



HAL
open science

Quelques applications de la propagation de contraintes sur les domaines continus en automatique

Luc Jaulin, Marc Christie, Laurent Granvilliers

► **To cite this version:**

Luc Jaulin, Marc Christie, Laurent Granvilliers. Quelques applications de la propagation de contraintes sur les domaines continus en automatique. 11e Journées Francophones de Programmation Logique et Programmation par Contraintes (JFPLC 2002), May 2002, Nice, France. hal-00845652

HAL Id: hal-00845652

<https://hal.science/hal-00845652>

Submitted on 17 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Quelques applications de la propagation de contraintes sur les domaines continus en automatique

Luc Jaulin* — Marc Christie** — Laurent Granvilliers**

* *Laboratoire d'Ingénierie des Systèmes Automatisés,
62 avenue Notre Dame du Lac,
F-49000 Angers
jaulin@univ-angers.fr,
<http://www.istia.univ-angers.fr/~jaulin>*

** *IRIN, Université de Nantes,
B.P. 92208 - F-44322 Nantes Cedex 3
{christie, granvilliers}@irin.univ-nantes.fr*

RÉSUMÉ. De nombreux problèmes d'automatique sont des problèmes de satisfaction de contraintes sur des domaines continus. L'analyse par intervalles combinée à des techniques de propagation de contraintes permet de résoudre efficacement ce type de problèmes en garantissant la complétude. L'approche sera utilisée pour la résolution de problèmes d'estimation ensembliste et de commande robuste.

ABSTRACT. A large class of problems arising in control theory can be cast into constraint satisfaction problems over continuous domains. Interval analysis combined with constraint propagation techniques makes it possible to solve them in an efficient and guaranteed way. The approach will be used for set estimation and robust control problems.

MOTS-CLÉS : analyse par intervalles, calcul ensembliste, commande robuste, estimation ensembliste, méthodes complètes, propagation de contraintes.

KEYWORDS: constraint propagation, interval analysis, reliable methods, robust control, set computation, set estimation.

1. Introduction

De nombreux problèmes d'automatique peuvent se ramener à la résolution de systèmes de n équations à n inconnues, où à la minimisation d'un critère. Mais cela n'est pas toujours le cas : il existe une multitude de problèmes, dits de *satisfaction de contraintes*, où les variables impliquées sont reliées par un nombre arbitraire de contraintes de tout type. Comme les variables impliquées proviennent souvent de systèmes physiques (comme par exemple la masse, la vitesse, l'énergie, la température, la tension, le courant, la résistance, . . .), elles prennent généralement leur valeur dans \mathbb{R} et non pas dans un ensemble discret, nous parlerons donc de *problèmes de satisfaction de contraintes sur des domaines continus*, ou plus simplement de *CCSP* (Continuous Constraint Satisfaction Problems) (voir par exemple, [DAV 87], [CLE 87], [BEN 97], [SAM 96]). Ces problèmes sont considérés comme mal posés dans la communauté des automaticiens car ils peuvent admettre une infinité de solutions comme ils peuvent n'en posséder aucune. De ce fait, les outils existants (comme ceux proposés par la toolbox *control* de Matlab par exemple) ne peuvent être utilisés sans une transformation préalable, plus ou moins approximative, du problème particulier que l'on cherche à résoudre.

Les méthodes de propagation de contraintes [MAC 77] semblent adaptées à la résolution des problèmes satisfaction de contraintes car elles ne transforment pas le problème en un autre dans le seul but d'utiliser les outils disponibles. Au contraire, ces méthodes traitent directement le problème avec une méthode déductive, proche du bon sens dont dispose l'ingénieur. De plus, ces méthodes sont efficaces et fournissent des résultats garantis (c'est-à-dire qu'aucune solution ne peut être perdue). Enfin, commencent à apparaître des solveurs gratuits permettant la résolution d'une grande classe de problèmes de satisfaction de contraintes sur les domaines continus. Des exemples de tels solveurs sont *SPACE SOLVER* [LOT 00], [LOT], *AQCS* [RAT 00] (Approximate Quantified Constraint Solving) ou *REALPAVER* [GRA 02], [GRA 01].

Dans cet article, nous utiliserons le solveur *REALPAVER*, pour résoudre quelques problèmes difficiles d'automatique. Comme *Numerica* [van 97], ce solveur possède pour entrée un fichier texte définissant le CCSP, c'est-à-dire que ce fichier contient,

- 1) une déclaration des n variables réelles x_1, \dots, x_n du problème,
- 2) un ensemble de n intervalles $[x_1], \dots, [x_n]$ contenant toutes les valeurs a priori possible pour ces variables,
- 3) l'ensemble de toutes les contraintes reliant ces variables.

