



HAL
open science

Génération des conditions d'existence d'une classe de systèmes de solides surcontraints avec les bases de Gröbner

Ruixian Liu Renaud

► **To cite this version:**

Ruixian Liu Renaud. Génération des conditions d'existence d'une classe de systèmes de solides surcontraints avec les bases de Gröbner. Autre. Ecole Centrale Paris, 2014. Français. NNT : 2014ECAP0015 . tel-01020268

HAL Id: tel-01020268

<https://theses.hal.science/tel-01020268v1>

Submitted on 8 Jul 2014

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

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



THÈSE

Présentée par **RENAUD Ruixian**

pour l'obtention du
GRADE DE DOCTEUR

École Doctorale : École Centrale Paris (ED287)

Spécialité : Sciences pour l'ingénieur

Laboratoire d'accueil : LISMMA (EA2336)

Génération des conditions d'existence d'une classe de systèmes de solides surcontraints avec les bases de Gröbner

Soutenue le : 6 février 2014

Devant un jury composé de :

M. Jean-Charles FAUGÈRE	Directeur de recherche	Président du Jury
M. Alain RIVIÈRE	Professeur des universités	Directeur de thèse
M. Luc MATHIEU	Professeur des universités	Rapporteur
M. Dominique MICHELUCCI	Professeur des universités	Rapporteur
M. Philippe SERRÉ	Maître de conférences	Examineur
M. André CLÉMENT	Professeur émérite	Examineur
M. Guillaume MANDIL	Maître de conférences	Examineur
M. Jean-François RAMEAU	Conseiller scientifique	Invité

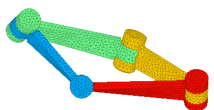
2009-2013

Année 2014

Numéro d'Ordre 2014ECAP0015

École Doctorale Sciences pour l'Ingénieur - ED287

École Centrale Paris



Remerciements

En premier lieu, je tiens à remercier mon directeur de thèse, monsieur Alain RIVIÈRE, directeur général de Supméca, pour m'avoir encouragé et avoir dirigé mon travail de recherche. Non seulement il m'a transmis ses connaissances scientifiques, mais il m'a aussi facilité les procédures administratives. Il m'a également offert des conditions de travail privilégiées. J'ai été extrêmement sensible à ses qualités humaines d'écoute et de compréhension tout au long de ces années.

J'adresse mille mercis à monsieur André CLÉMENT, professeur émérite et "moteur" de notre équipe de recherche, pour son soutien et son enthousiasme. Il m'a aidé à explorer des voies de recherches intéressantes. Sa présence et ses encouragements ont été décisifs dans ma thèse. Ses idées éclairées m'ont poussé à rester dans le monde de recherches.

Je suis infiniment reconnaissante envers monsieur Philippe SERRÉ, maître de conférences et co-encadrant, pour son suivi attentif de ma thèse. Sa grande disponibilité et sa méfiance face aux intuitions, son esprit entraîné à repérer les erreurs de raisonnement m'ont beaucoup aidé. Sa perspicacité m'a amené à approfondir, expliquer et ré-expliquer mes raisonnements tortueux jusqu'à ce qu'ils acquièrent une profondeur suffisante. Je me souviendrais longtemps nos longues discussions aux détours et contours multiples, parfois productives et souvent enrichissantes. J'ai beaucoup apprécié l'attention qu'il a su me prêter à chacune de mes irruptions intempestives dans son bureau. Philippe, merci pour ton aide précieuse, et son amitié.

Je souhaiterais également présenter ma gratitude à messieurs Luc MATHIEU et Dominique MICHELUCCI, mes rapporteurs de thèse, pour leur relecture scrupuleuse du manuscrit et leurs suggestions toujours avisées. Merci également aux examinateurs de ma soutenance qui ont été présents et ont su, par leurs remarques et leurs conseils, faire avancer ma réflexion : monsieur Jean-Charles FAUGÈRE et monsieur Guillaume MANDIL.

Ces remerciements seraient incomplets si je n'en adressais pas à l'ensemble des membres R&D de Dassault Systèmes : monsieur Patrick JOHNSON, monsieur Jean-François RAMEAU, monsieur Pascal SEBAH, monsieur Auxkin ORTUZAR et monsieur Michael DIGUET pour leurs échanges lors de nos réunions. Je remercie plus particulièrement Jean-François RAMEAU pour nos nombreuses discussions et ses connaissances dont il m'a fait bénéficier.

Je désire en outre remercier mes collègues pour la bonne ambiance qui règne à Supméca. Merci à eux de s'être rendus disponibles pour la soutenance blanche et la soutenance finale, et de m'avoir donné maints conseils.

J'aimerais remercier tous mes amis qui ont su me soutenir pendant toutes ces années, en particulier Mireille MOINET, Shangzhu LIFERMANN, Hervé ROGNON, Sylvain THOUVIOT, Christine YANG, Aurore TRAN VAN et tous les autres, pour leur aide et les bons moments passés ensemble durant ces années.

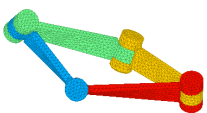
Ma reconnaissance va à ceux qui ont plus particulièrement assuré le soutien affectif de ce travail doctoral : ma famille, la famille RENAUD ainsi que la famille HRYSIEWICZ.

Je ne pourrais jamais assez remercier mon tendre époux, Franck RENAUD, mon premier lecteur et correcteur de fautes d'orthographe de cette thèse, pour toutes les heures qu'il a consacrées à la relecture, son soutien quotidien sans faille et son Amour indéfectible de chaque instant. Notre anniversaire de mariage a grandi en même temps que mon projet scientifique. Je t'aime.

Une pensée pour terminer ces remerciements pour toi qui n'a pas vu l'aboutissement de ma thèse mais je sais que tu aurais été très fière de ta fille!!! Merci d'avoir fait de moi ce que je suis aujourd'hui.

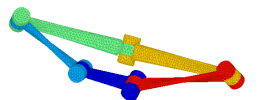
谨此博士论文献给我最爱的妈妈和爸爸！我爱你们！

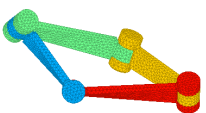
Ruixian
瑞贤





*À mes parents et aux parents de Franck,
et à tous ceux que je ne nomme pas, mais qui se reconnaîtront.*





Résumé

En cinématique il n'existe aucune méthode permettant de générer les conditions d'assemblage sous forme symbolique pour les systèmes de solides, éventuellement surcontraints. En revanche il existe un grand nombre de formules permettant - en principe - de calculer la mobilité. Les noms de Kutzbach, Grübler et Tchebycheff sont associés à différentes formules bien connues mais aucune formule infaillible n'a jamais été trouvée. C'est ce qui motive notre proposition de méthodes numériques et symboliques constructives. Celles-ci permettent de générer automatiquement les conditions d'assemblage et de mobilité à partir de la connaissance des équations de fermeture des boucles de solides.

Les méthodes numérique et symbolique présentées dans cette thèse se basent sur un socle commun. Elles utilisent les paramètres de Denavit-Hartenberg et les classent en deux catégories : la première catégorie, notée u pour usinage, représente les dimensions géométriques des solides, la seconde, notée m pour mobilité, représente les paramètres de position relative de deux solides en contact. Ensuite, les équations de fermeture sont obtenues par une méthode "coordinate free" à partir d'une matrice de Gram. C'est à cette étape que les deux méthodes diffèrent.

Dans le cas de l'analyse numérique locale, le système d'équations est linéarisé. La décomposition en valeur singulière est utilisée pour l'élimination des paramètres. Nous obtenons ensuite les conditions d'assemblage dans un voisinage de la configuration initiale. Les conditions de mobilité sont calculées à partir des conditions d'assemblage issues d'un nombre fini de configurations.

Dans le cas de l'analyse symbolique, nous calculons formellement la base de Gröbner associée aux équations de fermeture et cela grâce à l'algorithme FGb de Faugère. Il existe peu de références sur l'utilisation des Bases de Gröbner en cinématique, et aucune ne présente une analyse exhaustive du problème. Avec l'ordre spécifié, nous ne gardons que des paramètres u et éliminons tous les autres. Lorsque ces relations en u sont vérifiées, elles représentent les conditions d'assemblage, dites relations de surcontraintes. Cet ensemble peut être vide lorsque le système est isocontraint. Pour générer les conditions de mobilité, nous gardons tous les u et un paramètre de mobilité. En annulant les coefficients en facteur de ce paramètre de mobilité, un nouveau système qui ne dépend que de u est construit. Les conditions de mobilité sont obtenues en calculant la base de Gröbner du nouveau système. Pour atteindre les résultats désirés, nous avons également proposé deux outils : saturation et contrainte. Ils permettent de parcourir un sous-ensemble de solutions représentées par une base de Gröbner.

La confrontation des résultats obtenus dans la thèse avec ceux de la littérature, pour quelques de mécanismes connus, permet d'affirmer la validité et la complétude de la méthode.

Mots clés : conditions d'assemblage, conditions de mobilité, bases de Gröbner, analyses symboliques et numériques, système de solides 4R

Abstract

In mechanical kinematics analysis, no generic method allows to generate symbolic assembly conditions for mechanisms with rigid bodies (especially for over-constrained mechanisms). However, many formulas can help to compute mobility. Kutzbach, Grübler and Tchebitcheff are associated to different formulas but none of them works in every case. For this reason, we propose generic numerical and symbolic methods in order to obtain assembly and mobility conditions from closed-loop equations.

The numerical and symbolic methods presented in this thesis have the same starting point. They both use Denavit-Hartenberg parameters. These parameters are divided into two types : the first is called u , representing geometric dimensions of rigid bodies ; the second type is called m (for parameters of mobility), representing the relative motion between two rigid bodies in contact. First of all, closed-loop equations are written using a coordinate free method based on Gram matrix. After this step, the two methods differ.

In case of numerical analysis, the system of equations is linearized. The Singular Value Decomposition is used to eliminate parameters. We then compute assembly conditions in the neighborhood of the initial configuration. Mobility conditions are computed based on assembly conditions which result from a finite number of configurations.

In case of symbolic analysis, we calculate the Gröbner basis associated with closed-loop equations in order to eliminate parameters of mobility. There are few references talking about the application of Gröbner basis in kinematics analysis. None of these references provides a well-explained analysis of kinematics problem. Since we have been collaborating with Mr. Faugère, we use the package “FGb” for the elimination of parameters of mobility. With a specified order, relations between the u parameters could be computed by eliminating the m parameters. When verified, these relations are assembly conditions, also named over-constrained assembly conditions. This set of relations may be empty when the studied system is iso-constrained. To generate mobility conditions, one has to keep all u parameters and only one parameter of mobility. When mobility parameter’s coefficients vanish, the mechanism is mobile with 1 degree of freedom. To determine two kinds of conditions, we propose also two tools : saturation and constraint. They allow browsing a subset of system’s solutions which is represented by a Gröbner basis.

Since our results are identical to the ones found in literature, our method is valid and complete.

Key words : assembly conditions, mobility conditions, Gröbner basis, symbolic and numerical analysis, 4R linkage

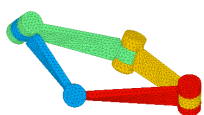
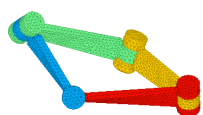


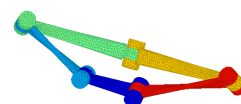
Table des matières

Remerciements	3
Table des matières	11
Introduction	17
1 État de l'art	25
1.1 Système de solides et modèle géométrique associé	26
1.1.1 Assemblage	27
1.1.2 Structure	27
1.1.3 Mobilité	28
1.1.4 Cas d'étude: mécanisme de Bennett	28
1.2 Paramétrages	29
1.2.1 Paramétrage de Denavit-Hartenberg	29
1.2.2 Paramétrage des modèles géométriques	31
1.3 Méthodes de mise en équation	32
1.3.1 Approche cartésienne	33
1.3.2 Approche non-cartésienne	34
1.4 Méthodes d'analyse	40
1.4.1 Méthodes d'analyse de structure	41
1.4.2 Méthodes d'analyse de mécanisme	41
1.4.3 Approche géométrique	43
1.5 Méthodes de résolution	45
1.5.1 Méthodes symboliques	45
1.5.2 Méthodes numériques	52
1.6 Conclusion	54
Bibliographie du chapitre 1	55
2 Approche numérique locale pour la mobilité	59
2.1 Introduction	61
2.1.1 Assembly and mobility check during Computer-Aided Design	61
2.1.2 Mobility vs. number of positions	62
2.1.3 A case study	63
2.2 Modeling approach, parameters and specifications	65
2.2.1 Modeling parameters	66
2.2.2 Specification parameters	69
2.2.3 Closed-loop equation	70

2.2.4	Equation system	70
2.3	Generation of compatibility equations	71
2.4	Compatibility equations of Goldberg $5R$ linkage	74
2.4.1	Equations of Goldberg $5R$ linkage	74
2.4.2	Calculation of mobility equations number	79
2.4.3	Mobility equations of specified Goldberg $5R$ linkage	81
2.5	Application of generated mobility equations	83
2.6	Conclusion and perspectives	84
Bibliographie du chapitre 2		86
3	Modélisation tensorielle des systèmes mécaniques	89
3.1	Introduction	90
3.2	Génération des Fonctions F_i	90
3.2.1	Formule d'Olinde-Rodrigues	91
3.2.2	Cas d'une chaîne ouverte de solides	91
3.2.3	Cas d'une chaîne fermée de solides	93
3.3	Mise en équation d'un système de solides	95
3.3.1	Méthode basée sur les fonctions F_i	95
3.3.2	Méthode de Jin et Yang	96
3.3.3	Variante de la méthode de Jin et Yang	99
3.4	Application au système de solides $4R$	100
3.4.1	Méthode basée sur les fonctions F_i	100
3.4.2	Méthode de Jin et Yang	108
3.4.3	Variante de la méthode de Jin et Yang	109
3.4.4	Discussion	109
3.5	Conclusion	116
Bibliographie du chapitre 3		117
4	Résolution formelle avec les bases de Gröbner	119
4.1	Introduction	120
4.2	Résolution symbolique d'un système de solides	121
4.2.1	Conditions d'assemblage	127
4.2.2	Conditions de mobilité	137
4.3	Validations	144
4.3.1	Résolution formelle et interprétation géométrique	144
4.3.2	Validation numérique	146
4.4	Application des équations d'assemblage	148
4.5	Conclusion	149
Bibliographie du chapitre 4		150
Conclusion générale		152
Bibliographie générale		156
A	Annexes de l'État de l'art	163
A.1	Quelques rappels	163
A.1.1	Espace vectoriels	163
A.1.2	Espace affine	163



A.1.3	Espace vectoriel normé	164
B	Annexes de la Résolution formelle avec les bases de Gröbner	165
B.1	Divers algorithmes basés sur les bases de Gröbner	165
B.1.1	Buchberger	166
B.1.2	Existence et unicité	168
B.1.3	Amélioration de l’algorithme de Buchberger	169
B.1.4	Autre algorithme	170
B.2	Simplification d’une équation d’assemblage de la famille du système à 4 barres .	170
B.3	Application de l’algorithme de conditions de mobilité avec la méthode Jin et Yang	172
C	Annexes du codes Maple: conditions d’assemblage et de mobilité	177



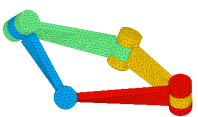
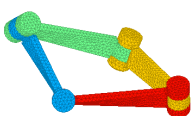


Table des figures

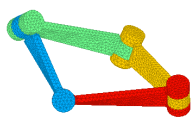
1	Système mécanique plan surcontraint avec 5 barres et 6 pivots (en 3D)	17
2	Représentation géométrique simplifiée du système mécanique de la figure 1	17
3	Représentation cartésienne du système à 5 barres dans l'espace géométrique	18
4	Les points rouges représentent les 6 solutions de la structure de figure 1	18
5	Un mécanisme à 5 barres avec 1 degré de liberté (1ddl)	19
6	La ligne noire représente l'ensemble des positions possibles du mécanisme sur- contraint à 5 barres. Le mécanisme en mouvement passe d'un point à l'autre de cette courbe.	19
7	L'ajout d'une barre quelconque au système à 5 barres	19
8	La structure (assemblage) d'un système à 6 barres	19
9	Un mécanisme à 6 barres avec 1 degré de liberté	19
10	Le processus de simulation des comportements physiques utilisant un maillage et le processus de détermination des caractéristiques géométriques	20
1.1	Assemblage impossible: une boucle ouverte	27
1.2	Assemblage rigide: c'est une structure	27
1.3	Assemblage mobile: c'est un mécanisme	27
1.4	Un mécanisme de Bennett	29
1.5	Notation de Denavit-Hartenberg	30
1.6	Le squelette associé au système de 4 solides et 4 liaisons	31
1.7	Le squelette avec la notation Denavit-Hartenberg associée à la figure 1.6	32
1.8	Les angles d'Euler	40
1.9	Mécanisme de Bennett à 1 degré de liberté	42
1.10	Une chaîne fermée	43
1.11	L'intersection de $f_1 = 0$ et $f_2 = 0$	52
1.12	Exemple: Schéma combinatoire de résolution par intervalles en bisection et éva- luation	54
2.1	Denavit-Hartenberg frame assignment	64
2.2	A general spatial 5-bar linkage. In this case, the displayed mechanism is rigid.	65
2.3	A Goldberg 5R linkage	66
2.4	A flowchart of the new numerical approach	75
2.5	Goldberg's 14 positions	80
2.6	Rank increasing with the growth of M_{stack}	81
3.1	Modèle vectoriel d'Olinde-Rodrigues	91
3.2	Un squelette composé de 5 vecteurs	92
3.3	mise en évidence de la boucle des paramètres angulaires	94

3.4	Le système de 6 solides et 6 liaisons	96
3.5	Le squelette 3D associé à la figure 3.4 avec la notation D-H	96
3.6	Le squelette associé au système de 6 solides et 6 liaisons [Yan02]	97
3.7	La coupure sur le squelette de la figure 3.6	97
3.8	Squelette d'un 4R représenté par les vecteurs et les segments de droites	100
3.9	La boucle des paramètres angulaires pour le 4R	100
3.10	La coupure effectuée sur les liaisons pivots \vec{n}_B et \vec{n}_D	108
3.11	La coupure sur les liaisons pivots \vec{n}_B et \vec{n}_D fait intervenir les angles θ_A et θ_C	108
3.12	Un mécanisme de 4R mobile en 3D: mécanisme de Bennett.	115
3.13	Un mécanisme 4R mobile avec deux solides superposés aux deux autres.	115
3.14	Un exemple de mécanisme de 4R mobile	116
4.1	Méthodes génériques pour générer les conditions d'existence qui ne dependent que de u	120
4.2	Le système mécanique plan sur-contraint composé de 5 barres	122
4.3	L'ajout d'une saturation revient à couper une branche d'un arbre. Ceci permet d'éliminer les solutions indésirables comme les longueurs nulles par exemple.	125
4.4	L'ajout d'une contrainte revient à parcourir une branche. Ceci permet de parcourir des solutions particulières.	125
4.5	Les étapes principales d'obtention des conditions d'assemblage	127
4.6	L'obtention des conditions d'assemblage dans un cas idéal	129
4.7	L'obtention des conditions de mobilité dans un cas relativement complexe	129
4.8	Recherche d'un ensemble de polynômes générateurs plus petits qui représente le même idéal que les n polynômes.	130
4.9	La technique pour minimiser le nombre de polynômes générateurs: comparer les 2 bases de Gröbner.	130
4.10	L'étude de la situation $L2 + L1 + L3 - L4 = 0$	137
4.11	L'étude de la situation $L2 + L1 + L3 - L4 \neq 0$	137
4.12	Les étapes principales pour obtenir les conditions de mobilité	138
4.13	Mécanisme de Bennett de la solution $a1 = a3$ et $a2 = a4$	145
4.14	Mécanisme de Bennett de la solution $a1 = -a3$ et $a2 = -a4$	145
4.15	Représentation géométrique de 4R pour le cas $L1 = L2$	146
4.16	Système à 4 barres assemblées de la configuration 2 (1 ^{ère} solution)	149
4.17	Système à 4 barres non assemblées de la 2 ^{ème} solution	149



Liste des tableaux

1	Tableau des contributions de la thèse	22
2.1	Initial configuration of Goldberg $5R$	66
2.2	Values of $(\mathbf{q}_0, \mathbf{s}_0)$ when $\theta = 20^\circ$	78
2.3	Second configuration of Goldberg $5R$	81
2.4	Initial and final configuration of Goldberg $5R$	84
3.1	Paramètres dimensionnels et de mobilité décrivant le squelette de Bennett	100
3.2	Matrice de Gram unitaire d'un 4R en fonction de la formule d'Olinde-Rodrigues	102
3.3	Matrice de Gram unitaire symétrique	105
3.4	Tableau des solutions des équations (3.64) – (3.68)	114
3.5	Les solutions des 3 figures	115
4.1	Tableau des solutions calculées avec les 7 facteurs de l'équation (4.24)	136
4.2	Différentes configurations d'un mécanisme du Bennett	147



Introduction

Nous avons choisi de débiter ce mémoire par l'étude d'un exemple afin de présenter les objectifs de nos travaux.

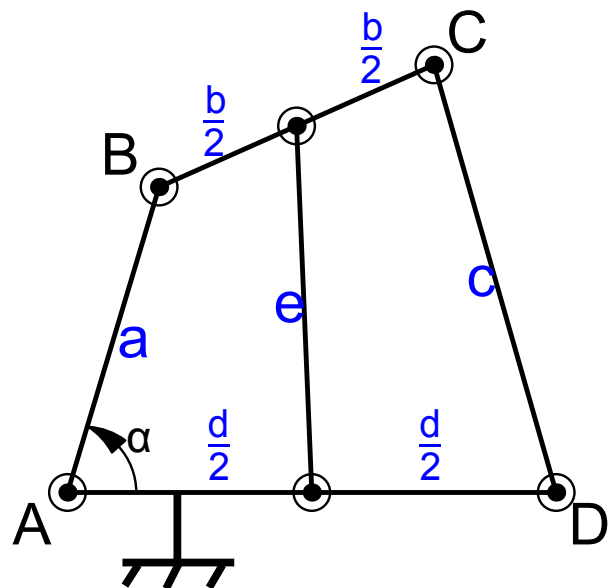
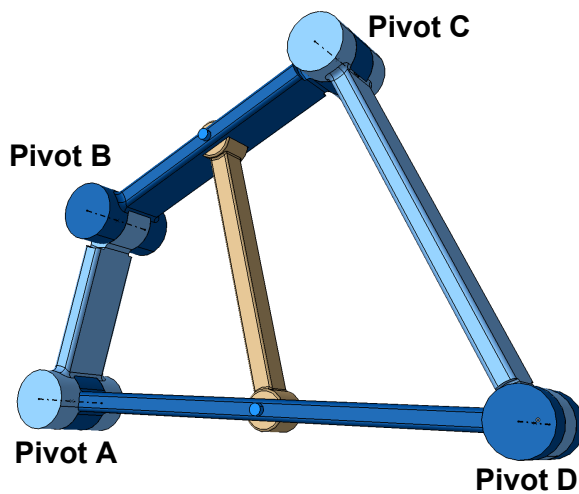


FIGURE 1 – Système mécanique plan surcontraint avec 5 barres et 6 pivots (en 3D)

FIGURE 2 – Représentation géométrique simplifiée du système mécanique de la figure 1

La figure 1 montre un système mécanique plan composé de cinq barres indéformables connectées deux à deux par 6 axes parallèles. Sa représentation simplifiée, exposée à la figure 2, est un squelette plan. Le point B tourne autour du point A. La barre AD est fixe, avec d sa longueur ; a la longueur de barre AB ; b la longueur de barre BC ; c la longueur de barre CD ; e la longueur de la barre centrale. L'angle α définit la position de AB par rapport AD.

Le modèle cartésien associé au système à 5 barres est montré dans la figure 3. Supposons que le point A est l'origine du repère, et que la barre AD coïncide avec l'abscisse x de ce repère. En notant (p, q) les coordonnées du point B ; (s, t) les coordonnées du point C et $(r, 0)$ les coordonnées du point D, les équations suivantes décrivent les relations entre la représentation

géométrique et la représentation cartésienne.

$$\begin{cases} p^2 + q^2 = a^2 \\ (s - p)^2 + (t - q)^2 = b^2 \\ (s - r)^2 + t^2 = c^2 \\ (p + s - r)^2 + (q + t)^2 = 4e^2 \end{cases}$$

Il est facile de trouver la relation $r = d$ grâce à la coïncidence entre AD et l'axe x . De la même manière, p et q peuvent être décrits par a et α . Remplaçons (p, q) par $p = a \cdot \cos \alpha$ et $q = a \cdot \sin \alpha$, les 4 équations précédentes deviennent :

$$\begin{cases} (s - a \cos \alpha)^2 + (t - a \sin \alpha)^2 - b^2 = 0 \\ (s - d)^2 + t^2 - c^2 = 0 \\ (a \cos \alpha + s - d)^2 + (a \sin \alpha + t)^2 - 4e^2 = 0 \end{cases} \quad (1)$$

En mécanique, un assemblage de solides complètement rigide s'appelle une **structure**. Lorsqu'elle possède une (ou plusieurs) mobilité on l'appelle un **mécanisme**. Les définitions mathématiques de l'assemblage, structure et mobilité se trouvent dans §1.1. Ainsi, le système montré dans la figure 1 est une structure.

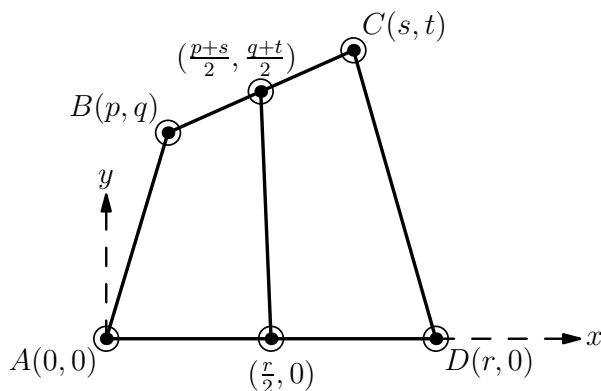


FIGURE 3 – Représentation cartésienne du système à 5 barres dans l'espace géométrique

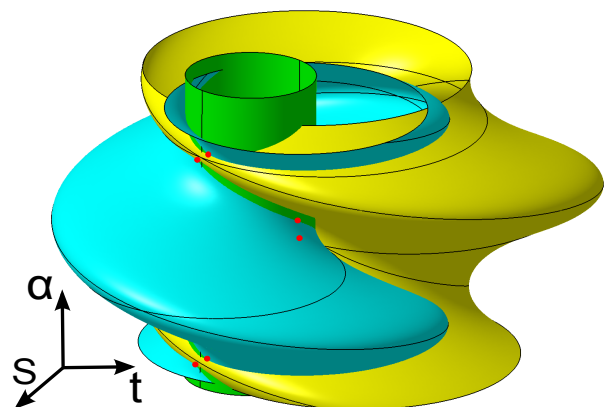
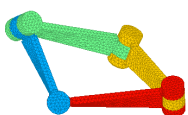


FIGURE 4 – Les points rouges représentent les 6 solutions de la structure de figure 1

Les 3 équations précédentes décrivent la représentation cartésienne (figure 3) dans l'espace paramétrique en fonction de (s, t, α) . La figure 4 est une représentation de ces 3 variétés algébriques (s, t, α) dans l'espace paramétrique associés aux 3 équations. La première équation $(s - a \cos \alpha)^2 + (t - a \sin \alpha)^2 - b^2 = 0$ définit l'enveloppe bleue d'un cercle horizontal de rayon b se déplaçant le long de l'hélice $\alpha \rightarrow (a \cos \alpha, a \sin \alpha, \alpha)$. La seconde équation $(s - d)^2 + t^2 - c^2 = 0$ définit un cylindre vert de rayon c et d'axe vertical passant par le point $(d,0,0)$. La troisième équation $(a \cos \alpha + s - d)^2 + (a \sin \alpha + t)^2 - 4e^2 = 0$ est caractérisée par l'enveloppe jaune de rayon $2e$ se déplaçant le long de l'hélice $\alpha \rightarrow (d - a \cos \alpha, -a \sin \alpha, \alpha)$. Les solutions des 3 équations sont les intersections de ces trois surfaces, représentée par les points rouges. Chaque point isolé correspond à une structure. Quand une structure peut passer de manière continue à une autre structure, le système est un mécanisme, voir l'exemple de la figure 5. Ainsi, l'intersection des surfaces devient une courbe lisse, comme la courbe noire montrée dans la figure 6. Après



une première étude, on constate que la structure de la figure 2 se transforme en un mécanisme lorsque $a = c = e$, $b = d$ (figure 5). Le mouvement du mécanisme suit la courbe noire dans la figure 6.

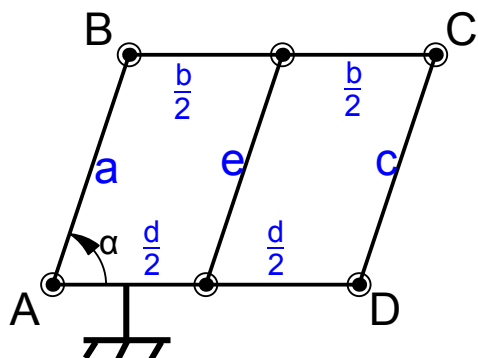


FIGURE 5 – Un mécanisme à 5 barres avec 1 degré de liberté (1ddl)

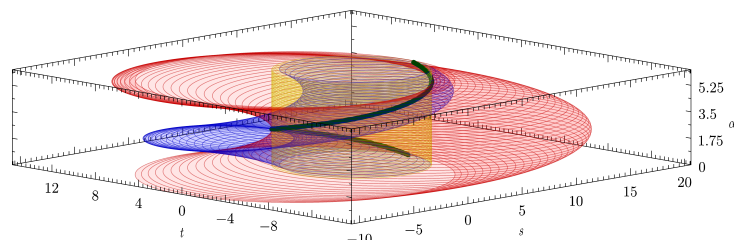


FIGURE 6 – La ligne noire représente l'ensemble des positions possibles du mécanisme surcontraint à 5 barres. Le mécanisme en mouvement passe d'un point à l'autre de cette courbe.

Supposons maintenant que l'on ajoute une barre supplémentaire à la structure précédente, les 6 barres ne s'assemblent pas dans la majorité des cas, comme montré dans la figure 7. Ainsi, il faut qu'un ingénieur modifie la longueur d'une (ou des) barres pour qu'elles s'assemblent comme dans la figure 8. Il est facile de transformer la structure en mécanisme à un degré de liberté en imposant $a = e = f = c$ et $b = d$, voir figure 9.

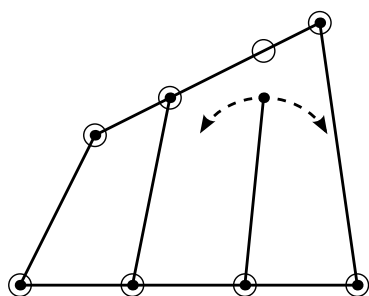


FIGURE 7 – L'ajout d'une barre quelconque au système à 5 barres

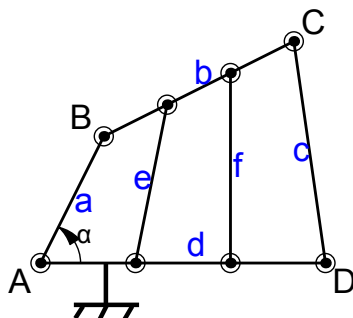


FIGURE 8 – La structure (assemblage) d'un système à 6 barres

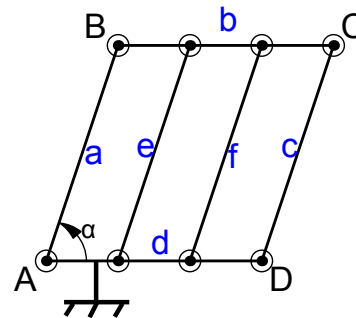
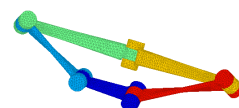


FIGURE 9 – Un mécanisme à 6 barres avec 1 degré de liberté

En général, plus il y a de pièces dans un système, plus il est compliqué de les assembler, et encore plus difficile de les rendre mobiles.

Le sujet de cette thèse est de proposer une méthode générique qui génère les conditions d'assemblage et de mobilité pour les systèmes mécaniques.



Contexte

Dans les années 70, les logiciels de CAO¹ n'étaient que des logiciels de DAO²[Hob]. Ils ont évolué petit à petit, d'une part grâce à l'augmentation des performances du matériel informatique et d'autre part grâce à la recherche dans le domaine algorithmique. Même maintenant, ce ne sont pas encore véritablement des systèmes d'aide à la conception. Néanmoins, les développements de CAO ont actuellement atteint un tel niveau de fiabilité et de maturité qu'il a été possible récemment d'obtenir un modèle numérique complet de grande précision pour des très grands ensembles mécaniques. Par exemple, le Boeing 787 et l'Airbus A380 comportent plusieurs millions de pièces ; le groupe moto-propulseur d'une Mercedes comporte plus de 10 000 pièces. Le modèle géométrique numérique est maintenant d'une telle précision qu'il est directement utilisé pour commander les Machine-Outils à Commande Numérique (MOCN). C'est le premier exemplaire de la série qui sert à mesurer expérimentalement la distance entre le modèle et le réel, et qui sert à évaluer la fiabilité du système ou à détecter les éventuelles erreurs humaines de conception.

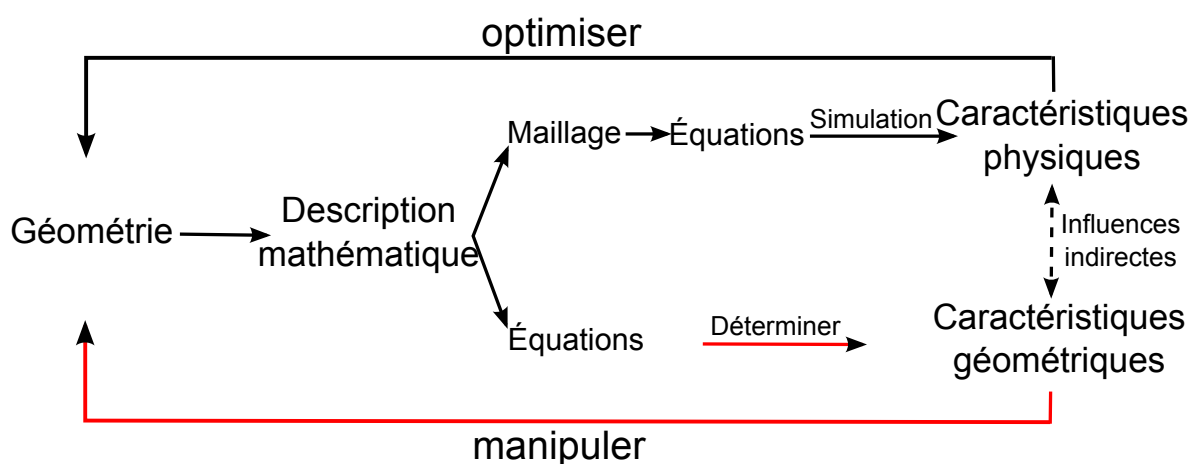
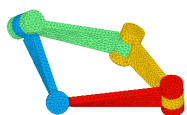


FIGURE 10 – Le processus de simulation des comportements physiques utilisant un maillage et le processus de détermination des caractéristiques géométriques

La représentation géométrique, également appelée “master geometrical model”, est le premier pilier de la conception. Le second pilier est constitué par de nombreux logiciels de simulations qui permettent de déterminer le comportement physique d'un objet : dynamique, thermique, etc. La plupart de ces logiciels reposent sur la méthode des éléments finis : nous pouvons citer ABAQUS, NASTRAN, ANSYS, Flux, Rayon, etc. La méthode des éléments finis permet de simuler le comportement physique de l'objet en prenant en compte la géométrie de l'objet grâce à un maillage. Le maillage est obtenu en discrétisant la géométrie définie par le master modèle. Il existe des logiciels tels COMSOL, qui permettent d'effectuer des simulations multiphysiques avec un couplage fort. L'équation, résolue par la discrétisation spatiale (maillage) et temporelle, intègre déjà toutes les physiques que l'utilisateur souhaite modéliser. Naturellement tous ces logiciels de simulation exigent en entrée une description mathématique de la géométrie adaptée à l'application physique envisagée (Cf. figure 10, chemin de flèches noires en haut).

1. Conception Assistée par Ordinateur
2. Dessin Assisté par Ordinateur



Il existe plusieurs paramétrages qui peuvent décrire le même système comme dans la figure 1. Afin de résoudre efficacement un problème, le choix de paramétrage est très important. Depuis Descartes, on sait représenter des objets géométriques de plus en plus complexes par un paquet d'équations (chemin de flèche noir en bas). Mais à l'inverse (chemin de flèches rouges), les scientifiques ont moins avancé pour découvrir les propriétés des géométries à partir de ses équations.

En 1900, Hilbert a introduit la dualité entre les propriétés des objets géométriques d'une portion de l'espace et les propriétés des polynômes de l'idéal correspondant. Cela conduira en 1964 aux bases de Gröbner. La force de cette dualité est d'associer à l'objet géométrique un objet algébrique, puis d'établir un dictionnaire permettant de traduire les propriétés géométriques en propriétés algébriques et vice et versa. Les mathématiciens appellent *géométrie algébrique* l'étude de cette dualité. Les ingénieurs préfèrent parler de *Géométrie* pour désigner la construction et la manipulation de ce dictionnaire.

Depuis le milieu de *XIX^{ème}* siècle, les recherches mathématiques sur l'étude des variétés se déclinent en deux points de vue : le point de vue local³ et le point de vue global⁴. L'étude locale porte vers l'élément unique de l'espace des relations géométriques. L'étude globale cherche au contraire à caractériser la totalité, indépendamment des éléments qui la composent ; elle s'attaque d'emblée à la représentation algébrique des connexions de l'ensemble.

- cette dualité locale versus globale s'est toujours présentée aux mécaniciens comme une opposition entre deux modes d'étude, irréductibles l'un à l'autre. Elle se traduit dans les systèmes CAD/CAM, où l'on étudie séparément la représentation finie⁵ d'un solide unique sous forme d'expressions algébriques finies faisant parties de la géométrie analytique et la représentation différentielle d'un système de solides sous forme d'expressions d'équations différentielles faisant parties de la cinématique. Mais il n'existe pas de pont aujourd'hui entre les deux approches ; ce pont que la Géométrie se propose de construire.
- Or l'ingénieur ou le mécanicien est intéressé essentiellement par des propriétés géométriques finies et systémiques de sa machine. C'est ce qu'on appelle ici la Géométrie.

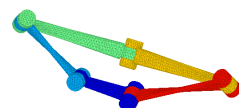
Souvent, le système d'équations associé aux boucles de fermeture des solides est non-linéaire.

L'étude des **bases de Gröbner** d'un tel système permet d'analyser formellement ses propriétés. Ce sera l'outil utilisé dans cette thèse pour le calcul formel. Jusqu'à aujourd'hui, en raison de la complexité exponentielle de cet outil, de la puissance des ordinateurs et du développement de la programmation, il était très difficile de s'attaquer au problème global. C'est pourquoi la résolution formelle de ces équations n'est possible que depuis une vingtaine d'années.

Grâce aux récents développements mathématiques de résolution formelle des systèmes algébriques non linéaires et le développement en France d'ordinateur puissant tel que "Curie", il est maintenant possible de développer une technologie "Géométrie" pour commencer à étudier le problème systémique des propriétés d'un système de solides.

Les travaux de cette thèse permettront de répondre aux besoins actuels de l'ingénieur-concepteur pour qu'il puisse

3. géométrie différentielle avec Riemann
4. géométrie algébrique avec Sylvester
5. "closed form" en anglais



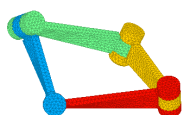
1. connaître automatiquement les caractéristiques de systèmes de solides dès son schéma conceptuel terminé. C'est ce qu'on appelle les conditions de compatibilité, et qui sont purement et simplement les conditions d'existence des variables dimensionnelles de chaque solide à l'intérieur d'un assemblage. Cette garantie est fondamentale pour l'ingénieur parce qu'elle n'est pas fournie expressément aujourd'hui pour un assemblage. Être informé de la mobilité ou au contraire de la rigidité de son assemblage. L'objectif étant d'établir l'existence des relations d'entrée/sortie des variables de position relative des différents solides.
2. être informé de la "robustesse" de cet assemblage au sens de Taguchi [Hwa96, Mad99, Tsu92], c'est-à-dire connaître automatiquement l'expression formelle des coefficients d'influence des variations de chaque paramètre pour pouvoir éventuellement les optimiser. Il n'y a plus de différences mathématiques entre la valeur nominale et la tolérance que le fabricant lui attribue.
3. définir un ordonnancement des spécifications qui hiérarchise la prise de décision d'un travail collaboratif et régularise ainsi les fréquents conflits décisionnels.
4. définir théoriquement et évaluer pratiquement le jeu fonctionnel minimum de la machine. Ceci est un domaine vierge scientifiquement, malgré un demi-siècle de recherches mondiales, car les résultats-empiriques-sont des secrets bien gardés par les entreprises.
5. enfin avoir la connaissance de ces propriétés globales, non seulement simplifierait et donc accélérerait le travail collaboratif au sein d'une entreprise, mais surtout, permettrait l'usage intensif des logiciels de simulation multi-physiques tels que Dymola ou Matlab/Simulink sans solution de continuité. En effet, ces logiciels sont tributaires, des dimensions géométriques de la machine et de leurs diverses relations. Étant donné les relations de compatibilité, la simulation des logiciels est plus réaliste. Le même problème se pose d'ailleurs, mais différemment, pour les logiciels de simulation qui reposent sur la méthode des éléments finis.

Contributions

	Approches numériques locales	Approches formelles globales
Assemblage	Article : An approach to manage over-constrained mechatronic system in teamwork (2011)	Conditions d'assemblage de 4R (2011-2013)
Mobilité	Article : A tool to check mobility under parameter variations in over-constrained mechanisms (Ch. 2) (2011-2012)	Conditions de mobilité de 4R (2011-2013)

TABLE 1 – Tableau des contributions de la thèse

En 2011, nous nous sommes concentrés sur les approches numériques locales qui permettent de générer les conditions de compatibilité locales d'assemblage et de mobilité. Un article de conférence a été publié en 2011. Il présente une approche pour générer les équations de compatibilité



sous les variables multi-physiques pour garantir l'assemblage d'un système piézo-électrique. Un autre article (voir Ch.2) sur les équations d'assemblage et de mobilité d'un mécanisme de solides a été publié dans une revue internationale. Néanmoins, à cause des caractéristiques de ces approches numériques locales (linéarisation, possibilité d'application sur des cas particuliers...), il est possible que nous perdions quelques informations si l'approche a été mal choisie. Pour cette raison, nous nous sommes intéressés depuis 2011 aux approches formelles globales pour obtenir, de manière générique, les conditions d'existence.

Une des contributions de cette thèse consiste à proposer une variante de l'écriture du système d'équations basée sur la matrice de Gram et la méthode de Jin et Yang [Yan02]. Une autre contribution est l'utilisation des bases de Gröbner pour résoudre formellement un problème polynomial issu d'un système mécanique. L'obtention des conditions d'assemblage et de mobilité se décompose en 3 phases :

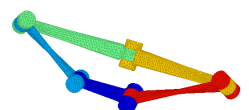
- *Identification* : cette phase a pour but d'identifier tous les angles et toutes les longueurs qui apparaissent dans le modèle à analyser en utilisant un squelette, puis à classer ces paramètres en deux catégories : *paramètres d'usinage* et *paramètres de mobilité*
- *Mise en équation* : cette phase établit le système d'équations à résoudre basé sur la matrice de Gram à l'aide de la méthode de Jin et Yang avec laquelle la condition de fermeture est obtenue. Cette méthode permet non seulement d'écrire moins d'équations par rapport aux mises en équations traditionnelles, mais aussi de faciliter la phase de résolution en ayant moins de paramètres à éliminer.
- *Résolution formelle* : le système d'équations, produit en phase 2, est résolu en utilisant les bases de Gröbner. Toutes les conditions d'assemblage ainsi que celles de mobilité sont exprimées formellement. Ceci permet de les réutiliser sous forme de contraintes.

La nouveauté de cette thèse réside dans l'utilisation des bases de Gröbner pour étudier formellement les équations algébriques associées aux boucles de fermeture des systèmes mécaniques. Cet outil permet également de découvrir les comportements géométriques d'une classe de mécanisme. Ces comportements sont représentés par les conditions d'assemblage et de mobilité.

Plan du mémoire

Le premier chapitre développe l'état de l'art concernant le modéleur géométrique déclaratif sur lequel l'équipe du LISMMA⁶ a travaillé pendant ces dernières années. Une synthèse est faite des différentes approches qui ont été utilisées dans cette thèse ou pourraient être utilisées par la suite pour la modélisation géométrique et la résolution symbolique d'une classe de mécanismes à concevoir. Ensuite, les notions d'assemblage et de mobilité de systèmes de solides sont données. Puis un paramétrage fréquemment utilisé pour modéliser les problèmes géométriques est présenté. La deuxième partie consiste en un article publié dans une revue internationale. Cette publication valide les approches numériques développées pendant la thèse qui permettent de générer les conditions d'existence locales pour un système de solides étudié. Le troisième chapitre propose une modélisation vectorielle basée sur les matrices de Gram issue de travaux déjà réalisés dans le domaine non-cartésien. Différentes mises en équation de système de solides sont présentées. Une étude spécifique sur les différentes mises en équations est faite pour comparer avec les résultats du chapitre suivant. Enfin, dans le dernier chapitre, sont détaillés les approches

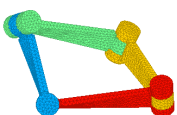
6. Laboratoire d'Ingénierie des Systèmes Mécaniques et des MATériaux



génériques et les algorithmes correspondants permettant de produire les conditions d'existence symboliques à l'aide des bases de Gröbner. Des exemples illustrent les résultats obtenus par les modèles développés. Des relations sous la forme symbolique entre les paramètres géométriques sont disponibles à la fin du chapitre.

Bibliographie de l'introduction

- [Hob] Hober. Introduction et développement de l'informatique à la SAT. Url : <http://www.satsouvenir.fr/index.php?p=4-3&contrib=71>.
- [Hwa96] J. M. Lim & K. S. Kim & B.J. Yum & H. Hwang. Determination of an optimal configuration of operating policies for direct-input-output manufacturing systems using the Taguchi method. *Computers Industrial Engineering*, 31(3/4) :555–560, 1996.
- [Mad99] I. E. Madu & C. N. Madu. Design optimization using signal-to-noise ration. *Simulation and Practice*, 7 :349–372, 1999.
- [Tsu92] K. L. Tsui. An overview of Taguchi method and newly developed statistical methods for robust design. *IIE Transaction*, 24(5) :44–57, 1992.
- [Yan02] Qiong Jin & Tingli Yang. Overconstraint analysis on spatial 6-link loops. *Mechanism and machine theory*, 37 :267–278, 2002.

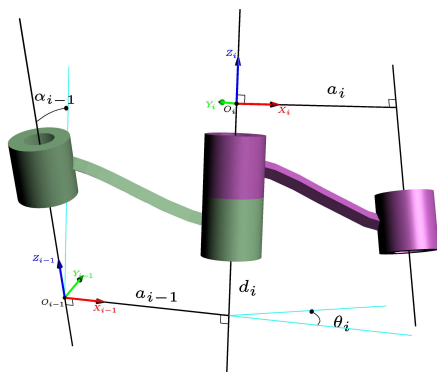


Chapitre 1

État de l'art

Sommaire

1.1	Système de solides et modèle géométrique associé	26
1.1.1	Assemblage	27
1.1.2	Structure	27
1.1.3	Mobilité	28
1.1.4	Cas d'étude: mécanisme de Bennett	28
1.2	Paramétrages	29
1.2.1	Paramétrage de Denavit-Hartenberg	29
1.2.2	Paramétrage des modèles géométriques	31
1.3	Méthodes de mise en équation	32
1.3.1	Approche cartésienne	33
1.3.2	Approche non-cartésienne	34
1.4	Méthodes d'analyse	40
1.4.1	Méthodes d'analyse de structure	41
1.4.2	Méthodes d'analyse de mécanisme	41
1.4.3	Approche géométrique	43
1.5	Méthodes de résolution	45
1.5.1	Méthodes symboliques	45
1.5.2	Méthodes numériques	52
1.6	Conclusion	54
	Bibliographie du chapitre 1	55



Résumé

Ce chapitre présente l'objectif de la thèse : *spécifier l'assemblage et la mobilité d'un système de solides*. Pour aborder l'identification des paramètres, nous proposons d'utiliser un squelette et la notation Denavit-Hartenberg. Les paramètres sont divisés en deux catégories : paramètres d'usinage et paramètres de mobilité. La mise en équation sert à établir les relations entre deux types de paramètres. Pour analyser le système de solides, des méthodes algébriques adaptées et efficaces sont nécessaires.

1.1 Système de solides et modèle géométrique associé

Un *système de solides* est un ensemble de solides indéformables, reliés entre eux par des liaisons mécaniques parfaites. Il est conçu pour réaliser une ou plusieurs fonctions. Plus formellement, un **système de solides** en dimension 3 se définit par une paire (S, L) où :

- S est un ensemble de **pièces mécaniques**, corps solides indéformables en dimension 3 ;
- L est un ensemble de **liaisons mécaniques**, c'est-à-dire la réalisation physique de la connexion de 2 solides (2 espaces affines distincts \mathbb{R}^3) obtenue par la mise en coïncidence de leurs EGRM¹, chaque liaison liant une paire de solides de S .

Une *liaison mécanique* permet de mettre en relation deux pièces en bloquant certains mouvements relatifs entre elles. Dans cette thèse, les liaisons mécaniques sont restreintes aux liaisons n'autorisant qu'un mouvement relatif clairement défini : les *liaisons cinématiques parfaites*. Elles ne présentent ni irrégularités, ni jeu, ni frottement dans les mouvements qu'elles autorisent entre les pièces impliquées. Il est possible de ramener tous les types de liaisons cinématiques à une combinaison particulière de liaisons cinématiques de type pivot et glissière [Tem84] ou seulement pivot-glissant avec certaines restrictions. Pour cette raison, nos recherches se portent d'abord sur les systèmes de solides réalisés avec des liaisons pivot parfait. A l'avenir, les études pourront s'étendre aux autres liaisons.

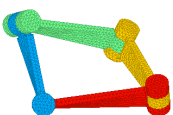
Beaucoup de recherches sont faites sur les systèmes de solides depuis le *XIX^{ème}* siècle. En 2007, la société *Information Européen* a démarré *ICT Policy Support Programme (ICT PSP)* [CIP] qui conduit au projet *thinkMotion* [Pro] dont le but est d'avoir une grande bibliothèque numérique en libre accès, dans le domaine des systèmes de mouvement, ce qui est essentiel pour tous les domaines d'ingénierie. Le projet *thinkMotion*, auquel cette thèse pourra servir, couvre à la fois les connaissances techniques originales et le progrès technique rapide. Dans cette bibliothèque se trouve de nombreux mécanismes plans et spatiaux possédant 3, 4 ou 5 liaisons. Certains d'entre eux ont été étudiés et fabriqués.

Ainsi, nous cherchons, dans cette thèse, des méthodes génériques qui permettent d'analyser et de caractériser les comportements de systèmes de solides. Concrètement, à quelles conditions un système de solides s'assemble ou est mobile.

Souvent, les ingénieurs associent des modèles géométriques aux systèmes de solides étudiés. Ensuite, l'étude de systèmes de solides se transforme en analyse des modèles géométriques. Pour analyser un modèle géométrique, il est inévitable de citer la discipline le *Problème de Satisfaction de Contraintes Géométriques*(GCSP). Cette discipline propose une solution incontournable dans l'identification des paramètres, grâce à l'outil du squelette, qui permet de simplifier des modèles géométriques tout en conservant les informations intactes.

Un squelette est constitué d'un ensemble de barres (segments de droites) de longueurs données. Ces barres sont jointes par leurs extrémités et sont soumises à un ensemble de relations géométriques (distances, angles, coïncidences, colinéarités, \dots). Chaque relation géométrique est représentée par une équation ou inégalité portant sur les variables des entités.

1. Éléments Géométriques de Référence Minimum



On peut considérer un squelette dans une *configuration* donnée. Ceci consiste à décrire la géométrie (angles, longueurs, dimensions, forme,...) de chacune des pièces et à déterminer les paramètres (rotations relatives, translations,...) des liaisons. Pour notre étude, nous avons classé les paramètres dans un squelette en deux types :

- **Les paramètres d'usinage**² notés u , représentent les dimensions géométriques des solides tels que les longueurs et les angles etc. Par exemple les paramètres a, b, c, d et e de la figure 2 sont des paramètres d'usinage en longueur. Les paramètres d'usinage s'appellent aussi les *paramètres dimensionnels*.
- **Les paramètres de mobilité** notés m , représentent les paramètres de position relative de 2 solides en contact telles que les rotations et les translations dans des liaisons. Par exemple le paramètre α de la figure 2 est un paramètre de mobilité en rotation.

Grâce à cette classification des paramètres, nous pouvons définir mathématiquement un système de solides. C'est un quadruplet $T = (U, M, E, \tilde{C})$ où $U \subset \mathbb{R}^n$ est l'espace des paramètres d'usinage, $M \subset \mathbb{R}^m$ est l'espace des paramètres de mobilité, $E = \mathbb{R}^k$ et $\tilde{C} : U \times M \rightarrow E$ est la fonction des contraintes.

1.1.1 Assemblage

Mathématiquement, l'*assemblage* d'un système de solides est réalisable s'il existe $(u_0, m_0) \in U \times M$ tel que $\tilde{C}(u_0, m_0) = 0$. De la même manière, un assemblage ayant u_0 pour paramètres d'usinage est irréalisable si $\tilde{C}(u_0, m) \neq 0$ quelque soit le paramètre de mobilité m .

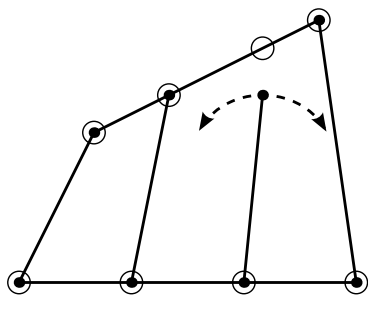


FIGURE 1.1 – Assemblage impossible : une boucle ouverte

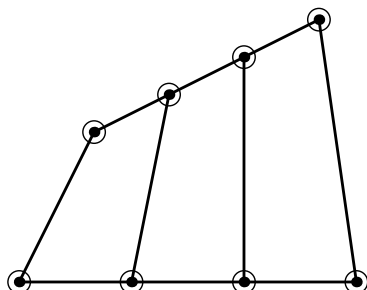


FIGURE 1.2 – Assemblage rigide : c'est une structure

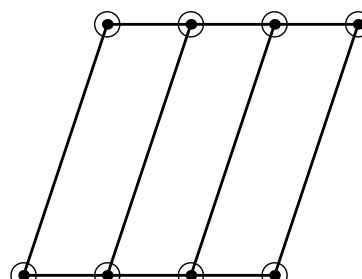


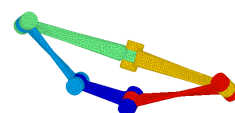
FIGURE 1.3 – Assemblage mobile : c'est un mécanisme

La figure 1.1 est un assemblage irréalisable. Les systèmes de la figure 1.2 et 1.3 sont des assemblages. Ils vérifient $\tilde{C}(u_0, m_0) = 0$. Pour distinguer la rigidité et la mobilité, deux définitions supplémentaires sont nécessaires.

1.1.2 Structure

Une *structure* est rigide si un assemblage est indéformable. C'est-à-dire que malgré ses liaisons cinématiques, les barres n'admettent aucun mouvement relatif les unes par rapport aux autres,

2. ou paramètres dimensionnels



voir la figure 1.2. Cependant, un système rigide comporte six mouvements dans l'espace en 3D : trois translations et trois rotations. Autrement dit, ce corps rigide doit être *déplaçable* car cette condition est vérifiée tant que les contraintes géométriques mises en jeu sont indépendantes du référentiel de l'espace géométrique considéré.

La rigidité, en mathématique (géométrie, algèbre), est le principal sujet d'étude de la commutativité de topologie structurale [Whi98, Gra02].

- les constructions sont réduites à leur squelette, à savoir des représentations de solides et de liaisons ;
- les propriétés physiques des matériaux utilisés dans la construction ne sont pas prises en compte.

Ces simplifications sont appliquées sur tous les squelettes de cette thèse.

1.1.3 Mobilité

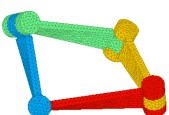
Quand un assemblage possède au moins un degré de liberté, il est considéré mobile, comme le squelette dans la figure 1.3. Un système mobile de solides est un *mécanisme*. Notons (u, m) un couple tel que $\tilde{C}(u, m) = 0$ et fixons u . Le mouvement du mécanisme noté $P(u)$ est le sous-ensemble de M défini par $P(u) = \{m \in M \text{ et } \tilde{C}(u, m) = 0\}$. Le mouvement est régulier si l'ensemble de $P(u)$ est une sous variété unidimensionnelle de m .

Une brève présentation et une analyse critique des différentes méthodes d'analyse de la mobilité présentées dans la littérature ces 150 dernières années est faite par Gogu [Gog05], pour bien situer les différentes contributions de l'étude sur la théorie des mécanismes. L'étude des mouvements d'un mécanisme en configuration connue peut se faire suivant différentes approches : l'*approche géométrique* considère les relations géométriques (positions et mouvements) du mécanisme ; l'*approche statique* considère les forces exercées dans le mécanisme, l'*approche cinématique* considère les vitesses relatives des pièces mécaniques les unes par rapport aux autres ; l'*approche dynamique* considère le travail et les puissances développées dans le mécanisme. Ces approches permettent de déterminer le *degré de mobilité* d'un mécanisme. Nous présentons dans la section 1.5.2 le principe de l'approche cinématique pour la détermination du degré de mobilité d'un mécanisme.

1.1.4 Cas d'étude : mécanisme de Bennett

Tout au long de l'étude, une chaîne fermée composée de 4 solides et 4 liaisons sert d'étude de cas. Il s'agit du mécanisme de Bennett, découvert en 1903 par G.T. Bennett [Ben03]. C'est un système de solides en 3D composé de quatre biellettes et n'ayant que des liaisons pivots, voir la figure 1.9. En général, un système est toujours mobile à partir de 7 solides et de 7 liaisons dans une boucle fermée. Le mécanisme de Bennett est intéressant pour notre étude parce qu'il est surcontraint et mobile.

Depuis l'article de Bennett "A new mechanism" [Ben03], un grand nombre de mécanismes 3D surcontraints sont présentés. Ils peuvent être considérés comme la combinaison de plusieurs mécanismes de Bennett, comme le mécanisme de Myard [Mya31], le mécanisme 5R de Goldberg



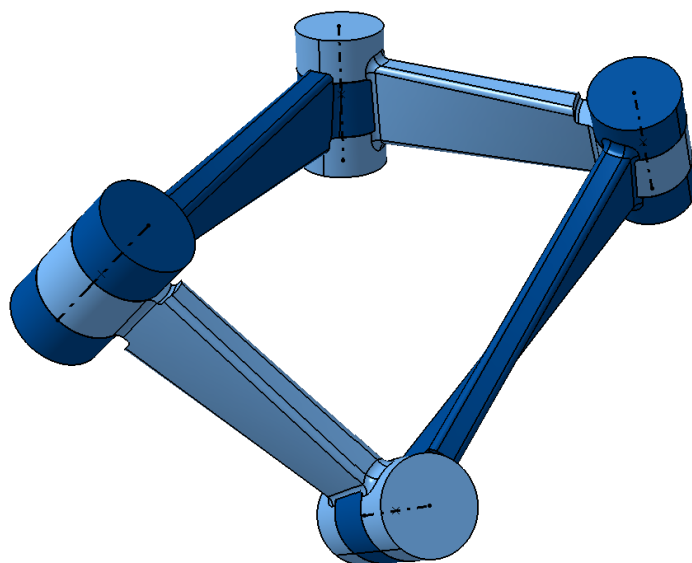


FIGURE 1.4 – Un mécanisme de Bennett

et le 6R [Gol43], le mécanisme 6R de Bennett-joint [Rot94], le mécanisme 6R de Dietmaier [Die95], le mécanisme de Wohlhart en double Goldberg [Woh91], etc. La plupart des mécanismes listés ci-dessus sont des systèmes de solides spatiaux. En raison des caractéristiques spatiales, ces mécanismes sont intéressants pour la conception industrielle où le mouvement spatial est nécessaire. Dans la thèse de Melin[Mel04], plusieurs mécanismes de Bennett sont utilisés pour construire un abri déployable et une structure de tente en assemblage rapide.

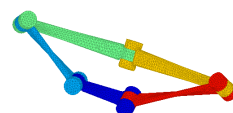
Il est connu qu'une chaîne fermée en 3D doit avoir au moins 7 liaisons pivots pour être mobile. Malgré ses 4 liaisons, le mécanisme de Bennett possède un degré de liberté. Bricard [Bri26] a donné les équations de compatibilité du mécanisme de Bennett. Ces équations sont les conditions nécessaires et suffisantes pour qu'un mécanisme de Bennett ait un degré de liberté. Mais les équations de compatibilité du mécanisme de 5R et 6R restent inconnues. Pour cette raison, le mécanisme de Bennett, ayant seulement 4 pivots, est intéressant à analyser pour étudier ensuite les mécanismes de 5R et 6R.

1.2 Paramétrages

Nous allons présenter dans cette partie comment paramétrer les systèmes de solides.

1.2.1 Paramétrage de Denavit-Hartenberg

La notation de Denavit-Hartenberg est très utilisée en robotique pour les chaînes ouvertes. Elle est utilisée dans cette thèse parce qu'elle permet d'identifier simplement les paramètres d'usinage et de mobilité définis au début de ce chapitre. La figure 2.1 montre une liaison Z_i liant une paire de solides $i - 1$ (en vert) et i (en magenta).



