

ÉCOLE DOCTORALE MSII

ICube (UMR7357)

THÈSE présentée par :

Gauthier HENTZ

Soutenue le : 18 septembre 2017

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Robotique

**Outils numériques pour la conception
de mécanismes**

THÈSE dirigée par :

Mme. CHARPENTIER Isabelle
M. RENAUD Pierre

Chargée de recherche CNRS, Laboratoire ICube
Professeur, INSA de Strasbourg

RAPPORTEURS :

M. CARO Stéphane
M. DAYA EI Mostafa

Chargé de recherche CNRS, Laboratoire LS2N
Professeur, Université de Lorraine

EXAMINATEURS :

M. ANDREFF Nicolas
M. RUBBERT Lennart

Professeur, Université de Franche-Comté
Maître de conférences, INSA de Strasbourg

Remerciements

« L'objectif de toute éducation devrait être de projeter chacun dans l'aventure d'une vie à découvrir, à orienter, à construire »,

Albert Jacquard.

Voici venu le moment de reprendre la plume pour rendre à l'aventure humaine ce qui appartient dans le reste de ce manuscrit à la science. Cette thèse n'aurait pu se dérouler dans de bonnes conditions sans l'attribution d'un financement du ministère de l'enseignement supérieur et de la recherche.

Je tiens tout d'abord à remercier Isabelle Charpentier et Pierre Renaud pour m'avoir proposé ce sujet de thèse et pour leur confiance renouvelée tout au long de nos travaux. Leurs nombreux conseils ont été précieux pour diriger mes recherches dans des directions innovantes. Merci pour votre investissement infailible autant lors des réunions que pour les relectures.

Je remercie les membres de mon jury Nicolas Andreff, Stéphane Caro et El Mostafa Daya pour la relecture attentive du manuscrit, les commentaires et questions qui m'ont permis de prendre du recul sur ce travail interdisciplinaire et d'améliorer encore la clarté du manuscrit.

Un grand merci à Lennart Rubbert pour ses nombreux partages d'expérience allant de la transmission de la vision de la recherche actuelle par des Anciens « Regarde tous ces gens qui tapent à l'ordinateur, rien ne te frappe ? Quand est-ce qu'ils lisent ? ! » à la réalisation de séduisantes figures et des gains de productivité « Le pouvoir de TikZ ». Merci aussi pour les relectures détaillées, les idées, les nombreuses réunions à 4 et les apartées.

Je tiens également à remercier tous les membres de l'équipe AVR pour leur accueil chaleureux, leur disponibilité, et de m'avoir intégré aux réunions d'équipe qui m'ont permis d'avoir un meilleur aperçu du travail d'enseignant-chercheur dans son ensemble.

Les doctorants et affiliés toujours partant pour un Xanadu et plus si affinité : Arnaud, Nadège, Quentin B., Nicole, Cédric, Benoît, Thomas, François, Paolo, Rahim, Markus, Meysam, Yinoussa, Andru, Nicolas, Laure, Laure-Anaïs, Quentin P., Kissoo, Antoine, Oscar,

Raphael, Ajitan. Et les nombreux stagiaires qui ont contribué à la très bonne ambiance du labo.

J'ai eu la chance de profiter de la bonne humeur des colocs successifs : Myriam, Chloé, Olivier, Danaé, Xavier, Flora et Samuel. Mais aussi des amis qui ont partagé ma vie sur Strasbourg : Mic, Pierre, Marieke, Marie et toute la diaspora du 67. Merci à Servann et aux amis aussi doués qu'humbles et inspirants qui m'ont encouragés dans ma voie. Parmi eux, une pensée pour Emilien qui nous a quitté prématurément, et que j'aurais aimé voir à ma place.

Je n'aurais pu me lancer dans l'aventure de la thèse sans le concours de mes parents, soutiens indéfectibles, et qui ont su me transmettre la liberté d'esprit qui m'a mené au choix de la recherche malgré les difficultés. C'est aussi Mylène, Arnaud et sa petite famille, les cousins, Grand-Papa qui m'ont chacun guidé à leur manière sur le chemin de la vie.

Finalement mes pensées vont à Victoire qui a embelli cette dernière année et m'a apporté un soutien précieux.

Sommaire

Liste des figures	vii
Liste des tableaux	ix
Liste des symboles	x
Introduction	1
1 Analyse des mécanismes en robotique	5
1.1 Architecture des mécanismes étudiés	6
1.2 Conception de mécanismes	10
1.3 Détermination d'un espace de travail	13
1.4 Caractérisation des propriétés cinématiques	20
1.5 Evaluation de la sensibilité des performances aux paramètres géométriques	25
1.6 Conclusion	27
2 Méthodes et outils numériques	31
2.1 Résolution de problèmes non-linéaires	32
2.2 Continuation et séries de Taylor	44
2.3 Différentiation automatique de codes informatiques	51
2.4 Diamanlab : une implémentation Matlab de Diamant	59
2.5 Conclusion	66
3 Détermination de l'espace de travail	67
3.1 Etat de l'art	68
3.2 Continuation de haut-degré	81
3.3 Application à un mécanisme plan	86
3.4 Application à un mécanisme spatial	91
3.5 Conclusion	94

4	Analyse de singularités	97
4.1	Définition et classification des singularités	98
4.2	Détermination automatique des singularités	110
4.3	Cas d'application	114
4.4	Conclusion	118
5	Analyse de sensibilité en conception de mécanismes	121
5.1	Etat de l'art	122
5.2	Différentiation de haut-degré	126
5.3	Conclusion	141
	Conclusion générale	143
	Liste des publications	147
	Table des annexes	148
	Bibliographie	171
	Index terminologique	184

Liste des figures

1.1	Exemple d'un robot SCARA.	7
1.2	Exemple d'un robot parallèle plan de type 3- <u>RRR</u>	8
1.3	Représentation d'un processus de conception itérative de mécanisme.	9
1.4	Schéma cinématique d'un mécanisme RRR.	14
1.5	Schéma cinématique du mécanisme 3- <u>RRR</u>	16
1.6	Photographies de deux actionneurs piézoélectriques amplifiés.	26
2.1	Système masse-ressort et branche de solutions d'équilibre.	34
2.2	Un diagramme de bifurcation fictif.	36
2.3	Principe de continuation par prédiction-correction.	38
2.4	Continuation naturelle et point de retournement.	40
2.5	Continuation avec prédiction d'ordre 1 et pilotage par le paramètre de pseudo-longueur d'arc.	41
2.6	Principe de la continuation de haut-degré.	49
2.7	Diagramme de classes du logiciel Diamanlab.	61
2.8	Interface graphique utilisateur de Diamanlab.	65
3.1	Méthode géométrique de détermination de l'espace de travail.	70
3.2	Méthode algébrique de détermination de l'espace de travail.	71
3.3	Méthode par discrétisation pour la détermination de l'espace de travail.	73
3.4	Méthode d'optimisation pour la détermination de l'espace de travail.	74
3.5	Méthode d'analyse par intervalles.	76
3.6	Processus de contraction de boîte de la méthode branch-and-prune.	78
3.7	Méthode de continuation pour la détermination de l'espace de travail.	80
3.8	Méthode de perturbation pour le changement de branches de solutions.	86
3.9	Frontières de l'espace des positions accessibles du mécanisme RRR.	89
3.10	Frontières de l'espace des positions accessibles avec les méthodes Diamant et branch-and-prune.	90

3.11	Représentation du mécanisme Orthoglide et de sa modélisation simplifiée. . .	92
3.12	Représentations des frontières de l'espace de travail de l'Orthoglide.	93
4.1	Distinction de trois ensembles de singularités.	99
4.2	Schéma cinématique d'un mécanisme 4 barres.	100
4.3	Description des 6 types de singularités.	101
4.4	Inclusion des types de singularités RPM et IIM dans les trois ensembles de singularités.	110
4.5	Schéma cinématique du mécanisme double-chaîne.	115
4.6	Ensemble des singularités déterminées du mécanisme double-chaîne.	116
5.1	Sensibilité de haut-degré de l'énergie de contrainte d'une poutre.	126
5.2	Photographies de deux actionneurs piézoélectriques amplifiés.	127
5.3	Modélisation du comportement cinématique d'un amplificateur par un mécanisme Scott-Russell.	129
5.4	Configurations et rapport d'amplification du mécanisme Scott-Russell pour sa course maximale d'actionnement.	131
5.5	Modification du point de fonctionnement du mécanisme pour une perturbation de ses paramètres géométriques.	133
5.6	Configurations nominale et perturbée du mécanisme Scott-Russell pour une perturbation équivalente.	134
5.7	Modification du rapport d'amplification du mécanisme Scott-Russell pour une perturbation.	134
5.8	Sensibilité de la configuration de l'amplificateur <i>APA900</i>	138
5.9	Sensibilité du rapport d'amplification de l'amplificateur <i>APA900</i>	139
5.10	Analyse de sensibilité de la configuration de l'amplificateur <i>APA400</i>	140

Liste des tableaux

2.1	Un ensemble de problèmes de robotique résolus par continuation.	33
2.2	Multiplicité des solutions sur un diagramme de bifurcations.	36
2.3	Un ensemble de problèmes non-linéaires résolus au moyen de méthodes de haut-degré.	45
2.4	Formules de dérivation des fonctions et opérations élémentaires.	53
2.5	Formules de récurrence pour le calcul des dérivées d'ordre élevé.	58
2.6	Attributs de la super-classe @Syst.	62
2.7	Propriétés de la classe @Checkpoint	63
3.1	Comparaison des temps de calcul avec les méthodes Diamant et branch-and-prune.	91
3.2	Positionnement de la méthode par comparaison aux méthodes de détermination de l'espace de travail existantes.	96
4.1	Les 21 classes de singularité.	106
5.1	Données constructeur sur les actionneurs piézoélectriques amplifiés.	136

Liste des symboles

\mathbf{u}	Vecteur des coordonnées opérationnelles
\mathbf{v}	Vecteur des coordonnées actives
\mathbf{w}	Vecteur des coordonnées passives
$\mathbf{y} = [\mathbf{u}^\top, \mathbf{w}^\top]^\top$	Vecteur des coordonnées opérationnelles et passives
$\mathbf{z} = [\mathbf{v}^\top, \mathbf{w}^\top]^\top$	Vecteur des coordonnées actives et passives
$\mathbf{q} = [\mathbf{u}^\top, \mathbf{v}^\top, \mathbf{w}^\top]^\top$	Vecteur de coordonnées géométriques du mécanisme
$\dot{\mathbf{q}} = [\dot{\mathbf{u}}^\top, \dot{\mathbf{v}}^\top, \dot{\mathbf{w}}^\top]^\top$	Vecteur des vitesses articulaires et opérationnelles du mécanisme
\mathcal{Q}	Espace des coordonnées \mathbf{q}
\mathcal{C}	Ensemble des coordonnées \mathbf{q} admissibles
\mathcal{W}	Ensemble des coordonnées \mathbf{u} admissibles
$\mathbf{R}^{\mathcal{C}}(\mathbf{q})$	Relations décrivant la géométrie du mécanisme
$\mathbf{R}_{\mathbf{q}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{q}}$	Matrice jacobienne associée à la fonction vectorielle $\mathbf{R}^{\mathcal{C}}(\mathbf{q})$
$\mathbf{R}_{\mathbf{w}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{w}}$	Sous-matrice de $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$ contenant les colonnes relatives à \mathbf{w}
$\mathbf{R}_{\mathbf{y}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{y}}$	Sous-matrice de $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$ contenant les colonnes relatives à \mathbf{y}
$\mathbf{R}_{\mathbf{z}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{z}}$	Sous-matrice de $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$ contenant les colonnes relatives à \mathbf{z}
\mathcal{S}	Ensemble des configurations singulières
$\mathcal{S}_{\mathbf{y}}$	Ensemble des configurations singulières directes
$\mathcal{S}_{\mathbf{z}}$	Ensemble des configurations singulières inverses
O	Origine du repère de référence lié au bâti

C	Origine du repère opérationnel lié à l'effecteur
N	Nombre de liaisons du mécanisme
m	Mobilité (globale) du mécanisme
n_e	Nombre de relations décrivant la géométrie du mécanisme
n_q	Nombre de coordonnées géométriques
n_u	Nombre de coordonnées opérationnelles
n_v	Nombre de coordonnées actives
$n_w = N - n_v$	Nombre de coordonnées passives
\mathbf{p}	Vecteur des paramètres géométriques
$\delta \mathbf{p}$	Vecteur de perturbation des paramètres
$\delta \mathbf{q}$	Vecteur de perturbation de la configuration
$\delta \mathbf{u}$	Vecteur de perturbation de sortie
$\delta \mathbf{v}$	Vecteur de perturbation d'entrée
$\delta \mathbf{w}$	Vecteur de perturbation intermédiaire
$\mathbf{R}_p^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{p}}$	Matrice jacobienne de $\mathbf{R}^{\mathcal{C}}$ différentié par rapport à \mathbf{p}
$\mathcal{R}(u, \lambda)$	Fonction résiduelle du système $\mathcal{R} = \mathbf{0}$
n_{eq}	Nombre d'équations décrivant un problème paramétrique $\mathcal{R} = \mathbf{0}$
$\mathbf{u} \in \mathbb{R}^{n_{eq}}$	Vecteur d'état du système $\mathcal{R} = \mathbf{0}$
$\lambda \in \mathbb{R}$	Paramètre naturel de continuation du système $\mathcal{R} = \mathbf{0}$
$u_\lambda = \frac{du}{d\lambda}$	Dérivée première de $u(\lambda)$
n_{inc}	Nombre d'inconnues du système $\mathcal{R}(\mathbf{U}) = \mathbf{0}$
$\mathbf{U} = (u, \lambda)$	Vecteur des n_{inc} inconnues du système $\mathcal{R}(\mathbf{U}) = \mathbf{0}$
$a \in \mathbb{R}$	Paramètre de chemin en continuation et variable de développement des séries de Taylor

$\mathbf{U}(a)$	Branche de solutions du système $\mathcal{R}(\mathbf{U}) = \mathbf{0}$
$\mathbf{U}(0)$	Point solution du système $\mathcal{R}(\mathbf{U}) = \mathbf{0}$ pour $a = 0$
$\mathbf{U}^{(k)}(a) = \frac{d^k \mathbf{U}}{da^k}(a)$	Dérivée de $\mathbf{U}(a)$ d'ordre k
$\mathbf{U}_k = \frac{1}{k!} \mathbf{U}^{(k)}(0)$	Coefficient de Taylor d'ordre k en $\mathbf{U}(0)$
K	Ordre de troncature des séries de Taylor
$\mathcal{R}^{(k)}(a) = \frac{d^k (\mathcal{R} \circ \mathbf{U})}{da^k}(a)$	Dérivée de $(\mathcal{R} \circ \mathbf{U})(a)$ d'ordre k
\mathcal{R}_k	Coefficient de Taylor de $(\mathcal{R} \circ \mathbf{U})(a)$ d'ordre k en $\mathbf{U}(0)$
$\{\mathcal{R}_1 \mathbf{U}_1 = Id\}$	Matrice jacobienne de $\mathcal{R}(\mathbf{U})$ contenant les dérivées partielles de \mathcal{R} par rapport à \mathbf{U}
$\{\mathcal{R}_k \mathbf{U}_k = \mathbf{0}\}$	Ensemble des contributions d'ordre $1, \dots, k-1$ et du terme $\frac{\partial^k \mathcal{R}}{\partial \mathbf{U}^k}$ dans \mathcal{R}_k . Dérivée de $\mathcal{R} \circ \mathbf{U}(a)$ d'ordre k évaluée pour $\mathbf{U}_k = \mathbf{0}$.
ε	Tolérance de précision de la méthode Diamant
ε_{nr}	Tolérance de précision de l'algorithme de Newton-Raphson
N_{nr}	Nombre maximal d'étapes de correction de Newton-Raphson autorisé
ε_{rang}	Seuil d'annulation des valeurs propres lors de l'évaluation du rang d'une matrice

Introduction

Les travaux de cette thèse ont été réalisés dans le contexte de la conception de mécanismes pour la robotique de manipulation. Nous nous intéressons en particulier au développement des robots pour l'assistance aux gestes de radiologie interventionnelle et chirurgicaux. Le choix de la robotisation dans ces applications est motivé par la précision et la répétabilité du geste dans le cadre des gestes médicaux et chirurgicaux assistés par ordinateur, ainsi que le couplage possible du robot avec des données d'imagerie.

Dans le domaine médical, l'espace disponible pour intégrer un dispositif est très réduit au regard des contextes rencontrés par exemple en robotique industrielle. De plus, les mouvements à réaliser sont complexes et variés. Par exemple les gestes percutanés en radiologie interventionnelle nécessitent un mouvement pour lequel un point de l'outil doit rester immobile. Ils peuvent nécessiter de grandes amplitudes et un niveau de précision élevé. Une autre nécessité est la compatibilité des robots avec les champs magnétiques et autres rayonnements présents dans les imageurs. Une solution dans ce contexte est l'utilisation de matériaux polymères. Ceux-ci possèdent cependant une rigidité limitée, qui peut être palliée notamment par l'utilisation de mécanismes à architecture parallèle.

Dans ce cadre, le concepteur d'un nouveau dispositif robotique doit développer une expertise sur de nombreux aspects allant de la synthèse structurelle à la synthèse dimensionnelle en passant par la modélisation. Concevoir de manière optimale un système est dès lors très difficile alors que les méthodes existantes sont souvent spécifiques, difficiles à prendre en main, et relativement lentes.

L'architecture mécanique d'un robot manipulateur a pour fonction de transformer le mouvement des actionneurs en un mouvement d'un préhenseur respectant :

1. un nombre, une nature, et une amplitude des mouvements désirés, caractérisés par la notion d'espace de travail, qui doit donc être évalué ;
2. le comportement en vitesse souhaité dans l'espace de travail, avec l'évaluation de la présence de singularités qui doit être réalisée ;

3. les performances en présence d'incertitudes sur les paramètres de conception, caractérisée par la sensibilité à ces paramètres qui doit donc être estimée.

Il est nécessaire pour le concepteur de pouvoir évaluer ces trois éléments de manière efficace. En d'autres termes des outils génériques, rapides et précis nous semblent nécessaires pour l'évaluation de l'espace de travail, de la localisation et la nature des singularités, et de la précision de positionnement.

Un nombre important d'outils spécifiques à une classe ou à une architecture de mécanismes existe pour l'évaluation d'une ou plusieurs de ces caractéristiques. Ils requièrent dans l'immense majorité des cas un niveau d'expertise élevé. Nous proposons dans ce travail de thèse de progresser sur ce point, en s'appuyant sur des outils numériques performants aujourd'hui disponibles.

Méthodes numériques en robotique Les méthodes numériques permettent a priori une résolution des problèmes pour tout mécanisme. Plusieurs approches existent, comme les méthodes d'échantillonnage, les méthodes d'analyse par intervalles ou la continuation. Elles requièrent en général une puissance de calcul importante, en particulier pour des problèmes de dimension élevée. C'est pourquoi les problèmes sont souvent traités de manière locale plutôt que globale, par exemple en déterminant l'espace de travail par ses frontières.

Nous proposons dans cette thèse de nous appuyer sur la continuation numérique pour l'analyse de mécanismes. La continuation numérique consiste à déterminer une courbe solution de proche en proche à partir d'un point solution initial. Les méthodes de continuation ont été exploitées pour la résolution de problèmes complexes de conception de mécanismes par suivi des solutions à partir d'un problème plus simple dont la solution est connue [Wampler et Sommese, 2013 ; McCarthy, 2013]. La méthode de prédiction-correction utilisée par Haug *et al.* [1996] pour la résolution de systèmes d'équations non-linéaires, résulte en l'obtention de branches de solutions sous forme discrète. Ces méthodes ont pour avantage d'être génériques, plus précises que les méthodes d'analyse par intervalles, plus rapides que les méthodes d'échantillonnage, mais sont limitées à la résolution de problèmes à une dimension.

Les méthodes de continuation ont également gagné en performance et en fiabilité. D'une part, les algorithmes intègrent des fonctionnalités de détection d'intersections de branches et de changement de branches de solutions, ce qui permet la résolution de problèmes dont l'ensemble de solutions est de structure complexe. D'autre part, la continuation de haut-degré permet de déterminer les branches de solution plus efficacement, et d'obtenir une représentation continue au moyen d'une résolution sous forme de polynômes de Taylor. Cette dernière est au centre de notre travail.

La continuation de haut-degré requiert la différentiation du modèle à résoudre à haut-degré, ce qui peut s'avérer fastidieux. Cette difficulté est levée avec l'usage de la différentiation automatique, une méthode algorithmique qui permet l'obtention des dérivées exactes de fonctions implémentées comme un code informatique. Elle exploite pour cela les règles de dérivation des fonctions et opérations élémentaires. La différentiation automatique a atteint un degré de maturité permettant son utilisation hors de la communauté des numériciens. Nous proposons de l'exploiter pour la continuation mais aussi pour extraire simplement les informations requises pour l'analyse de singularités et de la sensibilité.

Diamant est une méthode de continuation de haut-degré exploitant la différentiation automatique de haut-degré pour une résolution générique de tout système d'équations non-linéaires. Diamanlab implémente en langage Matlab la méthode Diamant. Ainsi l'utilisateur de Matlab peut implémenter les équations du problème à résoudre sans pré-traitement particulier, et réaliser une continuation interactive en un clic. L'outil Diamanlab est donc une base intéressante pour assister dans la résolution de problèmes de conception de mécanismes.

Contributions de la thèse

Associer continuation numérique, différentiation automatique et évaluation de mécanismes pour la conception requiert une étude méthodologique et de validation. Elle est conduite en considérant successivement les 3 volets que sont la détermination de l'espace de travail, l'analyse de singularités et l'analyse de sensibilité. Nos contributions sont présentées dans ce manuscrit au travers de 5 chapitres.

Le chapitre 1 introduit le cadre de notre approche de conception qui permet une **analyse des mécanismes** générique. Les caractéristiques principales **en robotique** que nous avons choisies d'évaluer sont alors détaillées. Une méthode de modélisation et un formalisme sont ensuite proposés. Ils permettent d'envisager une résolution générique des problèmes.

Le chapitre 2 présente les éléments essentiels des **méthodes numériques** sélectionnées. Les **outils numériques** sont présentés et leur choix justifié. Les propositions de couplage dans le cadre de l'analyse des mécanismes sont finalement introduites.

Le chapitre 3 expose les méthodes existantes pour la **détermination de l'espace de travail** afin d'illustrer la pertinence de notre approche numérique. Le formalisme permettant une approche de résolution générale est présenté. Notre méthode de détermination est décrite, puis son efficacité évaluée au regard de l'état de l'art.

Le chapitre 4 expose le formalisme d'**analyse de singularités** permettant une détection complète et une classification détaillée des singularités. Les méthodes de résolution exploitant ce formalisme sont présentées et évaluées au regard de nos besoins. Notre méthode de détermination des singularités est ensuite détaillée et évaluée sur un mécanisme de référence pour l'analyse de singularités.

Le chapitre 5 présente un état de l'art des approches d'**analyse de sensibilité**. Notre approche reposant sur l'étude du modèle géométrique et la différentiation automatique de haut-degré est ensuite développée. Sa pertinence pour l'estimation de la sensibilité des performances d'un mécanisme au voisinage de ses configurations singulières est évaluée sur un exemple simple issu d'une problématique industrielle.

En **conclusion**, nous discutons l'intérêt et les limitations de notre approche et des outils d'aide à la conception développés pour les trois problématiques sus-citées. Des perspectives sont formulées concernant le choix des outils numériques et les améliorations qui pourraient leur être apportées.

Chapitre 1

Analyse des mécanismes en robotique

« The past century shows that our ability to analyze and design mechanisms and robotic systems of increasing complexity has depended on our ability to derive and solve the associated increasingly complex polynomial systems. », J. Michael McCarthy.

Sommaire

1.1	Architecture des mécanismes étudiés	6
1.2	Conception de mécanismes	10
1.2.1	Démarche de conception	10
1.2.2	Caractéristiques d'intérêt	11
1.2.3	Prise en compte des erreurs de fabrication	12
1.3	Détermination d'un espace de travail	13
1.3.1	Cas d'un mécanisme sériel	13
1.3.2	Cas d'un mécanisme parallèle	15
1.3.3	Approche adoptée	18
1.4	Caractérisation des propriétés cinématiques	20
1.4.1	Difficultés de détection	20
1.4.2	Difficultés de classification	23
1.4.3	Approche adoptée	24
1.5	Evaluation de la sensibilité des performances aux paramètres géométriques	25
1.5.1	Difficultés d'évaluation	26
1.5.2	Approche proposée	27
1.6	Conclusion	27

Comme introduit précédemment, les travaux de cette thèse se placent dans le contexte de la conception de mécanismes pour la robotique de manipulation.

Un système robotique est constitué d'un mécanisme, d'actionneurs éventuellement associés à une chaîne de transmission de puissance, de capteurs, et d'un système de commande du mouvement. Nous nous intéressons à la conception robotique en contexte médical. Les contraintes de conception du mécanisme à la base du système robotique sont alors très restrictives. La compacité des dispositifs est indispensable pour ne pas gêner le praticien, et pour être compatible avec la présence par exemple d'imageurs médicaux. Une problématique centrale concerne alors la conception du mécanisme intégré au dispositif robotique. Pour la décrire, il nous faut tout d'abord introduire précisément l'architecture et les caractéristiques des mécanismes étudiées.

Ce chapitre présente le formalisme nécessaire à l'étude des caractéristiques considérés. Cette présentation est nécessaire à la compréhension des propositions faites dans cette thèse. Elle met en avant les difficultés d'évaluation existantes en soulignant les problématiques pour envisager leur traitement systématique avec les outils numériques proposés en chapitre 2.

Nous présentons en section 1.1 le contexte médical à l'origine de nos travaux. Nous caractérisons alors par leur architecture les mécanismes étudiés. En section 1.2, nous présentons la démarche de conception de mécanismes généralement considérée. Puis nous montrons le besoin d'outils d'évaluation des performances et précisons les caractéristiques essentielles étudiées dans la suite. Nous exposons alors en sections 1.3 et 1.4 l'approche adoptée pour la caractérisation de l'espace de travail d'un mécanisme. Enfin nous montrons en section 1.5 l'intérêt de la prise en compte des incertitudes sur la géométrie du mécanisme au plus tôt dans la démarche de conception.

1.1 Architecture des mécanismes étudiés

De manière générale, un *mécanisme*¹ est un système de corps conçu pour convertir des mouvements de corps en des mouvements contraints d'autres corps. Il s'agit de piloter les mouvements d'un corps auquel va par exemple être attaché un outil médical par le biais du mécanisme. Il est composé :

- d'une *chaîne cinématique* constituée d'un ensemble de corps mobiles appelés *membres* et reliés entre eux par des *liaisons* autorisant un ou plusieurs mouvements relatifs ;

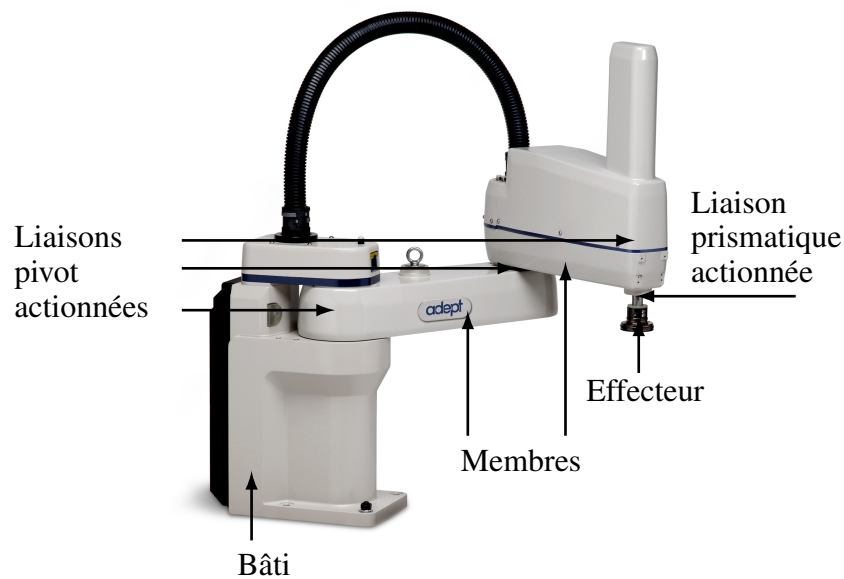
1. Les termes introduits sont indiqués en italique. La terminologie utilisée correspond par défaut à celle proposée par l'IFTOMM dans Breteler [2011] et les références incluses. Dans le cas contraire une référence est précisée. S'il s'agit d'un choix original de l'auteur la définition est mise en exergue.

- d'un *bâti* correspondant au corps fixe ;
- d'un *effecteur* correspondant au membre auquel un outil ou un préhenseur est attaché ;
- de liaisons actionnées.

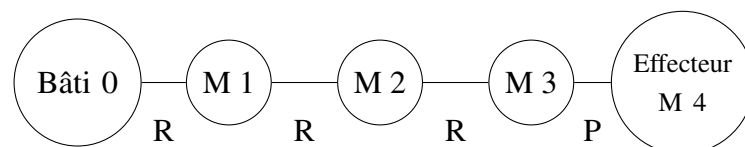
Par convention une liaison est usuellement désignée par une lettre associée au mouvement qu'elle permet [Merlet, 2006]. Les liaisons qui nous intéressent sont les liaisons de type :

- *prismatique* (P) autorisant une translation selon un axe ;
- *pivot* (R pour « revolute ») autorisant une rotation selon un axe ;
- *cardan* (U pour « universal ») autorisant deux rotations selon deux axes concourants ;
- *sphérique* (S) autorisant trois rotations selon trois axes concourants.

Le nombre de membres et de liaisons d'un mécanisme, leurs interconnexions, ainsi que le type de liaisons définissent l'*architecture* de ce mécanisme. Les mécanismes peuvent être classés selon leur architecture en 2 familles : mécanismes sériels et parallèles.



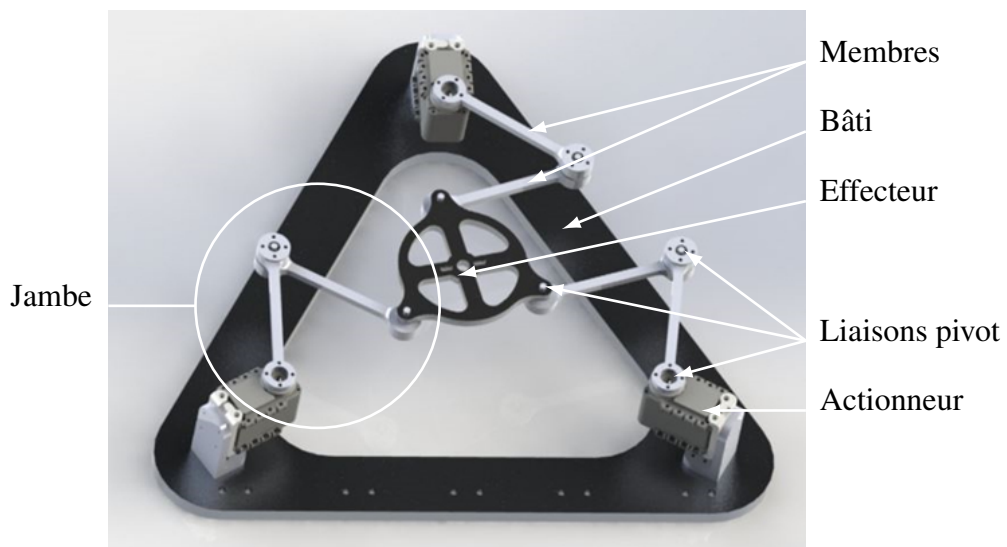
(a) Image d'un robot SCARA [Adept, 2017]



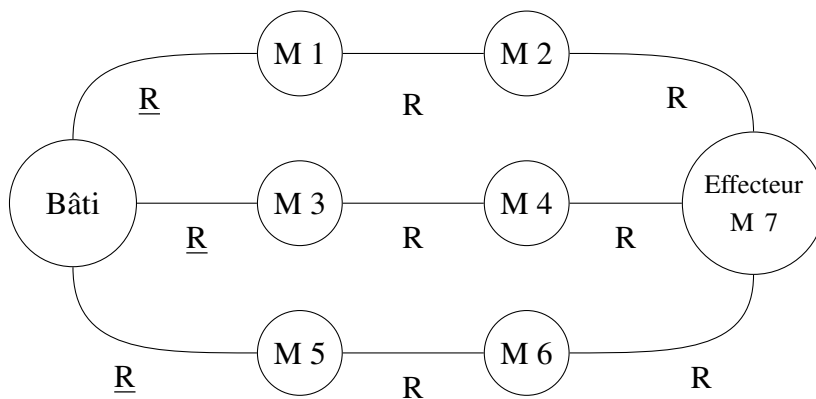
(b) Graphe des liaisons

FIGURE 1.1 – Exemple d'un robot SCARA.

Un *mécanisme sériel* est constitué d'une chaîne cinématique ouverte, *i.e.* l'effecteur est lié au bâti par une unique chaîne. Toutes les liaisons d'un mécanisme sériel sont actionnées afin de contraindre l'état de la chaîne cinématique. L'image et le graphe des liaisons décrivant l'architecture d'un robot SCARA sont fournis en figure 1.1. Sur le graphe des liaisons, les corps sont représentés par des cercles reliés par des lignes qui symbolisent les liaisons. Les quatre liaisons présentes pour cet exemple sont actionnées et permettent de piloter les quatre degrés de liberté de la chaîne cinématique. L'architecture d'une telle chaîne cinématique ouverte est communément décrite par l'énumération des liaisons la composant du bâti vers l'effecteur. Ainsi le robot SCARA est basé sur un mécanisme RRRP.



(a) Photographie d'un robot 3-RRR [TaarLab, 2017]



(b) Graphe des liaisons

FIGURE 1.2 – Exemple d'un robot parallèle plan de type 3-RRR.

Un *mécanisme parallèle* [Merlet, 2006] est constitué d'une chaîne cinématique fermée dans laquelle chacun des membres est connecté à au moins deux autres membres. Il n'est pas

nécessaire d'actionner toutes les liaisons afin de contraindre l'état de la chaîne cinématique. Un exemple est introduit en figure 1.2 par son image et son graphe des liaisons décrivant l'architecture. On observe la présence d'une chaîne cinématique fermée, qui peut être décomposée en trois chaînes cinématiques simples indépendantes reliant l'effecteur au bâti. Pour une de ces chaînes cinématiques simples, on désigne souvent par *jambe* l'ensemble des liaisons et membres connectant le bâti à l'effecteur. En terme de représentation, un mécanisme parallèle peut être décrit d'une manière similaire à celle d'un mécanisme sériel, *i.e.* chaque jambe est décrite par l'énumération des liaisons la composant, en partant du bâti et en allant vers l'effecteur. Les liaisons actionnées, dites actives, sont soulignées pour les identifier. Ainsi le mécanisme du robot de la figure 1.2, composé de trois jambes identiques RRR en parallèle, actionnées à leur base, est désigné par 3-RRR.

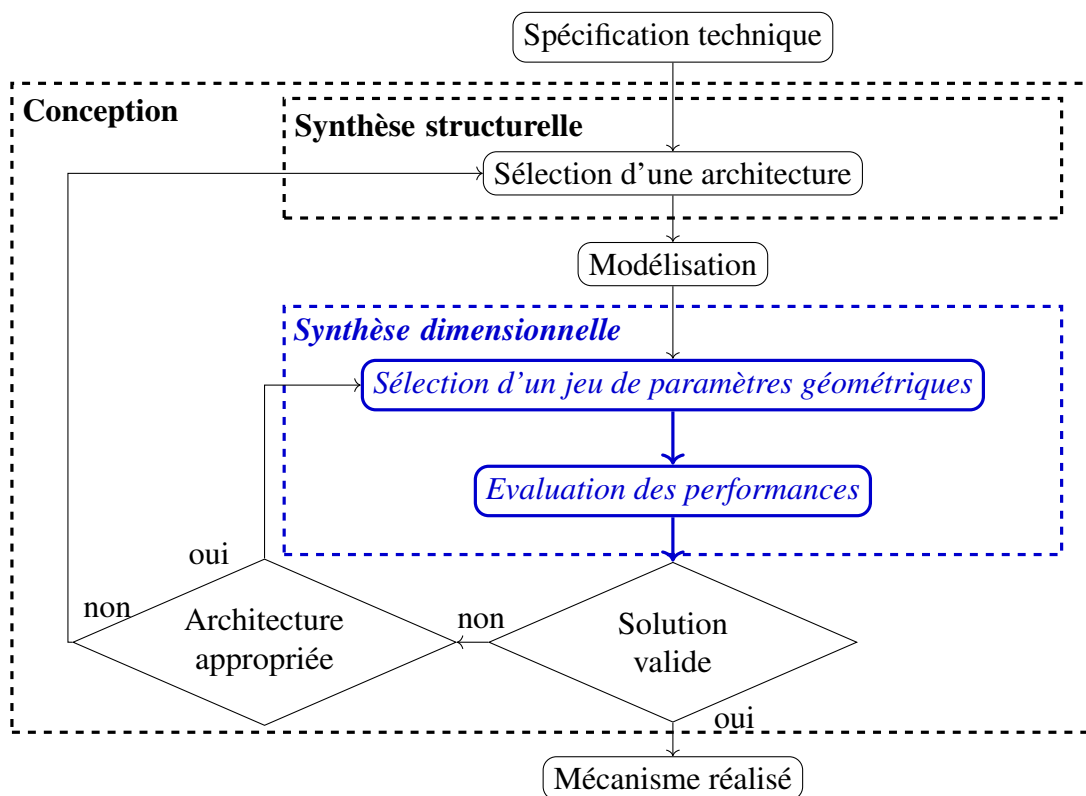


FIGURE 1.3 – Représentation d'un processus de conception itérative de mécanisme. Le cadre de cette thèse est indiqué en **bleu** et *italique*.

1.2 Conception de mécanismes

1.2.1 Démarche de conception

L'approche la plus couramment exploitée pour concevoir un mécanisme est de procéder en deux temps [Merlet, 2005b ; Gogu, 2008], comme décrit en figure 1.3.

Dans un premier temps, le type d'architecture est choisi dans la phase dite de **synthèse structurelle** [Merlet, 2006 - Chapitre 2 ; Gogu, 2008]. La sélection de l'architecture s'opère par comparaison de solutions préliminaires dont la génération dépend de l'expertise du concepteur. On obtient alors un graphe de liaisons caractérisant le mécanisme choisi.

Dans un second temps, la géométrie du mécanisme, *i.e.* les dimensions des membres et leurs configurations sont sélectionnées pendant la phase de **synthèse dimensionnelle** [Merlet et Daney, 2005 ; Merlet, 2006 - Chapitre 11]. Chacune des deux phases est généralement réalisée de manière itérative, comme illustré par les bouclages en figure 1.3, la validité de la solution étant évaluée en la confrontant à la spécification technique.

Des outils génériques permettant l'évaluation des solutions préliminaires sont nécessaires [Merlet, 2002a]. De tels outils doivent permettre au concepteur d'étudier une large famille de mécanismes sans expertise spécifique. On voit donc l'importance de la *généricité* d'une *approche de conception* et des *outils d'aide à la conception* de mécanismes.

Le processus de **conception** des mécanismes implique une étape d'évaluation des performances du mécanisme choisi, afin de pouvoir les comparer à la spécification technique. Concrètement, on évalue un critère de performance dont la valeur est jugée acceptable au regard d'une valeur seuil. Si les performances obtenues ne sont pas satisfaisantes, il appartient au concepteur de modifier les paramètres géométriques, voire de changer l'architecture du mécanisme pour se rapprocher d'une solution valide, comme indiqué en figure 1.3. Qu'elle soit réalisée à la main ou par une méthode d'optimisation, la synthèse implique un nombre important d'évaluations des solutions candidates. La durée du processus de conception en est fortement impactée. La *rapidité d'évaluation* des performances est donc un besoin important pour toute méthode visant à estimer les *caractéristiques d'un mécanisme*.

Lors de la conception d'un mécanisme, le concepteur s'intéresse en premier lieu aux performances pour une géométrie nominale du mécanisme, *i.e.* pour un modèle donné, et un jeu de paramètres géométriques fixé. Les performances sont bien sûr susceptibles de varier après production et intégration du mécanisme. En particulier la valeur réelle des paramètres géométriques diffère de la valeur nominale des paramètres à cause des erreurs de fabrication. L'impact peut être évalué en aval de la phase de conception, par des évaluations expérimentales. La réalisation de prototypes est cependant coûteuse en temps et en ressources, et ne permet que difficilement d'envisager une optimisation du niveau de performances

réelles : on va simplement vérifier que les performances restent acceptables. Une méthode permettant d'anticiper l'*impact de la variabilité des paramètres sur les performances* de la solution synthétisée est donc aussi d'un grand intérêt, *i.e.* une méthode d'analyse de sensibilité.

En conclusion, trois besoins fondamentaux ressortent pour une approche d'évaluation de mécanismes en phase de synthèse dimensionnelle et son implémentation :

- la *généricité* de l'évaluation, étant donnée la variété des architectures potentiellement étudiées lors de la synthèse structurelle,
- l'*efficacité* d'évaluation, *i.e.* un compromis acceptable entre rapidité et précision du processus itératif de conception,
- l'évaluation de la *robustesse des performances* en présence d'erreurs de fabrication.

1.2.2 Caractéristiques d'intérêt

Dans le contexte de la robotique médicale, un mécanisme est utilisé à des fins de manipulation. On s'intéresse donc à la pose de l'effecteur qui peut être décrite par un jeu de *coordonnées opérationnelles* souvent exploité [Khalil et Dombre, 2002]. La *configuration* du mécanisme désigne l'état de sa chaîne cinématique, *i.e.* la pose de chacun de ses membres [Bonev, 2002 ; Merlet, 2007 ; Gogu, 2008]. La configuration est liée elle-même à l'état des liaisons actionnées, caractérisé par un jeu de *coordonnées actives*. Ceci est exprimé par des relations de fermeture géométrique qui modélisent les contraintes imposées par les liaisons sur la configuration du mécanisme.

L'évaluation des performances d'un mécanisme concerne plusieurs *caractéristiques*. Étant donné le champ applicatif visé, nous nous focalisons sur deux d'entre elles.

Espace de travail Dans le contexte médical, l'espace disponible pour le robot et pour la manipulation de l'effecteur est limité. On cherche dès lors à évaluer le rapport entre le volume que peut parcourir l'effecteur et le volume occupé par le robot. Ce critère de performance permet de comparer différentes solutions selon ce que l'on peut considérer être un critère de compacité. La première caractéristique d'intérêt est donc l'espace de travail.

L'espace de travail [Chedmail *et al.*, 1998 - p.28 ; Bohigas *et al.*, 2012a] est défini comme l'ensemble des poses accessibles par l'effecteur. Une pose est accessible s'il existe un état des liaisons correspondant tel que l'ensemble des contraintes imposées par les liaisons aux membres soit satisfait. Une méthode par discrétisation [Bonev et Ryu, 2001 ; Castelli *et al.*, 2008] ou une méthode d'échantillonnage aléatoire [Rastegar et Perel, 1990 ; Alciatore et Ng, 1994], dite de Monte-Carlo, sont des approches génériques largement exploitées pour évaluer cet espace. Elles consistent à échantillonner l'ensemble des états des liaisons admissibles,

et à évaluer la pose de l'effecteur pour chacun des états considérés. L'espace à évaluer peut atteindre jusqu'à une dimension 6. Une évaluation précise de l'espace de travail implique donc un nombre très important d'évaluations. Si une expression du modèle géométrique direct n'est pas disponible, il faut parfois de plus effectuer une résolution numérique pour chaque état évalué. Ces approches sont donc très coûteuses en temps de calcul.

Le premier axe de travail de cette thèse développé au chapitre 3 vise à proposer une approche alternative pour la résolution des problèmes associés, détaillée en section 1.3.

Lieu des configurations singulières Un nombre donné de mouvements indépendants réalisables est nécessaire pour une tâche de manipulation dans l'espace de travail requis. Or le nombre de mouvements réalisables par l'effecteur d'un mécanisme peut varier dans l'espace de travail. En configuration singulière, des mouvements peuvent ne plus être effectués ou contrôlés. La seconde caractéristique d'intérêt est donc le lieu des configurations singulières pour lesquelles deux options se présentent. La première, la plus courante en phase de conception, consiste à éviter la présence de configurations singulières au sein de l'espace de travail [Arsenault et Boudreau, 2004 ; Merlet, 2007 ; Bohigas *et al.*, 2013]. La seconde, rencontrée dans des situations de conception plus spécifiques, consiste à exploiter des configurations singulières, par exemple pour amplifier des mouvements [Rubbert *et al.*, 2012, 2014]. Durant la synthèse dimensionnelle, il est donc nécessaire dans un cas comme dans l'autre de déterminer avec précision le lieu des configurations singulières et les mouvements obtenus en singularité.

Dans cette thèse, le deuxième axe de travail, développé au chapitre 4, concerne une démarche de localisation et d'analyse des configurations singulières. Les difficultés d'un traitement exhaustif et l'approche proposée sont introduites en section 1.4.

1.2.3 Prise en compte des erreurs de fabrication

La robotique apporte en contexte médical la précision qui peut manquer à des gestes manuels. Cette précision est notamment liée à la précision du mécanisme, qu'il s'agit donc d'assurer. Elle est en particulier liée aux incertitudes de fabrication, qui impactent également les caractéristiques d'intérêt décrites précédemment, à savoir l'espace de travail et le lieu des configurations singulières.

Comme notre contribution porte sur la conception préliminaire de mécanismes, et en particulier sur la phase de synthèse dimensionnelle, les sources d'incertitude concernent principalement les tolérances sur les paramètres géométriques du mécanisme. Nous avons identifié en section 1.2.1 le besoin d'une évaluation de la robustesse des performances en présence d'erreurs de fabrication. La conception robuste vise à optimiser les paramètres

de conception d'un système et de son procédé de fabrication afin de réduire la sensibilité de ses performances aux incertitudes [Caro, 2004]. Une évaluation de l'impact d'erreurs géométriques sur les caractéristiques du mécanisme revient à analyser la sensibilité de la configuration du mécanisme aux variations des paramètres géométriques. L'information de sensibilité peut au delà être très utile dans le processus itératif de conception. D'abord elle indique comment modifier le jeu de paramètres géométriques pour améliorer les performances de la solution et constitue ainsi une assistance à la synthèse dimensionnelle. Ensuite elle peut être exploitée dans le cadre d'une optimisation des paramètres. La qualité et la rapidité de la sélection de paramètres adaptés dépend donc de la précision de la méthode d'analyse de sensibilité des performances utilisée.

Le troisième axe de travail, développé au chapitre 5, vise à proposer une nouvelle approche pour répondre au besoin d'évaluation précise de l'impact d'erreurs de fabrication sur les caractéristiques d'intérêt. Les difficultés d'évaluation dans les situations d'intérêt que sont les configurations singulières et l'approche proposée sont introduites en section 1.5.

1.3 Détermination d'un espace de travail

La première caractéristique d'un mécanisme étudiée est son espace de travail. Deux problèmes ayant trait à la notion d'espace de travail peuvent se poser durant la synthèse d'un mécanisme. Le premier est d'évaluer la capacité à parcourir un espace de travail prescrit, *i.e.* le domaine d'évolution de la pose de l'effecteur est donné. Le deuxième consiste à générer l'espace de travail correspondant à une architecture pour un jeu de paramètres géométriques et des intervalles d'évolution des liaisons actionnées fixés. L'un ou l'autre problème pose des difficultés, comme exposé dans la suite.

1.3.1 Cas d'un mécanisme sériel

En prenant l'exemple du mécanisme RRR comme présenté en figure 1.4(a), les difficultés qui se posent avec un mécanisme sériel peuvent être mises en avant. Il s'agit d'un mécanisme sériel plan composé de trois liaisons pivot actives reliant trois membres de longueurs notées respectivement L_1 , L_2 et L_3 , en partant du bâti. Le premier membre pivote autour du bâti avec un angle α , le deuxième membre tourne autour du premier avec un angle β , et le troisième autour du deuxième avec un angle γ .

Un repère de référence $(O, \mathbf{x}_O, \mathbf{y}_O)$ est lié au bâti. Un repère $(C, \mathbf{x}_C, \mathbf{y}_C)$ est lié à l'effecteur du mécanisme et désigné comme le repère opérationnel. Sa position est paramétrée par les

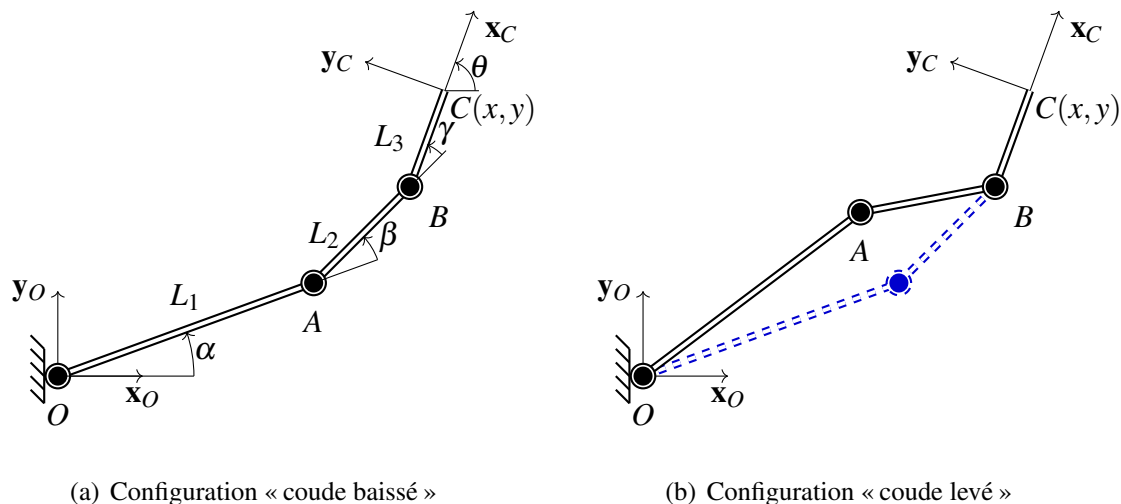


FIGURE 1.4 – Schéma cinématique d’un mécanisme RRR. Deux configurations existent pour une pose donnée (x, y, θ) de l’effecteur.

coordonnées x et y et son orientation selon l’axe \mathbf{z}_0 est paramétrée par la coordonnée θ par rapport au repère $(O, \mathbf{x}_O, \mathbf{y}_O)$.

Afin de résoudre les problèmes de génération de l’espace de travail et d’analyse d’un espace de travail prescrit, un modèle exprimant les contraintes imposées sur les membres par les liaisons est nécessaire. La projection des relations vectorielles de fermeture de la chaîne cinématique complétée des relations d’angle permet d’obtenir le système d’équations non-linéaires :

$$\begin{cases} x = L_1 \cos(\alpha) + L_2 \cos(\alpha + \beta) + L_3 \cos(\alpha + \beta + \gamma), \\ y = L_1 \sin(\alpha) + L_2 \sin(\alpha + \beta) + L_3 \sin(\alpha + \beta + \gamma), \\ \theta = \alpha + \beta + \gamma. \end{cases} \quad (1.1)$$

Ces relations expriment la pose de l’effecteur $X = (x, y, \theta)$ en fonction de l’état des liaisons actionnées $q = (\alpha, \beta, \gamma)$, et correspondent au modèle géométrique direct (MGD) [Merlet, 2006 ; Gogu, 2008] de la forme $X = \mathbf{f}(q)$. D’autres représentations vectorielles ou matricielles d’un mécanisme sont également possibles, par exemple décrites dans [Zlatanov 1998 ; Bonev 2002 ; Merlet 2007 ; Gogu 2008] L’entrée et la sortie du MGD sont respectivement les coordonnées actives et opérationnelles.

On obtient donc sans manipulation particulière une expression analytique permettant la résolution directe du problème de génération de l'espace de travail. Par exemple une méthode de discrétisation peut alors être directement exploitée.

La recherche de l'état des liaisons actionnées pour une pose de l'effecteur prescrite pose en revanche des difficultés. La résolution de ce problème implique de connaître une relation de la forme $q = \mathbf{g}(X)$. Il s'agit du *modèle géométrique inverse* (MGI) exprimant les coordonnées actives en fonction des coordonnées opérationnelles. L'entrée et la sortie du MGI sont alors respectivement les coordonnées opérationnelles et actives. L'établissement du MGI pour le mécanisme RRR, est un problème complexe consistant en l'expression des variables q en fonction des variables X , par exemple à partir des expressions du MGD décrites par le système (1.1). Par ailleurs l'observation de la figure 1.4 montre que, pour la position du point C et l'orientation de l'effecteur représentées, deux états des liaisons actionnées sont possibles, correspondant aux configurations « coude baissé » et « coude levé ».

Deux difficultés se posent donc dans l'étude d'espace de travail prescrit d'un mécanisme sériel :

- L'obtention d'une expression analytique $q = \mathbf{g}(X)$ n'est pas assurée,
- Le MGI peut être non-unique, donc une unique relation $q = \mathbf{g}(X)$ ne peut toujours être écrite : à un jeu de coordonnées opérationnelles peuvent correspondre plusieurs jeux de coordonnées actives appelés postures [Chablat et Wenger, 1998].

1.3.2 Cas d'un mécanisme parallèle

En prenant l'exemple d'un mécanisme 3-RRR présenté en figure 1.5(a), les difficultés qui se posent avec un mécanisme parallèle peuvent être mises en avant. Elles sont duales de celles qui se posent avec un mécanisme sériel. Le mécanisme 3-RRR consiste en une chaîne cinématique fermée. Les points fixes A_1 , A_2 et A_3 appartiennent au bâti, et l'effecteur de centre C est formé par le triangle $C_1C_2C_3$. La chaîne cinématique peut être vue comme comprenant 3 sous-chaînes cinématiques ouvertes identiques, appelées jambes, reliant le bâti et l'effecteur. Chaque jambe $A_iB_iC_i$ est composée de 2 membres connectés en série par 3 liaisons pivot. La pose de l'effecteur est donnée par la position (x, y) du point C et son orientation θ dans le repère de référence $(O, \mathbf{x}, \mathbf{y})$. Les liaisons actionnées sont situées à la base du mécanisme, *i.e.* aux points A_1 , A_2 et A_3 . L'état des liaisons actionnées est décrit par les coordonnées actives $(\alpha_1, \alpha_2, \alpha_3)$.

Afin de résoudre les problèmes de génération de l'espace de travail et d'analyse d'un espace de travail prescrit, on doit disposer d'un modèle exprimant les contraintes imposées sur les membres par les liaisons.

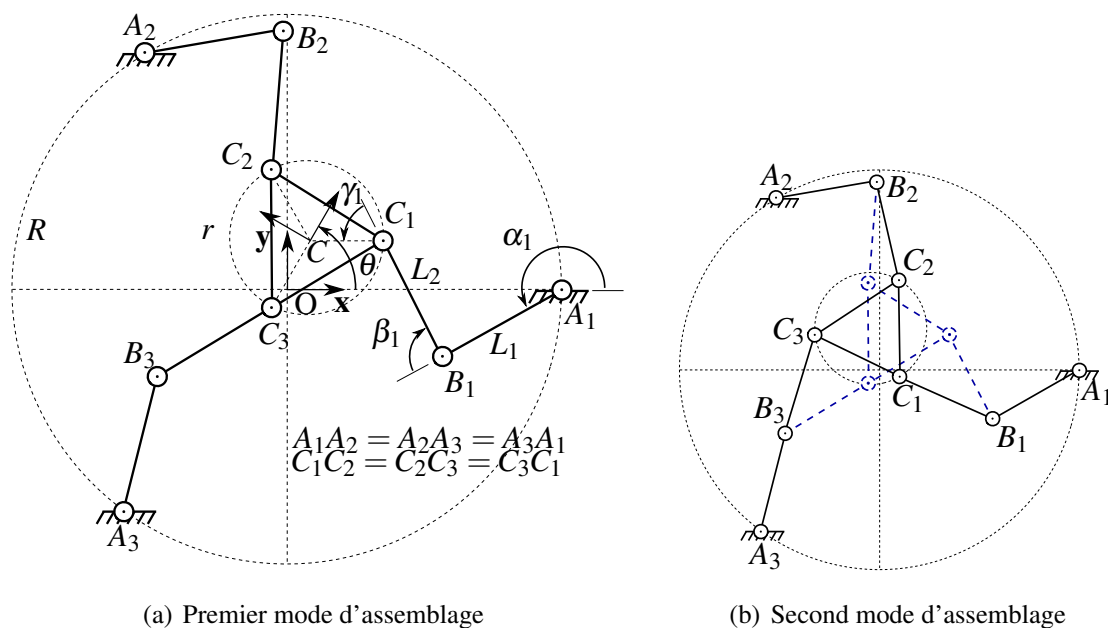


FIGURE 1.5 – Schéma cinématique du mécanisme 3-RRR.

Pour cela nous pouvons exprimer une relation entre les coordonnées actives et les coordonnées opérationnelles. De simples considérations géométriques permettent alors l'obtention du MGI [Merlet, 2006 - Chapitre 3]. L'établissement du MGD reste en général un problème ouvert [Merlet, 2006 - Chapitre 4], rendant très complexe la détermination de l'espace de travail plus complexe.

Comme pour le mécanisme sériel précédent, on peut établir des relations vectorielles de fermeture de chaîne cinématique pour une jambe du mécanisme parallèle :

$$\mathbf{OA}_i + \mathbf{A}_i\mathbf{B}_i + \mathbf{B}_i\mathbf{C}_i + \mathbf{C}_i\mathbf{C} - \mathbf{OC} = \mathbf{0}, \forall i \quad (1.2)$$

Si l'on procède par projection de ces relations et en ajoutant la relation d'angles pour chaque jambe du mécanisme on a :

$$\left\{ \begin{array}{l} R + L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1 + \beta_1) + r \cos(\theta + \frac{2\pi}{3}) - x = 0, \\ 0 + L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1 + \beta_1) + r \sin(\theta + \frac{2\pi}{3}) - y = 0, \\ \alpha_1 + \beta_1 + \gamma_1 - \frac{2\pi}{3} - \theta = 0, \\ -\frac{R}{2} + L_1 \cos(\alpha_2) + L_2 \cos(\alpha_2 + \beta_2) + r \cos(\theta - \frac{2\pi}{3}) - x = 0, \\ \frac{\sqrt{3}}{2}R + L_1 \sin(\alpha_2) + L_2 \sin(\alpha_2 + \beta_2) + r \sin(\theta - \frac{2\pi}{3}) - y = 0, \\ \alpha_2 + \beta_2 + \gamma_2 + \frac{2\pi}{3} - \theta = 0, \\ -\frac{R}{2} + L_1 \cos(\alpha_3) + L_2 \cos(\alpha_3 + \beta_3) + r \cos(\theta) - x = 0, \\ -\frac{\sqrt{3}}{2}R + L_1 \sin(\alpha_3) + L_2 \sin(\alpha_3 + \beta_3) + r \sin(\theta) - y = 0, \\ \alpha_3 + \beta_3 + \gamma_3 - \theta = 0. \end{array} \right. \quad (1.3)$$

Ce système d'équations n'exprime explicitement ni les coordonnées actives $(\alpha_1, \alpha_2, \alpha_3)$ en fonction des coordonnées opérationnelles (x, y, θ) , ni les coordonnées opérationnelles en fonction des coordonnées actives. Il ne s'agit donc ni d'un MGI ni d'un MGD, et le problème d'analyse de l'espace de travail ne peut être traité directement avec ce système. Ce système d'équations exprimant une relation implicite entre toutes les coordonnées opérationnelles, actives et passives du mécanisme, est appelé *modèle géométrique implicite* [Andreff, 2006 ; Hubert, 2010]. Il s'établit facilement pour tout mécanisme. Il fait intervenir les coordonnées passives $(\beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3)$, ce qui s'avère intéressant dans le cadre de notre étude. Supposons en effet que l'on fixe les coordonnées actives $(\alpha_1, \alpha_2, \alpha_3)$. L'observation de la figure 1.5(b) montre qu'il est possible d'assembler les seconds membres de chaque jambe et l'effecteur en différentes poses, appelées modes d'assemblage [Chablat et Wenger, 1998]. Un jeu de coordonnées actives peut donc être lié à plusieurs jeux de coordonnées opérationnelles via le MGD. Pour déterminer la pose de l'effecteur sans ambiguïté, il est nécessaire de fixer un nombre supérieur de coordonnées. Par exemple on peut fixer la valeur de la première coordonnée passive de chaque jambe $(\beta_1, \beta_2, \beta_3)$, car ces variables interviennent directement dans le modèle implicite.

En résumé, le modèle implicite ne requiert pas de manipulation particulière pour son écriture. La conservation des coordonnées passives $(\beta_1, \beta_2, \beta_3)$ dans la représentation du mécanisme permet de plus de traiter de manière explicite la multiplicité des solutions aux MGD et MGI.

Les exemples traités d'un mécanisme sériel et d'un mécanisme parallèle mettent en lumière certaines difficultés des problèmes d'analyse de l'espace de travail. L'obtention des MGD et MGI pour un mécanisme quelconque reste un problème ouvert [Merlet, 2002b].

1.3.3 Approche adoptée

De nombreuses approches ont été proposées par ailleurs pour la détermination de l'espace de travail des mécanismes. On trouve des méthodes algébriques et géométriques qui sont efficaces pour des architectures et des formulations très spécifiques [Gosselin, 1990 ; Merlet, 1995 ; Bonev et Gosselin, 2006].

Zlatanov *et al.* [1995] décrivent l'état du mécanisme par un jeu contenant les coordonnées de toutes les liaisons. Haug *et al.* [1996] et Bohigas *et al.* [2012b] utilisent un jeu minimal de *coordonnées* dites *généralisées* pour décrire l'état du mécanisme de manière unique. Haug *et al.* [1996] désignent par \mathbf{q} le jeu de coordonnées généralisées qui est décomposé en coordonnées active, passives et opérationnelles.

Chacun de ces jeux contient a minima les coordonnées actives du mécanisme. L'établissement d'un jeu minimal peut nécessiter une réflexion sur la sélection de coordonnées passives et constitue alors une tâche supplémentaire. C'est pourquoi nous choisissons de décrire l'état du mécanisme par toutes les coordonnées des liaisons.