REALPAVER génère un ensemble de boîtes de \mathbb{R}^n contenant tous les n -uplets (x_1, \dots, x_n) de $[x_1] \times \dots \times [x_n]$ qui satisfont toutes les contraintes. Les méthodes et outils utilisés par *REALPAVER* sont l'analyse par intervalles [MOO 79] et la propagation de contraintes utilisant des consistances plus ou moins fortes (voir [LHO 93]).

Les problèmes auxquels nous allons nous intéresser sont l'estimation ensembliste au paragraphe 2 et la commande robuste au paragraphe 3. Ces problèmes sont classiques en automatique et ne peuvent être résolus avec les outils disponibles (ceux de

Matlab ou Scilab, par exemple) que pour des cas particuliers. Nous montrerons qu'ils appartiennent à la classe des CCSP et nous proposerons une résolution à l'aide du solveur REALPAVER.

2. Estimation ensembliste

Considérons un système dont le comportement est décrit par un ensemble d'équations, ou plus généralement de contraintes. Certaines de ces variables impliquées dans ce système peuvent être mesurées, avec une certaine erreur, d'autres non. Certaines de ces variables nous intéressent, d'autres pas. Souvent, les variables qui nous intéressent ne correspondent pas à celles qui sont mesurées. L'*estimation* consiste à chercher des informations sur la valeur de certaines variables à partir d'une approximation de la valeur des variables mesurées et de la connaissance des contraintes modélisant le système que l'on considère (voir par exemple [WAL 97] pour un cours sur le sujet).

Deux types d'approches pour la représentation de l'incertitude sur les variables sont généralement considérés :

- l'*approche probabiliste* : la connaissance de la valeur d'une variable est représentée par une densité de probabilité sur cette variable,
- l'*approche ensembliste* : cette connaissance est représentée par un sous-ensemble de \mathbb{R} (par exemple un intervalle) dont on sait qu'il contient à coup sûr la valeur de la variable.

L'approche probabiliste est souvent plus satisfaisante d'un point de vue théorique, mais réclame des hypothèses que nous ne sommes pas toujours capables de satisfaire en pratique. De plus la manipulation de variables incertaines, sous un angle probabiliste, est très complexe dès que l'on sort du cadre des contraintes linéaires et des incertitudes gaussiennes. Prenons par exemple deux variables aléatoires gaussiennes a et b centrées en zéro et d'écart type $\sigma = 1$. Le calcul de la loi de probabilité de la variable $c = a.b$ nécessite la connaissance de la loi jointe du couple (a, b) , qui est souvent inconnue. Pour pouvoir continuer le calcul, nous pouvons supposer, comme cela est souvent fait, que les variables sont indépendantes. Dans ce cas, on peut trouver de façon unique la loi de probabilité, mais le calcul intégral qui en résulte peut s'avérer difficile.

L'approche ensembliste est beaucoup plus pauvre, dans le sens où l'information dont on dispose sur la variable n'est que partiellement représentée. On ne peut plus dire, dans quelle partie de l'ensemble la variable a le plus de chance de se trouver. En revanche, l'approche ensembliste rend plus aisés certains traitements non-linéaires de l'incertitude. Par exemple, si $a \in [-1, 5]$ et $b \in [3, 7]$, on peut en déduire que $a * b \in [-7, 35]$, mais rien ne nous permet de dire si la variable est plus proche de 35 ou de -7.

C'est cette deuxième approche que nous allons considérer dans ce paragraphe. Nous allons nous appuyer sur l'exemple d'un circuit électrique simple pour illustrer

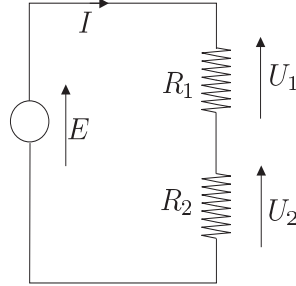


Figure 1. Circuit électrique comprenant une pile et deux résistances en série

l'approche par contraintes d'un problème d'estimation ensembliste. Dans le sous-paragraphe 2.1 nous allons considérer que toutes les variables intervenant dans le problème d'estimation ont le même statut. Le problème d'estimation se pose alors comme un problème de satisfaction de contraintes. Dans le sous-paragraphe 2.2, nous montrerons l'intérêt de sélectionner certaines des variables afin de caractériser de façon plus précise l'ensemble de toutes les solutions.