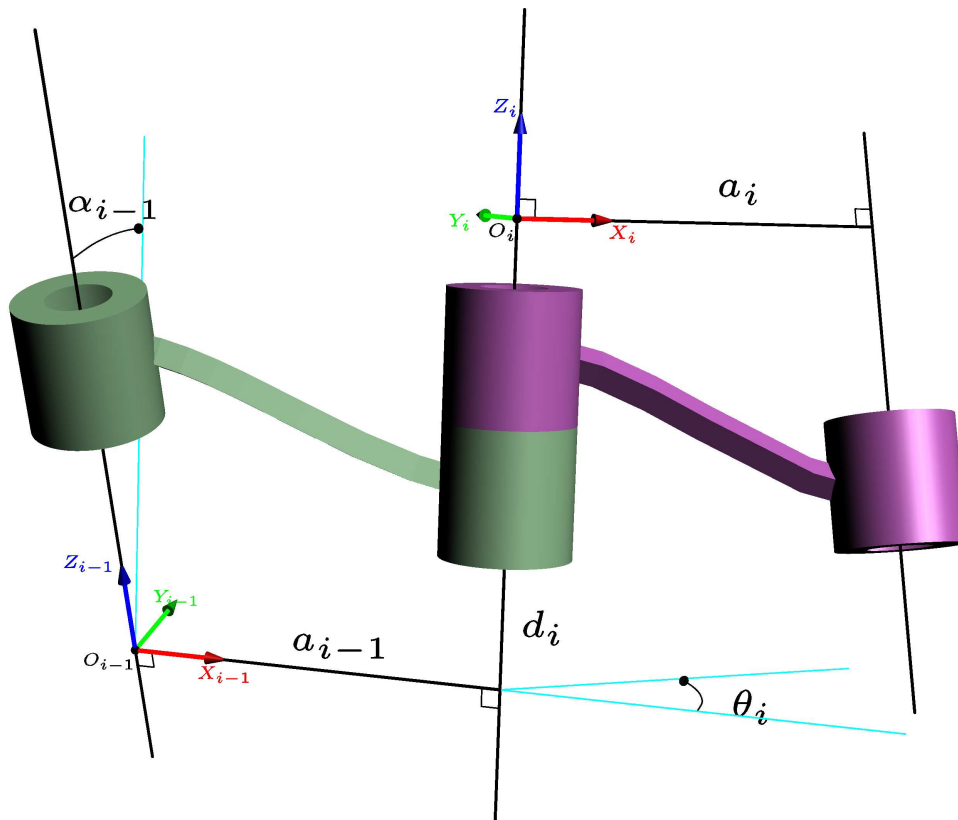
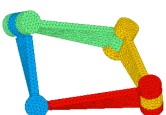


FIGURE 1.5 – Notation de Denavit-Hartenberg

- Paramètre d'usinage : a_{i-1} , la longueur d'usinage du solide $i - 1$, est définie comme la distance mesurée le long de la perpendiculaire commune aux deux axes des liaisons, $O_{i-1}\vec{Z}_{i-1}$ et $O_i\vec{Z}_i$;
- Paramètre d'usinage : α_{i-1} , l'angle d'usinage du solide $i - 1$, est l'angle entre les axes de liaisons, $(\vec{Z}_{i-1}, \vec{Z}_i) = \alpha_{i-1}$;
- Paramètre de mobilité : la position relative des deux solides est donnée par la distance d_i mesurée le long de l'axe de la liaison, entre les 2 perpendiculaires à l'axe ;
- Paramètre de mobilité : l'angle θ_i entre les perpendiculaires communes $\theta_i = (\vec{X}_{i-1}, \vec{X}_i)$ autour de \vec{Z}_i ;

Ainsi, les paramètres a_{i-1} et α_{i-1} sont les paramètres d'usinage du solide $i - 1$; les paramètres d_i et θ_i sont les paramètres de mobilité de la liaison Z_i . A noter qu'il y a deux contraintes fixes par défaut : les deux angles droits permettant de construire la perpendiculaire commune aux deux axes \vec{Z}_i et \vec{Z}_{i-1} .

Selon le type de liaison, un ou deux paramètres sont variables. Nous allons lister 4 liaisons principales :



- Pivot : $\theta_i = \theta_i(t)$ et $d_i = constante(cst)$;
- Prismatique : $\theta_i = cst$ et $d_i = d_i(t)$;
- Pivot glissant : $\theta_i = \theta_i(t)$ et $d_i = d_i(t)$;
- Hélicoïdal : $\theta_i = \theta_i(t)$ et $d_i = k\theta_i(t)$.

On peut généraliser cette méthode de paramétrage. Avec ce paramétrage, Mandil [Man11] a modélisé d'autres liaisons. Ces exemples sont présentés dans la section 3.1.3 de la thèse de Mandil [Man11]. Cailliau [Cai10] a décrit dans sa thèse des chaînes cinématiques comportant plusieurs boucles. Cailliau a présenté une méthode de synthèse qui permet de créer un squelette géométrique paramétrique. Ce squelette représente les différents éléments géométriques caractéristiques des solides et des liaisons par des structures filaires. Il utilise un graphe cinématique adjoint, dont les nœuds correspondent à des liaisons pivot-glissant et les arcs aux différents corps des pièces, pour modéliser les problèmes de synthèse dimensionnelle. Ce graphe peut représenter des mécanismes quelconques et intégrer les tâches à effectuer et les contraintes géométriques.

Nous nous limitons dans ce rapport au traitement d'une chaîne de solides fermée.

1.2.2 Paramétrage des modèles géométriques

Les parties précédentes se sont attachées à expliquer comment définir un paramétrage des variables d'usinage et de mobilité dans un système de solides dont les liaisons comportent un axe et en prenant en considération une seule boucle cinématique fermée. Ainsi, une fois que tous les solides et les liaisons sont déclarés, le squelette associé au système de solides forme une boucle constituée de segments de droites et de points. Chaque segment de droite est perpendiculaire à ses deux segments adjacents.

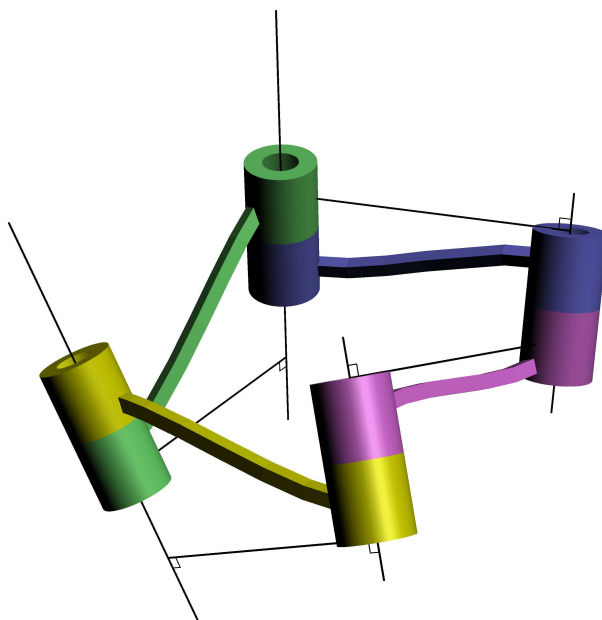
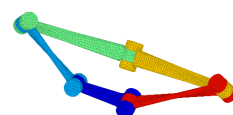


FIGURE 1.6 – Le squelette associé au système de 4 solides et 4 liaisons

La figure 1.6 montre un système de 4 solides et 4 liaisons. Ce système peut être représenté par le squelette (les droites noires) constitué par des droites ayant seulement des perpendiculaires



communes. Le squelette de la figure 1.7 est extrait de la figure 1.6. Par rapport à la notation de Denavit-Hartenberg, nous appelons L_i au lieu de a_i la longueur d'un solide.

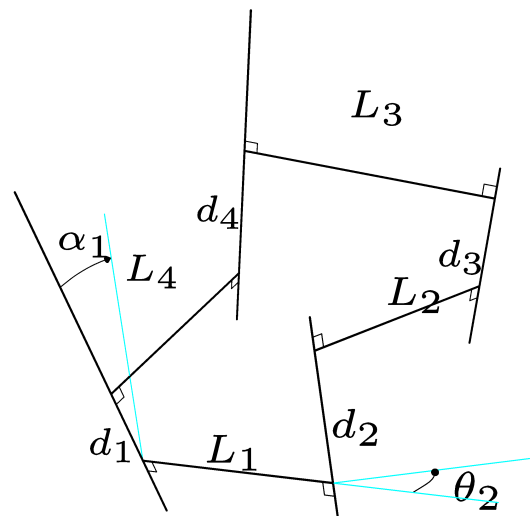


FIGURE 1.7 – Le squelette avec la notation Denavit-Hartenberg associée à la figure 1.6

Ainsi, les paramètres de ce squelette sont les suivants :

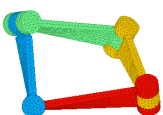
- paramètres d'usinage de longueur du solide : L_1 , L_2 , L_3 et L_4 ;
- paramètres d'usinage d'angle du solide : α_1 , α_2 , α_3 et α_4 ;
- paramètres de mobilité de rotation relative des liaisons : θ_1 , θ_2 , θ_3 et θ_4 ;
- paramètres de mobilité de translation relative d'une liaison : d_1 , d_2 , d_3 et d_4 .

Les 16 paramètres sont des spécifications du squelette de la figure 1.7.

1.3 Méthodes de mise en équation

Comme nous l'avons vu, les paramètres sont divisés en paramètres d'usinage et paramètres de mobilité. La mise en équation décrit les relations entre ces deux types de paramètres : $\tilde{C}(u, m) = 0$, ce que l'on appelle *équations de fermeture*. Les équations de fermeture décrivent la chaîne fermée de solides. Pour assembler les pièces mécaniques, un mécanicien ne se soucie pas de la position des pièces dans un repère absolu. Mais souvent, il est difficile d'écrire directement les relations qui ne dépendent que de u et m . Parfois, nous avons besoin d'une description en coordonnées cartésiennes pour faciliter l'écriture des équations de fermeture. Ainsi, nous allons présenter dans cette sous-section les méthodes de mise en équation avec les approches cartésiennes et non-cartésiennes.

En toute généralité, le système d'équations algébriques qui décrit la fermeture d'un système de solides s'écrit $C(x, u, m) = 0$ avec $x \in X$ un ensemble de paramètres de description (ex : coordonnées cartésiennes) ; $u \in U$ un ensemble de paramètres d'usinage ; $m \in M$ un ensemble



de paramètres de mobilité. Ces équations, souvent non-linéaires, définissent la position, la dimension et l'orientation de chaque élément de l'objet. Toutefois, pour retrouver $\tilde{C}(u, m) = 0$, x doit être éliminé par une méthode que nous présenterons plus tard.

1.3.1 Approche cartésienne

Il s'agit de l'approche la plus classique. Cette approche est basée sur l'utilisation du repère cartésien. L'idée principale est d'exprimer tous les objets géométriques dans un seul repère. Par conséquent, la position relative entre ces objets dépendra de la position absolue de chaque objet.

Prenons l'exemple de la figure 2, les points et les vecteurs sont représentés par les coordonnées cartésiennes usuelles. Des équations scalaires sont utilisées pour exprimer des distances et des angles entre deux vecteurs. Les contraintes de distance pour l'exemple de la figure 2 de la page 7 sont traduites de la manière suivante :

Les relations de distance sur les 5 barres :

$$\left\{ \begin{array}{l} a^2 - \langle \overrightarrow{AB}, \overrightarrow{AB} \rangle = 0 \\ b^2 - \langle \overrightarrow{BC}, \overrightarrow{BC} \rangle = 0 \\ c^2 - \langle \overrightarrow{CD}, \overrightarrow{CD} \rangle = 0 \\ d^2 - \langle \overrightarrow{DA}, \overrightarrow{DA} \rangle = 0 \\ e^2 - \left(\frac{\overrightarrow{DA}}{2} + \overrightarrow{AB} + \frac{\overrightarrow{BC}}{2} \right)^2 = 0 \end{array} \right.$$

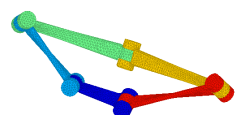
Et la relation d'angle :

$$\left\{ \begin{array}{l} \langle \overrightarrow{AB}, \overrightarrow{DA} \rangle - a \cdot d \cdot \cos \alpha = 0 \\ a^2 - \langle \overrightarrow{AB}, \overrightarrow{AB} \rangle = 0 \\ d^2 - \langle \overrightarrow{DA}, \overrightarrow{DA} \rangle = 0 \end{array} \right.$$

Soit, en remplaçant le produit scalaire par son expression

$$\left\{ \begin{array}{l} a^2 - (x_B - x_A)^2 + (y_B - y_A)^2 = 0 \\ b^2 - (x_C - x_B)^2 + (y_C - y_B)^2 = 0 \\ c^2 - (x_D - x_C)^2 + (y_D - y_C)^2 = 0 \\ d^2 - (x_A - x_D)^2 + (y_A - y_D)^2 = 0 \\ 4e^2 - (x_A + x_D - x_B - x_C)^2 + (y_A + y_D - y_B - y_C)^2 = 0 \\ (x_B - x_A)(x_A - x_D) + (y_B - y_A)(y_A - y_D) - a \cdot d \cdot \cos \alpha = 0 \end{array} \right.$$

Pour établir les relations entre u et m , le concepteur est obligé d'utiliser les coordonnées cartésiennes dans le repère absolu. L'orientation d'un solide par rapport à l'autre ainsi que la position doivent également être exprimées dans le même repère absolu. Souvent, le concepteur s'intéresse à l'orientation relative mais pas aux coordonnées du solide. C'est pour cette raison que ce modèle est mal adapté aux problèmes de spécifications d'objets géométriques. Au niveau



du système d'équations, un mauvais choix de repère peut impliquer une complication de génération du système d'équations et engendre des problèmes de résolution [Dur98, Mic95, The01]. Pour cette raison l'utilisation de coordonnées cartésiennes pour exprimer les relations entre les paramètres d'usinage et de mobilité n'est pas la meilleure approche. Nous proposons donc de remplacer la précédente par l'approche tensorielle qui permet de se libérer de la dépendance du repère cartésien.

1.3.2 Approche non-cartésienne

Le but de cette méthode est d'exprimer directement les relations entre u et m sans passer par le repère cartésien. Cette approche est basée sur l'utilisation de la matrice de Gram, qui permet de définir un espace métrique.

L'idée proposée est la suivante : l'utilisateur construit un squelette dans un espace métrique dont la taille est fonction du nombre d'éléments géométriques dessinés par le concepteur. Cet espace métrique est défini par une matrice de Gram. Une fois instancié, les propriétés de la matrice de Gram doivent être respectées. Par exemple, dans espace 3D, le rang de la matrice est égal à la dimension de l'espace de plongement, c'est-à-dire 3. Les notions de la matrice de Gram sont détaillées dans [Bas77, Car46, Lic83].

Les objets construits dans un modèleur CAO appartiennent à l'espace affine. La matrice de Gram décrit complètement un espace vectoriel. Il faut donc encore définir la technique qui va permettre de passer de l'espace vectoriel à l'espace affine. Après la présentation de la matrice de Gram, nous allons voir deux méthodes pour exprimer la fermeture vectorielle : *coupe métrique* et *projection*. Les définitions de l'espace vectoriel, de l'espace affine et de l'espace vectoriel normé se trouvent dans l'annexe A.1.

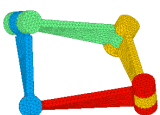
1.3.2.1 Matrice de Gram

Géométriquement, il est aisé de caractériser un repère quelconque : les 3 normes de chacun des 3 vecteurs de bases, les 3 angles que font leur direction, la séquence de définition du sens direct ou indirect et enfin un produit scalaire qui définit la relation entre les 3 normes de ces vecteurs. Une matrice de Gram permet précisément de représenter complètement ces repères quelconques comme les vecteurs. Les autres propriétés des matrices de Gram permettent d'effectuer simplement des changements de repères quelconques, en particulier d'exprimer le nouveau repère par un certain tenseur dépendant du premier, de conserver l'invariance des longueurs et des angles.

Soit E un espace vectoriel réel de dimension m . On appelle base de E la donnée de m vecteurs $\vec{b}_i (i = 1, \dots, m)$ telle que tout vecteur \vec{x} de E puisse s'exprimer d'une manière unique par une combinaison linéaire de m vecteurs \vec{b}_i .

$$\vec{x} = \sum_{i=1}^m x^i \vec{b}_i \tag{1.1}$$

Par définition, les x^i s'appellent les *composantes contravariantes* du vecteur \vec{x} dans la base $\langle \vec{b}_i \rangle$.



Dans E , le produit scalaire des deux vecteurs \vec{x} et \vec{y} s'écrit :

$$\langle \vec{x}, \vec{y} \rangle = \sum_{i=1}^m \sum_{j=1}^m \langle \vec{b}_i, \vec{b}_j \rangle x^i y^j \quad (1.2)$$

Intéressons nous maintenant aux propriétés des matrices de Gram.

Définition 1.3.2.1. La *matrice de Gram* associée à un ensemble V de n vecteurs $\vec{v}_1, \dots, \vec{v}_n$ sur un espace vectoriel muni d'un produit scalaire est notée $G(V)$. Elle est définie par l'équation (1.3)

$$G(V) = \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_1, \vec{v}_2 \rangle & \dots & \langle \vec{v}_1, \vec{v}_n \rangle \\ \langle \vec{v}_2, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & \dots & \langle \vec{v}_2, \vec{v}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{v}_n, \vec{v}_1 \rangle & \langle \vec{v}_n, \vec{v}_2 \rangle & \dots & \langle \vec{v}_n, \vec{v}_n \rangle \end{pmatrix} \quad (1.3)$$

Un élément de cette matrice s'écrit :

$$g_{ij}(V) = \langle \vec{v}_i, \vec{v}_j \rangle \quad (1.4)$$

Soient deux vecteurs \vec{w} et \vec{z} , définis par une combinaison linéaire des vecteurs \vec{v}_i , de la manière suivante :

$$\vec{w} = \sum_{i=1}^n w^i \vec{v}_i \quad (1.5)$$

$$\vec{z} = \sum_{i=1}^n z^i \vec{v}_i \quad (1.6)$$

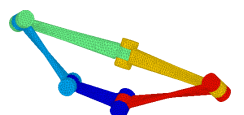
Le produit scalaire des deux vecteurs \vec{w} et \vec{z} s'écrit :

$$\langle \vec{w}, \vec{z} \rangle = \sum_{i=1}^n \sum_{j=1}^n \langle \vec{v}_i, \vec{v}_j \rangle w^i z^j = \sum_{i=1}^n \sum_{j=1}^n g_{ij}(V) w^i z^j \quad (1.7)$$

Les termes de la matrice de Gram $G(V)$ permettent de spécifier les normes des vecteurs et les angles entre ces derniers. On constate donc que la matrice de Gram d'un système contient tous les paramètres utiles (longueurs, angles).

$$\|\vec{v}_i\| = \sqrt{g_{ii}(V)} \quad (1.8)$$

$$\cos(\vec{v}_i, \vec{v}_j) = \frac{g_{ij}(V)}{\sqrt{g_{ii}(V)} \cdot \sqrt{g_{jj}(V)}} = \frac{g_{ij}(V)}{\|\vec{v}_i\| \|\vec{v}_j\|} \quad (1.9)$$



où le symbole $\|\cdot\|$ représente la norme d'un vecteur.

Il faut noter que :

- Le rang de $G(V)$ vaut la dimension de l'espace engendré par les n vecteurs ;
- Cette matrice est symétrique et semi définie positive ;
- Chaque terme diagonal vaut $\|\vec{v}_i\|$.

Pour des raisons pratiques, nous écrivons les vecteurs \vec{v}_i comme suit :

$$\vec{v}_i = L^i \vec{e}_i \quad \text{avec} \quad \|\vec{e}_i\| = 1 \quad (1.10)$$

Ainsi, la grandeur L^i représente la longueur du vecteur \vec{v}_i .

En substituant l'équation (1.10) dans l'expression (1.3), alors :

$$G(V) = \begin{pmatrix} (L^1)^2 \langle \vec{e}_1, \vec{e}_1 \rangle & L^1 L^2 \langle \vec{e}_1, \vec{e}_2 \rangle & \dots & L^1 L^n \langle \vec{e}_1, \vec{e}_n \rangle \\ L^2 L^1 \langle \vec{e}_2, \vec{e}_1 \rangle & (L^2)^2 \langle \vec{e}_2, \vec{e}_2 \rangle & \dots & L^2 L^n \langle \vec{e}_2, \vec{e}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ L^n L^1 \langle \vec{e}_n, \vec{e}_1 \rangle & L^n L^2 \langle \vec{e}_n, \vec{e}_2 \rangle & \dots & (L^n)^2 \langle \vec{e}_n, \vec{e}_n \rangle \end{pmatrix} \quad (1.11)$$

On définit une nouvelle matrice de Gram $G(N)$ associée à l'ensemble N des n vecteurs $\vec{e}_1, \dots, \vec{e}_n$. On l'appelle matrice de Gram *unitaire*. Elle s'écrit :

$$G(N) = \begin{pmatrix} \langle \vec{e}_1, \vec{e}_1 \rangle & \langle \vec{e}_1, \vec{e}_2 \rangle & \dots & \langle \vec{e}_1, \vec{e}_n \rangle \\ \langle \vec{e}_2, \vec{e}_1 \rangle & \langle \vec{e}_2, \vec{e}_2 \rangle & \dots & \langle \vec{e}_2, \vec{e}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{e}_n, \vec{e}_1 \rangle & \langle \vec{e}_n, \vec{e}_2 \rangle & \dots & \langle \vec{e}_n, \vec{e}_n \rangle \end{pmatrix} = \begin{pmatrix} 1 & \cos(\vec{e}_1, \vec{e}_2) & \dots & \cos(\vec{e}_1, \vec{e}_n) \\ \cos(\vec{e}_2, \vec{e}_1) & 1 & \dots & \cos(\vec{e}_2, \vec{e}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\vec{e}_n, \vec{e}_1) & \cos(\vec{e}_n, \vec{e}_2) & \dots & 1 \end{pmatrix} \quad (1.12)$$

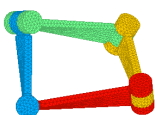
On définit également la matrice diagonale L de taille $n \times n$:

$$L = \begin{pmatrix} L^1 & 0 & \dots & 0 \\ 0 & L^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & L^n \end{pmatrix} \quad (1.13)$$

Finalement, la relation entre $G(V)$, $G(N)$ et L est :

$$G(V) = L \cdot G(N) \cdot L \quad (1.14)$$

A noter que :



- Le rang de $G(N)$ vaut la dimension de l'espace engendré par les n vecteurs ;
- Cette matrice est symétrique et semi définie positive ;
- Chaque terme diagonal vaut $\|\vec{e}_i\| = 1$.
- Un terme $g_{ij}(N)$ hors diagonal de la matrice vaut le cosinus de l'angle entre \vec{e}_i et \vec{e}_j .

Toutes les spécifications angulaires sont dans $G(N)$; toutes les spécifications de longueurs sont dans L .

1.3.2.2 Méthode des coupes métriques

Cette méthode consiste à découper la matrice de Gram en sous-matrices puis à établir les relations de fermeture. A noter que la méthode des coupes métriques s'applique à une chaîne fermée.

Soit V un ensemble de n vecteurs d'un espace euclidien et V_1 et V_2 deux sous-ensembles complémentaires de V ($V_1 \cap V_2 \neq \emptyset$ et $V_1 \cup V_2 = V$). Chaque sous-ensemble V_i est associé à une quantité nommée *coupe métrique*, correspondant à la somme de tous les termes de la sous-matrice de Gram associée.

Lorsque l'ensemble V forme une somme vectorielle nulle, les coupes métriques de toutes les paires de sous-ensembles complémentaires sont égales. Plus précisément, si $\vec{v}_1 + \dots + \vec{v}_n = \vec{0}$, il y a :

$$\forall k \in [1, n], (\vec{v}_1 + \dots + \vec{v}_k) = -(\vec{v}_{k+1} + \dots + \vec{v}_n)$$

$$(\vec{v}_1 + \dots + \vec{v}_k)^2 = (\vec{v}_{k+1} + \dots + \vec{v}_n)^2$$

Pour un contour fermé composé de n vecteurs, une matrice de Gram non-unitaire est de la forme :

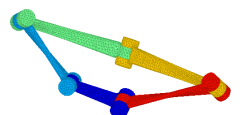
$$G(V) = \left(\begin{array}{ccc|ccc} \langle \vec{v}_1, \vec{v}_1 \rangle & \dots & \langle \vec{v}_1, \vec{v}_k \rangle & \langle \vec{v}_1, \vec{v}_{k+1} \rangle & \dots & \langle \vec{v}_1, \vec{v}_n \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \vec{v}_k, \vec{v}_1 \rangle & \dots & \langle \vec{v}_k, \vec{v}_k \rangle & \langle \vec{v}_k, \vec{v}_{k+1} \rangle & \dots & \langle \vec{v}_k, \vec{v}_n \rangle \\ \hline \langle \vec{v}_{k+1}, \vec{v}_1 \rangle & \dots & \langle \vec{v}_{k+1}, \vec{v}_k \rangle & \langle \vec{v}_{k+1}, \vec{v}_{k+1} \rangle & \dots & \langle \vec{v}_{k+1}, \vec{v}_n \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \vec{v}_n, \vec{v}_1 \rangle & \dots & \langle \vec{v}_n, \vec{v}_k \rangle & \langle \vec{v}_n, \vec{v}_{k+1} \rangle & \dots & \langle \vec{v}_n, \vec{v}_n \rangle \end{array} \right) \quad (1.15)$$

$$= \left(\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right)$$

La notion de coupe métrique permet d'établir la relation suivante :

$$\sum (A_k)_{ij} = \sum (D_k)_{mp} \quad (1.16)$$

Pour exprimer algébriquement la condition de fermeture vectorielle en 3D, il faut donc écrire que trois coupes métriques appliquées à la matrice de Gram non-unitaire sont nulles. L'inconvénient de cette méthode est le nombre de paramètres qu'elle fait intervenir lorsque la boucle vectorielle contient un grand nombre de vecteurs. En effet, les paramètres sont différents si la coupure change. Les équations obtenues sont souvent dépendantes.



1.3.2.3 Méthode des projections

Cette méthode consiste à écrire que les projections du vecteur somme sur d vecteurs indépendants sont nulles (d'où d est la dimension de l'espace).

Dans une boucle fermée, la somme des vecteurs est nulle.

$$\sum_i L^i \vec{e}_i = \vec{0} \quad \text{avec} \quad \|\vec{e}_i\| = 1$$

En conséquence, la projection de cette boucle fermée sur n'importe quel vecteur est égale à zéro.

$$\sum_{i,k} L^i \langle \vec{e}_i, \vec{e}_k \rangle = 0 \quad (1.17)$$

La difficulté de cette méthode est de choisir d vecteurs indépendants lorsque les vecteurs sont écrits de manière formelle. Pour simplifier le choix, nous pouvons projeter la boucle sur tous les vecteurs qui appartiennent à la boucle. Ceci induit que ces n équations sont dépendantes. L'équation (1.17) peut être exprimée avec la matrice de Gram et les longueurs des vecteurs de la manière suivante :

$$G(N) \cdot L \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \langle \vec{e}_1, \vec{e}_1 \rangle & \langle \vec{e}_1, \vec{e}_2 \rangle & \dots & \langle \vec{e}_1, \vec{e}_n \rangle \\ \langle \vec{e}_2, \vec{e}_1 \rangle & \langle \vec{e}_2, \vec{e}_2 \rangle & \dots & \langle \vec{e}_2, \vec{e}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{e}_n, \vec{e}_1 \rangle & \langle \vec{e}_n, \vec{e}_2 \rangle & \dots & \langle \vec{e}_n, \vec{e}_n \rangle \end{pmatrix} \cdot \begin{pmatrix} L^1 \\ L^2 \\ \vdots \\ L^n \end{pmatrix} = 0 \quad (1.18)$$

Comme la méthode précédente, la méthode de projection génère un nombre important d'équations dépendantes entre elles. Il est difficile d'extraire les équations indépendantes quand elles sont écrites de manière symbolique.

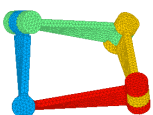
1.3.2.4 Autres approches non-cartésiennes

1.3.2.4.1 Approche Cayley-Menger

À l'origine, le déterminant de Cayley-Menger est utilisé pour exprimer des relations entre les distances des points. L'idée principale est ne pas utiliser des coordonnées mais des distances. Lorsqu'un point est à 3 distances de 3 points connus, alors il est connu.

Pour n points p_1, \dots, p_n , soit $d_{ij} = (p_j - p_i)^2$, le Déterminant de Cayley-Menger est défini par [Riv06] :

$$D(p_1, \dots, p_n) = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & d_{12} & d_{13} & d_{14} & \dots & d_{1n} \\ 1 & d_{21} & 0 & d_{23} & d_{24} & \dots & d_{2n} \\ 1 & d_{31} & d_{32} & 0 & d_{34} & \dots & d_{3n} \\ 1 & d_{41} & d_{42} & d_{43} & 0 & \dots & d_{4n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & d_{n1} & d_{n2} & d_{n3} & d_{n4} & \dots & 0 \end{vmatrix} \quad (1.19)$$



Théorème Pour un n -uplet des points p_1, \dots, p_n intégré dans un espace Euclidien de dimension d , il est nécessaire et suffisant que

- pour n'importe quel m -uplet des points avec $m \geq d + 2$

$$D(p_1, \dots, p_m) = 0$$

- pour tout k -uplet ($2 \leq k \leq d + 1$) parmi n points

$$\text{signe } D(p_1, \dots, p_k) = (-1)^k \text{ ou } 0$$

Par exemple avec 5 points donnés dans l'espace euclidien 3D, nous avons $5 \geq 3 + 2$. Ainsi, la relation (1.20) doit être respectée [Fou04] :

$$D(p_1, \dots, p_5) = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & \dots \\ 1 & 0 & d_{12} & d_{13} & d_{14} & d_{15} \\ 1 & d_{21} & 0 & d_{23} & d_{24} & d_{25} \\ 1 & d_{31} & d_{32} & 0 & d_{34} & d_{35} \\ 1 & d_{41} & d_{42} & d_{43} & 0 & d_{45} \\ 1 & d_{51} & d_{52} & d_{53} & d_{54} & 0 \end{vmatrix} = 0 \quad (1.20)$$

1.3.2.4.2 Quaternions

Les quaternions ont été découverts par William Rowan Hamilton en 1843 et ensuite améliorés par les représentations de Josiah Willard Gibbs et Hermann Grassmann [Kha08]. Un quaternion γ est défini par $\gamma = \gamma_0 + \gamma_1 i + \gamma_2 j + \gamma_3 k$, où les composantes $\gamma_0, \gamma_1, \gamma_2$ et γ_3 sont des scalaires ; i, j et k sont les opérateurs qui satisfont les règles suivantes :

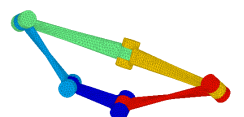
$$\begin{aligned} ii = jj = kk = ijk = -1, \\ ij = k, jk = i, ki = j, \\ ji = -k, kj = -i, ik = -j. \end{aligned}$$

Le conjugué du quaternion $\gamma = \gamma_0 + \gamma_1 i + \gamma_2 j + \gamma_3 k$ est le quaternion obtenu en conservant sa partie scalaire et en prenant l'opposé de sa partie vectorielle $\bar{\gamma} = \gamma_0 - \gamma_1 i - \gamma_2 j - \gamma_3 k$. Les quaternions unitaires sont, par définition, les quaternions γ de norme 1. Ils peuvent être utilisés pour décrire l'orientation. Soit, par exemple, un vecteur spatial $\mathbf{v} = (v_x, v_y, v_z)^T$ sous forme du quaternion noté $\mathbf{v} = v_x i + v_y j + v_z k$, si ϕ est un quaternion unitaire, le produit du quaternion $\phi \mathbf{v} \bar{\phi}$ effectue une rotation spatiale du vecteur \mathbf{v} autour de l'axe $(\phi_1, \phi_2, \phi_3)^T$.

1.3.2.4.3 Matrice de rotation

Une matrice de rotation Q est une matrice carrée avec des entités réelles qui effectuent une rotation dans un espace euclidien. Q est aussi une matrice orthogonale de déterminant 1, ce qui peut s'exprimer par les équations suivantes : $Q^t Q = I = Q Q^t$ et $\det(Q) = 1$, où Q^t est la matrice transposée de Q , et I est la matrice unité. Par exemple une rotation d'un angle θ autour de l'axe z est donnée par l'équation (1.21).

$$Q = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.21)$$



1.3.2.4.4 Angle d'Euler

Nous nous intéressons seulement ici à la description du mouvement du solide en rotation quelconque autour du point O , qui peut être un point fixe du solide dans le référentiel d'origine O_{xyz} (figure 1.8). L'orientation de O_{xyz} à $O_{x'y'z'}$ est indiquée comme un vecteur de trois angles. Les angles d'Euler sont choisis de façon à permettre une mémorisation simple de la construction du vecteur rotation instantané du solide. On passe du référentiel fixe O_{xyz} au référentiel lié au solide $O_{x'y'z'}$ par trois rotations successives.

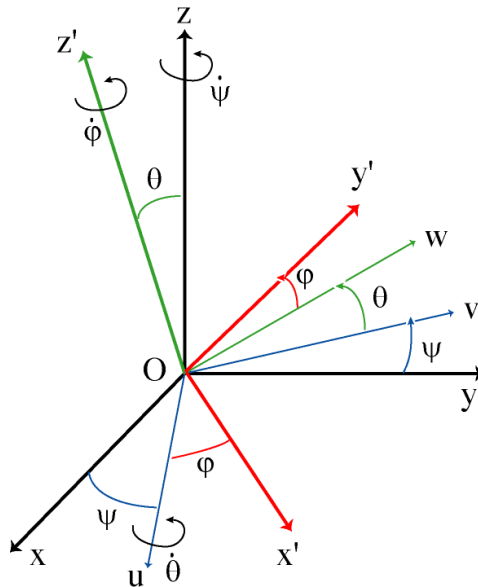


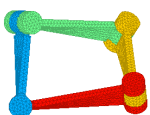
FIGURE 1.8 – Les angles d'Euler

- La précession ψ , autour de l'axe O_z , fait passer de O_{xyz} au référentiel O_{uvw} .
- La rotation θ , autour de l'axe O_u , fait passer de O_{uvw} à $O_{uvw'}$.
- La rotation propre φ , autour de l'axe $O_{z'}$, fait passer de $O_{uvw'}$ au référentiel lié au solide $O_{x'y'z'}$.

1.4 Méthodes d'analyse

Lorsque les équations de fermeture $C(x, u, m) = 0$ sont écrites, il faut les analyser et/ou les résoudre pour caractériser les propriétés du système. Les systèmes d'équations utilisées en CAO sont quelconques : linéaires, non linéaires, polynomiaux, algébriques ou transcendants, ce qui accroît la difficulté de la résolution. Souvent, une résolution efficace nécessite une analyse du système d'équations et une méthode adaptée.

Nous allons voir quelques méthodes d'analyse puis les méthodes de résolution.



1.4.1 Méthodes d'analyse de structure

La *théorie de la rigidité* apporte une méthode à l'analyse de structure. Elle étudie la rigidité d'assemblages de tiges métalliques reliées par des joints flexibles (liaisons rotules) dans un plan ou dans un espace 3D [Imb13]. Ces tiges, comme les barres de solides dans nos études, ne sont pas pliables, ni élastiques. Dans un assemblage, un jeu de longueurs des tiges est générique quand les longueurs sont algébriquement indépendantes. Selon les hypothèses émises par Hendrickson [Hen92], nous avons :

- ① Si un assemblage est rigide pour un jeu de longueurs génériques, il l'est pour tous les jeux de longueurs génériques [Glu71].
- ② Si un assemblage est rigide pour un jeu de longueurs génériques, il l'est presque pour tous les jeux de longueurs.
- ③ Si un assemblage est rigide pour un jeu de longueurs, il l'est presque pour tous les jeux de longueurs.

Différentes méthodes permettent de caractériser la constriction structurelle. Nous pouvons citer Cauchy [Cau31], Laman [Lam70] et la méthode du témoin publiée par Michelucci [Fou06, Fou07].

1.4.2 Méthodes d'analyse de mécanisme

L'étude des mécanismes est toujours possible à partir d'une analyse statique ou cinématique exhaustive. Cependant, celles-ci sont souvent longues et fastidieuses. Nous allons voir donc des formules globales qui permettront une analyse rapide d'un mécanisme du point de vue de la mobilité.

1.4.2.1 Formule de Grübler

Le formule de Grübler caractérise le degré de mobilité d'un mécanisme en toutes ses configurations *génériques*. En 1917, Grübler [Grü17, Ang82] a proposé une formule combinatoire pour le calcul du degré de mobilité d'un mécanisme indépendante de sa configuration :

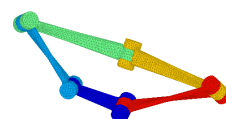
Formule de Grübler

Un mécanisme (S, L) ³ en dimension d a pour degré de mobilité $m = \frac{d(d+1)}{2} \cdot (|S| - |L| - 1) + \sum_{l \in L} ddl(l)$, où $ddl(l)$ est le nombre de degrés de liberté de la liaison l .

Le principe de la formule de Grübler est que le degré de mobilité d'un mécanisme est égal à la somme des degrés de liberté offerts par ses pièces mécaniques hormis le bâti $\frac{d(d+1)}{2} \cdot (|S| - 1)$ soustrait du nombre de degrés de liberté retirés par les liaisons $\frac{d(d+1)}{2} \cdot |L| - \sum_{l \in L} ddl(l)$.

La figure 1.9 présente le mécanisme 3D de Bennett constitué de 4 pièces (des solides) et de 4 liaisons pivots. Le degré de mobilité de ce mécanisme selon la formule de Grübler est $m = -2$ indiquant que le mécanisme devrait être une structure rigide. Cependant, la configuration du mécanisme présentée sur cette figure, où les 2 barres face à face sont de même longueur et où

3. avec S le nombre de pièces, L le nombre de liaison



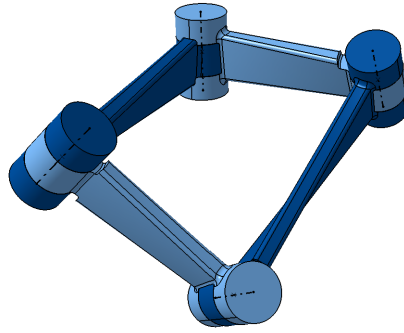


FIGURE 1.9 – Mécanisme de Bennett à 1 degré de liberté

les axes des liaisons pivots forment des angles particuliers, confère à ce mécanisme un unique degré de liberté. Ainsi, la formule de Grübler permet donc de décider du degré de mobilité d'un mécanisme en général mais peut être fausse pour des mécanismes spécifiques comme le mécanisme du Bennett.

1.4.2.2 Approches mécaniques

1.4.2.2.1 Analyse cinématique La théorie des mécanismes traite uniquement de modèles ayant des liaisons parfaites sans jeux, ni frottement, ni déformations. Les liaisons sont définies par un torseur cinématique duquel découle le torseur d'action mécanique transmissible.

Le degré de mobilité (cinématique) d'un mécanisme, noté m_c , est le nombre de paramètre cinématique indépendants à choisir pour définir le mouvement. C'est donc la différence entre le nombre d'inconnues cinématiques du mécanisme et le nombre d'équations cinématiques indépendantes que l'on peut écrire [Gir09].

Soit un système modélisé par une chaîne simple fermée constituée de S solides (bâti compris) et L liaisons. Soit $\vec{V}_{M \in R_i/R_1}$ le champ des vecteur vitesses d'un solide R_i ; $\vec{\Omega}_{R_i/R_1}$ le champ des moments d'un torseur de résultate. La notation M, R_i précise que le point M appartient au solide R_i . On appelle *torseur cinématique* du solide R_i dans son mouvement par rapport à R_1 [Gir97]. On le note

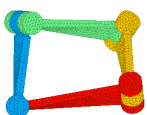
$$\{C_{R_i/R_1}\} = \left\{ \begin{array}{c} \vec{\Omega}_{R_i/R_1} \\ \vec{V}_{M \in R_i/R_1} \end{array} \right\}_M \text{ ou } \{C_{i/1}\} = \left\{ \begin{array}{c} \vec{\Omega}_{i/1} \\ \vec{V}_{M,i/1} \end{array} \right\}_M$$

Nous associons à chaque liaison le torseur cinématique $\{C_{i/i+1}\}$, voir la figure ci-dessous. La fermeture de chaîne cinématique permet d'écrire :

$$\{C_{0/1}\} + \{C_{1/2}\} + \dots + \{C_{i/i+1}\} + \dots + \{C_{S-1/S}\} + \{C_{S/0}\} = 0$$

Ce qui conduit à deux équations vectorielles :

$$\left\{ \begin{array}{l} \vec{\Omega}_{0/1} + \vec{\Omega}_{1/2} + \dots + \vec{\Omega}_{i/i+1} + \dots + \vec{\Omega}_{S-1/S} + \vec{\Omega}_{S/0} = \vec{0} \\ \vec{V}_{A \in 0/1} + \vec{V}_{A \in 1/2} + \dots + \vec{V}_{A \in i/i+1} + \dots + \vec{V}_{A \in S-1/S} + \vec{V}_{A \in S/0} = \vec{0} \end{array} \right.$$



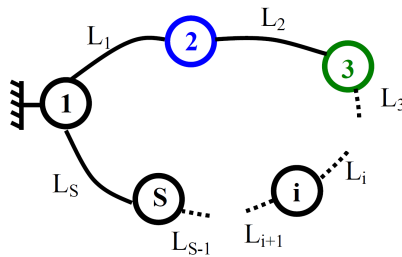


FIGURE 1.10 – Une chaîne fermée

Le point A étant judicieusement choisi pour donner les différentes expressions des vecteurs vitesses les plus simples possibles. Ces deux équations vectorielles conduisent à 6 équations scalaires qui doivent permettre de déterminer les N_C inconnues cinématiques. On appelle usuellement E_C le nombre d'équations scalaires issues des fermetures cinématiques. Ici, pour une seule fermeture cinématique, on a $E_C = 6$. Soit r_C le rang du système, c'est-à-dire le nombre d'équations indépendantes, issues des E_C équations de la cinématique.

On définit le degré de mobilité cinématique du mécanisme par la différence entre le nombre d'inconnues cinématiques et le rang du système d'équations de la cinématique :

$$m_C = N_C - r_C$$

m_C correspond au nombre d'inconnues cinématiques indépendantes qu'il faut se fixer pour déterminer les autres.

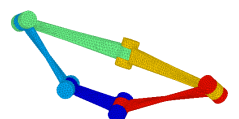
1.4.2.2 Analyse statique Le degré d'hyperstatisme (h) d'un mécanisme correspond au nombre d'inconnues statiques de liaisons que l'on ne peut calculer avec les équations issues du principe fondamental de la statique. C'est la différence entre le nombre d'inconnues de liaisons et le rang du système d'équations de la statique du système [Gir09]. Dans ce cas, on peut dire qu'il y a un excès de liaisons mécaniques sur le modèle du système. Après la modélisation des liaisons et par les seules équations, si toutes les composantes des actions mécaniques intervenant dans les liaisons du système peuvent être calculées par les seules équations de la statique alors le mécanisme est dit *isostatique*.

Pour un mécanisme constitué de p pièces, nous pouvons écrire $6(p - 1)$ équations qui lient les inconnues de liaison entre les solides⁴. Si le système est mobile d'ordre m , les $6(p - 1)$ équations obtenues sont de rang r_s tel que $6(p - 1) - r_s = m$, soit : $6(p - 1) - m = r_s$. La différence entre le nombre d'inconnues d'efforts de liaison, noté I_S , et le rang r_s du système d'équations cinématique correspond au degré d'hyperstatisme du mécanisme noté h : $I_S - (6(p - 1) - m) = h$.

1.4.3 Approche géométrique

Nous proposons une approche géométrique qui permet d'étudier l'assemblage et la mobilité d'un système grâce au *développement limité*. Cette approche d'analyse est utilisée dans ma publication qui se trouve au Chapitre 2. Le développement limité est une approche classique pour linéariser des fonctions. Il peut être appliqué sur les fonctions qui représentent les contraintes

4. On n'écrit pas les équations d'équilibre de la pièce support du type bâti ou carter



du système de solides, d'autre part, lorsque la dérivée seconde d'une fonction est nulle, il offre le moyen d'en connaître les extremas. Le développement limité est un outil pratique en termes d'analyse de la mobilité locale autour d'une configuration spécifique.

On suppose qu'un système de solides est défini par :

$$f_i(\mathbf{u}, \mathbf{m}) = 0, i = 1, 2, \dots \quad (1.22)$$

où $[\mathbf{u} \ \mathbf{m}] = [u_1 \ u_2 \ \dots \ u_p \ m_1 \ m_2 \ \dots \ m_q]$ est le vecteur des variables. Soit une configuration d'assemblage donnée, $[\mathbf{u}_0 \ \mathbf{m}_0] = [u_{01} \ u_{02} \ \dots \ u_{0p} \ m_{01} \ m_{02} \ \dots \ m_{0q}]$ tel que $f_i(\mathbf{u}_0, \mathbf{m}_0) = 0$. A partir de cette configuration, nous pouvons définir un déplacement \mathbf{h} tel que $[\mathbf{u} \ \mathbf{m}] = [\mathbf{u}_0 + \mathbf{h}_u \ \mathbf{m}_0 + \mathbf{h}_m]$. Ensuite, l'équation (1.22) se transforme en

$$f_i(\mathbf{u}_0 + \mathbf{h}_u, \mathbf{m}_0 + \mathbf{h}_m) = 0, i = 1, 2, \dots \quad (1.23)$$

avec $\mathbf{h} = [h_{u1} \ h_{u2} \ \dots \ h_{up} \ h_{m1} \ h_{m2} \ \dots \ h_{mq}]$.

Supposons l'équation (1.23) est $a + 1$ fois différentiable en $(\mathbf{u}_0, \mathbf{m}_0)$ dans \mathbb{R} , alors pour tout $(\mathbf{u}, \mathbf{m}) \in \mathbb{R}^{n+m}$, le développement limité est :

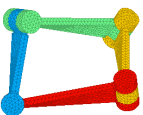
$$\begin{aligned} f_i(\mathbf{u}, \mathbf{m}) &= f_i(\mathbf{u}_0 + \mathbf{h}_u, \mathbf{m}_0 + \mathbf{h}_m) \\ &= f_i(\mathbf{u}_0, \mathbf{m}_0) + (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}}) f_i(\mathbf{u}_0, \mathbf{m}_0) \\ &\quad + \frac{1}{2!} (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}})^2 f_i(\mathbf{u}_0, \mathbf{m}_0) + \dots \\ &\quad + \frac{1}{a!} (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}})^a f_i(\mathbf{u}_0, \mathbf{m}_0) \\ &\quad + \frac{1}{(a+1)!} (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}})^{a+1} f_i(\mathbf{u}_0 + \beta \mathbf{h}_u, \mathbf{m}_0 + \beta \mathbf{h}_m) \end{aligned} \quad (1.24)$$

avec $i = 1, 2, \dots, n$ et $0 < \beta < 1$.

Or $f_i(\mathbf{u}_0, \mathbf{m}_0) = 0$. Dans l'équation (1.24), les termes d'ordre 2 et plus sont négligés, nous obtenons alors :

$$f_i(\mathbf{u}_0 + \mathbf{h}_u, \mathbf{m}_0 + \mathbf{h}_m) = (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}}) f_i(\mathbf{u}_0, \mathbf{m}_0), \quad i = 1, 2, \dots \quad (1.25)$$

L'équation (1.25) est considérée comme une approximation au 1^{er} ordre de l'équation (1.24). La condition nécessaire et suffisante pour que l'on puisse trouver une approximation du premier ordre de la fonction f_i en un point $(\mathbf{u}_0, \mathbf{m}_0)$ est que la fonction f_i soit dérivable en $(\mathbf{u}_0, \mathbf{m}_0)$. Cela traduit le fait que l'on peut approximer localement, c'est à dire dans un voisinage de $(\mathbf{u}_0, \mathbf{m}_0)$, $f_i(\mathbf{u}_0 + \mathbf{h}_u, \mathbf{m}_0 + \mathbf{h}_m)$ par $\mathbf{h} f'_i(\mathbf{u}_0, \mathbf{m}_0)$. A noter que l'équation (1.25) peut s'écrire sous forme matricielle :



$$\left[\begin{array}{cccc} \frac{\partial f_i(\mathbf{u}_0, \mathbf{m}_0)}{\partial u_1} & \cdots & \frac{\partial f_i(\mathbf{u}_0, \mathbf{m}_0)}{\partial u_p} & \frac{\partial f_i(\mathbf{u}_0, \mathbf{m}_0)}{\partial m_1} & \cdots & \frac{\partial f_i(\mathbf{u}_0, \mathbf{m}_0)}{\partial m_q} \end{array} \right] \begin{bmatrix} h_{u1} \\ \vdots \\ h_{up} \\ h_{m1} \\ \vdots \\ h_{mq} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

ou

$$\mathbf{Jh} = 0$$

où \mathbf{J} est la matrice jacobienne. De la même manière, l'approximation du 2^{nd} ordre s'écrit

$$f_i(\mathbf{u}_0 + \mathbf{h}_u, \mathbf{m}_0 + \mathbf{h}_m) = (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}}) f_i(\mathbf{u}_0, \mathbf{m}_0) + \frac{1}{2!} (\mathbf{h}_u \frac{\partial}{\partial \mathbf{u}} + \mathbf{h}_m \frac{\partial}{\partial \mathbf{m}})^2 f_i(\mathbf{u}_0, \mathbf{m}_0) \quad (1.26)$$

Ainsi, la mobilité locale d'ordre 1 et 2 d'un mécanisme autour d'une configuration $(\mathbf{u}_0, \mathbf{m}_0)$ est le nombre de h_i qui peuvent être choisis librement dans l'équation (1.25) et (1.26), voir [Che11]. La mobilité locale d'ordre ∞ autour de la configuration $(\mathbf{u}_0, \mathbf{m}_0)$ est équivalente à la mobilité globale dans une zone, y compris le point de la configuration $(\mathbf{u}_0, \mathbf{m}_0)$.

Puisque nous cherchons les conditions qui rendent les systèmes de solides globalement mobiles, le développement limité en série Taylor est également un outil de vérification des résultats.

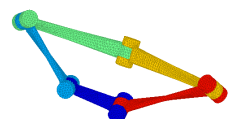
1.5 Méthodes de résolution

1.5.1 Méthodes symboliques

L'objectif des méthodes symboliques est de donner la résolution exacte et formelle d'un système d'équations. Elles fournissent l'ensemble des solutions sous forme symbolique sans erreur d'arrondi numérique. Elles ont deux avantages significatifs : la correction et la complétude. Les méthodes symboliques sont nombreuses : la méthode de Ritt-Wu [Wu86, Cho98], la méthode des résultants [Bon00], la méthode à partir des bases de Gröbner, qui est un apport de ma thèse. Nous avons aussi des méthodes à base de règles géométriques [Mid96] comme, par exemple les méthodes de construction formelle à la règle et au compas. Nous pouvons classer parmi ces méthodes l'approche de Mathis [Mat97, Sch98]. Le logiciel de calcul formel Maple[®] a été utilisé dans ce rapport puisque certaines méthodes symboliques qui y sont décrites ont été industrialisées.

1.5.1.1 Élimination et méthodes du résultant

Les résultants sont des expressions polynomiales dans les coefficients d'un système d'équations polynomiales qui sont dérivables après l'élimination des variables. L'importance des résultants



réside dans le fait que l'élimination d'une variable est une condition nécessaire et suffisante pour que le système ait une solution en la variable éliminée [Alf08]. Ainsi, le résultant de deux polynômes permet de vérifier s'ils possèdent des facteurs communs. Il existe deux expressions de résultant pour deux polynômes $f(x) = a_m x^m + \dots + a_0$ et $g(x) = b_n x^n + \dots + b_0$ d'ordre m et n : *résultant de Sylvester* et *résultant de Bézout*. Un des objectifs du théorème d'élimination est de développer des algorithmes pour calculer les résultants d'un système d'équations polynomial en multi-variables basé sur les résultants.

Le résultant de Sylvester et Bézout permettent d'éliminer une seule variable par itération, tandis que le résultant de Dixon et Macaulay peuvent éliminer plusieurs variables. Le désavantage de ces méthodes est que le temps de calcul est une fonction exponentielle de la taille de matrice. A cause de cette complexité exponentielle, nous sommes souvent limités par des modèles complexes comme le mécanisme de Bricard [Dah09]. La limite d'un ordinateur normal (1.95G de RAM, Processeur 2.53GHz) est le calcul du déterminant de 16×16 en formel. Nous avons testé la méthode de Sylvester avec l'exemple de Bennett. Pour cet exemple, le calcul est bloqué au niveau du déterminant d'une matrice de taille 18×18 .

1.5.1.1.1 Résultant de Sylvester

Le calcul de résultant de Sylvester est une méthode d'élimination bien connue. Soient deux polynômes

$$f(x) = a_m x^m + \dots + a_0, \quad g(x) = b_n x^n + \dots + b_0$$

Éliminer x entre ces deux équations polynômiales revient à trouver une condition vérifiée par les coefficients des deux équations, nécessaire et suffisante pour que ces deux équations aient au moins une racine commune. Cette condition est donnée par le résultant de Sylvester. Le résultant de Sylvester peut être calculé à partir des coefficients des polynômes à l'aide d'un déterminant.

La matrice de Sylvester est une matrice de taille $(m+n) \times (m+n)$. La première ligne est formée des coefficients de $f(x)$ suivis de $m-1$ zéros

$$\left(a_m \quad a_{m-1} \quad \dots \quad a_1 \quad a_0 \quad 0 \quad \dots \quad 0 \quad 0 \right)$$

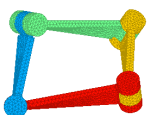
La seconde s'obtient à partir de la première en décalant les coefficients de $f(x)$ d'une case vers la droite :

$$\left(0 \quad a_m \quad a_{m-1} \quad \dots \quad a_1 \quad a_0 \quad 0 \quad \dots \quad 0 \right)$$

Les $n-2$ suivantes s'obtiennent par le même décalage jusqu'à obtenir la ligne :

$$\left(0 \quad \dots \quad 0 \quad 0 \quad a_m \quad a_{m-1} \quad \dots \quad a_1 \quad a_0 \right)$$

On complète ensuite la matrice de Sylvester en appliquant la même construction au polynôme $g(x)$. Ainsi, pour que $f(x)$ et $f(x)$ aient une racine commune, le déterminant de la matrice de Sylvester doit être égale à zéro.



$$Res_x(f, g) = \begin{vmatrix} a_m & a_{m-1} & \dots & a_0 & 0 & 0 & \dots & 0 \\ 0 & a_m & a_{m-1} & \dots & a_0 & 0 & \dots & 0 \\ 0 & 0 & a_m & a_{m-1} & \dots & a_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & 0 \\ 0 & 0 & 0 & \dots & a_m & a_{m-1} & \dots & a_0 \\ b_n & b_{n-1} & \dots & b_0 & 0 & 0 & \dots & 0 \\ 0 & b_n & b_{n-1} & \dots & b_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & b_n & b_{n-1} & \dots & b_0 \end{vmatrix} = 0 \quad (1.27)$$

Le déterminant (1.27) est appelé le *résultant* de f et g et se note $Res_x(f, g)$.

1.5.1.1.2 Déterminant de Bézout

Définition 1.5.1.1. Matrice de Bézout

Prenons encore les deux polynômes

$$f(x) = a_m x^m + \dots + a_0, \quad g(x) = b_n x^n + \dots + b_0$$

où $a_m, \dots, a_0, b_n, \dots, b_0$ sont des polynômes en autres variables, y par exemple.

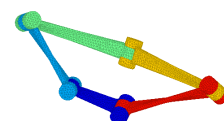
Bézout indique que l'élimination de x se réduit à trouver une fonction de x , par laquelle multipliant la première équation, et une autre fonction de x , par la quelle multipliant la seconde, la somme des deux produits soit telle que chaque puissance de x disparaisse [Alf]. Bézout multiplie la première équation par un polynôme en x à coefficients indéterminés de degré $(n - 1)$, et la deuxième par un polynômes de degré $(m - 1)$, et fait la somme des deux produits. Il obtien un polynôme en x de degré $(m + n - 1)$, dont les coefficients sont des combinaisons linéaires de $a_m, \dots, a_0, b_n, \dots, b_0$. En suivant le même schéma, les puissances de x peuvent être supprimées l'une après l'autre. Autrement dit, pour deux équations $\begin{cases} f(x) = 0 \\ g(x) = 0 \end{cases}$ La méthode consiste à postuler que l'équation finale est une équation de la forme $fr + gs = 0$. On cherche alors à déterminer les polynômes multiplicateurs r et s , ainsi que leur degré pour ramener à un problème linéaire.

Sachant que $m > n$, on appelle matrice de Bézout associée à deux polynômes $f(x)$ et $g(x)$, la matrice symétrique

$$B(f, g) = \begin{pmatrix} c_{0,0} & \dots & c_{0,m-1} \\ \vdots & \ddots & \vdots \\ c_{m-1,0} & \dots & c_{m-1,m-1} \end{pmatrix}$$

où les $c_{i,j}$ sont définis par l'expression de Cayley :

$$b(x, y) = \frac{f(x)g(y) - f(y)g(x)}{x - y}$$



telle que la fraction rationnelle $b(x, y)$ est une fonction à deux variables qui s'écrit de la manière suivante :

$$b(x, y) = \sum_{i,j=0}^{m-1} c_{i,j} x^i y^j$$

Le résultant de Bézout est le déterminant de la matrice $B(f, g)$.

Étant donné un système de $n + 1$ équations polynomiales $P_i(X) = f_i(x_1, x_2, \dots, x_n) = 0$, $i = 1, \dots, n + 1$, les méthodes permettant d'éliminer simultanément plusieurs variables sont *résultant de Dixon* et *résultant de Macaulay*.

1.5.1.1.3 Résultant de Dixon

Soit $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ un nouvel ensemble de variables et

$$\delta(\hat{X}) = \begin{vmatrix} Q_{1,1} & \dots & Q_{1,n+1} \\ Q_{2,1} & \dots & Q_{2,n+1} \\ \vdots & \vdots & \vdots \\ Q_{n,1} & \dots & Q_{n,n+1} \\ P_1(\hat{X}) & \dots & P_{n+1}(\hat{X}) \end{vmatrix}$$

où $Q_{j,1} = (f_i(\hat{x}_1, \dots, \hat{x}_{j-1}, x_j, \dots, x_n) - f_i(\hat{x}_1, \dots, \hat{x}_j, x_{j+1}, \dots, \hat{x}_n)) / (x_j - \hat{x}_j)$. $\delta(\hat{X})$ est connu comme polynôme de Dixon. Soit F un ensemble de polynômes composés de l'ensemble de tous les coefficients des termes de $\delta(\hat{X})$. La matrice des coefficient de F est la matrice de Dixon et son déterminant est vu comme le résultant de Dixon [Sax95].

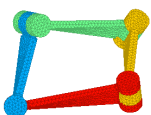
1.5.1.1.4 Résultant de Macaulay

Soit $1 \leq i \leq n + 1$, d_i le degré total du polynôme $P_i(X)$ et $d_m = 1 + \sum_1^{n+1} (d_i - 1)$, T est l'ensemble de degré d_m des variables x_1, x_2, \dots, x_n , autrement dit, $T = \{x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} | \alpha_1 + \alpha_2 + \dots + \alpha_n = d_m\}$. Maintenant, soit $T^{(i)}$ les termes de degré $d_m - d_i$ qui ne sont pas divisibles par $\{x_1^{d_1}, x_2^{d_2}, \dots, x_{i-1}^{d_{i-1}}\}$. Ensuite, le produit des termes dans $T^{(i)}$ par $P_i(X)$ donne un ensemble de polynômes dont les coefficients sont formés de la matrice de Macaulay. Le déterminant de cette matrice est le résultant de Macaulay [Li09].

Pour la même raison que le résultant de Sylvester, nous n'allons pas utiliser le résultant de Dixon et Macaulay. De plus, ces deux méthodes n'ont pas été industrialisées et le développement d'algorithme n'est pas le but de cette thèse.

1.5.1.2 Bases de Gröbner

L'utilisation des bases de Gröbner pour trouver les conditions d'assemblage et de mobilité est un des apports principaux de ma thèse. Ainsi, les bases de Gröbner sont ici définies en détail.



La division euclidienne et l'algorithme d'Euclide pour le PGCD⁵ sont des outils algorithmiques centraux en algèbre commutative computationnelle et en calcul formel sur les polynômes en une seule indéterminée. On s'attend à trouver la même universalité d'applications dans le cas de plusieurs indéterminées.

La théorie algorithmique bien adaptée pour le cas de plusieurs indéterminées est la théorie des bases de Gröbner, utilisée initialement pour les idéaux de polynômes de plusieurs indéterminées par B. Buchberger dans les années 1960. La méthode des bases de Gröbner est l'une des plus classiques en approche formelle. L'algorithme pour les calculer est aujourd'hui appelé algorithme de Buchberger. En plus, il existe des produits industriels, Maple[®] et Magma[®] par exemple, dans lesquels l'algorithme des Bases de Gröbner a été intégré.

Le champ des applications des Bases de Gröbner est vaste et varié : démonstration de théorèmes géométriques⁶ [Cho88, Kap86, Win90], géométrie algébrique algorithmique [Din], théorie des invariants [Ta], étude de systèmes hypergéométriques [Salc], manipulation de fonctions spéciales générales [Sala], sommation et intégration symboliques [Salb], preuve de théorèmes géométriques assistée par ordinateur [EIG], ...

Les bases de Gröbner vont être présentées en deux temps. Dans une première partie, nous rappelons les définitions algébriques et la théorie des bases de Gröbner ; nous donnerons ensuite une application de la base de Gröbner.

Définition 1.5.1.2. Anneaux

On appelle anneau $(\mathbb{A}, +, \times)$ tout triplet constitué d'un ensemble non vide \mathbb{A} , muni de deux lois de composition interne $+$ et \times sur \mathbb{A} qui vérifient :

- $(\mathbb{A}, +)$ est un groupe commutatif ;
- \times est associative ;
- \times admet un élément neutre dans \mathbb{A} ;
- \times est distributive par rapport à $+$.

De plus, l'anneau $(\mathbb{A}, +, \times)$ est dit commutatif si la loi \times est commutative. Dans la suite, par abus de langage, on dit que \mathbb{A} est un anneau lorsqu'il n'y a pas d'ambiguïté sur les lois de composition interne $+$ et \times .

