Systèmes d'équations algébriques et robots parallèles

Daniel Lazard*
LIP6, Université P. et M. Curie,
75252 Paris Cedex 05, France

Résumé

Les robots parallèles font intervenir des systèmes d'équations polynomiales difficiles, qui ont longtemps été au delà des possibilités des logiciels de résolution.

Dans cet article, on montre comment les progrès de ces logiciels ont régulièrement été suivis de progrès dans la connaissance de ces robots : Il y a quelques années les logiciels ne permettaient que de compter le nombre de solutions complexes. Aujourd'hui, ils permettent de calculer rapidement toutes les solutions réelles, avec une précision garantie, ce qui permet la simulation et la commande de haute précision de robots effectivement construits.

1 Introduction

Depuis environ 10 ans, les progrès significatifs dans la résolution des systèmes d'équations polynomiales ont eu des applications immédiates dans l'étude des robots parallèles.

Inversement, de nombreux progrès des logiciels de résolutions ont été impulsés par les besoins de la robotique.

Dans cet article, nous retraçons succinctement l'histoire de ces progrès conjoints, en nous concentrant sur trois point importants qui font l'objet des trois sections qui suivent :

- Détermination du nombre de solutions complexes.
- Calcul des solutions réelles.
- Simulation et commande d'une machine d'usinage à grande vitesse effectivement commercialisée.

*E-mail : Daniel.Lazard@lip6.fr

2 Nombre de solutions complexes

J'ai été confronté aux robots parallèles il y a tout juste 10 ans, à l'occasion d'un rapport que j'ai eu à faire sur un projet de recherches soumis à la NSF.

L'auteur de cette proposition, considérant que les bases de Gröbner n'étaient pas assez efficaces pour déterminer le nombre de positions possibles d'un robot parallèle, avait pour objectif de développer de nouveaux algorithmes plus efficaces, ce qui était manifestement au delà de ses compétences. Mais son affirmation sur l'inefficacité des bases de Gröbner méritait d'être vérifiée, car elle était basée sur des logiciels qui n'étaient pas les plus efficaces de l'époque.

En fait, la première version du logiciel GB de J.-C. Faugère permettait déjà de calculer la base de Gröbner pour l'ordre du degré dans le cas des plate-formes planaires, mais saturait la mémoire à cause de la taille des coefficients dans le cas général [5]. Mais, si les bases ainsi calculées permettaient de déterminer le nombre de solutions complexes, elles ne permettaient pas de calculer explicitement ces solutions.

Cependant, ces calculs ont permis de démontrer que le nombre maximal de solutions complexes pour un robot parallèle général (plate-forme de Mc Cough) était égal à 40, d'abord dans le cas d'une plate-forme plane [5], puis dans le cas général [6]. Il est remarquable que ces preuves utilisent des calculs de base de Gröbner, mais que ceux-ci étaient dès cette époque à la portée de MAPLE. Ainsi, pour la preuve du cas général, les calculs ont nécessité 3h30 en MAPLE (pour 3s de GB).

Enfin, nous avons montré que les techniques de bases de Gröbner, telles qu'implantées dans le logiciel GB, permettaient de déterminer le nombre maximal de solutions complexes de n'importe quelle classe de robots parallèles [3]. C'est la préparation de cet article qui a suscité l'introduction dans GB de l'algorithme de la trace (T-GROEBNER), qui a immédiatement permis de diviser par 20 certains des temps de calcul.

3 Calcul des solutions réelles

Comme nous l'avons dit plus haut, jusque vers 1994, les implémentations disponibles des bases de Gröbner permettaient de calculer les bases pour l'ordre du degré, mais avaient des difficultés pour calculer les bases pour l'ordre lexicographique, et ne permettaient pas de calculer explicitement les solutions.

Celles-ci sont devenues accessibles avec l'algorithme rapide de changement d'ordre[1] de la thèse de Faugère [1] et les résultats de la thèse de F. Rouillier [8].

[1]Il faut souligner que les bases de Gröbner pour l'ordre lexicographique constituent une étape presque inévitable pour le calcul des solutions : En effet, celles-ci ont usuellement la forme dite "shape lemma" $f_1(x_1), x_2 - f_2(x_2), \ldots, x_n - f_n(x_n)$ dans le cas des robots parallèles, forme qui ramène le calcul des solutions à la résolution d'une équation à une variable.