2.1. Approche non sélective

Le circuit de la figure 1 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

$$E \in [23V, 26V]; I \in [4A, 8A]; U_1 \in [10V, 11V];$$

$$U_2 \in [14V, 17V]; P \in [124W, 130W];$$

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 :

$$\begin{aligned} \text{(i)} \quad P &= EI, & \text{(ii)} \quad E &= (R_1 + R_2) I, & \text{(iii)} \quad U_1 &= R_1 I, \\ \text{(iv)} \quad U_2 &= R_2 I, & \text{(v)} \quad E &= U_1 + U_2. \end{aligned} \quad (1)$$

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

$$\mathbb{S} = \left\{ \begin{pmatrix} E \\ R_1 \\ R_2 \\ I \\ U_1 \\ U_2 \\ P \end{pmatrix} \in \begin{pmatrix} [23, 26] \\ [0, \infty[\\ [0, \infty[\\ [4, 8] \\ [10, 11] \\ [14, 17] \\ [124, 130]; \end{pmatrix} \text{ tels que } \left\{ \begin{array}{l} P = EI \\ E = (R_1 + R_2)I \\ U_1 = R_1I \\ U_2 = R_2I \\ E = U_1 + U_2 \end{array} \right. \right\}$$

Pour le programme source suivant,

```

Bisection none;
Variables
  E in [ 23 , 26],
  R1 in [ 0 , +oo[,
  R2 in [ 0 , +oo[,
  I in [ 4 , 8],
  U1 in [ 10 , 11],
  U2 in [ 14 , 17],
  P in [124 , 130];
Constraints
  P = E*I,
  E = (R1+R2)*I,
  U1 = R1*I,
  U2 = R2*I ,
  E = U1+U2;
    
```

et en moins de 10^{-3} seconde, REALPAVER génère une seule boîte solution

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

dont on est sûr qu'elle enferme \mathbb{S} . Ce résultat se traduit par les relations d'appartenance

$$\begin{array}{ll} E \in [24; 26], & R_1 \in [1.846; 2.307], \\ R_2 \in [2.584; 3.355], & I \in [4.769; 5.417], \\ U_1 \in [10; 11], & U_2 \in [14; 16], \\ P \in [124; 130]. & \end{array} \quad (2)$$

La propagation des contraintes faite par REALPAVER nous a permis d'obtenir avec une assez bonne précision¹ les valeurs des résistances R_1 et R_2 . De plus, elle a généré

1. La précision d'une mesure en physique peut être définie par la largeur de l'intervalle fourni par le capteur (ou l'appareil de mesure). Cet intervalle est censé contenir la vraie valeur de

des intervalles d'appartenance pour les variables 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 .

2.2. Approche sélective

Dans beaucoup de problèmes d'estimation, on peut trouver un groupe de variables (appelées *variables sources*, ou *paramètres*) qui, une fois instanciées, permettent de déterminer, de façon unique et par simple propagation, les valeurs des autres variables du système. Dans le problème de notre circuit électrique, le triplet (E, R_1, R_2) peut être considéré comme un groupe de variables sources car la contrainte (ii) de (1) permet de calculer I , puis les contraintes (iii) et (iv) permettent de calculer U_1 et U_2 . Enfin, la contrainte (i) permet de calculer P . Appelons $\mathbf{p} = (p_1, \dots, p_{n_p})$ le vecteur des variables sources, $\mathbf{y} = (y_1, \dots, y_{n_y})$ le vecteur des autres variables et \mathbf{f} la fonction (ou algorithme) qui permet de retrouver \mathbf{y} à partir de \mathbf{p} . Notons $[\mathbf{p}]$ le domaine *a priori* pour \mathbf{p} et $[\mathbf{y}]$ le domaine *a priori* pour \mathbf{y} . On appelle *ensemble de vraisemblance* l'ensemble

$$\mathbb{S}_p = \{\mathbf{p} \in [\mathbf{p}] \mid \mathbf{f}(\mathbf{p}) \in [\mathbf{y}]\} = [\mathbf{p}] \cap \mathbf{f}^{-1}([\mathbf{y}]). \quad (3)$$

La connaissance de \mathbb{S}_p permet, d'une certaine façon de représenter toutes les solutions de notre problème car chaque instanciation de \mathbf{p} dans \mathbb{S}_p peut être étendue à une solution, c'est-à-dire à un couple (\mathbf{p}, \mathbf{y}) qui satisfait toutes les contraintes. Le problème de la caractérisation de \mathbb{S}_p est appelé *problème d'inversion ensembliste* car \mathbb{S}_p se définit comme l'inverse de l'ensemble $[\mathbf{y}]$ par la fonction \mathbf{f} . Dans l'exemple de notre circuit électrique, on a

$$\mathbf{p} = \begin{pmatrix} E \\ R_1 \\ R_2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} I \\ U_1 \\ U_2 \\ P \end{pmatrix}, \quad \mathbf{f}(\mathbf{p}) = \begin{pmatrix} \frac{E}{R_1 + R_2} \\ \frac{R_1 E}{R_1 + R_2} \\ \frac{R_2 E}{R_1 + R_2} \\ \frac{E^2}{R_1 + R_2} \end{pmatrix}. \quad (4)$$

