



Etalonnage géométrique d'un robot à six degrés de liberté

Xavier Baguenard¹, Massa Dao¹, Luc Jaulin¹, Wisama Khalil²

¹LISA, FRE 2656,

Université d'Angers, 62 avenue notre Dame du lac, 49000 Angers, France

{baguenar, dao}@istia.univ-angers.fr,

jaulin@univ-angers.fr,

²IRCCyN, UMR, CNRS 6597,

1 rue de la Noe, BP 92101, 44321 Nantes cedex 03, France

wisama.khalil@irccyn.ec-nantes.fr

Mots clés : analyse par intervalles, calcul ensembliste, étalonnage des robots, estimation ensembliste, méthode garantie, propagation de contraintes, robotique.

Keywords: interval analysis, set computation, geometric calibration of robot, set estimation, reliable methods, constraint propagation.

Résumé : L'étalonnage des robots est généralement effectué par des méthodes locales qui ne donnent pas d'information fiable sur l'intervalle de confiance autour du résultat et qui sont souvent piégées dans des minima locaux. Afin de pallier ces problèmes, une méthode ensembliste est proposée dans cet article. Cette méthode permet de déterminer, pour chaque paramètre à estimer, un intervalle le contenant qui dépend de l'incertitude sur la mesure et de l'erreur de modélisation.

Abstract: Robot manipulators calibration is usually performed by local methods which may be trapped into local minima. In this paper, a reliable approach, based on set computation, is proposed. It makes it possible to compute for each parameter an interval containing its actual value.

I. Introduction

Afin de commander correctement les robots, il est fondamental de connaître avec précision leurs paramètres géométriques. Ces paramètres interviennent dans le calcul des modèles géométriques direct et inverse. Le problème de l'étalonnage consiste à estimer ces paramètres géométriques, à partir d'un ensemble de mesures prises sur le robot pour des configurations

différentes.

Pour les robots de type série, il est toujours possible d'obtenir un modèle mathématique, dit géométrique direct, qui permet de calculer les coordonnées de l'organe terminal \mathbf{x} en fonction des paramètres géométriques \mathbf{p} et des variables articulaires \mathbf{q} .

Le but de cet article est de tester les méthodes ensemblistes sur la résolution d'un problème d'étalonnage et déterminer ainsi \mathbf{p} .

Dans la section II., nous présenterons le robot que nous allons utiliser ainsi que le modèle géométrique direct [7] utilisé pour la génération des contraintes associées au problème de l'étalonnage [3], [4], [6] et [8]. Une technique de propagation de contraintes [10] sur les intervalles [5] sera alors présentée en section III. (voir [5], [2] et [9] pour une présentation) puis sera appliquée au problème d'étalonnage en section IV..

Notation : Dans la suite de cet article les notations seront les suivantes : une coordonnée ou un paramètre sera noté t , l'intervalle associé $[t]$, un vecteur sera noté en gras \mathbf{t} et le pavé associé $[\mathbf{t}]$, le pavé étant un *produit cartésien d'intervalles*.

II. Problème de l'étalonnage géométrique

Le but de l'étalonnage géométrique des robots est d'obtenir les valeurs des paramètres géométriques. Lever l'incertitude sur ces paramètres permet d'ajuster les modèles mathématiques afin de positionner de façon plus précise les robots. En effet les erreurs de positionnement en robotique sont dans la plupart des cas dues à des valeurs de paramètres \mathbf{p} mal ajustées.

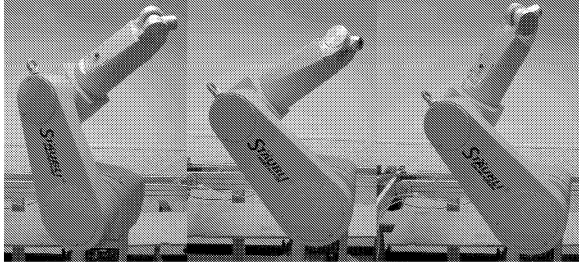


Figure 1 – Robot Staubli RX90 dans différentes configurations.

A. Présentation du robot

Le robot Staubli RX90 de la figure 1 est un robot de type série possédant six degrés de liberté d'axes rotoïdes. Une configuration du robot est donnée par le vecteur des coordonnées articulaires

$$\mathbf{q} = (q_1, \dots, q_6)^t \in \mathbb{R}^6, \quad (1)$$

où les q_i sont les angles des articulations du robot et sont exprimés en radians. Chacune de ces coordonnées correspond pour ce robot à une rotation d'axe \mathbf{k}_i . Sur la figure 2, les six axes associés à ces articulations rotoïdes sont représentés.