C'est ainsi, qu'il est devenu possible, pour n'importe quel robot parallèle, même le plus général, de déterminer et de certifier le nombre de positions réelles possibles, et de calculer explicitement ces positions [7].

À cette époque, la résolution complète du système d'équations associé à un robot parallèle prenait quelques dizaines d'heures. Les progrès algorithmiques intervenus depuis ont permis de ramener le temps de calcul à quelques secondes.

Le tableau suivant est extrait de [4]. Il illustre les progrès algorithmiques accomplis, indépendamment de l'évolution des matériels, car les temps de calculs ont été obtenus en ré-exécutant tous les programmes sur une même machine actuelle (station de travail).

Année	Temps de calcul de la base de Gröbner	Temps de calcul RUR et isolation des racines	Temps total
1994	T-GROEBNER 2 h 30 mn	HERMITE 7 h	9 h 30 mn
1995	T-GROEBNER 2 h 30 mn	RUR + STURM-HABICHT 20 mn + 2 h	4 h 50 mn
1996	T-GROEBNER 2 h 30 mn	RUR + USPENSKY 20 mn + 30 s	2 h 50 mn
1997	ACCEL-GB 10 s	RUR + USPENSKY 20 mn + 30 s	20 mn 40 s
1998	ACCEL-GB 10 s	RUR (RS) + USPENSKY II 30 s + 0.1 s	40 s
1999	F4 4 s	RUR2 + USPENSKY II 5 s + 0.1 s	10 s
2000	F4 LEX 3 s	USPENSKY II 0.1 s	3 s

TAB. 1 – Histoire du calcul du modèle géométrique direct d'un robot parallèle.

Il n'est pas possible, dans le cadre de cet article, de décrire tous les algorithmes utilisés, aussi nous contentons d'expliciter les sigles utilisés.

T-GROEBNER — Il s'agit d'un algorithme de Buchberger classique effectué simultanément avec des coefficients entiers et avec des coefficients réduits modulo un nombre premier fixe p. Les paires critiques donnant 0 modulo p ne sont pas calculés sur les entiers. Les éventuelles, mais improbables, erreurs introduites sont corrigées à la fin.

HERMITE — C'est un algorithme qui détermine le nombre de solutions réelles d'un système décrit par une base de Gröbner (pour un ordre quelconque) [8]. Cet algorithme ne permet pas de calculer effectivement ces solutions.

RUR — La Représentation Univariée Rationnelle ou RUR consiste à calculer, à partir d'une base de Gröbner pour un ordre quelconque, les solutions sous la forme

$$\begin{aligned} f_0(T) &= 0 \\ X_1 &= f_1(T)/f_0'(T) \\ \ldots & \\ X_n &= f_n(T)/f_0'(T) \end{aligned}$$

où T et une variable auxiliaire, et où les f_i sont des polynômes univariés [9] RUR (RS) et RUR2 sont les améliorations successives de cet algorithme.

STURM-HABITCH, USPENSKY — Ces algorithmes comptent le nombre de racines réelles d'un polynôme univarié dans un intervalle. En subdivisant cet intervalle, ces algorithmes permettent donc de calcules les solutions avec une précision garantie. USPENSKY II est la version améliorée qui est décrite dans [10].

ACCEL-GB, F4 — F4 est un algorithme permettant de calculer les bases de Gröbner par des méthodes d'algèbre linéaire [2]. F4 LEX consiste à utiliser F4 directement avec l'ordre lexicographique, et à donner le résultat sous la forme RUR.

ACCEL-GB est une forme préliminaire de cet algorithme où l'algèbre linéaire est simulée par des opérations sur les polynômes.

Il faut souligner que tous ces algorithmes travaillent avec des nombres rationnels exacts, ou dans certains cas avec de l'arithmétique d'intervalles. Ainsi, ils calculent, avec une précision garantie, *toutes* les solutions réelles du système.

4 La machine outil hexapode de CMW

La société *Constructions Mécaniques des Vosges Marioni* (CMW) fabrique une machine d'usinage à grande vitesse (50m/mn) basée sur une plate-forme de robot parallèle (plate-forme de Gough).

Pour une telle machine, les contraintes de précision sont sévères, puisque l'on voudrait atteindre une précision de l'ordre du micron (Dans son état actuel, l'hexapode de CMW permet une précision de 5μm dans la trajectoire d'un outil pesant plus de 300kg avec son support mobile et ayant un espace de travail de plus de 70cm.