En pratique, seule une approximation peut être obtenue pour \mathbb{S}_p . Un encadrement de la forme

$$\mathbb{S}_p^- \subset \mathbb{S}_p \subset \mathbb{S}_p^+$$

peut être obtenu par des méthodes intervalles [JAU 01]. Leur efficacité peut être accrue en les combinant à des méthodes de propagation. REALPAVER peut être utilisé pour caractériser la frontière $\partial\mathbb{S}_p$ de \mathbb{S}_p . Essayons pour cela de trouver les contraintes

la mesure, c'est-à-dire, la valeur que nous donnerait un capteur idéal qui aurait une précision infinie.

satisfaites par les éléments de $\partial\mathbb{S}_p$. Notons tout d'abord que la contrainte $\mathbf{f}(\mathbf{p}) \in [\mathbf{y}]$ définissant \mathbb{S}_p peut se réécrire comme suit :

$$\mathbf{f}(\mathbf{p}) \in [\mathbf{y}] \Leftrightarrow \begin{cases} f_1(\mathbf{p}) \in [y_1] \\ \vdots \\ f_{n_y}(\mathbf{p}) \in [y_{n_y}] \end{cases} \Leftrightarrow \begin{cases} f_1(\mathbf{p}) \leq y_1^+ & \text{et} & -f_1(\mathbf{p}) \leq -y_1^- \\ \vdots \\ f_{n_y}(\mathbf{p}) \leq y_{n_y}^+ & \text{et} & -f_{n_y}(\mathbf{p}) \leq -y_{n_y}^- \end{cases}$$

Ainsi,

$$\mathbb{S}_p = \left\{ \mathbf{p} \in [\mathbf{p}] \text{ tels que } \max \begin{pmatrix} f_1(\mathbf{p}) - y_1^+; -f_1(\mathbf{p}) + y_1^-; \\ \dots \\ f_{n_y}(\mathbf{p}) - y_{n_y}^+; -f_{n_y}(\mathbf{p}) + y_{n_y}^- \end{pmatrix} \leq 0 \right\}.$$

Remarquons qu'un sur-ensemble \mathbb{S}_p^+ pour \mathbb{S}_p pourrait être obtenu par REALPAVER, mais un nombre de boîtes exagérément grand seraient générés afin de recouvrir entièrement l'intérieur de l'ensemble \mathbb{S}_p . Il convient donc de ne pas découper les boîtes dont on peut prouver qu'elles se trouvent à l'intérieur de \mathbb{S}_p . Le complémentaire de \mathbb{S}_p est donné par

$$\bar{\mathbb{S}}_p = \left\{ \mathbf{p} \in [\mathbf{p}] \text{ tels que } \max \begin{pmatrix} f_1(\mathbf{p}) - y_1^+; -f_1(\mathbf{p}) + y_1^-; \\ \dots \\ f_{n_y}(\mathbf{p}) - y_{n_y}^+; -f_{n_y}(\mathbf{p}) + y_{n_y}^- \end{pmatrix} > 0 \right\}.$$

La frontière $\partial\mathbb{S}_p$ de \mathbb{S}_p pouvant être généralement définie comme l'intersection des adhérences de \mathbb{S}_p et $\bar{\mathbb{S}}_p$, on a

$$\partial\mathbb{S}_p = \left\{ \mathbf{p} \in [\mathbf{p}] \text{ tels que } \max \begin{pmatrix} f_1(\mathbf{p}) - y_1^+; -f_1(\mathbf{p}) + y_1^-; \\ \dots \\ f_{n_y}(\mathbf{p}) - y_{n_y}^+; -f_{n_y}(\mathbf{p}) + y_{n_y}^- \end{pmatrix} = 0 \right\}.$$

Dans le cadre de notre circuit électrique, la frontière $\partial\mathbb{S}_p$ est donc donnée par

$$\partial\mathbb{S}_p = \left\{ \begin{pmatrix} E \\ R_1 \\ R_2 \end{pmatrix} \in \begin{pmatrix} [23, 26] \\ [0, \infty[\\ [0, \infty[\end{pmatrix} \mid \max \begin{pmatrix} \frac{E}{R_1+R_2} - 8; \\ \frac{E}{R_1+R_2} + 4; \\ \frac{R_1 E}{R_1+R_2} - 11; \\ -\frac{R_1 E}{R_1+R_2} + 10; \\ \frac{R_2 E}{R_1+R_2} - 17; \\ -\frac{R_2 E}{R_1+R_2} + 14; \\ \frac{E^2}{R_1+R_2} - 130; \\ -\frac{E^2}{R_1+R_2} + 124 \end{pmatrix} = 0 \right\}.$$