Nous choisissons par ailleurs d'étudier le modèle géométrique implicite tel que défini dans [Hubert, 2010 - p.11] car la présence des coordonnées passives dans ce modèle permet la prise en compte des solutions multiples. Par ailleurs, dans le cadre de la conception des mécanismes, la problématique en termes d'analyse de l'espace de travail ne se limite pas à la relation entre la pose de l'effecteur et l'état des liaisons actionnées car il peut être nécessaire de tenir compte :

- des différents choix de mode d'actionnement,
- des collisions entre les membres [Merlet, 1995] ou avec des obstacles [FarzanehKaloorazi *et al.*, 2016 ; Wenger et Chedmail, 1991],
- du débattement des liaisons actives [Rastegar et Perel, 1990].
- du débattement des liaisons passives [Merlet, 1995].

La représentation de Zlatanov *et al.* contenant toutes les coordonnées des liaisons prend en compte de telles contraintes. En plus de la représentation de l'état du mécanisme, l'étude de l'espace de travail requiert une représentation explicite de l'état de l'effecteur. Nous proposons d'utiliser un jeu de variables que nous désignons comme un jeu de coordonnées géométriques permettant de caractériser la pose de l'effecteur et l'état de la chaîne cinématique.

Définition 1 : Un jeu de *coordonnées géométriques* est un jeu comprenant les coordonnées de toutes les liaisons d'un mécanisme et ses coordonnées opérationnelles.

Suivant la notation introduite par Haug *et al.* [1996] :

- les coordonnées opérationnelles forment un vecteur \mathbf{u} dans la suite,
- les coordonnées actives sont notées \mathbf{v} dans la suite,
- le jeu de coordonnées géométriques est décrit par un vecteur $\mathbf{q} = [\mathbf{u}^\top, \mathbf{v}^\top, \mathbf{w}^\top]^\top$, où le jeu de coordonnées passives \mathbf{w} est constitué des coordonnées restantes.

A des fins de concision des notations, on construit le vecteur $\mathbf{y} = [\mathbf{u}^\top, \mathbf{w}^\top]^\top$ contenant toutes les coordonnées géométriques à l'exception des coordonnées actives. De même, on construit le vecteur $\mathbf{z} = [\mathbf{v}^\top, \mathbf{w}^\top]^\top$ contenant toutes les coordonnées géométriques à l'exception des coordonnées opérationnelles.

Cette représentation offre deux avantages notables pour le concepteur. D'une part le jeu de coordonnées géométriques peut être construit en prenant toutes les coordonnées apparaissant dans les équations de fermeture de chaîne, voir section 1.3.2. D'autre part elle permet de résoudre les problèmes de conception de mécanismes traités dans cette thèse, notamment d'analyse de singularités (voir chapitre 4).

Ce jeu ne doit pas nécessairement contenir les coordonnées de toutes les liaisons. En pratique, il suffit que le jeu \mathbf{z} contienne suffisamment de coordonnées pour décrire l'état du mécanisme, *i.e.* qu'il constitue a minima un jeu de coordonnées généralisées au sens de Haug *et al.* [1996] et Bohigas *et al.* [2012b].

Le modèle géométrique implicite est défini comme « une relation entre la pose, la configuration articulaire et la géométrie du robot » [Andreff, 2006 - p.21]. L'introduction du jeu de coordonnées géométriques permet d'écrire le modèle géométrique implicite sous une forme générique.

Définition 2 : Le *modèle géométrique implicite* :

$$\mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \mathbf{0}, \quad (1.4)$$

est le modèle liant les coordonnées géométriques.

La représentation de la configuration d'un mécanisme par ses coordonnées géométriques mène à un traitement des problèmes d'analyse de l'espace de travail différent de celui introduit en section 1.3. En effet, on cherche à déterminer l'ensemble des configurations \mathbf{q} du mécanisme admissibles au lieu de l'ensemble des poses accessibles par l'effecteur.

L'ensemble des configurations admissibles, défini initialement par Chedmail *et al.* [1998] et Bohigas *et al.* [2012b] pour un jeu de coordonnées généralisées, est ici défini comme suit :

Définition 3 : L'ensemble des *configurations admissibles* \mathcal{C} satisfaisant :

$$\mathcal{C} = \left\{ \mathbf{q} \in \mathcal{Q} \mid \mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \mathbf{0} \right\}, \quad (1.5)$$

où \mathcal{Q} est l'espace des coordonnées \mathbf{q} , est appelé l'*espace des configurations*.

L'espace de travail tel qu'introduit en section 1.2.2 est une restriction de l'ensemble des configurations \mathbf{q} admissibles aux coordonnées opérationnelles \mathbf{u} . Il est défini comme suit :

Définition 4 : L'*espace de travail* \mathcal{W} est l'ensemble des poses accessibles par l'effecteur [Chedmail *et al.*, 1998 - p.28, Bohigas *et al.*, 2012a] :

$$\mathcal{W} = \left\{ \mathbf{u} \mid \exists \mathbf{z} \text{ tel que } \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) = \mathbf{0} \right\}, \quad (1.6)$$

où \mathbf{u} désigne les coordonnées opérationnelles, \mathbf{z} les coordonnées des liaisons actives et passives et $\mathbf{R}^{\mathcal{C}} = \mathbf{0}$ est le modèle géométrique implicite (1.4).

La détermination de l'espace des configurations admissibles est donc un problème plus général que la détermination de l'espace de travail.

1.4 Caractérisation des propriétés cinématiques

La seconde caractéristique d'intérêt est le lieu des configurations singulières. Contrairement à l'espace de travail, il s'agit d'une caractéristique locale, *i.e.* dépendante de la configuration du mécanisme. La détermination d'une configuration singulière peut poser des difficultés dans la détection de la configuration et la classification des mouvements en singularité, comme exposé ci-après.

1.4.1 Difficultés de détection

Pour un mécanisme sériel, toutes les liaisons sont actionnées, donc $\mathbf{q} = [\mathbf{u}^T, \mathbf{v}^T]^T$. Le MGD est obtenu directement par écriture des équations de fermeture vectorielle :

$$\mathbf{u} = \mathbf{R}^{MGD}(\mathbf{v}) \quad (1.7)$$

Par différentiation du modèle géométrique par rapport au temps, on obtient le modèle cinématique direct :

$$\dot{\mathbf{u}} = \mathbf{R}_{\mathbf{v}}^{MGD}(\mathbf{v})\dot{\mathbf{v}}. \quad (1.8)$$

Les configurations singulières sont définies comme des configurations pour lesquelles la jacobienne $\mathbf{R}_{\mathbf{v}}^{MGD}(\mathbf{v}) = \frac{\partial \mathbf{R}^{MGD}}{\partial \mathbf{v}}(\mathbf{v})$ connaît une perte de rang.

Reprenons l'exemple du mécanisme RRR présenté en figure 1.4. Son modèle géométrique direct est donné en équation (1.1). Afin de déterminer l'ensemble de ces configurations singulières, on construit la jacobienne $\mathbf{R}_{\mathbf{v}}^{MGD}(\mathbf{v})$ par différentiation du modèle géométrique par rapport aux coordonnées actives $\mathbf{v} = [\alpha, \beta, \gamma]^T$:

$$\mathbf{R}_{\mathbf{v}}^{MGD}(\mathbf{v}) = \begin{bmatrix} -L_1 \sin(\alpha) - L_2 \sin(\alpha + \beta) - L_3 \sin(\alpha + \beta + \gamma) & -L_2 \sin(\alpha + \beta) - L_3 \sin(\alpha + \beta + \gamma) & -L_3 \sin(\alpha + \beta + \gamma) \\ L_1 \cos(\alpha) + L_2 \cos(\alpha + \beta) + L_3 \cos(\alpha + \beta + \gamma) & L_2 \cos(\alpha + \beta) + L_3 \cos(\alpha + \beta + \gamma) & L_3 \cos(\alpha + \beta + \gamma) \\ 1 & 1 & 1 \end{bmatrix}. \quad (1.9)$$

Le déterminant de la jacobienne $\mathbf{R}_{\mathbf{v}}^{MGD}$ est :

$$\det(\mathbf{R}_{\mathbf{v}}^{MGD}) = -\sin(\alpha) \cos(\alpha + \beta) L_1 L_2 + \cos(\alpha) \sin(\alpha + \beta) L_1 L_2. \quad (1.10)$$

Les configurations singulières sont les configurations telles que :

$$\det(\mathbf{R}_{\mathbf{v}}^{MGD}) = 0 \Leftrightarrow \beta = 0 + 2k\pi. \quad (1.11)$$

Cette condition algébrique peut être interprétée sous forme d'une condition géométrique : les configurations singulières correspondent aux situations où les deux premiers membres du mécanisme sont alignés.

Les configurations singulières d'un mécanisme sériel correspondent par ailleurs à une perte locale de mouvements réalisables par l'effecteur : il existe une direction dans laquelle l'effecteur ne peut pas se déplacer. Il s'agit de la seule modification des mouvements réalisables existante pour les mécanismes sériels. Ainsi, l'interprétation de la modification de comportement de l'effecteur en configuration singulière pour un mécanisme sériel est triviale, tout comme donc le travail de classification des mouvements en singularités.

Pour l'exemple traité ici, on peut établir des conditions géométriques simples et il est aisé de se représenter l'ensemble des configurations singulières où les deux premiers membres du mécanisme sont alignés par observation de la figure 1.4. Mais dans le cas général, la détection de toutes les configurations singulières à partir de conditions algébriques ou géométriques est un problème complexe [Merlet, 2006 - Chapitre 6 ; Merlet, 2007].

Une configuration singulière est une configuration admissible, donc solution du système (1.1). La matrice jacobienne $\mathbf{R}_{\mathbf{v}}^{MGD}$ présente par ailleurs une perte de rang, ce qui peut

s'écrire :

$$\begin{cases} [\mathbf{R}_v^{MGD}(\mathbf{v})]^\top \xi = \mathbf{0}, \\ \xi \neq \mathbf{0}. \end{cases} \quad (1.12)$$

On peut interpréter la condition implicite de perte de rang en remarquant que l'équation (1.12) satisfait le modèle cinématique (1.8) avec $\dot{\mathbf{u}} = \mathbf{0}$ et $\dot{\mathbf{v}} = \xi$. Un vecteur ξ satisfaisant les conditions de perte de rang correspond en fait à un jeu de vitesses d'entrée $\dot{\mathbf{v}} \neq \mathbf{0}$, pour lequel les vitesses de sortie sont nulles $\dot{\mathbf{u}} = \mathbf{0}$. Autrement dit, s'il existe un vecteur ξ satisfaisant au système (1.12) en une configuration (\mathbf{u}, \mathbf{v}) , alors celui-ci indique une direction de l'espace des coordonnées actives dans laquelle un actionnement des liaisons n'entraîne aucun mouvement de l'effecteur.

En prenant de manière arbitraire ξ unitaire, une configuration singulière d'un mécanisme sériel est telle que le *système étendu* [Seydel 1979 et Haug *et al.* 1996] d'équations suivant est satisfait :

$$\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{v}, \xi) = \begin{bmatrix} \mathbf{R}^{MGD}(\mathbf{v}) - \mathbf{u} \\ [\mathbf{R}_v^{MGD}(\mathbf{v})]^\top \xi \\ \xi^\top \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (1.13)$$

où $\mathcal{S}_z = \{\mathbf{v} | \exists (\mathbf{u}, \xi) \text{ tel que } \mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{v}, \xi) = \mathbf{0}\}$ désigne l'ensemble des singularités inverses. Dans le cas du mécanisme RRR, on a :

$$\begin{bmatrix} \mathbf{R}^{MGD}(\mathbf{v}) - \mathbf{u} \\ [\mathbf{R}_v^{MGD}(\mathbf{v})]^\top \xi \\ \xi^\top \xi - 1 \end{bmatrix} = \begin{bmatrix} L_1 c_\alpha + L_2 c_{\alpha\beta} + L_3 c_{\alpha\beta\gamma} - x \\ L_1 s_\alpha + L_2 s_{\alpha\beta} + L_3 s_{\alpha\beta\gamma} - y \\ \alpha + \beta + \gamma - \theta \\ (-L_1 s_\alpha - L_2 s_{\alpha\beta} - L_3 s_{\alpha\beta\gamma}) \xi_1 + (L_1 c_\alpha + L_2 c_{\alpha\beta} + L_3 c_{\alpha\beta\gamma}) \xi_2 + \xi_3 \\ (-L_2 s_{\alpha\beta} - L_3 s_{\alpha\beta\gamma}) \xi_1 + (L_2 c_{\alpha\beta} + L_3 c_{\alpha\beta\gamma}) \xi_2 + \xi_3 \\ (-L_3 s_{\alpha\beta\gamma}) \xi_1 + (L_3 c_{\alpha\beta\gamma}) \xi_2 + \xi_3 \\ \xi_1^2 + \xi_2^2 + \xi_3^2 - 1 \end{bmatrix}, \quad (1.14)$$

où $c_\alpha = \cos(\alpha)$, $c_{\alpha\beta} = \cos(\alpha + \beta)$, $c_{\alpha\beta\gamma} = \cos(\alpha + \beta + \gamma)$, $s_\alpha = \sin(\alpha)$, $s_{\alpha\beta} = \sin(\alpha + \beta)$, $s_{\alpha\beta\gamma} = \sin(\alpha + \beta + \gamma)$, $\mathbf{u} = [x, y, \theta]^\top$, $\mathbf{v} = [\alpha, \beta, \gamma]^\top$ et $\xi = [\xi_1, \xi_2, \xi_3]^\top$. La dérivation du modèle géométrique écrit sous forme d'un système d'équations non-linéaires et l'écriture du système étendu peuvent nécessiter un effort important. Ces difficultés peuvent être levées par l'outil numérique qu'est la différentiation automatique, présentée au chapitre 2 et exploitée en section 3.2.1.

L'équation (1.14) montre que le système étendu est un système d'équations non-linéaires dépendant des variables $(\mathbf{u}, \mathbf{v}, \xi)$. Finalement, la détection des configurations singulières d'un mécanisme sériel consiste en la résolution d'un système d'équations non-linéaires de la

forme $\mathbf{R}^{\mathcal{L}}(\mathbf{u}, \mathbf{v}, \xi) = \mathbf{0}$ (1.13), que l'on peut résoudre de manière générique au même titre que le modèle géométrique implicite (1.4), ce que nous exploiterons en section 3.2.2.

1.4.2 Difficultés de classification

Dans le cas général, on dispose d'un modèle géométrique implicite (1.4) $\mathbf{R}^{\mathcal{L}}(\mathbf{q}) = \mathbf{0}$. Après dérivation en temps on obtient le modèle cinématique implicite reliant les vitesses des liaisons actives, des liaisons passives et de l'effecteur :

$$\frac{\partial \mathbf{R}^{\mathcal{L}}(\mathbf{q})}{\partial t} = \mathbf{R}_q^{\mathcal{L}} \dot{\mathbf{q}} = \mathbf{R}_u^{\mathcal{L}} \dot{\mathbf{u}} + \mathbf{R}_v^{\mathcal{L}} \dot{\mathbf{v}} + \mathbf{R}_w^{\mathcal{L}} \dot{\mathbf{w}} = \mathbf{0}, \quad (1.15)$$

où les matrices-colonne $\dot{\mathbf{u}}$, $\dot{\mathbf{v}}$ et $\dot{\mathbf{w}}$ décrivent respectivement les vitesses de l'effecteur, des liaisons actives et des liaisons passives, et sont respectivement appelées *vitesses de sortie*, *vitesses d'entrée* et *vitesses intermédiaires*. La matrice $\mathbf{R}_q^{\mathcal{L}} = \frac{\partial \mathbf{R}^{\mathcal{L}}}{\partial \mathbf{q}}$ est la jacobienne du modèle géométrique implicite dérivé par rapport aux coordonnées géométriques, et $\mathbf{R}_u^{\mathcal{L}} = \frac{\partial \mathbf{R}^{\mathcal{L}}}{\partial \mathbf{u}}$, $\mathbf{R}_v^{\mathcal{L}} = \frac{\partial \mathbf{R}^{\mathcal{L}}}{\partial \mathbf{v}}$ et $\mathbf{R}_w^{\mathcal{L}} = \frac{\partial \mathbf{R}^{\mathcal{L}}}{\partial \mathbf{w}}$ sont les matrices jacobiennes partielles du modèle géométrique implicite différencié respectivement par rapport aux coordonnées opérationnelles, actives et passives. Les pertes de rang de chacune de ces matrices définissent des types de configurations singulières, impliquant différentes modifications du nombre et de la nature des mouvements réalisables. La classification de [Gosselin et Angeles, 1990], fondée sur les relations entre coordonnées opérationnelles et actives, distingue trois types de configurations singulières selon la perte du rang de :

- $\mathbf{R}_v^{\mathcal{L}}$: singularités de « type I » ;
- $\mathbf{R}_u^{\mathcal{L}}$: singularités de « type II » ;
- $\mathbf{R}_v^{\mathcal{L}}$ et $\mathbf{R}_u^{\mathcal{L}}$: singularités de « type III ».

L'absence des coordonnées passives dans l'analyse cinématique d'un mécanisme parallèle résulte en une détermination incomplète des configurations singulières, comme montré dans Zlatanov *et al.* [1995, 2002]. Hunt [1978] et Sugimoto *et al.* [1982] analysent la chaîne cinématique dans son ensemble, mais sans la considérer comme un dispositif de manipulation, il en résulte également une détermination incomplète des configurations singulières.

La classification de toutes les configurations singulières pour un mécanisme quelconque a été proposée par Zlatanov *et al.* [1995]. L'état du mécanisme est alors représenté par un jeu de coordonnées $\mathbf{q} = [\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T]^T$ [Bohigas *et al.*, 2012b]. Ici, le jeu $\mathbf{z} = [\mathbf{v}^T, \mathbf{w}^T]^T$ décrit l'état de la chaîne cinématique, et le jeu \mathbf{u} décrit l'état de l'effecteur. Les singularités sont ensuite classifiées selon des conditions de perte de rang des 4 matrices jacobiennes $\mathbf{R}_q^{\mathcal{L}}$, $\mathbf{R}_w^{\mathcal{L}}$, $\mathbf{R}_y^{\mathcal{L}} = [\mathbf{R}_u^{\mathcal{L}}, \mathbf{R}_w^{\mathcal{L}}]$ et $\mathbf{R}_z^{\mathcal{L}} = [\mathbf{R}_v^{\mathcal{L}}, \mathbf{R}_w^{\mathcal{L}}]$.

En conclusion, la classification de tous les types de configurations singulières existantes pour un mécanisme quelconque est une tâche complexe car :

- la classification des configurations singulières en trois types est incomplète : elle ne permet ni de déterminer toutes les configurations singulières, ni de distinguer tous les types de modification de comportement pouvant apparaître ;
- la classification exhaustive décrite par Zlatanov *et al.* [1995] comporte 6 types de modification de comportement résultant en 21 classes de configurations singulières ;
- le calcul des 4 jacobiennes nécessaires peut être fastidieux et complexe analytiquement ou numériquement ;
- dans le cas d'une évaluation numérique, les configurations singulières doivent être déterminées très précisément pour pouvoir évaluer le rang de ces jacobiennes [Bohigas *et al.*, 2012b].

1.4.3 Approche adoptée

Nous proposons d'exploiter l'approche de Zlatanov *et al.* [1995] pour la détection et la classification exhaustives des configurations singulières d'un mécanisme. Dans notre approche, une configuration est décrite par un jeu de coordonnées géométriques \mathbf{q} .

Définition 5 : En une configuration \mathbf{q} , un *mouvement instantané* $\dot{\mathbf{q}}$ est admissible si et seulement si il satisfait aux équations cinématiques [Zlatanov *et al.*, 1995] :

$$\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{q}} = \mathbf{0}, \quad (1.16)$$

avec $\dot{\mathbf{q}} = [\dot{\mathbf{u}}^T, \dot{\mathbf{v}}^T, \dot{\mathbf{w}}^T]^T$ le vecteur des mouvements instantanés et $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{q}}$ est la matrice jacobienne des fonctions du système (1.4) différenciée par rapport aux coordonnées géométriques \mathbf{q} . Le système d'équations linéaires (1.16) est appelé *modèle cinématique implicite* en référence à Andreff [2006 - p.23] qui considérait un système issu de la dérivation temporelle « d'une relation entre la pose, la configuration articulaire et la géométrie du robot ».

Définition 6 : Une *singularité inverse* est une configuration \mathbf{q} pour laquelle la jacobienne partielle $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{z}}$ est singulière [Bohigas, 2013]. On note \mathcal{S}_z l'ensemble des singularités

inverses. Les configurations $\mathbf{q} \in \mathcal{S}_z \subset \mathcal{C}$ sont solutions du système étendu :

$$\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ [\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z})]^\top \cdot \xi \\ \xi^\top \cdot \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (1.17)$$

où ξ est un vecteur unitaire et les deux derniers termes du système (1.17) expriment la perte de rang de la jacobienne $\mathbf{R}_z^{\mathcal{C}}$.

Définition 7 : Une *singularité directe* est une configuration \mathbf{q} pour laquelle $\mathbf{R}_y^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{y}}$ est singulière [Bohigas, 2013]. On note \mathcal{S}_y l'ensemble des singularités directes. Les configurations $\mathbf{q} \in \mathcal{S}_y \subset \mathcal{C}$ sont solutions de :

$$\mathbf{R}^{\mathcal{S}_y}(\mathbf{v}, \mathbf{y}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{v}, \mathbf{y}) \\ [\mathbf{R}_y^{\mathcal{C}}(\mathbf{v}, \mathbf{y})]^\top \cdot \xi \\ \xi^\top \cdot \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (1.18)$$

où ξ est un vecteur unitaire.

Une configuration singulière est définie comme une configuration où les matrices $\mathbf{R}_z^{\mathcal{C}}$ ou $\mathbf{R}_y^{\mathcal{C}}$ sont singulières [Zlatanov *et al.*, 1995]. La détection de la totalité des configurations singulières du mécanisme passe par l'évaluation de $\mathcal{S} = \mathcal{S}_z \cup \mathcal{S}_y$ via la résolution de deux systèmes d'équations non-linéaires (1.17) et (1.18) [Zlatanov *et al.*, 1998 ; Bohigas, 2013]. Un état de l'art des méthodes permettant une analyse détaillée des singularités est proposé en section 4.1.

1.5 Evaluation de la sensibilité des performances aux paramètres géométriques

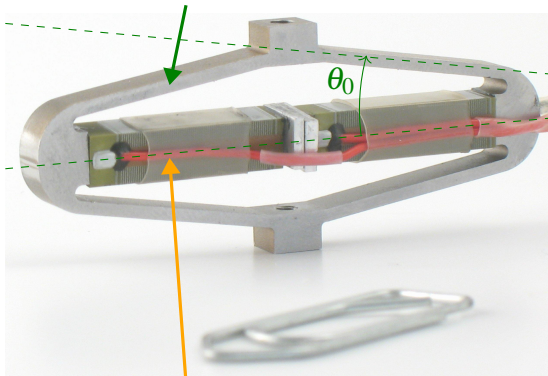
Le troisième volet considère l'évaluation de la robustesse des performances en présence d'incertitude sur les paramètres géométriques du mécanisme. Nous nous plaçons dans une approche de conception robuste, et cherchons à évaluer de manière efficace la sensibilité des caractéristiques du mécanisme aux paramètres géométriques dès la phase de conception préliminaire.

1.5.1 Difficultés d'évaluation

Les caractéristiques que sont l'espace de travail et le lieu des configurations singulières peuvent être très sensibles aux erreurs de fabrication et d'assemblage du mécanisme. Nous illustrons ici les difficultés rencontrées dans l'analyse de sensibilité avec l'exemple d'un mécanisme d'amplification.

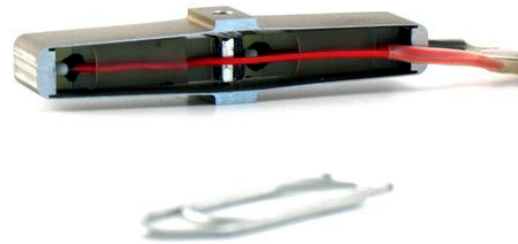
Les actionneurs piézoélectriques amplifiés de CEDRAT TECHNOLOGIES reposent sur la combinaison d'un actionneur à base de céramiques piézoélectriques multicouches empilées désigné comme la céramique et d'un mécanisme d'amplification, voir figure 1.6(a). La fonction principale du mécanisme d'amplification est de transformer le mouvement

Mécanisme d'amplification



Actionneur piézoélectrique

(a)



(b)

FIGURE 1.6 – Photographies de deux actionneurs (a) APA150M et (b) APA900M de CEDRAT TECHNOLOGIES [CEDRAT, 2017].

d'extension de la céramique en un mouvement amplifié de contraction du mécanisme selon un axe perpendiculaire. Ce mécanisme est étudié en section 5.2 et son fonctionnement est alors détaillé. Les performances de l'actionneur piézoélectrique amplifié en terme de course maximale dépendent du rapport d'amplification du mécanisme. Le rapport d'amplification dépend de la configuration du mécanisme exploitée, caractérisée par l'angle θ_0 (Figure 1.6(a)). Par exemple le mécanisme APA150M conçu avec un angle $\theta_0 = 9^\circ$ (figure 1.6(a)) possède un rapport d'amplification plus faible que celui du mécanisme APA900M conçu avec un angle $\theta_0 = 2^\circ$ (figure 1.6(b)). Nous souhaitons pouvoir évaluer l'impact d'une erreur de fabrication portant sur l'angle θ_0 sur le rapport d'amplification et les performances de l'APA en matière de course maximale.

Une approche classique est la linéarisation d'un modèle exprimant l'indice de performance étudié en fonction du paramètre incertain. La résolution du modèle linéarisé pour de

petites variations des paramètres permet alors d'établir les variations de performance. Cette approche échoue à proximité des configurations singulières [Tannous *et al.*, 2014], comme nous le détaillerons en section 5.1.1.

Dans l'exemple de l'actionneur piézoélectrique amplifié, le mécanisme est utilisé à proximité de sa configuration singulière, c'est à dire dans la situation où le rapport d'amplification est maximal. Une analyse de sensibilité par linéarisation n'est donc pas applicable.

1.5.2 Approche proposée

Nous proposons de développer une analyse de sensibilité de haut-degré pour permettre une analyse de sensibilité y compris à proximité des configurations singulières d'un mécanisme. Dans cette approche, le modèle exprimant la caractéristique en fonction du paramètre est alors dérivé à haut-degré par rapport au paramètre à étudier. L'usage de la différentiation automatique de haut-degré permet d'automatiser l'approche. La sensibilité est établie comme un polynôme de Taylor exprimant la dépendance non-linéaire entre la performance et le paramètre. La méthode échoue également en configuration singulière car la fonction n'est pas dérivable mais elle permet d'obtenir des résultats très précis y compris à proximité étroite des singularités, comme montré en section 5.2.4.

1.6 Conclusion

Dans ce chapitre, nous avons présenté la démarche de conception de mécanismes adoptée pour identifier le besoin pour une approche de conception générique et des outils d'aide à la conception permettant une évaluation automatisée et efficace des performances et de leur robustesse. L'objectif de la thèse est d'accélérer le processus itératif de conception de mécanismes. L'évaluation des performances est faite à travers l'analyse des caractéristiques du mécanisme. Dans la robotique médicale, il est tout d'abord nécessaire de déterminer l'espace de travail d'un mécanisme ainsi que ses singularités.

Nous avons identifié des points communs dans ces deux problématiques qui permettent d'envisager leur traitement systématique avec les outils numériques proposés en chapitre 2. Notre approche de détermination de l'espace de travail et des singularités repose sur le modèle géométrique implicite qui peut être établi pour tout mécanisme. La différentiation de ce modèle et la résolution de systèmes d'équations non-linéaires permet ensuite de formuler et résoudre de manière systématique les deux problèmes.

Seuls des outils numériques permettent d'envisager une résolution générique des problèmes sous la formulation proposée avec un niveau élevé d'automatisation. Nous avons vu

que les méthodes de discrétisation résultent en des temps de calcul élevés qui sont caractéristiques des méthodes numériques. Or nous souhaitons développer une méthode à la fois générique et efficace, compromis qui semble difficile à réaliser. La résolution des problèmes de conception nécessite par ailleurs la différentiation du modèle géométrique implicite et la résolution de différents systèmes d'équations non-linéaires. Pour notre approche numérique il est donc nécessaire de disposer du modèle géométrique implicite sous forme de code informatique dans un environnement informatique familier du concepteur. Le langage de programmation Matlab possède les caractéristiques recherchées. De plus, il permet la programmation orientée objet, nécessaire à l'implémentation des méthodes numériques exploitées présentée en section 2.3.3. La première contribution de nos travaux consiste à considérer la résolution de différents problèmes de conception d'un mécanisme à partir de la donnée seule du modèle géométrique implicite. Dans les chapitres 2 à 5, nous supposons donc disposer du modèle géométrique implicite du mécanisme implémenté dans une fonction Matlab définissant par ailleurs les coordonnées géométriques et les paramètres géométriques. Cette tâche ne requiert pas de compétence spécifique pour le concepteur de mécanismes.

Premièrement, nous avons fait le choix de manipuler le modèle géométrique implicite qui prend en compte l'ensemble des coordonnées passives du mécanisme. L'analyse de singularités repose sur l'étude du modèle cinématique implicite que nous choisissons d'établir par différentiation du modèle géométrique implicite. Dans le cadre de la classification de toutes les configurations singulières existantes, il est nécessaire d'établir 4 jacobiennes. La dimension du système traité peut rendre sa différentiation et donc l'établissement des jacobiennes fastidieuse. Dans notre approche numérique, l'évaluation du rang des jacobiennes requiert une évaluation très précise des dérivées combinée à une localisation très précise des configurations singulières.

Deuxièmement, nous avons vu que l'approche d'analyse de sensibilité par linéarisation du modèle géométrique est une approche générique et efficace, mais qui échoue à proximité des configurations singulières d'un mécanisme. Or l'analyse du comportement du mécanisme à proximité des configurations singulières s'est avérée nécessaire dans plusieurs situations de conception [Bachta *et al.*, 2009 ; Rubbert *et al.*, 2014 ; Tannous *et al.*, 2014]. Nous proposons une approche d'analyse de sensibilité de haut-degré dont la dissémination est faible car elle nécessite l'établissement de coefficients de Taylor d'ordre élevé qui peut être fastidieux. Elle consiste en la différentiation de haut-degré du modèle géométrique. Le calcul de ces coefficients de Taylor est par ailleurs nécessaire lors de la résolution des systèmes étendus avec la méthode Diamant.

Troisièmement, la détermination des frontières de l'espace de travail et des configurations singulières requiert l'établissement de systèmes étendus impliquant l'évaluation de matrices

jacobiennes. L'évaluation des jacobiennes peut être fastidieux et l'assemblage de ces systèmes nécessite une certaine expertise.

La différentiation automatique est une méthode algorithmique permettant d'évaluer les dérivées exactes de fonctions implémentées sous forme de code informatique. Le choix de son exploitation permet d'envisager une automatisation de l'établissement et de l'évaluation des matrices jacobiennes, systèmes étendus, et polynômes de Taylor impliqués dans nos méthodes. Nous pouvons ainsi proposer des outils génériques, automatisés et précis, ne nécessitant que le modèle géométrique implicite.

La résolution des problèmes de conception nécessite la résolution de différents systèmes d'équations non-linéaires. Considérer ces systèmes comme un seul système générique permet d'envisager la résolution de différents problèmes au sein d'un outil unique. Cet outil requiert alors une méthode de résolution générique. Nous proposons d'utiliser la méthode de continuation de haut-degré Diamant pour constituer le coeur de résolution des outils numériques proposés. Elle possède l'avantage de ne pas nécessiter de formulation particulière des systèmes pour une résolution automatisée. Diamant est une méthode générique et efficace qui permet l'obtention automatisée de branches de solutions continues sous forme de polynômes de Taylor. Ses fondements sont introduits au chapitre suivant.

Chapitre 2

Méthodes et outils numériques

« En mathématiques, on ne comprend pas les choses, on s’y habitue », John von Neumann.

Sommaire

2.1	Résolution de problèmes non-linéaires	32
2.1.1	Diagramme de bifurcation	33
2.1.2	Continuation numérique	37
2.1.3	Modes de pilotage	39
2.1.4	Prédiction d’ordre 1	41
2.1.5	Correction d’ordre 1	42
2.1.6	Conclusion sur la méthode de continuation d’ordre 1	44
2.2	Continuation et séries de Taylor	44
2.2.1	Méthodes numériques et séries	44
2.2.2	Continuation de haut-degré	48
2.2.3	Méthode Diamant	49
2.3	Différentiation automatique de codes informatiques	51
2.3.1	Motivation et principe	52
2.3.2	Différentiation par transformation de source	53
2.3.3	Différentiation par surcharge d’opérateur	56
2.4	Diamanlab : une implémentation Matlab de Diamant	59
2.4.1	Logiciels de continuation de haut-degré	60
2.4.2	Structure de la bibliothèque	60
2.4.3	Continuation avec Diamanlab et interface graphique	64
2.5	Conclusion	66

La nécessité d'une approche de résolution numérique générique pour la conception de mécanismes a été identifiée dans le chapitre 1. Nous avons alors identifié le besoin d'une méthode efficace d'évaluation des caractéristiques essentielles d'un mécanisme. Nous souhaitons de plus que les outils numériques soient facilement exploitables par le concepteur. Ce chapitre présente les méthodes et des outils numériques exploités pour y parvenir.

En section 2.1, sont introduites les notions fondamentales de la continuation numérique : les diagrammes de bifurcations et l'analyse de stabilité des systèmes [Seydel, 2010]. La résolution de problèmes non-linéaires par une méthode de continuation numérique d'ordre 1 [Allgower et Georg, 1990 ; Doedel *et al.*, 2007] est alors présentée. Les trois éléments essentiels sont décrits : le mode de pilotage (Section 2.1.3), la méthode de prédiction (Section 2.1.4) et la méthode de correction (Section 2.1.5). Les inconvénients des méthodes d'ordre 1 sont finalement soulignés.

Nous introduisons en section 2.2 le principe de la méthode *Diamant* pour la résolution des systèmes d'équations non-linéaires [Charpentier et Potier-Ferry, 2008]. *Diamant* est l'acronyme de Différentiation Automatique de la Méthode Asymptotique Numérique Typée. Il s'agit d'une méthode de continuation de haut-degré automatisée au moyen de la différentiation automatique.

Nous proposons également d'exploiter la différentiation automatique [Griewank et Walther, 2008] pour une obtention automatisée des matrices jacobiniennes utilisées pour la détection et la classification des configurations singulières d'un mécanisme. Nous présentons en conséquence en section 2.3 le principe des méthodes de différentiation automatique et les outils utilisés pour la formulation et la résolution automatiques des problèmes traités.

Nous décrivons en section 2.4 l'implémentation en langage Matlab du logiciel de continuation de haut-degré *Diamanlab* [Charpentier *et al.*, 2013]. Il est le support pour les développements de nos outils de conception de mécanismes présentés aux chapitres 3, 4 et 5. L'intérêt des outils numériques pour les problématiques de conception de mécanismes est discuté en section 2.5 sur la base des éléments décrits précédemment.

2.1 Résolution de problèmes non-linéaires

Des couplages entre les méthodes de continuation d'ordre 1 et des problèmes en analyse de mécanismes et en robotique ont déjà été proposés, comme illustré dans le tableau 2.1. Les méthodes de continuation par homotopie ont été introduites pour la résolution des systèmes polynomiaux apparaissant dans l'étude du mouvement des mécanismes par Tsai et Morgan [1985], Wampler *et al.* [1990] et Sommese *et al.* [2004]. Tsai et Morgan [1985]

ont par exemple proposé le calcul du modèle géométrique inverse de mécanismes sériels d'architecture quelconque. Les branches de solutions rencontrées dans la résolution de ces problèmes ne présentent alors pas de points particuliers comme des points de bifurcation. Haug *et al.* [1996] a proposé l'utilisation de continuation d'ordre 1 pour la résolution de systèmes étendus représentant les frontières de l'espace de travail, sur laquelle nous reviendrons au chapitre 3.

Par ailleurs, les séries entières ont été exploitées conjointement à la continuation polynomiale [Morgan *et al.*, 1992]. Récemment, Bohigas *et al.* [2013] ont développé une méthode pour une planification de trajectoire évitant les singularités combinant homotopie et continuation en dimension élevée [Henderson, 2007]. En robotique, Du et Hwang [2007] et Nik *et al.* [2013] ont utilisé la méthode des séries de Taylor pour la résolution de problèmes de commande optimale.

Problème	Formulation générique	Quelques références
Géométrie	$\mathcal{R}^{\mathcal{C}}(\mathbf{q}) = \mathbf{0}$	Espace des configurations [Rubbert <i>et al.</i> , 2017]
Cinématique	$\mathcal{F}(\mathbf{q}, b) = \mathbf{0}$	Synthèse de trajectoire [Bohigas <i>et al.</i> , 2013]
Homotopie	$\mathcal{H}(\mathbf{p}, a) = a(\mathcal{F}(\mathbf{p})) + (1 - a)(\mathcal{G}(\mathbf{p})) = \mathbf{0}$	Synthèse de mécanismes [Roth et Freudenstein, 1963]. Résolution de problèmes cinématiques sous forme polynomiale $\mathcal{F}(\mathbf{p}) = \mathbf{0}$ [Tsai et Morgan, 1985 ; Sommese <i>et al.</i> , 2004].
Géométrie-Statique	$\mathcal{R}^{ETC}(\mathbf{q}, \mathbf{w}_e) = \mathbf{0}$	Espace de travail sous charge [Boehler <i>et al.</i> , 2015]

TABLEAU 2.1 – Un ensemble de problèmes non-linéaires résolus par continuation en robotique et leur formulation générique.

2.1.1 Diagramme de bifurcation

Les difficultés de résolution numérique des problèmes non-linéaires en mécanique sont illustrés sur le cas du système masse-ressort plan décrit en figure 2.1 [Cochelin *et al.*, 2007 - p.21].

Il s'agit d'une masse m suspendue à un ressort, soumise à l'action de la pesanteur et d'une force horizontale d'amplitude λ variable. On note z et x les coordonnées de la masse,

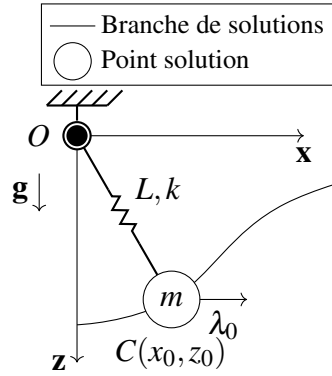


FIGURE 2.1 – Système masse-ressort plan soumis à l'action de la gravité et d'une force horizontale λ_0 . La branche de solutions décrit les positions (x_0, z_0) d'équilibre pour une force horizontale d'amplitude λ variable.

k la raideur du ressort et L_0 sa longueur à vide. La longueur courante s'exprime par :

$$L = \sqrt{z^2 + x^2}, \quad (2.1)$$

et l'énergie potentielle du système :

$$P = \frac{1}{2}k(L - L_0)^2 - mgz - \lambda x.$$

Les états admissibles pour ce système correspondent aux positions d'équilibre de la masse qui minimisent l'énergie potentielle du système par :

$$\begin{cases} \frac{\partial P}{\partial z} = k(L - L_0) \frac{\partial L}{\partial z} - mg = 0, \\ \frac{\partial P}{\partial x} = k(L - L_0) \frac{\partial L}{\partial x} - \lambda = 0, \end{cases} \quad (2.2)$$

qui peut s'écrire après introduction des coordonnées de la masse :

$$\mathcal{R}(x, z, \lambda) = \begin{bmatrix} k(\sqrt{z^2 + x^2} - L_0) \frac{z}{\sqrt{z^2 + x^2}} - mg \\ k(\sqrt{z^2 + x^2} - L_0) \frac{x}{\sqrt{z^2 + x^2}} - \lambda \end{bmatrix} = \mathbf{0}. \quad (2.3)$$

Ce système, de forme générique $\mathcal{R}(u, \lambda) = \mathbf{0}$ avec $u = (x, z)$, est composé d'équations non-linéaires appelées équations résiduelles. On retrouve la forme des problèmes introduits en chapitre 1, par exemple le modèle géométrique implicite (1.4). En mécanique, on s'intéresse à l'ensemble des états u admissibles pour ce système, *i.e.* satisfaisant (2.3) pour λ variable. Les

solutions (u, λ) forment une courbe continue, appelée *branche de solutions*, dans l'espace \mathbb{R}^3 . La projection selon (x, z) de la branche de solutions du système (2.3) est représentée par la courbe superposée au schéma de la figure 2.1.

Définition 8 : Un diagramme décrivant l'évolution d'une composante de u en fonction de λ tel que (u, λ) soit solution de $\mathcal{R}(u, \lambda) = \mathbf{0}$, est appelé *diagramme de bifurcation* [Seydel, 2010 - Section 2.2].

L'étude des changements d'état d'un système constitue un problème central en mécanique. Ces changements se manifestent le long d'une branche de solutions par l'émergence de points remarquables correspondant à un changement du nombre de solutions (u, λ) lorsque λ varie. Comme l'illustre le diagramme de bifurcations fictif présenté en figure 2.2, des branches de solutions peuvent émerger, se terminer, ou s'intersecter. Le nombre de solutions pour une valeur de λ donnée est reporté dans le tableau 2.2.

Définition 9 : Un *point de bifurcation* (u^{bif}, λ^{bif}) est un point solution du système d'équations $\mathcal{R}(u, \lambda) = \mathbf{0}$ en lequel le nombre de solutions change [Seydel, 2010 - Section 2.2]. Ce point (u^{bif}, λ^{bif}) peut être détecté par l'étude de dérivées du système. Par exemple un *point de bifurcation simple* (u^{bif}, λ^{bif}) remplit les conditions suivantes [Seydel, 2010 - Section 2.5] :

1. $\mathcal{R}(u^{bif}, \lambda^{bif}) = \mathbf{0}$;
2. La matrice $\mathcal{R}_{(u, \lambda)}(u^{bif}, \lambda^{bif})$ perd 1 rang.

Un point de bifurcation simple se manifeste sur un diagramme de bifurcation par une intersection de 2 branches de solutions avec deux tangentes distinctes, comme montré en figure 2.2. Notons que d'autres types de points de bifurcation peuvent apparaître dans le cadre de la résolution d'équations différentielles ordinaires [Seydel, 2010 - Section 2.6].

Définition 10 : Un *point de retournement* (u^{ret}, λ^{ret}) remplit les conditions suivantes [Seydel, 2010 - Section 2.5] :

1. $\mathcal{R}(u^{ret}, \lambda^{ret}) = \mathbf{0}$;
2. La matrice $\mathcal{R}_u(u^{ret}, \lambda^{ret})$ perd 1 rang ;
3. La matrice $\mathcal{R}_{(u, \lambda)}(u^{ret}, \lambda^{ret})$ est de rang plein ;
4. Il existe une paramétrisation $u(a), \lambda(a)$, avec $u(a^{ret}) = u^{ret}, \lambda(a^{ret}) = \lambda^{ret}$
et $\frac{d^2 \lambda(a^{ret})}{da^2} \neq 0$.

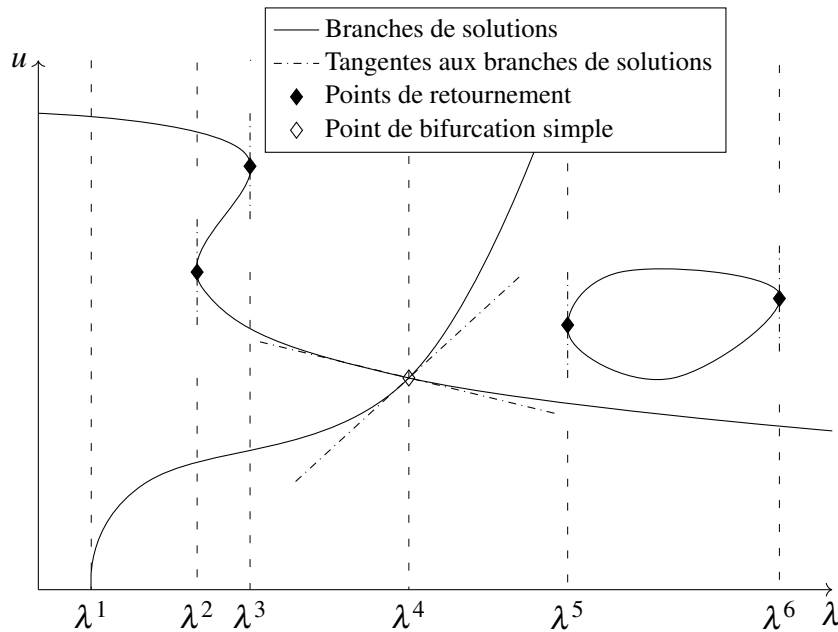


FIGURE 2.2 – Un diagramme de bifurcation fictif.

Valeurs de λ	Nombre de solutions $u(\lambda)$
$\lambda < \lambda^1$	1
$\lambda^1 \leq \lambda < \lambda^2$	2
λ^2	3
$\lambda^2 < \lambda < \lambda^3$	4
λ^3	3
$\lambda^3 < \lambda < \lambda^4$	2
λ^4	1
etc...	

TABLEAU 2.2 – Multiplicité des solutions sur le diagramme de bifurcations de la figure 2.2.

Les 3 premières conditions garantissent que la tangente à la branche de solutions au point (u^{ret}, λ^{ret}) est perpendiculaire à l'axe λ , comme montré en figure 2.2. La quatrième condition implique le point (u^{ret}, λ^{ret}) est à un extrema de λ et non un point d'inflexion.

Les situations en figure 2.2 illustrent les méthodes nécessaires à la détermination de l'ensemble de solutions du système $\mathcal{R}(u, \lambda) = \mathbf{0}$:

- une *méthode de continuation* pour tracer une branche de solutions à partir d'un point initial ;
- une méthode pour la *détection de points de bifurcation* ;

— une méthode pour permettre le *changement de branches*.

2.1.2 Continuation numérique

Soit le problème résiduel non-linéaire paramétrique générique :

$$\mathcal{R}(u, \lambda) = \mathbf{0}, \quad (2.4)$$

où \mathcal{R} est une fonction vectorielle à valeurs réelles dépendant du vecteur d'état $u \in \mathbb{R}^n$ et du paramètre $\lambda \in \mathbb{R}$:

$$\begin{aligned} \mathcal{R} : \mathbb{R}^{n_{inc}} &\longrightarrow \mathbb{R}^{n_{eq}} \\ (u, \lambda) &\longmapsto \mathcal{R}(u, \lambda) \end{aligned}, \quad (2.5)$$

où $n_{inc} = n + 1$ est le nombre d'inconnues du système (2.4) et $n_{eq} = n$ est le nombre de ses équations. L'ensemble des solutions du système sous-déterminé (2.4) est de dimension 1 et est constitué de *branches de solutions*.

Les méthodes de continuation d'ordre 1, dites de prédiction-correction, reposent sur un schéma itératif permettant d'approcher une branche de solutions par une succession de points, comme représenté en figure 2.3. La fonction \mathcal{R} est supposée différentiable par rapport à (u, λ) , on peut donc écrire son approximation linéaire et déterminer une tangente.

Le principe des *méthodes de continuation* est de déterminer une approximation numérique des branches de solutions, de proche en proche, à partir d'un point initial (u^0, λ^0) solution du système (2.4). Les différentes méthodes de continuation se distinguent par les choix [Cochelin *et al.* 2007 - Section 1.1 ; Seydel 2010 - Section 4] :

- du mode de pilotage qui permet de clore le système (2.4) sous-déterminé [Seydel, 2010 - Section 4.5] :
 - le pilotage par un paramètre naturel, présenté en section 2.1.3.2, utilise l'inconnue λ comme un paramètre variant de manière monotone : le nombre d'inconnues devient $n_{inc} = n$,
 - le pilotage par la pseudo-longueur d'arc, présenté en section 2.1.3.3, consiste à ajouter une équation et un paramètre : le nombre d'équations devient $n_{eq} = n + 1$,
- de la méthode de prédiction et notamment de l'ordre de différentiation utilisé [Seydel, 2010 - Section 4.4] :
 - la linéarisation de la fonction \mathcal{R} résulte en la méthode de prédiction d'ordre 1 présentée en section 2.1.4 ;
 - une différentiation de haut-degré de la fonction \mathcal{R} est exploitée dans les *méthodes de continuation de haut-degré* présentées en section 2.2 [Cochelin *et al.*, 2007 - Section 1.2.1] ;

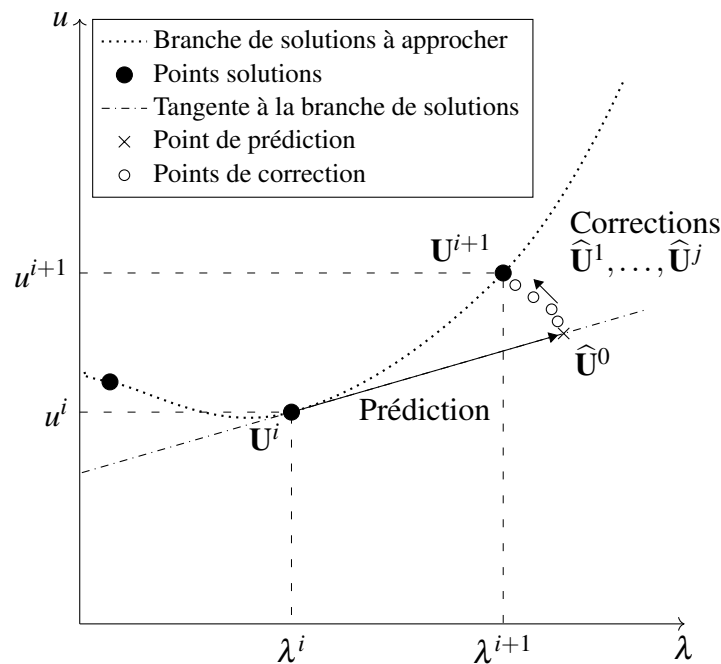


FIGURE 2.3 – Principe de continuation par prédiction-corrrection.

- de la méthode de correction exploitée pour améliorer l'approximation si nécessaire [Seydel, 2010 - Section 4.6]. L'étape de correction est généralement réalisée par l'algorithme de Newton-Raphson présenté en section 2.1.5.
- du contrôle de la longueur du pas de prédiction, décrit en section 2.1.6 [Seydel, 2010 - Section 4.7].

La différence principale entre les méthodes de continuation porte sur la méthode de prédiction de la solution utilisée. Les logiciels de continuation d'ordre 1 les plus connus sont AUTO [Doedel *et al.*, 2007], MATCONT [Dhooge *et al.*, 2004] et HOMPAC [Watson *et al.*, 1990]. Il est important de comprendre le principe des méthodes de continuation d'ordre 1 pour comprendre les avantages de la méthode de continuation de haut-degré Diamant.

Le processus de correction converge vers un point solution de la branche courante à condition que :

- la longueur du pas de prédiction soit suffisamment petite [Cochelin *et al.*, 2007 - Section 1.1.2].,
- les jacobiniennes pour la prédiction et les corrections soient évaluées hors du voisinage d'un point de bifurcation.

La définition préalable d'un mode de pilotage est donc primordiale pour l'application d'un schéma de prédiction-corrrection.

2.1.3 Modes de pilotage

2.1.3.1 Principe

Supposons connu un point (u^0, λ^0) solution du système d'équations (2.4) et appartenant à une branche de solutions. De manière générale, une courbe peut être paramétrée par un paramètre de pilotage noté a ayant une signification géométrique. Formellement, une solution du système (2.4) dépend alors de a :

$$\begin{cases} u = u(a) \\ \lambda = \lambda(a). \end{cases} \quad (2.6)$$

Pour une valeur donnée de a , le système (2.4) possède n équations en $n + 1$ inconnues : il est sous-déterminé. Le système est clos par l'ajout de l'équation de chemin $p(u, \lambda, a) = 0$ [Seydel, 2010 - Section 4.5] :

$$\begin{bmatrix} \mathcal{R}(u, \lambda) \\ p(u, \lambda, a) \end{bmatrix} = \mathbf{0}, \quad (2.7)$$

qui possède $n + 1$ équations en $n + 1$ inconnues et peut être résolu.

2.1.3.2 Paramètre naturel

Pour la continuation naturelle (ou paramétrique) [Allgower et Georg 1990 ; Doedel *et al.* 2007 ; Seydel 2010], on considère le vecteur d'état u comme fonction du paramètre λ , alors désigné comme *paramètre naturel*. Par exemple, pour le pendule de la figure 2.1, le choix de $a = \lambda$ est « naturel ». L'équation de chemin s'écrit alors : $p(u, \lambda) = \lambda - \lambda^0$. Comme montré en figure 2.4, la résolution par continuation du système (2.4) consiste alors à approcher la fonction $u(\lambda)$ par une succession de points $u^0, u^1, \dots, u^i, \dots$, en faisant varier λ d'un pas $\delta\lambda$ donné. Supposons connus un point solution initial (u^i, λ^i) et la tangente $u_\lambda(\lambda^i) = \frac{du}{d\lambda}(\lambda^i)$ à la branche de solutions en ce point. Nous voulons calculer le point \hat{u}^0 en $\hat{\lambda}^0 = \lambda^i + \delta\lambda$. L'étape de prédiction consiste alors à calculer le point de prédiction :

$$\hat{u}^0 = u^i + \delta\lambda u_\lambda(\lambda^i). \quad (2.8)$$

Celle-ci est complétée au besoin d'étapes de correction constituant ainsi un pas de continuation naturelle. Ces deux méthodes sont détaillées en sections 2.1.4 et 2.1.5.

Comme le montre la figure 2.4, la prédiction est impossible lorsqu'on atteint le point de retournement (u^{ret}, λ^{ret}) car la valeur de la dérivée $u_\lambda(\lambda^{ret})$ est infinie. C'est un inconvénient majeur du pilotage par un paramètre naturel. Dans le cadre de l'étude de mécanismes,

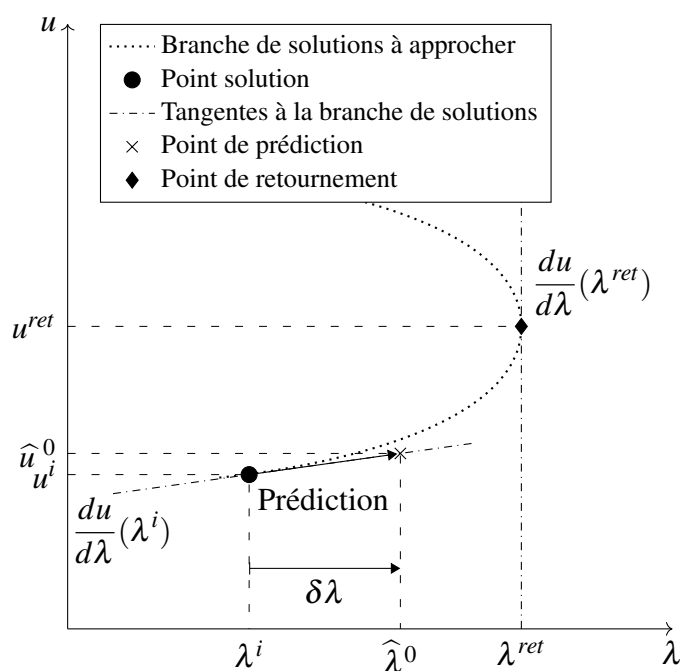


FIGURE 2.4 – Continuation naturelle et point de retournement par rapport au paramètre λ .

on remarque qu'aucun choix « naturel » de paramètre de pilotage ne s'impose parmi les coordonnées du mécanisme. Il est aussi nécessaire de choisir une méthode robuste aux points de retournement, voir Seydel [2010 - Section 4.5].

2.1.3.3 Pseudo-longueur d'arc

Keller [1987] propose une méthode de continuation pilotée par un paramètre mathématique a représentant l'abscisse curviligne le long de la branche de solutions. Les inconnues $\mathbf{U} = (u, \lambda)$ du système sont considérées comme une fonction de la variable réelle a appelée *paramètre de chemin* et satisfaisant l'équation de pseudo-longueur d'arc :

$$a = \left\langle \mathbf{U}(a) - \mathbf{U}(0), \frac{d\mathbf{U}}{da}(\mathbf{U}(0)) \right\rangle, \quad (2.9)$$

où $\langle \cdot, \cdot \rangle$ est un produit scalaire. Le paramètre a est une approximation tangente de la longueur de l'arc reliant les points $\mathbf{U}(a)$ et $\mathbf{U}(0)$, voir figure 2.5. L'ajout de cette équation permet de

clure le système (2.4). Celui-ci est reformulé comme suit :

$$\begin{cases} \mathcal{R}(\mathbf{U}(a)) = \mathbf{0}, \\ a = \left\langle \mathbf{U}(a) - \mathbf{U}(0), \frac{d\mathbf{U}}{da}(0) \right\rangle. \end{cases} \quad (2.10)$$

Le pilotage par un paramètre de chemin permet la prédiction en tout point de retournement et est indépendant des composantes de \mathbf{U} .

2.1.4 Prédiction d'ordre 1

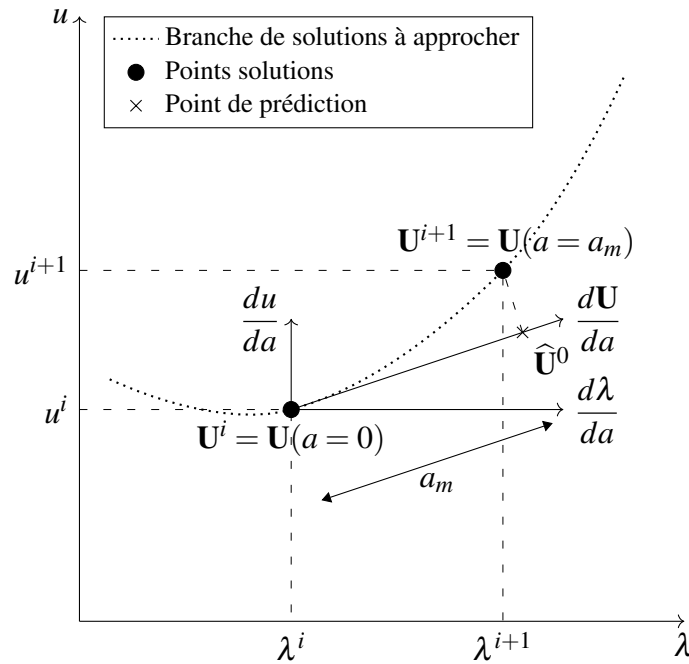


FIGURE 2.5 – Interprétation géométrique de la continuation avec prédiction d'ordre 1 et pilotage par le paramètre de pseudo-longueur d'arc.

La première étape du processus de continuation, dite de prédiction, consiste à perturber la solution dans une direction définie par la tangente à la branche de solutions avec un pas de prédiction à déterminer. La précision du point de prédiction se dégrade si la non-linéarité est importante et la longueur du pas de perturbation augmente. On appliquera des étapes de corrections si la norme du résidu $\mathcal{R}(\hat{\mathbf{U}}^0)$ au point prédit est jugée trop grande (section 2.1.5).

La résolution par continuation selon la pseudo-longueur d'arc du système (2.10) consiste à approcher la fonction $\mathbf{U}(a)$ par une succession de points $\mathbf{U}^0, \mathbf{U}^1, \dots, \mathbf{U}^i, \dots$, en faisant varier a d'un pas a_m donné. Arbitrairement, on peut fixer $a = 0$ pour chaque nouveau point.

Supposons connus un point solution initial $\mathbf{U}^i = \mathbf{U}(0)$ et la tangente $\frac{d\mathbf{U}}{da}(0)$ en ce point indiquant la direction de prédiction. Nous voulons calculer la solution $\mathbf{U}^{i+1} = \mathbf{U}(a_m)$ en a_m , comme représenté en figure 2.5. Pour cela, une approximation $\widehat{\mathbf{U}}^0$ du point $\mathbf{U}(a_m)$ est calculée telle que $\widehat{\mathbf{U}}^0 = \mathbf{U}(0) + \delta\mathbf{U}$ où $\delta\mathbf{U}$ est l'incrément de prédiction pour le pas a_m . L'étape de prédiction consiste à résoudre le système :

$$\begin{cases} \mathcal{R}(\widehat{\mathbf{U}}^0) = \mathbf{0}, \\ a_m = \left\langle \widehat{\mathbf{U}}^0 - \mathbf{U}(0), \frac{d\mathbf{U}}{da}(0) \right\rangle. \end{cases} \quad (2.11)$$

Comme \mathcal{R} est supposée différentiable, elle est linéarisée au point $\mathbf{U}(0)$ et évaluée au point $\widehat{\mathbf{U}}^0$:

$$\mathcal{R}(\mathbf{U}(0) + \delta\mathbf{U}) \approx \mathcal{R}(\mathbf{U}(0)) + \frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(0))\delta\mathbf{U}, \quad (2.12)$$

où $\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(0))$ est la matrice jacobienne de \mathcal{R} en $\mathbf{U}(0)$. Comme $\mathcal{R}(\mathbf{U}(0)) \approx \mathbf{0}$, l'incrément de prédiction $\delta\mathbf{U}$ est solution du problème linéaire :

$$\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(0))\delta\mathbf{U} = \mathbf{0}. \quad (2.13)$$

Le système (2.13) étant sous-déterminé, on lui ajoute l'équation de pseudo-longueur d'arc linéarisée. L'étape de prédiction consiste à déterminer l'inconnue $\delta\mathbf{U}$ solution de :

$$\begin{cases} \frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(0))\delta\mathbf{U} = \mathbf{0}, \\ \left\langle \delta\mathbf{U}, \frac{d\mathbf{U}}{da}(0) \right\rangle - a_m = 0. \end{cases} \quad (2.14)$$

Cette solution $\widehat{\mathbf{U}}^0$ généralement peu précise est corrigée comme suit.

2.1.5 Correction d'ordre 1

Étant donné un point prédit $\widehat{\mathbf{U}}^0$, nous cherchons une approximation \mathbf{U}^{i+1} appartenant à la branche de solutions. Pour cela nous cherchons itérativement une suite de points $\widehat{\mathbf{U}}^1, \dots, \widehat{\mathbf{U}}^j, \dots$ qui converge vers \mathbf{U}^{i+1} , voir figure 2.3.

Le processus présenté ici correspond à l'algorithme de Newton-Raphson. Il est implémenté dans l'outil de continuation exploité dans cette thèse, voir section 2.4. Comme \mathcal{R} est supposée différentiable, elle est linéarisée au point $\hat{\mathbf{U}}^j$ et évaluée au point $\hat{\mathbf{U}}^{j+1}$:

$$\mathcal{R}(\hat{\mathbf{U}}^{j+1}) \approx \mathcal{R}(\hat{\mathbf{U}}^j) + \left(\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\hat{\mathbf{U}}^j) \right) (\hat{\mathbf{U}}^{j+1} - \hat{\mathbf{U}}^j), \quad (2.15)$$

où $\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\hat{\mathbf{U}}^j)$ est la matrice jacobienne de \mathcal{R} en $\hat{\mathbf{U}}^j$.

