



HAL
open science

De l'amélioration des performances des robots manipulateurs

David Daney

► **To cite this version:**

David Daney. De l'amélioration des performances des robots manipulateurs . Robotique [cs.RO].
Université de Nice - Sophia Antipolis, 2016. tel-01418313

HAL Id: tel-01418313

<https://inria.hal.science/tel-01418313v1>

Submitted on 16 Dec 2016

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.

UNIVERSITÉ DE NICE SOPHIA ANTIPOLIS

De l'amélioration des performances des robots manipulateurs

On improving the performances of robot manipulators

David DANÉY

Mémoire pour obtenir le diplôme d'

HABILITATION À DIRIGER DES RECHERCHES

Sciences et Technologies de l'Information et de la Communication

Jury :

Rapporteurs :

Nicolas Andreff, Professeur Université de Franche-Comté

Luc Jaulin, Professeur ENSTA-Bretagne

Philippe Wenger, Directeur de recherche CNRS

Examineurs :

Philippe Gorce, Professeur Université de Toulon (Président du jury)

Jean-Pierre Merlet, Directeur de recherche INRIA

1 Juillet 2016, à Inria Sophia Antipolis

Table des matières

1	Introduction générale	4
1.1	Présentation	4
1.2	Parcours	5
1.3	Problématiques et positionnement général des contributions	6
1.4	Introduction aux contributions et plan du mémoire	7
1.4.1	Les robots étudiés et développés	7
1.4.2	Analyse par intervalles pour la robotique : motivation, outils et contributions	8
1.4.3	Conception appropriée des robots	9
1.4.4	Étalonnage des robots parallèles	9
2	Les robots étudiés	12
2.1	Introduction	12
2.2	Modèles de manipulateur à architecture parallèle	12
2.3	Les plate-formes de Gough	13
2.3.1	Hexapode 300, CMW, Constructions Mécaniques des Vosges	15
2.3.2	Plateforme Deltalab	15
2.3.3	Satellite d'observation Thales	15
2.4	Les robots à câbles	18
2.4.1	ReelAx8	18
2.4.2	CoGiRo	20
2.4.3	Marionet	21
3	Analyse par intervalles pour la robotique	23
3.1	Quelques notions de base de l'arithmétique et de l'analyse par intervalles	24
3.1.1	Arithmétique par intervalles	25
3.1.2	Propriétés algébriques	25
3.1.3	Évaluation : les extensions par intervalles	25
3.1.4	Résolution de systèmes par intervalles	27
3.1.5	Caractérisation des sous-espaces de recherche	30
3.2	Approches par intervalles pour la robotique	33
3.2.1	Modélisation et Contraintes épaissies	33
3.2.2	Formulation d'un problème	34
3.2.3	Adaptations des outils	35
3.3	Résolution de système, contributions à l'analyse par intervalles	38
3.3.1	Cas bien contraint	38
3.3.2	Cas sur-contraint	38

3.3.3	Cas sous-contraint	39
3.3.4	Cas des valeurs/vecteurs propres ou des valeurs singulières d'une matrice intervalle	39
3.3.5	Tests pour les boites intérieures pour les matrices symé- triques	42
3.4	Diffusion des contributions dans le domaine de l'analyse par in- tervalles	43
3.5	Conclusion	44
4	Conception appropriée des robots manipulateurs	51
4.1	Principes de la conception appropriée	51
4.1.1	Problématique	51
4.1.2	Principe de la méthode	53
4.2	Analyse par intervalles de performance des robots	54
4.2.1	Représentation par intervalles des propriétés des robots et conception approprié	54
4.2.2	Caractérisation de sous-espace de travail en fonction des propriétés de robots	55
4.3	Propriétés étudiées	56
4.3.1	Prise en compte de contraintes géométriques	57
4.3.2	Prise en compte de contraintes variationnelles et statiques	62
4.3.3	Vérification des performances statiques	65
4.4	Conception de robot	67
4.5	Conclusion	68
5	L'étalonnage des robots manipulateurs	72
5.1	Introduction	72
5.1.1	Problématiques de l'étalonnage des manipulateurs parallèles	72
5.1.2	Contexte	73
5.1.3	Origine des erreurs	74
5.1.4	Positionnement des recherches	76
5.2	Créer une information redondante sur l'état interne du robot . .	78
5.2.1	Obtenir une redondance métrologique du système	78
5.2.2	Problématique	78
5.2.3	Contributions	79
5.3	Modélisation des équations de l'étalonnage	81
5.3.1	Modèle implicite simplifications et erreurs	81
5.3.2	Élimination algébrique	83
5.3.3	Simplification d'un modèle augmenté des équations d'éta- lonnage	84
5.3.4	Modèle simplifié mais épaissi des équations d'étalonnage .	85
5.4	Méthode d'identification des paramètres	86
5.4.1	Approche classique aux moindres carrés non-linéaires . . .	86
5.4.2	Méthodes algébriques	89
5.4.3	Méthodes par intervalles	93
5.5	Conditions d'étalonnage et amélioration de l'identification des pa- ramètres	96
5.5.1	Conditions d'étalonnages	98
5.5.2	Identifiabilité et choix des configurations de mesures . . .	99
5.6	Conclusion	102

6	Conclusion générale	108
7	Curriculum Vitae	111
7.1	Parcours et activités	111
7.1.1	Responsabilités récentes	111
7.1.2	Responsabilités activités Collectives	112
7.1.3	Parcours de recherche	113
7.1.4	Formation	113
7.2	Encadrement	114
7.2.1	Encadrement de Doctorants	114
7.2.2	Encadrement de doctorants sur articles communs	114
7.2.3	Post doctorants/ATER	114
7.2.4	Jeunes ingénieurs	114
7.2.5	Étudiants Master	115
7.3	Dissémination	115
7.3.1	Cours	115
7.3.2	Projet étudiants	115
7.3.3	Grand public	116
7.3.4	Développement logiciels	116
7.4	Contrats de recherche et collaborations	116
7.4.1	Projets Internationaux	116
7.4.2	Projets Nationaux	116
7.4.3	Contrats Industriels	117
7.4.4	Collaborations actuelles	117
7.5	Visibilité Nationale - Internationale	117
7.5.1	Expertises éditoriales	117
7.5.2	Relecture	117
7.5.3	Participation des comités	117
7.5.4	Participation à des jury des thèses	118
7.5.5	Organisation Conférences	118
7.5.6	Missions Expertises	118

Chapitre 1

Introduction générale

1.1 Présentation

Ce document décrit les travaux de recherches auxquels j'ai participé, que j'ai entrepris ou que j'ai dirigés, principalement au sein de l'équipe Coprin (2002-2013) du Centre de Recherche Inria Sophia Antipolis Méditerranée.

L'essentiel de mes contributions ont traité de *l'amélioration des performances des robots manipulateurs*. Dans ce domaine, j'ai co-encadré 7 doctorants dont 2 en tant qu'encadrant principal, 4 post-doctorants, ainsi que des stagiaires et ingénieurs que j'ai formés à nos méthodes. J'ai publié plus de 70 articles de journaux et de conférences internationales, essentiellement dans les domaines relatifs à ces thématiques.

J'ai effectué un travail régulier de relecture dans des journaux de robotique mais aussi de mécanique, de mathématiques appliquées et d'intelligence artificielle. J'ai mis à disposition de la communauté mon expertise en participant à différents comités scientifiques, de programme, de sélection mais aussi à des jurys de thèse ou de concours. Je suis actuellement membre du bureau Aquitaine Robotics, cluster regional à fort caractère industriel, et président du comité de labellisation de ce cluster, ce qui me permet de m'intéresser à l'ensemble des projets de robotique en Aquitaine. J'effectue aussi un travail d'animation au sein du pôle Aerospace Valley sur des questions de robotique industrielle. J'ai participé à - ou coordonné - des travaux dans le cadre de projets variés (ANR, projets européens, projets Inria d'envergure nationale et de développement technologique). Ces dernières années, j'ai participé à l'organisation de plusieurs conférences (World Congress Iftomm 07, IROS 08, ASROB 15) et workshops dont un récurrent à IROS.

Tout au long de ma carrière, j'ai entretenu des relations étroites avec des industriels (CMW, Alcatel, Thales, Tecnalía, AIO, Explosense, Safran...) dont les projets ont, en partie, guidé les développements applicatifs de mes recherches.

L'ensemble de ces activités ont influencé mes contributions et orientations. Mais le contexte et mes collaborations scientifiques ont marqué ces choix.

1.2 Parcours

Ma thèse a porté sur l'étalonnage des robots parallèles, au sein des projets PRISM (1996-1997) puis SAGA (1997-1999) de l'Inria Sophia Antipolis. Elle a bénéficié des très hautes expertises au sein de ces équipes en matière de géométrie algorithmique puis algébrique. Jean-Pierre Merlet, alors directeur de ma thèse, avait montré l'intérêt de bien maîtriser ces domaines pour l'étude des robots parallèles notamment pour la résolution des modèles géométriques, pour le calcul des espaces de travail ou la localisation des singularités ; il préparait, d'ailleurs, une première synthèse qui sert toujours de référence dans le domaine [Mer06].

Cet environnement a fortement influencé mes contributions sur la modélisation des équations d'étalonnage. La façon dont sont prises les informations sur l'état interne du robot induit la forme symbolique des modèles d'étalonnage. Ainsi, nous pouvons modifier cette forme en jouant sur le type de mesures exero/proprioceptives ou de contraintes mécaniques limitant les degrés de liberté du robot. Dans ce contexte, avec l'aide de Ioannis Emiris de l'université d'Athènes alors chercheur à l'Inria, nous avons cherché à obtenir des systèmes de degrés faibles considérant le moins de variables possibles. Pour cela, nous avons utilisé des techniques d'élimination algébrique basés sur les résultants creux.

Parallèlement, j'ai pu montrer comment améliorer la qualité numérique de ces systèmes en choisissant, dans la mesure du possible, les valeurs des coefficients des systèmes considérés. Cette problématique reste encore une question centrale de l'étalonnage des robots.

Ces questions numériques/symboliques étaient déjà au cœur de mes préoccupations lorsque j'ai rejoint l'équipe-projet Coprin de l'Inria Sophia Antipolis, en fin 2002. Dirigée par Jean-Pierre Merlet, elle s'intéressait à l'étude des robots manipulateurs et plus particulièrement à l'analyse et à la synthèse de l'architecture mécanique des robots parallèles. Mon projet de recherche portait alors sur l'étude des robots à câbles afin d'étendre les résultats obtenus pour les robots parallèles en matière de conception et d'étalonnage. La spécificité de cette équipe est d'avoir su allier son expertise en matière de théorie des mécanismes et le développement d'outils d'analyse par intervalles afin de proposer des solutions innovantes et pertinentes en robotique notamment. Ce projet se dirigera progressivement vers la robotique d'assistance pour devenir Héphaistos en 2014.

De 2003 à 2013, j'ai mis en oeuvre mon projet de recherche, en participant aux orientations du projet en matière de robotique avec Jean-Pierre Merlet, Yves Papegay et Odile Pourtallier et en co-construisant avec Michel Rueher, Gilles Trombettoni et Bertrand Neuveu notre expertise en matière d'approche par intervalles. Tout au long de ces années, j'ai participé à l'encadrement des doctorants du projet, incitant les étudiants de robotique à utiliser l'analyse par intervalles (J. Hubert, M. Harshe) et, inversement, à fournir des applications réalistes aux étudiants en informatique (C. Grandon, G. Chabert, A. Goldsztejn).

Au cours de ces années, deux projets collaboratifs ont particulièrement contribué aux travaux décrits dans ce document. Tout d'abord, le projet ANR CoGiRo (2010-2013), avec le Lirimm, le Lasmea et Technalia qui a proposé de concevoir un robot à câbles de grande envergure. Notre contribution a porté sur l'étalonnage de ce type de structure au travers des résultats obtenus par Julien Alexandre dit Sandretto dans sa thèse [ADS13]. Le second projet est une colla-

boration à long terme avec Thales sur le conception d'une architecture parallèle de type 6 \underbrace{P}_{US} comme structure mécanique d'un télescope actif d'observation terrestre. L'originalité de notre proposition est le déploiement possible du manipulateur dans l'espace au moyen de rubans de métal enroulables. Ils sont utilisés en place des segments du robot et une fois déployés, leur courbure transversale assure la rigidité de la structure. En 2013, le groupe de recherche de Thales, dirigé par Frédéric Falzon, nous à demandé de qualifier le modèle de leur prototype, ce qui a été réalisé par Thibault Gayral dans sa thèse [Gay13].

En 2011-2014, j'ai coordonné l'Inria Project Lab "Personally Assisting Living" (PAL). Cette infrastructure, financée par la direction de la recherche d'Inria, propose des solutions technologiques de type capteurs intelligents, robotiques ou domotiques afin d'améliorer la qualité de vie des personnes fragiles et leur entourage. Plus particulièrement ces solutions visent à favoriser l'autonomie des personnes âgées et leur maintien à domicile par la sécurisation, la préservation ou le rétablissement des fonctions nécessaires à leur vie quotidienne. Cette action regroupait 10 équipes Inria (en robotique ou spécialisées dans le monitoring des personnes) associées à des partenaires de recherche (technologiques, médicaux, sciences sociales) travaillant ensemble autour de trois principaux thèmes : l'assistance à la mobilité, l'évaluation de l'état de fragilité des personnes et l'analyse de comportement.

Depuis septembre 2013, j'ai intégré le Centre de Recherche Inria Bordeaux Sud-ouest où je m'intéresse aux systèmes pour l'assistance aux opérateurs industriels. Cette nouvelle voie de recherche est, bien évidemment, influencée par ces deux dernières expériences. Elle répond à un enjeu important pour la robotique : comment concevoir des robots qui considèrent des contraintes humaines. Dans ce cadre, je m'intéresse aux gestes techniques des opérateurs industriels, leur analyse permettra à terme de mieux concevoir les robots d'assistance industriel.

1.3 Problématiques et positionnement général des contributions

Mon parcours scientifique est articulé autour de l'amélioration des performances des robots manipulateurs [SK08] et plus particulièrement des robots à architecture parallèle [Mer06], composés de segments rigides ou flexibles comme les robots à câbles. Pour ce type de manipulateur, l'organe terminal est relié à une base par plusieurs chaînes cinématiques indépendantes.

L'intérêt des robots manipulateurs (sériels ou parallèles) réside dans leur capacité à déplacer une charge le long d'une trajectoire dont les caractéristiques sont induites par les choix de conception du système mécanique utilisé. En effet, le type de mouvement produit et la qualité de sa réalisation, est en grande partie lié aux performances statiques et dynamiques du robot qui sont fonction de l'architecture mécanique choisie, des dimensions des éléments mécaniques, des tolérances de fabrication et d'assemblage, et du type de capteurs et d'actionneurs utilisés.

Si le choix d'une architecture, la qualité des éléments choisis, de leur fabrication et de leur assemblage jouent sur les performances globales du système, les dimensions mêmes du robot sont des paramètres essentiels qui déterminent ses propriétés.

Mes travaux de recherche visent à *améliorer les performances du robot*, en s'intéressant à :

- En *amont* de la construction du robot : *La conception des robots* qui détermine les dimensions des éléments du robot afin de garantir certaines de ses propriétés données par un cahier des charges.
- En *aval* de la construction du robot : *L'étalonnage des robots* détermine la valeur des dimensions réelles, mal connues à causes des erreurs de fabrication et d'assemblage. Plus exactement, nous chercherons ici à identifier les paramètres d'un modèle de robot, au travers des informations fournies par des mesures directes ou indirectes sur son état.

À travers ces deux axes, j'ai cherché à rendre le système mécanique et l'identification de ses paramètres robustes aux incertitudes afin de garantir ses propriétés. Mes contributions utilisent des outils provenant de l'analyse numérique et du calcul formel. De plus, une partie de l'originalité de mes contributions et de sa pertinence vient de l'utilisation de l'**analyse par intervalles** qui permet de prendre en compte et d'unifier la représentation de la variation des paramètres et des différentes incertitudes (tolérances, erreurs de mesures) dans un même formalisme. Ces techniques fournissent des résultats intéressants, qui sont repris dans différentes communautés notamment en robotique. Cependant, elles restent encore à développer et j'ai été amené à proposer de nouveaux outils pour la représentation des données, la résolution de systèmes de contraintes ou l'analyse matricielle. L'une des difficultés est aussi de spécifier ces outils dans un cadre robotique.

L'intérêt de ces travaux doit être démontré sur des manipulateurs réels. Tout au long de ma carrière, j'ai participé à des projets et bénéficié de collaborations industrielles et académiques qui m'ont permis de mettre en œuvre expérimentalement les méthodes que j'ai proposées. De plus, il m'a été donné de participer, au sein de l'équipe-projet Coprin, à l'élaboration d'un certain nombre de prototypes de robots à câbles, rassemblés dans une famille dénommée *Marionet*.

Ce document traite de ces problématiques. Il est articulé autour de trois thèmes principaux :

1. L'utilisation et le développement de méthodes d'analyse par intervalles. J'ai donc contribué à développer des outils mathématiques et logiciels afin de répondre aux enjeux suivants.
2. Les méthodes de conception dimensionnelle de robots manipulateurs. Nous présenterons une méthodologie de conception fondée sur l'analyse des propriétés des robots et de leurs espaces de travail.
3. Les méthodes d'étalonnage des robots parallèles et mes contributions sur les différentes étapes du processus d'identification des paramètres de ces robots.

1.4 Introduction aux contributions et plan du mémoire

1.4.1 Les robots étudiés et développés

En robotique, comme dans beaucoup de domaine, la validation d'un résultat passe par l'expérimentation. La modélisation du réel fournit les plus grandes dif-

difficultés auxquelles j'ai été confronté et qui ont orienté mes recherches. Nous aurons dans ce chapitre une vue d'ensemble des robots étudiés et de leurs modèles présentés génériquement afin d'introduire les notations et quelques équations qui seront utilisés dans ce mémoire.

Les robots manipulateurs étudiés ont, en grande majorité, une architecture mécanique de type plate-forme de Gough. Le choix de ce robot comme support de mes travaux est lié à son espace de travail de 6 degrés de liberté (ddl) qui induit une grande dimension aux problèmes traités et qui nécessite une algorithmie adaptée et efficace. Les résultats obtenus sont facilement transposable aux cas de robots à moindre nombre de degrés de liberté.

Nous présenterons aussi la famille de robots à câbles *Marionet* qui a été développée au sein du projet Coprin. L'utilisation de câbles induit de nouveaux problèmes qui occupent fortement la communauté à l'heure actuelle.

1.4.2 Analyse par intervalles pour la robotique : motivation, outils et contributions

L'analyse par intervalle est un outil puissant de calcul ensembliste. Elle est particulièrement intéressante pour gérer deux challenges en robotique : la gestion des incertitudes et l'étude des systèmes globalisée pour un ensemble de points de fonctionnement.

La gestion des incertitudes est un enjeu majeur dans tous les domaines scientifiques. Dans ce mémoire, nous identifions trois grands types d'incertitudes qui ferons l'objet de notre attention :

- les incertitudes liées aux mesures : ces erreurs peuvent avoir des distributions variées ou mal identifiées,
- les incertitudes liées aux données de modélisation : typiquement des tolérances sur les valeurs des dimensions du robot,
- les incertitudes sur le modèle du robot : il n'est pas possible de prendre en compte l'ensemble des phénomènes physiques auquel est soumis le système ; alors les modèles considérés ne sont qu'une approximation simplificatrice de la réalité.

Une représentation par intervalles permet d'unifier la représentation des variations des mesures, des données et des paramètres du système dans un même formalisme. Cette approche ensembliste nécessite pour la résolution des modèles des outils d'*analyse par intervalles* et de *programmation par contraintes*.

Le second intérêt d'une approche par intervalles est sa capacité à modéliser les comportements d'un système dans un ensemble d'états au travers de l'extension par intervalles de son modèle scalaire. Ceci permet de gérer de multiples problèmes en robotique notamment d'étudier les propriétés des manipulateurs sur l'intégralité d'un espace de travail donné et non plus seulement en quelques points.

Afin de bien comprendre l'intérêt de mes contributions, nous présenterons l'analyse par intervalles, ses propriétés et les méthodes qui seront utilisés pour les développements en robotique. Ce domaine est encore jeune, une quarantaine d'année, et beaucoup de problèmes, même basiques, n'ont pas de solutions sur étagère. Il a donc été nécessaire, dans un objectif applicatif en robotique, de s'intéresser à des méthodes de résolutions ou d'algèbre linéaire adaptées aux intervalles. Nous présenterons donc des contributions sur l'évaluation par intervalles des modèles robotiques, la résolution des systèmes d'équations bien- ou

sur-contraints, et des calculs de bornes sur les valeurs propres de matrices par intervalles.

1.4.3 Conception appropriée des robots

J'ai contribué à la formalisation et au développement d'une méthodologie de conception nommée *Conception Appropriée* mise en place au sein de l'équipe-projet Coprin sous la direction de Jean-Pierre Merlet. Cette méthode est notamment utilisée pour la conception dimensionnelle, c'est-à-dire pour donner des valeurs numériques aux paramètres géométriques caractéristiques des dimensions des corps qui composent le robot.

La conception est vue ici avant tout comme un problème de faisabilité et non comme un problème d'optimalité. En effet, un cahier des charges définissant les propriétés souhaitées pour un manipulateur est généralement fourni sous forme de contraintes bornées. Nous cherchons alors à identifier un ensemble continu de valeurs pour les paramètres géométriques qui vérifient un système d'égalités/inégalités dont les coefficients sont bornés.

Là encore, nous utilisons l'analyse par intervalle comme outil de modélisation et de résolution de systèmes de contraintes. Cette méthodologie est originale et, afin de traiter les problèmes robotiques, nécessite d'adapter ou de développer des outils numériques propres.

Nous verrons comment ce problème se ramène à déterminer plusieurs type d'espaces de travail vérifiant chaque propriété pour une dimension scalaire donné du robot : l'espace total des positions vérifiant toutes les propriétés étant alors à l'intersection de ces ensembles. Cette première étape correspond alors à une analyse des performances du robot. Une seconde étape est ensuite nécessaire pour vérifier qu'un ensemble de dimensions possibles réalise les propriétés souhaitées. Cette seconde étape est possible par une extension aux intervalles des paramètres géométriques du robot.

1.4.4 Étalonnage des robots parallèles

Nous cherchons ici à améliorer les performances d'un manipulateur existant. En effet, lorsqu'on mesure ses performances réelles, comme par exemple sa précision de positionnement, elle ne correspond pas avec ce qui peut être calculé au travers d'un modèle théorique fourni lors de la phase précédente de conception.

Ceci est dû au cumul de nombreuses erreurs d'origines diverses :

- géométriques, comme par exemple les tolérances de fabrication et d'assemblage ou les positionnement des repères ;
- statiques, notamment l'influence des forces internes sur la structure qui ne peut pas être infiniment rigide ;
- environnementales comme la température, etc.
- les erreurs de modèles qui ne peuvent être qu'une approximation d'une réalité plus complexe.

Dans ce mémoire, nous nous concentrons sur ces erreurs lorsque le robot est positionné dans un état statique. Ceci nous permet de négliger les erreurs dues aux aspects dynamiques : ces questions faisant l'objet de travaux importants en dehors de mes contributions.

Classiquement, l'étalonnage est présenté comme l'identification de ces erreurs ce qui n'est pas exact. En effet, ceci ne peut être possible que si le modèle

considéré est parfait et que tous les phénomènes physiques auxquels sont soumis le robot sont mesurés directement ou indirectement. Nous préférons définir l'étalonnage comme l'identification de valeurs pour les paramètres d'un modèle considéré afin qu'il approxime au mieux le comportement du robot. Celui-ci est mesuré directement ou indirectement par ses seuls capteurs proprioceptifs (méthodes d'auto-étalonnage) ou en les couplant avec des informations extéroceptives (méthodes d'étalonnage).

Nos contributions sont alors de rendre cette identification robuste aux erreurs de mesures par :

- une analyse symbolique des équations d'étalonnage : le but est de proposer des traitements algébriques de ces systèmes,
- une analyse numérique des équations d'étalonnage : les conditions d'étalonnage sur les mesures effectuées et les configurations dans lesquelles elles sont pratiquées influencent cette robustesse aux erreurs.
- des méthodes d'identification qui prennent en compte les erreurs de modèles.

Pour chacun des ces trois points, nous présenterons les méthodes que nous avons mis en oeuvre.

Bibliographie

- [ADS13] Julien Alexandre Dit Sandretto. *Étalonnage des robots à câbles : identification et qualification*. Theses, Université Nice Sophia Antipolis, September 2013.
- [Gay13] Thibault Gayral. *Étalonnage d'un instrument d'observation spatial actif*. PhD thesis, Université Nice Sophia Antipolis, 2013.
- [Mer06] Jean-Pierre Merlet. *Parallel Robots, 2nd Edition*, volume 128. Springer, 2006.
- [SK08] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.

Chapitre 2

Les robots étudiés

2.1 Introduction

Nous présentons ici quelques robots qui ont été conçus ou utilisés afin de servir de support à nos contributions. Ils proviennent de collaborations industrielles (Hexapode 300 pour CMW, Poignet actif pour Thales), académiques (Dentalab avec le Lasmea, ReelAx8, CoGiRo avec le Lirmm et Technalia) ou de projets (ANR CogiRo par exemple).

Nous avons essentiellement étudié deux grands types de robots à architecture parallèle : les robots parallèles classiques (plateforme de Gough et poignet actif) et les robots à câbles. On les présente brièvement afin d'aider le lecteur à mieux cerner leurs caractéristiques, ce qui les différencie et d'introduire quelques modèles génériques ainsi que leurs notations.

2.2 Modèles de manipulateur à architecture parallèle

Un manipulateur parallèle est composé d'une base fixe et d'un mobile reliés par plusieurs chaînes cinématiques rigides, pour les robots parallèles classiques et par des câbles pour les robots à câbles.

Dans ce mémoire, nous noterons :

- Ω_o le repère de référence associé à la base du robot,
- Ω_c le repère associé au mobile (ou organe terminal),
- $X = [P, R]^t$, les coordonnées généralisées, représentant la position P et l'orientation R de Ω_c exprimées dans Ω_o ,
- $Q = [q_1, \dots, q_m]$, le vecteur représentant les coordonnées articulaires, leurs valeurs étant fournies par un capteur de mesures associé à l'actionneur installé sur l'articulation $i = 1, \dots, m$,
- ξ les paramètres géométriques du robot qui le caractérisent : dimensions des segments ou des plateaux, offsets articulaires etc ...
- Dans certain cas nous noterons $[X, Q]$ le vecteur associé à une configuration du robot si ses modes d'assemblages le nécessitent,
- F les forces et moments appliqués sur le mobile,
- τ les forces et couples appliqués par les actionneurs

En position statique, il existe une relation implicite qui lie les coordonnées articulaires et généralisées ainsi que les paramètres géométriques du robot :

$$\mathcal{F}(X, Q, \xi) = 0 \quad (2.1)$$

Cette relation est donnée, dans des conditions idéales, par les équations de fermeture des chaînes cinématiques. Le modèle géométrique inverse (resp. direct) est noté $F_{mgi}(X, \xi)$ (resp. $F_{mgd}(Q, \xi)$), il fournit les coordonnées articulaires (resp. généralisées) en fonction des coordonnées généralisées (resp. articulaires). Pour obtenir ces modèles, il sera nécessaire de résoudre le système $\mathcal{F}(X, Q) = 0$ en Q ou X . Dans le cas général, les modèles inverses et directs sont susceptibles de fournir plusieurs solutions réelles et complexes : $[Q_1, Q_2, \dots] = F_{mgi}(X, \xi)$, $[X_1, X_2, \dots] = F_{mgd}(Q, \xi)$.

En différenciant le modèle implicite 2.1, par rapport aux trois variables introduites (X, Q, ξ) , nous obtenons les jacobiniennes cinématiques, $J_X = \frac{\partial \mathcal{F}}{\partial X}$, et $J_Q = \frac{\partial \mathcal{F}}{\partial Q}$, ainsi que la jacobienne d'identification $J_\xi = \frac{\partial \mathcal{F}}{\partial \xi}$.

Autour d'un point de fonctionnement, nous aurons donc approximativement :

$$J_X \cdot \Delta X + J_Q \cdot \Delta Q + J_\xi \cdot \Delta \xi \approx 0 \quad (2.2)$$

La jacobienne cinématique inverse sera notée génériquement $J_{inv} = -J_Q^{-1} \cdot J_X$. Pour les robots parallèles J_Q est généralement diagonale ce qui facilite son inversion si elle n'est pas singulière.

Pour des variations nulles des paramètres géométriques nous obtenons donc :

— Le modèle cinématique :

$$J_{inv} \cdot \dot{X} = \dot{Q} \quad (2.3)$$

— Le modèle variationnel :

$$J_{inv} \cdot \Delta X = \Delta Q \quad (2.4)$$

— Le modèle statique :

$$J_{inv}^T \cdot \tau = F \quad (2.5)$$

Dans ce mémoire, nous allons vérifier les propriétés des robots. Elles sont calculées en utilisant les modèles précédents par leur évaluation directe ou par leur résolution en certaine(s) de ses inconnues.

Par exemple, la précision en un point \tilde{X} de l'espace de travail sera donnée par le modèle variationnel en calculant, si possible, $\Delta \tilde{X} = J_{inv}^{-1}(\tilde{X}) \cdot \Delta \tilde{Q}$ pour une erreur donnée sur les mesures articulaires $\Delta \tilde{Q}$. Cette précision est alors à comparer avec une erreur maximum souhaitée et donnée pour le robot noté ΔX_{max} . Ainsi, si $\|\Delta \tilde{X}\| < \Delta X_{max}$ alors le robot est considéré comme suffisamment précis en \tilde{X} .

De façon générique, ces propriétés sont données sous la forme de contraintes d'égalité ou d'inégalité notées $C(X, \xi)$ avec $C(X, \xi) = 0$; $\neq 0$ ou $\underline{c}_r \leq C(X, \xi) \leq \overline{c}_r$; les valeurs de $\underline{c}_r, \overline{c}_r$ étant données par un cahier des charges.

2.3 Les plate-formes de Gough

Une plate-forme de Gough [GW62, Ste65] est représentée sur la Figure 2.1.

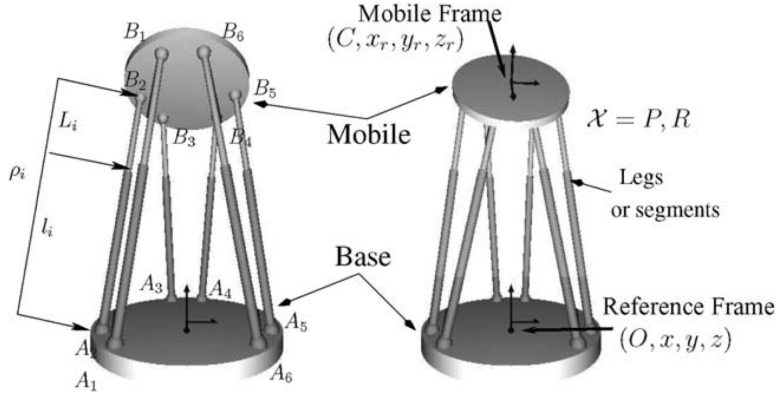


FIGURE 2.1 – Plateforme de Gough

Les plates-formes base et mobile sont reliées par six jambes $i = 1, \dots, 6$ constituées de chaînes cinématiques de type 2RP3R¹. Pour ce type de manipulateur :

- les joints prismatiques sont actifs, actionnés et leurs longueurs sont mesurées,
- les joints rotoïdes sont passifs.

Le modèle de ce robot est généralement simplifié en considérant les articulations cardans (2R) et rotules (3R) comme parfaites. Sous cette hypothèse, la chaîne est considérée comme SS (S pour joint Sphérique). Pour chaque jambe, les attaches des segments avec les plates-formes sont alors ponctuelles et leurs distances (i.e. les longueurs des jambes) sont notées L_i . Ainsi les paramètres géométriques de ce robot sont :

- Les points d'attaches de ces segments avec la base sont notés A_i de coordonnées $a_i = a_i^x, a_i^y, a_i^z$ dans Ω_o ,
- les points d'attaches de ces segments avec le mobile sont notés B_i de coordonnées $b_i = b_i^x, b_i^y, b_i^z$ dans Ω_c
- les longueurs des jambes sont $L_i = \|\mathbf{A}_i \mathbf{B}_i\|$
- une longueur morte l_i est associée à chaque segment $i = 1, \dots, 6$ telle que $L_i = q_i + l_i$, avec les coordonnées articulaires q_i mesurées par un capteur proprioceptif.

Le vecteur ξ des paramètres géométriques du robot est alors $\xi = [\xi_1, \dots, \xi_6]^t$
 $\xi_i = [a_i^x, a_i^y, a_i^z, b_i^x, b_i^y, b_i^z, l_i]^t$ avec pour $i = 1, \dots, 6$,

Les équations implicites sont pour chaque jambe $i = 1, \dots, 6$:

$$\mathcal{F}_i(\mathcal{X}, q_i, \xi_i) = \|P + R \cdot b_i - a_i\| - (q_i + l_i) \quad (2.6)$$

1. R pour joint rotoïde, 1 degré de liberté en rotation, et P pour joint prismatique, 1 degré de liberté en translation

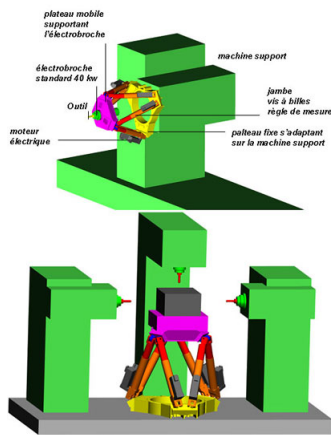


FIGURE 2.2 – Concept machine UGV, CMW



FIGURE 2.3 – Hexapode 300, CMW

2.3.1 Hexapode 300, CMW, Constructions Mécaniques des Vosges

C.M.W. (Constructions Mécaniques des Vosges) a été l'un des premiers constructeurs au monde à avoir proposé d'améliorer un centre d'usinage classique 3 axes par l'ajout, à la place de l'outil, d'un mécanisme visant à fournir un usinage à grande vitesse avec 5 degrés de liberté (un 6ème degré de liberté est utilisé pour augmenter l'efficacité du mécanisme - voir [MPD00]) (Figure 2.2). Cette tête d'usinage supplémentaire, l'Hexapode 300 de CMW (Figure 2.3), est basée sur une plate-forme de Gough. Ce manipulateur a été conçu dans l'équipe-projet SAGA en 1998.

2.3.2 Plateforme Deltalab

Nous avons étudié dans [DACP06], la table de Stewart commercialisée par Deltalab, voir 2.4, et acquise par le LASMEA avec qui nous avons conduit des expérimentations d'étalonnage. Ce robot est une plate-forme de Gough comme décrit ci-dessus. A titre indicatif, la base est de rayon 270 mm, et le mobile de rayon 195 mm; la longueur des six segments peuvent varier entre 345 mm et 485 mm. Le DeltaLab tableau de Stewart a été conçu à des fins académiques et pédagogiques, et n'offre pas la précision d'un robot industriel.

Nous avons étalonné ce robot, dans [DACP06], en utilisant une méthode par intervalle, voir section 5.4.3.

2.3.3 Satellite d'observation Thales

Dans le cadre d'une collaboration industrielle avec Thales, nous avons conçu puis étalonné un robot parallèle de type "poignet actif" utilisé comme architecture mécanique d'un satellite actif d'observation terrestre.

La solution que nous avons proposée à Thales répond à deux enjeux auxquels les architectures modernes des télescopes doivent faire face :



FIGURE 2.4 – Table de Stewart, Deltalab

- Une correction accrue des défauts d'images. En effet, ces défauts sont en majorités du à un mauvais positionnement relatif des miroirs. Actuellement les corrections sont essentiellement logicielles, mais il est possible de rendre le télescope *actif* en utilisant des actionneurs afin de corriger mécaniquement les erreurs de position.
- une diminution de l'encombrement du télescope lors de sa mise en orbite. Le coût de la mise en orbite est conditionné aussi bien par la masse du système que par son volume.

Ainsi, afin de répondre à ces deux enjeux, nous avons proposé d'utiliser un manipulateur parallèles inspiré par [HHK⁺01] qui, à terme, est déployable dans l'espace. Pour cela les segments rigides sont remplacés par des rubans flexibles, enroulés lors des phases de transport puis déroulés dans l'espace, voir figure 2.5. La rigidité des rubans une fois la structure déployée est assurée par sa courbure pré-formatée tel un mètre à mesurer. La correction de positionnement du miroir secondaire (installée sur la partie mobile) relativement au miroir primaire (installée sur la partie mobile) est obtenue par l'actionnement des liaisons prismatiques des chaînes cinématiques de type $6\underline{P}RS$ du "poignet actif".

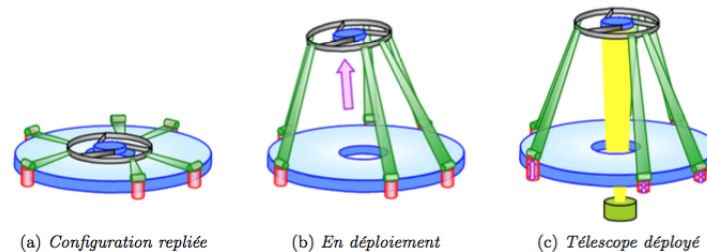


FIGURE 2.5 – Concept de déploiement d'un satellite d'observation terrestre

Le concept a été validé expérimentalement, voir figure 2.6. Ces tests ont mis en évidence les erreurs importantes de positionnement du miroir secondaire (attaché au mobile) par rapport au miroir primaire (attaché à la base). Ceci est dû à l'enroulement des rubans encore peu maîtrisé.

L'actionnement des jambes doit donc servir à corriger les défauts de déploiement aussi bien que les autres défauts de la structure. Mais ils doivent être

2.4 Les robots à câbles

Ces structures sont de même type que les manipulateurs parallèles classiques mais les segments rigides sont remplacés par des câbles. Ceux-ci ne pouvant créer qu'une force de traction, la modélisation et la commande de ces manipulateurs deviennent beaucoup plus compliquées. En effet, il est nécessaire soit d'ajouter des câbles (plus de câbles que de degrés de liberté à l'organe terminal) pour générer des forces en opposition (on parlera de robot pleinement contraint) soit d'utiliser la force d'attraction terrestre pour générer une force verticale, suffisante pour déplacer le mobile dans les directions souhaitées, on parlera alors de grue suspendue.

Les nouvelles problématiques qu'engendrent ces robots à câbles, par exemple voir [MD09], ont conditionné une partie de mes recherches post-doctorales tant sur les aspects de conception que d'étalonnage. Le développement et l'étude de ce type de structure a fait l'objet de mon programme de recherche de recrutement comme chargé de recherche à l'Inria.

J'ai participé au développement d'une famille de robot à câbles (nommés MARIONET) au sein du projet Coprin que nous verrons en sous-section 2.4.3. De plus, une action collaborative importante a structuré les recherches sur ce *nouveau* type d'architecture : l'ANR Cogiro (2010-2013). Son objectif principal était de proposer et de valider des méthodologies innovantes pour concevoir, calibrer et contrôler des robots parallèles à câbles ayant un très grand espace de travail et des capacités de manipulation de lourdes charges. Quatre partenaires ont été impliqués dans ce projet : Tecnalia, et trois laboratoires universitaires : Institut Pascal (CNRS - Université Blaise Pascal), LIRMM (CNRS - Université Montpellier 2), et EPI COPRIN (INRIA Sophia Antipolis). Le projet Européen CableBot (2011-2014) fera suite à CoGiRo. Dans le cadre de ces projets, je me suis intéressé à l'étalonnage de ce type de structure avec l'aide de Julien Alexandre dit Sandretto.

2.4.1 ReelAx8

Le robot ReelAx8, voir figure 2.8, est un robot à câbles (8 câbles, 6 degrés de liberté) de type pleinement contraint développé par Technalia (mais pouvant être placé en position suspendue) en collaboration avec le Lirmm [LGMT12].

Son espace de travail est de 1 à 2 mètres en translation et de ± 5 degrés en orientation ; nous verrons que ce faible espace de travail en rotation a des conséquences sur l'identification de ses paramètres.

L'actionnement des câbles se fait par un enroulement autour d'un tambour. Sa rotation est couplée avec une translation afin de guider le câble lors de son enroulement/déroulement et ainsi d'éviter la superposition en couches des câbles, voir 2.9. Les tensions dans les câbles sont mesurées via un capteur lié à une poulie de renvoi. Il faut cependant noter que leur utilisation s'avère problématique en raison de la gamme des forces très étendues à mesurer, ce qui ne permet pas une mesure fine des tensions.

Les câbles actionnés sont alors renvoyés vers des œillères permettant de considérer leurs sorties vers le mobile comme ponctuelle, ce qui est visible sur la figure 2.9.

Ce prototype a servi aux premières expérimentations d'étalonnage dans la thèse de Julien Alexandre dit Sandretto dans le cadre de l'ANR CoGiRo.

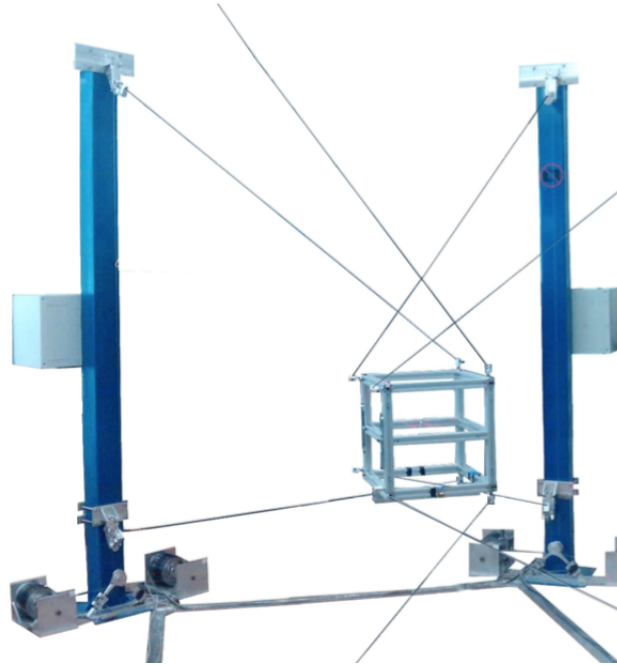


FIGURE 2.8 – ReelAx8 : Configuration pleinement contraint

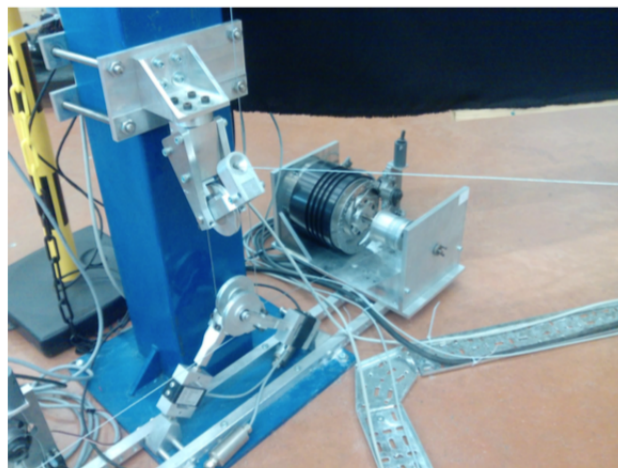


FIGURE 2.9 – Actionnement par enroulement autour d'un tambour ; la mesure de la force est assurée via un renvoi de câble

2.4.2 CoGiRo

Le robot CoGiRo, voir figure 2.10, est un robot à câbles (8 câbles, 6 degrés de liberté) de type suspendu.