Une caractérisation extérieure de $\partial\mathbb{S}_p$ peut être obtenue par REALPAVER. Rentrons pour cela le programme source suivant.

```

Output
  precision = 0.01,      /* calcul des boites a 10-2 */
  solutions = all,      /* processus complet */
  mode      = hull;    /* renvoie l'enveloppe convexe */
Variables
  E in [23,26],
  R1 in [succ 0, +oo[, /* succ 0 est le premier flottant >0 */
  R2 in [succ 0, +oo[,
  $X in ]-oo,+oo];    /* partage d'une expression */
Constraints
  X = E/(R1+R2),
  0 = max(X-8, max(-X+4, max(R1*X-11, max(-R1*X+10,
    max(R2*X-17, max(-R2*X+14, max(E*X-130, -E*X+124))))));

```

En 21690 ms sur un PC-Celeron 733 MHz, REALPAVER génère alors 97671 boîtes de côté inférieur à 10^{-2} dont l'union contient $\partial\mathbb{S}_p$. Ces boîtes sont représentées sur la figure 2 obtenue à l'aide du visualiseur *Universal Solution Viewer* [CHR 00]. Le gros cube fixant les échelles est donné par $[23.9, 26] \times [1.8, 2.4] \times [2.5, 3.4]$. Les boîtes étant petites, il nous a été difficile de visualiser les éléments individuellement. Pour plus de clarté, le détourage des boîtes a été supprimé. L'ensemble \mathbb{S}_p a une forme d'obélisque creux. On peut vérifier, par un simple calcul que la plus petite boîte englobant tous ces boîtes est

$$[23.970, 26] \times [1.844, 2.307] \times [2.582, 3.356], \quad (5)$$

ce qui est cohérent avec le résultat (2) obtenu au paragraphe 2.1.

3. Stabilité robuste

Dans ce paragraphe, nous allons traiter un problème de commande robuste. L'adjectif *robuste*, signifie dans la communauté des automaticiens : "dans tous les cas possibles, même le pire". Les méthodes garanties (c'est-à-dire complètes) comme celles proposées par l'analyse par intervalles ont donc eu un certain succès pour le traitement de problèmes de commande robuste [KOL 88, KEA 89, KHL 92, KOL 93, MAL 97]. Dans ce paragraphe, nous allons montrer que les méthodes de propagation combinées à l'arithmétique des intervalles permet de résoudre efficacement et simplement des problèmes pourtant considérés comme difficiles (voir [BAR 94]).

Considérons tout d'abord un système linéaire d'entrée $u(t)$ et de sortie $y(t)$ pouvant être représenté par l'équation différentielle linéaire

$$a_n y^{(n)} + \dots + a_1 \dot{y} + a_0 y = b_m u^{(m)} + \dots + b_1 \dot{u} + b_0 u. \quad (6)$$

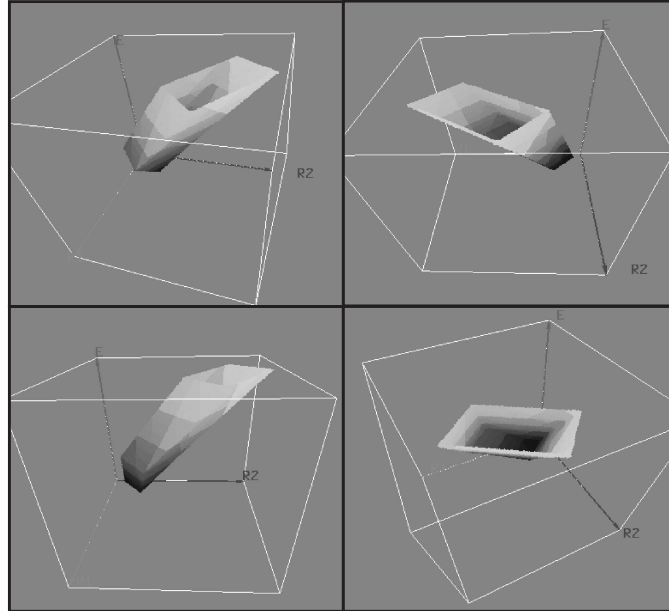


Figure 2. Représentation 3D de la frontière de l'ensemble \mathbb{S}_p .

L'entrée $u(t)$ peut être choisie arbitrairement, par exemple en fonction de $y(t)$ de façon à changer le comportement du système.