Pour améliorer l'approximation de la solution, il faut parvenir à

$$\left\| \mathcal{R}(\hat{\mathbf{U}}^{j+1}) \right\| < \varepsilon_{nr}, \quad (2.16)$$

où ε_{nr} est la tolérance de précision de l'approximation. Le système linéaire en l'incrément de correction $\delta \hat{\mathbf{U}}^j = (\hat{\mathbf{U}}^{j+1} - \hat{\mathbf{U}}^j)$

$$\left(\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\hat{\mathbf{U}}^j) \right) \delta \hat{\mathbf{U}}^j = -\mathcal{R}(\hat{\mathbf{U}}^j) \quad (2.17)$$

doit être résolu. Une condition supplémentaire est nécessaire pour clore le système (2.17). Dans les méthodes de continuation selon la pseudo-longueur d'arc on linéarise l'équation de pseudo-longueur d'arc (2.9) pour écrire :

$$\begin{cases} \left(\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\hat{\mathbf{U}}^j) \right) \delta \hat{\mathbf{U}}^j = -\mathcal{R}(\hat{\mathbf{U}}^j) \\ \delta \hat{\mathbf{U}}^j \cdot \delta \mathbf{U} = \mathbf{0} \end{cases}, \quad (2.18)$$

qui exprime que les incréments de correction doivent être orthogonaux à l'incrément de prédiction. Le calcul de $\delta \hat{\mathbf{U}}^j$ par la résolution du système (2.18) permet de déduire $\hat{\mathbf{U}}^{j+1} = \hat{\mathbf{U}}^j + \delta \hat{\mathbf{U}}^j$.

On obtient de manière pratique une suite de points qui converge vers \mathbf{U}^{i+1} en une dizaine d'itérations à condition que :

- le point prédit $\hat{\mathbf{U}}^0$ soit suffisamment proche de la branche de solutions,
- la jacobienne $\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}^{i+1})$ évaluée au point solution ne soit pas singulière.

En général, on définit un nombre maximal de répétitions N_{nr} du processus de correction.

2.1.6 Conclusion sur la méthode de continuation d'ordre 1

Avec la méthode de continuation pilotée par un paramètre naturel, le nouveau point solution s'écrit :

$$\mathbf{U}^{i+1} = \mathbf{U}^i + \delta\mathbf{U} + \delta\hat{\mathbf{U}}^1 + \dots + \delta\hat{\mathbf{U}}^j + \dots \quad (2.19)$$

où l'incrément de prédiction $\delta\mathbf{U}$ est solution du système (2.14) et les incréments de correction $\delta\hat{\mathbf{U}}^j$ sont solutions du système (2.18). La résolution peut s'avérer coûteuse en temps de calcul si les incréments $\delta\mathbf{U}$ sont petits.

Les 3 difficultés rencontrées en analyse des mécanismes sont :

1. le nombre élevé - voire infini en présence de fonctions trigonométriques - de branches de solutions, le nombre élevé de points de bifurcations,
2. l'approximation de la branche sous forme discrète,
3. le choix du pas a_m qui est délicat car il résulte d'un compromis d'efficacité entre l'algorithme de Newton-Raphson et les étapes de prédiction.

Les méthodes de continuation de haut-degré implémentent une méthode de prédiction de haut-degré qui permet l'obtention de solutions continues par morceau.

2.2 Continuation et séries de Taylor

Les méthodes présentées dans cette section ont été développées pour une résolution fiable et efficace des problèmes non-linéaires suffisamment différentiables. Leur principe est de représenter l'inconnue par une série de Taylor.

2.2.1 Méthodes numériques et séries

Le principe des méthodes numériques reposant sur les séries de Taylor est d'exploiter les dérivées d'ordre supérieur pour une meilleure prise-en-compte des non-linéarités. Elles permettent la résolution de nombreux problèmes non-linéaires.

Un tableau issu de [Charpentier, 2012] présentant un ensemble de problèmes non-linéaires résolus au moyen de méthodes basées sur la méthode asymptotique numérique (présentée en section 2.2.2) est reproduit en tableau 2.3 en ajoutant le problème d'analyse des mécanismes récemment traité par Rubbert *et al.* [2017].

Fonction analytique et développement en séries de Taylor Une *fonction analytique* est une fonction indéfiniment dérivable qui peut être approchée par un développement en série

Type d'équations	Problème	Formulation générique	Quelques références
Algébrique	Problèmes aux limites	$\mathcal{R}(u, \lambda) = 0$	Bratu [Cochelin <i>et al.</i> , 2007 ; Charpentier et Potier-Ferry, 2008] [◊] Lois de comportement non-linéaire [Koutsawa <i>et al.</i> , 2008] [◊] Vibration forcée non-linéaire [Azrar <i>et al.</i> , 1993] Contact [Boudaoud <i>et al.</i> , 2009] [◊]
	Géométrique	$\mathcal{R}^{\mathcal{C}}(\mathbf{q}) = 0$	Espace des configurations [Rubbert <i>et al.</i> , 2017] [◊]
	Valeurs propres	$\mathcal{P}(\lambda)u = 0$ $\mathcal{Q}(u, \lambda)u = 0$	Problème propre non-linéaire [Guillaume, 1999] Vibrations libres non-linéaires [Daya et Potier-Ferry, 2001] [◊]
Temporelle	EDO/EDA	$\mathcal{R}(u, \dot{u}, \lambda) = 0$	Pendule [Hoefkens <i>et al.</i> , 2002] Plasticité [Cochelin <i>et al.</i> , 2007] Frottement [Béchet <i>et al.</i> , 2010] [◊]
	EDP	$\mathcal{R}(u, \dot{u}) = 0$	Problèmes d'EDP stationnaires [Jamal <i>et al.</i> , 2002]

TABLEAU 2.3 – Un ensemble de problèmes non-linéaires résolus au moyen de méthodes basées sur la méthode asymptotique numérique (Adapté de [Charpentier, 2012] et augmenté de [Rubbert *et al.*, 2017]) et les applications traitées avec la méthode Diamant ([◊]). Les formulations génériques représentent les types de fonctions et de variables traitées dans les différents domaines.

de Taylor. Soit \mathbf{U} une fonction analytique d'une variable réelle a définie au voisinage de 0. La série de Taylor de \mathbf{U} en 0 s'écrit :

$$\mathbf{U}(a) = \mathbf{U}(0) + \frac{a}{1!} \frac{d\mathbf{U}}{da}(0) + \frac{a^2}{2!} \frac{d^2\mathbf{U}}{da^2}(0) + \frac{a^3}{3!} \frac{d^3\mathbf{U}}{da^3}(0) + \dots, \quad (2.20)$$

où $\mathbf{U}^{(k)}(0) = \frac{d^k\mathbf{U}}{da^k}(0)$ est la dérivée k -ième de \mathbf{U} au point 0. La représentation d'une fonction analytique par une série n'est valable que dans le rayon de convergence de la série.

Un *polynôme de Taylor* est une série de Taylor dont la somme est tronquée à un ordre donné K :

$$\mathbf{U}(a) \approx \sum_{k=0}^K a^k \frac{1}{k!} \mathbf{U}^{(k)}(0), \forall a \in \mathcal{V}(0), \quad (2.21)$$

où $k!$ est la factorielle de k et $\mathcal{V}(0)$ est un voisinage de 0. On parlera dans la suite d'approximation d'une fonction par un polynôme de Taylor d'ordre K .

Coefficients de Taylor Pour des raisons de stabilité des algorithmes, on ne calcule pas des dérivées de haut-degré, mais des dérivées pondérées appelées *coefficients de Taylor d'ordre k* et notés $\mathbf{U}_k = \frac{1}{k!} \mathbf{U}^{(k)}(0), k \in \mathbb{N}$.

Soit $\mathcal{R}(\mathbf{U})$ une fonction analytique qui dépend de \mathbf{U} qui est elle-même une fonction analytique de a . La fonction composée $(\mathcal{R} \circ \mathbf{U})(a) = \mathcal{R}(\mathbf{U}(a))$ est également analytique et peut être approchée au voisinage de $\mathbf{U}(0)$ par son polynôme de Taylor :

$$\mathcal{R}(\mathbf{U}(a)) \approx \mathcal{R}(\mathbf{U}(0)) + \sum_{k=1}^K a^k \frac{1}{k!} \frac{d^k(\mathcal{R} \circ \mathbf{U})}{da^k}(0), \forall a \in [0, a_m], \quad (2.22)$$

où $\mathcal{R}(\mathbf{U}(0)) = \mathbf{0}$, et $[0, a_m]$ est le domaine de validité de l'approximation.

Le principe de résolution du système générique (2.4) est d'approcher sa solution \mathbf{U} à l'aide de polynômes de Taylor (2.21). Le système étant supposé analytique, trouver

$$\mathbf{U}(a) = \sum_{k=0}^K a^k \mathbf{U}_k \quad (2.23)$$

revient à chercher les \mathbf{U}_k tels que

$$(\mathcal{R} \circ \mathbf{U})(a) = \sum_{k=0}^K a^k \frac{1}{k!} \frac{d^k(\mathcal{R} \circ \mathbf{U})}{da^k}(0) = \mathbf{0}, \quad (2.24)$$

i.e.

$$\frac{d^k(\mathcal{R} \circ \mathbf{U})}{da^k}(0) = \mathbf{0}, \forall k \leq K \quad (2.25)$$

La résolution de système reformulé suppose la différentiation de haut-degré de la fonction composée $(\mathcal{R} \circ \mathbf{U})(a)$.

Différentiation de haut-degré d'une fonction composée Pour illustrer nos propos, nous utilisons quatre fois la règle de dérivation d'une fonction composée $(\mathcal{R} \circ \mathbf{U})(a)$. On obtient :

$$\begin{aligned}
(\mathcal{R} \circ \mathbf{U})(a) &= \mathcal{R}(\mathbf{U}(a)), \\
\mathcal{R}^{(1)}(a) &= \left(\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(a)) \right) \cdot \left(\mathbf{U}^{(1)}(a) \right), \\
\mathcal{R}^{(2)}(a) &= \left(\frac{\partial^2 \mathcal{R}}{\partial \mathbf{U}^2}(\mathbf{U}(a)) \right) \cdot \left(\mathbf{U}^{(1)}(a) \right)^2 + \left(\frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(a)) \right) \cdot \left(\mathbf{U}^{(2)}(a) \right), \\
\mathcal{R}^{(3)} &= \frac{\partial^3 \mathcal{R}}{\partial \mathbf{U}^3} \cdot \left(\mathbf{U}^{(1)} \right)^3 + 3 \frac{\partial^2 \mathcal{R}}{\partial \mathbf{U}^2} \cdot \mathbf{U}^{(2)} \cdot \mathbf{U}^{(1)} + \frac{\partial \mathcal{R}}{\partial \mathbf{U}} \cdot \mathbf{U}^{(3)}, \\
\mathcal{R}^{(4)} &= \frac{\partial^4 \mathcal{R}}{\partial \mathbf{U}^4} \cdot \left(\mathbf{U}^{(1)} \right)^4 + 6 \frac{\partial^3 \mathcal{R}}{\partial \mathbf{U}^3} \cdot \mathbf{U}^{(2)} \cdot \left(\mathbf{U}^{(1)} \right)^2 + 4 \frac{\partial^2 \mathcal{R}}{\partial \mathbf{U}^2} \cdot \mathbf{U}^{(3)} \cdot \mathbf{U}^{(1)} \\
&\quad + 3 \frac{\partial^2 \mathcal{R}}{\partial \mathbf{U}^2} \cdot \left(\mathbf{U}^{(2)} \right)^2 + \frac{\partial \mathcal{R}}{\partial \mathbf{U}} \cdot \mathbf{U}^{(4)},
\end{aligned} \tag{2.26}$$

où $\frac{\partial \mathcal{R}}{\partial \mathbf{U}}$ est la matrice jacobienne de \mathcal{R} différentié par rapport à \mathbf{U} , $\mathcal{R}^{(k)}(a) = \frac{d^k (\mathcal{R} \circ \mathbf{U})}{da^k}(a)$ et $\mathbf{U}^{(k)}(a) = \frac{d^k \mathbf{U}}{da^k}(a)$ sont respectivement les dérivées d'ordre k de $(\mathcal{R} \circ \mathbf{U})$ et \mathbf{U} par rapport à a . La dépendance de \mathcal{R} en \mathbf{U} et de \mathbf{U} en a est omise à partir de l'ordre 3 pour plus de clarté. A chaque ordre k , seuls les termes de plus haut-degré $\frac{\partial^k \mathcal{R}}{\partial \mathbf{U}^k}$ et $\mathbf{U}^{(k)}$ sont à calculer, les autres termes dépendant des dérivées calculées aux ordres $1, \dots, (k-1)$. On remarque que la matrice jacobienne $\frac{\partial \mathcal{R}}{\partial \mathbf{U}}$ apparaît à chaque ordre. Deux conclusions importantes sont déduites de ces observations :

- le calcul des dérivées de haut-degré de $\mathcal{R} \circ \mathbf{U}$ par récurrence peut être envisagé. Ainsi, la règle de dérivation des fonctions composées a été généralisée aux dérivées d'ordre supérieur par Arbogast en 1800, puis reformulée de manière plus moderne par Faà di Bruno en 1855 [Charpentier *et al.*, 2016]. ;
- à l'ordre k , la dérivée peut être décomposée en un terme linéaire $\frac{\partial \mathcal{R}}{\partial \mathbf{U}} \cdot \mathbf{U}^{(k)}$ et un terme non-linéaire noté $\{\mathcal{R}_k^{nl}\}$ rassemblant les termes de haut-degré restants, c'est-à-dire $\frac{\partial^2 \mathcal{R}}{\partial \mathbf{U}^2}, \dots, \frac{\partial^k \mathcal{R}}{\partial \mathbf{U}^k}$, et $\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(k-1)}$.

Plutôt que les dérivées $\mathcal{R}^{(k)}$, on calcule donc les coefficients de Taylor $\mathcal{R}_k = \frac{1}{k!} \mathcal{R}^{(k)}(0)$. Dans la méthode Diamant présentée en section 2.2.3, les termes non-linéaires de haut-degré présents dans l'équation (2.26) et notés $\{\mathcal{R}_k^{nl}\}$ ci-dessus sont calculés comme des coefficients

de Taylor $\{\mathcal{R}_k|_{\mathbf{U}_k=\mathbf{0}}\}$ évalués avec $\mathbf{U}_k = \mathbf{0}$ [Bilasse *et al.*, 2009]. Ce choix particulier annule la contribution du terme linéaire tangent $\frac{\partial \mathcal{R}}{\partial \mathbf{U}} \cdot \mathbf{U}^{(k)}$.

Le coefficient de Taylor \mathcal{R}_k peut ainsi être décomposé en deux termes et calculé efficacement [Bilasse *et al.*, 2009] :

$$\mathcal{R}_k = \{\mathcal{R}_1|_{\mathbf{U}_1=Id}\} \mathbf{U}_k + \{\mathcal{R}_k|_{\mathbf{U}_k=\mathbf{0}}\}, \quad (2.27)$$

où $\{\mathcal{R}_1|_{\mathbf{U}_1=Id}\} = \frac{\partial \mathcal{R}}{\partial \mathbf{U}}(\mathbf{U}(0))$ est la matrice jacobienne de \mathcal{R} par rapport à \mathbf{U} . Cette décomposition ainsi que le calcul des coefficients de Taylor par récurrence constituent deux fondements de la méthode de continuation de haut-degré Diamant.

2.2.2 Continuation de haut-degré

Les systèmes d'équations non-linéaires paramétriques peuvent être résolus avec les méthodes de continuation de haut-degré. Nous proposons leur application dans le domaine de la conception de mécanismes. Le principe de résolution des méthodes de continuation de haut-degré est d'approcher $\mathbf{U}(a)$ par une succession de morceaux de branches continus $\mathbf{U}^1(a), \dots, \mathbf{U}^i(a), \mathbf{U}^{i+1}(a), \dots$, voir figure 2.6. On suppose que les fonctions étudiées sont suffisamment dérivables pour pouvoir être approchées par un polynôme de Taylor d'ordre K représentant un morceau de branche de solutions. Les méthodes de continuation de haut-degré considèrent l'inconnue \mathbf{U} comme une fonction d'une variable réelle et implémentent généralement le pilotage par un paramètre de chemin satisfaisant l'équation de pseudo-longueur d'arc (2.9).

Thompson et Walker [1968] ont proposé l'association de la technique de prédiction de haut-degré et de la méthode des éléments finis et l'ont exploité dans le cadre de la mécanique des structures pour l'analyse de stabilité. Ce principe a été repris et développé dans la méthode asymptotique numérique [Cochelin *et al.*, 1994]. L'idée est de considérer une approximation par un polynôme de Taylor d'ordre K au lieu de l'approximation linéaire considérée pour une prédiction d'ordre 1 (2.12) afin de réaliser une étape de prédiction qui tende vers la solution exacte lorsque K est grand.

La résolution du système (2.4) a alors été partiellement automatisée pour des équations quadratiques en u [Cochelin *et al.*, 2007 - Section 2.4] :

$$\mathbf{L}(u) + \mathbf{Q}(u, u) - \lambda \mathbf{F} = \mathbf{0}, \quad (2.28)$$

où u est le vecteur inconnu, \mathbf{L} est un opérateur linéaire, \mathbf{Q} est un opérateur quadratique, \mathbf{F} est un vecteur donné et λ est un paramètre de contrôle. La formulation quadratique des problèmes requiert une certaine expertise ce qui peut expliquer que la méthode asymptotique numérique

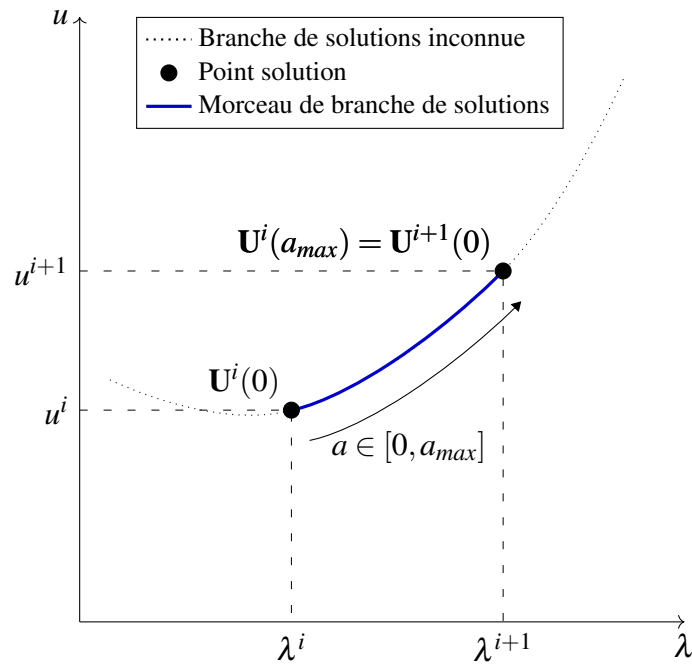


FIGURE 2.6 – Principe de la continuation de haut-degré : on approche la branche de solution $\mathbf{U}(a)$ en $\mathbf{U}^i(0)$ par un morceau de branche reliant de manière continue les points solution $\mathbf{U}^i(0)$ et $\mathbf{U}^i(a_{max}) = \mathbf{U}^{i+1}(0)$ (Adapté de Cochelin *et al.* [2007]).

soit restée en grande partie confinée au domaine de la mécanique des structures [Charpentier et Potier-Ferry, 2008 ; Charpentier, 2012].

La méthode de continuation de haut-degré Diamant [Charpentier et Potier-Ferry, 2008] présentée en section 2.2.3 exploite la différentiation automatique de haut-degré présentée en section 2.3.3, ce qui lui confère des niveaux de généricité et d'automatisation particulièrement intéressants dans notre approche.

2.2.3 Méthode Diamant

La méthode Diamant est une méthode de continuation de haut-degré qui permet la résolution automatisée de problèmes non-linéaires à partir de \mathcal{R} lui-même. La variété des problèmes non-linéaires pouvant être traités est illustrée en tableau 2.3. Un problème non-linéaire résolu par la méthode Diamant est mis sous la forme générique (2.4) :

$$\mathcal{R}(\mathbf{U}(a)) = \mathbf{0}. \quad (2.29)$$

Ce problème est approché sous la forme :

$$\mathcal{R}(\mathbf{U}(a)) \approx \mathcal{R}(\mathbf{U}(0)) + \sum_{k=1}^K a^k \mathcal{R}_k = \mathbf{0}, \forall a \in [0, a_m], \quad (2.30)$$

où $\mathcal{R}(\mathbf{U}(0)) \approx 0$, $[0, a_m]$ est le domaine de validité de l'approximation et \mathcal{R}_k sont les coefficients de Taylor de $(\mathcal{R} \circ \mathbf{U})(a)$ d'ordre k en $\mathbf{U}(0)$.

Comme \mathcal{R} est analytique, cette relation est vérifiée quelque soit a , ce qui implique que :

$$\mathcal{R}_k = \mathbf{0}, \text{ pour } k = 1, \dots, K. \quad (2.31)$$

A chaque ordre k , la séquence (2.31) est un système linéaire en l'inconnue \mathbf{U}_k . La résolution du système d'ordre k de la séquence (2.31) implique l'évaluation du coefficient de Taylor \mathcal{R}_k . Celui-ci peut être calculé automatiquement [Bilasse *et al.*, 2009] au moyen de la décomposition (2.27). La méthode Diamant [Charpentier et Potier-Ferry, 2008 ; Bilasse *et al.*, 2009] propose de résoudre (2.10) à l'ordre k sous la forme :

$$\begin{cases} \{\mathcal{R}_1 | \mathbf{U}_1 = Id\} \mathbf{U}_k = -\{\mathcal{R}_k | \mathbf{U}_k = \mathbf{0}\}, \forall k \geq 1 \\ \langle \mathbf{U}_k, \mathbf{U}_1 \rangle = \delta_{k,1} \end{cases} \quad (2.32)$$

où $\{\mathcal{R}_k | \mathbf{U}_k = \mathbf{0}\}$ ne dépend que des termes aux ordres inférieurs à k . Les dérivées contenues dans $\{\mathcal{R}_1 | \mathbf{U}_1 = Id\}$ et $\{\mathcal{R}_k | \mathbf{U}_k = \mathbf{0}\}$ sont calculées par différentiation automatique de haut-degré comme présenté en section 2.3.3. En pratique, les inconnues \mathbf{U}_k ainsi que le terme de haut-degré $\{\mathcal{R}_k | \mathbf{U}_k = \mathbf{0}\}$ sont évalués alternativement.

L'approximation de la branche de solutions par le polynôme de Taylor $\sum_{k=0}^K a^k \mathbf{U}_k$ est exploitée pour calculer le *domaine de validité* $[0, a_m]$ défini tel que [Cochelin *et al.*, 2007] :

$$\left\| \mathcal{R} \left(\sum_{k=0}^K a^k \mathbf{U}_k \right) \right\| < \varepsilon, \forall a \in [0, a_m], \quad (2.33)$$

où ε est la précision exigée sur le résidu de $\mathcal{R}(\mathbf{U})$. En pratique, a_m est calculé a posteriori :

$$a_m = \left(\frac{\varepsilon}{\|\{\mathcal{R}_{K+1} | \mathbf{U}_{K+1} = \mathbf{0}\}\|} \right)^{\frac{1}{K+1}}, \quad (2.34)$$

Ce calcul *a posteriori* permet :

- de maximiser le domaine de validité du polynôme et par conséquent de minimiser le nombre de résolutions de systèmes,
- d'assurer une précision donnée de la représentation continue de la branche sur le domaine de validité et de limiter les corrections par l'algorithme de Newton-Raphson.

Finalement, les étapes de résolution de la méthode de continuation de haut-degré Diamant sont décrites par l'algorithme (1).

Algorithm 1 – Algorithme de prédiction dans la méthode Diamant pour le calcul d'une série solution de l'équation (2.32).

Point solution initial $\mathbf{U}(0) = \mathbf{U}^i$ connu

Initialisation des \mathbf{U}_k à $\mathbf{0}$

Calcul de la jacobienne $\{\mathcal{R}_{1|\mathbf{U}_1=Id}\}$ en exploitant la différentiation automatique sur \mathcal{R}

Calcul du vecteur tangent \mathbf{U}_1

Assemblage de la jacobienne et de l'équation de chemin

Pour $k = 2$ à K **faire**

Calcul du coefficient $\{\mathcal{R}_{k|\mathbf{U}_k=\mathbf{0}}\}$ par différentiation automatique

Résolution de $\{\mathcal{R}_{1|\mathbf{U}_1=Id}\} \mathbf{U}_k = -\{\mathcal{R}_{k|\mathbf{U}_k=\mathbf{0}}\}$ et $\langle \mathbf{U}_k, \mathbf{U}_1 \rangle = \delta_{k,1}$

Assemblage du polynôme d'ordre k

Fin Pour

Calcul de a_m

$\mathbf{U}^{i+1} = \mathbf{U}(a_m)$

La méthode Diamant permet donc une résolution entièrement automatisée d'un système d'équations non-linéaires écrit sous une forme générique. Cette automatisation repose sur les développements théoriques introduits et sur l'exploitation de la différentiation automatique de haut-degré pour le calcul des dérivées contenues dans $\{\mathcal{R}_{1|\mathbf{U}_1=Id}\}$ et $\{\mathcal{R}_{k|\mathbf{U}_k=\mathbf{0}}\}$.

2.3 Différentiation automatique de codes informatiques

Le calcul des dérivées est une problématique importante autant pour la formulation des problèmes d'analyse de mécanismes que pour la continuation de haut-degré. Le besoin pour une approche de conception générique et automatisée implique une dérivation transparente pour l'utilisateur.

En robotique, les dérivées jouent un rôle central dans l'étude du mouvement des mécanismes. Deux approches pour l'obtention du modèle cinématique sont à distinguer [Bonev, 2002 - p.22]. La première exploite l'étude du mouvement via la théorie des visseurs. Dans cette théorie, l'état de l'effecteur est représenté par un système de torseurs, et leur dérivation permet l'étude des mouvements des corps composant le mécanisme [Bonev 2002 ;

Cervantes-Sánchez *et al.* 2009]. La seconde est celle que nous choisissons d'adopter et consiste en la différentiation des équations du modèle géométrique à l'ordre 1. Haug *et al.* [1996] ont proposé l'utilisation de différentiation automatique par surcharge d'opérateur pour l'établissement et la résolution des systèmes étendus exprimant alors les frontières de l'espace de travail. Nous proposons d'exploiter la différentiation automatique de manière plus large.

Dans cette section, nous mettons en évidence la pertinence de l'approche différentiation automatique (DA). Nous présentons ensuite la DA par transformation de source et la DA par surcharge d'opérateur, toutes deux exploitées dans les outils développés aux chapitres 3, 4 et 5.

2.3.1 Motivation et principe

Les dérivées d'une fonction peuvent être calculées manuellement, numériquement par la méthode des différences finies, ou par différentiation symbolique.

Le besoin pour une approche de conception générique et automatisée écarte le recours à un calcul manuel des dérivées. La méthode des différences finies est facile à mettre en œuvre, mais introduit des erreurs de troncature dans les dérivées. Sa précision dépend du pas de calcul et une dérivation à un ordre élevé est difficilement envisageable. La différentiation symbolique, par exemple à l'aide de l'outil Maple, devient rapidement fastidieuse à mesure que le nombre d'équations et l'ordre de différentiation augmente [Dürrbaum *et al.*, 2002]. Aucune de ces alternatives n'est pleinement satisfaisante dans le cadre de notre approche.

Une alternative est la différentiation des codes informatiques : c'est la *différentiation automatique* (DA). Les outils de DA augmentent un code ou un programme informatique en ajoutant des instructions pour le calcul des dérivées [Griewank et Walther, 2008]. Les dérivées ainsi calculées sont exactes à la précision machine près. Des outils ont été développés pour de nombreux langages informatiques. La DA possède l'avantage d'éviter les erreurs humaines de différentiation et l'imprécision des méthodes approchées. La maturité des outils de DA rend leur prise en main envisageable hors de la communauté des numériciens.

Le principe fondamental de la DA est l'application de la règle de dérivation des fonctions composées à un programme informatique. A l'exécution, un code ou un programme informatique est une séquence d'instructions implémentant des fonctions et des opérations sous forme d'assignation de variables. Le code source peut comporter des milliers de lignes, des boucles et des instructions conditionnelles.

Pour une DA à l'ordre 1, il existe un mode linéaire tangent (direct) exploitant des principes de différentiation vus dans l'enseignement secondaire. Ce mode est adapté au calcul des dérivées directionnelles, de jacobiniennes de dimensions réduites et aux calculs de sensibilité

Fonction	Dérivée	Opération	Dérivée
$u(x)$	$u^{(1)}(x)$	$u + v$	$u^{(1)} + v^{(1)}$
$\exp(u)$	$\exp(u) \cdot u^{(1)}$	$u - v$	$u^{(1)} - v^{(1)}$
$\log(u)$	$\frac{u^{(1)}}{u}$	$u \cdot v$	$u^{(1)} \cdot v + u \cdot v^{(1)}$
$\cos(u)$	$-\sin(u) \cdot u^{(1)}$	$\frac{u}{v}$	$\frac{u^{(1)} \cdot v - u \cdot v^{(1)}}{v^2}$
$\sin(u)$	$\cos(u) \cdot u^{(1)}$	u^n	$n \cdot u^{(1)} \cdot u^{n-1}$ si n entier

TABLEAU 2.4 – Formules de dérivation des fonctions et opérations élémentaires.

par rapport à un nombre réduit de paramètres. Le mode adjoint (inverse), adapté au calcul du gradient d'un vecteur d'état de grande dimension [Griewank et Walther, 2008], n'est pas utilisé dans cette thèse.

Deux types d'implémentation de DA sont disponibles. La DA par *transformation de source* génère les instructions constituant un code différencié. La DA par *surcharge d'opérateur* exploite la programmation orientée objet pour travailler avec des variables de type polynômes de Taylor. Nous nous intéressons plus particulièrement aux logiciels exploitables avec des codes Matlab et Fortran.

2.3.2 Différentiation par transformation de source

Un outil de DA par *transformation de source* génère automatiquement un *code différencié* à partir d'un *code source* fourni par l'utilisateur. Cette approche produit un code différencié lisible et permet d'évaluer des dérivées directionnelles.

2.3.2.1 Logiciels

Des outils de DA par transformation de source sont disponibles en Fortran, en Matlab et marginalement en C. Un outil de DA par transformation de source de qualité requiert de nombreuses années de développement et de débogage afin de prendre en compte toutes les spécificités de syntaxe et de structure du langage. Il existe 6 outils de DA en langage Matlab d'après le site internet de référence Autodiff [2017]. Parmi ceux-ci, seuls 3 sont décrits comme implémentant la transformation de source :

- ADiGator exploite une approche basée sur la surcharge d'opérateur. Il ne s'agit donc pas d'un réel outil de transformation de source. De plus il a été développé récemment, et il y a donc un manque de recul sur son utilisation pratique ;
- ADiMat adopte une approche hybride et produit un code source différencié qui est moyennement lisible [Lampoh, 2012 - p.54] ;

— TomSym est une classe du logiciel propriétaire TOMLAB.

L'absence d'un outil simple d'utilisation en Matlab, l'existence d'une version de Diamant codé en Fortran 95, le faible nombre d'instructions des codes à différentier dans notre approche d'analyse de mécanismes et l'expérience issue de travaux antérieurs proches [Lampoh *et al.*, 2011 ; Lampoh, 2012 - Section 2.3] ont mené à l'utilisation du logiciel Tapenade pour la DA. Tapenade est un outil de DA par transformation de codes source C ou Fortran [Hascoet et Pascual, 2013].

En pratique l'utilisateur fournit un code source Fortran implémentant un problème. Il doit définir les variables de différentiation (dites indépendantes) qui constituent l'entrée du code et les variables (dites dépendantes) qui constituent la sortie du code. La traduction de codes Matlab en codes Fortran 95 et inversement est très aisée dans le cas des codes étudiés en analyse des mécanismes. Le code différentié obtenu est lisible, comme montré dans la suite.

2.3.2.2 Mode linéaire tangent

Prenons l'exemple de la fonction :

$$\begin{aligned} f : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ (x,y) &\longmapsto f(x,y) = \sqrt{x^2 + y^2} = z \end{aligned} \quad (2.35)$$

Supposons que l'on souhaite évaluer la dérivée de f par rapport aux variables x ou y . La fonction est implémentée dans un code source Fortran 95 dans la routine `NORME` dont les arguments sont les variables indépendantes x et y et la variable dépendante z . Ce code source est différentié avec Tapenade par rapport à x et y afin de générer un code différentié

qui permet d'évaluer la dérivée directionnelle $\begin{pmatrix} \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \end{pmatrix}$.

Le fonctionnement de Tapenade est comparable à celui d'un compilateur. Une analyse syntaxique est menée sur le code fourni par l'utilisateur afin d'écarter toute erreur de programmation et de signaler l'éventuelle présence d'une structure syntaxique non-supportée par Tapenade pour le langage choisi :

```

1  !           Generated by TAPENADE      (INRIA, Ecuador team)
2  ! Tapenade 3.12 (r6213) – 13 Oct 2016 10:54
3  !
4  ! Top routine
5  SUBROUTINE NORME(z, x, y)
6  IMPLICIT NONE
7  ! Independent input variables

```

```

8  DOUBLE PRECISION :: x, y
9  ! Dependent output variables
10 DOUBLE PRECISION :: z
11  INTRINSIC SQRT
12  DOUBLE PRECISION :: arg1
13  arg1 = x**2 + y**2
14  z = SQRT(arg1)
15 END SUBROUTINE NORME

```

Les instructions contenant des compositions de fonctions sont décomposées par l'introduction de variables intermédiaires. Par exemple $\text{SQRT}(x**2 + y**2)$ est modifiée en $\text{SQRT}(\text{arg1})$ où $\text{arg1} = x**2 + y**2$ est une variable intermédiaire.

Si le code ne comporte pas d'erreurs syntaxiques, celui-ci est différentiable. L'utilisateur fournit :

- le code,
- le mode de différentiation, ici mode tangent,
- la routine principale, ici NORME,
- les variables dépendantes, ici la variable z dont la dérivée est recherchée,
- les variables indépendantes, ici les variables x et y par rapport à laquelle on dérive,

avec le code :

```

1  !           Generated by TAPENADE      (INRIA, Ecuador team)
2  ! Tapenade 3.12 (r6213) – 13 Oct 2016 10:54
3  ! Differentiation of residual in forward (tangent) mode:
4  ! variations of useful results: f
5  ! with respect to varying inputs: x y
6  ! RW status of diff variables: f:out x:in y:in
7  ! Top routine
8  SUBROUTINE NORME_D(z, zd, x, xd, y, yd)
9  IMPLICIT NONE
10 DOUBLE PRECISION :: x, y, xd, yd ! Independent input variables
11 DOUBLE PRECISION :: z, zd ! Dependent output variables
12 INTRINSIC SQRT
13 DOUBLE PRECISION :: arg1, arg1d
14 arg1d = 2*x*xd + 2*y*yd
15 arg1 = x**2 + y**2
16 IF (arg1 .EQ. 0.0) THEN
17     zd = 0.D0
18 ELSE

```

```

19     zd = arg1d / (2.0 * SQRT( arg1 ))
20 END IF
21 z = SQRT( arg1 )
22 END SUBROUTINE NORME_D

```

Le code différencié contient la routine différenciée `NORME_D` et respecte les conventions de Tapenade :

- les variables dérivées sont suffixées de la lettre `d`, et sont introduites comme arguments de la routine pour chaque variable indépendante et dépendante,
- le type d'une variable dérivée est déclaré identique à la variable source,
- une instruction différenciée est introduite au dessus de chaque instruction source dépendante d'une variable indépendante et dont dépend la variable dépendante,
- l'instruction du code direct est reproduit pour fournir la trajectoire lors de l'évaluation des dérivées.

Pour comprendre l'utilisation du code obtenu, on peut par exemple chercher à évaluer la dérivée de $f(x, y) = z$ par rapport à la variable x au point $(x_0, y_0) = (2, 3)$. L'évaluation de la dérivée est effectuée par l'appel de la routine `NORME_D` avec les arguments suivants :

- la direction de différenciation, définie dans la base des variables indépendantes : on initialise les variables indépendantes dérivées `xd=1` et `yd=0` ;
- le point d'évaluation : on initialise les variables indépendantes `x=2` et `y=3`.

Le résultat est constitué de la valeur de la variable `z` et de la valeur de sa dérivée `zd`.

2.3.3 Différenciation par surcharge d'opérateur

2.3.3.1 Mode linéaire tangent

La *surcharge d'opérateur* repose sur la programmation orientée objet. Elle consiste à redéfinir les opérations mathématiques et les fonctions élémentaires intrinsèques à un langage de programmation pour un nouveau type de variables informatiques créé par le développeur.

Une classe contient :

- un type et ses *attributs* pour décrire des objets,
- des *méthodes* représentant leur comportement,
- une méthode, appelée *constructeur*, permettant de construire un objet hérité de la classe par l'assignation à une variable du résultat.

Par exemple pour le calcul des dérivées premières, on peut définir la classe `@Deriv` possédant les attributs `val` et `deriv`. Considérons maintenant une variable réelle `x` dont la dérivée est notée `xd`. En créant un objet `X` de classe `@Deriv`, on peut stocker la variable et

sa dérivée dans une unique variable :

$$X = \begin{cases} X.\text{val} = x \\ X.\text{deriv} = xd \end{cases} \quad (2.36)$$

La surcharge des opérations et fonctions pour des variables de type `@Deriv` implémente les formules de dérivation du tableau 2.4. Par exemple le produit $Z = X * Y$ de deux variables de type `@Deriv` est effectué au moyen de la formule de dérivation du produit de fonctions :

$$Z = \begin{cases} Z.\text{val} = X.\text{val} * Y.\text{val} \\ Z.\text{deriv} = X.\text{deriv} * Y.\text{val} + X.\text{val} * Y.\text{deriv} \end{cases} \quad (2.37)$$

La surcharge d'opérateur permet donc de propager la valeur des dérivées des variables lors de l'évaluation des instructions du code source.

2.3.3.2 Haut-degré

Pour la *différentiation automatique de haut-degré* (DAHD), le nouveau type de variables contient les informations nécessaires à la représentation d'un polynôme de Taylor. Un objet `t` de classe `@Taylor` a pour attributs :

$$t = \begin{cases} t.\text{value} \\ t.\text{order} \\ t.\text{coef} \end{cases} \quad (2.38)$$

qui correspondent respectivement au point $U(0)$, à l'ordre K et aux coefficients de Taylor $U_k, k = 1 \dots K$. La classe `@Taylor` est une classe du logiciel `Diamanlab` présenté en section 2.4.

La classe `@Taylor` contient des méthodes spécifiques servant à la manipulation des polynômes de Taylor telle que la fonction `evalseries(t, a, k)`. Celle-ci permet d'évaluer le polynôme de Taylor stocké dans la variable Taylor `t` en un point `a`, à l'ordre `k`. Son code source est le suivant :

```
function [val] = evalseries(t,a,k)
val = t.value ; ak = 1;
3 for i=1:k % Accumulation des termes par recurrence
    ak = ak*a; % a^i
    val = val + t.coef(:,i)*ak; % ...+...+ t.coef_(i)*a^i
end
```

La DAHD repose sur la surcharge d'opérateur pour les variables de type Taylor et implémente des formules de récurrence pour la dérivation de haut-degré.

Formules de récurrence Soient deux variables u et v représentant respectivement les approximations de fonctions $u(a)$ et $v(a)$ par des polynômes de Taylor en 0. La DAHD exploite des formules de récurrence généralisant les formules de dérivation des fonctions et opérations aux ordres élevés [Griewank et Walther, 2008], voir tableau 2.5. Par exemple la DAHD du produit $w = uv$ de deux variables représentant des polynômes de Taylor implémente la formule de Leibniz pour calculer le coefficient de Taylor w_k :

$$w_k = (u.v)_k = \sum_{j=0}^k u_j.v_{k-j},$$

où u_k , v_k et w_k sont respectivement les coefficients de Taylor des fonctions $u(a)$, $v(a)$ et $w(a)$.

Opération	Dérivée d'ordre k
$w = u + v$	$w_k = u_k + v_k$
$w = u.v$	$w_k = \sum_{j=0}^k u_j.v_{k-j}$
$w = \frac{u}{v}$	$w_k = \frac{1}{v_0} \left(u_k - \sum_{j=0}^k w_j.v_{k-j} \right)$
Fonction	
$w = \sqrt{u}$	$w_k = \frac{1}{2w_0} \left(u_k - \sum_{j=0}^{k-1} w_j.w_{k-j} \right)$
$w = \exp(u)$	$\tilde{w}_k = \sum_{j=1}^k w_{k-j}.\tilde{u}_j$
$w = \log(u)$	$\tilde{w}_k = \frac{1}{u_0} \left(\tilde{u}_k - \sum_{j=1}^k u_{k-j}.\tilde{w}_j \right)$
$w = u^r$	$\tilde{w}_k = \frac{1}{u_0} \left(r \sum_{j=1}^k w_{k-j}.\tilde{u}_j - \sum_{j=1}^{k-1} u_{k-j}.\tilde{w}_j \right)$

TABLEAU 2.5 – Formules de récurrence pour le calcul des dérivées d'ordre k de fonctions et opérations élémentaires. Les variables mises en valeur par un tilde sont telles que $\tilde{u}_k = ku_k$.

Méthodes de surcharge d'opérateur Les autres méthodes de la classe `@Taylor` implémentent ces formules de récurrence au moyen de la surcharge d'opérateur. La bibliothèque de surcharge d'opérateur du logiciel Diamanlab est constituée de méthodes placées dans le répertoire de la classe `@Taylor`.

Un exemple est la fonction `exp(u)` d'une variable de type Taylor `u`. Le code source de la méthode correspondante est :

```

function v = exp(u)
% @Taylor/exp est la formule de recurrence pour l'exponentielle
global Ck
4
co = u.order;
cu = u.coef;
for k=1:Ck
    tcu(:,:,k)=k*cu(:,:,k);
9 end
value=exp(u.value);
cv=cu*0; tcv=cv;
for k=1:Ck
    tcv(:,:,k) = times(value,tcu(:,:,k));
14 for j=1:k-1
        tcv(:,:,k) = tcv(:,:,k) + times(cv(:,:,k-j),tcu(:,:,j));
    end
    cv(:,:,k) = tcv(:,:,k)/k;
end
19 v = Taylor(co,value,cv);

```

Le langage Matlab est un langage haut-niveau conçu pour le calcul matriciel, ainsi toutes les variables sont gérées sous forme de tableaux. La méthode `exp(u)` pour des variables de type Taylor fait apparaître des tableaux. Afin de rendre la programmation la plus simple dans le cadre de calculs impliquant des scalaires, des vecteurs, des matrices, voire des tableaux de dimension supérieure, il exploite intensivement la technique de surcharge d'opérateur.

2.4 Diamanlab : une implémentation Matlab de Diamant

La méthode Diamant présentée en section 2.2.3 exploite la différentiation automatique de haut-degré par surcharge d'opérateur présentée en section 2.3.3.2 pour proposer une méthode de continuation de haut-degré générique. Diamanlab [Charpentier *et al.*, 2013] est un logiciel

implémentant cette méthode en langage Matlab. Nous présentons le fonctionnement de Diamanlab et illustrons ses limites dans le cadre de notre approche.

2.4.1 Logiciels de continuation de haut-degré

Manlab Les développements principaux de la méthode asymptotique numérique présentée brièvement en section 2.2.2 ont conduit à la production du logiciel de continuation en Matlab nommé Manlab [Cochelin *et al.*, 2009]. Manlab est à la fois un solveur de systèmes d'équations non-linéaires de haut niveau et un logiciel de continuation assisté par des fonctionnalités de détection de points de bifurcation. Il propose une interface graphique. L'approximation des branches de solutions par une collection de polynômes de Taylor est réalisée par *continuation interactive* alternant :

- la continuation le long d'une branche par le calcul automatique d'un nombre de polynômes consécutifs défini par l'utilisateur,
- le changement de branches de solutions assisté par l'utilisateur via l'interface graphique.

Diamanlab Son successeur Diamanlab combine la robustesse et l'aspect interactif de ManLab avec les atouts de la DA et de la programmation orientée objet prônés par la méthode Diamant. L'utilisateur implémente le système d'équations comme une fonction Matlab, définit un point initial et exploite l'interface graphique pour établir le diagramme de bifurcation. La résolution d'un problème (2.4) exploite le principe de la continuation interactive de ManLab. Nous verrons en section 3.2.2 que la continuation interactive n'est pas adaptée à la détermination de l'espace de travail. Diamanlab fonctionne avec les versions de Matlab 7.0 ou supérieur sans aucune bibliothèque tierce. Notons que la classe `@Taylor` est exploitable hors du cadre de la continuation, ce que nous ferons dans le cadre de l'analyse de sensibilité qui sera menée au chapitre 5.

2.4.2 Structure de la bibliothèque

La généralité dérive de Diamant, de la DA et de la programmation orientée objet. Nous présentons ici les éléments du logiciel Diamanlab nécessaires à la compréhension et à l'évaluation des modifications apportées en section 3.2.2.

Avec Diamanlab, tout système d'équations non-linéaires satisfaisant (2.4) peut être codé dans une classe `@UserSyst` comprenant un constructeur et une méthode `R` implémentant le système d'équations que l'utilisateur souhaite résoudre. La classe `@UserSyst` dérive

de la classe `@Syst`, comme le montre le diagramme de classes présenté en figure 2.7. `@UserSyst` contient des méthodes spécifiques au problème traité : la méthode `R` codant le système d'équations (2.29) à résoudre et éventuellement une méthode `disp` définissant une représentation spécifique des solutions. Son entrée et sa sortie peuvent être des variables de type `@Taylor` ou `@double`.

Diamanlab est structuré en 4 classes, comme décrit par le diagramme de classes de la figure 2.7. Par exemple la classe Taylor de DAHD par surcharge d'opérateur correspond au répertoire `@Taylor` et contient le fichier constructeur `Taylor.m` et d'autres méthodes.

Les 3 autres classes sont dédiées à la gestion interactive de la continuation (`@ContDriver`), à la manipulation des systèmes d'équations non-linéaires (`@Syst`) et au stockage des informations sur les polynômes (`@CheckPoint`). Nous présentons ici les éléments qui sont exploités pour la détermination de l'espace de travail et des singularités.

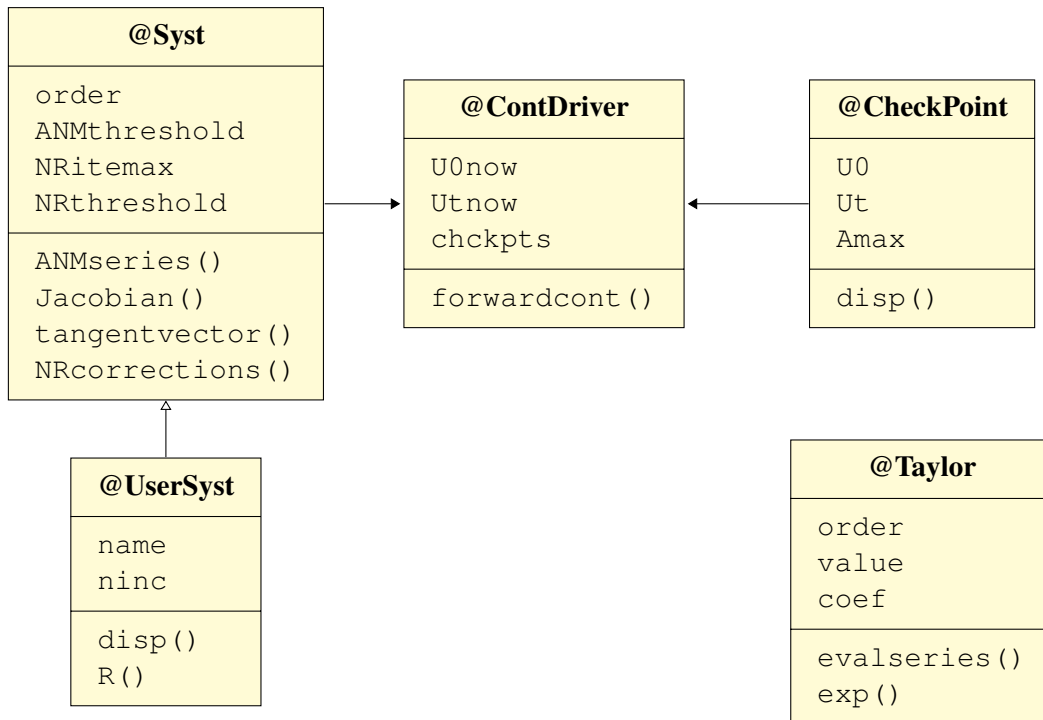


FIGURE 2.7 – Diagramme de classes du logiciel Diamanlab avec attributs et méthodes

2.4.2.1 Gestion d'une branche

La méthode de continuation Diamant nécessite la définition d'un certain nombre de paramètres de résolution numérique, tels que l'ordre de troncature K , la précision ε exigée sur le résidu et la tolérance de précision ε_{nr} de l'algorithme de Newton-Raphson. Ces paramètres sont définis par défaut et peuvent être adaptés. Ils sont stockés dans un objet

@UserSys de la classe @Sys. Deux propriétés dépendantes du problème indiquent son nom `name` et sa dimension `ninc`. Les propriétés principales de la classe @Sys sont précisées dans le tableau 2.6. Les méthodes `Jacobian`, `tangentvector`, `NRcorrections` et `ANMseries` de la classe @Sys sont utilisées pour la résolution du système.

Variable	Type	Init	Description
<code>name</code>	Texte		Nom du problème
<code>ninc</code>	Scalaire		Nombre d'inconnues
<code>order</code>	Scalaire	20	Ordre de troncature K des polynômes de Taylor
<code>ANMthreshold</code>	Scalaire	1e-6	Critère de précision ε pour le domaine de validité
<code>NRitemax</code>	Scalaire	10	Nombre maximal de corrections N_{nr}
<code>NRthreshold</code>	Scalaire	1e-6	Précision du résidu ε_{nr} exigée pour le N-R

TABLEAU 2.6 – Attributs de la super-classe @Sys.

La méthode `ANMseries` prend en charge le calcul des séries, *i.e.* de la résolution des équations (2.32). Ses variables d'entrée sont un objet de la classe @UserSys représentant le système, la valeur `U0val` d'une variable `U0` de type @Taylor représentant le point de développement, et la direction de la tangente `Ut0`. Ses variables de sortie sont la série `U0`, son domaine de validité `Amax` et les données concernant une potentielle bifurcation `BifData`. L'algorithme de résolution 1 est codé de la manière suivante :

```

1 function [U0,Amax, BifData] = ANMseries(sys,U0val,Ut0)
% Compute ANM series for sys(U0val) in the tangent direction Ut0
  global Ck % Current diff. order
  dRdU = Jacobian(sys,U0val);
  U1 = tangentvector(sys,U0val,dRdU); if (U1'*Ut0) < 0, U1 = - U1;
  end
6 % ANM Tangent Matrix including the path equation
  K = [dRdU ; (sys.arclengthdef.*U1)'];
  [LK,UK,pK,qK] = lu(K,'vector');
% series computation : Zero and First order
  U0=Taylor(get(sys,'order'),U0val); U0=set(U0,'coef1',U1);
11 % series computation : Orders Ck\ensuremath{\geq}2
  for Ck=2:sys.order
    rhs=[get(-R(sys,U0),'coefk',Ck);0]; % right-handside term with
      U0_Ck=0
    Usol(qK)= UK\ (LK\rhs(pK)); % solution
    U0=set(U0,'coefk',Usol,Ck); % set Taylor coef U0_Ck
16 end

```

```

% domain of utility Amax
Amax=1;
if (norm(rhs)~=0)
    Amax = (sys.ANMthreshold/norm(rhs))^(1/(sys.order));
end
% Detection and extraction of emerging geometric series
% ... see the source code ...
end

```

La fonction Matlab `ANMseries` appelle d'abord la méthode `Jacobian` pour calculer la jacobienne $\{\mathcal{R}_1|_{\mathbf{U}_1=Id}\}$. Ensuite, la méthode `tangentvector` calcule une tangente à la branche de solutions au point $\mathbf{U}(0)$. La jacobienne $\{\mathcal{R}_1|_{\mathbf{U}_1=Id}\}$ est assemblée avec le terme linéarisant l'équation de chemin dans la matrice \mathbb{K} qui est factorisée selon la méthode `LU`. La série `U0` est initialisée par la donnée de sa valeur `U0val` (ordre 0), de son ordre et de son vecteur tangent `U1` (ordre 1). Les coefficients de la série sont alors calculés de manière itérative à partir de l'ordre 2. Pour chaque itération, le terme $rhs = -\{\mathcal{R}_k|_{\mathbf{U}_k=0}\}$ est évalué, puis le système (2.32) est résolu. Le domaine de validité de la série `Amax` est finalement calculé.

La méthode `NRcorrections` implémente l'algorithme de Newton-Raphson pour améliorer la précision des solutions si nécessaire. Les méthodes `ANMserieBif`, `GeomSerie` et `locatebif`, sont dédiées au traitement des points de bifurcations [Charpentier *et al.*, 2013].

2.4.2.2 Stockage des morceaux de branche de solutions

Les branches de solutions sont calculées sous forme de polynômes de Taylor. Un objet de classe `@Checkpoint` est créé au cours du processus de continuation à chaque fois qu'un polynôme est calculé, puis stocké. Les propriétés principales de la classe `@Checkpoint` sont décrites dans le Tableau 2.7.

Variable	Type	Init	Description
<code>U0</code>	Taylor	0	Point de développement et valeur des coefficients du polynôme
<code>Ut</code>	Vecteur	0	Direction de continuation au point initial
<code>Amax</code>	Scalaire	0	Domaine de validité du polynôme
<code>params</code>	Structure	/	Paramètres d'affichage
<code>BifData</code>	Structure	/	Point de bifurcation et série géométrique

TABLEAU 2.7 – Propriétés de la classe `@Checkpoint`

La méthode `evalseries` permet d'évaluer un checkpoint pour construire un morceau de branche de solutions. Le stockage des polynômes sous forme de checkpoints est adapté à la continuation interactive. L'utilisateur peut sélectionner un point d'une branche de solutions affichée sur le diagramme de bifurcation. Le checkpoint le plus proche est alors défini comme point courant et la continuation peut reprendre.

L'information contenue dans un checkpoint est riche et précise. Nous proposons le post-traitement de cette information pour la classification des singularités étudiée au chapitre 4.

2.4.3 Continuation avec Diamanlab et interface graphique

La classe `@ContDriver` implémente la gestion du processus de continuation. Ses propriétés sont le point courant de continuation et une liste d'objets `@CheckPoint` stockés au long du processus. L'algorithme de continuation interactive est présenté en 2

Algorithm 2 – Algorithme de continuation interactive dans la version initiale de Diamanlab. Méthode `forwardcontinuation`.

```

Pour  $i = 1$  à  $nbSeries$  faire
  Si  $|\mathbf{R}(\mathbf{U}_0)| > NRthreshold$  Alors
    Corrections de Newton-Raphson
  Fin Si
  Si Point non-singulier Alors
    Calcul des termes  $\{\mathbf{U}_k\}_{k=1,\dots,K}$  de la série et domaine de validité  $a_m$  (Algorithme 1)
  Sinon
    Calcul de la série  $\mathbf{U}^b$  sur la seconde branche au point de bifurcation [Charpentier
    et Cochelin, 2016]
     $\mathbf{U} = \mathbf{U}^b$ 
  Fin Si
  Création d'un objet checkpoint
  Stockage du checkpoint
   $\mathbf{U}_0 = \mathbf{U}(a_m)$  et  $\mathbf{U}_t = \mathbf{U}'(a_m)$ 
Fin Pour

```

A l'exécution, Diamanlab lance l'interface graphique et une figure pour tracer le future diagramme de bifurcation. La continuation interactive est pilotée par l'interface graphique et contrôlée sur le diagramme de bifurcation. L'interface graphique est divisée en 5 cadres :

- **Continuation** : permet de lancer i pas de continuation avec une précision ε (voir équation (2.34)). La méthode appelée est `forwardcontinuation`.
- **Correction** : permet d'activer les corrections de Newton-Raphson quand la précision d'un point solution dépasse le seuil de tolérance ε_{nr} .

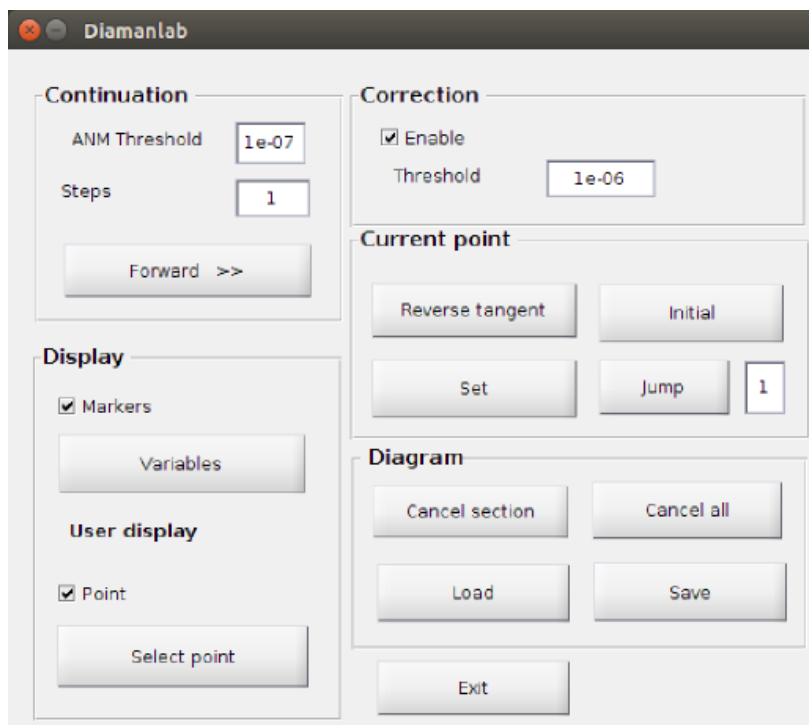


FIGURE 2.8 – Interface graphique utilisateur de Diamanlab.

- **Current point** : permet d’agir sur le point courant. Le bouton **Reverse tangent** permet d’inverser la direction de continuation. Le bouton **Set** permet de positionner le point courant sur une branche de solutions déjà calculée en cliquant sur le diagramme de bifurcation. Ceci permet en particulier de parcourir une nouvelle branche de solutions en cliquant sur un point de bifurcation. Le bouton **Jump** permet de lancer une étape de correction à partir d’un nouveau point afin par exemple de découvrir une branche de solutions qui n’est pas connectée à celle initialement parcourue.
- **Display** : **Variables** permet de modifier les variables affichées sur le diagramme de bifurcation, **Markers** permet d’afficher les checkpoints. **User display** permet d’afficher la représentation du problème définie par l’utilisateur.
- **Diagram** : permet de sauvegarder, modifier et charger les résultats stockés comme une collection d’objets checkpoints.

Afin de mieux répondre au besoin de l’outil de détermination de l’espace de travail d’un mécanisme, une nouvelle méthode de parcours est développée au chapitre 3. Cette proposition permet de proposer un parcours automatisé d’un jeu de branches de solutions connectées, contrairement au parcours manuel possible en l’état.

2.5 Conclusion

Dans ce chapitre nous avons décrit les outils mathématiques utilisés pour développer notre outil générique, automatisé, efficace et fiable d'évaluation des caractéristiques d'un mécanisme. Il s'agit :

- d'une approche générique de résolution des problèmes non-linéaires,
- de la méthode de continuation de haut-degré Diamant,
- de la différentiation automatique par transformation de source en mode linéaire tangent et de la différentiation automatique de haut-degré par surcharge d'opérateur.

Notre approche de résolution de problèmes de conception recourt à :

- des dérivées d'ordre 1 du modèle géométrique implicite, par rapport aux coordonnées géométriques pour le calcul du modèle cinématique implicite. Celles-ci peuvent être obtenues à l'aide de la différentiation automatique par transformation de source, qui permet d'évaluer des dérivées d'ordre 1 via la génération d'un code source différentié. Ce code linéaire tangent peut aussi être exploité pour l'évaluation des dérivées nécessaires à la détection et la classification des singularités abordé au chapitre 4 ;
- des dérivées d'ordre élevé dans la méthode de continuation Diamant et dans le cadre de l'analyse de sensibilité de haut-degré. Pour ce faire, nous utilisons la différentiation automatique par surcharge d'opérateur qui permet d'évaluer des dérivées de haut-degré à l'exécution du code de manière efficace et transparente pour l'utilisateur.

Ces méthodes et outils numériques sont utilisés pour la représentation des frontières de l'espace de travail par des branches de solutions continues au chapitre 3, pour l'analyse de singularité des mécanismes au chapitre 4 et pour l'analyse de sensibilité au chapitre 5.

Chapitre 3

Détermination de l'espace de travail

« Aujourd'hui on peut faire de la musique avec des ordinateurs, mais l'ordinateur a toujours existé dans la tête des compositeurs »,

Milan Kundera.

Sommaire

3.1	Etat de l'art	68
3.1.1	Méthodes algébriques et géométriques	69
3.1.2	Méthodes numériques	72
3.1.3	Méthode numérique d'analyse par intervalles	75
3.1.4	Méthodes de continuation d'ordre 1	79
3.1.5	Méthode de continuation de haut-degré	81
3.2	Continuation de haut-degré	81
3.2.1	Formulation automatique des frontières de l'espace de travail	82
3.2.2	Détermination des frontières continues de l'espace de travail	85
3.3	Application à un mécanisme plan	86
3.3.1	Mécanisme RRR avec liaisons à débattements limités	87
3.3.2	Espace des positions accessibles 2D	88
3.3.3	Evaluation de l'efficacité	89
3.4	Application à un mécanisme spatial	91
3.4.1	Le mécanisme orthoglide	91
3.4.2	Stratégie de discrétisation	92
3.4.3	Espace de travail 3D	94
3.5	Conclusion	94

La première caractéristique du mécanisme étudiée est son espace de travail. Ce chapitre traite du développement d'un outil numérique pour sa détermination.