Soit \mathbb{K} un corps commutatif, on désigne par X_1, \dots, X_n les indéterminées et par $\mathbb{A} = \mathbb{K}[X_1, \dots, X_n]$ l'anneau des polynômes en ces indéterminées, à coefficients dans \mathbb{K} . La notation multi-exposant $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}$ intervient afin d'exprimer un polynôme de la forme $X^\alpha = X_1^{\alpha_1} X_2^{\alpha_2} \dots X_n^{\alpha_n}$

Définition 1.5.1.3. Soit $f \in \mathbb{K}[X_1, \dots, X_n]$ non identiquement nul, s'écrivant :

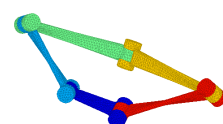
$$f = \sum c(\alpha_1, \dots, \alpha_n) X_1^{\alpha_1} X_2^{\alpha_2} \dots X_n^{\alpha_n}$$

où $c(\alpha_1, \dots, \alpha_n) \in \mathbb{K}$. L'ensemble $T(f)$ des termes de f est

$$T(f) = \{c(\alpha_1, \dots, \alpha_n) X_1^{\alpha_1} X_2^{\alpha_2} \dots X_n^{\alpha_n} \mid c(\alpha_1, \dots, \alpha_n) \neq 0\}$$

5. Plus Grand Commun Diviseur

6. Geometric theorem-proving



L'ensemble $M(f)$ des monômes de f est tel que

$$M(f) = \{X_1^{\alpha_1} X_2^{\alpha_2} \dots X_n^{\alpha_n} \mid c(\alpha_1, \dots, \alpha_n) \neq 0\}$$

Dans la suite, $M(X_1, \dots, X_n) = M$ est l'ensemble des monômes que l'on peut former avec les variables (X_1, \dots, X_n) .

Définition 1.5.1.4. (Ordre monomiaux) Un *ordre monomial* sur M est une relation d'ordre strict \prec qui est :

- totale : deux monômes peuvent toujours être comparés ;
- compatible avec le produit : dès que $m_1 \prec m_2$, on a $m' m_1 \prec m' m_2$ pour tout m' ;
- un bon ordre : tout ensemble non vide de monômes a un plus petit élément, ou de façon équivalente, toute suite strictement décroissante de monômes se termine.

En particulier, pour tout ordre monomial, on a la relation $1 \prec X_i$ pour chaque i . Sinon il y aura $X_i \prec 1$ pour certain i , puis de proche en proche $X_i^{k+1} \prec X_i^k$, d'où une suite infinie strictement décroissante. En conséquence, $1 = X_i^0 \dots X_n^0$ est le plus petit élément de M pour tout ordre monomial, car tout monôme m peut être obtenu comme dernier élément d'une chaîne

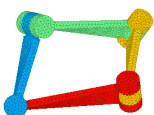
$$1 \prec X_{i_1} \prec X_{i_1} X_{i_2} \prec \dots \prec X_{i_1} \dots X_{i_r}$$

Pour la suite, nous adoptons les notations $|\alpha| = \alpha_1 + \dots + \alpha_n$, $X^\alpha = X_1^{\alpha_1} \dots X_n^{\alpha_n}$ et $X^{\alpha+\beta} = X^\alpha X^\beta$ pour tous multi-exposants $\alpha = (\alpha_1, \dots, \alpha_n)$ et $\beta = (\beta_1, \dots, \beta_n)$. On notera \preceq l'ordre large associé à l'ordre strict \prec .

En général, il y a trois ordres monomiaux employés sur des monômes et un ordre qui n'est pas un ordre monomial : l'ordre lexicographique inverse.

- *ordre lexicographique* (ordre du dictionnaire) : $X^\alpha \prec_{lex} X^\beta$ si $\alpha_k < \beta_k$ pour $k = \min\{i : \alpha_i \neq \beta_i\}$, ou autrement dit, si la première valeur non nulle de la suite $\alpha_1 - \beta_1, \alpha_2 - \beta_2, \dots$ est strictement négative.
- *ordre lexicographique gradué* (ordre du degré total raffiné par \prec_{glex}) : $X^\alpha \prec_{glex} X^\beta$ si $|\alpha| < |\beta|$ ou $(|\alpha| = |\beta|$ et $X^\alpha \prec_{lex} X^\beta)$.
- *ordre lexicographique inverse* (n'est pas un ordre monomial) : $X^\alpha \prec_{revlex} X^\beta$ si $\alpha_k > \beta_k$ pour $k = \max\{i : \alpha_i \neq \beta_i\}$, ou autrement dit, si la dernière valeur non nulle de la suite $\alpha_1 - \beta_1, \alpha_2 - \beta_2, \dots$ est strictement positive.
- *ordre lexicographique inverse gradué* (ordre du degré total raffiné par \prec_{revlex}) : $X^\alpha \prec_{revlex} X^\beta$ si $|\alpha| < |\beta|$ ou $(|\alpha| = |\beta|$ et $X^\alpha \prec_{revlex} X^\beta)$.

Pour tous ces ordres, on vérifie la relation $X_1 \succ X_2 \succ \dots \succ X_n$. Il est aisé de confondre ces différents ordres, aussi donnons-nous explicitement les quelques premiers termes de la suite ordonnée des monômes pour chacun de ces ordres. Il est nécessaire d'avoir au moins trois indéterminées



(n=3) et d'aller jusqu'au degré trois pour mettre en évidence les différences entre les ordres.

- *ordre lexicographique* : $1 \prec X_3 \prec X_3^2 \prec X_3^3 \prec \dots \prec X_2 \prec X_2X_3 \prec X_2X_3^2 \prec X_2X_3^3 \prec \dots \prec X_2^2 \prec X_2^2X_3 \prec X_2^2X_3^2 \prec X_2^2X_3^3 \prec \dots \prec X_1 \prec X_1X_3 \prec X_1X_3^2 \prec X_1X_3^3 \prec \dots \prec X_1X_2 \prec X_1X_2X_3 \prec X_1X_2X_3^2 \prec X_1X_2X_3^3 \prec \dots \prec X_1X_2^2 \prec X_1X_2^2X_3 \prec X_1X_2^2X_3^2 \prec X_1X_2^2X_3^3 \prec \dots$
- *ordre lexicographique gradué* : $1 \prec X_3 \prec X_2 \prec X_1 \prec X_3^2 \prec X_2X_3 \prec X_2^2 \prec X_1X_3 \prec X_1X_2 \prec X_1^2 \prec X_3^3 \prec X_2X_3^2 \prec X_2^2X_3 \prec X_3^2 \prec X_1X_3^2 \prec X_1X_2X_3 \prec X_1X_2^2 \prec X_1^2X_3 \prec X_1^2X_2 \prec X_1^3 \prec \dots$
- *ordre lexicographique inverse* : $\dots \prec X_1^3X_2^2X_3 \prec X_1^2X_2^2X_3 \prec X_1X_2^2X_3 \prec X_2^2X_3 \prec \dots \prec X_1^3X_3 \prec X_1^2X_3 \prec X_1X_3 \prec X_3 \prec \dots \prec X_1^3X_2^2 \prec X_1X_2^2 \prec X_2^2 \prec \dots \prec X_1^3X_2 \prec X_1X_2 \prec X_2 \prec \dots \prec X_1^3 \prec X_1^2 \prec X_1 \prec 1$
- *ordre lexicographique inverse gradué* : $1 \prec X_3 \prec X_2 \prec X_1 \prec X_3^2 \prec X_2X_3 \prec X_1X_3 \prec X_2^2 \prec X_1X_2 \prec X_1^2 \prec X_3^3 \prec X_2X_3^2 \prec X_2^2X_3 \prec X_1X_2X_3 \prec X_1^2X_3 \prec X_1X_2^2 \prec X_1^2X_2 \prec X_1^3 \prec \dots$

Encore une fois, l'ordre \prec_{revlex} n'est pas un ordre monomial : il fournit une suite infinie décroissante de monômes. En revanche, les ordres $\prec_{lex}, \prec_{grlex}, \prec_{grevlex}$ sont des ordres monomiaux. Une fois un ordre admissible \prec sur M fixé, il est facile de trier l'ensemble $M(f)$ des monômes de f .

Définition 1.5.1.5. Soit $M_{\prec}(f)$ l'ensemble $M(f)$ trié par \prec ,

- monôme de tête de f : noté $LM_{\prec}(f)$, le plus grand monôme de f ;
- $LC(f)$: coefficient de $LM_{\prec}(f)$ dans l'écriture monomiale de f ;
- terme de tête de f : noté $LT_{\prec}(f)$ de f , le terme correspondant au monôme de tête, est le produit $LC(f)LM_{\prec}(f)$.

Lorsqu'il n'y a pas d'ambiguïté sur l'ordre monomial utilisé, le monôme de tête, le terme de tête et le coefficient de tête de f seront simplement notés $LM(f)$, $LT(f)$ et $LC(f)$. La compatibilité de l'ordre monomial avec le produit entraîne la relation $LT(fg) = LT(f)LT(g)$ pour tous $f, g \in \mathbb{A}$. Nous pouvons maintenant définir les idéaux d'anneaux.

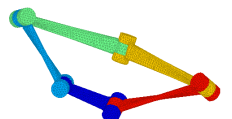
Définition 1.5.1.6. (Idéaux d'anneaux) Soit \mathbb{A} l'anneau des polynômes et f, g des équations polynomiales dans \mathbb{K} . Un idéal \mathbb{I} de \mathbb{A} est un sous-ensemble de \mathbb{A} tel que :

1. $\forall (f, g) \in \mathbb{I} \times \mathbb{I}, f + g \in \mathbb{I}$;
2. $\forall f \in \mathbb{I}, \text{ et } \forall p \in \mathbb{A}, fp \in \mathbb{I}$.

La définition suivante d'une base de Gröbner est purement mathématique et permet donc de définir une base canonique d'un idéal indépendamment de tout algorithme.

Définition 1.5.1.7. (Bases de Gröbner) Soit I un idéal de $\mathbb{Q}[X_1, \dots, X_n]$ et un ordre monomial sur \mathbb{A} étant fixé. On dit qu'un sous-ensemble fini $G = (g_1, \dots, g_k)$ est une base de Gröbner de I pour l'ordre admissible \prec si pour tout $f \in I$, il existe $1 \leq i \leq k$ tel que $LM(g_i)$ divise $LM(f)$.

Il n'y a pas unicité des bases de Gröbner. Par exemple, si G est une base de Gröbner d'un idéal I et si $g \in I$ alors $G \cup g$ est encore une base de Gröbner de I . En effet, si G est une base de



Gröbner d'un idéal I , alors G engendre I . De plus, une base de Gröbner d'un idéal I contient nécessairement un polynôme qui engendre I . Par exemple, si $f_1, \dots, f_n \in \mathbb{K}[x]$, alors une base de Gröbner de l'idéal engendrée par les f_i contient un PGCD des f_i .

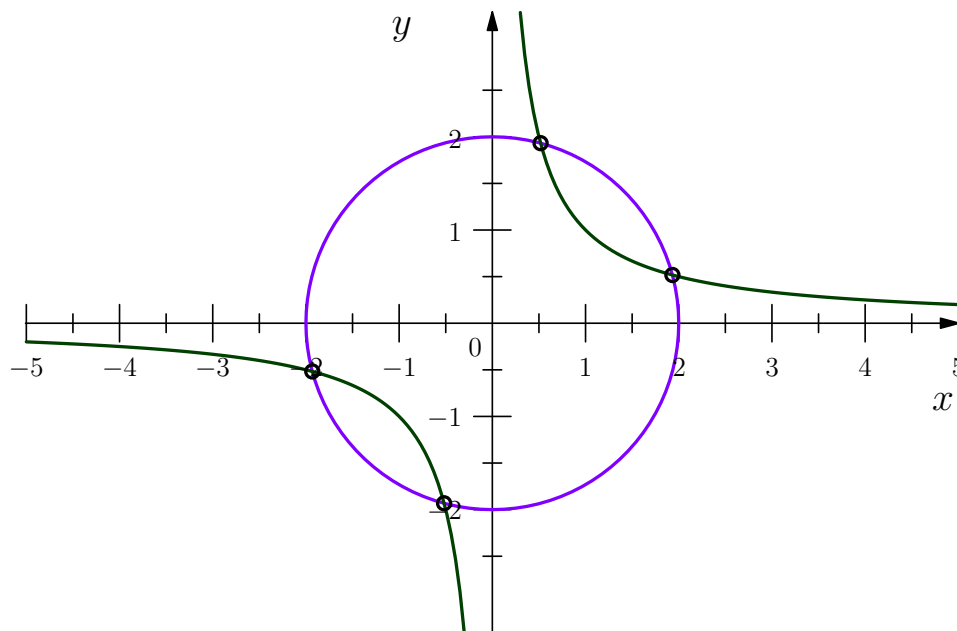


FIGURE 1.11 – L'intersection de $f_1 = 0$ et $f_2 = 0$

Nous avons choisi d'utiliser les bases de Gröbner parce qu'elles généralisent l'élimination incrémentale des indéterminées. Considérons le système suivant qui symbolise une équation d'hyperboles et une équation de cercle (cf. figure (1.11))

$$f_1 = x^2 + y^2 - 4, \quad f_2 = xy - 1$$

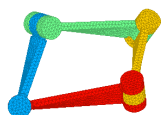
Résoudre ces deux équations revient à trouver les points d'intersection d'un cercle et d'une hyperbole. L'ordre lexicographique \prec est obtenu dans Maple par la fonction "plex" signifiant "pur lexicographie". Une base de Gröbner pour $plex(x, y)$ est donnée par le système

$$y^4 - 4y + 1, \quad x + y^3 - 4y.$$

L'intérêt des bases de Gröbner est qu'il n'y a plus de x dans le premier polynôme qui admet pour racines les ordonnées des points d'intersections. Le second permet de calculer les valeurs des abscisses correspondantes.

1.5.2 Méthodes numériques

Les méthodes numériques permettent de résoudre tous les systèmes d'équations mais font face au problème de l'inexactitude du calcul numérique. Dans cette section, nous ne décrivons que brièvement les méthodes numériques classiques car elles ne répondent pas aux critères attendus pour pouvoir résoudre un GCSP de manière formelle : elles ne fournissent qu'une solution du système à résoudre et ne sont donc pas complètes ; de plus elles produisent des solutions entachées de l'erreur d'arrondi numérique. Ces erreurs dépendent du conditionnement des matrices.



1.5.2.1 Méthodes itératives classiques

Les méthodes itératives, aussi appelées méthodes numériques locales, permettent de calculer une solution approchée. Parmi ces méthodes, la plus classique est la méthode de Newton-Raphson. Elle utilise la jacobienne J du système $F(X)$ à résoudre, pour s'approcher d'une solution grâce à la formule suivante : $X_{n+1} = X_n - J^{-1}(X_n)F(X_n)$. Cette méthode converge rapidement mais localement vers une solution sous certaines conditions (Théoreme de Newton-Kantorovitch [Ort68]).

1.5.2.2 Méthode d'homotopie

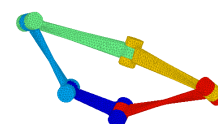
La méthode d'homotopie (ou par continuation homotopique) est destinée à résoudre des problèmes de satisfaction des contraintes en CAO, mécanique et robotique. D. Michelucci et H. Lamure [Mic95] ont utilisé cette méthode pour résoudre des problèmes géométriques en 2D. C. Durand [Dur98] a adapté cette méthode pour résoudre des problèmes géométriques dans des espaces euclidiens 3D.

La fonction homotopique $H : \mathbb{X} \times [0, 1] \rightarrow \mathbb{S}$ est définie par

$$H(X, t) = (1 - t)G(X) + tF(X)$$

d'où $F(X)$ est un système d'équations sur l'ensemble de variables X , $X \in \mathbb{X}$, produisant des valeurs dans \mathbb{S} ; $G : \mathbb{X} \rightarrow \mathbb{S}$ est un système d'équations possédant autant de solutions que F . La méthode homotopique consiste à faire varier itérativement la valeur de t de 0 à 1 pour faire évoluer la fonction homotopique de G vers F . Chaque solution X^* du système G est associée à une courbe $H(t, X^*) : [0, 1] \rightarrow \mathbb{R}$ ou \mathbb{C} que l'on appelle *la courbe homotopique*. L'idée principale de la résolution par homotopie est de suivre la courbe homotopique démarrant en $t = 0$ afin d'obtenir les solutions de F lorsque $t = 1$. Pour ce faire, la méthode la plus utilisée est la méthode de prédiction-corrrection. En CAO, les valeurs de l'approximation initiale données par le concepteur seront les points de départ de l'homotopie.

Même si l'homotopie permet de trouver toutes les solutions dans le cas algébrique iso-contraint [Dur98, Ari97], le temps de calcul est du même ordre que celui de Newton-Raphson ce qui s'explique par le calcul de l'inversion de la matrice jacobienne [Moi08]. De plus, le choix du système G est une étape cruciale de la méthode d'homotopie. G doit avoir au moins autant de solutions que F et toutes ces solutions doivent être connues pour servir de points de départ aux chemins homotopiques.



1.5.2.3 Résolution par intervalles

La méthode de résolution par intervalles est une technique mixte de l'analyse par intervalles et de la programmation par contraintes [Dav87]. C. Jermann [Jer02] a utilisé cette méthode dans sa thèse pour la résolution de GCSP en 2D et 3D. Il décompose d'abord le GCSP objectif en un ensemble de sous-systèmes plus faciles à résoudre. Chaque sous-système produit en phase décomposition est ensuite résolu par intervalles et les solutions de ces sous-systèmes sont combinées en solutions du GCSP entier.

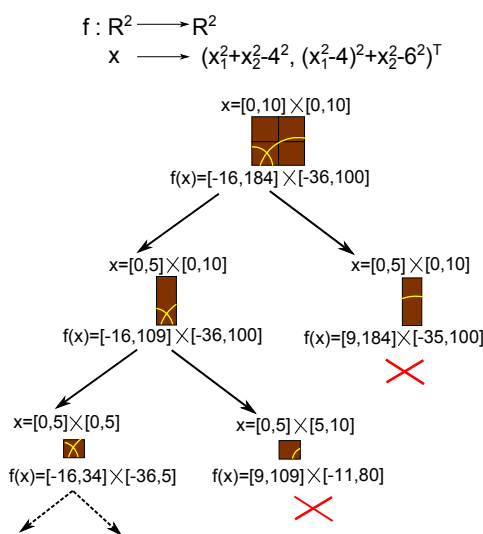


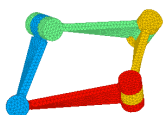
FIGURE 1.12 – Exemple : Schéma combinatoire de résolution par intervalles en bisection et évaluation

La résolution par intervalles consiste à isoler les intervalles-solutions contenus dans les domaines des variables du CSP. Ainsi, la solution fournie ne sera pas seulement une valeur numérique, mais un intervalle encadrant des solutions du GCSP. Par exemple, le CSP présenté à la figure 1.12 possède une solution dans $x = [0, 10] \times [0, 10]$. La bisection et l'évaluation nous permet de trouver la solution.

1.6 Conclusion

Dans la phase d'identification de paramètres, nous envisageons la même solution qu'avec la discipline GCSP. Ainsi, la méthode de squelette et la notation de Denavit-Hartenberg sont retenues. Les paramètres trouvés sur un squelette sont divisés en deux catégories : paramètres d'usinage et paramètres de mobilité.

Dans la phase de mise en équation, les approches cartésiennes et non-cartésiennes sont présentées. Avec l'approche cartésienne et contravariante, les comportements d'un mécanisme sont mélangés avec sa position et son orientation. Dans notre contexte, nous ne nous intéressons qu'aux dimensions tel que la longueur des pièces, l'angle entre les pièces, etc. Pour cette raison, l'approche cartésienne et contravariante est mise de coté.



En effet, la meilleure méthode de mise en équation n'existe pas car les performances de chaque méthode varient selon le problème à résoudre. Par exemple, si l'ingénieur s'intéresse à la hauteur dans l'exemple à 5 barres dans la figure 2, il vaut mieux utiliser l'approche cartésienne. Parce que les coordonnées (s, t) du point C contiennent la solution du problème : t est la hauteur du mécanisme.

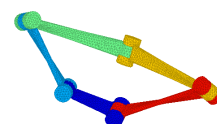
Donc, le système d'équations doit représenter à la fois le comportement du mécanisme étudié et, d'autre part, il faut qu'il soit le plus proche de la forme de solutions pour que l'ingénieur le résolve efficacement. Dans le meilleur des cas, la mise en équation est elle-même la solution. Ainsi, la phase de résolution peut être omise. Par exemple, avec la méthode Cayley-Menger, nous obtenons directement les relations de distances en carré ; la méthode des Quaternion est plus adaptée pour obtenir des relations entre les angles.

Au niveau de la résolution, nous avons choisi la méthode formelle des bases de Gröbner. Nous fournissons ici la justification de ce choix :

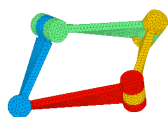
- **Méthodes numériques** : Bien que les méthodes numériques soient efficaces et résolvent des systèmes d'équations, nous cherchons des relations d'assemblage et de mobilité, ce que ces solveurs ne font pas. Nous nous sommes donc tournés vers les méthodes formelles.
- **Méthodes formelles** : Les méthodes formelles sont complètes et correctes mais présentent une complexité exponentielle. Pour cette raison, le temps du calcul est impacté par la taille du modèle étudié : le nombre de variables, le nombre d'équations et le degré de chaque variable. En revanche, nous avons choisi d'utiliser les bases de Gröbner parce qu'elles généralisent à la fois le PGCD et la réduction de Gauss, qui répondent mieux à nos besoins. De plus, l'algorithme pour la résolution formelle des bases de Gröbner a déjà été intégré dans un logiciel de calcul : Maple[®]. Ceci nous permet d'assurer le processus de résolution, étant aussi une des raisons pour lesquelles nous avons choisi cette méthode.

Bibliographie du chapitre 1

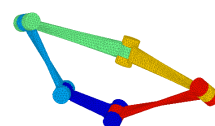
- [Alf] Liliane Alfonsi. Bézout et les intersection de courbes algébriques. Url : <http://www.bibnum.education.fr/files/Bezout-analyse.pdf>.
- [Alf08] Liliane Alfonsi. Étienne Bézout : Analyse algébrique au siècle des lumières. *Revue d'Histoire des Mathématiques*, 14 :211–287, 2008.
- [Ang82] J. Angeles. *Spatial Kinematic Chains*. Springer-Verlag, Berlin, 1982.
- [Ari97] C. M. Hoffmann & A.J. Arinyo. Symbolic constraints in constructive geometric constraint solving. *Journal of Symbolic Computation*, 23 :287–299, 1997.
- [Bas77] J. Bass. *Cours de mathématiques. Tome 1, fascicule 1 : algèbre linéaire, intégrales simple, fonction définies par des séries et des intégrales*. Masson, 1977.
- [Ben03] G.T. Bennett. A new mechanism. *Engineering*, 76 :777–778, 1903.
- [Bon00] D. Bondyfalat. *Interaction entre symbolique et numérique : application à la vision artificielle*. PhD thesis, Université de Nice Sophia Antipolis, 2000.



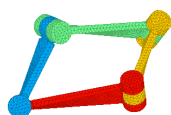
- [Bri26] R. Bricard. *Leçons de cinématique*, chapter Leçons de cinématique : Cinématique appliquée. Gauthier-Villars, 1926.
- [Cai10] P. É. Cailliau. *Modélisation et résolution de problèmes de synthèse dimensionnelle de mécanismes*. PhD thesis, École Centrale de Paris, 2010.
- [Car46] E. Cartan. *Leçons sur la géométrie des espaces de Riemann*. Jacques Gabay, 1946.
- [Cau31] A. L. Cauchy. Sur les polygones et polyèdres, second mémoire. *J. Ecole Polytechnique*, 9 :87, 1831.
- [Che11] C. Chen. The order of local mobility of mechanisms. *Mechanism and Machine Theory*, 46 :1251–1264, 2011.
- [Cho88] S.-C. Chou. *Mechanical Geometry Theorem Proving*. D. Reidel Publishing Co., 1988.
- [Cho98] X-S. Gao & S-C. Chou. Solving geometric constraint systems. ii. a symbolic approach and decision if rc-constructibility. *Computer-Aided Design*, 30, no. 2 :115–122, 1998.
- [CIP] CIP. Competitiveness and Innovation framework Programme. Url : http://ec.europa.eu/information_society/activities/ict_psp/index_en.htm.
- [Dah09] L. Racila & M. Dahan. Méthode géométrique de calcul pour un translateur 6R. In *19ème Congrès Français de Mécanique*, 2009.
- [Dav87] E. Davis. Constraint propagation with interval labels. *AI*, 32(3) :281–331, 1987.
- [Die95] P. Dietmaier. A new 6R space mechanism. in : *Proceeding of 9th World Congress IFToMM, Milano*, 1 :52–56, 1995.
- [Din] Mohab Safey El Din. Algorithmes efficaces en géométrie algébrique réelle. 2007, Url : <http://www-polsys.lip6.fr/safey/Articles/coursJNCF.pdf>.
- [Dur98] C. Durand. *Symbolic and numerical techniques for constraint solving*. PhD thesis, Department of Computer Science, Purdue University, December, 1998.
- [EIG] Spécification, contraintes et preuves. Url : http://icube-igg.unistra.fr/index.php/Sp%C3%A9cifications,_contraintes_et_preuves.
- [Fou04] D. Michelucci & S. Foufou. Using Cayley-Menger determinants for geometric constraint solving. *ACM Symposium on Solid Modeling and Application*, 2004.
- [Fou06] D. Michelucci & S. Foufou. Geometric constraint solving : the witness configuration method. *Computer-Aided Design*, 38 :284–299, 2006.
- [Fou07] D. Michelucci & S. Foufou. Detecting all dependences in systems of geometric constraints using the witness method. *Automated Deduction in Geometry*, pages 98–112, 2007.
- [Gir97] G. Colombari & J. Giraud. *Science industrielles, classes préparatoires 1re année*. 1997.
- [Gir09] G. Colombari & J. Giraud. *Science industrielles pour l'ingénieur, classes préparatoires 2ème année*. 2009.



- [Glu71] H. Gluck. The converse to the four-vertex theorem. *L'Enseignement Math.*, 17 :295–309, 1971.
- [Gog05] Grigore Gogu. Mobility of mechanisms : a critical review. *Mechanism and Machine Theory*, 40 :1068–1097, 2005.
- [Gol43] M. Goldberg. New five-bar and six-bar linkages in three dimensions. *ASME Trans*, 65 :183–210, 1943.
- [Grü17] Grübler. *Getriebelehre*. Springer, Berlin, 1917.
- [Gra02] J. Graver. Counting on frameworks : Mathematics to aid the design of rigid structures. *Dolciani Mathematical Expositions. Mathematical Association of America*, 25, 2002.
- [Hen92] B. Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21 :65–84, 1992.
- [Imb13] Rémi Imbach. *Résolution de contraintes géométriques en guidant une méthode homotopique par la géométrie*. PhD thesis, Université de Strasbourg, 2013.
- [Jer02] C. Jermann. *Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles*. PhD thesis, L'université de Nice-Sophia Antipolis UFR Sciences, 2002.
- [Kap86] D. Kapur. Using Gröbner bases to reason about geometry problems. *J. SymbolicComputation*, 2(4) :399–408, 1986.
- [Kha08] B. Siciliano & O. Khatib, editor. *Springer handbook of robotics*. Springer, 2008.
- [Lam70] G. Laman. Theorem and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4 :331–340, 1970.
- [Li09] Y. Li. New method to extend Macaulay resultant. *In Second International Conference on Intelligent Computation Technology and Automation, ICICTA '09*, 4 :562–565, 2009.
- [Lic83] A. Lichnerowicz. *Algèbre et analyse linéaires*. Masson, 1983.
- [Man11] G. Mandil. *Modèle de représentation géométrique intégrant les états physiques du produit*. PhD thesis, Supméca Paris France and Sherbrooke Québec Canada, 2011.
- [Mat97] P. Mathis. *Constructions géométriques sous contraintes en modélisation à base topologique*. PhD thesis, Université Louis Pasteur, Strasbourg, 1997.
- [Mel04] N. O. Melin. *Application of Bennett Mechanisms to Long-Span Shelters*. PhD thesis, Department of Engineering Science at the University of Oxford, 2004.
- [Mic95] H. Lamure & D. Michelucci. Solving geometric constraints by homotopy. In *Third Symposium on Solid Modelling and its Applications*, 1995.
- [Mid96] R. Latham & A. Middleditch. Connectivity analysis : a tool for processing geometric constraints. *Computer-Aided Design*, 28, no. 11 :917–928, 1996.
- [Moi08] M.T. Moinet. *Descriptions non cartésiennes et résolution de problèmes géométriques sous contraintes*. PhD thesis, École Centrale Paris, 2008.



- [Mya31] F. E. Myard. Contribution à la géométrie des systèmes articulés. *Société Mathématiques de France*, 59 :183–21, 1931.
- [Ort68] J. M. Ortega. The Newton-Kantorovich Theorem. *The American Mathematical Monthly*, 75(6) :658–660, Jun. - Jul. 1968.
- [Pro] ThinkMOTION Project. ThinkMOTION Project - Competitiveness and Innovation framework Programme. Url : <http://www.thinkmotion.eu/>.
- [Riv06] P. Serré & A. Ortuzar & A. Rivière. Non-cartesian modelling for analysis of the consistency of a geometric specification for conceptual design. *International Journal of Computational Geometry and Applications*, 16(5/6) :549–565, 2006.
- [Rot94] C. Mavroidis & B. Roth. Analysis and synthesis of overconstrained mechanisms. *in : Proceedings of the 1994 ASME Design Technical Conferences, Minneapolis, MI, September*, pages 115–133, 1994.
- [Sala] A. Bostan & B. Salvy. Identités de fonction spéciales et séries D-finies. Url : <http://perso.ens-lyon.fr/bruno.salvy/M1ENS/cours8.pdf>.
- [Salb] A. Bostan & B. Salvy. Somme hypergéométrique. Url : <http://perso.ens-lyon.fr/bruno.salvy/M1ENS/somme.pdf>.
- [Salc] A. Bostan & F. Chyzak & B. Salvy. D-finitude : algorithmes et applications. Url : <http://algo.inria.fr/EJCIM07/EJCIM07-LN.pdf>.
- [Sax95] D. Kapur & T. Saxena. Comparison of various multivariate resultant formulations. *In ISSAC'95 : Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, pages 187–194, 1995.
- [Sch98] J.F. Dufourd & P. Mathis & P. Schreck. Geometric construction by assembling solved subfigures. *Artificial Intelligence*, 99(1) :73–119, 1998.
- [Ta] F. Peyraut & M. Domaszewski & S. Gomes & D. Chamoret & N. Labeled & S. Roth & N. Harb & A.T. Ta. Modélisation éléments finis en hyperélasticité. Technical report, Laboratoire Mécatronique, Méthodes, Modèles et Métiers.
- [Tem84] A. Clément & A. Rivière & M. Temmerman. *Théorie et pratique*, chapter Cotation tridimensionnelle des systèmes mécanique. 1984.
- [The01] I. Macdonald & J. Pach & T. Theobald. Common tangents to four unit balls. *Discr. Comp. Geometry*, 26 :1–17, 2001.
- [Whi98] W. Whiteley. *Rigidity and scene analysis, Handbook for discrete and computational geometry*. CRC Press LLC, 1998.
- [Win90] F. Winkler. Gröbner bases in geometry theorem proving and simplest degeneracy conditions. *Mathematica Pannonica*, 1(1) :15–32, 1990.
- [Woh91] K. Wohlhart. Merging two general Goldberg 5R linkages to obtain a new 6R space mechanism. *Mechanism and Machine Theory*, 26(2) :659–668, 1991.
- [Wu86] W. Wu. Basic principles of mechanical theorem proving in elementary geometries. *J. Automated Reasoning*, 2 :221–254, 1986.

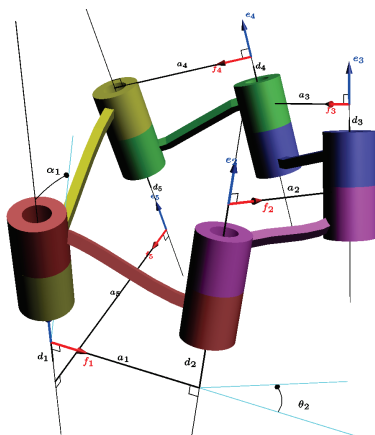


Chapitre 2

Approche numérique locale pour la mobilité

Sommaire

2.1	Introduction	61
2.1.1	Assembly and mobility check during Computer-Aided Design	61
2.1.2	Mobility vs. number of positions	62
2.1.3	A case study	63
2.2	Modeling approach, parameters and specifications	65
2.2.1	Modeling parameters	66
2.2.2	Specification parameters	69
2.2.3	Closed-loop equation	70
2.2.4	Equation system	70
2.3	Generation of compatibility equations	71
2.4	Compatibility equations of Goldberg 5R linkage	74
2.4.1	Equations of Goldberg 5R linkage	74
2.4.2	Calculation of mobility equations number	79
2.4.3	Mobility equations of specified Goldberg 5R linkage	81
2.5	Application of generated mobility equations	83
2.6	Conclusion and perspectives	84
	Bibliographie du chapitre 2	86



Résumé

Ce chapitre est constitué d'un article publié pendant la thèse. Cette publication présente une approche numérique qui permet de générer les conditions d'assemblage et de mobilité locales. A la fin de l'article, une application numérique est présentée. Elle montre un cas d'application des conditions générées.

Ce chapitre est constitué de l'article «A tool to check mobility under parameter variations in over-constrained mechanisms» publié en 2013 dans la revue internationale *Mechanism and Machine Theory*, vol.69.

Auteurs :

Ruixian LIU : Étudiante en doctorat au LISMMA et à l'École Centrale Paris

Philippe SERRÉ : Maître de conférences au LISMMA

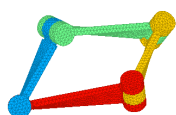
Jean-François RAMEAU : Conseiller scientifique chez Dassault Systèmes, professeur associé au LISMMA

Cet article présente une approche numérique qui détermine le nombre d'équations de mobilité pour une famille de mécanismes surcontraints. L'approche proposée permet également de générer les équations d'assemblage et de mobilité locale. La famille du mécanisme de Goldberg 5R est utilisée tout au long de l'article pour illustrer cette approche.

Le squelette associé au système de solides et la notation Denavit-Hartenberg sont utilisés pour identifier les paramètres du modèle. L'écriture des équations de fermeture sous forme $C(x, u, m) = 0$ est basée sur la matrice de Gram $\{q_1, \dots, q_{29}\} \subset X$ qui sont les paramètres de description initiaux; $\{s_1, \dots, s_{20}\} \subset U$ sont les paramètres d'usinage; $s_{21} \in M$ est un paramètre de mobilité. Parmi les s_i , s_1, \dots, s_5 sont les longueurs de barres, s_6, \dots, s_{20} sont des angles d'usinage. En raison de la perpendicularité entre la barre et ses deux axes de liaison, s_6, \dots, s_{15} restent constants. s_{16}, \dots, s_{20} sont les angles d'usinage entre les deux axes de liaisons. Dans un premier temps, nous linéarisons les équations de fermeture puis nous éliminons les dq_i . Pour obtenir les conditions d'assemblage, nous avons utilisé la décomposition en Valeurs Singulières (SVD) pour l'élimination des dq_i . Les conditions d'assemblage ne dépendent que des ds_i et garantissent l'assemblage du système étudié dans un voisinage de la configuration initiale.

La méthode d'obtention des conditions de mobilité locale est originale. Nous générons les conditions d'assemblage du mécanisme pour plusieurs positions. Pour chaque position, les dimensions du mécanisme restent constantes mais le paramètre de mobilité s_{21} varie. Puis toutes les équations sont stockées dans une matrice. Les conditions de mobilité sont calculées à partir de cette matrice en utilisant la SVD. En effet, la SVD permet de déterminer la dépendance des conditions d'assemblage des différentes positions. Finalement, les conditions de mobilité calculées par la SVD sont indépendantes.

Les équations d'assemblage et de mobilité générées dans cet article sont locales. Elles sont valides dans le voisinage des mécanismes nominaux. A noter que pour générer les conditions d'assemblage, le modèle initial doit être mobile. Une application des équations d'existence est exposée à la fin de l'article.



A tool to check mobility under parameter variations in over-constrained mechanisms

Ruixian LIU¹, Philippe SERRÉ¹, Jean-François RAMEAU²

¹ *Laboratoire d'ingénierie des Systèmes Mécanique et des MATériaux, 3 rue Fernand Hainaut, 93407 Saint-Ouen Cedex, France*

² *Dassault Systèmes, 10, rue Marcel Dassault, 78140 Vélizy-Villacoublay, France*

Abstract

This paper explores a numerical approach, which determines the number of mobility equations for a family of over-constrained mechanism and generates mobility equations under dimensional parameter variations. These equations can be used to check the mobility of the over-constrained mechanism. They are produced from a specified mobile system and are valid in the neighborhood of nominal mechanisms. This generic approach is applied on Goldberg 5R linkages.

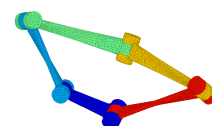
Keywords : Over-constrained mechanism, Mobility, 5R linkage, Specification

2.1 Introduction

2.1.1 Assembly and mobility check during Computer-Aided Design

Mechanical products are often composed of links connected by joints, forming open or closed mobile chains. The main issue that is addressed when all the parts of the mechanism are accomplished, is to determine both how it can be assembled and how inter-active it will be. The more parts there are in a mechanism, the more complex the search of solution for mobility is. Despite the discovery of many over-constrained mechanisms and their kinematic studies, only a few of them have been used in industrial application. Most of engineers in industry are unaware of the existence of spatial over-constrained mechanisms and their properties. Moreover, there are few CAD software managing intelligently over-constrained mechanisms. When several designers are involved in the design process, maintaining the consistency of over-constrained models is even harder. So, our research team works together with Dassault Systèmes[®] in order to develop a tool for dealing with over-constrained-mechanism-based modeling during teamwork. The development of the modeling tool is not only a problem of solving equations but also making all design parameters editable. So that the design team can ensure the consistency of design parameters. As we presented recent years the approaches based on Cayley-Menger determinants and topological matrix to guarantee the assembly of a rigid mechanism [Cho11, Moi10], this paper will show more about mobility.

This paper presents a generic approach, which determines the number of mobility equations for an over-constrained mechanism family and generates these equations from a specified configuration in view to guiding users to ensure the consistency of the sets of parameters used. When engineers want to modify a parameter of their mechanism, the others can be calculated



thanks to generated mobility equations. Therefore, the new mechanism is always mobile in a neighborhood of some points in the configuration space.

In next subsection, we begin by giving theorems which inspire us during the mobility calculus. Following this, an over-constrained mechanism and its initial configuration are displayed. Definitions of modeling parameters and specifications are given in section 2. In section 3, one can find the explanation on how to determine the number of mobility equations and how to generate them. The discussion on how many compatibility equations there should be for determining the global mobility of Goldberg 5R linkage family is in section 4. The mobility equations of a Goldberg 5R linkage are provided thanks to the numerical method presented in section 3. Moreover, a numerical example illustrating how the above equations preserve the mobility of over-constrained mechanism is presented in section 5.

2.1.2 Mobility vs. number of positions

Definitions of global mobility and local mobility are given by Chen [Che11]. He pointed out that the rank-deficiency of Jacobian matrix is equivalent to the first-order local mobility, and the global mobility is equivalent to the infinite-order local mobility. Other major approaches, like motion group theory [Her78] and theory of linear transformation [Gog08, Gog09], concentrate on the global mobility of mechanisms.

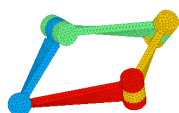
The numerical investigation is based on the claim that a number of first order mobility in different positions are sufficient to conclude the overall mobility of the mechanism. Despite this claim is acceptable from the engineering point of view, it deserves a rigorous proof from the mathematical point of view. The mathematics are based on the fact that equations of the mechanism are a polynomial system, that the configuration parameters are the variables of the polynomials, and that any point in the zero set of the polynomial system is a particular position of the mechanism. It is not difficult to change trigonometric functions into polynomial functions by symbolic substitution and by adding new equations. The first idea that comes to mind is that the zero set of a polynomial system is an algebraic variety [O'S00], which features the following property.

Theorem 2.1.2.1. *A zero dimensional algebraic variety is a finite set of points.*

This means that if the zero set of the polynomial system includes an infinite subset of separated points, then it is a non zero dimensional variety, meaning that the corresponding mechanism is mobile. Nevertheless, using theorem 2.1.2.1 for mechanism mobility is questionable for two reasons. The first one is that the theorem holds for complex numbers as opposed to real life mechanisms and real solutions. The non zero dimensional variety could be a complex variety, providing some imaginary mobility to a real mechanism. The second reason is that theorem 2.1.2.1 does not give any numerical bound on the number of separated solutions beyond which the mobility is guaranteed.

Fortunately, these two objections can be overcome thanks to a real algebraic geometry theorem [Mil64] linking the topology of the real zero set of a real polynomial system to a formula involving its degree and its number of variables.

Theorem 2.1.2.2. *Let $F(x_1, \dots, x_m) = (f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m))$ be a real polynomial function of m real variables and n scalar coordinates. Let $d = \max\{\deg(f_i), i = 1, \dots, n\}$*



be the degree of function F . Let

$$V = \{(x_1, \dots, x_m) \in \mathbb{R}^m; F(x_1, \dots, x_m) = 0\}$$

be the zero set of F . Then, the sum of the Betti numbers of V is less than $d(2d - 1)^{m-1}$.

Betti numbers β_j are well known in algebraic topology [Wal07] and, in the context of mechanism mobility, it is enough to remember that they are non-negative integers and that the first Betti number β_0 is the number of connected components of V . Let $s \geq 0$ be the number of isolated zeros of function F . Each one of these s points in \mathbb{R}^m is one particular connected component of V . Consequently, from theorem 2.1.2.2,

$$s \leq \beta_0 \leq \sum_j \beta_j \leq d(2d - 1)^{m-1}$$

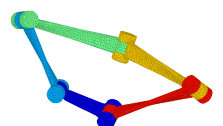
meaning that the number of isolated zeros of function F cannot be larger than $d(2d - 1)^{m-1}$. Suppose now that V includes a finite subset of r points and that $r > d(2d - 1)^{m-1}$. Then V is a non zero dimensional set. This is because V includes at most $d(2d - 1)^{m-1}$ isolated solutions so that $r - d(2d - 1)^{m-1}$ points belong to non zero dimensional connected components of V . Keeping in mind that V is the set of all possible positions of the mechanism, reformulating the previous discussion in terms of mobility yields the expected theorem.

Theorem 2.1.2.3. *A mechanism defined by the polynomial function of theorem 2.1.2.2 and featuring a number of positions that is strictly larger than $d(2d - 1)^{m-1}$ is mobile.*

2.1.3 A case study

Since Bennett published his acclaimed paper “A new mechanism” [Ben03] which gave birth to the Bennett linkage, it has been found that a number of 3D over-constrained linkages can be regarded as the combination of several Bennett linkages, including the Myard linkage [Mya31], the Goldberg 5R and 6R linkages [Gol43], the Bennett-joint 6R linkage [Rot94], the Dietmaier 6R linkage [Die95], Wohlhart’s double-Goldberg linkages [Woh91], etc. A lot of over-constrained mechanisms are spatial linkages. Due to the spatial kinematic characteristics, they are advantageous candidates in modern linkage design where spatial motion is needed. It is well known that a linkage should have at least seven links to be mobile in normal closed-loop revolute joint spatial mechanisms [Rot94]. Few links and joints in a mechanism means reduction in fabrication cost. Nowadays, more and more research based on over-constrained family started for military purposes and humanitarian aid [Che03]. Chen [Che03] and Melin [Mel04] developed a new type of deployable, long-span assembled shelter frame based upon tiled Bennett mechanisms. Other applications of over-constrained mechanisms can be found in [Cua06, Bed07]. According to this trend, over-constrained mechanisms being mobile with fewer links and fewer joints are interesting topics to be studied.

Denavit and Hartenberg set forth a standard approach to the analysis of linkages. Their approach is a commonly used convention for attaching reference frames to the links of a spatial kinematic chain. In this paper, the research is based on four parameters in Denavit-Hartenberg representation, shown in Fig. 2.1. Each joint is assigned a coordinate frame. Four D-H parameters describe how a frame i relates to a previous frame $i-1$. The four parameters a_i , α_i , d_i and θ_i are generally given the names link length (length of the perpendicular between the joint axis), link twist (twist angle between the joint axis), joint offset (perpendicular and shortest



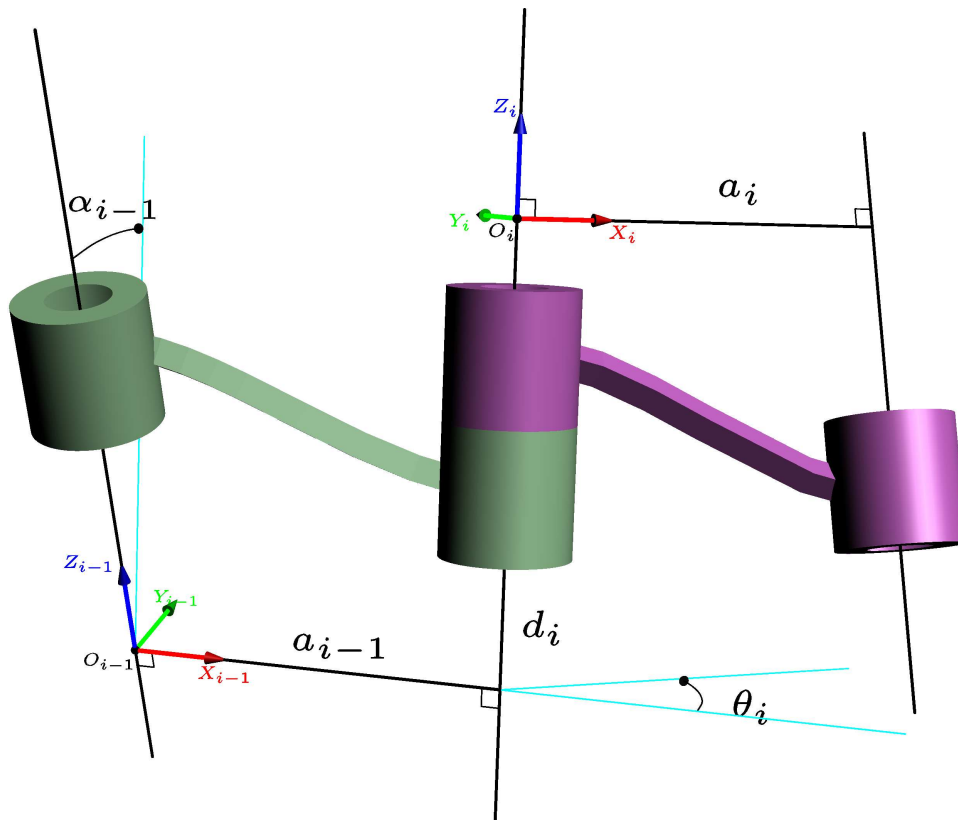
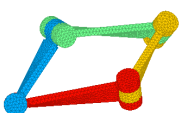


FIGURE 2.1 – Denavit-Hartenberg frame assignment

line between two links), and joint angle (rotation angle around a joint axis), respectively. Twist angle is measured from one joint axis to the other joint axis in the same link. Positive twist angle, α_1 in Fig. 2.2 for example, results from counterclockwise rotation, and negative twist angle results from clockwise rotation. Thus, for a general spatial five-bar linkage, D-H notation is derived in Fig. 2.2, where e_i is the unit vector along i th revolute axis, while f_i if the unit vector along the common perpendicular between e_i and e_{i+1} .

Among the best known of over-constrained kinematic loops, Goldberg's $5R$ linkage and Myard linkage are the mobile ones that are connected with five revolute joints. Nowadays, the family of $5R$ linkages is growing : $5R$ linkages include Myard $5R$ [Mya31], Goldberg $5R$ [Gol43], Generalized Goldberg $5R$ [Woh91] and extended Myard $5R$ [You08]. So in this paper, Goldberg $5R$ linkage will be studied. Goldberg has shown that the Bennett linkage can be used as a building-block for other multi-link mechanisms [Gol43]. As can be seen in Fig. 2.3, the Goldberg $5R$ linkage is composed of five bars connected by five revolute joints. This mechanism is obtained by combining a pair of Bennett linkages with the same ratio, defined as the ratio between the sinusoidal value of twist and length of the same link. Respecting certain equations, the system has a common link and, there are two other links in each Bennett made co-linear and rigidly connected. Finally, the Goldberg $5R$ linkage is completed after the removal of the common link [You07].

Initial dimensions of the studied mechanism are listed in Table 2.1. In order to easily compare the initial and final configuration, it is better to use integer for initial length during modeling.



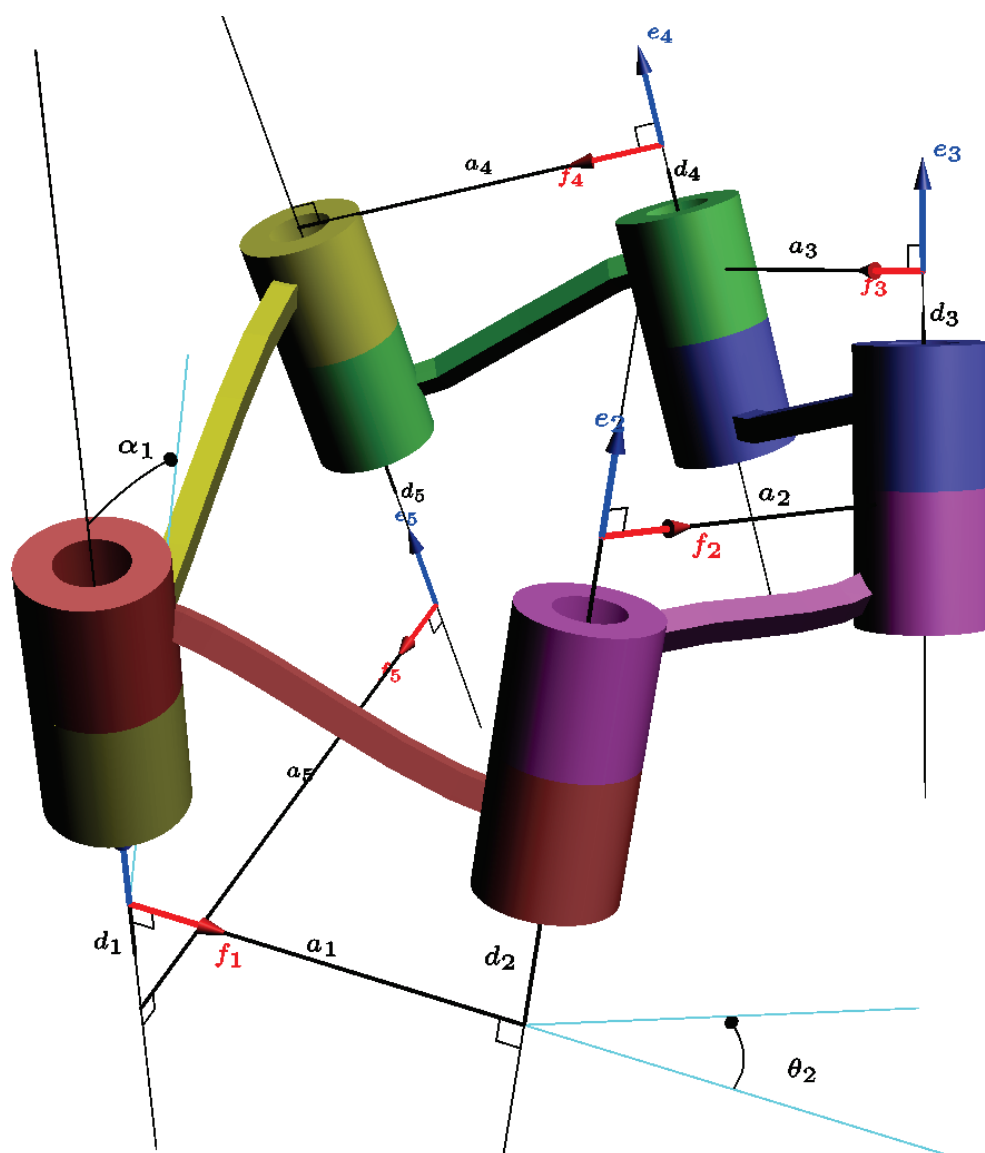
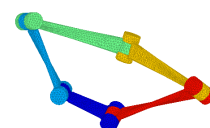


FIGURE 2.2 – A general spatial 5-bar linkage. In this case, the displayed mechanism is rigid.

And the CAD software can ensure at least six decimal places on millimeter. Thus, there are six zeros in initial length in Table 2.1, Table 2.3 and Table 2.4. All the dimensions in this paper use six decimal places.

2.2 Modeling approach, parameters and specifications

The construction of equation system is based on vector and tensor modeling. Let us define \mathbf{q} a set of parameters called modeling parameters, which forms a complete, consistent and minimal system. The set of parameters \mathbf{s} imposed by designers to specify a link and its joints is named “specification parameters”. Among \mathbf{s} , magnitudes associated to links are “design parameters”; magnitudes associated to joints are “configuration parameters”. Design parameters are invariant



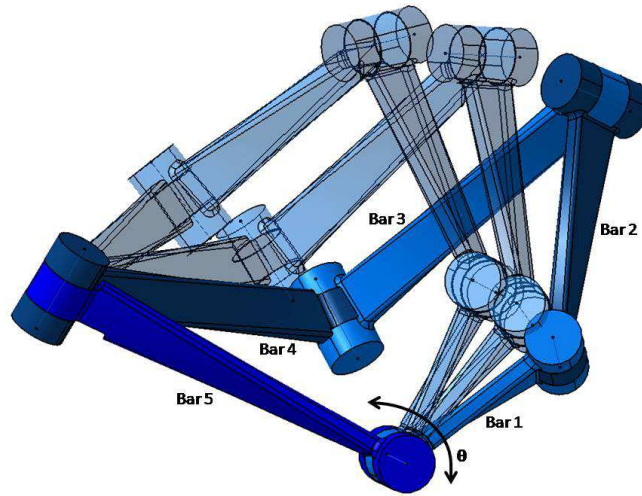


FIGURE 2.3 – A Goldberg 5R linkage

Bar's number	Length(mm) - a_i	Angle(degree) - α_i
1	100.000000	28.359350
2	140.000000	41.682325
3	200.000000	-71.805128
4	240.000000	70.041675
5	200.000000	-71.805128

TABLE 2.1 – Initial configuration of Goldberg 5R

during the movement of mechanism. Specification parameters might be physical parameters such as mass and density. In this study, we are only interested in geometrical specification parameters such as length and angle. When there are several links and joints in the same system, defined by different designers, no one knows if the mechanism can ultimately be constructed or not : there are sometimes too few or too many specification parameters, anyway, the over-constrained mechanism can be assembled if, and only if, its specification parameters are consistent.

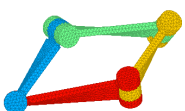
In this paper, equation system consists of three types of lists : $T(\mathbf{q})$, $K(\mathbf{s})$ and $B(\mathbf{q})$. $T(\mathbf{q})$ is a list in which there are angular, norm and length modeling parameters. And $K(\mathbf{s})$ has angular, norm and length specifications. $B(\mathbf{q})$ is the projection of closed loop on three independent vectors. These lists are as follow :

2.2.1 Modeling parameters

2.2.1.1 Angular modeling

Definition 2.2.1.1. Given a n -dimension euclidean space \mathbb{V} and one of its basis $\{\mathbf{e}_i\}$, G is called metric tensor. The 2-order tensor is defined by

$$G : \{\mathbf{u}, \mathbf{v}\} \in \mathbb{V}^2 \rightarrow G(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{R}$$



where $\langle \mathbf{u}, \mathbf{v} \rangle$ is the scalar product of two vectors. Since the tensor is 2-order, there are four kinds of components :

$$\begin{cases} g_{ij} = \mathbf{G}(\mathbf{e}_i, \mathbf{e}_j) = \langle \mathbf{e}_i, \mathbf{e}_j \rangle \\ g^{ij} = \mathbf{G}(\mathbf{e}^i, \mathbf{e}^j) = \langle \mathbf{e}^i, \mathbf{e}^j \rangle \\ g_i^j = \mathbf{G}(\mathbf{e}_i, \mathbf{e}^j) = \langle \mathbf{e}_i, \mathbf{e}^j \rangle = \delta_i^j \\ g_j^i = \mathbf{G}(\mathbf{e}^i, \mathbf{e}_j) = \langle \mathbf{e}^i, \mathbf{e}_j \rangle = \delta_j^i \end{cases} \quad (2.1)$$

where basis $\{\mathbf{e}^j\}$, called contravariant basis, is the dual basis of covariant basis $\{\mathbf{e}_i\}$. The symbol of Kronecker δ_j^i is defined by

$$\delta_j^i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

The components of two matrix $[g_{ij}]$ and $[g^{ij}]$ depend on basis $\{\mathbf{e}_i\}$ and $\{\mathbf{e}^j\}$. These two matrix are symmetric : $g_{ij} = g_{ji}$ and $g^{ij} = g^{ji}$ because the scalar product of two vectors is commutative.

In 3D metric space, there are two ways to express vector \mathbf{u} : with covariant basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ or with contravariant basis $(\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3)$:

$$\mathbf{u} = u^1 \mathbf{e}_1 + u^2 \mathbf{e}_2 + u^3 \mathbf{e}_3 = u_1 \mathbf{e}^1 + u_2 \mathbf{e}^2 + u_3 \mathbf{e}^3$$

Where u^1, u^2, u^3 are contravariant components in covariant basis and u_1, u_2, u_3 are the covariant components in contravariant basis.

Let us admit another vector \mathbf{v} in 3D metric space, we have

$$\mathbf{v} = v^1 \mathbf{e}_1 + v^2 \mathbf{e}_2 + v^3 \mathbf{e}_3 = v_1 \mathbf{e}^1 + v_2 \mathbf{e}^2 + v_3 \mathbf{e}^3$$

By definition, the different ways to write the scalar product of two vectors \mathbf{u} and \mathbf{v} in function of their components are :

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{G}(\mathbf{u}, \mathbf{v}) = g_{ij} u^i v^j = g^{ij} u_i v_j \quad (2.2)$$

One of tensors properties is that the contracted product between \mathbf{G} and a vector \mathbf{u} is

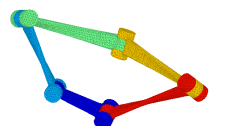
$$\begin{aligned} \mathbf{G} \cdot \mathbf{u} &= g^{ij} \mathbf{e}_i \otimes \mathbf{e}_j \cdot u_k \mathbf{e}^k \\ &\stackrel{\text{def}}{=} g^{ij} u_k \mathbf{e}_i \otimes (\mathbf{e}_j \cdot \mathbf{e}^k) \\ &= g^{ij} u_k \delta_j^k \mathbf{e}_i \end{aligned} \quad (2.3)$$

$$\begin{aligned} \mathbf{G} \cdot \mathbf{u} &= g_{ij} \mathbf{e}^i \otimes \mathbf{e}^j \cdot u^k \mathbf{e}_k \\ &\stackrel{\text{def}}{=} g_{ij} u^k \mathbf{e}^i \otimes (\mathbf{e}^j \cdot \mathbf{e}_k) \\ &= g_{ij} u^k \delta_k^j \mathbf{e}^i \end{aligned} \quad (2.4)$$

Thanks to Einstein summation convention (Einstein notation), we have :

$$\begin{aligned} g^{ij} u_k \delta_j^k \mathbf{e}_i &= g^{ij} u_j \mathbf{e}_i \\ &= \mathbf{u} \end{aligned} \quad (2.5)$$

$$\begin{aligned} g_{ij} u^k \delta_k^j \mathbf{e}^i &= g_{ij} u^j \mathbf{e}^i \\ &= \mathbf{u} \end{aligned} \quad (2.6)$$



Eqs. (2.5) and (2.6) allow us to obtain

$$\mathbf{u} = g^{ij} u_j \mathbf{e}_i \iff u^i = g^{ij} u_j \quad \text{and} \quad \mathbf{u} = g_{ij} u^j \mathbf{e}^i \iff u_i = g_{ij} u^j \quad (2.7)$$

In matrix calculus, Eq. (2.7) can be written as follow :

$$[u^i] = [g^{ij}][u_j] \quad \text{and} \quad [u_i] = [g_{ij}][u^j] \quad (2.8)$$

So, Eq. (2.9) can be concluded from Eq. (2.8)

$$[g^{ij}] = [g_{ij}]^{-1} \quad (2.9)$$

Since we choose the covariant modeling, Eq. (2.9) can be substituted into expression $g^{ij} u_i v_j$. Eq. (2.2) becomes

$$\langle \mathbf{u}, \mathbf{v} \rangle = [u_i][g_{ij}]^{-1}[v_j] = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix} \begin{pmatrix} 1 & g_{12} & g_{13} \\ g_{12} & 1 & g_{23} \\ g_{13} & g_{23} & 1 \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (2.10)$$

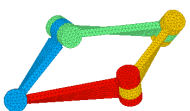
Afterwards, a set of modeling parameters \mathbf{q} is associated to $[u_i]$, $[g_{ij}]$ and $[v_j]$. Hence, Eq. (2.10) becomes

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle &= \begin{pmatrix} q_{u1} & q_{u2} & q_{u3} \end{pmatrix} \begin{pmatrix} 1 & q_1 & q_2 \\ q_1 & 1 & q_3 \\ q_2 & q_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} q_{v1} \\ q_{v2} \\ q_{v3} \end{pmatrix} \\ &= \frac{1}{\Delta} (q_{v1} q_{u1} q_3^2 - q_{v1} q_{u1} + q_{v1} q_{u2} q_1 - q_{v1} q_{u2} q_2 q_3 - q_{v1} q_{u3} q_1 q_3 + \\ &\quad q_{v1} q_{u3} q_2 + q_{v2} q_{u1} q_1 - q_{v2} q_{u1} q_2 q_3 + q_{v2} q_{u2} q_2^2 - q_{v1} q_{u2} + \\ &\quad q_{v2} q_{u3} q_3 - q_{v2} q_{u3} q_2 q_1 - q_{v3} q_{u1} q_1 q_3 + q_{v3} q_{u1} q_2 + q_{v3} q_{u2} q_3 - \\ &\quad q_{v3} q_{u2} q_2 q_1 - q_{v3} q_{u3} + q_{v3} q_{u3} q_1^2) \end{aligned} \quad (2.11)$$

where $\Delta = -2q_2 q_1 q_3 + q_2^2 + q_1^2 - 1 + q_3^2$. Therefore, all angular modelings such as $\langle \mathbf{u}, \mathbf{v} \rangle$ are in function of \mathbf{q} and they are stored in list $T(\mathbf{q})$. Moreover, a mechanism with a single closed loop often has more than three vectors. For instance, a 5R Goldberg linkage has ten vectors according to Denavit-Hartenberg notation. So, engineers can calculate the angle of any two vectors thanks for Eq. (2.11).