Le robot Staubli est conçu pour manipuler un outil, appelé organe terminal, qui se trouve en bout de la chaîne cinématique. L'outil sera repéré par trois points dont les coordonnées dans le repère de l'atelier forment le vecteur

$$\mathbf{x} = (a_x^1, a_y^1, a_z^1, a_x^2, a_y^2, a_z^2, a_x^3, a_y^3, a_z^3)^t \in \mathbb{R}^9. \quad (2)$$

Rappelons que la structure du robot est définie par un ensemble de paramètres géométriques. Ils représentent les paramètres de construction ou d'installation du robot (voir [7]). Ils sont illustrés sur la figure 2 représentant le modèle associé au robot.

1) Les paramètres θ_0 , r_0 , α_1 , d_1 , θ_1^o et r_1 définissent la base du robot par rapport au repère atelier, nous possédons peu de connaissance sur ces paramètres qui correspondent à l'installation du robot dans le repère atelier.

2) Les paramètres b_x^i , b_y^i et b_z^i , $i = 1, 2, 3$ fixant trois points sur l'organe terminal, permettent de définir un repère outil par rapport au repère fixé au dernier corps. Trois contraintes relient ces paramètres car les distances entre les trois points sont connues.

3) Les paramètres géométriques α_j , d_j , θ_j^o et r_j sont liés au robot lui-même. Ce sont

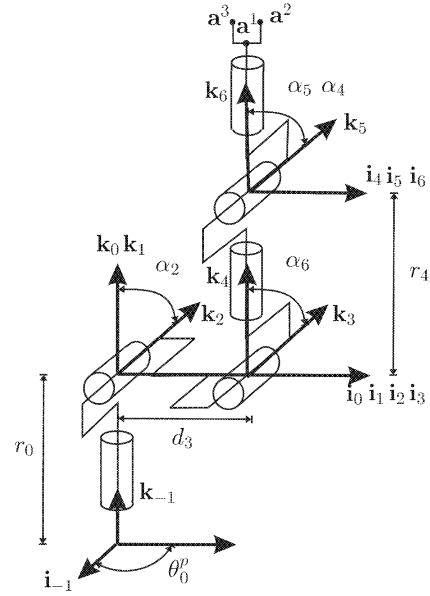


Figure 2 – Schéma du robot STAUBLI RX90 avec ses paramètres non nuls.

des paramètres constructeurs bien connus. Par exemple, d_3 correspond à la longueur du corps 3 du robot et r_4 à la longueur du corps 4. Les θ_j^o correspondent à l'offset sur les variables articulaires q_j , $j = 2, 3, 4, 5$.

Ces paramètres seront désignés par le vecteur

$$\mathbf{p} = (r_0, \alpha_1, d_1, r_1, \dots, \alpha_5, d_5, r_5, \alpha_6, d_6, \theta_0, \theta_1^o, \dots, \theta_5^o, b_x^1, b_y^1, b_z^1, b_x^2, b_y^2, b_z^2, b_x^3, b_y^3, b_z^3)^t. \quad (3)$$

B. Modèle associé au robot

Connaissant le vecteur des variables articulaires \mathbf{q} et le vecteur des paramètres géométriques \mathbf{p} , il est possible de calculer la position d'un point de l'organe terminal dans le repère atelier avec le modèle géométrique direct [7]. Ce modèle est appliqué aux trois points de l'outil pour obtenir la relation

$$\mathbf{x} = \mathbf{f}(\mathbf{p}, \mathbf{q}). \quad (4)$$

Le développement scalaire des relations matricielles du modèle géométrique direct s'exprime comme une composition de produits, de sommes, et de fonctions trigonométriques. Le modèle est donc fortement non linéaire.

C. Principe de l'étalonnage

Afin d'estimer les paramètres géométriques \mathbf{p} , le principe de l'étalonnage que nous allons utiliser est le suivant

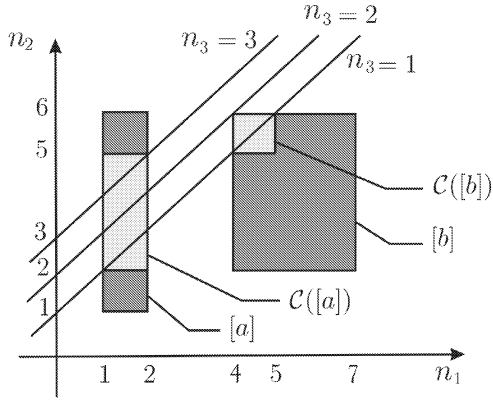


Figure 3 – Opérateur de contraction de la contrainte $n_3 = n_2 - n_1$ sur les pavés $[a]$ et $[b]$.