La section 3.1 présente un état de l'art des méthodes de détermination de l'espace de travail. Les limites identifiées permettent de préciser le besoin introduit au chapitre 1 d'une méthode à la fois générique, efficace et robuste. Un nouvel outil est présenté en conséquence en section 3.2. L'espace de travail est décrit par son enveloppe dont l'expression est dérivée automatiquement grâce à la différentiation automatique avec l'outil Tapenade. La détermination de l'enveloppe est ensuite réalisée avec l'outil de continuation de haut-degré Diamanlab. L'efficacité de l'outil est évaluée sur un mécanisme plan en section 3.3 puis l'outil est évalué sur un espace de travail 3D en section 3.4 avec l'exemple du mécanisme Orthoglide. Une discussion des résultats est finalement menée en section 3.5.

3.1 Etat de l'art

L'espace de travail \mathcal{W} défini par l'équation (1.6) comme l'ensemble des coordonnées opérationnelles accessibles peut être de dimension 6, ce qui pose un problème de représentation. Il est parfois par ailleurs intéressant de se limiter à certains sous-espaces selon la tâche de manipulation étudiée. Par exemple l'espace des positions accessibles est défini comme l'ensemble des positions que l'effecteur peut atteindre pour au moins une orientation. L'espace primaire ou dextre est défini comme l'ensemble des positions accessibles par l'effecteur avec toutes les orientations possibles. L'évaluation de l'espace de travail passe donc classiquement par la détermination de projections dans un espace de dimension 3 [Kumar et Waldron, 1981 ; Chedmail *et al.*, 1998 ; Bonev et Gosselin, 2006 ; Merlet, 2006].

L'espace des positions accessibles est au maximum de dimension 3. En dimension 3, il peut être vu comme un volume. Deux approches peuvent alors être envisagées :

- la représentation du volume occupé,
- ou bien la représentation de l'enveloppe de ce volume.

Les approches d'analyse de l'espace de travail peuvent être distinguées selon ces modes de représentation. Elles diffèrent par ailleurs par la méthode de détermination sur un plan mathématique. On a dans la littérature d'une part des méthodes algébriques ou géométriques et par ailleurs des méthodes numériques. Nous les analysons ici pour mettre en évidence leurs avantages et inconvénients au regard des besoins exprimés pour notre approche d'évaluation de l'espace de travail de mécanismes, à savoir : généricité, efficacité et robustesse.

3.1.1 Méthodes algébriques et géométriques

3.1.1.1 Méthode basée sur la théorie des torseurs

Sugimoto *et al.* [1982] exploitent la théorie des torseurs pour exprimer les conditions d'appartenance aux frontières de l'espace de travail d'un point de l'effecteur de mécanismes sériels. Le mouvement instantané de chaque liaison du mécanisme est représenté par un visseur cinématique. Lorsque le mécanisme est dans une configuration à la frontière de l'espace de travail, l'effecteur ne peut se déplacer dans la direction normale à la frontière, autrement dit il perd un degré de liberté. Ainsi les visseurs cinématiques, bien que non-nuls, ne permettent pas d'agir indépendamment sur chacun des degrés de liberté de l'effecteur. La détermination des frontières de l'espace de travail consiste donc à trouver toutes les configurations où le système de visseurs cinématiques dégénère [Sugimoto *et al.*, 1982]. La dégénérescence correspond à la dépendance linéaire de visseurs qui peut être déterminée algébriquement grâce à la condition d'annulation d'un déterminant. Avec cette méthode, on recherche donc toutes les conditions algébriques d'occurrence des singularités.

Agrawal [1991] ont proposé une méthode basée sur la théorie des torseurs pour la détermination des frontières de l'espace de travail de mécanismes parallèles.

Avantages La méthode a l'avantage de permettre un lien entre configuration singulière en bord d'espace de travail et représentation géométrique. Ces conditions sont généralement simples pour des mécanismes sériels, mais leur interprétation devient complexe pour des mécanismes parallèles.

Inconvénients Une méthode pour transformer les conditions implicites en une représentation de l'espace de travail est nécessaire et n'existe pas à notre connaissance de manière générique.

3.1.1.2 Méthode géométrique

Cette méthode repose sur la manipulation de primitives géométriques représentant les mouvements autorisés par les degrés de liberté des liaisons du mécanisme étudié. Dans le cas d'un mécanisme parallèle, on détermine pour chaque jambe un objet géométrique qui décrit toutes les poses de l'effecteur satisfaisant aux contraintes de cette jambe. L'espace de travail est constitué de l'intersection de tous les objets. Les intersections sont déterminées au moyen d'opérations définies dans une bibliothèque de géométrie algorithmique. Les travaux de Gosselin [1990], Merlet [1992, 1995] et Merlet *et al.* [1998] sont représentatifs de ces méthodes géométriques.

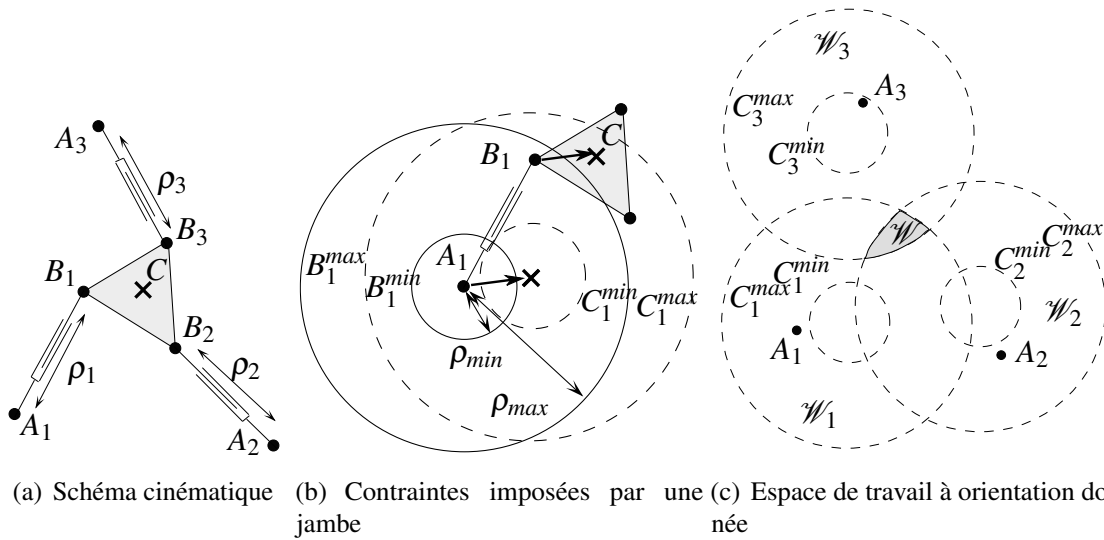


FIGURE 3.1 – (a) Schéma cinématique du mécanisme 3 – RPR . (b) Primitives géométriques en lignes pointillées générées par les positions du point C autorisées par la jambe 1 du mécanisme 3 – RPR pour une orientation de l'effecteur constante. (c) Espace de travail à orientation constante (en gris) déterminé par l'intersection des primitives [Adapté de Merlet, 2006 - Section 7.3.1.1].

Le principe peut être illustré par l'exemple du mécanisme parallèle 3 – RPR tel que décrit dans [Merlet, 2006 - Section 7.3.1.1] et représenté en figure 3.1(a). L'ensemble des positions C de l'effecteur accessibles pour une orientation donnée est recherché. Les points A_i sont fixes sur la base, chaque jambe A_iB_i est constituée d'une liaison pivot en A_i en série avec une liaison prismatique d'allongement ρ_i et d'une seconde liaison pivot en B_i . Les liaisons prismatiques actionnées ont un débattement limité entre ρ_{min} et ρ_{max} . Comme illustré en figure 3.1(b), deux cercles B_i^{min} et B_i^{max} de rayon ρ_{min} et ρ_{max} et de centre A_i constituent alors les frontières des positions admissibles par les points B_i . Considérons la jambe A_1B_1 du mécanisme. L'orientation de l'effecteur est supposée constante, donc le vecteur $\overrightarrow{B_1C}$ conserve son orientation relative à $\overrightarrow{A_1B_1}$ lorsque le point B_1 tourne autour du point A_1 . Les positions admissibles pour le point C sont donc comprises dans deux cercles C_1^{min} et C_1^{max} correspondant à une translation des cercles B_1^{min} et B_1^{max} selon le vecteur $\overrightarrow{B_1C}$. Pour que les contraintes de la jambe i sur l'effecteur soient satisfaites, le point C doit donc être situé entre les deux cercles en pointillés C_i^{min} et C_i^{max} . L'espace de travail à orientation constante est l'intersection des zones \mathcal{W}_i comme indiqué en gris sur la figure 3.1(c).

Avantages Une fois que la méthode a été développée pour une architecture particulière, l'estimation de l'espace de travail et de son volume est très rapide et précise. La représentation

qui résulte d'intersections entre des objets géométriques simples est continue et permet de comprendre l'impact des différentes contraintes sur l'espace de travail.

Inconvénients Il est difficile de prendre toutes les contraintes en compte, comme par exemple un débattement limité des liaisons passives. La méthode géométrique est spécifique : un raisonnement doit être construit pour chaque nouvelle architecture.

3.1.1.3 Méthode algébrique

Les équations de fermeture vectorielle d'une chaîne cinématique liant les coordonnées actives et opérationnelles [Abdel-Malek et Yeh, 1997] ou la modélisation de Denavit-Hartenberg liant des coordonnées homogènes [Ahmad et Luo, 1988] permettent l'obtention d'une expression algébrique liant la pose de l'effecteur à l'état des liaisons actives.

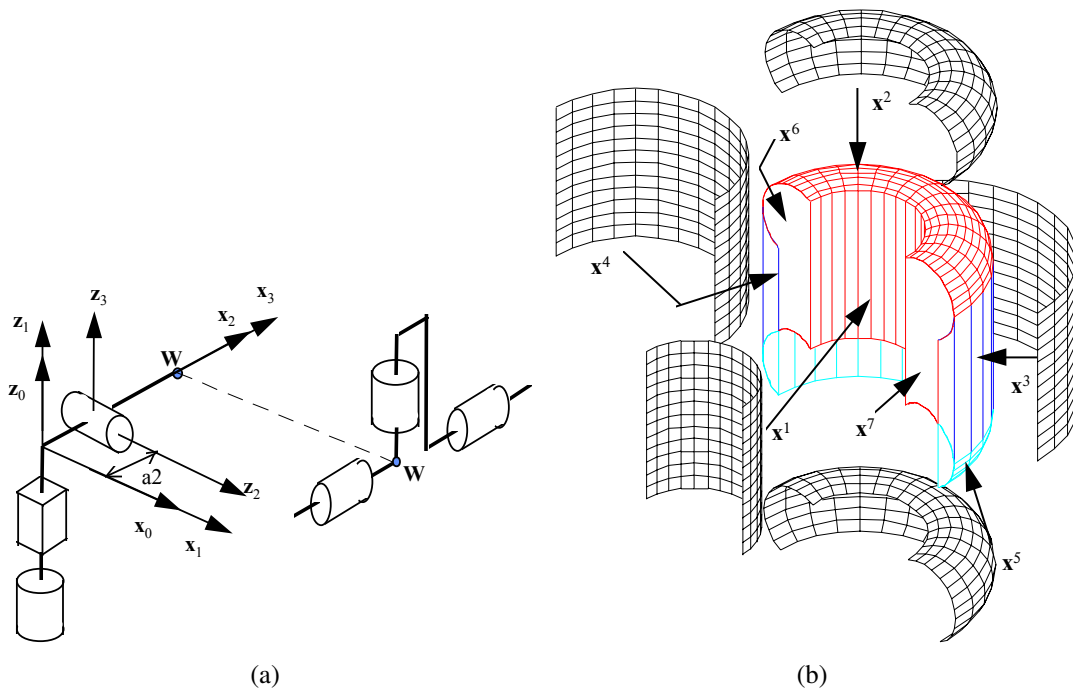


FIGURE 3.2 – (a) Schéma cinématique d'un mécanisme *RPR* couplé à un poignet sphérique *RRR*. (b) Représentation des surfaces de singularité assemblées pour former les frontières de l'espace de travail [Issu de Abdel-Malek, 1995].

Une méthode algébrique exploite ensuite des manipulations algébriques pour obtenir une expression explicite des configurations singulières définissant le bord d'espace dans l'espace des coordonnées opérationnelles en fonction des coordonnées actives, et réciproquement.

Ces manipulations sont relativement aisées pour les mécanismes sériels. La méthode est par exemple utilisée pour déterminer les frontières de l'espace de travail d'un mécanisme

RPRRRR [Abdel-Malek, 1995] décomposé en un mécanisme *RPR* et un poignet *RRR* décrit en figure 3.2(a). Une surface est établie pour une condition de singularité associée à chaque axe, comme montré en figure 3.2(b). Finalement les frontières de l'espace de travail sont représentées dans l'espace cartésien des coordonnées opérationnelles à l'aide d'un logiciel de calcul symbolique.

On peut par ailleurs citer les travaux de Jha *et al.* [2015] qui ont proposé une méthode de détermination de l'espace de travail reposant sur des manipulations algébriques des équations de contrainte suivies de la résolution de systèmes algébriques par un algorithme « Cylindrical Algebraic Decomposition ».

Avantages Cette méthode est applicable à tout mécanisme pour lequel une expression explicite du modèle géométrique direct ou inverse est disponible. Le débattement limité des liaisons peut être pris en compte. L'ensemble de solutions obtenu est continu selon une ou plusieurs dimensions. Les expressions algébriques permettent une évaluation très rapide de certaines caractéristiques du mécanisme.

Inconvénients Le succès de la méthode dépend fortement du choix de représentation du mécanisme, par exemple de coordonnées opérationnelles adaptées aux symétries de son espace de travail. La manipulation formelle des expressions algébriques est par ailleurs rapidement complexe. La méthode requiert donc une expertise importante. La manipulation manuelle des équations ne garantit pas une détermination complète des frontières de l'espace de travail.

3.1.2 Méthodes numériques

3.1.2.1 Méthode par discrétisation

La méthode repose généralement sur la disponibilité d'une expression du modèle géométrique direct pour les mécanismes sériels et du modèle géométrique inverse pour les mécanismes parallèles. La méthode par discrétisation est alors très simple dans son principe. Par exemple pour un mécanisme parallèle, l'espace des coordonnées opérationnelles est représenté par un ensemble discrétisé de points. Les coordonnées actives correspondant à chaque point sont calculées au moyen du modèle géométrique inverse pour déterminer si ce point satisfait aux contraintes introduites par la mécanique. Les méthodes se distinguent par :

- le choix des coordonnées qui sont discrétisées : discrétisation dans l'espace des coordonnées opérationnelles [Monsarrat et Gosselin, 2003] ou dans l'espace des co-

- ordonnées actives [Rastegar et Perel, 1990] respectivement si le modèle géométrique inverse ou direct est exploité.
- le schéma de discrétisation utilisé : grille de points régulière ou ensemble de points générés aléatoirement dans une représentation cartésienne [Rastegar et Perel, 1990], cylindrique [Bonev et Gosselin, 2006] ou sphérique [Monsarrat et Gosselin, 2003] de la pose de l'effecteur,

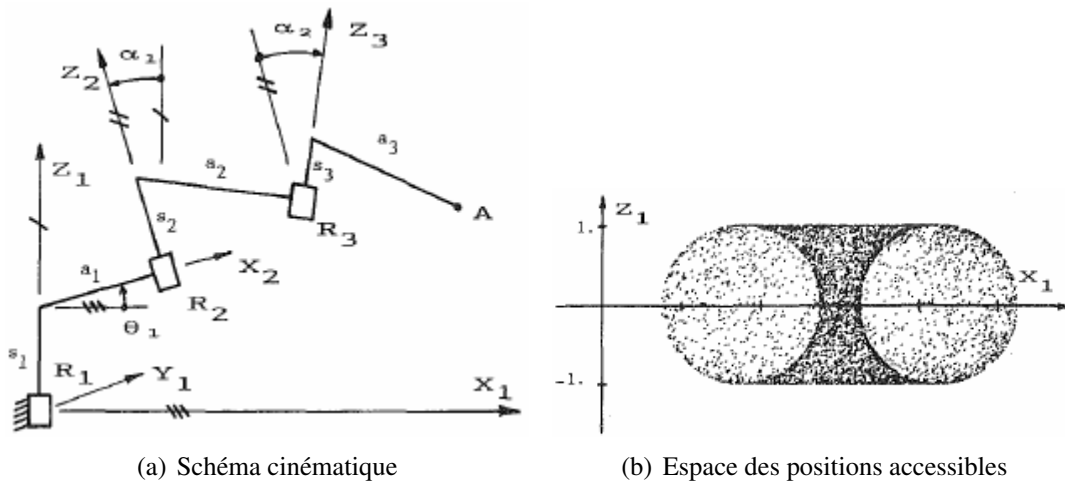


FIGURE 3.3 – (a) Schéma cinématique du mécanisme spatial *RRR*. (b) Projection exploitant la symétrie d'axe Z_1 de l'espace des positions accessibles dans le plan (X_1, Z_1) [Issues de Rastegar et Perel, 1990].

Rastegar et Perel [1990] utilisent une méthode par discrétisation où une grille de noeuds est générée selon une loi normale uniforme. L'exemple du mécanisme *RRR* spatial présenté en figure 3.3(a) est traité. L'image de 150000 points de l'espace articulaire est calculée au moyen du modèle géométrique direct. L'espace de travail est représenté par un nuage de points dans l'espace des coordonnées opérationnelles, comme illustré en figure 3.3(b). D'autres exemples représentatifs des méthodes par discrétisation sont présentés dans Fichter [1986], Masory et Wang [1994], Bonev et Ryu [2001] et Monsarrat et Gosselin [2003].

Avantages La méthode est simple d'implémentation et peut utiliser soit un modèle géométrique direct ou soit un modèle géométrique inverse. Une résolution numérique du modèle géométrique implicite est également envisageable. La méthode présente par ailleurs l'avantage d'être facilement généralisable à tout mécanisme et de rendre envisageable la prise en compte de toutes les contraintes liées à l'espace de travail, par exemple des débattements limités au niveau des liaisons.

Inconvénients Les solutions sont obtenues sous forme discrète. Supposons qu'une discontinuité apparaisse dans le nuage de points obtenu dans l'espace des coordonnées opérationnelles. Cette discontinuité peut être due à l'inexistence d'antécédent via le modèle géométrique direct ou alors un antécédent existe mais il est absent dans la discrétisation de l'espace articulaire choisie. Or deux points proches dans l'espace articulaire peuvent avoir des images distantes par le modèle géométrique direct. Un nombre de points élevé est nécessaire pour lever l'ambiguïté. Le nombre de points est par ailleurs d'autant plus important que la dimension de l'espace de travail et la précision exigée sont élevées. Le temps de calcul croît exponentiellement avec la densité de la grille [Merlet et Gosselin, 2008].

3.1.2.2 Méthode par optimisation

Snyman *et al.* [1998] ont proposé une méthode d'optimisation sous contrainte pour la détermination des frontières de l'espace de travail. L'expression d'un modèle géométrique est initialement nécessaire. Pour les mécanismes sériels, la contrainte est exprimée par le MGD et pour les mécanismes parallèles par le MGI.

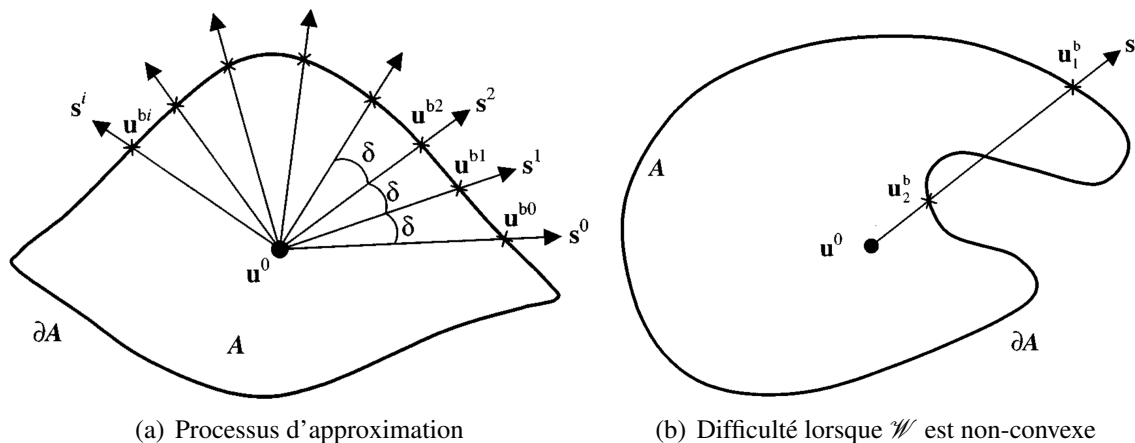


FIGURE 3.4 – (a) Processus d'approximation des frontières de l'espace de travail par résolution d'un problème d'optimisation numérique [Issu de Snyman *et al.*, 1998]. Pour chaque direction s^0 , la distance entre les point u^0 et u^{b0} est maximisée sous contrainte d'assemblage du mécanisme. (b) Des difficultés apparaissent lorsque l'espace de travail est non-convexe.

Dans le cas des mécanismes parallèles, on suppose connus un jeu de coordonnées opérationnelles admissible u^0 à l'intérieur d'une frontière et une direction initiale s^0 , voir figure 3.4(a). On recherche un jeu de coordonnées opérationnelles u sur la frontière dans la direction s^0 . La fonction de coût à maximiser est la distance entre la coordonnée opérationnelle recherchée u et le point initial u^0 . Les contraintes sont composées d'inégalités exprimant le débattement limité des liaisons, où la configuration des liaisons est calculée

avec le MGI. Une contrainte supplémentaire exprime que \mathbf{u} doit se trouver sur la droite paramétrée $\mathbf{u}(\lambda) = \mathbf{u}^0 + \lambda \mathbf{s}^0$, où λ est un paramètre réel. Le résultat de l'optimisation sous contrainte retourne un point \mathbf{u}^{b0} situé sur la frontière de l'espace de travail. Le processus est répété en modifiant la direction \mathbf{s} d'un angle δ , comme décrit en figure 3.4(a).

Avantages La méthode est applicable aux mécanismes sériels et aux mécanismes parallèles pour lesquels un MGI est disponible.

Inconvénients Une représentation discrète est obtenue. Chaque point solution résulte d'une optimisation numérique sous contrainte, ce qui en fait une méthode coûteuse. Cette méthode est initialement limitée aux mécanismes plan, même si elle peut être étendue aux dimensions supérieures en appliquant une discrétisation supplémentaire. Des difficultés apparaissent lorsque l'espace de travail est non-convexe, voir figure 3.4(b).

3.1.3 Méthode numérique d'analyse par intervalles

Les méthodes d'analyse par intervalles sont des méthodes intéressantes par leur généralité qui en fait des méthodes adaptées dans le cadre du développement de notre méthode.

Il s'agit d'une famille de méthodes pouvant être exploitées pour la résolution de systèmes d'équations non-linéaires et généralisant le principe de la méthode de dichotomie aux dimensions supérieures ou égales à 2. En analyse par intervalles, la résolution d'un système débute par la création d'une boîte encadrant les solutions dans l'espace des solutions. Une liste de boîtes contenant les solutions du système et éventuellement une seconde liste de boîtes ne contenant pas de solution sont déterminées de manière itérative à partir de cette boîte, comme illustré en figure 3.5. Deux approches principales peuvent être utilisées, voire combinées [Moore, 1966 - Chapitre 7] :

- la contraction d'intervalles consistant à réduire les bornes d'un ou plusieurs intervalles de la boîte encadrant la solution,
- le processus d'élimination consistant à déterminer des régions de la boîte ne contenant aucune solution, par exemple par bisection et application d'une fonction test.

Plusieurs approches d'estimation de l'espace de travail exploitent l'analyse par intervalles [Merlet, 1999 ; Oetomo *et al.*, 2008 ; Bohigas *et al.*, 2012a]. Deux d'entre elles nous intéressent particulièrement car elles présentent des caractéristiques intéressantes au regard de nos besoins.

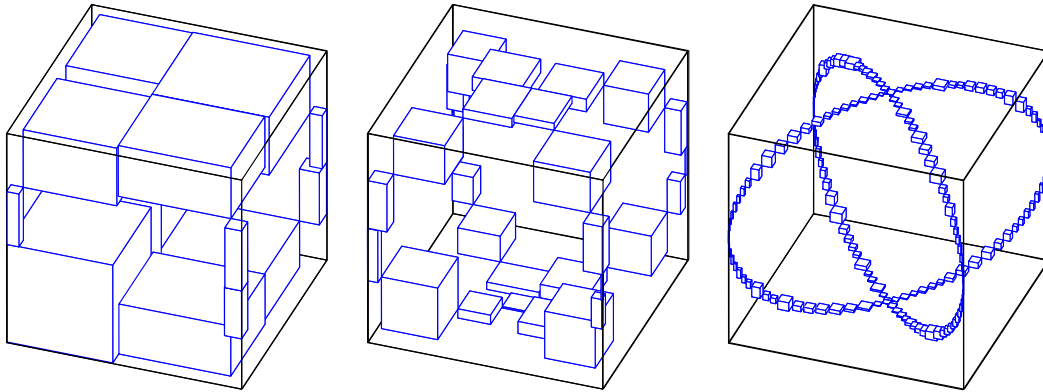


FIGURE 3.5 – Illustration du processus de résolution en analyse par intervalles [Issu de Bohigas *et al.*, 2014]. La boîte initiale en noir est décomposée par l'application itérative (de gauche à droite) d'un processus de contraction de boîte et d'élimination par bisection. Un ensemble de boîtes approximant les solutions est obtenu.

3.1.3.1 La méthode branch-and-bound

La méthode de « *branch-and-bound* » a été utilisée pour l'étude de l'espace de travail de dimension 6 du mécanisme 6 – *UPS* d'une plateforme de Stewart-Gough [Merlet, 1999]. L'approche est basée sur une exploration complète de l'espace des 6 coordonnées opérationnelles. Considérons une boîte de cet espace représentée par 6 intervalles. Le MGI permet de calculer une boîte correspondante de l'espace des coordonnées actives. La méthode branch-and-bound exploite alors un processus d'élimination : la validité de la boîte est testée au regard de la course admissible par les actionneurs et une bisection est effectuée en cas d'indétermination. Ce processus peut être décrit en considérant $f(x = \{x_1, x_2, \dots, x_n\})$ une fonction de n variables. Les variables manipulées dans l'algèbre d'intervalles sont des intervalles $X_i = [\underline{x}_i, \bar{x}_i]$ où \underline{x}_i désigne la borne inférieure et \bar{x}_i la borne supérieure de l'intervalle X_i . On appelle *boîte* un jeu de n intervalles noté $\mathbf{B} = \{X_1, X_2, \dots, X_n\}$. L'évaluation d'intervalles $F(\mathbf{B})$ de f pour la boîte \mathbf{B} est définie comme l'intervalle :

$$[\underline{F}, \bar{F}] \mid \underline{F} \leq f(x_1, \dots, x_i, \dots, x_n) \leq \bar{F}, \forall x_i \in X_i, \text{ pour tout } i. \quad (3.1)$$

L'évaluation d'intervalles d'une fonction vectorielle $\mathbf{f} = \{f_1, f_2, \dots, f_m\}$ correspond à la boîte $\mathbf{F}(\mathbf{B}) = \{F_1(\mathbf{B}), F_2(\mathbf{B}), \dots, F_m(\mathbf{B})\}$. Pour un mécanisme, les contraintes imposées par les liaisons sur la pose de l'effecteur sont exprimées par un système d'inéquations :

$$\mathbf{F}(\mathbf{B}) \leq \mathbf{0}. \quad (3.2)$$

L'espace de travail est alors défini comme l'ensemble des boîtes \mathbf{B} satisfaisant le système (3.2).

Dans la méthode proposée, une boîte initiale définissant le domaine de recherche des solutions est fournie par l'utilisateur. La présence de solutions dans la boîte est ensuite testée à l'aide d'un opérateur d'élimination noté S (pour « shrinking »). L'opérateur S calcule l'évaluation d'intervalle de chacune des composantes de \mathbf{F} pour la boîte courante \mathbf{B} et retourne :

- -1 si la borne inférieure de l'évaluation d'intervalle d'une des composantes $F_j(\mathbf{B})$ de $\mathbf{F}(\mathbf{B})$ est positive : la boîte ne contient pas de solution ;
- 1 si les bornes supérieures des évaluations d'intervalle de toutes les composantes $F_j(\mathbf{B})$ de $\mathbf{F}(\mathbf{B})$ sont négatives : l'ensemble représenté par la boîte est solution ;
- 0 sinon : la boîte est indéterminée.

Les boîtes déterminées sont archivées. Si une boîte est indéterminée, une dichotomie selon un des intervalles la composant est effectuée. Chacune des deux boîtes obtenues est testée à l'aide de l'opérateur S . L'algorithme se termine lorsque toutes les boîtes ont été déterminées ou lorsque toutes les boîtes en cours de traitement ont une taille inférieure à la précision voulue σ .

Avantages La méthode d'analyse par intervalles a l'avantage d'autoriser la prise en compte de tout type de contrainte. La méthode est applicable à tout mécanisme. Les erreurs numériques d'arrondi ou de troncature peuvent être prises en compte [Merlet, 1999] et l'ensemble de solutions peut ainsi être garanti sur le domaine décrit par la boîte initiale.

Inconvénients La méthode est générique au prix d'une efficacité limitée. L'évaluation par intervalles est sensible à la forme analytique utilisée pour définir le problème. Des temps de calcul acceptables sont obtenus essentiellement grâce à la définition d'un opérateur S efficace. Ceci implique une formulation spécifique du problème et une bonne connaissance de la méthode. Par exemple dans l'implémentation de Merlet [1999], c'est la possibilité de formuler des inéquations qui permet de déterminer qu'une boîte dans l'espace de travail est entièrement solution du système ($S = 1$) et qui rend l'algorithme efficace. Cependant une boîte comprenant des points situés sur les frontières de l'espace de travail est indéterminée. Une telle boîte est soumise au processus de bisection jusqu'à obtention de la résolution voulue, la complexité de l'algorithme est alors exponentielle.

3.1.3.2 La méthode branch-and-prune

Bohigas *et al.* [2012a] proposent l'exploitation de l'analyse par intervalles pour la détermination des frontières de l'espace de travail telle qu'exprimée par Haug *et al.* [1996]. Ils résolvent le système étendu des singularités inverses (1.17) à l'aide d'une méthode dite de « *branch-and-prune* » [Van Hentenryck *et al.*, 1997 ; Castellet et Thomas, 1998 ; Porta *et al.*, 2009] qui combine un processus d'élimination et de contraction d'intervalles.

Dans cette méthode un opérateur de contraction S permet « l'élagage » (« prune ») des bords d'une boîte par l'exploitation de la décomposition du système initial $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ en systèmes d'équations linéaires $\mathbf{L}(\mathbf{x}) = \mathbf{0}$, paraboliques $\mathbf{P}(\mathbf{x}) = \mathbf{0}$ ou hyperboliques $\mathbf{H}(\mathbf{x}) = \mathbf{0}$. Le processus de contraction alors effectué sur chaque élément est illustré en figure 3.6.

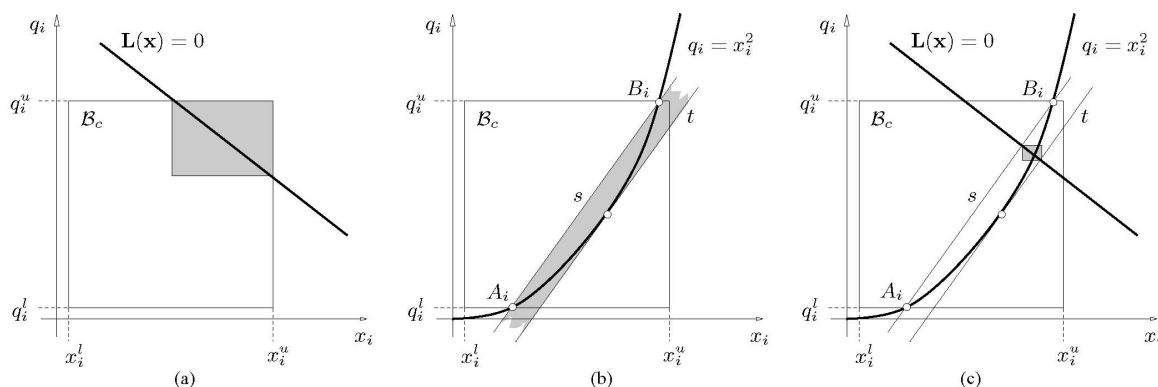


FIGURE 3.6 – Principe de l'opérateur de contraction utilisant les propriétés (a) des droites et (b) des paraboles. (c) Plus petite boîte contenant l'intersection de $\mathbf{L}(x) = \mathbf{0}$ et des demi-plans de (b) Porta *et al.* [Issu de 2009].

Le système doit être réduit à une forme quadratique. Ce processus peut être fastidieux si le nombre d'équations de contraintes et d'inconnues est important. La forme analytique obtenue peut par ailleurs ne pas être appropriée à une résolution efficace par l'analyse par intervalles.

Pour ces raisons, les auteurs ont proposé dans le cadre de la méthode branch-and-prune une approche de modélisation dédiée. Si elle est applicable à tout mécanisme, elle requiert l'adoption d'une formulation « bas niveau » adoptant une approche du domaine de la dynamique des systèmes multi-corps, mais peu commune en robotique [Jalón et Bayo 1994 ; Porta *et al.* 2009]. Un effort spécifique doit donc être réalisé pour exploiter la démarche.

La méthode branch-and-prune est implémentée dans le logiciel CUIK++ (pour « Complete Universal Inverse Kinematics ») [Porta *et al.*, 2014] d'analyse cinématique des mécanismes. Ce logiciel sera considéré plus tard pour s'y confronter en terme d'efficacité.

Avantages Le processus de contraction permet une réduction des boîtes sans avoir un recours systématique à la bisection. Ceci rend la méthode branch-and-prune plus efficace que la méthode branch-and-bound.

Inconvénients L'apport du processus de contraction est conditionné à une formulation spécifique des équations qui est obtenue par l'emploi d'une méthode de modélisation peu commune. Par ailleurs cette formulation résulte en de larges systèmes plus coûteux à résoudre que les systèmes obtenus par fermeture de chaîne. Malgré les efforts de parallélisation réalisés par les auteurs, les temps de calcul demeurent très élevés sur les exemples développés. Ainsi les auteurs exploitent finalement souvent une discrétisation de l'ensemble de solutions afin de déterminer l'espace de travail dans des temps raisonnables.

3.1.4 Méthodes de continuation d'ordre 1

Les méthodes de continuation d'ordre 1 sont également génériques et sont de plus efficaces, deux propriétés d'intérêt dans le cadre du développement de nos méthodes.

Litvin [1980] utilise le théorème des fonctions implicites pour exprimer le fait que les frontières de l'espace de travail correspondent à des configurations pour lesquelles l'effecteur perd un degré de liberté. Il en déduit le critère d'annulation du déterminant de la jacobienne $\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$, avec \mathbf{u} les coordonnées opérationnelles et \mathbf{z} les coordonnées actives et passives, comme condition pour qu'une configuration soit sur les frontières de l'espace de travail. Haug *et al.* [1992] formalisent cette condition à partir du modèle géométrique implicite et de ses dérivées :

$$\partial\mathcal{W} = \left\{ \mathbf{u} \in \mathcal{W} \mid \exists \mathbf{z} \text{ tel que : } \text{rang} \left(\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \right) < n_e \right\}, \quad (3.3)$$

où $\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) = \frac{\partial \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{z}}$ est une matrice $n_e \times n_z$ avec n_e la dimension de $\mathbf{R}^{\mathcal{C}}$ et n_z la dimension de \mathbf{z} .

Jo et Haug [1989b] et Haug *et al.* [1992] ont formulé le critère analytique suivant pour les frontières de l'espace de travail d'un mécanisme quelconque :

$$\partial\mathcal{W} = \left\{ \mathbf{u} \in \mathcal{W} \mid \exists (\mathbf{z}, \xi) \text{ tel que : } \left[\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \right]^T \cdot \xi = \mathbf{0} \text{ et } \xi^T \cdot \xi = 1 \right\}, \quad (3.4)$$

où ξ est un vecteur unitaire inconnu de dimension n_e orthogonal aux vecteurs-colonne de $\mathbf{R}_z^{\mathcal{C}}$. Finalement les frontières de l'espace de travail $\partial\mathcal{W}$ sont les projections sur l'espace des coordonnées opérationnelles des solutions \mathcal{S}_z du système étendu $\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \mathbf{0}$ [Haug *et al.*, 1992].

Haug *et al.* [1996] proposent de résoudre le système étendu $\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \mathbf{0}$ (1.17), vu comme un système d'équations non-linéaires sous-déterminé (2.4) de la forme générique $\mathcal{R}(\mathbf{U}) = \mathbf{0}$, par une méthode de continuation d'ordre 1, telle que présentée en section 2.1.4. A partir d'un point solution initial, toutes les branches de solutions connectées constituant \mathcal{S}_z sont déterminées grâce à la gestion des points de bifurcation. Les branches de solutions définies dans l'espace des coordonnées \mathbf{U} sont projetées dans un diagramme de bifurcation en (x, y) , comme montré en figure 3.7.

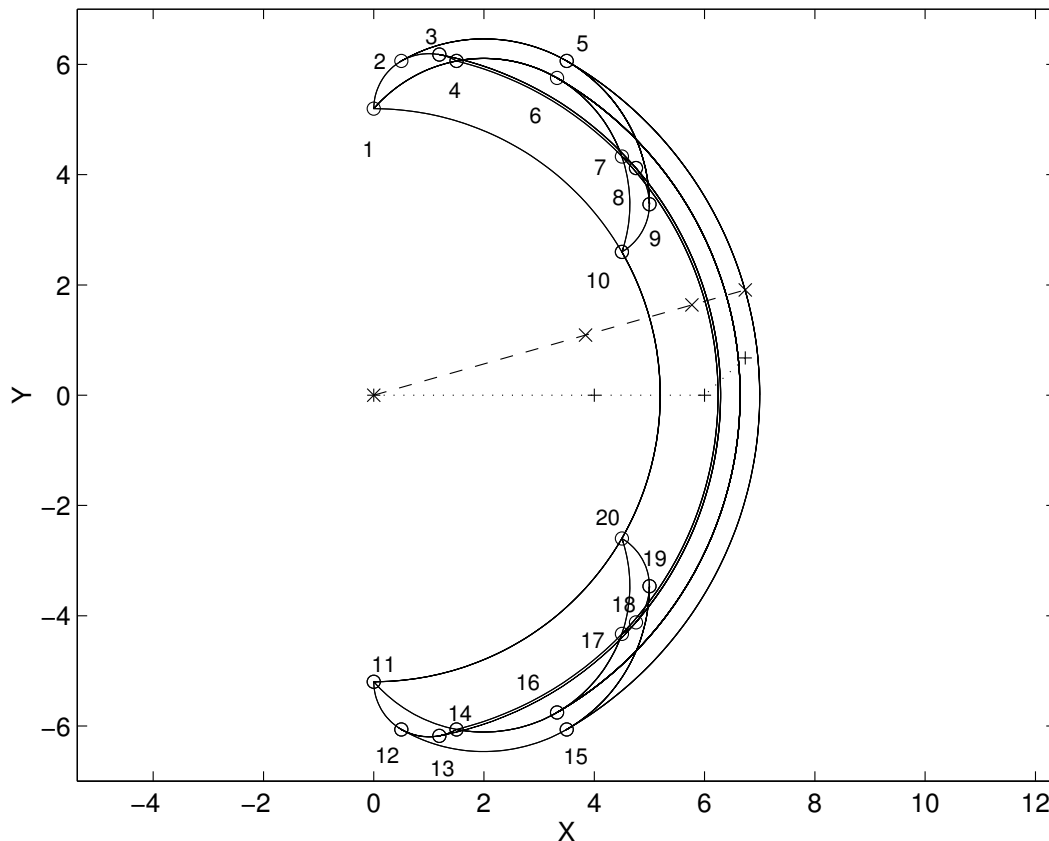


FIGURE 3.7 – Frontières de l'espace de travail $\partial\mathcal{W}$ du mécanisme *RRR*. Les traits en pointillés et en tirets représentent respectivement les configurations initiale et finale du mécanisme pour le processus de continuation [Issu de Haug *et al.*, 1996 - Figure 3].

Avantages Cette approche est efficace et peut être appliquée à tout mécanisme. Elle exploite le modèle géométrique implicite du mécanisme, toujours disponible. Elle permet la prise en compte de limites de débattement des liaisons. Yang *et al.* [2010] ont d'ailleurs proposé une extension de la méthode pour prendre en compte également des collisions avec des obstacles.

Inconvénients La formulation du système étendu incombe à l'utilisateur, ce qui peut rendre moins abordable la méthode. Plusieurs difficultés se posent par ailleurs pour la détermination de l'espace de travail de tout mécanisme. D'abord une gestion des points de bifurcation possédant plus de 2 tangentes est nécessaire. Ensuite cette méthode ne permet pas d'aborder de front le problème des frontières d'un espace de travail de dimension supérieure à 2, une discrétisation étant réalisée si nécessaire. De manière plus essentielle, la méthode de continuation d'ordre 1 utilisée ne génère des résultats que sous forme discrète, avec les inconvénients inhérents décrits dans le cas des méthodes par discrétisation.

3.1.5 Méthode de continuation de haut-degré

Le logiciel Diamanlab a été récemment exploité pour le parcours de l'espace des configurations du mécanisme et donc la détermination des configurations admissibles d'un mécanisme parallèle 3 – US [Rubbert *et al.*, 2017]. La gestion des bifurcations a alors permis d'obtenir différentes branches de solutions correspondant aux multiples solutions des problèmes géométriques direct et inverse. Boehler *et al.* [2015] ont par ailleurs exploité Diamanlab pour la détermination des frontières de l'espace de travail sous charge d'un mécanisme de tenségrité. Un tel mécanisme est composé de membres rigides en compression reliés par des ressorts en tension. La géométrie du mécanisme est dépendante des efforts internes qui s'exercent entre les différents éléments qui le composent. L'espace de travail sous charge correspond à l'ensemble des configurations admissibles respectant les efforts internes admissibles. Pour le mécanisme étudié, les conditions algébriques de singularité cinématique ont été établies et ajoutées aux équations de contraintes géométriques et statiques afin d'obtenir un système représentant les frontières de l'espace de travail sous charge. Les frontières de l'espace de travail sous charge ont alors été déterminées avec Diamanlab.

Ces travaux ont montré l'intérêt de l'approche, mais aussi le besoin de développements spécifiques pour rendre l'outil aisé à prendre en main et pour permettre une étude systématique des frontières de l'espace de travail d'un mécanisme.

3.2 Continuation de haut-degré

Dans ce chapitre, nous proposons de pallier les limites identifiées en nous appuyant sur les outils numériques décrits au chapitre 2 pour développer une méthode efficace et automatisée pour la détermination de l'espace de travail de tout mécanisme. Notre méthode combine la continuation de haut-degré présentée en section 2.2.2 et la DA présentée en section 2.3, toutes deux implémentées dans l'outil Diamanlab décrit en section 2.4.

Notre méthode repose sur une modélisation initiale du mécanisme pour obtenir un modèle géométrique implicite. La section 3.2.1 présente le processus automatisé de formulation du système d'équations étendu décrivant les frontières de l'espace de travail $\partial\mathcal{W}$ à partir de ce modèle. Nous décrivons ensuite en section 3.2.2 les modifications de Diamanlab proposées pour disposer d'un outil automatisé de détermination des branches de solutions connectées.

3.2.1 Formulation automatique des frontières de l'espace de travail

3.2.1.1 Système étendu par ses dérivées

Nous exploitons comme Haug *et al.* [1996], la représentation de l'espace de travail par son enveloppe, en partant de la définition des frontières de l'espace de travail :

Définition 11 : Les frontières de l'espace de travail $\partial\mathcal{W}$ sont définies comme l'ensemble :

$$\partial\mathcal{W} = \left\{ \mathbf{u} \mid \exists (\mathbf{z}, \xi) \text{ tel que } \mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \mathbf{0} \right\}, \quad (3.5)$$

où ξ est un vecteur unitaire de dimension n_e et $\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \mathbf{0}$ est le système étendu :

$$\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}_z^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (3.6)$$

Le système étendu (3.6) possède $n_e + n_z + 1$ équations comportant $n_u + n_z + n_e$ inconnues alors que le système $\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) = \mathbf{0}$ (1.4) possède n_e équations en n_q inconnues. Pour un mécanisme non-redondant, l'ensemble des configurations non-singulières solutions de ce système est de dimension $n_u = n_q - n_e$ [Haug *et al.*, 1996]. L'ensemble $\partial\mathcal{W}$ est ainsi de dimension $n_u - 1$, donc inférieur d'une dimension à l'espace de travail \mathcal{W} défini par l'équation (1.6) comme l'ensemble des coordonnées \mathbf{u} satisfaisant le système (1.4).

3.2.1.2 Traitement des dérivées mixtes

Pour sa résolution, le système étendu (3.6) est considéré comme un système générique de la forme $\mathcal{R}(\mathbf{U}(a)) = \mathbf{0}$ introduite au chapitre 2 avec l'équation (2.29). Dans Diamant, la fonction $\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}(a), \mathbf{z}(a), \xi(a))$ est différenciée à haut-degré par rapport à a . Ceci implique une différenciation de $\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$ et de $(\mathbf{R}_z^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi$ par rapport à $\mathbf{U} = [\mathbf{u}^{\top}, \mathbf{z}^{\top}, \xi^{\top}]^{\top}$. Or la matrice jacobienne $\mathbf{R}_z^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$ contient des dérivées de la fonction $\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$ par rapport à \mathbf{z} . On voit donc apparaître des dérivées mixtes $\frac{\partial^2 \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{z} \partial \mathbf{u}}$ et $\frac{\partial^2 \mathbf{R}^{\mathcal{C}}}{\partial \mathbf{z} \partial \xi}$.

Haug *et al.* [1996] proposaient l'utilisation de ADOL-C [Griewank *et al.*, 1996], un outil de DA de haut-degré par surcharge d'opérateur, pour l'évaluation de la jacobienne $\mathbf{R}_z^{\mathcal{L}}(\mathbf{u}, \mathbf{z}, \xi)$ intervenant dans la définition des frontières de l'espace de travail. Or la continuation requiert également le calcul de matrices hessiennes. Elles peuvent être évaluées en utilisant une fois de plus ADOL-C dans une direction de différentiation adaptée, mais ce processus est complexe à mettre en oeuvre.

Les outils de différentiation par transformation de source sont plus intuitifs pour le calcul de jacobienes et l'établissement du système étendu. En effet ils permettent d'obtenir le code différentié qui peut être exploité pour l'évaluation de $\mathbf{R}_z^{\mathcal{L}}(\mathbf{u}, \mathbf{v})$ tout en reproduisant les instructions de $\mathbf{R}^{\mathcal{L}}(\mathbf{u}, \mathbf{z})$. Nous choisissons l'outil de DA Tapenade [Hascoet et Pascual, 2013] présenté en section 2.3.2.1 car son code est lisible contrairement aux autres outils de DA par surcharge d'opérateur disponibles en Matlab. De plus, son mode de DA « multi-directionnel tangent » permet une évaluation simultanée des dérivées composant la jacobienne. Il ne traite que des codes Fortran. La traduction du langage Matlab au langage Fortran ne pose cependant pas de difficulté dans notre application. En effet le modèle géométrique implicite du mécanisme peut être implémenté sans opération ou fonction spécifique au langage Matlab. Le nombre de lignes de code est par ailleurs réduit, même pour un mécanisme parallèle complexe. La syntaxe en langage Fortran 95 est par ailleurs relativement proche de celle du langage Matlab.

Comme exposé au chapitre 2, la différentiation à haut-degré est efficace avec la DA par surcharge d'opérateur. Nous choisissons donc d'exploiter dans ce but la classe `@Taylor` codée en Matlab et présentée en section 2.3.3.2. Nous proposons donc finalement pour les tâches de DA :

- d'exploiter la DA par surcharge d'opérateur pour la continuation de haut-degré,
- de calculer les dérivées mixtes en effectuant la dérivation d'ordre 1 avec Tapenade puis en surchargeant le code obtenu à l'aide de la classe `@Taylor`.

3.2.1.3 Choix de construction du système étendu $\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi)$

Deux approches de couplage entre Matlab et Tapenade sont envisageables pour construire le code Matlab correspondant au système étendu (3.6) à partir du modèle géométrique implicite (1.4) :

- Établir le code Fortran d'un système augmenté différentié avec Tapenade pour obtenir un code qui contienne notamment les équations du système augmenté $\mathbf{R}^{\mathcal{S}_z} = \mathbf{0}$. L'extraction du système étendu peut ensuite être réalisée et le code Fortran converti en Matlab ;

$$\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \xrightarrow[\text{le système}]{\text{Augmenter}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{pmatrix} \xrightarrow[\text{par rapport à } \mathbf{z}]{\text{Différentier}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ \mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ (\mathbf{R}_{\mathbf{z}}^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{pmatrix} \xrightarrow[\text{système étendu}]{\text{Réduire en}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}_{\mathbf{z}}^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{pmatrix} \quad (3.7)$$

— Etablir le code Fortran du modèle géométrique, différentier ensuite avec Tapenade pour obtenir un code qui permette d'évaluer $\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$ et la jacobienne $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$. Le code Fortran obtenu peut alors être converti en code Matlab et intégré dans le code standard qui permet d'assembler le système étendu.

$$\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \xrightarrow[\text{par rapport à } \mathbf{z}]{\text{Différentier}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ \mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \end{pmatrix} \xrightarrow[\text{système étendu}]{\text{Assembler le}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}_{\mathbf{z}}^{\mathcal{C}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{pmatrix} \quad (3.8)$$

La première approche a été adoptée, car le code généré par Tapenade peut être directement intégré dans Diamanlab sans traitement spécifique. L'expertise nécessaire pour la détermination du problème est minime.

Le processus (3.7), proposé pour construire le système (3.6) est donc finalement composé de 3 étapes [Hentz *et al.*, 2016a] :

1. Définition des équations $\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z})$ et des coordonnées \mathbf{u} et \mathbf{z} du mécanisme étudié,
2. Implémentation du système augmenté dans une routine `RESIDUAL(u, z, xi, res, restxi, nxi)`. Les variables indépendantes sont u , z et xi , et les variables dépendantes sont res , $restxi$ et nxi , respectivement.
3. Différentiation de `RESIDUAL(u, z, xi, res, restxi, nxi)` par rapport à z avec Tapenade en mode « multi-directionnel tangent » pour générer la routine `RESIDUAL_DV(u, z, zd, xi, res, resd, restxi, restxid, nxi)` contenant le système étendu (3.6).

A l'évaluation du code, la variable dérivée zd prend pour valeur la matrice identité $n_z \times n_z$ qui décrit la base canonique de \mathbb{R}^{n_z} . La routine `RESIDUAL_DV` contient les informations nécessaires pour l'évaluation du système (3.6) sans manipulation supplémentaire. Ce code est traduit en Matlab puis implémenté dans un fichier utilisateur `R`. Il peut alors être résolu par Diamanlab sous forme de branches de solutions continues représentant les frontières de l'espace de travail. Cette approche est détaillée en annexe A et illustrée par les codes exploités pour la détermination de l'espace des positions accessibles du mécanisme RRR traitée en section 3.3.

3.2.2 Détermination des frontières continues de l'espace de travail

3.2.2.1 Implémentation initiale de Diamanlab

L'implémentation initiale de Diamanlab permet un parcours interactif de branches de solutions qui repose sur une détection automatique des points de bifurcation et un changement de branches de solutions assisté par l'utilisateur. Les points de bifurcation sont détectés par l'émergence de séries géométriques. Ils peuvent être localisés par détection d'un changement de signe du déterminant de la matrice $\{\mathcal{R}_1|_{\mathbf{U}_1=Id}\}$ (Voir paragraphe 2.1.1). Pour un point de bifurcation simple, le changement de branches est assisté par le calcul automatique d'un vecteur indiquant la direction de la seconde branche. L'utilisateur doit simplement cliquer sur le point de bifurcation pour lancer la continuation dans cette direction.

Si l'ensemble de solutions est constitué de plusieurs branches de solutions connectées par différents points de bifurcation, son parcours interactif tel que décrit en section 2.4.3 et nécessitant une intervention de l'utilisateur à chaque changement de branche de solutions peut être fastidieux. Une approche interactive du changement de branches n'est donc pas adaptée.

L'implémentation courante de Diamanlab ne permet par ailleurs pas la gestion des points d'intersection de branches avec plus de 2 tangentes distinctes, pourtant visibles dans l'exemple du mécanisme RRR, voir le point 1 en figure 3.7. Ces points correspondent à des points de bifurcation complexes pour lesquels la jacobienne analysée perd plus d'un rang. Dans le cas d'une perte de rang multiple, le point de bifurcation peut être localisé mais il est difficile de parcourir toutes les branches au moyen de la continuation interactive. Cela nécessite la détermination d'un point initial sur chacune des branches de solution à l'intersection.

3.2.2.2 Méthode de perturbation aléatoire

Dans l'implémentation proposée, les points de bifurcation sont détectés par détection du changement de signe du déterminant de la jacobienne $\{\mathcal{R}_1|_{\mathbf{U}_1=Id}\}$ [Keller, 1987]. Nous proposons de gérer le changement de branches de solutions par une méthode de perturbation aléatoire [Allgower et Georg, 1990]. La continuation à partir d'un point initial perturbé sans application de corrections de Newton-Raphson permet un changement de branche aléatoire [Seydel, 2010 - Section 2.10 - p.104], comme indiqué en figure 3.8.

Lorsque la continuation est initialisée en un point solution, la branche fondamentale en traits pointillés est parcourue. Le point de bifurcation \mathbf{U}_{bif} est détecté et traversé en suivant la branche fondamentale dans la direction \mathbf{U}_{tf} . On désire maintenant effectuer un changement de branches de solutions afin de parcourir la branche bifurquée en traits pointillés.

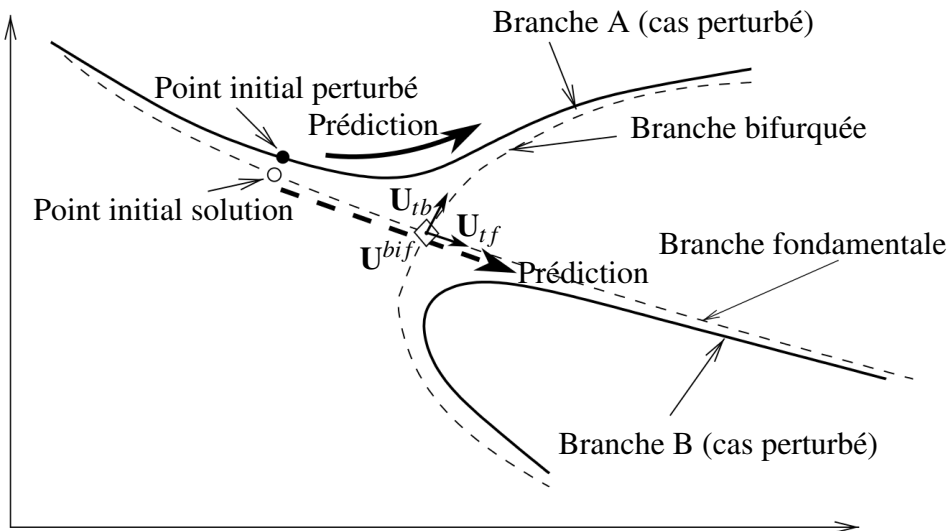


FIGURE 3.8 – Effet de la perturbation du point initial sur la connectivité des branches de solutions [Adapté de Seydel, 2010 - Figure 2.47].

L'implémentation initiale de Diamanlab calcule la direction U_{tb} et l'utilisateur lance une continuation dans cette direction. Ceci n'est possible que pour un point de bifurcation simple.

La méthode de perturbation aléatoire permet un changement de branche y compris dans le cas de bifurcations complexes. La continuation est initialisée en un point perturbé proche d'un point solution. Le point perturbé appartient à la branche A, solution d'un système perturbé pour lequel deux branches de solutions sont disjointes et le point de bifurcation n'est pas présent. Un changement de branche continu entre la branche fondamentale et la branche bifurquée est réalisé par le parcours de la branche A.

En appliquant une perturbation aléatoire au point initial solution, chacune des branches émanant d'un point de bifurcation rencontré pourra être aléatoirement parcourue. Finalement en passant par chaque point de bifurcation un nombre suffisant de fois, le parcours d'un ensemble de branches de solutions connectées est automatisé. L'algorithme 3 est utilisé pour le parcours automatisé d'un ensemble de branches de solutions connectées.

3.3 Application à un mécanisme plan

Nous choisissons un mécanisme précédemment étudié par Haug *et al.* [1996] afin de réaliser une comparaison précise des résultats obtenus par notre méthode. Le même mécanisme est aussi fourni comme exemple pour le calcul des frontières de l'espace de travail avec le logiciel CUIK++ [Porta *et al.*, 2014], permettant une comparaison quantitative de l'efficacité

Algorithm 3 – Algorithme de continuation automatisée par un changement de branche par perturbation de la solution.

Pour $i = 1$ à $nbSeries$ **faire**

Si $|\mathbf{R}(\mathbf{U}_0)| > NRthreshold$ **Alors**

 Corrections de Newton-Raphson \mathbf{U}_0

Fin Si

 Définition du vecteur de perturbation

$pert = ampliPert * rand([-1, 1], size(U0))$

 Perturbation de la solution $\mathbf{U}_{0p} = \mathbf{U}_0 + pert$

 Calcul de la série \mathbf{U} à partir du point solution \mathbf{U}_0 (Algorithme 1)

 Calcul de la série \mathbf{U}_p à partir du point perturbé \mathbf{U}_{0p}

 Création d'un objet checkpoint solution \mathbf{U}

 Création d'un objet checkpoint perturbé \mathbf{U}_p

 Stockage du checkpoint solution \mathbf{U}

$\mathbf{U}_0 = \mathbf{U}_p(a_m)$ et $\mathbf{U}_t = \mathbf{U}'_p(a_m)$

Fin Pour

réalisée en section 3.3.3. La résolution est effectuée sur un Intel Core i7, 2.7GHz, 16GB RAM. Les polynômes de Taylor sont évalués avec une précision $\varepsilon = 1e - 6$ et un ordre de troncature $K = 20$.

3.3.1 Mécanisme RRR avec liaisons à débattements limités

Nous considérons l'exemple du mécanisme sériel *RRR* plan de mobilité $m = 3$. L'espace de travail du mécanisme est défini dans l'espace des coordonnées opérationnelles (x, y, θ) . Le premier espace d'intérêt étudié couramment est l'espace des positions accessibles défini dans l'espace des coordonnées opérationnelles (x, y) [Haug *et al.*, 1996 ; Snyman *et al.*, 1998]. Le modèle géométrique implicite est donné en équation (1.1). Pour l'étude de l'espace des positions accessibles, nous extrayons les 2 premières relations qui correspondent au modèle étudié par Haug *et al.* [1996] :

$$\begin{cases} x = L_1 \cos(\alpha) + L_2 \cos(\alpha + \beta) + L_3 \cos(\alpha + \beta + \gamma), \\ y = L_1 \sin(\alpha) + L_2 \sin(\alpha + \beta) + L_3 \sin(\alpha + \beta + \gamma), \end{cases} \quad (3.9)$$

Nous considérons que les coordonnées actives du mécanisme ont des débattements limités, par exemple $\alpha \in [\alpha^{min}; \alpha^{max}]$. Pour prendre en compte cette contrainte, nous effectuons un changement de variables $\alpha = a_1 + b_1 \sin(v_1)$, avec $a_1 = \frac{\alpha^{min} + \alpha^{max}}{2}$ et $b_1 = \frac{\alpha^{max} - \alpha^{min}}{2}$ [Jo et Haug, 1989a]. Dans cet exemple nous considérons $\alpha^{min} = \beta^{min} = \gamma^{min} = -\frac{\pi}{2}$ et $\alpha^{max} = \beta^{max} = \gamma^{max} = \frac{\pi}{2}$ pour les débattements limités sur α , β et γ . Le changement de variable

est effectué dans le système (3.9) et les nouvelles coordonnées actives considérées sont $\mathbf{v} = [v_1, v_2, v_3]^T$.

3.3.2 Espace des positions accessibles 2D

L'espace des positions accessibles pour le mécanisme plan est de dimension 2. Ses frontières sont donc constituées d'un ensemble de branches de solutions qui peuvent être résolues directement par continuation de haut-degré. Les paramètres géométriques du mécanisme sont fixés à $(L_1, L_2, L_3) = (4, 2, 1)$. Le point initial \mathbf{U}^0 correspond au mécanisme dans une configuration étendue horizontale $(\mathbf{u}, \mathbf{v}) = (x, y, v_1, v_2, v_3) = (7, 0, 0, 0, 0)$ avec le vecteur $\xi = (1, 0)$.

Les branches de solutions $(\mathbf{u}(a), \mathbf{v}(a), \xi(a))$ sont calculées dans \mathcal{C} dont la représentation est difficile. Elles sont projetées sur l'espace des coordonnées (x, y) . Le diagramme de bifurcations correspondant permet de représenter les frontières de l'espace des positions accessibles, voir figure 3.9. Le diagramme de bifurcations est constitué de 20 branches de solutions s'intersectant en des points de bifurcations, ce qui correspond bien aux résultats obtenus par Haug *et al.* [1996]. L'espace des positions accessibles est bordé par des branches de solutions extérieures, alors que les branches de solutions situées à l'intérieur correspondent à des configurations pour lesquelles le mouvement de l'effecteur est restreint.

La projection des branches nécessaire à leur représentation entraîne nécessairement une perte d'information par rapport à l'information calculée dans l'espace des configuration \mathcal{C} . Par exemple deux branches qui se croisent dans (x, y) peuvent correspondre à une intersection de branches dans \mathcal{C} ou à une simple superposition due à la projection. La représentation par un diagramme de bifurcations permet de lever cette ambiguïté car on obtient les lieux de singularités mais en plus l'information de connectivité des branches représentée par les points de bifurcation.