Son espace de travail est d'une dizaine de mètres en translation (5 mètres en z) et de ± 30 degrés en orientation autour des 3 axes. L'actionnement des câbles et la structure du robot est du même type que pour le robot ReelAx8 présenté précédemment en configuration suspendue.

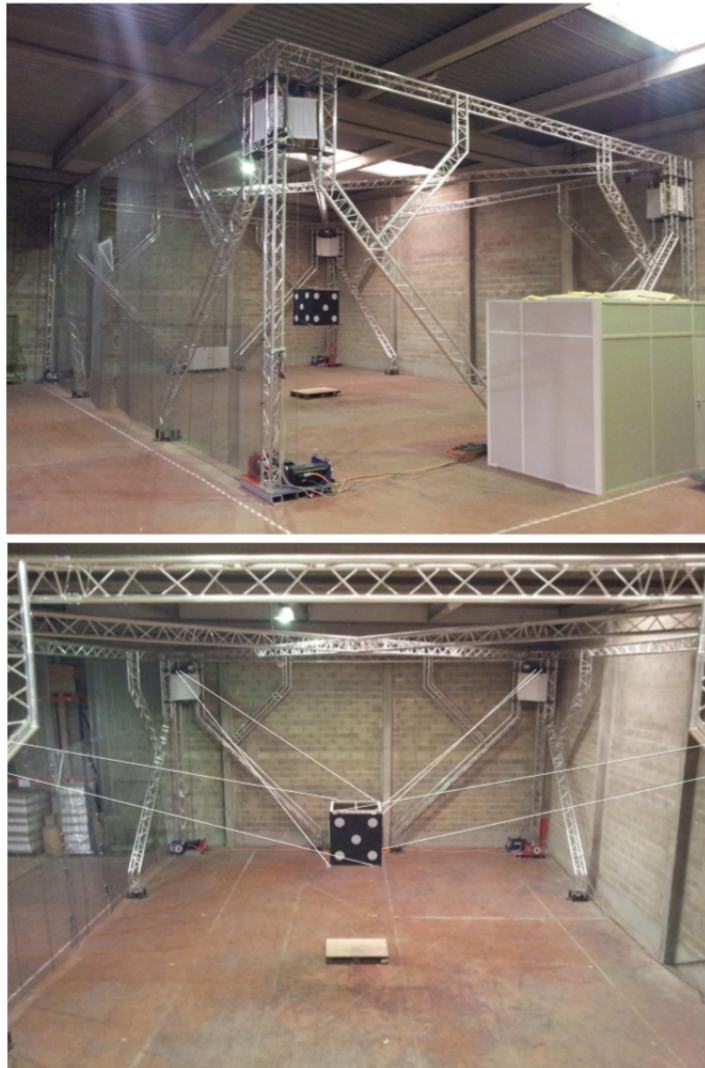


FIGURE 2.10 – CoGiRo : configuration suspendue

L'un des objectifs principaux de l'ANR CoGiRo a été de valider une commande référencée vision. Le robot a donc été instrumenté par des caméras permettant de visualiser le départ des câbles en sortie d'œillère mais aussi la plate-

forme mobile. Malheureusement, nous n'avons pas pu exploiter ces informations pour l'étalonnage du robot.

2.4.3 Marionet

Les robots de la série MARIONET ont été développés au sein du projet CO-PRIN comme des démonstrateurs technologiques et des preuves de concepts vers des applications cibles de la robotique parallèle à câbles. Les quatre principaux sont :

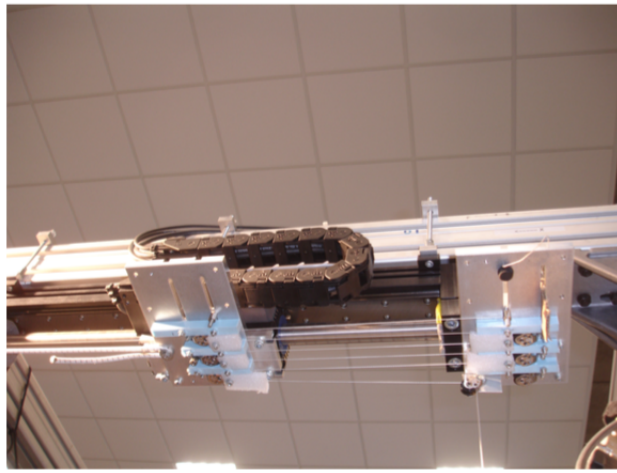


FIGURE 2.11 – Système d'actionnement linéaire utilisant un palan

- MARIONET-REHAB, pour la rééducation et les déplacements ultrarapides ; l'actionnement des câbles utilise un système original de palan, voir [MD07],
- MARIONET-CRANE, utilisé comme grue de secours, portable à dos d'hommes, totalement autonome et pouvant soulever plus d'une tonne [MD10],
- MARIONET-ASSIST, pour le transfert des personnes âgées ou handicapées,
- MARIONET-VR, installé dans la salle immersive du centre de recherche Sophia Antipolis Méditerranée de Inria ; il utilise 6 actionneurs linéaires et peut soulever une personne. Il est utilisé pour de la rééducation et de la simulation avec le but de permettre, à terme, un retour haptique. De nombreux développements sont envisagés dans le futur.

Bibliographie

- [DACP06] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval method for calibration of parallel robots : Vision-based experiments. *Mechanism and Machine Theory*, 41(8) :929 – 944, 2006.
- [GW62] VE Gough and SG Whitehall. Universal tyre test machine. In *Proc. FISITA 9th Int. Technical Congress*, pages 117–137, 1962.
- [HHK⁺01] J Hesselbach, MB Helm, H Kerle, S Soetebier, and M Krefft. A parallel robot with spread-band elements. In *32nd International Symposium on Robotics, Seoul*, pages 1731–1736, 2001.
- [LGMT12] Johann Lamaury, Marc Gouttefarde, Micaël Michelin, and Olivier Tempier. Design and control of a redundant suspended cable-driven parallel robots. In Jadran Lenarcic and Manfred Husty, editors, *Latest Advances in Robot Kinematics*, pages 237–244. Springer Netherlands, 2012.
- [MD07] Jean-Pierre Merlet and David Daney. A New Design for Wire-Driven Parallel Robot. In *Inter. Cong. Design and Modelling of Mechanical Systems (CMSM 2007)*, Monastir, Tunisie, 2007.
- [MD09] Jean-Pierre Merlet and David Daney. Kinematic analysis of a spatial four-wire driven parallel crane without constraining mechanism. In Andrés Kecskeméthy and Andreas Muller, editors, *Computational Kinematics*, pages 1–8. Springer, 2009.
- [MD10] Jean-Pierre Merlet and David Daney. A portable, modular parallel wire crane for rescue operations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2834–2839, Anchorage, États-Unis, 2010. IEEE.
- [MPD00] Jean-Pierre Merlet, M.-W. Perng, and David Daney. Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator. In Jadran Lenarcic and Michael M. Stanisic, editors, *International symposium on Advances in robot kinematics*, pages 315–322, Piran, Slovénie, 2000. Springer.
- [Ste65] Doug Stewart. A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1) :371–386, 1965.

Chapitre 3

Analyse par intervalles pour la robotique

La représentation par intervalles des données est intéressante car elle permet une approche ensembliste pour :

- vérifier le comportement d'un système pour un ensemble possible de valeurs des paramètres,
- prendre en compte les incertitudes (de mesures ou de modèles).

La force de cette approche est qu'elle **garantie** une caractérisation complète du comportement, en particulier du pire des cas.

Pour cela nous utilisons une arithmétique par intervalles qui permet, au travers d'une extension des opérateurs élémentaires, d'évaluer des fonctions, des systèmes d'équations, sur des ensembles bornés. Plus encore, des algorithmes provenant de l'*analyse par intervalles* ou de la *programmation par contrainte* fournissent un encadrement des solutions d'un système linéaire ou non-linéaire, pour répondre à un problème d'algèbre linéaire ou d'optimisation, ceci pour des systèmes à coefficients intervalles. Mais le prix à payer est une sur-estimation des bornes exactes des ensembles recherchés qui est due aux propriétés de l'arithmétique d'intervalles que nous verrons en section 3.1.2.

Une grande partie de mes contributions consiste à utiliser ces approches afin de répondre de façon originale à de nombreux problèmes en robotique que nous verrons dans les chapitres suivants.

Mais il est à souligner que la qualité et l'efficacité des solutions que nous avons mis en place sont directement dépendantes du travail sur la modélisation des problèmes en robotique et des outils intervalles utilisés pour les résoudre. Ainsi, cela demande à développer deux expertises importantes :

- Formuler le problème afin que les propriétés des solutions obtenues par ces approches intervalles offre une réponse pertinente et originale à la problématique. Nous y reviendrons de façon générique en section 3.2.2.
- Maîtriser les avantages et les inconvénients de ces approches afin de concevoir une solution adaptée au problème traité.

La diffusion de nos approches dans la communauté est freinée par l'apparente simplicité des fondements des approches par intervalles. En effet, l'algorithme de base de type évaluation-bissection est facilement compréhensible et peut être implanté facilement, voir section 3.1.5.

Or, son efficacité devient vite limitée en raison de l'explosion combinatoire générée par des problèmes en haute dimension. L'erreur classique est de penser qu'il suffit d'étendre aux intervalles un résultat obtenu sur les scalaires. Ce n'est, en général, pas le cas. Nous pouvons prendre par exemple le calcul d'une trilatération par intervalle : nous montrons dans [GDP⁺07] l'intérêt d'appliquer des algorithmes de filtrage directement sur le système des équations de base plutôt que d'étendre par intervalle le résultat d'une résolution classique de ce problème.

De plus, les outils classiques (résolution de systèmes linéaires, polynomiaux, calcul d'un déterminant, optimisation sous contraintes etc ...) doivent être modifiés afin de prendre en compte les particularités de l'analyse par intervalles. Ainsi, des algorithmes obsolètes ou oubliés s'avèrent alors parfaitement adaptés à une utilisation avec des données intervalles.

Ainsi, il a été nécessaire :

- d'offrir une bibliothèque qui regroupe ces méthodes de bases adaptées. Le projet Coprin a développé la bibliothèque ALIAS qui a été utilisée dans nombre de mes contributions. Elle regroupe de nombreuses fonctionnalités pour la résolution de systèmes, l'optimisation ou en algèbre linéaire. Afin de diffuser ces travaux dans le domaine de l'ingénierie, j'ai proposé une interface Scilab à cette bibliothèque [Dan10] et j'ai contribué au même type d'interface pour Mathematica [PD08].
- de développer de nouveaux algorithmes dont les besoins sont guidés par des manques de solutions pour nos problèmes de robotique.

3.1 Quelques notions de base de l'arithmétique et de l'analyse par intervalles

Le but de cette section est de présenter les difficultés liées à l'utilisation de l'analyse par intervalle. Ceci a des conséquences sur les traitements des problèmes de robotique que nous avons considérés et explique nos choix en matière de développement logiciel.

La très grande force de l'analyse par intervalles est de pouvoir calculer sur les ensembles, c'est à dire de borner les propriétés d'un système pour des plages continues de valeurs de leurs paramètres. Nous verrons dans les chapitres suivants que cette capacité est largement exploitée pour résoudre des problèmes de robotique. En effet, il devient simple de montrer que certaines valeurs limites d'une fonction, représentant des propriétés d'un système robotique, ne peuvent pas être atteintes (ex : espace de travail sous contraintes articulaires) ou que des équations ne peuvent s'annuler pour un ensemble d'états du système (vérification des singularités).

Sur un domaine initial de recherche borné (ce qui est généralement le cas), nous pouvons obtenir des informations globales, sans perte de solution. En effet, les algorithmes que nous présenterons ci-après sont fondés sur les contractions successives certifiées autour des solutions et non pas, classiquement, sur des convergences qui peuvent être affectées par des minima locaux. Mais ceci à un prix : s'il est facile de montrer la non-existence d'une solution sur un ensemble, il est plus difficile d'en prouver l'existence (mais pas impossible ...). De plus, l'efficacité et la qualité des solutions sont directement liées à la forme de l'expres-

sion utilisée pour modéliser. Il devient alors difficile de proposer des solutions génériques et automatiques. Le travail sur la modélisation des problèmes, du choix des outils mathématiques utilisés devient donc essentiel.

3.1.1 Arithmétique par intervalles

L'arithmétique par intervalle est proposée par Moore dans [Moo66]. Nous synthétisons une petite introduction qui est reprise de N. Revol dans [Rev01]. Un intervalle $\mathbf{x} = [\underline{x}, \bar{x}]$ est défini par $\{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}$. L'ensemble des intervalles de \mathbb{R} est noté \mathbb{IR} . On peut généraliser cette définition pour des vecteurs $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_n]^T \in \mathbb{IR}^n$, appelés aussi boîtes, ou à des matrices $\mathbf{A} \in \mathbb{IR}^{n \times m}$. Le résultat d'une opération entre intervalles $\mathbf{x} \circ \mathbf{y}$ est donné par un intervalle qui inclut $\{x \circ y \mid x \in \mathbf{x}, y \in \mathbf{y}\}$. De façon générique, nous noterons \square cette approximation par intervalles qui inclut les ensembles recherchés $\mathbf{x} \circ \mathbf{y} = \square\{x \circ y \mid x \in \mathbf{x}, y \in \mathbf{y}\}$. Nous pouvons ainsi définir les opérateurs arithmétiques $(+, -, \times, ^n, /, \sqrt{\cdot})$, les fonctions élémentaires $(\exp, \sin, \cos, \dots)$. Afin de garantir ces opérateurs et ces fonctions élémentaires il est important lors de l'implantation de cette arithmétique de choisir convenablement le sens des arrondis lors des opérations sur les scalaires. C'est dans cette optique que nous avons utilisé pour la bibliothèque C++, BIAS/Profil de l'université technologique de Hambourg []. Cette prise en compte de l'arrondis permet la certification de l'arithmétique ce qui explique son utilisation dans bien d'autre domaine, par exemple en calcul formel. En robotique, à cause des problème d'arrondis, il est possible de *perdre* une solution d'une résolution ou de confondre une solution double, ce qui peut avoir des conséquences sur l'interprétation du résultat. L'arithmétique d'intervalle peut être une solution à ce problème.

3.1.2 Propriétés algébriques

L'arithmétique par intervalles perd un certain nombre de propriétés comparée à son équivalent scalaire. Ainsi, la soustraction (resp. la division) n'est pas la réciproque de l'addition (resp. la multiplication) : $\mathbf{x} - \mathbf{x} = \{x - y \mid x \in \mathbf{x}, y \in \mathbf{y}\} \supset \{x - x \mid x \in \mathbf{x}\}$. Nous avons donc une perte de corrélation entre les occurrences des variables dans une expression. Cette remarque se retrouve dans nombre de cas, par exemple $\mathbf{x} \times \mathbf{x} \supset \mathbf{x}^2$. Autre point, la multiplication est dite sous-distributive par rapport à l'addition $\mathbf{x} \times (\mathbf{y} + \mathbf{z}) \subset \mathbf{x} \times \mathbf{y} + \mathbf{x} \times \mathbf{z}$.

3.1.3 Évaluation : les extensions par intervalles

Nous pouvons introduire l'extension aux intervalles d'une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, fondée sur les opérateurs arithmétiques et les fonctions élémentaires.

Ainsi, $\mathbf{f} : \mathbb{IR} \rightarrow \mathbb{IR}$ est une extension aux intervalles de f si $\forall \mathbf{x} \in \mathbb{IR}, \mathbf{f}(\mathbf{x}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{x}\}$ et $\forall x \in \mathbb{R}, f(x) = \mathbf{f}(\mathbf{x})$. Nous pouvons alors étendre ces notions dans les dimensions supérieures. Les deux principales extensions utilisées en analyse par intervalle sont :

- L'extension naturelle, notée $\mathbf{F}_{\mathbf{N}}$, qui consiste à remplacer les opérateurs scalaires par les opérateurs intervalles dans l'expression de la fonction. Par exemple, l'extension naturelle des fonctions suivantes, équivalentes dans les scalaires, pour $\mathbf{x} = [-1, 1]$ est donnée par :
 - $f_1(x) = x(x + 1) = [-1, 1]([-1, 1] + 1) = [-2, 2]$

- $f_2(x) = x.x + x = [-1, 1].[-1, 1] + [-1, 1] = [-2, 2]$
- $f_3(x) = x^2 + x = [-1, 1]^2 + [-1, 1] = [-1, 2]$
- $f_4(x) = (x + 0.5)^2 - 0.25 = ([-1, 1] + 0.5)^2 - 0.25 = [-0.25, 2]$

Notez que la dernière expression, où la variable x n'apparaît qu'une seule fois, donne les bornes exactes de l'expression évaluée sur $\mathbf{x} = [-1, 1]$.

- L'extension de Taylor, noté \mathbf{F}_T , qui approxime la fonction par le développement en série au 1er ou 2ème ordre donné sous la forme :

$$\mathbf{F}_{T,1}(\mathbf{X}) \subseteq \mathbf{F}(\mathbf{x}_0) + \mathbf{J}(\mathbf{X})(\mathbf{X} - \mathbf{x}_0) \quad (3.1)$$

$$\mathbf{F}_{T,2}(\mathbf{X}) \subseteq \mathbf{F}(\mathbf{x}_0) + \mathbf{J}(\mathbf{X}_0)(\mathbf{X} - \mathbf{x}_0) + \frac{1}{2} \cdot (\mathbf{X} - \mathbf{x}_0)^T \cdot \mathbf{H}(\mathbf{X}) \cdot (\mathbf{X} - \mathbf{x}_0) \quad (3.2)$$

où $x_0 \in \mathbf{X}$, $J(X) = \frac{\partial F}{\partial X}$ la jacobienne du système et H son Hessienne.

Mentionnons aussi l'extension de Hansen qui propose de développer la forme de Taylor pour chacune des variables intervalles successivement. Cette forme incrémentale permet d'obtenir une jacobienne (au 1er ordre) dont le diamètre des coefficients est inférieur à ceux obtenu par une approche classique.

Mais il existe aussi de nombreuses formes d'extension aux intervalles suivant la nature des fonctions : par exemple pour les polynômes, la forme standard, centrée, de Horner, de Berstein, interpolée etc. En raison des propriétés énoncées en sous-section 3.1.2, chacune des extensions $\mathbf{F}(\mathbf{X})$ de $F(X)$ donne une solution intervalle plus ou moins sur-estimée des bornes de l'ensemble $\{F(X)|X \in \mathbf{X}\}$. Cependant il existe deux façons de garantir les bornes exactes de cet ensemble :

- 1 Limiter l'apparition des occurrences des variables : si chacune des variables n'apparaît qu'une seul fois dans l'expression choisie, nous pouvons garantir l'obtention les bornes exactes, aux arrondis près, voir section 3.1.1.
- 2 Prendre en compte la monotonicité des expressions : si la fonction est strictement croissante ou décroissante sur les domaines des variables il est alors possible de déterminer des bornes exactes de l'ensemble recherché par une évaluation scalaire de l'expression en les bornes des variables intervalles.

Ainsi si $\forall X \in \mathbf{X}, J(X) \geq 0$ (resp. $J(X) \leq 0$) alors $\mathbf{F} = [\underline{\mathbf{X}}, \overline{\mathbf{X}}]$ (resp. $\mathbf{F} = [\overline{\mathbf{X}}, \underline{\mathbf{X}}]$). Le calcul de la condition $\forall X \in \mathbf{X}, J(X) \geq 0$ ou ≤ 0 peut faire appel a l'arithmétique d'intervalles en évaluant directement $J(\mathbf{X})$ mais le problème devient alors récursif puisque l'évaluation de $J(\mathbf{X})$ est soumis au même propriété de l'arithmétique et donc potentiellement surévalué. Pour l'implantation effective de l'algorithme, nous conseillons de voir <http://www-sop.inria.fr/hephaistos/logiciels/ALIAS/ALIAS-C++/node37.html>.

Si le challenge est de déterminer les bornes exactes de $\{F(X)|X \in \mathbf{X}\}$, ceci ne répond pas pleinement au problème de la description complète et minimal de cet ensemble. Pour des problèmes de dimension > 1 , nous devons considérer le problème de l'effet enveloppant (wrapping effect). En effet, l'image d'un vecteur intervalle par une fonction n'est généralement pas un pavé, c'est un ensemble de forme quelconque. Cependant l'arithmétique impose que le résultat de l'évaluation soit donné sous la forme d'un pavé : nous obtenons donc forcément une sur-estimation de l'image. Un exemple simple de robotique dans le plan est le

suisant. Soit un vecteur intervalle $u = ([-1, 1], [-1, 1])^t$ multiplié par une matrice de rotation de 45° . Nous obtenons le vecteur $u' = ([-\sqrt{2}, \sqrt{2}], [-\sqrt{2}, \sqrt{2}])^t$ qui contient bien l'ensemble de l'image de u par la rotation R . Cependant :

- $(R.R.R.R).u = ([-1, 1], [-1, 1])^t$ produit l'évaluation exacte de l'image de u qui est bien un pavé droit,
- $R.(R.(R.(R.u))) = ([-4, 4], [-4, 4])^t$ Les quatre rotations successives démultiplient, sous cette forme, l'effet enveloppant, sur-estimant grandement l'évaluation de cette fonction.

L'évaluation d'une fonction ou d'un système par intervalle est donc un problème clé qu'il est nécessaire de spécifier pour traiter convenablement un problème. Pour synthétiser, la décorrélation des occurrences des variables et l'effet enveloppant sont les deux problèmes à prendre en compte lors de la phase de la mise en équation. Travailler les expressions des contraintes et de leur jacobiniennes, tester les différents types d'extension possible sont nécessaires afin d'obtenir une certaine efficacité par la minimisation des effets de sur-estimation de l'évaluation par intervalles.

3.1.4 Résolution de systèmes par intervalles

L'analyse par intervalle offre des méthodes afin de déterminer les zéros d'un système d'équations linéaires ou non sur un domaine initial borné¹. Plus particulièrement, nous pouvons considérer deux cas :

- Les coefficients des équations sont scalaires : ces méthodes peuvent être utilisées afin de déterminer l'ensemble des solutions d'un système.
- Les coefficients des équations sont intervalles : ces méthodes sont utilisées afin de rendre robuste l'identification des solutions aux erreurs d'arrondis. Mais nous nous sommes plutôt intéressé à déterminer des continuum de solutions générés par l'épaississement des équations du à l'extension aux intervalles de leurs coefficients.

Résolution de systèmes linéaires par intervalles

Considérons une matrice intervalle $\mathbf{A}_{m \times n}$ et une boîte $\mathbf{b}_{m \times 1}$, et plus particulièrement le cas bien-contraint $m = n$, nous cherchons à déterminer l'ensemble des vecteurs x qui vérifie l'équation $A.x = b$ pour $A, b \in \mathbf{A}, \mathbf{b}$. Si nous utilisons une arithmétique par intervalles, le problème revient à déterminer :

$$\Sigma_{\exists, \exists}(\mathbf{A}, \mathbf{b}) = \square \{x | \exists A \in \mathbf{A}, \exists b \in \mathbf{b}, A.x = b\} \quad (3.3)$$

avec \square une approximation externe certifié du pavé qui contient l'ensemble $D = \{x | \exists A \in \mathbf{A}, \exists b \in \mathbf{b}, A.x = b\}$, généralement non convexe.

Il existe de nombreuses solutions afin d'obtenir une approximation plus ou moins bonne (c'est à dire la boîte englobante de diamètre la plus petite possible). Nous pouvons citer l'algorithme intervalle d'élimination de Gauss directement transposé de son équivalent scalaire, des méthodes itératives (Gauss-Seidel) ou de relaxation convexe (Krawczyk). Cependant, nombre d'algorithmes sont fondés sur l'utilisation du théorème d'Oettli et Pragger qui montre que :

1. Nous n'évoquerons pas les méthodes dédiés à la résolution de problème d'optimisation contraint peu utilisés dans mes contributions.

$$x \in \Sigma_{\exists, \exists}(\mathbf{A}, \mathbf{b}) \leftrightarrow |mid(\mathbf{A}).x - mid(\mathbf{b})| \leq rad(\mathbf{A}).|x| + rad(\mathbf{b}) \quad (3.4)$$

avec $mid(\mathbf{Z}) = 1/2.(\underline{Z} + \overline{Z})$ milieu de l'intervalle Z , et $rad(\mathbf{Z}) = 1/2.(\overline{Z} - \underline{Z})$ son rayon.

Toute la difficulté est de faire disparaître les valeurs absolues soit en traitant ce problème quadrant par quadrant (ex : méthode Albert, Rump) ou par des relaxations ou approximations successives (ex : méthode de Hansen-Bliek-Rohn-Ning-Kearfott, Beaumont). Dans beaucoup de cas, un pré-conditionnement peut s'avérer utile, voir [Rev01], en déterminant par exemple :

$$\Sigma_{\exists, \exists}(mid(\mathbf{A})^{-1}.\mathbf{A}, (mid(\mathbf{A})^{-1}.\mathbf{b})) \quad (3.5)$$

Remarques :

- Il est à noter qu'il peut être intéressant de changer les quantificateurs associés à la matrice \mathbf{A} et/ou au vecteur \mathbf{b} .

$$(3.6)$$

$$\Sigma_{\forall, \exists}(\mathbf{A}, \mathbf{b}) = \square\{x | \forall A \in \mathbf{A}, \exists b \in \mathbf{b}, A.x = b\}$$

$$\Sigma_{\exists, \forall}(\mathbf{A}, \mathbf{b}) = \square\{x | \exists A \in \mathbf{A}, \forall b \in \mathbf{b}, A.x = b\} \quad (3.7)$$

$$(3.8)$$

- Les méthodes appropriées dans les cas sur-contraints $m > n$ et sous-contraints $n > m$ seront examinées respectivement dans le chapitre concernant l'étalonnage 5.4.3 et celui concernant la conception appropriée 4.3.3.

Résolution de systèmes non-linéaires par intervalles

Nous cherchons maintenant à déterminer une approximation intervalle (noté \square) de l'ensemble $\{x \in \mathbf{D} | \exists p \in \mathbf{p}, F(x, p) = 0\}$ dans un domaine initial \mathbf{D} avec f un système d'équations non-linéaires.

L'algorithme de base en analyse par intervalles est le *Newton par intervalle*. Il est fondé sur une extension par intervalles de la forme de Taylor au 1er ordre du système d'équations considéré, voir 3.1.

$$\mathbf{F}(\mathbf{X}, \mathbf{p}) \subseteq \mathbf{F}(\mathbf{x}_0, \mathbf{p}) + \mathbf{J}(\mathbf{X}, \mathbf{p})(\mathbf{X} - \mathbf{x}_0) = \mathbf{0} \quad (3.9)$$

Il consiste à borner par inclusions successives les solutions d'un système linéaire par intervalles qui englobe les solutions du système non-linéaire de départ.

L'opérateur de Newton est alors donné par :

$$N(\mathbf{X}) = x_0 + \Sigma_{\exists, \exists}(J(\mathbf{X}, \mathbf{p}), -\mathbf{F}(\mathbf{x}_0, \mathbf{p})) \quad (3.10)$$

avec $\Sigma_{\exists, \exists}$ l'opérateur de résolution d'un système linéaire par intervalles vu en section 3.1.4 et x_0 un point scalaire de \mathbf{X} .

Le schéma de Newton d'inclusions successives est alors $\mathbf{X} \leftarrow N(\mathbf{X}) \cap \mathbf{X}$. Si $N(\mathbf{X}) \cap \mathbf{X} = \emptyset$ alors le système n'admet pas de solution dans \mathbf{X} . De plus, il est à noter une de ses propriétés intéressantes : sa capacité à prouver l'existence et l'unicité (les jacobiniennes contenues dans $J(\mathbf{X}, \mathbf{p})$ doivent être régulières) d'une

solution dans un intervalle donné si une stricte inclusion est obtenue au cours des itérations successives.

Exemple simple :

- Déterminer x tel que $x^2 + 2x - 1 = 0$ pour $x \in D = [-0.9, 0.9]$
- Le schéma de Newton nous donne : $\mathbf{X}_{k+1} = \mathbf{X}_k \cap \frac{1}{2\mathbf{X}_k+2} \cdot [-x_{m,k}^2 - 2x_{m,k} + 1] + x_{m,k}$
- Itérativement le domaine initial D se réduit (se contracte) autour d'une solution (unique dans le domaine initial) $\mathbf{X}_1 = [0.2631, 0.9001], \dots, \mathbf{X}_4 = [0.4142, 0.4143]$

Les méthodes utilisées pour résoudre le système linéaire donnent des variantes à cette algorithmme (Moore, Hansen-Sengupta, ...).

Il existe d'autres approches provenant de la programmation par contraintes qui s'avèrent très efficaces pour diminuer l'espace de recherche initial D des solutions du système fondé sur la vérification de la cohérence des équations du système, notamment la 2B-cohérence, la Box-cohérence ou la 3B-cohérence. Nous vous conseillons de lire la thèse de Gilles Chabert [Cha07] qui introduit bien les concepts.

Une façon simpliste de présenter la 2B-cohérence consiste à réécrire les équations (ou contraintes) sous forme de projection d'une variable ou d'une occurrence de la variable sur chaque contrainte ; puis d'en vérifier la cohérence.

Ainsi, si nous reprenons l'exemple précédent

- Déterminer x tel que $x^2 + 2x - 1 = 0$ pour $x \in [-1, 1]$
- | | | |
|--------------------|--|-------------------|
| Projections | $y = \frac{1}{2} \cdot (1 - x^2)$ (1.) | $y = 1 - 2x$ (2.) |
| | $y = x$ | $y = x^2$ |

- En utilisant la projection (1.), pour $x_0 = [-1, 1]$

Algo : $x_{k+1} \leftarrow x_k \cap \frac{1}{2} \cdot (1 - x_k^2)$
 $x_1 = [0.1, 0.5], \dots, x_6 = [0.413, 0.416]$

Nous pouvons remarquer que la première itération de l'algorithme de type 2B donne des résultats meilleurs que l'utilisation de schéma de Newton par intervalles. D'une façon général, les deux algorithmes sont assez complémentaires et nos contributions en robotique les utilisent séquentiellement afin d'obtenir une efficacité maximale.

Algorithme de filtrage

Les algorithmes présentés à la section précédente permettent de diminuer l'espace de recherche en contractant une boîte initiale, autour de la solution (continuum de solutions) définie par les équations du système de contraintes $\mathbf{C}(\mathbf{X}, \mathbf{p}) = 0$ où p est un paramètre et éventuellement un intervalle. Nous appelons ce type d'opérateurs, contracteurs ou filtres.

Ils permettent de réduire les bornes de l'intervalle (ou vecteur intervalle) sans perdre de solution ou, d'un autre point de vue, d'éliminer les parties de $X \subset D$ telles que $\{X \in \mathbf{X} | X \notin \mathcal{D}\}$.

Ainsi, nous pouvons définir ces filtres tel que

$$\mathbf{X}_{new} = Filtering(\mathbf{X}_{old}, \mathbf{C}(\mathbf{x}) = 0) \text{ tel que } \mathbf{X}_{new} \subseteq \mathbf{X}_{old} \tag{3.11}$$

Il existe de nombreuses possibilités de construire ces algorithmes de filtrage utilisant un opérateur de Newton, les opérateurs 2B, 3B mais aussi des tests d'existence d'une solution sur un domaine : en robotique nous vous conseillons

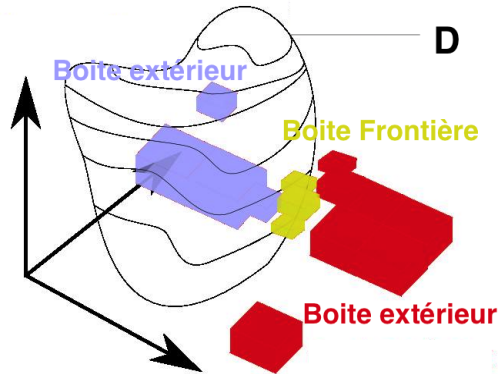


FIGURE 3.1 – Caractérisation de boîtes

de lire les solutions proposés dans [Mer04] qui offre un éventail astucieux des différentes approches.

3.1.5 Caractérisation des sous-espaces de recherche

Nous cherchons à décrire des ensembles continus de points notés \mathcal{D} caractérisés par des contraintes d'égalités $\mathcal{D} = \{X | C(X, \mathbf{p}) = 0\}$ ou d'inégalités $\mathcal{D} = \{X | \underline{c}_r \leq C(X, \mathbf{p}) \leq \overline{c}_r\}$.

La représentation par intervalles nous oblige à adopter une description de ces ensembles par des boîtes (rectangle en 2D, cube en 3D et hypercube dans les dimensions supérieures) dont les cotés sont parallèles aux axes des repères de références choisies. Nous n'obtiendrons donc une approximation de ces ensembles dans l'espace de recherche par un pavage dont chacune des boîtes possède des caractéristiques décrites dans la sous-section 3.1.5.

Type de boîtes caractérisant un espace de recherche

Le principe est de tester la nature d'une boîte courante, notée \mathbf{X}_s , représentant un espace de recherche donné par les contraintes $\mathcal{D} = \{X | \underline{c}_r \leq C(X, \mathbf{p}) \leq \overline{c}_r \text{ ou } C(X, \mathbf{p}) = 0\}$. Nous pouvons caractériser cette boîte suivant trois types, voir Figure 3.1 :

- *Boîte extérieure* si $\mathbf{X}_s \cap \mathcal{D} = \emptyset$, la boîte est strictement en dehors de l'espace de recherche.
- *Boîte intérieure* si $\mathbf{X}_s \subseteq \mathcal{D}$, la boîte est strictement contenue de l'espace de recherche.
- *Boîte frontière* qui ne sont ni des boîtes intérieures ni des boîtes extérieures. De plus, afin de donner une taille minimale à une boîte, nous ajoutons un critère sur le rayon de la boîte² : si $rad(\mathbf{X}_s) < \epsilon$ et que nous ne pouvons la caractériser comme une boîte intérieure/extérieure elle est alors considérée comme frontière ou de nature indéterminée. En effet, due à l'arithmétique des ordinateurs, il n'est pas possible de considérer

2. Pour rappel, $rad(\mathbf{x}) = 1/2(\bar{x} - \underline{x})$.

un intervalle non représentable par le type de flottant utilisé. Ainsi, ϵ ne peut être plus petit que le *ulp*, *unit in the last place* voir un multiple si les opérateurs arithmétiques prennent en compte les erreurs d'arrondi. Plus encore, l'épaisseur de la frontière d'un espace de recherche doit correspondre à la réalité de l'exploitation des résultats obtenus et donc la taille minimale d'une boîte doit être une donnée du problème.

Caractériser la boîte \mathbf{X}_s est donc un enjeux important qui dépendra fortement de la nature de la propriété testée et donc de la contrainte considérée $C(X, \mathbf{p})$.

- Si la contrainte est une inégalité
 - une boîte extérieure est caractérisée par $\mathbf{C}(\mathbf{X}_s, \mathbf{p}) \cap [\underline{c}_r, \overline{c}_r] = \emptyset$
 - une boîte intérieure est caractérisée par $\mathbf{C}(\mathbf{X}_s, \mathbf{p}) \subseteq [\underline{c}_r, \overline{c}_r]$
- Si la contrainte est une égalité
 - une boîte extérieure est caractérisée par $0 \notin \mathbf{C}(\mathbf{X}_s, \mathbf{p})$
 - une boîte intérieure est difficile à obtenir simplement. En effet, généralement, les bornes de $\mathbf{C}(\mathbf{X}_s, \mathbf{p})$ étant surestimées le test $0 \in \mathbf{C}(\mathbf{X}_s, \mathbf{p})$ ne nous garantit pas que $\exists X_s \in \mathbf{X}_s$ tel que $C(X_s, \mathbf{p}) = 0$. Plus encore, considérons le cas où la contrainte $C(X, \mathbf{p})$ est unidimensionnelle (donc pas d'effet enveloppant) et que l'extension par intervalles \mathbf{C} de C nous garantit les bornes exactes de l'évaluation de \mathbf{C} (par exemple si la fonction est monotone sur l'intervalle considéré). Dans ce cas une simple évaluation de la contrainte peut nous indiquer que $\exists X_s \in \mathbf{X}_s, C(X_s, \mathbf{p}) = 0$ mais pour faire de \mathbf{X}_s une boîte intérieure il faut montrer que $\forall X_s \in \mathbf{X}_s, C(X_s, \mathbf{p}) = 0$.

Algorithme de base : Branch and bound

Une fois ces tests établis, l'algorithme de base, pour déterminer un espace de travail qui vérifie les propriétés considérées d'un système, est un *Branch and Bound*. Plus précisément, une liste de boîte à traiter \mathcal{L} est initialisée par une boîte englobant l'ensemble des solutions décrit par les contraintes $C(X_s, \xi)$. Pour chaque élément de la liste, nous vérifions si nous pouvons catégoriser la boîte courante \mathbf{X}_c , comme boîte intérieure (stockée dans \mathcal{L}_{int}), extérieure (stockée dans \mathcal{L}_{ext}) ou frontière (stockée dans \mathcal{L}_{front}) : si tous les tests échouent alors la boîte est bissectée en plusieurs sous-boîtes puis ajoutée à la liste \mathcal{L} . L'algorithme se termine lorsque la liste \mathcal{L} est vide. La complexité de cet algorithme est exponentielle, dans le pire cas, ce qui peut faire craindre son inefficacité. Cependant, il est à noter que pour des problèmes de dimensions faibles (par exemple pour la recherche d'un espace de dimension 3D) le critère d'arrêt sur une taille minimal de boîte à considérer ϵ est relativement vite atteint au regard de l'interprétation physique d'une taille minimal de boîte versus un espace de travail. Typiquement, il faut une vingtaine de bisections successives d'un cube

de $1m^3$ pour le découper un cube de $1cm^3$ de côté.

```

0.  $\mathcal{L} \leftarrow$  Boite englobante de  $\mathcal{D}$ 
while  $\mathcal{L} \neq \emptyset$  do
  1.  $\mathbf{X}_c \leftarrow \mathcal{L}_k$  avec  $k$  indice d'un des éléments de  $\mathcal{L}$ 
  if Test_boite_frontière( $\mathbf{X}_c$ ) is ok then
    2.  $\mathcal{L}_{\text{front}} \leftarrow \mathcal{L}_{\text{front}}, \mathbf{X}_c$ 
  else if Test_boite_extérieure( $\mathbf{X}_c$ ) is ok then
    3.  $\mathcal{L}_{\text{ext}} \leftarrow \mathcal{L}_{\text{ext}}, \mathbf{X}_c$ 
  else if Test_boite_intérieure( $\mathbf{X}_c$ ) is ok then
    4.  $\mathcal{L}_{\text{int}} \leftarrow \mathcal{L}_{\text{int}}, \mathbf{X}_c$ 
  else
    5. Bissecter( $\mathbf{X}_c$ )  $\rightarrow \{\mathbf{X}_1, \mathbf{X}_2\}$ ;
    6.  $\mathcal{L} \leftarrow \mathcal{L}, \mathbf{X}_1, \mathbf{X}_2$ 
end

```

Algorithm 1: Algorithme de base de vérification d'un espace de recherche : Branch and bound

Cependant, il faut remarquer que :

- si la dimension du problème est grande, en fonction de la nature des fonctions, l'algorithme peut entraîner une explosion combinatoire du nombre de boites à traiter.
- l'évaluation des contraintes 3.1.3 est un enjeu majeur : du à la surestimation possible de l'évaluation des tests 3 et 4, l'algorithme risque de ne pas pouvoir conclure et donc de devoir bissecter la boite.
- comme nous l'avons vu précédemment, le test de boite intérieur n'est pas toujours possible et souvent difficile.

Ainsi, cet algorithme de base doit être amélioré : le but est alors de réduire l'explosion combinatoire créée par la bisection.

Les solutions sont alors :

- 1 d'affiner l'évaluation des contraintes, c'est à dire de diminuer leur surestimation due aux propriétés de l'arithmétique. Il faut donc :
 - modéliser les contraintes afin que leur expression soit favorable à une évaluation par intervalles. Cette remarque est importante car elle explique l'intérêt pour un roboticien de bien comprendre les enjeux afin d'orienter le choix de l'outil de modélisation (mathématique, cinématique, mécanique) pour formuler le problème.
 - tester les différents type d'extension par intervalle,
 - utiliser des outils de calcul formel afin de travailler la forme symbolique des contraintes, voir [PD07]
 - s'efforcer de prendre en compte les dépendances entre les paramètres,
 - utiliser la monotonie des équations.
- 2 de trouver un test efficace pour vérifier les boites intérieures. Si l'équation est paramétrée par un intervalle \mathbf{p} (voir fonction épaissie 3.2.1), il faut montrer que $\forall x \in \mathbf{x}, \exists p \in \mathbf{p}, C(x, p) = 0$. Pour beaucoup de cas étudiés en robotique nous arrivons à déterminer une condition équivalente. Mais ce test doit être étudié au cas par cas : le challenge est alors de déterminer cette condition.
- 3 d'ajouter aux tests sur les boites intérieures/extérieures, des méthodes de filtrage, vu précédemment en section 3.1.4. Les gains de performances sont généralement considérables : ces étapes permettent de réduire fortement l'espace courant de recherche sans augmenter la liste des boites à

traiter. Les tests d'existence, voir d'unicité, permet de rejeter ou non une boîte, voir d'utiliser d'autres algorithmes pour converger (ou contracter) vers une solution possiblement unique.

- 4 d'améliorer l'efficacité de l'algorithme de branch and bound en utilisant des stratégies de bisection adaptées au problème : gestion du stockage des boîtes, choix des variables à bissecter (round robin, smear function ...).

3.2 Approches par intervalles pour la robotique

Cette section a pour ambition de montrer quelles sont les interactions entre contributions en analyse par intervalles et en robotique. Plus concrètement ces dernières seront plus développées dans les chapitres suivant. Mais nous montrons que tirer avantages de ces approches nécessite aussi de prendre en compte les inconvénients de leurs propriétés.

3.2.1 Modélisation et Contraintes épaissies

L'un des intérêts de l'arithmétique d'intervalles est d'offrir des extensions par intervalles d'une fonction, ou plus génériquement d'une contrainte, données sous la forme $\mathbf{C}(\mathbf{X}, \mathbf{m}, \xi)$. En robotique, les variables intervalles $\mathbf{X}, \mathbf{m}, \xi$ permettent de représenter plusieurs types de phénomènes bornés :

- \mathbf{X} des ensembles continus d'états de la variable par exemple un ensemble de configurations du robot ou bien une portion de son espace de travail.
- \mathbf{m} une incertitude de mesure, donnée au pire cas c'est à dire par un ensemble borné, sans pré-supposer une quelconque distribution statistique.
- ξ des tolérances associés à une variable qui permettent, par exemple, de bien représenter des incertitudes de fabrication et d'assemblage.

Il est très difficile de prendre en compte l'ensemble des phénomènes physiques auquel est soumis un système. En effet, pour cela, il faudrait pouvoir mesurer ces phénomènes et donc ajouter des variables les symbolisant et ainsi complexifier le modèle. Dans la pratique, le modèle retenu est généralement le fruit de simplifications d'une version plus complexe, non-accessible. Dans les cas étudiés en robotique, elles sont typiquement le fruit d'hypothèses sur la rigidité des corps ou sur la nature des articulations... La question est alors de les valider : l'ajustement de sa complexité est objectivé par sa capacité prédictible au regard d'une tolérance sur les propriétés recherchées.

