}], [\mathbf{y}]) & \rightarrow & \mathcal{C}^{\cup [\mathbf{y}]} ([\mathbf{x}]) \\ \mathcal{C} ([\mathbf{x}], [\mathbf{y}]) & \rightarrow & \mathcal{C}^{\cap [\mathbf{y}]} ([\mathbf{x}]) \end{array} \tag{10}$$

qui permettent de frabriquer les deux contracteurs $\mathcal{C}^{\cup[\mathbf{y}]}([\mathbf{x}])$ et $\mathcal{C}^{\cap[\mathbf{y}]}([\mathbf{x}])$ à partir du contracteur $\mathcal{C}([\mathbf{x}],[\mathbf{y}])$ sont respectivement appelés *union projection* et *intersection projection*. On a les propriétés suivantes

$$\begin{array}{ll} \text{(i)} & \mathcal{C}^{\cap} \subset \mathcal{C}^{\cup}, \\ \text{(ii)} & \mathcal{C}^{\cup[\mathbf{y}]} \text{ et } \mathcal{C}^{\cap[\mathbf{y}]} \text{ sont des contracteurs (\textit{i.e.} contractance, consistance, convergence)} \\ \text{(iii)} & \text{set}\left(\mathcal{C}^{\cup[\mathbf{y}]}\right) = \{\mathbf{x}, \exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \text{set}(\mathcal{C})\} \\ \text{(vii)} & \text{set}\left(\mathcal{C}^{\cap[\mathbf{y}]}\right) = \{\mathbf{x}, \forall \mathbf{y} \in [\mathbf{y}], (\mathbf{x},\mathbf{y}) \in \text{set}(\mathcal{C})\} \end{array} \tag{11}$$

L'intérêt de ces deux contracteurs est donc évident pour le calcul d'ensembles définis par des quantificateurs, comme c'est le cas pour la projection d'ensembles. Une collection de contracteurs $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ est dite *complémentaire* si

$$\text{set}(\mathcal{C}_1) \cap \cdots \cap \text{set}(\mathcal{C}_m) = \emptyset. \tag{12}$$

QUIMPER est un langage interprété qui permet de fabriquer et manipuler aisément des contracteurs afin de les utiliser pour résoudre un problème ensembliste. Un programme QUIMPER décrit tout simplement une collection complémentaire de contracteurs. Lorsqu'un programme QUIMPER est exécuté, il construit tout d'abord la collection de contracteurs puis il lance un paveur. Un paveur est un simple algorithme de bisection qui appelle tous les contracteurs disponibles sur tous les pavés courants. Seuls sont bissectés les pavés qui n'arrivent plus à être contractés par aucun des contracteurs. Le paveur balaye donc tout l'espace de recherche $\mathbb{R}^n$ et range les zones contractées dans des ensembles appelés *sous-pavages*. Ainsi, à chaque contracteur correspond un unique sous-pavage. Puisque les contracteurs sont complémentaires, le paveur termine.

3 Problème du circuit électrique

Le premier exemple que nous allons considérer concerne le circuit de la figure 1. Ce dernier est composé d'une pile délivrant une tension E et de deux résistances R_1 et R_2. Les tensions aux bornes de ces deux résistances sont notées U_1 et U_2. La puissance et le courant délivrés par la pile sont notés P et I. Supposons que dans notre circuit, différentes mesures aient été prélevées. Dans un contexte ensembliste, chaque mesure se traduit par une relation d'appartenance. On obtient par exemple

$$E \in [23V, 26V]; I \in [4A, 8A]; U_1 \in [10V, 11V]; U_2 \in [14V, 17V]; P \in [124W, 130W];$$

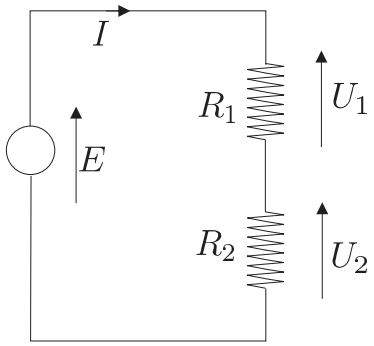

Figure 1: Circuit électrique comprenant une pile et deux résistances en séries

Nous ne savons rien sur les valeurs des résistances à part qu'elles sont toutes les deux positives. Cette information se traduit à nouveau par les relations d'appartenance $R_1 \in [0, \infty[$ et $R_2 \in [0, \infty[$. De plus, les lois de la physique nous donnent les contraintes :

$$\begin{array}{lll} \text{(i) } P = EI, & \text{(ii) } E = (R_1 + R_2)\, I, & \text{(iii) } U_1 = R_1 I, \\ \text{(iv) } U_2 = R_2 I, & \text{(v) } E = U_1 + U_2. & \end{array} \tag{13}$$