2.2.1.2 Norm modeling

Norm modeling uses the scalar product of the same vector. This expression is given by :



$$\begin{aligned}
 \langle \mathbf{u}, \mathbf{u} \rangle = \|\mathbf{u}\|^2 &= \begin{pmatrix} q_{u1} & q_{u2} & q_{u3} \end{pmatrix} \begin{pmatrix} 1 & q_1 & q_2 \\ q_1 & 1 & q_3 \\ q_2 & q_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} q_{u1} \\ q_{u2} \\ q_{u3} \end{pmatrix} \\
 &= \frac{1}{\Delta} (q_{u1}^2 q_3^2 - q_{u1}^2 + 2q_{u1} q_{u2} q_1 - 2q_{u1} q_{u2} q_2 q_3 - 2q_{u1} q_{u3} q_1 q_3 + \\
 &\quad 2q_{u1} q_{u3} q_2 - 2q_{u2} q_{u1} q_2 q_3 + q_{u2}^2 q_2^2 - q_{u1} q_{u2} + 2q_{u3} q_{u2} q_2 q_1 - \\
 &\quad q_{u3}^2 + q_{u3}^2 q_1^2)
 \end{aligned} \tag{2.12}$$

where $\Delta = -2q_2 q_1 q_3 + q_2^2 + q_1^2 - 1 + q_3^2$. And it is the same modeling way for the norm of other vectors if the studied model has more than four vectors. This relation is one part of list $T(\mathbf{q})$.

2.2.1.3 Length modeling

Suppose a couple of points AB represented by vector \overrightarrow{AB} :

$$\overrightarrow{AB} = q_{AB} \mathbf{u}$$

where \mathbf{u} is a normalized vector. Therefore, the length of \overrightarrow{AB} is

$$\|\overrightarrow{AB}\| = \|q_{AB} \mathbf{u}\| = q_{AB} \|\mathbf{u}\| = q_{AB} \tag{2.13}$$

And q_{AB} is also saved in list $T(\mathbf{q})$. To sum up, elements in list $T(\mathbf{q})$ are angular modeling such as Eq. (2.11), norm modeling such as Eq. (2.12) and length modeling such as q_{AB} in Eq. (2.13).

2.2.2 Specification parameters

2.2.2.1 Angular specification

In vector space, if s_{uv} is an angular specification, the scalar product of \mathbf{u} and \mathbf{v} is

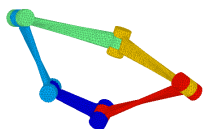
$$\langle \mathbf{u}, \mathbf{v} \rangle = \cos(\mathbf{u}, \mathbf{v}) = \cos(s_{uv}) \tag{2.14}$$

This kind of specification expression is stored in list $K(\mathbf{s})$. Attention should be paid that angular specification can represent design parameters (twist angles) as well as configuration parameters (joint angles).

2.2.2.2 Norm specification

Since basis $\{e_i\}$ is composed of unit vectors, the other vectors in the studied model are also normalized. Thus, the specification of vectors norm is 1 :

$$s_{norm} = \|\mathbf{u}\|^2 = 1 \tag{2.15}$$



Norm specification is also one part of list $K(\mathbf{s})$.

2.2.2.3 Specification of length

For a non-unit vector \overrightarrow{AB} , its length can be specified by

$$\|\overrightarrow{AB}\| = L_{AB} = s_{AB} \quad (2.16)$$

This kind of specification can represent design parameters (length of links) as well as configuration parameters (joint offset).

This is the last part in list $K(\mathbf{s})$. To sum up, list $K(\mathbf{s})$ includes angular specification such as Eq. (2.14), norm specification such as Eq. (2.15) and length specification such as Eq. (2.16).

2.2.3 Closed-loop equation

In vector space, the vector sum of a closed loop is vector zero. For a closed loop of four vectors for example, we have

$$\overrightarrow{AB} + \overrightarrow{BC} + \overrightarrow{CD} + \overrightarrow{DA} = q_{AB}\mathbf{u} + q_{BC}\mathbf{v} + q_{CD}\mathbf{w} + q_{DA}\mathbf{t} = \vec{0} \quad (2.17)$$

The projections of Eq. (2.17) on basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ are scalar zeros :

$$\begin{cases} q_{AB}\langle\mathbf{u}, \mathbf{e}_1\rangle + q_{BC}\langle\mathbf{v}, \mathbf{e}_1\rangle + q_{CD}\langle\mathbf{w}, \mathbf{e}_1\rangle + q_{DA}\langle\mathbf{t}, \mathbf{e}_1\rangle = 0 \\ q_{AB}\langle\mathbf{u}, \mathbf{e}_2\rangle + q_{BC}\langle\mathbf{v}, \mathbf{e}_2\rangle + q_{CD}\langle\mathbf{w}, \mathbf{e}_2\rangle + q_{DA}\langle\mathbf{t}, \mathbf{e}_2\rangle = 0 \\ q_{AB}\langle\mathbf{u}, \mathbf{e}_3\rangle + q_{BC}\langle\mathbf{v}, \mathbf{e}_3\rangle + q_{CD}\langle\mathbf{w}, \mathbf{e}_3\rangle + q_{DA}\langle\mathbf{t}, \mathbf{e}_3\rangle = 0 \end{cases} \quad (2.18)$$

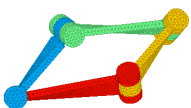
Note that $\mathbf{e}_1, \mathbf{e}_2$ and \mathbf{e}_3 should be independent. After being associated with the scalar product of two vectors, angular modeling parameters can be used in the development of Eq. (2.18).

$$\begin{cases} q_{AB}q_{u1} + q_{BC}q_{v1} + q_{CD}q_{w1} + q_{DA}q_{t1} = 0 \\ q_{AB}q_{u2} + q_{BC}q_{v2} + q_{CD}q_{w2} + q_{DA}q_{t2} = 0 \\ q_{AB}q_{u3} + q_{BC}q_{v3} + q_{CD}q_{w3} + q_{DA}q_{t3} = 0 \end{cases} \quad (2.19)$$

So, three projection equations of vectors' closed loop are considered as function $B(\mathbf{q})$ (Eq. (2.19)).

2.2.4 Equation system

Afterwards, writing the equivalence of the two sets of parameters, \mathbf{s} and \mathbf{q} , we obtain the equations of system. The establishment of Eq. (2.20) composed of m equations reveals the relation between modeling parameters and specification parameters [Clé09]. Theoretically, the system of equations is as follows :



$$\begin{cases} T(\mathbf{q}) - K(\mathbf{s}) = 0 \\ B(\mathbf{q}) = 0 \end{cases} \quad (2.20)$$

Where B are closed-loop functions of n modeling parameters $\mathbf{q} = (q_1, q_2, \dots, q_n)$ and $T(\mathbf{q}) - K(\mathbf{s})$ are modeling functions involving \mathbf{q} and l specification parameters $\mathbf{s} = (s_1, s_2, \dots, s_l)$ [Riv07]. The initial configuration in Table 1 will be substituted into \mathbf{q} and \mathbf{s} after some developments.

Eq. (2.20) can also be abbreviated into a set of equations

$$F(\mathbf{q}, \mathbf{s}) = 0 \quad (2.21)$$

2.3 Generation of compatibility equations

Eq. (2.21) is regarded as an implicit function. According to the implicit function theorem, if Eq. (2.21) satisfies some mild conditions on its partial derivatives, then one can, in principle, solve this equation for \mathbf{s} , at least over some small intervals. To state the implicit function theorem, the Jacobian of F is required. Given a fixed point $(\mathbf{q}_0, \mathbf{s}_0) = (q_{01}, q_{02}, \dots, q_{0n}, s_{01}, s_{02}, \dots, s_{0l})$ which satisfies $F(\mathbf{q}_0, \mathbf{s}_0) = 0$, the Jacobian matrix is

$$(DF)(\mathbf{q}_0, \mathbf{s}_0) = \left(\begin{array}{cccc|cccc} \frac{\partial f_1}{\partial q_1}(\mathbf{q}_0, \mathbf{s}_0) & \cdot & \cdot & \cdot & \frac{\partial f_1}{\partial q_n}(\mathbf{q}_0, \mathbf{s}_0) & \frac{\partial f_1}{\partial s_1}(\mathbf{q}_0, \mathbf{s}_0) & \cdot & \cdot & \cdot & \frac{\partial f_1}{\partial s_l}(\mathbf{q}_0, \mathbf{s}_0) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_m}{\partial q_1}(\mathbf{q}_0, \mathbf{s}_0) & \cdot & \cdot & \cdot & \frac{\partial f_m}{\partial q_n}(\mathbf{q}_0, \mathbf{s}_0) & \frac{\partial f_m}{\partial s_1}(\mathbf{q}_0, \mathbf{s}_0) & \cdot & \cdot & \cdot & \frac{\partial f_m}{\partial s_l}(\mathbf{q}_0, \mathbf{s}_0) \end{array} \right) \quad (2.22)$$

$$= (\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0) \mid \mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0))$$

where $\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0)$ is the first-order partial derivatives matrix of F in the \mathbf{q} 's at the point $(\mathbf{q}_0, \mathbf{s}_0)$. Similarly, $\mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0)$ is the first-order partial derivatives matrix of F in the \mathbf{s} 's at the point $(\mathbf{q}_0, \mathbf{s}_0)$.

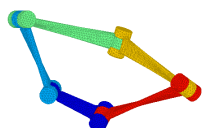
Except when it is a singular point, \mathbf{q}_0 is a set of modeling parameters that forms a complete, consistent and minimal system. Thus, $\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0)$ is invertible. In this case, there exists a function g such that $\mathbf{q} = g(\mathbf{s})$ in the neighborhood of point $(\mathbf{q}_0, \mathbf{s}_0)$. Then Eq. (2.21) becomes

$$F(g(\mathbf{s}), \mathbf{s}) = 0 \quad (2.23)$$

Since $\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0)$ is invertible, a solution of Eq. (2.23) exists according to the implicit function theorem. However, it is too difficult to find out $\mathbf{q} = g(\mathbf{s})$ and to solve Eq. (2.23). Hence, the aim of this study is not to solve Eq. (2.23) for \mathbf{s} , but to find out the relation between specification parameter variations : $d\mathbf{s} = \mathbf{s} - \mathbf{s}_0$.

As point $(\mathbf{q}_0, \mathbf{s}_0)$ satisfies Eq. (2.21), $(\mathbf{q}_0, \mathbf{s}_0)$ is also a configuration of studied mechanism. From this configuration, a displacement can be defined such as $(\mathbf{q}_0 + d\mathbf{q}, \mathbf{s}_0 + d\mathbf{s})$. Then, Eq. (2.21) should be written in a new set of variables, given by

$$F(\mathbf{q}_0 + d\mathbf{q}, \mathbf{s}_0 + d\mathbf{s}) = 0 \quad (2.24)$$



This formulation is particularly useful in the analysis of local mobility around a specific configuration. Expanding Eq. (2.24) via Taylor's theorem yields

$$\begin{aligned} F(\mathbf{q}_0 + d\mathbf{q}, \mathbf{s}_0 + d\mathbf{s}) &= F(\mathbf{q}_0, \mathbf{s}_0) + \frac{\partial F}{\partial \mathbf{q}}(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{q} + \frac{\partial F}{\partial \mathbf{s}}(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s} + o(\|\boldsymbol{\omega}\|) \\ &= 0 \end{aligned} \quad (2.25)$$

with $\boldsymbol{\omega} = (d\mathbf{q}, d\mathbf{s})$.

It is known that $F(\mathbf{q}_0, \mathbf{s}_0) = 0$. And $F(\mathbf{q}_0 + d\mathbf{q}, \mathbf{s}_0 + d\mathbf{s}) = 0$ is required. At the first-order, neglecting $o(\|\boldsymbol{\omega}\|)$, Eq. (2.25) becomes a linear system :

$$\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{q} + \mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s} = 0 \quad (2.26)$$

From above relation one can obtain the expression $\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{q} = -\mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s}$. In order to get the relation under specification parameter variations $d\mathbf{s}$, $d\mathbf{q}$ should be eliminated. For this purpose, a numerical method named Singular Value Decomposition (SVD) is used to develop $\mathbf{J}_q(\mathbf{q}_0, \mathbf{s}_0)$

$$\begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \cdot \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{pmatrix}^T \cdot d\mathbf{q} = -\mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s} \quad (2.27)$$

with $\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \in \mathbb{R}^{m \times m}$; $\mathbf{U}_1 \in \mathbb{R}^{m \times r}$, $\mathbf{U}_2 \in \mathbb{R}^{m \times (m-r)}$

and $\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{pmatrix} \in \mathbb{R}^{n \times n}$; $\mathbf{V}_1 \in \mathbb{R}^{n \times r}$, $\mathbf{V}_2 \in \mathbb{R}^{n \times (n-r)}$

respecting $U^T \cdot U = I$ and $V^T \cdot V = I$.

In Eq. (2.27), $\Sigma_1 = \text{diag}(\sigma_i)$, $i = 1, 2, \dots, r$ with $\sigma_1 > \sigma_2 > \dots > \sigma_r > 0$. The rank of Σ_1 is r . As set \mathbf{q}_0 is complete, consistent and minimal in our case, r equals n .

Now pass \mathbf{U} matrix to the right side of Eq. (2.27)

$$\begin{pmatrix} \Sigma_1 \cdot \mathbf{V}_1^t \\ 0_{(m-r) \times n} \end{pmatrix} \cdot d\mathbf{q}_{n \times 1} = - \begin{pmatrix} \mathbf{U}_1^t \\ \mathbf{U}_2^t \end{pmatrix} \cdot \mathbf{J}_{s_{m \times l}}(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s}_{l \times 1} \quad (2.28)$$

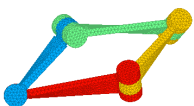
So the $m - r$ compatibility equations for assembly are obtained from

$$-\mathbf{U}_2^t \cdot \mathbf{J}_s(\mathbf{q}_0, \mathbf{s}_0) \cdot d\mathbf{s} = 0 \quad (2.29)$$

For a specified position, the above equation can also be written as follows :

$$M_{CA} \cdot d\mathbf{s} = 0 \quad (2.30)$$

Eq. (2.30) illustrates variations dependence on the specification parameters. The studied mechanism can be assembled around the initial position in respecting Eq. (2.30). But the so-called assembly equation (Eq. (2.30)) does not give any information about global mobility.



Generally, designers are interested in both assembly and mobility. For this reason, the relations in different specific positions are absolutely necessary. According to theorem 2.1.2.3, it is reasonable to ensure a mechanism's mobility with the help of the assembly relations in several positions. But theorem 2.1.2.3 gives only a boundary condition. There is no need to analyze so many positions. The question is : how many positions should be studied to obtain needed information on mobility? The first who dealt with the question of a Bennett linkage was Veldkamp [Vel67]. He reported that three instantaneous poses can be reached by two spatial RR-chains which form one mobile 4R-mechanism as solution. Tsai and Roth [Rot73] confirmed this statement by solving 10 quadratic equations for three finitely separated poses and showed that it always has two solutions which form a Bennett linkage. Using Huang's result [Hua97] and that of Tsai et al, Perez and McCarthy [McC03] improved this study by reducing 10 quadratic equations to 4, which can be solved. The combination of the two solutions of 4 polynomial equations leads to a unique Bennett linkage. In spite of all activity on Bennett linkages, it appears that few papers answer this question for Goldberg 5R linkages.

Therefore, we firstly try to analyze as many as possible positions in order to get enough information on mobility. For each position, modeling parameters \mathbf{q}_0 , as well as specification parameters which represent joint offset and joint angle, can be considered as real variables x_i in theorem 2.1.2.2. By definition, the mobility compatibility relations are given by Eq. (2.31) :

$$M_{CM} \cdot d\mathbf{s} = 0 \quad (2.31)$$

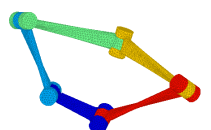
$$M_{stack} = \begin{pmatrix} M_{CA1} \\ \cdot \\ \cdot \\ \cdot \\ M_{CAk} \end{pmatrix} \quad (2.32)$$

The number of the studied positions is supposed to be k . To build M_{CM} , k assembled matrix M_{CAi} are stored in a stack matrix M_{stack} shown in Eq. (2.32). Assume that M_{stack} 's rank is r_{stack} , the number of compatibility equations, with which we need to check mobility for the same class of over-constrained mechanisms, is also r_{stack} . Moreover, the number of position that should be studied for this class is defined by

$$n_{position} = \frac{r_{stack}}{n_{assembly}} \quad (2.33)$$

where $n_{assembly}$ is the number of assembly equations for one position. That means, for each family of mobile over-constrained mechanisms, one can determine the mobility by using r_{stack} equations. And it is enough to analyze $n_{position}$ to get these r_{stack} mobility equations.

Then, matrix M_{CM} can be constructed r independent lines upon matrix M_{stack} . Thus, Eq. (2.31) transforms to



$$\begin{pmatrix} M_{CA1} \\ \cdot \\ \cdot \\ M_{CAr} \end{pmatrix} \cdot d\mathbf{s} = 0 \quad (2.34)$$

Besides, the choice of independent lines can also be automated by SVD. Expanding Eq. (2.31) via SVD yields :

$$\begin{pmatrix} \mathbf{U}_3 & \mathbf{U}_4 \end{pmatrix} \cdot \begin{pmatrix} \Sigma_2 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{V}_3 & \mathbf{V}_4 \end{pmatrix}^T \cdot d\mathbf{s} = 0 \quad (2.35)$$

Multiplying term $\begin{pmatrix} \mathbf{U}_3^{-1} & \mathbf{U}_4^{-1} \end{pmatrix}^T$ on the left in both sides of Eq. (2.35)

$$\begin{pmatrix} \Sigma_2 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{V}_3 & \mathbf{V}_4 \end{pmatrix}^T \cdot d\mathbf{s} = 0 \quad (2.36)$$

Finally, Eq. (2.36) can be written as

$$\Sigma_2 \cdot \mathbf{V}_3^t \cdot d\mathbf{s} = 0 \quad (2.37)$$

This equation allows checking the mobility of an over-constrained mechanism. Thanks to Eq. (2.37), one can produce M_{CM} with

$$M_{CM} = \Sigma_2 \cdot \mathbf{V}_3^t \quad (2.38)$$

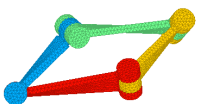
The flowchart in Fig. 2.4 shows the process of this new numerical approach. The approach will be applied on a Goldberg 5R linkage.

When expanding Eq. (2.24) via Taylor's theorem at $(\mathbf{q}_0, \mathbf{s}_0)$, we assume that the partial derivatives of \mathbf{q} and \mathbf{s} are continuous around $(\mathbf{q}_0, \mathbf{s}_0)$ and Eq. (2.21) is a function defined in the neighborhood of $(\mathbf{q}_0, \mathbf{s}_0)$. Thus, the studied mechanism can be assembled in the neighborhood of $(\mathbf{q}_0, \mathbf{s}_0)$. After that, several positions of the same size are studied because a mobile mechanism can be assembled in infinite positions. To obtain the mobility equations, the Jacobian matrix of all the positions are stored in M_{stack} . There should be enough positions for the rank of M_{stack} reaching to a plateau. SVD is then applied on M_{stack} to find out independent lines of coefficients. After combining the result coefficients of SVD with their corresponding variations of specifications, we get the mobility equations for studied mechanism.

2.4 Compatibility equations of Goldberg 5R linkage

2.4.1 Equations of Goldberg 5R linkage

There are 31 equations for the example of Goldberg 5R linkage. They are built from Denavit-Hartenberg skeleton : 5 equations represent 5 axes-distances ; 10 equations constrain perpendi-



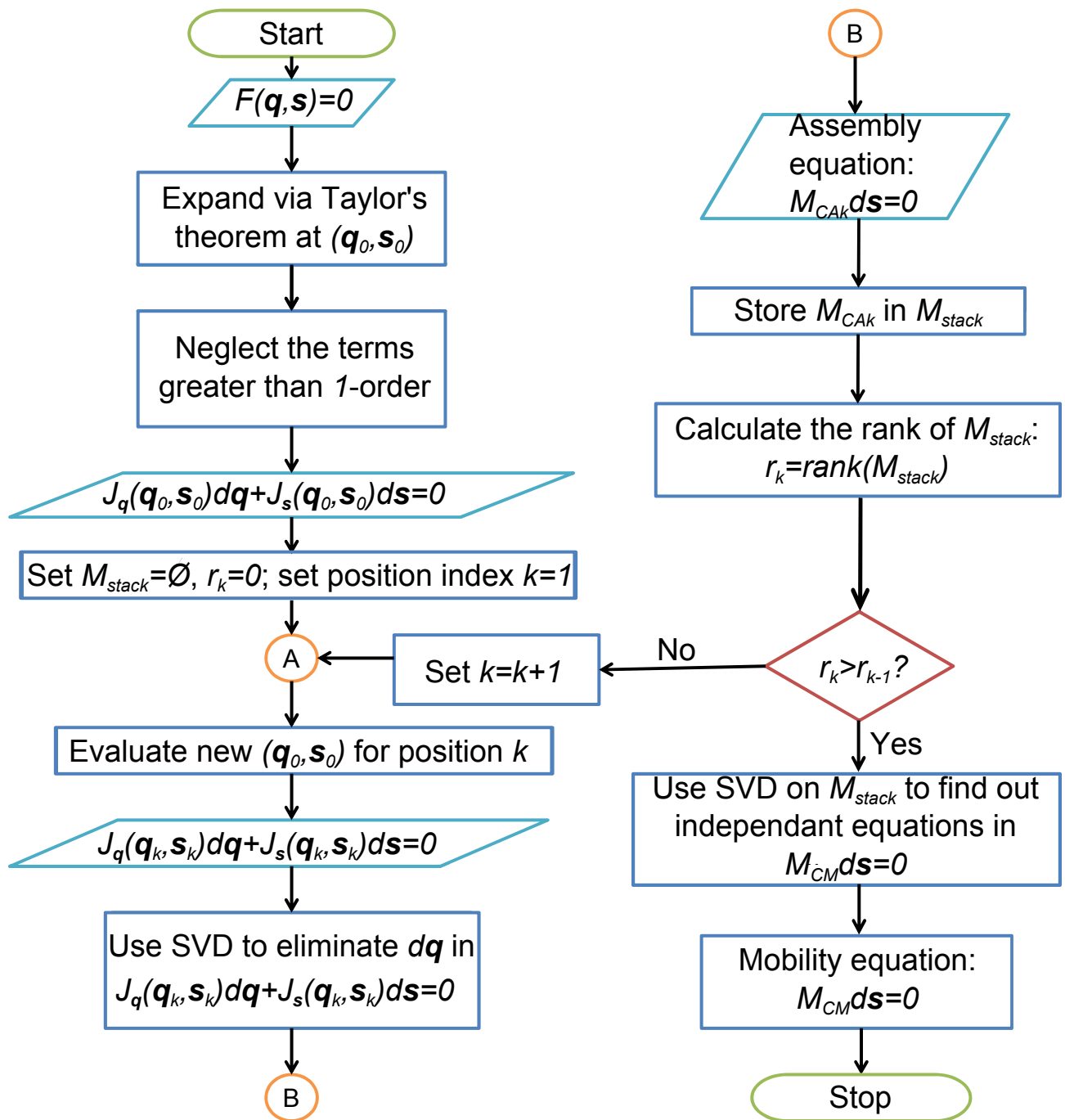
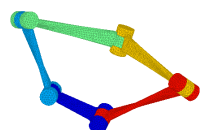


FIGURE 2.4 – A flowchart of the new numerical approach

cularity between the length of the shortest axes-distance and its 2 adjacent axes ; 5 equations reveal twist angles ; 1 equation represents the constraint of control angle named θ shown in Fig. 2.3 ; 3 closure equations result from projection ; 7 equations indicate the constraints of 7 vectors' norms. In Denavit-Hartenberg skeleton, Goldberg 5R linkage has 10 vectors : 5 vectors for axes of rotation and 5 vectors for common perpendiculars between those axes. The norm information



of 3 vectors can be found in matrix M of Eqs. (2.40d)-(2.40i). That is why $10 - 3 = 7$ other vectors norms need to be constrained.

In following equations, q_i ($i = 1, \dots, 29$) carry vectors and tensor information of skeleton. s_i ($i = 1, \dots, 21$) are specification parameters : 5 axes-distances, 5 twist angles, 1 control angle and 10 perpendicular angles.

5 equations of link length :

$$q1 - s1 = 0 \quad (2.39a)$$

$$q2 - s2 = 0 \quad (2.39b)$$

$$q3 - s3 = 0 \quad (2.39c)$$

$$q4 - s4 = 0 \quad (2.39d)$$

$$q5 - s5 = 0 \quad (2.39e)$$

16 angular equations :

$$q6 - \cos(s6) = 0 \quad (2.40a)$$

$$q15 - \cos(s7) = 0 \quad (2.40b)$$

$$q23 - \cos(s8) = 0 \quad (2.40c)$$

$$\begin{pmatrix} q8 & 16 & q23 \end{pmatrix} M \begin{pmatrix} q9 & q17 & q24 \end{pmatrix}^T - \cos(s9) = 0 \quad (2.40d)$$

$$\begin{pmatrix} q9 & 17 & q24 \end{pmatrix} M \begin{pmatrix} q10 & q18 & q25 \end{pmatrix}^T - \cos(s10) = 0 \quad (2.40e)$$

$$\begin{pmatrix} q10 & q18 & q25 \end{pmatrix} M \begin{pmatrix} q11 & q19 & q26 \end{pmatrix}^T - \cos(s11) = 0 \quad (2.40f)$$

$$\begin{pmatrix} q11 & q19 & q26 \end{pmatrix} M \begin{pmatrix} q12 & q20 & q27 \end{pmatrix}^T - \cos(s12) = 0 \quad (2.40g)$$

$$\begin{pmatrix} q12 & q20 & q27 \end{pmatrix} M \begin{pmatrix} q13 & q21 & q28 \end{pmatrix}^T - \cos(s13) = 0 \quad (2.40h)$$

$$\begin{pmatrix} q13 & q21 & q28 \end{pmatrix} M \begin{pmatrix} q14 & q22 & q29 \end{pmatrix}^T - \cos(s14) = 0 \quad (2.40i)$$

$$q14 - \cos(s15) = 0 \quad (2.40j)$$

$$q7 - \cos(s16) = 0 \quad (2.40k)$$

$$q24 - \cos(s17) = 0 \quad (2.40l)$$

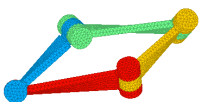
$$\begin{pmatrix} q9 & q17 & q24 \end{pmatrix} M \begin{pmatrix} q11 & q19 & q26 \end{pmatrix}^T - \cos(s18) = 0 \quad (2.40m)$$

$$\begin{pmatrix} q11 & q19 & q26 \end{pmatrix} M \begin{pmatrix} q13 & q21 & q28 \end{pmatrix}^T - \cos(s19) = 0 \quad (2.40n)$$

$$q13 - \cos(s20) = 0 \quad (2.40o)$$

$$q22 - \cos(s21) = 0 \quad (2.40p)$$

where $M = \begin{pmatrix} 1 & q6 & q7 \\ q6 & 1 & q15 \\ q7 & q15 & 1 \end{pmatrix}^{-1} = \frac{1}{\Delta} \begin{pmatrix} -1 + q15^2 & q6 - q7q15 & q7 - q6q15 \\ q6 - q7q15 & -1 + q7^2 & q15 - q7q6 \\ q7 - q6q15 & q15 - q7q6 & -1 + q6^2 \end{pmatrix}$ and $\Delta = -2q7q6q15 + q7^2 + q6^2 - 1 + q15^2$



Eqs. (2.40a)-(2.40j) are angular equations of $\langle \mathbf{e}_i, \mathbf{n}_i \rangle$ and $\langle \mathbf{e}_i, \mathbf{n}_{i+1} \rangle$ ($i = 1, \dots, 5$). Eqs. (2.40k)-(2.40o) indicate $\langle \mathbf{n}_i, \mathbf{n}_{i+1} \rangle$ ($i = 1, \dots, 5$). Similarly, Eq. (2.40p) represents $\langle \mathbf{e}_1, \mathbf{e}_5 \rangle$. Note that results of Eqs. (2.40d)-(2.40i) and Eqs. (2.40m)-(2.40n) can be calculated by Eq. (2.11).

7 equations of norm :

$$\begin{pmatrix} q8 & q16 & q23 \end{pmatrix} M \begin{pmatrix} q8 & q16 & q23 \end{pmatrix}^T - 1 = 0 \quad (2.41a)$$

$$\begin{pmatrix} q9 & q17 & q24 \end{pmatrix} M \begin{pmatrix} q9 & q17 & q24 \end{pmatrix}^T - 1 = 0 \quad (2.41b)$$

$$\begin{pmatrix} q10 & q18 & q25 \end{pmatrix} M \begin{pmatrix} q10 & q18 & q25 \end{pmatrix}^T - 1 = 0 \quad (2.41c)$$

$$\begin{pmatrix} q11 & q19 & q26 \end{pmatrix} M \begin{pmatrix} q11 & q19 & q26 \end{pmatrix}^T - 1 = 0 \quad (2.41d)$$

$$\begin{pmatrix} q12 & q20 & q27 \end{pmatrix} M \begin{pmatrix} q12 & q20 & q27 \end{pmatrix}^T - 1 = 0 \quad (2.41e)$$

$$\begin{pmatrix} q13 & q21 & q28 \end{pmatrix} M \begin{pmatrix} q13 & q21 & q28 \end{pmatrix}^T - 1 = 0 \quad (2.41f)$$

$$\begin{pmatrix} q14 & q22 & q29 \end{pmatrix} M \begin{pmatrix} q14 & q22 & q29 \end{pmatrix}^T - 1 = 0 \quad (2.41g)$$

Using Eq. (2.12), we can derive results of Eqs. (2.41a)-(2.41g). Matrix M is the same in Eqs. (2.40d)-(2.40i).

The loop-closure equation of a general spatial five-bar linkage, shown in Fig. 2.2, is given by

$$a_1 \mathbf{f}_1 + d_1 \mathbf{e}_1 + a_2 \mathbf{f}_2 + d_2 \mathbf{e}_2 + a_3 \mathbf{f}_3 + d_3 \mathbf{e}_3 + a_4 \mathbf{f}_4 + d_4 \mathbf{e}_4 + a_5 \mathbf{f}_5 + d_5 \mathbf{e}_5 = 0 \quad (2.42)$$

Because there is no joint offset in Goldberg 5R linkage, we assume that $d_i = 0$ for $i = 1, 2, \dots, 5$. Hence, the loop-closure equation becomes

$$a_1 \mathbf{f}_1 + a_2 \mathbf{f}_2 + a_3 \mathbf{f}_3 + a_4 \mathbf{f}_4 + a_5 \mathbf{f}_5 = 0 \quad (2.43)$$

Furthermore, by dotproducting $(\mathbf{e}_1, \mathbf{f}_1, \mathbf{e}_2)$ with Eq. (2.43), we obtain closure equations.

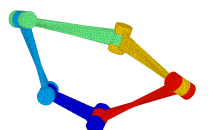
3 closure equations :

$$q1q6 + q2q8 + q3q10 + q4q12 + q5q14 = 0 \quad (2.44a)$$

$$q1 + q2q16 + q3q18 + q4q20 + q5q22 = 0 \quad (2.44b)$$

$$q1q15 + q2q23 + q3q25 + q4q27 + q5q29 = 0 \quad (2.44c)$$

Therefore, Eqs. (2.40a)-(2.40p), Eqs. (2.41a)-(2.41g), Eqs. (2.39a)-(2.39e) and Eqs. (2.44a)-(2.44c) are equations system of Goldberg 5R linkage. These equations can be grouped into Eq. (2.20).



q1	0.100000	q11	-0.760301	q21	0.324912	s1	0.100000	s11	1.570806
q2	0.140000	q12	0.228039	q22	0.939725	s2	0.140000	s12	1.570806
q3	0.200000	q13	0.312265	q23	0	s3	0.200000	s13	1.570806
q4	0.240001	q14	0	q24	0.746803	s4	0.240001	s14	1.570806
q5	0.200000	q15	0	q25	-0.227496	s5	0.200000	s15	1.570806
q6	0	q16	0.936088	q26	-0.360457	s6	1.570806	s16	0.495032
q7	0.880000	q17	-0.234146	q27	0.324953	s7	1.570806	s17	0.727544
q8	-0.167258	q18	-0.969502	q28	0.698865	s8	1.570806	s18	-1.253272
q9	0.361609	q19	-0.003724	q29	-0.162585	s9	1.570806	s19	1.222572
q10	-0.156608	q20	-0.937987	s21	0.349111	s10	1.570806	s20	-1.253272

TABLE 2.2 – Values of $(\mathbf{q}_0, \mathbf{s}_0)$ when $\theta = 20^\circ$

2.4.1.1 Compatibility equations of one position

As discussed in section 3, Eq. (2.26) is Taylor’s expansion at the first-order. Substituting the conditions of $\theta = 20^\circ$ into Eq. (2.26) yields

$$\mathbf{J}_{\mathbf{q}(\theta=20)} \cdot \begin{pmatrix} dq_1 & \dots & dq_{29} \end{pmatrix}^{\mathbf{T}} + \mathbf{J}_{\mathbf{s}(\theta=20)} \cdot \begin{pmatrix} ds_1 & \dots & ds_{21} \end{pmatrix}^{\mathbf{T}} = 0 \quad (2.45)$$

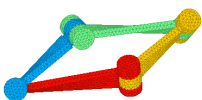
whence $d\mathbf{q} = \begin{pmatrix} dq_1 & \dots & dq_{29} \end{pmatrix}^{\mathbf{T}}$ is the vector of modeling parameters’ variation, while $d\mathbf{s} = \begin{pmatrix} ds_1 & \dots & ds_{21} \end{pmatrix}^{\mathbf{T}}$ is the vector of specification parameters’ variation.

$$\mathbf{J}_{\mathbf{q}(\theta=20)} = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1.0531 & \dots & -1.6257 & \dots & 1.9 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0.1 & \dots & 0.14 & \dots & 0.2 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & -1.1908 & \dots & 1.3532 & \dots & 0 & \dots & -1.4401 \end{pmatrix}$$

and

$$\mathbf{J}_{\mathbf{s}(\theta=20)} = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & -0.9399 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & -0.3420 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

The sizes of $\mathbf{J}_{\mathbf{q}(\theta=20)}$ and $\mathbf{J}_{\mathbf{s}(\theta=20)}$ are 31×29 and 31×21 , respectively. Moreover the structure of the jacobian is simple : it is quite sparse, most of the nonzero terms form diagonal blocks and triangular blocks. The values of $(\mathbf{q}_0, \mathbf{s}_0)$ in position $\theta = 20^\circ$ are in Table 2.



After SVD calculus, the rank of $\mathbf{J}_{q(\theta=20)}$ is 29. According to Eq. (2.29), there are $31 - 29 = 2$ compatibility equations for the specified position $\theta = 20^\circ$.

Now, we select 11 of 21 *dsi* because there is no need to impose the variation of perpendicular angles. This perpendicularity plays central role in Denavit-Hartenberg representation and should be always guaranteed. Suppose that $ds = \left(da_1 \ da_2 \ da_3 \ da_4 \ da_5 \ d\alpha_1 \ d\alpha_2 \ d\alpha_3 \ d\alpha_4 \ d\alpha_5 \ d\theta \right)^T$, when θ is 20° , the compatibility relation is

$$M_{CA_{\theta=20}^\circ} \cdot ds = 0 \quad (2.46)$$

with

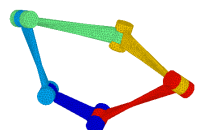
$$M_{CA_{\theta=20}^\circ} = -\mathbf{U}_2^t \cdot \mathbf{J}_s = \begin{pmatrix} 0.356383 & -0.192224 \\ 0.327308 & -0.138595 \\ -0.285693 & 0.257982 \\ -0.419513 & 0.076885 \\ 0.381802 & -0.157116 \\ 0.115033 & 0.261872 \\ 0.117835 & 0.250207 \\ -0.100496 & -0.247902 \\ -0.103338 & -0.240504 \\ 0.106814 & 0.254533 \\ 0 & 0 \end{pmatrix}^T$$

This set of relations is called assembly equation in which $M_{CA_{\theta=20}^\circ}$ is assembly matrix. Each line in $M_{CA_{\theta=20}^\circ}$ corresponds to an assembly equation while each column relates to a specification parameter. When the above relations are respected, the mechanism may be assembled in the neighborhood of the initial position $\theta = 20$ degree. Here, we focus only on variations of link's lengths, variations of twist angles and control angle (eleven specification variations). Note that in $M_{CA_{\theta=20}^\circ}$ matrix, the coefficients of $d\theta$ parameter vanishes. This proves that the dimensional variations of the bars cannot be "corrected" by the variation of the control angle. It is the general case of mechanisms with one single degree of freedom.

In fact, specification parameters mentioned in this paper includes constant parameters and variables. For example, the length of a link (a_i) could be a constant parameter and joint angle θ could be a variable. When Goldberg 5R linkage is in different positions, only θ varies. In a specific position, the set of specification parameters \mathbf{s} becomes \mathbf{s}_0 , which is a part of the given fixed point $(\mathbf{q}_0, \mathbf{s}_0)$. At the beginning of this section, the 11 specification parameters are fixed values when $\theta = 20^\circ$. Then \mathbf{q}_0 and \mathbf{s}_0 are assigned to Eq. (2.26) to calculate Eq. (2.46). That is why Eq. (2.46) is in function of ds , but not \mathbf{s} .

2.4.2 Calculation of mobility equations number

Afterwards, we study 14 Goldberg's positions that the control angle θ varies from 20 degree to 85 degree with a step of 5 degree. Fig. 2.5 demonstrates the Goldberg's gradual process with each bar and its axis. Bar 5 is immobile. So M_{stack} is given by



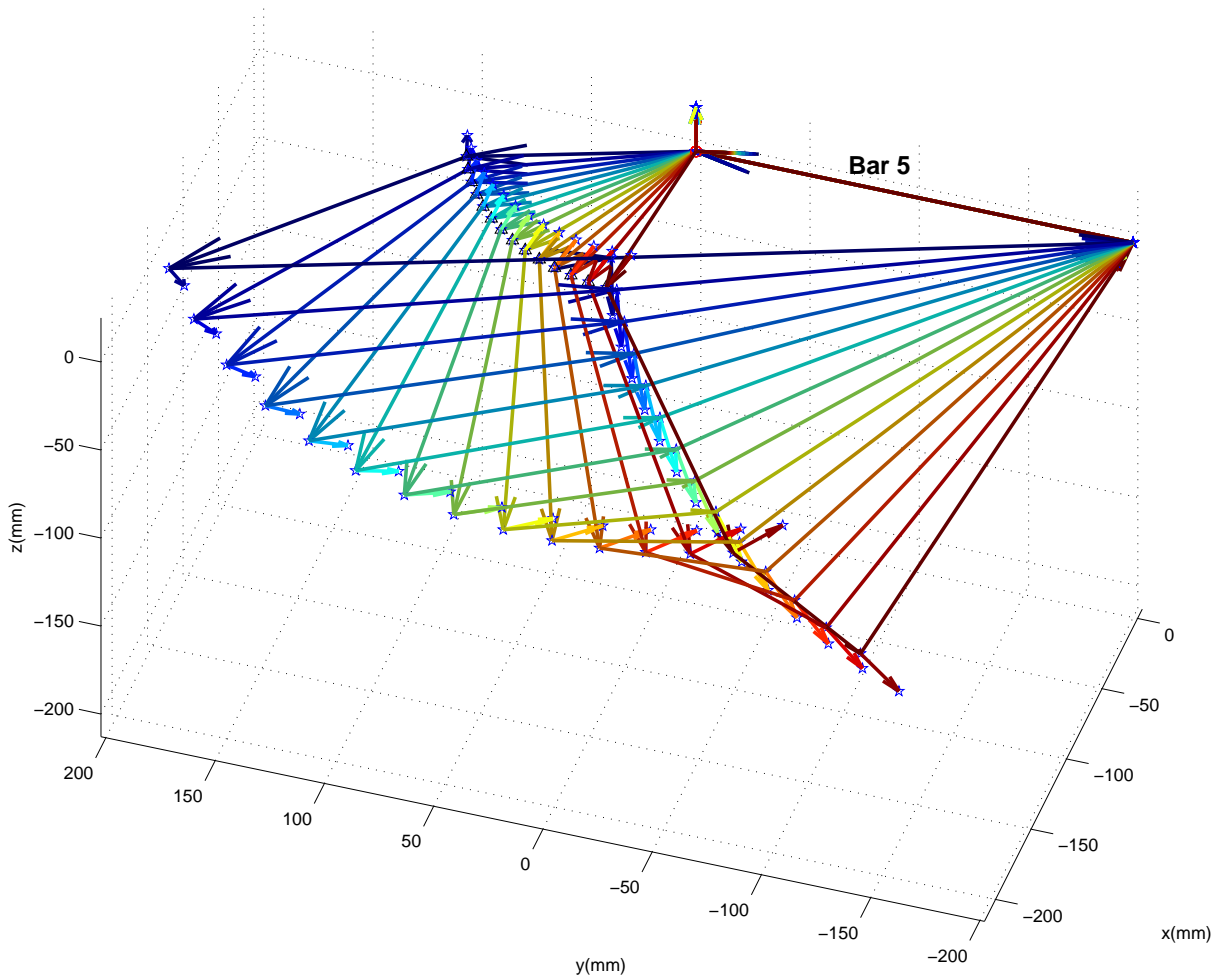
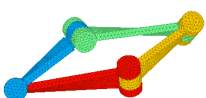


FIGURE 2.5 – Goldberg's 14 positions

$$M_{stack} = \begin{pmatrix} M_{CA}_{\theta=20^\circ} \\ M_{CA}_{\theta=25^\circ} \\ M_{CA}_{\theta=30^\circ} \\ \vdots \\ \vdots \\ \vdots \\ M_{CA}_{\theta=85^\circ} \end{pmatrix}_{28 \times 11} \quad (2.47)$$

As mentioned earlier, there are two compatibility equations for each position. And each column represents a deviation parameter. Hence, the M_{stack} 's size is 28×11 (14 positions and 11 specification variations).

Fig. 2.6 shows the growth of M_{stack} 's rank, with an accuracy of six decimal places. M_{stack} 's rank climbs by following the size of M_{stack} . From 4th position, the rank of M_{stack} holds steady on six. Hence, we deduce that there should be six independent equations to determine a mobile Goldberg 5R linkage. According to the flowchart in Fig. 2.4 and the rank of M_{stack} in Fig. 2.6,



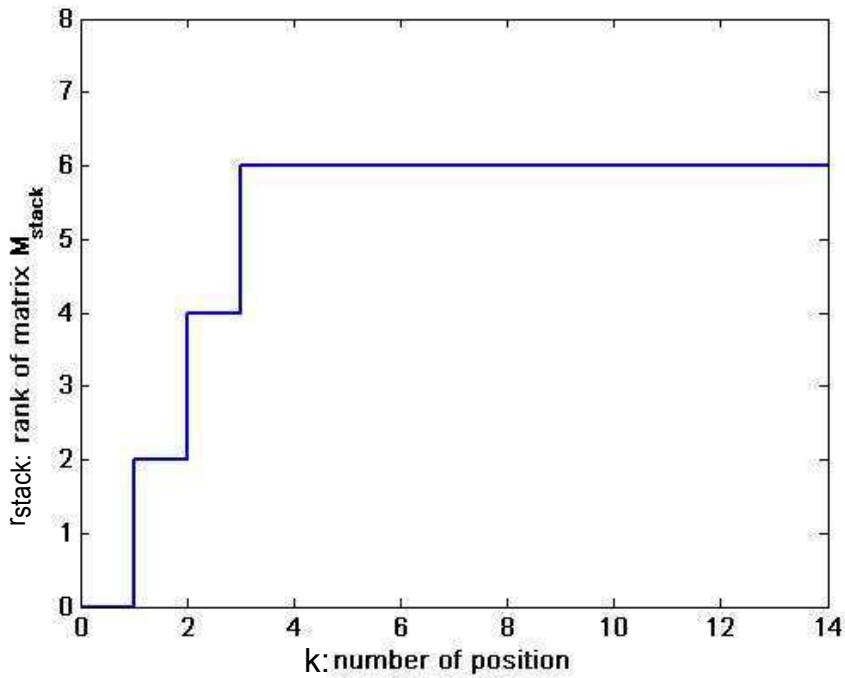


FIGURE 2.6 – Rank increasing with the growth of M_{stack}

Bar's number	Length(mm) - a_i	Angle(degree) - α_i
1	90.000000	17.457603
2	75.000000	14.477512
3	150.000000	-30.000000
4	165.000000	31.935115
5	150.000000	-30.000000

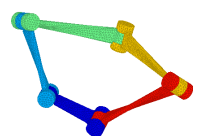
TABLE 2.3 – Second configuration of Goldberg 5R

four positions are enough for the first study. But 14 positions are studied because we want to make sure that the mobility equations are correct. To prove this, the author also tests another configuration of Goldberg 5R linkage (listed in Table. 2.3) and M_{stack} 's rank is always six. That means, it is enough to study maximum 3 positions for any mobile 5R linkage in order to get its six compatibility equations. Similarly, this numerical method can be used to determine any other mobile over-constrained mechanism.

Till now, the obtained result reveals that there should be six equations for Goldberg 5R linkage family. These equations will be presented in next subsection.

2.4.3 Mobility equations of specified Goldberg 5R linkage

The mobility equations for configuration 1 in Table. 2.1 are displayed in Eq. (2.48), resulting from the presented method in section 3.



$$M_{CM_{config1}} \cdot ds = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.48)$$

with

$$M_{CM_{config1}} = \begin{pmatrix} 1.017389 & 0.787801 & 0.052111 & -0.034556 & -0.059004 & -0.010678 \\ 0.778659 & -0.013343 & -0.151619 & 0.396105 & 0.002374 & 0.008439 \\ 0.033706 & -1.553493 & -0.339085 & -0.123146 & -0.010228 & -0.004135 \\ -1.982313 & 0.442776 & 0.243400 & 0.026330 & -0.004775 & 0.001669 \\ 1.291313 & 0.637601 & 0.127083 & -0.168447 & 0.043798 & 0.001564 \\ 0.386923 & -0.622911 & 0.267006 & -0.003308 & -0.004231 & 0.013043 \\ 0.397411 & -0.462454 & 0.307322 & -0.071251 & -0.015669 & 0.009785 \\ -0.123260 & 0.493642 & -0.315846 & -0.105576 & -0.009720 & 0.016623 \\ -0.208161 & 0.394934 & -0.321752 & 0.004831 & 0.016046 & -0.011374 \\ 0.210363 & -0.553850 & 0.301910 & 0.086408 & 0.011927 & -0.016792 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad (2.49)$$

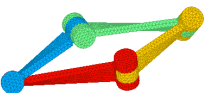
And for the second configuration, the compatibility equations are

$$M_{CM_{config2}} \cdot ds = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.50)$$

with

$$M_{CM_{config2}} = \begin{pmatrix} -1.296935 & -0.104730 & -0.099015 & -0.004031 & 0.004191 & 0.004019 \\ -1.267785 & -0.167582 & -0.109018 & -0.046566 & 0.003633 & -0.003824 \\ 0.935920 & -0.509198 & -0.194681 & 0.013387 & -0.005195 & 0.000063 \\ 1.377502 & 0.202487 & 0.135640 & -0.012232 & 0.006534 & -0.000184 \\ -1.039118 & 0.433091 & 0.159395 & 0.025771 & -0.006323 & -0.000360 \\ -0.249058 & -0.362607 & 0.110266 & 0.056906 & 0.000314 & -0.001503 \\ -0.257872 & -0.344770 & 0.113014 & 0.069332 & 0.000430 & 0.000759 \\ 0.054489 & 0.316198 & -0.165595 & 0.084867 & 0.000317 & -0.000673 \\ 0.226002 & 0.334631 & -0.120747 & -0.052252 & -0.003383 & 0.000405 \\ -0.081301 & -0.335971 & 0.156427 & -0.074694 & -0.003310 & 0.000596 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad (2.51)$$

In this section, the number of mobility equations for the family of 5R linkage is determined. Then two types of compatibility equations are interpreted : assembly and mobility. The first ensures that the system can be assembled. The second checks mechanism's mobility. To generate assembly equations Eq. (2.46), we substitute the values of $(\mathbf{q}_{\theta=20}, \mathbf{s}_{\theta=20})$ into Eq. (2.26). SVD is then used to eliminate $d\mathbf{q}$. That is why the assembly equations are in function of ds . Afterwards, mobility equations Eq. (2.48) result from the assembly equations of several positions. And the number of mobility equations is determined by the matrix M_{stack} 's rank. The application of compatibility equations will be illustrated later.



2.5 Application of generated mobility equations

In the example of Goldberg five-bar linkage, matrix M_{CM} 's rank is six. It means that any six $ds\mathbf{1} = (ds_1, ds_2, ds_3, ds_4, ds_5, ds_6)$ of eleven ds can be calculated while the other five $ds\mathbf{2} = (ds_7, ds_8, ds_9, ds_{10}, ds_{11})$ are given. The partitioning choice of eleven specifications is quite diverse. For instance, the partition of ds for numerical application in this subsection is $ds\mathbf{1} = (da_1, d\alpha_1, da_2, d\alpha_2, da_5, d\alpha_5,)$ and $ds\mathbf{2} = (da_3, d\alpha_3, da_4, d\alpha_4, d\theta)$.

According to the above definition, Eq. (2.48) can be written as follows :

$$\left(A_{6 \times 6} \mid B_{6 \times 5} \right) \cdot \left(\frac{ds_{16 \times 1}}{ds_{25 \times 1}} \right) = 0 \quad (2.52)$$

$ds_{16 \times 1}$ is the target set of specifications that should be identified thanks to generated mobility equations. Matrix $A_{6 \times 6}$ is invertible, so that Eq. (2.52) becomes

$$ds\mathbf{1} = -A^{-1} \cdot B \cdot ds\mathbf{2} \quad (2.53)$$

And the relation of eleven ds for studied partition gives

$$\begin{pmatrix} da_1 \\ d\alpha_1 \\ da_2 \\ d\alpha_2 \\ da_5 \\ d\alpha_5 \end{pmatrix} = \begin{pmatrix} -7.431227 & -0.488505 & 6.609356 & -1.039190 & 0 \\ -42.811293 & -2.814278 & 35.676078 & -5.609356 & 0 \\ 7.431227 & 0.488505 & -5.609356 & 1.039190 & 0 \\ 42.811293 & 2.814278 & -35.676078 & 6.609356 & 0 \\ 1.000000 & 0 & 0 & 0 & 0 \\ 0 & 1.000000 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} da_3 \\ d\alpha_3 \\ da_4 \\ d\alpha_4 \\ d\theta \end{pmatrix} \quad (2.54)$$

As predicted, the variation of the control parameter θ will not affect the mechanism's size because the last column of the matrix vanishes.

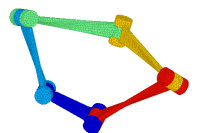
A numerical application is given below

$$ds\mathbf{1} = \begin{pmatrix} da_1 \\ d\alpha_1 \\ da_2 \\ d\alpha_2 \\ da_5 \\ d\alpha_5 \end{pmatrix} = \begin{pmatrix} 0.004066 \\ 0.024044 \\ -0.003066 \\ -0.016045 \\ -0.001000 \\ 0.003401 \end{pmatrix} \quad (2.55)$$

with

$$ds\mathbf{2} = \begin{pmatrix} da_3 \\ d\alpha_3 \\ da_4 \\ d\alpha_4 \\ d\theta \end{pmatrix} = \begin{pmatrix} -0.00100 \\ 0.003401 \\ 0.001000 \\ 0.008000 \\ 1.000000 \end{pmatrix} \quad (2.56)$$

In Eq. (2.55) and (2.56), the length (a_i) unity is meter and the angle (α_i and θ) unit is radian. Suppose that there are five designers, each designer works on one link. If designer 1, 2 and 5



Bar's Num.	Initial Length (mm)	Final Length (mm)	Initial Angle (deg.)	Final Angle (deg.)
1	100.000000	104.066326	28.359350	29.737011
2	140.000000	136.933674	41.682325	40.762997
3	200.000000	199.000000	-71.805128	-72.000000
4	240.000000	241.000000	70.041675	70.500000
5	200.000000	199.000000	-71.805128	-72.000000

TABLE 2.4 – Initial and final configuration of Goldberg 5R

modify respectively the length and the twist angle of link 1, 2 and 5 ; designer 3 and 4 can directly enter correspondent values with the help of Eq. (2.54).

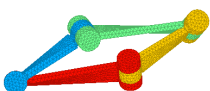
This application is also a validation which proves the feasibility of the proposed method. It does not allow obtaining the formal relations but it works for all over-constrained mechanisms. The authors would like to point out that the obtained compatibility relations are valid only in the neighborhood of the nominal mechanisms.

Table 2.4 reveals the variation between the initial size and the final size of Goldberg 5R linkage. The geometric conditions in initial configuration such as $\alpha_3 = \alpha_5$ and $a_3 = a_5$ are also respected in final configuration. Subsequently, it is quite evident that the compatibility equations can help designers to determine mobility. As said in section 1.1, these compatibility equations can be used by designers who do not know the geometric conditions of studied mechanism because these conditions are already taken into account in mobility equations. Therefore, the mobility equations should result from a mobile mechanism in order to ensure the correctness.

2.6 Conclusion and perspectives

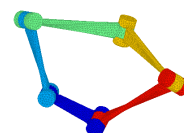
This paper presents a numerical approach to determine the number of mobility equations for a family of over-constrained mechanisms. This generic approach works on all spatial over-constrained mechanisms and helps generating mobility equations for a specified configuration based on mobile system. Currently, only one family of over-constrained mechanism is tested. Regarding the result, there should be six mobility equations to preserve the mobility of Goldberg 5R family. For future work, other special position such as overlap position should be proceed in order to improve the robustness of the prototype for industrial use. An application which allows creating a new configuration of mobile Goldberg 5R linkage is presented. A set of compatibility equations is provided for a studied mechanism Goldberg 5R linkage, and the compatibility equations are valid in the neighborhood of the nominal mechanism. All these efforts are aimed at promoting the consistency of over-constrained mechanisms during teamwork, which can not only be important from a theoretical point of view but also have industrial application value : the proposed approach makes all design parameters directly and independently editable by a design team.

Although the presented approach shows its ability to assist mechanical designers, there remains much to be done before a truly practical system can be built to guarantee the consistency of a complex mechanism. The proposed method addresses well the 5-bar family, but more tests on complex mechanisms such as 6R linkage are required in order to enrich this method. Besides,



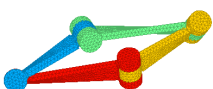
Conclusion and perspectives

the formal compatibility relation is also an area where further research is required. Another aspect lies in merging this method with other domain such as thermodynamics, electricity, vibration and acoustics, etc. When more types of engineering data is taken into consideration, compatibility equations will possess more significant physical meanings.

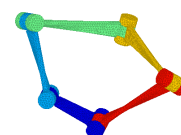


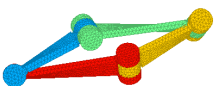
Bibliographie du chapitre 2

- [Bed07] C. Mavroidis & M. Beddows. A spatial overconstrained mechanism that can be used in practical applications, 2007.
- [Ben03] G.T. Bennett. A new mechanism. *Engineering*, 76 :777–778, 1903.
- [Che03] Y. Chen. *Design of Structural Mechanisms*. PhD thesis, Department of Engineering Science at the University of Oxford, 2003.
- [Che11] C. Chen. The order of local mobility of mechanisms. *Mechanism and Machine Theory*, 46 :1251–1264, 2011.
- [Cho11] R. Liu & P. Serré & J.-Y. Choley. An approach to manage over-constrained mechatronic system in teamwork. In *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA 2011)*, volume 2, pages 983–988, Beijing (China), August 7-11, 2011.
- [Clé09] P. Serré & F. MHenni & A. Clément. A Parametric Approach to Determine Minimum Clearance in Overconstrained Mechanisms. In *11th CIRP International Seminar on Computer-Aided Tolerancing, Annecy, France, March 26-27, 2009*, 2009.
- [Cua06] M. González & D. Dopico & U. Lugrís & J. Cuadrado. A benchmarking system for mbs simulation software : Problem standardization and performance measurement. *Multibody System Dynamics*, 16 (2) :179–190, 2006.
- [Die95] P. Dietmaier. A new 6R space mechanism. in : *Proceeding of 9th World Congress IFToMM, Milano*, 1 :52–56, 1995.
- [Gog08] G. Gogu. *Structural Synthesis of Parallel Robots*, volume 149 of *Solid Mechanics and Its Applications*, chapter Part 1 : Methodology, page 716. Springer, 2008.
- [Gog09] G. Gogu. *Structural Synthesis of Parallel Robots*, chapter Part 2 : Translational Topologies with Two and Three Degrees of Freedom, page 775. *Solid Mechanics and Its Applications*. Springer, 2009.
- [Gol43] M. Goldberg. New five-bar and six-bar linkages in three dimensions. *ASME Trans*, 65 :183–210, 1943.
- [Her78] J. Hervé. Principes fondamentaux d’une théorie des mécanismes. *Revue Roumaine des Sciences Techniques - Série de Mécanique Appliquée*, 23(5) :693–709, 1978.
- [Hua97] C. Huang. The cylindroid associated with finite motions of the bennett mechanism. *Journal of Mechanical Design*, 119(4) :521–524, 1997.
- [McC03] A. Perez & J. M. McCarthy. Dimensional synthesis of Bennett linkages. *ASME Journal of mechanical design*, 125 :98–104, 2003.
- [Mel04] N. O. Melin. *Application of Bennett Mechanisms to Long-Span Shelters*. PhD thesis, Department of Engineering Science at the University of Oxford, 2004.
- [Mil64] J. Milnor. On the Betti numbers of real varieties. In *Proceedings of the American Mathematical Society*, volume 15, pages 275–280, 1964.



- [Moi10] G. Mandil & P. Serré & M. Moinet. Geometric tool for the description and control of functional specification at the conceptual design phase. *International Journal of Design and Innovation Research (IJODIR)*, 5 :23–39, 2010.
- [Mya31] F. E. Myard. Contribution à la géométrie des systèmes articulés. *Société Mathématiques de France*, 59 :183–21, 1931.
- [O’S00] D. Cox & J. Little & D.-D. O’Shea. *Ideals, Varieties and Algorithms*, chapter An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, 2000.
- [Riv07] Faïda Mhenni & Philippe Serré & Abdelfattah Mlika & Lotfi Romdhane & Alain Rivière. Dependency between dimensional deviations in overconstrained mechanisms. In *Conception et Production Intégrées*, 2007.
- [Rot73] L. W. Tsai & B. Roth. A note on the design of Revolute-Revolute cranks. *Mechanisms and Machine Theory*, 3 :23–31, 1973.
- [Rot94] C. Mavroidis & B. Roth. Analysis and synthesis of overconstrained mechanisms. in : *Proceedings of the 1994 ASME Design Technical Conferences, Minneapolis, MI, September*, pages 115–133, 1994.
- [Vel67] G.R. Veldkamp. Canonical systems and instantaneous invariants in spatial kinematics. *Journal of Mechanisms*, 3 :329–388, 1967.
- [Wal07] A.H. Wallace. *An Introduction to algebraic topology, Dover books on Mathematics*. 2007.
- [Woh91] K. Wohlhart. Merging two general Goldberg 5R linkages to obtain a new 6R space mechanism. *Mechanism and Machine Theory*, 26(2) :659–668, 1991.
- [You07] Y. Chen & Z. You. Spatial 6R linkages based on the combination of two Goldberg 5R linkage. *Mechanism and Machine Theory*, 42 :1484–1498, 2007.
- [You08] Y. Chen & Z. You. An extended Myard linkage and its derived 6R linkage. *Journal of Mechanical Design*, 130(5) :23011–23018, 2008.



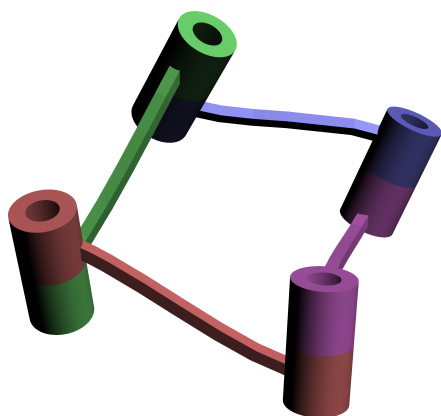


Chapitre 3

Modélisation tensorielle des systèmes mécaniques

Sommaire

3.1	Introduction	90
3.2	Génération des Fonctions F_i	90
3.2.1	Formule d'Olinde-Rodrigues	91
3.2.2	Cas d'une chaîne ouverte de solides	91
3.2.3	Cas d'une chaîne fermée de solides	93
3.3	Mise en équation d'un système de solides	95
3.3.1	Méthode basée sur les fonctions F_i	95
3.3.2	Méthode de Jin et Yang	96
3.3.3	Variante de la méthode de Jin et Yang	99
3.4	Application au système de solides 4R	100
3.4.1	Méthode basée sur les fonctions F_i	100
3.4.2	Méthode de Jin et Yang	108
3.4.3	Variante de la méthode de Jin et Yang	109
3.4.4	Discussion	109
3.5	Conclusion	116
	Bibliographie du chapitre 3	117



Résumé

Ce chapitre présente une écriture formelle de la matrice de Gram de rang 3. On utilise les fonctions d'Olinde-Rodrigues basées sur les vecteurs dans l'espace $3D$. Différentes méthodes pour générer des équations sont détaillées et illustrées sur le 4R. Une analyse manuelle est faite à la fin du chapitre.

3.1 Introduction

Le chapitre précédent a présenté pour la notation D-H les trois types de paramètres qui peuvent apparaître dans le système d'équations :

- les paramètres d'usinage, c'est-à-dire l'angle dimensionnel α_i dans une pièce et la longueur d'une pièce L_i ;
- les paramètres de mobilité, c'est-à-dire l'angle relatif d'une pièce par rapport à l'autre (θ_i pour la liaison pivot) et la distance de translation Lj_i d'une liaison prismatique.
- les paramètres de description, qui permettent de décrire la position et l'orientation des pièces par rapport à une référence.

En toute généralité, le système d'équations algébriques qui décrit la fermeture d'un système de solides s'écrit :

$$C(x, u, m) = 0$$

avec $x \in X$, $u \in U$, $m \in M$. X est l'espace des paramètres de description (exemple : coordonnées cartésiennes) ; U est l'espace des paramètres d'usinage ; M est l'espace des paramètres de mobilité.

Si l'on veut obtenir les conditions d'assemblage et de mobilité qui ne dépendent que des paramètres d'usinage u , il faut donc éliminer x et m . Pour cette raison, moins il y a de paramètres à éliminer, plus l'obtention de ces conditions aisée sera. Mais, la méthode qui permet d'obtenir les conditions de mobilité n'est pas unique. Dans le chapitre 2, nous avons généré les équations de mobilité à partir de plusieurs positions d'un système de solides. Ce sujet reste ouvert pour la suite.

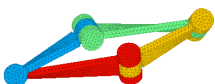
Les mises en équations "Fonction F_i " et "Jin et Yang" ont attiré notre attention car elles ne dépendent que de u et m et nous pensons que le traitement de ce système d'équations sera plus simple. Dans les deux prochaines sections, nous n'aborderons ces deux mises en équations.