Les contraintes $\mathbf{C}(\mathbf{X}, \mathbf{m}, \xi)$ ainsi obtenues sont alors erronées. Cependant, nous pouvons souvent borner cette approximation $\mathbf{C}(\mathbf{X}, \mathbf{m}, \xi, \epsilon_{\mathbf{m}})$ par un paramètre $\epsilon_{\mathbf{m}}$ qui modélise cette erreur de modèle.

La modélisation intervalle permet d'unifier, dans un même formalisme, la représentation des variabilités des paramètres/données/approximations qui le composent. Bien plus qu'une représentation simple des incertitudes, c'est cette capacité qui rend l'analyse par intervalle intéressante et pertinente. Le principe est alors de vérifier les propriétés d'un système à travers leur évaluation par des contraintes dites épaissies : le comportement réel du système, quelle que soit la variation de ses variables et des approximations de son modèle est alors contenu dans ces contraintes épaissies. Ainsi, si les propriétés d'un système sont vérifiées

pour une représentation englobante et certifiée de son comportement, elles le seront alors pour son comportement réel.

Nous retrouverons cette notion de contrainte épaissie dans les deux chapitres suivants consacrés aux contributions robotiques pour la conception et l'étalonnage.

3.2.2 Formulation d'un problème

Afin d'utiliser au maximum les forces de l'analyse par intervalles et de minimiser ses désavantages, il est souhaitable d'adapter la formulation des problèmes de robotique, ce qui permet d'obtenir de bons résultats tout en étant pertinent en pratique.

Prenons l'exemple fondateur qui a orienté les contributions du projet Coprin vers l'analyse par intervalle. Les singularités des robots parallèles sont des points de l'espace pour lesquelles les forces internes du robot tendent vers l'infini. Il est donc souhaitable de garantir que l'espace de travail effectif du robot n'en contient pas. L'approche classique consisterait à déterminer leurs localisations; l'approche intervalle a plutôt tendance à vérifier leur non-existence dans une boîte (l'espace de travail) donnée. Pour cette exemple, vérifier l'existence revient génériquement à garantir qu'une équation (le déterminant d'une matrice jacobienne) s'annule pour une certaine valeur des variables : ce qui est difficile à faire. En utilisant une approche par intervalle, il est plus facile de montrer la non-existence, ce qui revient à vérifier que l'équation ne s'annule pas pour la boîte considérée : une simple évaluation par intervalles de l'équation suffit à donner la condition nécessaire de la non-existence qui sont garanties par l'arithmétique d'intervalles.

Cependant, pour des robots à plus de trois degrés de liberté, le développement du déterminant devient problématique car son degré, en un grand nombre de variables, devient élevé induisant une surestimation de son évaluation par intervalle. Une solution consiste à développer des outils spécifiques (voir ALIAS [Mer00]) qui ne passent pas par le développement du déterminant, en prenant en compte les dépendances entre les paramètres ou la monotonie du déterminant pour en déduire des bornes exactes (aux arrondis près).

Mais une autre approche, plus intéressante, est développée dans la thèse de Julien Hubert [Hub10] que j'ai co-encadré avec Jean-Pierre Merlet. En effet, pour des problèmes de contrôlabilité du robot, il est important de ne pas s'approcher de ces singularités; ce qui pose la question de la distance aux singularités. En effet, il est difficile d'associer une métrique à la valeur du déterminant. Comme décrit en début de section, le robot est soumis à des efforts internes élevés, dès lors qu'elles sont "proches" d'une singularité. Il est alors possible de fixer une limite physique à ces efforts données afin d'"éviter toute détérioration du robot". Ainsi, le problème de singularité peut être vu de façon plus juste (c'est à dire avec un sens physique et non plus mathématique). On se ramène alors à une étude statique de l'espace de travail qui revient à vérifier les solutions d'un système linéaire. Ce problème est mieux maîtrisé par la communauté comme nous l'avons vu en section 3.1.4.

Ainsi cet exemple illustre le travail fourni en robotique en choisissant les problèmes compatibles avec l'utilisation de l'analyse par intervalle, comment les formuler pour mettre en valeur ces approches tout en étant pertinent dans

la forme de la solution fournie. D'autres exemples seront présentés dans les chapitres suivants.

3.2.3 Adaptations des outils

Comme nous l'avons vu en section 3.1.3, l'évaluation en elle-même d'une contrainte intervalle est un enjeu important : la boîte résultante doit contenir, au plus juste, l'image des ensembles définis par des variables intervalles par les contraintes. La surestimation a de lourdes conséquences notamment sur le temps de calcul et une complexité alourdie. Ainsi, l'une des caractéristiques du travail de développement en robotique est de choisir :

R.1 un outil de modélisation mathématique adapté. Typiquement, nous avons à notre disposition de nombreuses façons de représenter la position et l'orientation d'un solide dans l'espace : toutes ne sont pas équivalentes en terme de surestimation par l'arithmétique d'intervalle.

Exemple : Considérons un robot mobile non-holonome à 2 degrés de liberté obtenus par une commande à deux roues actionnées. Le robot peut soit tourner sur lui-même d'un angle θ soit avancer d'une distance ρ . Si nous cherchons à décrire par intervalle sa position aux incertitudes près sur les consignes θ et ρ il est plus judicieux d'utiliser une représentation polaire de sa position qu'une représentation en x, y qui sur-estimera l'ensemble de position possible dans le plan par un rectangle, ceci à cause de l'effet enveloppant. Cette remarque est semblable à la représentation d'un complexe sur les intervalles. En terme d'effet enveloppant, une représentation algébrique des complexes est plus adaptée si nous souhaitons décrire la somme de deux complexes tandis qu'une représentation polaire décrira de façon plus précise le produit.

R.2 une forme symbolique qui diminue la surestimation de l'arithmétique par intervalle, par exemple en diminuant le nombre d'occurrence des variables.

Exemple : Un exemple simple est celui donné en section 3.1.3 sur le calcul de la composition des rotations successives : le calcul formel permet de le simplifier donner une forme optimale à l'évaluation, [PD07].

R.3 des outils avancés. L'analyse par intervalle est un domaine encore jeune et peu enseignée. Ce domaine se développe et de plus en plus d'applications font appel à ces techniques pour fournir des solutions pertinentes. Il existe de nombreuses solutions à tester pour l'évaluation (différent type d'extensions de Taylor, de Hansen etc...), pour la résolution de systèmes linéaires ou non, pour la prise en compte de la dépendance entre les paramètres. Cependant le domaine reste en construction et les interactions vertueuses entre développements théoriques et applicatifs le nourrisse. Ainsi, il nous a été nécessaire de nous intéresser à des outils basiques d'algèbre linéaire ou pour la résolution de systèmes bien/sous/sur-contraints afin de répondre à nos besoins de méthodes pour la robotique. Ceci explique les développements décrits en section 5.4.3 et 4.3.3.

Ces choix interagissent entre eux. C'est donc un travail d'évaluation difficile et malheureusement assez peu automatisable.

Modélisation mathématique des problèmes en robotique

Nous présentons ici un exemple particulier d'une problématique fondamentale en robotique concernant l'évaluation des chaînes cinématiques. En effet, cela illustre des efforts nécessaires pour rendre notre approche efficace. L'évaluation de ces équations dont la forme est générique est un élément basique dans nombre de problèmes pour les manipulateurs, les exosquelettes, l'analyse de mouvement humain etc... Cette chaîne cinématique est habituellement décrite en utilisant une modélisation de type Denavit-Hartenberg. Avec l'aide de T. Adelman, étudiant de Master, nous avons développé une évaluation par intervalles adaptée à ces équations géométriques. La difficulté est de ne pas perdre un certain nombre de propriétés de ces chaînes décrites par des produits de matrices homogènes. En effet, l'effet enveloppant de l'arithmétique d'intervalles rend son extension aux intervalles très pessimiste en ne prenant pas en compte la corrélation entre certaines variables. De plus, certaines propriétés sont perdues, par exemple que le produit de deux matrices de rotation intervalles fournit une matrice sur-estimant l'ensemble des composés des deux matrices mais perd ses propriétés. Il est alors nécessaire de réintroduire dans le système d'équations ces propriétés au travers de contraintes supplémentaires. Nous avons montré comment utiliser des contracteurs ou méthodes de filtrages pour minimiser cet effet tout au long de l'évaluation de ces chaînes. Les résultats ont été utilisés pour la conception de chaînes série pour un micro-robot dans [ODM09]. Cependant le sujet reste d'actualité, dans le cadre de la thèse de P-O Martin, nous allons le reprendre afin d'évaluer les performances de l'humain dont la modélisation fondée sur une représentation cinématique de son squelette. Plusieurs pistes sont à prendre telles celles de [BDJK03], ou plus récemment [PP13, PRK⁺13] ou [LRA15] qui utilise des résultats de [Gol09].

Utilisation du calcul formel

Le calcul formel est souvent nécessaire pour obtenir une forme efficace d'évaluation des contraintes, voir nos contributions dans [PD07, PD08]. Nous pouvons résumer cela en quelques points qui ont été étudiés dans chacun des articles auxquelles j'ai participé :

- Travailler une forme symbolique des contraintes favorables à l'arithmétique par intervalles : factoriser, minimiser le nombre d'occurrences des variables intervalles,
- Tester différentes modélisations mathématiques du problème. Il peut être aussi intéressant de simplifier la structure des systèmes en introduisant de nouvelles variables. Prenons un exemple simple, il peut être intéressant de remplacer les fonctions trigonométriques par des variables $C_\alpha = \text{Cos}(\alpha)$, $S_\alpha = \text{Sin}(\alpha)$ au prix d'un ajout de contraintes supplémentaires $C_\alpha^2 + S_\alpha^2 - 1 = 0$. Cette solution permet de simplifier les équations du système au prix d'une inconnue et d'une contrainte supplémentaires : les algorithmes de filtrages deviennent plus efficace. Nous avons utilisé ce type d'astuce par exemple dans [ODM09].
- Pré-calculer des termes intermédiaires, des jacobiniennes des systèmes etc. Comme nous l'avons vu, l'un des problèmes importants est la prise en compte des dépendances entre les paramètres. Nous pouvons le considérer, par exemple en développant formellement des algorithmes afin de

simplifier leur forme ou d'appliquer une extension de type Taylor pour le calcul des coefficients. Ceci qui a été appliqué à l'algorithme d'élimination de Gauss pour résoudre un système linéaire considérant une matrice paramétrée dans [MD05].

Cependant, cette démarche peut avoir ses limites : le pré-calcul symbolique d'une solution n'est pas forcément la bonne approche. Nous pouvons donner comme exemple le calcul du modèle géométrique direct d'un robot parallèle 6SPS de type 3-2-1, voir [GDP⁺07], où les incertitudes sur les paramètres géométriques et sur les variables articulaires sont prises en compte. La forme symbolique de ce modèle est suffisamment simple pour obtenir analytiquement les solutions. Cependant, nous pouvons montrer qu'une extension naturelle ce calcul donne un encadrement des résultats fortement sur-évalué. Dans [GDP⁺07, GDP06] C. Grandon montre qu'il est nécessaire d'utiliser des algorithmes de filtrages, appliqué au système de base (c'est à dire sans pré-calcul des solutions) afin de diminuer très fortement le volume de la boîte résultat encapsulant les ensembles solutions.

Prise en compte des quantificateurs

Comme nous l'avons vu en remarque de la section 3.1.4 et en section 3.1.5, les quantificateurs associés aux variables jouent un rôle essentiel pour décrire :

- Certains types de problèmes pour lesquels un des paramètres doit être quantifié universellement afin par exemple de garantir une commande pour une tolérance donnée,
- De caractériser des ensembles notamment des boites intérieures tel que nous l'avons vu en section 3.1.5.

La théorie des intervalles modaux considère ces quantificateurs, elle fournit une interprétation plus riche que les intervalles classiques. Cependant, en dépit de ses applications prometteuses, elle est peu utilisée, en raison de sa construction compliquée. Nous avons proposé dans [GDRT05] une nouvelle formulation de cette théorie qui utilise des intervalles généralisées et permet de manipuler ces propositions quantifiées avec des intervalles bornés mais dont l'ordre n'est pas contraint.

Cette nouvelle formulation rendre plus facile à comprendre et à utiliser la théorie, tout en nous permettant d'introduire de nouveaux développements : à savoir, un processus de linéarisation qui est compatible avec les concepts de la théorie des intervalles modaux. Cela a conduit à un nouvel outil d'analyse : une nouvelle extension de la valeur moyenne à intervalles généralisés qui est illustrée dans le papier [GDRT05]. Cependant, afin d'être efficace, les extensions de valeur moyenne doivent être appliquées à des intervalles suffisamment petits de sorte que la matrice jacobienne intervalle impliquée soit régulière. Ceci nous a conduit à proposer une implantation en C d'une librairie (BiasK) qui étend l'arithmétique d'intervalles classique aux intervalles généralisées. Cette arithmétique permet alors d'utiliser des variables quantifiées universellement en plus de celle quantifiée existentiellement. La structure du code est similaire et compatible avec la bibliothèque bien connue BIAS/Profil.

Cette approche est élégante, cependant, nous n'avons pas pu trouver d'exemple clair pour lequel les intervalles classiques n'offrent pas de formulation possible afin de caractériser les ensembles recherchés et quantifiées. Nous n'avons alors pas continué sur cette voie.

3.3 Résolution de système, contributions à l'analyse par intervalles

Nous avons proposé plusieurs contributions sur le développement de techniques d'analyses par intervalles comprenant aussi des résultats pour la résolution de systèmes non-linéaires : polynomiaux dans [LMR⁺05], pour les systèmes sur-contraints [DAP05, ADSTDG13, DACP06a, AdSDT14], et les moyens de les implanter [PD08, PDM03, PD08].

Nous nous attarderons sur le problème des valeurs propres réelles de matrices intervalles. Ce travail, important en pratique, a été traité dans [HDT10, HDT11b, HDT11a, HDT11c].

3.3.1 Cas bien contraint

Dans [LMR⁺05], j'ai participé à une contribution commune principalement porté par Yahia Lebbah, Claude Michel et Michel Rueher. Cette méthode permet de déterminer les zéros d'un système d'équations polynomiales.

Revenons à la méthode de Newton par intervalles. Pour schématiser son fonctionnement cette méthode peut être divisée en deux étapes permettant deux approximations d'un système non-linéaire par intervalle : l'extension de Taylor permet d'approximer l'ensemble solution par un système linéaire d'inégalité généralement non-convexe ; puis la résolution de ce système *linéaire par intervalle* passera généralement par une seconde approximation afin de le transformer en un (ou plusieurs) système(s) convexe(s). Le schéma est alors répété pour contracter ces approximations linéaires autour d'une solution possible. Utilisée dans un *Branch and Bounds*, nous pouvons alors déterminer l'ensemble des solutions possibles, voir un continuum de solutions, si les systèmes sont paramétrés par des intervalles.

Le principe de la méthode proposée dans [LMR⁺05] reprend ce schéma général en cherchant à approximer les solutions dans un polyèdre convexe puis de calculer les bornes sur chacune des dimensions en se ramenant à un problème de programmation linéaire. Le point clé est alors de déterminer les relaxations linéaires à partir des polynômes initiaux. La première étape dite de *quadrification*, voir [], consiste à réécrire chaque polynôme par un système d'équation équivalant qui introduit de nouvelles variables fonctions de termes quadratiques en les inconnues. Il est alors possible d'approximer ces contraintes quadratiques par des inégalités linéaires (donc convexe) suivant un schéma donné dans le papier. Ce nouveau type de contacteur local (qui ne s'applique que sur une équation indépendamment des autres) sert alors de méthode de filtrage dans un algorithme de type *Branch and Bound*. Ce papier montre qu'il est possible de proposer des solutions pertinentes en mixant des approches de programmation par contrainte et d'analyse par intervalles.

Il est à noter que cette méthode peut aussi être utilisée pour fournir une solution aux systèmes sur-contraints comme nous allons le voir à la section suivante.

3.3.2 Cas sur-contraint

Dans le cas où nous avons plus d'observations du comportement d'un système que d'inconnues $x_{n \times 1}$, à identifier nous obtenons un *système sur-contraint*

d'équations $F_{m \times 1}(x)$ avec $m > n$.

Dans le cas scalaire, ce type de système n'offre, généralement pas, de solutions telles que $\forall k \in \{1, \dots, m\}, F_k(x) = 0$. En effet, si nous considérons $F_k(x, M)$ comme incertain (erreurs de mesures des coefficients M , erreur de modèles etc), les m équations ne peuvent être liées entre elles. Nous devons donc caractériser de façon différente des solutions à ce problème, ce qui se fait en général par une caractérisation aux moindres carrées.

Dans le cas intervalle, le problème est différent. En effet, les fonctions épaissies $\mathbf{F}_k(x, \mathbf{M})$ représente des ensembles : il est donc possible de trouver des solutions communes à leurs intersections, même dans le cas sur-contraint, décrites par :

$$\{x | \forall k \in \{1, \dots, m\}, \exists M \in \mathbf{M}; F_k(x, M) = 0 \quad (3.12)$$

Plusieurs méthodes par intervalle permettent alors d'approximer cet ensemble par un pavage englobant.

Dans le cas où cet ensemble est vide, nous pouvons alors considérer une caractérisation aux moindres carrés par intervalles. Ces problèmes ont été étudiés spécifiquement pour l'étalonnage des robots. Nous préférons donc présenter les résultats dans la section consacrée 5.4.3.

3.3.3 Cas sous-contraint

L'expérience que nous avons dans ce domaine est principalement liée à l'étude de l'espace de travail des robots à câbles. Nous reviendrons donc sur ce problème à la section 4.3.3.

3.3.4 Cas des valeurs/vecteurs propres ou des valeurs singulières d'une matrice intervalle

J'ai initié, participé et dirigé les recherches sur l'analyse des valeurs/vecteurs propres ou des valeurs singulières d'une matrice intervalle. Ces travaux sont publiés, avec l'aide d'un post-doctorant, M. Hladik dans 4 articles de journal.

Les valeurs/vecteurs propres/singuliers sont intéressants pour la détermination des propriétés des systèmes mécaniques, biologiques, automatiques, etc. En ce qui concerne la robotique, voir [SK08], de nombreux indices de performance comme la dextérité, la manipulabilité, l'isotropie les utilisent mais elles sont aussi utiles pour déterminer les solutions d'un système d'équation [MC94] comme nous l'avons fait dans [DE04], ou dans [DPM05] pour déterminer des configurations de mesure pour l'étalonnage. De façon simple, déterminer les valeurs propres ou singulières est un des méthodes les plus intéressants pour comprendre les systèmes paramétrés ou incertains. Des approches par intervalles ont été utilisées pour l'analyse de systèmes de suspensions automobiles [QMF01], de structure [QCE96], de vibrations [Dim95], des propriétés des robots [CWMM04], et même analyse en composantes principales [GL06] et de l'analyse en composantes indépendantes [Com94]. Un état de l'art un peu ancien est donnée dans [Sha00].

Ce sujet est donc un enjeux important en analyse par intervalles et se développe depuis les années 90. Les premiers résultats sont données par [Dei91] pour le calcul des valeurs propres complexes puis réelles dans [RD92]. Ces résultats sont conditionnés par des hypothèses fortes sur les vecteurs propres. Des

approximations de l'ensemble des valeurs propres sont fournies dans [QMF01] après avoir utilisé des approches de perturbation de matrices.

Le problème de la détermination des valeurs propres réelles d'une matrice par intervalles \mathbf{A} est décrit par l'ensemble suivant :

$$\mathbf{\Lambda} = \{\lambda \in \mathbb{C} \text{ ou } \mathbb{R} \mid A.x = \lambda.x, x \neq 0, A \in \mathbf{A}\} \quad (3.13)$$

Remarquons que nous nous sommes concentrés sur le problème des valeurs propres réelles qui nous permet de bien répondre aux demandes en terme de robotique. De plus, ce problème permet le calcul des valeurs singulières d'une matrice $A_{m \times n}$, puisqu'elles sont identiques aux $q = \min(m, n)$ plus grandes valeurs propres de la matrice de Jordan-Wielandt.

$$\mathcal{A} = \begin{pmatrix} 0 & \mathbf{A}^T \\ \mathbf{A} & 0 \end{pmatrix} \quad (3.14)$$

Nous noterons la matrice milieu $A_c = \frac{1}{2}(\underline{A} + \overline{A})$ et la matrice rayon $A_\Delta = \frac{1}{2}(\overline{A} - \underline{A})$

Borner les valeurs propres et améliorations dans le cas symétrique

Dans un premier article [HDT10] nous avons cherché à améliorer un théorème de Rohn [Roh98] qui borne $\mathbf{\Lambda}$ en utilisant les valeurs propres et le rayon spectral de matrices scalaires calculées à partir des matrices milieu et rayon.

Pour cela nous avons utilisé le théorème de Bauer-Fike [GVL12] puis une généralisation de Chu [Chu86] qui permet, après une adaptation aux intervalles et quelques améliorations, d'obtenir de meilleurs bornes que celles de Rohn, pour plus de précision voir [HDT10].

Si maintenant, la matrice intervalle \mathbf{A} est symétrique (noté \mathbf{A}^S), nous pouvons la encore améliorer les propositions de Rohn, adaptées à ce cas particulier. Pour cela, on adapte le théorème de Courant-Fischer [GVL12]. L'astuce supplémentaire revient alors à utiliser la propriété d'entrelacement de Cauchy des valeurs propres des matrices symétriques afin de proposer un algorithme qui permet de borner chacune d'entre elles. De nombreuses variantes, améliorations et comparaisons sont présentées dans l'article.

Algorithmes de *branch and bound*

Le second article [HDT11b] présente plusieurs algorithmes pour décrire (ou paver) l'ensemble $\mathbf{\Lambda}$ fondés sur une approche de type *branch and bound* présentée en section 3.1.5.

Nous avons alors proposé plusieurs tests afin de vérifier qu'un ensemble intervalle noté λ_k est

- une boîte extérieure λ_{out}^k , c'est à dire qu'elles ne contiennent pas des valeurs possibles de $\mathbf{\Lambda}$
- une boîte intérieure λ_{in}^k , c'est à dire que cet ensemble est contenu dans $\mathbf{\Lambda}$

En ce qui concerne le test de boîte extérieure, il repose essentiellement sur la proposition simple suivante : si la matrice intervalle $\mathbf{M} = \mathbf{A} - \lambda_t.I$ est régulière alors le vecteur intervalle λ_t est une boîte extérieure. Ainsi, ce test se réduit à vérifier la régularité de la matrice \mathbf{M} . Pour cela nous avons sélectionné, un test

de Rex et Rohn [RR98] offrant une condition suffisante forte à la régularité des matrices intervalles puis un algorithme de Jansson et Rohn de [JR99] fondé sur l'existence d'une solution à un système linéaire si la matrice est régulière.

Nous avons aussi proposé notre propre test fondé sur la proposition suivante : \mathbf{M} est régulière si et seulement si le système $\mathbf{M}.x = 0$, $x \neq 0$ n'admet pas de solution. Nous avons alors joué sur la nécessité de normaliser le vecteur x (qui correspond aux vecteurs propres) pour utiliser une norme infinie, $\|x\|_\infty$, $x_k = 1$ pour un $k = 1, \dots, n$. Ceci nous permet de mettre le système sous la forme d'un système linéaire intervalle sur-contraint de type : $\mathbf{M}_k.x_{1,\dots,k-1,k+1,\dots,n} = -\mathbf{m}_k$ avec $\mathbf{M}_k = [\mathbf{m}_1, \dots, \mathbf{m}_{k-1}, \mathbf{m}_{k+1}, \dots, \mathbf{m}_n]$, \mathbf{m}_k étant la k -ième colonne de \mathbf{M} . Si ce système ne possède pas de solution alors l'intervalle λ_t choisi pour construire la matrice $\mathbf{M} = \mathbf{A} - \lambda_t.I$, ne contient pas de valeurs propres de \mathbf{A} ; sinon le vecteur X est une approximation englobante des vecteurs propres de \mathbf{A} . L'enjeu de cette approche est la solution utilisée pour résoudre le système sur-contraint. Attention, ce ne peut-être une solution au sens des *Moindres-carrés* mais bien de déterminer une solution commune à toutes les équations linéaires du système ce qui a un sens dans le cas intervalle. Nous avons utilisé les mêmes techniques que nous avons développées pour résoudre le problème de l'étalonnage des robots parallèles en utilisant des techniques par intervalles, voir [DACP06b].

Enfin, nous avons proposé deux autres algorithmes. Le premier détermine si la boîte considérée est intérieure; pour cela nous avons adapté le test de Jansson et Rohn de [JR99]. Le second algorithme permet de vérifier si les bornes calculées sont exactes (aux erreurs d'arrondis près). Ces tests et algorithmes sont alors utilisés dans une algorithme général de *branch and bound* afin de paver les valeurs propres réelles d'une matrice intervalle. Un petit exemple en robotique vient illustrer l'article.

Une méthode de filtrage

Dans l'article [HDT11a], nous proposons une méthode de filtrage pour réduire la surestimation produite par diverses méthodes présentées précédemment. En règle générale, les méthodes de filtrage commencent par une approximation externe initiale puis l'améliore en utilisant une approche itérative (on parlera plutôt de contraction). Même si le filtrage est couramment utilisé en programmation par contraintes, il est peu utilisé pour le problème de valeurs propres intervalles. Nous pouvons, bien sûr, appliquer un algorithme de filtrage classique tel que présenté précédemment, sur le système d'équations des valeurs propres. Nous avons testé dans [HDT11b] une approche par Newton sans réel succès.

Au meilleur de notre connaissance, le seul travail connexe est de celui de Beaumont, où un algorithme itératif est présenté, sur la base approximation convexe de valeurs propres [Bea00]. La nouvelle méthode de filtrage que nous proposons est plus simple et applicable à la fois dans le cas de matrice symétrique ou non. Elle est aussi plus efficace car elle ne prend pas en compte les vecteurs propres pour filtrer directement les valeurs propres.

Pour notre filtre, nous utilisons des conditions suffisantes sur la régularité des matrices proposées par [Bee75] et [Rum83]. Nous montrons comment obtenir une borne autour de chaque valeur extrême d'une approximation externe (elles ne font donc pas partie de $\mathbf{\Lambda}$) qui définit un ensemble. Nous montrons que cet ensemble ne peut contenir de valeurs propres appartenant à l'une des matrices de \mathbf{A} .

Dans ce sens la méthode est un algorithme de type rognage qui permet d'éliminer des parties extrêmes d'une approximation externe. Les résultats montrent une grande amélioration dans le calcul des bornes présenté en section 3.3.4.

3.3.5 Tests pour les boites intérieures pour les matrices symétriques

Le cas des matrices symétriques noté \mathbf{A}^S est intéressant en robotique car les calculs de nombreux indices de performances sont fondés sur des propriétés de ce type de matrice. De plus leur structure particulière permet de simplifier les approches. Ceci est important car si nous utilisons des méthodes générales, comme précédemment, les dépendances entre les coefficients de la matrice ne sont pas pris en compte et les algorithmes sur-estiment alors les résultats.

Plusieurs études permettent de réaliser cet objectif. Hertz [Her92] a proposé une formule pour calculer les points extrêmes de l'ensemble des valeurs propres. Mais, au cours des dernières années, plusieurs algorithmes d'approximation ont été développés. Au moyen de la théorie de perturbation des matrices, Qiu et al. [QCE96] ont proposé un algorithme afin d'approximer les bornes de l'ensemble recherché, et Leng et He [LHY08] proposent de fournir une enveloppe externe par un algorithme de type "Branch and Bounds". Une estimation externe a également été examinée par Kolev [Kol06] dans le cas général, mais il prend en compte des dépendances non-linéaires entre les coefficients. À noter un algorithme itératif pour l'estimation externe a été donnée par Beaumont dans [Bea00].

Dans notre article [HDT11c], nous nous concentrons sur les approximations internes (pavage interne) des ensembles de valeurs propres. Il y a beaucoup moins d'articles consacrés à l'approximation intérieure. Citons une méthode de stratégie d'évolution par Yuan et al. [YHL08] ou une méthode générale pour les systèmes non-linéaires dans [Rum10].

Dans notre article [HDT11c], les propositions sont fondées sur un théorème qui permet de décrire les bornes des valeurs propres de la matrice symétrique \mathbf{A} à partir de valeurs propres calculées à partir des matrices extrêmes A_z mais aussi de sous-matrices dites principales : c'est à dire en éliminant des lignes et colonnes correspondantes de \mathbf{A} .

Les matrices extrêmes sont donnée par :

$$A_z = A_c + \text{diag}(z).A_\Delta.\text{diag}(z), \quad z \in \pm 1^n, \quad z_1 = 1 \quad (3.15)$$

Le premier algorithme présenté est repose sur une technique de recherche locale. L'idée est de commencer avec une valeur propre associée à la matrice milieu, et du vecteur propre correspondant, puis ensuite de passer à une matrice extrême A_z dans \mathbf{A} selon le modèle orienté par le signe du vecteur propre (voir théorème 1 de [HDT11c]). La procédure est répétée jusqu'à ce qu'aucune amélioration ne soit plus possible. La deuxième méthode présentée est fondée sur l'énumération complète des matrices extrêmes.

Afin, la troisième méthode utilise pleinement le théorème 1 de [HDT11c]. Il produit habituellement des approximations internes très serrées, même exactes dans certains cas. L'idée de base qui sous-tend l'algorithme consiste à énumérer tous les sommets de toutes les sous-matrices principales de \mathbf{A} y compris \mathbf{A} elle-même. Ainsi, nous passons par plus de matrices que les méthodes précédentes avec une complexité plus élevée. L'algorithme est alors amélioré en

terme de complexité (exponentielle dans ce cas), en utilisant une approche de type "Branch and Bound" afin d'éliminer certains cas.

Les trois méthodes donnent de très bon résultats au regard des exemples fournis par la littérature. Les gains en termes de précision sont importants entre les méthodes 1, 2 et la méthode 3 mais au prix d'un temps de calcul largement supérieur.

3.4 Diffusion des contributions dans le domaine de l'analyse par intervalles

Nous verrons comment nous avons utilisé l'analyse par intervalles pour répondre au problème de la conception dimensionnelle des robots manipulateurs puis afin d'offrir des alternatives aux méthodes classiques pour l'étalonnage des robots. Ce ne sont que quelques exemples, parmi tant d'autres, de problèmes qui peuvent bénéficier de ces approches. Nous avons remarqué que le travail de formulation des problèmes ou le choix des outils mathématiques de modélisation ont une influence sur l'efficacité de ces méthodes. De plus, il est souvent nécessaire de varier les approches. En effet, suivant la forme des contraintes, le diamètre des coefficients intervalles, les méthodes de filtrage sont plus ou moins efficaces (contraction autour de la solution). Malheureusement, même si de gros efforts sont faits par la communauté, il n'existe pas de méthodes universelles qui fonctionnent quel que soit le problème traité. Il est donc essentiel d'offrir à un utilisateur non expert un ensemble de méthodes soit génériques mais modulaires (par exemple qui permettent de tester différentes méthodes de filtrage) soit qui répondent à un problème clairement identifié par exemple, l'évaluation des valeurs propres ou d'un déterminant d'une matrice intervalles, la résolution de polynôme à coefficients intervalles, etc ...

Afin d'offrir ces outils et ainsi de favoriser la diffusion de l'analyse par intervalle, l'équipe-projet Coprin a fait d'important efforts pour permettre d'utiliser des méthodes d'analyse par intervalles en proposant la bibliothèque ALIAS [Mer00]. Nous avons alors proposé de nombreux outils :

- pour l'évaluation. Nous avons implanté les extensions décrites en section 3.1.3 mais d'autre type d'évaluation : par exemple en considérant la monotonie des contraintes, pour les polynômes, pour le calcul du déterminant d'une matrice intervalle en prenant en compte la dépendance paramétrée entre ses coefficients etc ...
- pour une résolution générique de systèmes. L'utilisateur a accès à plusieurs types de solvers en fonction de sa capacité à fournir la Jacobienne ou l'Hessienne du système. La connaissance de ces matrices permet généralement d'augmenter l'efficacité des méthodes. La gestion des paramètres du "Branch and Bound" sont accessibles.
- pour une résolution adaptée d'un système de forme spécifique. La forme des contraintes (ou équations) sont alors prise en compte : trigonométrique, polynôme univarié, équations de distance ...
- pour l'analyse des systèmes : bornes sur le nombre racines, test d'unicité, test de régularité et calcul de valeurs propres etc.

Nous vous conseillons de vous référer à la documentation pour le détails des

fonctions disponibles³. Il est à noter que ces outils peuvent être parallélisés pour une meilleure efficacité calculatoire comme nous l'avons montré dans [PDM03].

Cette bibliothèque est efficace et offre de très nombreuses possibilités. Cependant, dans ma communauté (robotique), de nombreux chercheurs ou ingénieurs ne peuvent pas forcément s'investir dans une programmation bas-niveau de leur algorithme. Ceci peut être un frein à la diffusion de l'analyse par intervalle. Nous avons donc décidé de proposer des outils d'interface de haut-niveau. Le projet Coprin a donc développé trois interfaces pour ALIAS : en Maple, pour ses capacités de traitement formel des systèmes, en Mathematica ([PD08]) pour des raisons similaires et en Scilab pour sa diffusion dans notre communauté, pour l'enseignement (coût des licences). J'ai principalement contribué sur les deux dernières interfaces.

Tout d'abord, j'ai initié, développé et mis en place une toolbox Scilab offrant aux utilisateurs une arithmétique d'intervalles mais aussi la possibilité d'avoir accès à des modules de plus hauts niveaux (résolution de polynômes, résolution de systèmes linéaires ...), [Dan10]. Cette action a été soutenue par le CRISAM avec la mise à disposition d'un ingénieur associé, R. Pereira, pendant un an. Une approche similaire est toujours en développement pour Mathematica, dont les prémisses sont données dans [PD08, PD07].

3.5 Conclusion

L'analyse par intervalle est un outil intéressant qui propose des solutions originales et puissantes pour nombre de problèmes. Nous verrons dans les chapitres suivants son utilisation pratique en robotique.

Même si le domaine est de plus en plus accepté comme une alternative crédible à nombre de méthodes, il reste assez *jeune*. De nombreux problèmes basiques pour la résolution de systèmes, l'optimisation et l'algèbre linéaire étendue aux intervalles restent à approfondir. J'ai participé aux développements dans ce domaine au sein de l'équipe Coprin, ce qui a été très formateur. La direction du projet qui a réussi à mêler des techniques de programmation par contraintes avec d'autres plus classiques d'analyse par intervalles a montré qu'il était possible de traiter des problèmes de taille réaliste dans des temps acceptables en pratique. L'historique du parcours de recherche montre bien que les aspects algébriques et le développement formel des contraintes et équations sont essentiels pour maximiser l'efficacité des méthodes. Plus encore, les connaissances en terme de représentation géométrique et mécanique des problèmes de robotique jouent un rôle important dans les choix de modélisation mathématique adaptés à l'analyse par intervalle.

Ainsi, il est difficile de séparer les rôles entre communauté. Les outils manquants notamment sur le traitement des systèmes sur-contraints et le calcul des valeurs/vecteurs propres réelles des matrices intervalles m'ont poussé à développer ces contributions et à diriger des recherches dans ce sens. C'est le besoin applicatif qui a dicté ces choix.

Il reste de nombreuses questions à traiter et il est clair que de beaux jours sont à venir pour l'analyse par intervalles. Tout au long de ce chapitre et dans les chapitres suivants, nous avons insisté sur l'utilisation du calcul formel, de la

3. voir <http://www-sop.inria.fr/teams/hephaistos/logiciels/ALIAS/>

prise en compte des dépendances entre les paramètres et de la vérification de la monotonie afin de mieux évaluer les contraintes.

Ces voies de recherches restent intéressantes et il est important de continuer à creuser dans ce sens. Les travaux de Milan Hladik, après son passage à inria, sur la prise en compte de la dépendance des paramètres dans les algorithmes de base (système linéaire etc...) vont dans ce sens ce qui est une très bonne chose pour nous, roboticien.

Bibliographie

- [AdSDT14] Julien Alexandre dit Sandretto, David Daney, and Gilles Trombettoni. Interval Methods for Model Qualification : Methodology and Advanced Application. *Mathematics in Computer Science*, 2014.
- [ADSTDG13] Julien Alexandre Dit Sandretto, Gilles Trombettoni, David Daney, and Chabert Gilles. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *6th Int. Workshop on Computational Kinematics (CK2013)*, volume 15 of *Mechanisms and Machine Science*, pages 209–217, Barcelona, Spain, Espagne, 2013. Springer.
- [BDJK03] Xavier Baguenard, Massa Dao, Luc Jaulin, and Wisama Khalil. Méthodes ensemblistes pour l'étalonnage géométrique. *JOURNAL EUROPEEN DES SYSTEMES AUTOMATISES.*, 37(9) :1059–1074, 2003.
- [Bea00] Olivier Beaumont. *An algorithm for symmetric interval eigenvalue problem*. IRISA, 2000.
- [Bee75] Helmut Bееck. Zur problematik der hüllenbestimmung von intervallgleichungssystemen. In *Interval Mathematics*, pages 150–159. Springer, 1975.
- [Cha07] Gilles Chabert. *Techniques d'intervalles pour la résolution de systèmes d'équations*. PhD thesis, Université Nice Sophia Antipolis, 2007.
- [Chu86] King-wah Eric Chu. Generalization of the bauer-fike theorem. *Numerische Mathematik*, 49(6) :685–691, 1986.
- [Com94] Pierre Comon. Independent component analysis, a new concept ? *Signal processing*, 36(3) :287–314, 1994.
- [CWMM04] Damien Chablat, Ph Wenger, Félix Majou, and J-P Merlet. An interval analysis based study for the design and the comparison of three-degrees-of-freedom parallel kinematic machines. *The International Journal of Robotics Research*, 23(6) :615–624, 2004.
- [DACP06a] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval Method for Calibration of Parallel Robots : A Vision-based Experimentation. *Mechanism and Machine Theory*, 41(8) :929–944, 2006.

- [DACP06b] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval method for calibration of parallel robots : Vision-based experiments. *Mechanism and Machine Theory*, 41(8) :929 – 944, 2006.
- [Dan10] David Daney. Int4Sci : Scilab Interface for Interval Analysis. In *14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, Lyon, France, 2010.
- [DAP05] David Daney, Nicolas Andreff, and Yves Papegay, A. Interval method for calibration of parallel robots. In *Computational Kinematics (CK)*, Cassino, Italie, 2005.
- [DE04] David Daney and Ioannis Emiris. Algebraic Elimination for Parallel Robot Calibration. In *World Congress of International Federation for the Theory of Machines and Mechanisms (IFToMM)*, pages 212–217, Tianjin, Chine, 2004.
- [Dei91] Assem Deif. The interval eigenvalue problem. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 71(1) :61–64, 1991.
- [Dim95] Andrew D Dimarogonas. Interval analysis of vibrating systems. *Journal of Sound and Vibration*, 183(4) :739–749, 1995.
- [DPM05] David Daney, Yves Papegay, and Blaise Madeline. Choosing Measurement Poses for Robot Calibration with the Local Convergence Method and Tabu Search. *International Journal of Robotics Research*, 24(6) :501–518, 2005.
- [GDP06] Carlos Grandón, David Daney, and Yves Papegay. Combining CP and Interval Methods for solving the Direct Kinematic of a Parallel Robot under Uncertainties. In *Workshop of International Conference on Principles and Practice of Constraint Programming (IntCP 06)*, Nantes, France, 2006.
- [GDP⁺07] Carlos Grandón, David Daney, Yves Papegay, Cristina Tavolieri, Erika Ottaviano, and Marco Ceccarelli. Handling Uncertainties with Symbolic/Numerical Solvers for a Class of Parallel Robots. In *World Congress of International Federation for the Theory of Machines and Mechanisms (IFToMM)*, Besançon, France, June 2007.
- [GDRT05] Alexandre Goldsztejn, David Daney, Michel Rueher, and Patrick Taillibert. Modal intervals revisited : a mean-value extension to generalized intervals. In *In International Workshop on Quantification in Constraint Programming (International Conference on Principles and Practice of Constraint Programming, CP-2005)*, Barcelona, Espagne, 2005.
- [GL06] Federica Gioia and Carlo N Lauro. Principal component analysis on interval data. *Computational Statistics*, 21(2) :343–363, 2006.
- [Gol09] Alexandre Goldsztejn. On the exponentiation of interval matrices. *arXiv preprint arXiv :0908.3954*, 2009.
- [GVL12] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

- [HDT10] Milan Hladik, David Daney, and Elias Tsigaridas. Bounds on real eigenvalues and singular values of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4) :2116–2129, 2010.
- [HDT11a] Milan Hladik, David Daney, and Elias Tsigaridas. A filtering method for the interval eigenvalue problem. *Applied Mathematics and Computation*, 217(12) :5236 – 5242, 2011.
- [HDT11b] Milan Hladik, David Daney, and Elias Tsigaridas, P. An algorithm for addressing the real interval eigenvalue problem. *Journal of Computational and Applied Mathematics*, 235(8) :2715 – 2730, 2011.
- [HDT11c] Milan Hladik, David Daney, and Elias Tsigaridas, P. Characterizing and Approximating Eigenvalue Sets of Symmetric Interval Matrices. *Computers and Mathematics with Applications*, 62(8) :3152–3163, 2011.
- [Her92] David Hertz. The extreme eigenvalues and stability of real symmetric interval matrices. *Automatic Control, IEEE Transactions on*, 37(4) :532–535, 1992.
- [Hub10] Julien Hubert. *Static analysis of parallel manipulators singularities*. Theses, École Nationale Supérieure des Mines de Paris, September 2010.
- [JR99] Christian Jansson and Jiri Rohn. An algorithm for checking regularity of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(3) :756–776, 1999.
- [Kol06] Lubomir V. Kolev. Outer interval solution of the eigenvalue problem under general form parametric dependencies. *Reliable Computing*, 12(2) :121–140, 2006.
- [LHY08] Huinan Leng, Zhiqing He, and Quan Yuan. Computing bounds to real eigenvalues of real-interval matrices. *International journal for numerical methods in engineering*, 74(4) :523–530, 2008.
- [LMR⁺05] Yahia Lebbah, Claude Michel, Michel Rueher, David Daney, and Jean-Pierre Merlet. Efficient and Safe Global Constraints for Handling Numerical Constraint Systems. *SIAM Journal on Numerical Analysis*, 42(5) :pp. 2076–2097, 2005.
- [LRA15] Sergio Lescano, Micky Rakotondrabe, and Nicolas Andreff. Precision prediction using interval exponential mapping of a parallel kinematic smart composite microstructure. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1994–1999. IEEE, 2015.
- [MC94] Dinesh Manocha and John F Canny. Efficient inverse kinematics for general 6r manipulators. *Robotics and Automation, IEEE Transactions on*, 10(5) :648–657, 1994.
- [MD05] Jean-Pierre Merlet and David Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 942–947, Barcelona, Espagne, 2005. IEEE.