Il est clair que la contrainte (ii) est redondante car elle peut être obtenue à partir des trois dernières. De telles redondances sont difficiles à détecter pour des systèmes plus complexes et perturbent souvent les méthodes de résolutions formelles ou numériques classiques. En revanche, l'existence de contraintes redondantes a plutôt tendance à rendre plus performantes les méthodes de propagation. L'ensemble des solutions de notre problème est

$$\mathbb{S} = \left\{ \begin{pmatrix} E \\ R_1 \\ R_2 \\ I \\ U_1 \\ U_2 \\ P \end{pmatrix} \in \begin{pmatrix} [23, 26] \\ [0, \infty[\\ [0, \infty[\\ [4, 8] \\ [10, 11] \\ [14, 17] \\ [124, 130]; \end{pmatrix} \text{ tels que } \left\{ \begin{array}{l} P = EI \\ E = (R_1 + R_2)\, I \\ U_1 = R_1 I \\ U_2 = R_2 I \\ E = U_1 + U_2 \end{array} \right\} \right. \tag{14}$$

Pour le programme source suivant,

```
variables
 E in [23 ,26];
```

```
 I in [4,8];
 U1 in [10,11];
 U2 in [14 ,17];
 P in [124,130];
 R1 in [0 ,1e08 ];
 R2 in [0 ,1e08 ];
contractor_list L
 P=E*I;
 E=(R1+R2)*I;
 U1=R1*I;
 U2=R2*I;
 E=U1+U2;
end
contractor C=Compose(L)
contractor epsilon=precision(1)
```

QUIMPER fabrique tout d'abord une liste de 5 contracteurs qui sont rangés dans une liste de contracteurs. Chaque contracteur est lié à une contrainte, ce qui signifie par exemple que si $\mathcal{C}_i$ est le $i$ième contracteur de la liste, on a

$$\begin{aligned} \text{set}\left(\mathcal{C}_1\right) &= \left\{(E, R_1, R_2, I, U_1, U_2, P)\,,\ P = EI\right\} && (15)\\ \text{set}\left(\mathcal{C}_2\right) &= \left\{(E, R_1, R_2, I, U_1, U_2, P)\,,\ E = (R_1 + R_2)\, I\right\}, \ldots && (16) \end{aligned}$$

Ensuite, on fabrique le contracteur $\mathcal{C} = \mathcal{C}_1 \circ \cdots \circ \mathcal{C}_5$ et enfin, un contracteur de précision `epsilon` est défini afin d'avoir la propriété de complémentarité demandée par QUIMPER pour que le programme termine. En 10^{-3} seconde QUIMPER retourne le pavé

$$[24; 26] \times [1.846; 2.307] \times [2.584; 3.355] \times [4.769; 5.417] \times [10; 11] \times [14; 16] \times [124; 130]\,. \quad (17)$$

Nous sommes alors certains que ce pavé enferme $\mathbb{S}$. Ce résultat se traduit par les relations d'appartenance

$$\begin{array}{ll} E \in [24; 26]\,, & R_1 \in [1.846; 2.307]\,,\\ R_2 \in [2.584; 3.355], & I \in [4.769; 5.417]\,,\\ U_1 \in [10; 11]\,, & U_2 \in [14; 16]\,,\\ P \in [124; 130]\,. & \end{array} \quad (18)$$

Ainsi QUIMPER nous a permis d'obtenir avec une assez bonne précision les valeurs des résistances R_1 et R_2. De plus, il a généré des intervalles d'appartenance pour les variables