3.2 Génération des Fonctions F_i

Dans cette partie, la mise en équation est réalisée à partir d'un modèle vectoriel. Pour définir l'espace vectoriel associé, nous utilisons la matrice de Gram $G(N)$ unitaire associée à l'ensemble N des n vecteurs $\vec{e}_1, \dots, \vec{e}_n$. Chaque terme de cette matrice représente les spécifications angulaires nécessaires à la description du système de solides.

$$G(N) = \begin{pmatrix} 1 & \langle \vec{e}_1, \vec{e}_2 \rangle & \dots & \langle \vec{e}_1, \vec{e}_n \rangle \\ \langle \vec{e}_2, \vec{e}_1 \rangle & 1 & \dots & \langle \vec{e}_2, \vec{e}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{e}_n, \vec{e}_1 \rangle & \langle \vec{e}_n, \vec{e}_2 \rangle & \dots & 1 \end{pmatrix} \quad (3.1)$$

La mise en équation du problème est réalisée par l'écriture des équations qui proviennent de la spécification implicite suivante : puisque la réalisation appartient à l'espace 3D, il faut que la matrice de Gram $G(N)$ soit au maximum de rang 3.



3.2.1 Formule d'Olinde-Rodrigues

Les fonctions F_i avec ($i = 1, 2, \dots$) sont utilisées pour modéliser les relations entre les termes de la matrice de Gram. Il s'agit des rotations successives qui permettent de passer d'un vecteur à un autre. Ces rotations sont exprimées avec la formule vectorielle d'Olinde-Rodrigues [Rod40] (3.2).

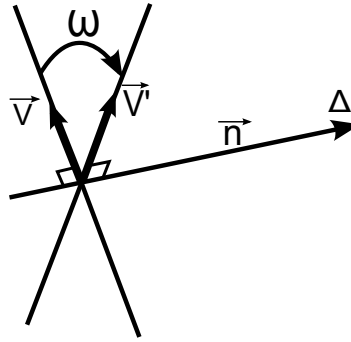


FIGURE 3.1 – Modèle vectoriel d'Olinde-Rodrigues

Voyons la figure 3.1, on désigne par R_{ω}^{Δ} la rotation d'angle ω autour de la droite Δ dans l'espace vectoriel à 3 dimensions. Soient \vec{n} le vecteur unitaire de Δ , \vec{V} un vecteur quelconque. L'expression du vecteur \vec{V}' , résultant de la rotation R_{ω}^{Δ} du vecteur \vec{V} d'un angle ω autour du vecteur Δ est donnée par la formule (3.2) :

$$\vec{V}' = \langle \vec{n}, \vec{V} \rangle \cdot (1 - \cos \omega) \cdot \vec{n} + \cos \omega \cdot \vec{V} + \sin \omega \cdot (\vec{n} \times \vec{V}) \quad (3.2)$$

Notons $\vec{V}' = R[\vec{n}, \omega](\vec{V})$.

3.2.2 Cas d'une chaîne ouverte de solides

L'équation (3.2) appelée la *formule d'Olinde-Rodrigues* est appliquée au squelette d'une chaîne ouverte, voir la figure 3.2. Le vecteur \vec{n}_2 est l'image du vecteur \vec{n}_1 par la rotation $R_{\alpha_1}^{\Delta_1}$ d'angle α_1 autour de la droite Δ_1 de vecteur unitaire \vec{e}_1 , c'est-à-dire :

$$\vec{n}_2 = R[\vec{e}_1, \alpha_1](\vec{n}_1)$$

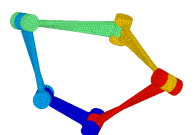
De la même manière, le vecteur \vec{e}_2 est l'image du vecteur \vec{e}_1 par la rotation $R_{\alpha_1}^{\Delta_1}$ et la rotation d'angle θ_2 autour de la droite Δ_2 de vecteur unitaire \vec{n}_2 , c'est à dire :

$$\vec{e}_2 = R[\vec{n}_2, \theta_2](\vec{e}_1) = R[R[\vec{e}_1, \alpha_1](\vec{n}_1), \theta_2](\vec{e}_1)$$

En continuant encore une fois, le vecteur \vec{n}_3 est l'image du vecteur \vec{n}_2 par la rotation $R_{\alpha_2}^{\Delta_2}$ et la rotation d'angle α_2 autour de la droite Δ_2 de vecteur unitaire, c'est à dire :

$$\vec{n}_3 = R[\vec{e}_2, \alpha_2](\vec{n}_2) = R[R[R[\vec{e}_1, \alpha_1](\vec{n}_1), \theta_2](\vec{e}_1), \alpha_2](R[\vec{e}_1, \alpha_1](\vec{n}_1))$$

Ainsi, nous savons exprimer de manière systématique un vecteur par rapport à une base de trois vecteurs, ici la base $(\vec{n}_1, \vec{e}_1, (\vec{n}_1 \times \vec{e}_1))$.



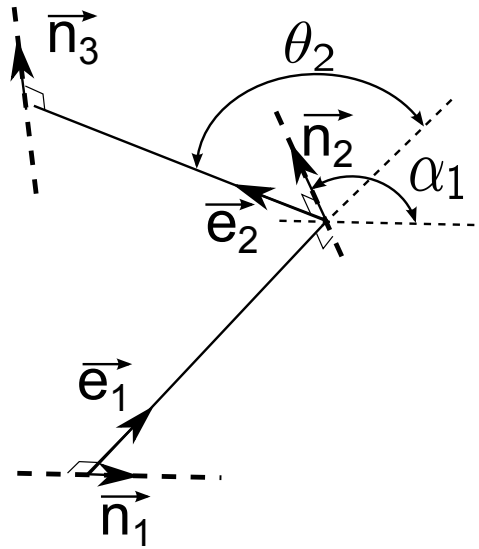


FIGURE 3.2 – Un squelette composé de 5 vecteurs

Connaissant l'expression de tous les vecteurs de la séquence, il est possible d'exprimer les produits scalaires de tous les vecteurs les uns par rapport aux autres.

Écrivons, par exemple, les 2 produits scalaires $\langle \vec{n}_1, \vec{e}_2 \rangle$ et $\langle \vec{n}_3, \vec{n}_1 \rangle$ pour le squelette de la figure 3.2.

3.2.2.1 Expression formelle du produit scalaire $\langle \vec{n}_1, \vec{e}_2 \rangle$

Nous venons de montrer que :

$$\vec{e}_2 = R[\vec{n}_2, \theta_2](\vec{e}_1) = \langle \vec{n}_2, \vec{e}_1 \rangle \cdot (1 - \cos \theta_2) \cdot \vec{n}_2 + \cos \theta_2 \cdot \vec{e}_1 + \sin \theta_2 \cdot (\vec{n}_2 \times \vec{e}_1) \quad (3.3)$$

Dans l'exemple, l'axe de liaison est perpendiculaire à l'axe de la pièce, le produit scalaire $\langle \vec{n}_2, \vec{e}_1 \rangle$ est nul et l'équation (3.3) devient

$$\vec{e}_2 = \cos \theta_2 \cdot \vec{e}_1 + \sin \theta_2 \cdot (\vec{n}_2 \times \vec{e}_1) \quad (3.4)$$

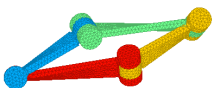
Or, il est facile de calculer \vec{n}_2 à l'aide des équations (3.2) et $\langle \vec{e}_1, \vec{n}_1 \rangle = 0$:

$$\begin{aligned} \vec{n}_2 &= R[\vec{e}_1, \alpha_1](\vec{n}_1) \\ &= \langle \vec{e}_1, \vec{n}_1 \rangle \cdot (1 - \cos \alpha_1) \cdot \vec{e}_1 + \cos \alpha_1 \cdot \vec{n}_1 + \sin \alpha_1 \cdot (\vec{e}_1 \times \vec{n}_1) \\ &= \cos \alpha_1 \cdot \vec{n}_1 + \sin \alpha_1 \cdot (\vec{e}_1 \times \vec{n}_1) \end{aligned} \quad (3.5)$$

Alors, en substituant l'équation (3.5) dans l'équation (3.4), nous obtenons le résultat suivant

$$\begin{aligned} \vec{e}_2 &= \cos \theta_2 \cdot \vec{e}_1 + \sin \theta_2 \cdot (\cos \alpha_1 \cdot \vec{n}_1 + \sin \alpha_1 \cdot (\vec{e}_1 \times \vec{n}_1)) \times \vec{e}_1 \\ &= \cos \theta_2 \cdot \vec{e}_1 + \sin \theta_2 \cdot \cos \alpha_1 \cdot (\vec{n}_1 \times \vec{e}_1) + \sin \theta_2 \cdot \sin \alpha_1 \cdot (\vec{e}_1 \times \vec{n}_1) \times \vec{e}_1 \end{aligned} \quad (3.6)$$

Or $(\vec{e}_1 \times \vec{n}_1) \times \vec{e}_1 = \langle \vec{e}_1, \vec{e}_1 \rangle \cdot \vec{n}_1 - \langle \vec{n}_1, \vec{e}_1 \rangle \cdot \vec{e}_1 = \vec{n}_1$, qui est l'égalité du double produit vectoriel.



Ainsi,

$$\vec{e}_2 = \cos \theta_2 \cdot \vec{e}_1 + \sin \theta_2 \cdot \cos \alpha_1 \cdot (\vec{n}_1 \times \vec{e}_1) + \sin \theta_2 \cdot \sin \alpha_1 \cdot \vec{n}_1$$

Le produit scalaire dans l'équation (3.7) se déduit facilement grâce aux relations (3.8) et (3.9) :

$$\begin{aligned} \langle \vec{n}_1, \vec{e}_2 \rangle &= \cos \theta_2 \cdot \langle \vec{n}_1, \vec{e}_1 \rangle + \sin \theta_2 \cdot \cos \alpha_1 \cdot \langle \vec{n}_1, (\vec{n}_1 \times \vec{e}_1) \rangle + \sin \theta_2 \cdot \sin \alpha_1 \cdot \langle \vec{n}_1, \vec{n}_1 \rangle \\ &= \sin \alpha_1 \cdot \sin \theta_2 \end{aligned} \quad (3.7)$$

$$\langle \vec{n}_1, \vec{e}_1 \rangle = 0 \quad (3.8)$$

$$\langle \vec{n}_1, (\vec{n}_1 \times \vec{e}_1) \rangle = 0 \quad (3.9)$$

3.2.2.2 Expression formelle du produit scalaire entre $\langle \vec{n}_3, \vec{n}_1 \rangle$

Pour calculer ce produit scalaire, d'une part nous savons que :

$$\vec{n}_3 = \cos \alpha_2 \cdot \vec{n}_2 + \sin \alpha_2 \cdot (\vec{e}_2 \times \vec{n}_2) \quad (3.10)$$

Et d'autre part, nous savons que :

$$\begin{cases} \vec{e}_2 = \sin \alpha_1 \cdot \sin \theta_2 \cdot \vec{n}_1 + \cos \theta_2 \cdot \vec{e}_1 + \cos \alpha_1 \cdot \sin \theta_2 \cdot (\vec{n}_1 \times \vec{e}_1) \\ \vec{n}_2 = \cos \alpha_1 \cdot \vec{n}_1 - \sin \alpha_1 \cdot (\vec{n}_1 \times \vec{e}_1) \end{cases} \quad (3.11)$$

Alors, en substituant l'équation (3.11) dans l'équation (3.10), nous obtenons le résultat suivant :

$$\vec{n}_3 = (\cos \alpha_1 \cdot \cos \alpha_2 - \sin \alpha_1 \cdot \cos \theta_2 \cdot \sin \alpha_2) \cdot \vec{n}_1 + \sin \theta_2 \cdot \sin \alpha_2 \cdot \vec{e}_1 - (\sin \alpha_1 \cdot \cos \alpha_2 + \cos \alpha_1 \cdot \cos \theta_2 \cdot \sin \alpha_2) \cdot (\vec{n}_1 \times \vec{e}_1)$$

Le produit scalaire cherché est l'équation (3.13) car

$$\begin{aligned} \langle \vec{n}_1, \vec{n}_3 \rangle &= (\cos \alpha_1 \cdot \cos \alpha_2 - \sin \alpha_1 \cdot \cos \theta_2 \cdot \sin \alpha_2) \langle \vec{n}_1, \vec{n}_1 \rangle + \sin \theta_2 \cdot \sin \alpha_2 \cdot \langle \vec{e}_1, \vec{n}_1 \rangle + \\ &\quad (\sin \alpha_1 \cdot \cos \alpha_2 + \cos \alpha_1 \cdot \cos \theta_2 \cdot \sin \alpha_2) \cdot \langle \vec{n}_1 \times \vec{e}_1, \vec{n}_1 \rangle \end{aligned} \quad (3.12)$$

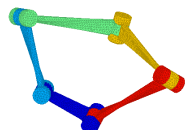
avec $\langle \vec{n}_1, \vec{n}_1 \rangle = 1$, $\langle \vec{e}_1, \vec{n}_1 \rangle = 0$ et $\langle \vec{n}_1 \times \vec{e}_1, \vec{n}_1 \rangle = 0$, l'équation (3.12) devient

$$\langle \vec{n}_1, \vec{n}_3 \rangle = \cos \alpha_1 \cdot \cos \alpha_2 - \sin \alpha_1 \cdot \cos \theta_2 \cdot \sin \alpha_2 \quad (3.13)$$

3.2.3 Cas d'une chaîne fermée de solides

Dans le cas d'une chaîne ouverte, la méthode systématique de calcul du produit scalaire présentée ci-dessus permet d'exprimer formellement chaque terme de la matrice de Gram en fonction d'un ensemble de paramètres angulaires indépendants.

Toutefois, le cas d'une chaîne fermée présente une particularité puisque les paramètres angulaires que l'utilisateur peut spécifier ne forment pas une base de variables indépendantes. Il est donc impératif de trouver les relations entre elles. La méthode la plus classique consiste à utiliser la propriété suivante : le produit de toutes les rotations est égal à une matrice identité. Dans cette



sous-section, nous allons exprimer la matrice de Gram unitaire avec des fonctions F_i . Nous nous intéressons à la composition des rotations pour une chaîne comportant p liaisons monoaxiales et p solides. Dans une boucle fermée, il faut que le vecteur \vec{n}_{p+1} et \vec{n}_1 soient confondus. De même, le vecteur \vec{e}_1 doit être l'image du vecteur \vec{e}_p par la rotation autour du vecteur \vec{n}_1 . Ces deux conditions sont exprimées dans la figure 3.3. La méthode basée sur les fonctions F_i consiste à exprimer cette fermeture en passant d'un vecteur à un autre en suivant deux séquences possibles de rotation.

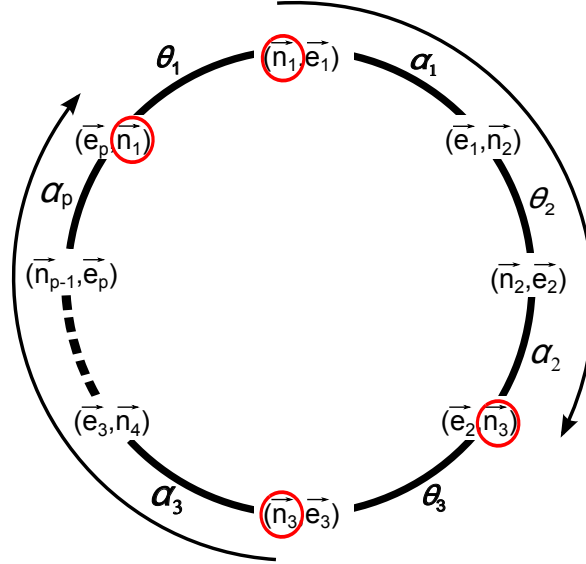


FIGURE 3.3 – mise en évidence de la boucle des paramètres angulaires

En effet, pour passer du vecteur \vec{n}_3 au vecteur \vec{n}_1 , il est possible de suivre les 2 séquences suivantes :

$$\text{Seq. A : } (\vec{n}_1, \vec{e}_1) \xrightarrow{\alpha_1} (\vec{e}_1, \vec{n}_2) \xrightarrow{\theta_2} (\vec{n}_2, \vec{e}_2) \xrightarrow{\alpha_2} (\vec{e}_2, \vec{n}_3) \Rightarrow \langle \vec{n}_1, \vec{n}_3 \rangle = f(\alpha_1, \theta_2, \alpha_2)$$

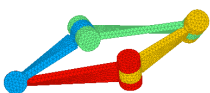
ou

$$\text{Seq. B : } (\vec{n}_3, \vec{e}_3) \xrightarrow{\alpha_3} (\vec{e}_3, \vec{n}_4) \dots \xrightarrow{\alpha_p} (\vec{e}_p, \vec{n}_1) \Rightarrow \langle \vec{n}_3, \vec{n}_1 \rangle = g(\alpha_3, \theta_4, \alpha_4, \dots, \alpha_p)$$

Ces deux séquences permettent d'écrire deux expressions différentes d'un même produit scalaire. Puisque ceci est vrai quelque soit la paire de vecteurs choisis, ce résultat est synthétisé dans la représentation formelle suivante de la matrice de Gram $G(N)$, où la première expression d'un produit scalaire apparaît dans le triangle supérieur droit tandis que la seconde apparaît dans le triangle inférieur gauche.

	\vec{n}_1	\vec{e}_1	\vec{n}_2	\vec{e}_2	...	\vec{n}_p	\vec{e}_p
\vec{n}_1	1	0	$F_1(\alpha_1)$	$F_2(\alpha_1, \theta_2)$...	$F_{2,p-3}(\alpha_1, \dots)$	$F_{2,p-2}(\alpha_1, \dots)$
\vec{e}_1	$F_{2,p-2}(\theta_2, \dots)$	1	0	$F_1(\theta_2)$...	$F_{2,p-4}(\theta_2, \dots)$	$F_{2,p-3}(\theta_2, \dots)$
\vec{n}_2	$F_{2,p-3}(\alpha_2, \dots)$	$F_{2,p-2}(\alpha_2, \dots)$	1	0	...	$F_{2,p-5}(\alpha_2, \dots)$	$F_{2,p-4}(\alpha_2, \dots)$
\vec{e}_2	$F_{2,p-4}(\theta_3, \dots)$	$F_{2,p-3}(\theta_3, \dots)$	$F_{2,p-2}(\theta_3, \dots)$	1	...	$F_{2,p-6}(\theta_3, \dots)$	$F_{2,p-5}(\theta_3, \dots)$
...
\vec{e}_{p-1}	$F_2(\theta_p, \alpha_1)$	$F_3(\theta_p, \alpha_1, \theta_2)$	$F_4(\theta_p, \alpha_1, \theta_2, \alpha_2)$	$F_5(\theta_p, \alpha_1, \theta_2, \alpha_2, \theta_3)$...	0	$F_1(\theta_p)$
\vec{n}_p	$F_1(\alpha_p)$	$F_2(\alpha_p, \theta_p)$	$F_3(\alpha_p, \theta_p, \alpha_1)$	$F_4(\alpha_p, \theta_p, \alpha_1, \theta_2)$...	1	0
\vec{e}_p	0	$F_1(\theta_1)$	$F_2(\theta_1, \alpha_1)$	$F_3(\theta_1, \alpha_1, \theta_2)$...	$F_{2,p-2}(\theta_1, \dots)$	1

(3.14)



Les fonctions $F_{i(i=1\dots 2p-2)}$ seront utilisées pour exprimer le produit scalaire entre deux vecteurs de la chaîne représentée sur la figure 3.3. Le long de cette chaîne deux vecteurs adjacents (représentés sur un même nœud) sont orthogonaux. Par exemple, θ est l'angle entre \vec{e}_1 et \vec{e}_2 autour de \vec{n}_2 ; α est l'angle entre \vec{n}_2 et \vec{n}_3 autour de \vec{e}_2 . La génération de ces fonctions a été automatisée dans Maple[®].

3.3 Mise en équation d'un système de solides

3.3.1 Méthode basée sur les fonctions F_i

Les contraintes géométriques imposées par le concepteur induisent un ensemble d'équations. Selon le type de spécifications, les équations se classent en deux catégories : *équations de fermeture angulaire* et *équations vectorielles* (ou *équations affines*).

Les équations angulaires sont obtenues à partir des fonctions F_i tandis que les équations vectorielles font intervenir les longueurs. Les longueurs des vecteurs sont stockées dans la matrice diagonale L comme dans l'équation (1.13).

3.3.1.1 Équations angulaires

Dans la matrice de Gram unitaire (Cf. équation 3.14) exprimée avec les fonctions F_i , les longueurs sont absentes. Donc, pour un mécanisme en boucle fermée de p vecteurs, il y a $2p$ paramètres angulaires dont p angles dimensionnels α et p angles de mobilité θ .

On constate que la matrice de Gram remplie avec les fonctions F_i n'est pas symétrique. La commutativité du produit scalaire assure que la matrice de Gram est symétrique. Par conséquence, pour assurer la fermeture angulaire de la boucle vectorielle, il faudra écrire l'égalité des termes diagonalement opposés de la matrice présentée dans l'équation (3.14). Cette opération s'écrit ainsi :

$$g_{ij}(N) = g_{ji}(N)$$

Pour une boucle de p vecteurs, le nombre d'équations angulaires est :

$$Nb_{ang} = \frac{p \cdot (p - 1)}{2}$$

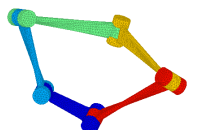
Ainsi, nous écrivons les différentes relations diagonale par diagonale.

Sur la 2^{ème} diagonale, il y a $2p - 1$ relations :

$$\begin{cases} F_{2(p-2)}(\theta_2, \dots) = 0 \\ F_{2(p-3)}(\alpha_2, \dots) = F_1(\alpha_1) \\ \vdots \end{cases}$$

Sur la 3^{ème} diagonale, il y a $2p - 2$ relations :

$$\begin{cases} F_{2(p-2)}(\alpha_2, \dots) = 0 \\ F_{2(p-3)}(\theta_3, \dots) = F_1(\alpha_1) \\ \vdots \end{cases}$$



A noter que l'espace euclidien de travail est de dimension 3 et le rang de la matrice de Gram unitaire est égal à 3. De ce fait, il suffit d'écrire seulement trois égalités des termes diagonalement opposés. Ceci signifie que les $\frac{p \cdot (p-1)}{2}$ équations sont forcément dépendantes entre elles. Grâce à l'automatisation de l'écriture des équations angulaires, nous savons générer toutes ces équations angulaires à partir d'une matrice de Gram issue d'une chaîne fermée. Cependant, il n'est pas facile de traiter automatiquement toutes ces équations en même temps dans la phase d'analyse et de résolution symbolique et manuelle. Au §3.4.4, nous avons fait un choix parmi les $p - 1$ équations angulaires pour une résolution symbolique manuelle. En raison de l'intervention des paramètres de mobilité θ_i que nous voulons éliminer, le choix des "bonnes"¹ équations est très difficile.

3.3.1.2 Équation vectorielle

Pour assurer l'assemblage du mécanisme, non seulement la fermeture angulaire doit être vérifiée mais aussi la fermeture vectorielle. Ces dernières équations représentent le fait que certains vecteurs forment des boucles lorsque le concepteur définit des contraintes de coïncidence. Bien évidemment ces dernières doivent être respectées. Deux méthodes sont présentées dans § 1.3.2.2 et §1.3.2.3 ; une est basée sur le concept de coupe métrique [Riv06] tandis que l'autre utilise la méthode des projections.

3.3.2 Méthode de Jin et Yang

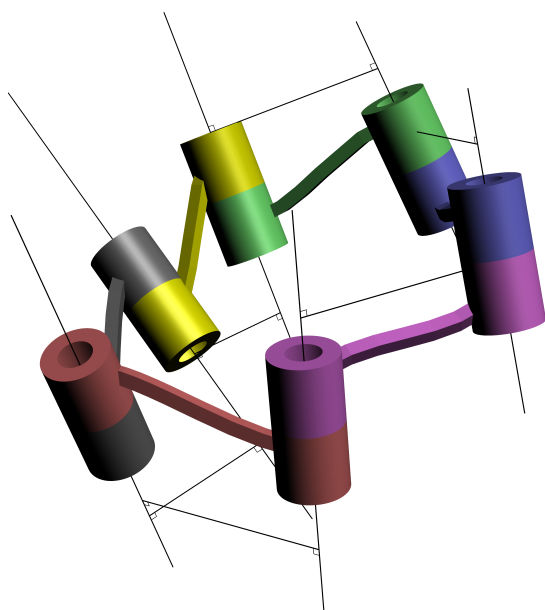


FIGURE 3.4 – Le système de 6 solides et 6 liaisons

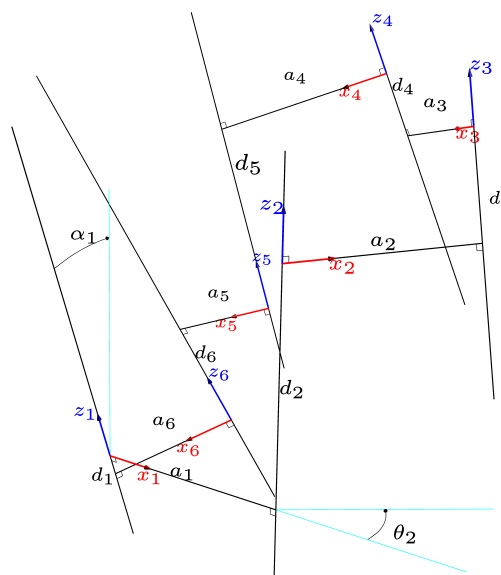
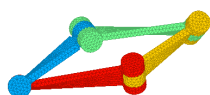


FIGURE 3.5 – Le squelette 3D associé à la figure 3.4 avec la notation D-H

Dans la 1^{ère} méthode présentée, le nombre d'équations varie en fonction de la taille du squelette. Plus il y a de vecteurs dans le modèle, plus le nombre d'équations du système est élevé.

1. Nous souhaitons prendre des équations indépendantes avec le moins possible de paramètres à éliminer qui contiennent toute les informations du système étudié.



Jin *et al.* [Yan02] propose une autre technique pour construire un système algébrique indépendant.

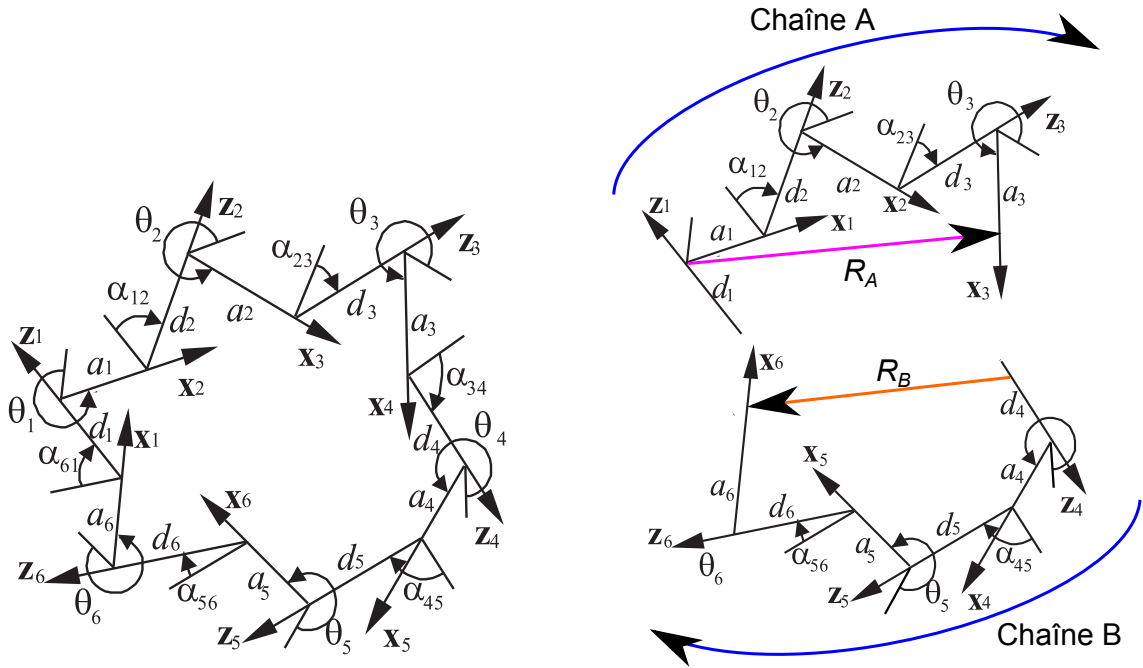


FIGURE 3.6 – Le squelette associé au système de 6 solides et 6 liaisons [Yan02]

FIGURE 3.7 – La coupure sur le squelette de la figure 3.6

Dans leur article [Yan02], ils ont appliqué la méthode sur un mécanisme fermé de 6 liaisons en une seule boucle (voir la figure 3.4). Le squelette avec la notation D-H associé au système étudié est montré dans la figure 3.5. Les directions des axes et des pièces sont représentées respectivement par les vecteurs unitaires \vec{z}_i et \vec{x}_i . Les paramètres d'usinage et ceux de mobilité sont définis ainsi : d_i représente la translation le long de l'axe \vec{z}_i , a_i est la distance le long de \vec{x}_{i+1} , α_{ij} est l'angle de torsion entre l'axe \vec{z}_i et l'axe \vec{z}_{i+1} et θ_i est l'angle de rotation par rapport à l'axe \vec{z}_i . La boucle peut être coupée en deux chaînes ouvertes, d'où viennent un ensemble de 4 équations en fonction de 4 variables de mobilité et de 12 paramètres d'usinage. Ces 4 équations possèdent un nombre fini de solutions.

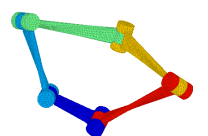
Par exemple, en coupant le squelette aux liaisons \vec{z}_1 et \vec{z}_4 , apparaissent alors deux chaînes ouvertes A et B. La chaîne A suit le chemin $(\vec{z}_1, \vec{x}_1) \rightarrow (\vec{z}_2, \vec{x}_2) \rightarrow (\vec{z}_3, \vec{x}_3) \rightarrow \vec{z}_4$. De la même manière, la chaîne B suit le chemin $(\vec{z}_4, \vec{x}_4) \rightarrow (\vec{z}_5, \vec{x}_5) \rightarrow (\vec{z}_6, \vec{x}_6) \rightarrow \vec{z}_1$. Les vecteurs qui lient les extrémités de deux chaînes sont notés \vec{R}_A et \vec{R}_B . Ces notations se trouvent également dans la publication [Yan02]. Les 4 équations de Jin et Yang s'écrivent alors :

$$\langle \vec{z}_1, \vec{z}_4 \rangle_A = \langle \vec{z}_1, \vec{z}_4 \rangle_B \tag{3.15}$$

$$\langle \vec{R}_A, \vec{z}_1 \times \vec{z}_4 \rangle_A = \langle \vec{R}_B, \vec{z}_1 \times \vec{z}_4 \rangle_B \tag{3.16}$$

$$\langle \vec{R}_A, \vec{z}_1 \rangle_A = \langle \vec{R}_B, \vec{z}_1 \rangle_B \tag{3.17}$$

$$\langle \vec{R}_A, \vec{z}_4 \rangle_A = \langle \vec{R}_B, \vec{z}_4 \rangle_B \tag{3.18}$$



avec :

$$\begin{aligned}\vec{R}_A &= a_1\vec{x}_1 + d_2\vec{z}_2 + a_2\vec{x}_2 + d_3\vec{z}_3 + a_3\vec{x}_3 \\ \vec{R}_B &= -(d_4\vec{z}_4 + a_4\vec{x}_4 + d_5\vec{z}_5 + a_5\vec{x}_5 + d_6\vec{z}_6 + a_6\vec{x}_6)\end{aligned}$$

L'équation (3.15) est une équation angulaire. Il s'agit du produit scalaire entre les vecteurs \vec{z}_1 et \vec{z}_4 . Les équations (3.16), (3.17) et (3.18) peuvent être considérées comme les équations de fermeture vectorielle. Plus concrètement, l'équation (3.16) est la projection de \vec{R}_A (resp. \vec{R}_B) sur $\vec{z}_1 \times \vec{z}_4$; l'équation (3.17) est la projection de \vec{R}_A (resp. \vec{R}_B) sur \vec{z}_1 ; l'équation (3.18) est la projection de \vec{R}_A (resp. \vec{R}_B) sur \vec{z}_4 . Le principe de cette modélisation peut être résumé par l'équation suivante :

$$\left(\begin{array}{c|ccc} & \vec{z}_1 & \vec{z}_4 & \vec{z}_1 \times \vec{z}_4 & \vec{R}_A \\ \hline \vec{z}_1 & 1 & \cos \gamma_A & 0 & M_A \\ \vec{z}_4 & \cos \gamma_A & 1 & 0 & N_A \\ \vec{z}_1 \times \vec{z}_4 & 0 & 0 & 1 - \cos^2 \gamma_A & P_A \end{array} \right) \equiv \left(\begin{array}{c|ccc} & \vec{z}_1 & \vec{z}_4 & \vec{z}_1 \times \vec{z}_4 & \vec{R}_B \\ \hline \vec{z}_1 & 1 & \cos \gamma_B & 0 & M_B \\ \vec{z}_4 & \cos \gamma_B & 1 & 0 & N_B \\ \vec{z}_1 \times \vec{z}_4 & 0 & 0 & 1 - \cos^2 \gamma_B & P_B \end{array} \right) \quad (3.19)$$

Le squelette est coupé en deux chaînes ouvertes. Les deux chaînes ouvertes peuvent être représentées par les deux matrices de l'équation (3.19), respectivement. Pour assembler les deux chaînes, il faut que les deux matrices soient égales. L'idée de Jin et Yang est d'écrire l'égalité entre les éléments de deux matrices.

Géométriquement, l'équation (3.15) traduit le fait que quelque soit le chemin suivi (séquence A ou séquence B), l'angle entre \vec{z}_1 et \vec{z}_4 doit être identique. Ceci explique également la symétrie du tableau 3.19. Nous avons donc $\cos \gamma_A = \cos \gamma_B$. Notez que les 1 sur la diagonale indiquent que les vecteurs \vec{z}_1 et \vec{z}_4 sont unitaires.

Le produit vectoriel $\vec{z}_1 \times \vec{z}_4$ définit le vecteur orthogonal aux deux vecteurs donnés. Quand \vec{z}_1 et \vec{z}_4 ne sont pas colinéaires, la norme du produit vectoriel s'écrit :

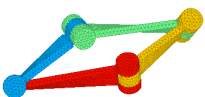
$$\|\vec{z}_1 \times \vec{z}_4\| = \|\vec{z}_1\| \cdot \|\vec{z}_4\| \cdot |\sin(\vec{z}_1, \vec{z}_4)| \quad (3.20)$$

avec $\vec{z}_1 \times \vec{z}_4$ orthogonal aux vecteurs \vec{z}_1 et \vec{z}_4 . Pour cette raison, nous trouvons le terme $\cos^2 \gamma = 1 - \sin^2(\vec{z}_1, \vec{z}_4)$ dans la 3^{ème} case de la diagonale.

Une autre raison pour calculer le produit vectoriel est le fait de faire la projection. En effet, l'équation (3.16) est une projection du vecteur \vec{R}_A (resp. \vec{R}_B) sur le vecteur orthogonal de \vec{z}_1 et \vec{z}_4 , et se trouve dans la case P_A (resp. P_B). De la même manière, les projections de \vec{R}_A (resp. \vec{R}_B) sur \vec{z}_1 et \vec{z}_4 correspondent respectivement aux cases M_A (resp. M_B) et N_A (resp. N_B). Les produits scalaires $\langle \vec{z}_1, \vec{z}_1 \times \vec{z}_4 \rangle$ et $\langle \vec{z}_4, \vec{z}_1 \times \vec{z}_4 \rangle$ sont nuls car $\vec{z}_1 \perp (\vec{z}_1 \times \vec{z}_4)$ et $\vec{z}_4 \perp (\vec{z}_1 \times \vec{z}_4)$.

Finalement, nous avons pu remplir les 3 premières lignes du tableau 3.19 et écrire les 4 équations indépendantes grâce à la symétrie du tableau.

La limite de cette méthode apparaît lorsque les deux vecteurs \vec{z}_1 et \vec{z}_4 sont colinéaires car le produit vectoriel $\vec{z}_1 \times \vec{z}_4$ est nul. Par conséquent, les 4 équations deviennent dépendantes. Pour éviter cela, il faut toujours vérifier que les deux vecteurs choisis ne sont pas colinéaires. À l'heure actuelle, nous avons automatisé la génération de la matrice de Gram et l'écriture des



fonctions F_i . Puisque le produit mixte n'apparaît pas dans la matrice de Gram, il n'est pas encore automatisé.

3.3.3 Variante de la méthode de Jin et Yang

S'inspirant de l'idée de Jin et Yang, nous avons remplacé l'équation (3.16) par l'équation (3.22) afin d'éviter de calculer le produit mixte. Les trois autres équations resteront inchangées. Pour l'exemple de la figure 3.4 coupée sur \vec{z}_1 et \vec{z}_4 , les équations (3.15), (3.16), (3.17) et (3.18) deviennent :

$$\langle \vec{z}_1, \vec{z}_4 \rangle_A = \langle \vec{z}_1, \vec{z}_4 \rangle_B \quad (3.21)$$

$$\langle \vec{R}_A, \vec{R}_A \rangle = \langle \vec{R}_B, \vec{R}_B \rangle \quad (3.22)$$

$$\langle \vec{R}_A, \vec{z}_1 \rangle_A = \langle \vec{R}_B, \vec{z}_1 \rangle_B \quad (3.23)$$

$$\langle \vec{R}_A, \vec{z}_4 \rangle_A = \langle \vec{R}_B, \vec{z}_4 \rangle_B \quad (3.24)$$

avec :

$$\begin{aligned} \vec{R}_A &= a_1 \vec{x}_1 + d_2 \vec{z}_2 + a_2 \vec{x}_2 + d_3 \vec{z}_3 + a_3 \vec{x}_3 \\ \vec{R}_B &= -(d_4 \vec{z}_4 + a_4 \vec{x}_4 + d_5 \vec{z}_5 + a_5 \vec{x}_5 + d_6 \vec{z}_6 + a_6 \vec{x}_6) \end{aligned}$$

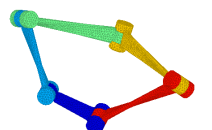
L'intérêt de cette proposition est d'éviter de calculer le produit vectoriel $\vec{z}_1 \times \vec{z}_4$ sans perdre d'information sur le système.

$$\left(\begin{array}{c|ccc} & \vec{z}_1 & \vec{z}_4 & \vec{R}_A \\ \hline \vec{z}_1 & 1 & \cos \gamma_A & M_A \\ \vec{z}_4 & \cos \gamma_A & 1 & N_A \\ \vec{R}_A & M_A & N_A & R_A^2 \end{array} \right) \equiv \left(\begin{array}{c|ccc} & \vec{z}_1 & \vec{z}_4 & \vec{R}_B \\ \hline \vec{z}_1 & 1 & \cos \gamma_B & M_B \\ \vec{z}_4 & \cos \gamma_B & 1 & N_B \\ \vec{R}_B & M_B & N_B & R_B^2 \end{array} \right) \quad (3.25)$$

En effectuant la même coupure que Jin et Yang, nous décrivons les deux chaînes ouvertes par deux matrices de Gram. Les cases bleues, vertes et jaunes correspondent aux équations (3.21), (3.23) et (3.24). Nous constatons qu'il nous manque une seule case pour remplir le noyau de la matrice. Ce sont les termes R_A^2 et R_B^2 dans la case rouge, d'où vient l'équation (3.22), qui est la longueur de \vec{R}_A et \vec{R}_B au carré. Finalement, nous avons écrit 4 équations indépendantes sans utiliser le produit vectoriel. Cette méthode nous convient mieux au niveau de l'automatisation de la mise en équation.

Le même inconvénient que la méthode de Jin et Yang apparaît. C'est la colinéarité de deux vecteurs donnés qui rend les équations dépendantes, et la matrice devient alors de rang inférieur à 3. Il faut donc éviter de choisir des vecteurs colinéaires pour la projection.

Grâce à ces deux dernières méthodes, on écrit seulement 4 équations au lieu de 6 ou plus. Lorsque les 2 vecteurs de la coupure ne sont pas colinéaires, ces 4 équations sont indépendantes et contiennent toutes les informations géométriques nécessaires et suffisantes du système.



3.4 Application au système de solides 4R

Dans cette section, nous appliquons les 3 méthodes présentées sur un système de solides en 3D. Ces 3 méthodes sont : méthode des fonctions F_i , méthode de Jin et Yang et la variante de la méthode de Jin et Yang.

3.4.1 Méthode basée sur les fonctions F_i

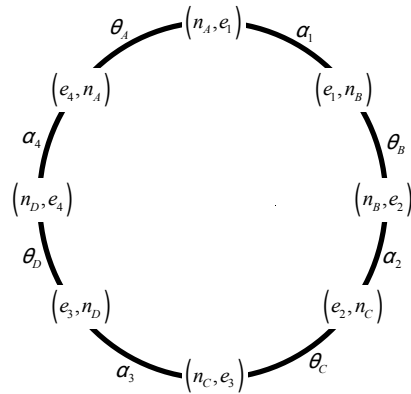
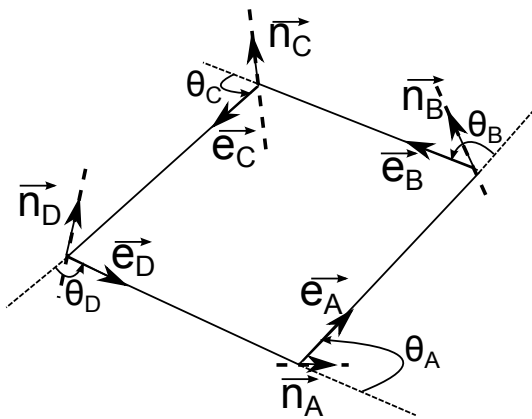


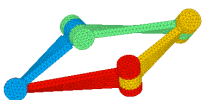
FIGURE 3.8 – Squelette d'un 4R représenté par les vecteurs unitaires et les segments de droites

FIGURE 3.9 – La boucle des paramètres angulaires pour le 4R

La figure 3.8 est le squelette d'un 4R. L'axe et l'orientation des liaisons pivots sont représentés par les vecteurs unitaires \vec{n}_i . De la même manière, la direction et l'orientation des pièces sont représentées par les vecteurs unitaires \vec{e}_i . Les expressions sont données dans le tableau 3.1. Il faut retenir que la différence entre le squelette de 4R et un squelette quelconque est que l'axe du pivot de 4R est perpendiculaire aux deux pièces qui les lie. La figure 3.9 montre la boucle de passage en paramètres angulaires pour un 4R. Les relations entre la figure 3.8 et la figure 3.9 sont dans le tableau 3.3.

Angle de mobilité (liaison)	Angle dimensionnel (solide)	Longueur dimensionnelle (solide)
$\theta_A = \text{angle}(\vec{e}_A, \vec{e}_B)$	$\alpha_1 = \text{angle}(\vec{n}_A, \vec{n}_B)$	$L_1 = \text{longueur}(\vec{n}_A, \vec{n}_B)$
$\theta_B = \text{angle}(\vec{e}_B, \vec{e}_C)$	$\alpha_2 = \text{angle}(\vec{n}_B, \vec{n}_C)$	$L_2 = \text{longueur}(\vec{n}_B, \vec{n}_C)$
$\theta_C = \text{angle}(\vec{e}_C, \vec{e}_D)$	$\alpha_3 = \text{angle}(\vec{n}_C, \vec{n}_D)$	$L_3 = \text{longueur}(\vec{n}_C, \vec{n}_D)$
$\theta_D = \text{angle}(\vec{e}_D, \vec{e}_A)$	$\alpha_4 = \text{angle}(\vec{n}_D, \vec{n}_A)$	$L_4 = \text{longueur}(\vec{n}_D, \vec{n}_A)$

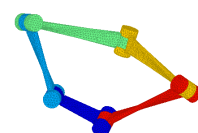
TABLE 3.1 – Paramètres dimensionnels et de mobilité décrivant le squelette de Bennett



Le tenseur de 4R s'écrit alors :

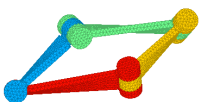
$$G(N) = \begin{pmatrix} 1 & \langle \vec{n}_A, \vec{e}_1 \rangle & \langle \vec{n}_A, \vec{n}_B \rangle & \langle \vec{n}_A, \vec{e}_2 \rangle & \langle \vec{n}_A, \vec{n}_C \rangle & \langle \vec{n}_A, \vec{e}_3 \rangle & \langle \vec{n}_A, \vec{n}_D \rangle & \langle \vec{n}_A, \vec{e}_4 \rangle \\ \langle \vec{e}_1, \vec{n}_A \rangle & 1 & \langle \vec{e}_1, \vec{n}_B \rangle & \langle \vec{e}_1, \vec{e}_2 \rangle & \langle \vec{e}_1, \vec{n}_C \rangle & \langle \vec{e}_1, \vec{e}_3 \rangle & \langle \vec{e}_1, \vec{n}_D \rangle & \langle \vec{e}_1, \vec{e}_4 \rangle \\ \langle \vec{n}_B, \vec{n}_A \rangle & \langle \vec{n}_B, \vec{e}_1 \rangle & 1 & \langle \vec{n}_B, \vec{e}_2 \rangle & \langle \vec{n}_B, \vec{n}_C \rangle & \langle \vec{n}_B, \vec{e}_3 \rangle & \langle \vec{n}_B, \vec{n}_D \rangle & \langle \vec{n}_B, \vec{e}_4 \rangle \\ \langle \vec{e}_2, \vec{n}_A \rangle & \langle \vec{e}_2, \vec{e}_1 \rangle & \langle \vec{e}_2, \vec{n}_B \rangle & 1 & \langle \vec{e}_2, \vec{n}_C \rangle & \langle \vec{e}_2, \vec{e}_3 \rangle & \langle \vec{e}_2, \vec{n}_D \rangle & \langle \vec{e}_2, \vec{e}_4 \rangle \\ \langle \vec{n}_C, \vec{n}_A \rangle & \langle \vec{n}_C, \vec{e}_1 \rangle & \langle \vec{n}_C, \vec{n}_B \rangle & \langle \vec{n}_C, \vec{e}_2 \rangle & 1 & \langle \vec{n}_C, \vec{e}_3 \rangle & \langle \vec{n}_C, \vec{n}_D \rangle & \langle \vec{n}_C, \vec{e}_4 \rangle \\ \langle \vec{e}_3, \vec{n}_A \rangle & \langle \vec{e}_3, \vec{e}_1 \rangle & \langle \vec{e}_3, \vec{n}_B \rangle & \langle \vec{e}_3, \vec{e}_2 \rangle & \langle \vec{e}_3, \vec{n}_C \rangle & 1 & \langle \vec{e}_3, \vec{n}_D \rangle & \langle \vec{e}_3, \vec{e}_4 \rangle \\ \langle \vec{n}_D, \vec{n}_A \rangle & \langle \vec{n}_D, \vec{e}_1 \rangle & \langle \vec{n}_D, \vec{n}_B \rangle & \langle \vec{n}_D, \vec{e}_2 \rangle & \langle \vec{n}_D, \vec{n}_C \rangle & \langle \vec{n}_D, \vec{e}_3 \rangle & 1 & \langle \vec{n}_D, \vec{e}_4 \rangle \\ \langle \vec{e}_4, \vec{n}_A \rangle & \langle \vec{e}_4, \vec{e}_1 \rangle & \langle \vec{e}_4, \vec{n}_B \rangle & \langle \vec{e}_4, \vec{e}_2 \rangle & \langle \vec{e}_4, \vec{n}_C \rangle & \langle \vec{e}_4, \vec{e}_3 \rangle & \langle \vec{e}_4, \vec{n}_D \rangle & 1 \end{pmatrix} \quad (3.26)$$

Il est détaillé dans le tableau 3.2 en utilisant les formules d'Olinde-Rodrigues.



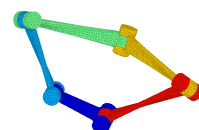
	\vec{n}_A	\vec{e}_1^A	\vec{n}_B	\vec{e}_2^B	\vec{n}_C	\vec{e}_3^C	\vec{n}_D	\vec{e}_4^D
\vec{n}_A	1	0	$F_1(\alpha_1)$	$F_2(\alpha_1, \theta_B)$	$F_3(\alpha_1, \theta_B, \alpha_2)$	$F_4 \left(\begin{matrix} \alpha_1, \theta_B \\ \alpha_2, \theta_C \end{matrix} \right)$	$F_5 \left(\begin{matrix} \alpha_1, \theta_B, \alpha_2 \\ \theta_C, \alpha_3 \end{matrix} \right)$	$F_6 \left(\begin{matrix} \alpha_1, \theta_B, \alpha_2 \\ \theta_C, \alpha_3, \theta_D \end{matrix} \right)$
\vec{e}_1^A	$F_6 \left(\begin{matrix} \theta_B, \alpha_2, \theta_C \\ \alpha_3, \theta_D, \alpha_4 \end{matrix} \right)$	1	0	$F_1(\theta_B)$	$F_2(\theta_B, \alpha_2)$	$F_3(\theta_B, \alpha_2, \theta_C)$	$F_4 \left(\begin{matrix} \theta_B, \alpha_2 \\ \theta_C, \alpha_3 \end{matrix} \right)$	$F_5 \left(\begin{matrix} \theta_B, \alpha_2, \theta_C \\ \alpha_3, \theta_D \end{matrix} \right)$
\vec{n}_B	$F_5 \left(\begin{matrix} \alpha_2, \theta_C, \alpha_3 \\ \theta_D, \alpha_4, \theta_B \end{matrix} \right)$	$F_6 \left(\begin{matrix} \alpha_2, \theta_C, \alpha_3 \\ \theta_D, \alpha_4, \theta_B \end{matrix} \right)$	1	0	$F_1(\alpha_2)$	$F_2(\alpha_2, \theta_C)$	$F_3(\alpha_2, \theta_C, \alpha_3)$	$F_4 \left(\begin{matrix} \alpha_2, \theta_C \\ \alpha_3, \theta_D \end{matrix} \right)$
\vec{e}_2^B	$F_5 \left(\begin{matrix} \theta_C, \alpha_3, \theta_D \\ \alpha_4, \theta_A \end{matrix} \right)$	$F_6 \left(\begin{matrix} \theta_C, \alpha_3, \theta_D \\ \alpha_4, \theta_A \end{matrix} \right)$	$F_6 \left(\begin{matrix} \theta_C, \alpha_3, \theta_D \\ \alpha_4, \theta_A, \alpha_1 \end{matrix} \right)$	1	0	$F_1(\theta_C)$	$F_2(\theta_C, \alpha_3)$	$F_3(\theta_C, \alpha_3, \theta_D)$
\vec{n}_C	$F_4 \left(\begin{matrix} \alpha_3, \theta_D, \alpha_4 \\ \theta_D, \alpha_4 \end{matrix} \right)$	$F_5 \left(\begin{matrix} \alpha_3, \theta_D, \alpha_4 \\ \theta_D, \alpha_4 \end{matrix} \right)$	$F_6 \left(\begin{matrix} \theta_D, \alpha_4, \theta_A \\ \alpha_1, \theta_B \end{matrix} \right)$	$F_5 \left(\begin{matrix} \theta_D, \alpha_4, \theta_A \\ \alpha_1, \theta_B \end{matrix} \right)$	1	0	$F_1(\alpha_3)$	$F_2(\alpha_3, \theta_D)$
\vec{e}_3^C	$F_2(\theta_D, \alpha_4)$	$F_3(\theta_D, \alpha_4, \theta_A)$	$F_4 \left(\begin{matrix} \theta_D, \alpha_4 \\ \theta_A, \alpha_1 \end{matrix} \right)$	$F_5 \left(\begin{matrix} \theta_D, \alpha_4, \theta_A \\ \alpha_1, \theta_B \end{matrix} \right)$	$F_6 \left(\begin{matrix} \theta_D, \alpha_4, \theta_A \\ \alpha_1, \theta_B, \alpha_2 \end{matrix} \right)$	1	0	$F_1(\theta_D)$
\vec{n}_D	$F_1(\alpha_4)$	$F_2(\alpha_4, \theta_A)$	$F_3(\alpha_4, \theta_A, \alpha_1)$	$F_4 \left(\begin{matrix} \alpha_4, \theta_A \\ \alpha_1, \theta_B \end{matrix} \right)$	$F_5 \left(\begin{matrix} \alpha_4, \theta_A, \alpha_1 \\ \theta_B, \alpha_2 \end{matrix} \right)$	$F_6 \left(\begin{matrix} \alpha_4, \theta_A, \alpha_1 \\ \theta_B, \alpha_2, \theta_C \end{matrix} \right)$	1	0
\vec{e}_4^D	0	$F_1(\theta_A)$	$F_2(\theta_A, \alpha_1)$	$F_3(\theta_A, \alpha_1, \theta_B)$	$F_4 \left(\begin{matrix} \theta_A, \alpha_1 \\ \theta_B, \alpha_2 \end{matrix} \right)$	$F_5 \left(\begin{matrix} \theta_A, \alpha_1, \theta_B \\ \alpha_2, \theta_C \end{matrix} \right)$	$F_6 \left(\begin{matrix} \theta_A, \alpha_1, \theta_B \\ \alpha_2, \theta_C, \alpha_3 \end{matrix} \right)$	1

TABLE 3.2 – Matrice de Gram unitaire d'un 4R en fonction de la formule d'Olinde-Rodrigues



Les expressions de F_1 à F_6 sont indiquées dans l'équation (3.27) [Riv02] :

$$\left\{ \begin{array}{l}
 F_1(\alpha_1) = \cos \alpha_1 \\
 F_2(\alpha_1, \theta_B) = \sin \alpha_1 \cdot \sin \theta_B \\
 F_3(\alpha_1, \theta_B, \alpha_2) = \cos \alpha_1 \cdot \cos \alpha_2 - \sin \alpha_1 \cdot \sin \alpha_2 \cdot \cos \theta_B \\
 F_4(\alpha_1, \theta_B, \alpha_2, \theta_C) = \sin \theta_C \cdot (\cos \alpha_1 \cdot \sin \alpha_2 + \cos \alpha_2 \cdot \sin \alpha_1 \cdot \cos \theta_B) \\
 \quad + \sin \alpha_1 \cdot \cos \theta_C \cdot \sin \theta_B \\
 F_5(\alpha_1, \theta_B, \alpha_2, \theta_C, \alpha_3) = \cos \alpha_3 \cdot (\cos \alpha_1 \cdot \cos \alpha_2 - \sin \alpha_1 \cdot \sin \alpha_2 \cdot \cos \theta_B) \\
 \quad - \sin \alpha_3 \cdot (\cos \alpha_1 \cdot \sin \alpha_2 \cdot \cos \theta_C - \sin \alpha_1 \cdot \sin \theta_B \cdot \sin \theta_C) \\
 \quad + \cos \alpha_2 \cdot \sin \alpha_1 \cdot \cos \theta_B \cdot \cos \theta_C \\
 F_6(\alpha_1, \theta_B, \alpha_2, \theta_C, \alpha_3, \theta_D) = \sin \alpha_4 \cdot (\cos \alpha_1 \cdot \cos \alpha_B \cdot \sin \alpha_3 + \cos \alpha_1 \cdot \cos \alpha_3 \cdot \sin \alpha_2 \cdot \cos \theta_C) \\
 \quad - \sin \alpha_1 \cdot \sin \alpha_2 \cdot \sin \alpha_3 \cdot \cos \theta_B - \cos \alpha_3 \cdot \sin \alpha_1 \cdot \sin \theta_B \cdot \sin \theta_C \\
 \quad + \cos \alpha_2 \cdot \cos \alpha_3 \cdot \sin \alpha_1 \cdot \cos \theta_B \cdot \cos \theta_C) \\
 \quad + \cos \theta_D \cdot (\sin \theta_C \cdot (\cos \alpha_1 \cdot \sin \alpha_2 + \cos \alpha_2 \cdot \sin \alpha_1 \cdot \cos \theta_B) \\
 \quad + \sin \alpha_1 \cdot \cos \theta_C \cdot \cos \theta_B)
 \end{array} \right. \quad (3.27)$$



3.4.1.1 Fermeture angulaire

Les équations de fermeture angulaire d'un 4R sont basées sur la symétrie de la matrice présentée dans le tableau 3.4.1. Dans ce cas, nous obtenons 28 équations angulaires au total, qui sont dépendantes entre elles. En raison de leur taille, nous n'allons pas les énumérer ici. Du fait que le rang de la matrice de Gram soit égale à 3 (la dimension du modèle), il suffit d'écrire 3 équations indépendantes qui contiennent les informations complètes de la fermeture angulaire.

Nous constatons que les produits scalaires entre $\langle \vec{e}_i, \vec{e}_j \rangle$ dépendent au maximum de trois angles de liaisons θ_i (angles de mobilité). Il nous faut considérer aussi le choix parmi les 28 équations. Un exemple d'équations angulaires est donné avec (3.28), (3.29) et (3.30).

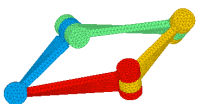
$$F_1(\alpha_3) = F_5(\alpha_4, \theta_A, \alpha_1, \theta_B, \alpha_2) \quad (3.28)$$

$$F_3(\alpha_2, \theta_C, \alpha_3) = F_3(\alpha_4, \theta_A, \alpha_1) \quad (3.29)$$

$$F_3(\alpha_1, \theta_B, \alpha_2) = F_3(\alpha_3, \theta_D, \alpha_4) \quad (3.30)$$

3.4.1.1.1 Coupe métrique

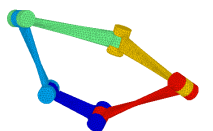
En plus de la fermeture angulaire, il faut aussi écrire les équations vectorielles en fonction des longueurs. Avant d'établir ces relations, il est nécessaire de rendre symétrique la matrice de Gram présentée dans le tableau 3.2 pour simplifier l'écriture des équations. Voyons le tableau 3.3, chaque produit scalaire exprimé par la fonction F_6 est remplacé par un 0 ; chaque fonction F_5 est remplacé par F_1 ; enfin chaque fonction F_4 est remplacé par F_2 . Nous constatons que malgré ces simplifications, la matrice n'est toutefois pas encore symétrique. Les paramètres d'entrée des fonctions F_3 en haut et en bas sur les quatrièmes diagonales sont différents. Il existe donc deux possibilités afin d'atteindre la symétrie : conserver une des deux, celle du haut ou celle du bas. Le choix de l'une ou l'autre n'a aucune importance. Dans le cas du tableau 3.3, la 4^{ème} diagonale du haut a été conservée.



	\vec{n}_A	\vec{e}_1	\vec{n}_B	\vec{e}_2	\vec{n}_C	\vec{e}_3	\vec{n}_D	\vec{e}_4
\vec{n}_A	1	0	$F_1(\alpha_1)$	$F_2(\alpha_1, \theta_B)$	$F_3(\alpha_1, \theta_B, \alpha_2)$	$F_2(\theta_D, \alpha_4)$	$F_1(\alpha_4)$	0
\vec{e}_1	0	1	0	$F_1(\theta_B)$	$F_2(\theta_B, \alpha_2)$	$F_3(\theta_B, \alpha_2, \theta_C)$	$F_2(\alpha_4, \theta_A)$	$F_1(\theta_A)$
\vec{n}_B	$F_1(\alpha_1)$	0	1	0	$F_1(\alpha_2)$	$F_2(\alpha_2, \theta_C)$	$F_3(\alpha_2, \theta_C, \alpha_3)$	$F_2(\theta_A, \alpha_1)$
\vec{e}_2	$F_2(\alpha_1, \theta_B)$	$F_1(\theta_B)$	0	1	0	$F_1(\theta_C)$	$F_2(\theta_C, \alpha_3)$	$F_3(\theta_C, \alpha_3, \theta_D)$
\vec{n}_C	$F_3(\alpha_1, \theta_B, \alpha_2)$	$F_2(\theta_B, \alpha_2)$	$F_1(\alpha_2)$	0	1	0	$F_1(\alpha_3)$	$F_2(\alpha_3, \theta_D)$
\vec{e}_3	$F_2(\theta_D, \alpha_4)$	$F_1(\theta_C)$	$F_2(\alpha_2, \theta_C)$	$F_1(\theta_C)$	0	1	0	$F_1(\theta_D)$
\vec{n}_D	$F_1(\alpha_4)$	$F_2(\alpha_4, \theta_A)$	$F_3(\alpha_2, \theta_C, \alpha_3)$	$F_2(\theta_C, \alpha_3)$	$F_1(\alpha_3)$	0	1	0
\vec{e}_4	0	$F_1(\theta_A)$	$F_2(\theta_A, \alpha_1)$	$F_3(\theta_C, \alpha_3, \theta_D)$	$F_2(\alpha_3, \theta_D)$	$F_1(\theta_D)$	0	1

TABLE 3.3 – Matrice de Gram unitaire symétrique

Une fois que la matrice de Gram unitaire simplifiée est rendue symétrique, il faut écrire la matrice de Gram non-unitaire pour pouvoir appliquer la méthode des coupes métriques. Dans



le cas du 4R, la matrice des longueurs, notée L , permettant d'effectuer le produit est présentée par l'équation (3.31). Le résultat du produit est montré dans l'équation (3.32)

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_4 \end{pmatrix} \quad (3.31)$$

$$L \cdot G(N) \cdot L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_1^2 & 0 & L_1 L_2 F_1(\theta_B) & 0 & L_1 L_3 F_3(\theta_B, \alpha_2, \theta_C) & 0 & L_1 L_4 F_1(\theta_A) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_2 L_1 F_2(\theta_B) & 0 & L_2^2 & 0 & L_2 L_3 F_1(\theta_C) & 0 & L_2 L_4 F_3(\theta_C, \alpha_3, \theta_D) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_3 L_1 F_3(\theta_B, \alpha_2, \theta_C) & 0 & L_3 L_2 F_1(\theta_C) & 0 & L_3^2 & 0 & L_3 L_4 F_1(\theta_D) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_4 L_1 F_1(\theta_A) & 0 & L_4 L_2 F_3(\theta_C, \alpha_3, \theta_D) & 0 & L_4 L_3 F_1(\theta_D) & 0 & L_4^2 \end{pmatrix} \quad (3.32)$$

Avec l'équation (3.32), il est facile d'effectuer deux coupes métriques (voir §3.3.2.1).