La mise au point et la commande de cette machine nécessitent donc de simuler son comportement, afin de déterminer les causes des pertes de précision. Celles-ci sont multiples :

- Manque de rigidité mécanique du système. C'est ce qui détermine la précision maximale qu'une machine donnée ne pourra jamais dépasser. C'est pour cette raison que l'usinage à grande vitesse de précision est impossible avec d'autres architectures.

– Erreurs de calibration de la machine et de positionnement de la pièce à usiner.
– Discrétisation introduite par le logiciel de CAO définissant la trajectoire de l'outil.
– Seconde discrétisation (généralement avec un pas différent) introduite par le logiciel de commande qui transforme cette trajectoire théorique en impulsions envoyées aux moteurs.
– Troisième discrétisation due au temps de retour des informations fournies par les capteurs (avec les boîtiers de commande standard, ce temps de retour est de l'ordre de 1,5ms, pendant lequel l'outil parcourt plus d'un millimètre !).

La simulation consiste donc à calculer la trajectoire décrite par l'outil en fonction des commandes effectivement envoyées aux moteurs. Compte tenu des spécifications ci-dessus, cette trajectoire doit être déterminée avec une précision supérieure à celle des nombres flottants de simple précision.

Pour un tel calcul, la méthode traditionnelle consiste à utiliser une méthode numérique de suivi de trajectoire (méthode de Newton). À ce niveau de précision, elle pose des problèmes réels de mise en œuvre et de certification. En outre, cette approche ne calcule qu'une seule des positions possibles du robot, ce qui ne permet pas de détecter le saut éventuel entre deux trajectoires voisines, ni la proximité d'une éventuelle singularité dans le domaine de travail.

C'est pour ces raisons que l'approche décrite dans la section précédente a été choisie : Une simulation d'un millier de points nécessite un petit nombre d'heures sur une station de travail, et fournit toutes les trajectoires avec une précision arbitraire et certifiée.

Il est intéressant de noter que ce robot parallèle à été conçu par J.-P. Merlet avec un support et une plate-forme plans, et un certain nombre de symétries ; mais les erreurs de fabrication font que le robot réel est tout à fait général, et que la planarité et les symétries ne peuvent pas être utilisées pour simplifier les calculs.

Il serait évidemment souhaitable de réduire ces temps de simulation à quelques secondes. Cela ne semble pas actuellement accessible au calcul purement exact. Des tests préliminaires indiquent que c'est tout à fait possible, soit en calculant les bases de Gröbner avec des coefficients flottants (il apparaît que l'algorithme F4 le permette) soit en utilisant une méthode de Newton avec arithmétique d'intervalles. Cependant, pour ne pas perdre la garantie de précision des résultats, il faut mettre au points des tests permettant de revenir automatiquement à la méthode exacte pour les points des trajectoires où ces méthodes approchés n'ont pas une précision suffisante.

Références

[1] J.-C. Faugère. *Résolution des systèmes d'équations polynomiales.* Thèse, Université de Paris VI, 1994.

[2] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra* **139** (1999), 61–88.

[3] J.-C. Faugère et D. Lazard. The combinatorial classes of parallels manipulators. *Mechanism and Machine Theory*, **30** (1995), 765–776.

[4] J.-C. Faugère, F. Rouillier, L. Rolland. *Forward position analysys for parallel robots.* Article en préparation.

[5] D. Lazard. Stewart platforms and Gröbner basis. *3rd International Workshop on Advances in Robot Kinematic (3ARK, Ferrare, Italy)*, V. Parenti-Castelli et J. Lenarcic (éd.), pp. 136–142, Sept. 1992.

[6] D. Lazard. On the representation of Rigid-Body Motions and its Application to Generalized Platform Manipulators. *Computational Kinematics*, J. Angeles et al. ed., Solid Mechanics and its Applications **28** (Kluwer, 1993), 175–182.

[7] F. Rouillier. Real root counting for some robotics problems. *Solid Mechanics and its Applications* **40** (Kluwer, 1995), 73–82.

[8] F. Rouillier. *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux.* Thèse, Université de Rennes I, 1996.

[9] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, **5** (1999), 433–461.

[10] F. Rouillier et P. Zimmermann. Efficient isolation of a polynomial real roots. Rapport de recherche 4113, Institut National de Recherche en Informatique et en Automatique, 2001. (16 pages).