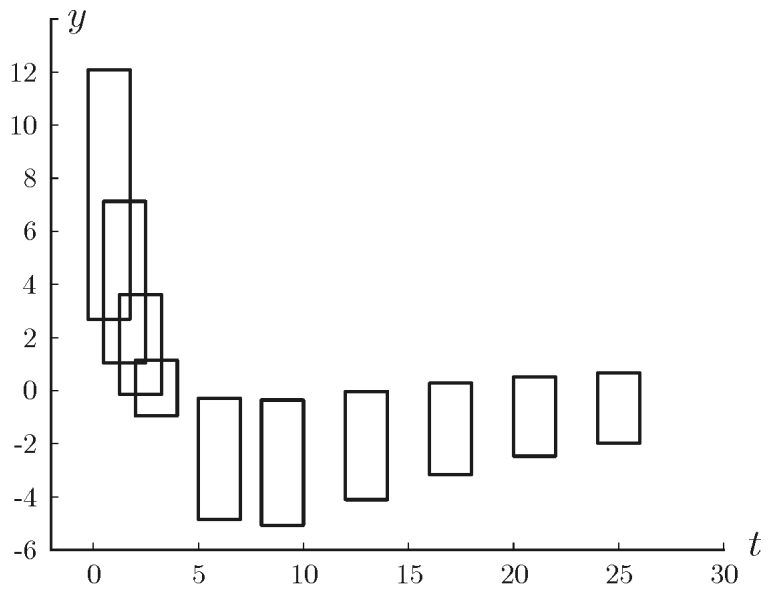

Figure 2: Tous les vecteurs (p_1, p_2) tels que le modèle traverse chacune des barres erreurs sont considérés comme acceptables

I et U_2, pourtant accessibles à la mesure, plus précis que les intervalles *a priori*. En revanche, une meilleure précision n'a pu être obtenue pour U_1 et P.

4 Problème des exponentielles

Ce problème est pris dans le livre [8] page 165. On considère le modèle donné par

$$y_m(\mathbf{p}, t) = 20\exp(-p_1 t) - 8\exp(-p_2 t). \tag{19}$$

Dix mesures ont été faites à différents instants. Les pavés représentés sur la figure 2 montrent une incertitude à la fois sur la mesure y_i mais aussi sur l'instant de la mesure t_i, $i \in \{1, \ldots, 10\}$.

Les intervalles pour t_i et y_i sont donnés ci-dessous.

i	$[t_i]$	$[y_i]$
1	$[-0.25, 1.75]$	$[2.7, 12.1]$
2	$[0.5, 2.5]$	$[1.04, 7.14]$
3	$[1.25, 3.25]$	$[-0.13, 3.61]$
4	$[2, 4]$	$[-0.95, 1.15]$
5	$[5, 7]$	$[-4.85, -0.29]$
6	$[8, 10]$	$[-5.06, -0.36]$
7	$[12, 14]$	$[-4.1, -0.04]$
8	$[16, 18]$	$[-3.16, 0.3]$
9	$[20, 22]$	$[-2.5, 0.51]$
10	$[24, 26]$	$[-2, 0.67]$

L'ensemble des paramètres acceptables est défini par

$$\begin{array}{rcl} \mathbb{S} & = & \{\mathbf{p} \in \mathbb{R}^2 \mid \exists t_1 \in [t_1], \ldots, \exists t_{10} \in [t_{10}] \mid y_m(\mathbf{p}, t_1) \in [y_1], \ldots, y_m(\mathbf{p}, t_{10}) \in [y_{10}]\} \\ & = & \bigcap_{i \in \{1,\ldots,10\}} \underbrace{\left\{\mathbf{p} \in \mathbb{R}^2 \mid \exists t_i \in [t_i] \mid y_m(\mathbf{p}, t_i) \in [y_i]\right\}}_{\mathbb{S}_i}. \end{array} \tag{20}$$

Soient $\mathcal{C}(\mathbf{p}, t)$ et $\bar{\mathcal{C}}(\mathbf{p}, t)$ deux contracteurs tels que

$$\text{set}\left(\mathcal{C}(\mathbf{p}, t)\right) = \{(\mathbf{p}, t), y_m(\mathbf{p}, t) \in [y]\} \tag{21}$$

$$\text{set}\left(\bar{\mathcal{C}}(\mathbf{p}, t)\right) = \{(\mathbf{p}, t), y_m(\mathbf{p}, t) \notin \text{int}([y])\}. \tag{22}$$