- 1) Positionner le robot dans r configurations différentes $\mathbf{q}(1), \dots, \mathbf{q}(r)$,
- 2) Mesurer la position des trois points fixés sur l'organe terminal $\mathbf{x}(1), \dots, \mathbf{x}(r)$,
- 3) Générer les contraintes grâce au modèle géométrique direct. Les contraintes sont de la forme

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{p}, \mathbf{q}(k)), \quad k = \{1 \dots r\}, \quad (5)$$

- 4) Résoudre par la méthode de propagation de contraintes présentée en section III..

Rappelons que pour chaque configuration du robot, le vecteur articulaire $\mathbf{q}(k)$ et le vecteur de position des trois points de l'organe terminal $\mathbf{x}(k)$ sont mesurés avec une certaine précision.

III. Propagation de contraintes sur les intervalles

Le but de cette section est de présenter le principe de la propagation de contraintes que nous allons utiliser pour l'étalonnage du robot.

A. Opérateur de contraction

Pour un ensemble \mathbb{S} , notons $\mathcal{C}_{\mathbb{S}}([\mathbf{n}])$ le plus petit pavé (c'est à dire le produit cartésien de d'intervalles réels) contenant $\mathbb{S} \cap [\mathbf{n}]$. $\mathcal{C}_{\mathbb{S}}$ est appelé *opérateur de contraction* associé à l'ensemble \mathbb{S} . Nous allons maintenant voir sur deux exemples simples les opérateurs de contraction. Soit

$$\mathbb{S} = \{(n_1, n_2, n_3) \mid n_3 = n_2 - n_1\}, \quad (6)$$

si nous prenons les pavés $[a]$ et $[b]$ suivant

$$[a] = [1, 2] \times [1, 6] \times [1, 3], \quad (7)$$

$$[b] = [4, 7] \times [2, 6] \times [1, 3], \quad (8)$$

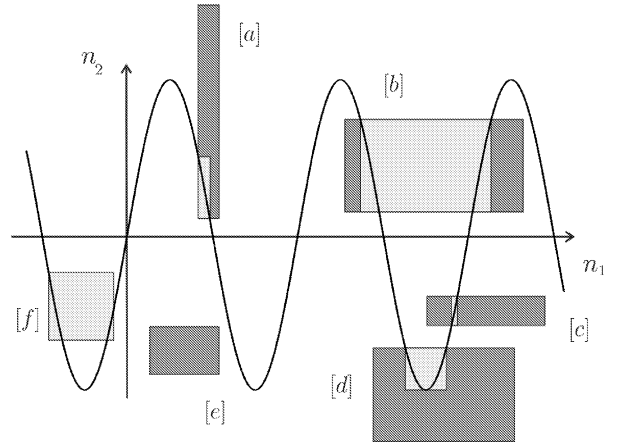


Figure 4 – Opérateur de contraction de la contrainte sinus.

alors

$$\mathcal{C}_{\mathbb{S}}([a]) = [1, 2] \times [2, 5] \times [1, 3], \quad (9)$$

$$\mathcal{C}_{\mathbb{S}}([b]) = [4, 5] \times [5, 6] \times [1, 2], \quad (10)$$

(voir figure 3 pour interprétation géométrique). Le calcul associé peut être effectué automatiquement à l'aide du calcul par intervalle

Des contracteurs associés à des contraintes trigonométriques peuvent aussi être envisagés. Par exemple, si

$$\mathbb{S} = \{(n_1, n_2) \mid n_2 = \sin(n_1)\},$$

et si les pavés $[a], \dots, [f]$ et $[e]$ sont représentés en gris foncé sur la figure 4, alors les $\mathcal{C}_{\mathbb{S}}([n_1] \times [n_2])$ sont en gris clair.