- [Mer00] Jean Pierre Merlet. Alias : an interval analysis based library for solving and analyzing system of equations. *SEA, June. Automation*, pages 1964–1969, 2000.
- [Mer04] J-P Merlet. Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. *The International Journal of robotics research*, 23(3) :221–235, 2004.
- [Moo66] Ramon E Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.
- [ODM09] Denny Oetomo, David Daney, and Jean-Pierre Merlet. Design Strategy of Serial Manipulators With Certified Constraint Satisfaction. *IEEE Transactions on Robotics*, 25(1) :1–11, 2009.
- [PD07] Yves Papegay and David Daney. Computing with Intervals : the Ultimate Symbolic Computation? In *Wolfram Technology Conference*, Illinois, États-Unis, 2007.
- [PD08] Yves Papegay, A. and David Daney. UnCertainties, a Package for Interval Analysis. In *9th International Mathematica Symposium*, Maastricht, Pays-Bas, 2008.
- [PDM03] Yves Papegay, David Daney, and Jean-Pierre Merlet. Parallel Implementation of Interval Analysis for Equations Solving. In Jack Dongarra, Domenico Laforenza, and Salvatore Orlando, editors, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, volume 2840 of *Lecture Notes in Computer Science*, pages 555–559. Springer Berlin Heidelberg, 2003.
- [PP13] Muhammed R Pac and Dan O Popa. Interval analysis of kinematic errors in serial manipulators using product of exponentials formula. *Automation Science and Engineering, IEEE Transactions on*, 10(3) :525–535, 2013.
- [PRK⁺13] Muhammed R Pac, Micky Rakotondrabe, Sofiane Khadraoui, Dan O Popa, and Philippe Lutz. Guaranteed manipulator precision via interval analysis of inverse kinematics. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V001T09A017–V001T09A017. American Society of Mechanical Engineers, 2013.
- [QCE96] Zhiping Qiu, Suhuan Chen, and Isaac Elishakoff. Bounds of eigenvalues for structures with an interval description of uncertain-but-non-random parameters. *Chaos, Solitons & Fractals*, 7(3) :425–434, 1996.
- [QMF01] Zhiping Qiu, Peter C Müller, and Andreas Frommer. An approximate method for the standard interval eigenvalue problem of real non-symmetric interval matrices. *Communications in numerical methods in engineering*, 17(4) :239–251, 2001.
- [RD92] Jiri Rohn and Assem Deif. On the range of eigenvalues of an interval matrix. *Computing*, 47(3) :373–377, 1992.
- [Rev01] Nathalie Revol. Arithmétique par intervalles. *Calculateurs Parallèles et Systèmes Répartis*, 13(4-5) :387–426, 2001.

- [Roh98] J. Rohn. Bounds on eigenvalues of interval matrices. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 78(S3) :1049–1050, 1998.
- [RR98] Georg Rex and Jiri Rohn. Sufficient conditions for regularity and singularity of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(2) :437–445, 1998.
- [Rum83] Siegfried M Rump. Solving algebraic problems with high accuracy. In *Proc. of the symposium on A new approach to scientific computation*, pages 51–120. Academic Press Professional, Inc., 1983.
- [Rum10] Siegfried M Rump. Verification methods : Rigorous results using floating-point arithmetic. *Acta Numerica*, 19 :287–449, 2010.
- [Sha00] Mohamed A Shalaby. The interval eigenvalue problem. In *ECCOMAS, European Congress on Computational Methods in Applied Sciences and Engineering*, 2000.
- [SK08] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.
- [YHL08] Quan Yuan, Zhiqing He, and Huinan Leng. An evolution strategy method for computing eigenvalue bounds of interval matrices. *Applied mathematics and computation*, 196(1) :257–265, 2008.

Chapitre 4

Conception appropriée des robots manipulateurs

4.1 Principes de la conception appropriée

La conception de robot est une problématique très importante pour notre communauté, dont les enjeux seront présentés en section 4.1.1. Afin d'y répondre, nous avons développé une méthodologie, maintenant éprouvée, qui utilise les approches intervalles décrites dans le chapitre précédent.

J'ai participé à la conception de cette méthodologie lors d'études pour la conception de certains manipulateurs, essentiellement plate-forme de Gough et robots à câble. Cette contribution est le résultat du programme de recherche de l'équipe Coprin : nous avons ainsi formé de nombreux étudiants et chercheur à cette méthodologie.

Elle est fondée sur la vérification ensembliste des propriétés des robots développées en section 4.2 puis sur l'utilisation de cette approche pour détermination de l'ensemble des paramètres vérifiant des performances souhaitées. Les principes de l'algorithme seront présentés en section 4.4.

4.1.1 Problématique

Les propriétés d'un système sont obtenues à partir d'un modèle qui établit des relations entre l'état du robot et des paramètres qui sont intrinsèques à sa structure.

Pour les manipulateurs dont l'architecture mécanique est fixée, les propriétés du robot sont donc fonction :

- des paramètres géométriques intrinsèques, typiquement la valeur des dimensions, au sens large, de ses composants ou de ses offsets. Ces paramètres sont considérés comme indépendants de l'état du robot (ou configuration) mais aussi, sous certaines hypothèses, des conditions environnementales.
- son état, donné par des consignes d'entrée (coordonnées articulaires) et/ou sa localisation (position/orientation) dans son espace de travail (coordonnées généralisées).

L'objectif de la conception dimensionnelle est de déterminer des valeurs des paramètres géométriques du robot afin, soit d'*optimiser* ses performances, soit d'assurer la vérification d'un cahier des charges. Il existe de nombreuses approches utilisées pour la conception des robots parallèles : méthode de type essai-erreur, algorithme génétique [ZWL05, SDPN01], méthode par atlas ou visuelle, ou le nombre de paramètres est limité [Cla91, BHG95, LWOK], ou par des méthodes d'optimisation à convergence locale [ZA97, OC02].

Quelles que soit les méthodes utilisées, les difficultés sont toujours :

- les critères de performances considérés : quel est leur pertinence, comment les choisir puis les calculer efficacement et exactement sur un espace de travail ... Ces critères n'étant le reflet que d'une propriété du robot, il est naturel de vouloir en considérer plusieurs. La solution est habituellement de rassembler les critères dans une fonction de coût unique qui pose le problème de la pondération entre des critères généralement hétérogènes.
- la dimension du vecteur des variables d'état et des paramètres à optimiser : les performances étant dépendantes de variables définissant la configuration du robot ou des conditions externes, il est intéressant d'évaluer les performances de façon globale, par exemple sur un espace de travail. Une solution simple est alors de discrétiser cet espace ce qui devient difficile voire impossible pour un espace de travail de grande dimension. Cette approche ne permet cependant pas de garantir une quelconque optimalité.
- les propriétés de la méthode d'optimisation choisie : par exemple la garantie d'une convergence globale d'un algorithme, ce qui n'est généralement pas le cas.
- la robustesse des critères aux incertitudes de fabrication et d'assemblage du robot. En effet, si la sensibilité de l'optimum trouvé aux variations des paramètres est grande, il est possible que le robot, construit à partir de ces résultats, n'ait pas les performances espérées.

Nous avons préféré reformuler le problème en terme de garantie de performance. En effet, l'expérience montre que les demandes des utilisateurs¹ sont données sous la forme d'une liste (souvent ordonnée) de propriétés attendues. Ces demandes peuvent être généralement modélisées sous forme de contraintes d'inégalités sans nécessairement exiger un optimum.

Il est donc question de pouvoir définir et évaluer une ou plusieurs performances pour un robot donné, c'est à dire pour une instance d'un système. Mais :

- Comment synthétiser une performance d'un système par un critère significatif? En effet, le critère qui est utilisé pour modéliser la performance d'un robot dont pouvoir être interprété physiquement, par un *client de la conception* pour être décrit dans un cahier des charges.
- Comment évaluer ce critère sur un ensemble de points de fonctionnement? Un manipulateur évolue sur une trajectoire ou dans un espace de travail; les conditions environnementales peuvent être variables : ce critère doit pouvoir être donné pour un ensemble de points.
- Comment propager les incertitudes dans l'évaluation de ce critère? Le robot réellement construit ne sera pas exactement celui imaginé lors de la conception, ceci est du à un certain nombre d'erreurs liées aux capteurs

1. Ceux qui vont utiliser les résultats de la conception dimensionnelle : bureau d'étude, constructeur de robot, etc.

proprioceptifs, au tolérances de la fabrication et l'assemblage ...

- Comment certifier la performance d'un système? Ceci pose la question d'une évaluation robuste des critères.

Si nous pouvons répondre à ces questions pour une instance de robot, pour chacune des performances décrites dans le cahier des charges, nous chercherons alors à généraliser l'approche de certification des performances, non pas pour une seule instance du robot mais pour un ensemble d'instances. Nous pouvons alors nous demander :

- Comment évaluer un ou plusieurs critères pour un ensemble d'état du système?
- Comment certifier qu'un ensemble d'états du système répond à un ensemble de critères?

4.1.2 Principe de la méthode

L'approche proposée par la conception appropriée a pour objectif de :

- **Déterminer un ensemble d'instances d'un système qui certifie un ensemble de performances souhaitées.** Contrairement aux approches classiques, nous ne cherchons pas une seule instance des paramètres intrinsèques mais plutôt un ensemble qui satisfasse les contraintes décrites dans le cahier des charges. Cette question sera vue en fin de chapitre à la section 4.4. En effet, il sera d'abord nécessaire d'être capable de traiter le défi suivant ...
- un sous-objectif est de **Certifier des performances d'un système.** Pour cela nous déterminons un espace de travail (un ensemble de points continus ou trajectoires) pour lesquels la ou les performances sont vérifiées et certifiées. Nous devons pouvoir le faire pour une instance du système (la valeur des paramètres intrinsèque est fixée) comme pour un ensemble d'instances du système. Nous approfondirons cette question à la section 4.2 de façon générique et à la section 4.3 sur les cas effectivement étudiés.

Par exemple, un constructeur de robot souhaite déterminer les dimensions de son manipulateur tel qu'il puisse réaliser :

- Propriété 1 : un espace de travail garanti en terme de position et orientation accessible par l'organe terminal du manipulateur,
- Propriété 2 : que cet espace de travail ne contienne pas de singularités,
- Propriété 3 : que, dans cet espace de travail, la précision de positionnement soit inférieur à une certaine valeur de façon garanti.

Chaque propriété considérée peut, avec plus ou moins d'effort, être modélisée au travers d'une ou plusieurs contraintes impliquant la performance souhaitée. Le problème revient donc à déterminer les solutions d'un système d'égalités et d'inégalités de dimension positive. Là encore, le problème difficile en raison du nombre important du nombre de variables à considérer en robotique : les paramètres du système/robot mais aussi ses différent points de fonctionnement (par exemple ses positionnements possibles au sein de son espace de travail).

Il conviendra donc d'être efficace dans l'évaluation de ses contraintes en cherchant à la faire, non pas sur un point, mais pour un ensemble de points. Pour cela nous utilisons une arithmétique d'intervalle [Moo66] qui permet de modéliser ces ensembles par des intervalles et d'*étendre aux intervalles* nos contraintes, c'est à dire de borner leur évaluation pour un ensemble de valeurs possible des

variables considérées — voir [HW03, Jau01]. Comme nous l’avons vu au chapitre précédent, cette arithmétique est pessimiste, dans le sens où les bornes peuvent être surestimées ; elle est cependant garantie, ce qui nous permet de certifier nos résultats non seulement par rapport aux incertitudes mais aussi par rapport aux erreurs d’arrondis qui apparaissent dans le calcul informatique.

4.2 Analyse par intervalles de performance des robots

La conception intervient avant la fabrication du robot en considérant un modèle théorique simplifié. Pour des raisons de faisabilité, le nombre de paramètres de conception ne peut être trop élevé pour ne pas exploser les temps de calculs lors de la conception. Par exemple, un modèle géométrique complet de la plate-forme de Gough compte 138 paramètres [Vis96] pour moins d’une dizaine de paramètres de conception généralement considérés. Plus encore, le comportement du robot doit être considéré dans un ensemble d’états possibles (ensemble de configurations du robot) et pour des conditions matérielles et environnementales variables (précision des actionneurs, forces appliquées, etc).

Les méthodes que nous proposons, considèrent des modèles de taille suffisante pour être significative en terme d’analyse de comportement mais prennent aussi en compte les incertitudes ou variations des variables et des paramètres du problème. Pour cela nous utilisons le principe de modélisation épaissie introduite en section 3.2.1 par une approche intervalle. Ainsi nous considérons un ensemble borné de modèles possibles qui contient (ou contiendra après fabrication) le modèle effectif réel soumis à des conditions variables.

4.2.1 Représentation par intervalles des propriétés des robots et conception approprié

Nous utilisons les notations introduites en section 2.2. Nous notons :

- ξ les paramètres géométriques du robot qui le définissent
- E_t l’ensemble des variables qui sont liées à l’état du robot : par exemple sa position/orientation X ou les coordonnées articulaires Q .
- E_v l’ensemble des variables décrivant les conditions auxquelles peut être soumis le robot (exemple : F les forces extérieures et τ intérieures) ou ses caractéristiques matériels (exemple : précision des capteurs proprioceptifs)
- pour plus de simplicité nous notons le vecteur $E = [E_t, E_v]$

De façon générique, les propriétés souhaitées du manipulateur sont données sous la forme de contraintes d’égalité ou d’inégalité notées $C(E, \xi)$ avec $C(E, \xi) = 0$; $\neq 0$ ou $\underline{c}_r \leq C(E, \xi) \leq \overline{c}_r$ décrivant un ensemble compact de points ; les valeurs de $\underline{c}_r, \overline{c}_r$ étant donnés par un cahier des charges par exemple. Ces contraintes seront principalement obtenues à partir du modèle implicite $\mathcal{F}, (X, Q, \xi)$ et de ses dérivées (jacobiennes) en fonction des variables $X, Q, \xi, F, \tau, \delta X \dots$ qui sont décrites dans la section 2.2.

L’ensemble de ces contraintes peut être *étendu par intervalle*, c’est-à-dire que nous pouvons borner l’évaluation de ces équations/inéquations pour un ensemble de points représentés par des intervalles, tel que nous l’avons vu en section 3.1.

L'extension intervalle des contraintes $C(E, \xi)$ est $\mathbf{C}(\mathbf{E}, \xi) \supseteq \{C(E, \xi), \forall E \in \mathbf{E}\}$; elle nous permet d'évaluer cette contrainte sur l'ensemble des points de E , $E \in \mathbf{E} = [\underline{E}, \overline{E}]$. Notons qu'en raison des propriétés de l'arithmétique d'intervalle nous ne pouvons garantir qu'une sur-estimation de $\{C(E, \xi), \forall E \in \mathbf{E}\}$. Le premier challenge est alors d'obtenir la meilleure extension intervalle de C dans le sens où les bornes de $\mathbf{C}(\mathbf{E}, \xi)$ sont le plus proches possible des bornes de $\{C(E, \xi), \forall E \in \mathbf{E}\}$. Pour cela nous utilisons les techniques introduites en section 3.1.3. Cependant, il n'est pas toujours possible d'obtenir une évaluation direct de ces contraintes comme nous le verrons en section 4.3. En effet, elles peuvent être le fruit de la résolution d'un système (le plus souvent linéaire dans les cas que nous avons traités), d'un déterminant etc. Nos contributions en matière d'analyse par intervalles ont été influencées par cette difficulté. Il est aussi à noter que la quantification des variables E peut jouer un rôle important pour décrire les ensembles recherchés, voir 3.2.3.

4.2.2 Caractérisation de sous-espace de travail en fonction des propriétés de robots

Il nous faut maintenant décrire et évaluer chacune des propriétés recherchées pour une instance du robot (les valeurs de ξ sont fixées et scalaires). Pour fusionner ces propriétés nous passerons par le calcul *des* espaces de travail (pour chacune des propriétés) et ainsi certifier que le robot, positionné dans un ensemble continu de points de l'espace de travail, possède les propriétés décrites dans le cahier des charges. Cet espace de travail est donné par des contraintes sur les coordonnées généralisées X (notation voir section 2.2) ce qui est plus adapté pour les robots parallèles. Le cœur de l'algorithme est alors décrit.

Tout d'abord nous devons modéliser les propriétés souhaitées par un système d'équations, d'inéquations ou d'inégalités présenté de façon générale dans la section précédente 4.2.1. Ainsi, l'espace de travail du robot est donné par l'ensemble de valeurs de $X \in \mathbf{X} = [\underline{X}, \overline{X}] : \forall x \in \mathbf{X}, \forall, \exists E_v \in \mathbf{E}_v, C(X, \xi, E_v) = 0, \neq 0, \leq 0$.

Il est à noter que le(s) quantificateur(s) associé(s) à E_v peut(peuvent) décrire différentes propriétés suivant les cas. Ces ensembles sont généralement difficile à obtenir. Une approche est d'en déterminer les frontières, voir par exemple notre article [GMD06b]. Ceci peut être difficile à obtenir pour des dimensions de paramètres E_v ou X élevés.

Une alternative intervalle permet de les décrire. Cependant, la représentation par intervalles nous oblige à adopter une description de ces ensembles par des boites (rectangle en 2D, cube en 3D et hypercube dans les dimensions supérieures dont les cotés sont parallèles aux axes des repères de référence choisis. Nous n'obtiendrons donc qu'une approximation de ces ensembles par un pavage. Ainsi, afin d'obtenir une description précise, nous utilisons l'approche décrite à la section 3.3.4, centrée sur l'algorithme de *branch and bound*.

Ainsi, nous déterminons si un ensemble de positions modélisées par un vecteur intervalle boite, \mathbf{X}_t est à l'intérieur, à l'extérieur ou à la frontière de l'espace décrit par les contraintes C en évaluant $C(\mathbf{X}_t, \xi, \mathbf{E}_v)$.

Comme nous l'avons vu les difficultés sont généralement de déterminer le test de boite intérieur, ce qui nécessite de jouer sur les quantificateurs ($\forall, \exists, \#$) associés aux paramètres de C . L'algorithme est initialisé avec une boite englobante de l'espace de travail qu'il est usuellement facile d'obtenir en utilisant

les contraintes géométriques liées aux débattements du robot. L'arrêt de l'algorithme est liée au nombre nul de boîtes à traiter et donc à la taille minimum pour laquelle une boîte est considérée comme frontière (on ne peut pas décider si elle est interne ou externe à l'espace recherché). La variable bissectée étant un fragment de l'espace de travail du robot il est toujours possible de déterminer une valeur limite associée à une incertitude de positionnement physiquement acceptable. Les contributions pour chacun des problèmes traités sont aussi liées au type de contracteur utilisé ou adapté au problème. En effet, les algorithmes de filtrage comme l'algorithme de Newton par intervalle ou les algorithmes provenant de la programmation par contrainte (2B, 3B, box consistency, HC4, 3Bcid ...) [CJ09] ou ceux utilisant des test d'unicité (Newton, Kantorovitch) [Mer04] jouent un rôle très important. Ces algorithmes permettent de réduire l'espace de recherche sans introduire de point de choix, ce qui permet de limiter la combinatoire. Ainsi, en fonction des contraintes considérées, à partir d'un domaine de recherche initial (boîte de départ), le filtre retourne une boîte de largeur plus petite sans perte de solution. Certains algorithmes certifient l'existence et/ou l'unicité de la solution.

Ainsi, nous notons $\mathcal{D} = \mathcal{D}_1 \cap \dots \cap \mathcal{D}_k$ l'espace de travail d'un robot l'intersection de k sous-espace de travail caractérisés par k différentes propriétés donnée par des contraintes $C_k(\mathbf{X}_t, \xi, \mathbf{E}_v)$.

Nous caractérisons deux cas :

- **Vérification d'un espace de travail \mathcal{B}** : ceci consiste à vérifier qu'une boîte ou un ensemble de boîtes \mathcal{B} est entièrement incluse dans l'espace de travail du robot, $\mathcal{B} \subseteq \mathcal{D}$, plus précisément que \mathcal{B} n'est constitué que de boîtes intérieures.
- **Approximation par intervalle de l'espace de travail \mathcal{D}** : ceci consiste à subdiviser une boîte englobant l'espace de travail, notée $\mathcal{D}_e \supset \mathcal{D}$ en trois listes de sous-boîtes : les boîtes intérieures $\mathcal{B}_{int} \subset \mathcal{D}$, les boîtes extérieures $\mathcal{B}_{ext} \not\subset \mathcal{D}$ et \mathcal{B}_{bound} si les boîtes ne peuvent être rangées dans les deux catégories précédentes.

4.3 Propriétés étudiées

Nous donnons ici quelques exemples de travaux, auxquels j'ai participé ou co-dirigé, utilisant l'analyse par intervalles pour certifier un espace de travail d'un robot en fonction d'une ou plusieurs propriétés considérées.

De nombreux travaux se sont intéressés (et s'intéressent encore) à ces calculs d'espace de travail [Mer06], et ceci pour tout les types de manipulateurs : séries ou parallèles. Les deux méthodes classiques utilisent soit les approches géométriques pour déterminer ses frontières, par exemple nos travaux sur l'espace de travail des robots à câbles [GMD06a], soit les approches vérifiant chaque point sur une grille de segmentation [CW98].

L'approche géométrique garantit une description exacte de l'espace de travail, mais elle est complexe à mettre en œuvre pour des problèmes de grandes dimensions. Nous avons toutefois utilisé cette approche reposant sur la détermination de la frontière de l'espace recherché [GMD06b]. Pour plus de cohérence dans ce document, principalement centré sur l'approche intervalle développé pour les principes de la conception appropriée, nous nous étendrons peu sur ce type de méthode. Cependant, nous pouvons la résumer. Les contraintes décri-

vant les positions pour lesquelles le robot possède les propriétés requises, sont données par les inégalités $c_r \leq C(X, \xi) \leq \bar{c}_r$. Les lieux frontières étant ceux pour lesquels au moins une des contraintes atteint ses bornes soit : $C(X, \xi) = \bar{c}_r$ ou $C(X, \xi) = c_r$. Les contraintes étant vectorielles, nous pouvons constituer des sous-ensembles bien-contraint de systèmes de ce type et identifier ces points extrêmes. Le but est alors de relier ces points par des courbes (dans le cas d'une représentation par coupe 2D) ou surfaces qui modélisent alors les frontières. On utilisera pour cela des algorithmes de suivi de courbes, qui garantissent l'existence et l'unicité d'un point sur cette frontière. L'analyse par intervalles s'avère, là encore, un outils intéressant que nous avons utilisé.

Une seconde approche par segmentation est plus simple à implanter car il suffit de savoir vérifier les propriétés du robot en un point : la procédure sera alors répétée en chacun des points d'un cube (hypercube) englobant une approximation de l'espace de travail. Imaginons une segmentation de 100 points par axes pour un temps de calcul des propriétés de 0.01 ms : il faut 10 s pour vérifier un espace de travail à 3 ddl et approximativement 115 jours pour 6 ddl. Ainsi, la distance entre les points de la grille de segmentation doit être assez grande pour permettre un calcul dans un temps réaliste ; calcul qui ne donnera toutefois aucune garantie pour les points hors de la grille quel que soit le mode d'échantillonnage aléatoire ou régulier. Nous préférons le gruyère à l'emmental...

L'arithmétique par intervalle nous permet donc de vérifier les contraintes sur des ensembles compacts, ce qui justifie son utilisation pour la vérification et la certification (si l'arithmétique est correctement arrondie) de l'espace de travail des robots. Nous utilisons, dans leur grandes lignes, les principes présentés précédemment au chapitre précédent afin de garantir les espaces de travail des robots en utilisant les techniques de l'analyse par intervalle.

4.3.1 Prise en compte de contraintes géométriques

Dans cette section nous présentons nos contributions sur l'analyse des performances des robots et le calcul de l'espaces de travail définis par des contraintes dites géométriques. Tout d'abord, elles sont données par les débattements articulaires, puis par le lieu singularités de l'espace de travail et enfin pour les possibles collisions entre les éléments du manipulateur.

Débattements articulaires

Les premières propriétés généralement considérées pour un robot sont les limitations de son espace de travail contraint par les butés articulaires. Un point d'accès simple et pédagogique de nos travaux est l'article [ODSM08a] qui présente l'analyse de l'espace de travail d'un micro-manipulateur parallèle 3RRR plan à joints flexibles voir 4.1. Ce papier décrit l'algorithme de base de l'analyse de l'espace de travail mais aussi les difficultés, quelques astuces et les outils du projet Coprin/Hephaistos pour améliorer l'efficacité de la méthode.

Ainsi, notons $\mathbf{C}_q(\mathbf{X}_s, \xi)$ les équations qui décrivent les débattements articulaires et Q_{min}, Q_{max} leurs bornes. Ces équations sont obtenues en utilisant le modèle géométrique inverse (MGI), noté $\mathcal{F}_{MGI}(X, \xi) = Q = [q_1, q_2, \dots]^T$ des robots parallèles. Ce modèle est généralement simple : les contraintes \mathbf{C}_q peuvent alors être données directement par ces équations. Dans le cas de la plate-forme de Gough ou autre mécanisme qui admettent une seule solution au MGI, une simple

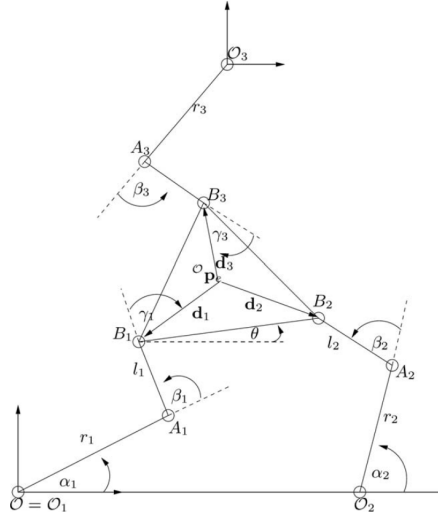


FIGURE 4.1 – Micro-manipulateur parallèle 3RPR plan à joint flexible

évaluation suffit pour décrire ces contraintes et les tests de boîtes intérieures ou extérieures, voir 3.1.5. Ainsi, l'évaluation de $\mathbf{F}_{MGI}(\mathbf{X}_s, \xi) = \mathbf{Q}_s$, une extension par intervalle de $\mathcal{F}_{MGI}(X_s, \xi)$, sur l'intervalle \mathbf{X}_s donne une sur-estimation de l'ensemble des valeurs des coordonnées articulaires \mathbf{Q}_s pour l'ensemble continu des points de l'espace $X_s \in \mathbf{X}_s$. Ceci nous permet d'obtenir des tests simples pour déterminer la nature des boîtes :

- *Test boîte intérieur* : $\mathbf{F}_{MGI}(\mathbf{X}_s, \xi) \subseteq [Q_{min}, Q_{max}]$
- *Test boîte extérieur* : $\mathbf{F}_{MGI}(\mathbf{X}_s, \xi) \cap [Q_{min}, Q_{max}] = \emptyset$

L'une des particularités de l'analyse par intervalle est qu'il faut remettre en question nos habitudes notamment en terme d'évaluation de fonction – voir 3.1.3. Ainsi, nous remarquons dans [ODSM08a] que le MGI du 3RPR plan admet jusqu'à 2 solutions (les équations du modèles implicite étant quadratique). Il est alors facile d'obtenir une forme analytique de ces solutions. Mais l'article montre que l'évaluation directe de ne permet pas d'obtenir des résultats satisfaisant notamment pour le pavage des boîtes intérieurs : les bornes $\mathbf{F}_{MGI}(\mathbf{X}_s, \xi)$ sont surestimés lors de l'évaluation du MGI. Nous avons donc proposé un certain nombre d'astuces que nous utilisons afin d'améliorer nos résultats :

- simplification de la structure des équations en introduisant des variables et contraintes supplémentaires notamment pour éliminer des fonctions trigonométrique inverse.
- utilisation de filtres, voir section 3.1.4
- ajout de contraintes redondantes pour améliorer l'efficacité des filtres

Un autre avantage de l'analyse par intervalle est la prise en compte des incertitudes dans nos modèles. Ainsi dans [ODSM08a], le micro-manipulateur 3RPR plan est constitué d'articulations passives flexibles. Afin de certifier les performances, il est intéressant de prendre en compte des sources possibles d'erreurs telles que les tolérances de fabrication et les inexactitudes de modélisation dans

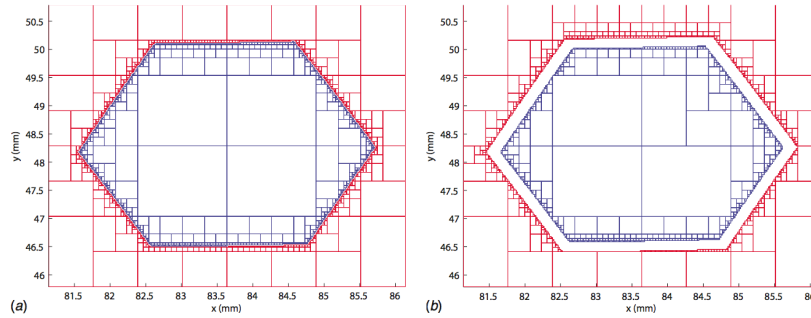


FIGURE 4.2 – Coupe (x,y) , orientation fixée à -10.3° , d’un espace de travail certifié d’un 3RRR plan prenant en compte les débattements articulaires ($\pm 3^\circ$) ; en bleu l’espace de travail admissible et en rouge l’espace de travail ne satisfaisant pas les limites du débattement : en (a) les longueurs des barres sont considérées exactes, en (b) en prenant en compte des tolérances de fabrication sur les longueurs de barres de $\pm 50\mu m$

ces articulations.

La figure 4.2(a) montre un espace de travail contraint par des débattements articulaires, produit en ne supposant aucune incertitude dans les tolérances de fabrication et la modélisation des articulations pivots flexibles. Dans ce cas, les longueurs des barres r_i et l_i (voir Figure 4.1), qui font partie des paramètres géométriques ξ de ce robot, ont été définies comme des intervalles dégénérés. Les boîtes intérieures, soit l’espace de travail certifié, sont représentées en bleu et les boîtes extérieures en rouge.

Maintenant, nous souhaitons certifier les performances de ce robot en prenant en compte les incertitudes de modélisation des articulations flexibles. Il est montré dans la littérature que modéliser ces défauts est complexe mais qu’ils se traduisent par des translations supplémentaires de l’articulation qui peuvent être reportées dans les valeurs des longueurs de barres l_i et r_i . Ainsi on ajoute à ces variables une incertitude qui borne les erreurs de modélisation : on épaissira alors leur valeurs. Cette démarche est décrite en section 3.2.1 mais nous reverrons cette approche dans le cadre de l’étalonnage qui consiste à considérer un *modèle épais*. Des incertitudes supplémentaires, dues aux erreurs de fabrication de ces barres, sont également ajoutées à ces variables.

Dans notre exemple, l’ensemble des incertitudes dues à des défauts non modélisés et aux tolérances est défini comme étant limité au sein d’un intervalle $\pm 50\mu m$. L’espace de travail résultant, dans les limites de déviation admissible des joints flexibles est représenté sur la figure 4.2(b) pour comparaison (coupe en translation pour une orientation fixée à -10.3°). Comme prévu, la surface des boîtes internes, en bleu, est plus petite si les erreurs sur les paramètres sont prises en compte. Nous constatons aussi que la frontière de l’espace de travail est plus grande (espace entre les boîtes bleues et rouges). Cette zone représente des points pour lesquels il n’est pas possible de garantir leur accessibilité pour les tolérances considérées. Mais l’objectif, à savoir certifier un espace de travail interne, est atteint.

L’épaississement de modèles pour prendre en compte les incertitudes sur

les paramètres géométriques est très intéressant en pratique, ce qui donne à nos résultats un intérêt supplémentaire. La cerise sur le gâteau est que cette plus-value est obtenue sans changer l’algorithme.

Prise en compte des singularités

À cet espace de travail contraint par les débattement articulaires, nous pouvons ajouter des contraintes afin de prendre en compte les singularités du robot. Typiquement, ces contraintes sont définies par l’annulation des déterminants des matrices jacobiniennes $\det(J_X = \frac{\partial F}{\partial X}) = 0$ (singularité de type II), et $\det(J_Q = \frac{\partial F}{\partial Q}) = 0$ (singularité de type I).

Pour les cas que nous traitons en robotique, en utilisant un outil de calcul formel, il est possible d’obtenir symboliquement ces déterminants et, éventuellement, les factoriser. Cependant, pour le cas décrit dans [ODSM08a] où la matrice jacobienne est de taille relativement faible 3×3 , il est montré que cette approche d’évaluation directe du déterminant n’est pas efficace en raison d’une trop grande surestimation de ses bornes par l’arithmétique par intervalles.

Ce problème est encore plus grand lorsqu’on cherche à déterminer la présence de singularités pour des robots évoluant dans un espace de dimension 6. Dans [MD01], nous cherchons à vérifier cela pour une plate-forme de Gough. Le terme *vérifier* est un point essentiel des approches que nous avons mises en place. En effet, la localisation exacte (si ceci est possible concrètement) d’une singularité n’est généralement pas ce qui est recherché : nous souhaitons, avant tout, certifier l’absence de ces points. Prenons le cas d’un point X_1 d’une boîte (partie compacte de l’espace de travail) à vérifier. Si dans cette boîte il existe un point X_2 tel que $\text{sign}(\det(J(X_2))) \neq \text{sign}(\det(J(X_1)))$ alors il existe une singularité au sein de l’espace considéré. L’analyse par intervalle nous permet de vérifier cela – voir [MD01]. Le problème est là encore de fournir des bornes sur l’évaluation de ce déterminant le plus efficacement possible, c’est à dire en minimisant leur sur-estimation. Ce papier souligne les solutions pour améliorer le problème de l’évaluation de ces fonctions en utilisant :

- **des outils du calcul formel-numérique.** En exploitant la forme semi-symbolique² du déterminant il est possible d’optimiser son traitement : il apparaît comme une somme de termes génériques en les coordonnées généralisées. La vérification de la monotonie sert la encore pour une meilleur évaluation par intervalle. Enfin, ces formes sont exploités pour mettre en place des filtres adaptés, voir section 3.1.4.
- **des outils du calcul numérique.** Le problème peut être aussi donné sous une forme d’un problème d’optimisation globale garantie sur un domaine de recherche. La librairie ALIAS fournit de tels outils.

Cependant, il ne faut pas voir ce problème uniquement comme un problème d’évaluation directe par intervalle en utilisant une forme développée du déterminant. L’analyse numérique adaptée aux intervalles nous donne de nombreux outils pour calculer ou borner les valeurs du déterminant sans nécessairement le développer : en passant, par exemple par une décomposition LU, élimination de Gauss, inégalité de Hadamar ... La librairie ALIAS de Hephaistos³ fournit

². Une partie des paramètres (les paramètres géométriques) est ici donnée sous forme de rationnel afin de certifier les calculs

³. Voir <http://www-sop.inria.fr/hephaistos/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html>

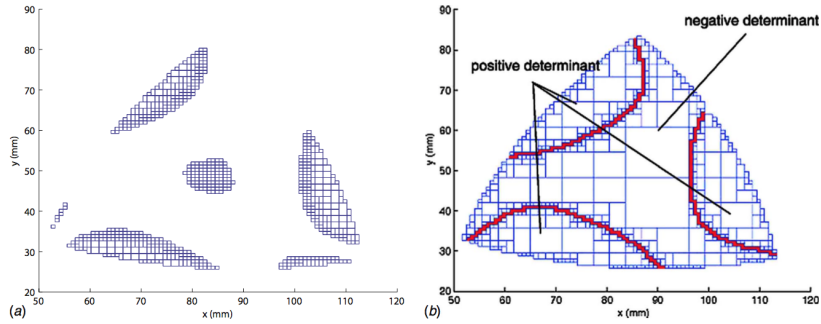


FIGURE 4.3 – Coupe (x,y) d’un espace de travail certifié d’un 3RRR plan sans singularité (en bleu) : en (a) les courbes de singularité sont supprimés en utilisant une évaluation directe du déterminant, en (b) en utilisant des tests de régularité de la matrice

ces fonctions. Plus encore, il est à souligner les implantations des tests qui permettent de vérifier la régularité des matrices. Nous avons utilisé dans [ODSM08a] deux tests, spécifiques à l’analyse par intervalle et disponibles dans ALIAS. Ils se sont avérés les plus efficaces pour détecter les singularités des robots.

Le premier test, proposé par Rex et Rohn [RR98] est fondé sur des conditions suffisantes qui montrent la régularité d’une matrice intervalle proposée. Ce test de cohérence indique comment, à partir d’une matrice intervalle \mathbf{A} de taille $n \times n$, bâtir 2^{2n-1} matrice scalaire telles que leurs déterminants est de même signe alors il n’existe pas de matrices singulières dans \mathbf{A} . C’est un test puissant pour certifier une boîte extérieure pour la contrainte étendue aux intervalles $\det(\mathbf{J}_X(\mathbf{X})) = 0$, à savoir qu’une boîte de l’espace de travail \mathbf{X} ne contient aucune singularité.

Cependant, les éléments des jacobiennes sont paramétrés en les variables X, Q, ξ et donc symboliquement liés ; hors, l’utilisation de l’arithmétique par intervalles perd cette corrélation. Merlet et Donovan proposent dans [MD06a] de prendre en compte les liens linéaires entre les éléments de la matrice en fonction des paramètres. Ceci permet d’améliorer grandement les performances de l’algorithme précédent : ainsi, dans [ODSM08a], nous comparons en figure 4.3 deux coupes de l’espace de travail en (x, y) du 3RRR plan pour une orientation de 40° obtenue par l’algorithme présenté précédemment. En Figure 4.3(a), une évaluation directe du déterminant nous permet de rejeter des boites dans l’espace travail mais susceptibles de contenir une singularité. En figure 4.3(b), la même chose est obtenue mais en utilisant les algorithmes disponible dans ALIAS utilisant les tests de régularité : les résultats sont bien meilleurs, les courbes de singularités à supprimer de l’espace de travail sont alors soulignées en rouge pour une meilleur clarté.

Prise en compte des collisions

Nous avons étudié un autre type de contrainte géométrique importante car elle limite l’espace de travail du manipulateur : les collisions des éléments internes du robot. L’analyse par intervalle est la encore bien adaptée à ce pro-

blème. En effet, nous souhaitons garantir que, pour une boîte (ensemble continu de position/orientation de l'organe terminal) aucun de ces éléments ne soit en contact.

Dans [MD06b], nous avons mis en oeuvre un algorithme de détection de ces collisions pour une architecture de type plate-forme de Gough. La difficulté est de considérer toutes des intersections possibles de l'ensemble des éléments, de formes hétérogènes, qui constituent un manipulateur. Il est alors nécessaire de faire des hypothèses simplificatrices tout en garantissant les non collisions. Nous avons choisi d'englober chaque élément du robot classiquement considérés – jambes et plateaux – par un cylindre droit. De plus, afin de répondre à un besoin industriel, nous avons ajouté sur les plateaux des cylindres supplémentaires englobant des éléments volumineux (outils, objets à déplacer ...).

Afin de vérifier les possibles collisions, nous proposons différentes approches fondées sur les distances entre les axes des cylindres et les intersections possibles entre les sections droites des cylindres. Ces approches fournissent un ensemble de contraintes d'inégalités qui doivent être vérifiées mais dont la nature, le nombre (à maximiser) et les variables à considérer (à minimiser) diffèrent. Là encore une approche globale s'avère efficace : on préférera considérer un ensemble sur-contraint et redondant d'inégalités mixant les différentes approches. Par exemple, la première approche considérant la distance entre les axes des cylindres devient numériquement instable lorsque ces axes deviennent parallèles. Les autres approches deviennent alors plus efficaces en terme intervalle : les surestimations des bornes des inégalités à évaluer est moindre.

Comme pour les précédentes études, un soin particulier est porté sur l'évaluation des contraintes en utilisant leur monotonie, et l'utilisation de filtre s'avère essentiel.

Les tests ont été effectués pour un robot à câbles considérant 9 cylindres (6 pour les câbles, 2 associés au mobile et 1 à la base) générant 33 intersections possibles à vérifier deux à deux.

Plusieurs espaces de travail 6D, boîtes en position et orientation et trajectoires sont alors vérifiés en des temps tout à fait acceptables (max de 1 heure pour un espace de travail complet et de l'ordre de quelques dizaines de secondes pour les trajectoires).

4.3.2 Prise en compte de contraintes variationnelles et statiques

Pour le moment, nous avons vu que les contraintes nous permettant de définir un espace de travail étaient directement données sous une forme directe d'un système d'équations par intervalles (qui peuvent être vue comme des inégalités). Les difficultés sont alors liées à l'évaluation de ces contraintes. Cependant, d'autres types de contraintes modélisant des propriétés attendues d'un manipulateur sont intéressantes.

Nous allons voir comment certifier la précision et les performances statiques d'un manipulateur. La difficulté est ici que ces propriétés sont données sous la forme d'un problème inverse, et dans les cas présentés ici, au travers de la solution (plus précisément un continuum de solutions) d'un système linéaire d'équations par intervalles.

Vérification de la précision du manipulateur

La précision des manipulateurs est une propriété importante des manipulateurs au regard des applications qui leur sont dédiées. Nous avons considéré ce problème dans plusieurs articles [MD05, ODSM08b, ODSM08a] dont nous présentons les grandes lignes dans cette section.

De façon générique, la différentiation du modèle implicite $\mathcal{F}, (X, Q, \xi)$, voir section 2.2, des manipulateurs nous donne le modèle variationnel :

$$J_X(X, Q, \xi) \cdot \Delta X = -J_Q(X, Q, \xi) \cdot \Delta Q \quad (4.1)$$

Où $J_X = \frac{\partial F}{\partial X}$ et $J_Q = \frac{\partial F}{\partial Q}$.

Les jacobienes $J_X(X, \xi)$ et $J_Q(X, \xi)$ ⁴ étant dépendantes du positionnement X du manipulateur, sa précision n'est alors généralement⁵ pas la même dans tout son espace de travail.

Dans cette équation 4.1, le vecteur ΔQ est une variation des coordonnées articulaires autour d'une configuration du robot dont les valeurs articulaires sont notées \tilde{Q} . Cette variation peut être une incertitude liée à la mesure de \tilde{Q} donnée par les capteurs proprioceptifs du robot : c'est donc une donnée du problème fournie par un cahier des charges du constructeur du robot et fonction de la résolution $\pm\alpha$ des capteurs de mesure installés sur les axes des articulations actives. Associer un modèle de bruit à cette variable peut s'avérer difficile : on peut cependant la modéliser au pire cas à travers un intervalle qui représente l'ensemble des valeurs bornées possibles autour de \tilde{Q} , noté $\tilde{\mathbf{Q}} = [\mathbf{Q} - \alpha, \mathbf{Q} + \alpha] = \mathbf{Q} + \widetilde{\Delta\mathbf{Q}}$ avec $\widetilde{\Delta\mathbf{Q}} = [-\alpha, +\alpha]$.

Le vecteur ΔX représente l'incertitude de positionnement du robot dû à la propagation de $\widetilde{\Delta Q}$ dans le système modélisé par l'équation 4.1. Une approche pourrait alors consister à déterminer, pour chaque point de l'espace de travail, la valeur de ΔX en fonction de $\widetilde{\Delta Q}$. Nous proposons une alternative qui s'inscrit dans la méthodologie de la conception appropriée fondée sur la certification *des* espaces de travail du robot.