où int$([y])$ est l'intérieur de $[y]$. On supposera que la fonction $y_m(\mathbf{p}, t)$ est continue afin de s'assurer que les deux ensembles ci-dessus sont bien fermés (sinon, la notion même de contracteur n'a pas de sens). On peut dériver les deux contracteurs sur $[\mathbf{p}]$ suivants

$$\mathcal{C}^{\cup[t]}([\mathbf{p}]) = \left[\bigcup_{t \in [t]} \pi_x\left(\mathcal{C}([\mathbf{p}], t)\right)\right] \tag{23}$$

$$\bar{\mathcal{C}}^{\cap[t]}([\mathbf{p}]) = \left[\bigcap_{t \in [t]} \pi_x\left(\bar{\mathcal{C}}([\mathbf{p}], t)\right)\right] \tag{24}$$

où $[t]$ est vu ici comme un paramètre des contracteurs. On a

$$\begin{array}{rcl} \text{set}\left(\mathcal{C}^{\cup[t]}\right) & = & \{\mathbf{p}, \exists t \in [t], \mathcal{C}(\mathbf{p}, t)\} = \{\mathbf{p}, \exists t \in [t], y_m(\mathbf{p}, t) \in [y]\} \\ \text{set}\left(\bar{\mathcal{C}}^{\cap[t]}\right) & = & \left\{\mathbf{p}, \forall t \in [t], \bar{\mathcal{C}}(\mathbf{p}, t)]\right\} = \{\mathbf{p}, \forall t \in [t], y_m(\mathbf{p}, t) \notin \text{int}([y])\}. \end{array}$$

Définissons les deux contracteurs suivants

$$\mathcal{C}\left([\mathbf{p}]\right) = \bigcap_{i\in\{1,\ldots,10\}} \mathcal{C}^{\cup[t_i]}\left([\mathbf{p}],t_i\right) \tag{25}$$

$$\bar{\mathcal{C}}\left([\mathbf{p}]\right) = \bigsqcup_{i\in\{1,\ldots,10\}} \bar{\mathcal{C}}^{\cap[t_i]}\left([\mathbf{p}],t_i\right). \tag{26}$$

D'après la définition de $\mathbb{S}$ (voir (20), il est clair que $\text{set}(\mathcal{C}) = \mathbb{S}$. De plus, puisque le complémentaire de $\mathbb{S}$ est donné par

$$\bar{\mathbb{S}} = \bigcup_{i\in\{1,\ldots,10\}} \underbrace{\left\{\mathbf{p}\in\mathbb{R}^2 \mid \forall t_i\in[t_i] \mid y_m(\mathbf{p},t_i)\notin[y_i]\right\}}_{\bar{\mathbb{S}}_i}, \tag{27}$$

on a $\text{set}\big(\bar{\mathcal{C}}\big) \subset \bar{\mathbb{S}}$. Le programme QUIMPER ci-dessous fabrique trois contracteurs : le contracteur $\mathcal{C}$ pour $\mathbb{S}$, le contracteur $\mathcal{C}$ pour son complémentaire et un contracteur de précision $\mathcal{C}_\varepsilon$. Ces trois contracteurs sont complémentaires. Nous obtenons en 2.7 secondes les sous-pavages représentés sur la figure 3.

```
constant
  Y[10] = [[2.7,12.1]; [1.04,7.14]; [-0.13,3.61]; [-0.95,1.15]; [-4.85,-0.29];
           [-5.06,-0.36]; [-4.1,-0.04]; [-3.16,0.3]; [-2.5,0.51]; [-2,0.67]];
variables
  p1 in [0,1.2]; p2 in [0,0.5];
parameters
  t[10] in [[-0.25,1.75]; [0.5,2.5]; [1.25,3.25]; [2,4];
            [5,7]; [8,10]; [12,14]; [16,18]; [20,22]; [24,26]];
function z=f(p1,p2,t)
  z=20*exp(-p1*t)-8*exp(-p2*t);
end
contractor outer=intersection (i=1:10,
       union_projection(t[i],f(p1,p2,t[i]) in Y[i]));
contractor inner=union (i=1:10,
       intersection_projection(t[i],f(p1,p2,t[i]) notin Y[i]));
contractor epsilon=precision(0.01).
```