Exemple 3.1 Choisissons pour loi de commande, $u(t) = w(t) - y(t)$, où $w(t)$ est une nouvelle entrée. L'équation du nouveau système ainsi formé est

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 \dot{y} + a_0 y = b_m (w^{(m)} - y^{(m)}) + \dots + b_0 (w - y),$$

c'est-à-dire

$$a_n y^{(n)} + \dots + (b_m + a_m) y^{(m)} + \dots + (b_0 + a_0) y + a_1 \dot{y} + a_0 y = b_m w^{(m)} + \dots + b_0 w.$$

Nous avons ainsi changé l'équation différentielle qui régit le système. Un des problèmes de base de l'automatique est de trouver une loi de commande qui fasse que le système se comporte comme nous le souhaitons. ■

Revenons à l'équation de notre système (6) et prenons sa transformée de Laplace. Cette opération revient à remplacer l'opérateur de dérivation par s . Cette équation devient

$$a_n s^n y(s) + \dots + a_1 s y(s) + a_0 y(s) = b_m s^m u(s) + \dots + b_1 s u(s) + b_0 u(s).$$

Le polynôme

$$A(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0$$

est appelé *polynôme caractéristique*. C'est lui qui détermine le comportement dynamique du système. Par exemple, si toutes ses racines sont à partie réelles négatives, alors le système est stable, c'est-à-dire que si on met soudainement l'entrée $u(t)$ du système à zéro, alors la sortie va converger vers zéro. Le critère de Routh que nous allons maintenant présenter (voir [DOR 95] page 278 pour plus de détails) permet de prouver la stabilité d'un système avec un nombre d'opérations très faible, sans avoir à calculer toutes les racines du polynôme caractéristique.

Théorème 3.1 (Critère de Routh). Soit $P(s) = a_n s^n + \dots + a_1 s + a_0$ un polynôme. La table de Routh se construit comme suit

a_n	a_{n-2}	a_{n-4}	a_{n-6}	\dots	0	0
a_{n-1}	a_{n-3}	a_{n-5}	a_{n-7}	\dots	0	0
b_1	b_2	b_3			0	0
c_1	c_2	c_3			0	0
\vdots	\vdots	\vdots			\vdots	\vdots

(7)

avec

$$\begin{aligned}
 b_1 &= \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} & b_2 &= \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}} & \dots \\
 c_1 &= \frac{b_1 a_{n-3} - a_{n-1} b_2}{b_1} & c_2 &= \frac{b_1 a_{n-5} - a_{n-1} b_3}{b_1} & \dots \\
 & \dots & & \vdots &
 \end{aligned}
 \tag{8}$$

Notons que les deux premières lignes sont les coefficients de $P(s)$. Les autres éléments t_{ij} à la ligne $i > 2$ et à la colonne j sont obtenues par le calcul

$$t_{ij} = \frac{t_{i-1,1} t_{i-2,j+1} - t_{i-2,1} t_{i-1,j+1}}{t_{i-1,1}} \tag{9}$$

Les racines de $P(s)$ sont toutes à parties réelles strictement négatives si et seulement si les éléments de la première colonne de la table de Routh sont tous de même signe. ■

Considérons à titre d'exemple, le problème d'une moto roulant à une vitesse faible et constante de 1m/s (voir [DOR 95] 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 :

$$\phi(s) = \frac{1}{s^2 - \alpha_1} \theta(s) \tag{10}$$

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

$$\theta(s) = \frac{\alpha_2 + \alpha_3 s}{\tau s + 1} (\phi_d(s) - \phi_m(s)) \tag{11}$$

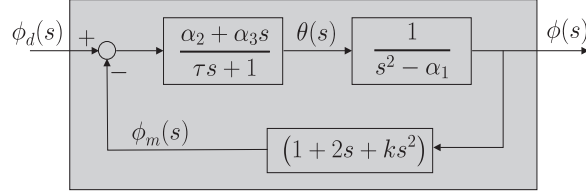


Figure 3. Moto stabilisée par un régulateur du premier ordre

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

$$\phi_m(s) = (1 + 2s + ks^2) \phi(s) \quad (12)$$

L'ensemble de ces équations est représenté par le schéma de la figure 3. En manipulant les trois équations précédentes de façon à supprimer $\theta(s)$ et $\phi_m(s)$, il vient

$$\phi(s) = \frac{1}{s^2 - \alpha_1} \left(\frac{\alpha_2 + \alpha_3 s}{\tau s + 1} (\phi_d(s) - (1 + 2s + ks^2) \phi(s)) \right) \quad (13)$$

En isolant $\phi(s)$, on obtient la relation entrée-sortie du système régulé :

$$\phi(s) = \frac{\alpha_2 + \alpha_3 s}{(s^2 - \alpha_1)(\tau s + 1) + (\alpha_2 + \alpha_3 s)(1 + 2s + ks^2)} \phi_d(s) \quad (14)$$

Son polynôme caractéristique est

$$(s^2 - \alpha_1)(\tau s + 1) + (\alpha_2 + \alpha_3 s)(1 + 2s + ks^2) = a_3 s^3 + a_2 s^2 + a_1 s + a_0$$

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	(15)
a_2	a_0	
$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0	
a_0	0	

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.