Pour une position accessible et une orientation de l'effecteur donnée, les solutions du modèle géométrique implicite sont des jeux de coordonnées actives appelées solutions géométriques inverses. Des branches correspondant à différentes solutions géométriques inverses peuvent se superposer (branches 1-10-20 et 10-20-11 par exemple) ou s'intersecter en des points de bifurcations. Bien qu'il semble possible de changer de branche de manière continue, ce n'est pas toujours possible. Un changement de branche de manière continue implique une trajectoire admissible par le mécanisme et donc un changement de configuration pour des valeurs continues des coordonnées. Par exemple le parcours d'une trajectoire décrite par les points 1-10-9 impliquerait le changement instantané d'une posture « coude baissé » à une posture « coude levé » au point 10.

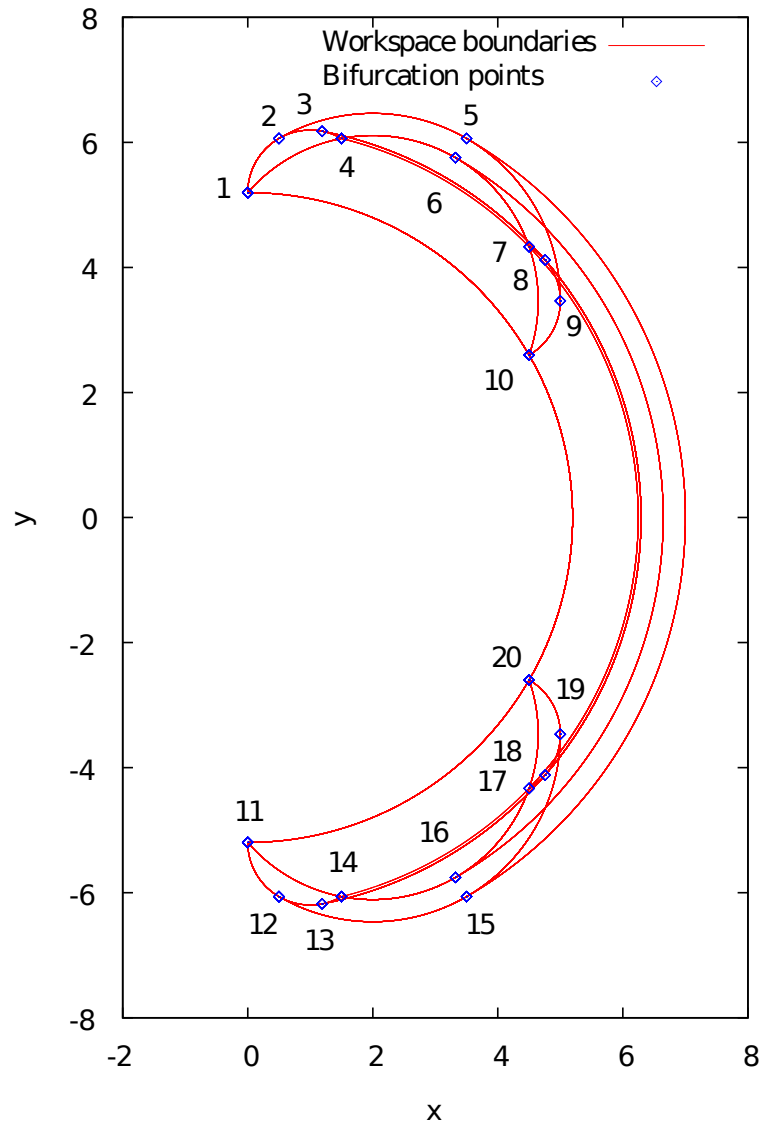


FIGURE 3.9 – Frontières de l'espace des positions accessibles du mécanisme RRR représentées dans un diagramme de bifurcations.

3.3.3 Evaluation de l'efficacité

La figure 3.10 et le tableau 3.1 présentent une comparaison de l'efficacité de la méthode Diamant et de la méthode branch-and-prune [Bohigas *et al.*, 2012a] implémentée dans le logiciel CUIK++ [Porta *et al.*, 2014]. Nous nous sommes appuyés sur le tutoriel disponible sur le site du laboratoire IRI [CSIC-UPC, 2017] pour l'installation et l'utilisation du logiciel.

Le logiciel CUIK++ étant codé en C, nous choisissons d'utiliser l'implémentation de la méthode Diamant en Fortran 90. En effet, le langage Matlab est un langage interprété dont les temps de calcul ne sont pas comparables au C. Premièrement la figure 3.10(b) présente les

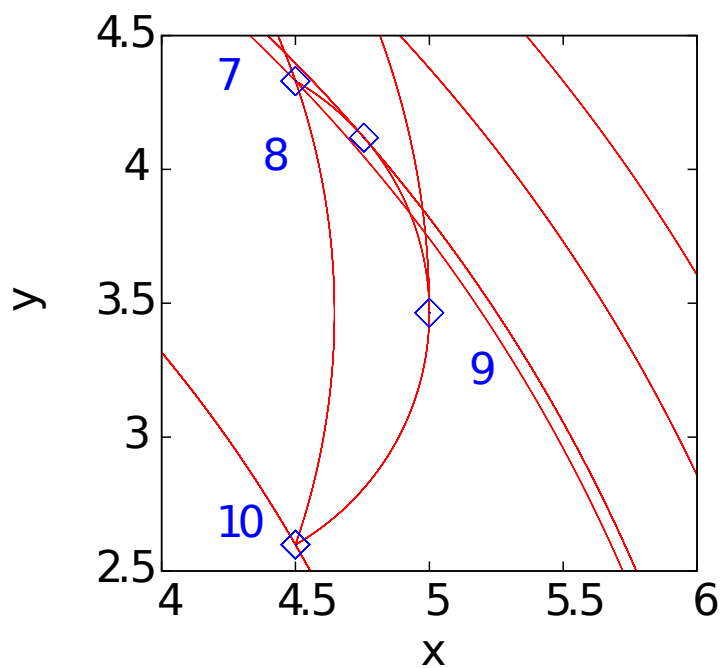
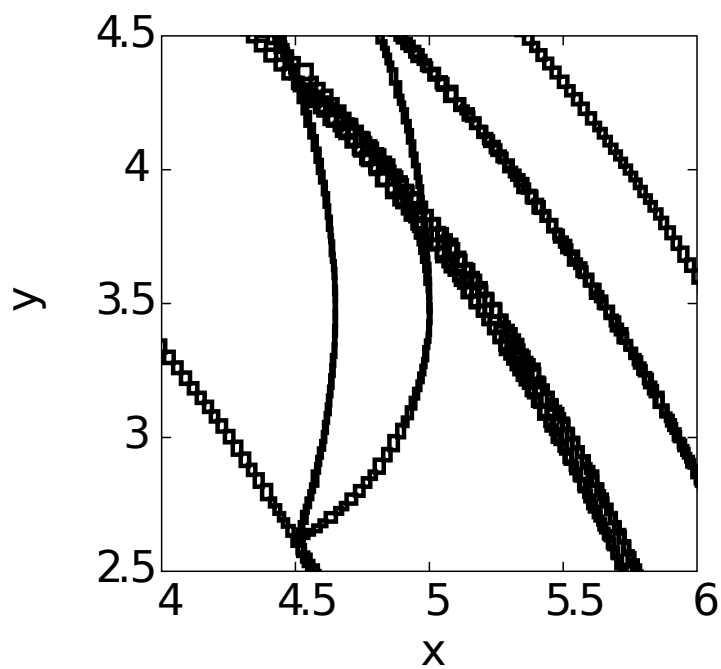
(a) Diamant $\varepsilon = 1e-6$ (b) Branch-and-prune $\sigma = 5e-2$

FIGURE 3.10 – (a) Agrandissement des branches de solutions de la figure 3.9. (b) Agrandissement sur la même zone des frontières déterminées avec la méthode branch-and-prune implémentée dans CUIK++ [Porta *et al.*, 2014].

frontières de l'espace des positions accessibles obtenues avec CUIK++ sous forme de boîtes dont la plus grande dimension ne dépasse pas le seuil de précision noté σ . Deuxièmement, le tableau 3.1 met en évidence le temps de calcul qui augmente de manière exponentielle avec la précision pour la méthode branch-and-prune. La méthode Diamant s'avère à la fois plus rapide et précise que la méthode branch-and-prune. Les points de bifurcation ne sont par ailleurs pas déterminés avec précision avec cette dernière.

Méthode	Temps de calcul	Précision
Branch-and-prune	1min15s	$\sigma = 5e - 2$
Branch-and-prune	4min31s	$\sigma = 1e - 2$
Branch-and-prune	7min36s	$\sigma = 5e - 3$
Branch-and-prune	196min42s	$\sigma = 1e - 4$
Diamant	7s	$\varepsilon = 1e - 6$

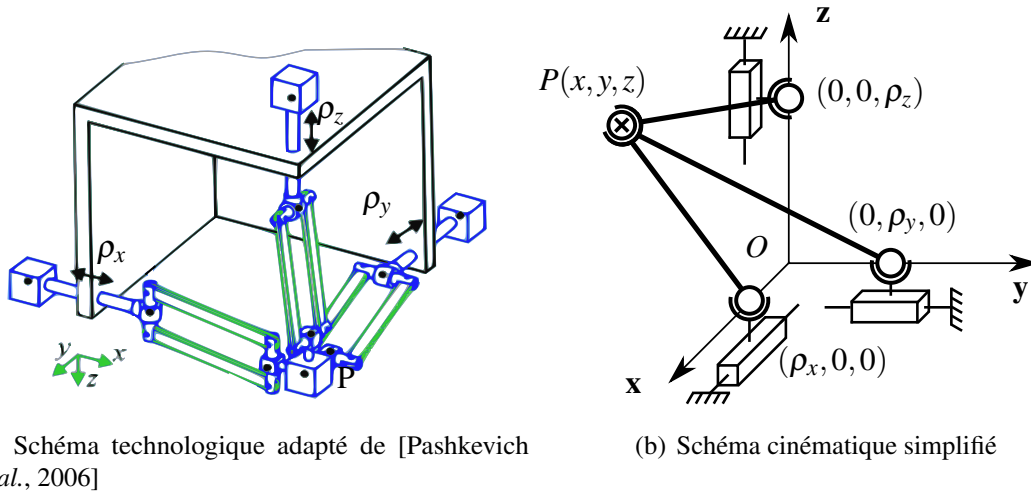
TABLEAU 3.1 – Temps de calcul des frontières de l'espace des positions accessibles du mécanisme RRR pour différentes précisions avec les méthodes Diamant et branch-and-prune.

3.4 Application à un mécanisme spatial

Nous évaluons maintenant notre outil en considérant l'espace de travail de dimension 3 du mécanisme Orthoglide.

3.4.1 Le mécanisme orthoglide

Le mécanisme parallèle Orthoglide [Pashkevich *et al.*, 2006] décrit en figure 3.11(a) est un mécanisme spatial. Il présente des caractéristiques cinématiques intéressantes, comme l'existence d'une configuration isotrope autour de laquelle le rapport de transmission entre les vitesses actives et opérationnelles est proche de l'unité. Dans une modélisation simplifiée exploitée ici et présentée en figure 3.11(b), il est composé de trois jambes identiques de longueur L . Chaque jambe est connectée d'un côté à l'effecteur P par une liaison sphérique et de l'autre à la base par une liaison prismatique actionnée via une liaison sphérique passive. Les coordonnées actives $\mathbf{v} = [\rho_x, \rho_y, \rho_z]^T$ représentent les positions des liaisons actives et les coordonnées opérationnelles $\mathbf{u} = [x, y, z]^T$ décrivent la position du point P . Il n'y a pas de coordonnées passives donc $\mathbf{z} = \mathbf{v}$. La modélisation et la paramétrisation introduites



(a) Schéma technologique adapté de [Pashkevich *et al.*, 2006]

(b) Schéma cinématique simplifié

FIGURE 3.11 – Représentation du mécanisme Orthoglide et de sa modélisation simplifiée.

dans Pashkevich *et al.* [2006] sont écrites dans le système non-linéaire :

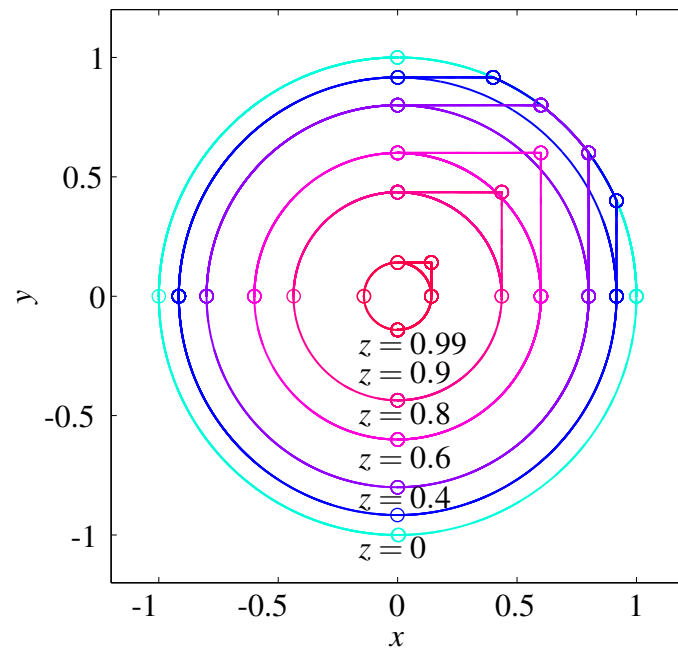
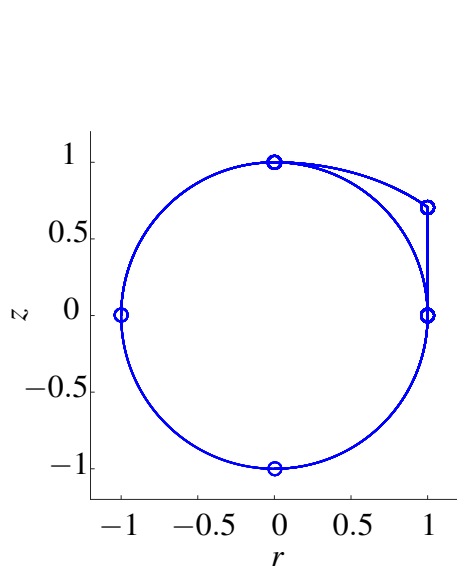
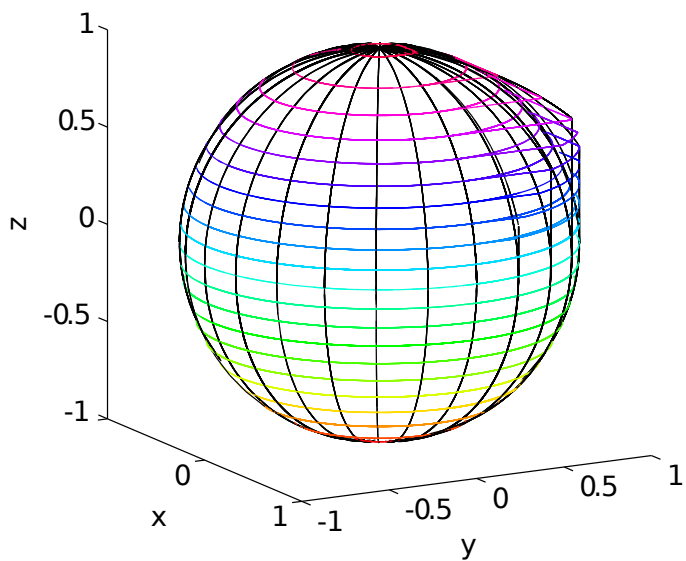
$$\mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \begin{bmatrix} (x - \rho_x)^2 + y^2 + z^2 - L^2 \\ x^2 + (y - \rho_y)^2 + z^2 - L^2 \\ x^2 + y^2 + (z - \rho_z)^2 - L^2 \end{bmatrix} = \mathbf{0}, \quad (3.10)$$

où $\mathbf{q} = [x, y, z, \rho_x, \rho_y, \rho_z]^T = [\mathbf{u}^T, \mathbf{z}^T]^T$. Un débattement limité dans l'intervalle $[0, 2L]$ pour chaque coordonnée active est pris en compte par le changement de variables $\underline{\mathbf{v}} = L + L \sin(\underline{\mathbf{v}})$.

L'espace des configurations est de dimension 3 et donc l'ensemble de singularités inverses \mathcal{S}_z est de dimension 2. Il est nécessaire de discrétiser le système étendu (3.6) pour pouvoir le résoudre sous forme de branche de solutions 1D par une méthode de continuation.

3.4.2 Stratégie de discrétisation

La continuation peut être appliquée pour le calcul des frontières d'un espace de travail de dimension supérieure ou égale à 3 [Haug *et al.*, 1996]. Pour cela, une stratégie de discrétisation de l'ensemble de solutions doit être adoptée. De manière générale il s'agit d'ajouter au système étendu une ou plusieurs relations linéaires liant des composantes du vecteur \mathbf{q} . Nous fixons simplement une coordonnée opérationnelle, ce qui correspond à un parcours dans une section de l'ensemble de solutions.

(a) Diagramme de bifurcations à z donné(b) Diagramme de bifurcations pour $\phi = \frac{\pi}{4}$ 

(c) Vue isométrique

FIGURE 3.12 – Représentations des frontières de l'espace de travail de l'Orthoglide. (a) Sections avec $z = \{0, 0.4, 0.6, 0.8, 0.9, 0.99\}$ représentées dans un diagramme de bifurcations en (x, y) . (b) Section avec $\phi = \frac{\pi}{4}$ représentée dans un diagramme de bifurcations en (r, z) . (c) Représentation cartésienne isométrique des sections assemblées.

Etant données les symétries du mécanisme, il est intéressant de déterminer des sections planes de l'espace de travail passant par l'axe z . Afin d'améliorer la représentation, des sections planes de l'espace de travail orthogonales à l'axe z sont de plus déterminées. Le processus de résolution par discrétisation consiste à :

1. convertir les coordonnées opérationnelles cartésiennes (x, y, z) en coordonnées cylindriques (r, ϕ, z) ,
2. fixer z ou ϕ ,
3. résoudre le système respectivement pour les inconnues $\mathbf{U} = [r, \phi, \mathbf{v}^\top, \xi^\top]^\top$ ou $\mathbf{U} = [r, z, \mathbf{v}^\top, \xi^\top]^\top$.

Le code utilisé pour le changement de variables \mathbf{v} , la conversion de \mathbf{u} et la résolution du système étendu (3.6) avec Diamanlab pour ϕ donné est présenté en annexe B. Le processus est répété pour différentes valeurs z ou ϕ , respectivement, pour obtenir un ensemble d'isolignes, comme montré en figure 3.12(a).

3.4.3 Espace de travail 3D

Les branches de solutions du système discrétisé sont assemblées en figure 3.12(c) pour figurer les frontières de l'espace de travail. La figure 3.12(c) laisse apparaître une sphère de rayon L et une surface convexe limitée par les configurations où au moins une jambe du mécanisme est orthogonale à l'axe définissant sa liaison prismatique active. La forme de la surface convexe, délimitée géométriquement par 3 cylindres de rayon L , peut être mieux observée sur des sections à z constant, comme représenté en figure 3.12(a). Chaque courbe de couleur représente l'ensemble des branches de solutions obtenues pour une valeur donnée de z . Les points de bifurcation représentent des positions de l'effecteur correspondant à plusieurs jeux de positions des liaisons actives. Le nombre de solutions au problème géométrique inverse y varie. La représentation obtenue est en accord avec la description géométrique élaborée dans Pashkevich *et al.* [2006].

3.5 Conclusion

L'usage de DA et de la méthode Diamant ont permis de développer un outil pour la détermination des frontières de l'espace de travail qui répond de manière plus satisfaisante aux besoins exprimés pour accélérer le processus de conception. Le tableau 3.2 compare les méthodes en termes :

- de généralité par rapport aux architectures de mécanismes,

- d'expertise requise pour la prise en main, la modélisation nécessaire, le temps d'implémentation,
- de précision de détermination d'un ensemble de solutions,
- de temps de calcul,
- de complétude de l'ensemble de solutions déterminé.

L'outil proposé hérite des capacités de la continuation de haut-degré en termes de précision, de temps de calcul et de continuité des branches de solutions. Les performances obtenues sont très encourageantes. En effet, les frontières de l'espace de travail sont calculées bien plus rapidement avec la méthode Diamant qu'avec la méthode branch-and-prune avec une précision obtenue très supérieure. Le processus développé pour la construction du système des frontières est intéressant car il permet l'automatisation du traitement de systèmes étendus. L'emploi de la DA permet la résolution du problème automatiquement à partir de la donnée seule du modèle géométrique implicite mis sous forme augmentée du mécanisme par l'utilisateur.

Comme illustré sur le mécanisme Orthoglide, les branches de solutions calculées comme polynômes de Taylor permettent la représentation de frontières d'un espace de travail de dimension 3. La discrétisation puis la reconstruction nécessaires à la représentation d'un espace de travail de dimension supérieure 2 nécessitent cependant encore une expertise. Le besoin pour une stratégie de discrétisation pourrait être contourné par l'emploi d'une stratégie de continuation en dimension élevée [Henderson, 2007]. La reconstruction de surfaces à partir des polynômes de Taylor pourrait permettre d'améliorer la représentation des frontières.

La stratégie adoptée pour le parcours automatisé du jeu de branches de solutions ne permet pas de garantir que l'ensemble des solutions est complètement déterminé. Le couplage des méthodes de continuation et d'analyse par intervalles pourrait permettre de pallier ce défaut [Ratschek et Rokne, 2005 ; Emeliyanenko *et al.*, 2009]. Ratschek et Rokne [2005] proposent une méthode qui garantit les solutions obtenues par continuation. Pour cela un couplage avec une méthode d'analyse par intervalles est réalisé. Le système est résolu par bisection jusqu'à ce que le continuum de solutions contenu dans la boîte soit régulier et puisse être résolu efficacement par continuation. Emeliyanenko *et al.* [2009] proposent un couplage inverse, autrement dit la continuation est principalement utilisée et l'analyse par intervalles exploitée dans les cas difficiles. Un tel couplage permettrait d'obtenir une méthode possédant les atouts des méthodes d'analyse par intervalles et de continuation qui sont complémentaires.

Méthode	Généricité	Expertise	Précision	Temps de calcul	Complétude des solutions
Géométrique	Méthodes valides pour une classe d'architecture	Prise en main et adaptation de la méthode complexe	Élevée	/	Dépend de la bonne application de la méthode
Algébrique	Méthodes valides pour une classe d'architecture	Prise en main et adaptation de la méthode complexe	Élevée	/	Dépend de la bonne application de la méthode
Analyse par intervalles	Formulation quadratique	Modélisation spécifique nécessaire. Systèmes étendus établis par l'utilisateur	Moyenne	Moyen	Solutions garanties mais discrètes
Continuation d'ordre 1	Équations de fermeture géométrique	Résolution dans des plans de coupes	Élevée	Faible	Ensemble de branches de solutions discrètes connectées par des points de bifurcation
Diamant	Équations de fermeture géométrique	Outil codé en Matlab. Systèmes étendus obtenus par DATS. Coupes en 3D	Élevée	Faible	Ensemble de branches de solutions continues connectées par des points de bifurcation

TABLEAU 3.2 – Positionnement de la méthode par comparaison aux méthodes de détermination de l'espace de travail existantes.

Chapitre 4

Analyse de singularités

« Plus l'univers se standardise, plus la singularité m'intéresse », Claude Sautet.

Sommaire

4.1 Définition et classification des singularités	98
4.1.1 Comportement en singularité	98
4.1.2 Classification de Zlatanov <i>et al.</i>	100
4.1.3 Méthodes de détection et d'identification	107
4.2 Détermination automatique des singularités	110
4.2.1 Stratégie de détection et de classification	110
4.2.2 Détection de l'ensemble des singularités	112
4.2.3 Classification des configurations singulières	114
4.3 Cas d'application	114
4.3.1 Description du mécanisme	114
4.3.2 Implémentation et résultats	115
4.4 Conclusion	118

Dans ce chapitre, nous nous intéressons à la deuxième caractéristique d'intérêt identifiée, à savoir les singularités cinématiques d'un mécanisme, et développons un outil dédié.

Un état de l'art des méthodes d'estimation et classification des singularités est tout d'abord réalisé en section 4.1, en revenant sur la définition des singularités et la classification du comportement en singularité. Il met l'accent sur les méthodes permettant une classification détaillée des types de singularités. Les limites des méthodes existantes permettent alors de préciser le besoin introduit au chapitre 1. L'approche de détermination des singularités introduite au chapitre 1 et son couplage avec les outils numériques exposés au chapitre 2 conduisent au développement de l'outil d'analyse de singularités présenté en section 4.2.

Notre outil est évalué sur un mécanisme plan de référence en section 4.3. Les apports et limites de l'outil de détection et de classification des singularités sont discutés en section 4.4.

4.1 Définition et classification des singularités

4.1.1 Comportement en singularité

La configuration d'un mécanisme est décrite par le vecteur des coordonnées géométriques $\mathbf{q} = [\mathbf{u}^\top, \mathbf{v}^\top, \mathbf{w}^\top]^\top$ où \mathbf{u} sont les coordonnées opérationnelles, \mathbf{v} sont les coordonnées actives et \mathbf{w} sont les coordonnées passives. Deux problèmes cinématiques peuvent être posés pour une configuration donnée \mathbf{q} d'un mécanisme [Zlatanov *et al.*, 1995] :

- le *problème cinématique inverse* (PCI) : Trouver un mouvement instantané $\dot{\mathbf{q}}$ admissible pour des vitesses de sortie $\dot{\mathbf{u}}$ données ;
- le *problème cinématique direct* (PCD) : Trouver un mouvement instantané $\dot{\mathbf{q}}$ admissible pour des vitesses d'entrée $\dot{\mathbf{v}}$ données.

Un mouvement instantané $\dot{\mathbf{q}} = [\dot{\mathbf{u}}^\top, \dot{\mathbf{v}}^\top, \dot{\mathbf{w}}^\top]^\top$ est défini par l'assemblage de jeux de vitesses de sortie $\dot{\mathbf{u}}$, vitesses d'entrée $\dot{\mathbf{v}}$ et vitesses intermédiaires $\dot{\mathbf{w}}$. Le vecteur $\dot{\mathbf{z}} = [\dot{\mathbf{v}}^\top, \dot{\mathbf{w}}^\top]^\top$ contenant les vitesses d'entrée $\dot{\mathbf{v}}$ et les vitesses intermédiaires $\dot{\mathbf{w}}$ recherchées dans le PCI est introduite. De même on introduit le vecteur $\dot{\mathbf{y}} = [\dot{\mathbf{u}}^\top, \dot{\mathbf{w}}^\top]^\top$ contenant les vitesses de sortie $\dot{\mathbf{u}}$ et les vitesses intermédiaires $\dot{\mathbf{w}}$ recherchées dans le PCD.

Les singularités inverses et directes sont des situations d'indétermination respectivement dans le PCI et le PCD. Le modèle cinématique implicite a été défini comme :

$$\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{q}} = \mathbf{0}, \quad (4.1)$$

Dans le cas du PCI, ce modèle peut être reformulé en :

$$\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{z}} + \mathbf{R}_{\mathbf{u}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{u}} = \mathbf{0},$$

avec $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{z}}$ et $\mathbf{R}_{\mathbf{u}}^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{u}}$. Si la matrice $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{q})$ est inversible, alors une solution au PCI est $\dot{\mathbf{z}} = -(\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{q}))^{-1} \mathbf{R}_{\mathbf{u}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{u}}$. Une *singularité inverse* est une configuration \mathbf{q} pour laquelle la matrice $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}$ connaît une perte de rang [Bohigas, 2013], *i.e.* $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{q})$ n'est pas inversible.

Dans le cas du PCD, le modèle cinématique implicite (4.1) peut être reformulé :

$$\mathbf{R}_{\mathbf{y}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{y}} + \mathbf{R}_{\mathbf{v}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{v}} = \mathbf{0},$$

avec $\mathbf{R}_y^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{y}}$ et $\mathbf{R}_v^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{v}}$. Une *singularité directe* est une configuration \mathbf{q} pour laquelle la matrice $\mathbf{R}_y^{\mathcal{C}} = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{y}}$ connaît une perte de rang.

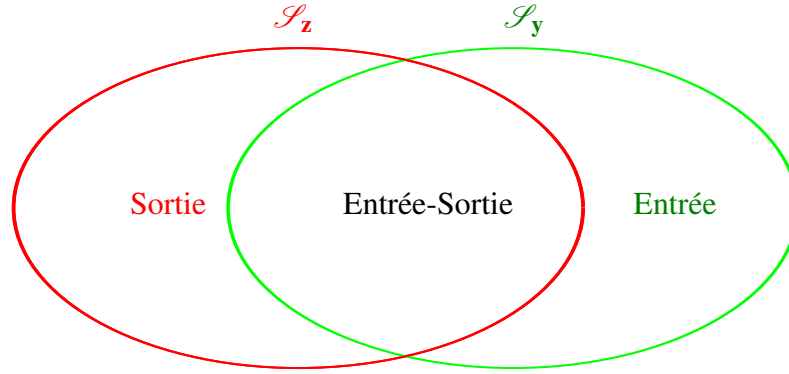


FIGURE 4.1 – Distinction de trois ensembles de singularités.

Comme présenté en figure 4.1, l'ensemble des singularités \mathcal{S} est constitué de l'union de l'ensemble des singularités inverses noté \mathcal{S}_z et de l'ensemble des singularités directes noté \mathcal{S}_y : $\mathcal{S} = \mathcal{S}_z \cup \mathcal{S}_y$ [Zlatanov *et al.*, 1998]. Une configuration singulière peut appartenir à :

- l'ensemble des *singularités de sortie* [Bohigas, 2013] $\mathcal{S}_z \setminus \mathcal{S}_y$, incluant les singularités de « type I » [Gosselin et Angeles, 1990] ou singularités sérielles ;
- l'ensemble des *singularités d'entrée* $\mathcal{S}_y \setminus \mathcal{S}_z$, incluant les singularités de « type II » ou singularités parallèles ;
- l'ensemble des *singularités entrée-sortie* $\mathcal{S}_z \cap \mathcal{S}_y$, incluant les singularités de « type III » ou singularités combinées.

Comme introduit en section 1.4.2, les 3 ensembles de « type I », « type II » et « type III » classiquement étudiés [Gosselin et Angeles, 1990] et définis respectivement par la perte de rang de $\mathbf{R}_v^{\mathcal{C}}$, de $\mathbf{R}_u^{\mathcal{C}}$ et de $(\mathbf{R}_u^{\mathcal{C}} \text{ et } \mathbf{R}_v^{\mathcal{C}})$ ne permettent pas de détecter tous les lieux de singularités, ni de classer tous les comportements. D'autres singularités ont ainsi été identifiées, d'une part en tenant compte de la présence des liaisons passives, d'autre part en modélisant les 6 mouvements possibles pour l'effecteur même pour des mécanismes à mobilité réduite. En effet, le nombre de mobilités peut augmenter en configuration singulière. Zlatanov *et al.* [1995] proposent donc une classification plus détaillée permettant de déterminer tous les comportements dégénérés.

Dans cette thèse, nous considérons cette classification intéressante et requise pour fournir une interprétation fine du comportement cinématique en singularité.

4.1.2 Classification de Zlatanov *et al.*

L'ensemble $\mathcal{S}_z \cup \mathcal{S}_y$ des singularités directes et inverses englobe toutes les configurations pour lesquelles le comportement d'un mécanisme est dégénéré. En ces singularités, le mécanisme connaît une modification du nombre, de la nature ou de l'amplitude des mouvements admissibles par l'effecteur, voire du degré de mobilité de la chaîne cinématique. La classification des singularités par Zlatanov *et al.* [1995, 1998] est fondée sur la formulation des mouvements dégénérés pouvant apparaître. Un *type de singularité* décrit une situation d'indétermination des problèmes cinématiques, impliquant une modification des mouvements admissibles par un mécanisme, comme illustré en figure 4.3. On dit alors qu'on *identifie* le ou les types de singularité auxquels appartient une configuration. Une *classe de singularité* est définie comme une combinaison de types de singularités. On dit qu'on peut alors *classifier* une configuration dans la classe de singularité à laquelle elle appartient.

4.1.2.1 Les types de singularité

Six types de singularité correspondant chacun à une situation d'indétermination élémentaire des PCI et PCD ont été introduits dans [Zlatanov *et al.*, 1995]. Nous détaillons ici leur formulation, leur interprétation en terme de modification des mouvements admissibles en s'appuyant sur la figure 4.3.

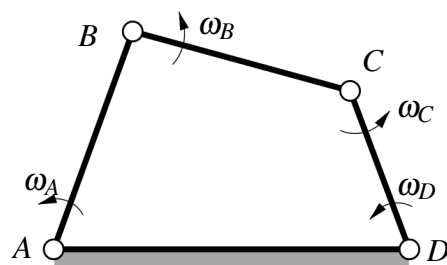


FIGURE 4.2 – Schéma cinématique d'un mécanisme 4 barres. ω_A désigne la vitesse de rotation de AB par rapport à AD . Les vitesses d'entrée, de sortie et intermédiaires sont respectivement notées ω_A , ω_D et (ω_B, ω_C) .

Le mécanisme 4 barres présenté en figure 4.2 est utilisé pour illustrer l'identification de types de singularités. Il est constitué d'une chaîne cinématique fermée RRRR composée de liaisons pivot aux points A , B , C et D . La base du mécanisme est le segment AD . L'effecteur est le segment CD dont la pose est caractérisée par son orientation par rapport à la base AD . La vitesse de sortie est donc la vitesse de rotation de la liaison D , *i.e.* $\dot{\mathbf{u}} = \omega_D$. La liaison pivot au point A est actionnée et sa vitesse de rotation est la vitesse d'entrée $\dot{\mathbf{v}} = \omega_A$. Les liaisons des points B et C sont passives, les vitesses intermédiaires sont donc $\dot{\mathbf{w}} = [\omega_B, \omega_C]^T$.

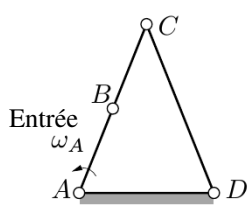
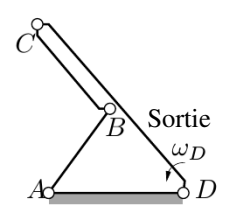
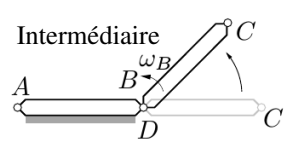
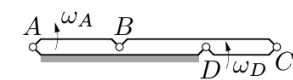
Types	RI	IO	RO	II	RPM	IIM
Indétermination	Le PCI a une infinité de solutions pour $\dot{\mathbf{u}} = \mathbf{0}$	Le PCI n'a pas de solution pour $\dot{\mathbf{u}}$ donné	Le PCD a une infinité de solutions pour $\dot{\mathbf{v}} = \mathbf{0}$	Le PCD n'a pas de solution pour $\dot{\mathbf{v}}$ donné	Les PCI et PCD ont une infinité de solutions pour $\dot{\mathbf{u}} = \mathbf{0}$ et $\dot{\mathbf{v}} = \mathbf{0}$	Les PCI et PCD ont une infinité de solutions pour $\dot{\mathbf{u}}$ et $\dot{\mathbf{v}}$ quelconques
Formulation	$\exists(\dot{\mathbf{v}} \neq \mathbf{0}, \dot{\mathbf{w}}) \mathbf{R}_q^{\mathcal{L}} \cdot \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{bmatrix} = \mathbf{0}$	$\exists \dot{\mathbf{u}} \nexists(\dot{\mathbf{v}}, \dot{\mathbf{w}}) \mathbf{R}_q^{\mathcal{L}} \cdot \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{bmatrix} = \mathbf{0}$	$\exists(\dot{\mathbf{u}} \neq \mathbf{0}, \dot{\mathbf{w}}) \mathbf{R}_q^{\mathcal{L}} \cdot \begin{bmatrix} \dot{\mathbf{u}} \\ \mathbf{0} \\ \dot{\mathbf{w}} \end{bmatrix} = \mathbf{0}$	$\exists \dot{\mathbf{v}} \nexists(\dot{\mathbf{u}}, \dot{\mathbf{w}}) \mathbf{R}_q^{\mathcal{L}} \cdot \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{bmatrix} = \mathbf{0}$	$\exists \dot{\mathbf{w}} \neq \mathbf{0} \mathbf{R}_q^{\mathcal{L}} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{w}} \end{bmatrix} = \mathbf{0}$	$\text{rang}(\mathbf{R}_q^{\mathcal{L}}) < N$
Interprétation cinématique	ω_A quelconque pour $\omega_D = 0$	$\omega_D = 0$	ω_D quelconque pour $\omega_A = 0$	$\omega_A = 0$	ω_B quelconque pour $\omega_A = \omega_C = 0$	$\omega_A, \omega_B, \omega_C$ et ω_D quelconques
Une condition géométrique	A, B et C alignés	A, B et C alignés	B, C et D alignés	B, C et D alignés	$AB = AD, CB = CD$, et B et D confondus	A, B et C alignés, et B, C et D alignés
Une configuration	 (a) (RI,IO)		 (b) (RO,II)		 (c) (RPM,IO,II)	 (d) (RI,RO,IIM)

FIGURE 4.3 – Description des 6 types de singularités, avec illustration dans le cas d'un mécanisme 4 barres.

Des configurations du mécanisme 4 barres décrivant les 6 types de singularité sont données en figure 4.3. Le comportement cinématique dégénéré ainsi qu'une condition géométrique d'occurrence sont ainsi illustrés.

L'appartenance d'une configuration \mathbf{q} à un type de singularité est définie par la satisfaction des relations cinématiques suivantes :

Entrée redondante (RI, pour « redundant input ») Une configuration est une singularité de type RI s'il existe une solution au PCI non-nulle pour une vitesse de sortie $\dot{\mathbf{u}} = \mathbf{0}$:

$$\{RI\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \exists (\dot{\mathbf{v}} \neq \mathbf{0}, \dot{\mathbf{w}}) \text{ tel que } \mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \cdot [\mathbf{0}^T, \dot{\mathbf{v}}^T, \dot{\mathbf{w}}^T]^T = \mathbf{0} \right\}. \quad (4.2)$$

Un exemple de ce type de singularité est donné en figure 4.3(a). Pour une valeur de vitesse de sortie nulle $\omega_D = \mathbf{0}$, la vitesse d'entrée ω_A peut prendre n'importe quelle valeur : l'entrée est redondante.

Sortie impossible (IO, pour « impossible output ») Une configuration est une singularité de type IO s'il existe une vitesse de sortie $\dot{\mathbf{u}}$ pour laquelle le PCI n'a pas de solution :

$$\{IO\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \exists \dot{\mathbf{u}} \text{ pour lequel } \nexists (\dot{\mathbf{v}}, \dot{\mathbf{w}}) \text{ tel que } \mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \cdot [\dot{\mathbf{u}}^T, \dot{\mathbf{v}}^T, \dot{\mathbf{w}}^T]^T = \mathbf{0} \right\}. \quad (4.3)$$

Un exemple de ce type de singularité est donné en figure 4.3(a). En cette configuration, toute vitesse de sortie ω_D non-nulle est impossible : la sortie est impossible.

Dans le cas général, l'effecteur connaît une perte de degré(s) de liberté.

Notons que les singularités de type IO correspondent effectivement à l'indétermination du PCI comme énoncé par Zlatanov [1998 - Proposition 3.16] : « an IO-type singularity implies an insolvable IIKP ». En effet, les singularités de « type 1 » sont des configurations où seul le PCI est indéterminé et sont de type RI et IO, comme illustré en figure 4.3(a). L'énoncé « An IO-type singularity implies an insolvable FIKP » dans [Zlatanov *et al.*, 1995 - p.567] est par contre erroné.

Sortie redondante (RO, pour « redundant output ») Une configuration est une singularité de type RO s'il existe une solution au PCD non-nulle pour une vitesse d'entrée $\dot{\mathbf{v}} = \mathbf{0}$:

$$\{RO\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \exists (\dot{\mathbf{u}} \neq \mathbf{0}, \dot{\mathbf{w}}) \text{ tel que } \mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \cdot [\dot{\mathbf{u}}^T, \mathbf{0}^T, \dot{\mathbf{w}}^T]^T = \mathbf{0} \right\}. \quad (4.4)$$

Un exemple de ce type de singularité est donné en figure 4.3(b). Pour une valeur de vitesse d'entrée nulle $\omega_A = \mathbf{0}$, la vitesse de sortie ω_D peut prendre n'importe quelle valeur : la sortie est redondante.

Entrée impossible (II, pour « impossible input ») Une configuration est une singularité de type II s'il existe une vitesse d'entrée $\dot{\mathbf{v}}$ pour laquelle le PCD n'a pas de solution :

$$\{II\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \exists \dot{\mathbf{v}} \text{ pour lequel } \nexists (\dot{\mathbf{u}}, \dot{\mathbf{w}}) \text{ tel que } \mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \cdot [\dot{\mathbf{u}}^T, \dot{\mathbf{v}}^T, \dot{\mathbf{w}}^T]^T = \mathbf{0} \right\}. \quad (4.5)$$

Un exemple de ce type de singularité est donné en figure 4.3(b). En cette configuration, toute vitesse d'entrée ω_A non-nulle est impossible : l'entrée est impossible.

Dans le cas général, les liaisons actives ne peuvent toutes être actionnées indépendamment.

Mouvement passif redondant (RPM, pour « redundant passive motion ») Une configuration est une singularité de type RPM si les PCD et PCI ont une infinité de solutions pour une vitesse d'entrée $\dot{\mathbf{v}} = \mathbf{0}$ et une vitesse de sortie $\dot{\mathbf{u}} = \mathbf{0}$ données :

$$\{RPM\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \exists \dot{\mathbf{w}} \neq \mathbf{0} \text{ tel que } \mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \cdot [\mathbf{0}^T, \mathbf{0}^T, \dot{\mathbf{w}}^T]^T = \mathbf{0} \right\}. \quad (4.6)$$

Un exemple de ce type de singularité est donné en figure 4.3(c). Pour des valeurs des vitesses d'entrée ω_A et de sortie ω_D nulles, la vitesse intermédiaire ω_B peut prendre n'importe quelle valeur : le mouvement passif est redondant.

Gain instantané de mobilité (IIM, pour « increased instantaneous motion ») Une configuration est une singularité de type IIM si les PCD et PCI ont une infinité de solutions pour toute vitesse d'entrée $\dot{\mathbf{v}}$ et toute vitesse de sortie $\dot{\mathbf{u}}$:

$$\{IIM\} = \left\{ \mathbf{q} \in \mathcal{S} \subset \mathcal{C} \mid \text{rang} \left(\mathbf{R}_q^{\mathcal{C}}(\mathbf{q}) \right) < N \right\}. \quad (4.7)$$

Un exemple de ce type de singularité est donné en figure 4.3(d). Les vitesses d'entrée ω_A et de sortie ω_D peuvent prendre indépendamment n'importe quelle valeur : on a un gain instantané de mobilité. Autrement dit, il n'est pas possible d'assurer l'absence de mouvement dans le mécanisme par l'actionnement d'une seule liaison. Une singularité de type IIM est

solution du système étendu [Bohigas, 2013 - Section 2.3] :

$$\mathbf{R}^{\{IIM\}}(\mathbf{q}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{L}}(\mathbf{q}) \\ [\mathbf{R}_{\mathbf{q}}^{\mathcal{L}}(\mathbf{q})]^{\top} \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{bmatrix} = \mathbf{0}. \quad (4.8)$$

Des singularités dites architecturales et de contrainte ont été étudiées dans la littérature.

Les *singularités de contrainte* [Gregorio et Parenti-Castelli, 1999 ; Zlatanov *et al.*, 2002], aussi appelées « uncertainty configuration » [Hunt, 1978], correspondent à des configurations pour lesquelles la chaîne cinématique dans son ensemble gagne un degré de mobilité. Peu importe les m liaisons bloquées, le mécanisme ne peut être immobilisé.

Les singularités de contrainte n'ont été observées que pour des mécanismes parallèles à mobilité réduite. L'analyse des singularités d'un tel mécanisme est complète à condition qu'il soit étudié au moyen d'un modèle qui puisse décrire les singularités de contrainte. Celui-ci doit donc permettre de modéliser les 6 mouvements possibles de l'effecteur. Une singularité de contrainte est toujours une singularité de type IIM. Si le gain de mobilité résulte en des mouvements d'amplitude finie, il s'agit de *singularités architecturales* [Ma et Angeles, 1991 ; Gregorio et Parenti-Castelli, 1999 ; Borrás *et al.*, 2008], aussi appelées « self motion » [Hunt, 1978].

4.1.2.2 Conditions algébriques de singularité

Les 6 relations (4.2)-(4.7) caractérisent donc les différentes situations d'indétermination des PCI et PCD possibles en configuration singulière. Zlatanov *et al.* [1998] ont ensuite formulé 8 conditions permettant d'identifier l'appartenance d'une configuration aux types de singularités et aux ensembles de singularités inverses et directes :

$$\mathbf{q} \in \{RI\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{z}}^{\mathcal{L}}(\mathbf{q})) < \text{rang}(\mathbf{R}_{\mathbf{w}}^{\mathcal{L}}(\mathbf{q})) + m \Leftrightarrow r_{\mathbf{z}}(\mathbf{q}) > r_{\mathbf{w}}(\mathbf{q}) \geq 0, \quad (4.9)$$

$$\mathbf{q} \in \{RO\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{y}}^{\mathcal{L}}(\mathbf{q})) < \text{rang}(\mathbf{R}_{\mathbf{w}}^{\mathcal{L}}(\mathbf{q})) + m \Leftrightarrow r_{\mathbf{y}}(\mathbf{q}) > r_{\mathbf{w}}(\mathbf{q}), \quad (4.10)$$

$$\mathbf{q} \in \{RPM\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{w}}^{\mathcal{L}}(\mathbf{q})) < N - m \Leftrightarrow r_{\mathbf{w}}(\mathbf{q}) > 0 \quad (4.11)$$

$$\mathbf{q} \in \{IO\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{z}}^{\mathcal{L}}(\mathbf{q})) < \text{rang}(\mathbf{R}_{\mathbf{q}}^{\mathcal{L}}(\mathbf{q})) \Leftrightarrow r_{\mathbf{z}}(\mathbf{q}) > r_{\mathbf{q}}(\mathbf{q}) \geq 0, \quad (4.12)$$

$$\mathbf{q} \in \{II\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{y}}^{\mathcal{L}}(\mathbf{q})) < \text{rang}(\mathbf{R}_{\mathbf{q}}^{\mathcal{L}}(\mathbf{q})) \Leftrightarrow r_{\mathbf{y}}(\mathbf{q}) > r_{\mathbf{q}}(\mathbf{q}) \geq 0, \quad (4.13)$$

$$\mathbf{q} \in \{IIM\} \Leftrightarrow \text{rang}(\mathbf{R}_{\mathbf{q}}^{\mathcal{L}}(\mathbf{q})) < N \Leftrightarrow r_{\mathbf{q}}(\mathbf{q}) > 0, \quad (4.14)$$

$$\begin{aligned} \mathbf{q} \in \mathcal{S}_z &\Leftrightarrow \mathbf{q} \in \{RI\} \text{ ou } \mathbf{q} \in \{RPM\} \Leftrightarrow \mathbf{q} \in \{IO\} \text{ ou } \mathbf{q} \in \{IIM\} \\ &\Leftrightarrow \mathbf{R}_z^{\mathcal{C}} \text{ est singulière} \Leftrightarrow r_z(\mathbf{q}) > 0, \end{aligned} \quad (4.15)$$

$$\begin{aligned} \mathbf{q} \in \mathcal{S}_y &\Leftrightarrow \mathbf{q} \in \{RO\} \text{ ou } \mathbf{q} \in \{RPM\} \Leftrightarrow \mathbf{q} \in \{II\} \text{ ou } \mathbf{q} \in \{IIM\} \\ &\Leftrightarrow \mathbf{R}_y^{\mathcal{C}} \text{ est singulière} \Leftrightarrow r_y(\mathbf{q}) > 0. \end{aligned} \quad (4.16)$$

Dans ces formules, $r_x = \dim(\mathbf{R}_x^{\mathcal{C}}(\mathbf{q})) - \text{rang}(\mathbf{R}_x^{\mathcal{C}}(\mathbf{q}))$ désigne l'ordre de la perte de rang de $\mathbf{R}_x^{\mathcal{C}}$. Ainsi $r_x = 0$ signifie que $\mathbf{R}_x^{\mathcal{C}}$ est régulière, $r_x > 0$ signifie que $\mathbf{R}_x^{\mathcal{C}}$ est singulière. La *mobilité* du mécanisme, notée m , correspond au nombre de coordonnées indépendantes qu'il est nécessaire de commander pour contrôler le déplacement de chacun des membres. Le nombre de liaisons du mécanisme est noté N .

4.1.2.3 Les 21 classes de singularités

Toutes les situations d'indétermination élémentaires décrites précédemment ne peuvent être rencontrées indépendamment. Zlatanov *et al.* [1995] ont montré qu'une configuration singulière d'un mécanisme doit appartenir à au moins deux types de singularités. Elle doit notamment appartenir à au moins un des types RI, RO ou RPM, et par ailleurs au moins un des types IO, II et IIM. Seules certaines combinaisons, appelées *classes de singularité*, sont en fait possibles. Elles sont répertoriées dans le tableau 4.1. Au total 21 classes correspondant à une situation d'indétermination spécifique des PCI et PCD et un comportement cinématique donné du mécanisme existent [Zlatanov *et al.*, 1995]. Une numérotation des classes est introduite dans le tableau 4.1 pour faciliter leur désignation.

Le tableau 4.1 montre que si seul le PCI est indéterminé ($r_z > 0$), alors la singularité est de classe (RI,IO) correspondant aux singularités de sortie $\mathcal{S}_z \setminus \mathcal{S}_y$ où l'entrée est redondante et la sortie est impossible. Elle est numérotée 1 et elle correspond aux singularités de « type I » [Gosselin et Angeles, 1990]. De même si seul le PCD est indéterminé, alors la singularité est de classe (RO,II) correspondant aux singularités d'entrée $\mathcal{S}_y \setminus \mathcal{S}_z$ où la sortie est redondante et l'entrée est impossible. Elle est numérotée 2 et elle correspond aux singularités de « type II » [Gosselin et Angeles, 1990].

Les 19 autres classes de singularités ne peuvent être rencontrées que lorsque le PCD et le PCI sont indéterminés et correspondent aux singularités entrée-sortie $\mathcal{S}_z \cap \mathcal{S}_y$. Ces singularités incluent les singularités de « type III » [Gosselin et Angeles, 1990] qui incluent les singularités de la classe (RI,IO,RO,II) numérotée 3. Les singularités entrée-sortie $\mathcal{S}_z \cap \mathcal{S}_y$ incluent également des singularités de type RPM ou IIM numérotées de 4 à 21.

Type	Condition	IO (4.12) ($r_z > r_q = 0$)	II (4.13) ($r_y > r_q = 0$)	IIM (4.14) ($r_q > 0$)	IO,II (4.12) et (4.13)	IO,IIM (4.12) et (4.14)	II,IIM (4.13) et (4.14)	IO,II,IIM (4.12), (4.13) et (4.14)
RI	(4.9) ($r_z > r_w = 0$)	1						
RO	(4.10) ($r_y > r_w = 0$)		2					
RPM	(4.11) ($r_w > 0$)			6	5			7
RI,RO	(4.9) et (4.10)			4	3	9	11	15
RI,RPM	(4.9) et (4.11)				10	13		17
RO,RPM	(4.10) et (4.11)				12		14	19
RI,RO,RPM	(4.9) et (4.10) et (4.11)			8	16	18	20	21

TABLEAU 4.1 – Les 21 classes de singularité numérotées comme combinaisons possibles des 6 types de singularité. Les conditions sont données par la relation et par l'ordre de la perte de rang r_x de la jacobienne $\mathbf{R}_x^{\mathcal{C}}$, avec $r_x = \dim(\mathbf{R}_x^{\mathcal{C}}) - \text{rang}(\mathbf{R}_x^{\mathcal{C}})$.

4.1.3 Méthodes de détection et d'identification

Nous avons présenté au chapitre 3 une méthode générique de détermination des frontières de l'espace de travail. Ces frontières correspondent à la projection des singularités inverses dans l'espace des coordonnées opérationnelles. La détermination des frontières de l'espace de travail constitue donc déjà un problème de détection de singularités. Nous avons alors souligné en section 3.1 l'existence d'approches algébriques, géométriques, et numériques pour ce faire.

Des méthodes basées sur la théorie des visseurs [Tsai, 1998 ; Joshi et Tsai, 2002 ; Bonev *et al.*, 2003 ; Rubbert *et al.*, 2014], des méthodes algébriques [Ben-Horin et Shoham, 2006 ; Kong et Gosselin, 2001 ; Amine *et al.*, 2012 ; Rubbert *et al.*, 2012], géométriques [Merlet, 1989 ; Hao et McCarthy, 1998 ; Park et Kim, 1999 ; Zlatanov *et al.*, 1998 ; Amine *et al.*, 2012], analytiques [Lai et Yang, 1986 ; Ahmad et Luo, 1988 ; Gosselin et Angeles, 1990 ; Kong et Gosselin, 2001] et numériques [Merlet, 2007] ont été proposées pour l'analyse des singularités. Leur usage n'a pas été envisagé pour permettre une classification détaillée des singularités, qui va en fait au delà de la notion de singularités inverse et directe comme décrit en section 4.1.2. Nous présentons dans la suite un état de l'art focalisé sur les méthodes permettant une telle classification détaillée.

4.1.3.1 Méthode algébrique et géométrique

La classification des singularités a été proposée par Zlatanov *et al.* [1998]. Une approche est alors également proposée pour la détermination des singularités avec séquentiellement la détection d'un ensemble de singularités suivie de son identification.

Les auteurs tirent parti de la structure du tableau 4.1 pour proposer leur stratégie de détermination. Si l'identification d'une singularité écarte l'appartenance aux singularités de type IIM et de type RPM, alors seules les classes 1, 2 et 3 de singularités (RI,IO), (RO,II) et (RI,IO,RO,II) sont possibles. Les singularités peuvent être classifiées au moyen des conditions (4.15) et (4.16) seulement. De plus, Zlatanov *et al.* [1998] supposent que les ensembles de singularités {IIM} et {RPM} sont de dimension inférieure à celle de l'ensemble de singularités \mathcal{S} . Cette stratégie possède l'avantage de diminuer le nombre de classes possibles lors de l'identification.

La stratégie proposée consiste donc en :

1. la détection des singularités IIM et RPM avec les conditions (4.14) et (4.11),
2. la classification des singularités IIM et RPM (classes 4 à 21) avec les conditions (4.9), (4.10), (4.12) et (4.13) et par opérations de soustraction et d'intersection sur les ensembles obtenus,

3. la détection des singularités de classes (RI,IO), (RO,II) et (RI,IO,RO,II) avec les conditions (4.15) et (4.16), et par soustraction des ensembles obtenus en 1.,
4. la classification des singularités de classes (RI,IO), (RO,II) et (RI,IO,RO,II) par intersection des 2 ensembles obtenus en 3.

Des méthodes algébriques et géométriques sont exploitées pour la détermination successive des différentes classes de singularités d'un mécanisme plan de mobilité 2 et spatial de mobilité 6. Mais aucune méthode de détermination générique n'est proposée dans l'article. Zlatanov *et al.* [1998] suggèrent la résolution d'un système d'équations non-linéaires pour les étapes de détection, mais soulignent la difficulté d'une résolution symbolique et le coût de calcul d'une résolution numérique, en particulier lorsque le nombre de mobilités augmente.

4.1.3.2 Méthode numérique issue de [Bohigas *et al.*, 2014]

Bohigas *et al.* ont proposé 2 méthodes pour déterminer les singularités en exploitant le cadre de travail de Zlatanov *et al.* [1998]. Dans une première méthode décrite dans [Bohigas *et al.*, 2014], des systèmes d'équations non-linéaires définissant les types de singularités sont introduits et il est proposé leur résolution par une méthode d'analyse par intervalles.

Des systèmes étendus représentant chacun des 6 types de singularités sont établis par assemblage du modèle géométrique implicite et des équations de vitesse. Par exemple les singularités de type RPM sont solutions du système étendu exprimant la perte de rang de la matrice $\mathbf{R}_w^{\mathcal{C}}$:

$$\mathbf{R}^{\{RPM\}}(\mathbf{q}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{q}) \\ (\mathbf{R}_w^{\mathcal{C}}(\mathbf{q}) \cdot \xi) \\ \xi^T \cdot \xi - 1 \end{bmatrix} = \mathbf{0}.$$

Les systèmes sont traités un à un par la méthode branch-and-prune [Porta *et al.*, 2009 ; Bohigas *et al.*, 2012a] de résolution de systèmes d'équations non-linéaires présentée au chapitre 3. Un effort de formulation spécifique est alors nécessaire pour appliquer cette méthode avec des temps de calcul raisonnables.

La méthode décompose une boîte initiale englobant l'ensemble des configurations admissibles \mathcal{C} en un ensemble spécifique de boîtes approchant les solutions avec une précision donnée. Chaque type de singularités est représenté par un ensemble de boîtes issu d'une décomposition spécifique due aux différents systèmes. Chaque configuration singulière appartient à au moins deux types de singularité. Pour classifier une singularité, il est donc nécessaire de déterminer la coïncidence de deux boîtes de types différents. Celles-ci étant établies lors de résolutions spécifiques, une intersection seulement partielle des boîtes de

deux types est possible. La classification de la singularité dans ce cas pose problème et nécessite un post-traitement manuel.

4.1.3.3 Méthode numérique issue de [Bohigas *et al.*, 2012b]

Dans une seconde méthode, Bohigas *et al.* [2012b] proposent tout comme dans [Zlatanov *et al.*, 1998] une détermination par une étape de détection suivie d'une étape de classification. Dans un premier temps, la détection de l'ensemble des singularités est réalisée par résolution successive des 2 systèmes étendus représentant les sous-ensembles de singularités inverses \mathcal{S}_z et directes \mathcal{S}_y . La méthode de branch-and-prune est alors exploitée.

Dans un second temps, chacune des boîtes est classifiée. Des systèmes étendus d'équations non-linéaires dont les solutions appartiennent à un ou plusieurs types de singularités sont formulés [Bohigas *et al.*, 2012b - Tableau 2]. La méthode branch-and-prune est de nouveau exploitée pour cette phase de classification des singularités. Si la méthode renvoie une absence de solutions pour une boîte et un système testé, alors la boîte ne peut être du type défini par le système. Une série de tests est appliquée, permettant d'éliminer progressivement les classes de singularités telles que décrites dans le tableau 4.1.

La plupart des singularités appartiennent uniquement aux sous-ensembles \mathcal{S}_y ou \mathcal{S}_z et correspondent aux singularités d'entrée (RO,II) et de sortie (RI,IO). Les autres classes de singularités apparaissent à l'intersection $\mathcal{S}_y \cap \mathcal{S}_z$ des deux sous-ensembles. On retrouve ici le problème d'intersection d'ensembles de boîtes identifié en section 4.1.3.2 sur la méthode de Bohigas *et al.* [2014]. La classification par élimination ne permet pas toujours de déterminer une unique classe de singularité pour les boîtes de $\mathcal{S}_y \cap \mathcal{S}_z$. Ceci amène Bohigas *et al.* à lever les dernières indéterminations par inspection directe et au cas par cas des figures.

La méthode présente l'avantage de garantir la détection complète de $\mathcal{S}_y \cup \mathcal{S}_z$, mais la classification de $\mathcal{S}_y \cap \mathcal{S}_z$ peut échouer. En effet, la méthode ne permet pas d'identifier avec précision les singularités de l'ensemble $\mathcal{S}_y \cap \mathcal{S}_z$ auquel appartiennent 19 des 21 classes de singularités. La détermination des singularités de $\mathcal{S}_y \cap \mathcal{S}_z$ est donc partielle.

Les besoins d'efficacité et de généricité pour nos outils d'analyse des mécanismes nous portent naturellement à exploiter la méthode Diamant pour la détection des configurations singulières. Nous proposons d'étendre l'outil de détermination des frontières de l'espace de travail proposé en section 3.2 à la détermination des configurations singulières.

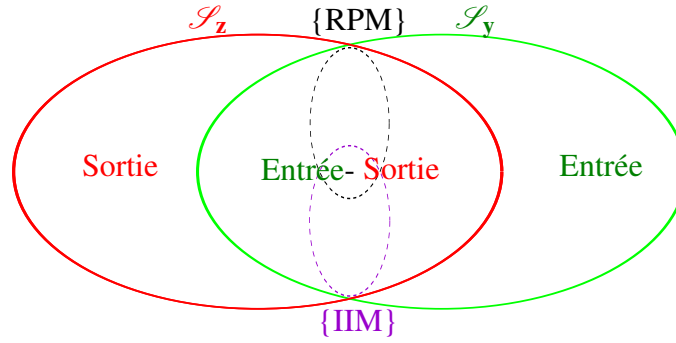


FIGURE 4.4 – Inclusion des types de singularités RPM et IIM dans les trois ensembles de singularités.

4.2 Détermination automatique des singularités

La démarche que nous proposons pour la détermination des configurations singulières \mathcal{S} exploite la structure décrite sur la figure 4.4 qui a déjà été exploitée dans [Zlatanov *et al.*, 1998] et [Bohigas *et al.*, 2012b] pour construire leurs stratégies de détermination telles que présentées en section 4.1.3. Notre méthode de détermination des singularités repose sur la détection de toutes les configurations singulières \mathcal{S} suivie de leur classification.

4.2.1 Stratégie de détection et de classification

Notre stratégie de détermination de singularités consiste en la détection des ensembles de singularités inverses \mathcal{S}_z et directes \mathcal{S}_y . Les singularités inverses sont solutions du système étendu (1.17) :

$$\mathbf{R}^{\mathcal{S}_z}(\mathbf{u}, \mathbf{z}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{L}}(\mathbf{u}, \mathbf{z}) \\ [\mathbf{R}_z^{\mathcal{L}}(\mathbf{u}, \mathbf{z})]^T \cdot \xi \\ \xi^T \cdot \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (4.17)$$

où $\mathbf{q} = [\mathbf{u}^T, \mathbf{z}^T]^T$ sont les coordonnées géométriques, $\mathbf{z} = [\mathbf{v}^T, \mathbf{w}^T]^T$ et ξ est un vecteur unitaire. Les singularités directes sont solutions du système étendu (1.18) rappelé ici :

$$\mathbf{R}^{\mathcal{S}_y}(\mathbf{v}, \mathbf{y}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{L}}(\mathbf{v}, \mathbf{y}) \\ [\mathbf{R}_y^{\mathcal{L}}(\mathbf{v}, \mathbf{y})]^T \cdot \xi \\ \xi^T \cdot \xi - 1 \end{bmatrix} = \mathbf{0}, \quad (4.18)$$

où $\mathbf{y} = [\mathbf{u}^T, \mathbf{w}^T]^T$ et ξ est un vecteur unitaire.