Coupure 1	$(L_1 \vec{e}_1 + L_2 \vec{e}_2)^2 = (L_3 \vec{e}_3 + L_4 \vec{e}_4)^2$
Coupure 2	$(L_2 \vec{e}_2 + L_3 \vec{e}_3)^2 = (L_1 \vec{e}_1 + L_4 \vec{e}_4)^2$
Coupure 3	$(L_1 \vec{e}_1 + L_2 \vec{e}_2 + L_3 \vec{e}_3)^2 = (L_4 \vec{e}_4)^2$

Les résultats des coupures 1 et 2 sont représentés par les équations (3.33) et (3.34).

$$L_1^2 + L_2^2 + 2L_1 L_2 F_1(\theta_B) = L_3^2 + L_4^2 + 2L_3 L_4 F_1(\theta_D) \implies L_1^2 + L_2^2 + 2L_1 L_2 \cos(\theta_B) = L_3^2 + L_4^2 + 2L_3 L_4 \cos(\theta_D) \quad (3.33)$$

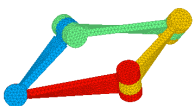
$$L_2^2 + L_3^2 + 2L_2 L_3 F_1(\theta_C) = L_1^2 + L_4^2 + 2L_1 L_4 F_1(\theta_A) \implies L_2^2 + L_3^2 + 2L_2 L_3 \cos(\theta_C) = L_1^2 + L_4^2 + 2L_1 L_4 \cos(\theta_A) \quad (3.34)$$

Ainsi, le 1^{er} système d'équations qui décrit la fermeture angulaire et la fermeture affine est

$$\left\{ \begin{array}{l} F_1(\alpha_3) = F_5(\alpha_4, \theta_A, \alpha_1, \theta_B, \alpha_2) \\ F_3(\alpha_2, \theta_C, \alpha_3) = F_3(\alpha_4, \theta_A, \alpha_1) \\ F_3(\alpha_1, \theta_B, \alpha_2) = F_3(\alpha_3, \theta_D, \alpha_4) \\ L_1^2 + L_2^2 + 2L_1 L_2 F_1(\theta_B) = L_3^2 + L_4^2 + 2L_3 L_4 F_1(\theta_D) \\ L_2^2 + L_3^2 + 2L_2 L_3 F_1(\theta_C) = L_1^2 + L_4^2 + 2L_1 L_4 F_1(\theta_A) \\ L_4^2 = L_1^2 + L_2^2 + L_3^2 + 2L_1 L_2 F_2(\alpha_1, \theta_B) + 2L_2 L_3 F_2(\alpha_2, \theta_C) + 2L_1 L_3 F_3(\theta_B, \alpha_2, \theta_C) \end{array} \right. \quad (3.35)$$

L'avantage de cette méthode est que nous pouvons choisir librement les paramètres contenus dans les équations de coupe métrique. Avec les différentes symétries de la matrice de Gram non-unitaire, le nombre d'équations que nous pouvons écrire est trop élevé. Comme cette méthode n'a pas été automatisée, il est impossible d'écrire toutes les équations avec toutes les combinaisons. Il faut donc faire un choix compatible avec les paramètres de mobilité des équations angulaires et des équations de coupes métriques pour pouvoir les résoudre ensuite.

Le 1^{er} exemple du système d'équations est obtenu lorsque les fonctions F_i de l'équation (3.35) sont développées :



$$\left\{ \begin{array}{l} \cos \alpha_2 (\cos \alpha_4 \cos \alpha_1 - \sin \alpha_4 \sin \alpha_1 \cos \theta_A) - \sin \alpha_2 (\cos \alpha_4 \sin \alpha_1 \cos \theta_B - \\ \quad \sin \alpha_4 \sin \theta_A \sin \theta_B + \cos \alpha_1 \sin \alpha_4 \cos \theta_A \cos \theta_B) - \cos \alpha_3 = 0 \\ \cos \alpha_4 \cos \alpha_1 - \sin \alpha_4 \sin \alpha_1 \cos \theta_A - \cos \alpha_2 \cos \alpha_3 + \sin \alpha_2 \sin \alpha_3 \cos \theta_C = 0 \\ \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 \cos \theta_B - \cos \alpha_3 \cos \alpha_4 + \sin \alpha_3 \sin \alpha_4 \cos \theta_D = 0 \\ \quad L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_B - L_3^2 - L_4^2 - 2L_3L_4 \cos \theta_D = 0 \\ \quad L_2^2 + L_3^2 + 2L_2L_3 \cos \theta_C - L_1^2 - L_4^2 - 2L_1L_4 \cos \theta_A = 0 \\ \quad L_1^2 + L_2^2 + L_3^2 - L_4^2 + 2L_1L_2 \cos \theta_B + 2L_2L_3 \cos \theta_C + \\ \quad \quad 2L_1L_3 (\cos \theta_B \cos \theta_C - \sin \theta_B \sin \theta_C \cos \alpha_2) = 0 \end{array} \right. \quad (3.36)$$

3.4.1.1.2 Projection

Puisque cette méthode est aussi basée sur la matrice de Gram unitaire symétrique, nous allons utiliser le résultat de symétrie présenté dans le tableau 3.3. Pour écrire les équations de projection, appliquons l'équation (1.18). Le résultat est donné par l'équation (3.37).

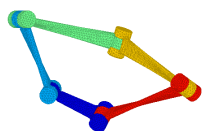
$$G(N) \cdot L = \begin{pmatrix} 0 + 0 + 0 + L_2F_2(\alpha_1, \theta_B) + 0 + L_3F_2(\theta_D, \alpha_4) + 0 + 0 \\ 0 + L_1 + 0 + L_2F_1(\theta_B) + 0 + L_3F_3(\theta_B, \alpha_2, \theta_C) + 0 + L_4F_1(\theta_A) \\ 0 + 0 + 0 + 0 + 0 + L_3F_2(\alpha_2, \theta_C) + 0 + L_4F_2(\theta_A, \alpha_1) \\ 0 + L_1F_2(\theta_B) + 0 + L_2 + 0 + L_3F_1(\theta_C) + 0 + L_4F_3(\theta_C, \alpha_3, \theta_D) \\ 0 + L_1F_2(\theta_B, \alpha_2) + 0 + 0 + 0 + 0 + 0 + L_4F_2(\alpha_3, \theta_D) \\ 0 + L_1F_3(\theta_B, \alpha_2, \theta_C) + 0 + L_2F_1(\theta_C) + 0 + L_3 + 0 + L_4F_1(\theta_D) \\ 0 + L_1F_2(\alpha_4, \theta_A) + 0 + L_2F_2(\theta_C, \alpha_3) + 0 + 0 + 0 + 0 \\ 0 + L_1F_1(\theta_A) + 0 + L_2F_3(\theta_C, \alpha_3, \theta_D) + 0 + L_3F_1(\theta_D) + 0 + L_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.37)$$

Après une première étude, les équations peuvent être séparées en deux parties : une partie qui contient 3 paramètres de mobilité et une partie qui contient 2 paramètres de mobilité. Pour faciliter la résolution, nous nous intéressons aux équations avec 2 paramètres de mobilité. Elles sont :

$$\left\{ \begin{array}{l} L_2F_2(\alpha_1, \theta_B) + L_3F_2(\theta_D, \alpha_4) = 0 \\ L_3F_2(\alpha_2, \theta_C) + L_4F_2(\theta_A, \alpha_1) = 0 \\ L_1F_2(\theta_B, \alpha_2) + L_4F_2(\alpha_3, \theta_D) = 0 \\ L_1F_2(\alpha_4, \theta_A) + L_2F_2(\theta_C, \alpha_3) = 0 \end{array} \right. \quad (3.38)$$

Ainsi, le 2^{ème} exemple du système d'équations qui décrit le 4R :

$$\left\{ \begin{array}{l} \cos \alpha_2 (\cos \alpha_4 \cos \alpha_1 - \sin \alpha_4 \sin \alpha_1 \cos \theta_A) - \sin \alpha_2 (\cos \alpha_4 \sin \alpha_1 \cos \theta_B - \\ \quad \sin \alpha_4 \sin \theta_A \sin \theta_B + \cos \alpha_1 \sin \alpha_4 \cos \theta_A \cos \theta_B) - \cos \alpha_3 = 0 \\ \cos \alpha_4 \cos \alpha_1 - \sin \alpha_4 \sin \alpha_1 \cos \theta_A - \cos \alpha_2 \cos \alpha_3 + \sin \alpha_2 \sin \alpha_3 \cos \theta_C = 0 \\ \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 \cos \theta_B - \cos \alpha_3 \cos \alpha_4 + \sin \alpha_3 \sin \alpha_4 \cos \theta_D = 0 \\ \quad L_2 \sin \alpha_1 \sin \theta_B + L_3 \sin \theta_D \sin \alpha_4 = 0 \\ \quad L_3 \sin \alpha_2 \sin \theta_C + L_4 \sin \theta_A \sin \alpha_1 = 0 \\ \quad L_1 \sin \alpha_2 \sin \theta_B + L_4 \sin \theta_D \sin \alpha_3 = 0 \\ \quad L_1 \sin \alpha_4 \sin \theta_A + L_2 \sin \theta_C \sin \alpha_3 = 0 \end{array} \right. \quad (3.39)$$



3.4.2 Méthode de Jin et Yang

Appliquons la méthode de Jin et Yang dans le cas du 4R. Dans la figure 3.10, la coupure est réalisée entre \vec{n}_B et \vec{n}_D . Le 4R est séparé en deux sous-systèmes : \vec{R}_A et \vec{R}_B , voir la figure 3.11.

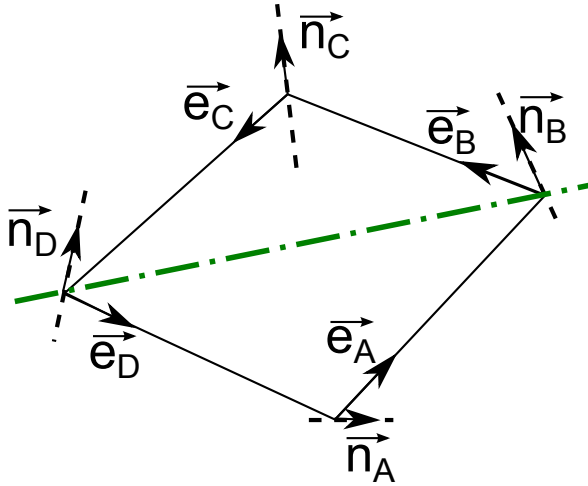


FIGURE 3.10 – La coupure effectuée sur les liaisons pivots \vec{n}_B et \vec{n}_D

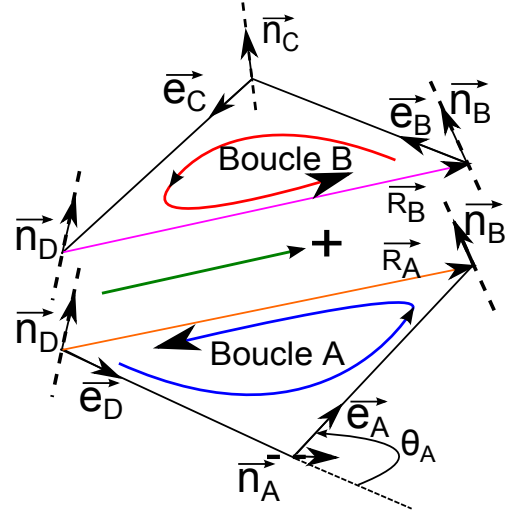


FIGURE 3.11 – La coupure sur les liaisons pivots \vec{n}_B et \vec{n}_D fait intervenir les angles θ_A et θ_C

En projetant \vec{R}_A et \vec{R}_B , les équations (3.15) (3.16) (3.17) (3.18) se transforment en équations (3.40) (3.41) (3.42) (3.43). Comme indiquée dans la figure 3.11, la direction de la somme des vecteurs est définie par le vecteur \vec{R}_A ou \vec{R}_B .

$$\langle \vec{n}_B, \vec{n}_D \rangle_A = \langle \vec{n}_B, \vec{n}_D \rangle_B \tag{3.40}$$

$$\langle \vec{R}_A, \vec{n}_B \times \vec{n}_D \rangle_A = \langle \vec{R}_B, \vec{n}_B \times \vec{n}_D \rangle_B \tag{3.41}$$

$$\langle \vec{R}_A, \vec{n}_B \rangle_A = \langle \vec{R}_B, \vec{n}_B \rangle_B \tag{3.42}$$

$$\langle \vec{R}_A, \vec{n}_D \rangle_A = \langle \vec{R}_B, \vec{n}_D \rangle_B \tag{3.43}$$

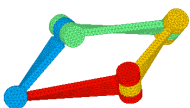
avec :

$$\vec{R}_A = L_4 \vec{e}_D + L_1 \vec{e}_A$$

$$\vec{R}_B = -(L_2 \vec{e}_B + L_3 \vec{e}_C)$$

$$\vec{R}_A = -\vec{R}_B$$

Après avoir coupé la boucle au niveau des liaisons \vec{n}_B et \vec{n}_D , il ne reste que les variables θ_A et θ_C . Le système d'équations (3.44) est le 3^{ème} exemple qui décrit le 4R :



$$\left\{ \begin{array}{l} \cos \alpha_2 \cos \alpha_3 - \sin \alpha_3 \sin \alpha_2 \cos \theta_C - \cos \alpha_4 \cos \alpha_1 + \sin \alpha_1 \sin \alpha_4 \cos \theta_A = 0 \\ L_2(\cos \alpha_3 \sin \alpha_2 + \cos \alpha_2 \sin \alpha_3 \cos \theta_C) + L_3(\cos \alpha_2 \sin \alpha_3 + \cos \alpha_3 \sin \alpha_2 \cos \theta_C) \\ -L_4(\cos \alpha_1 \sin \alpha_4 + \cos \alpha_4 \sin \alpha_1 \cos \theta_A) - L_1(\cos \alpha_4 \sin \alpha_1 + \cos \alpha_1 \sin \alpha_4 \cos \theta_A) = 0 \\ L_3 \sin \alpha_2 \sin \theta_C + L_4 \sin \theta_A \sin \alpha_1 = 0 \\ L_2 \sin \theta_C \sin \alpha_3 + L_1 \sin \alpha_4 \sin \theta_A = 0 \end{array} \right. \quad (3.44)$$

3.4.3 Variante de la méthode de Jin et Yang

À l'instar de la méthode de Jin et Yang, les équations (3.21), (3.23) et (3.24) sont identiques aux équations (3.40), (3.42) et (3.43) si la coupure est effectuée en \vec{n}_B et \vec{n}_D . En revanche, l'équation (3.22) devient l'équation (3.45).

$$(L_4\vec{e}_D + L_1\vec{e}_A)^2 = (L_2\vec{e}_B + L_3\vec{e}_C)^2 \quad (3.45)$$

Après l'avoir développée, l'équation précédente se transforme en équation (3.46) :

$$L_4^2 + 2L_4L_1 \cos \theta_A + L_1^2 = L_2^2 + 2L_2L_3 \cos \theta_C + L_3^2 \quad (3.46)$$

Ainsi, le 4^{ème} exemple du système d'équations de 4R est :

$$\left\{ \begin{array}{l} \cos \alpha_2 \cos \alpha_3 - \sin \alpha_3 \sin \alpha_2 \cos \theta_C - \cos \alpha_4 \cos \alpha_1 + \sin \alpha_1 \sin \alpha_4 \cos \theta_A = 0 \\ L_4^2 + 2L_4L_1 \cos \theta_A + L_1^2 - L_2^2 - 2L_2L_3 \cos \theta_C - L_3^2 = 0 \\ L_3 \sin \alpha_2 \sin \theta_C + L_4 \sin \theta_A \sin \alpha_1 = 0 \\ L_2 \sin \theta_C \sin \alpha_3 + L_1 \sin \alpha_4 \sin \theta_A = 0 \end{array} \right. \quad (3.47)$$

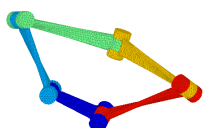
Comme la méthode de Jin et Yang, grâce à la coupure aux liaisons \vec{n}_B et \vec{n}_D , le système d'équations ne dépend que de deux paramètres de mobilité θ_A et θ_C . Ceci facilite l'élimination des paramètres de mobilité pour la suite. Cette méthode permet d'écrire 4 équations indépendantes même si \vec{n}_B et \vec{n}_D sont parallèles.

3.4.4 Discussion

3.4.4.1 Résolution manuelle des équations générées

Dans cette partie, nous cherchons à résoudre manuellement les systèmes d'équations dans le but de retrouver les conditions de mobilité du mécanisme de Bennett publié par Bennett lui-même [Ben03]. En effet, le mécanisme de Bennett est un système de 4R mobile en 3D avec 1 *ddl*.

En regroupant les différents exemples du système d'équations (3.36), (3.39), (3.44) et (3.47), nous obtenons 13 équations (3.48)-(3.60) avec 4 variables θ_A , θ_B , θ_C et θ_D . Les L_i et α_i sont



considérés comme les paramètres. Nous allons éliminer les variables par la suite.

$$L2 \sin \alpha_1 \sin \theta_B + L3 \sin \theta_D \sin \alpha_4 = 0 \quad (3.48)$$

$$L1 \sin \alpha_2 \sin \theta_B + L4 \sin \theta_D \sin \alpha_3 = 0 \quad (3.49)$$

$$\cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 \cos \theta_B - \cos \alpha_3 \cos \alpha_4 + \sin \alpha_3 \sin \alpha_4 \cos \theta_D = 0 \quad (3.50)$$

$$\cos \alpha_1 \cos \alpha_4 - \sin \alpha_1 \cos \theta_A \sin \alpha_4 - \cos \alpha_2 \cos \alpha_3 + \sin \alpha_2 \sin \alpha_3 \cos \theta_C = 0 \quad (3.51)$$

$$L1^2 + 2 L1 L2 \cos \theta_B + L2^2 - L3^2 - L4^2 - 2 L3 L4 \cos \theta_D = 0 \quad (3.52)$$

$$L2^2 + 2 L2 L3 \cos \theta_C + L3^2 - L1^2 - L4^2 - 2 L1 L4 \cos \theta_A = 0 \quad (3.53)$$

$$L1^2 + L2^2 + L3^2 - L4^2 + 2 L1 L2 \cos \theta_B + 2 L2 L3 \cos \theta_C + 2 L1 L3 (\cos \theta_B \cos \theta_C - \sin \theta_B \sin \theta_C \cos \alpha_2) = 0 \quad (3.54)$$

$$L3 \sin \alpha_2 \sin \theta_C + L4 \sin \theta_A \sin \alpha_1 = 0 \quad (3.55)$$

$$L1 \sin \alpha_4 \sin \theta_A + L2 \sin \theta_C \sin \alpha_3 = 0 \quad (3.56)$$

$$\cos \alpha_2 \cos \alpha_3 - \sin \alpha_2 \sin \alpha_3 \cos \theta_C - \cos \alpha_1 \cos \alpha_4 + \sin \alpha_1 \cos \theta_A \sin \alpha_4 = 0 \quad (3.57)$$

$$L2 (\cos \alpha_3 \sin \alpha_2 + \cos \alpha_2 \sin \alpha_3 \cos \theta_C) + L3 (\cos \alpha_2 \sin \alpha_3 + \cos \alpha_3 \sin \alpha_2 \cos \theta_C) - L4 (\cos \alpha_1 \sin \alpha_4 + \cos \alpha_4 \sin \alpha_1 \cos \theta_A) - L1 (\cos \alpha_4 \sin \alpha_1 + \cos \alpha_1 \sin \alpha_4 \cos \theta_A) = 0 \quad (3.58)$$

$$L1^2 + 2 L1 L4 \cos \theta_A + L4^2 - L2^2 - L3^2 - 2 L2 L3 \cos \theta_C = 0 \quad (3.59)$$

$$\cos \alpha_2 (\cos \alpha_1 \cos \alpha_4 - \sin \alpha_1 \cos \theta_A \sin \alpha_4) + \sin \alpha_2 (-\cos \theta_B \cos \theta_A \cos \alpha_1 \sin \alpha_4 - \sin \alpha_1 \cos \theta_B \cos \alpha_4 + \sin \theta_B \sin \theta_A \sin \alpha_4) - \cos \alpha_3 = 0 \quad (3.60)$$

Nous nous sommes rendu compte que les paramètres de mobilité θ_B et θ_D interviennent dans les équations (3.48)-(3.52). Après avoir substitué l'équation (3.50) dans (3.52), nous obtenons l'équation

$$2(L1L2 \sin \alpha_3 \sin \alpha_4 - L3L4 \sin \alpha_1 \sin \alpha_2) \cos \theta_B + 2L3L4(\cos \alpha_1 \cos \alpha_2 - \cos \alpha_3 \cos \alpha_4) + \sin \alpha_3 \sin \alpha_4(L1^2 + L2^2 - L3^2 - L4^2) = 0 \quad (3.61)$$

Maintenant, il ne reste plus qu'un seul paramètre de mobilité θ_B . Si nous considérons θ_B comme une variable, l'équation (3.61) doit toujours être respectée pour un 4R mobile. Ainsi, quelque soit θ_B , l'équation (3.61) est identiquement nulle quand un système est mobile. Il faut donc que le coefficient de $\cos \theta_B$ et que le terme constant soient nuls, ce qui donne l'équation (3.62) :

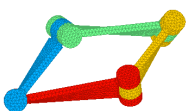
$$\begin{cases} 2L3L4(\cos \alpha_1 \cos \alpha_2 - \cos \alpha_3 \cos \alpha_4) + \sin \alpha_3 \sin \alpha_4(L1^2 + L2^2 - L3^2 - L4^2) = 0 \\ L1L2 \sin \alpha_3 \sin \alpha_4 - L3L4 \sin \alpha_1 \sin \alpha_2 = 0 \end{cases} \quad (3.62)$$

De la même manière, en substituant l'équation (3.57) dans (3.59), nous obtenons :

$$2(L2L3 \sin \alpha_1 \sin \alpha_4 - L1L4 \sin \alpha_2 \sin \alpha_3) \cos \theta_A + 2L3L2(\cos \alpha_3 \cos \alpha_2 - \cos \alpha_1 \cos \alpha_4) - \sin \alpha_3 \sin \alpha_2(L1^2 + L4^2 - L3^2 - L2^2) = 0 \quad (3.63)$$

Pour que le 4R soit mobile, il faut aussi respecter l'équation (3.64) quelque soit θ_A :

$$\begin{cases} 2L3L2(\cos \alpha_3 \cos \alpha_2 - \cos \alpha_1 \cos \alpha_4) - \sin \alpha_3 \sin \alpha_2(L1^2 + L4^2 - L3^2 - L2^2) = 0 \\ L3L2 \sin \alpha_1 \sin \alpha_4 - L1L4 \sin \alpha_3 \sin \alpha_2 = 0 \end{cases} \quad (3.64)$$



A partir des équations (3.48) et (3.49), il est facile d'obtenir l'équation (3.65) :

$$(L_1 \sin \alpha_2 - \frac{L_2 L_4 \sin \alpha_1 \sin \alpha_3}{L_3 \sin \alpha_4}) \sin \theta_B = 0 \quad (3.65)$$

θ_B est considéré comme une variable. Nous souhaitons que le 4R soit mobile avec 1 *ddl*, θ_B , variant par rapport au temps. L'équation (3.65) doit toujours être vérifiée. Il faut que le coefficient de $\sin \theta_B$ soit identiquement nul, voir l'équation (3.66).

$$L_1 L_3 \sin \alpha_2 \sin \alpha_4 - L_2 L_4 \sin \alpha_1 \sin \alpha_3 = 0 \quad (3.66)$$

Si nous traitons le groupe d'équations (3.55) et (3.56) qui dépendent de θ_A et θ_C , nous pouvons substituer l'équation (3.55) dans l'équation (3.56), le résultat est l'équation (3.67) qui ne fait intervenir aucun paramètre de mobilité.

$$L_2 L_4 \sin \alpha_1 \sin \alpha_3 - L_1 L_3 \sin \alpha_2 \sin \alpha_4 = 0 \quad (3.67)$$

Après avoir regroupé les différents exemples du système d'équations, nous avons analysé quelques équations parmi (3.48)-(3.60). Nous constatons que les formes des équations dans chaque résultat sont similaires. Ainsi, nous obtenons :

$$2L_3 L_4 (\cos \alpha_1 \cos \alpha_2 - \cos \alpha_3 \cos \alpha_4) + \sin \alpha_3 \sin \alpha_4 (L_1^2 + L_2^2 - L_3^2 - L_4^2) = 0 \quad (3.68)$$

$$2L_3 L_2 (\cos \alpha_3 \cos \alpha_2 - \cos \alpha_1 \cos \alpha_4) - \sin \alpha_3 \sin \alpha_2 (L_1^2 + L_4^2 - L_3^2 - L_2^2) = 0 \quad (3.69)$$

$$L_1 L_2 \sin \alpha_3 \sin \alpha_4 - L_3 L_4 \sin \alpha_1 \sin \alpha_2 = 0 \quad (3.70)$$

$$L_3 L_2 \sin \alpha_1 \sin \alpha_4 - L_1 L_4 \sin \alpha_3 \sin \alpha_2 = 0 \quad (3.71)$$

$$L_2 L_4 \sin \alpha_1 \sin \alpha_3 - L_1 L_3 \sin \alpha_2 \sin \alpha_4 = 0 \quad (3.72)$$

Dans un premier temps, l'équation (3.70) peut être exprimée par l'équation (3.73).

$$L_4 = \frac{L_1 L_2 \sin \alpha_3 \sin \alpha_4}{L_3 \sin \alpha_1 \sin \alpha_2} \quad (3.73)$$

Après avoir substitué l'équation (3.73) dans l'équation (3.71), elle se transforme en l'équation (3.74) :

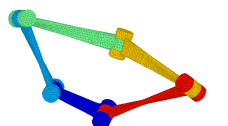
$$\frac{L_2 \sin \alpha_4}{L_3 \sin \alpha_1} (L_3^2 (\sin \alpha_1)^2 - L_1^2 (\sin \alpha_3)^2) = 0 \quad (3.74)$$

En réalité, la longueur d'une pièce mécanique et son angle dimensionnel sont supérieurs à 0. Pour que l'équation (3.74) soit toujours vérifiée, il faut donc que :

$$L_3^2 (\sin \alpha_1)^2 = L_1^2 (\sin \alpha_3)^2 \quad (3.75)$$

De la même façon, après substitution de l'équation (3.73) dans l'équation (3.72), nous obtenons :

$$L_3^2 (\sin \alpha_2)^2 = L_2^2 (\sin \alpha_3)^2 \quad (3.76)$$



L'équation (3.75) exprime la relation entre les paramètres dimensionnels de la pièce 1 et ceux de la pièce 3; l'équation (3.76) est une expression entre les paramètres dimensionnels de la pièce 2 et ceux de la pièce 3. De plus, une troisième équation engendrée par l'équation (3.75) et l'équation (3.76) est :

$$L_1^2(\sin \alpha_2)^2 = L_2^2(\sin \alpha_1)^2 \quad (3.77)$$

Si nous changeons l'expression de l'équation (3.73), il est facile de trouver l'équation (3.78), (3.79) et (3.80) :

$$L_2^2(\sin \alpha_4)^2 = L_4^2(\sin \alpha_2)^2 \quad (3.78)$$

$$L_3^2(\sin \alpha_4)^2 = L_4^2(\sin \alpha_3)^2 \quad (3.79)$$

$$L_1^2(\sin \alpha_4)^2 = L_4^2(\sin \alpha_1)^2 \quad (3.80)$$

Après le changement de variables $\sin \alpha_1 \rightarrow s1$, $\sin \alpha_2 \rightarrow s2$, $\sin \alpha_3 \rightarrow s3$ et $\sin \alpha_4 \rightarrow s4$, développons maintenant les équations (3.78)-(3.80) en fonction de L_4 :

$$L1 = \epsilon_1 \frac{L_4 s1}{s4} \quad (3.81)$$

$$L2 = \epsilon_2 \frac{L_4 s2}{s4} \quad (3.82)$$

$$L3 = \epsilon_3 \frac{L_4 s3}{s4} \quad (3.83)$$

avec $\epsilon_i = \pm 1$. En substituant les équations (3.81)-(3.83) dans l'équation (3.68), nous obtenons :

$$Eq(3.68) = 2\epsilon_3(c1c2 - c3c4) + (s1^2 + s2^2 - s3^2 - s4^2) \quad (3.84)$$

$$= 2\epsilon_3(c1c2 - c3c4) + (1 - c1^2 + 1 - c2^2 - 1 + c3^2 - 1 + c4^2) \quad (3.85)$$

$$= 2\epsilon_3(c1c2 - c3c4) - c1^2 - c2^2 + c3^2 + c4^2 \quad (3.86)$$

$$= (c3 + \epsilon_3 c4)^2 - (c1 + \epsilon_3 c2)^2 = 0 \quad (3.87)$$

En substituant les équations (3.81)-(3.83) dans l'équation (3.69), nous obtenons :

$$Eq(3.69) = 2\epsilon_2\epsilon_3(c3c2 - c1c4) + (s1^2 + s4^2 - s3^2 - s2^2) \quad (3.88)$$

$$= 2\epsilon_2\epsilon_3(c3c2 - c1c4) + (1 - c1^2 + 1 - c4^2 - 1 + c3^2 - 1 + c2^2) \quad (3.89)$$

$$= 2\epsilon_2\epsilon_3(c3c2 - c1c4) - c1^2 - c4^2 + c3^2 + c2^2 \quad (3.90)$$

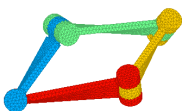
$$= (c2 + \epsilon_2\epsilon_3 c3)^2 - (c1 + \epsilon_2\epsilon_3 c4)^2 = 0 \quad (3.91)$$

Supposons que $\epsilon = \pm 1$ et $\epsilon' = \pm 1$, les équations (3.87) et (3.91) s'écrivent :

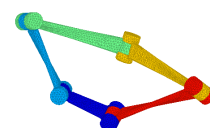
$$Eq. (3.87) \implies (c3 + \epsilon_3 c4) = \epsilon(c1 + \epsilon_3 c2) \quad (3.92)$$

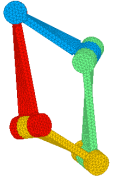
$$Eq. (3.91) \implies (c2 + \epsilon_2\epsilon_3 c3) = \epsilon'(c1 + \epsilon_2\epsilon_3 c4) \quad (3.93)$$

L'analyse des signes de ϵ , ϵ' , ϵ_2 et ϵ_3 permet de dresser le tableau 3.4. Parmi les 16 combinaisons, 2 cases sont vides car pour ces deux cas, les équations (3.92) et (3.93) sont identiques. Nous ne pouvons pas déduire d'autres relations. Dans les cases non vides, on obtient d'abord 2 relations de θ_i en résolvant les équations (3.92) et (3.93). A partir de ces 2 relations, on peut calculer les 2 relations de longueurs grâce aux équations (3.68)-(3.72). Avec les relations en $\cos \alpha$, nous



avons déduit toutes les solutions en $\sin \alpha$. Puisque toutes les longueurs doivent être positives, certaines solutions en $\sin \alpha$ sont déjà supprimées grâce aux équations (3.81)-(3.83). Dans les cas où $(\epsilon = -1, \epsilon' = -1, \epsilon_2 = 1, \epsilon_3 = 1)$ et $(\epsilon = 1, \epsilon' = -1, \epsilon_2 = -1, \epsilon_3 = 1)$, la résolution manuelle n'a pas abouti car nous n'avons que deux équations au lieu d'en avoir quatre.





ϵ_2 ϵ_3		$\epsilon = 1$		$\epsilon = -1$		$\epsilon = 1$		$\epsilon = -1$			
		$\epsilon' = 1$		$\epsilon' = -1$		$\epsilon' = -1$		$\epsilon' = 1$			
1	1	$c_1 = c_3 \xrightarrow{(3.64)} L_1 = L_3$		La résolution manuelle n'a pas abouti.				$c_1 = -c_2 \xrightarrow{(3.67)} L_1 = L_2$		$c_1 = -c_4 \xrightarrow{(3.66)} L_1 = L_4$	
		$c_2 = c_4 \xrightarrow{(3.65)} L_2 = L_4$						$c_3 = -c_4 \xrightarrow{(3.68)} L_3 = L_4$		$c_2 = -c_3 \xrightarrow{(3.68)} L_2 = L_3$	
		$s_1 = s_3$	$s_1 = -s_3$					$s_1 = s_2$	$s_1 = -s_2$	$s_1 = s_4$	$s_1 = -s_4$
		$s_2 = s_4$	$s_3 = s_4$	$s_3 = s_4$	$s_3 = s_4$	$s_2 = s_3$	$s_2 = s_3$	$s_2 = s_3$	$s_2 = s_3$		
		$s_1 = s_2$	$s_3 = s_4$	$s_1 = s_2$	$s_1 = -s_2$	$s_1 = s_4$	$s_1 = -s_4$	$s_2 = -s_3$	$s_2 = -s_3$		
		$s_2 = s_4$	$s_3 = -s_4$	$s_3 = -s_4$	$s_3 = -s_4$	$s_2 = -s_3$	$s_2 = -s_3$	$s_2 = -s_3$	$s_2 = -s_3$		
1	-1	$c_1 = c_2 \xrightarrow{(3.67)} L_1 = L_2$		$c_1 = c_4 \xrightarrow{(3.66)} L_1 = L_4$		$c_1 = c_3 \xrightarrow{(3.64)} L_1 = L_3$		$c_1 = c_4 \xrightarrow{(3.66)} L_1 = L_4$			
		$c_3 = c_4 \xrightarrow{(3.68)} L_3 = L_4$		$c_2 = c_3 \xrightarrow{(3.68)} L_2 = L_3$		$c_2 = c_4 \xrightarrow{(3.65)} L_2 = L_4$		$c_2 = c_3 \xrightarrow{(3.68)} L_2 = L_3$			
		$s_1 = s_2$	$s_1 = -s_2$	$s_1 = s_4$	$s_1 = s_4$	$s_1 = s_3$	$s_1 = -s_3$	$s_1 = s_4$	$s_1 = s_4$		
		$s_3 = s_4$	$s_3 = -s_4$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_4$	$s_2 = s_4$	$s_2 = s_3$	$s_2 = -s_3$		
		$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$		
		$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_3$	$s_2 = -s_3$		
-1	1	$c_1 = c_4 \xrightarrow{(3.66)} L_1 = L_4$		$c_1 = -c_2 \xrightarrow{(3.67)} L_1 = L_2$		La résolution manuelle n'a pas abouti.				$c_1 = -c_3 \xrightarrow{(3.64)} L_1 = L_3$	
		$c_2 = c_3 \xrightarrow{(3.68)} L_2 = L_3$		$c_3 = -c_4 \xrightarrow{(3.68)} L_3 = L_4$						$c_2 = -c_4 \xrightarrow{(3.65)} L_2 = L_4$	
		$s_1 = s_4$	$s_1 = s_4$	$s_1 = s_2$	$s_1 = -s_2$					$s_1 = s_3$	$s_1 = -s_3$
		$s_2 = s_3$	$s_2 = -s_3$	$s_3 = s_4$	$s_3 = s_4$	$s_2 = s_4$	$s_2 = s_4$	$s_2 = -s_3$	$s_2 = -s_3$		
		$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = s_2$	$s_1 = -s_2$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$		
		$s_2 = s_3$	$s_2 = -s_3$	$s_3 = -s_4$	$s_3 = -s_4$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = -s_4$	$s_2 = -s_4$		
-1	-1	$c_1 = -c_4 \xrightarrow{(3.66)} L_1 = L_4$		$c_1 = -c_3 \xrightarrow{(3.64)} L_1 = L_3$		$c_1 = -c_3 \xrightarrow{(3.64)} L_1 = L_3$		$c_1 = -c_3 \xrightarrow{(3.64)} L_1 = L_3$			
		$c_2 = -c_3 \xrightarrow{(3.68)} L_2 = L_3$		$c_2 = -c_4 \xrightarrow{(3.65)} L_2 = L_4$		$c_2 = c_4 \xrightarrow{(3.65)} L_2 = L_4$		$c_2 = -c_4 \xrightarrow{(3.65)} L_2 = L_4$			
		$s_1 = s_4$	$s_1 = s_4$	$s_1 = s_3$	$s_1 = -s_3$	$s_1 = s_3$	$s_1 = -s_3$	$s_1 = s_3$	$s_1 = -s_3$		
		$s_2 = s_3$	$s_2 = -s_3$	$s_2 = -s_4$	$s_2 = -s_4$	$s_2 = -s_4$	$s_2 = -s_4$	$s_2 = -s_4$	$s_2 = -s_4$		
		$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$	$s_1 = -s_4$		
		$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = s_3$	$s_2 = -s_3$	$s_2 = -s_4$	$s_2 = -s_4$		

TABLE 3.4 – Tableau des solutions des équations (3.64) – (3.68)

3.4.4.2 Interprétation géométrique

Les solutions du tableau 3.4 respectent les équations (3.81)-(3.83), (3.92) et (3.93). Nous avons construit dans CATIA les systèmes de 4R à partir des solutions calculées. Grâce à la résolution manuelle, nous avons découvert 3 systèmes de 4R mobiles. Ils sont présentés dans les figures 3.12, 3.13 et 4.14.

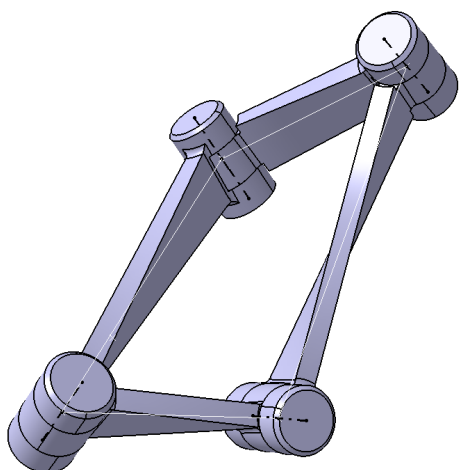


FIGURE 3.12 – Un mécanisme de 4R mobile en 3D : mécanisme de Bennett.

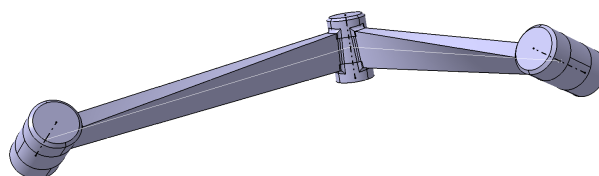
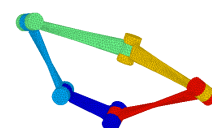


FIGURE 3.13 – Un mécanisme 4R mobile avec deux solides superposés aux deux autres.

Figure 3.12	Figure 3.13	Figure 4.14
$c_1 = c_3 \quad s_1 = s_3 \quad L_1 = L_3$ $c_2 = c_4 \quad s_2 = s_4 \quad L_2 = L_4$	$c_1 = c_2 \quad s_1 = s_2 \quad L_1 = L_2$ $c_3 = c_4 \quad s_3 = s_4 \quad L_3 = L_4$ ou $c_1 = c_4 \quad s_1 = s_4 \quad L_1 = L_4$ $c_2 = c_3 \quad s_2 = s_3 \quad L_2 = L_3$	$c_1 = -c_3 \quad s_1 = -s_3 \quad L_1 = L_3$ $c_2 = -c_4 \quad s_2 = -s_4 \quad L_2 = L_4$

TABLE 3.5 – Les solutions des 3 figures

Le 4R de la figure 3.12 est un mécanisme de Bennett présenté dans §1.1.4. Les conditions de mobilité sont publiées par Bennett [Ben03]. Le 4R de la figure 3.13 est construit par 4 solides avec deux superposés aux deux autres. Théoriquement, il est mobile, mais il est impossible de le fabriquer. Le 4R de la figure 3.14 est également mobile. Les deux solides jaunes se croisent mais il n’y a pas de collision entre eux. Les solutions correspondant aux 3 figures sont dans le tableau 3.5. Nous n’avons pas testé toutes les solutions car certaines représentent les mêmes systèmes de solides que ceux montrés dans les figures. Néanmoins, dans le tableau 3.4, toutes les solutions ne permettent pas de construire un système 4R fermé et mobile. Ainsi, une vérification géométrique est nécessaire pour valider les solutions obtenues.



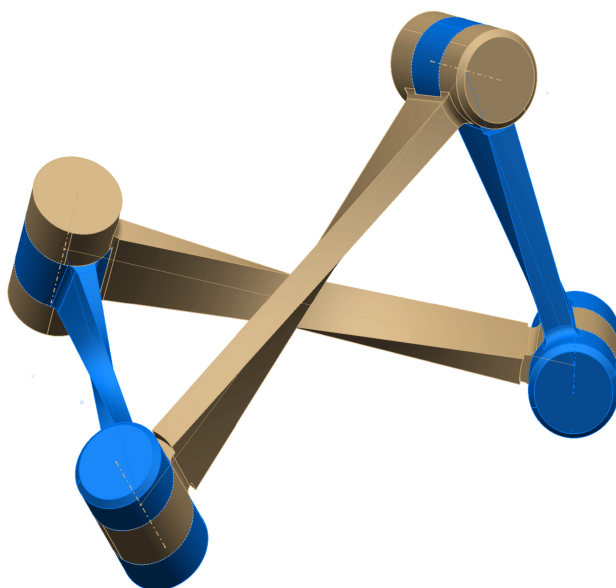


FIGURE 3.14 – Un exemple de mécanisme de 4R mobile

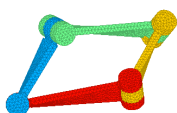
3.5 Conclusion

Dans ce chapitre, nous avons proposé l'écriture formelle de la matrice de Gram de rang 3 avec les fonctions F_i . Les équations de fermeture angulaire sont obtenues en écrivant la symétrie de la matrice de Gram unitaire. Deux méthodes ont été présentées pour assurer la fermeture de boucle : la méthode des coupes métriques et méthode des projections. L'ensemble des équations angulaires et de fermeture de boucle forment les équations de fermeture du système de solides.

L'inconvénient de la méthode des fonctions F_i est que le nombre d'équations dépend du nombre de vecteurs de la boucle. Par exemple, le 4R est défini par 36 équations dont 28 équations angulaires et 8 équations de fermeture de boucle ; le 5R est défini par 55 équations dont 45 équations angulaires et 10 équations vectorielles ; le 6R est défini par 78 équations dont 66 équations angulaires 12 équations vectorielles. En général, 6 équations sont suffisantes pour caractériser un système de solides fermé en 3D. Lors de la résolution formelle, le nombre d'équations impacte directement le temps de calcul. Nous savons écrire beaucoup d'équations mais il est difficile de les traiter toutes en même temps.

Ainsi, nous avons sélectionné quelques équations parmi celles que nous avons générées. L'objectif est de prendre les équations indépendantes ayant moins de paramètres de mobilité sans perdre d'information sur les solutions. Trois équations angulaires indépendantes et trois équations de fermeture de boucle indépendantes sont sélectionnées pour atteindre ce but. Néanmoins, le choix d'équations est compliqué. Nous ne pouvons pas garantir, a priori, l'indépendance des 6 équations.

Pour résoudre ce problème, la proposition de Jin et Yang a été utilisée. Il s'agit d'une méthode de mise en équation basée sur une coupure n'écrivant que 4 équations. Ceci permet de diminuer le nombre d'équations du système et le temps de résolution. En revanche, la projection sur $\vec{z}_i \times \vec{z}_j$ nécessite un calcul supplémentaire en écrivant les équations spécifiquement car le vecteur n'appartient pas à la liste des vecteurs du squelette. L'inconvénient de cette méthode apparaît



quand $\vec{z}_i // \vec{z}_j$. Dans ce cas, le rang de la matrice est inférieur à 3 car $\vec{z}_i \times \vec{z}_j = 0$. Les équations écrites deviennent dépendantes entre elles.

En s'inspirant de la méthode de Jin et Yang, nous avons proposé une variante : $R_A^2 = R_B^2$, qui est basé sur la matrice de Gram. Nous constatons aussi que le degré de l'équation $R_A^2 = R_B^2$ est plus élevé que l'équation $\langle \vec{R}, \vec{z}_i \times \vec{z}_j \rangle$ mais la taille de la première est beaucoup plus petite que la deuxième. Le résultat de cette proposition est identique à celui de la méthode de Jin et Yang. Ce fait confirme la faisabilité de cette méthode. La difficulté est la même que la méthode de Jin et Yang. Il faut donc toujours vérifier que les deux vecteurs choisis ne sont pas colinéaires.

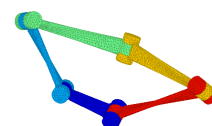
À la fin du chapitre, nous avons regroupé les différents exemples du système d'équations et les avons résolus manuellement. Nous avons discuté des solutions du 4R à l'aide d'un tableau. En construisant les modèles géométriques dans CATIA, nous avons découvert 3 types de 4R mobiles. Néanmoins, les solutions calculées ne permettent pas d'obtenir un système 4R mobile et fermé à chaque fois. Ainsi, une vérification géométrique est nécessaire pour valider ces solutions.

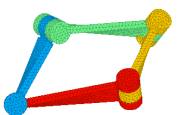
Malgré la réussite de la résolution manuelle pour le 4R, la taille des équations pour les mécanismes de 5 pivots et 6 pivots est plus grande ; il est alors trop difficile d'obtenir les conditions de mobilité avec un calcul manuel. Il faut donc une approche automatisée afin de trouver les conditions de mobilité et d'assemblage de la forme symbolique.

Dans le prochain chapitre, nous allons présenter la méthode de résolution formelle utilisée dans cette thèse : Bases de Gröbner. Cette méthode confortera les résultats trouvés. Une application de cette méthode sur le 4R sera également détaillée. Une analyse du résultat sera aussi fournie.

Bibliographie du chapitre 3

- [Ben03] G.T. Bennett. A new mechanism. *Engineering*, 76 :777–778, 1903.
- [Riv02] P. Serré & A. Clément & A. Rivière. Vers une approche déclarative en CFAO application au mécanisme de Bennett. In *Proceedings of the 4th International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME)*, 2002.
- [Riv06] P. Serré & A. Ortuzar & A. Rivière. Non-cartesian modelling for analysis of the consistency of a geometric specification for conceptual design. *International Journal of Computational Geometry and Applications*, 16(5/6) :549–565, 2006.
- [Rod40] Olindes Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace. *Journal de mathématiques pures et appliquées*, 5 :380–440, 1840.
- [Yan02] Qiong Jin & Tingli Yang. Overconstraint analysis on spatial 6-link loops. *Mechanism and machine theory*, 37 :267–278, 2002.



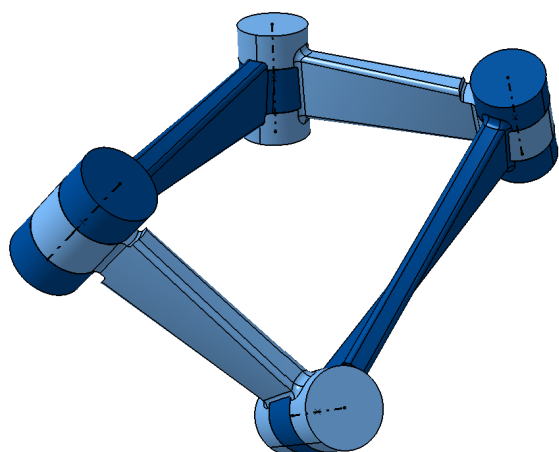


Chapitre 4

Résolution formelle avec les bases de Gröbner

Sommaire

4.1	Introduction	120
4.2	Résolution symbolique d'un système de solides	121
4.2.1	Conditions d'assemblage	127
4.2.2	Conditions de mobilité	137
4.3	Validations	144
4.3.1	Résolution formelle et interprétation géométrique	144
4.3.2	Validation numérique	146
4.4	Application des équations d'assemblage	148
4.5	Conclusion	149
	Bibliographie du chapitre 4	150



Résumé

Ce chapitre présente l'obtention des conditions d'assemblage et de mobilité pour la famille des systèmes de solides 4R. Les approches peuvent être utilisées pour d'autres système de solides. Appliquées au système 4R, 5 équations de mobilité et 2 équations d'assemblage sont trouvées grâce à l'algorithme de J.-C. Faugère qui permet de calculer des bases de Gröbner. Ces équations sont symboliques. Elles sont validées sur plusieurs configurations numériques.

4.1 Introduction

Dans le chapitre précédent, deux méthodes de mise en équations pour modéliser les systèmes de solides ont été présentées. Les équations de fermeture, obtenues avec les deux mises en équations, ont été résolues manuellement pour la mobilité du 4R en 3D. Les solutions obtenues sont symboliques.

L'objectif de ce chapitre est de proposer une méthode de génération automatique des conditions symboliques d'assemblage et de mobilité.

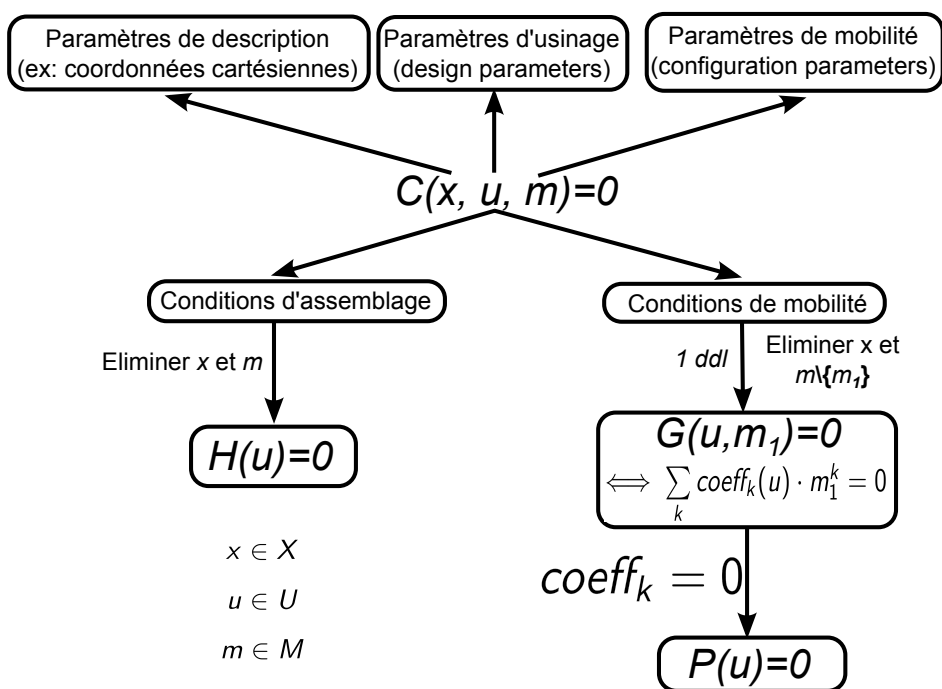


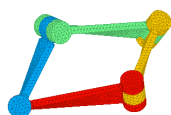
FIGURE 4.1 – Méthodes génériques pour générer les conditions d'existence qui ne dépendent que de u

Rappelons que $U \subset \mathbb{R}^n$ est l'espace des paramètres d'usage¹; $M \subset \mathbb{R}^m$ est l'espace des paramètres de mobilité; et $X \subset \mathbb{R}^p$ est l'espace des paramètres de description. La figure 4.1 montre les méthodes génériques pour obtenir formellement les conditions d'assemblage et de mobilité. Pour générer les conditions d'assemblage H en fonction de u , à partir des équations de fermeture $C(x, u, m) = 0$, il faut éliminer x et m . Après une étude bibliographique, le calcul de la base de Gröbner est retenu pour réaliser cette élimination.

La technique qui permet d'obtenir les conditions de mobilité en fonction de u est analogue à celle utilisée au chapitre 3. Dans le cas d'un unique degré de liberté, elle consiste à éliminer x et m à l'exception d'un paramètre de mobilité, m_1 par exemple. L'idée de cette proposition est de construire un nouveau système univarié, fonction d'une variable de mobilité², pour ensuite imposer que ce système soit identiquement nul.

1. ou paramètres dimensionnels

2. on considère que le paramètre de mobilité après l'élimination est une variable.



Pour avoir un système à 1 *ddl*, il faut que le nouveau système d'équations $G(u, m_1) = 0$ soit toujours vérifié quelque soit m_1 . Avec le calcul des bases de Gröbner, nous obtenons cette équation sous une forme polynomiale telle que $\sum_k \text{coeff}_k(u) \cdot m_1^k = 0$, où les facteurs $\text{coeff}_k(u)$ sont fonction des paramètres d'usinage u . Alors pour que le système soit mobile, il faut que tout ces coefficients soient nuls.

Cette technique est une des possibilités pour l'obtention des conditions de mobilité. Il en existe d'autres comme celle utilisée lors du chapitre 2. Elles ne sont pas toutes présentées dans ce rapport.

Les équations sont symboliques tout au long de la procédure ; nous avons besoin d'un outil efficace et pratique pour le calcul, surtout pour l'élimination des paramètres de mobilité et des paramètres de description. Maple[®] est un logiciel propriétaire de calcul symbolique développé par Maplesoft. Ce logiciel a été utilisé dans cette thèse en raison des packages préexistants. Le package [*Groebner*] est directement intégré dans Maple. En outre, M. Faugère a développé un package appelé "FGb" qui permet aussi de faire le calcul des bases de Gröbner. Avec la collaboration de M. Faugère, nous avons utilisé le package "FGb" pour l'obtention des conditions d'assemblage et de mobilité.

Plusieurs algorithmes existent pour calculer les bases de Gröbner : Buchberger, F4, F5, etc. La présentation de ces algorithmes se trouve dans l'annexe B. A noter que le package "FGb" est basé sur l'algorithme F4.

Les algorithmes de calcul symbolique que nous proposons pour obtenir des conditions d'existence sont explorés successivement. D'abord, nous allons reprendre l'analyse de l'exemple 2D à 5 barres plan présenté dans l'introduction. Quelques notions de bases sur l'utilisation du package "FGb" sont données. Ensuite, les méthodes génériques pour obtenir les conditions d'assemblage et de mobilité seront explorées. Ces méthodes sont appliquées au système de solides 4R. Puis, les deux types de conditions d'existence sont vérifiés avec des modèles numériques dans CATIA. Enfin, nous montrerons une application pratique des conditions d'assemblage et de mobilité.

4.2 Résolution symbolique d'un système de solides

Reprenons l'exemple du système plan à 5 barres (Figure 4.2) présenté dans l'introduction.

Au début du mémoire, nous nous demandons : quelles sont les conditions pour que ce système s'assemble ? Quelles sont les conditions pour qu'il soit mobile ?

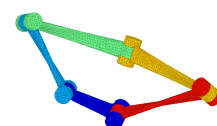
Le système d'équations qui caractérise le système de la figure 4.2 est celui déjà présenté dans l'introduction :

$$\begin{cases} (s - a \cos \alpha)^2 + (t - a \sin \alpha)^2 - b^2 = 0 \\ (s - d)^2 + t^2 - c^2 = 0 \\ (a \cos \alpha + s - d)^2 + (a \sin \alpha + t)^2 - 4e^2 = 0 \end{cases} \quad (4.1)$$

Le système d'équations (4.1) peut s'écrire sous la forme

$$C(x, u, m) = 0$$

Avec



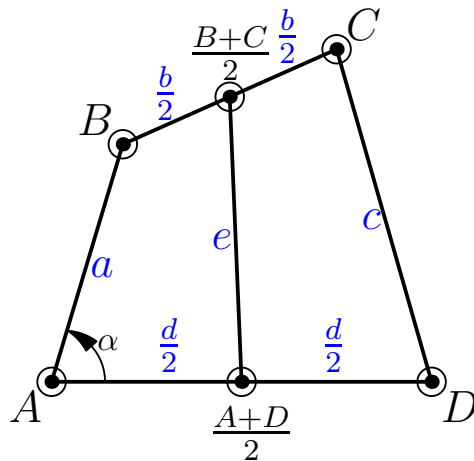


FIGURE 4.2 – Le système mécanique plan sur-contraint composé de 5 barres

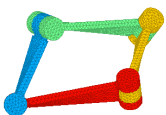
- $x : \{s, t\}$ des paramètres de description ;
- $u : \{a, b, c, d, e\}$ des paramètres d’usinage ;
- $m : \{\alpha\}$ un paramètre de mobilité.

Le calcul des bases de Gröbner est basé sur les notions et les théorèmes des idéaux. Un idéal est engendré par un polynôme ou un ensemble de polynômes dans l’anneau des polynômes. Ainsi, les objets traités dans le calcul des bases de Gröbner sont toujours des polynômes. Pour cette raison, nous devons transformer l’équation (4.1) en système polynômial (4.2). L’équation (4.2) est la réécriture de l’équation (4.1) quand on remplace $\sin \alpha$ et $\cos \alpha$ par su et cu , respectivement. A cause de cette substitution, une relation supplémentaire $su^2 + cu^2 - 1 = 0$, doit être ajoutée au système (4.2) car elle exprime le lien entre su et cu . Ainsi, l’équation (4.2) est le système polynômial qui caractérise la figure 4.2.

$$\begin{cases} (s - a \cdot cu)^2 + (t - a \cdot su)^2 - b^2 = 0 \\ (s - d)^2 + t^2 - c^2 = 0 \\ (a \cdot cu + s - d)^2 + (a \cdot su + t)^2 - 4e^2 = 0 \\ su^2 + cu^2 - 1 = 0 \end{cases} \quad (4.2)$$

Si nous voulons assembler le système de solides de la figure 4.2, il faut que le système d’équations (4.2) admette une solution pour un jeu de paramètres donné. Un de nos objectifs est d’obtenir les relations d’assemblage qui dépendent des paramètres d’usinage. Ces relations forment les conditions d’assemblage. Elles établissent les relations que doivent respecter les dimensions des pièces pour qu’elles puissent s’assembler. Pour cela, il faut éliminer les paramètres de description et les paramètres de mobilité. Pour que le système de solides soit mobile avec 1 *ddl* (cu par exemple), il faut que le système s’assemble quelque soit cu . C’est-à-dire qu’il faut éliminer s , t et su^3 .

3. Nous pouvons choisir n’importe quel paramètre de mobilité. Pour cet exemple, nous avons choisi de garder cu .



Mais comment vérifier que le système d'équation (4.1) possède une solution ? Comment éliminer les (su, s, t) pour obtenir les relations en fonction de (cu, a, b, c, d, e) ? Une réponse est donnée par les bases de Gröbner.

Le package développé par M. Faugère de Paris VI pour calculer les bases de Gröbner est FGb ; l'instruction Maple pour appeler ce package est :

```
> with(FGb);
[ModuleUnload, cpu, date, estime_size, fgb_debug, fgb_exit, fgb_gbasis, fgb_gbasis_elim,
fgb_gbasis_lm, fgb_hilbert, fgb_interface, fgb_minpoly_sqfr, fgb_nop, fgb_normalForm,
fgb_radical2, fgb_radical2_quiet, fgb_rr, fgb_rr_crit, fgb_rr_sqfr, pack_fgb_init]
```

La commande Maple permettant de trouver un ou des polynômes dans la base de Gröbner, qui ne font intervenir que les paramètres d'usinage, est la suivante :

```
> BasisFGb_elim := fgb_gbasis_elim(C, 0, [su, cu, s, t], [a, b, c, d, e]);
BasisFGb_elim := [ ]
```

avec

- C est la liste des polynômes. Dans le cas précédent, C est la liste d'équations (4.2).
- le premier paquet $[su, cu, s, t]$ est la liste de variables que l'on souhaite éliminer.
- le deuxième paquet $[a, b, c, d, e]$ est la liste de variables que l'on souhaite garder dans la base.

Le résultat “[]” est un ensemble vide. Cela signifie qu'il n'y a aucune relation nécessaire entre les paramètres d'usinage pour que le mécanisme s'assemble. C'est-à-dire que si un ingénieur prend 5 barres quelconques, il existe toujours une position pour que ces 5 barres s'assemblent⁴. En général, l'assemblage obtenu est rigide.

Pour que le système soit mobile avec 1 ddl , il faut que les 5 barres s'assemblent quelque soit cu :

```
> fgb_gbasis_elim(C, 0, [su, s, t], [cu, a, b, c, d, e]) : BasisElMo := collect(%, cu);
```

```
BasisElMo = (8 a^4 d^2 + 8 a^2 c^2 d^2 - 16 a^2 d^2 e^2) cu^2 + (8 a d^3 e^2 - 32 a d e^4 - 4 a c^4 d -
6 a^3 d^3 - 2 a c^2 d^3 + 40 a^3 d e^2 - 8 a b^2 d e^2 + 24 a c^2 d e^2 + 6 a^3 b^2 d - 16 a^3 c^2 d + 2 a b^2 c^2 d -
12 a^5 d) cu - 16 a^4 e^2 + 8 a^2 b^2 e^2 - 16 a^2 d^2 e^2 + 16 a^2 e^4 - 4 a^4 b^2 + a^2 b^4 + 8 a^4 c^2 - 4 a^2 b^2 c^2 +
4 a^2 c^4 + 5 a^4 d^2 - 2 a^2 b^2 d^2 + 2 a^2 c^2 d^2 + a^2 d^4 - 8 c^2 d^2 e^2 + 16 d^2 e^4 - 16 a^2 c^2 e^2 + c^4 d^2 + 4 a^6
```

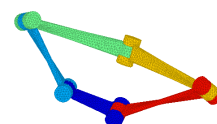
Le résultat (texte en bleu) est un polynôme. “collect(A, B)” est une fonction de factorisation. collect($\%, cu$) retourne un résultat factorisé en fonction de la variable de mobilité cu . $\%$ représente le résultat de la commande précédente. Dans cet exemple, $\%$ est le résultat de la fonction “fgb_gbasis_elim”. La factorisation est de la forme :

$$\text{Coeff}_2 cu^2 + \text{Coeff}_1 cu + \text{Coeff}_0 = 0 \quad (4.3)$$

L'équation (4.3) doit être respectée quelque soit cu . Ceci implique que les coefficients Coeff_2 , Coeff_1 , Coeff_0 soient tous nuls.

$$T = \begin{cases} \text{Coeff}_2 = 0 \\ \text{Coeff}_1 = 0 \\ \text{Coeff}_0 = 0 \end{cases} \quad (4.4)$$

4. la différence de la longueur de barres ne doit pas être trop importante. Par exemple, quand la longueur d'une barre est supérieure à la somme des quatre autres, ces 5 barres ne peuvent pas être assemblées.