Les valeurs nominales pour les coefficients sont : $\alpha_1 = 9$, $\alpha_2 = 3$, $\alpha_3 = 1$, $\tau = 2$, $k = -3$. Or, ces coefficients ne sont connus que de façon approximative. En pratique, on connaît un intervalle d'appartenance pour chacune de ces variables. Notons les $[\alpha_1]$, $[\alpha_2]$, $[\alpha_3]$, $[\tau]$ et $[k]$. Le problème de stabilité robuste consiste à montrer la

stabilité du système pour toutes les valeurs possibles des coefficients. Cela revient à dire que l'on a la stabilité robuste si

$$\forall \alpha_1 \in [\alpha_1], \forall \alpha_2 \in [\alpha_2], \forall \alpha_3 \in [\alpha_3], \forall \tau \in [\tau], \forall k \in [k], \\ a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2} \text{ et } a_0 \text{ sont de même signe.}$$

Or, pour quatre nombres b_1, b_2, b_3 et b_4 on a l'équivalence

$$b_1, b_2, b_3 \text{ et } b_4 \text{ sont de même signes} \\ \Leftrightarrow \max(\min(b_1, b_2, b_3, b_4), -\max(b_1, b_2, b_3, b_4)) > 0$$

La condition de stabilité robuste revient donc à dire que la proposition

$$\exists \alpha_1 \in [\alpha_1], \exists \alpha_2 \in [\alpha_2], \exists \alpha_3 \in [\alpha_3], \exists \tau \in [\tau], \exists k \in [k], \\ \max\left(\min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right), -\max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right)\right) \leq 0$$

est fausse. Prouver la stabilité robuste de notre système revient donc à montrer que le CCSP dont les variables, les domaines et les contraintes sont donnés respectivement par

$$\mathcal{V} = \{a_0, a_1, a_2, a_3, \alpha_1, \alpha_2, \alpha_3, \tau, k\}, \\ \mathcal{D} = \{[\alpha_0], [\alpha_1], [\alpha_2], [\alpha_3], [\alpha_2], [\alpha_3], [\tau], [k]\}, \\ \mathcal{C} = \left\{ \begin{array}{l} a_3 = \tau + \alpha_3 k ; a_2 = \alpha_2 k + 2\alpha_3 + 1 ; a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2, \\ a_0 = -\alpha_1 + \alpha_2 ; b = \frac{a_2 a_1 - a_3 a_0}{a_2} ; \\ \max(\min(a_3, a_2, b, a_0), -\max(a_3, a_2, b, a_0)) \leq 0. \end{array} \right\}$$

possède un ensemble solution vide.

Supposons que les valeurs pour les coefficients de la moto, du régulateur et du capteur satisfont

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

alors REALPAVER prouve en moins de 10^{-3} secondes la stabilité robuste du système bouclé. Pour le programme source donné ci-dessous, REALPAVER retourne un ensemble vide.

```

Variables
a0      in  ]-oo,+oo[,   a1      in  ]-oo,+oo[,
a2      in  ]-oo,+oo[,   a3      in  ]-oo,+oo[,
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], $b      in  ] -oo,+oo[;
Constraints
a3 = tau + alpha3*k,
```

```

a2 = alpha2*k + 2*alpha3 + 1,
a1 = alpha3 - alpha1*tau + 2*alpha2,
a0 = alpha2 - alpha1,
b = (a2*a1 - a3*a0)/a2,
0 >= max( min(a3,min(a2,min(b,a0))),
          -max(a3,max(a2,max(b,a0))));

```

4. Conclusion

De nombreux problèmes issus de l'automatique comme la commande robuste ou l'estimation ensembliste possèdent pour solution un ensemble vide ou bien un ensemble de volume non nul contenant une infinité non-dénombrable d'éléments. Dans ce contexte, les approches classiques conçues pour traiter des problèmes d'optimisation globales ou des systèmes d'équations non-linéaires avec autant d'équations que d'inconnues, semblent inadaptées.