L'objectif est alors de déterminer l'ensemble \mathcal{B} des positions/orientations de l'organe terminal du robot X tel que sa précision ΔX soit inférieure à une incertitude acceptable $\widetilde{\Delta X}$ donnée, là encore, par un cahier des charges. Dans le formalisme intervalle, il nous faut donc $\Delta X \subset \widetilde{\Delta X}$. La description du problème est alors donnée par :

$$\forall X \in \mathcal{B}, \forall \Delta Q \in \Delta \mathbf{Q}, \Delta X \subset \widetilde{\Delta X} \quad (4.2)$$

Afin de généraliser le problème, nous considérons chacune des composantes $\widetilde{\Delta Q}_i$ du vecteur $\widetilde{\Delta Q}$ comme une incertitude distribuée sur l'intervalle $[-\alpha_{q,i}, +\alpha_{q,i}]$. Dans le cadre d'une représentation par intervalle nous avons :

4. Pour les robots étudiés les jacobienes sont exprimées en fonction des seules coordonnées généralisés.

5. Ceci pour les robots non isotropes.

$$\Delta \mathbf{Q} = D_{\alpha_q} \cdot \mathbf{O}_I \text{ avec } D_{\alpha_q} = \begin{pmatrix} \alpha_{q,1} & 0 & \dots \\ 0 & \alpha_{q,2} & \dots \\ \dots & \dots & \dots \end{pmatrix} \quad (4.3)$$

$$\text{et le vecteur de variation d'unité } \mathbf{O}_I = \begin{pmatrix} [-1, 1] \\ [-1, 1] \\ \dots \end{pmatrix}$$

Dans le cas bien contraint ($\dim(\Delta X) = \dim(\Delta Q) = n$), il peut être possible d'obtenir l'inverse de J_X^{-1} (si $Det(J_X) \neq 0$). L'équation 4.1 peut alors s'écrire :

$$\Delta X = \begin{pmatrix} \Delta X_1 \\ \Delta X_2 \\ \dots \end{pmatrix} = W(X, \xi) \cdot \mathbf{O}_I \quad (4.4)$$

$$\text{avec } W = \begin{pmatrix} W_1(X, \xi) \\ W_2(X, \xi) \\ \dots \end{pmatrix}_{n \times n} = -J_X(X, \xi)^{-1} \cdot J_Q(X, \xi) \cdot D_{\alpha_q} \quad (4.5)$$

L'erreur maximale pour chaque composante ΔX_i , pour une erreur unitaire \mathbf{O}_I , est facilement obtenue en calculant la norme l_1 de chaque ligne de la matrice $W_i(X, \xi) : \Delta X_i^{max}(X, \xi) = \|W_i(X, \xi)\|_1 = \sum_{j=1}^n |W_{i,j}|$. Nous pouvons comparer alors ΔX_i^{max} avec la précision désirée $\widetilde{\Delta X}_i$: si $\Delta X_i^{max}(X, \xi) < \widetilde{\Delta X}_i$ alors l'erreur de positionnement du robot est jugée satisfaisante au point X .

Ainsi l'espace de travail en précision du robot peut être défini comme suit :

$$\mathcal{B} = \{X \mid \|W_i(X, \xi)\|_1 < \widetilde{\Delta X}_i \forall i = 1, \dots, n\} \quad (4.6)$$

Un algorithme de type *branch and bound* permet d'obtenir un pavage certifié de cet ensemble en vérifiant la condition étendue aux intervalles $\forall X \in \mathbf{X}, \|W_i(\mathbf{X}, \xi)\|_1 \subset \widetilde{\Delta X}_i$.

Cependant, le calcul de la matrice $W(X, \xi)$ nécessite d'inverser la matrice $J_X(X, \xi)$. En fonction de sa taille, de la complexité des coefficients en les paramètres, il n'est pas toujours évident d'obtenir une inverse dont la forme symbolique est compatible avec une bonne évaluation par intervalles des critères présentés précédemment.

Nous avons donc étudié des alternatives qui consistent, tout d'abord, à résoudre le système linéaire 4.1 étendu aux intervalles. En effet, une approximation extérieure de la précision (c'est à dire contenant la précision effective du robot) est donnée par :

$$\Delta \mathbf{X}^\square = \Sigma_{\exists, \exists}(\mathbf{J}_X(\mathbf{X}, \xi), -\mathbf{J}_Q(\mathbf{X}, \xi) \cdot \widetilde{\Delta \mathbf{Q}}) \quad (4.7)$$

Ainsi, l'espace de travail en précision \mathcal{B} peut-être obtenu en calculant par un algorithme de type "Branch and Bound" sur \mathbf{X} .

$$\mathcal{B} = \{X \mid \Delta \mathbf{X}^\square \subset \widetilde{\Delta \mathbf{X}}\} \quad (4.8)$$

Les solutions de l'équation intervalle selon 4.3.2 sont sur-estimées : la condition de la précédente définition est alors faible mais suffisante pour décrire une

approximation de l'ensemble. Afin de l'améliorer, le cœur de l'algorithme est alors d'obtenir la meilleure estimation de $\Delta \mathbf{X}^\square$ fournie par la résolution du système linéaire 4.7.

Nous avons proposé dans [MD05, ODSM08b, ODSM08a] plusieurs solutions qui sont en fait complémentaires :

- une adaptation des algorithmes itératifs de type Gauss-Seidel et Krawczyk : l'algorithme cherchant juste à vérifier l'inclusion de la précision dans un intervalle donné
- le système linéaire étant dépendant de la position du robot X , nous avons amélioré l'algorithme d'élimination de Gauss afin de prendre en compte les inter-dépendances entre les paramètres par l'utilisation de leurs dérivées dans l'évaluation du schéma de Gauss
- un pré-conditionnement symbolique de l'algorithme d'élimination de Gauss par intervalle.
- une utilisation du théorème de Oettli-Pragger (voir équation 3.4) pour déterminer s'il y a un système qui admet une solution en les bornes de $\Delta \mathbf{X}$, ce qui permet de rejeter des boîtes.

Les tests et algorithmes présentés dans nos papiers sont adaptés au cas bien-contraint et l'approche est généralisable à d'autres propriétés des robots comme la vérification des performances statiques du robot.

4.3.3 Vérification des performances statiques

D'une façon très similaire à la section précédente, l'espace de travail statique du robot peut être déterminé. Nous vous conseillons de lire la thèse de Julien Hubert [Hub10] que j'ai pu co-encadrer sur l'analyse statique par intervalles des robots parallèles et leur liens avec les singularités : cette approche par intervalles offre des perspectives intéressantes.

Nous avons montré dans [GMD06b, GDM11] qu'il est possible de calculer un espace de travail qui garantit la capacité de l'organe terminal du robot à produire un ensemble de forces/moments, en fonction de forces/couples générés par les actionneurs installés sur les articulations actives du robot.

Pour cela nous utilisons le modèle statique des robots parallèles présenté en section 2.2 en posant $W = J_{inv}^T$:

$$W(X).\tau = F \tag{4.9}$$

avec F les forces/moments appliquées sur la plate-forme mobile et τ les forces/couple articulaires. Notez que la matrice W est paramétrée par la position/orientation du robot X .

Le but est de déterminer l'ensemble des positions/orientations du robot tel qu'un ensemble de force noté $\mathbf{F}_d = [\underline{\mathbf{F}}, \overline{\mathbf{F}}]$ puisse être généré par un vecteur des forces/couples τ donné par un intervalle noté $\tau_d = [\underline{\tau}_d, \overline{\tau}_d]$. Les vecteurs intervalles \mathbf{F}_d, τ_d sont des données du problème : respectivement les capacités de charges de robot souhaitées et de ses capacités de force/couple fournies par les moteurs. Ainsi, la description de l'espace de travail statique est donnée par \mathcal{B} :

$$\mathcal{B} = \{X | \forall F \in \mathbf{F}_d, \exists \tau \in \tau_d; W(X).\tau = F\} \tag{4.10}$$

Là encore, cela revient à déterminer des tests de boîtes intérieures et extérieures à cet ensemble afin de mettre en œuvre un algorithme de *Branch and Bound*. Dans le cas des robots bien-contraints (la dimension de l'espace généralisé est la même que celle de l'espace articulaire, $\dim F = \dim \tau$), l'approche est similaire à la détermination de l'espace de travail en précision, vue précédemment. La résolution du système linéaire 4.9, étendu aux intervalles tel que :

$$\tau^\square = \Sigma_{\exists, \exists}(\mathbf{W}(\mathbf{X}_t), \tau_{\mathbf{d}}) \quad (4.11)$$

fournit une condition suffisante pour montrer que \mathbf{X}_t est une boîte intérieure de l'ensemble \mathcal{B} si $\tau^\square \subseteq \tau_{\mathbf{d}}$. Nous pouvons utiliser le même type d'astuces présentées dans la section précédente afin d'améliorer les bornes de 4.11 et ainsi rendre le test plus efficace. Pour le test de boîtes extérieures, il suffit de montrer qu'au moins une des forces/moments $\mathbf{F}_{\mathbf{d}}$ ne peut être générée par un $\tau \in \tau_{\mathbf{d}}$, notamment pour des valeurs limites scalaires de $\mathbf{F}_{\mathbf{d}}$. Ce test peut être difficile à monter à cause de la sur-estimation de l'arithmétique, là encore pour le rendre plus efficace nous utilisons des techniques de filtrage présentées en 3.1.4 afin de vérifier la consistance des équations.

Dans le cas où $\dim F < \dim \tau$, le test de boîte intérieure est plus compliqué, ce qui arrive par exemple pour les robots à câbles de type grue. En effet, pour générer un effort sur le mobile, nous ne pouvons que tirer sur le câble mais pas le pousser. Ainsi, il est nécessaire d'installer plus de câbles que le nombre de degrés de liberté du mobile. Par exemple, le robot CoGiRo possède 8 câbles pour actionner un solide suivant 6 axes. Le système peut être alors vu comme redondant mais les systèmes impliquant le torseur statique restent sous-contraints.

Ainsi le *Wrench-feasible workspace* (WFW) est l'ensemble des poses du robot tel qu'un ensemble donné de tensions dans les câbles (entre une tension minimum pour tendre les câbles et une tension maximum de rupture) peuvent équilibrer tous les torseurs statiques donnés dans un ensemble (sous certaines hypothèses simplificatrices du modèle des câbles). De façon plus simple, dans quelles positions le robot est-il capable de générer un ensemble de forces/couples voulues en tenant compte des capacités de traction des câbles.

Nous présentons dans [GDM11] une condition adaptée pour ce type de manipulateur générant une forme sous-contrainte de l'équation 4.9. Pour cela nous utilisons un théorème de Rohn au chapitre 2 de [FNR⁺06].

Une condition suffisante pour qu'une boîte X soit intérieure au WFW est donnée si le système linéaire par intervalle 4.11 est dit *fortement faisable*. Pour cela, et dans le cas difficile de système rectangulaire, Rohn montre qu'il suffit de vérifier la faisabilité de 2^n système linéaire classique (scalaire) noté $W_y \cdot \tau = F_y$ avec W_y et F_y pris sur les sommets de la matrice \mathbf{W} et du vecteur $\mathbf{F}_{\mathbf{d}}$; y étant un indice d'énumération des sommets à vérifier. La faisabilité est montrée par la première partie d'une programmation linéaire (Nous avons utilisé ici GLPK). La certification nécessaire des résultats du simplex peut être obtenue par des arrondis rigoureux (voir [NS04]).

Les tests de boîtes intérieures et extérieures étant établis, nous utilisons, là encore, une approche de type *Branch and bound* afin de calculer l'espace de travail. Nous conseillons la lecture de [GDM11] afin de voir comment, à partir de cet algorithme de base, des améliorations sont possibles : bisection, simplification de la structure formelle des équations... Les résultats montrent qu'il est possible de vérifier un espace de travail total (une boîte 6D en position/orientation est

donnée) d'un robot à 8 câbles en quelques dizaines de secondes (dépendant de la taille de la boîte à vérifier, des dimensions du robot ...) ce qui est inférieur à une approche classique de discrétisation pour des résultats exacts : l'analyse par intervalle si elle est correctement pratiquée n'est pas plus lente qu'une approche *classique*. L'obtention de l'espace de travail total est de l'ordre de la demi heure mais comme nous allons le voir, ce calcul n'est pas nécessaire dans le cadre de la conception appropriée.

4.4 Conception de robot

La partie *conception* dimensionnelle est fondée sur les algorithmes précédents de certification de l'espace de travail. Ces algorithmes permettent de décrire l'ensemble des positions accessibles d'un robot (dans le sens où ses paramètres dimensionnels sont fixés pour une valeur scalaire) pour lesquelles ses propriétés sont certifiées et compatibles avec le cahier des charges donné.

Pour la partie conception, on ne s'intéresse qu'à une partie de cet espace modélisée par une boîte intervalle \mathbf{X}_d , là encore donnée par le constructeur. En effet, son robot doit être capable de décrire un hypercube englobant les trajectoires permettant de réaliser un certain nombre de tâches. Cette remarque est importante car le temps de calcul nécessaire pour certifier un hypercube est largement inférieur à celui nécessaire pour calculer les espaces de travail.

L'extension aux intervalles des contraintes $C(\mathbf{X}, \Xi, \mathbf{E}_v)$ pour un ensemble de valeurs des dimensions du robot $\xi \in \Xi$ nous permet de vérifier les propriétés d'un ensemble de robots pour l'espace continu considéré \mathbf{X}_d .

Nous devons donc déterminer :

$$\{\xi | \forall X \in \mathbf{X}_d, \forall \mathbf{E}_v \in \mathbf{E}_v, C(\mathbf{X}, \Xi, \mathbf{E}_v) = \mathbf{0} \text{ ou } \leq \mathbf{0}\} \quad (4.12)$$

Le principe est alors simple : nous utilisons un algorithme de *branch and bound* qui applique des bisections sur le vecteur intervalle Ξ jusqu'à certification de l'ensemble des propriétés du robot, $C(\mathbf{X}_d, \Xi, \mathbf{E}_v)$, pour l'ensemble des points de l'espace de travail \mathbf{X}_d .

En général, nous avons appliqué deux niveaux de bisection principaux : une procédure de haut niveau qui stocke (ou élimine) des ensembles de paramètres géométriques du robot sous forme de boîtes Ξ_t qui certifient (ou non) les performances pour l'ensemble des points \mathbf{X}_d .

Dans un second niveau de bisection, la procédure de haut-niveau appelle une sous-procédure qui vérifie que les performances \mathbf{E}_v sont réalisables sur l'ensemble continu de points de l'espace de travail \mathbf{X}_d pour un ensemble de robots Ξ_t . Nous vérifions alors les propriétés via une forme épaissie des contraintes C pour un ensemble de géométries possibles pour le robot. Il est alors nécessaire de bissecter les propriétés \mathbf{E}_v ainsi que l'espace de recherche \mathbf{X}_d , ce qui constitue de le second niveau de recherche. Mais ceci est en tout point semblable aux approches présentées précédemment sur la vérification des propriétés.

Ce principe général est décrit dans [MD08] et j'ai dirigé les travaux de plusieurs postdoctorants afin de l'implémenter pour des robots particuliers (voir [ODM09, HOS⁺10]).

Notons que, suivant les propriétés recherchées, la dimension de l'espace généralisé du robot et le nombre de variables peut être grand. Les calculs deviennent

assez lourd. Nous nous appuyons alors sur la parallélisation possible des algorithmes disponible sous ALIAS.

4.5 Conclusion

La conception appropriée d'un robot est une méthodologie bien structurée où chaque étape est clairement identifiée. Cependant, son implantation reste un savoir-faire un peu particulier qui nécessite une certaine expertise et une bonne maîtrise de l'utilisation de l'analyse par intervalles. En effet, il faut avoir conscience des avantages et des inconvénients d'une modélisation par intervalles, prendre en compte les défauts de cette arithmétique et formuler correctement les problèmes pour cette approche. La vision ensembliste n'est pas intuitive : remplacer des approches itératives par des approches contractantes demande un changement de point de vue pourtant essentiel pour créer des algorithmes efficaces. La formation d'étudiants et de chercheurs est alors un point essentiel et j'ai participé activement à cette tâche. J'ai participé à l'encadrement de 3 doctorants, en dehors de la robotique, sur ces questions d'analyse par intervalles : 2 d'entre eux, chercheurs et enseignants-chercheurs en informatique, continuent de contribuer sur des questions de robotique en utilisant des approches par intervalles.

En matière de perspectives, quelques pistes mentionnées restent toujours d'actualité, notamment la prise en compte de la dépendance des paramètres dans les algorithmes de filtrages et de calcul des contraintes. Ces pistes restent un travail collaboratif entre roboticiens modélisant les problèmes et numériciens pour le développement d'algorithmes d'analyse par intervalles. Nous verrons, par la suite, que mes recherches sur la vérification des valeurs propres des matrices intervalles sont dictées par ce constat.

Mais cette étape de conception intervient avant la fabrication du robot, conçu en considérant un modèle théorique souvent simplifié. Le système réellement réalisé est bien évidemment différent de celui proposé lors de la conception : de nombreuses erreurs détériorent le comportement réel au regard de ce qui est attendu théoriquement. Si nous voulons améliorer des performances comme par exemple la précision d'un robot, il est alors nécessaire de l'étalonner, c'est à dire de déterminer la valeur *effective* des paramètres d'un modèle de robot.

Bibliographie

- [BHG95] Soumya Bhattacharya, Himanshu Hatwal, and Arindam Ghosh. On the optimum design of stewart platform type parallel manipulators. *Robotica*, 13(02) :133–140, 1995.
- [CJ09] Gilles Chabert and Luc Jaulin. Contractor programming. *Artificial Intelligence*, 173(11) :1079–1100, 2009.
- [Cla91] Reymond Clavel. Conception d" un robot parallèle rapide à 4 degrés de liberté. 1991.
- [CW98] Damien Chablat and Philippe Wenger. Moveability and collision analysis for fully-parallel manipulators. In *RoManSy*, pages 1–8. Iftom, 1998.
- [FNR⁺06] Miroslav Fiedler, Josef Nedoma, Jaroslav Ramik, Jiri Rohn, and Karel Zimmermann. *Linear optimization problems with inexact data*. Springer Science & Business Media, 2006.
- [GDM11] Marc Gouttefarde, David Daney, and Jean-Pierre Merlet. Interval-Analysis-Based Determination of the Wrench-Feasible Workspace of Parallel Cable-Driven Robots. *IEEE Transactions on Robotics*, 27(1) :001–013, 2011.
- [GMD06a] Marc Gouttefarde, J-P Merlet, and David Daney. Determination of the wrench-closure workspace of 6-dof parallel cable-driven mechanisms. In *Advances in Robot Kinematics*, pages 315–322. Springer, 2006.
- [GMD06b] Marc Gouttefarde, Jean-Pierre Merlet, and David Daney. Determination of the Wrench-Closure Workspace of 6-DOF Parallel Cable-Driven Mechanisms. In B. Roth J. Lenarcic, editor, *Advances in Robot Kinematics*, pages 315–322, Ljubljana, Slovenia, 2006. Springer.
- [HOS⁺10] Kanako Harada, Denny Oetomo, Ekawahyu Susilo, Arianna Menciassi, David Daney, Jean-Pierre Merlet, and Paolo Dario. A reconfigurable modular robotic endoluminal surgical system : vision and preliminary results. *Robotica*, 28 :171–183, 2010.
- [Hub10] Julien Hubert. *Static analysis of parallel manipulators singularities*. Theses, École Nationale Supérieure des Mines de Paris, September 2010.
- [HW03] Eldon Hansen and G William Walster. *Global optimization using interval analysis : revised and expanded*, volume 264. CRC Press, 2003.

- [Jau01] Luc Jaulin. *Applied interval analysis : with examples in parameter and state estimation, robust control and robotics*, volume 1. Springer, 2001.
- [LWOK] Xin-Jun Liu, Jinsong Wang, Kun-Ku Oh, and Jongwon Kim. A new approach to the design of a delta robot with a desired workspace. *Journal of Intelligent and Robotic Systems*, 39(2) :209–225.
- [MD01] Jean-Pierre Merlet and David Daney. A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot. In *In 2nd Workshop on Computational Kinematics*, pages 167–176, Seoul, Corée, République De, May 2001.
- [MD05] Jean-Pierre Merlet and David Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 942–947, Barcelona, Espagne, 2005. IEEE.
- [MD06a] J-P Merlet and Peter Donelan. On the regularity of the inverse jacobian of parallel robots. In *Advances in Robot Kinematics*, pages 41–48. Springer, 2006.
- [MD06b] Jean-Pierre Merlet and David Daney. Legs interference checking of parallel robots over a given workspace or trajectory. In *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, États-Unis, 2006. IEEE.
- [MD08] J.-P. Merlet and D. Daney. Appropriate Design of Parallel Manipulators. In Lihui Wang and Jeff Xi, editors, *Smart Devices and Machines for Advanced Manufacturing*, pages 1–25. Springer London, 2008.
- [Mer04] J-P Merlet. Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. *The International Journal of robotics research*, 23(3) :221–235, 2004.
- [Mer06] Jean-Pierre Merlet. *Parallel Robots, 2nd Edition*, volume 128. Springer, 2006.
- [Moo66] Ramon E Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.
- [NS04] Arnold Neumaier and Oleg Shcherbina. Safe bounds in linear and mixed-integer linear programming. *Mathematical Programming*, 99(2) :283–296, 2004.
- [OC02] Erika Ottaviano and Marco Ceccarelli. Optimal design of capaman (cassino parallel manipulator) with a specified orientation workspace. *Robotica*, 20(02) :159–166, 2002.
- [ODM09] Denny Oetomo, David Daney, and Jean-Pierre Merlet. Design Strategy of Serial Manipulators With Certified Constraint Satisfaction. *IEEE Transactions on Robotics*, 25(1) :1–11, 2009.
- [ODSM08a] D Oetomo, David Daney, Bijan Shirinzadeh, and Jean-Pierre Merlet. An Interval-Based Method for Workspace Analysis of Planar Flexure-Jointed Mechanism. *Journal of Mechanical Design*, 131(1) :011014–011014, 2008.

- [ODSM08b] Denny Oetomo, David Daney, Bijan Shirinzadeh, and Jean-Pierre Merlet. Certified workspace analysis of 3RRR planar parallel flexure mechanism. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3838–3843, Pasadena, États-Unis, 2008. IEEE.
- [RR98] Georg Rex and Jiri Rohn. Sufficient conditions for regularity and singularity of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(2) :437–445, 1998.
- [SDPN01] Yu Xin Su, Bao Yan Duan, B Peng, and RD Nan. A real-coded genetic optimal kinematic design of a stewart fine tuning platform for a large radio telescope. *Journal of Robotic Systems*, 18(9) :507–516, 2001.
- [Vis96] Peter Vischer. *Improving the accuracy of parallel robots*. PhD thesis, 1996.
- [ZA97] Kouros E Zanganeh and Jorge Angeles. Kinematic isotropy and the optimum design of parallel manipulators. *The International Journal of Robotics Research*, 16(2) :185–197, 1997.
- [ZWL05] Dan Zhang, Lihui Wang, and Sherman Y Lang. Parallel kinematic machines : design, analysis and simulation in an integrated virtual environment. *Journal of Mechanical Design*, 127(4) :580–588, 2005.

Chapitre 5

L'étalonnage des robots manipulateurs

5.1 Introduction

5.1.1 Problématiques de l'étalonnage des manipulateurs parallèles

Lors de la mise en fonction d'un robot, le comportement observé n'est généralement pas celui attendu. Ses performances effectives ne sont pas celles prédites par son modèle théorique. Ainsi, la commande du manipulateur, fondée sur ce modèle, est détériorée par des erreurs d'origines diverses. Afin d'améliorer la qualité d'une tâche, et ainsi de répondre aux attentes données par un cahier des charges, il est nécessaire de procéder à un étalonnage. Cette procédure doit permettre d'obtenir des corrections du modèle afin qu'il approxime mieux le comportement du robot.

Les détériorations de performances sont dues à un ensemble d'erreurs que nous décrirons à la sous-section 5.1.3. Comme nous le verrons dans ce chapitre, le but de l'étalonnage n'est pas de déterminer directement la valeur de ces erreurs mais plutôt de compenser leur influence sur le comportement du robot.

Pour cela, il faut identifier les paramètres d'un modèle choisi qui approxime, *au mieux*, le comportement du robot. Mais cette identification n'est possible que si nous créons une information redondante sur l'état interne de robot – voir section 5.2. Dans la première partie de mes contributions j'ai cherché à savoir comment créer cette information redondante à partir de mesures et/ou de contraintes. Le challenge est de diminuer l'influence des incertitudes liées à ces informations (erreurs de mesure ou tolérances) sur l'identification – voir section 5.4. Nous cherchons alors à analyser les solutions de l'identification, et à mieux les caractériser et à mieux maîtriser leur identifiabilité.

Mes contributions sur ce sujet sont principalement appliquées aux robots parallèles et à câbles. Par rapport à l'étalonnage des robots séries, il existe plusieurs particularités que nous décrirons en sous-section 5.1.2. Cependant les principes et étapes de l'étalonnage restent similaires.

5.1.2 Contexte

Nous nous intéresserons à l'étalonnage statique des manipulateurs en nous focalisant sur des critères de performance statique et en particulier la précision. Le robot sera donc immobilisé lors des mesures de ses performances pour une charge invariante. Ainsi nous ne prendrons pas en compte les phénomènes dynamiques tels que les accélérations, la variation des efforts ou autres paramètres fonction du temps. L'étalonnage dynamique est un domaine à part entière – voir [PG01, RVA⁺06].

L'étalonnage des robots manipulateurs est bien souvent la dernière étape avant leur mise en service. Elle est alors tributaire de nombreux choix réalisés en amont : l'architecture, les dimensions, les modèles, les matériaux utilisés, les capteurs et actionneurs installés, la qualité de la fabrication et de l'assemblage ou la commande sont, en général, pré-déterminés. Ces choix se font au regard d'un cahier des charges dont un des éléments principaux est la précision du positionnement. De façon caricaturale, les étapes précédant l'étalonnage (la conception, la commande) cherchent toutes à introduire le minimum d'incertitude ou d'erreur, ou à être robustes par rapport à celles-ci. Or le but de l'étalonnage est, lui aussi, d'améliorer la précision du robot, d'où le paradoxe qu'un robot bien conçu est intrinsèquement un robot difficile à étalonner. En effet, l'identification de l'erreur sur les paramètres est d'autant plus difficile que le modèle est peu sensible aux variations de ces paramètres. Quelques uns des robots que j'ai eu à étalonner ont été conçus au sein de l'équipe Coprin en utilisant les méthodes de conception robuste décrites précédemment : leur étalonnage a été d'autant plus difficile ...

Une autre particularité importante de mes travaux est le type d'architecture des manipulateurs étudiés : les robots parallèles. Ce sujet, comparé à l'étalonnage des robots séries, reste relativement récent et peu de papiers en ont traité avant les années 2000 [ZYM98, WHA95, Inn95]. Beaucoup de points communs relient l'étalonnage des deux types de manipulateur : les principes généraux, les différentes étapes, les outils d'identification ... cependant les modèles utilisés ne sont pas les mêmes.

Pour les robots séries, toutes les articulations sont actives et la valeur de leurs coordonnées articulaires est mesurée par un capteur proprioceptif. Pour les robots parallèles, les chaînes cinématiques qui les composent peuvent contenir des articulations passives, c'est-à-dire non-actionnées, et surtout non-instrumentées. Leurs valeurs sont alors des inconnues de plus à considérer dans nos systèmes d'équations. Afin d'éliminer ces variables, les modèles géométriques des robots parallèles sont bâtis sur les hypothèses fortes que certaines articulations sont parfaites et parfaitement assemblées. Par exemple, les chaînes cinématiques de la plate-forme de Gough sont de type RRP₃RR. Les équations de son modèle géométrique inverse sont bâties sur l'hypothèse que le cardan (RR) et l'articulation sphérique (RRR) sont parfaites afin de simplifier sa représentation par une chaîne SPS. La commande est alors fondée sur ces modèles simplifiés sans nécessairement calculer les valeurs des paramètres de ces articulations passives. Nous pouvons faire des remarques semblables pour l'ensemble des robots parallèles quel que soit leur nombre de degrés de liberté. Dans la plupart des cas, leurs modèles impliquent des hypothèses sur leur géométrie : sur la direction de certains axes (parallèles, orthogonales), sur des symétries, sur certaines dimensions etc. Nous aurons donc une différence entre :

- un modèle qui décrit l'ensemble des défauts du manipulateur, par exemple en utilisant une modélisation de type Denavit-Harenberg modifiée qui nous donnera l'ensemble minimal mais complet des paramètres géométriques à considérer,
- un modèle simplifié qui sera celui généralement utilisé par la commande.

Nous verrons dans la section 5.3.3 qu'il est intéressant de faire une analyse globale de l'influence des erreurs sur les performances d'un robot, et d'en extraire l'influence des principales erreurs afin de valider un modèle : les différents types d'incertitudes qui seront listés dans la prochaine sous-section. Cependant c'est le modèle final, celui de la commande qui nous intéresse. Ainsi, il sera intéressant d'identifier ses paramètres afin qu'ils minimisent un critère censé refléter les performances du robot.

Enfin, dans ce document, nous nous limiterons à des *approches basées modèle*. En effet, même s'il est difficile de prendre en compte l'ensemble des phénomènes physiques pouvant détériorer les performances d'un manipulateur, la géométrie et la mécanique nous aide grandement à modéliser ces phénomènes. Il n'est alors pas nécessaire d'utiliser des méthodes d'Intelligence Artificielle, comme par exemple dans [YYLW08] pour déterminer les compensations nécessaires pour corriger le comportement du robot. Dans nos recherches nous n'avons que peu l'occasion de nous intéresser aux robots de moins de 6 degrés de liberté alors que leur étalonnage est particulier. En effet, pour nombre d'entre eux, leurs degrés de liberté sont mécaniquement limités par des contraintes sur leur géométrie (direction de certaines axes ...). Dans le cadre d'un étalonnage, il faut remettre en cause ces hypothèses qui ne peuvent pas être réalisées exactement en pratique et donc considérer ces défauts. De plus, l'identification, même précise, de ces erreurs ne permettra pas de compenser l'erreur de positionnement car elle évoluera dans des dimensions non contrôlées du robot. Nous conseillons au lecteur des travaux en la matière [PCDG00, HKKP02].

5.1.3 Origine des erreurs

Les dégradation des performances d'un robot sont dues à des erreurs. Leurs origines sont multiples : on peut citer par exemple [SK08] :

- Les erreurs d'origines géométriques
 - erreurs de fabrication et d'assemblage qui influent sur les valeurs des dimensions, des orientations des articulations et des corps du robots. Elles peuvent être estimées par les tolérances de fabrication et d'assemblage.
 - erreurs sur l'estimation du positionnement des repères associés aux corps du robot. Par exemple, le repère de base du robot par rapport au repère global de référence (placement du robot dans une scène) ou d'un outil spécifiquement utilisé pour l'étalonnage (ex : une mire).
- Les erreurs d'origines non-géométriques
 - erreurs sur les paramètres des modèles des capteurs, par exemple un gain et un offset d'un capteur angulaire, paramètre intrinsèque d'une caméra... Une estimation des erreurs peut être donnée par un étalonnage spécifique de ces capteurs.
 - erreurs sur les paramètres inertiels,
 - erreurs dues aux flexibilités des corps,

- erreurs dues à l'influence des variations de température ou des vibrations,

Il est difficile et, souvent pas nécessaire, de chercher à identifier l'ensemble de ces erreurs. Leur observation directe ou indirecte nécessite des mesures additionnelles. En effet, pour la plupart des manipulateurs, le nombre de capteurs proprioceptifs installés suffit à la commande mais ne permet pas d'obtenir une information redondante sur l'état interne complet du robot. Il est alors nécessaire d'instrumenter le robot avec des capteurs proprioceptifs et/ou extéroceptifs supplémentaires, comme nous le verrons à la section 5.2. Le type de ces capteurs sera alors fonction de la nature physique des erreurs à identifier.

La prise en compte de l'ensemble des erreurs possibles augmente donc le nombre d'informations à considérer mais aussi la complexité des modèles utilisés afin de les lier. Est-ce que ceci est nécessaire? C'est-à-dire, l'influence des ces erreurs sur les performances du robot est-elle suffisamment importante pour qu'il soit nécessaire de la considérer? La réponse ne peut être donnée qu'au regard d'un critère de qualité attendu fourni par un cahier des charges ou de montrer qu'elles sont négligeables comparées à l'influence d'autres erreurs. Cependant, la littérature [MWZ93] considère que les erreurs d'origine géométrique ainsi que la mésévaluation des offsets liés aux capteurs sont prépondérants : c'est donc un étalonnage géométrique qui est généralement considéré. Cependant, l'amélioration des performances des manipulateurs ces dernières années et l'utilisation de segments flexibles font évoluer cette tendance et un étalonnage géométrico-statique est maintenant considéré [EM05, ENM10].

Le modèle choisi pour synthétiser le comportement du robot doit pouvoir considérer l'ensemble des paramètres qui ont le plus d'influence et négliger les autres. Concrètement, une partie des phénomènes physiques ne sont pas considérés (ex : influence de la température) ou simplifiés (ex : certaines articulations sont considérées comme parfaites) : ces approximations introduisent une erreur globale que nous appellerons une **erreur de modèle**, voir [AAC12]. Cette erreur est assez peu étudiée dans la littérature, et cela a conditionné une partie de mes contributions et des travaux que j'ai pu diriger.

Nous aurons donc à considérer trois grandes classes d'erreurs :

- Les erreurs sur les paramètres que nous chercherons à identifier : ces erreurs sont systématiques, proviennent d'une mauvaise connaissance des dimensions géométriques, des offset articulaires ou des conditions environnementales. Nous aurons accès à une approximation de ces valeurs (données lors de la conception). Elles sont indépendantes de la configuration du robot.
- Les erreurs sur les informations fournies sur l'état interne du robot : elles proviendront soit de mesures, soit de jeux mécaniques ; elles sont donc aléatoires mais connues à une incertitude près. Leurs valeurs sont dépendantes de la configuration statique dans laquelle se trouve le robot.
- Les erreurs sur les modèles considérés pour lier les deux précédentes classes d'erreurs. Elles sont fonction des approximations et des hypothèses considérées. Elles doivent être négligeables afin de pouvoir procéder à l'étalonnage, nous verrons cela à la section 5.5.

Le choix du modèle est alors une question importante car il doit être en lien avec celui qui est implanté dans la commande afin de positionner le robot lors de son fonctionnement. L'étalonnage revient donc à identifier la valeur des paramètres du modèle utilisés par la commande en fonction d'informations (mesures

directes ou indirectes) sur l'état observé du manipulateur.

5.1.4 Positionnement des recherches

La vision précédente de l'étalonnage n'est pas si simple et ceci pour trois raisons :

- L'information sur l'imprécision du robot est obtenue au travers des informations supplémentaires sur son état, ce qui nécessite d'ajouter des capteurs internes/externes ou d'imposer des contraintes mécaniques. Il est alors nécessaire de considérer l'ensemble de ce nouveau système (robot+système de mesures/contraintes) et donc un nouveau modèle que nous appellerons **modèle d'étalonnage**, reliant tout ou une partie des paramètres du modèle de la commande du robot et d'autres paramètres liés à l'ajout des mesures supplémentaires. Les erreurs de mesures ne se propageront donc pas de la même façon dans les deux modèles.
- Comme nous l'avons mentionné, le modèle utilisé ne représente pas exactement le comportement du robot : la nature et le nombre des paramètres qui le composent est lié au niveau de simplification considéré via des hypothèses (articulations parfaites, influence de la température ou des forces internes négligées...). Ces erreurs de modèle ont, là encore, une influence sur la qualité de l'étalonnage et donc sur les performances finales du robot.
- Une conséquence du point précédent est que la valeur des paramètres obtenue par le processus d'étalonnage ne reflète peut-être pas la valeur physique de ces paramètres. En effet, leur identification est fondée sur un modèle erroné instancié par des mesures incertaines mais, nous rappelons que nous recherchons un modèle qui approxime le comportement du robot afin de le commander.

Ces erreurs de modèle deviennent une question centrale de l'étalonnage mais elles ne sont pas, ou très peu, considérées dans la littérature. Cette question est un point d'entrée de mes contributions scientifiques. Elle a motivé mes axes de recherches et l'orientation des deux thèses qui j'ai encadrées.

- Dans la thèse de Thibault Gayral, nous avons cherché à identifier le modèle complexe du robot (structure d'un satellite d'observation terrestre à articulation flexible) à l'aide d'informations très riches¹ permettant de mesurer les positionnements de l'ensemble des corps du robot. Puis, nous avons dégradé le modèle en vérifiant que les conséquences restent acceptables sur la précision du robot et sur l'identifiabilité des paramètres des modèles incrémentaux. Ceci nous a permis de proposer un modèle simple qui autorise l'auto-étalonnage à partir d'informations fournies par l'appareil optique embarqué sur le télescope.
- Dans la thèse de Julien Alexandre dit Sandretto, nous avons cherché à garder la structure des modèles simples des robots à câbles directement exportés de la plate-forme de Gough. Cependant, il faut prendre en compte les incertitudes liées à la nature déformable des câbles, ce qui introduit une erreur de modèle peu identifiable au regard de la qualité des informations sur les tensions des câbles qui auraient pu nous renseigner. Nous introduisons alors le concept de modèle épaissi, [STD13]. En

1. Nous avons utilisé un système de photogrammétrie fourni par notre partenaire industriel Thales.

effet, l'idée est d'utiliser une représentation par intervalles des variables d'un modèle simplifié qui inclut l'ensemble des modèles plus complexes.

Il a alors été nécessaire de développer des outils afin de caractériser une solution adaptée à ce type d'étalonnage par intervalles.

Ces questions portent sur la modélisation et la structure des équations d'étalonnage et des incertitudes. Mais ces modèles sont instanciés par des valeurs numériques qui vont plus ou moins exciter des paramètres à identifier, ceci en fonction des configurations du robot dans lesquelles sont pratiquées les mesures. Nous pouvons donc agir sur leur choix afin d'améliorer l'identifiabilité des inconnues et la robustesse de l'étalonnage par rapport aux erreurs de mesures. Cet autre aspect de mes travaux a fait l'objet de travaux avec :

- B. Madeline sur l'élaboration d'une méthode qui détermine les configurations du robot dans lesquelles sont pratiquées les prises de mesures statiques. L'algorithme de base est sous-optimal, et fait appel à de l'optimisation locale afin de fournir des positions maximisant certains critères. Nous avons proposé d'utiliser des méta-heuristiques afin de limiter le problème de convergence locale dans l'optimisation des configurations de mesures pour l'étalonnage.
- T. Gayral sur les conditions de l'étalonnage : comment déterminer la précision nécessaire sur les paramètres inconnus compatibles avec un objectif de performance du robot, sur les mesures et sur les critères d'arrêt des algorithmes d'identification.

La quasi-totalité des papiers sur l'étalonnage utilise une approche aux moindres carrés non-linéaires afin de fournir une solution à l'identification des paramètres. Ces méthodes, aux nombreuses variantes, sont globalement bien adaptées à la problématique. Cependant, il est difficile d'interpréter les résultats obtenus. En effet, il est intéressant en pratique de fournir des informations complémentaires à l'utilisateur sur les dispersions des résultats en fonction de la propagation des erreurs dans le système. Ceci n'est possible qu'à condition d'avoir des modèles de bruits associés à chacune des variables. Cela peut s'avérer compliqué et dépendant d'hypothèses fortes quant à la distribution des erreurs, parfois peu réalistes.

Nous proposons alors une solution au pire cas où les mesures et leurs incertitudes sont modélisées par des intervalles. Ceci nous permet d'unifier au sein des systèmes d'équations d'étalonnage des erreurs sur les mesures mais aussi sur les modèles (modèles épaissis vus précédemment). Il est alors nécessaire de caractériser une solution pour ces systèmes sur-contraints par intervalles. Une première approche publiée a montré les limites de ne pas considérer les erreurs de modèles. La thèse de J. Alexandre dit Sandretto propose une seconde caractérisation des solutions qui prend en compte ces erreurs.

J'ai fourni des solutions originales sur l'ensemble des grandes étapes de l'étalonnage aux travers de mes contributions et de celles des étudiants que j'ai encadrés sur ces sujets. Les sections suivantes les présentent en suivant ce cheminement : de l'obtention des informations nécessaires à l'étalonnage à la construction des modèles jusqu'à l'identification de leurs paramètres. Cependant, deux idées maîtresses ont guidé mes recherches : constituer des modèles qui approximent au mieux le comportement du robot et rendre leur identification la plus robuste possible par rapport aux incertitudes.

5.2 Créer une information redondante sur l'état interne du robot

L'objectif est d'identifier les paramètres d'un modèle qui approxime le comportement du manipulateur. Ces paramètres sont considérés comme des inconnues mais nous en avons une estimation théorique fournie lors de la conception. Il est d'abord nécessaire de **créer une information redondante sur l'état interne du robot**. En effet, les informations disponibles sont fournies par les capteurs proprioceptifs associés aux articulations actives. Pour la plupart des manipulateurs, elles suffisent pour déterminer la configuration du robot en résolvant un système d'équation bien contraint. Une information supplémentaire doit alors être obtenue pour déterminer les paramètres du robot en plus de sa configuration.

5.2.1 Obtenir une redondance métrologique du système

Nous avons trois possibilités afin d'obtenir cette information redondante :

- Ajouter un système de mesure extéroceptif qui fournit des informations sur la configuration du robot. Plus précisément, ces informations proviennent de la mesure partielle ou complète de la position/orientation de la plate-forme mobile ou des corps du manipulateur.
- Ajouter des capteurs proprioceptifs. Pour l'étalonnage géométrique des robots parallèles, cela revient généralement à ajouter des capteurs sur les articulations passives du robot [Dan03] ou sur les corps du robot (mobile ou segment). Une autre possibilité est d'ajouter des informations de nature non-géométrique. Par exemple, pour un étalonnage géométrico-statique, des capteurs de forces additionnels permettent de considérer les relations statiques. Dans ces cas, le manipulateur est en redondance métrologique – voir [MAR00, Cor08].
- Contraindre mécaniquement les degrés de liberté du manipulateur. Pour cela, la mobilité de plusieurs corps (plateforme mobile, segment ou articulation) est alors fixée. Dans ce cas encore, le mécanisme contraint se comporte comme un robot en redondance métrologique : en effet, la diminution de ses degrés de liberté, et donc de l'espace généralisé, rend redondants les capteurs proprioceptifs installés par défaut.

Dans la littérature, on parlera d'étalonnage lorsque l'information est apportée par un système de mesure ou par des contraintes externes. Ce système n'est alors utilisé que pour étalonner le robot : il est supprimé lors du fonctionnement courant du manipulateur. D'autre part, on parlera d'auto-étalonnage lorsque on n'utilisera que les capteurs installés à demeure sur le manipulateur.

5.2.2 Problématique

Comme le propose le paradigme de Wampler et Hollerbach [WHA95], nous pouvons unifier ces trois façons d'obtenir une information redondante par un *étalonnage implicite*. Pour cela il faut considérer le système globalement : le robot et son système de mesures ou de contraintes mécanique. Dans le cas d'une mesure de positionnement externe sans contact (par exemple, en utilisant le système de vision par ordinateur) l'information de mesure est considérée comme une liaison mécanique virtuelle. On peut alors décrire le système complet, c'est à

dire le manipulateur et le système de mesure ou de contrainte, comme un mécanisme à part entière, en redondance métrologique. De plus, les auteurs proposent d’assimiler les erreurs de mesures aux jeux mécaniques produits par les systèmes de contraintes. Cela permet de décrire toutes les formes d’étalonnage dans une même vision. On pourra néanmoins remarquer que la nature des incertitudes n’est pas homogène : tolérance mécanique vs erreur de mesure.