La formulation de ces systèmes nécessite l'évaluation des matrices jacobiennes $\mathbf{R}_z^{\mathcal{L}}$ et $\mathbf{R}_y^{\mathcal{L}}$. Par ailleurs la classification exploite les relations (4.9)-(4.14) dont la formulation nécessite

l'évaluation des matrices jacobiennes $\mathbf{R}_z^{\mathcal{C}}$ et $\mathbf{R}_y^{\mathcal{C}}$, $\mathbf{R}_w^{\mathcal{C}}$ et $\mathbf{R}_q^{\mathcal{C}}$. Quatre matrices jacobiennes doivent donc être évaluées, et ce à la fois lors de la détection et de l'identification.

La méthode proposée pour la détermination des singularités repose finalement sur le processus suivant [Hentz *et al.*, 2016b] :

1. Détection des singularités de \mathcal{S} et détection spécifique des singularités de $\mathcal{S}_z \cap \mathcal{S}_y$ (Section 4.2.2) :
 - (a) Exploitation de la différentiation automatique pour construire la jacobienne $\mathbf{R}_q^{\mathcal{C}}$, et ses sous-matrices $\mathbf{R}_z^{\mathcal{C}}$, $\mathbf{R}_y^{\mathcal{C}}$ et $\mathbf{R}_w^{\mathcal{C}}$ par différentiation du modèle géométrique implicite (1.4) $\mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \mathbf{0}$ par rapport aux coordonnées géométriques \mathbf{q} (Section 4.2.2.1) ;
 - (b) Formulation des systèmes étendus (4.17) et (4.18) par assemblage générique des éléments obtenus en 1.(a) (Section 4.2.2.1) ;
 - (c) Détection des configurations singulières \mathcal{S} par résolution des systèmes étendus (4.17) et (4.18) avec une méthode de continuation de haut-degré ;
 - (d) Surveillance de la perte de rang de \mathbf{R}_y pendant la continuation sur le système (4.17) et de \mathbf{R}_z pendant la continuation sur le système (4.18). Ainsi les singularités de $\mathcal{S}_z \cap \mathcal{S}_y$ sont détectées avec précision (Section 4.2.2.2) ;
2. Classification en singularités de sortie ou d'entrée des branches de solutions soit du système (4.17), ou du système (4.18) où aucun point de $\mathcal{S}_z \cap \mathcal{S}_y$ n'est détecté (Section 4.2.3.1) :
3. Classification de chaque singularité $\mathbf{q} \in \mathcal{S}_z \cap \mathcal{S}_y$ (Section 4.2.3.2) :
 - (a) Évaluation numérique des rangs des matrices $\mathbf{R}_z^{\mathcal{C}}$, $\mathbf{R}_y^{\mathcal{C}}$, $\mathbf{R}_w^{\mathcal{C}}$ et $\mathbf{R}_q^{\mathcal{C}}$. Les valeurs propres des matrices carrées \mathbf{R}_z et \mathbf{R}_y et les valeurs singulières des matrices rectangulaires \mathbf{R}_w et \mathbf{R}_q sont considérées nulles si leur valeur est inférieure à un seuil ε_{rang} ;
 - (b) Identification des types de singularités par test des conditions de perte de rang (4.9)-(4.14) ;
 - (c) Classification automatisée en testant successivement l'appartenance à (a) des combinaisons de types RI, RO, RPM correspondant à l'assignation à une ligne du tableau 4.1, (b) le type IIM, (c) le type IO et (d) le type II correspondants à l'assignation d'un numéro de classe du tableau 4.1.

Ces étapes du processus de détermination sont détaillées dans les sections 4.2.2 et 4.2.3.

4.2.2 Détection de l'ensemble des singularités

4.2.2.1 Construction des systèmes étendus

Nous choisissons d'établir les équations du modèle cinématique implicite (4.1) par différentiation du modèle géométrique implicite (1.4)

$$\mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \mathbf{0}$$

par rapport au temps t [Andreff, 2006 ; Bohigas *et al.*, 2012b] :

$$\frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial t} = \mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) \cdot \dot{\mathbf{q}} = \mathbf{0},$$

où $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) = \frac{\partial \mathbf{R}^{\mathcal{C}}(\mathbf{q})}{\partial \mathbf{q}}$ et $\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt}$ contient les dérivées temporelles des coordonnées géométriques.

Pour la détermination des frontières de l'espace de travail présentée en section 3.2.1, seule la fonction globale $\mathbf{R}^{\mathcal{S}_z}(\mathbf{q}, \xi)$ était évaluée. L'approche correspondant au processus (3.7) était ainsi adoptée. Ce processus possède l'avantage d'éviter une étape d'assemblage de la fonction $\mathbf{R}^{\mathcal{S}_z}(\mathbf{q}, \xi)$. Pour l'analyse des singularités il est nécessaire de calculer les 4 jacobiennes $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}$ et $\mathbf{R}_{\mathbf{y}}^{\mathcal{C}}$, $\mathbf{R}_{\mathbf{w}}^{\mathcal{C}}$ et $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$, et le processus (3.7) est donc moins adapté. Les matrices $\mathbf{R}_{\mathbf{w}}^{\mathcal{C}}$, $\mathbf{R}_{\mathbf{y}}^{\mathcal{C}} = [\mathbf{R}_{\mathbf{u}}^{\mathcal{C}}, \mathbf{R}_{\mathbf{w}}^{\mathcal{C}}]$ et $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}} = [\mathbf{R}_{\mathbf{v}}^{\mathcal{C}}, \mathbf{R}_{\mathbf{w}}^{\mathcal{C}}]$ correspondent à des sous-matrices issues de $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}} = [\mathbf{R}_{\mathbf{u}}^{\mathcal{C}}, \mathbf{R}_{\mathbf{v}}^{\mathcal{C}}, \mathbf{R}_{\mathbf{w}}^{\mathcal{C}}]$. L'évaluation des dérivées contenues dans $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$ permet donc l'évaluation de toutes les matrices nécessaires. Nous proposons donc le processus suivant [Hentz *et al.*, 2016b] :

$$\mathbf{R}^{\mathcal{C}}(\mathbf{q}) \xrightarrow[\text{par rapport à } \mathbf{q}]{\text{Différencier}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{q}) \\ \mathbf{R}_{\mathbf{q}}^{\mathcal{C}}(\mathbf{q}) \end{pmatrix} \xrightarrow[\text{fonctions test}]{\text{Assembler les}} \begin{pmatrix} \mathbf{R}^{\mathcal{C}}(\mathbf{q}) \\ [\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}(\mathbf{q})]^{\top} \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \\ \text{rang}(\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}) < \text{rang}(\mathbf{R}_{\mathbf{w}}^{\mathcal{C}}) + m \\ \text{rang}(\mathbf{R}_{\mathbf{y}}^{\mathcal{C}}) < \text{rang}(\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}) \\ \dots \end{pmatrix}, \quad (4.19)$$

où les matrices $\mathbf{R}_{\mathbf{z}}^{\mathcal{C}}$ et $\mathbf{R}_{\mathbf{y}}^{\mathcal{C}}$, $\mathbf{R}_{\mathbf{w}}^{\mathcal{C}}$ sont des sous-matrices de $\mathbf{R}_{\mathbf{q}}^{\mathcal{C}}$.

La différentiation du modèle géométrique implicite (1.4) par rapport aux coordonnées géométriques \mathbf{q} est effectuée par transformation de source. L'annexe C détaille la construction du code différentié en mode multi-directionnel tangent avec le logiciel Tapenade. L'exploitation du code différentié pour l'évaluation du modèle cinématique implicite et des différentes jacobiennes le composant est alors présentée pour le cas d'application introduit dans la suite.

Afin d'intégrer le code différentié de manière générique dans Diamanlab, des fonctions Matlab ont été développées qui permettent de :

1. définir les directions de différentiation nécessaires,
2. assembler la matrice $\mathbf{R}_q^{\mathcal{C}}$,
3. décomposer $\mathbf{R}_q^{\mathcal{C}}$ pour obtenir les sous-matrices $\mathbf{R}_z^{\mathcal{C}}$, $\mathbf{R}_y^{\mathcal{C}}$ et $\mathbf{R}_w^{\mathcal{C}}$,
4. assembler les systèmes étendus (4.17) ou (4.18) en fonction de la variable `extsys`,
5. assembler les fonctions tests (4.9)-(4.14)
6. réaliser les tests conditionnels pour classifier une singularité à partir des types identifiés.

Les 4 premiers points sont détaillés dans la fonction présentée en annexe D pour le cas d'application introduit dans la suite.

4.2.2.2 Résolution

Chacun des sous-ensembles \mathcal{S}_z et \mathcal{S}_y forme un ou plusieurs lieux de solutions continues. Les sous-ensembles de singularités inverses et directes d'un mécanisme spatial ont généralement une intersection $\mathcal{S}_z \cap \mathcal{S}_y$ non-nulle [Bonev, 2002 - Section 2.5.3 - p.68]. Nous proposons de déterminer l'ensemble des singularités $\mathcal{S}_z \cup \mathcal{S}_y$ comme un seul ensemble de solutions continues à partir d'un point initial représentant une singularité inverse ou directe. La continuation de haut-degré est exploitée pour résoudre successivement les systèmes étendus (4.17) et (4.18). Pour cela, la perte de rang de $\mathbf{R}_y^{\mathcal{C}}$ est surveillée pendant le parcours de \mathcal{S}_z . La perte de rang de $\mathbf{R}_z^{\mathcal{C}}$ pendant le parcours de \mathcal{S}_y est également surveillée. Si une perte de rang est détectée, la configuration correspondante est située dans l'intersection $\mathcal{S}_z \cap \mathcal{S}_y$. Le point correspondant peut être utilisé pour passer d'une branche de \mathcal{S}_z à une branche de \mathcal{S}_y , et inversement. A partir d'un point initial :

- la détection est réalisée dans l'ensemble des singularités inverses et directes $\mathcal{S}_z \cup \mathcal{S}_y$ connectées,
- la détection de singularités entrée-sortie $\mathcal{S}_z \cap \mathcal{S}_y$ est fait avec une précision qui doit permettre leur classification ultérieure en singularités de classes 3 à 21 (Voir tableau 4.1),
- l'absence de détection d'un point dans l'intersection $\mathcal{S}_z \cap \mathcal{S}_y$ permet d'identifier des singularités de sortie $\mathcal{S}_z \setminus \mathcal{S}_y$ (RI,IO) ou d'entrée $\mathcal{S}_y \setminus \mathcal{S}_z$ (RO,II).

La différentiation automatique, la précision apportée par la méthode Diamant, la surveillance de la perte de rang et le changement de branche automatique doivent permettre de parcourir l'ensemble de singularités $\mathcal{S}_z \cup \mathcal{S}_y$ connectées au point initial. L'ensemble de singularités est obtenu comme un jeu de branches de solutions \mathcal{S}_z connectées par des points de bifurcation de \mathcal{S}_z connecté à un jeu similaire de \mathcal{S}_y par des points d'intersection de $\mathcal{S}_z \cap \mathcal{S}_y$. Ces points d'intersections sont classifiés comme décrit en section 4.2.3.2.

4.2.3 Classification des configurations singulières

4.2.3.1 Branches de solutions

Supposons que la continuation soit réalisée sur une branche de solutions où seule la matrice $\mathbf{R}_z^{\mathcal{C}}$ perd un rang. La surveillance du signe des déterminants des matrices $\mathbf{R}_y^{\mathcal{C}}$ et $\mathbf{R}_z^{\mathcal{C}}$ pendant la continuation permet d'assurer que les matrices $\mathbf{R}_q^{\mathcal{C}}$, $\mathbf{R}_w^{\mathcal{C}}$, $\mathbf{R}_y^{\mathcal{C}}$ et $\mathbf{R}_z^{\mathcal{C}}$ ont un rang constant entre deux points d'intersection de \mathcal{S}_z et \mathcal{S}_y . Si le point initial est une singularité de sortie, la branche de solutions peut alors être classifiée en singularité de sortie. De même les branches de solutions parcourues à partir d'une singularité d'entrée sont constituées exclusivement de singularités d'entrée si aucune perte de rang supplémentaire n'est détectée.

Si le point initial est une singularité combinée, la branche ne peut pas être classifiée globalement.

4.2.3.2 Points de $\mathcal{S}_z \cap \mathcal{S}_y$

Les points détectés dans l'intersection de $\mathcal{S}_z \cap \mathcal{S}_y$ sont des singularités d'une des classes 3 à 21 listées dans le tableau 4.1. Une étape de classification est dès lors nécessaire. La précision de détection des points d'intersection $\mathbf{q} \in \mathcal{S}_z \cap \mathcal{S}_y$ permet d'envisager une évaluation numérique du rang des matrices jacobiniennes $\mathbf{R}_z^{\mathcal{C}}(\mathbf{q})$, $\mathbf{R}_y^{\mathcal{C}}(\mathbf{q})$, $\mathbf{R}_w^{\mathcal{C}}(\mathbf{q})$ et $\mathbf{R}_q^{\mathcal{C}}(\mathbf{q})$ exploitée pour la classification. Les valeurs propres des matrices carrées \mathbf{R}_z et \mathbf{R}_y et les valeurs singulières des matrices rectangulaires \mathbf{R}_w et \mathbf{R}_q sont considérées nulles si leur valeur est inférieure au seuil ε_{rang} . Nous proposons donc une classification numérique directe des singularités de $\mathcal{S}_z \cap \mathcal{S}_y$. Chaque point correspondant est testé en utilisant la fonction implémentant les conditions de perte de rang (4.9) à (4.14) afin d'identifier les différents types de singularité auquel il appartient. La dernière étape de d'identification de la classe de singularité consiste en une séquence de tests conditionnels permettant de classifier la singularité dans une ligne puis dans une colonne du tableau 4.1 qui définissent finalement sa classe.

4.3 Cas d'application

4.3.1 Description du mécanisme

Le mécanisme parallèle plan « double-chaîne » de mobilité 2 décrit en figure 4.5 est considéré afin de valider notre méthode. Le mécanisme est choisi car il représente un défi en termes de détermination des singularités, en particulier avec les paramètres géométriques proposés par Bohigas *et al.* [2012b], pour lesquels tous les types de singularités apparaissent.

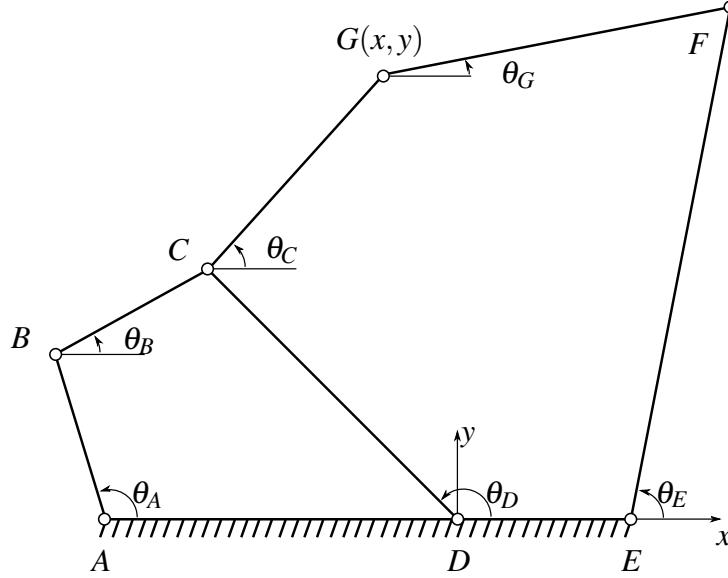


FIGURE 4.5 – Schéma cinématique du mécanisme double-chaîne.

La nature plane de l'architecture et de ses mouvements permet une interprétation du comportement cinématique dégénéré provoqué par chaque classe de singularité [Zlatanov *et al.*, 1998 ; Bohigas *et al.*, 2012b]. Il s'agit donc d'un cas intéressant pour évaluer notre méthode.

Le modèle géométrique implicite du mécanisme, adapté de Bohigas *et al.* [2012b], est obtenu par l'écriture des équations de fermeture vectorielle pour chacune des deux chaînes cinématiques fermées et deux équations qui contraignent la position du point G :

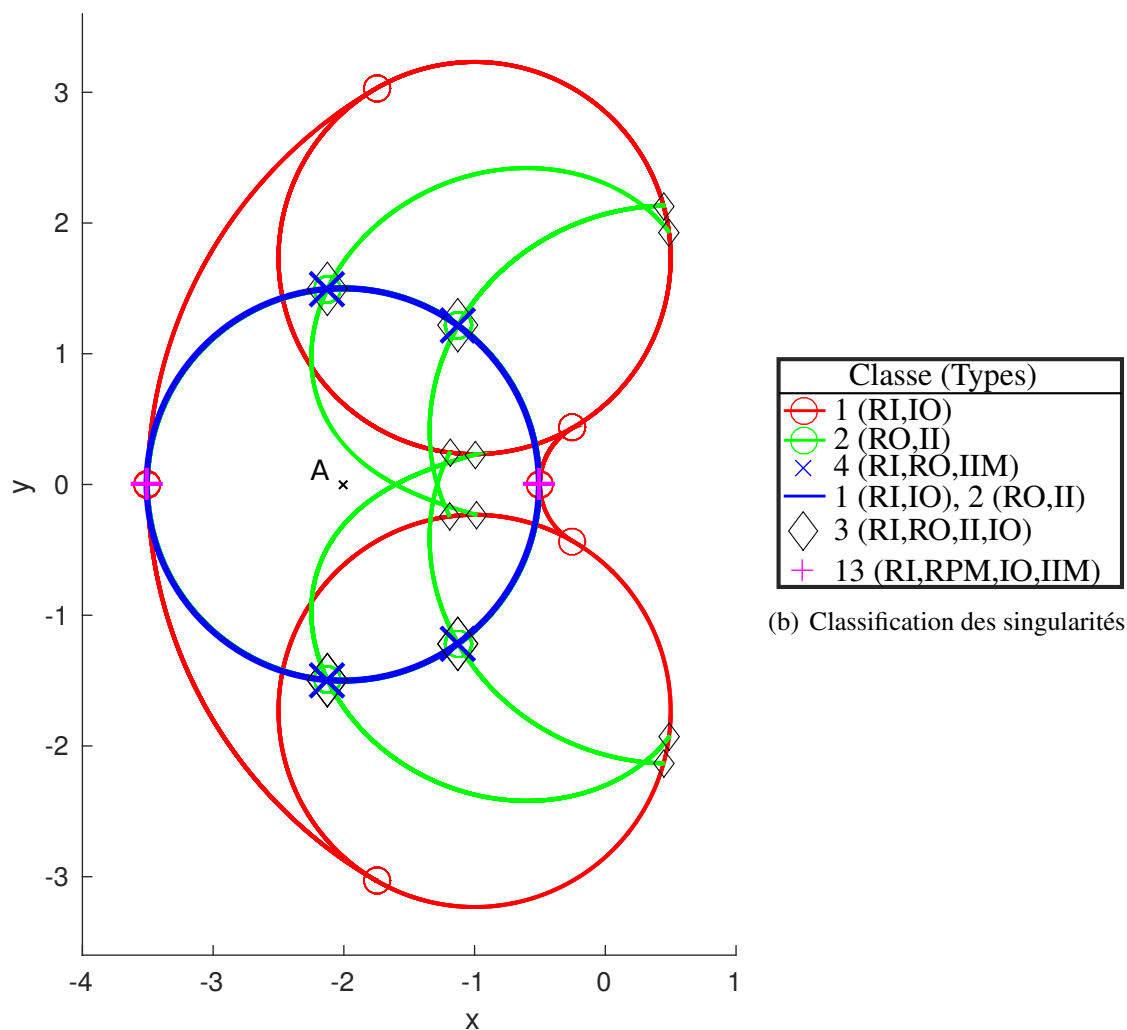
$$\mathbf{R}^{\mathcal{C}}(\mathbf{q}) = \begin{bmatrix} CD\cos(\theta_D) + CG\cos(\theta_C) - x \\ CD\sin(\theta_D) + CG\sin(\theta_C) - y \\ AB\cos(\theta_A) + BC\cos(\theta_B) - CD\cos(\theta_D) - AD \\ AB\sin(\theta_A) + BC\sin(\theta_B) - CD\sin(\theta_D) \\ CD\cos(\theta_D) + CG\cos(\theta_C) + FG\cos(\theta_G) - EF\cos(\theta_E) - DE \\ CD\sin(\theta_D) + CG\sin(\theta_C) + FG\sin(\theta_G) - EF\sin(\theta_E) \end{bmatrix} = \mathbf{0}, \quad (4.20)$$

où $\mathbf{v} = [\theta_A, \theta_E]^T$ sont les coordonnées actives, $\mathbf{u} = [x, y]^T$ sont les coordonnées opérationnelles. Les coordonnées des liaisons passives sont $\mathbf{w} = [\theta_B, \theta_C, \theta_D, \theta_G]^T$. Les paramètres géométriques sont $AB = BC = DE = 1$, $AD = CD = FG = 2$, $CG = 1.5$ et $EF = 3$.

4.3.2 Implémentation et résultats

L'implémentation est réalisée avec un ordre de troncature des polynômes de Taylor $K = 20$ et une précision de $\varepsilon = 1e - 6$ pour la méthode Diamant. Les valeurs propres des

matrices carrées $\mathbf{R}_z^{\mathcal{C}}$ et $\mathbf{R}_y^{\mathcal{C}}$ et les valeurs singulières des matrices rectangulaires $\mathbf{R}_w^{\mathcal{C}}$ et $\mathbf{R}_q^{\mathcal{C}}$ sont considérées nulles si leur valeur est inférieure au seuil $\varepsilon_{rang} = 1e - 5$. La fonction Matlab utilisée pour la détermination des singularités du mécanisme avec Diamanlab est présentée en annexe D. Les singularités de $\mathcal{S}_z \cup \mathcal{S}_y$ sont déterminées dans l'espace des variables $(x, y, \theta_A, \theta_E, \theta_B, \theta_C, \theta_D, \theta_G, \xi)$, où ξ est le vecteur introduit dans les systèmes étendus (4.17) et (4.18). Elles sont projetées dans l'espace des coordonnées opérationnelles (x, y) et représentées en figure 4.6(a).



(b) Classification des singularités

(a) Ensemble des singularités $\mathbf{q} \in \mathcal{S}$ déterminées, projetées sur (x, y)

FIGURE 4.6 – (a) Ensemble des singularités de $\mathcal{S} = \mathcal{S}_z \cup \mathcal{S}_y$ du mécanisme double-chaîne décrit en figure 4.5, déterminées avec notre méthode, après projection sur l'espace des coordonnées opérationnelles (x, y) . Chaque singularité appartient à une des classes dont le numéro et les types correspondants (Voir tableau 4.1) sont indiqués dans (b).

Les points de bifurcation sont automatiquement détectés durant le processus de continuation et indiqués avec un marqueur circulaire (\circ). Ils représentent des intersections de branches rencontrées lors du parcours soit de \mathcal{S}_z ou de \mathcal{S}_y . Les points dans l'intersection $\mathcal{S}_z \cap \mathcal{S}_y$, détectés par la surveillance de la perte de rang des matrices $\mathbf{R}_z^{\mathcal{C}}$ et $\mathbf{R}_y^{\mathcal{C}}$, sont indiqués par les autres marqueurs. La détection des points de $\mathcal{S}_z \cap \mathcal{S}_y$ et des points de bifurcation permet de calculer les branches de solutions de $\mathcal{S}_z \cup \mathcal{S}_y$ à partir d'un point initial. Les singularités de sortie (RI,IO) et d'entrée (RO,II) sont indiquées respectivement en rouge et en vert.

Toutes les singularités de $\mathcal{S}_z \cap \mathcal{S}_y$ sont ensuite classifiées en utilisant la seconde étape de la méthode. Les marqueurs \times désignent des singularités de classe 4 (RI,RO,IIM). Les points indiqués par un marqueur \diamond correspondent à des singularités combinées de classe 3 (RI,IO,RO,II). Deux points de classe 13 (RI,RPM,IO,IIM) sont indiqués par un marqueur $+$. Certains points sur le diagramme de bifurcation sont identifiés par plusieurs classes. Cela signifie qu'ils correspondent à plusieurs configurations de l'espace des configurations \mathcal{C} projetées dans (x,y) , car une configuration ne peut appartenir qu'à une unique classe [Zlatanov *et al.*, 1995]. Le cercle bleu de centre A et de rayon CG est en particulier une superposition de singularités de classes (RI,IO) et de singularités de classe (RO,II). Toutes les singularités précédemment déterminées [Bohigas *et al.*, 2012b] sont obtenues, à l'exception de singularités de dimension accrue qui correspondent au cercle bleu.

De plus, notre méthode permet la classification des singularités de classe 3 (RI,IO,RO,II), 4 (RI,RO,IIM) et 13 (RI,RPM,IO,IIM) qui ne peuvent être classifiées automatiquement par la méthode de Bohigas *et al.* [2012b].

La présence de singularités de type IIM implique un gain de mobilité. Une investigation supplémentaire est nécessaire. Une analyse approfondie des singularités IIM déterminées montre que ces points correspondent à des configurations où les liaisons en A et en C sont confondus.

Un découplage du mouvement des deux chaînes cinématiques fermées composant le mécanisme initial a lieu. On peut dès lors considérer que chacune des 2 chaînes cinématiques fermées est un mécanisme, dit équivalent, d'une mobilité chacun. Considérons d'abord la chaîne ABCD. La liaison actionnée au point A permet d'orienter les segments AB et BC alors que les segments AD et CD sont fixes avec un angle $\theta_D = \pi$. La chaîne CDEFG considérée avec un angle $\theta_D = \pi$ peut être assimilée à un mécanisme 4 barres CDEFG. La liaison actionnée au point E permet de positionner le point G sur un cercle de centre A et de rayon CG.

En conséquence, pour une valeur de (x,y) donnée sur le cercle bleu, θ_A peut prendre n'importe quelle valeur dans $[-\pi, \pi]$.

La jambe ABC est en une configuration repliée qui correspond à une singularité de sortie (RI,IO). Ainsi, la classe (RI,IO) indiquée par un cercle dans l'espace (x, y) est une projection d'un cylindre dans l'espace (x, y, θ_A) . Le même raisonnement peut être appliqué pour conclure que les points de la classe 3 (RI,IO,RO,II) sur le cercle sont en fait des lignes dans l'espace (x, y, θ_A) . Notons que seules les configurations à l'intersection entre le cylindre de classe (RI,IO) et les singularités de classe (RO,II) ($\theta_A \in \{-\pi, 0, \pi\}$) correspondent à des singularités de classe 4 (RI,RO,IIM).

L'ensemble des singularités gagne donc une dimension sur le cercle bleu. Les singularités présentes dans cette dimension sont de classe (RI,IO) et (RI,IO,RO,II) qui sont solutions du système étendu (4.17).

La détermination des configurations singulières du mécanisme double-chaîne pour lesquelles les points A et C sont confondus peut être réalisée en fixant θ_A . La chaîne ABCD étant ainsi immobilisée, on cherche à déterminer les configurations admissibles par le mécanisme 4 barres équivalent CEF. Si l'on fixe θ_A pour la résolution du système (4.20), on obtient un système de 6 équations à 7 inconnues qui exprime les configurations admissibles par le mécanisme 4 barres équivalent CEF et peut être résolu par continuation. Les configurations obtenues pour un ensemble de valeurs $\theta_A \in [-\pi, \pi]$ devront encore être classifiées.

4.4 Conclusion

Dans ce chapitre, nous avons proposé l'exploitation de la continuation de haut-degré pour la détection et la classification de singularités. La précision des polynômes de Taylor calculés, combinée à la différentiation automatique, permet la classification directe des singularités en utilisant les conditions de perte de rang. La détermination des singularités est effectuée à partir du modèle géométrique implicite du mécanisme qui peut être obtenu facilement par application des équations de fermeture des chaînes cinématiques.

L'évaluation de la méthode montre des résultats prometteurs : la plupart des singularités sont détectées et toutes les singularités détectées sont classifiées sans ambiguïté. Seuls deux ensembles de singularités de dimension accrue ne sont pas détectés avec l'implémentation actuelle. Leur existence peut être déduite de la présence de singularités IIM qui sont bien déterminées. Leur détection requiert la mise en place d'une stratégie de parcours par continuation telle que la discrétisation ou le parcours d'autres systèmes étendus.

L'outil proposé peut être complété de trois manières. Une stratégie pour la sélection automatique de la valeur du seuil ϵ_{rang} en dessous de laquelle les valeurs propres et singulières des matrices sont considérées nulles doit être développée afin de garantir la classification des singularités de $\mathcal{S}_z \cap \mathcal{S}_y$.

Concernant les configurations de type IIM déterminées : elles peuvent être spécifiquement étudiées afin de déterminer le comportement cinématique y apparaissant. Si elles engendrent un sous-ensemble de dimension accrue, celui-ci pourrait être détecté par la résolution du système étendu (4.8) exprimant la perte de rang de la jacobienne $\mathbf{R}_q^{\mathcal{L}}$:

$$\mathbf{R}^{\{IIM\}}(\mathbf{q}, \xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{L}}(\mathbf{q}) \\ [\mathbf{R}_q^{\mathcal{L}}(\mathbf{q})]^T \cdot \xi \\ \xi^T \cdot \xi - 1 \end{bmatrix} = \mathbf{0} .$$

Le sous-ensemble détecté, qui appartient à $\mathcal{S}_z \cap \mathcal{S}_y$, restera à classifier.

Concernant la classification d'une branche de solutions de $\mathcal{S}_z \cap \mathcal{S}_y$: celle-ci peut être conduite par des tests le long de la branches. On peut également envisager de résoudre les systèmes étendus dont les solutions sont les singularités d'un type donné, ce qui correspondrait alors au principe décrit en section 4.1.3.2 de Bohigas [2013 - Section 2.3].

Chapitre 5

Analyse de sensibilité en conception de mécanismes

« Parallel mechanisms are highly sensitive to dimensioning : a classical example is that by changing the radius of the platform of Stewart-Gough platform by 10% we may change the minimal stiffness of the robot over its workspace by 700% »,

Jean-Pierre Merlet.

Sommaire

5.1	Etat de l'art	122
5.1.1	Méthodes d'analyse de sensibilité en conception de mécanismes	123
5.1.2	Analyse de sensibilité par dérivation de haut-degré	125
5.2	Différentiation de haut-degré	126
5.2.1	Présentation du cas d'étude	127
5.2.2	Modélisation par des corps pseudo-rigides	128
5.2.3	Problème d'analyse de sensibilité	132
5.2.4	Exploitation de la DA de haut-degré	136
5.3	Conclusion	141

Le concepteur s'intéresse en premier lieu aux caractéristiques nominales d'un mécanisme. Les caractéristiques sont cependant susceptibles de varier après fabrication et intégration du mécanisme. La précision de positionnement de l'effecteur, l'amplitude de ses mouvements admissibles, ou le lieu des singularités par exemple peuvent ne plus satisfaire aux spécifications visées.

Assurer que les performances restent acceptables après fabrication et intégration d'un mécanisme est bien sûr d'importance. Afin d'avancer dans cette direction, nous proposons dans ce chapitre une démarche d'évaluation de la sensibilité des performances d'un mécanisme aux incertitudes de fabrication et d'intégration.

En section 5.1, notre approche pour prendre en compte l'impact des erreurs en phase de conception préliminaire est introduite. Les limites des principales méthodes exploitées pour analyser la sensibilité des caractéristiques d'un mécanisme sont alors exposées. Des méthodes d'analyse de sensibilité issues de la mécanique des structures et reposant sur le calcul de dérivées de haut-degré au moyen de la différentiation automatique sont ensuite présentées.

L'intérêt d'une analyse de sensibilité de haut-degré est mis en avant en section 5.2 en analysant le comportement d'un mécanisme amplificateur pour actionneur piézoélectrique.

L'exploitation possible de l'outil et ses limites sont discutées en section 5.3, et des perspectives de développement finalement proposées.

5.1 Etat de l'art

Nous considérons ici la présence d'erreurs de fabrication et d'assemblage des membres et de la base du mécanisme, qui sont les sources d'erreur les plus étudiées [Di Gregorio et Parenti-Castelli, 1999 ; Wu et Rao, 2004 ; Caro *et al.*, 2005 ; Binaud *et al.*, 2010b ; Qu *et al.*, 2012 ; Tannous *et al.*, 2014 ; Corradi *et al.*, 2014 ; Chen *et al.*, 2014 ; Simas et Di Gregorio, 2016 ; Pickard *et al.*, 2016]. Ces erreurs ont un impact sur plusieurs caractéristiques d'un mécanisme. Ainsi a été étudié leur impact sur la précision de positionnement de l'effecteur, sur le nombre et sur la nature de mouvements parasites non contrôlés, les indices de manipulabilité, ou encore sur la localisation des singularités [Di Gregorio et Parenti-Castelli, 1999 ; Gregorio et Parenti-Castelli, 1999 ; Voglewede et Ebert-Uphoff, 2004 ; Caro *et al.*, 2005 ; Merlet, 2005a ; Binaud *et al.*, 2010a,b ; Cardou *et al.*, 2010 ; Qu *et al.*, 2012 ; Chen *et al.*, 2013, 2014 ; Corradi *et al.*, 2014 ; Rezaei et Akbarzadeh, 2015 ; Simas et Di Gregorio, 2016].

Durant la conception d'un mécanisme, on étudie en premier lieu le modèle géométrique implicite du mécanisme $\mathbf{R}(\mathbf{q}) = \mathbf{0}$. Les erreurs de fabrication et d'assemblage du mécanisme influent sur la valeur réelle des *paramètres géométriques* \mathbf{p} . Leur influence sur le modèle géométrique peut être étudiée en considérant les paramètres géométriques comme des variables du modèle géométrique implicite [Caro, 2004] :

$$\mathbf{R}(\mathbf{q}, \mathbf{p}) = \mathbf{0} \quad (5.1)$$

Saltelli *et al.* [2008] définissent l'*analyse de sensibilité* comme l'étude de la manière dont des incertitudes sur les entrées d'un modèle contribuent aux incertitudes sur les sorties d'un modèle. Nous nous intéressons dans la suite à l'analyse de sensibilité du comportement d'un mécanisme aux variations de ses paramètres géométriques.

5.1.1 Méthodes d'analyse de sensibilité en conception de mécanismes

Différentes approches ont été proposées en robotique pour conduire une analyse de sensibilité des coordonnées opérationnelles aux paramètres géométriques. La méthode la plus simple et la plus utilisée pour l'analyse de sensibilité consiste simplement à résoudre le modèle pour différentes valeurs du paramètre étudié. Les solutions sont généralement représentées dans des abaques. Aucune information n'est cependant obtenue entre deux points solutions. Cette méthode est par ailleurs coûteuse lorsque le nombre de paramètres augmente.

5.1.1.1 Méthode par linéarisation

La méthode par linéarisation est une autre méthode exploitée. Di Gregorio et Parenti-Castelli [1999] et Caro *et al.* [2005, 2009] ont étudié la sensibilité de mécanismes parallèles aux erreurs géométriques au moyen d'une linéarisation. Nous présentons ici le principe de cette méthode aussi appelée analyse de sensibilité d'ordre 1 [Haug et Ehle, 1982 ; Hou *et al.*, 1998].

Pour tout mécanisme, on peut disposer du modèle géométrique implicite. Celui-ci lie la pose de l'effecteur, la configuration des liaisons et les paramètres géométriques :

$$\mathbf{R}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}, \mathbf{p}) = \mathbf{0}.$$

Supposons une erreur nulle sur l'état des liaisons ($\delta \mathbf{z} = \mathbf{0}$). Pour conduire une analyse de sensibilité d'ordre 1 de la pose de l'effecteur aux erreurs géométriques, le modèle est linéarisé par différentiation par rapport aux coordonnées opérationnelles \mathbf{u} et aux paramètres géométriques \mathbf{p} :

$$\mathbf{R}_{\mathbf{u}}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}, \mathbf{p}) \delta \mathbf{u} + \mathbf{R}_{\mathbf{p}}^{\mathcal{C}}(\mathbf{u}, \mathbf{z}, \mathbf{p}) \delta \mathbf{p} = \mathbf{0}, \quad (5.2)$$

où l'erreur de pose de l'effecteur $\delta \mathbf{u}$ est la perturbation de sortie, et l'erreur sur les paramètres géométriques $\delta \mathbf{p}$ est la perturbation des paramètres. Avec $\mathbf{R}_{\mathbf{u}}^{\mathcal{C}}$ de rang plein, l'erreur de pose de l'effecteur satisfait :

$$\delta \mathbf{u} = -(\mathbf{R}_{\mathbf{u}}^{\mathcal{C}})^{-1} \mathbf{R}_{\mathbf{p}}^{\mathcal{C}} \delta \mathbf{p},$$

où $-(\mathbf{R}_u^{\mathcal{C}})^{-1}\mathbf{R}_p^{\mathcal{C}}$ est la matrice de sensibilité de la pose de l'effecteur aux erreurs géométriques.

La méthode est simple à mettre en oeuvre. En une configuration, la matrice de sensibilité $-(\mathbf{R}_u^{\mathcal{C}})^{-1}\mathbf{R}_p^{\mathcal{C}}$ représente la contribution relative des différents paramètres géométriques sur l'erreur de pose. Néanmoins, le calcul des matrices peut être fastidieux. La méthode n'est pas applicable à proximité des singularités directes où la matrice $\mathbf{R}_u^{\mathcal{C}}$ est mal conditionnée. Du fait de l'utilisation d'un modèle linéaire, seules de petites perturbations des paramètres peuvent par ailleurs être prises en compte.

5.1.1.2 Méthode de linéarisation d'intervalles

Le principe général des méthodes d'analyse par intervalles a été présenté en section 3.1.3. Oetomo *et al.* [2008] étudient par ce biais la sensibilité globale de l'espace de travail aux erreurs sur les paramètres géométriques. La méthode branch-and-bound est alors exploitée pour la détermination de l'ensemble des poses accessibles pour des intervalles de tolérance donnés sur les paramètres géométriques. La méthode ne permet pas d'évaluer avec précision la variation de la pose de l'effecteur pour une variation du paramètre mais garantit l'obtention d'une représentation conservatrice de l'espace de travail.

Dans une approche de sensibilité locale, Tannous *et al.* [2014] proposent de combiner la méthode par linéarisation avec une méthode d'analyse par intervalles pour étudier la sensibilité de la pose de l'effecteur à des erreurs de fabrication, des jeux dans les liaisons ou à des erreurs d'actionnement. La méthode repose sur la linéarisation du modèle géométrique implicite du mécanisme. Le système linéaire est résolu par une méthode d'analyse par intervalles [Krawczyk, 1969 ; Moore, 1979]. Pour une configuration du mécanisme et pour des intervalles de tolérance sur les paramètres donnés, la méthode permet de calculer des intervalles encadrant les coordonnées opérationnelles perturbées admissibles qui sont conservatifs [Krawczyk, 1969 ; Neumaier, 1990].

L'aspect conservatif des résultats est particulièrement intéressant lorsqu'on s'intéresse aux configurations voisines des singularités directes. On a en effet mentionné qu'à proximité de ces singularités, la méthode de linéarisation ne peut être employée. La méthode permet d'envisager une étude à proximité des singularités. Les intervalles calculés par la méthode qui repose sur une linéarisation sont exacts mais leur largeur peut être grande si la dépendance est non-linéaire, notamment à proximité des singularités.

5.1.2 Analyse de sensibilité par dérivation de haut-degré

Comme nous l'avons souligné au chapitre 1, pouvoir évaluer l'impact d'erreurs de fabrication ou d'assemblage y compris à proximité de singularités nous semble d'importance. Au voisinage de singularités, l'étude du comportement non-linéaire peut être envisagé en utilisant les dérivées de haut-degré, ce qui a été considéré hors du domaine de la robotique.

Des méthodes d'analyse de sensibilité exploitant les dérivées de haut-degré ont en effet été proposées dans de nombreux domaines de la physique [Barrio, 2006]. La sensibilité est alors définie comme la dérivée d'un indice de performance par rapport à une variable de conception. Une revue des approches pour l'analyse de sensibilité de systèmes d'équations linéaires réalisée par Van Keulen *et al.* [2005] cite des travaux exploitant les dérivées de haut-degré.

La DA pour l'analyse de sensibilité a été utilisée dans le domaine de la mécanique des structures. La DA par transformation de source et par surcharge d'opérateur a été notamment exploitée. Hou *et al.* [1998] ont exploité l'outil de DA par transformation de source ADIFOR pour des analyses de sensibilité d'ordre 1 et 2 de structures modélisées par éléments finis. Ozaki *et al.* [1995] ont par ailleurs proposé l'analyse de sensibilité de modèles éléments finis au moyen de la DA. Les dérivées partielles mixtes par rapport à un nombre élevé de variables sont alors évaluées jusqu'à l'ordre 3 avec le programme DAFOR de transformation de codes sources Fortran [Berz, 1987]. La sensibilité des contraintes et des déformations dans la structure mécanique d'un pont en treillis par rapport au rayon des éléments sont évaluées. L'apport du calcul de polynômes de Taylor est significatif lorsque la dépendance aux paramètres de conception est non-linéaire et lorsque les perturbations appliquées aux structures mécaniques sont grandes.

Kolchuzhin *et al.* [2007] proposent l'exploitation de la DA pour analyser la sensibilité de la réponse d'un système micro-électro-mécanique (MEMS) notamment à ses dimensions. Les équations modélisant le système sont discrétisées par une méthode d'éléments finis puis différenciées à haut-degré avec la bibliothèque de surcharge d'opérateurs FFADLib [Tsukanov et Hall, 2003]. Un exemple traité concerne la sensibilité de l'énergie de contrainte à la variation d'épaisseur d'une poutre en double encastrement, comme illustré en figure 5.1. La sensibilité est établie sous forme de polynômes de Taylor développés jusqu'à l'ordre 5 en le paramètre d'épaisseur. La méthode d'analyse de sensibilité permet de réaliser une abaque d'un coût moindre qu'une méthode d'échantillonnage où une résolution par la méthode des éléments finis est nécessaire pour chaque valeur discrète du paramètre. Les solutions à l'ordre 5 permettent alors d'observer la dépendance non-linéaire à l'épaisseur sur un domaine important comme le montre la figure 5.1. Cette approche a été utilisée pour l'optimisation de forme par Guillaume et Masmoudi [1994].

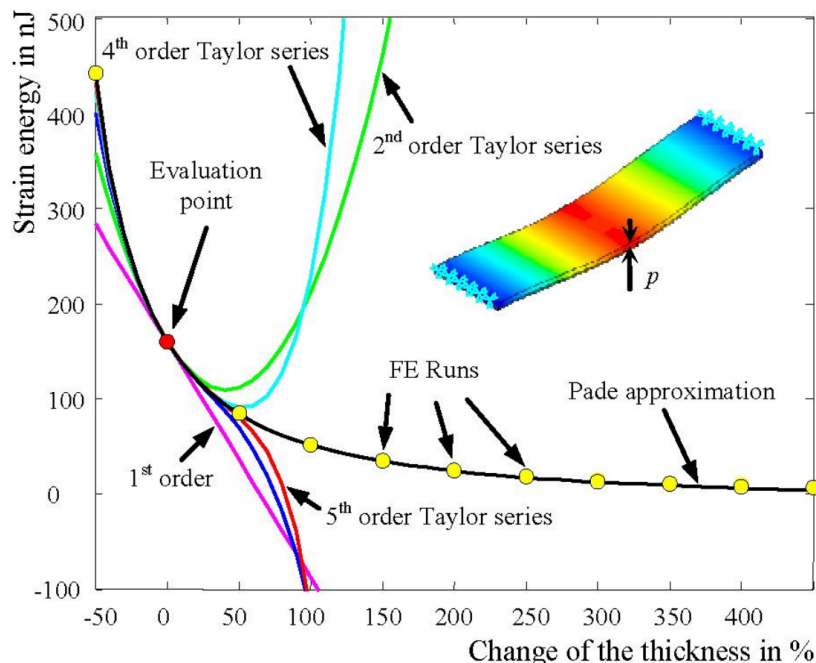


FIGURE 5.1 – Sensibilité de haut-degré de l'énergie de contrainte d'une poutre encastree par rapport à son épaisseur [Issu de Kolchuzhin *et al.*, 2007]. La poutre est modélisée par éléments finis et la précision des approximations de haut-degré comparées à un calcul direct par éléments finis pour d'autres valeurs des paramètres.

Berz et Hoffstätter [1998] ont proposé l'approximation de fonctions par des polynômes de Taylor tronqués à l'ordre K et l'encadrement de la contribution des termes d'ordre supérieur à K par une méthode d'analyse par intervalles. L'approximation d'un système d'équations non-linéaires au moyen de cette approche est appelée modèle de Taylor. Le logiciel COSY INFINITY [Makino et Berz, 2006] implémente cette méthode qui est principalement exploitée pour la résolution d'équations différentielles algébriques et d'équations différentielles ordinaires en physique nucléaire.

5.2 Différentiation de haut-degré

Nous proposons d'appliquer l'analyse de sensibilité de haut-degré au moyen de la DA de haut-degré dans le cadre de l'analyse de mécanismes. Nous cherchons dans la suite à évaluer l'intérêt d'exploiter l'information de haut-degré dans le cas d'un mécanisme au voisinage de ses configurations singulières. Le cas d'étude sélectionné est un mécanisme d'amplification.

5.2.1 Présentation du cas d'étude

5.2.1.1 Les actionneurs piézoélectriques amplifiés

Les *actionneurs* piézoélectriques amplifiés tels que proposés par exemple par CEDRAT TECHNOLOGIES [CEDRAT, 2017] reposent sur la combinaison d'un actionneur à base de céramiques piézoélectriques multicouches empilées appelées *céramique* et d'un *mécanisme d'amplification*, voir figure 5.2(a). La céramique est alimentée par une source de tension continue. L'application d'une tension positive entraîne une déformation du matériau piézoélectrique qui génère un mouvement d'extension.

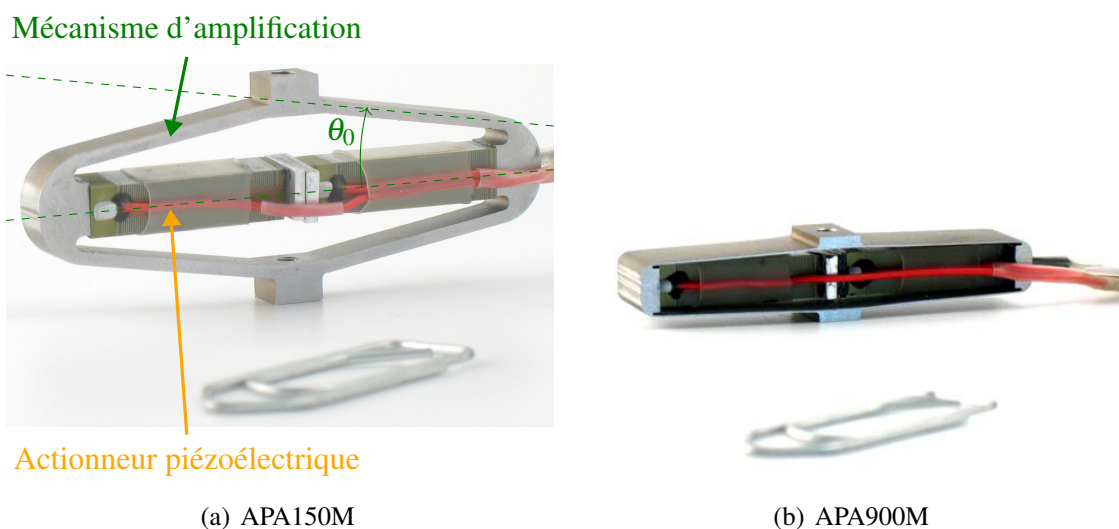


FIGURE 5.2 – Photographies de deux actionneurs piézoélectriques amplifiés de CEDRAT TECHNOLOGIES [CEDRAT, 2017]. La longueur de leur céramique est de l'ordre de $g = 50\text{mm}$, mais le déplacement autorisé par le modèle APA900M (a) est beaucoup plus grand que celui du modèle APA150M (b).

La fonction principale du mécanisme d'amplification est de transformer le mouvement d'extension de faible amplitude de la céramique en un mouvement amplifié de contraction du mécanisme selon un axe perpendiculaire, voir figure 5.3. Le mouvement d'extension de la céramique est le déplacement d'entrée du mécanisme et le mouvement amplifié de contraction le déplacement de sortie. Les actionneurs sont conçus pour réaliser de petits déplacements autour d'un point de fonctionnement. Le mécanisme compliant a également pour fonction la précharge de la céramique. Ceci permet un allongement de la durée de vie et des performances dynamiques supérieures à celles des actionneurs amplifiés par un mécanisme rigide ou une architecture sérielle. Enfin le rôle d'interface du mécanisme d'amplification limite les efforts exercés sur la céramique hors de son axe principal assurant ainsi sa protection.

5.2.1.2 Mécanisme compliant

L'amplification du déplacement d'entrée est donc réalisée par un mécanisme compliant. Le mécanisme est monolithique et la céramique y est assemblée avec une précharge dans le mécanisme. L'actionneur est ainsi également considéré comme une structure monolithique, ce qui implique l'absence de jeu dans le mécanisme.

Les mouvements du mécanisme sont obtenus à partir de déformations du matériau au niveau d'amincissements de matière. On considère que les parties de section les plus épaisses qui servent d'interface avec la céramique et avec les systèmes extérieurs sont rigides. Les parties de section mince, appelées lames, (voir figure 5.2(a)) se déforment sur toute leur longueur. Dès lors, à l'inverse des mécanismes rigides, la mobilité du mécanisme compliant n'est pas nécessairement localisée au niveau de liaisons mécaniques.

5.2.2 Modélisation par des corps pseudo-rigides

Si le mécanisme compliant possède de nombreux avantages, sa modélisation peut être rendue complexe du fait de la répartition spatiale des déformations. Un modèle linéarisé peut être considéré pour l'étude du mouvement du mécanisme si :

- les déplacements étudiés sont petits au regard des dimensions du mécanisme,
- les configurations considérées sont éloignées des singularités.

Or dans notre étude la proximité des configurations singulières est considérée. Des non-linéarités dues à la géométrie doivent donc être prises en compte.

Le modèle des corps pseudo-rigides permet d'approcher le comportement d'un mécanisme compliant au moyen d'un mécanisme rigide équivalent. L'approximation est acceptable sous les hypothèses et en suivant les règles de conversion détaillées dans Howell [2001]. Une telle approche est intéressante en phase de conception préliminaire pour l'estimation des caractéristiques cinématiques du système.

La modélisation par des corps rigides et la prise en compte des symétries du système permettent une modélisation des actionneurs CEDRAT par un mécanisme Scott-Russell, voir figure 5.3 et section 5.2.2.1. La longueur $AB = L_0$ du bras du mécanisme est la dimension caractéristique principale.

5.2.2.1 Modélisation géométrique

Dans la représentation simplifiée de la figure 5.3, le mécanisme plan Scott-Russell est constitué d'un membre AB relié au bâti aux points A et B par une liaison glissière via une liaison pivot. Les axes des deux glissières sont perpendiculaires. Nous nous intéressons au mouvement de la glissière située au point B qui est paramétré par la coordonnée de sortie y

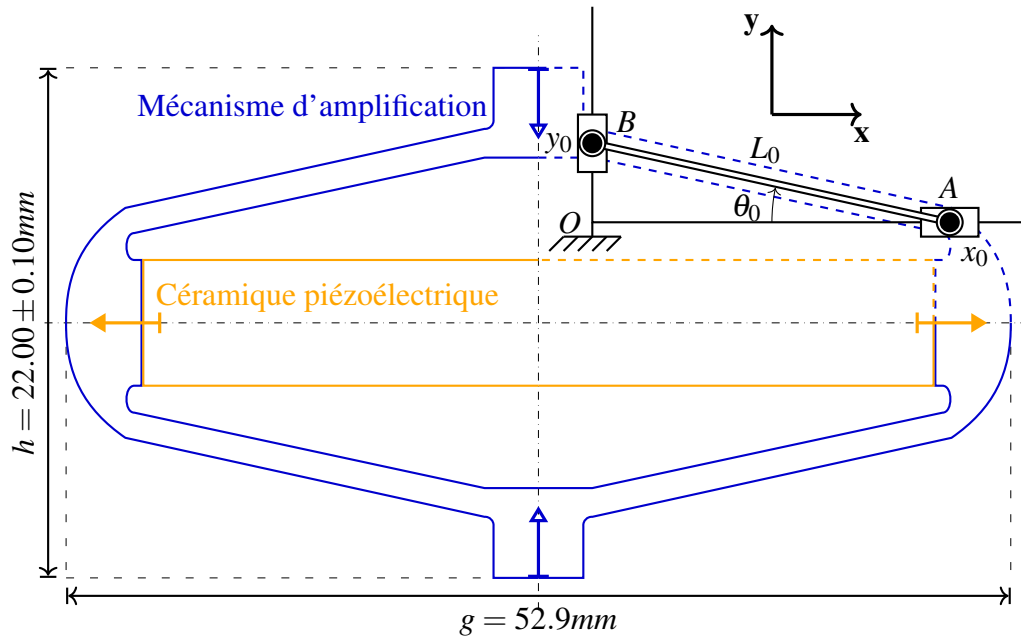


FIGURE 5.3 – Modélisation du comportement cinématique d'un amplificateur CEDRAT APA150M par un mécanisme Scott-Russell.

en fonction du mouvement de la glissière actionnée située au point A qui est paramétré par la coordonnée d'entrée x . L'angle entre le segment AB et l'axe x est noté θ .

L'étude du fonctionnement nominal du mécanisme revient donc à étudier les déformations du triangle OAB avec le paramètre géométrique $L = AB$ constant. Les relations entrée-sortie disponibles pour l'étude du mécanisme sont :

$$L^2 = x^2 + y^2, \quad (5.3)$$

avec le théorème de Pythagore.

Avec les relations trigonométriques, on écrit :

$$\cos(\theta) = x/L \quad (5.4)$$

$$\sin(\theta) = y/L, \quad (5.5)$$

et en combinant les 2 équations précédentes :

$$\tan(\theta) = y/x. \quad (5.6)$$

La géométrie du triangle rectangle OAB peut être décrite par plusieurs paramètres : la longueur de l'hypoténuse L , l'angle $\theta = \widehat{BAO}$, et les longueurs des segments adjacents x et opposé y .

Le jeu (x, y) ou l'angle θ sont suffisants pour décrire une configuration du mécanisme. Les mouvements du segment se font sous les contraintes exercées par les liaisons qui peuvent être exprimées par différents modèles. Le modèle géométrique direct s'exprime par :

$$\begin{aligned} \mathbf{R}^{MGD} : [x_0, x_0 + x_c] &\rightarrow [y_0 + y_c, y_0] \\ x &\mapsto y = \sqrt{L_0^2 - x^2} \end{aligned} \quad (5.7)$$

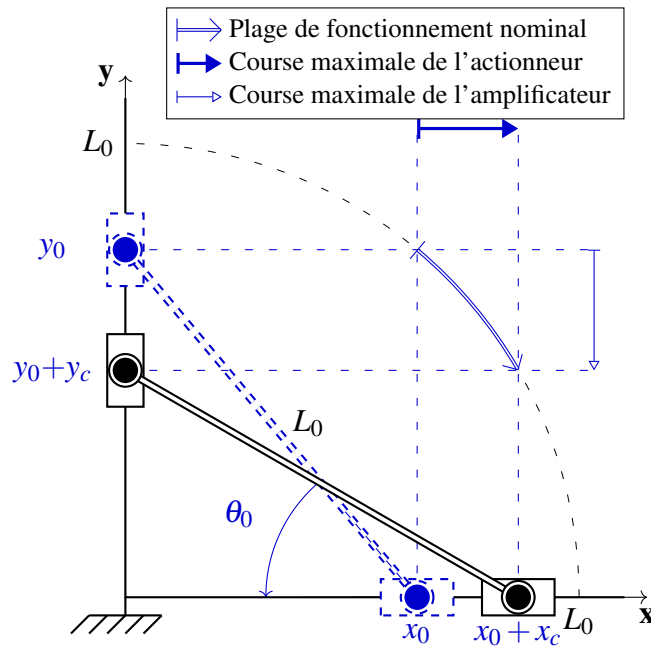
Remarquons que la fonction $\sqrt{L_0^2 - x^2}$ n'est dérivable que pour $x \in]-L_0, L_0[$. Ainsi pour $x = L_0$ le mécanisme est en configuration singulière et la dérivée du MGD n'est pas définie. Nous nous intéressons précisément à l'étude du mécanisme au voisinage de cette configuration singulière exploitée par les concepteurs pour son rapport d'amplification élevé.

5.2.2.2 Fonctionnement nominal

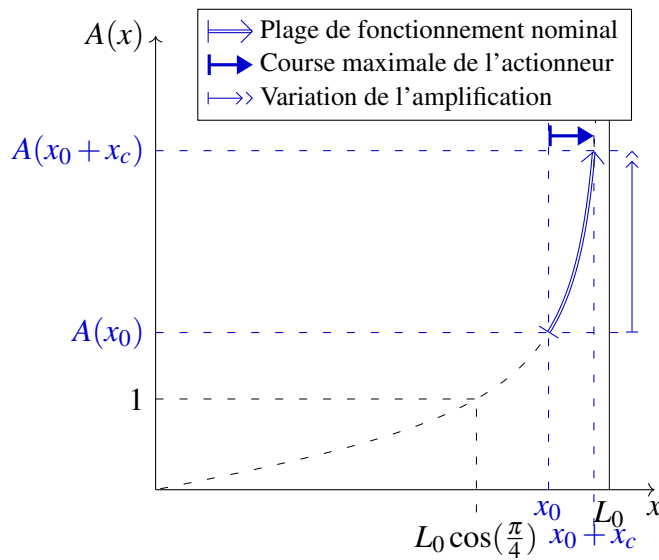
On appelle *point de fonctionnement* du mécanisme la donnée d'une valeur du *paramètre géométrique* L_0 et d'une *configuration initiale* du mécanisme (x_0, y_0, θ_0) telle que le mécanisme puisse être assemblé. Un point de fonctionnement décrit la géométrie d'un actionneur à vide et est représenté en traits pointillés bleus sur la figure 5.4(a). La course maximale $x_c > 0$ d'un actionneur piézoélectrique est limitée, ainsi les déplacements du mécanisme sont très réduits autour de la configuration initiale (x_0, y_0, θ_0) . Lorsque le mécanisme est actionné, la longueur x augmente jusqu'à une valeur maximale $x_0 + x_c$. On note y_c et θ_c les variations maximales de y et de θ correspondant à la course maximale de la céramique. Le mécanisme se situe alors en une nouvelle configuration $(x_0 + x_c, y_0 + y_c, \theta_0 + \theta_c)$ représentée en traits pleins noirs sur la figure 5.4(a). Finalement pour un actionneur donné, les coordonnées varient dans les intervalles $x \in [x_0, x_0 + x_c] \subset [0, L_0[$, $y \in [y_0 + y_c, y_0] \subset]0, L_0]$ et $\theta \in [\theta_0 + \theta_c, \theta_0] \subset]0, \frac{\pi}{2}]$ définissant la *plage de fonctionnement*.

5.2.2.3 Rapport d'amplification

La fonction principale du mécanisme est l'amplification du mouvement de la céramique. En une configuration donnée, le rapport entre l'amplitude des déplacements de sortie et d'entrée est appelé *rapport d'amplification*. Il peut être exprimé en fonction de la coordonnée



(a) Configurations du mécanisme



(b) Rapport d'amplification

FIGURE 5.4 – Représentation de la modification (a) de la configuration et (b) du rapport d'amplification du mécanisme pour la course maximale d'actionnement. A des fins de lisibilité, les configurations sont représentées avec une exagération des angles.

d'entrée x et obtenu par dérivation de la fonction (5.7) par rapport à x :

$$A^x : [x_0, x_0 + x_c] \rightarrow]-\infty, 0[\quad (5.8)$$

$$x \mapsto \frac{dy}{dx} = -\frac{x}{\sqrt{L^2 - x^2}}$$

En introduisant les équations (5.7) puis (5.6) dans l'équation (5.8), le rapport d'amplification peut aussi être exprimé en fonction de la coordonnée passive θ :

$$\begin{aligned} A^\theta : [\theta_0 + \theta_c, \theta_0] &\rightarrow]-\infty, 0[\\ \theta &\mapsto \frac{dy}{dx} = -\frac{1}{\tan(\theta)} \end{aligned} \quad (5.9)$$

Supposons que l'angle θ soit compris dans l'intervalle $]0, \frac{\pi}{4}[$. Le rapport d'amplification A^θ est négatif et il augmente en valeur absolue lorsque θ diminue. Afin de simplifier l'interprétation des résultats sur le rapport d'amplification, nous considérerons par la suite la valeur absolue $|A^x| = -A^x$ du rapport d'amplification, notée A . Plus l'angle $\theta < \frac{\pi}{4}$ est proche de 0, plus le rapport d'amplification $A > 1$ du mécanisme est grand. Le comportement de A en fonction de x est montrée en figure 5.4(b). La fonction $A(x)$ n'est pas définie en L_0 .

La course maximale d'un actionneur est déterminée par la course maximale de la céramique et par le rapport d'amplification du mécanisme. L'amplitude du mouvement d'entrée est proportionnelle à la longueur de la céramique. A encombrement donné, on ne peut donc pas modifier la longueur de la céramique pour obtenir l'amplitude désirée. C'est donc le rapport d'amplification que le concepteur doit faire varier. Suivant l'équation (5.9), celui-ci dépend de la configuration du mécanisme caractérisée par l'angle θ . La figure 5.2 montre les *APA150M* et *APA900M* qui sont les actionneurs dotés respectivement du plus petit et du plus grand rapport d'amplification de la gamme d'encombrement. Bien que les longueurs de leurs céramiques soient similaires, leurs déplacements libres maximaux sont de $187 \mu\text{m}$ et $802 \mu\text{m}$. Pour une longueur de la céramique et une course maximale d'actionnement très proches, c'est l'angle initial du mécanisme qui est modifié pour obtenir une course maximale de l'actionneur très variable.

5.2.3 Problème d'analyse de sensibilité

5.2.3.1 Erreur de fabrication sur le paramètre géométrique

Les erreurs de fabrication vont altérer le point de fonctionnement nominal $(x_0, y_0, \theta_0, L_0)$ visé pendant la conception. Il résulte des erreurs de fabrication une valeur réelle du paramètre \underline{L}_0 et les coordonnées réelles $(\underline{x}_0, \underline{y}_0, \underline{\theta}_0)$ définissent alors un nouveau point de fonctionnement, dit réel.