Appliquer l'opérateur de contraction $\mathcal{C}_{\mathbb{S}}$ sur un pavé de recherche $[n_1] \times [n_2]$ permet d'obtenir le pavé minimal contenant l'ensemble des points appartenant à \mathbb{S} . Sur la figure 4 les cas de contraction suivants sont exposés,

- le cas où le pavé de recherche $[e]$ n'intersecte aucun point de la contrainte sinus

$$\mathcal{C}_{\mathbb{S}}([e]) = \emptyset, \quad (11)$$

- celui où le pavé de recherche est déjà le pavé minimal encadrant \mathbb{S}

$$\mathcal{C}_{\mathbb{S}}([f]) = [f], \quad (12)$$

- les cas où la contraction s'effectue suivant uniquement un axe pour les pavés $[b]$ et $[c]$,
- et les cas où les deux axes n_1 et n_2 des pavés $[a]$ et $[d]$ sont contractés

B. Propagation de contrainte

Pour illustrer la propagation de contrainte, considérons les deux contraintes suivantes

$$c_1 : n_1 = \sin(n_2), \quad (13)$$

$$c_2 : n_1 = n_2^3, \quad (14)$$

et cherchons des intervalles, si possible petits, qui contiennent toutes les valeurs possibles pour n_1 et n_2 .

A chacune de ces contraintes, nous associons l'ensemble des valeurs pour n_1, n_2 qui satisfont ces contraintes. C'est-à-dire

$$\mathbb{S}_1 = \{(n_1, n_2) | n_1 = \sin(n_2)\}, \quad (15)$$

$$\mathbb{S}_2 = \{(n_1, n_2) | n_1 = (n_2)^3\}. \quad (16)$$

Les opérateurs de contraction $\mathcal{C}_{\mathbb{S}_1}$ et $\mathcal{C}_{\mathbb{S}_2}$ associés à ces ensembles, permettent de contracter un pavé $[\mathbf{n}]$

$$[\mathbf{n}] =] - \infty, +\infty[\times] - \infty, +\infty[.$$

Nous avons ainsi

$$\mathcal{C}_{\mathbb{S}_1}([\mathbf{n}]) = [-1, 1] \times] - \infty, +\infty[\quad (17)$$

et

$$\mathcal{C}_{\mathbb{S}_2}(\mathcal{C}_{\mathbb{S}_1}([\mathbf{n}])) = [-1, 1] \times [-1, 1] \quad (18)$$

et encore

$$\mathcal{C}_{\mathbb{S}_2}(\mathcal{C}_{\mathbb{S}_2}(\mathcal{C}_{\mathbb{S}_1}([\mathbf{n}]))) = [-0.84, 0.85] \times [-1, 1].$$

Nous assistons donc pas à pas à la contraction du pavé de recherche jusqu'à l'obtention d'un point fixe du type

$$\mathcal{C}_{\mathbb{S}}([\mathbf{n}]) = [\mathbf{n}]. \quad (19)$$

Le principe de propagation de contrainte est donc de composer itérativement les opérateurs de contraction $\mathcal{C}_{\mathbb{S}_2}$ et $\mathcal{C}_{\mathbb{S}_1}$. Le point fixe contient toutes les solutions possibles pour les n_1, n_2 qui satisfont les contraintes c_1 et c_2 .

C. Décomposition en contraintes primitives

Les contraintes de notre problème d'étalonnage ne sont pas sous la forme de contraintes primitives, il faut donc effectuer la décomposition expliquée ci-dessous.

Soient deux contraintes

$$n_1 = l \cdot \cos(\theta) - m \cdot \sin(\theta), \quad (20)$$

$$n_2 = l \cdot \sin(\theta) + m \cdot \cos(\theta), \quad (21)$$

peut être décomposée de la façon suivante

$$\begin{aligned} a_1 &= \cos(\theta), \\ a_2 &= l \cdot a_1, \\ a_3 &= \sin(\theta), \\ a_4 &= m \cdot a_3, \\ n_1 &= a_2 - a_4, \end{aligned} \quad (22)$$

pour la première contrainte et pour la deuxième

$$\begin{aligned} a_5 &= \cos(\theta), \\ a_6 &= m \cdot a_5, \\ a_7 &= \sin(\theta), \\ a_8 &= l \cdot a_7, \\ n_2 &= a_8 + a_6. \end{aligned} \quad (23)$$

Pour chacune de ces contraintes primitives, des variables intermédiaires sont introduites. Elles permettent de transmettre l'information entre chaque contrainte primitive. Ces variables sont instanciées avec des intervalles infinis $] - \infty, +\infty[$.

Cette décomposition produit une liste de contraintes primitives mais cette liste comporte de nombreuses redondances.

Dans l'exemple précédent, le cosinus de la variable θ apparaît deux fois avec les variables temporaires a_1 et a_5 . Afin d'éviter d'appliquer deux fois l'opérateur de contraction, une génération spécifique des contraintes doit être effectuée.