Les 3 coefficients de cu sont :

$$\text{Coeff}_2 = 16 d^2 e^4 + 4 a^6 + c^4 d^2 - 8 c^2 d^2 e^2 - 4 a^4 b^2 + a^2 b^4 + 8 a^4 c^2 - 4 a^2 b^2 c^2 + 4 a^2 c^4 + 5 a^4 d^2 - 2 a^2 b^2 d^2 + 2 a^2 c^2 d^2 + a^2 d^4 - 16 a^4 e^2 + 8 a^2 b^2 e^2 - 16 a^2 c^2 e^2 - 16 a^2 d^2 e^2 + 16 a^2 e^4$$

$$\text{Coeff}_1 = 8 a d^3 e^2 - 32 a d e^4 + 2 a b^2 c^2 d - 12 a^5 d + 6 a^3 b^2 d - 16 a^3 c^2 d + 40 a^3 d e^2 - 4 a c^4 d - 6 a^3 d^3 - 2 a c^2 d^3 - 8 a b^2 d e^2 + 24 a c^2 d e^2$$

$$\text{Coeff}_0 = -16 a^2 d^2 e^2 + 8 a^4 d^2 + 8 a^2 c^2 d^2$$

Pour obtenir les conditions d'assemblage, nous calculons la base de Gröbner à partir de l'équation (4.4). La base de Gröbner du système (4.4) est :

> `fgb_gbasis(T, 0, [], [cu, a, b, c, d, e]) : BasisElMo2 := factor(%);`

$$\begin{aligned} \text{BasisElMo2} = & [-d^4 e^2 (b-d)(b+d)(c^2+3ae-4e^2)(-c^2+3ae+4e^2), d^3(4b^2c^6 - \\ & 48b^2c^4e^2 - 27d^2a^2c^2e^2 + 9d^2c^4e^2 + 108a^2b^2e^4 + 192b^2c^2e^4 - 54d^2a^2e^4 - 72d^2c^2e^4 - \\ & 256b^2e^6 + 144d^2e^6), d^3(c-e)(c+e)(c^6-12c^4e^2+27a^2e^4+48c^2e^4-64e^6), a d^3(9b^2c^4 - \\ & 16c^6 - 9c^4d^2 + 54a^2c^2e^2 - 45e^2b^2c^2 + 102c^4e^2 + 45c^2d^2e^2 + 36e^4b^2 - 156c^2e^4 - \\ & 36d^2e^4 + 16e^6), -d^4(-4c^6+27a^2c^2e^2+39c^4e^2-54a^2e^4-120c^2e^4+112e^6), a d^2(b^4c^2 - \\ & 2b^2c^4 + 9a^2c^2d^2 - 2d^2b^2c^2 - c^4d^2 + d^4c^2 - 4b^4e^2 + 4a^2c^2e^2 + 10e^2b^2c^2 + 4c^4e^2 + \\ & 18a^2d^2e^2 + 8b^2d^2e^2 + 14c^2d^2e^2 - 4d^4e^2 - 16a^2e^4 - 8e^4b^2 - 24c^2e^4 - 40d^2e^4 + \\ & 32e^6), a d^2(c-e)(2e+c)(-2e+c)(c+e)(2c^2-4e^2-b^2+d^2+2a^2), d^3(9a^2c^4+c^6 - \\ & 18a^2c^2e^2 - 12c^4e^2 + 36a^2e^4 + 48c^2e^4 - 64e^6), a d(6a^2b^2c^2 - 3b^4c^2 - 8a^2c^4 + 10b^2c^4 - \\ & 8c^6 - 33a^2c^2d^2 + 6d^2b^2c^2 - c^4d^2 - 3d^4c^2 - 24a^2b^2e^2 + 12b^4e^2 + 40a^2c^2e^2 - 56e^2b^2c^2 + \\ & 56c^4e^2 - 30a^2d^2e^2 - 24b^2d^2e^2 - 16c^2d^2e^2 + 12d^4e^2 - 32a^2e^4 + 64e^4b^2 - 112c^2e^4 + \\ & 80d^2e^4 + 64e^6), d^2(a^2b^4 - 2a^2b^2d^2 - 3a^2c^2d^2 + c^4d^2 + a^2d^4 - 6a^2d^2e^2 - 8c^2d^2e^2 + \\ & 16d^2e^4), a^2d^2(c-e)(c+e)(b-d)(b+d), d(6a^4b^2 - 3a^2b^4 - 8a^4c^2 + 10a^2b^2c^2 - 8a^2c^4 + \\ & 6a^2b^2d^2 + 5a^2c^2d^2 - 3c^4d^2 - 3a^2d^4 + 8a^4e^2 - 16a^2b^2e^2 + 24a^2c^2e^2 + 22a^2d^2e^2 + \\ & 24c^2d^2e^2 - 16a^2e^4 - 48d^2e^4), a d^2(3a^2b^2 - 2a^2c^2 + b^2c^2 - 2c^4 - 3a^2d^2 - c^2d^2 + 8a^2e^2 - \\ & 4b^2e^2 + 12c^2e^2 + 4d^2e^2 - 16e^4), 4a^6 - 4a^4b^2 + a^2b^4 + 8a^4c^2 - 4a^2b^2c^2 + 4a^2c^4 - \\ & 2a^2b^2d^2 - 3a^2c^2d^2 + c^4d^2 + a^2d^4 - 16a^4e^2 + 8a^2b^2e^2 - 16a^2c^2e^2 - 6a^2d^2e^2 - 8c^2d^2e^2 + \\ & 16a^2e^4 + 16d^2e^4, a d(3a^2+c^2-4e^2)(2c^2-4e^2-b^2+d^2+2a^2), a^2d^2(a^2+c^2-2e^2)] \end{aligned}$$

La base est constituée de 16 polynômes séparés par des virgules. Comparés aux conditions que nous avons montrées dans l'introduction du rapport ⁵, cette base obtenue n'est pas identique à l'expression désirée. Le résultat obtenu est beaucoup plus complexe car il contient toutes les solutions algébriques du problème géométrique posé. En fait, chaque solution peut être considérée comme une feuille de l'arbre dans la figure 4.3. Généralement, nous ne souhaitons pas obtenir toutes les solutions possibles mais uniquement celles qui permettent de fabriquer le 4R. C'est la raison pour laquelle nous proposons des outils pour simplifier les résultats.

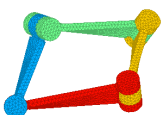
Nous introduisons pour cela le concept de *spécialisation*. Il y a deux types de spécialisation : *saturation* et *contrainte*.

◆ On appelle *saturation* l'ajout d'un polynôme supplémentaire en vue d'écartier des solutions indésirables.

⇨ Exemple de saturation : pour spécifier qu'une variable x ne peut pas être nulle, nous ajoutons $x \cdot v - 1 (= 0)$ à la fin de la liste polynomiale. En ajoutant cette saturation, nous devons éliminer le *cofacteur* v .

Cette opération revient à couper une branche d'un arbre (figure 4.3). Certaines solutions

5. Les conditions de mobilité du mécanisme plan surcontraint 5 barres sont : $a = e = c, b = d$.



indésirables peuvent ainsi être éliminées.



FIGURE 4.3 – L'ajout d'une saturation revient à couper une branche d'un arbre. Ceci permet d'éliminer les solutions indésirables comme les longueurs nulles par exemple.

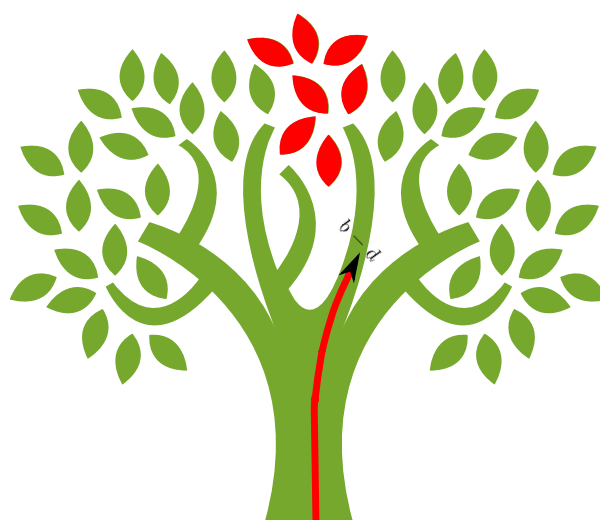


FIGURE 4.4 – L'ajout d'une contrainte revient à parcourir une branche. Ceci permet de parcourir des solutions particulières.

• On appelle *contrainte* l'ajout d'un polynôme supplémentaire en vue de particulariser le problème.

⇨ Exemple de contrainte : $b = d \implies b - d (= 0)$. L'ajout d'une contrainte n'engendre aucun paramètre supplémentaire à supprimer.

⇨ Dans cette thèse, nous proposons deux "astuces" pour ajouter des contraintes :

⇨ Ajouter le facteur qui apparaît le plus souvent en comptant son ordre de multiplicité ;

⇨ Ajouter le facteur du dernier polynôme résultat de la base de Gröbner.

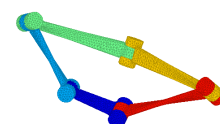
Prenons encore l'exemple de l'arbre. Le fait d'ajouter une contrainte revient à garder une branche de l'arbre et à couper toutes les autres (figure 4.4). Cette opération nous permet de parcourir certaines solutions qui semblent intéressantes.

Les saturations et les contraintes peuvent être utilisées séparément ou en même temps pendant le calcul. Mais il faut faire attention à l'utilisation des contraintes car en imposant le "chemin", nous risquons de perdre des informations potentiellement utiles sur les autres branches. Les contraintes seront utilisées dans les prochaines sections.

Pour l'exemple du mécanisme à 5 barres, il faut ajouter la saturation suivante car nous ne souhaitons pas étudier la situation où une (ou plusieurs) barre soit de longueur nulle. Ainsi, les longueurs des barres doivent être non nulles.

$$S = a \cdot b \cdot c \cdot d \cdot e \cdot v - 1 = 0 \quad (4.5)$$

Pour éliminer ces cas, il faut ajouter l'équation (4.5) à la liste précédente de polynômes puis éliminer le cofacteur v . Les cofacteurs sont similaires aux variables. Si un des a, b, c, d ou e est égale à 0, c par exemple, l'équation (4.5) donne en effet $0 = 1$, ce qui est impossible. Il n'existe



donc aucune solution pour le système d'équations. Après l'ajout de l'équation (4.5), la base de Gröbner du nouveau système est :

```
> fgb_gbasis_elim([op(Q), op(S)], 0, [v], [a, b, c, d, e]) : BasisElMo2 :=
factor(%); map(print, %)
```

$$\begin{aligned} & (b-d)^3 (b+d)^3 \\ & (c-e)^3 (c+e)^3 \\ & 4c^4 d^2 + b^4 e^2 - 2b^2 d^2 e^2 - 8c^2 d^2 e^2 + d^4 e^2 + 4d^2 e^4 \\ & (c-e)(c+e)(b-d)(b+d) \\ & -2e^2 + a^2 + c^2 \end{aligned}$$

L'ajout de cette saturation a permis de diminuer le nombre de polynômes de 16 à 5.

Nous allons maintenant ajouter des contraintes en appliquant les astuces proposées, les facteurs suivants sont les contraintes :

- le facteur $b-d$ est répété 4 fois (en comptant les multiplicités).
- le facteur $c-e$ est répété 4 fois (en comptant les multiplicités).

$b+d$ et $c+e$ sont également répétés 4 fois. Mais les longueurs ne peuvent pas être opposées car les longueurs négatives ne peuvent pas être fabriquées. En conséquence, ces deux facteurs sont plutôt des saturations. Ainsi, les nouvelles saturations et les contraintes sont :

- Saturations :
 - les longueurs de barres sont non nulles : $a \cdot b \cdot c \cdot d \cdot e \cdot v - 1 (= 0)$;
 - les longueurs de barres ne peuvent être opposées : $(b+d) \cdot m - 1 (= 0)$, $(c+e) \cdot n - 1 (= 0)$;
- Contraintes :
 - $b-d (= 0)$, $c-e (= 0)$.

Le nouveau système de spécialisation est

$$S = \begin{cases} a \cdot b \cdot c \cdot d \cdot e \cdot v - 1 (= 0) \\ (b+d) \cdot m - 1 (= 0) \\ (c+e) \cdot n - 1 (= 0) \\ b-d (= 0) \\ c-e (= 0) \end{cases} \quad (4.6)$$

Nous calculons la base de Gröbner de BasisElMo2 avec les nouvelles spécialisations (4.6).

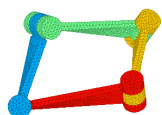
```
> BasisElMo3 := factor(fgb_gbasis_elim([op(BasisElMo2), op(S)], 0, [v,m,n], [a, b, c, d, e]))
```

$$\text{BasisElMo3} := [(a-e)(a+e), b-d, c-e]$$

L'ensemble des équations correspondant à la liste des polynômes est :

$$\begin{cases} (a-e)(a+e) = 0 \\ b-d = 0 \\ c-e = 0 \end{cases} \quad (4.7)$$

A partir de l'équation (4.7), il est aisé de conclure car $a+e$ ne peut pas être nul :



$$\begin{cases} c = e \\ b = d \\ a = c \end{cases} \quad (4.8)$$

L'équation (4.8) est la même condition de mobilité que nous avons obtenu dans l'introduction du mémoire.

Dans cet exemple, nous avons analysé et résolu la mise en équation $C(x, u, m) = 0$ (équation (4.1)). Dans la prochaine sous-section, nous allons exposer des méthodes génériques pour obtenir les conditions d'assemblage et de mobilité pour des systèmes de solides formant une boucle fermée. Ces méthodes sont appliquées au système d'équation $C(u, m) = 0$. La procédure est exprimée sous forme de texte. La génération des équations $C(x, u, m) = 0$ et $C(u, m) = 0$ est décrite au chapitre 3.

4.2.1 Conditions d'assemblage

4.2.1.1 Principe général

La méthode permettant d'obtenir les conditions d'assemblage est une des contributions la plus importante de cette thèse. L'application de cette méthode nous a permis d'obtenir des conditions d'assemblage de 4R 3D qui n'ont jamais été publiées auparavant. Dans cette partie, nous allons explorer comment générer les conditions d'assemblage pour le système de solides en boucle fermée. La méthode proposée est schématisée par la figure 4.5.

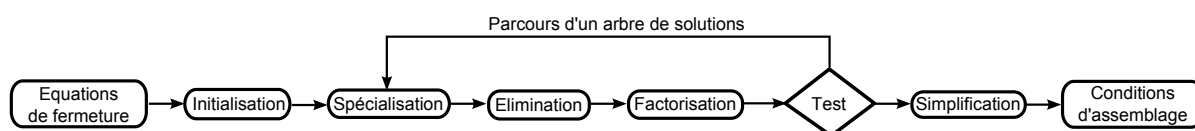


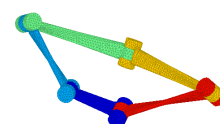
FIGURE 4.5 – Les étapes principales d'obtention des conditions d'assemblage

Elle est constituée de 5 étapes principales :

✎ **Initialisation** : le but de cette étape est de convertir le système d'équations initial en un système polynomial. Il est évoqué dans la partie précédente que le calcul des bases de Gröbner est basé sur les notions et les théorèmes des idéaux. Un idéal est engendré par un polynôme ou un ensemble de polynômes dans l'anneau des polynômes. Ainsi, les objets manipulés dans le calcul des bases de Gröbner sont toujours des polynômes.

✎ **Spécialisation** : dans cette phase, l'utilisateur spécifie des saturations et/ou des contraintes afin de parcourir des solutions spécifiques.

✎ **Elimination** : l'algorithme calcule les bases de Gröbner du système ou élimine les paramètres de mobilité spécifiés.



✎ **Factorisation** : cette étape facilite l'analyse du résultat. Elle permet notamment de trouver de nouvelles saturations et/ou contraintes.

✎ **Simplification** : l'objectif est de trouver une expression "simple" du résultat.

Les données d'entrée sont une liste de polynômes (système d'équations) et un ordre admissible des paramètres, ainsi qu'une liste de spécifications. Les spécifications ne sont pas obligatoires lors de la 1^{ère} élimination. Les inconnues du système sont les paramètres de mobilité. Les paramètres d'usinage sont les éléments à conserver. Dans un premier temps, nous éliminons toutes les inconnues (paramètres de mobilité). Si le résultat est non-vide, nous factorisons les polynômes obtenus et nous identifions des saturations et des contraintes afin d'éliminer les solutions inutiles. Lorsqu'aucune nouvelle spécification n'est ajoutée, nous passons à l'étape de simplification dans le but d'obtenir des conditions d'assemblage courtes et simples.

4.2.1.2 Description de l'algorithme

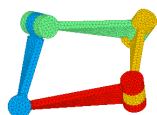
Avec la mise en équation proposée au chapitre 3, nous pouvons écrire les équations de fermeture d'un système de solides en boucle fermée en fonction de u et m . Les paramètres d'usinage comportent les angles d'usinage u_α et les longueurs d'usinage u_L ; les paramètres de mobilité comportent les angles de mobilité m_θ et les translations de mobilité m_L . Dans un premier temps, les équations de fermeture sont transformées en un système polynômial. C'est-à-dire que $\sin(u_\alpha)$, $\cos(u_\alpha)$, $\sin(u_\theta)$, $\cos(u_\theta)$ sont remplacés par $s\alpha$, $c\alpha$, $s\theta$ et $c\theta$, respectivement. L'ensemble des équations P et Q sont des relations supplémentaires qui lient $(s\theta, c\theta)$ et $(s\alpha, c\alpha)$ tel que $s\theta^2 + c\theta^2 = 1$ et $s\alpha^2 + c\alpha^2 = 1$, respectivement. Ensuite, nous éliminons les variables $s\theta$ et $c\theta$ après avoir ajouté au système d'équations d'éventuelles spécialisations. Quelque soit le nombre d'itération du sous-processus factoriser \rightarrow spécialiser \rightarrow éliminer \rightarrow factoriser \rightarrow spécialiser, un résultat simplifié $H(u_\alpha, u_L)$ est souhaitable. Ce processus fonctionne bien quand le système étudié est simple.

Cependant, le calcul direct d'élimination peut être mis en échec pour un exemple compliqué tel que le mécanisme de Bennett. Nous proposons donc des étapes supplémentaires pour résoudre un système plus compliqué. Ces étapes sont illustrées par la figure 4.7. Au lieu d'ajouter toutes les relations $P(s\theta, c\theta) = 0$ et $Q(s\alpha, c\alpha) = 0$ au départ de l'algorithme, nous n'ajoutons que $P(s\theta, c\theta) = 0$. Dans ce cas, l'élimination est plus rapide. Ensuite, nous injectons $Q(s\alpha, c\alpha) = 0$ dans le résultat intermédiaire et calculons la base de Gröbner du nouveau système (complet). De cette manière, nous obtenons la même base de Gröbner $\tilde{G}(s\alpha, c\alpha, u_L)$ que celle de la figure 4.6. Ces idées sont présentées au §4.2.1.4 avec un système de solides 4R. On s'intéresse ensuite à la simplification.

4.2.1.3 Etape de simplification

Cette étape comporte deux opérations :

- Réduction du nombre de polynômes : le but est de trouver le plus petit ensemble de polynômes générateurs du système obtenu afin de simplifier les résultats obtenus. Cette opération est automatisée.



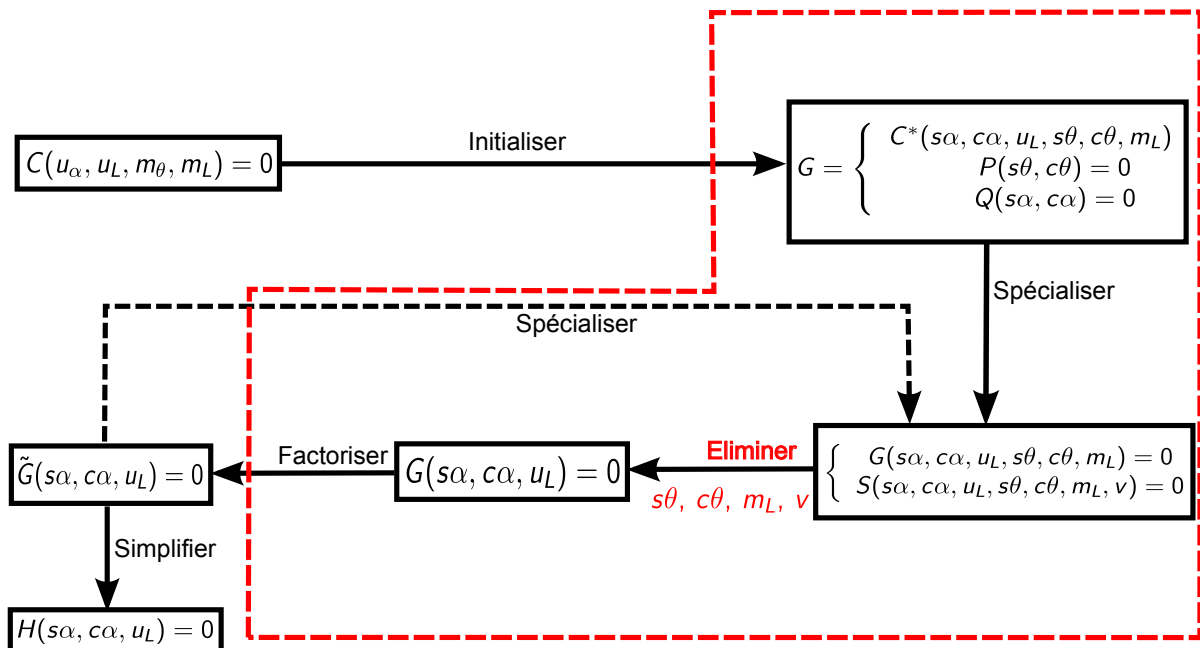


FIGURE 4.6 – L'obtention des conditions d'assemblage dans un cas idéal

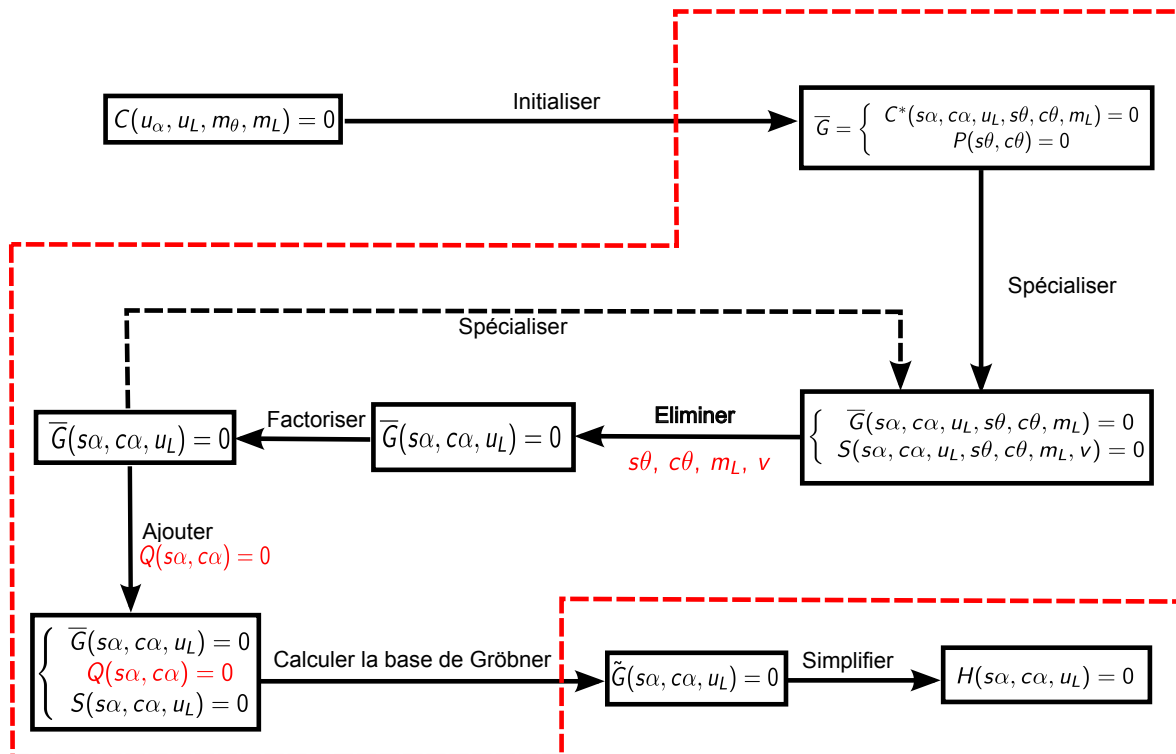
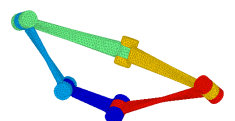


FIGURE 4.7 – L'obtention des conditions de mobilité dans un cas relativement complexe



➤ Simplification trigonométrique : le but est d'utiliser les relations trigonométriques pour rendre les équations plus courtes. C'est une opération manuelle utilisant Maple si nous voulons obtenir des relations optimisées⁶.

L'objectif de la 1^{ère} opération est illustré par la figure 4.8. Les n polynômes de la base de Gröbner représentent un idéal. Existe-il un autre ensemble de p polynômes générateurs, avec p inférieur à n , qui représente le même idéal ? La réponse est à nouveau obtenue par le calcul des bases de Gröbner. On cherche ici le plus petit ensemble de polynômes générateurs qui représente le même idéal que celui généré par les n équations.

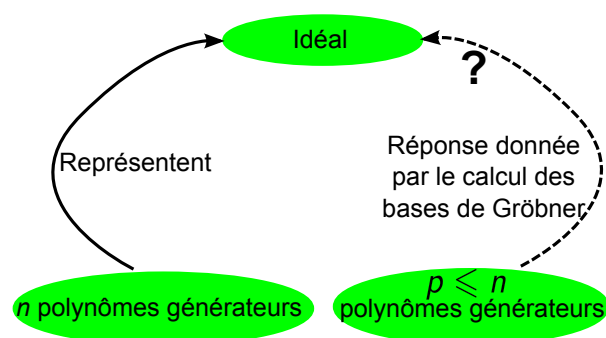


FIGURE 4.8 – Recherche d'un ensemble de polynômes générateurs plus petits qui représente le même idéal que les n polynômes.

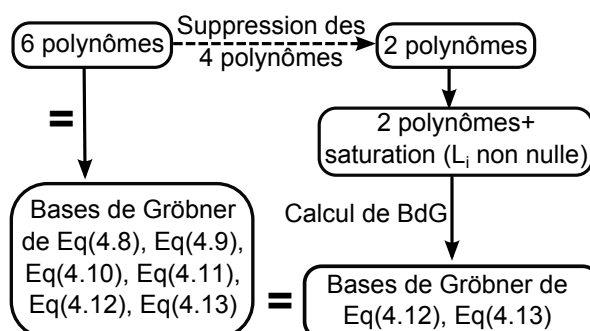


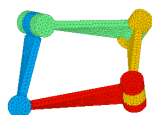
FIGURE 4.9 – La technique pour minimiser le nombre de polynômes générateurs : comparer les 2 bases de Gröbner.

Le principe de l'algorithme est donné dans la figure 4.9. Les données d'entrée sont les n polynômes. D'abord, il faut calculer la base de Gröbner des n polynômes. Puis nous supprimons le premier qui apparaît dans cette base et nous calculons la nouvelle base de Gröbner de ces $n - 1$ polynômes plus la saturation. Si les 2 bases de Gröbner ne sont pas identiques, cela veut dire qu'il y a eu une perte d'information, auquel cas le polynôme supprimé est important. Donc, il ne faut pas le supprimer. Si les 2 bases de Gröbner sont identiques, cela veut dire qu'il n'y a pas eu de perte d'information. Dans ce cas, nous gardons ces $n - 1$ polynômes. Ensuite, nous recommençons la même procédure avec les polynômes restants en supprimant un autre jusqu'à les avoir tous testés.

4.2.1.4 Application du système 4 barres

Nous avons appliqué notre méthode au système 4R. Nous allons montrer comment générer les équations d'assemblage à partir de la mise en équation de la variante de la méthode Jin et Yang. Nous avons vu dans la section 3.4.3 du chapitre 3 que cette méthode fournit les 4 équations suivantes :

6. On souhaite avoir des relations courtes, simples.



$$C(u_\alpha, u_L, m_\theta) = \begin{cases} \cos \alpha_2 \cdot \cos \alpha_3 - \sin \alpha_3 \cdot \sin \alpha_2 \cdot \cos \theta_C - \cos \alpha_4 \cdot \cos \alpha_1 + \sin \alpha_1 \cdot \sin \alpha_4 \cdot \cos \theta_A = 0 \\ L4^2 + 2 \cdot L4 \cdot L1 \cdot \cos \theta_A + L1^2 - L2^2 - 2 \cdot L2 \cdot L3 \cdot \cos \theta_C - L3^2 = 0 \\ L3 \cdot \sin \alpha_2 \cdot \sin \theta_C + L4 \cdot \sin \theta_A \cdot \sin \alpha_1 = 0 \\ L2 \cdot \sin \theta_C \cdot \sin \alpha_3 + L1 \cdot \sin \alpha_4 \cdot \sin \theta_A = 0 \end{cases} \quad (4.9)$$

Nous appliquons l'algorithme présenté à la figure 4.7 car, celui décrit à la figure 4.6 n'a pas permis d'atteindre le résultat sur une machine standard.

☞ Initialisation :

Les variables $\sin(\alpha_i)$, $\cos(\alpha_i)$, $\sin(\theta_i)$ et $\cos(\theta_i)$ sont remplacées par les symboles si et ci .

$$\begin{cases} \sin(\alpha_1) = s1 \\ \cos(\alpha_1) = c1 \\ \sin(\alpha_2) = s2 \\ \cos(\alpha_2) = c2 \end{cases} \quad \begin{cases} \sin(\alpha_3) = s3 \\ \cos(\alpha_3) = c3 \\ \sin(\alpha_4) = s4 \\ \cos(\alpha_4) = c4 \end{cases} \quad \begin{cases} \sin(\theta_A) = sA \\ \cos(\theta_A) = cA \\ \sin(\theta_C) = sC \\ \cos(\theta_C) = cC \end{cases}$$

Puis, les relations qui lient (sA, cA) et (sC, cC) sont ajoutées. Ainsi, le système d'équation $\overline{G}(s1, s2, s3, s4, c1, c2, c3, c4, L1, L2, L3, L4, sA, cA, sC, cC)$ s'écrit :

$$\overline{G} = \begin{cases} c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA = 0 \\ L4^2 + 2 L4 L1 cA + L1^2 - L2^2 - L3^2 - 2 L2 L3 cC = 0 \\ L3 s2 sC + L4 sA s1 = 0 \\ L2 sC s3 + L1 s4 sA = 0 \\ cC^2 + sC^2 - 1 = 0 \\ cA^2 + sA^2 - 1 = 0 \end{cases}$$

☞ 1^{ère} élimination :

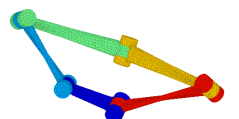
Dans la 1^{ère} élimination, nous n'imposons aucune spécialisation. Pour obtenir les conditions d'assemblage qui ne dépendent que des paramètres d'usinage, nous allons d'abord éliminer les 4 inconnues : sC, cC, sA, cA . Le résultat de l'élimination fournit une solution comportant 80 polynômes parmi lesquels beaucoup sont triviaux.

☞ Factorisation et spécialisation :

Après la factorisation, nous avons identifié des spécialisations permettant de parcourir l'arbre des solutions.

☞ Saturation : $L_i \neq 0$, les longueurs de barres ne peuvent pas être nulles ;

☞ Contrainte : après la factorisation, le facteur $s4 s2 L1 L3 - s3 s1 L2 L4$ a attiré notre attention car il apparaît 16 fois sous forme de facteur dans les 80 polynômes. Selon la 1^{ère} astuce de l'ajout de contraintes, ce facteur peut devenir une spécification. Nous allons l'étudier en priorité. La situation $s4 s2 L1 L3 - s3 s1 L2 L4 \neq 0$ sera discutée dans § 4.2.1.5. Remarque : ce facteur avait déjà été mis en évidence au cours de la résolution manuelle du Chapitre 3.



☞ 2^{ème} élimination :

Après la 2^{ème} élimination, nous obtenons 6 équations. Ces équations ne dépendent que des paramètres d'usinage et s'écrivent :

$$L1 L2^2 L3 L4 (-4 c4^2 c1^2 + 8 c3 c2 c4 c1 + 4 s3^2 s2^2 - 4 c3^2 c2^2 + 4 s4^2 s1^2 - 2 s4^2 s2^2) + s4^2 s2^2 L1 L3 L4 (L4^2 - 2 L3^2 - 2 L1^2) + s4 s3 s2 s1 L2 (L2^4 - 2 L2^2 L1^2 + L3^4 - 2 L3^2 L2^2 + L1^4 - 2 L3^2 L1^2) = 0 \quad (4.10)$$

$$L3^2 L2 L1^2 (-4 c4^2 c1^2 + 8 c3 c2 c4 c1 - 2 s4^2 s2^2 + 4 s3^2 s2^2 - 4 c3^2 c2^2 + 4 s4^2 s1^2) + s3^2 s1^2 L2 (L4^2 - 2 L3^2 - 2 L1^2) + s4 s3 s2 s1 L4 L3 L1 (L2^4 - 2 L2^2 L1^2 + L3^4 - 2 L3^2 L2^2 + L1^4 - 2 L3^2 L1^2) = 0 \quad (4.11)$$

$$L4^2 L2^2 (-4 c4^2 c1^2 + 8 c3 c2 c4 c1 - 2 s4^2 s2^2 + 4 s3^2 s2^2 - 4 c3^2 c2^2 + 4 s4^2 s1^2) + s4^2 s2^2 (L4^4 - 2 L4^2 L3^2 - 2 L2^2 L1^2 + L2^4 - 2 L4^2 L1^2 + L3^4 - 2 L3^2 L2^2 + L1^4 - 2 L3^2 L1^2) = 0 \quad (4.12)$$

$$L4 L3 L2 L1 (-4 c4^2 c1^2 + 8 c3 c2 c4 c1 - 2 s4^2 s2^2 + 4 s3^2 s2^2 - 4 c3^2 c2^2 + 4 s4^2 s1^2) + s4 s3 s2 s1 (L4^4 - 2 L4^2 L3^2 - 2 L2^2 L1^2 + L2^4 - 2 L4^2 L1^2 + L3^4 - 2 L3^2 L2^2 + L1^4 - 2 L3^2 L1^2) = 0 \quad (4.13)$$

$$L3^2 L1^2 (-4 c4^2 c1^2 + 8 c3 c2 c4 c1 - 2 s4^2 s2^2 + 4 s3^2 s2^2 - 4 c3^2 c2^2 + 4 s4^2 s1^2) + s3^2 s1^1 (L4^4 - 2 L4^2 L3^2 - 2 L2^2 L1^2 + L2^4 - 2 L4^2 L1^2 + L3^4 - 2 L3^2 L2^2 + L1^4 - 2 L3^2 L1^2) = 0 \quad (4.14)$$

$$s3 s1 L4 L2 - s4 s2 L3 L1 = 0 \quad (4.15)$$

Parmi les 6 polynômes, aucune spécialisation n'est identifiée. Nous passons donc à l'étape suivante.

☞ Ajout de $Q(s\alpha, c\alpha) = 0$:

Ensuite, nous calculons la base de Gröbner de l'ensemble des 10 polynômes formés par les 6 précédents polynômes additionnés des 4 polynômes supplémentaires :

$$\triangleleft s1^2 + c1^2 - 1 = 0 \quad (\sin^2 \alpha_1 + \cos^2 \alpha_1 = 1)$$

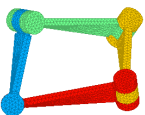
$$\triangleleft s2^2 + c2^2 - 1 = 0 \quad (\sin^2 \alpha_2 + \cos^2 \alpha_2 = 1)$$

$$\triangleleft s3^2 + c3^2 - 1 = 0 \quad (\sin^2 \alpha_3 + \cos^2 \alpha_3 = 1)$$

$$\triangleleft s4^2 + c4^2 - 1 = 0 \quad (\sin^2 \alpha_4 + \cos^2 \alpha_4 = 1)$$

☞ Calcul de la base de Gröbner :

Le calcul renvoie 32 polynômes, parmi lesquels les 4 derniers sont les polynômes que nous avons ajoutés à l'instant. Ceci veut dire que si ces 4 polynômes sont ajoutés dès le début dans



le système de départ, nous devons théoriquement obtenir 32 polynômes. Cette technique est applicable sur l'exemple du 4R parce que nous n'éliminons pas les $\sin \alpha_i$ et $\cos \alpha_i$.

☞ Simplification - réduction du nombre de polynômes :

Les 32 équations obtenues ne peuvent plus être factorisées sous la forme $A \cdot B \cdot C = 0$, ainsi, ni saturation, ni contrainte ne peuvent être ajoutées pour trier les solutions. Donc, nous passons à l'étape de simplification.

Cette étape a permis de réduire le nombre d'équations à 6. Les 4 derniers correspondent aux relations entre les $\sin \alpha_i$ et $\cos \alpha_i$.

$$\begin{aligned}
 & -2 c_2^2 L_3^2 L_1^2 - 2 c_3^2 L_3^2 L_1^2 + L_4^4 - 2 L_4^2 L_3^2 + 8 c_4 c_3 c_2 c_1 L_3^2 L_1^2 - 2 c_3^2 c_1^2 L_3^2 L_2^2 + \\
 & c_3^2 c_1^2 L_2^4 - 2 c_3^2 c_1^2 L_4^2 L_1^2 - 2 c_3^2 c_1^2 L_3^2 L_1^2 - 2 c_3^2 c_1^2 L_2^2 L_1^2 + c_3^2 c_1^2 L_1^4 - c_1^2 L_4^4 - \\
 & c_3^2 L_4^4 + 2 c_3^2 L_4^2 L_3^2 + 2 c_1^2 L_4^2 L_3^2 - c_3^2 L_3^4 + 2 c_3^2 L_3^2 L_2^2 + 2 c_1^2 L_3^2 L_2^2 - c_3^2 L_2^4 + \\
 & 2 c_3^2 L_4^2 L_1^2 + 2 c_1^2 L_4^2 L_1^2 + 2 c_3^2 L_2^2 L_1^2 + 2 c_1^2 L_2^2 L_1^2 - c_3^2 L_1^4 - 2 c_4^2 c_2^2 L_3^2 L_1^2 + \\
 & c_3^2 c_1^2 L_4^4 - 2 c_3^2 c_1^2 L_4^2 L_3^2 + c_3^2 c_1^2 L_3^4 - c_1^2 L_3^4 - c_1^2 L_2^4 - c_1^2 L_1^4 - 2 c_4^2 L_3^2 L_1^2 - 2 c_1^2 L_3^2 L_1^2 + \\
 & L_1^4 + 4 L_3^2 L_1^2 - 2 L_3^2 L_2^2 - 2 L_4^2 L_1^2 - 2 L_2^2 L_1^2 + L_2^4 + L_3^4 = 0 \quad (4.16)
 \end{aligned}$$

$$s_3 s_1 L_4 L_2 - s_4 s_2 L_3 L_1 = 0 \quad (4.17)$$

$$s_1^2 + c_1^2 - 1 = 0 \quad (4.18)$$

$$s_2^2 + c_2^2 - 1 = 0 \quad (4.19)$$

$$s_3^2 + c_3^2 - 1 = 0 \quad (4.20)$$

$$s_4^2 + c_4^2 - 1 = 0 \quad (4.21)$$

Si nous calculons la base de Gröbner des équations (4.16) - (4.21) en mettant le même ordre monomial et la saturation, nous retrouvons les 32 équations. Ceci prouve non seulement qu'il n'y a pas eu de perte d'information en supprimant les équations algébriquement dépendantes, mais aussi que le plus petit ensemble de générateurs qui représentent le même idéal contient deux (ou 6 avec les relations trigonométriques) polynômes.

☞ Simplification trigonométrique

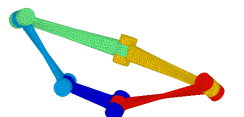
Après un traitement algébrique des équations (4.16) et (4.17) à l'aide des relations trigonométriques, nous obtenons finalement les 2 conditions d'assemblage réécrites en fonction des tangentes :

$$\left\{ \begin{array}{l} Eq.(4.16) \\ Eq.(4.17) \end{array} \right\} \implies (L_2^2 - L_1^2 - L_3^2 + L_4^2)^2 - \left(\frac{2 L_2 L_4 c_2 c_4}{s_2 s_4} - \frac{2 L_1 L_3 c_1 c_3}{s_1 s_3} \right)^2 = 0$$

$$\xrightarrow[s_i = \sin(a_i)]{c_i = \cos(a_i)} (L_2^2 - L_1^2 - L_3^2 + L_4^2)^2 - \left(\frac{2 L_2 L_4 \cos(a_2) \cos(a_4)}{\sin(a_2) \sin(a_4)} - \frac{2 L_1 L_3 \cos(a_1) \cos(a_3)}{\sin(a_1) \sin(a_3)} \right)^2 = 0$$

$$\implies (L_2^2 - L_1^2 - L_3^2 + L_4^2)^2 - \left(\frac{2 L_2 L_4}{\tan(a_2) \tan(a_4)} - \frac{2 L_1 L_3}{\tan(a_1) \tan(a_3)} \right)^2 = 0 \quad (4.22)$$

Ce développement se trouve dans l'annexe C.



Puisque les équations (4.16) et (4.17) impliquent l'équation (4.22), nous gardons finalement les deux équations les plus courtes. Ces deux relations constituent les conditions d'assemblage. Le système d'équation (4.23) est donc la condition d'assemblage de la classe du système de solides $4R$ en 3D.

$$\left\{ \begin{array}{l} (L_2^2 - L_1^2 - L_3^2 + L_4^2)^2 = \left(\frac{2 L_2 L_4}{\tan(a_2) \tan(a_4)} - \frac{2 L_1 L_3}{\tan(a_1) \tan(a_3)} \right)^2 \\ \frac{L_2 L_4}{\sin(a_2) \sin(a_4)} = \frac{L_1 L_3}{\sin(a_1) \sin(a_3)} \end{array} \right. \quad (4.23)$$

A noter que les angles a_i du système d'équations (4.23) doivent être différents de 0. Cela exclut les systèmes plan.

4.2.1.5 L'étude de la spécialisation $s_4 s_2 L_1 L_3 - s_3 s_1 L_2 L_4 \neq 0$

L'ensemble des solutions obtenues avec la contrainte $s_4 s_2 L_1 L_3 - s_3 s_1 L_2 L_4 (= 0)$ est discuté dans § 4.2.1.4. Dans cette section, nous allons explorer les situations où $s_4 s_2 L_1 L_3 - s_3 s_1 L_2 L_4 \neq 0$. Ces développements sont fournis dans l'annexe C.

☞ Spécialisations :

– Saturations :

1. $L_i \neq 0$, les longueurs de barres ne peuvent pas être nulles ;
2. $s_4 s_2 L_1 L_3 - s_3 s_1 L_2 L_4 \neq 0$.

☞ 2^{ème} élimination :

Après la 1^{ère} élimination, le calcul de la base de Gröbner donne 80 polynômes. Dans la 2^{ème} élimination, nous appliquons les saturations ci-dessus sur les 80 polynômes. Le résultat de l'élimination fournit une solution comportant 17 polynômes parmi lesquels beaucoup peuvent être factorisés.

☞ Identification de la saturation : $L_1 + L_2 + L_3 + L_4 \neq 0$

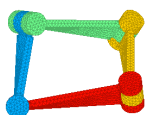
Dans le 16^{ème} polynôme, le facteur $(L_1 + L_2 + L_3 + L_4)$ apparaît. Ce facteur nous oriente vers des solutions telle que la somme des longueurs de barres est nulle. Cet ensemble de solutions n'est pas désirables. Nous choisissons d'ajouter $L_1 + L_2 + L_3 + L_4 \neq 0$ comme nouvelle saturation.

Liste de saturations :

1. $L_i \neq 0$, les longueurs de barres ne peuvent pas être nulles ;
2. $s_4 s_2 L_1 L_3 - s_3 s_1 L_2 L_4 \neq 0$.
3. $L_1 + L_2 + L_3 + L_4 \neq 0$

☞ 3^{ème} élimination :

Après la 3^{ème} élimination, le calcul renvoie 18 polynômes. Le 18^{ème} polynôme contient 7 facteurs.



$$(L2 + L1 + L3 - L4) (-L2 + L1 - L3 - L4) (L4 + L1 - L3 - L2) (L2 + L1 - L3 + L4) \\ (L2 + L1 - L3 - L4) (-L2 + L1 + L3 - L4) (-L2 + L1 + L3 + L4) = 0 \quad (4.24)$$

Dans la suite, nous allons étudier respectivement chaque facteur de l'équation (4.24). Pendant chaque l'analyse, le facteur correspondant est considéré comme une contrainte.

Résultat calculé à partir du facteur $(L2 + L1 + L3 - L4)$:

En imposant cette contrainte, nous obtenons les 3 equations suivantes :

$$-(c4 c1 - c2 c3 - s3 s2 - s4 s1) (c4 c1 - c2 c3 + s3 s2 + s4 s1) = 0 \quad (4.25)$$

$$(c4 c1 - c2 c3 + s3 s2 + s4 s1) (L3 + L1) (L2 + L1) = 0 \quad (4.26)$$

$$L4 - L3 - L2 - L1 = 0 \quad (4.27)$$

A partir des 3 équations précédentes, nous pouvons déduire les équations (4.28) et (4.29) car $L1 + L2 \neq 0$ et $L1 + L3 \neq 0$.

$$c4 c1 - c2 c3 + s3 s2 + s4 s1 = 0 \quad (4.28)$$

$$L4 - L3 - L2 - L1 = 0 \quad (4.29)$$

Après le changement de variables, les équations (4.28) et (4.29) deviennent :

$$\cos(a4) \cos(a1) - \cos(a2) \cos(a3) + \sin(a3) \sin(a2) + \sin(a4) \sin(a1) = 0 \quad (4.30)$$

$$L4 - L3 - L2 - L1 = 0 \quad (4.31)$$

Grâce à la formulaire trigonométrique, nous injectons les équations (4.32) et (4.33) dans l'équation (4.30).

$$\cos(a - b) = \cos a \cos b + \sin a \sin b \quad (4.32)$$

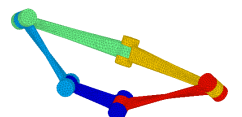
$$\cos(a + b) = \cos a \cos b - \sin a \sin b \quad (4.33)$$

Finalement, les solutions des équations (4.30) et (4.31) sont

$$\begin{cases} a1 + a2 + a3 = a4 \\ L1 + L2 + L3 = L4 \end{cases} \quad (4.34) \quad \begin{cases} a2 + a3 + a4 = a1 \\ L1 + L2 + L3 = L4 \end{cases} \quad (4.35)$$

Ces solutions signifient que la somme des 3 longueurs de barres égale à la longueur de la 4^{ème}. Cette situation n'est possible que lorsque le système de solides à 4 barres est aligné.

Pour continuer l'étude sans oublier de solutions, nous imposons $(L2 + L1 + L3 - L4) \neq 0$ sur les 18 polynômes de la 3^{ème} élimination. En effet, supposons que toutes les solutions des 18 polynômes de la 3^{ème} élimination sont sur un arbre. Une feuille représente une solution. Une branche symbolise un facteur. Donc, il y a au total 7 branches sur cet arbre. En imposant $L2 + L1 + L3 - L4 = 0$, nous exploitons l'ensemble des feuilles sur la première branche, comme illustré dans la figure 4.10. Puis cette branche est coupée en imposant $L2 + L1 + L3 - L4 \neq 0$, voir la figure 4.11.



Spécialisation de facteur	Equations calculées	Solutions correspondantes
$L2 + L1 + L3 - L4$	$c4 c1 - c2 c3 + s3 s2 + s4 s1 = 0$ $L4 - L3 - L2 - L1 = 0$	$a4 - a1 = a2 + a3$ $L1 + L2 + L3 = L4$
		$a4 - a1 = -(a2 + a3)$ $L1 + L2 + L3 = L4$
$-L2 + L1 - L3 - L4$	$c4 c1 - c2 c3 + s3 s2 + s4 s1 = 0$ $-L2 + L1 - L3 - L4 = 0$	$a1 - a4 = a2 + a3$ $L2 + L3 + L4 = L1$
		$a1 - a4 = -(a2 + a3)$ $L2 + L3 + L4 = L1$
$L2 + L1 - L3 + L4$	$s3 s2 + c2 c3 + s4 s1 - c4 c1 = 0$ $L2 + L1 - L3 + L4 = 0$	$a1 + a4 = a2 - a3$ $L1 + L2 + L4 = L3$
		$a1 + a4 = -(a2 - a3)$ $L1 + L2 + L4 = L3$
$-L2 + L1 - L3 + L4$	$c4 c1 - c2 c3 + s3 s2 - s4 s1$ $L4 + L1 - L3 - L2 = 0$	$a1 + a4 = a2 + a3$ $L4 + L1 = L3 + L2$
		$a1 + a4 = -(a2 + a3)$ $L4 + L1 = L3 + L2$
	$c4 c1 - c2 c3 - s3 s2 + s4 s1$ $L1 - L3 = 0, L2 - L4 = 0$	$a1 - a4 = a2 - a3$ $L1 = L3, L2 = L4$
		$a1 - a4 = a3 - a2$ $L1 = L3, L2 = L4$
	$c4 c1 - c2 c3 - s3 s2 + s4 s1$ $L1 - L2 = 0, L3 - L4 = 0$	$a1 - a4 = a2 - a3$ $L1 = L2, L3 = L4$
		$a1 - a4 = a3 - a2$ $L1 = L2, L3 = L4$
$L2 + L1 - L3 - L4$	$-c4 c1 + c2 c3 + s3 s2 - s4 s1 = 0$ $L1 + L2 - L3 - L4 = 0$	$a4 + a1 = \pi - (a2 - a3)$ $L1 + L2 = L3 + L4$
		$a4 + a1 = \pi + (a2 - a3)$ $L1 + L2 = L3 + L4$
$L1 + L3 + L4 - L2$	$-c4 c1 + c2 c3 + s3 s2 - s4 s1 = 0$ $L1 + L3 + L4 - L2 = 0$	$a4 + a1 = a2 - a3$ $L1 + L3 + L4 = L2$
		$a4 + a1 = -(a2 - a3)$ $L1 + L3 + L4 = L2$
$L2 - L1 - L3 + L4$	$-c4 c1 + c2 c3 + s3 s2 - s4 s1 = 0$ $L2 - L1 - L3 + L4 = 0$	$a1 - a4 = a2 - a3$ $L1 + L3 = L2 + L4$
		$a1 - a4 = -(a2 - a3)$ $L1 + L3 = L2 + L4$

TABLE 4.1 – Tableau des solutions calculées avec les 7 facteurs de l'équation (4.24)

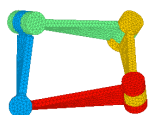




FIGURE 4.10 – L'étude de la situation $L2 + L1 + L3 - L4 = 0$ FIGURE 4.11 – L'étude de la situation $L2 + L1 + L3 - L4 \neq 0$

Nous suivons la même démarche sur les branches restantes. Le tableau 4.1 récapitule les solutions engendrées par le facteur correspondant. Parmi les 18 solutions, 8 représentent les situations pour lesquelles la somme des 3 longueurs de barres égale à la longueur de la 4^{ème}. Les restes signifient que la somme des 2 longueurs de barres égale à la somme des 2 autres. Le code Maple lié à ce calcul se trouve dans l'annexe

Finalement, nous avons séparément parcouru dans §4.2.1.4 et §4.2.1.5 les cas $s4 s2 L1 L3 - s3 s1 L2 L4 = 0$ et $s4 s2 L1 L3 - s3 s1 L2 L4 \neq 0$. Toutes les solutions ont été explorées.

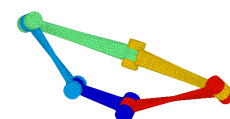
4.2.2 Conditions de mobilité

4.2.2.1 Principe général

Dans cette partie, nous allons explorer comment générer les conditions de mobilité pour le système de solides en boucle fermée. L'algorithme est appliqué sur la mise en équation $C(u, m) = 0$, obtenue avec la variante de la méthode Jin et Yang. Les programmes Maple correspondant aux explications se trouvent en annexe C.

La figure 4.12 expose les étapes principales d'obtention des conditions de mobilité. Comme pour les conditions d'assemblage, cette méthode de génération des conditions de mobilité est également générique. Elle est constituée de 9 étapes principales. Par rapport à la génération des conditions d'assemblage, il y a 4 étapes supplémentaires parce qu'un nouveau système univarié doit être construit et analysé lors du calcul des conditions de mobilité. Les deux rectangles rouges sont deux procédés identiques appliqués aux différents groupes d'équations : le 1^{ère} est appliqué aux équations de fermeture ; le 2^{ème} est appliqué au nouveau système. Le schéma est itératif lors du traitement du nouveau système. Il utilise les spécialisations pour trier les solutions du nouveau système. Les 4 étapes supplémentaires pour obtenir les conditions de mobilité sont :

⚡ **Extraction des coefficients** : après la 1^{ère} élimination, nous factorisons les polynômes du résultat précédent puis nous récupérons les coefficients devant le paramètre de mobilité res-



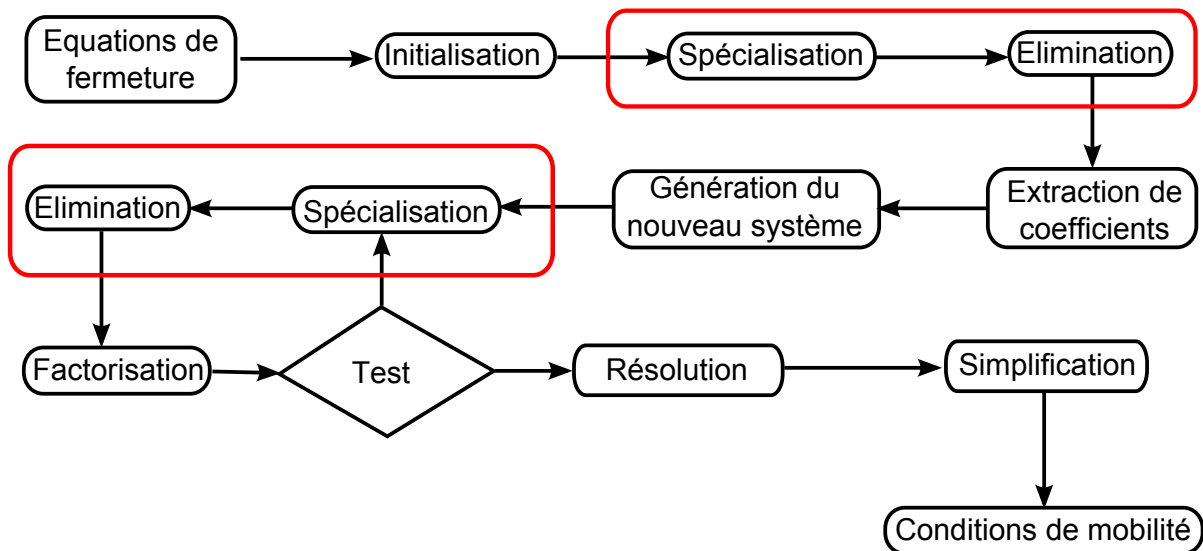


FIGURE 4.12 – Les étapes principales pour obtenir les conditions de mobilité

tant⁷, et également les termes constants.

⚡ **Génération du nouveau système** : le résultat de cette étape est une liste de coefficients. Ces coefficients sont fonction des paramètres d'usinage.

⚡ **Spécialisation (2^{ème} rectangle)** : dans cette étape, l'utilisateur impose des saturations et/ou des contraintes afin de parcourir des solutions spécifiées du nouveau système. Si le temps d'élimination dans l'étape suivante est trop élevé, il faut essayer de diminuer le nombre de saturations et augmenter le nombre de contraintes.

⚡ **Elimination (2^{ème} rectangle)** : l'algorithme des bases de Gröbner élimine les cofacteurs issus des spécifications du nouveau système.

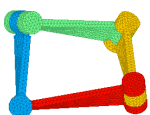
4.2.2.2 Description de l'algorithme

L'écriture du système d'équations est expliquée dans le chapitre 3. Les données d'entrées sont une liste de polynômes (équations de fermeture), un ordre admissible des paramètres et une liste de spécialisations. Les spécifications ne sont pas obligatoires. Les inconnues du système sont les paramètres de mobilité. Les paramètres d'usinage sont les éléments à conserver.

Supposons qu'il y ait n inconnues. Nous éliminons d'abord $n - 1$ inconnues. Le résultat de cette élimination est une liste polynômiale. Puis nous factorisons la liste polynômiale par rapport à la variable de mobilité restante. Ensuite, il faut récupérer les coefficients de la variable de mobilité restante. Le nouveau système est construit avec ces coefficients. Après avoir ajouté les spécialisations, on élimine les cofacteurs s'il y en a⁸. Une nouvelle liste polynômiale est

7. pour le cas d'un *ddl*, tous les paramètres de mobilité doivent être éliminés sauf un.

8. s'il n'y a que des contraintes, il n'y aura pas de cofacteur.



calculée. Puis nous factorisons ces polynômes et nous cherchons à ajouter des contraintes et des saturations afin de trouver les solutions désirées. Lorsqu'aucune nouvelle spécification n'est ajoutée, nous passons aux étapes de simplification pour obtenir les conditions de mobilité courtes et simples.

Dans les prochaines sous-sections, nous allons appliquer l'algorithme sur la variante de la méthode de Jin et Yang. Un autre test, présenté dans l'annexe B.3, est effectué en utilisant la méthode de Jin et Yang. Malheureusement, il n'est pas possible d'atteindre les conditions de mobilité avec la méthode de Jin et Yang en raison de la puissance de l'ordinateur utilisé (Processeur Intel Core i5 2.50GHz, 64 bits et 4,00 Go de RAM). Les conditions de mobilité sont difficiles à obtenir.

La technique présentée dans la sous-section précédente est également utilisée dans cette partie. Ainsi, les relations qui lient $s\alpha$ et $c\alpha$ ne sont pas mises dans la première phase de l'élimination. Ceci n'impacte pas l'exactitude du résultat final.

Avec la variante de la méthode de Jin et Yang, nous obtenons les conditions de mobilité du mécanisme de Bennett. Ces conditions sont nécessaires et suffisantes pour caractériser la mobilité du Bennett. Ces conditions de mobilité, dites aussi équations de compatibilité de la mobilité, ont été découvertes en 1966 par M. Beggs [Beg66].

4.2.2.3 Application du système 4 barres

Nous avons appliqué cette méthode au système 4R. Nous allons montrer comment générer les équations de mobilité à partir la mise en équation de la variante de la méthode Jin et Yang. Nous avons vu dans la section 3.4.3 du chapitre 3 que cette méthode fournit les 4 équations suivantes :

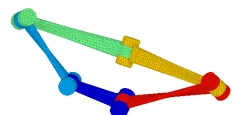
$$C(u_\alpha, u_L, m_\theta) = \begin{cases} \cos \alpha_2 \cdot \cos \alpha_3 - \sin \alpha_3 \cdot \sin \alpha_2 \cdot \cos \theta_C - \cos \alpha_4 \cdot \cos \alpha_1 + \sin \alpha_1 \cdot \sin \alpha_4 \cdot \cos \theta_A = 0 \\ L_4^2 + 2 \cdot L_4 \cdot L_1 \cdot \cos \theta_A + L_1^2 - L_2^2 - 2 \cdot L_2 \cdot L_3 \cdot \cos \theta_C - L_3^2 = 0 \\ L_3 \cdot \sin \alpha_2 \cdot \sin \theta_C + L_4 \cdot \sin \theta_A \cdot \sin \alpha_1 = 0 \\ L_2 \cdot \sin \theta_C \cdot \sin \alpha_3 + L_1 \cdot \sin \alpha_4 \cdot \sin \theta_A = 0 \end{cases} \quad (4.36)$$

☞ Initialisation :

Comme dans la sous-section précédente, les variables $\sin(\alpha_i)$, $\cos(\alpha_i)$, $\sin(\theta_i)$ et $\cos(\theta_i)$ sont remplacées par les symboles s_i et c_i .

$$\begin{cases} \sin(\alpha_1) = s1 \\ \cos(\alpha_1) = c1 \\ \sin(\alpha_2) = s2 \\ \cos(\alpha_2) = c2 \end{cases} \quad \begin{cases} \sin(\alpha_3) = s3 \\ \cos(\alpha_3) = c3 \\ \sin(\alpha_4) = s4 \\ \cos(\alpha_4) = c4 \end{cases} \quad \begin{cases} \sin(\theta_A) = sA \\ \cos(\theta_A) = cA \\ \sin(\theta_C) = sC \\ \cos(\theta_C) = cC \end{cases}$$

Puis, les relations qui lient (sA, cA) et (sC, cC) sont ajoutées. Ainsi, le système d'équations s'écrit :



$$\overline{G} = \begin{cases} c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA = 0 \\ L4^2 + 2 L4 L1 cA + L1^2 - L2^2 - L3^2 - 2 L2 L3 cC = 0 \\ L3 s2 sC + L4 sA s1 = 0 \\ L2 sC s3 + L1 s4 sA = 0 \\ cC^2 + sC^2 - 1 = 0 \\ cA^2 + sA^2 - 1 = 0 \end{cases} \quad (4.37)$$

↳ 1^{ère} élimination :

Dans le système d'équations de fermeture (4.36), il n'y a que deux angles de mobilité θ_A et θ_C . Après l'étape d'initialisation, nous avons 4 variables de mobilité dans le système polynomial (4.37) : sC , cC , sA et cA . L'idée suivante est d'éliminer 3 variables de mobilité et de n'en laisser qu'une. Nous avons choisi dans cet exemple de garder cA et d'éliminer sC , cC et sA . Il n'y a pas de spécialisation dans cette étape. Le résultat de cette élimination comporte 289 équations univariées en fonction de cA . Le degré de cA le plus élevé est 3. Puis nous factorisons ces 289 équations par rapport à cA .

$$\left\{ \begin{array}{l} \vdots \\ (s4 s1 L1^2 + 2 L4 c2 c3 L1 - 2 L4 c4 c1 L1 + s4 s1 L4^2 - s4 s1 L3^2 - s4 s1 L2^2) cA + \\ 2 s4 s1 L4 L1 + c3 c2 L1^2 - c4 c1 L1^2 - c4 c1 L4^2 - c3 c2 L3^2 + c4 c1 L3^2 + \\ c3 c2 L4^2 - c3 c2 L2^2 + c4 c1 L2^2 - 2 s3 s2 L3 L2 = 0 \\ (2 s4 L3 s1 L2 - 2 L1 s2 L4 s3) cA + 2 c3 c2 L3 L2 - \\ 2 c4 c1 L3 L2 - s3 s2 L4^2 + s3 s2 L3^2 + s3 s2 L2^2 - s3 s2 L1^2 = 0 \end{array} \right. \quad (4.38)$$

↳ Extraction de coefficients :

Selon la technique de la figure 4.1, comme nous souhaitons imposer que les polynômes soient nuls quelque soit cA , il faut que les coefficients en facteur de cA soient tous nuls. L'étape suivante consiste à extraire les coefficients à partir des 289 équations et de les mettre dans une liste. Il y a 537 coefficients à annuler, eux-mêmes considérés comme des polynômes qui dépendent des paramètres d'usinage.

↳ Ajout de $Q(s\alpha, c\alpha) = 0$:

Comme pour le cas précédent, nous devons aussi ajouter les 4 relations qui lient $s\alpha$ et $c\alpha$:

$$\triangleleft s1^2 + c1^2 - 1 = 0 \quad (\sin^2 \alpha_1 + \cos^2 \alpha_1 = 1)$$

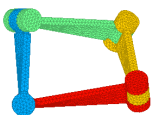
$$\triangleleft s2^2 + c2^2 - 1 = 0 \quad (\sin^2 \alpha_2 + \cos^2 \alpha_2 = 1)$$

$$\triangleleft s3^2 + c3^2 - 1 = 0 \quad (\sin^2 \alpha_3 + \cos^2 \alpha_3 = 1)$$

$$\triangleleft s4^2 + c4^2 - 1 = 0 \quad (\sin^2 \alpha_4 + \cos^2 \alpha_4 = 1)$$

↳ Spécialisation :

Nous venons de construire le nouveau système polynomial. Il faut maintenant ajouter les spécialisations pour trier les solutions de ces 537 polynômes. Nous commençons souvent par l'ajout des saturations pour garantir l'absence de solutions indésirables. Les cofacteurs u et v doivent être éliminés par la suite.