Nous considérons que la perturbation du mécanisme n'affecte pas les symétries du mécanisme, autrement dit l'angle \widehat{BOA} reste droit et l'axe des glissières n'est pas modifié. La figure 5.5 décrit les conséquences d'une telle perturbation sur la géométrie du mécanisme. Un trait continu jaune représente l'écart entre la valeur nominale (croix bleue) et la valeur

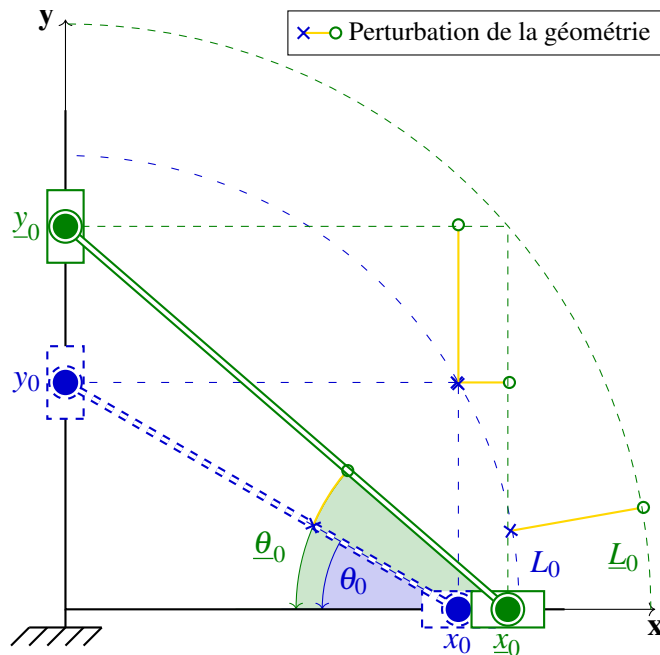


FIGURE 5.5 – Représentation de la modification du point de fonctionnement (configuration et paramètre géométrique) du mécanisme pour une perturbation arbitraire. A des fins de lisibilité, les configurations sont représentées avec une exagération des angles.

perturbée (rond vert) de chaque coordonnée du mécanisme et du paramètre géométrique après perturbation.

L'analyse de sensibilité consiste à relaxer le paramètre L_0 pour étudier l'influence d'une perturbation arbitraire de celui-ci sur le comportement du mécanisme. Supposons que la configuration réelle du mécanisme soit inconnue, autrement dit la relation entre θ_0 et $\underline{\theta}_0$ n'est pas définie. Le problème de sensibilité est indéterminé car \underline{L}_0 et $\underline{\theta}_0$ sont inconnus. En revanche si une des coordonnées réelles à l'état initial est connue, le problème peut être résolu.

5.2.3.2 Perturbation équivalente du paramètre géométrique

La propriété principale étudiée, le rapport d'amplification, ne dépend que de l'angle θ_0 , voir l'équation (5.9). Dès lors, pour une géométrie réelle d'angle initial $\underline{\theta}_0$, on peut étudier un mécanisme équivalent en terme de rapport d'amplification en appliquant une homothétie $\tilde{L}_0 = \frac{x_0}{\underline{x}_0} \underline{L}_0$, voir figure 5.6. Plutôt que d'appliquer la perturbation réelle \underline{L}_0 , on applique donc une perturbation équivalente \tilde{L}_0 . L'angle $\tilde{\theta}_0 = \underline{\theta}_0$ et donc le rapport d'amplification $\tilde{A} = \underline{A}$ sont conservés. Cela permet finalement une analyse de sensibilité avec une erreur nulle sur le

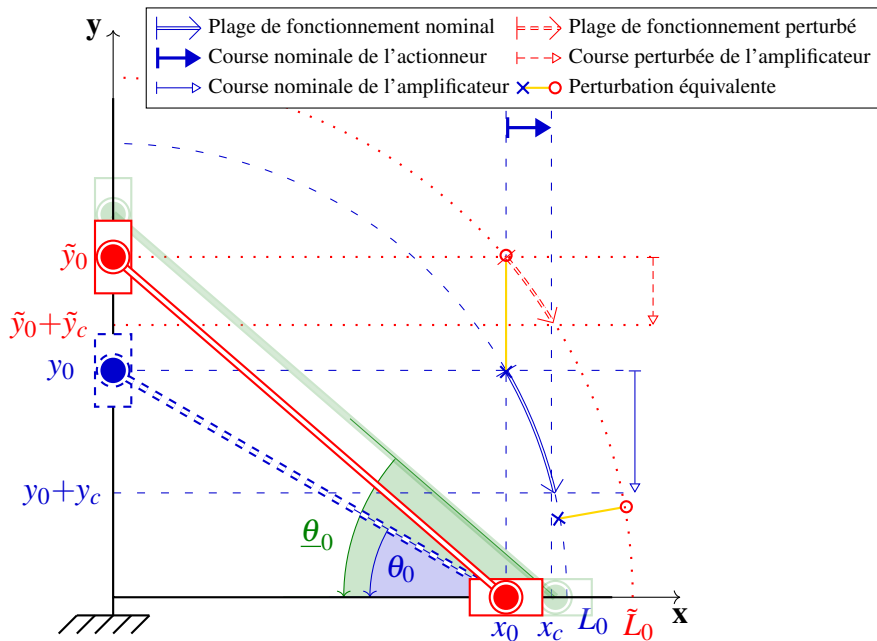


FIGURE 5.6 – Configurations nominale (tirets bleus) et perturbée (trait rouge) du mécanisme pour une perturbation équivalente $\tilde{L}_0 = \frac{x_0}{x_0} L_0$. La configuration avec perturbation réelle est rappelée en vert transparent.

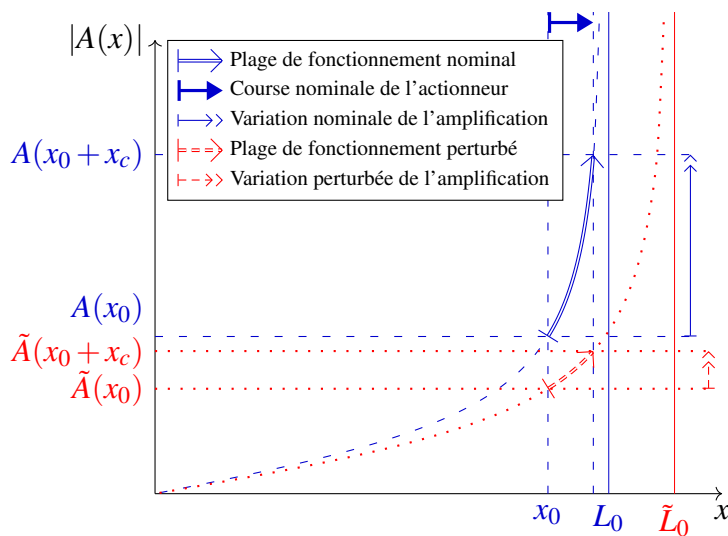


FIGURE 5.7 – Modification du rapport d'amplification du mécanisme pour une perturbation à $\delta x_0 = 0$. Le rapport d'amplification équivalent correspond bien au rapport d'amplification réel après mise à l'échelle $\underline{x} = \frac{L_0}{L_0} x$.

déplacement de la céramique c'est-à-dire $\tilde{x}_0 = x_0$. Ceci correspond à la possibilité d'ajuster expérimentalement la longueur à vide de la céramique au moyen de cales.

5.2.3.3 Extraction de données

Afin d'évaluer l'intérêt de l'analyse de sensibilité de haut degré, nous choisissons de travailler avec des données issues directement des caractéristiques des actionneurs précédemment décrits.

Géométrie nominale Le tableau 5.1 résume les cotes nécessaires à l'étude des 3 actionneurs couvrant la gamme M caractérisée par une longueur de l'ordre de $g = 50\text{mm}$. La sortie initiale y_0 et l'angle initial θ_0 du mécanisme sont extraits directement du modèle 3D disponible sur le site du constructeur [CEDRAT, 2017]. Le point de fonctionnement nominal est ainsi défini. Le "déplacement maximal sans charge" fourni par le fournisseur dans la documentation correspond à la valeur absolue de la contraction maximale h_c de l'actionneur. Le déplacement maximal de la sortie du mécanisme y_c correspond à la moitié de cette valeur car deux bras du mécanisme se déforment pour générer ce déplacement. Les valeurs relatives y_c/y_0 minimale, nominale et maximale du déplacement maximal sont fournies.

Afin de simplifier la comparaison des différents actionneurs étudiés, nous proposons de normaliser leurs points de fonctionnement nominaux. Celui-ci peut être entièrement décrit par la donnée de θ_0 et L_0 . Le paramètre géométrique L_0 est ainsi normalisé arbitrairement à 1. Notons que la valeur de sortie initiale y_0 est la seule donnée avant normalisation dans le tableau 5.1.

Erreur géométrique L'observation de la figure 5.3 indique que les incertitudes sur la hauteur de l'actionneur h et sa longueur g peuvent résulter de l'usinage du mécanisme d'amplification et de son assemblage avec la céramique.

Supposons que l'incertitude soit constante sur la hauteur de l'actionneur. On peut alors calculer l'erreur relative sur la sortie initiale du mécanisme $\frac{y_0 - y_0}{y_0} = \frac{\delta h}{h}$ à partir de l'incertitude relative sur la hauteur de l'actionneur. Nous choisissons de ne pas considérer d'erreur sur l'entrée initiale du mécanisme : $\frac{x_0 - x_0}{x_0} = 0$. Ce choix est justifié par le fait qu'une étude ne considérant pas d'erreur sur l'entrée peut être ramenée à une étude équivalente en considérant une mise à l'échelle du problème. Cela permet finalement une étude de sensibilité à $\tilde{x}_0 = x_0$. En effet, la longueur à vide de l'actionneur peut être ajustée au moyen de cales. Les résultats de sensibilité pour les actionneurs réels pourront être retrouvés en prenant en compte la normalisation de L_0 et le cas échéant la mise à l'échelle.

Les données constructeur brutes ou déduites de la documentation en terme de géométrie réelle (incertitudes de fabrication) sont synthétisées dans le tableau 5.1.

Caractéristique Modèle	Configuration initiale		Erreur		Plage de fonctionnement $\frac{y_c}{y_0} = \frac{h_c}{2h}$ [<i>min, nom, max</i>]
	θ_0	y_0 [mm]	$\frac{\tilde{y}_0 - y_0}{y_0} = \frac{\delta h}{h}$	$\delta \theta_0$	
<i>APA150M</i>	9.366°	3.689	0.00045	0.041°	[0.0235, 0.0295, 0.0347]
<i>APA400M</i>	3.517°	1.165	0.00154	0.054°	[0.1575, 0.1966, 0.2322]
<i>APA900M</i>	2.469°	0.815	0.004	0.099°	[0.3933, 0.4920, 0.5805]

TABLEAU 5.1 – Données constructeur brutes ou déduites de la documentation sur les actionneurs piézoélectriques amplifiés *APA150M*, *APA400M* et *APA900M* de CEDRAT TECHNOLOGIES.

5.2.4 Exploitation de la DA de haut-degré

5.2.4.1 Démarche

Nous souhaitons réaliser une analyse de sensibilité locale de la sortie y , dont dépend le rapport d'amplification A , à une perturbation du paramètre géométrique L_0 . Dans le cas du mécanisme Scott-Russell, cette analyse peut être faite analytiquement car le MGD est disponible et ses équations sont simples. Nous choisissons cependant de réaliser celle-ci en exploitant la DA de haut-degré afin d'évaluer l'intérêt de l'exploitation de cet outil générique. Les équations (5.7) et (5.8) sont implémentées dans un code Matlab.

En un point de fonctionnement donné (x_0, y_0, θ_0) , nous cherchons donc à calculer δy et δA pour une perturbation δL arbitraire, la perturbation de l'entrée étant considérée nulle ($\tilde{x}_0 = x_0$). Des polynômes de Taylor sont utilisés pour approcher la fonction $\mathbf{R}^{MGD}(x = x_0, L)$ au voisinage de L_0 :

$$y(L_0 + \delta L) \approx \sum_{k=0}^K (\delta L)^k \frac{1}{k!} \frac{d^k y}{dL^k}(L_0), \forall L_0 + \delta L \in \mathcal{V}(L_0), \quad (5.10)$$

Une approximation de la fonction $A^x(x = x_0, L)$ est également recherchée.

Nous proposons de calculer les polynômes de Taylor (5.10) avec la différentiation automatique de haut-degré (DAHD) par surcharge d'opérateur présentée en section 2.3.3.2. Nous exploitons donc l'outil de DAHD que constitue la classe Taylor du logiciel Diamanlab. Le paramètre géométrique L par rapport auquel on veut différentier la fonction est déclaré comme variable de type @Taylor de valeur $L_0 = 1$ et de coefficient d'ordre 1 initialisé à 1. Le code Matlab permettant le calcul des polynômes (5.10) et leur évaluation pour une perturbation du paramètre est présenté en annexe E.

Les équations (5.7) et (5.8) sont alors évaluées grâce à la bibliothèque de surcharge d'opérateurs de manière transparente pour l'utilisateur. On obtient une variable de type

Taylor représentant le polynôme de Taylor (5.10) par sa valeur y_0 , son ordre de troncature K et ses coefficients de Taylor $y_k = \frac{1}{k!} \frac{\partial^k y}{\partial L^k}(x_0, L_0)$.

Au point de fonctionnement $(x_0, y_0, \theta_0, L_0)$ où le polynôme a été calculé, on peut alors estimer la perturbation de sortie δy_0 résultant d'une perturbation δL dans un voisinage $\mathcal{V}(L_0)$ défini par le rayon de convergence. Le polynôme peut être évalué au point $L_0 + \delta L$ à un ordre $k \leq K$ par l'utilisation de la fonction `evalseries`.

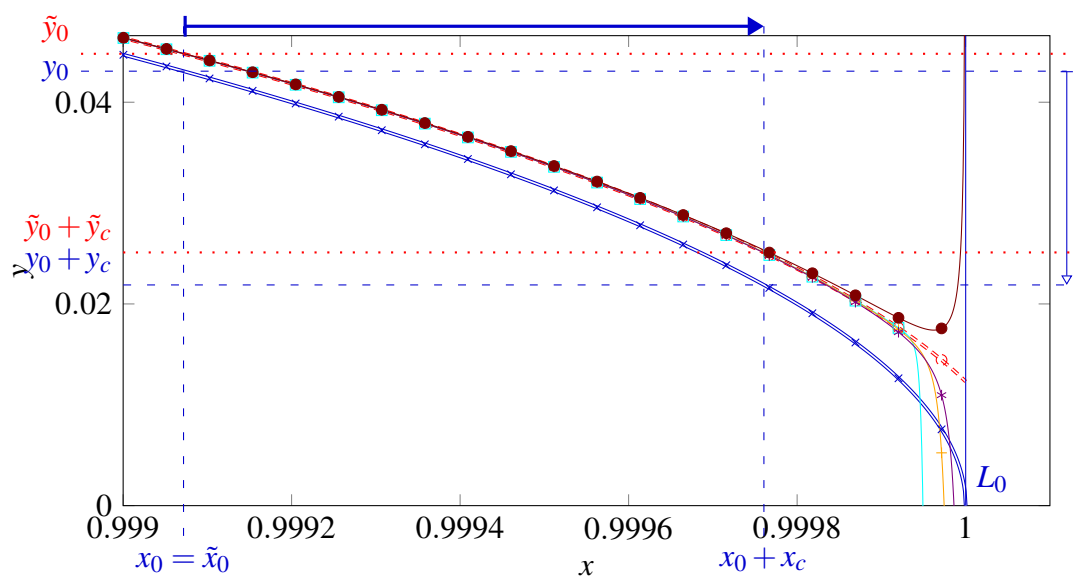
L'analyse de sensibilité décrite précédemment est réalisée en chaque configuration du mécanisme dans la plage de fonctionnement $[x_0, x_0 + x_c]$ de l'amplificateur étudié. En pratique, pour l'actionneur APA900M, on représente le *domaine de variation* de la céramique $[0.999, 1[\supset [x_0, x_0 + x_c]$ qui est discrétisé en 1000 points et on affiche un marqueur tous les 50 points de calcul. Pour l'actionneur APA400M le domaine de variation de la céramique est $[0.998, 1[\supset [x_0, x_0 + x_c]$. Ceci permet d'étudier les résultats de l'analyse de sensibilité de haut-degré jusqu'au plus proche de la configuration singulière $x = 1$. Un polynôme de Taylor est ainsi calculé en chaque point du domaine.

5.2.4.2 Résultats

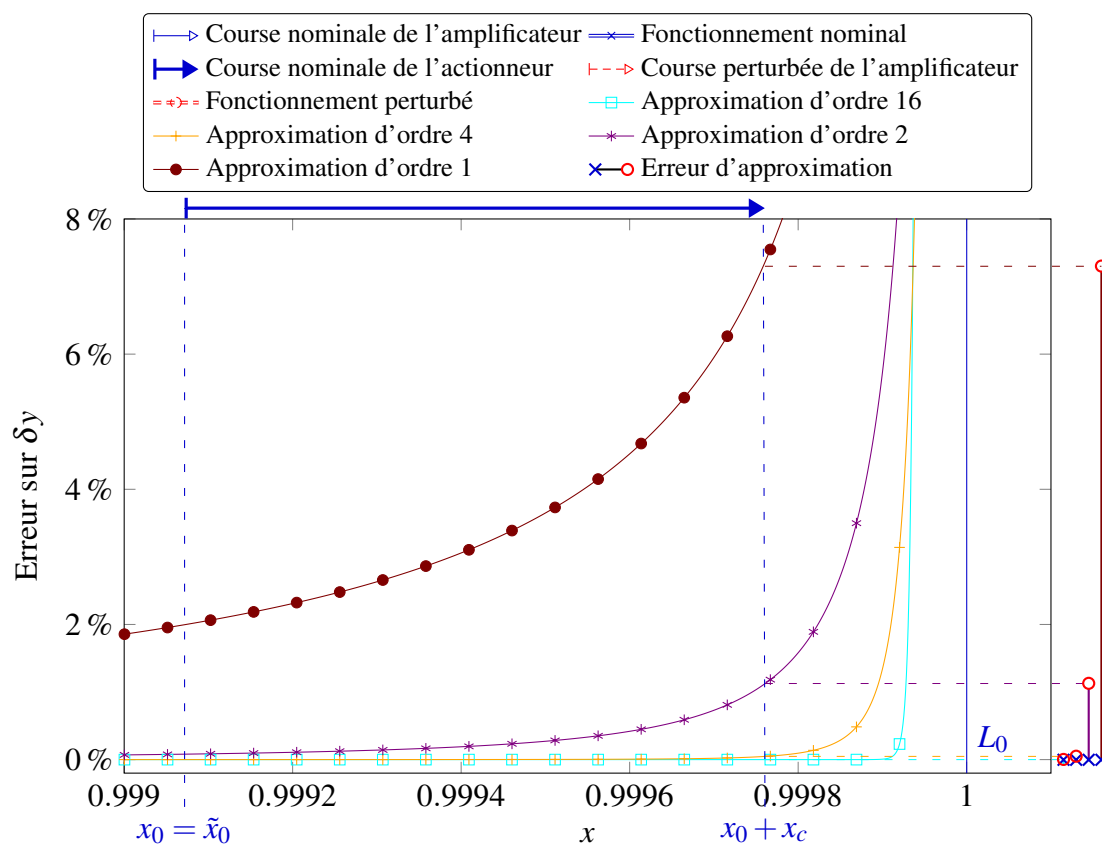
Le déplacement maximal d'entrée x_c est calculé via les formules trigonométriques appliquées à y_{max} et y_{min} en fonctionnement nominal. Les polynômes sont évalués pour la perturbation équivalente $\delta y_0/y_0 = 4\%$, puis tronqués aux ordres 1, 2, 4 et 16 pour produire différentes courbes.

La figure 5.8(a) présente la sensibilité de la configuration du mécanisme APA900M à la perturbation δL du paramètre pour $x = \tilde{x}$. La plage de fonctionnement nominal est calculée par application de l'équation (5.7) aux différents points de calcul. Les configurations dans la plage de fonctionnement nominal sont représentées en double-trait marqué de croix bleus. La plage de fonctionnement perturbé est calculée par application directe de l'équation (5.7) avec le paramètre perturbé \tilde{L}_0 à des fins de validation des résultats de sensibilité. Les configurations dans la plage de fonctionnement perturbé sont représentées en double-trait pointillés marqué de cercles rouges. La flèche à tête creuse en pointillés rouges représente la course maximale de l'amplificateur à géométrie perturbée par rapport à la géométrie nominale représentée en bleu.

Les approximations d'ordre 1, 2, 4 et 16 reliant les points d'évaluation des polynômes pour la perturbation donnée sont représentées par des lignes aux marqueurs \bullet , $*$, $+$ et \square . Comme attendu, la figure 5.8(a) montre que l'approximation diverge pour une valeur de x proche de L_0 . Cela correspond à un rétrécissement du rayon de convergence des polynômes construits à proximité du point $x = L_0$.

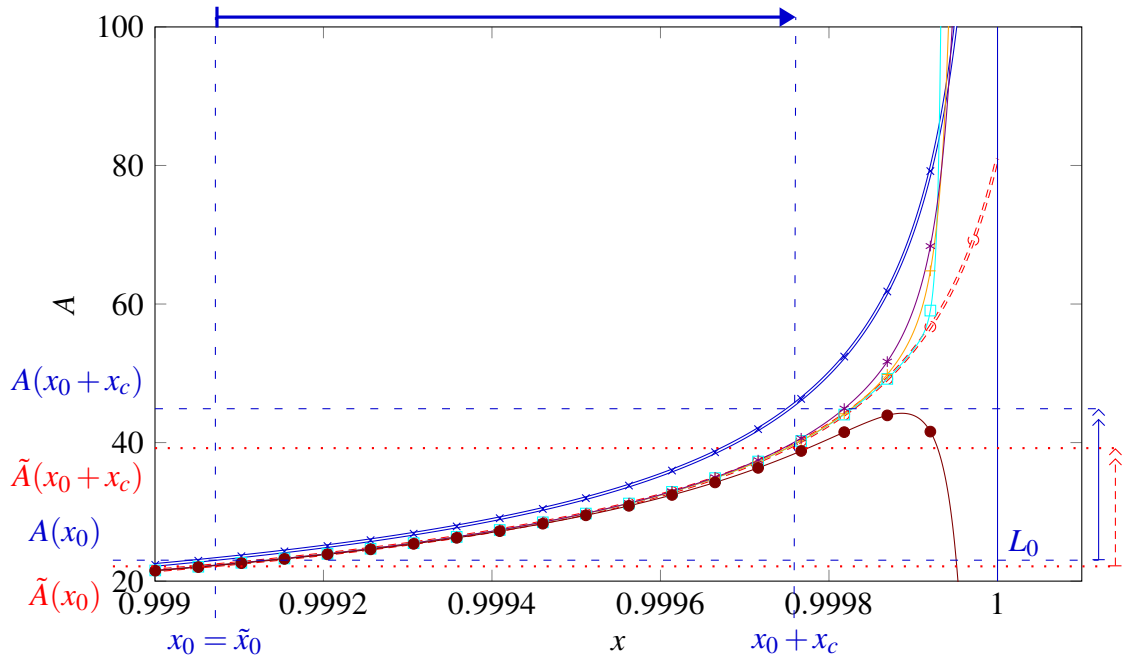


(a) Configurations assemblables

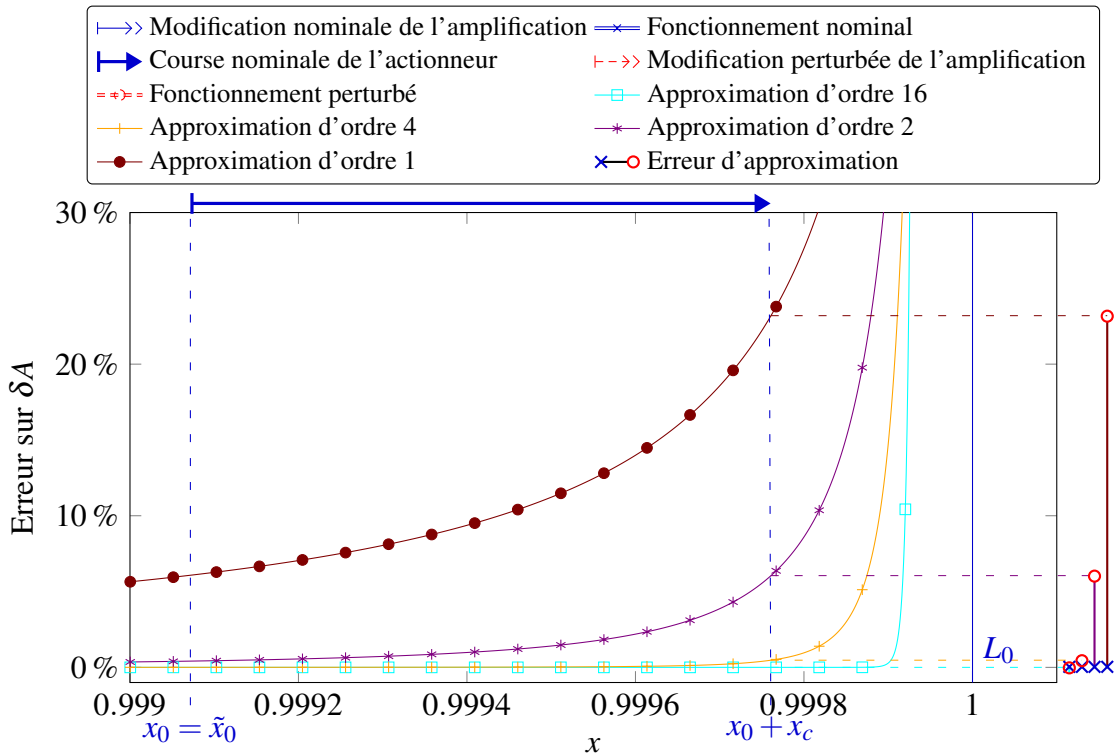


(b) Erreur relative d'approximation des configurations perturbées

FIGURE 5.8 – Analyse de sensibilité de la configuration de l'amplificateur APA900 à une perturbation de la hauteur initiale $\delta y_0/y_0 = \delta h/h = 4\%$. Le point de fonctionnement nominal est fixé à $y_0 = 0.815\text{mm}$. Le déplacement maximal nominal est de $\Delta y = 0.401\text{mm}$.

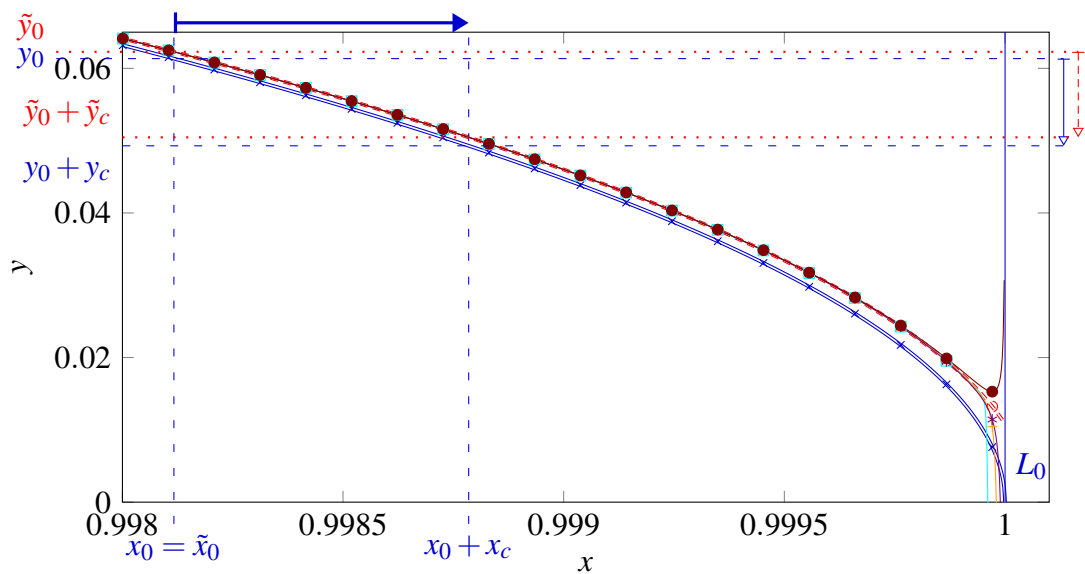


(a) Rapport d'amplification

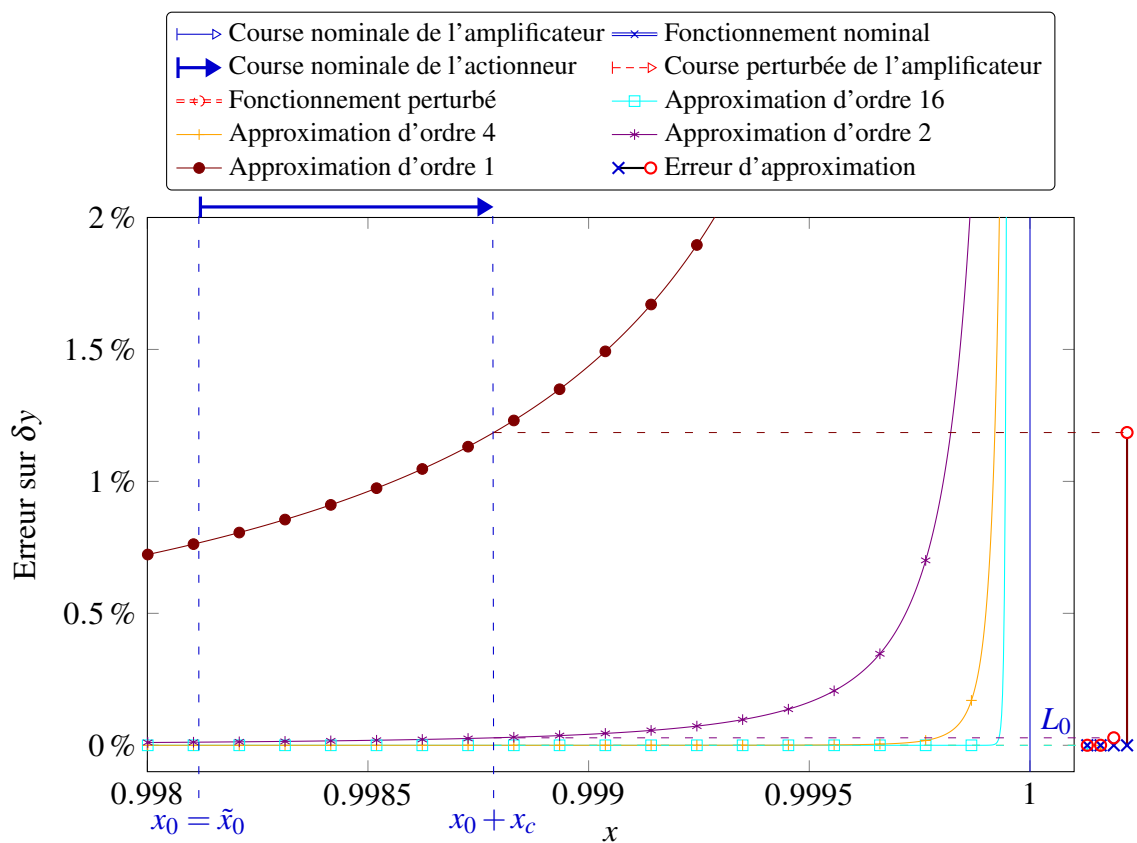


(b) Erreur relative d'approximation du rapport d'amplification perturbé

FIGURE 5.9 – Sensibilité du rapport d'amplification de l'amplificateur APA900 à une perturbation de la hauteur initiale $\delta y_0/y_0 = \delta h/h = 4\%$. Le point de fonctionnement nominal est fixé à $y_0 = 0.815mm$. Le déplacement maximal nominal est de $\Delta y = 0.401mm$.



(a) Configurations assemblées



(b) Erreur relative d'approximation des configurations perturbées

FIGURE 5.10 – Analyse de sensibilité de la configuration de l'amplificateur APA400 à une perturbation de la hauteur initiale $\delta y_0/y_0 = \delta h/h = 1.54\%$. Le point de fonctionnement nominal est fixé à $y_0 = 1.165\text{mm}$. Le déplacement maximal nominal est de $\Delta y = 0.229\text{mm}$.

L'erreur d'approximation des configurations perturbées dans le domaine de variation de l'APA900M est représentée en figure 5.8(b) pour une analyse de sensibilité d'ordre 1, 2, 4 et 16. L'analyse d'ordre 1 fournit une approximation de la configuration perturbée avec une erreur comprise entre 2% et 7%. L'analyse d'ordre 2 permet une bien meilleure approximation avec une erreur inférieure à 1%. L'analyse à un ordre 16 permet d'obtenir des résultats avec une erreur négligeable. Pour l'erreur considérée sur le paramètre considéré, l'analyse de sensibilité ne fournit pas d'approximation acceptable au-delà de $x = 0.9999$. Cependant la figure montre que l'analyse de sensibilité de haut-degré permet d'étudier l'amplificateur fonctionnant proche de la configuration singulière sur toute sa plage de fonctionnement.

La figure 5.9(a) présente le rapport d'amplification du mécanisme en fonction de sa configuration repérée par $x = \tilde{x}$. L'analyse d'ordre 1 fournit une approximation du rapport d'amplification avec une erreur comprise entre 6% et 24%. Les analyses d'ordre 2 et 4 permettent une bien meilleure approximation avec des erreurs respectivement inférieures à 6% et 1%. L'analyse à un ordre 16 permet d'obtenir des résultats avec une erreur négligeable.

L'erreur d'approximation des configurations perturbées de l'APA400M est représentée en figure 5.10(b) pour une analyse de sensibilité d'ordre 1, 2, 4 et 16. L'analyse d'ordre 1 fournit une approximation de la configuration perturbée avec une erreur comprise entre 0.7% et 1.2%. L'analyse d'ordre 2 permet une approximation avec une erreur inférieure à 0.05% sur la plage de fonctionnement. L'analyse à un ordre supérieur à 4 permet de fournir une très bonne approximation sur toute la plage de fonctionnement.

On constate que l'analyse de sensibilité présente un intérêt dans l'étude de ce mécanisme aux caractéristiques cinématiques particulières. En partant de données réalistes issues de Cedrat Technologies, nous avons constaté qu'une analyse de sensibilité de haut-degré permet d'étudier avec un haut-degré de précision des modèles dont la plage de fonctionnement est très proche de la configuration singulière, tels que les APA400M et APA900M. Nos résultats montrent que ceux-ci ne peuvent être étudiés avec une analyse de sensibilité d'ordre 1. L'approximation obtenue est très bonne sur le domaine où les polynômes calculés convergent. A un ordre élevé, la divergence des polynômes se produit de manière soudaine. Ceci peut potentiellement permettre de définir un seuil pour le domaine où l'analyse de sensibilité est valide.

5.3 Conclusion

Nous avons présenté dans ce chapitre une méthode d'analyse de sensibilité de haut-degré. Le principe de cette méthode est d'exploiter des dérivées de haut-degré du modèle étudié là où la méthode de linéarisation n'exploite qu'une dérivée d'ordre 1. La mise en

oeuvre de la méthode est facilitée par l'exploitation de la différentiation automatique de haut-degré par surcharge d'opérateur. Celle-ci est illustrée sur l'exemple d'un mécanisme d'amplification dont le rapport d'amplification est grand par un fonctionnement à proximité de sa configuration singulière.

La modification de la course maximale et du rapport d'amplification d'actionneurs piézoélectriques amplifiés ont été étudiés. En partant de données réalistes issues de Cedrat Technologies, nous avons pu constater qu'une analyse de sensibilité de haut-degré permet d'étudier avec un degré de précision très intéressant des actionneurs dont la plage de fonctionnement est très proche de la configuration singulière, tels que les *APA400M* et *APA900M*. Ceux-ci ne peuvent être étudiés avec une analyse de sensibilité d'ordre 1. L'approximation obtenue est très bonne sur le domaine où les polynômes calculés sont valides. Pour une approximation de haut-degré, la divergence des polynômes est constatée, mais une procédure pour détecter automatiquement le domaine de validité est envisageable.

A notre sens, l'analyse de sensibilité présente donc un intérêt dans l'étude de mécanismes au comportement cinématique caractérisé par de fortes non-linéarités ou dont le fonctionnement est envisagé à proximité des configurations singulières comme les dispositifs d'amplification.

Il serait intéressant d'intégrer un outil d'analyse de sensibilité générique dans les outils de détermination de l'espace de travail et des singularités développés. Cela permettrait d'assister le concepteur dans la proposition de paramètres géométriques. Cette intégration nécessite une formulation générique de problèmes de sensibilité. Des systèmes d'équations doivent être formulés automatiquement à partir du modèle géométrique implicite du mécanisme au moyen de la DA. La méthode de continuation Diamant pourrait être exploitée pour calculer les polynômes de Taylor dans le cadre d'une analyse de sensibilité de haut-degré et calculer automatiquement un domaine de validité du polynôme.

Une autre approche intéressante pour l'analyse de sensibilité du comportement cinématique des mécanismes est le calcul de branches de sensibilité [Charpentier et Lampoh, 2016]. Celles-ci constituent une information de sensibilité d'ordre 1, continue dans une dimension, qui peut être calculée pour un faible coût de calcul supplémentaire conjointement au calcul de branches de solutions d'un système.

Conclusion générale

« We can only see a short distance ahead, but we can see plenty there that needs to be done »,

Alan Turing.

Dans ce travail de thèse, nous nous sommes attachés à proposer de nouveaux outils numériques pour assister la conception de mécanismes, durant laquelle il est encore difficile de s'appuyer sur des outils génériques d'évaluation.

Synthèse des contributions

Nous avons proposé plus précisément trois outils. La différentiation automatique par transformation de source permet l'obtention automatique d'un code différencié permettant la génération d'un modèle cinématique implicite à partir de la simple écriture d'un modèle géométrique implicite. La différentiation automatique de haut-degré par surcharge d'opérateur permet le calcul de coefficients de Taylor d'ordre élevé de manière générique et efficace et en toute transparence pour l'utilisateur. La méthode de continuation de haut-degré Diamant permet alors une résolution générique et efficace de tout problème formulé comme système d'équations non-linéaires.

Le couplage de ces trois outils numériques a permis de développer au chapitre 3 un nouvel outil pour la détermination des frontières de l'espace de travail. La différentiation automatique permet de formuler les équations décrivant les frontières. La différentiation automatique de haut-degré combinée à la continuation de haut-degré et au changement de branche par perturbation permettent d'envisager de déterminer l'ensemble des solutions connectées sous forme continue à partir d'un point initial. Une stratégie de discrétisation permet de traiter des espaces de travail de dimension 3 et supérieure.

Ce nouvel outil permet donc au concepteur d'analyser l'espace de travail d'un mécanisme dont il ne connaît que le modèle géométrique implicite. Les frontières de l'espace des positions accessibles du mécanisme RRR ont été établies avec précision et avec une efficacité largement supérieure à celle de la méthode d'analyse par intervalles branch-and-prune. Les frontières de l'espace de travail du mécanisme Orthoglide ont également été établies avec succès.

Le couplage d'outils numériques pour la détection et la classification des configurations singulières d'un mécanisme a été proposé au chapitre 4. La détection des singularités est réalisée selon une méthode comparable à celle de détermination des frontières de l'espace de travail. Nous avons implémenté toutes les relations nécessaires à la classification en exploitant une étape de différentiation automatique du modèle géométrique implicite. Durant la résolution, la précision des dérivées et des solutions générées sont exploitées pour détecter spécifiquement les singularités entrée-sortie qui sont les plus difficiles à déterminer.

Ce nouvel outil doit permettre de déterminer les lieux de singularités ainsi que le comportement cinématique dégénéré y apparaissant. Des résultats encourageants ont ainsi été obtenus sur un mécanisme connu pour la complexité de ses singularités. Les singularités sont correctement déterminées à l'exception d'un jeu de singularités de dimension accrue.

Nous avons montré au chapitre 5 l'apport d'une méthode d'analyse de sensibilité de haut-degré. Sur un exemple simple, la méthode permet d'obtenir une information de sensibilité précise même à proximité de configurations singulières d'intérêt pour les performances de certains mécanismes.

Perspectives

L'expression du modèle géométrique sous forme scalaire est aisée à obtenir à partir des équations de fermeture de chaînes géométriques. Nous avons alors pu nous appuyer sur la continuation pour résoudre le système d'équations obtenu, et sur l'emploi de la différentiation automatique tel que décrit dans ce manuscrit. Il est intéressant en robotique du fait de la nature des mouvements de paramétrer la pose de l'effecteur d'un mécanisme par un vecteur de $SE(3)$ et d'exprimer une relation avec l'état des liaisons en utilisant des transformations décrivant des mouvements de corps rigides.

Les travaux récents de Sommer *et al.* [2016] doivent permettre d'envisager d'exploiter de telles représentations avec un gain potentiel d'efficacité. Les auteurs proposent alors une bibliothèque qui code les opérateurs et leurs dérivées d'ordre 1. Une application dans le cadre

de nos outils requiert le développement d'une bibliothèque de surcharge d'opérateurs pour la différentiation de haut-degré sur les variétés différentiables.

Des limites subsistent dans la détermination de l'espace de travail et la détection des singularités au moyen de la continuation de haut-degré. Si la résolution d'ensembles de solution de dimension supérieure à 1 est possible, une stratégie générique pour leur discrétisation reste à définir. En particulier lors de la présence de singularités impliquant un gain local de mobilité, une stratégie pour la prise en compte automatisée de ce gain de dimension serait à développer. Sur ce volet, Henderson [2007] propose de manière intéressante une méthode de continuation dite « higher-dimensional » pour aborder de front le calcul d'ensembles de solutions de dimension supérieure à 1. Cette méthode peut constituer une solution pour une stratégie générique de détermination des ensembles de dimension élevée.

Le calcul d'ensembles composés de solutions non-connectées est par essence même difficile avec une méthode de continuation. Une voie pour pallier cette faiblesse dans notre contexte consiste en un couplage avec une méthode d'analyse par intervalles [Ratschek et Rokne, 2005 ; Emeliyanenko *et al.*, 2009]. Un tel couplage permettrait d'obtenir une méthode possédant les atouts complémentaires des méthodes de continuation et d'analyse par intervalles, cette dernière possédant l'avantage de garantir les solutions.

Les résultats concernant l'utilisation de termes de sensibilité de haut-degré soulignent par ailleurs l'intérêt d'aller plus loin dans l'utilisation d'outils évaluant la sensibilité. Un premier niveau d'exploitation est l'intégration de fonctionnalités d'analyse de sensibilité générique dans les outils d'évaluation des caractéristiques de mécanismes. Un deuxième niveau est de s'appuyer sur l'analyse de sensibilité pour mettre en place un outil d'optimisation.

Enfin tous les travaux effectués pour des mécanismes à corps rigides sont potentiellement d'intérêt pour des systèmes déformables comme les mécanismes compliant ou les robots à tubes concentriques. Sur ce plan, des travaux sont en cours pour exploiter les diagrammes de bifurcation et la continuation numérique.

« Devenir soi nécessite un détour par les autres, car il faut s'immerger dans toutes les richesses apportées par nos prédécesseurs. Ce détour peut être, périlleux, douloureux, mais en faire l'économie serait délibérément s'appauvrir »,

Albert Jacquard.

Liste des publications

Revue nationale avec comité de lecture

G. Hentz, I. Charpentier, P. Renaud, « Higher-order continuation for the determination of robot workspace boundaries », *Comptes Rendus Mécanique*, Volume 344, Pages 95-101, 2016.

Conférence internationale avec comité de lecture et édition des actes

G. Hentz, I. Charpentier, L. Rubbert, P. Renaud, « A Taylor-based continuation method for the determination and classification of robot singularities », dans *Advances in Robot Kinematics (ARK)*, Nice, Juin 2016, Présentation orale.

Conférence internationale avec comité de lecture sans actes

G. Hentz, I. Charpentier, L. Rubbert, P. Renaud, « A general bifurcation analysis method for robot workspaces », *Engineering Mechanics Institute (EMI) International Conference of ASCE*, Metz, Octobre 2016, Présentation orale.

Autres communications

G. Hentz, I. Charpentier, L. Rubbert, P. Renaud, « Développement d'un outil numérique générique pour l'analyse cinématique d'un robot », *Journée Recherche de l'INSA de Strasbourg*, Strasbourg, Mars 2016, Poster.

G. Hentz, I. Charpentier, P. Renaud, « Outils numériques pour la conception des mécanismes », *Journée « Poster » de l'école doctorale MSII (ED269)*, Strasbourg, Octobre 2015, **Prix du meilleur poster**.

G. Hentz, I. Charpentier, P. Renaud, « Une approche générique pour la détermination des frontières de l'espace de travail d'un robot », *Journée nationale de la recherche en robotique (JNRR)*, Cap Hornu, Octobre 2015, Poster.

G. Hentz, I. Charpentier, P. Renaud, « Higher-order continuation for the determination of robot workspace boundaries », *Journées des jeunes chercheurs en robotique (JJCR)*, Amiens, Octobre 2015, Présentation orale.

G. Hentz, I. Charpentier, P. Renaud, « Higher-order continuation for the determination of robot workspace boundaries », *Journée du GT6 « Conception de robots »*, Paris, Septembre 2015, Présentation orale.

Table des annexes

A	Codes Fortran et Matlab pour la détermination des frontières de l'espace des positions accessibles du mécanisme RRR	149
B	Code Matlab pour la détermination des frontières de l'espace de travail du mécanisme Orthoglide	155
C	Differentiation automatique multidirectionnelle pour le code Fortran du mécanisme double-chaîne	159
D	Fonction Matlab utilisée pour la détermination des singularités du mécanisme double-chaînes	165
E	Code Matlab pour l'analyse de sensibilité de haut-degré d'un mécanisme d'amplification	169

Annexe A : Codes Fortran et Matlab pour la détermination des frontières de l'espace des positions accessibles du mécanisme RRR

Les codes présentés ici sont exploités pour la détermination des frontières de l'espace des positions accessibles du mécanisme RRR avec le logiciel Diamanlab.

Code Fortran différentié avec Tapenade

Le système étendu (3.6) est construit en suivant le processus (3.8). Le modèle géométrique implicite $\mathbf{R}^{\mathcal{L}}(\mathbf{q}) = \mathbf{0}$ est implémenté sous forme augmentée

$$\begin{pmatrix} \mathbf{R}^{\mathcal{L}}(\mathbf{u}, \mathbf{z}) \\ (\mathbf{R}^{\mathcal{L}})^{\top}(\mathbf{u}, \mathbf{z}) \cdot \xi \\ \xi^{\top} \cdot \xi - 1 \end{pmatrix} = \mathbf{0}$$

dans une routine Fortran. Le code source correspondant est différentié par rapport aux coordonnées $\mathbf{z} = \mathbf{v}$ représentées par la variable z avec Tapenade. Les étapes d'obtention du code différentié avec Tapenade sont détaillées en annexe D. Le code différentié obtenu est :

```
1  !! Code Fortran du systeme augmente du RRR differentie par rapport
    aux coordonnes actives
    !           Generated by TAPENADE      (INRIA, Ecuador team)
    ! Tapenade 3.12 (r6213) - 13 Oct 2016 10:54
    !
    ! Differentiation of residual in forward (tangent) mode:
6  ! variations of useful results: res restxi
```

```

! with respect to varying inputs: z
! RW status of diff variables: res:out restxi:out z:in
SUBROUTINE RESIDUAL_DV(u, z, zd, xi, res, resd, restxi, restxid,
    nxi, nbdirs)
    USE DIFFSIZES
11 ! Hint: nbdirsmax should be the maximum number of differentiation
    directions
    IMPLICIT NONE
    DOUBLE PRECISION :: l1, l2, l3, a, b, u(2), z(3), xi(2), eq1
    DOUBLE PRECISION :: zd(nbdirsmax, 3), eq1d(nbdirsmax)
    DOUBLE PRECISION :: xi1, xi2, res(2), restxi, nxi, eq2
16 DOUBLE PRECISION :: resd(nbdirsmax, 2), restxid(nbdirsmax), eq2d
    (nbdirsmax)
    DOUBLE PRECISION :: v1, v2, v3, x, y, alpha, beta, gamma
    DOUBLE PRECISION, DIMENSION(nbdirsmax) :: v1d, v2d, v3d, alphad,
    betad, gammad
    INTRINSIC COS, SIN
    DOUBLE PRECISION :: arg1, arg2,
21 DOUBLE PRECISION, DIMENSION(nbdirsmax) :: arg1d, arg2d
    INTEGER :: nd, nbdirs
! Coordonnees
    x = u(1); y = u(2); v1 = z(1); v2 = z(2); v3 = z(3)
    alpha = a + b*SIN(v1); beta = a + b*SIN(v2)
26 gamma = a + b*SIN(v3)
    arg1 = alpha + beta; arg2 = alpha + beta + gamma
    DO nd=1, nbdirs
        v1d(nd) = zd(nd, 1); v2d(nd) = zd(nd, 2)
        v3d(nd) = zd(nd, 3)
31 ! Changements de variables
        alphad(nd) = b*v1d(nd)*COS(v1)
        betad(nd) = b*v2d(nd)*COS(v2)
        gammad(nd) = b*v3d(nd)*COS(v3)
! Equations de fermeture vectorielle
36 arg1d(nd) = alphad(nd) + betad(nd)
        arg2d(nd) = alphad(nd) + betad(nd) + gammad(nd)
        eq1d(nd) = l1*alphad(nd)*SIN(alpha) + l2*arg1d(nd)*SIN(arg1) +
            l3*arg2d(nd)*SIN(arg2)
        arg1d(nd) = alphad(nd) + betad(nd)
        arg2d(nd) = alphad(nd) + betad(nd) + gammad(nd)

```

```

41 ! Systeme augmente
    resd(nd, :) = 0.D0
    resd(nd, 1) = eq1d(nd)
    END DO
    xi1 = xi(1); xi2 = xi(2)
46 eq1 = x - (11*COS(alpha)+12*COS(arg1)+13*COS(arg2))
    arg1 = alpha + beta; arg2 = alpha + beta + gamma
    DO nd=1,nbdirs
        eq2d(nd) = -(11*alphad(nd)*COS(alpha)) - 12*arg1d(nd)*COS(arg1)
            - 13*arg2d(nd)*COS(arg2)
        resd(nd, 2) = eq2d(nd)
51 restxid(nd) = xi1*eq1d(nd) + xi2*eq2d(nd)
    END DO
    eq2 = y - (11*SIN(alpha)+12*SIN(arg1)+13*SIN(arg2))
    res(1) = eq1; res(2) = eq2
    restxi = eq1*xi1 + eq2*xi2
56 nxi = xi1*xi1 + xi2*xi2 - 1
    END SUBROUTINE RESIDUAL_DV

```

Fonction Matlab pour la détermination des frontières de l'espace des positions accessibles

Le code augmenté différentié est réduit par la sélection :

- des n_e premières composantes de res correspondant à \mathbf{R}^e ,
- des n_v composantes de $restxid$ correspondant à $(\mathbf{R}_z^e)^T(\mathbf{u}, \mathbf{z}) \cdot \xi$,
- de n_{xi} correspondant à $\xi^T \cdot \xi - 1$.

Le code Fortran est traduit en une fonction Matlab contenant les équations résiduelles relatives au système étendu (3.6) :

```

%% Code Matlab R.m pour la determination des frontieres de l'
    espace des positions accessibles du mecanisme RRR
2 function [res]= R(obj,U)
    global Ck

%% Modele du mecanisme RRR
% Topologie
7 nu = 2; % Nombre de coordonnees operationnelles

```

```

nv = 3; % Nombre de coordonnees actives
nw = 0; % Nombre de coordonnees passives
nz = nv+nw; nq = nv+nw+nu; % Nombre de coordonnees geometriques
% Parametres geometriques
12 l1=4; l2=2; l3=1;
% Debatement articulaire limite : theta varie dans [a-b;a+b]
a=0; b=pi/3;

if isa(U, 'Taylor') % initialisation de res de type Taylor
17   res=U(obj.ninc-1)*0;
else % res est de type double pour le Newton-Raphson et la
    classification
    res=zeros(obj.ninc-1,1);
end
%% Inconnues Diamanlab
22 % Coordonnees geometriques
    x = U(1); y = U(2);
    v1 = U(3); v2 = U(4); v3 = U(5);
% Changement de variable pour prise en compte du debatement
    alpha = a + b*sin(v1);
27   beta = a + b*sin(v2);
    gamma = a + b*sin(v3);

%% Determination des frontieres de l'espace des positions
    accessibles
neqcontr=2; % Nombre d'equations de contraintes
32 % Vecteur dont l'existence implique la perte de rang
    xi1 = U(6); xi2 = U(7); % dim(neqcontr)
nbdirs=nz; % Nombre de directions de differentiation
zd=eye(nz,nz); % Matrice des directions de differentiation
% le nb de colonnes correspond au nombre de variables
    independantes considerees dans Tapenade
37 % une ligne est une direction de perturbation par rapport au
    vecteur de variables independantes

%% Code differentie avec Tapenade
    arg1 = alpha + beta; arg2 = alpha + beta + gamma;
    v1d = zd(:, 1); v2d = zd(:, 2); v3d = zd(:, 3);
42 % Changements de variables
    
```

```

    alphad = b*v1d*cos(v1); betad = b*v2d*cos(v2);
    gammad = b*v3d*cos(v3);
% Equations de fermeture vectorielle
    arg1d = alphad + betad; arg2d = alphad + betad + gammad;
47    eq1d = l1*alphad*sin(alpha) + l2*arg1d*sin(arg1) + l3*arg2d*
        sin(arg2);
    arg1d = alphad + betad; arg2d = alphad + betad + gammad;
% Systeme augmente
    resd(:, 1) = eq1d;
    xi1 = xi(1); xi2 = xi(2);
52    eq1 = x - (l1*cos(alpha)+l2*cos(arg1)+l3*cos(arg2));
    arg1 = alpha + beta; arg2 = alpha + beta + gamma;
    eq2d = -(l1*alphad*cos(alpha)) - l2*arg1d*cos(arg1) - l3*arg2d
        *cos(arg2);
    resd(:, 2) = eq2d;
    restxid = xi1*eq1d + xi2*eq2d;
57    eq2 = y - (l1*sin(alpha)+l2*sin(arg1)+l3*sin(arg2));
    res(1) = eq1; % 1ere equation du systeme etendu
    res(2) = eq2; % 2eme equation du systeme etendu
    restxi = eq1*xi1 + eq2*xi2;
    nxi = xi1*xi1 + xi2*xi2 - 1;
62
% Reduction du systeme augmente differentie en systeme etendu
res(3:5)=restxid; % equations 3 a 5 du systeme etendu
res(6)=nxi; % 6eme equation du systeme etendu
end

```


Annexe B : Code Matlab pour la détermination des frontières de l'espace de travail du mécanisme Orthoglide

Le code présenté ici est exploité pour la détermination des frontières de l'espace de travail du mécanisme Orthoglide avec le logiciel Diamanlab. Le système étendu est construit en suivant le processus (3.8). Les étapes d'obtention du code différencié avec Tapenade et de son intégration dans Diamanlab sont détaillées en annexe D. Il permet le calcul d'une section des frontières de l'espace de travail pour une valeur de ϕ donnée, par exemple le diagramme de bifurcation en figure 3.12(b).

```
%% Code source Matlab pour la determination des frontieres de l'
    espace de travail du mecanisme Orthoglide
function [res]= R(obj,U)
global Ck phi
4
%% Modele du mecanisme Orthoglide
% Topologie
nu = 3; % Nombre de coordonnees operationnelles
nv = 3; % Nombre de coordonnees actives
9 nw = 0; % Nombre de coordonnees passives
nq = nv+nw+nu; % Nombre de coordonnees geometriques
% Parametres geometriques
L = 1; % Longueur d'une jambe
% Course des actionneurs limitee : theta varie dans [a-b;a+b]
14 a=[L L L]'; % dim(nv) offset de l'intervalle de contrainte des
    articulations
b=[L L L]'; % dim(nv) semi-amplitude de l'intervalle de contrainte

%% Systeme des frontieres l'espace de travail
```

```

neqcontr=3; % Nombre d'equations de contraintes
19 % directions de differentiation : toutes les coordonnees sauf
    celles definissant l'espace de travail que l'on veut etudier.
nbdirs=nv+nw; % Nombre de directions de differentiation des
    equations de contraintes
ud=eye(nbdirs,nq); % on a differentie dans tapenade par rapport a
    (v,u)
% une ligne est une direction de perturbation par rapport au
    vecteur de variables independantes

24 if isa(U,'Taylor') % initialisation de res de type Taylor
    res=U(obj.ninc-1)*0;
else % res est de type double pour le Newton-Raphson et la
    classification
    res=zeros(obj.ninc-1,1);
end
29 %% Inconnues diamanlab
% Variables d'entree
v1=U(6);
v2=U(7);
v3=U(8);
34 % Coordonnees actives – Course limitee prise en compte par
    changement de variable
rhox=a(1)+b(1)*v1;
rhoy=a(2)+b(2)*v1;
rhoz=a(3)+b(3)*v1;
% Coordonnees operationnelles – cylindriques
39 pr=U(4);
pphi=phi;
pz=U(5);
% Coordonnees operationnelles – cartesiennes
px=pr*cos(pphi);
44 py=pr*sin(pphi);
% Vecteur xi
xi = U(1:3); % dim(neqcontr)

%% Code differentie obtenu par Tapenade
49 % Ici on inclue les lignes d'interet converties en matlab et la
    boucle DO est geree par des tableaux
    
```



```

v1d = ud(:, 1);
v2d = ud(:, 2);
v3d = ud(:, 3);
54 rhoxd = b(1)*v1d*cos(v1);
rhozd = b(2)*v2d*cos(v2);
rhozd = b(3)*v3d*cos(v3);
pxd = ud(:, 4);
pyd = ud(:, 5);
59 pzd = ud(:, 6);
eq1d = -(2*(px-rhox)*(pxd-rhoxd)) - 2*py*pyd - 2*pz*pzd;
eq2d = -(2*px*pxd) - 2*(py-rhoy)*(pyd-rhoyd) - 2*pz*pzd;
eq3d = -(2*px*pxd) - 2*py*pyd - 2*(pz-rhoz)*(pzd-rhozd);
resd1 = eq1d;
64 resd2 = eq2d;
resd3 = eq3d;
% Equations de fermeture de chaine
eq1=L^2-((px-rhox)^2+py^2+pz^2);
eq2=L^2-(px^2+(py-rhoy)^2+pz^2);
69 eq3=L^2-(px^2+py^2+(pz-rhoz)^2);

% Assemblage de la matrice jacobienne Rz
resd= [resd1' ; resd2' ; resd3'];

74 %% Assemblage du systeme etendu Sz(q,xi)
res(1)=eq1;
res(2)=eq2;
res(3)=eq3;
res(4:6)=resd'*xi; % xi est orthogonal aux vecteurs propres de
resd
79 res(7)=xi'*xi-1; % xi est un vecteur unitaire
end

```


Annexe C : Differentiation automatique multidirectionnelle pour le code Fortran du mécanisme double-chaîne

Les étapes du processus (4.19), nécessaires à la construction et l'évaluation du modèle cinématique implicite sont :

- établir le code Fortran du modèle géométrique implicite ;
- différentier le code avec Tapenade pour obtenir un code différentié qui permette d'évaluer $\mathbf{R}^{\mathcal{L}}(\mathbf{q})$ et la jacobienne $\mathbf{R}_{\mathbf{q}}^{\mathcal{L}}(\mathbf{q})$;
- traduire le code Fortran en Matlab et l'intégrer dans Diamanlab comme présenté en annexe D.

Ecriture du code Fortran du modèle géométrique implicite

Supposons que l'on souhaite évaluer les dérivées de $\mathbf{R}^{\mathcal{L}}(\mathbf{q})$ par rapport aux coordonnées géométriques \mathbf{q} . Dans Diamanlab, le système générique résolu est $\mathcal{R}(\mathbf{U}) = \mathbf{0}$, donc les coordonnées géométriques sont représentées par la variable `u` et le problème résiduel $\mathcal{R}(\mathbf{U})$ est représenté par la variable `res`. La génération du code source requiert :

- la déclaration des variables dépendante `res` et indépendante `u`, et de leurs dimensions correspondant respectivement au nombre d'équations dans le modèle géométrique et au nombre de coordonnées géométriques ;
- la déclaration des fonctions intrinsèques du langage utilisées ;
- l'implémentation du modèle géométrique implicite en Fortran dans la routine `residual(u, res)` avec pour arguments `u` et `res`.

Le code source du modèle géométrique implicite du mécanisme double-chaîne est :

```

!! Code source Fortran du modele geometrique du mecanisme double-
chaîne
SUBROUTINE RESIDUAL(u , res )
IMPLICIT NONE
!! Declaration des variables et leurs dimensions !!
5 DOUBLE PRECISION :: u(8) , thetaa , thetab , thetac , thetad , thetae , thetag
, px , py
DOUBLE PRECISION :: eq1 , eq2 , eq3 , eq4 , eq5 , eq6 , res (6)
! Parametres geometriques
DOUBLE PRECISION :: ab , bc , cd , ad , cg , fg , ef , de
! Fonctions intrinseques au langage exploitees
10 INTRINSIC COS
INTRINSIC SIN

!! Modele geometrique implicite !!
! Coordonnees geometriques
15 px=u(1) ; py=u(2)
thetaa=u(3) ; thetae=u(4)
thetab=u(5) ; thetac=u(6) ; thetad=u(7) ; thetag=u(8)
! Equations de fermeture de chaines
eq1=ab*COS(thetaa)+bc*COS(thetab)-cd*COS(thetad)-ad
20 eq2=ab*SIN(thetaa)+bc*SIN(thetab)-cd*SIN(thetad)
eq3=cd*COS(thetad)+cg*COS(thetac)+fg*COS(thetag)-ef*COS(thetae)-de
eq4=cd*SIN(thetad)+cg*SIN(thetac)+fg*SIN(thetag)-ef*SIN(thetae)
eq5=cd*COS(thetad)+cg*COS(thetac)-px
eq6=cd*SIN(thetad)+cg*SIN(thetac)-py
25

!! Calcul de res !!
res(1)=eq1 ; res(2)=eq2 ; res(3)=eq3
res(4)=eq4 ; res(5)=eq5 ; res(6)=eq6
END SUBROUTINE RESIDUAL

```

Différentiation avec Tapenade

Le code source est ensuite différentié avec Tapenade afin de générer un code différentié qui permette d'évaluer plusieurs dérivées directionnelles en un appel. Tapenade est lancé avec les arguments suivants :

- « Name of the top routine » = « RESIDUAL »,
- « Dependent output variables » = « res »,
- « Independent input variables » = « u »,
- « Differentiate in » = « Multidirectional Tangent Mode ».