En revanche, une approche par propagation de contraintes sur les intervalles peut se montrer efficace pour traiter ce type de problèmes. Avec cette approche, nous avons traité avec succès un problème d'estimation ensembliste et un problème de commande robuste. Ces deux problèmes, classiques en automatique sont considérés comme difficiles dans le cas non linéaire et avec une approche garantie. Une multitude d'autres problèmes d'automatique ou de robotique particulièrement adaptés à cette approche par contraintes peuvent être trouvés dans [JAU 01].

Le solveur REALPAVER accompagné d'un visualiseur graphique [CHR 00] peut résoudre de façon très efficace la plupart d'entre eux. Sa distribution récente devrait, nous l'espérons, avoir un impact important dans la communauté des automaticiens et des roboticiens.

5. Bibliographie

- [BAR 94] BARMISH B. R., *New Tools for Robustness of Linear Systems*, MacMillan, New York, NY, 1994.
- [BEN 97] BENHAMOU F., OLDER W., « Applying Interval Arithmetic to Real, Integer and Boolean Constraints », *Journal of Logic Programming*, , 1997, p. 1-24.
- [CHR 00] CHRISTIE M., « Universal Solution Viewer : User's Manual, IRIN », 2000.
- [CLE 87] CLEARY J. G., « Logical arithmetic », *Future Computing Systems*, vol. 2, n° 2, 1987, p. 125-149.
- [DAV 87] DAVIS E., « Constraint propagation with interval labels », *Artificial Intelligence*, vol. 32, n° 3, 1987, p. 281-331.
- [DOR 95] DORF R., BISHOP R., *Modern Control Systems*, Electrical and Computer Engineering : Control Engineering, Addison-Wesley Publishing Compagny, 1995.

- [GRA 01] GRANVILLIERS L., « On the Combination of Interval Constraint Solvers », *Reliable Computing*, vol. 7, n° 6, 2001, p. 467-483.
- [GRA 02] GRANVILLIERS L., « RealPaver, disponible sur <http://www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver/%20guilf>, 2002.
- [JAU 01] JAULIN L., KIEFFER M., DIDRIT O., WALTER E., *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, Londres, 2001.
- [KEA 89] KEARFOTT R. B., « Interval mathematics techniques for control theory computations », BOWERS K., LUND J., Eds., *Computation and Control. Proceedings of the Bozeman Conference*, vol. 20 de *Progress in Systems and Control Theory*, p. 169-178, Birkhäuser, Boston, MA, 1989.
- [KHL 92] KHLEBALIN N. A., « Interval automatic systems - Theory, computer-aided design and applications », *Interval Computations*, vol. 3, 1992, p. 106-115.
- [KOL 88] KOLEV L. V., MLADENOV V. M., VLADOV S. S., « Interval mathematics algorithms for tolerance analysis », *IEEE Transactions on Circuits and Systems*, vol. 35, n° 8, 1988, p. 967-974.
- [KOL 93] KOLEV L., « An interval first-order method for robustness analysis », *Proceedings of the IEEE International Symposium on Circuits and Systems*, Chicago, IL, 1993, p. 2522-2524.
- [LHO 93] LHOMME O., « Consistency Techniques for Numeric CSPs », *Proceedings of the International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993, p. 232-238.
- [LOT] LOTTAZ C., SAM-HAROUD D., « SpaceSolver, disponible sur <http://liawww.epfl.ch/~lottaz/SpaceSolver/> ».
- [LOT 00] LOTTAZ C., « Collaborative design using solution spaces », PhD dissertation 2119, Swiss Federal Institute of Technology in Lausanne, Suisse, 2000.
- [MAC 77] MACKWORTH A. K., « Consistency in networks of relations », *Artificial Intelligence*, vol. 8, n° 1, 1977, p. 99-118.
- [MAL 97] MALAN S. A., MILANESE M., TARAGNA M., « Robust analysis and design of control systems using interval arithmetics », *Automatica*, vol. 33, n° 7, 1997, p. 1363-1372.
- [MOO 79] MOORE R. E., *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, PA, 1979.
- [RAT 00] RATSCHAN S., « Approximate Quantified Constraint Solving (AQCS) », disponible sur : <http://www.risc.uni-linz.ac.at/research/software/AQCS>, 2000.
- [SAM 96] SAM-HAROUD D., FALTINGS B., « Consistency Techniques for Continuous Constraints », *Constraints*, vol. 1, n° 1-2, 1996, p. 85-118.
- [van 97] VAN HENTENRYCK P., DEVILLE Y., MICHEL L., *Numerica : A Modeling Language for Global Optimization*, MIT Press, Boston, MA, 1997.
- [WAL 97] WALTER E., PRONZATO L., *Identification of Parametric Models from Experimental Data*, Springer-Verlag, Londres, UK, 1997.