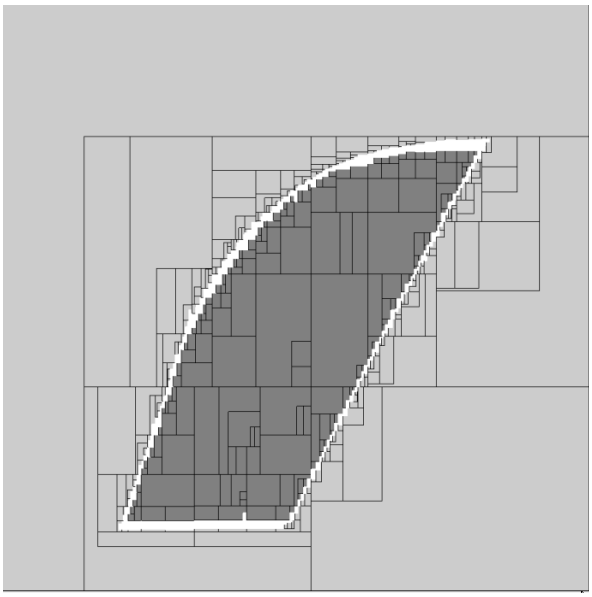

Figure 3: Ensemble des paramètres acceptables

5 Problème de stabilité robuste

Dans ce paragraphe, nous allons utiliser QUIMPER pour montrer la stabilité robuste d'un système linéaire. L'adjectif *robuste*, signifie ici "dans tous les cas possibles, même le pire". Nous allons ici montrer comment résoudre efficacement et simplement des problèmes pourtant considérés comme difficiles (voir [1]). Considérons, à titre d'exemple, le problème d'une moto roulant à une vitesse constante et faible de 1m/s (voir [6] page 313 pour plus de détails sur ce problème). L'entrée du système est l'angle θ du guidon et la sortie est l'angle de roulis ϕ de la moto. La relation entrée-sortie du système est :

$$\phi(s) = \frac{1}{s^2 - \alpha_1}\theta(s). \tag{28}$$

Vu la très faible vitesse de la moto, l'effet gyroscopique de la roue avant (qui maintient la moto stable à vive allure) a été négligé et le système est donc instable. On se propose de commander l'angle du guidon de la moto par un régulateur du premier ordre dont la relation entrée-sortie, donnée dans le domaine de Laplace, est

$$\theta(s) = \frac{\alpha_2 + \alpha_3 s}{\tau s + 1}\left(\phi_d(s) - \phi_m(s)\right), \tag{29}$$

où ϕ_d est l'angle de roulis désiré et ϕ_m est l'angle de roulis mesuré. Notons que comme le capteur n'est pas parfait, l'angle mesuré ϕ_m, c'est-à-dire celui qui apparaît en sortie du

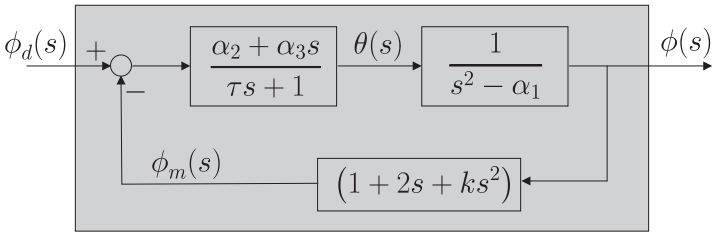

Figure 4: Moto stabilisée par un régulateur du premier ordre

capteur, n'est pas exactement l'angle réel ϕ de la moto. La relation différentielle entre ces deux quantités est donnée par :

$$\phi_m(s) = \left(1 + 2s + ks^2\right) \phi(s). \tag{30}$$

L'ensemble de ces équations est représenté par le schéma de la figure 4. Supposons que les valeurs pour les coefficients de la moto, du régulateur et du capteur satisfont

$$\alpha_1 \in [8.8; 9.2], \alpha_2 \in [2.8; 3.2], \alpha_3 \in [0.8; 1.2], \tau \in [1.8; 2.2], k \in [-3.2; -2.8] \tag{31}$$

La relation entrée-sortie du système régulé est

$$\phi(s) = \frac{\alpha_2 + \alpha_3 s}{\left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right)} \phi_d(s). \tag{32}$$