Cependant, nous pouvons faire plusieurs remarques :

- L’étalonnage est à différencier de l’auto-étalonnage car il nécessite l’apport d’un appareil externe qui n’est utilisé que pour cette procédure. Il sera donc désinstallé lors de l’utilisation courante du robot. Cependant, lors de la phase d’identification, l’ensemble des paramètres est déterminé : aussi bien ceux du robot, *paramètres internes*, que des paramètres additionnelles pour modéliser l’appareil externe dans la boucle *paramètres externe*, voir [Dan03]. De façon simple, la méthode classiquement utilisée des moindres-carrés, voir section 5.4.1, revient à répartir les erreurs entre la prédiction théorique de la configuration du système et son observation, sur l’ensemble des paramètres internes comme externes. Cette répartition des erreurs est liée à la structure des équations mais aussi aux erreurs de mesure et de modèle. Il est aussi possible d’ajouter des pondérations afin de prendre en compte un a-priori sur les erreurs commises comme nous l’avons fait dans [JADG12]. Cependant, lors de la phase de compensation, après étalonnage, seuls les paramètres internes interviennent dans les modèles utilisés par la commande. Il n’est pas clair que les valeurs déterminées permettront d’améliorer les performances du robot sachant qu’elles ont été identifiées en présence de paramètres externes.
- Les manipulateurs en redondance d’actionnement qui sont, pourtant, naturellement en redondance métrologique, posent problème. En effet, avant étalonnage le modèle du robot est incertain. Une commande imposée à ce type de robot entraîne une augmentation des forces internes afin de compenser les défauts. Ainsi, soit le robot peut être détérioré, soit une certaine flexibilité dans ses articulations ou dans ses corps absorbera ces défauts au prix d’une déformation. Dans le second cas, ces degrés de liberté supplémentaires doivent être pris en compte dans un modèle d’étalonnage plus complexe où ces déformations internes sont mesurées. Ainsi, le robot ne peut plus être considéré en redondance d’actionnement. Dans certains cas, nous avons considéré une autre solution en débrayant (ou en laissant libre) autant de moteurs que nécessaire afin de supprimer la redondance d’actionnement sans pour autant revenir sur la redondance métrologique. Ces solutions sont à privilégier, si possible, lors d’un étalonnage sous contraintes mécaniques, voir [Dan99].

5.2.3 Contributions

Des états de l’art récents et complets à ce sujet sont disponibles au sein des thèses [Gay13] et [ADS13]. Cependant, au cours de ces années de recherches, nous avons utilisé différentes sources d’informations redondantes, par le biais de :

- **un système de mesure externe.** Afin de connaître la précision des manipulateurs, il est naturel de considérer, tout d’abord, les systèmes de mesure qui fournissent directement des informations sur la position et/ou

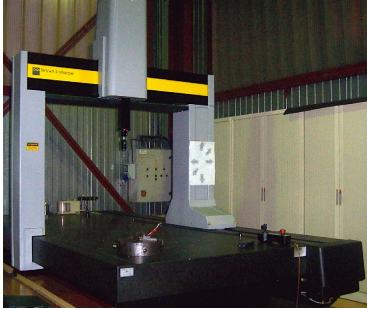


FIGURE 5.1 – Machine à mesurer tridimensionnelle, Brown & Sharpe, CMW



FIGURE 5.2 – Laser tracker, robot CoGiRo, Lirimm

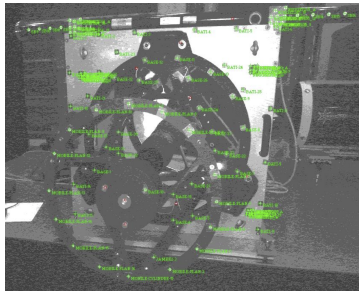


FIGURE 5.3 – Points mesurés par photogrammétrie, Thales



FIGURE 5.4 – Vision par ordinateur, Lasmea

l'orientation de l'organe terminal du robot dans un repère de référence. Nous avons utilisé plusieurs types de système : machine à mesurer dans [Dan03], voir Fig. 5.2, système de vision dans [DACP06b], voir Fig. 5.4, "laser tracker" dans [ADSTDG13, JADG12], voir Fig. 5.2, photogrammétrie dans [GDB13, GDD13], voir Fig. 5.3 ou un simple télémètre laser dans [MD10].

- **l'utilisation de capteurs internes.** Nous avons utilisé des capteurs de rotation sur certaines articulations passives dans [Dan03], la redondance d'actionnement des robots à câbles dans [ADSTDC14] ou un instrument optique embarqué dans [Gay13].
- **l'ajout de contraintes mécaniques.** Les études théoriques de ce type d'étalonnage sont disponibles dans [Dan99, Dan00].

Nous avons maintenant une longue expérience en matière de choix du type de mesure pour l'étalonnage. Le cas traité pour CMW, présenté dans [Dan03], a été formateur. Il montre que les méthodes d'auto-étalonnages (vs étalonnage) donnaient de meilleurs résultats pour une mise en œuvre expérimentale beaucoup plus simple.

Nous nous sommes alors efforcés de privilégier cette approche. Ainsi, dans

les thèses de T. Gayral et J. Alexandre dit Sandretto, nous avons développé des méthodes d’auto-étalonnage :

- Pour les robots à câbles, dans certaines conditions, nous avons montré expérimentalement dans [JADG12] comment obtenir des résultats comparables aux méthodes d’étalonnages classiques sans utiliser une instrumentation externe. En effet, le robot peut être vu en redondance métrologique (plus de câbles que de degrés de liberté du robot). Il est à noter que cet article souligne l’importance de la modélisation des équations de contraintes sans l’utilisation du modèle géométrique direct, ce qui est pourtant courant dans des calibrations comparables mais pose des problèmes numériques importants.
- Dans la thèse de T. Gayral, qui consiste à étalonner un satellite d’observation terrestre, nous utilisons l’instrument optique installé sur l’un des miroirs afin de mesurer les erreurs de positionnement du robot. Nous montrons qu’il est alors possible d’auto-étalonner le robot dans l’espace, ce qui est une avancée essentielle pour la mise en orbite de ce type de télescope actif.

5.3 Modélisation des équations de l’étalonnage

5.3.1 Modèle implicite simplifications et erreurs

Une fois que le type d’information redondante est choisi, il est nécessaire de le lier avec les paramètres du robot au travers d’un système d’équations.

Comme le présentent Wampler et Hollerbach dans [WHA95], un **modèle implicite** peut être obtenu en considérant la ou les boucle(s) de fermeture des chaînes cinématiques (réelle ou virtuelle) du système complet (robot, système de mesures ou de contraintes). Nous avons repris cette approche dans nombre de nos contributions dont la plus aboutie sur ce sujet est décrite dans [Dan03].

Considérons un **modèle implicite parfait** qui lie des paramètres, dénommés par le vecteur ξ et des informations données par le vecteur M_k sur la configuration k du robot.

$$\mathcal{F}_k^*(M_k^*, \xi_t^*) = 0 \tag{5.1}$$

Remarques :

- Le vecteur ξ_t^* représente l’ensemble des paramètres géométriques réels et des paramètres modélisant exactement les phénomènes physiques auquel est soumis le robot. Ces paramètres sont indépendants de la configuration statique dans lequel il se trouve.
- Le vecteur M_k^* représente les informations exactes collectées sur le robot décrivant l’état du robot dans la configuration k . Elles comprennent les coordonnées articulaires mais aussi d’autres types d’informations suivant l’étalonnage mis en place : par exemple, les coordonnées généralisées ou certaines valeurs d’articulations passives en cas d’auto-étalonnage, direction des segments pour un étalonnage basé vision etc.

La construction de ces équations est plus simple pour les robots série. En effet, il est généralement possible d’obtenir une modélisation quasi-systématique en utilisant une paramétrisation de type DH, par exemple pour représenter la chaîne cinématique qui le compose : nous obtiendrons donc le nombre minimale,

mais complet, des paramètres à identifier. Pour les robots parallèles, l'approche consistant à modéliser toutes les chaînes cinématiques puis leurs contraintes de fermeture, nous donnera bien l'ensemble minimale de paramètres géométriques indépendants à considérer. Cependant, les variables articulaires des articulations passives (non-instrumentées) n'étant pas connues, elle doivent être ajoutées à la liste de inconnues, ce qui entraîne la non-identifiabilité d'une partie des paramètres géométriques. Là encore la simplification des modèles s'impose, ce qui correspond à l'approche utilisée pour le modèle de la commande (voir section 5.1.2)².

La forme symbolique des équations d'étalonnage statique est donc dépendante du choix du modèle et des simplifications retenues mais constante quelle que soit la configuration du robot choisie pour la mesure. Par contre son instanciation dépend de la configuration du robot dénotée par l'index k .

Le modèle parfait 5.1 est inaccessible à cause :

- des hypothèses simplificatrices décrites précédemment : négligence des phénomènes physiques auxquels est soumis le robot, simplification du comportement des articulations et corps du robot.
- des erreurs décrites à la section 5.1.3.

Ainsi, nous ne garderons alors qu'un sous-ensemble de ξ_t^* dénommé ξ^* qui représentera les paramètres géométriques et, éventuellement, statiques, à considérer effectivement en s'efforçant de garder les mêmes paramètres que ceux de la commande. Cela introduit donc une erreur de modèle noté $\epsilon_{\mathcal{F}} = \mathcal{F}_k(M_k^*, \xi) - \mathcal{F}_k^*(M_k^*, \xi_t^*)$ où \mathcal{F}_k note le modèle simplifié.

Aux erreurs de modèle viennent s'ajouter :

- Les erreurs ϵ_{M_k} sur les informations $M_k^* = M_k + \epsilon_{M_k}$ qui sont dues aux erreurs de mesures ou aux jeux dans les contraintes imposées sur les axes du robot.
- Les erreurs ϵ_{ξ} sur les paramètres considérés pour l'étalonnage $\xi^* = \xi_0 + \epsilon_{\xi}$. Nous souhaitons identifier ces erreurs ϵ_{ξ} grâce à la connaissance d'une estimée initiale ξ_0 fournie lors de la phase de fabrication du robot.

Ainsi le **modèle implicite** considéré sera de la forme, lorsque le robot est dans une configuration statique k :

$$\mathcal{F}_k(M_k + \epsilon_{M_k}, \xi_0 + \epsilon_{\xi}) = \epsilon_{\mathcal{F}} \quad (5.2)$$

Ce système fournit $\dim \mathcal{F}_k$ équations alors que les inconnues sont de dimension $\dim(\xi)$. Afin d'identifier ξ il sera donc nécessaire que le robot soit placé dans un nombre minimal de configurations de mesure K tel que $K * \dim \mathcal{F}_k > \dim(\xi)$. En effet, cette condition nous permet de considérer un système complet sur-contraint d'équations et ainsi de minimiser l'influence des erreurs ϵ .

Avant de nous intéresser à l'identification des inconnues ξ , nous présentons nos travaux sur les formes possibles des équations 5.2 sur trois aspects :

2. Prenons l'exemple d'une plate-forme de Gough composée de 6 chaînes de type 2RP3R. Vischer, dans [Vis96], montre qu'il est nécessaire de considérer 138 paramètres pour modéliser ce robot. Parmi ces paramètres, figurent les défauts de fabrication et d'assemblage des articulations cardans et rotules : défauts d'orthogonalité et de distance entre les axes des liaisons rotoïdes. Il n'est pas possible d'identifier ces défauts sans connaître les valeurs de variables articulaires des liaisons rotoïdes voir [Dan03]. Il est alors nécessaire, par hypothèse, de considérer ces articulations comme parfaites. Ceci nous permet de simplifier la modélisation des jambes de ce robot en les considérant comme des chaînes cinématiques de type SPS. Cette simplification nécessaire, permet alors de considérer les équations du modèle géométrique inverse telles que présentées par l'équation 2.6 en section 2.3.

- comment reformuler algébriquement ce système afin d'éliminer certaines variables du problème nécessaires à sa formulation mais inutiles en pratique pour l'étalonnage ?
- comment garder une structure simple de ses équations compatible avec le modèle de la commande mais considérant des approximations de phénomènes non modélisés ?
- comment à partir d'informations très riches sur la structure, dégrader un modèle complexe pour le simplifier ?

5.3.2 Élimination algébrique

Nous nous sommes tout d'abord intéressés à un problème particulier de l'étalonnage des robots parallèles qui est l'élimination de certaines variables dans les équations d'étalonnage. En effet, la modélisation du problème peut conduire à un système d'équations différent du système présenté en (5.2). Il peut être alors nécessaire d'introduire une troisième classe de variables, noté ζ et qui ne sont ni des informations du type M_k , ni des paramètres du robot à identifier.

Ces variables, dépendantes, notées ζ_k , ou non, notées ζ , de la configuration du robot k , peuvent être à considérer quel que soit le type d'étalonnage :

- lors d'un étalonnage utilisant un système de mesures externes mais qui ne mesure que partiellement la position/orientation du robot. Les variables ζ_k représentent alors l'information sur le positionnement qui n'est pas mesurée et donc inconnue.
- lors d'un auto-étalonnage. Les variables ζ_k peuvent représenter les valeurs d'une articulation passive non mesurée mais intervenant dans la modélisation
- lors d'un étalonnage sous contraintes mécaniques. Les variables ζ peuvent représenter une valeur de l'axe contraint, inconnue mais constante quelle que soit la configuration k .

Les systèmes obtenus sont alors de la forme :

$$\mathcal{F}_k(M_k, \xi, \zeta_k, \zeta) \sim 0 \quad (5.3)$$

L'obtention d'un système global sur-contraint nécessite de générer plus d'équations de contraintes \mathcal{F}_k que de paramètres, non mesurés, dépendants de la configuration du robot lors de la prise de mesure k .

Ce type de variables apparaît assez souvent, et nous verrons que c'est le cas dans les étalonnages que nous avons effectués. Dans beaucoup d'études, ces variables sont re-calculées à chaque itération de l'algorithme d'identification présenté en 5.7. Le problème est que ce calcul dépend des paramètres à identifier par l'étalonnage³. Cette approche pose des problèmes de convergence et rend

3. Par exemple, prenons l'approche de l'étalonnage direct des robots parallèles empruntée aux robots séries. Elle est fondée sur la mesure par un appareil externe de la position et de l'orientation de l'organe terminal du robot. La méthode directe consiste à comparer la position/orientation mesurée avec celle déduite du modèle géométrique direct (MGD) du robot en fonction des paramètres à identifier et des informations fournies par les capteurs proprioceptifs. Ce critère mesure exactement la performance que l'on souhaite améliorer : la précision du robot. Cependant, la méthode est peu adoptée car elle nécessite de calculer un MGD, généralement difficile à déterminer pour les robots parallèles, aux solutions multiples réelles et complexes. En effet, le MGD, au cours des itérations du processus d'identification, dépend de paramètres géométriques erronés (en cours d'approximations successives) et de mesures

le critère à minimiser, voir eq. (5.5), numérique et non plus algébrique.

Nous avons montré dans [DEP⁺06, DE02, DE01] qu'il peut être intéressant d'éliminer algébriquement du système (5.3) ces variables ζ pour obtenir un système du type (5.2). Plus encore, ces méthodes algébriques nous permettent de jouer avec la redondance d'information pour diminuer le degré du système final. Nous avons montré dans [DE02] que les méthodes de type *boîte noire* pour estimer ces variables ζ_k fonctionnent très mal. Expérimentalement, nous avons plus récemment utilisé ces résultats, voir [JADG12], pour étalonner un robot à câbles.

Les manipulations mathématiques des équations obtenues ont néanmoins une limite. En effet, elle sont bâties sur l'hypothèse forte qu'une erreur de modèle $\epsilon_{\mathcal{F}}$ est négligeable face aux erreurs sur les paramètres à identifier ϵ_{ξ} . Cette hypothèse n'est pas forcément réalisée, ce qui a été montré expérimentalement dans [DACP06b].

Ces travaux ont réorienté mes recherches vers ce problème de prise en compte des erreurs de modèle qui est assez peu étudié dans le cadre de l'étalonnage. Il a été à l'origine des deux sujets de thèse que j'ai encadrés spécifiquement sur l'étalonnage des robots à structures parallèles.

Comme nous l'avons vu, le modèle d'étalonnage est souvent un modèle du robot simplifié par hypothèse pour correspondre à celui qui sera utilisé de façon effective par la commande, mais aussi parce que l'ensemble des paramètres ne sont pas forcément identifiables à partir des informations M_k . Cette question sera revue à la section 5.5.

Cette erreur de modèle est donc directement liée à son niveau de complexité considérée. Afin de considérer ce problème, nous avons proposé d'étudier deux approches décrites dans les sous-sections suivantes.

5.3.3 Simplification d'un modèle augmenté des équations d'étalonnage

Dans la thèse de Thibault Gayral, nous avons cherché à identifier un modèle complexe du robot à l'aide d'informations très riches⁴ permettant de mesurer le positionnement de l'ensemble des corps du robot. Puis, nous avons dégradé le modèle en vérifiant que les conséquences soient acceptables sur la précision du robot et l'identifiabilité des paramètres des modèles incrémentaux. Ainsi, deux sources additionnelles d'imprécisions statiques sont identifiées et modélisées : une provient de la déformation de la plate-forme mobile, l'autre résulte du comportement des articulations flexibles.

Cependant, dans l'espace, les informations riches fournies par la photogrammétrie ne sont pas disponibles : seules des informations indirectes de positionnement du robot sont fournies par le dispositif optique du robot. Elles sont

proprioceptives incertaines. Dans le cas de l'utilisation d'un MGD exact nous devons donc considérer le problème dans les complexes et choisir l'une des solutions possibles. Dans le cas d'une utilisation d'un MGD numérique approché, les erreurs de convergences du MGD sont à ajouter au problème ce qui rend les algorithmes instables. Cette méthode directe n'est pas utilisée en pratique au profit de la méthode inverse [ZYM98]. Cependant, dans bien des cas d'auto-étalonnage [PE00] où on utilise une mesure partielle de la position/orientation, nous retrouvons les fondements de cette méthode. Nous avons montré ce problème dans [JADG12] et qu'il est préférable d'identifier directement l'ensemble des inconnues.

4. Nous avons utilisé un système de photogrammétrie fournie par notre partenaire industriel Thales.

données au travers des coefficients des polynômes de Zernik qui rendent compte des défauts optiques résultant des erreurs de positionnement. Le challenge a été de proposer un modèle simplifié du robot qui considère des paramètres identifiables et qui permet des corrections de positionnement micrométriques. Ainsi les résultats de l'étalonnage riches en informations nous ont permis de proposer et de valider un modèle simple qui autorise l'auto-étalonnage à partir d'informations fournies par l'appareil optique embarqué sur le télescope.

Les modélisations, l'étalonnage des paramètres et les résultats expérimentaux sont disponibles dans [GDD13, GDB13].

5.3.4 Modèle simplifié mais épaissi des équations d'étalonnage

Dans la thèse de Julien Alexandre dit Sandretto [ADS13], nous avons cherché à garder la structure des modèles simples des robots à câbles directement exportés de la plate-forme de Gough tout en prenant en compte les incertitudes liées à la nature flexible des câbles en *épaississant* ces modèles. En effet, l'idée est d'utiliser une représentation par intervalles des variables du problème afin de borner un ensemble possible de modèles plus complexes, soumis aux incertitudes, dans une représentation par intervalles simple. Il a alors été nécessaire de développer les outils afin de caractériser une solution adaptée à ce type d'étalonnage par intervalles que nous verrons en section 5.4.3.

Mais avant cela, il faut répondre à la question de la validité d'un modèle simple mais épaissi versus un modèle complet. La thèse de J. Alexandre dit Sandretto montre pourquoi et comment obtenir un modèle simplifié, possiblement identifiable et compatible avec la commande pour les robots à câbles dans le cadre du projet CoGiRo. En effet, le **modèle complet** d'un câble prend en compte sa masse et son élasticité. Ainsi, s'il est tendu entre deux points, il décrit dans l'espace une chaînette dont les équations sont données par Irvine [IC74]. Le système obtenu est non-linéaire et non-algébrique, fonction des tensions appliquées en les points d'attaches et des propriétés physiques du câble. Pour les robots de faible taille (quand la longueur de câble déployée est faible), pour des câbles légers et peu élastiques, on peut négliger la masse et l'élasticité pour obtenir un modèle simplifié où le câble décrit une ligne droite entre les points d'attaches. On peut alors considérer un **modèle simple** si l'erreur entre la longueur déployée et la distance entre points d'attaches est faible, c'est à dire de l'ordre de grandeur de l'incertitude de mesure de la longueur de câble.

Pour étalonner le modèle complet (avec prise en compte de la masse et de l'élasticité du câble), il faut mesurer la tension dans les câbles. Or, pour les robots étudiés (CoGiRo, ReelAx8) cette information n'est pas facilement accessible. En effet, elle demande un capteur performant pour des gammes de mesures larges de la tension ; la technologie actuelle ne permet pas d'obtenir, pour des prix abordables, cette mesure avec un bruit suffisamment faible pour être utilisée de la cadre de l'étalonnage. Il est donc nécessaire de n'utiliser que des informations géométriques (mesures externes de positionnement de l'organe terminal et/ou mesures internes des longueurs des câbles) pour cet auto-étalonnage.

Nous avons proposé une stratégie de contournement de ce problème fondée sur la vérification de l'erreur de modélisation, c'est à dire l'erreur entre la longueur de câble du modèle simple vs modèle complet, voir [AdSDT14, ADSTDC14, STD13, ADSTDG13, ADSTD12]. En effet, les performances des

robots n'étant pas les mêmes sur tout l'espace de travail, en fonction des spécifications du manipulateur (dimensions, gamme de charge utile, variation des tensions dans les câbles), un algorithme d'analyse par intervalle permet de déterminer les positions du robot pour lesquelles l'hypothèse de simplification du modèle de câble est réaliste, voir [STD13, AdSDT14].

Les résultats ont montré que pour le robot ReelAx8 les hypothèses sont acceptables, mais que pour le robot CoGiRo, seul un sous-espace de travail est compatible avec les conditions de l'étalonnage, voir section 5.5.1.

Cet algorithme est ici développé pour l'étalonnage mais il peut être utilisé pour valider les hypothèses simplificatrices dans nombre de travaux de recherches.

Cette étude de l'espace de travail permet deux choses :

- D'obtenir une estimation des erreurs de modèle dues aux caractéristiques du câble pour un ensemble de tâche.

Ce dernier point est important car il illustre la notion de modèle épaissi que nous avons établie en section 3.2.1. En effet, on montre qu'au moyen d'une étude théorique, il est possible d'estimer cette erreur de modèle qui n'est pas mesurable directement. En utilisant une approche intervalle, nous pouvons établir une approximation de l'ensemble des incertitudes : sur les mesures, fondées sur une estimation fournie par le constructeur ou expérimentalement, mais aussi sur les modèles en utilisant l'approche précédente.

5.4 Méthode d'identification des paramètres

5.4.1 Approche classique aux moindres carrés non-linéaires

L'identification des paramètres du robot revient à fournir une solution au système d'équations non linéaires fourni par les modèles d'étalonnage du type des équations 5.2.

Afin d'identifier les paramètres du robot ξ , il est nécessaire d'obtenir un système au moins bien contraint, i.e. de positionner le robot dans K configurations avec $K \times \dim(\mathcal{F}_k) \geq \dim(\xi)$ où $\dim(\mathcal{F}_k)$ est la dimension du sous-système d'équations \mathcal{F}_k obtenues pour une configuration du robot k et $\dim(\xi)$ la dimension du vecteur des paramètres inconnus.

A cause des incertitudes, il ne sera généralement pas possible de déterminer une solution ponctuelle qui annule l'ensemble de ces équations. En effet, pour cela il faudrait que ces équations soient linéairement liées ce qui n'est pas possible, en pratique, à cause des erreurs sur leurs coefficients (mesures) et sur leurs structures (modèle), d'où la création de résidus.

L'approche classique consiste donc à caractériser une solution aux moindres carrés non-linéaires, c'est à dire une solution qui minimise la somme au carré des résidus.

Nous prenons $K \times \dim(\mathcal{F}_k) > \dim(\xi)$ afin d'obtenir un système **sur-contraint**, noté $\mathcal{F} = [\mathcal{F}_1^T, \dots, \mathcal{F}_k^T, \dots, \mathcal{F}_K^T]^T$ et nous caractériserons les solutions ξ_{ls} afin qu'elles minimisent les erreurs résiduelles de l'évaluation de $\mathcal{F}(M_k, \xi_{ls})$ pour $k = 1 \dots K$. La caractérisation utilisée est alors définie par un critère aux moindres carrés, tel que :

$$\text{Déterminer } \xi_{ls} \quad \text{tel que :} \quad (5.4)$$

$$\begin{aligned} \xi_{ls} &= \min_{\xi} \mathcal{C}(M, \xi) \\ \text{avec } \mathcal{C}(M, \xi) &= \frac{1}{2} \mathcal{F}(M, \xi)^T \mathcal{F}(M, \xi) \end{aligned} \quad (5.5)$$

Dans les sous-sections suivantes, nous proposerons d'autres types de caractérisations alternatives et complémentaires. Mais la minimisation du critère \mathcal{C} est à la base de la quasi-totalité des méthodes d'identification utilisées pour l'étalonnage.

Les équations \mathcal{F} étant généralement non linéaires, le problème 5.4 est résolu par une méthode itérative de type Newton. Partant d'une estimée initiale ξ^i à l'itération i , le but est de déterminer ξ^{i+1} telle que $\mathcal{C}(\xi^{i+1}) < \mathcal{C}(\xi^i)$.

Le développement au second ordre du critère \mathcal{C} nous donne l'approximation suivante, au voisinage d'une estimée ξ^i :

$$\mathcal{C}(\xi) = \mathcal{C}(\xi^i) + \mathcal{J}_{\xi^i}^T \mathcal{F}_{\xi^i}(\xi - \xi^i) + \frac{1}{2}(\xi - \xi^i)^T [\mathcal{J}_{\xi^i}^T \mathcal{J}_{\xi^i} + \mathcal{H}_{\xi^i}^T F_{\xi^i}](\xi - \xi^i) \quad (5.6)$$

avec

- la matrice jacobienne d'identification $\mathcal{J} = \frac{\partial F}{\partial \xi}$,
- et le tenseur \mathcal{H} représentant le hessien de \mathcal{F} avec
 - $\mathcal{H}^T F = \sum_{k=1}^K \mathcal{H}_k^T F_k$
 - $\mathcal{H}_k = \frac{\partial^2 F_k}{\partial \xi \partial \xi^T}$ pour chaque configuration k .

Une condition nécessaire pour déterminer les ξ tel que (5.4) nous est donnée par l'annulation de la dérivé première de (5.6) soit :

$$[\mathcal{J}_{\xi^i}^T \mathcal{J}_{\xi^i} + \mathcal{H}_{\xi^i}^T F_{\xi^i}] \Delta \xi = -\mathcal{J}_{\xi^i}^T \mathcal{F}_{\xi^i} \quad (5.7)$$

Ainsi, si le système linéaire 5.7 n'est pas singulier, sa résolution admet une solution unique $\Delta \xi$.

Le schéma itératif est donc donné par :

$$\text{Tant que} \quad \text{Critère}(F_{\xi_i}, \xi_i, i) \quad (5.8)$$

$$\xi^{i+1} \leftarrow \xi^i + \Delta \xi \quad (5.9)$$

$$\text{avec } \Delta \xi \text{ solution de (5.7)} \quad (5.10)$$

$$(5.11)$$

La condition d'arrêt est fonction de la valeur relative ou absolue de la valeur des inconnues ξ et des valeurs des fonctions à minimiser F_{ξ_i} ou du nombre d'itérations. Dans le cadre de l'étalonnage ce critère sera discuté à la section 5.5.1.

Nous avons vu qu'autour de la solution nous avons donc la relation suivante :

$$\mathcal{J}^T \mathcal{J} \Delta \xi = \mathcal{J}^T \Delta \mathcal{F} \quad (5.12)$$

Où $\mathcal{J}^T \mathcal{J}$ est dite matrice d'information. Si cette matrice n'est pas singulière, il est possible d'obtenir une solution aux moindres carrés de ce système linéaire est par :

$$\Delta\xi = \mathcal{J}^+ \Delta\mathcal{F} \quad (5.13)$$

Où $\mathcal{J}^+ = (\mathcal{J}^T \mathcal{J})^{-1} \mathcal{J}^T$ est la pseudoinverse de \mathcal{J} et $\Delta\mathcal{F}$ l'erreur de l'évaluation des équations d'étalonnage.

Mais nous préférons voir ce problème directement sur le système linéarisé sur-constraint :

$$\mathcal{J} \Delta\xi = \Delta\mathcal{F} \quad (5.14)$$

Pour déterminer une solution à ce système, on utilisera une décomposition en valeurs singulières de $\mathcal{J} = U\Sigma V^T$, où $U_{K \times K}$ et $V_{N \times N}$ sont des matrices orthogonales avec K la dimension de \mathcal{F} et N le nombre de paramètres considérés. $\Sigma_{K \times N}$ contient dans ses coefficients diagonaux supérieurs les valeurs singulières σ_n de $\mathcal{J}_{K \times N}$.

Nous aurons donc :

$$\Delta\xi = V\Sigma^+ U^T \Delta\mathcal{F} \quad (5.15)$$

avec Σ^+ la transposée de Σ où toute valeur singulière $\sigma_n > \epsilon_\Sigma$ est remplacée par son inverse $1/\sigma_n$. Cette projection nous permettra d'obtenir une solution *aux moindres carrés* mathématique équivalente à la pseudo-inverse $\mathcal{J}^+ = (\mathcal{J}^T \mathcal{J})^{-1} \mathcal{J}^T$ mais numériquement plus stable à calculer. Le critère ϵ_Σ est très important car il conditionne le nombre de paramètres que nous pourrions identifier. Il est généralement déterminé, de façon automatique, en fonction de la taille, de la norme de \mathcal{J} et de la précision de l'arithmétique utilisée (valeur du plus petit flottant représentable) : cependant ceci ne prend pas en compte les incertitudes associées aux variables du problème, nous verrons ceci dans les contributions de la section 5.5.

La prise en compte des incertitudes est possible, si la distribution des erreurs de mesure est connue. Ainsi, les critères à minimiser peuvent être pondérés par un a priori sur les incertitudes [WHA95].

Nous avons testé et comparé plusieurs approches notamment dans [DACP06a] pour une plate-forme de Gough ou [JADG12] pour les robots à câble :

- *Weighted Least-Square* : le critère est ici pondéré par un a priori sur l'incertitude de mesure. Ainsi le critère devient $\mathcal{C} = \mathcal{F}^T \cdot \Sigma_F \cdot \mathcal{F}$ où Σ_F est approximé par la matrice de covariance Σ_M associée aux mesures M avec $\Sigma_F = J_M^T \cdot \Sigma_M \cdot J_M$ et $J_M = \frac{\partial \mathcal{F}}{\partial M}$. Nous avons aussi utilisé cette approche dans [DACP06a]
- *Orthogonal Distance Regression* : nous chercherons ici à minimiser aussi les erreurs de mesures en ajoutant au critère précédent \mathcal{C} une estimation des erreurs de mesures, voir [BBS87]
- *le χ^2* : Nous cherchons ici à prendre en compte un a-priori sur les paramètres à identifier en ajoutant au critère précédent \mathcal{C} une estimation de l'erreur sur les paramètres, voir [PE00]

Nous avons obtenu de bons résultats dans [JADG12] en utilisant ce dernier critère χ^2 pour l'étalonnage d'un robot à câbles au regard de sa précision en orientation. En fait, pour cet étalonnage, l'espace de travail en orientation du robot est faible, ce qui entraîne une mauvaise identifiabilité de certains paramètres (voir section 5.5.1). La méthode fondée sur l'utilisation de χ^2 permet de

limiter une certaine dérive dans l'estimation d'un sous-ensemble de paramètres à identifier.

Il existe de nombreuses possibilités afin de prendre en compte les incertitudes mais la plupart sont fondées sur des hypothèses fortes quant à leurs distributions qui ne reflètent pas forcément la réalité des incertitudes (par exemple, tolérance mécanique). Ainsi, il est difficile de conclure sur l'intérêt général de telle ou telle méthode tant les modèles de bruit et les a priori sont divers et plus ou moins difficiles à obtenir voir, inappropriés. De plus, l'erreur de modèle telle que nous l'avons définie précédemment n'est pas considérée.

La très grande majorité des études utilise une solution aux moindres carrés telle que (5.4) afin de caractériser une solution à l'étalonnage. Nous l'avons aussi utilisée pour étalonner de nombreux robots (voir [JADG12] pour une comparaison des différentes approches pour l'étalonnage des robots à câbles). Cependant, nous avons proposé d'autres types de caractérisation des solutions de l'étalonnage. En effet, il peut être difficile d'interpréter les résultats obtenus notamment pour mieux comprendre comment ils sont influencés par les incertitudes. Nous avons donc proposé des solutions alternatives.

- La première utilise des méthodes algébriques [DE04] pour résoudre des systèmes sur-contraints. Dans cet article nous avons montré qu'il est possible de jouer avec la redondance d'informations pour diminuer le degré des équations d'étalonnage. De plus, cette solution permet de s'affranchir d'une estimée initiale sur les paramètres, de détecter d'éventuelles solutions multiples (rares mais possibles) et de déterminer une certaine distribution des erreurs sur les solutions dues aux formes symboliques des équations considérées.
- La seconde est adaptée à l'extension par intervalles des équations d'étalonnage. Cette approche considère l'ensemble des variables (mesures, paramètres à identifier...) comme des intervalles de variations bornées. Ceci permet d'unifier les différents types d'incertitude dans une représentation commune mais pessimiste. Elle permet aussi la certification des solutions, les erreurs étant considérées par l'arithmétique. Les bases théoriques ont été développées dans [DPN04] puis expérimentées et discutées dans [DACP06b] pour un étalonnage utilisant un système de vision externe, en collaboration avec le LASMEA. Nous avons obtenu des résultats très intéressants qui ont souligné la nécessité de prendre en compte les erreurs de modèles. Ces travaux prometteurs ont fait l'objet d'une suite, au travers des contributions développées dans la thèse de Julien Alexandre dit Sandretto. Il a, en effet, proposé d'autres types de caractérisation des solutions d'étalonnage par intervalles et les a expérimentées dans le cadre de CoGiRo afin d'identifier les paramètres de plusieurs prototypes de robots à câbles. Ces méthodes sont présentées et comparées à l'approche aux moindres-carrés dans [ADSTDC14, ADSTDG13].

5.4.2 Méthodes algébriques

Nous avons étudié l'utilisation des méthodes d'élimination algébrique pour un étalonnage robuste des robots parallèles, appliquées à la plate-forme de Gough. Précédemment, nous avons vu que ces techniques permettent d'éliminer les variables des modèles d'étalonnages qui sont à considérer dans les modèles mais dont les valeurs sont inconnues (non-mesurées) et qui ne fournissent pas

d'information sur le modèle du robot, voir section 5.3.2. Dans cette section nous utilisons ces techniques dans le cadre de la résolution des systèmes d'étalonnage pour l'identification des paramètres.

Ainsi, dans [DE04] deux approches, à savoir l'élimination de variables algébriques et la linéarisation de monômes, sont comparées à une approche classique des moindres carrés. Le principe revient à utiliser des techniques d'élimination algébrique de type résultant creux [EC95] pour ramener le problème de l'identification des paramètres du robot à un problème de détermination de valeurs/vecteurs propres numériquement robuste.

Peu de travaux sur l'étalonnage des robots parallèles explorent cette approche algébrique. Innocenti [Inn95] propose une résolution des équations implicites (voir [ZYM98]) de la plate-forme Gough. Il utilise une configuration supplémentaire pour simplifier les équations algébriques puis applique une élimination dyalytique. Vischer [Vis96] présente une méthode de linéarisation monômiale (qu'il appelle semi-paramétrique) qui consiste à linéariser les équations d'étalonnage en les monômes des inconnues. Pour cela, il se sert de la redondance d'informations (plus d'équations disponibles que d'inconnues) pour constituer un système d'équations plus grand que le nombre total des monômes à considérer. Il applique cette technique pour les robots Argos et Delta, mais il souligne le problème de projection des résultats donnés dans la base de monômes sur la base des inconnues, les paramètres géométriques.

La contribution est, ici, de proposer une alternative aux méthodes des moindres carrés qui souffrent de plusieurs inconvénients : une nécessaire estimation initiale des paramètres qui peuvent influencer le résultat mais qui peuvent se révéler difficile à interpréter. En effet, ces méthodes fournissent une solution qui minimise un critère mais qui ne donnent pas d'idée de comment les erreurs de mesures se répartissent dans l'espace des solutions de chaque système bien-contraint pris parmi le système sur-contraint. Cette dispersion donne des éléments pour analyser les solutions possibles de l'étalonnage et ainsi participer à sa validation. En ramenant le problème de l'étalonnage non-linéaire en un problème d'algèbre linéaire dont la résolution est rapide, les méthodes symboliques nous permettent d'accéder à ce type d'information.

Pour cela, on détermine le résultant de $m + 1$ polynômes en m variables avec des coefficients génériques. Ce résultant est un seul polynôme, donné sous la forme d'un déterminant d'une matrice, en les coefficients du système d'origine qui s'annulent si, et seulement si, il existe un zéro non trivial commun pour les $m + 1$ polynômes. Nous nous concentrons sur ces résultants pour deux raisons. D'abord, leur construction ne souffre pas de l'instabilité due aux perturbations des coefficients : la matrice résultante peut être déterminée hors ligne pour les coefficients génériques, instanciés pour les données d'entrées spécifiques. Ainsi, le deuxième avantage est que la même matrice permet le calcul de la solution de problèmes répétés lorsque seules les valeurs des coefficients spécifiques changent. Nous utilisons des techniques d'élimination creuse (ou torique) et de résultants creux : ils généralisent la théorie classique en exploitant la structure des équations. En particulier, la borne BKK donne des informations sur le nombre de racines communes du système de $n + 1$ polynômes qui est généralement meilleur que le nombre fourni par la borne classique de Bezout.

Le cas bien contraint

Dans le cadre de l'étalonnage de robot, nous considérons, dans un premier temps, un système bien contraint (5.2), c'est à dire construit à partir d'un nombre minimum d'observations $n = 1 \dots N$ de l'état du robot afin de fournir autant d'équations $\mathcal{F}_{1,\dots,N}(M_{1,\dots,N}, \Xi)$ que d'inconnues $\Xi = [\xi_1, \dots, \xi_K]$ à identifier tel que $K = N$. Les coefficients M_n du système sont symboliques dans un premier temps pour fournir une forme générique du résultant puis instanciée par les mesures pratiquées sur le robot.

Il est possible d'appliquer un méthode de résultant en 2 étapes :

- 1 Choisir une variable $\lambda = \xi_k$, parmi les Ξ à cacher dans l'espace des coefficients (M_n) ; les autres variables seront éliminées.
- 2 Multipliez chaque équation par une liste des monômes $\xi_1^{c_{1,j}}, \dots, \xi_{k-1}^{c_{k-1,j}}, \xi_{k+1}^{c_{k+1,j}}, \dots, \xi_K^{c_{K,j}}$ dans les variables éliminées avec $c_{i,j}$ leurs degrés en les monômes considérés.

On obtient ainsi un système d'équations linéaires en les monômes des variables éliminées χ mis sous forme matricielle :

$$\mathcal{M}(\lambda)\chi = 0 \quad (5.16)$$

Ce principe est également connu comme l'élimination de dialytique, ou par linéarisation. Lorsque le système est bien-contraint, le résultante est soit le déterminant de cette matrice \mathcal{M} soit un diviseur non trivial du déterminant. Pour produire ces matrices de résultant creux, lors de la 2ème étape de la procédure ci-dessus, nous utilisons l'algorithme proposé dans [EC95]. Ce choix d'une méthode de type Sylvester nous permet un bon contrôle des dimensions de la matrice et peut-être appliqué au cas des systèmes sur-contraints. Si $\det(\mathcal{M})(\lambda) = 0$ le système $\mathcal{F}(M, \Xi) = 0$ admet une solution commune. La résolution de l'équation 5.16 en $\chi = \chi_s$ pour différentes valeurs de $\lambda = \lambda_s$ se réduit à un problème de valeurs propres généralisées pour une instance des coefficients $M_{1,\dots,N}$.

Afin d'identifier les inconnues $\Xi = \Xi_s$ à partir du vecteur des monômes χ_s déterminé précédemment, nous utilisons une projection orthogonale du type $\min_{\xi=\xi_s} \|\chi(\Xi) - \chi_s\|^2$. Pour cela, nous avons besoin d'une estimation initiale des ξ qui est, en fait, donnée par les premières composantes de monômes χ .

Nous appliquons cette méthode dans le cas de l'étalonnage classique de la plateforme de Gough avec mesures externes de la position et de l'orientation de l'organe terminal pour l'identification de 42 paramètres (7 par jambes) géométriques. Ceci nécessite de positionner la plate-forme dans au minimum 7 configurations de mesures.

Ainsi le cas bien contraint est obtenu pour $N = 7$. La borne BKK qui fournit une estimation du nombre de solutions complexes de $\mathcal{F}(M, \Xi) = 0$ est égale à 40, ce qui est optimal. La dimension optimale de la matrice résultante serait 130. Notre logiciel, basé sur [EC95], donne un matrice résultante \mathcal{M} de taille 205×205 contenant une variable cachée (composante en x du point d'attache de la jambe avec la base). Le système 5.16 est donc obtenu symboliquement, son instanciation des coefficients par des mesures permet de résoudre numériquement le cas bien contraint.

Gestion du cas sur-contraint

Dans la pratique, l'étalonnage est un problème sur-contraint. Les informations supplémentaires, fournissant un grand nombre d'équations redondantes mais perturbées par les erreurs de mesures, peuvent être utilisées de deux façons. La première est de simplifier la formulation algébrique en diminuant la taille de la matrice d'élimination. La deuxième façon est de produire un grand nombre de systèmes bien contraints, comme expliqué dans la section précédente, et d'utiliser une approche statistique d'observation des solutions obtenues.

La technique la plus simple consiste à linéariser les équations du système de base en les monômes. Dans le cas étudié, on obtient 18 monômes (en considérant, le monôme de degré nul en chacune des inconnues et un terme rassemblant les monômes quadratiques). Si le robot est placé dans au moins 17 configurations de mesures, nous pouvons donc construire un système linéaire (sur-contraint si $N > 17$) en les monômes puis résoudre classiquement le système en utilisant, par exemple, une technique de décomposition SVD. La projection de l'espace des monômes vers celui des inconnues se fait comme précédemment par projection orthogonale. Une autre solution consiste à utiliser des configurations de mesures supplémentaires (donc des équations en plus) pour diminuer la taille de la matrice résultante. Innocenti dans [Inn95] montre qu'avec une équation supplémentaire on peut trouver une matrice de taille 60×60 . Nous montrons qu'avec 14 configurations de mesure nous obtenons une matrice de taille 13×13 , ce qui est la plus petite matrice de résultant que nous ayons pu produire.