Le code différentié RESIDUAL_DV respecte les conventions de Tapenade :

- le paramètre `nbdirsmax` correspond à la dimension des variables déclarées comme indépendantes pour la phase de différentiation. Elle représente donc le nombre maximal de directions d'évaluation ;
- l'argument `nbdirs` correspond au nombre de directions de différentiation pour la phase d'évaluation ;
- des variables dérivées, suffixées de la lettre `d`, sont introduites comme arguments de la routine pour chaque variable indépendante, dépendante et intermédiaire ;
- le type de la variable dérivée `ud` est identique à la variable `u` et sa dimension est `DIMENSION(nbdirsmax, DIMENSION(u))`. Il en est de même pour les variables dépendantes et intermédiaires ;
- des boucles d'indice `nd` sur les directions sont introduites dans le code :

Le code source différentié est :

```

1  !! Code differentie Fortran du modele geometrique du mecanisme
    double-chaine par rapport aux coordonnees geometriques
    !           Generated by TAPENADE      (INRIA, Ecuador team)
    ! Tapenade 3.10 (r5717) - 30 Jul 2015 16:03
    !
    ! Differentiation of residual in forward (tangent) mode:
6  ! variations of useful results: res
    ! with respect to varying inputs: u
    ! RW status of diff variables: res:out u:in
SUBROUTINE RESIDUAL_DV(u, ud, res, resd, nbdirs)
    USE DIFFSIZES
11 ! Hint: nbdirsmax should be the maximum number of differentiation
    directions
    IMPLICIT NONE

```

```

DOUBLE PRECISION :: u(8), thetaa, thetab, thetac, thetad, thetae
, thetag, px, py
DOUBLE PRECISION :: ud(nbdirsmax, 8), thetaad(nbdirsmax),
thetabd(nbdirsmax), thetacd(nbdirsmax), thetadd(nbdirsmax),
thetaed(nbdirsmax), thetagd(nbdirsmax), pxd(nbdirsmax), pyd(
nbdirsmax)
DOUBLE PRECISION :: eq1, eq2, eq3, eq4, eq5, eq6, res(6)
16 DOUBLE PRECISION :: eq1d(nbdirsmax), eq2d(nbdirsmax), eq3d(
nbdirsmax), eq4d(nbdirsmax), eq5d(nbdirsmax), eq6d(nbdirsmax)
, resd(nbdirsmax, 6)
DOUBLE PRECISION :: ab, bc, cd, ad, cg, fg, ef, de
INTRINSIC COS
INTRINSIC SIN
INTEGER :: nd, nbdirs
21 thetaa = u(3); thetae = u(4);
thetab = u(5); thetac = u(6); thetad = u(7); thetag = u(8)
DO nd=1,nbdirs
!! Modele geometrique implicite !!
pxd(nd) = ud(nd, 1); pyd(nd) = ud(nd, 2)
26 thetaad(nd) = ud(nd, 3); thetaed(nd) = ud(nd, 4)
thetabd(nd) = ud(nd, 5); thetacd(nd) = ud(nd, 6)
thetadd(nd) = ud(nd, 7); thetagd(nd) = ud(nd, 8)
! Equations de fermeture de chaines
eq1d(nd) = cd*thetadd(nd)*SIN(thetad) - bc*thetabd(nd)*SIN(
thetab) - ab*thetaad(nd)*SIN(thetaa)
31 eq2d(nd) = ab*thetaad(nd)*COS(thetaa) + bc*thetabd(nd)*COS(
thetab) - cd*thetadd(nd)*COS(thetad)
eq3d(nd) = ef*thetaed(nd)*SIN(thetae) - cg*thetacd(nd)*SIN(
thetac) - fg*thetagd(nd)*SIN(thetag) - cd*thetadd(nd)*SIN(
thetad)
eq4d(nd) = cd*thetadd(nd)*COS(thetad) + cg*thetacd(nd)*COS(
thetac) + fg*thetagd(nd)*COS(thetag) - ef*thetaed(nd)*COS(
thetae)
eq5d(nd) = -(cd*thetadd(nd)*SIN(thetad)) - cg*thetacd(nd)*SIN(
thetac) - pxd(nd)
eq6d(nd) = cd*thetadd(nd)*COS(thetad) + cg*thetacd(nd)*COS(
thetac) - pyd(nd)
36 !! Calcul de res !!
resd(nd, :) = 0.D0

```

```

    resd(nd, 1) = eq1d(nd); resd(nd, 2) = eq2d(nd)
    resd(nd, 3) = eq3d(nd); resd(nd, 4) = eq4d(nd)
    resd(nd, 5) = eq5d(nd); resd(nd, 6) = eq6d(nd)
41  END DO
    px = u(1); py = u(2)
    eq1 = ab*COS(thetaa) + bc*COS(thetab) - cd*COS(thetad) - ad
    eq2 = ab*SIN(thetaa) + bc*SIN(thetab) - cd*SIN(thetad)
    eq3 = cd*COS(thetad) + cg*COS(thetac) + fg*COS(thetag) - ef*COS(
        thetae) - de
46  eq4 = cd*SIN(thetad) + cg*SIN(thetac) + fg*SIN(thetag) - ef*SIN(
        thetae)
    eq5 = cd*COS(thetad) + cg*COS(thetac) - px
    eq6 = cd*SIN(thetad) + cg*SIN(thetac) - py
    res(1) = eq1; res(2) = eq2; res(3) = eq3
    res(4) = eq4; res(5) = eq5; res(6) = eq6
51  END SUBROUTINE RESIDUAL_DV

```

Le concept clé du mode multi-directionnel tangent est l'usage d'une matrice de dimension $nbdirs \times nbdirsmax$ des directions de perturbation. Dans cet exemple, elle correspond à ud . Chaque ligne de cette dernière définit une direction de différentiation dans la base constituée par les variables indépendantes.

En pratique, pour évaluer les dérivées du modèle géométrique par rapport au jeu de coordonnées $\mathbf{z} = [\mathbf{v}^\top, \mathbf{w}^\top]^\top$ à l'aide du code différentié par rapport aux coordonnées géométriques $\mathbf{q} = [\mathbf{u}^\top, \mathbf{v}^\top, \mathbf{w}^\top]^\top$, le code linéaire tangent multidirectionnel est appelé avec les arguments suivants :

1. u : le point de différentiation
2. $nbdirs$: le nombre de directions de différentiation, *i.e.* la dimension de \mathbf{z}
3. ud : la matrice des directions de différentiation par rapport aux variables indépendantes \mathbf{v} et \mathbf{w} , c'est-à-dire :

Variables indépendantes u

$$\text{ud} = \left(\begin{array}{ccc|c}
 \mathbf{u} & \mathbf{v} & \mathbf{w} & \\
 [x \ y] & [\theta_a \ \theta_e] & [\theta_b \ \theta_c \ \theta_d \ \theta_f] & \\
 \mathbf{0} & [1 \ 0] & \mathbf{0} & \theta_a \\
 \mathbf{0} & [0 \ 1] & \mathbf{0} & \theta_e \\
 \mathbf{0} & \mathbf{0} & [1 \ 0 \ 0 \ 0] & \theta_b \\
 \mathbf{0} & \mathbf{0} & [0 \ 1 \ 0 \ 0] & \theta_c \\
 \mathbf{0} & \mathbf{0} & [0 \ 0 \ 1 \ 0] & \theta_d \\
 \mathbf{0} & \mathbf{0} & [0 \ 0 \ 0 \ 1] & \theta_f
 \end{array} \right) \left. \vphantom{\begin{array}{ccc|c} \end{array}} \right\} \begin{array}{l} \text{Directions de} \\ \text{différentiation} \end{array}$$

Dans le cas particulier où l'on veut établir les dérivées dans les directions de chacune des `nbdirsmax` variables indépendantes sélectionnées lors de la différentiation, la matrice des directions de différentiation est une matrice identité de dimensions `nbdirsmax` × `nbdirsmax`. C'est le cas rencontré en annexe D

Annexe D : Fonction Matlab utilisée pour la détermination des singularités du mécanisme double-chaînes

Le code présenté ici est exploité pour la détermination et la classification des singularités du mécanisme double-chaînes avec le logiciel Diamanlab. Il s'agit de la fonction Matlab implémentant le code différentié obtenu par différentiation avec Tapenade, comme montré en annexe C. Les arguments de la fonction $R()$ de la classe `@UserSyst` sont :

- la variable `obj` de type `@UserSyst` qui contient en particulier l'ordre de troncature des polynômes K dans la méthode Diamant et le nombre d'inconnues n_{inc} du système à résoudre,
- la variable `U` qui est de type :
 - soit `@Taylor` lorsque la fonction est appelée par la fonction `ANMseries()`
 - soit `double` lorsque la fonction est appelée par la fonction `NRcorrection()` ou pour la classification d'une configuration singulière

Ce code contient :

- les paramètres géométriques du mécanisme ;
- l'inconnue U pour la continuation contenant les coordonnées \mathbf{q} et le vecteur ξ ;
- les directions de perturbation `ud`.

Il permet :

- d'intégrer et d'évaluer le modèle géométrique implicite et son code différentié ;
- d'assembler la matrice $\mathbf{R}_q^{\mathcal{L}}$;
- de décomposer $\mathbf{R}_q^{\mathcal{L}}$ pour obtenir les sous-matrices $\mathbf{R}_z^{\mathcal{L}}$, $\mathbf{R}_y^{\mathcal{L}}$ et $\mathbf{R}_w^{\mathcal{L}}$;
- d'assembler le système étendu (4.17) ou (4.18) sélectionné par la variable `extsys`.

La fonction retourne le résidu du système étendu sélectionné $\mathbf{R}^{\mathcal{L}z}(\mathbf{q}, \xi)$ ou $\mathbf{R}^{\mathcal{L}y}(\mathbf{q}, \xi)$, et la jacobienne $\mathbf{R}_q^{\mathcal{L}}(\mathbf{q})$.

Le code suivant est exploité pour la détermination et la classification des singularités avec le logiciel Diamanlab :

```

%% Code Matlab R.m pour la détermination des singularités du
    mécanisme double-chaînes
function [res,Rq]= R(obj,U)
global Ck % Ordre de troncature des polynomes
4 global extsys % Systeme des singularités 'inverse' ou 'directe'
%% Modele du mécanisme double-chaînes
% Topologie
nu = 2; % Nombre de coordonnées opérationnelles
nv = 2; % Nombre de coordonnées actives
9 nw = 4; % Nombre de coordonnées passives
nq = nv+nw+nu; % Nombre de coordonnées géométriques
% Paramètres géométriques
ab=1; bc=ab; de=bc; ad=2; cd=ad; fg=ad; cg=1.5; ef=3;

14 if isa(U,'Taylor') % initialisation de res de type Taylor
    res=U(obj.ninc-1)*0;
else % res est de type double pour le Newton-Raphson et la
    classification
    res=zeros(obj.ninc-1,1);
end
19 %% Inconnues diamanlab
thetaa=U(1); thetae=U(2); % Coordonnées actives
% Coordonnées passives
thetab=U(3); thetac=U(4); thetad=U(5); thetag=U(6);
px=U(13); py=U(14); % Coordonnées opérationnelles

24 %% Détermination des singularités
neqcontr=6; % Nombre d'équations de contraintes
% Vecteur dont l'existence implique la perte de rang
xi = U(7:12); % dim(neqcontr)
29 nbdirs=nv+nw; % Nombre de directions de différentiation
ud=eye(nbdirs,nq); % Matrice des directions de différentiation
% le nb de colonnes correspond au nombre de variables
    indépendantes considérées dans Tapenade
% une ligne est une direction de perturbation par rapport au
    vecteur de variables indépendantes

34 %% Code différentiel obtenu par Tapenade

```

```

% Ici on inclue les lignes d'interet converties en matlab et la
% boucle DO est geree par des tableaux

pxd = ud(:, 1); pyd = ud(:, 2);
thetaad = ud(:, 3); thetaed = ud(:, 4); thetabd = ud(:, 5);
39 thetacd = ud(:, 6); thetadd = ud(:, 7); thetagd = ud(:, 8);
% Equations de fermeture de chaines differentiees
eq1d = cd*thetadd*sin(thetad) - bc*thetabd*sin(thetab) - ab*
thetaad*sin(thetaa); % vecteur colonne dR1dq
eq2d = ab*thetaad*cos(thetaa) + bc*thetabd*cos(thetab) - cd*
thetadd*cos(thetad); % vecteur colonne dR2dq
eq3d = ef*thetaed*sin(thetae) - cg*thetacd*sin(thetac) - fg*
thetagd*sin(thetag) - cd*thetadd*sin(thetad); % vecteur
44 colonne dR3dq
eq4d = cd*thetadd*cos(thetad) + cg*thetacd*cos(thetac) + fg*
thetagd*cos(thetag) - ef*thetaed*cos(thetae); % vecteur
colonne dR4dq
eq5d = -(cd*thetadd*sin(thetad)) - cg*thetacd*sin(thetac) -
pxd; % vecteur colonne dR5dq
eq6d = cd*thetadd*cos(thetad) + cg*thetacd*cos(thetac) - pyd;
% vecteur colonne dR6dq
% Equations de fermeture de chaines R(q)=0
eq1=ab*cos(thetaa)+bc*cos(thetab)-cd*cos(thetad)-ad; % R1
49 eq2=ab*sin(thetaa)+bc*sin(thetab)-cd*sin(thetad); % R2
eq3=cd*cos(thetad)+cg*cos(thetac)+fg*cos(thetag)-ef*cos(thetae)-de
; % R3
eq4=cd*sin(thetad)+cg*sin(thetac)+fg*sin(thetag)-ef*sin(thetae); %
R4
eq5=cd*cos(thetad)+cg*cos(thetac)-px; % R5
eq6=cd*sin(thetad)+cg*sin(thetac)-py; % R6
54 % Calcul de resd
resd1 = eq1d; resd2 = eq2d; resd3 = eq3d;
resd4 = eq4d; resd5 = eq5d; resd6 = eq6d;
res(1)=eq1; res(2)=eq2; res(3)=eq3;
res(4)=eq4; res(5)=eq5; res(6)=eq6;
59 %% Assemblage de la matrice jacobienne Rq apparaissant dans le
modele cinematique implicite : Rq dq = 0
Rq = [resd1' ; resd2' ; resd3' ; resd4' ; resd5' ; resd6'];

```

```
Rw = Rq(:, nu+nv+1:nq); % R(q) différentie par rapport a w
Ry = [Rq(:, 1:nu) Rw]; % R(q) différentie par rapport a y=(u,w)
64 Rz = [Rq(:, nu+1:nu+nv) Rw]; % R(q) différentie par rapport a z=(v,
    w)

%% Assemblage du systeme etendu
switch extsys
    case 'directe'
69     res(7:12)=Ry'*xi;
    case 'inverse'
        res(7:12)=Rz'*xi;
end
res(13)=xi'*xi-1; % xi est un vecteur unitaire
74 end
```

Annexe E : Code Matlab pour l'analyse de sensibilité de haut-degré d'un mécanisme d'amplification

Le code suivant est exploité pour l'analyse de sensibilité de haut-degré du mécanisme d'amplification de l'actionneur piézoélectrique CEDRAT APA900M. Les données numériques sont issues du tableau 5.1. Ce code permet le calcul des polynômes de Taylor y_{TL} représentant la sensibilité de la configuration du mécanisme à une perturbation dy_0 de la hauteur initiale y_0 reportée sur le paramètre géométrique L . Les polynômes sont calculés au moyen de la DA par surcharge d'opérateur implémentée dans la classe @Taylor et évalués pour la perturbation Ld au moyen de la fonction `evalseries`. Le diagramme représentant la configuration approchée (x, y_{Ld1}) dans la plage de fonctionnement de l'actionneur, La figure 5.8(a) est obtenue par tracé de la variable y_{Ld1} en fonction de la variable x

```
1 %% Code Matlab pour l'analyse de sensibilité de haut-degré de l'  
    actionneur APA900M  
    global Ck  
    Ck=16 % Ordre de troncature des polynomes  
  
    %% Point de fonctionnement nominal  
6 L = 1; % Etude normalisee  
    theta0=2.469/180*pi; x0=L*cos(theta0); y0=L*sin(theta0);  
  
    % Plage de fonctionnement  
    ymaxdisp=0.4920*y0; % Deplacement maximal de la sortie du  
        mecanisme  
11 yinf=y0-ymaxdisp; % Borne inferieure de la plage pour y  
    xinf = x0; % Borne inferieure de la plage pour x  
    xsup = sqrt(L.^2 - yinf.^2); % Course maximale de la ceramique
```

170 Code Matlab pour l'analyse de sensibilité de haut-degré d'un mécanisme d'amplification

```
%% Perturbation – Erreur de fabrication sur y0
16 % Point de fonctionnement
dy0=0.004*y0; % Perturbation sur la hauteur initiale
y0dy0=y0+dy0;
LdL = sqrt(x0^2+y0dy0^2); % x0 n'est pas perturbe
Ld=LdL-L % Perturbation reportee sur le parametre L
21
% Domaine de variation etudie
xmin=0.9999; xmax=L; ymin=-0.01; ymax=0.05; nbpts=1000;
x = linspace(xmin,xmax,nbpts);
26 %% DA de haut-degre par surcharge d'operateur par rapport a L %%
Lt = Taylor(Ck,ones(size(x)).*L); % Declaration de la variable
    Taylor associee a la variable L
Lt = set(Lt,'coef1',1); % Initialisation des coefficients d'ordre
    1
%% Modele geometrique et sensibilite
ytL = (Lt.^2-x.^2).^(1/2); % Calcul du polynome de Taylor y(L)
31
%% Evaluation du polynome de Taylor pour la perturbation Ld
yLd16=evalseries(ytL,Ld,16); % Ordre 16
yLd4=evalseries(ytL,Ld,4); % Ordre 4
yLd2=evalseries(ytL,Ld,2); % Ordre 2
36 yLd1=evalseries(ytL,Ld,1); % Ordre 1
hold on
plot(x,get(ytL,'value'));
plot(x,yLd1); plot(x,yLd2); plot(x,yLd4); plot(x,yLd16);
axis([xmin xmax ymin ymax])
41 end
```

Bibliographie

- Abdel-Malek, K. 1995, «Dexterity of manipulator arms at an operating point», dans *Proceedings of the 21st ASME Advances in Design Automation*, vol. 82, ASME, p. 781–788.
- Abdel-Malek, K. et H.-J. Yeh. 1997, «Analytical boundary of the workspace for general 3-DOF mechanisms», *International Journal of Robotics Research*, vol. 16, n° 2, p. 198–213.
- Adept. 2017, «Cobra i600», http://www.adept.com/products/robots/scara/cobra-i600/downloads/doc_view/177-adept-cobra-i600-image. En ligne ; atteint 03 Avril 2017.
- Agrawal, S. 1991, «Workspace boundaries of in-parallel manipulator systems», dans , *Fifth International Conference on Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR*, p. 1147–1152 vol.2, doi :10.1109/ICAR.1991.240401.
- Ahmad, S. et S. Luo. 1988, «Analysis of kinematic singularities for robot manipulators in Cartesian coordinate parameters», dans , *1988 IEEE International Conference on Robotics and Automation, 1988. Proceedings*, p. 840–845 vol.2, doi :10.1109/ROBOT.1988.12165.
- Alciatore, D. et C. Ng. 1994, «Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation», *ASME Robotics : Kinematics, Dynamics and Controls*, vol. 72, p. 141–146.
- Allgower, E. L. et K. Georg. 1990, *Numerical continuation methods*, vol. 33, Springer-Verlag Berlin.
- Amine, S., M. Tale Masouleh, S. Caro, P. Wenger et C. Gosselin. 2012, «Singularity analysis of 3t2r parallel mechanisms using Grassmann–Cayley algebra and Grassmann geometry», *Mechanism and Machine Theory*, vol. 52, p. 326–340, doi :10.1016/j.mechmachtheory.2011.11.015.
- Andreff, N. 2006, «Des droites et des robots. modélisation, identification et commande référencées vision des machines complexes», Thèse d'habilitation à diriger des recherches, Université Blaise Pascal, Clermont-Ferrand.
- Arsenault, M. et R. Boudreau. 2004, «The Synthesis of Three-Degree-of-Freedom Planar Parallel Mechanisms with Revolute Joints (3-RRR) for an Optimal Singularity-Free Workspace», *Journal of Robotic Systems*, vol. 21, n° 5, p. 259–274, doi :10.1002/rob.20013.
- Autodiff. 2017, «www.Autodiff.org - AD Tools for MATLAB», <http://www.autodiff.org/?module=Tools&language=MATLAB>. En ligne ; atteint 21 Avril 2017.

- Azrar, L., B. Cochelin, N. Damil et M. Potier-Ferry. 1993, «An asymptotic-numerical method to compute the postbuckling behaviour of elastic plates and shells», *International Journal for Numerical Methods in Engineering*, vol. 36, n° 8, p. 1251–1277, doi :10.1002/nme.1620360802.
- Bachta, W., P. Renaud, E. Laroche et J. Gangloff. 2009, «Cardiolock2 : Parallel singularities for the design of an active heart stabilizer», dans *2009 IEEE International Conference on Robotics and Automation*, p. 3839–3844, doi :10.1109/ROBOT.2009.5152483.
- Barrio, R. 2006, «Sensitivity Analysis of ODES/DAES Using the Taylor Series Method», *SIAM Journal on Scientific Computing*, vol. 27, n° 6, p. 1929–1947, doi :10.1137/030601892.
- Ben-Horin, P. et M. Shoham. 2006, «Singularity condition of six-degree-of-freedom three-legged parallel robots based on grassmann-cayley algebra», *IEEE Transactions on Robotics*, vol. 22, n° 4, p. 577–590, doi :10.1109/TRO.2006.878958.
- Berz, M. 1987, «The differential algebra FORTRAN precompiler DAFOR», cahier de recherche AT-3 : TN-87-32, Los Alamos National Laboratory.
- Berz, M. et G. Hoffstätter. 1998, «Computation and Application of Taylor Polynomials with Interval Remainder Bounds», *Reliable Computing*, vol. 4, n° 1, p. 83–97, doi :10.1023/A:1009958918582.
- Bilasse, M., I. Charpentier, E. M. Daya et Y. Koutsawa. 2009, «A generic approach for the solution of nonlinear residual equations. Part II : Homotopy and complex nonlinear eigenvalue method», *Computer Methods in Applied Mechanics and Engineering*, vol. 198, n° 49–52, p. 3999–4004, doi :10.1016/j.cma.2009.09.015.
- Binaud, N., P. Cardou, S. Caro et P. Wenger. 2010a, «The kinematic sensitivity of robotic manipulators to joint clearances», dans *ASME 2010 International Design engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, p. 1371–1380.
- Binaud, N., S. Caro et P. Wenger. 2010b, «Comparison of 3-RPR planar parallel manipulators with regard to their kinetostatic performance and sensitivity to geometric uncertainties», *Meccanica*, vol. 46, n° 1, p. 75–88, doi :10.1007/s11012-010-9390-0.
- Boehler, Q., I. Charpentier, M. S. Vedrines et P. Renaud. 2015, «Definition and Computation of Tensegrity Mechanism Workspace», *Journal of Mechanisms and Robotics*, vol. 7, n° 4, p. 044 502–044 502, doi :10.1115/1.4029809.
- Bohigas, O. 2013, «Numerical computation and avoidance of manipulator singularities», Thèse de doctorat, Universitat Politècnica de Catalunya, <http://digital.csic.es/handle/10261/98394>.
- Bohigas, O., M. E. Henderson, L. Ros, M. Manubens et J. M. Porta. 2013, «Planning singularity-free paths on closed-chain manipulators», *IEEE Transactions on Robotics*, vol. 29, p. 888–898.

- Bohigas, O., M. Manubens et L. Ros. 2012a, «A complete method for workspace boundary determination on general structure manipulators», *IEEE Transactions on Robotics*, vol. 28, p. 993–1006.
- Bohigas, O., D. Zlatanov, M. Manubens et L. Ros. 2012b, «On the numerical classification of the singularities of robot manipulators», dans *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 4, American Society of Mechanical Engineers, p. 1287–1296.
- Bohigas, O., D. Zlatanov, L. Ros, M. Manubens et J. Porta. 2014, «A General Method for the Numerical Computation of Manipulator Singularity Sets», *IEEE Transactions on Robotics*, vol. 30, n° 2, p. 340–351, doi :10.1109/TRO.2013.2283416.
- Bonev, I. et C. Gosselin. 2006, «Analytical determination of the workspace of symmetrical spherical parallel mechanisms», *IEEE Transactions on Robotics*, vol. 22, n° 5, p. 1011–1017, doi :10.1109/TRO.2006.878983.
- Bonev, I. A. 2002, *Geometric analysis of parallel mechanisms*, thèse de doctorat, Université Laval Canada, Québec.
- Bonev, I. A. et J. Ryu. 2001, «A new approach to orientation workspace analysis of 6-DOF parallel manipulators», *Mechanism and Machine Theory*, vol. 36, n° 1, p. 15–28, doi :10.1016/S0094-114X(00)00032-X.
- Bonev, I. A., D. Zlatanov et C. M. Gosselin. 2003, «Singularity Analysis of 3-DOF Planar Parallel Mechanisms via Screw Theory», *Journal of Mechanical Design*, vol. 125, n° 3, p. 573, doi :10.1115/1.1582878.
- Borras, J., F. Thomas et C. Torras. 2008, «Architecture singularities in flagged parallel manipulators», dans *2008 IEEE International Conference on Robotics and Automation*, p. 3844–3850, doi :10.1109/ROBOT.2008.4543801.
- Boudaoud, H., A. Lejeune, M. Potier-Ferry, H. Zahrouni et I. Charpentier. 2009, «Différentiation automatique et MAN : Application au contact unilatéral», dans *9e Colloque national en calcul des structures*, CSMA, Giens, France.
- Breteler, A. J. K. 2011, «On the Development of Terminology and an Electronic Dictionary for Mechanism and Machine Science», dans *Technology Developments : the Role of Mechanism and Machine Science and IFToMM*, édité par M. Ceccarelli, n° 1 dans *Mechanisms and Machine Science*, Springer Netherlands, p. 95–105, doi : 10.1007/978-94-007-1300-0_7.
- Béchet, F., A. Lejeune et M. Potier-Ferry. 2010, «Taylor series to solve friction problems», *Comptes Rendus Mécanique*, vol. 338, n° 6, p. 327–332, doi :10.1016/j.crme.2010.05.003.
- Cardou, P., S. Bouchard et C. Gosselin. 2010, «Kinematic-Sensitivity Indices for Dimensionally Nonhomogeneous Jacobian Matrices», *IEEE Transactions on Robotics*, vol. 26, n° 1, p. 166–173, doi :10.1109/TRO.2009.2037252.
- Caro, S. 2004, *Conception robuste de mécanismes*, thèse de doctorat, Université de Nantes.

- Caro, S., N. Binaud et P. Wenger. 2009, «Sensitivity Analysis of 3-RPR Planar Parallel Manipulators», *Journal of Mechanical Design*, vol. 131, n° 12, p. 121 005–121 005, doi : 10.1115/1.4000216.
- Caro, S., P. Wenger, F. Bennis et D. Chablat. 2005, «Sensitivity Analysis of the Orthoglide : A Three-DOF Translational Parallel Kinematic Machine», *Journal of Mechanical Design*, vol. 128, n° 2, p. 392–402, doi :10.1115/1.2166852.
- Castellet, A. et F. Thomas. 1998, «An Algorithm for the Solution of Inverse Kinematics Problems Based on an Interval Method», dans *Advances in Robot Kinematics : Analysis and Control*, édité par J. Lenarčič et M. L. Husty, Springer Netherlands, doi :10.1007/978-94-015-9064-8_40.
- Castelli, G., E. Ottaviano et M. Ceccarelli. 2008, «A Fairly General Algorithm to Evaluate Workspace Characteristics of Serial and Parallel Manipulators», *Mechanics Based Design of Structures and Machines*, vol. 36, n° 1, p. 14–33, doi :10.1080/15397730701729478.
- CEDRAT. 2017, «Actionneurs piézoélectriques amplifiés», <http://www.cedrat-technologies.com/fr/mechatronic-products/actuators/apa.html>.
- Cervantes-Sánchez, J. J., J. M. Rico-Martínez, G. González-Montiel et E. J. González-Galván. 2009, «The differential calculus of screws: Theory, geometrical interpretation, and applications», *Proceedings of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science*, vol. 223, n° 6, p. 1449–1468, doi : 10.1243/09544062JMES1195.
- Chablat, D. et P. Wenger. 1998, «Working modes and aspects in fully parallel manipulators», dans *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, vol. 3, p. 1964–1969, doi :10.1109/ROBOT.1998.680601.
- Charpentier, I. 2012, «On higher-order differentiation in nonlinear mechanics», *Optimization Methods and Software*, vol. 27, doi :10.1080/10556788.2011.577775.
- Charpentier, I. et B. Cochelin. 2016, «A full higher-order AD continuation and bifurcation framework», dans *AD2016 - The 7th International Conference on Algorithmic Differentiation - Programme and Abstracts*, Christ Church Oxford, UK, p. 50–53.
- Charpentier, I., B. Cochelin et K. Lampoh. 2013, «Diamanlab - An interactive Taylor-based continuation tool in MATLAB», <https://hal.archives-ouvertes.fr/hal-00853599>.
- Charpentier, I., J.-P. Friedelmeyer et J. Gustedt. 2016, «Arbogast – Origine d’un outil de dérivation automatique», Rapport de recherche n° 8911, INRIA. <https://hal.inria.fr/hal-01313355/document>.
- Charpentier, I. et K. Lampoh. 2016, «Sensitivity computations in higher order continuation methods», *Applied Mathematical Modelling*, vol. 40, n° 4, p. 3365–3380, doi :10.1016/j.apm.2015.10.033.
- Charpentier, I. et M. Potier-Ferry. 2008, «Automatic differentiation of the asymptotic numerical method : the Diamant approach», *Comptes Rendus - Mécanique*, vol. 336, p. 336–340.

- Chedmail, P., E. Dombre et P. Wenger. 1998, *La CAO en robotique : outils et méthodologies*, Hermes.
- Chen, G., H. Wang et Z. Lin. 2013, «A unified approach to the accuracy analysis of planar parallel manipulators both with input uncertainties and joint clearance», *Mechanism and Machine Theory*, vol. 64, p. 1–17, doi :10.1016/j.mechmachtheory.2013.01.005.
- Chen, Y., F. Xie, X. Liu et Y. Zhou. 2014, «Error modeling and sensitivity analysis of a parallel robot with SCARA(selective compliance assembly robot arm) motions», *Chinese Journal of Mechanical Engineering*, vol. 27, n° 4, p. 693–702, doi :10.3901/CJME.2014.0423.082.
- Cochelin, B., N. Damil et M. Potier-Ferry. 1994, «Asymptotic–numerical methods and Pade approximants for non-linear elastic structures», *International Journal for Numerical Methods in Engineering*, vol. 37, n° 7, p. 1187–1213, doi :10.1002/nme.1620370706.
- Cochelin, B., N. Damil et M. Potier-Ferry. 2007, *Méthode asymptotique numérique*, Méthodes numériques, Paris : Hermes science publications : Lavoisier, cop. 2007 1 vol. (297 p.) : ill. ; 24 cm.
- Cochelin, B., N. Damil et M. Potier-Ferry. 2009, «Manlab - An interactive path-following and bifurcation analysis software», <http://manlab.lma.cnrs-mrs.fr/>.
- Corradi, D., S. Caro, D. Chablat et P. Cardou. 2014, «Assembly conditions of parallel manipulators considering geometric errors, joint clearances, link flexibility and joint elasticity», dans *2014 IEEE International Conference on Robotics and Automation (ICRA)*, p. 4067–4072, doi :10.1109/ICRA.2014.6907450.
- CSIC-UPC, I. 2017, «The CuikSuite documentation», http://www.iri.upc.edu/people/porta/Soft/CuikSuite2_1-Doc/html/3R_8world.html.
- Daya, E. M. et M. Potier-Ferry. 2001, «A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures», *Computers & Structures*, vol. 79, n° 5, p. 533–541, doi :10.1016/S0045-7949(00)00151-6.
- Dhooge, A., W. Govaerts et Y. A. Kuznetsov. 2004, «MATCONT : a Matlab package for numerical bifurcation analysis of ODEs», *ACM SIGSAM Bulletin*, vol. 38, n° 1, p. 21–22.
- Di Gregorio, R. et V. Parenti-Castelli. 1999, «Influence of the geometric parameters of the 3-UPU parallel manipulator on the singularity loci», dans *The International Workshop on Parallel Kinematics Machines, UCIMU, Milano*.
- Doedel, E. J., T. F. Fairgrieve, B. Sandstede, A. R. Champneys, Y. A. Kuznetsov et X. Wang. 2007, «AUTO-07p : Continuation and bifurcation software for ordinary differential equations», cahier de recherche.
- Du, D. et I. Hwang. 2007, «A Computational Approach to Solve Optimal Control Problems Using Differential Transformation», dans *2007 American Control Conference*, p. 2322–2327, doi :10.1109/ACC.2007.4282305.
- Dürrbaum, A., W. Klier et H. Hahn. 2002, «Comparison of Automatic and Symbolic Differentiation in Mathematical Modeling and Computer Simulation of Rigid-Body Systems», *Multibody System Dynamics*, vol. 7, n° 4, p. 331–355, doi :10.1023/A:1015523018029.

- Emeliyanenko, P., E. Berberich et M. Sagraloff. 2009, «Visualizing Arcs of Implicit Algebraic Curves, Exactly and Fast», dans *Advances in Visual Computing*, Springer, Berlin, Heidelberg, p. 608–619, doi :10.1007/978-3-642-10331-5_57.
- FarzanehKaloorazi, M., M. T. Masouleh et S. Caro. 2016, «Collision-free workspace of parallel mechanisms based on an interval analysis approach», *Robotica*, p. 1–14, doi : 10.1017/S0263574716000497.
- Fichter, E. 1986, «A Stewart Platform- Based Manipulator: General Theory and Practical Construction», *The International Journal of Robotics Research*, vol. 5, n° 2, p. 157–182, doi :10.1177/027836498600500216.
- Gogu, G. 2008, «Kinematic analysis», dans *Structural Synthesis of Parallel Robots*, n° 149 dans *Solid Mechanics and its Applications*, Springer Netherlands, doi :10.1007/978-1-4020-5710-6_4.
- Gosselin, C. 1990, «Determination of the Workspace of 6-DOF Parallel Manipulators», *Journal of Mechanical Design*, vol. 112, n° 3, p. 331–336, doi :10.1115/1.2912612.
- Gosselin, C. et J. Angeles. 1990, «Singularity analysis of closed-loop kinematic chains», *IEEE Transactions on Robotics and Automation*, vol. 6, n° 3, p. 281–290, doi :10.1109/70.56660.
- Gregorio, R. d. et V. Parenti-Castelli. 1999, «Mobility analysis of the 3-UPU parallel mechanism assembled for a pure translational motion», dans *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 1999. Proceedings*, p. 520–525, doi : 10.1109/AIM.1999.803224.
- Griewank, A., D. Juedes et J. Utke. 1996, «ADOL-C : a package for the automatic differentiation of algorithms written in C/C++», *ACM Transactions on Mathematical Software*, vol. 22, n° 2, p. 131–167, doi :10.1145/229473.229474.
- Griewank, A. et A. Walther. 2008, *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, Society for Industrial and Applied Mathematics.
- Guillaume, P. 1999, «Nonlinear Eigenproblems», *SIAM Journal on Matrix Analysis and Applications*, vol. 20, n° 3, p. 575–595, doi :10.1137/S0895479897324172.
- Guillaume, P. et M. Masmoudi. 1994, «Computation of high order derivatives in optimal shape design», *Numerische Mathematik*, vol. 67, n° 2, p. 231–250, doi :10.1007/s002110050025.
- Hao, F. et J. M. McCarthy. 1998, «Conditions for line-based singularities in spatial platform manipulators», *Journal of Robotic Systems*, vol. 15, n° 1, p. 43–55, doi :10.1002/(SICI)1097-4563(199812)15:1<43::AID-ROB4>3.0.CO;2-S.
- Hascoet, L. et V. Pascual. 2013, «The Tapenade automatic differentiation tool : Principles, model, and specification», *ACM Transactions on Mathematical Software*, vol. 39, p. 20 :1–20 :43.
- Haug, E. J. et P. E. Ehle. 1982, «Second-order design sensitivity analysis of mechanical system dynamics», *International Journal for Numerical Methods in Engineering*, vol. 18, n° 11, p. 1699–1717, doi :10.1002/nme.1620181109.

- Haug, E. J., C.-M. Luh, F. A. Adkins et J.-Y. Wang. 1996, «Numerical algorithms for mapping boundaries of manipulator workspaces», *Journal of Mechanical Design, Transactions of the ASME*, vol. 118, p. 228–234.
- Haug, E. J., J. Y. Wang et J. K. Wu. 1992, «Dextrous Workspaces of Manipulators. I. Analytical Criteria», *Mechanics of Structures and Machines*, vol. 20, n° 3, p. 321–361, doi :10.1080/08905459208905172.
- Henderson, M. E. 2007, «Higher-Dimensional Continuation», dans *Numerical Continuation Methods for Dynamical Systems*, édité par D. B. Krauskopf, D. H. M. Osinga et D. J. Galán-Vioque, Springer Netherlands, p. 77–115.
- Hentz, G., I. Charpentier et P. Renaud. 2016a, «Higher-order continuation for the determination of robot workspace boundaries», *Comptes Rendus Mécanique*, vol. 344, n° 2, p. 95–101, doi :10.1016/j.crme.2015.10.001.
- Hentz, G., I. Charpentier, P. Renaud et L. Rubbert. 2016b, «A taylor-based continuation method for the determination and classification of robot singularities», dans *Advances in Robot Kinematics*, édité par J. Lenarčič et J.-P. Merlet, France, p. 419–426.
- Hoefkens, J., M. Berz et K. Makino. 2002, «Efficient High-Order Methods for ODEs and DAEs», dans *Automatic Differentiation of Algorithms*, édité par G. Corliss, C. Faure, A. Griewank, L. Hascoët et U. Naumann, Springer New York, doi :10.1007/978-1-4613-0075-5_41.
- Hou, G., A. Satyanarayana et S. Tiwari. 1998, «First- and second-order sensitivity analysis of finite element equations via automatic differentiation», dans *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics.
- Howell, L. L. 2001, *Compliant mechanisms*, Wiley.
- Hubert, J. 2010, *Manipulateurs parallèles, singularités et analyse statique*, thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- Hunt, K. H. 1978, *Kinematic Geometry of Mechanisms*, Clarendon Press.
- Jalón, J. G. d. et E. Bayo. 1994, «Dependent Coordinates and Related Constraint Equations», dans *Kinematic and Dynamic Simulation of Multibody Systems*, Mechanical Engineering Series, Springer New York, p. 16–70, doi :10.1007/978-1-4612-2600-0_2.
- Jamal, M., B. Braikat, S. Boutmir, N. Damil et M. Potier-Ferry. 2002, «A high order implicit algorithm for solving instationary non-linear problems», *Computational Mechanics*, vol. 28, n° 5, p. 375–380, doi :10.1007/s00466-002-0301-7.
- Jha, R., D. Chablat, F. Rouillier et G. Moroz. 2015, «Workspace and Singularity analysis of a Delta like family robot», *arXiv :1505.05388 [cs]*. ArXiv : 1505.05388.
- Jo, D.-Y. et E. J. Haug. 1989a, *Workspace analysis of closed-loop mechanisms with unilateral constraints*, Center for Simulation and Design Optimization and Department of Mechanical Engineering, University of Iowa.

- Jo, D.-Y. et E. J. Haug. 1989b, «Workspace Analysis of Multibody Mechanical Systems Using Continuation Methods», *Journal of Mechanical Design*, vol. 111, n° 4, p. 581–589, doi :10.1115/1.3259040.
- Joshi, S. A. et L.-W. Tsai. 2002, «Jacobian Analysis of Limited-DOF Parallel Manipulators», dans *27th Biennial Mechanisms and Robotics Conference*, vol. 5, ASME, Montreal, Quebec, Canada, p. 341–348, doi :10.1115/DETC2002/MECH-34238.
- Keller, H. B. 1987, «Lectures on numerical methods in bifurcation problems», *Applied Mathematics*, vol. 217, p. 50.
- Khalil, W. et E. Dombre. 2002, *Modelisation, Identification and Control of Robots*.
- Kolchuzhin, V., J. Mehner, T. Gessner et W. Doetze. 2007, «Application of Higher Order Derivatives Method to Parametric Simulation of MEMS», dans *2007 International Conference on Thermal, Mechanical and Multi-Physics Simulation Experiments in Microelectronics and Micro-Systems. EuroSime 2007*, p. 1–6, doi :10.1109/ESIME.2007.359962.
- Kong, X. et C. M. Gosselin. 2001, «Uncertainty Singularity Analysis of Parallel Manipulators Based on the Instability Analysis of Structures», *The International Journal of Robotics Research*, vol. 20, n° 11, p. 847–856, doi :10.1177/02783640122068146.
- Koutsawa, Y., I. Charpentier, E. M. Daya et M. Cherkaoui. 2008, «A generic approach for the solution of nonlinear residual equations. part I : the Diamant toolbox», *Computer Methods in Applied Mechanics and Engineering*, vol. 198, p. 572–577.
- Krawczyk, R. 1969, «Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehler-schranken», *Computing*, vol. 4, n° 3, p. 187–201, doi :10.1007/BF02234767.
- Kumar, A. et K. J. Waldron. 1981, «The Workspaces of a Mechanical Manipulator», *Journal of Mechanical Design*, vol. 103, n° 3, p. 665–672, doi :10.1115/1.3254968.
- Lai, Z. C. et D. C. H. Yang. 1986, «A New Method for the Singularity Analysis of Simple Six-link Manipulators», *The International Journal of Robotics Research*, vol. 5, n° 2, p. 66–74, doi :10.1177/027836498600500207.
- Lampoh, K. 2012, *Différentiation automatique de codes mécaniques : application à l'analyse de sensibilité des tôles sandwich aux paramètres de modélisation*, thèse de doctorat, Université de Lorraine.
- Lampoh, K., I. Charpentier et E. M. Daya. 2011, «A generic approach for the solution of nonlinear residual equations. Part III : Sensitivity computations», *Computer Methods in Applied Mechanics and Engineering*, vol. 200, n° 45–46, p. 2983–2990, doi :10.1016/j.cma.2011.06.009.
- Litvin, F. L. 1980, «Application of theorem of implicit function system existence for analysis and synthesis of linkages», *Mechanism and Machine Theory*, vol. 15, p. 115–125.
- Ma, O. et J. Angeles. 1991, «Architecture singularities of platform manipulators», dans *1991 IEEE International Conference on Robotics and Automation Proceedings*, p. 1542–1547 vol.2, doi :10.1109/ROBOT.1991.131835.

- Makino, K. et M. Berz. 2006, «COSY INFINITY Version 9», *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 558, n° 1, p. 346–350, doi :10.1016/j.nima.2005.11.109.
- Masory, O. et J. Wang. 1994, «Workspace evaluation of Stewart platforms», *Advanced Robotics*, vol. 9, n° 4, p. 443–461, doi :10.1163/156855395X00508.
- McCarthy, J. M. 2013, «Polynomials, Computers, and Kinematics for the 21st Century», dans *21st Century Kinematics*, édité par J. M. McCarthy, Springer London, p. 1–12.
- Merlet, J.-P. 1989, «Singular Configurations of Parallel Manipulators and Grassmann Geometry», *The International Journal of Robotics Research*, vol. 8, n° 5, p. 45–56, doi : 10.1177/027836498900800504.
- Merlet, J.-P. 1992, «Geometrical determination of the workspace of a constrained parallel manipulator», *Advances in Robot Kinematics, Ferrare, Italy*, p. 7–9.
- Merlet, J.-P. 1995, «Determination of the orientation workspace of parallel manipulators», *Journal of Intelligent and Robotic Systems*, vol. 13, n° 2, p. 143–160, doi :10.1007/BF01254849.
- Merlet, J.-P. 1999, «Determination of 6D-workspaces of Gough-type parallel manipulator and comparison between different geometries», *The International Journal of Robotics Research*, vol. 18, p. 902–916.
- Merlet, J. P. 2002a, «The need for a systematic methodology for the evaluation and optimal design of parallel manipulators», dans *3rd Chemnitzer Parallelkinematik Seminar*, p. 49–62.
- Merlet, J.-P. 2002b, «Still a long way to go on the road for parallel mechanisms», dans *ASME 27th Biennial Mechanisms and Robotics Conference*.
- Merlet, J. P. 2005a, «Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots», *Journal of Mechanical Design*, vol. 128, n° 1, p. 199–206, doi :10.1115/1.2121740.
- Merlet, J.-P. 2005b, «Optimal design of robots», dans *Robotics : Science and systems*.
- Merlet, J.-P. 2006, *Parallel robots*, 2^e éd., n° 74 dans *Solid mechanics and its applications*, Kluwer Academic Publishers.
- Merlet, J.-P. 2007, «A formal-numerical approach for robust in-workspace singularity detection», *IEEE Transactions on Robotics*, vol. 23, n° 3, p. 393–402.
- Merlet, J.-P. et D. Daney. 2005, «Dimensional Synthesis of Parallel Robots with a Guaranteed Given Accuracy over a Specific Workspace», dans *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, p. 942–947, doi :10.1109/ROBOT.2005.1570238.
- Merlet, J.-P. et C. Gosselin. 2008, «Parallel Mechanisms and Robots», dans *Springer Handbook of Robotics*, édité par B. S. Prof et O. K. Prof, Springer Berlin Heidelberg, p. 269–285.

- Merlet, J.-P., C. M. Gosselin et N. Mouly. 1998, «Workspaces of planar parallel manipulators», *Mechanism and Machine Theory*, vol. 33, n° 1, p. 7–20, doi :10.1016/S0094-114X(97)00025-6.
- Monsarrat, B. et C. M. Gosselin. 2003, «Workspace analysis and optimal design of a 3-leg 6-DOF parallel platform mechanism», *IEEE Transactions on Robotics and Automation*, vol. 19, n° 6, p. 954–966, doi :10.1109/TRA.2003.819603.
- Moore, R. 1979, *Methods and Applications of Interval Analysis*, Studies in Applied and Numerical Mathematics, Society for Industrial and Applied Mathematics. DOI : 10.1137/1.9781611970906 DOI : 10.1137/1.9781611970906.
- Moore, R. E. 1966, *Interval analysis*, Prentice-Hall, Englewood Cliffs, N.J. OCLC : 383416.
- Morgan, A. P., A. J. Sommese et C. W. Wampler. 1992, «A power series method for computing singular solutions to nonlinear analytic systems», *Numerische Mathematik*, vol. 63, n° 1, p. 391–409, doi :10.1007/BF01385867.
- Neumaier, A. 1990, *Interval Methods for Systems of Equations*, Cambridge University Press.
- Nik, H. S., S. Effati et A. Yildirim. 2013, «Solution of linear optimal control systems by differential transform method», *Neural Computing and Applications*, vol. 23, n° 5, p. 1311–1317, doi :10.1007/s00521-012-1073-4.
- Oetomo, D., D. Daney, B. Shirinzadeh et J. P. Merlet. 2008, «Certified workspace analysis of 3rrr planar parallel flexure mechanism», dans *2008 IEEE International Conference on Robotics and Automation*, p. 3838–3843, doi :10.1109/ROBOT.2008.4543800.
- Ozaki, I., F. Kimura et M. Berz. 1995, «Higher-order sensitivity analysis of finite element method by automatic differentiation», *Computational Mechanics*, vol. 16, n° 4, p. 223–234, doi :10.1007/BF00369867.
- Park, F. et J. W. Kim. 1999, «Singularity analysis of closed kinematic chains», *Journal of mechanical design*, vol. 121, n° 1, p. 32–38.
- Pashkevich, A., D. Chablat et P. Wenger. 2006, «Kinematics and workspace analysis of a three-axis parallel manipulator : the Orthoglide», *Robotica*, vol. 24, n° 01, p. 39–49, doi :10.1017/S0263574704000347.
- Pickard, J. K., J. A. Carretero et J.-P. Merlet. 2016, «Accounting for tolerances in the design parameters of the 3-RRR», .
- Porta, J., L. Ros et F. Thomas. 2009, «A Linear Relaxation Technique for the Position Analysis of Multiloop Linkages», *IEEE Transactions on Robotics*, vol. 25, n° 2, p. 225–239, doi :10.1109/TRO.2008.2012337.
- Porta, J. M., L. Ros, O. Bohigas, M. Manubens, C. Rosales et L. Jaillet. 2014, «The CUIK suite : Analyzing the motion of closed-chain multibody systems», *IEEE Robotics Automation Magazine*, vol. 21, p. 105–114.
- Qu, H., Y. Fang et S. Guo. 2012, «Parasitic rotation evaluation and avoidance of 3-UPU parallel mechanism», *Frontiers of Mechanical Engineering*, vol. 7, n° 2, p. 210–218, doi :10.1007/s11465-012-0317-4.

- Rastegar, J. et D. Perel. 1990, «Generation of Manipulator Workspace Boundary Geometry Using the Monte Carlo Method and Interactive Computer Graphics», *Journal of Mechanical Design*, vol. 112, n° 3, p. 452–454, doi :10.1115/1.2912630.
- Ratschek, H. et J. Rokne. 2005, «Scci-hybrid methods for 2d curve tracing», *International Journal of Image and Graphics*, vol. 05, n° 03, p. 447–479, doi :10.1142/S0219467805001859.
- Rezaei, A. et A. Akbarzadeh. 2015, «Study on Jacobian, singularity and kinematics sensitivity of the FUM 3-PSP parallel manipulator», *Mechanism and Machine Theory*, vol. 86, p. 211–234, doi :10.1016/j.mechmachtheory.2014.11.009.
- Roth, B. et F. Freudenstein. 1963, «Synthesis of Path-Generating mechanisms by numerical methods», *Journal of Engineering for Industry*, vol. 85, p. 298–304.
- Rubbert, L., S. Caro, J. Gangloff et P. Renaud. 2014, «Using Singularities of Parallel Manipulators to Enhance the Rigid-Body Replacement Design Method of Compliant Mechanisms», *Journal of Mechanical Design*, vol. 136, n° 5, p. 051 010–051 010, doi : 10.1115/1.4026949.
- Rubbert, L., S. Caro, P. Renaud et J. Gangloff. 2012, «A Planar Compliant Mechanism with RRP Mobilities Based on the Singularity Analysis of a 3-US Parallel Mechanism», dans *Latest Advances in Robot Kinematics*, édité par J. Lenarčič et M. Husty, Springer Netherlands, p. 381–388, doi :10.1007/978-94-007-4620-6_48.
- Rubbert, L., I. Charpentier, S. Henein et P. Renaud. 2017, «Higher-order continuation method for the rigid-body kinematic design of compliant mechanisms», *Precision Engineering*, doi :10.1016/j.precisioneng.2017.06.021. Disponible en ligne.
- Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana et S. Tarantola. 2008, *Global sensitivity analysis : the primer*, John Wiley & Sons.
- Seydel, R. 1979, «Numerical computation of branch points in nonlinear equations», *Numerische Mathematik*, vol. 33, n° 3, p. 339–352, doi :10.1007/BF01398649.
- Seydel, R. 2010, *Practical Bifurcation and Stability Analysis, Interdisciplinary Applied Mathematics*, vol. 5, 3^e éd., Springer New York, New York, doi :10.1007/978-1-4419-1740-9.
- Simas, H. et R. Di Gregorio. 2016, «Geometric Error Effects on Manipulators' Positioning Precision : A General Analysis and Evaluation Method», *Journal of Mechanisms and Robotics*, vol. 8, n° 6, p. 061 016.
- Snyman, J. A., L. J. du Plessis et J. Duffy. 1998, «An optimization approach to the determination of the boundaries of manipulator workspaces», *Journal of Mechanical Design*, vol. 122, p. 447–456.
- Sommer, H., C. Pradalier et P. Furgale. 2016, «Automatic Differentiation on Differentiable Manifolds as a Tool for Robotics», dans *Robotics Research*, édité par M. Inaba et P. Corke, n° 114 dans Springer Tracts in Advanced Robotics, Springer International Publishing, p. 505–520, doi :10.1007/978-3-319-28872-7_29.

- Sommese, A. J., J. Verschelde et C. W. Wampler. 2004, «Advances in polynomial continuation for solving problems in kinematics», *Journal of Mechanical Design*, vol. 126, p. 262–268.
- Sugimoto, K., J. Duffy et K. H. Hunt. 1982, «Special configurations of spatial mechanisms and robot arms», *Mechanism and Machine Theory*, vol. 17, n° 2, p. 119–132, doi : 10.1016/0094-114X(82)90042-8.
- TaarLab. 2017, «3-RRR», <http://www.taarlab.com/en/images/PM/3RRR.jpg>. En ligne ; atteint 27 Février 2017.
- Tannous, M., S. Caro et A. Goldsztejn. 2014, «Sensitivity analysis of parallel manipulators using an interval linearization method», *Mechanism and Machine Theory*, vol. 71, p. 93–114, doi :10.1016/j.mechmachtheory.2013.09.004.
- Thompson, J. M. T. et A. C. Walker. 1968, «The non-linear perturbation analysis of discrete structural systems», *International Journal of Solids and Structures*, vol. 4, n° 8, p. 757–768, doi :10.1016/0020-7683(68)90054-1.
- Tsai, L. et A. P. Morgan. 1985, «Solving the kinematics of the most general six- and Five-Degree-of-Freedom manipulators by continuation methods», *Journal of Mechanical Design*, vol. 107, p. 189–200.
- Tsai, L.-W. 1998, «The Jacobian Analysis of a Parallel Manipulator Using Reciprocal Screws», dans *Advances in Robot Kinematics : Analysis and Control*, édité par J. Lenarčič et M. L. Husty, Springer Netherlands, p. 327–336. DOI : 10.1007/978-94-015-9064-8_33.
- Tsukanov, I. et M. Hall. 2003, «Data structure and algorithms for fast automatic differentiation», *International Journal for Numerical Methods in Engineering*, vol. 56, n° 13, p. 1949–1972, doi :10.1002/nme.647.
- Van Hentenryck, P., D. McAllester et D. Kapur. 1997, «Solving Polynomial Systems Using a Branch and Prune Approach», *SIAM Journal on Numerical Analysis*, vol. 34, n° 2, p. 797–827, doi :10.1137/S0036142995281504.
- Van Keulen, F., R. T. Haftka et N. H. Kim. 2005, «Review of options for structural design sensitivity analysis. Part 1 : Linear systems», *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n° 30–33, p. 3213–3243, doi :10.1016/j.cma.2005.02.002.
- Voglewede, P. A. et I. Ebert-Uphoff. 2004, «Measuring "closeness" to singularities for parallel manipulators», dans *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 5, p. 4539–4544 Vol.5, doi :10.1109/ROBOT.2004.1302433.
- Wampler, C. W., A. P. Morgan et A. J. Sommese. 1990, «Numerical continuation methods for solving polynomial systems arising in kinematics», *Journal of Mechanical Design*, vol. 112, p. 59–68.
- Wampler, C. W. et A. J. Sommese. 2013, «Applying Numerical Algebraic Geometry to Kinematics», dans *21st Century Kinematics*, édité par J. M. McCarthy, Springer London, p. 125–159.

- Watson, L. T., S. C. Billups et A. P. Morgan. 1990, «HOMPACK : A Suite of Codes for Globally Convergent Homotopy Algorithms», cahier de recherche, Virginia Polytechnic Institute & State University, Blacksburg, VA, USA.
- Wenger, P. et P. Chedmail. 1991, «On the connectivity of manipulator free workspace», *Journal of Robotic Systems*, vol. 8, n° 6, p. 767–799, doi :10.1002/rob.4620080604.
- Wu, W. et S. S. Rao. 2004, «Interval Approach for the Modeling of Tolerances and Clearances in Mechanism Analysis», *Journal of Mechanical Design*, vol. 126, n° 4, p. 581–592, doi : 10.1115/1.1760775.
- Yang, X., H. Wang, C. Zhang et K. Chen. 2010, «A method for mapping the boundaries of collision-free reachable workspaces», *Mechanism and Machine Theory*, vol. 45, n° 7, p. 1024–1033, doi :10.1016/j.mechmachtheory.2010.02.002.
- Zlatanov, D. 1998, «Generalized singularity analysis of mechanisms», Thèse de doctorat, University of Toronto.
- Zlatanov, D., I. Bonev et C. Gosselin. 2002, «Constraint singularities of parallel mechanisms», dans *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, vol. 1, p. 496–502, doi :10.1109/ROBOT.2002.1013408.
- Zlatanov, D., R. G. Fenton et B. Benhabib. 1995, «A Unifying Framework for Classification and Interpretation of Mechanism Singularities», *Journal of Mechanical Design*, vol. 117, n° 4, p. 566–572, doi :10.1115/1.2826720.
- Zlatanov, D., R. G. Fenton et B. Benhabib. 1998, «Identification and classification of the singular configurations of mechanisms», *Mechanism and Machine Theory*, vol. 33, n° 6, p. 743–760, doi :10.1016/S0094-114X(97)00053-0.

Index terminologique

- Analyse de sensibilité, 123
- Analytique, Fonction, 44
- Approche de conception, 10, 11
- Architecture de mécanisme, 7
- Attribut (informatique), 56
- Bâti, 7
- Besoin pour l'évaluation, 11
- Boîte, 76
- Branch-and-bound, Méthode, 76
- Branch-and-prune, Méthode, 78
- Branche de solutions, 35, 37
- Caractéristique d'un mécanisme, 10–12
- Chaîne cinématique, 6
- Changement de branche, 37
- Classification des singularités, 100, 110
- Code différencié, 53
- Code source, 53
- Configuration du mécanisme, 11
- Configuration admissible, 20
- Constructeur (informatique), 56
- Continuation numérique, Méthode de, 36, 37
- Continuation de haut-degré, 37
- Continuation interactive, 60
- Coordonnées actives, 11, 19
- Coordonnées généralisées, 18
- Coordonnées géométriques, 19
- Coordonnées opérationnelles, 19
- Coordonnées passives, 19
- Critère de performance, 10
- Détection des singularités, 110
- Détection de point de bifurcation, 36
- Détermination des singularités, 110
- Diagramme de bifurcation, 35
- Diamant, Méthode, 32
- Différentiation Automatique (DA), 52
- DA de haut-degré (DAHD), 57, 136
- Effecteur, 7
- Efficacité d'évaluation, 11
- Erreur de fabrication, 132
- Espace des configurations, 20
- Espace de travail, 20
- Evaluation des performances du mécanisme, 10, 11
- Généricité de l'évaluation, 10, 11
- Identification des singularités, 100
- Itératif, processus de conception, 10, 11
- Jambe d'un mécanisme parallèle, 9
- Liaison, 6
- Mécanisme, 6
- Membre, 6
- Méthode (informatique), 56
- Mobilité du mécanisme, 105
- Modèle cinématique implicite, 24, 98, 112

- Modèle géométrique direct (MGD), 14, 130
- Modèle géométrique implicite, 17, 19, 112, 122
- Modèle géométrique inverse (MGI), 15
- Mouvement instantané, 24, 98
- Outils d'aide à la conception, 10, 11
- Parallèle, mécanisme, 8
- Paramètre de chemin, 40
- Paramètre géométrique, 122, 130
- Paramètre naturel, 39
- Point de bifurcation, 35
- Point de bifurcation simple, 35
- Point de fonctionnement, 130, 132
- Point de retournement, 35
- Précision d'évaluation, 11, 12, 24
- Problème cinématique direct (PCD), 98
- Problème cinématique inverse (PCI), 98
- Rapidité d'évaluation, 10, 11, 13
- Rapport d'amplification, 130
- Robustesse des performances, 11
- Sériel, mécanisme, 8
- Singularité architecturale, 104
- Singularité, Classe de, 100, 105
- Singularité de contrainte, 104
- Singularité directe, 25, 99
- Singularité d'entrée, 99
- Singularité entrée-sortie, 99
- Singularité inverse, 24, 98
- Singularité de sortie, 99
- Singularité, Type de, 100
- Spécification technique, 10
- Surcharge d'opérateur, DA, 53, 56
- Synthèse dimensionnelle, 10
- Système étendu, 22
- Coefficient de Taylor, 46
- Série de Taylor, 45
- Polynôme de Taylor, 45
- Transformation de source, DA, 53
- Validité, Domaine de, 50
- Vitesse d'entrée, 23, 98
- Vitesse intermédiaire, 23, 98
- Vitesse de sortie, 23, 98

Outils numériques pour la conception de mécanismes

Résumé

Dans le contexte médico-chirurgical, la robotique peut être d'un grand intérêt pour des procédures plus sûres et plus précises. Les contraintes d'encombrement sont cependant très fortes et des mobilités complexes peuvent être nécessaires.

A ce jour, la conception de mécanismes non conventionnels dédiés est alors difficile à réaliser faute d'outils génériques permettant une évaluation rapide de leurs performances.

Cette thèse associe la continuation de haut-degré et la différentiation automatique pour répondre à cette problématique en introduisant une méthode de modélisation et un formalisme génériques pour la conception de mécanismes. Nos contributions concernent en particulier le développement d'outils numériques pour l'évaluation de l'espace de travail, et de la localisation et la nature des singularités d'un mécanisme, et une analyse de sensibilité de haut-degré. Ceux-ci sont évalués sur des mécanismes de référence.

Mots-clés : Conception de mécanismes, Espace de travail, Analyse de singularité, Continuation de haut-degré, Différentiation automatique, Sensibilité.

Abstract

In the medical and surgical background, robotics can be of great interest for safer and more accurate procedures. Size constraints are however strong and complex movements may be necessary.

To date, the design of dedicated non-conventional mechanisms is then a difficult task because of a lack of generic tools allowing a fast evaluation of their performances.

This thesis combines higher-order continuation and automatic differentiation to address this issue through the introduction of a generic modelling method and a generic formalism for mechanism design. Our contributions especially concern the development of numerical tools for the evaluation of the workspace, of the singularity localization and nature, and for a higher-order sensitivity analysis. These tools are evaluated on reference mechanisms.

Keywords: Mechanism design, Workspace, Singularity analysis, Higher-order continuation, Automatic differentiation, Sensitivity.