Cette compilation des contraintes (voir [1]) permet de minimiser le nombre de calculs. Pour les problèmes de grande dimension, la suppression de ces opérations inutiles permet un gain de temps lors de la résolution.

Pour le problème traité dans cet article, le nombre de contraintes primitives après compilation est de 20558.

IV. Paramètres de simulation

Pour la génération des contraintes, nous avons choisi les paramètres géométriques avec les valeurs nominales données par la table suivante :

j	α_j	d_j	θ_j^o	r_j
0	-	-	$\frac{\pi}{2}$	0.5
1	0.1	0	0	0
2	$-\frac{\pi}{2}$	0	0	0
3	0	0.5	0	0
4	$\frac{\pi}{2}$	0	0	0.5
5	$-\frac{\pi}{2}$	0	0	0
6	$\frac{\pi}{2}$	0	-	-

Les trois points fixés sur l'organe terminal sont repérés par leurs coordonnées b_x^i , b_y^i et b_z^i dans

le repère lié à l'outil. Ces coordonnées sont des paramètres géométriques pour l'étalonnage et sont choisies pour la simulation avec les valeurs de la table suivante

i	b_x^i	b_y^i	b_z^i
1	0.1	0.2	0.1
2	0.1	0.1	0.2
3	0.2	0.1	0.1

Les configurations (\mathbf{x}, \mathbf{q}) que nous utilisons pour la simulation sont obtenues de la manière suivante. Nous prenons une configuration aléatoire du vecteur \mathbf{q} , où chaque $q_i \in [-\pi, \pi]$. Nous effectuons une simulation du modèle direct avec les valeurs nominales de \mathbf{p} , il nous donne les positions \mathbf{x} des trois points de l'organe terminal.

Pour cinquante configurations \mathbf{q} différentes, nous avons ainsi obtenu par simulation un ensemble de cinquante positions \mathbf{x} .

Afin de prendre en compte les incertitudes de mesure présentes dans les mesures réelles et les erreurs de modélisation, nous avons ajouté à chaque composante des vecteurs \mathbf{q} et \mathbf{x} un bruit aléatoire uniforme compris dans l'intervalle d'incertitude sur la mesure.

Nous obtenons ainsi des intervalles pour chaque configuration (\mathbf{x}, \mathbf{q}) contenant les points de simulation sans avoir d'information sur la localisation de ces points dans les intervalles obtenus.

Exemple : La variable articulaire q_1 est tirée de façon aléatoire dans son domaine de variation $[-\pi, \pi]$. La coordonnée a_x^1 du premier point de l'organe terminal est calculée par simulation. Pour simuler des valeurs réelles, les valeurs calculées sont bruitées tel que

$$\tilde{a}_x^1 = a_x^1 + e_x, \quad (24)$$

où e_x appartient à l'intervalle $[-10^{-4}, 10^{-4}]$ correspondant à l'incertitude de mesure sur les capteurs mesurant \mathbf{x} , et

$$\tilde{q}_1 = q_1 + e_q, \quad (25)$$

où e_q appartient à l'intervalle $[-10^{-5}, 10^{-5}]$ correspondant au pas du codeur générant \mathbf{q} .

Nous obtenons donc les intervalles pour a_x^1 et q_1 qui sont utilisés pour la simulation de l'étalonnage géométrique

$$[a_x^1] = [\tilde{a}_x^1 - 10^{-4}, \tilde{a}_x^1 + 10^{-4}], \quad (26)$$

$$[q_1] = [\tilde{q}_1 - 10^{-5}, \tilde{q}_1 + 10^{-5}]. \quad (27)$$

Pour chaque paramètre que nous cherchons à étalonner, il faut déterminer un intervalle de recherche. Ces domaines de recherche correspondent à la connaissance que nous avons sur les paramètres avant de réaliser l'étalonnage.

Par exemple, nous avons peu de connaissance sur les paramètres de la base qui correspondent à l'installation du robot dans le repère atelier. De même les paramètres b^i sont mal connus. Nous prenons donc un domaine de recherche de $0.1m$ sur les longueurs et de $0.17rad$ pour les angles autour des valeurs nominales. Pour les paramètres du robot lui-même nous disposons d'une meilleure information, ce sont des paramètres constructeurs, la recherche s'effectue sur un domaine de recherche de $0.02m$ pour les longueurs et de $0.035rad$ pour les angles.