En fonction du nombre total de configurations de mesures produites pour étalonner le robot fournissant N équations et du nombre d'équations U nécessaires pour construire les résultants, nous pouvons construire $\frac{N!}{U!(N-U)!}$ ensemble de solutions réelles ou complexes pour chacun des sous-systèmes d'équations étudiés. Dans un cas générique, seule une solution est commune pour des sous-systèmes et doit correspondre à la solution physique de l'étalonnage. Par exemple, si nous plaçons le robot dans 20 configurations de mesures, nous obtenons 20 équations d'étalonnage ; hors, 7 équations sont nécessaires pour construire la matrice de résultant 205×205 dans le cas bien-contraint : nous pouvons donc construire $Nb_t = 77520$ systèmes d'équations bien contraints à partir des 20 équations disponibles. Chacun de ces systèmes fournit jusqu'à 40 solutions réelles ou complexes mais une seule est commune à l'ensemble des nb_t systèmes. Dans [DE04], nous construisons et résolvons 10000 de ces systèmes. Le temps de calcul nécessaire est compatible avec le problème de l'étalonnage : il peut se faire off-line, les systèmes sont pré-calculés symboliquement, la résolution numérique par une méthode de vecteur propre généralisé est rapide. Après avoir sélectionné l'une des solutions possibles communes à chacun d'entre eux, nous utilisons les lois des grands nombres afin de visualiser la dispersion des solutions. La figure 5.5 montre ce que l'on peut obtenir en l'un des paramètres (coordonnée en x de la coordonnée du premier point d'attache de la plateforme main gauche de Inria).

Dans [DE04] nous reproduisons ce type de résultats pour différentes tailles de matrices de résultant, pour des systèmes linéarisés en les monômes et nous les comparons à une approche classique aux moindres carrés.

Nous contribuons ici en proposant une méthode algébrique puissante et précise pour la calibration des robots qui, en outre, fournit un indice de qualité associé aux solutions. Cette approche offre une alternative aux moindres carrés classiques et ne nécessite aucune hypothèse sur la répartition du bruit de

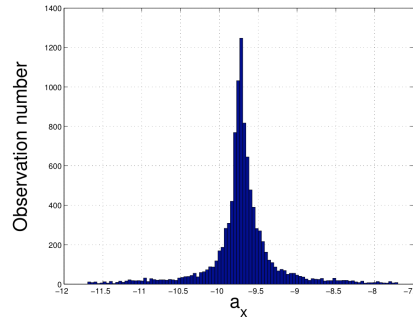


FIGURE 5.5 – 10000 observations d’un paramètre identifié

mesure. Les résultats montrent que la robustesse à ces erreurs est supérieure aux méthodes classiques lorsque le nombre de mesures est relativement faible et ne nécessite aucune estimation initiale de la solution. Dans le cas d’un grand nombre de mesures disponibles, il n’est pas évident que la méthode proposée soit préférable en terme d’erreurs sur l’identification. Cependant, il faut noter que d’autres critères de qualité de la méthode peuvent être pertinents, tels que la difficulté d’estimation d’une valeur initiale pour les paramètres d’étalonnage et des hypothèses pour la distribution de bruit : dans ces cas l’alternative reste intéressante.

Notre approche repose sur certaines notions avancées d’algèbre et d’outils spécialisés, qui peuvent constituer un obstacle pour sa large utilisation. En outre, il n’est pas clair que l’approche puisse être appliquée lorsque le nombre d’inconnues (7 dans le cas étudié) augmente.

5.4.3 Méthodes par intervalles

L’identification des paramètres du robot revient à fournir une solution aux systèmes d’équations non-linéaires et sur-contraints. À cause des incertitudes, il ne sera généralement pas possible de déterminer une solution unique qui annule l’ensemble de ces équations ou résidus. L’approche classique consiste donc à caractériser une solution aux moindres carrés non-linéaire, c’est à dire qui minimise la somme des résidus. La prise en compte des incertitudes est possible, si la distribution de l’erreur de mesure est connue. Ainsi, il est possible de pondérer les critères à minimiser par un a priori sur les incertitudes. Dans [WHA95], l’estimation d’indice de qualité pour chaque solution est proposée dans le cadre de l’étalonnage des robots. Cependant, le modèle classique de distribution gaussienne n’est pas réaliste pour certaines incertitudes comme les tolérances mécaniques pour lesquelles une distribution uniforme est mieux appropriée.

J’ai alors proposé d’étudier une autre caractérisation des solutions : l’étalonnage par intervalles. Le principe repose sur une approximation au pire cas des incertitudes des variables considérées. S’il est difficile (voir faux) d’associer une distribution particulière aux mesures, il est plus simple et réaliste de les borner et d’utiliser une représentation de leur état par un vecteur intervalle $\mathcal{M}_k \in \mathbf{M}_k = [\mathcal{M}_k - \epsilon, \mathcal{M}_k + \epsilon]$. Cette approche permet d’unifier dans un

même formalisme les incertitudes des autres types de variables qui interviennent dans le problème de l'étalonnage : les tolérances mécaniques de fabrications et d'assemblage qui sont associées aux contraintes mécaniques ou aux paramètres géométriques et qui, elles, sont naturellement fournies sous la forme d'intervalles bornés. Les équations d'étalonnages, étendues par intervalles grâce à une arithmétique adaptée, sont notées \mathbf{F}_k et représentent l'ensemble des contraintes \mathcal{F}_k instanciées par chaque valeur de \mathcal{M}_k pris dans \mathbf{M}_k .

J'ai tout d'abord proposé dans [DPN04] une caractérisation des solutions par intervalles (un ensemble compact) qui garantit d'inclure les valeurs des paramètres qui satisfèrent les équations d'étalonnages étendues aux intervalles ; leurs coefficients intervalles sont donnés par des bornes sur la variation des mesures. Cette caractérisation peut être donnée sous la forme :

$$C_1(\mathbf{M}) = \{\xi | \mathcal{F}(\xi, \mathcal{M}_k) = 0 \text{ avec } \mathcal{M}_k \in \mathbf{M}_k, k = 1, \dots, K\} \quad (5.17)$$

Attention, cette caractérisation ne donne pas de solution *aux moindres-carrés* mais fournit une approximation, si elle existe, d'un ensemble de paramètres qui annulent des équations prises dans le modèle épaissi d'étalonnage.

Fournir une approximation (enveloppe externe) par intervalles à cette caractérisation a nécessité de proposer une nouvelle méthode basée sur une version de l'algorithme de Newton par intervalles adaptée au cas sur-contraint. Ceci revient à résoudre, à chaque itération, un problème linéaire sur-contraint par intervalles. L'originalité est ici d'inclure l'ensemble des solutions dans une enveloppe convexe de contraintes linéaires fournies par différentes techniques (Beaumont [Bea00], Krawczyk [KN86]). Un algorithme de programmation linéaire permet alors de fournir un encadrement par intervalle des paramètres ξ du robot.

La validation expérimentale de cette méthode a été particulièrement intéressante et a conditionné mes travaux sur l'étalonnage à plus long terme [DACP06a, DAP05]. En effet, nous avons mis en place l'étalonnage d'une plateforme de Gough (DeltaLab, Table de Stewart) avec Nicolas Andreff, alors au Lasmea. Nous avons utilisé un système de vision par ordinateur pour mesurer la position et l'orientation de l'organe terminal du robot (étalonnage avec mesure externe). Ce choix nous a permis d'automatiser la procédure de mesure et de générer de très fortes quantités d'informations (14784 images prises et traitées) afin de déterminer très précisément des bornes sur les incertitudes de mesures. L'étalonnage par intervalles nous a permis de montrer que les erreurs de modèle de ce robot (d'une qualité de fabrication mécanique médiocre) n'étaient pas négligeables. Or cette erreur de modèle n'est pas considérée dans la littérature. Nous avons vu au paragraphe 5.1.3 qui nous y avons attaché une importance particulière dans les thèses que j'ai encadrées. Concrètement, nous avons montré qu'il n'existait pas de solution telle que l'ensemble des équations *épaissies* par une variation possible des mesures dans un intervalle s'annule. La première explication est alors que les bornes sur les variations des mesures sont sous-estimées : ceci est peu probable car la forte quantité et la redondance des mesures nous ont poussé à croire à la qualité de l'estimation des bornes utilisées. Nous avons donc privilégié une seconde explication due à la simplification trop importante des équations de fermeture de la plateforme de Gough qui considère les segments sur robots comme une chaîne cinématique de type *SS*. Ainsi, en épaississant successivement, notant l'itération par j , les équations d'étalonnage, c'est à dire en considérant $\mathcal{F}(\xi, \mathcal{M}_k) = [-\epsilon_j, +\epsilon_j]$ nous voyons apparaître un

ensemble de solutions possibles, proches de celles trouvées par une méthode d'étalonnage aux moindres carrés.

J'ai alors orienté la thèse de Julien Alexandre dit Sandretto afin de proposer d'autres types de caractérisations par intervalles des solutions d'une identification de paramètres.

$$\mathcal{C}_2(\mathbf{M}) = \{\xi \mid \min_{\xi} \sum_{k=1}^K \mathcal{F}(\xi, \mathcal{M}_k)^t \mathcal{F}(\xi, \mathcal{M}_k) \text{ avec } \mathcal{M}_k \in \mathbf{M}_k, k = 1, \dots, K\} \quad (5.18)$$

Là encore une adaptation de l'algorithme de Newton par intervalles a été nécessaire. Comme précédemment, une extension de Taylor permet de considérer itérativement un système sur-contraint linéaire par intervalles. Mais, afin de fournir une solution aux moindres-carrés par intervalles, le système est mis sous une forme augmentée de Bjork [BG67].

Le système linéaire intervalle à résoudre devient alors :

$$\Sigma_{\exists, \exists}^*(\mathcal{J}, \Delta \mathcal{F}) = \{\Delta \xi, r \mid \exists \mathcal{J} \in \mathcal{J}, \Delta \mathcal{F} \in \Delta \mathcal{F}, \begin{pmatrix} Id & \mathcal{J} \\ \mathcal{J}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} r \\ \Delta \xi \end{pmatrix} = \begin{pmatrix} \Delta \mathcal{F} \\ 0 \end{pmatrix}\} \quad (5.19)$$

Nous avons alors montré comment obtenir une enveloppe externe (toutes les solutions sont contenues) et une approximation interne (tous les points de l'ensemble sont solutions) qui bornent l'ensemble \mathcal{C}_2 . L'interprétation du reste r en la solution est intéressante. En effet, si $0 \in \mathbf{r}$ alors il existe bien une solution dans la solution intervalle ξ qui annule l'ensemble des équations d'étalonnage pour au moins une valeur des mesures prises dans \mathbf{M}_k . Dans le cas contraire, il donne une indication sur l'erreur de modèle que nous faisons si l'incertitude sur les mesures a été correctement estimée.

Ces solutions ont été validées expérimentalement sur le robot CoGiro du Lirmm. Elles ont permis d'obtenir l'identification des paramètres de façon certifiée.

Il reste une question importante qui n'a pas été soulevée pour le moment. Dans nombre de méthodes où l'information est redondante voir très redondante, il faut gérer le problème des mesures dites aberrantes. Ces mesures fausses, comme dans le cas d'une approche classique aux moindres carrés, peuvent affecter la solution, c'est à dire contribuer de façon trop importante à s'éloigner de la solution réelle. Il convient alors de ne pas prendre en compte ces mesures.

Dans la thèse de Julien Alexandre dit Sandretto [ADS13] une méthode, la *q-intersection*, utilisant la programmation par contrainte est proposée. Elle est appliquée au robot à câble de Cogiro. L'idée est de chercher une solution qui annule le plus possible d'équations d'étalonnage. S'il existe une solution qui annule l'ensemble, ceci revient à la première méthode présentée dans cette section ou si le reste de l'approche aux moindres carrés par intervalle contient zéro. Dans le cas où il existe des mesures aberrantes, plusieurs équations ne peuvent s'annuler en la solution réelle : au cours de l'identification, il n'est alors pas possible de trouver une solution commune. L'algorithme propose alors une stratégie qui permet de trouver un sous-ensemble de contraintes qui possède une solution commune et de maximiser ce nombre. Les tests que nous avons pratiqués montrent que nous avons pu de façon automatique détecter des mesures

aberrantes lors de l'étalonnage du robot à câbles.

L'algorithme peut néanmoins être amélioré en fusionnant les trois approches par intervalles proposées dans cette section.

5.5 Conditions d'étalonnage et amélioration de l'identification des paramètres

Nous venons de voir comment identifier les paramètres d'un robot. Mais, il est nécessaire de revenir sur un certain nombre de conditions qui permettent ce processus.

La principale condition d'étalonnage, classiquement étudiée, porte sur l'identifiabilité des paramètres [SK08] ; plus précisément, que tous les paramètres considérés doivent être identifiables indépendamment, c'est-à-dire que leurs valeurs ne dépendent pas de la valeur des autres.

Au début des années 2000, de nombreuses contributions sont apparues sur de nouvelles façons d'obtenir une information redondante et donc de générer des équations d'étalonnage. Mon travail de relecture m'a amené à remarquer que l'essentiel des erreurs dans de nombreux articles provenait d'une non-vérification de l'identifiabilité. Ces questions sont maintenant très bien décrites dans le chapitre Handbook of robotics [SK08] où trois types de problèmes d'identifiabilité sont considérés : la non-identifiabilité, l'identifiabilité liée, et la faible identifiabilité. Cependant, on peut voir ce problème de façon plus pragmatique.

Le problème de l'identifiabilité revient à vérifier que la matrice d'identifiabilité $\mathcal{J}(M_{1,\dots,K}, \xi) = \frac{\partial F(M_{1,\dots,K})}{\partial \xi}$, voir équation 5.6, est de rang égale au nombre de paramètres considérés, $\text{rang}(\mathcal{J}) = \dim(\xi)$. Une condition suffisante est que le déterminant de la matrice d'information $\mathcal{J}^t \mathcal{J}$ soit différent de zéro. Nous remarquons que les coefficients de la matrice \mathcal{J} sont dépendants des paramètres ξ mais aussi de mesures $M_{1,\dots,K}$. Ainsi :

- les propriétés formelles de cette matrice dépendent de la structure des coefficients, c'est à dire de la forme symbolique des équations d'étalonnage.
- les propriétés numériques de la matrice dépendent des mesures M_k et donc du choix des configurations de mesures dans lesquelles le robot est placé, afin d'instancier M_k .

La chute du rang de la matrice peut donc provenir de plusieurs problèmes.

Considérons, dans un premier temps, les coefficients de la matrice \mathcal{J} comme symboliques : les mesures $M_{1,\dots,K}$ et les paramètres à identifier ξ ne sont alors pas instanciés. Il se peut que le déterminant de $\mathcal{J}^t \mathcal{J}$ soit formellement nul. Dans ce cas, les colonnes de \mathcal{J} peuvent soit être nulles ou alors plusieurs colonnes ou lignes de la jacobienne sont linéairement liées. Ceci peut arriver lorsque :

- un certain nombre de paramètres ξ_i n'apparaît tout simplement pas dans les équations,
- le nombre de configurations de mesure (c'est-à-dire le nombre d'équations considérées) n'est pas suffisant,
- les références (repère, zéro capteur) ne sont pas physiquement définies, par exemple dans le cas d'un auto-étalonnage,
- le nombre de paramètres n'est pas minimal pour représenter le processus géométrique ou physique considéré,

- les mesures sont symboliquement contraintes liant ainsi les paramètres M_k , par exemple pour un étalonnage imposant des contraintes mécaniques sur la structure du robot.

Il est assez facile de vérifier ces cas par une simulation de la méthode d'étalonnage, sans considérer d'incertitude. Il est cependant rarement possible de calculer formellement de déterminant de $\mathcal{J}^t \mathcal{J}$.

Dans un second temps, la vérification doit aussi se faire avec les mesures réelles pratiquées pour étalonner le robot. En effet, le calcul d'un rang se fait généralement numériquement en déterminant le nombre de valeurs singulières de \mathcal{J} différentes de zéro au travers d'une SVD.

Cependant, il est clair que le calcul est difficile en pratique car le seuil de nullité est lié à la taille de la matrice, à la normalisation utilisée, à la précision des calculs il peut être ambigu en pratique : à partir de quelle valeur peut-on considérer la valeur comme nulle, ou trop faible ?

Ces notions d'identifiabilité ou de faible identifiabilité sont discutées dans les travaux que j'ai pu diriger. Elles s'illustrent notamment dans [JADG12], où nous présentons un étalonnage d'un robot à câbles dans un faible espace de travail en orientation. En effet, les contraintes géométriques et statiques sur l'espace de travail de ce prototype ne permettent pas de faire varier sur de grandes amplitudes les orientations du mobile. Or, dans le cadre de l'étalonnage classique l'inverse est utilisé, si les orientations du mobile sont constantes alors certains paramètres ne sont pas identifiables indépendamment. Mais le problème est que ces orientations ne sont pas de grande ampleur : l'ambiguïté doit être tranchée.

Si la chute de rang est clairement avérée, il existe des solutions pour fixer les liens entre les paramètres, voir [SK08]. En utilisant des simulations de mesures suffisamment génériques, ce qui est conseillé dans un premier temps, puis utilisant des valeurs réelles on pourra utiliser une méthode de décomposition QR de la matrice \mathcal{J} comme le propose Besnard et Khalil dans [BK01]. Elle nous permet de vérifier l'identifiabilité et éventuellement de détecter les contraintes linéaires entre les paramètres ξ . Il suffira alors de ne pas considérer certains paramètres ou de les fixer à une valeur constante.

Les cas ambigus sont plus difficile à détecter et il nous a fallu proposer une méthodologie stricte. Nous avons alors travaillé avec Thibault Gayral afin d'affiner les conditions d'étalonnage afin de montrer qu'il était possible de fixer des seuils numériques pour lesquels nous pouvons associer des significations physiques que nous introduirons en sous-section 5.5.1.

Comme nous venons de le voir, le manque de généralité des informations peut amener à revoir la paramétrisation du problème pour rendre ces paramètres identifiables. Nous avons introduit deux critères : le produit des valeurs singulières de la matrice d'identification (ou le déterminant de la matrice d'informations) ainsi que la (ou les) plus petite(s) valeur(s) singulière(s). Vérifier l'identifiabilité revient à vérifier que ces critères sont non-nuls (ou trop proches d'un zéro), mais nous pouvons aller plus loin. En effet, nous pouvons utiliser les positionnements du robot pour chaque configuration de mesure pour améliorer (augmenter) ces critères. Mais d'autres critères additionnels sur la robustesse aux incertitudes peuvent être introduits, voir par exemple dans [SH08]. Nous avons proposé un algorithme dans [Dan02, DPM05] qui permet d'optimiser ces critères : ceci sera présenté dans la sous-section 5.5.2.

5.5.1 Conditions d'étalonnages

Dans [GD14], nous proposons de définir un certain nombre de conditions nécessaires et/ou suffisantes afin de garantir une bonne qualité de l'étalonnage.

Les modèles géométriques de robot nous donnent des relations qui lient linéairement les incertitudes sur la position/orientation de l'organe terminal, sur les variables articulaires et sur les paramètres géométriques. En utilisant ces relations, nous montrons comment :

- déterminer une précision nécessaire sur les paramètres de l'étalonnage afin de garantir la précision souhaitée pour le robot,
- en déduire la précision nécessaire sur les mesures pratiquées pour étalonner le robot afin de garantir la condition précédente,
- donner une condition d'arrêt pour les algorithmes d'identification itératifs de type Newton, un seuil sous lequel la précision sur les paramètres est atteinte. Noter que ce critère a alors une véritable signification physique car il est lié à une précision souhaitée pour le robot,
- normaliser les jacobiniennes d'identification afin de donner des seuils simples qui garantissent l'identifiabilité des paramètres fondés sur des objectifs physiques de précision et non plus seulement mathématiques.

L'influence des erreurs de modèles sur l'identifiabilité des paramètres est aussi à analyser car elle a des conséquences sur les choix des paramètres à garder ou négliger dans les modèles, [GDD13, GDB13].

Comme nous l'avons vu dans la section conception, les dimensions d'un manipulateur sont calculées afin de pouvoir permettre au robot de réaliser des performances souhaitées. Ainsi, après fabrication et assemblage, les erreurs de son modèle vont dégrader ses performances. Dans [GD14], nous proposons de définir un certain nombre de conditions nécessaires et/ou suffisantes afin de garantir une bonne qualité de l'étalonnage.

Les modèles géométriques de robots nous donnent des relations qui lient linéairement les incertitudes sur la position/orientation de l'organe terminal avec les variables articulaires et les paramètres géométriques. En utilisant ces relations, nous montrons comment :

- **Déterminer une précision nécessaire sur les paramètres de l'étalonnage afin de garantir la précision souhaitée pour le robot.** La précision d'un manipulateur est fonction de la qualité de ses mesures proprioceptives mais aussi de son modèle. Nous avons vu en section 4.3.2 qu'il était possible de garantir la précision sur un espace de travail en fonction des incertitudes connues sur les variables articulaires. Cet algorithme est robuste par rapport aux incertitudes données sur les paramètres géométriques.

Il nous faut aller plus loin dans la démarche et déterminer avec quelle précision doivent être déterminés les paramètres géométriques afin de garantir la précision du robot en prenant en compte les erreurs sur les capteurs proprioceptifs. Nous donnons alors les conditions qui doivent être respectées.

- **En déduire la précision nécessaire sur les mesures pratiquées pour étalonner le robot afin de garantir la condition précédente.** Nous avons maintenant connaissance d'un objectif sur la précision nécessaire sur les paramètres à l'identification : nous pouvons donc en déduire quelle doit être l'incertitude sur les mesures (interne et externe) afin de

réaliser cet objectif.

- **Donner une condition d'arrêt pour les algorithmes d'identification itératifs de type Newton, seuil sous lequel la précision sur les paramètres est atteinte.** En effet, à cause des incertitudes, le critère ne sera jamais nul. Ainsi, passé un seuil, il n'est pas possible de répartir les erreurs du critère sur les paramètres. Les conditions précédentes nous aident alors à calculer ce seuil. Noter qu'il a alors une véritable signification physique car il est lié à une précision souhaitée pour le robot.
- **Normaliser les jacobienes d'identification.** Nous avons bâti les conditions au regard de la précision souhaitée pour le robot. Ces informations nous servent alors à normaliser les matrices jacobienes. Ceci nous permet de donner des seuils simples qui garantissent l'identifiabilité des paramètres fondés sur des objectifs physiques de précision et non plus seulement mathématiques.

L'influence des erreurs de modèles sur l'identifiabilité des paramètres est aussi à analyser car elle a des conséquences sur les choix des paramètres à garder ou négliger dans les modèles, [GDD13, GDB13].

5.5.2 Identifiabilité et choix des configurations de mesures

Nous avons vu à la sous-section précédente 5.5.1 qu'il était nécessaire de vérifier les conditions de l'étalonnage et l'identifiabilité des paramètres. Mais nous pouvons aussi agir sur ces configurations afin de rendre cette identifiabilité, et plus généralement la qualité de l'identification, meilleure. Mes travaux [DPM05, Dan02, KDCP14] ont contribué à résoudre cette question.

Tout d'abord, dans [DPM05], nous avons adapté une méthode provenant du domaine de la conception de plans d'expérimentations (Design Of Experiments, DOE). Nous avons ainsi proposé un algorithme itératif permettant de choisir un ensemble de configurations de mesures afin de maximiser l'identifiabilité des paramètres d'étalonnage : cette méthode a été reprise dans nombre de travaux.

Le but est ici de choisir les poses statiques du robot X_K dans lesquelles les mesures M_K seront pratiquées. En effet, considérons les équations d'étalonnage $\mathcal{F}(M_{1,\dots,K}, \xi)$ fonction des mesures M_K prises dans une configuration de mesures X_K et $\mathcal{J}(M_{1,\dots,K}, \xi) = \frac{\mathcal{F}(M_{1,\dots,K}, \xi)}{\xi}$ la matrice d'identification. Pour faire simple, les méthodes aux moindres carrés non-linéaires pseudo-inversent itérativement la matrice \mathcal{J} pour identifier le vecteur ξ . Ainsi, en jouant sur les valeurs de X_K et donc des mesures M_k , nous pouvons améliorer la qualité numérique de \mathcal{J} .

Il nous faut tout d'abord définir des indices permettant de quantifier les qualités numériques. Dans le domaine de l'étalonnage des robots, la communauté utilise des indices d'observabilité, notés $\mathcal{O}_{1,2,3,4}$, ce qui revient à utiliser des indices équivalents définis par le domaine de la conception de plan d'expérimentation (D-, A-, T-, E-optimalité), et repris maintenant en robotique [SK08]. Ces critères sont bâtis en fonction des valeurs singulières $\mu = \mu_1, \dots, \mu_L$ de la matrice d'identification \mathcal{J} ou des valeurs propres $\lambda = \lambda_1, \dots, \lambda_L$ de la matrice d'information $\mathcal{J}^t \mathcal{J}$.

Chacun de ces indices privilégie des propriétés numériques différentes de la matrice d'identification (ou de la matrice d'information), il est donc illusoire de considérer un critère meilleur qu'un autre *généralement*. Par contre, cela peut se faire au regard d'un objectif d'identification. Nous pouvons noter que le critère

\mathcal{O}_1 (ou D-optimalité, le déterminant de $\mathcal{J}^t \cdot \mathcal{J}$) représente un volume (produit des valeurs singulières) qui montre comment les erreurs sur la fonction $\Delta\mathcal{F}$ à minimiser aux moindres-carrés influenceront globalement les corrections à apporter sur les paramètres $\Delta\xi$. La forme de ce volume sera caractérisée par l'indice \mathcal{O}_2 (conditionnement de $\mathcal{J}^t \cdot \mathcal{J}$) que nous chercherons proche de 1 si nous ne voulons pas privilégier une erreur sur l'une des composantes de la correction $\Delta\xi$ au détriment d'une autre et ainsi obtenir une estimation de ξ homogène sur toutes ses composantes. Enfin, si c'est une garantie d'estimation qui est recherchée, la relation $\mu_L \cdot \|\Delta\xi\| \leq \Delta\mathcal{F}$ nous permet d'améliorer la sensibilité sur la correction à apporter aux paramètres en maximisant la plus petite valeur singulière représentée par l'indice \mathcal{O}_3 .

Améliorer ces indices revient donc à améliorer l'identification des paramètres et pour cela nous pouvons jouer sur le choix des configurations de mesures \mathcal{X} . Le but est alors, avant l'étalonnage, de préparer les expérimentations afin de déterminer ces positionnements optimaux du robot notés \mathcal{X}_{opt} .

Pour cela, il est nécessaire de déterminer une solution au problème suivant :

$$\mathbf{Max}_{\mathcal{X}_{opt}=\mathcal{X}} \mathcal{O}_{nu}(\mathcal{J}[\mathcal{X} = \{\mathcal{X}_1, \dots, \mathcal{X}_K\}, \xi) \text{ avec } \mathcal{X} \in \mathcal{W}_{\mathcal{X}} \quad (5.20)$$

où $\mathcal{W}_{\mathcal{X}}$ représente l'espace de travail du robot.

Cependant, nous sommes confrontés à deux problèmes :

- \mathcal{J} est fonction des mesures $M_{1,\dots,K}$ mais aussi d'une estimée des paramètres ξ qui sont les inconnues avant leur identification à partir de mesures effectivement pratiquées.
- Le vecteur \mathcal{X} est de grande dimension car il est fonction de la dimension dans laquelle évolue le robot multipliée par le nombre de mesures K . Le problème d'optimisation (5.20) devient donc difficile. Par exemple, pour un étalonnage classique d'une plate-forme de Gough par une méthode inverse utilisant K configurations de mesure, il est nécessaire de considérer $6 * K$ paramètres pour définir \mathcal{X} .

Concernant le premier problème, les paramètres géométriques étant inconnus lors de la phase de conception du plan d'expérience, il est nécessaire de faire une hypothèse forte : les configurations de mesures qui maximisent l'identifiabilité des paramètres sont *proches* de celles calculées à partir d'une estimée initiale de ξ . Cette hypothèse n'a pas pour le moment été vérifiée, mais semble parfaitement réaliste à la vue des résultats obtenus dans la littérature.

Le problème du nombre de paramètres à considérer pour le problème (5.20) (dimension de \mathcal{X}) est important lorsque nous considérons l'étalonnage de robots spatiaux pour un nombre important de configurations de mesures, ce qui est le cas dans nos travaux des recherches. Ainsi, une approche classique d'optimisation sous contraintes donne des résultats décevants en raison d'une convergence locale des algorithmes. Nous avons donc décidé d'adapter des stratégies sous-optimales, mais efficaces, proposées dans le domaine de la conception d'expériences fondées sur l'algorithme DETMAX proposé par Mitchell [Mit74]. Cette méthode permet de déterminer un ensemble de K configurations de mesures (le nombre K est fixé) et consiste en deux phases répétées itérativement :

- 1. Phase d'ajout. Cette phase consiste à choisir dans l'espace de travail une configuration additionnelle notée \mathcal{X}^+ telle que $\mathbf{Max}_{\mathcal{X}^+=\mathcal{X}_{K+1}} \mathcal{O}_{nu}(\{\mathcal{X}_1, \dots, \mathcal{X}_K\} + \{\mathcal{X}_{K+1} \in \mathcal{W}_{\mathcal{X}}\}, \xi)$ avec $nu = 1, 2, \dots$ numéro de l'indice à considérer. On utilisera ici un algorithme d'optimisation locale.

- 2. Phase de suppression. Cette phase consiste à supprimer une configuration \mathcal{X}^- choisie parmi les $K + 1$ configurations $\mathcal{X}_1, \dots, \mathcal{X}_{K+1}$ telle que $\mathcal{O}_{nu}(\{\mathcal{X}_1, \dots, \mathcal{X}_K - 1\} - \{\mathcal{X}_-\})$ soit maximal.

L’algorithme sélectionne ainsi de manière itérative une pose après une autre à l’intérieur de l’espace de travail. Après quelques itérations (proche du nombre minimal de configurations nécessaires), un ensemble de configurations est obtenu, qui maximise un indice d’observabilité associé à la jacobienne d’identification.

Nos résultats obtenus dans [DPM05] montrent que, en choisissant les configurations de mesures qui maximisent un des indices d’observabilité, nous pouvons augmenter considérablement la robustesse de l’étalonnage de plateforme de Gough par rapport au bruit de mesure.

Cependant, cet algorithme est sensible aux minima locaux. C’est pourquoi nous proposons dans [Dan02] des méthodes de méta-heuristique pour diminuer la sensibilité de notre algorithme. Elles nous ont permis de sortir des minima locaux et ainsi d’améliorer encore les résultats.

Toutefois, nous observons que le choix de cet indice \mathcal{O}_1 (le produit des valeurs singulières) est le mieux adapté à notre algorithme, en effet :

- Le processus d’optimisation est moins sensible aux minima locaux en utilisant cet indice (\mathcal{O}_1) en place de $\mathcal{O}_{2,3}$ ou 4 (nombre de conditions, la plus petite valeur singulière ou le produit des deux).
- Tous les indices d’observabilité ($\mathcal{O}_{1,2,3,4}$) sont maximisés ou près d’être maximisés si le critère optimisé est \mathcal{O}_1 .
- Nous pouvons observer que les configurations optimales convergent vers la bordure de l’espace de travail.
- La simulation présentée montre l’intérêt que la sélection de l’étalonnage pose sur les résultats de l’identification : ce qui a été confirmé dans toutes nos expérimentations. L’erreur sur les paramètres géométriques due aux bruits de mesures est largement réduite (divisé par 10 à 15). En outre, nous montrons qu’il n’est pas nécessaire de multiplier le nombre de configurations de mesures. En effet, nous observons que passées une vingtaine de configurations, la robustesse de l’identification ne s’améliore pas de façon significative. En comparaison, ce seuil est de l’ordre de 40 à 50, si les poses ne sont pas sélectionnées par un algorithme. La diminution du nombre de mesures est, évidemment, un résultat intéressant pour la mise en pratique de l’étalonnage.

Cet algorithme, et nombre de ses variations, ont été repris par la littérature et cette question est devenue une problématique récurrente du domaine. Dans des travaux plus récents [KDCP14], nous avons commencé à développer symboliquement les critères afin d’obtenir des motifs géométriques de configurations à reproduire, et ceci de façon exacte, afin de définir formellement les configurations optimales pour l’étalonnage. Dans des cas simples, il est possible de trouver ces *motifs* définis dans l’espace des configurations minimales pour étalonner qui permettent d’obtenir une matrice \mathcal{J} optimale au sens de l’ensemble des indices d’optimalité. Ces matrices, calculées par des approches géométriques expliquent les résultats obtenus numériquement.

5.6 Conclusion

J'ai contribué sur tous les aspects de l'étalonnage des manipulateurs et plus particulièrement pour les robots parallèles et à câbles : de la modélisation des équations à l'amélioration de l'identification des paramètres.

Nous retiendrons mon intérêt pour les erreurs de modèles qui sont un des marqueurs de différenciation fort de l'étalonnage des robots séries. Ces erreurs ne sont pas mesurables directement, mais ont pourtant une influence importante sur la qualité des résultats. En ce qui concerne les alternatives aux méthodes des moindres carrés, les approches par intervalles sont particulièrement intéressantes, originales et prometteuses. Il serait intéressant de pouvoir continuer à expérimenter dans ce sens. De plus, les approches peuvent être adaptées pour d'autres problèmes sur-contraints de robotique.

Mais, à mon avis, il reste encore des perspectives importantes sur 3 points essentiels :

- **Il serait intéressant d'explorer la relation entre la géométrie de la configuration des mesures pour l'étalonnage et la géométrie du robot en lui-même.** Quelques résultats préliminaires montrent qu'il est possible de déterminer des motifs de configurations qui maximisent les critères d'identifiabilité de l'étalonnage. En effet, nous pouvons nous limiter au nombre minimal de configurations de mesures quitte à répéter ces motifs plusieurs fois afin d'augmenter le nombre de mesures. Il serait intéressant de développer des méthodes géométriques afin de déterminer ces motifs et d'en étudier les propriétés. Cela permettrait d'associer génériquement, pour chaque architecture de robot, les configurations de mesure pour l'étalonnage et non plus de les calculer numériquement.
- **Comment choisir les configurations de mesures pour valider l'étalonnage ?** Pour le moment, la communauté utilise des configurations prises aléatoirement dans l'espace de travail afin de valider les résultats de l'étalonnage. Elles doivent être, bien évidemment, différentes de celles utilisées pour l'identification des paramètres. Cependant, une étude spécifique doit pouvoir montrer comment choisir judicieusement certaines configurations maximisant la sensibilité du positionnement en fonction des paramètres du modèle. La difficulté est de minimiser, dans le même temps, l'influence des erreurs sur les coordonnées articulaires du robot sur son positionnement. Quelques éléments de réponses se trouvent dans la thèse de T. Gayral, mais il serait bon de développer cet axe.
- Nous pouvons constater que le prix des capteurs chute, que leur qualité augmente et qu'il est donc maintenant envisageable de sur-instrumenter systématiquement les robots. Dans ces conditions, **l'auto-étalonnage est possible et pourquoi ne pas l'effectuer directement en ligne lors du fonctionnement ?** Ainsi, il serait intéressant de repenser l'étalonnage avec la commande et, peut-être, remettre à la mode les méthodes de type filtrage de Kalman qui ont été abandonnées dans le domaine (voir [PG00]).

Bibliographie

- [AAC12] Paul D Arendt, Daniel W Apley, and Wei Chen. Quantification of model uncertainty : Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10) :100908, 2012.
- [ADS13] Julien Alexandre Dit Sandretto. *Étalonnage des robots à câbles : identification et qualification*. Theses, Université Nice Sophia Antipolis, September 2013.
- [AdSDT14] Julien Alexandre dit Sandretto, David Daney, and Gilles Trombettoni. Interval Methods for Model Qualification : Methodology and Advanced Application. *Mathematics in Computer Science*, 2014.
- [ADSTD12] Julien Alexandre Dit Sandretto, Gilles Trombettoni, and David Daney. Confirmation of Hypothesis on Cable Properties for Cable-Driven Robots. In Fernando Viadero and Marco Ceccarelli, editors, *4th European Conference on Mechanism Science*, volume 7 of *Mechanisms and Machine Science*, pages 85–93, Santander, Espagne, September 2012. Springer.
- [ADSTDC14] Julien Alexandre Dit Sandretto, Gilles Trombettoni, David Daney, and Gilles Chabert. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *Computational Kinematics*, volume 15 of *Mechanisms and Machine Science*, pages 209–217. Springer Netherlands, 2014.
- [ADSTDG13] Julien Alexandre Dit Sandretto, Gilles Trombettoni, David Daney, and Chabert Gilles. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *6th Int. Workshop on Computational Kinematics (CK2013)*, volume 15 of *Mechanisms and Machine Science*, pages 209–217, Barcelona, Spain, Espagne, 2013. Springer.
- [BBS87] Paul T Boggs, Richard H Byrd, and Robert B Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal on Scientific and Statistical Computing*, 8(6) :1052–1078, 1987.
- [Bea00] Olivier Beaumont. *An algorithm for symmetric interval eigenvalue problem*. IRISA, 2000.

- [BG67] Å Björk and GH Golub. Iterative refinement of linear least squares solutions by householder transformation. *BIT*, 7 :322–337, 1967.
- [BK01] Sébastien Besnard and Wisama Khalil. Identifiable parameters for parallel robots kinematic calibration. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2859–2866. IEEE, 2001.
- [Cor08] David Corbel. *Contribution to accuracy improvement of parallel robots*. Theses, Université Montpellier II - Sciences et Techniques du Languedoc, December 2008.
- [DACP06a] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval Method for Calibration of Parallel Robots : A Vision-based Experimentation. *Mechanism and Machine Theory*, 41 (8) :929–944, 2006.
- [DACP06b] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval method for calibration of parallel robots : Vision-based experiments. *Mechanism and Machine Theory*, 41(8) :929 – 944, 2006.
- [Dan99] David Daney. Self Calibration of Gough Platform using leg mobility constraints. In *10th world congress of International Federation for the Theory of Machines and Mechanisms (IFTToMM)*, pages 104–109, Oulu, Finlande, June 1999.
- [Dan00] David Daney. *Etalonnage géométrique des robots parallèles*. PhD thesis, 2000.
- [Dan02] David Daney. Optimal Measurement Configurations for Gough Platform Calibration. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 147–152, Washington, États-Unis, 2002.
- [Dan03] David Daney. Kinematic calibration of the Gough Platform. *Robotica*, 21 :677–690, 2003.
- [DAP05] David Daney, Nicolas Andreff, and Yves Papegay, A. Interval method for calibration of parallel robots. In *Computational Kinematics (CK)*, Cassino, Italie, 2005.
- [DE01] David Daney and Ioannis Z. Emiris. Robust parallel robot calibration with partial information. In *IEEE International Conference on Robotics and Automation - ICRA '2001*, page 6 p, Corée, Séoul, 2001. none. Colloque avec actes et comité de lecture. internationale.
- [DE02] David Daney and Ioannis Z. Emiris. Variable elimination for reliable parallel robot calibration. *Electronic Journal of Computational Kinematics (EJCK)*, 2002.
- [DE04] David Daney and Ioannis Emiris. Algebraic Elimination for Parallel Robot Calibration. In *World Congress of International Federation for the Theory of Machines and Mechanisms (IFTToMM)*, pages 212–217, Tianjin, Chine, 2004.
- [DEP⁺06] David Daney, Ioannis Z. Emiris, Yves Papegay, Elias Tsigaridas, and Jean-Pierre Merlet. Calibration of parallel robots : on the

- Elimination of Pose-Dependent Parameters. In *European Conference on Mechanism Science, (EuCoMeS)*, Obergurgl, Autriche, 2006.
- [DPM05] David Daney, Yves Papegay, and Blaise Madeline. Choosing Measurement Poses for Robot Calibration with the Local Convergence Method and Tabu Search. *International Journal of Robotics Research*, 24(6) :501–518, 2005.
- [DPN04] David Daney, Yves Papegay, A., and Arnold Neumaier. Interval Methods for Certification of the Kinematic Calibration of Parallel Robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1913–1918, New Orleans, États-Unis, 2004. IEEE.
- [EC95] Ioannis Z Emiris and John F Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *Journal of Symbolic Computation*, 20(2) :117–149, 1995.
- [EM05] Gaë Ecorchard and Patrick Maurine. Self-calibration of delta parallel robots with elastic deformation compensation. In *IROS*, pages 1283–1288, 2005.
- [ENM10] Gaël Ecorchard, Reimund Neugebauer, and Patrick Maurine. Elasto-geometrical modeling and calibration of redundantly actuated pkms. *Mechanism and machine theory*, 45(5) :795–810, 2010.
- [Gay13] Thibault Gayral. *Étalonnage d'un instrument d'observation spatiale actif*. PhD thesis, Université Nice Sophia Antipolis, 2013.
- [GD14] Thibault Gayral and David Daney. A Sufficient Condition for Parameter Identifiability in Robotic Calibration. In Federico Thomas and Alba Perez Gracia, editors, *Computational Kinematics*, volume 15 of *Mechanisms and Machine Science*, pages 131–138. Springer Netherlands, 2014.
- [GDB13] Thibault Gayral, David Daney, and Marc Bernot. Model Discrepancy in Robotic Calibration : Its Influence on the Experimental Parameter Identification of a Parallel Space Telescope. In *IROS - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - 2013*, Tokyo, Japon, 2013.
- [GDD13] Thibault Gayral, David Daney, and Julien Ducarne. Flexure joints modeling for micrometer accuracy of an active 6-PUS space telescope through experimental calibration. In *ICRA - IEEE Int. Conf. on Robotics and Automation - 2013*, pages 4525–4530, Karlsruhe, Allemagne, 2013.
- [HKKP02] Chanhee Han, Jinwook Kim, Jongwon Kim, and Frank Chongwoo Park. Kinematic sensitivity analysis of the 3-upu parallel mechanism. *Mechanism and Machine Theory*, 37(8) :787–798, 2002.
- [IC74] H Max Irvine and Ti K Caughey. The linear theory of free vibrations of a suspended cable. In *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, volume 341, pages 299–315. The Royal Society, 1974.

- [Inn95] C Innocenti. Algorithms for kinematic calibration of fully-parallel manipulators. In *Computational Kinematics' 95*, pages 241–250. Springer, 1995.
- [JADG12] Dit Sandretto Julien Alexandre, David Daney, and Marc Gouttefarde. Calibration of a Fully-Constrained Parallel Cable-Driven Robot. In *ROMANSY'2012 : 19th CISM-IFTOMM Symposium on Robot Design, Dynamics, and Control*, pages 1–8, Paris, France, June 2012.
- [KDCP14] Alexandr Klimchik, David Daney, Stéphane Caro, and Anatoly Pashkevich. Geometrical Patterns for Measurement Pose Selection in Calibration of Serial Manipulators. In *Advances in Robot Kinematics, 14th Inter. Symposium*, Ljubljana, Slovénie, 2014.
- [KN86] Rudolf Krawczyk and Arnold Neumaier. An improved interval newton operator. *Journal of mathematical analysis and applications*, 118(1) :194–207, 1986.
- [MAR00] Frédéric MARQUET. *Contribution à l'étude de l'apport de la redondance en robotique parallèle*. Ph.d. thesis, Université Montpellier II, Montpellier, France, October 2000.
- [MD10] Jean-Pierre Merlet and David Daney. A portable, modular parallel wire crane for rescue operations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2834–2839, Anchorage, États-Unis, 2010. IEEE.
- [Mit74] Toby J Mitchell. An algorithm for the construction of “d-optimal” experimental designs. *Technometrics*, 16(2) :203–210, 1974.
- [MWZ93] Oren Masory, Jian Wang, and Hanqi Zhuang. On the accuracy of a stewart platform. ii. kinematic calibration and compensation. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 725–731. IEEE, 1993.
- [PCDG00] V Parenti-Castelli and R Di Gregorio. Influence of manufacturing errors on the kinematic performance of the 3-upu parallel mechanism. In *2nd Chemnitzer Parallelkinematik Seminar*, pages 85–99, 2000.
- [PE00] Amit J Patel and Kornel F Ehmann. Calibration of a hexapod machine tool using a redundant leg. *International Journal of Machine Tools and Manufacture*, 40(4) :489–512, 2000.
- [PG00] Ph Poignet and Maxime Gautier. Comparison of weighted least squares and extended kalman filtering methods for dynamic identification of robots. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 4, pages 3622–3627. IEEE, 2000.
- [PG01] Ph Poignet and M Gautier. Extended kalman filtering and weighted least squares dynamic identification of robots. *Control Engineering Practice*, 9(12) :1361–1372, 2001.
- [RVA+06] Pierre Renaud, Andres Vivas, Nicolas Andreff, Philippe Poignet, Philippe Martinet, François Pierrot, and Olivier Company. Kinematic and dynamic identification of parallel mechanisms. *Control engineering practice*, 14(9) :1099–1109, 2006.