Son polynôme caractéristique est

$$\left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right) = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \tag{33}$$

avec $a_3 = \tau + \alpha_3 k$, $a_2 = \alpha_2 k + 2\alpha_3 + 1$, $a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2$ et $a_0 = -\alpha_1 + \alpha_2$. La table de Routh associée est donc :

a_3	a_1
a_2	a_0
$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0
a_0	0

(34)

Ainsi, le système est stable si $a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}$ et a_0 sont de même signe. Or, pour quatre nombres b_1, b_2, b_3 et b_4 on a l'équivalence

$$\begin{array}{c} b_1,\ b_2,\ b_3 \text{ et } b_4 \text{ sont de même signes} \\ \Leftrightarrow \quad \begin{cases} \min(b_1, b_2, b_3, b_4) > 0 \text{ ou} \\ \max(b_1, b_2, b_3, b_4) < 0. \end{cases} \end{array} \tag{35}$$

Le problème de stabilité robuste consiste à montrer la stabilité du système pour toutes les valeurs possibles des coefficients. La condition de stabilité robuste revient donc à dire que la proposition

$$
\begin{array}{l}
\exists \alpha_1 \in [8.8; 9.2], \exists \alpha_2 \in [2.8; 3.2], \exists \alpha_3 \in [0.8; 1.2], \exists \tau \in [1.8; 2.2], \exists k \in [-3.2; -2.8], \\
a_3 = \tau + \alpha_3 k \ ; \ a_2 = \alpha_2 k + 2\alpha_3 + 1 \ ; \ a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2, \\
a_0 = -\alpha_1 + \alpha_2 \ ; \ b = \frac{a_2 a_1 - a_3 a_0}{a_2}; \\
\left\{ \begin{array}{rcl} \min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) & \leq & 0 \text{ et} \\ \max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) & \geq & 0 \end{array} \right.
\end{array}
$$

est toujours fausse. QUIMPER prouve en moins de 10^{-3} seconde la stabilité robuste du système bouclé. Le programme source pour QUIMPER est donné ci-dessous

```
variables
 alpha1 in [8.8,9.2];
 alpha2 in [2.8,3.2];
 alpha3 in [0.8,1.2];
 tau in [1.8,2.2];
 k in [-3.2,-2.8];
 r in [-1e08,0];
 b1 in [-1e08,0];
 b2 in [0,-1e08];
 a3,a2,a1,a0,b;
contractor_list L
 a3=tau+alpha3*k;
 a2=alpha2*k+2*alpha3+1;
 a1=alpha3-alpha1*tau+2*alpha2;
 a0=alpha2-alpha1;
 b1=min(a3,a2,(a2*a1-a3*a0)/a2,a0);
 b2=max(a3,a2,(a2*a1-a3*a0)/a2,a0);
end
contractor C=Compose(L)
```

6 Conclusion

Dans cet article, nous avons présenté un nouveau langage appelé QUIMPER qui permet de résoudre de façon élégante un grand nombre de problèmes ensemblistes. Ce langage utilise abondamment la notion de contracteur afin de pouvoir manipuler efficacement des ensembles définis par des inégalités ou des égalités non-linéaires. Afin d'illustrer comment utiliser un tel langage, quelques exemples issus de l'automatique ont été traités. Bien que le langage soit interprété, les temps de calcul sont largement compétitifs par rapport à ceux que l'on pourrait obtenir avec un langage compilé.

References

[1] B. R. Barmish. *New Tools for Robustness of Linear Systems*. MacMillan, New York, NY, 1994.

[2] G. Chabert. *IBEX library, available at* , `http://www.ibex-lib.org/`. ENSIETA-INRIA, 2007.

[3] G. Chabert and L. Jaulin. *QUIMPER: QUick Interval Modeling and Programming in a bounded-ERror context, available at* , `http://www.ibex-lib.org/`. ENSIETA, 2008.

[4] J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.

[5] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32(3):281–331, 1987.

[6] R.C. Dorf and R.H. Bishop. *Modern Control Systems*. Electrical and Computer Engineering: Control Engineering. Addison-Wesley Publishing Compagny, 1995.

[7] L. Granvilliers. *RealPaver, available at* , `www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver/`. IRIN, University of Nantes, 2002.

[8] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[9] D.J. Sam-Haroud and B. Faltings. Consistency techniques for continuous constraints. *Constraints*, 1(1-2):85–118, 1996.