V. Résultats

Le premier test réalisé pour l'étalonnage de *tous les paramètres* de \mathbf{p} n'a pas permis d'obtenir une contraction des domaines de recherche. La méthode d'étalonnage doit être améliorée.

Pour le deuxième test, l'étalonnage est limité aux longueurs mal connues et aux longueurs non nulles des paramètres du robot. Tous les autres paramètres non étalonnés sont supposés pour cet essai connus avec une précision de 10^{-5} .

La contraction des domaines de recherche est reportée dans la table suivante

	domaines de recherche	domaines obtenus
r_0	[0.4, 0.6]	[0.488772, 0.501644]
d_1	[0, 0.1]	[0, 0.000964102]
r_1	[0, 0.1]	[0, 0.00911161]
d_3	[0.49, 0.51]	[0.497796, 0.50185]
r_4	[0.49, 0.51]	[0.499031, 0.502242]
b_x^1	[0, 0.2]	[0.0993651, 0.10123]
b_y^1	[0.1, 0.3]	[0.199615, 0.200586]
b_z^1	[0, 0.2]	[0.0992392, 0.100921]
b_x^2	[0, 0.2]	[0.0993244, 0.101355]
b_y^2	[0, 0.2]	[0.099531, 0.100757]
b_z^2	[0.1, 0.3]	[0.199172, 0.200965]
b_x^3	[0.1, 0.3]	[0.199372, 0.201203]
b_y^3	[0, 0.2]	[0.0995955, 0.100753]
b_z^3	[0, 0.2]	[0.0991525, 0.100918]

L'arrêt de la propagation s'effectue quand la contraction du vecteur d'intervalles représentant les variables n'est plus que d'un facteur 10^{-5} . Le résultat a été obtenu en $8.34 s$ sur un pentium

1GHz. Ce temps correspond à 46 contractions suivant l'ensemble des contraintes primitives.

Nous obtenons donc pour les 15 paramètres que nous avons étalonnés un intervalle autour des valeurs nominales que nous avons prises pour la simulation. La taille de ces intervalles varie entre $0.01m$ et $0.001m$. La recherche d'intervalles encadrant de façon plus fine les valeurs des paramètres doit être envisagée.

VI. Conclusion

Dans cet article, nous avons présenté les premiers résultats concernant le problème de l'étalonnage géométrique des robots. Notre contribution principale est d'avoir, pour la première fois, réalisé cet étalonnage par méthode de propagation de contrainte, méthode dont l'un des avantages est de ne perdre aucune solution.

Cette propriété permet d'obtenir une information sur l'intervalle d'incertitude autour de la valeur du paramètre recherché. Mais cette méthode ne donne pas encore la précision nécessaire pour la connaissance de l'ensembles des paramètres du robot.

Afin de pallier ce problème, l'ajout de contraintes redondantes permettra de mieux cerner la solution. Une autre possibilité est la mise en place de la notion de contrainte globale. Les opérateurs de contraction associés à ces contraintes globales permettent d'obtenir le plus petit pavé encadrant l'ensemble des solutions. Ces opérateurs permettront donc d'obtenir plus rapidement un meilleur encadrement pour nos paramètres à estimer.

REFERENCES

- [1] A. Aho, R. Sethi, and J. Ullman. *Compilateurs Principes, techniques et outils*. Informatique intelligence artificielle. InterEditions, 1995.
- [2] C. Bliet and All. *Algorithms for Solving Nonlinear Constrained and Optimisation Problems: State of the Art*. Coconut project, 2001.
- [3] L. Everett. Forward calibration of closed loop jointed manipulators. *The Int. J. of Robotics Research*, 8(4):85–91, August 1989.
- [4] J. Hollerbach. A survey of kinematic calibration. *The Robotics Review, MIT Press Cambridge*, (1):207–242, 1989.

- [5] 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.
- [6] W. Khalil and S. Besnard. Comparison study of the geometric parameters calibration methods. *Int. Journal of robotics and Automation*, 15(2):56–67, 2000.
- [7] W. Khalil and E. Dombre. *Modélisation, identification et commande des robots, Collection Robotique*. Hermès, Paris, 1999.
- [8] W. Khalil and P. Lemoine. Gecaro: a system for the geometric calibration of robots. *Revue APII-JESA*, 33, 1999.
- [9] M. VanEmden. Algorithmic power from declarative use of redundant constraints. *Constraints*, 4(4):363–381, 1999.
- [10] D. Waltz. Generating semantic descriptions from drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, NY, 1975.