- [SH08] Yu Sun and John M Hollerbach. Observability index selection for robot calibration. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 831–836. IEEE, 2008.
- [SK08] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.
- [STD13] JulienAlexandredit Sandretto, Gilles Trombettoni, and David Daney. Confirmation of Hypothesis on Cable Properties for Cable-Driven Robots. In Fernando Viadero and Marco Ceccarelli, editors, *New Trends in Mechanism and Machine Science*, volume 7, pages 85–93. Springer Netherlands, 2013.
- [Vis96] Peter Vischer. *Improving the accuracy of parallel robots*. PhD thesis, 1996.
- [WHA95] Charles W Wampler, John M Hollerbach, and Tatsuo Arai. An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *Robotics and Automation, IEEE Transactions on*, 11(5) :710–724, 1995.
- [YYLW08] Liu Yu, Jiang Yanshu, Bin Liang, and XU Wenfu. Calibration of a 6-dof space robot using genetic algorithm. *Chinese Journal of Mechanical Engineering*, 21(6) :6–13, 2008.
- [ZYM98] Hanqi Zhuang, Jiahua Yan, and Oren Masory. Calibration of stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. *Journal of Robotic Systems*, 15(7) :395–405, 1998.

Chapitre 6

Conclusion générale

L'objectif de ce document est de montrer comment nous avons amélioré les performances des robots en amont et en aval de leur fabrication. Dans les deux cas, nous avons cherché et obtenu des valeurs pour les paramètres intrinsèques qui définissent le modèle du robot : lors de la conception, nous avons déterminé des ensembles de valeurs qui garantissent des propriétés données par un cahier des charges et lors de l'étalonnage, nous avons précisé ces valeurs afin que le modèle prédise au mieux les propriétés réelles du robot.

Nous avons présenté les outils qui nous ont permis de réaliser ces objectifs, notamment l'analyse par intervalles qui est une approche intéressante pour prendre en compte les variabilités des paramètres et qui permet de traiter, de façon unifiée, les incertitudes et les différents états du systèmes.

Nous avons montré qu'il est intéressant de bien connaître les concepts et l'algorithmique d'intervalles pour en tirer profit dans l'utilisation des outils mathématiques, géométriques, et mécaniques pour la modélisation de robots. Cette technicité requise peut être vue comme un désavantage et un frein à l'utilisation de ces méthodes. Mais au vu des résultats obtenus - et ceci bien au delà des mes contributions - je plaide plutôt pour un enseignement plus poussé de ces approches au sein de nos formations. Sur les aspects fondamentaux, il faut continuer à développer les briques de base nécessaires à la résolution de nos problèmes : algèbre linéaire, résolution de systèmes, etc, et approfondir certains points essentiels pour la diffusion des résultats en dehors de notre communauté, par exemple la prise en compte des dépendances entre les paramètres au sein des méthodes génériques. Sur les aspects applicatifs, nombre de problèmes peuvent trouver des solutions en utilisant ces approches. L'essor de bibliothèques traitant des intervalles (ALIAS, Intel®MKL, Mathematica, Ibex ...) est un vecteur d'utilisation mais qui ne répond que partiellement aux problématiques et notre expertise sur la formalisation par intervalles des problèmes doit pouvoir être transmise. Et même si l'analyse généralement au pire cas fournie par les approches par intervalles n'est pas toujours la plus pertinente, et si d'autres représentations de l'incertitude sont possibles, souhaitables et complémentaires, le pouvoir calculatoire de l'analyse par intervalles est néanmoins très efficace sur les problèmes réels.

La conception appropriée offre une véritable méthodologie applicable bien au-delà de la robotique. Nous nous sommes efforcés de la présenter de façon systématique et l'enjeu futur est de la généraliser à d'autres systèmes. Cela

peut nécessiter d'étendre les capacités de traitement de l'analyse par intervalles. Ainsi, mes contributions sur le calcul des valeurs propres des matrices intervalles sont directement liées au calcul de la dextérité ou de la manipulabilité des robots comme propriétés nouvelles à considérer dans nos algorithmes de conception. De plus, nous avons peu, pour le moment, considéré les propriétés élasto-statiques/élasto-dynamiques des robots mais l'étude des robots à câbles va nous amener vers ce type de contribution. D'un point de vue général, l'essor des robots flexibles nous pousse vers cette voie qui nous conduira probablement à de nouveaux développements de méthodes par intervalles.

Une approche similaire a été faite concernant l'étalonnage. Les caractérisations par intervalles des solutions d'un système sur-contraint que j'ai mises en place, offrent de nouvelles interprétations des résultats obtenus, complémentaires aux approches classiques, notamment sur la certification de l'identification. De plus, l'extension aux intervalles des coefficients des équations d'étalonnage permet de considérer un modèle épaissi qui prend en compte les erreurs de modèles tout en gardant une structure symbolique simple.

Une des directions de recherche forte que j'ai poussée concerne les conditions de l'étalonnage, ce qui peut être vu comme la conception d'un plan d'expérimentation. Le choix des configurations de mesures dans lesquelles positionner le robot reste un des enjeux importants de l'étalonnage. Même si les méthodes numériques pour le calcul des poses sont largement utilisées dans les articles récents, nous pouvons maintenant les calculer géométriquement, et mes travaux avec l'IRCCyN vont dans ce sens.

Les perspectives générales sont liées à l'évolution de la robotique ces dernières années. Nous vivons une révolution importante en matière de hardware qui peut nous faire repenser le robot. En effet, l'électronique (arduino, composants embarqués, capteurs) aussi bien que la mécanique (flexibilité de matériaux, imprimante 3D) deviennent modulaires, accessibles et à des coûts abordables. Si les robots sont plus faciles à concevoir, prototyper et construire, ils sont moins fiables dans leurs fabrications et leurs modèles deviennent plus compliqués, ce qui renforce les axes que nous avons mis en place. Ainsi, on ne peut se passer d'une bonne conception de ces robots afin de minimiser l'influence de ces défauts sur ses performances. De plus, la sur-instrumentation par des capteurs additionnels (qu'ils soient internes ou externes) est envisageable : ceci ouvre des perspectives en matière de couplage commande-auto-étalonnage et d'amélioration de la fiabilité des systèmes, tolérance aux pannes.

Mais les perspectives qui sont au coeur de mon projet de recherche futur, concernent la collaboration de l'humain avec le robot. En effet, la conception de systèmes robotisés en interaction avec l'homme nécessite de prendre en compte de nouveaux types de contraintes pour leur sécurité mais aussi pour le partage des tâches opérationnelles. L'homme, dans son modèle bio-mécanique, peut être vu comme une chaîne cinématique qui participe à l'exécution d'une trajectoire ou de l'application d'une force. Le robot d'assistance est alors vu comme une partie d'un système cybernétique complet qu'il faut concevoir afin d'aider un opérateur sans l'entraver, ou le contraindre. Cette conception ne peut se faire que si les mouvements et la capacité de l'homme sont mesurés puis analysés. Le monitoring de l'opérateur aussi bien que la synthèse de nouveaux indices d'évaluation de ses capacités motrices sont alors à développer. Là encore la variabilité et les incertitudes des paramètres humains, environnementaux et des systèmes d'assistances sont grandes. Les approches présentées dans ce document

me paraissent intéressantes à adapter et développer.

Chapitre 7

Curriculum Vitae

7.1 Parcours et activités

David Daney, 43 ans 2 enfants, 35 rue Stéhélin, 33200 Bordeaux, 0678301765.

7.1.1 Responsabilités récentes

2015-2016 | **Mise en place d'un projet de recherche** sur CRI Bordeaux-Sud Ouest
AGOI

L'une des forces de nos entreprises est liée au savoir-faire de nos opérateurs industriels capables de réaliser des tâches complexes qui nécessitent une expertise manuelle technique. Comment la préserver, la transmettre, la certifier, la faire évoluer et l'assister : nos travaux cherchent à répondre à ces enjeux.

En effet, il est possible d'aider l'opérateur en concevant des systèmes robotiques collaboratifs dans le but d'améliorer la productivité et la qualité de la tâche tout en préservant sa sécurité, son confort et sa santé. Pour cela, il faut évaluer le geste technique par le développement d'indices novateurs complémentaires aux approches ergonomiques et biomécaniques afin d'identifier les phases de fragilité et d'assistance. Ces informations sont nécessaires pour la conception de robots. Nous proposons d'utiliser une modélisation robotique ensembliste des postures et trajectoires des membres de l'opérateur afin de gérer les très grandes variabilités de paramètres à considérer et de les rendre compatible avec les modèles de robots d'assistance permettant la synthèse des interactions physique homme-robot.

2011-2014 | **Responsable Inria Project Lab (IPL)** : 80 chercheurs impliqués
Personally Assisted Living, PAL

Inria Project Lab PAL est une infrastructure financée par la direction de la recherche d'Inria qui propose des solutions technologiques de type capteurs intelligents, robotique ou domotique afin d'améliorer la qualité de vie des personnes fragiles et leur entourage. Plus particulièrement ces solutions visent à favoriser l'autonomie des personnes âgées et leur maintien à domicile par la sécurisation, la préservation ou le rétablissement des fonctions nécessaires à leur vie quotidienne. Cette action regroupe 10 équipes Inria (robotique ou spécialisées dans

le monitoring des personnes) associées à des partenaires de recherche (technologiques, médicaux, sciences sociales) qui travaillent ensemble autour de trois principaux thème : l'assistance à la mobilité, l'évaluation de l'état de fragilité des personnes et l'analyse de comportement.

7.1.2 Responsabilités activités Collectives

Depuis 2015	Président du comité de labellisations du Cluster Aquitain Robotics
Depuis 2015	Co-animateur Cobotique Pôle Aerospace Valley
Depuis 2014	Membre du bureau Aquitain Robotics
2011-2014	Responsable Inria Project Lab (IPL) , Personally Assisted Living, PAL
2013-2014	Responsable permanent équipe Phoenix Inria
2011-2013	Membre du comité d'évaluation ANR TecSan
2011-2012	Membre du conseil scientifique du CNR SDA (France Silver éco) Centre National de Référence, Santé à Domicile + Autonomie
2009-2010	Correspondant Plate-formes Expérimentales et Technologiques
2007-2008	Président de la Commission des Utilisateurs des Moyens Informatiques pour la Recherche du CRISAM
2004-2006	Membre de la Commission des Utilisateurs des Moyens Informatiques pour la Recherche du CRISAM

7.1.3 Parcours de recherche

Depuis sept. 2013	Chargé de recherche 1er classe , Inria Bordeaux - Sud Ouest Robotique, Assistance à la personne, Analyse d'activité, Placement de capteurs
2011-2014	Responsable Inria Project Lab , IPL Personally Assisted Living
2013-2014	Responsable permanent , équipe Phoenix
2006-2013	Chargé de recherche 1er classe , équipe Coprin
2003-2006	Chargé de recherche 2ème classe , équipe Coprin Inria Sophia Antipolis Méditerranée Robots parallèles, robot à câble, Analyse par intervalles
2002	Chercheur invité Université Rutgers, USA, Conception de robot Université Laval, Québec, Canada, Conception de robot Université McGill, Montréal, Canada, Théorie des mécanismes
2001	Post Doctorant , Loria, Inria Lorraine, équipe Space MPFR : bibliothèque de calcul numérique en précision arbitraire (Library GNU Public Licence)
2000	Post Doctorant , CMW - Constructions Mécaniques des Vosges, Lorraine Commande et étalonnage de l'Hexapode 300, Machines-outils 5 axes UGV
1997-2000	Doctorant , Université de Nice Sophia Antipolis, INRIA étalonnage de robots parallèles, Directeur de thèse : Jean-Pierre Merlet
1998	Ingénieur Recherche , CMW Constructions Mécaniques des Vosges, Lorraine Développement de logiciels nécessaires à la commande de l'Hexapode PR48
1995	Ingénieur développement , IBM, Pessac Développement et maintenance d'un applicatif (Transmission d'images par le réseau RTC)

7.1.4 Formation

1997-2000	Doctorat , Université de Nice Sophia Antipolis, INRIA, équipe SAGA, étalonnage de robots parallèles
1995-1996	DEA Aravis, Université de Nice - Sophia Antipolis, ESSI, Algorithmes, Robotique, Automatique, Vision, Image, Signal
1993-1995	Licence, Maîtrise Génie Mathématique et Informatique, Université Paul-Sabatier Toulouse, LAAS-IRIT option Systèmes Intelligents, Interactifs et Robotique

7.2 Encadrement

7.2.1 Encadrement de Doctorants

2015-2018	75%	Pierre-Olivier Martin, Biomechanical analyse and certification of industrial gestures, DT : JM Salotti
2014-2017	85%	Aurélien Massein, Conception d'environnement instrumenté pour la reconnaissance de comportements humains Financement : Cordis Inria, DT : Y. Papegay
2010-2013	95%	Thibault Gayral, Étalonage d'un instrument d'observation terrestre Financement : Thales Alenia Space / PACA, DT : J-P Merlet
2010-2013	75%	Julien Alexandre dit Sandretto, Étalonage des robots à câbles Financement ANR Cogiro, DT : G. Trombettoni
2009-2012	30%	Mandar Harshe, Conception d'un robot pour la rééducation et l'assistance des personnes Financement Cordis, DT : J-P Merlet
2007-2010	30%	Julien Hubert, Étude des singularités de robots parallèles Financement ANR Siropa, DT : J-P Merlet
2007-2010	30%	Christos Syrseloudis, Conception d'un robot pour la rééducation de la cheville, Financement Reflexion partenaire industriel, ministère Grec du développement, DT : I. Emiris

Le taux d'encadrement total : 420 %

7.2.2 Encadrement de doctorants sur articles communs

2004-2007	Carlos Grandon, Solving Systems of Distance Equations with Uncertainties Financement CONICYT, Chili, DT : B. Neuveu
2004-2006	Gilles Chabert, Techniques d'intervalles pour la résolution de systèmes d'équations Financement ministère, DT : G. Trombettoni
2003-2005	Alexandre Goldsztejn, Définition et applications des extensions des fonctions réelles aux intervalles généralisés : révision de la théorie des intervalles aux intervalles généralisés, Thales, DT : M. Rueher

7.2.3 Post doctorants/ATER

2015-2016	Charles Fage, PostDoc RAPP
2008-2009	Milan Hladik, PostDoc Cordis-Inria
2006-2008	Denny Oetomo, PostDoc Strep-IST/Nest ARES
2006-2007	Marc Gouttefarde, PostDoc Cordis-Inria
2003-2004	Blaise Madeline, ATER Unice

7.2.4 Jeunes ingénieurs

2009-2010	Guillaume Aubertin, Drivers, interface et commande ADT Marionet
2008-2009	Bogda Rosca, Drivers, interface et commande ADT Marionet
2007-2008	Raphael Peirera, Développement de Int4Sci

7.2.5 Étudiants Master

2016	Hadrien Carrion, Robot Collaboratif pour la prévention des Troubles Musculo-Squelettiques,
2015	Maïlys Pointreau, Construction d'une carte d'a priori sur les activités et déplacements des personnes dans leur domicile, Master – Sciences Cognitives et Ergonomie, Univ. Bordeaux
2015	Quentin Lebour, Analyse du geste des opérateurs industriels, Ensc, INP Bordeaux
2014	Maxime Girard, Analyse de signaux pour le monitoring du sommeil, Enseirb-Matmeca, INP Bordeaux
2014	B. Frugier, Monitoring du Sommeil, M1 Master Sciences Cognitives, Univ. Bordeaux
2005	T. Adelmar, Évaluation par intervalles des chaînes cinématiques, M1 Imerir

7.3 Dissémination

7.3.1 Cours

Depuis 2015	Cours robotique, Master Sciences cognitives (37.5h EPT/ans)
Depuis 2015	Cours robotique, Ensc (3h EPT/ans)
Depuis 2013	Responsable du module Mathématique pour la robotique, Enseirb-matmeca (22h ETP/ans)
2010-2013	Responsable du module régulation industrielle, Master, ITII, Polytech'Nice (47h ETP/ans)
2004-2013	Responsable du module robotique médicale, Master Génie Bio-Médical, Univ. Nice Sophia Antipolis (22h ETP/an)
2007-2008	Cours d'analyse par intervalles, Master, antenne Sophia Antipolis, ENS Lyon (12h ETP/an)
2007	Cours de robotique, Master, ISIA, école de mines de Paris (12h ETP)

7.3.2 Projet étudiants

2015	Modélisation robotique et simulation du geste d'un opérateur industriel, M2, Enseirb-Mathmeca, Explosense
2014	Monitoring de cuisine, M2, Enseirb-Mathmeca, G. Schahl et Al
2014	Conception d'une dessert robotisée, M2, Enseirb-Mathmeca, L. Leclec'h, C. Brisset, T. Guilbaud
2014	Hypnos, Lit intelligent pour le monitoring du sommeil, M2, Enseirb-Mathmeca, Ensc, F. Chenebault, C. Dallet, M. Gauquelin, T. Guitard, J. Seguin

7.3.3 Grand public

2012	Participation à Sciences en fête
2011	Participation EU Robotics week
2010	Article de vulgarisation : Les aides abordables pour nos aînés, La recherche, (no 446), Novembre 2010
2010	Conférences Lycée, robotique et métiers de la recherche

7.3.4 Développement logiciels

Int4Sci	Module Scilab d'analyse par intervalles, leadership scientifique et technique
Alias	Bibliothèque d'analyse par intervalles, participation
MPFR	Bibliothèque de calcul numérique en précision arbitraire (Library GNU Public Licence), participation
HexaUtils	Bibliothèque Robotique pour Machine Outils UGV, leadership scientifique et technique

7.4 Contrats de recherche et collaborations

7.4.1 Projets Internationaux

2013-2016	FP7-ICT-2013.5.3, RAPP, Robotic Applications for Delivering Smart User Empowering Applications, Certh, Wut, Sigma, Ortelio, Ormylia, Matia
2011-2014	FP7-ICT-FoF CableBot, Parallel Cable Robotics for Improving Maintenance and Logistics of Large-Scale Products, EADS, Lirmm, Fraunhofer-Ipa, Airbus, Acciona, Technalia, Vicinay
2007-2010	Pined (Fin. Ministère Grec), Robot de rééducation de la cheville, Athens Univ., Reflexion
2006-2009	Strep-IST/Nest ARES, Assembling Reconfigurable Endoluminal Surgical system, Sssa, Ethz, UB-Cben
2006-2009	3+3 Med Roras Project, Conception de robot de rééducation, Cassino Univ., Monastir engineer school, Oran Univ., Sousse Engineer school, Mentouri Univ.
2003-2004	PAI Platon, Calibration de robot, Athens Univ.

7.4.2 Projets Nationaux

2010-2013	ANR CoGiRo, Commande de robot géant, Lirmm, Lasmea, Technalia
2009-2010	ADT Marionet, Développement de robots à câbles
2007-2011	ANR Siropa, étude des singularités des robots parallèles, Ircsyn, Univ. Rennes, Lina, Inria Rocquencourt
2007-2008	RobPaca Color, Conception optimale et calibration de robot à câbles, Lirmm
2003-2005	ACI V3F Validation and checking of floating point number computations, Lirc, Irisa, Cea
2003-2004	Robea MP2, Précision des mécanismes complexes, Lirmm, Lasmea, Ircsyn

7.4.3 Contrats Industriels

Depuis 2015	AIO, analyse du geste industriel
Depuis 2015	Aerocampus, outils pour la formation
2010-2013	Thales Alenia Space, étalonnage d'un instrument d'observation spatiale
2010-2013	EADS, Grue à câbles de très grande dimension
2009-2013	Technalia, Modélisation et Identification des robots à câbles
2007-2010	Reflexion, PME Grecque
2003-2004	Alcatel Space Industries, Conception d'un satellite à structure déployable
1998-2003	CMW, Conception et étalonnage d'une machine outil 5x UGV

7.4.4 Collaborations actuelles

Industrielles	AIO, Explosense, Aerospline, Safran
Académiques	Univ. Toulon, Inria Nancy, Inria Sophia, Lasmae, Lirmm, Ircyn, Athens Univ. (Grèce), Charles Univ ...

7.5 Visibilité Nationale - Internationale

7.5.1 Expertises éditoriales

éd. Invité	Elsevier journal Robotics and Autonomous Systems RAS, Elsevier, Special Issue 2015 Autonomous robots, Springer, Special Issue 2016
éd. Associé	ICRA'12, IROS'15, IROS'16

7.5.2 Relecture

Journaux	IEEE trans. IJRR, Robotica, MMT, J. Mech. Des., IJRA, AR, AJC, RAS, Robotics and Computer Integrated Manufacturing, Simulation Modeling Practice and Theory, Precision Engineering...
Conférences	Robotique : IROS, ICRA, ARK, CK, Iftomm, CIFA ... Programmation par contrainte et analyse par intervalles : CPAIOR, CP, ACM SAC

7.5.3 Participation des comités

Scientifique	Centre National de Référence Santé à Domicile et Autonomie, 2011-2012
évaluation	ANR TecSan, 2011-2013
Programme	IntCP, JNRR, IFTOMM'07, IROS'08, ICRA'12, ICRA'15, WASRHE'12'13'14
Sélection	Université de Lorraine, MCF, Loria, Nancy, Numéro de poste : 27MCF0911

7.5.4 Participation à des jury des thèses

24-05-16	A. Bib, Université de Lorraine
15-12-15	M. Andries, Université de Lorraine
10-12-15	A. Panchéa, Université de Bourges
22-04-15	A Berti, University of Bologna, Italy.
15-02-14	Y. Wu, école centrale de Nantes
08-01-13	J. Rios Martinez, Univ. Grenoble
29-11-13	T. Gayral , Univ. Nice Sophia Antipolis
11-09-13	J. Alexandre dit Sandretto , Univ. Nice Sophia Antipolis
21-12-12	M. Harshe , école Nationale Supérieure des Mines de Paris
07-07-11	S. Nourizadesh, Univ. Nancy
01-06-11	C. Syrseloudis , Athens Univ.
28-10-10	J. Hubert , école Nationale Supérieure des Mines de Paris
06-12-07	T. Dallej, Univ. Blaise Pascal, Clermont-Ferrand
19-01-07	G. Chabert , Univ. Nice Sophia Antipolis

7.5.5 Organisation Conférences

2017	JNRR, Biarritz, France
2016	Local chair, The 15th Advances in Robot Kinematics conference, (ARK'16), June 27-30, Sophia Antipolis, France
2012-2016	Organisateur Int. Workshop on Assistance and Service robotics in a human environment, IROS'16 Daejeon, IROS'15 Hamburg, IROS'14, Chicago, USA, 60 participants, IROS'13, Tokyo, Japon, 70 Participants, IROS'12, Vilamoura, Portugal, 50 Participants
2015	Publication chair, The 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO 2015) July 1-3, Lyon, France
2008	Local Chair IROS'08, Nice, France, 1400 Participants
2007	Chargé du processus de relecture, World Congress de l'IF-TOMM'07, Besançon, France, 800 Participants

7.5.6 Missions Expertises

Dec. 2010	IT Shanghai, Direction des relations Internationales, Consulat de France, pour le thème Robotique, Perception, Interaction Homme-Robot, Assistance à la personne
Déc 2011	Philips, Eindhoven, Pays-Bas

Bibliographie

- [ADSDGB12] Julien Alexandre Dit Sandretto, David Daney, Marc Gouttefarde, and Cédric Baradat. Calibration of a fully-constrained parallel cable-driven robot. Rapport de recherche RR-7879, INRIA, February 2012.
- [AdSDT14] Julien Alexandre dit Sandretto, David Daney, and Gilles Trombettoni. Interval Methods for Model Qualification : Methodology and Advanced Application. *Mathematics in Computer Science*, 2014.
- [ADSTD12a] Julien Alexandre Dit Sandretto, Gilles Trombettoni, and David Daney. Confirmation of Hypothesis on Cable Properties for Cable-Driven Robots. In Fernando Viadero and Marco Ceccarelli, editors, *4th European Conference on Mechanism Science*, volume 7 of *Mechanisms and Machine Science*, pages 85–93, Santander, Espagne, September 2012. Springer.
- [ADSTD12b] Julien Alexandre Dit Sandretto, Gilles Trombettoni, and David Daney. Interval Methods for Model Qualification : Methodology and Advanced Application. Rapport de recherche RR-7880, INRIA, February 2012.
- [ADSTDC14] Julien Alexandre Dit Sandretto, Gilles Trombettoni, David Daney, and Gilles Chabert. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *Computational Kinematics*, volume 15 of *Mechanisms and Machine Science*, pages 209–217. Springer Netherlands, 2014.
- [ADSTDG13] Julien Alexandre Dit Sandretto, Gilles Trombettoni, David Daney, and Chabert Gilles. Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming. In Federico Thomas and Alba Perez Gracia, editors, *6th Int. Workshop on Computational Kinematics (CK2013)*, volume 15 of *Mechanisms and Machine Science*, pages 209–217, Barcelona, Spain, Espagne, 2013. Springer.
- [DACP06a] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval Method for Calibration of Parallel Robots : A Vision-based Experimentation. *Mechanism and Machine Theory*, 41 (8) :929–944, 2006.
- [DACP06b] David Daney, Nicolas Andreff, Gilles Chabert, and Yves Papegay. Interval method for calibration of parallel robots : Vision-based

- experiments. *Mechanism and Machine Theory*, 41(8) :929 – 944, 2006.
- [Dan98] David Daney. Mobility constraints on the legs of a parallel robot to improve the kinematic calibration. In *New machine concepts for handling and manufacturing devices on the basis of parallel structures*, pages 187–200, Braunschweig, Allemagne, 1998.
- [Dan99] David Daney. Self Calibration of Gough Platform using leg mobility constraints. In *10th world congress of International Federation for the Theory of Machines and Mechanisms (IFTOMM)*, pages 104–109, Oulu, Finlande, June 1999.
- [Dan02] David Daney. Optimal Measurement Configurations for Gough Platform Calibration. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 147–152, Washington, États-Unis, 2002.
- [Dan03a] David Daney. Étalonnage des robots parallèles. In *Journée robots/Max*, Sophia Antipolis, France, April 2003.
- [Dan03b] David Daney. Kinematic calibration of the Gough Platform. *Robotica*, 21 :677–690, 2003.
- [Dan10] David Daney. Int4Sci : Scilab Interface for Interval Analysis. In *14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, Lyon, France, 2010.
- [Dan11] David Daney. Robotic services for the elderly improving their autonomy at home. In *workshop on Robotics for Neurology and Rehabilitation, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, États-Unis, September 2011.
- [Dan14] David Daney. Collaboration Between Robots and Humans. In *ITechLaw European Conference*, Paris, France, September 2014.
- [DAP05a] David Daney, Nicolas Andreff, and Yves Papegay. Interval method for calibration of parallel robots : a vision-based experimentation. *Mechanism and machine theory 0094-11X*, page 1, 2005.
- [DAP05b] David Daney, Nicolas Andreff, and Yves Papegay, A. Interval method for calibration of parallel robots. In *Computational Kinematics (CK)*, Cassino, Italie, 2005.
- [DE01a] David Daney and Ioannis Z. Emiris. Robust parallel robot calibration with partial information. In *IEEE International Conference on Robotics and Automation - ICRA '2001*, page 6 p, Corée, Séoul, 2001. none. Colloque avec actes et comité de lecture. internationale.
- [DE01b] David Daney and Ioannis Z. Emiris. Variable elimination for reliable parallel robot calibration. In Franck C. Park and Cornel C. Iurascu, editors, *In 2nd Workshop on Computational Kinematics - CK'2001*, page 12 p, Seoul, Korea, 2001. School of Mechanical and Aerospace Engineering, none. Colloque avec actes et comité de lecture. internationale.

- [DE02] David Daney and Ioannis Z. Emiris. Variable elimination for reliable parallel robot calibration. *Electronic Journal of Computational Kinematics (EJCK)*, 2002.
- [DE03] David Daney and Ioannis Z. Emiris. Élimination de variables appliquée à l'étalonnage des robots parallèles. In *Journées Nationales de Calcul Formel*, Luminy, France, 2003.
- [DE04] David Daney and Ioannis Emiris. Algebraic Elimination for Parallel Robot Calibration. In *World Congress of International Federation for the Theory of Machines and Mechanisms (IFTOMM)*, pages 212–217, Tianjin, Chine, 2004.
- [DEP⁺06] David Daney, Ioannis Z. Emiris, Yves Papegay, Elias Tsigaridas, and Jean-Pierre Merlet. Calibration of parallel robots : on the Elimination of Pose-Dependent Parameters. In *European Conference on Mechanism Science, (EuCoMeS)*, Obergurgl, Autriche, 2006.
- [DLRW00] David Daney, Sylvain Lazard, Steve Robbins, and Sue Whitesides. Géométrie algorithmique pour la CAO et la conception optimale de robots. In *68ème Congrès ACFAS*, Montréal, Canada, May 2000.
- [DP12] David Daney and Robert Philippe. Research in the service of personal assistance. In *Health Information Technologies, HIT*, Paris, France, May 2012.
- [DPM05] David Daney, Yves Papegay, and Blaise Madeline. Choosing Measurement Poses for Robot Calibration with the Local Convergence Method and Tabu Search. *International Journal of Robotics Research*, 24(6) :501–518, 2005.
- [DPN04] David Daney, Yves Papegay, A., and Arnold Neumaier. Interval Methods for Certification of the Kinematic Calibration of Parallel Robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1913–1918, New Orleans, États-Unis, 2004. IEEE.
- [GD12] Thibault Gayral and David Daney. Experimental calibration of an active space telescope with flexure joints. Rapport de recherche RR-8096, INRIA, October 2012.
- [GD13] Thibault Gayral and David Daney. A sufficient condition for parameter identifiability in robotic calibration. In Federico Thomas and Alba Perez Gracia, editors, *CK2013 - 6th Int. Workshop on Computational Kinematics*, volume 15, pages 131–138, Barcelona, Espagne, 2013. Springer Netherlands.
- [GD14] Thibault Gayral and David Daney. A Sufficient Condition for Parameter Identifiability in Robotic Calibration. In Federico Thomas and Alba Perez Gracia, editors, *Computational Kinematics*, volume 15 of *Mechanisms and Machine Science*, pages 131–138. Springer Netherlands, 2014.
- [GDB13] Thibault Gayral, David Daney, and Marc Bernot. Model Discrepancy in Robotic Calibration : Its Influence on the Experimental Parameter Identification of a Parallel Space Telescope. In *IROS -*

- IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - 2013*, Tokyo, Japon, 2013.
- [GDB14] Thibault Gayral, David Daney, and Marc Bernot. Modeling and experimental calibration of an active space telescope based on a parallel robotic architecture. *Submitted to Mechanism and Machine Theory*, 2014.
- [GDD13] Thibault Gayral, David Daney, and Julien Ducarne. Flexure joints modeling for micrometer accuracy of an active 6-PUS space telescope through experimental calibration. In *ICRA - IEEE Int. Conf. on Robotics and Automation - 2013*, pages 4525–4530, Karlsruhe, Allemagne, 2013.
- [GDM11] Marc Gouttefarde, David Daney, and Jean-Pierre Merlet. Interval-Analysis-Based Determination of the Wrench-Feasible Workspace of Parallel Cable-Driven Robots. *IEEE Transactions on Robotics*, 27(1) :001–013, 2011.
- [GDP06] Carlos Grandón, David Daney, and Yves Papegay. Combining CP and Interval Methods for solving the Direct Kinematic of a Parallel Robot under Uncertainties. In *Workshop of International Conference on Principles and Practice of Constraint Programming (IntCP 06)*, Nantes, France, 2006.
- [GDP⁺07] Carlos Grandón, David Daney, Yves Papegay, Cristina Tavolieri, Erika Ottaviano, and Marco Ceccarelli. Handling Uncertainties with Symbolic/Numerical Solvers for a Class of Parallel Robots. In *World Congress of International Federation for the Theory of Machines and Mechanisms (IFToMM)*, Besançon, France, June 2007.
- [GDRT05] Alexandre Goldsztejn, David Daney, Michel Rueher, and Patrick Taillibert. Modal intervals revisited : a mean-value extension to generalized intervals. In *In International Workshop on Quantification in Constraint Programming (International Conference on Principles and Practice of Constraint Programming, CP-2005)*, Barcelona, Espagne, 2005.
- [GMD06a] Marc Gouttefarde, J-P Merlet, and David Daney. Determination of the wrench-closure workspace of 6-dof parallel cable-driven mechanisms. In *Advances in Robot Kinematics*, pages 315–322. Springer, 2006.
- [GMD06b] Marc Gouttefarde, Jean-Pierre Merlet, and David Daney. Determination of the Wrench-Closure Workspace of 6-DOF Parallel Cable-Driven Mechanisms. In B. Roth J. Lenarcic, editor, *Advances in Robot Kinematics*, pages 315–322, Ljubljana, Slovenia, 2006. Springer.
- [GMD07] Marc Gouttefarde, Jean-Pierre Merlet, and David Daney. Wrench-Feasible Workspace of Parallel Cable-Driven Mechanisms. In *ICRA'07 : International Conference on Robotics and Automation*, pages 1492–1497, Roma, Italy, April 2007.
- [HDT08] Milan Hladik, David Daney, and Elias P. Tsigaridas, P. An Algorithm for the Real Interval Eigenvalue Problem. Rapport de recherche RR-6680, INRIA, 2008.

- [HDT09a] Milan Hladik, David Daney, and Elias Tsigaridas. Bounds on Eigenvalues of Symmetric Interval Matrices. In *Small Workshop on Interval Methods (SWIM)*, Lausanne, Suisse, 2009.
- [HDT09b] Milan Hladik, David Daney, and Elias P. Tsigaridas. A filtering method for the interval eigenvalue problem. Rapport de recherche RR-7057, INRIA, 2009.
- [HDT09c] Milan Hladik, David Daney, and Elias P. Tsigaridas, P. Bounds on eigenvalues and singular values of interval matrices. Rapport de recherche, 2009.
- [HDT10] Milan Hladik, David Daney, and Elias Tsigaridas. Bounds on real eigenvalues and singular values of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4) :2116–2129, 2010.
- [HDT11a] Milan Hladik, David Daney, and Elias Tsigaridas. A filtering method for the interval eigenvalue problem. *Applied Mathematics and Computation*, 217(12) :5236 – 5242, 2011.
- [HDT11b] Milan Hladik, David Daney, and Elias Tsigaridas. Characterizing and approximating eigenvalue sets of symmetric interval matrices. Rapport de recherche RR-7544, INRIA, February 2011.
- [HDT11c] Milan Hladik, David Daney, and Elias Tsigaridas, P. An algorithm for addressing the real interval eigenvalue problem. *Journal of Computational and Applied Mathematics*, 235(8) :2715 – 2730, 2011.
- [HDT11d] Milan Hladik, David Daney, and Elias Tsigaridas, P. Characterizing and Approximating Eigenvalue Sets of Symmetric Interval Matrices. *Computers and Mathematics with Applications*, 62(8) :3152–3163, 2011.
- [HMDB11] Mandar Harshe, Jean-Pierre Merlet, David Daney, and Samir Bennour. A multi-sensors system for human motion measurement : Preliminary setup. In *The 13th World Congress in Mechanism and Machine Science, Guanajuato, Mexico*, Guanajuato, Mexique, June 2011.
- [HOS⁺10] Kanako Harada, Denny Oetomo, Ekawahyu Susilo, Arianna Men-ciassi, David Daney, Jean-Pierre Merlet, and Paolo Dario. A re-configurable modular robotic endoluminal surgical system : vision and preliminary results. *Robotica*, 28 :171–183, 2010.
- [JADG12] Dit Sandretto Julien Alexandre, David Daney, and Marc Goutte-farde. Calibration of a Fully-Constrained Parallel Cable-Driven Robot. In *ROMANSY'2012 : 19th CISM-IFTOMM Symposium on Robot Design, Dynamics, and Control*, pages 1–8, Paris, France, June 2012.
- [KDCP14] Alexandr Klimchik, David Daney, Stéphane Caro, and Anatoly Pashkevich. Geometrical Patterns for Measurement Pose Selection in Calibration of Serial Manipulators. In *Advances in Robot Kinematics, 14th Inter. Symposium*, Ljubljana, Slovénie, 2014.
- [LMR⁺05] Yahia Lebbah, Claude Michel, Michel Rueher, David Daney, and Jean-Pierre Merlet. Efficient and Safe Global Constraints for

- Handling Numerical Constraint Systems. *SIAM Journal on Numerical Analysis*, 42(5) :pp. 2076–2097, 2005.
- [MD01] Jean-Pierre Merlet and David Daney. A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot. In *In 2nd Workshop on Computational Kinematics*, pages 167–176, Seoul, Corée, République De, May 2001.
- [MD05] Jean-Pierre Merlet and David Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 942–947, Barcelona, Espagne, 2005. IEEE.
- [MD06] Jean-Pierre Merlet and David Daney. Legs interference checking of parallel robots over a given workspace or trajectory. In *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, États-Unis, 2006. IEEE.
- [MD07] Jean-Pierre Merlet and David Daney. A New Design for Wire-Driven Parallel Robot. In *Inter. Cong. Design and Modelling of Mechanical Systems (CMSM 2007)*, Monastir, Tunisie, 2007.
- [MD08] J.-P. Merlet and D. Daney. Appropriate Design of Parallel Manipulators. In Lihui Wang and Jeff Xi, editors, *Smart Devices and Machines for Advanced Manufacturing*, pages 1–25. Springer London, 2008.
- [MD09a] Jean-Pierre Merlet and David Daney. Kinematic analysis of a spatial four-wire driven parallel crane without constraining mechanism. In Andrés Kecskeméthy and Andreas Muller, editors, *Computational Kinematics*, pages 1–8. Springer, 2009.
- [MD09b] Jean-Pierre Merlet and David Daney. Kinematic analysis of a spatial four-wire driven parallel crane without constraining mechanism. In Andrés Kecskeméthy and Andreas Muller, editors, *Computational Kinematics*, volume 1, pages 1–8, Duisburg, Allemagne, May 2009. Springer.
- [MD10a] Jean-Pierre Merlet and David Daney. A portable, modular parallel wire crane for rescue operations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2834–2839, Anchorage, États-Unis, 2010. IEEE.
- [MD10b] Jean-Pierre Merlet and David Daney. Robotique : des aides abordables pour nos aînés. *La Recherche Les Cahiers de l’Inria*, (446 novembre 2010), November 2010.
- [MPD00a] Jean-Pierre Merlet, M.-W. Perng, and David Daney. Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator. In J. Lenarčič and M.M. Stanišić, editors, *Advances in Robot Kinematics*, pages 315–322. Springer, 2000.
- [MPD00b] Jean-Pierre Merlet, M.-W. Perng, and David Daney. Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator. In Jadran Lenarcic and Michael M. Stanisic, editors, *International symposium on Advances in robot kinematics*, pages 315–322, Piran, Slovénie, 2000. Springer.

- [ODH⁺09] Denny Oetomo, David Daney, Kanako Harada, Jean-Pierre Merlet, Arianna Menciassi, and Paolo Dario. Topology design of surgical reconfigurable robots by interval analysis. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3085–3090, Kobe, Japon, 2009. IEEE.
- [ODM09] Denny Oetomo, David Daney, and Jean-Pierre Merlet. Design Strategy of Serial Manipulators With Certified Constraint Satisfaction. *IEEE Transactions on Robotics*, 25(1) :1–11, 2009.
- [ODSM08a] D Oetomo, David Daney, Bijan Shirinzadeh, and Jean-Pierre Merlet. An Interval-Based Method for Workspace Analysis of Planar Flexure-Jointed Mechanism. *Journal of Mechanical Design*, 131(1) :011014–011014, 2008.
- [ODSM08b] Denny Oetomo, David Daney, Bijan Shirinzadeh, and Jean-Pierre Merlet. Certified workspace analysis of 3RRR planar parallel flexure mechanism. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3838–3843, Pasadena, États-Unis, 2008. IEEE.
- [OGD⁺14] Celia Ormea, Alain Giboin, David Daney, Jean-Pierre Merlet, and Patrick Rives. Quel dispositif d’assistance aux courses concevoir pour les futures personnes âgées ? In *Handicap 2014 - 8ème Édition*, Versailles, France, 2014.
- [PD07] Yves Papegay and David Daney. Computing with Intervals : the Ultimate Symbolic Computation ? In *Wolfram Technology Conference*, Illinois, États-Unis, 2007.
- [PD08] Yves Papegay, A. and David Daney. UnCertainties, a Package for Interval Analysis. In *9th International Mathematica Symposium*, Maastricht, Pays-Bas, 2008.
- [PDM03a] Yves Papegay, David Daney, and Jean-Pierre Merlet. Parallel Implementation of Interval Analysis for Equations Solving. In Jack Dongarra, Domenico Laforenza, and Salvatore Orlando, editors, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, volume 2840 of *Lecture Notes in Computer Science*, pages 555–559. Springer Berlin Heidelberg, 2003.
- [PDM03b] Yves Papegay, A., David Daney, and Jean-Pierre Merlet. Parallel Implementation of Interval Analysis for Equations Solving. In Jack Dongarra, Domenico Laforenza, and Salvatore Orlando, editors, *10th European PVM/MPI Users’ Group Conference (EuroPVM/MPI03)*, volume 2840 of *Lecture Notes in Computer Science*, pages 555–559, Venice, Italie, 2003. Springer.
- [PMD05] Yves Papegay, Jean-Pierre Merlet, and David Daney. Exact kinematics analysis of Car’s suspension mechanisms using symbolic computation and interval analysis. *Mechanism and Machine Theory*, 40(4) :395 – 413, 2005.
- [PTG⁺14] Fotis Psomopoulos, Emmanouil Tsardoulis, Alexandros Giokas, Cezary Zielinski, Vincent Prunet, Ilias Trochidis, David Daney, Manuel Serrano, Ludovic Courtès, Stratos Arampatzis, and Pericles A Mitkas. RAPP System Architecture. In *Assistance and*

Service Robotics in a Human Environment, IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, United States, September 2014.

[STD13] JulienAlexandredit Sandretto, Gilles Trombettoni, and David Daney. Confirmation of Hypothesis on Cable Properties for Cable-Driven Robots. In Fernando Viadero and Marco Ceccarelli, editors, *New Trends in Mechanism and Machine Science*, volume 7, pages 85–93. Springer Netherlands, 2013.

[YA16] Samer Mohammed Anne Spalanzani Abdelghani Chibani Yacine Amirat, David Daney. Assistance and Service Robotics in a Human Environment. *accepted, Robotics and Autonomous Systems*, 2016.