↷ 1^{ère} saturation : les longueurs de barres sont non nulles.

$$\begin{cases} L1 \neq 0 \\ L2 \neq 0 \\ L3 \neq 0 \\ L4 \neq 0 \end{cases} \implies L1 \cdot L2 \cdot L3 \cdot L4 \cdot u - 1 (= 0)$$

↷ 2^{ème} saturation : le système de solides n'est pas plan.

$$\begin{cases} \sin \alpha_1 \neq 0 \\ \sin \alpha_2 \neq 0 \\ \sin \alpha_3 \neq 0 \\ \sin \alpha_4 \neq 0 \end{cases} \implies s1 \cdot s2 \cdot s3 \cdot s4 \cdot v - 1 (= 0)$$

↷ 2^{ème} élimination :

Cette fois-ci, le système de départ comporte 537 polynômes, 4 relations trigonométriques en paramètres d'usinage plus deux saturations. Après la 2^{ème} élimination, nous récupérons 107 polynômes.

↷ Spécialisation :

Après avoir factorisé ces polynômes, certains facteurs tel que $L1 + L3$ et $L1 - L3$, $L1 + L2$ et $L1 - L2$, sont répétés plusieurs fois :

- ↷ 7 fois $L1 - L3$
- ↷ 7 fois $L1 + L3$
- ↷ 6 fois $L1 - L2$
- ↷ 6 fois $L1 + L2$

Les solutions $L1 + L3$ et $L1 + L2$ ne sont pas réalisables car il n'y a pas de longueurs opposées. Pour être rigoureux, nous avons effectué les tests sur $L1 - L3$ et $L1 - L2$. Le résultat avec la contrainte $L1 - L3$ est présenté en premier. Celui avec la contrainte $L1 - L2$ sera présenté à la suite.

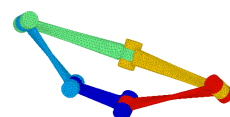
1^{er} cas : $L1 - L3$

La nouvelle liste de spécialisation est :

- ↷ Saturations : $L1 \cdot L2 \cdot L3 \cdot L4 \cdot u - 1$, $s1 \cdot s2 \cdot s3 \cdot s4 \cdot v - 1$
- ↷ Contrainte : $L1 - L3$

↷ 3^{ème} élimination :

Dans la prochaine itération, u et v doivent être éliminés. Après cette élimination, le nombre de polynômes passe de 107 à 43.



☞ Nouvelle spécialisation :

Parmi les 43 polynômes, il n'y a plus de répétition de facteur. Le facteur $L2 - L4$ a attiré notre attention car il apparaît dans l'avant dernier polynôme. Le dernier polynôme est $L1 - L3$, nous l'avons ajouté dans l'étape précédente. Donc, nous ajoutons $L2 - L4$ comme nouvelle contrainte :

☞ Saturations : $L1 \cdot L2 \cdot L3 \cdot L4 \cdot u - 1, s1 \cdot s2 \cdot s3 \cdot s4 \cdot v - 1$

☞ Contraintes : $L1 - L3, L2 - L4$

☞ 4^{ème} élimination :

Après la 4^{ème} élimination, le nombre de polynômes passe de 43 à 22. Néanmoins, ces 22 polynômes peuvent être simplifiés.

☞ Simplification : réduction automatique du nombre de polynômes

A l'étape de simplification, nous cherchons toujours le nombre de polynômes générateurs le plus petit. Après cette opération, 9 polynômes sont supprimés. Finalement, il ne reste que 13 polynômes. Ces 13 polynômes sont les conditions de mobilité calculées par la méthode de la base de Gröbner pour le système 4R en 3D.

Nous pouvons remarquer que la dimension Hilbert des 13 polynômes est égale à 3. Puisqu'il y a 12 paramètres dans le système, il doit y avoir $12 - 3 = 9$ relations de conditions de mobilité. Parmi les 9 équations, 4 sont les relations trigonométriques tel que $s\alpha^2 + c\alpha^2 - 1 = 0$. Donc, $9 - 4 = 5$ équations sont suffisantes pour caractériser la mobilité du système 4R en 3D. Afin de retrouver les 5 équations publiées dans [Ben03], quelques opérations manuelles seront nécessaires. Nous les développerons au §4.3.1.1.

$$s4 s3 L2^2 - s2 s1 L2^2 - s4 s3 L1^2 + s2 s1 L1^2 = 0 \quad (4.39)$$

$$c3^2 L2^2 - c3^2 L1^2 - c2^2 L1^2 + c1^2 L1^2 - L2^2 + L1^2 = 0 \quad (4.40)$$

$$s3 s1 L2^2 - s4 s2 L1^2 = 0 \quad (4.41)$$

$$c1^2 L2^2 - c2^2 L1^2 - L2^2 + L1^2 = 0 \quad (4.42)$$

$$s4^2 - 1 + c3^2 + c2^2 - c1^2 = 0 \quad (4.43)$$

$$c4^2 - c3^2 - c2^2 + c1^2 = 0 \quad (4.44)$$

$$s3^2 + c3^2 - 1 = 0 \quad (4.45)$$

$$s3 s2 - s4 s1 = 0 \quad (4.46)$$

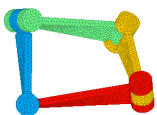
$$s2^2 + c2^2 - 1 = 0 \quad (4.47)$$

$$c2 c3 - c4 c1 = 0 \quad (4.48)$$

$$s1^2 + c1^2 - 1 = 0 \quad (4.49)$$

$$L4 - L2 = 0 \quad (4.50)$$

$$L3 - L1 = 0 \quad (4.51)$$



2^{ème} cas : L1-L2

La nouvelle liste de spécialisation est :

☞ Saturations : $L1 \cdot L2 \cdot L3 \cdot L4 \cdot u - 1, s1 \cdot s2 \cdot s3 \cdot s4 \cdot v - 1$

☞ Contrainte : $L1 - L2$

☞ 3^{ème} élimination :

Dans cette étape, nous obtenons 28 polynômes.

☞ Nouvelle spécialisation :

Parmi les 28 polynômes obtenus, $L3 - L4$ qui est identifiée comme une contrainte. La nouvelle liste de spécialisation est :

☞ Saturations : $L1 \cdot L2 \cdot L3 \cdot L4 \cdot u - 1, s1 \cdot s2 \cdot s3 \cdot s4 \cdot v - 1$

☞ Contraintes : $L1 - L2, L3 - L4$

☞ 4^{ème} élimination :

Avec la nouvelle liste de spécialisation, l'algorithme fournit 16 polynômes. Puisqu'il n'y a plus de spécialisation à ajouter, nous passons à l'étape de simplification.

☞ Simplification : réduction automatique du nombre de polynômes

Cette étape permet d'obtenir 9 polynômes au lieu de 13 :

$$c4^2 - c3^2 = 0 \tag{4.52}$$

$$s3^2 + c3^2 - 1 = 0 \tag{4.53}$$

$$s3 s2 - s4 s1 = 0 \tag{4.54}$$

$$s2^2 - 1 + c1^2 = 0 \tag{4.55}$$

$$c2 c3 - c4 c1 = 0 \tag{4.56}$$

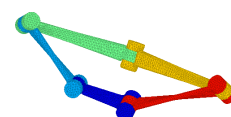
$$c2^2 - c1^2 = 0 \tag{4.57}$$

$$s1^2 + c1^2 - 1 = 0 \tag{4.58}$$

$$L4 - L3 = 0 \tag{4.59}$$

$$L2 - L1 = 0 \tag{4.60}$$

Ces deux résultats polynomiaux sont réécrits dans la section 4.3.1 sous forme trigonométrique pour être comparés avec les conditions de mobilité déjà obtenues.



4.3 Validations

4.3.1 Résolution formelle et interprétation géométrique

4.3.1.1 Cas $L1 = L3$:

A noter que cette opération n'est pas obligatoire. Elle est toutefois présentée pour cet exemple car nous voulons trouver les conditions de mobilité connues à partir des polynômes (4.39)-(4.51) que notre méthode a générés.

En remplaçant les symboles si et ci par sinus et cosinus, puis $\alpha_1 = a1$, $\alpha_2 = a2$, $\alpha_3 = a3$ et $\alpha_4 = a4$, les polynômes (4.39)-(4.51) se transforment en 8 équations.

$$\sin(a4) \sin(a3) L2^2 - \sin(a2) \sin(a1) L2^2 - \sin(a4) \sin(a3) L1^2 + \sin(a2) \sin(a1) L1^2 = 0 \quad (4.61)$$

$$\cos(a3)^2 L2^2 - \cos(a3)^2 L1^2 - \cos(a2)^2 L1^2 + \cos(a1)^2 L1^2 - L2^2 + L1^2 = 0 \quad (4.62)$$

$$\sin(a3) \sin(a1) L2^2 - \sin(a4) \sin(a2) L1^2 = 0 \quad (4.63)$$

$$\sin(a2)^2 L1^2 - \sin(a1)^2 L2^2 = 0 \quad (4.64)$$

$$\sin(a3) \sin(a2) - \sin(a4) \sin(a1) = 0 \quad (4.65)$$

$$\cos(a2) \cos(a3) - \cos(a4) \cos(a1) = 0 \quad (4.66)$$

$$L4 - L2 = 0 \quad (4.67)$$

$$L3 - L1 = 0 \quad (4.68)$$

Nous avons constaté que les Eqs.(4.65)-(4.66) ne dépendent que des angles dimensionels. Après une résolution symbolique en $a1$ et $a2$, nous avons

$$\begin{cases} a1 = a3 \\ a2 = a4 \end{cases} \quad (4.69)$$

$$\begin{cases} a1 = -a3 \\ a2 = -a4 \end{cases} \quad (4.70)$$

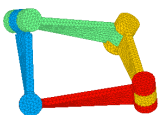
Ensuite, nous réinjectons les deux solutions angulaires (4.69) et (4.70) dans les équations (4.61)-(4.68), et nous obtenons :

$$\begin{cases} L1^2 \sin^2(a2) - L2^2 \sin^2(a1) = 0 \\ L1 = L3 \\ L2 = L4 \end{cases} \quad (4.71)$$

Les figures 4.13 et 4.14 montrent les mécanismes 4R correspondant aux solutions calculées : (4.69) et (4.71) pour la figure 4.13; (4.70) et (4.71) pour la figure 4.14.

La première solution, appelée mécanisme de Bennett, est celle donnée par Beggs [Beg66]. La 2^{ème} solution respecte aussi le système d'équations (4.36) et les barres peuvent être assemblées à π près. Le mécanisme de la figure 4.14 possède une mobilité globale.

Finalement, avec $\epsilon = \pm 1$, les conditions de mobilité de la famille du mécanisme de Bennett



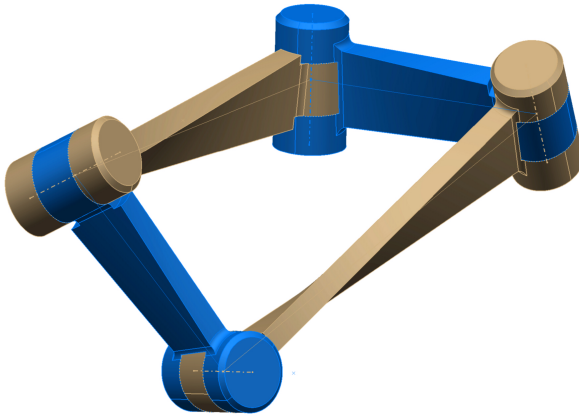


FIGURE 4.13 – Mécanisme de Bennett de la solution $a1 = a3$ et $a2 = a4$

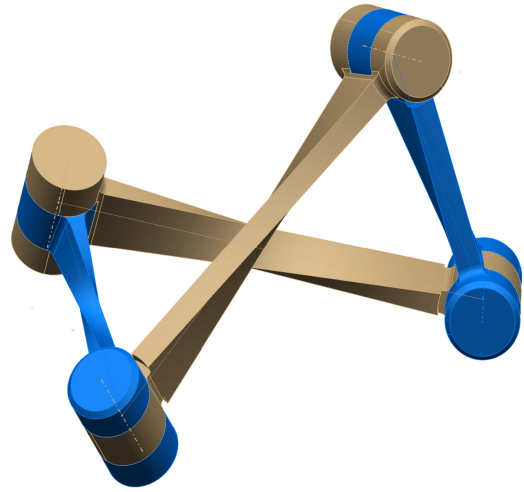


FIGURE 4.14 – Mécanisme de Bennett de la solution $a1 = -a3$ et $a2 = -a4$

sont :

$$\left\{ \begin{array}{l} L1^2 \sin^2(a2) - L2^2 \sin^2(a1) = 0 \\ L2 = L4 \\ L1 = L3 \\ a2 = \epsilon a4 \\ a1 = \epsilon a3 \end{array} \right. \quad (4.72)$$

4.3.1.2 Cas $L1 = L2$:

Après l'étape de simplification automatique et trigonométrique, nous obtenons 6 équations :

$$\cos(a4)^2 - \cos(a3)^2 = 0 \quad (4.73)$$

$$\sin(a3) \sin(a2) - \sin(a4) \sin(a1) = 0 \quad (4.74)$$

$$\cos(a2) \cos(a3) - \cos(a4) \cos(a1) = 0 \quad (4.75)$$

$$\cos(a2)^2 - \cos(a1)^2 = 0 \quad (4.76)$$

$$L4 - L3 = 0 \quad (4.77)$$

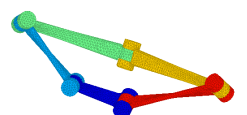
$$L2 - L1 = 0 \quad (4.78)$$

En résolvant l'équation (4.74) et l'équation (4.75), nous avons

$$\left\{ \begin{array}{l} a1 = a2 \\ a3 = a4 \end{array} \right. \quad (4.79)$$

$$\left\{ \begin{array}{l} a1 = -a2 \\ a3 = -a4 \end{array} \right. \quad (4.80)$$

En substituant les deux solutions précédentes dans les équations (4.73)-(4.76), ces 6 équations sont vérifiées.



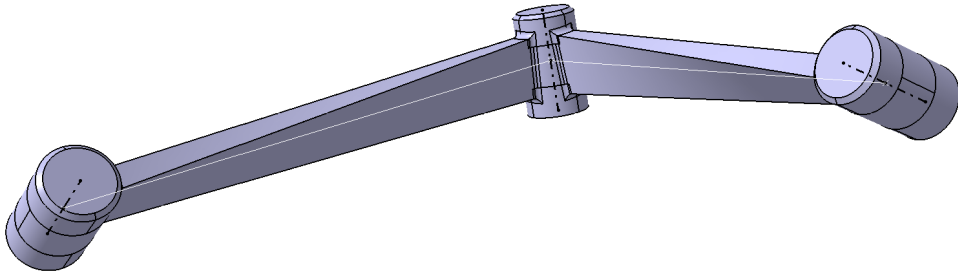


FIGURE 4.15 – Représentation géométrique de 4R pour le cas $L1 = L2$

$$\left\{ \begin{array}{l} L3 = L4 \\ L1 = L2 \\ a3 = \epsilon a4 \\ a2 = \epsilon a1 \end{array} \right. \quad (4.81)$$

Finalement, avec $\epsilon = \pm 1$, un autre exemple de condition de mobilité est l'équation (4.81).

La figure 4.15 est une représentation géométrique du système d'équation (4.81). Dans ce système à quatre barres, deux barres successives se superposent de telle sorte que deux des liaisons pivot habituellement distinctes se trouvent confondues. Ainsi, ce système particulier devient équivalent à un système à deux barres.

4.3.2 Validation numérique

Nous avons trouvé les conditions de mobilité et d'assemblage. Pour les valider, les cas tests numériques sont nécessaires. Dans cette sous-section, nous allons vérifier numériquement l'exactitude des conditions d'existence. Plusieurs configurations du mécanisme de 4R existent dans notre base de donnée. Elles sont listées dans le tableau 4.2.

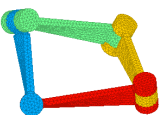
Les dimensions indiquées dans le tableau 4.2 sont données avec 9 chiffres significatifs, ce qui est bien supérieur à la précision atteignable en fabrication. Cette précision excessive sert uniquement à valider les équations (4.82), (4.83), (4.84) et (4.85) obtenues précédemment.

4.3.2.1 Condition d'existence d'assemblage

Les conditions d'assemblage du 4R sont :

$$(L2^2 - L1^2 - L3^2 + L4^2)^2 - \left(\frac{2 L2 L4}{\tan(a2) \tan(a4)} - \frac{2 L1 L3}{\tan(a1) \tan(a3)} \right)^2 = 0 \quad (4.82)$$

$$\frac{L2 L4}{\sin(a2) \sin(a4)} - \frac{L1 L3}{\sin(a1) \sin(a3)} = 0 \quad (4.83)$$



	Mobile	Assemblé	Non-assemblé
	Config. 1	Config. 2	Config. 3
L1(m)	1.000000000	1.000000000	1.500000000
a1(rad)	0.523598776	-0.291456795	-1.29145679
L2(m)	1.732050808	0.8104936767	0.810493677
a2(rad)	-1.04719755	0.45323040952	0.453230410
L3(m)	1.000000000	0.96695398029	0.966953980
a3(rad)	0.523598776	-0.3354427351	-0.335442735
L4(m)	1.732050808	1.4149558297	1.41495583
a4(rad)	-1.04719755	0.25909586837	0.259095868

TABLE 4.2 – Différentes configurations d'un mécanisme du Bennett

Trois configurations de système 4R sont testées dont une mobile, une rigide et une non-assemblable. Nous obtenons les résultats désirés.

	Config. 1	Config. 2	Config. 3
Eq. (4.82)	0	$1.80 \cdot 10^{-22}$	-0.29774906042184027196
Eq. (4.83)	0	0	0.20016507400554069343

Dans le cas de la configuration non-assemblable (3^{ème}), les équations ne sont pas égales à 0 comme prévu.

Ainsi, nous concluons que les conditions d'assemblage (4.82) et (4.83) sont correctes.

4.3.2.2 Conditions de mobilité

Les conditions de mobilité sont :

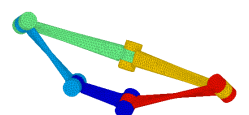
$$L2^2 \sin^2(a1) - L1^2 \sin^2(a2) = 0 \quad (4.84)$$

$$L1 - L3 = 0 \quad (4.85)$$

$$L2 - L4 = 0 \quad (4.86)$$

$$a1 - a3 = 0 \quad (4.87)$$

$$a2 - a4 = 0 \quad (4.88)$$



	Config. 1	Config. 2	Config. 3
Eq. (4.84)=	$-2.53557811 \cdot 10^{-12}$	0.13749250223777324821	-0.17556426795280304124
Eq. (4.85)=	0	0.03304601970931419098	0.53304601970931419098
Eq. (4.86)=	0	-0.60446215298120262368	-0.60446215298120262368
Eq. (4.87)=	$-3.207 \cdot 10^{-12}$	0.04398594061184479851	-0.95601405938815520149
Eq. (4.88)=	$3.459 \cdot 10^{-12}$	0.19413454114080238181	0.19413454114080238181

En substituant les valeurs numériques données par une colonne du tableau 4.2, dans les équations (4.84)-(4.88), les conditions de mobilité sont respectées. Les autres valeurs numériques d'un système 4R assemblé ou non assemblé ne respectent pas les conditions de mobilité.

4.4 Application des équations d'assemblage

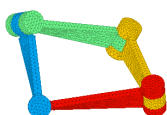
Les équations d'assemblage pour l'exemple du 4R sont générées. Nous vous montrons comment utiliser ces équations d'assemblage symboliques pour construire un système 4R rigide. Dans la phase de modélisation géométrique d'un 4R, un ingénieur peut imposer quelques paramètres d'usinage. Mais il n'est pas obligé de donner toutes les valeurs. Grâce aux conditions d'assemblage, il peut calculer des paramètres d'usinage restant. Néanmoins, cette méthode fonctionne sous une condition : le nombre d'inconnues des paramètres d'usinage doit être au moins égal au nombre d'équations d'assemblage.

Pour le système de solides 4R, il y a 2 équations d'assemblage. Le système a 12 variables d'usinage intervenant dans les équations d'assemblage⁹. Si nous fixons 10 d'entre elles, les 2 restantes peuvent être déterminées à l'aide des équations. Nous allons illustrer cette application avec un exemple.

$$\begin{cases} \text{Eq. (4.82)} \\ \text{Eq. (4.83)} \end{cases} = \begin{cases} 0.112610 \cdot 10^{-3} (L2^2 - 1. L1^2 + 1.06710)^2 - 1. (0.232626 L2 - 0.196224 L1)^2 = 0 \\ 0.133842 L2 - 0.108478 L1 = 0 \end{cases} \quad (4.89)$$

Après avoir évalué 10 paramètres ($\sin(a1)$, $\cos(a1)$, $\sin(a2)$, $\cos(a2)$, $L3$, $\sin(a3)$, $\cos(a3)$, $L4$, $\sin(a4)$, $\cos(a4)$), les équations d'assemblage du 4R deviennent l'équation (4.89). Les solutions

9. Elles sont $L1$, $\sin(a1)$, $\cos(a1)$, $L2$, $\sin(a2)$, $\cos(a2)$, $L3$, $\sin(a3)$, $\cos(a3)$, $L4$, $\sin(a4)$, $\cos(a4)$



Conclusion

des longueurs $L1$ et $L2$ de l'équation (4.89) sont :

Solution 1 :	$L1 = 1.00000000000000000007$	$L2 = 0.81049367671808519850$
Solution 2 :	$L1 = 3.1101719615272515278$	$L2 = 2.5207747083237211134$
Solution 3 :	$L1 = -1.00000000000000000007$	$L2 = -0.81049367671808519850$
Solution 4 :	$L1 = -3.1101719615272515278$	$L2 = -2.5207747083237211134$

La 3^{ème} et la 4^{ème} solution ont des longueurs négatives, elles peuvent être ignorées. Il est clair que la première solution est celle de la configuration 2 du tableau 4.2. Cette configuration est vérifiée dans CATIA, voir figure 4.16.

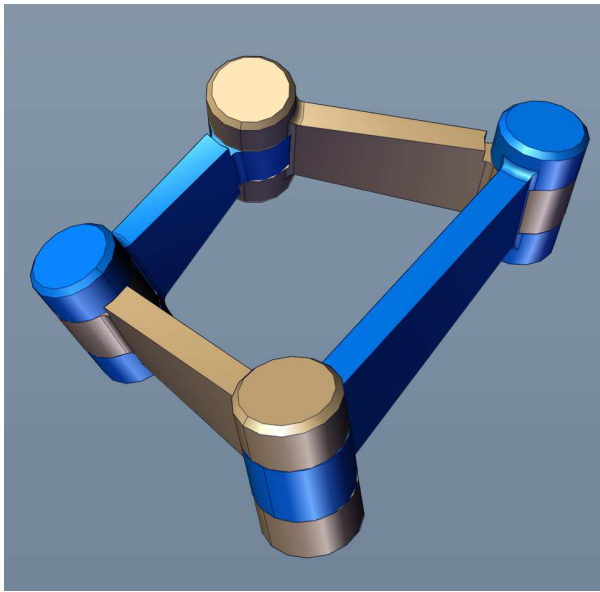


FIGURE 4.16 – Système à 4 barres assemblées de la configuration 2 (1^{ère} solution)

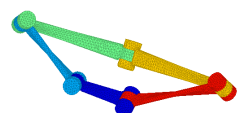
FIGURE 4.17 – Système à 4 barres non assemblées de la 2^{ème} solution

La 2^{ème} solution est compatible avec les paramètres déterminés et les équations de départ. Néanmoins, les pièces de cette configuration ne s'assemblent pas dans CATIA. Cela est dû à un défaut de la mise en équation : le produit scalaire $\langle \vec{R}, \vec{R} \rangle$ annule l'orientation du passage vectoriel dans chaque boucle de coupure. Autrement dit, nous perdons l'orientation du vecteur \vec{R}_A et \vec{R}_B en faisant le produit scalaire $\langle \vec{R}, \vec{R} \rangle$. Dans le meilleur des cas, le système à 4 barres de la 2^{ème} solution est la figure 4.17.

Par contre, si nous réinjectons la 2^{ème} solution dans le système d'équations de Jin et Yang du chapitre 3, une des équations n'est pas respectée car le produit mixte $\langle \vec{R}, \vec{n}_B \times \vec{n}_D \rangle$ préserve bien l'orientation des vecteurs. Il faudra donc vérifier la faisabilité des solutions avant de la proposer à l'ingénieur qui ne connaît pas les propriétés des mécanismes.

4.5 Conclusion

Dans ce chapitre, nous avons proposé des méthodes générales et automatiques pour générer les conditions d'assemblage et de mobilité. Ces relations symboliques dépendent des paramètres



d'usinage. Ces méthodes sont appliquées au système de solides 4R 3D. Grâce à elles, nous avons retrouvé les conditions de mobilité connues. Nous avons aussi découvert les conditions d'assemblage du 4R 3D. Ces conditions obtenues n'ont jamais été publiées auparavant.

La base de Gröbner est un système polynômial équivalent au système de départ. L'algorithme de la base de Gröbner permet d'éliminer plusieurs variables dans un ensemble d'équations. Nous avons choisi cette approche pour résoudre le système d'équations polynomiales. Grâce à la base de Gröbner, les conditions de mobilité et d'assemblage pour la famille de 4R sont générées. Les solutions symboliques exposées dans la partie 4.2 illustrent les conditions d'existence. Pour générer les conditions d'assemblage, nous avons exploré toutes les branches de solutions où les longueurs de barres ne sont pas nulles et les axes des liaisons pivots dans une barre ne sont pas parallèles. Une application montre comment utiliser les conditions d'existence pour aider les ingénieurs à assembler les systèmes à 4 barres.

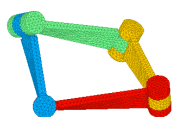
Néanmoins, nous n'avons pas pu trouver les conditions d'assemblage pour le mécanisme de Bennett avec la mise en équation de Jin et Yang. La résolution symbolique des bases de Gröbner requièrent des ressources informatiques variant exponentiellement avec la taille du modèle à traiter. Par exemple, les conditions de mobilité du système mécanique plan surcontraint mentionné dans l'introduction sont obtenues dans moins de 0.03 secondes. Et celles du mécanisme de Bennett le sont en 27.56 secondes. Le problème du coût de calcul ne doit pas être négligé. Cependant, certains développements récents tel que F5 [Fau02, Ver] permettent d'améliorer sensiblement les performances de la résolution symbolique des bases de Gröbner.

Au niveau de la mise en équation, il apparaît que celle de la variante de la méthode de Jin et Yang perd l'information de l'orientation des vecteurs. Ceci a conduit à une solution irréalisable lors de l'application sur une configuration d'un 4R 3D. Dans ce cas, il faut reinjecter cette solution dans tous les systèmes d'équations générés dans le chapitre 3 et vérifier si ces équations sont respectées. Souvent, s'il y a une équation qui n'est pas respectée, les solides ne s'assemblent pas.

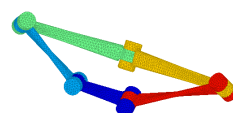
Les algorithmes employés pour l'obtention des conditions d'assemblage et de mobilité sont généraux et pourront être appliqués aux autres familles de mécanismes, Goldberg 5R et Bricard 6R par exemple.

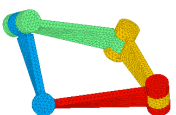
Bibliographie du chapitre 4

- [Beg66] J. S. Beggs. *Advance Mechanism*. Macmillan, 1966.
- [Ben03] G.T. Bennett. A new mechanism. *Engineering*, 76 :777–778, 1903.
- [Fau02] Jean-Charles Faugère. A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM.
- [Ver] Thibaut Verron. Bases de Gröbner, à la rencontre entre géométrie algébrique et algorithmique. Url : https://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDIQFjAA&url=http%3A%2F%2Fwww.math.ens.fr%2Fenseignement%2Ftelecharger_fichier.php%3Ffichier%3D788&ei=cC7QUsrCCIZb7Ab1_YG4CA&usg=AFQjCNGTsO



9RI9n5jajV8-xNsjEdZeSXjw&sig2=X7t99QjlTg0TBThG76ohhA&bvm=bv.59026428,d.Yms&cad=rja.





Conclusion générale

Peu d'ingénieurs sont conscients qu'il existe en 3D, quelques mécanismes constitués de solides en chaîne fermée ayant moins de 7 liaisons et qui sont cependant mobiles. Les propriétés de ces systèmes sont très particulières, ce qui rend difficile leur application industrielle. En effet, la mobilité de ces systèmes surcontraints semble impossible a priori. A l'heure actuelle, aucun outil de CAO n'offre une aide à la conception pour de tels systèmes. C'est la raison pour laquelle, l'objectif de la thèse était de trouver leurs conditions d'assemblage et de mobilité. Au début du mémoire, nous avons introduit les problèmes liés aux systèmes de solides en chaîne fermée en prenant l'exemple d'un système plan à 5 barres.

Le premier chapitre a dressé l'état de l'art sur le paramétrage, les méthodes génériques de mise en équation, d'analyse et de résolution. Les définitions des différents systèmes de solides ont été données. Des squelettes ont été associés aux systèmes de solides pour modéliser le problème. Les paramètres de Denavit-Hartenberg ont été classés en deux catégories : les paramètres d'usinage et les paramètres de mobilité. Selon la méthode utilisée, les équations de fermeture sont souvent composées de plusieurs types de paramètres. Pour obtenir des conditions qui ne dépendent que des paramètres d'usinage, des méthodes d'analyse et d'élimination, numériques et symboliques, ont été exposées.

Le deuxième chapitre a présenté une méthode numérique locale pour générer les conditions d'assemblage et de mobilité. Cette méthode est générique et elle a été appliquée au système de solides 5R en 3D. L'écriture du système d'équations était basée sur une approche non-cartésienne présentée dans le chapitre 1. Ensuite, le système d'équations a été linéarisé. La décomposition en valeur singulière a été utilisée pour éliminer les paramètres. Ensuite, les conditions d'assemblage ont été obtenues. Elles sont valides au voisinage de la configuration initiale. Les conditions de mobilité ont été obtenues à partir des conditions d'assemblage calculées pour un nombre fini de configurations.

Le troisième chapitre a détaillé différentes méthodes de mise en équation non-cartésienne : la première approche est basée sur la matrice de Gram construite avec des fonctions F_i issues de la formule d'Olinde-Rodrigues. Dans une chaîne fermée, la fermeture angulaire est assurée par la symétrie de la matrice de Gram. Celle-ci est formalisée en écrivant l'égalité de deux éléments symétriques de cette matrice. La fermeture vectorielle est garantie soit par la méthode des coupes métriques soit par celles des projections. L'inconvénient de cette approche est que le nombre d'équations est proportionnel au nombre de vecteurs dans la boucle. Ce nombre peut être limité en sélectionnant un sous-ensemble d'équations mais le choix est compliqué à faire a priori. Pour éviter ce problème, l'approche de Jin et Yang a été testée. Elle consiste à écrire quatre équations quel que soit le nombre de vecteurs dans une boucle. Néanmoins, cette approche requiert un calcul supplémentaire par rapport à la première approche : le produit

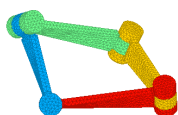
vectorel $\vec{z}_i \times \vec{z}_j$. En s'inspirant de la méthode de Jin et Yang, une variante originale a été proposée. En revanche, comme avec la méthode de Jin et Yang, les deux vecteurs de référence ne doivent pas être colinéaires. Nous avons aussi présenté dans ce chapitre quatre exemples de système d'équations pour le système 4R. A la fin du chapitre, les systèmes d'équations du 4R sont regroupés et résolus à la main. Un tableau synthétise les différentes solutions. Finalement, nous avons montré qu'une vérification géométrique est nécessaire pour valider les solutions obtenues.

Le chapitre 4 a ouvert la voie d'une généralisation à l'obtention des conditions d'assemblage et de mobilité. Afin de générer ces conditions en fonction des paramètres d'usinage, il faut éliminer des paramètres. C'est la méthode des bases de Gröbner qui s'en charge. Avec la variante de la méthode de Jin et Yang, nous avons trouvé les conditions de mobilité du 4R en 3D. De plus, les conditions d'assemblage, qui n'ont jamais été publiées pour cette même famille, ont été calculées grâce à notre méthode générique. Ces conditions sont fonction des paramètres d'usinage. Pour obtenir ces résultats, nous avons également utilisé deux outils de spécialisation : la saturation et la contrainte. La simplification automatique avec la base de Gröbner est une des contributions de cette thèse. Cette opération permet de calculer le plus petit ensemble de polynômes générateurs qui représentent le même idéal que le système de départ.

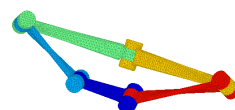
Les contributions de cette thèse sont multiples. Tout d'abord, nous avons généré les conditions locales d'assemblage et de mobilité d'un système 5R. Ces conditions ont été validées et ont fait l'objet d'un article publié dans une revue internationale. Ensuite, pour écrire les équations de fermeture, plusieurs mises en équation ont été présentées. Parmi elles, une mise en équation originale basée sur la méthode de Jin et Yang, a été privilégiée pour la résolution symbolique. Nous avons proposé des algorithmes de génération des conditions formelles d'assemblage et de mobilité pour un système de solides. Ces algorithmes utilisent les bases de Gröbner pour éliminer plusieurs paramètres dans le système d'équations. Une autre contribution de cette thèse est le développement d'une fonction de simplification automatique en utilisant les bases de Gröbner. Enfin, nous avons fourni deux outils de spécialisation pour parcourir l'arbre des solutions. Ces travaux, développés sous Maple, ont été illustrés sur le système de solides 4R.

Les perspectives de ce travail sont nombreuses. À l'avenir, la mise en équation devra s'adapter automatiquement au problème étudié. Il faudra donc développer un générateur automatique de système d'équations à partir du squelette du modèle. De plus, nous avons constaté des limites au calcul symbolique, principalement liées à la complexité exponentielle. Ceci conduit souvent le calcul à consommer une quantité de mémoire trop importante. Pour contourner le problème, nous pourrions nous orienter vers les points suivants :

- recherches de nouveaux algorithmes performants adaptés aux problèmes étudiés ;
- recherches de nouvelles spécialisations afin de mieux d'exploiter l'ensemble des solutions fournies par les méthodes de calcul symbolique ;
- traiter des problèmes partiellement instanciés dans lesquels certains paramètres ont des valeurs constantes ;
- les solutions présentées peuvent être utilisées pour établir des bibliothèques de mécanismes. Dans ce cas, l'utilisateur n'a plus besoin de calculer, à la demande, les conditions d'assemblage et de mobilité mais de les utiliser directement.

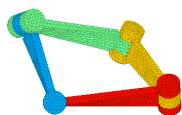


Il serait également souhaitable d'enrichir les différentes liaisons prises en compte et de traiter le cas de systèmes ayant plusieurs boucles. Enfin, il est indispensable de proposer une interface homme/machine qui permettrait de parcourir l'arbre des solutions de manière visuelle, directement dans l'interface CAO [Duf00, Mat04]. De nouvelles spécialisations pourraient ainsi être facilement identifiées.

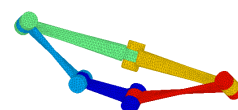


Bibliographie générale

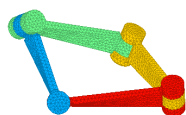
- [Alf] Liliane Alfonsi. Bézout et les intersection de courbes algébriques. Url : <http://www.bibnum.education.fr/files/Bezout-analyse.pdf>.
- [Alf08] Liliane Alfonsi. Étienne Bézout : Analyse algébrique au siècle des lumières. *Revue d'Histoire des Mathématiques*, 14 :211–287, 2008.
- [Ang82] J. Angeles. *Spatial Kinematic Chains*. Springer-Verlag, Berlin, 1982.
- [Ari97] C. M. Hoffmann & A.J. Arinyo. Symbolic constraints in constructive geometric constraint solving. *Journal of Symbolic Computation*, 23 :287–299, 1997.
- [Bar04] Magali Turrel Bardet. *Étude des systèmes algébriques surdéterminés-Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Paris 6, 2004.
- [Bas77] J. Bass. *Cours de mathématiques. Tome 1, fascicule 1 : algèbre linéaire, intégrales simple, fonction définies par des séries et des intégrales*. Masson, 1977.
- [Bed07] C. Mavroidis & M. Beddows. A spatial overconstrained mechanism that can be used in practical applications, 2007.
- [Beg66] J. S. Beggs. *Advance Mechanism*. Macmillan, 1966.
- [Ben03] G.T. Bennett. A new mechanism. *Engineering*, 76 :777–778, 1903.
- [Bon00] D. Bondyfalat. *Interaction entre symbolique et numérique : application à la vision artificielle*. PhD thesis, Université de Nice Sophia Antipolis, 2000.
- [Bri26] R. Bricard. *Leçons de cinématique*, chapter Leçons de cinématique : Cinématique appliquée. Gauthier-Villars, 1926.
- [Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Innsbruck, 1965.
- [Cai10] P. É. Cailliau. *Modélisation et résolution de problèmes de synthèse dimensionnelle de mécanismes*. PhD thesis, École Centrale de Paris, 2010.
- [Car46] E. Cartan. *Leçons sur la géométrie des espaces de Riemann*. Jacques Gabay, 1946.
- [Cau31] A. L. Cauchy. Sur les polygones et polyèdres, second mémoire. *J. Ecole Polytechnique*, 9 :87, 1831.
- [Che03] Y. Chen. *Design of Structural Mechanisms*. PhD thesis, Department of Engineering Science at the University of Oxford, 2003.
- [Che11] C. Chen. The order of local mobility of mechanisms. *Mechanism and Machine Theory*, 46 :1251–1264, 2011.
- [Cho88] S.-C. Chou. *Mechanical Geometry Theorem Proving*. D. Reidel Publishing Co., 1988.
- [Cho98] X-S. Gao & S-C. Chou. Solving geometric constraint systems. ii. a symbolic approach and decision if re-constructibility. *Computer-Aided Design*, 30, no. 2 :115–122, 1998.



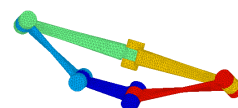
- [Cho11] R. Liu & P. Serré & J.-Y. Choley. An approach to manage over-constrained mechatronic system in teamwork. In *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA 2011)*, volume 2, pages 983–988, Beijing (China), August 7-11, 2011.
- [CIP] CIP. Competitiveness and Innovation framework Programme. Url : http://ec.europa.eu/information_society/activities/ict_psp/index_en.htm.
- [Clé09] P. Serré & F. MHenni & A. Clément. A Parametric Approach to Determine Minimum Clearance in Overconstrained Mechanisms. In *11th CIRP International Seminar on Computer-Aided Tolerancing, Annecy, France, March 26-27, 2009*, 2009.
- [Cua06] M. González & D. Dopico & U. Lugrís & J. Cuadrado. A benchmarking system for mbs simulation software : Problem standardization and performance measurement. *Multibody System Dynamics*, 16 (2) :179–190, 2006.
- [Dah09] L. Racila & M. Dahan. Méthode géométrique de calcul pour un translateur 6R. In *19ème Congrès Français de Mécanique*, 2009.
- [Dav87] E. Davis. Constraint propagation with interval labels. *AI*, 32(3) :281–331, 1987.
- [Die95] P. Dietmaier. A new 6R space mechanism. in : *Proceeding of 9th World Congress IFToMM, Milano*, 1 :52–56, 1995.
- [Din] Mohab Safey El Din. Algorithmes efficaces en géométrie algébrique réelle. 2007, Url : <http://www-polsys.lip6.fr/safey/Articles/coursJNCF.pdf>.
- [Din09] J.C. Faugère & M. Safey El Din. *De l'algèbre linéaire à la résolution des systèmes polynomiaux*, chapter 1. 2009.
- [Duf00] C. Essert-Villard & P. Schreck & J.-F. Dufourd. Sketch-based pruning of a solution space within a formal geometric constraint solver. *Artificial Intelligence Journal*, 124(1) :139–159, Nov. 2000.
- [Dur98] C. Durand. *Symbolic and numerical techniques for constraint solving*. PhD thesis, Department of Computer Science, Purdue University, December, 1998.
- [EIG] Spécification, contraintes et preuves. Url : http://icubegg.unistra.fr/index.php/Sp%C3%A9cifications,_contraintes_et_preuves.
- [Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases F4. *Journal of Pure and Applied Algebra*, 139(1) :61–68, 1999.
- [Fau02] Jean-Charles Faugère. A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC '02*, pages 75–83, New York, NY, USA, 2002. ACM.
- [Fou04] D. Michelucci & S. Foufou. Using Cayley-Menger determinants for geometric constraint solving. *ACM Symposium on Solid Modeling and Application*, 2004.
- [Fou06] D. Michelucci & S. Foufou. Geometric constraint solving : the witness configuration method. *Computer-Aided Design*, 38 :284–299, 2006.



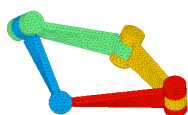
- [Fou07] D. Michelucci & S. Foufou. Detecting all dependences in systems of geometric constraints using the witness method. *Automated Deduction in Geometry*, pages 98–112, 2007.
- [Gir97] G. Colombari & J. Giraud. *Science industrielles, classes préparatoires 1re année*. 1997.
- [Gir09] G. Colombari & J. Giraud. *Science industrielles pour l'ingénieur, classes préparatoires 2ème année*. 2009.
- [Glu71] H. Gluck. The converse to the four-vertex theorem. *L'Enseignement Math.*, 17 :295–309, 1971.
- [Gog05] Grigore Gogu. Mobility of mechanisms : a critical review. *Mechanism and Machine Theory*, 40 :1068–1097, 2005.
- [Gog08] G. Gogu. *Structural Synthesis of Parallel Robots*, volume 149 of *Solid Mechanics and Its Applications*, chapter Part 1 : Methodology, page 716. Springer, 2008.
- [Gog09] G. Gogu. *Structural Synthesis of Parallel Robots*, chapter Part 2 : Translational Topologies with Two and Three Degrees of Freedom, page 775. Solid Mechanics and Its Applications. Springer, 2009.
- [Gol43] M. Goldberg. New five-bar and six-bar linkages in three dimensions. *ASME Trans*, 65 :183–210, 1943.
- [Grü17] Grübler. *Getriebelehre*. Springer, Berlin, 1917.
- [Gra02] J. Graver. Counting on frameworks : Mathematics to aid the design of rigid structures. *Dolciani Mathematical Expositions. Mathematical Association of America*, 25, 2002.
- [Hen92] B. Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21 :65–84, 1992.
- [Her78] J. Hervé. Principes fondamentaux d'une théorie des mécanismes. *Revue Roumaine des Sciences Techniques - Série de Mécanique Appliquée*, 23(5) :693–709, 1978.
- [Hob] Hober. Introduction et développement de l'informatique à la SAT. Url : <http://www.satsouvenir.fr/index.php?p=4-3&contrib=71>.
- [Hua97] C. Huang. The cylindroid associated with finite motions of the bennett mechanism. *Journal of Mechanical Design*, 119(4) :521–524, 1997.
- [Hwa96] J. M. Lim & K. S. Kim & B.J. Yum & H. Hwang. Determination of an optimal configuration of operating policies for direct-input-output manufacturing systems using the Taguchi method. *Computers Industrial Engineering*, 31(3/4) :555–560, 1996.
- [Imb13] Rémi Imbach. *Résolution de contraintes géométriques en guidant une méthode homotopique par la géométrie*. PhD thesis, Université de Strasbourg, 2013.
- [Jer02] C. Jermann. *Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles*. PhD thesis, L'université de Nice-Sophia Antipolis UFR Sciences, 2002.



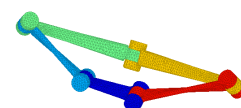
- [Kap86] D. Kapur. Using Gröbner bases to reason about geometry problems. *J. Symbolic-Computation*, 2(4) :399–408, 1986.
- [Kha08] B. Siciliano & O. Khatib, editor. *Springer handbook of robotics*. Springer, 2008.
- [Lam70] G. Laman. Theorem and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4 :331–340, 1970.
- [Li09] Y. Li. New method to extend Macaulay resultant. *In Second International Conference on Intelligent Computation Technology and Automation, ICICTA '09*, 4 :562–565, 2009.
- [Lic83] A. Lichnerowicz. *Algèbre et analyse linéaires*. Masson, 1983.
- [Lom] H. Lombardi. Un nouvel algorithme de calcul d'une base de Gröbner. Janvier, 1998. Url : <http://hlombardi.free.fr/publis/Smic-Smac.pdf>.
- [Mad99] I. E. Madu & C. N. Madu. Design optimization using signal-to-noise ration. *Simulation and Practice*, 7 :349–372, 1999.
- [Man11] G. Mandil. *Modèle de représentation géométrique intégrant les états physiques du produit*. PhD thesis, Supméca Paris France and Sherbrooke Québec Canada, 2011.
- [Mat97] P. Mathis. *Constructions géométriques sous contraintes en modélisation à base topologique*. PhD thesis, Université Louis Pasteur, Strasbourg, 1997.
- [Mat04] C. Essert-Villard & P. Mathis. *Interactive Handling of a Construction Plan in CAD*, chapter 3, pages 413–429. Springer Netherlands, 2004.
- [McC03] A. Perez & J. M. McCarthy. Dimensional synthesis of Bennett linkages. *ASME Journal of mechanical design*, 125 :98–104, 2003.
- [McK04] Clint McKay. An analysis of improvements to Buchberger's algorithm for Gröbner basis computation. Master's thesis, University of Maryland, College Park, 2004.
- [Mel04] N. O. Melin. *Application of Bennett Mechanisms to Long-Span Shelters*. PhD thesis, Department of Engineering Science at the University of Oxford, 2004.
- [Mic95] H. Lamure & D. Michelucci. Solving geometric constraints by homotopy. In *Third Symposium on Solid Modelling and its Applications*, 1995.
- [Mid96] R. Latham & A. Middleditch. Connexivity analysis : a tool for processing geometric constraints. *Computer-Aided Design*, 28, no. 11 :917–928, 1996.
- [Mil64] J. Milnor. On the Betti numbers of real varieties. In *Proceedings of the American Mathematical Society*, volume 15, pages 275–280, 1964.
- [Moi08] M.T. Moinet. *Descriptions non cartésiennes et résolution de problèmes géométriques sous contraintes*. PhD thesis, École Centrale Paris, 2008.
- [Moi10] G. Mandil & P. Serré & M. Moinet. Geometric tool for the description and control of functional specification at the conceptual design phase. *International Journal of Design and Innovation Research (IJODIR)*, 5 :23–39, 2010.

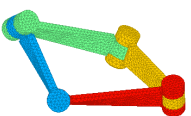


- [Mya31] F. E. Myard. Contribution à la géométrie des systèmes articulés. *Société Mathématiques de France*, 59 :183–21, 1931.
- [Ort68] J. M. Ortega. The Newton-Kantorovich Theorem. *The American Mathematical Monthly*, 75(6) :658–660, Jun. - Jul. 1968.
- [O'S00] D. Cox & J. Little & D.-D. O'Shea. *Ideals, Varieties and Algorithms*, chapter An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, 2000.
- [Pro] ThinkMOTION Project. ThinkMOTION Project - Competitiveness and Innovation framework Programme. Url : <http://www.thinkmotion.eu/>.
- [Riv02] P. Serré & A. Clément & A. Rivière. Vers une approche déclarative en CFAO application au mécanisme de Bennett. In *Proceedings of the 4th International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME)*, 2002.
- [Riv06] P. Serré & A. Ortuzar & A. Rivière. Non-cartesian modelling for analysis of the consistency of a geometric specification for conceptual design. *International Journal of Computational Geometry and Applications*, 16(5/6) :549–565, 2006.
- [Riv07] Faïda Mhenni & Philippe Serré & Abdelfattah Mlika & Lotfi Romdhane & Alain Rivière. Dependency between dimensional deviations in overconstrained mechanisms. In *Conception et Production Intégrées*, 2007.
- [Rod40] Olindes Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace. *Journal de mathématiques pures et appliquées*, 5 :380–440, 1840.
- [Rot73] L. W. Tsai & B. Roth. A note on the design of Revolute-Revolute cranks. *Mechanisms and Machine Theory*, 3 :23–31, 1973.
- [Rot94] C. Mavroidis & B. Roth. Analysis and synthesis of overconstrained mechanisms. in : *Proceedings of the 1994 ASME Design Technical Conferences, Minneapolis, MI, September*, pages 115–133, 1994.
- [Sala] A. Bostan & B. Salvy. Identités de fonction spéciales et séries D-finies. Url : <http://perso.ens-lyon.fr/bruno.salvy/M1ENS/cours8.pdf>.
- [Salb] A. Bostan & B. Salvy. Somme hypergéométrique. Url : <http://perso.ens-lyon.fr/bruno.salvy/M1ENS/somme.pdf>.
- [Salc] A. Bostan & F. Chyzak & B. Salvy. D-finitude : algorithmes et applications. Url : <http://algo.inria.fr/EJCIM07/EJCIM07-LN.pdf>.
- [Sax95] D. Kapur & T. Saxena. Comparison of various multivariate resultant formulations. In *ISSAC'95 : Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, pages 187–194, 1995.
- [Sch98] J.F. Dufourd & P. Mathis & P. Schreck. Geometric construction by assembling solved subfigures. *Artificial Intelligence*, 99(1) :73–119, 1998.
- [Ta] F. Peyraut & M. Domaszewski & S. Gomes & D. Chamoret & N. Labed & S. Roth & N. Harb & A.T. Ta. Modélisation éléments finis en hyperélasticité. Technical report, Laboratoire Mécatronique, Méthodes, Modèles et Métiers.



- [Tem84] A. Clément & A. Rivière & M. Temmerman. *Théorie et pratique*, chapter Cotation tridimensionnelle des systèmes mécanique. 1984.
- [The01] I. Macdonald & J. Pach & T. Theobald. Common tangents to four unit balls. *Discr. Comp. Geometry*, 26 :1–17, 2001.
- [Tsu92] K. L. Tsui. An overview of Taguchi method and newly developed statistical methods for robust design. *IIE Transaction*, 24(5) :44–57, 1992.
- [Vel67] G.R. Veldkamp. Canonical systems and instantaneous invariants in spatial kinematics. *Journal of Mechanisms*, 3 :329–388, 1967.
- [Ver] Thibaut Verron. Bases de Gröbner, à la rencontre entre géométrie algébrique et algorithmique. Url : https://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDIQFjAA&url=http%3A%2F%2Fwww.math.ens.fr%2Fenseignement%2Ftelecharger_fichier.php%3Ffichier%3D788&ei=cC7QUsrCCIZb7Ab1_YG4CA&usg=AFQjCNGTsO9RI9n5jajV8-xNsjEdZeSXjw&sig2=X7t99QjITg0TBThG76ohhA&bvm=bv.59026428,d.Yms&cad=rja.
- [Wal07] A.H. Wallace. *An Introduction to algebraic topology*, Dover books on Mathematics. 2007.
- [Whi98] W. Whiteley. *Rigidity and scene analysis*, Handbook for discrete and computational geometry. CRC Press LLC, 1998.
- [Win90] F. Winkler. Gröbner bases in geometry theorem proving and simplest degeneracy conditions. *Mathematica Pannonica*, 1(1) :15–32, 1990.
- [Woh91] K. Wohlhart. Merging two general Goldberg 5R linkages to obtain a new 6R space mechanism. *Mechanism and Machine Theory*, 26(2) :659–668, 1991.
- [Wu86] W. Wu. Basic principles of mechanical theorem proving in elementary geometries. *J. Automated Reasoning*, 2 :221–254, 1986.
- [Yan02] Qiong Jin & Tingli Yang. Overconstraint analysis on spatial 6-link loops. *Mechanism and machine theory*, 37 :267–278, 2002.
- [You07] Y. Chen & Z. You. Spatial 6R linkages based on the combination of two Goldberg 5R linkage. *Mechanism and Machine Theory*, 42 :1484–1498, 2007.
- [You08] Y. Chen & Z. You. An extended Myard linkage and its derived 6R linkage. *Journal of Mechanical Design*, 130(5) :23011–23018, 2008.





Annexe A

Annexes de l'État de l'art

A.1 Quelques rappels

A.1.1 Espace vectoriels

Un espace vectoriel est un ensemble d'éléments, dénommés vecteurs, possédant un ensemble de propriétés axiomatiques intéressantes. Soit K un corps commutatif, les éléments du corps K sont appelés des scalaires.

$(E, +, \cdot)$ est un K -espace vectoriel :

- $+$ est une loi de composition interne ;
- \cdot est une loi de composition externe à opérateurs dans K ;
- $(E, +)$ est un groupe commutatif ;
- distributivité par rapport à l'addition des vecteurs :
 $\forall \lambda \in K, \forall (\vec{u}, \vec{v}) \in E \times E, \lambda \cdot (\vec{u} + \vec{v}) = (\lambda \cdot \vec{u}) + (\lambda \cdot \vec{v})$
- distributivité par rapport à l'addition des scalaires :
 $\forall (\lambda, \mu) \in K \times K, \forall \vec{u} \in E, (\lambda + \mu) \cdot \vec{u} = (\lambda \cdot \vec{u}) + (\mu \cdot \vec{u})$
- associativité : $\forall (\lambda, \mu) \in K \times K, \forall \vec{u} \in E, \lambda \cdot (\mu \cdot \vec{u}) = (\lambda \cdot \mu) \cdot \vec{u}$
- élément neutre : $\forall \vec{u} \in E, 1 \cdot \vec{u} = \vec{u}$.

A.1.2 Espace affine

Soit E un ensemble et V un K -espace vectoriel. Soit $\varphi : E \times E \rightarrow V$ une application satisfaisant :

- $\forall (A, B) \in E^2, \forall \vec{v} \in V, A + v = B \Leftrightarrow AB = \vec{v}$,
- $\forall (A, B, C) \in E \times E \times E, \varphi(A, B) + \varphi(B, C) = \varphi(A, C)$.

Le couple (E, φ) est appelé espace affine défini sur V .

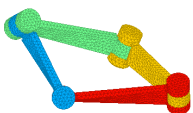
A.1.3 Espace vectoriel normé

On dit que l'espace vectoriel E est un espace préhilbertien si l'on a défini dans E un produit scalaire, c'est à dire si l'on a associé à tout couple \vec{X}, \vec{Y} de E un nombre complexe, noté $\langle \vec{X}, \vec{Y} \rangle$. D'autre part, un espace préhilbertien de dimension finie s'appelle un espace hermitien s'il est complexe, un espace euclidien s'il est réel. Il est démontré qu'un espace préhilbertien est un espace vectoriel normé, c'est à dire que le produit scalaire de deux vecteurs peut servir à définir une norme.

L'espace vectoriel normé E comporte les propriétés suivantes :

- E est normable et toutes les normes sur E sont équivalentes ;
- E est complet, c'est donc un espace de Banach ;
- les parties compactes de E sont les fermés bornés ;
- la boule unité fermée de E est compacte ;
- toute application linéaire de E dans un espace vectoriel normé quelconque est continue ;

Les éléments de cet espace sont aussi des vecteurs.



Annexe B

Annexes de la Résolution formelle avec les bases de Gröbner

B.1 Divers algorithmes basés sur les bases de Gröbner

Rappelons que les définitions de bases telles que $LC(f)$, $LM(f)$ et $M(f)$ se trouvent dans la section 1.5.1.2.

Pour commencer, il faut introduire la notion de réduction d'un polynôme f par rapport à un polynôme p , cette notion $f \xrightarrow{p} g$ est purement mathématique. Elle pourrait être paraphrasée par : “ f peut se réduire en g modulo p ”. Cette notion est ensuite étendue pour réduire un polynôme f par plusieurs polynômes. Il est donc nécessaire d'ordonner les polynômes d'un point de vue algorithmique puisque l'on manipule des listes de polynômes $F = [f_1, \dots, f_m]$. Il reste maintenant à définir, pour un ordre monomial fixé, la notion de réduction d'un polynôme, qui généralise la division euclidienne.

Définition B.1.0.1. Soient f , g et p dans le corps des rationnels $\mathbb{Q}[X_1, \dots, X_n]$ tels que $p \neq 0$. Alors on dit que :

1. f se réduit en g modulo p (notation $f \xrightarrow{p} g$), s'il existe $t \in M(f)$; et s'il existe $s \in M$ tels que $s \cdot LM(p) = t$ et $g = f - \frac{c}{LC(p)} \cdot s \cdot p$, où c est le coefficient de t dans f ;
2. f se réduit en g modulo P (notation $f \xrightarrow{P} g$) si $f \xrightarrow{p} g$ existe pour un certain $p \in P$;

Définition B.1.0.2. Soient f et g dans $\mathbb{Q}[X_1, \dots, X_n]$, et soit P un sous-ensemble fini de $\mathbb{Q}[X_1, \dots, X_n]$.

1. f est réductible modulo P s'il existe $g \in \mathbb{Q}[X_1, \dots, X_n]$ tel que $f \xrightarrow{P} g$;
2. f est top-réductible modulo P s'il existe $g \in \mathbb{Q}[X_1, \dots, X_n]$ tel que $f \xrightarrow{P} g$ et $LM(g) < LM(f)$.
3. f se réduit en g modulo P ($f \xrightarrow{*P} g$) si $g = f$ ou s'il existe $k \in \mathbb{N}$ et g_1, \dots, g_k dans $\mathbb{Q}[X_1, \dots, X_n]$ tels que $g_k = g$, $f \xrightarrow{P} g_1$, et pour $1 \leq i \leq k - 1$, $g \xrightarrow{P} g_{i+1}$.

Ainsi, l'algorithme de réduction d'un polynôme par une liste de polynômes est donné par

<p>Algorithm 1: Algorithme de RÉDUCTION</p> <p>Entrées: Un polynôme p, une liste de polynômes $F = [f_1, \dots, f_m]$ et un ordre admissible \prec</p> <p>Sorties: Un polynôme r tel que $p \xrightarrow{F} r$.</p> <p>Tant que $p \neq 0$ <i>et</i> p est top-réductible modulo F faire</p> <div style="margin-left: 20px;"> <p>$k = \min(i \in \{1, \dots, m\} LM(f_i) \text{ divise } LM(p));$</p> <p>$p := p - \frac{LT(p)}{LT(f_k)} f_k;$</p> </div> <p>finTantque</p> <p>Retourner p;</p>

L'algorithme 1 est le pendant multivarié de la division euclidienne de polynômes en une variable [Din09]. Elle se termine. La preuve se trouve dans [Din09]. Il est important de noter que la réduction d'un polynôme par un ensemble de polynômes n'est pas unique et que le résultat dépend, a priori, de la façon d'ordonner les polynômes dans la liste F .

Exemple B.1.0.1. Voici un exemple de réduction : $f = 9x^2 - y^2$, $f_1 = 3x - y$, $f_2 = xy^2 - 3$, l'ordre monomial \prec choisi ici est l'ordre lexicographique $y \prec x$.

1. On calcule $RÉDUCTION(f, [f_1, f_2], \prec)$: f est top-réductible modulo $[f_1, f_2]$ puisque $LM(f_1) = 3x$ divise $LM(f) = 9x^2$; On calcule donc $f' = f - \frac{9x^2}{3x} f_1 = 3xy - y^2$.
2. De nouveau f' est top-réductible modulo $[f_1, f_2]$ puisque $LM(f_1) = 3x$ divise $LM(f') = 3xy$; On calcule donc $f'' = f' - \frac{3xy}{3x} f_1 = 0$;
3. L'algorithme termine car $LM(f'') = 0$. Avec la notation mathématique, nous avons $f \xrightarrow{[f_1, f_2]}^* 0$.

Nous venons de définir la notion de réduction d'un polynôme. Maintenant des algorithmes de calcul des bases de Gröbner vont être présentés. La version la plus simple repose sur un couple de polynôme (f, g) pour lesquels on va calculer un S-polynôme. L'algorithme de Buchberger¹ [Buc65] et de Faugère (F4) sont tous basés sur ce calcul.

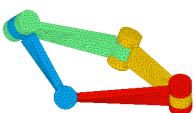
B.1.1 Buchberger

Définition B.1.1.1. (S-Polynôme). Soient deux polynômes non nuls p_1 et p_2 . Posons $m_1 = LM(p_1)$, $m_2 = LM(p_2)$ et $m = PPCM(m_1, m_2) = n_1 m_1 = n_2 m_2$, avec PPCM le *Plus Petit Commun Mutiple*. On appelle *S-polynôme* de deux polynômes p_1 et p_2 toute combinaison linéaire non nulle de la forme $l_1 p_1 + l_2 p_2$ pour tous polynômes non nuls l_1 et l_2 tels que $LM(l_i) = n_i$ et $LT(l_1)LT(p_1) + LT(l_2)LT(p_2) = 0$. En pratique, on se restreint aux termes de polynômes et on pose :

$$Spoly(p_1, p_2) = l_1 p_1 + l_2 p_2 \quad \text{pour} \quad l_1 = LC(p_2) n_1, \quad l_2 = -LC(p_1) n_2$$

Comme les *S-polynômes* sont éléments de l'idéal considéré, ils se réduisent nécessairement à zéro par toute base de Gröbner de l'idéal. L'algorithme rudimentaire de Buchberger se résume

1. du nom de son concepteur, Bruno Buchberger



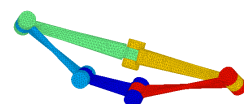
ci-dessous :

<p>Algorithm 2: Algorithme de Buchberger - Calcul des bases de Gröbner</p> <p>Entrées: Une liste de polynômes $F = [f_1, \dots, f_s]$ et un ordre admissible \prec</p> <p>Sorties: G un sous-ensemble fini de $\mathbb{Q}[X_1, \dots, X_n]$ qui est une base de Gröbner de $\langle f_1, \dots, f_s \rangle$.</p> <p>$G := F$ et $m := s$;</p> <p>$P := \{(f_i, f_j) 1 \leq i < j \leq m\}$;</p> <p>Tant que $P \neq \emptyset$ faire</p> <p style="padding-left: 20px;">Choisir et retirer de P une paire de polynômes (f, g);</p> <p style="padding-left: 20px;">$f_{m+1} := \text{Spol}(f, g)$;</p> <p style="padding-left: 20px;">$f_{m+1} := \text{RÉDUCTION}(f_{m+1}, G, \prec)$;</p> <p style="padding-left: 20px;">si $f_{m+1} \neq 0$ alors</p> <p style="padding-left: 40px;">$m := m + 1$;</p> <p style="padding-left: 40px;">$P := P \cup \{(f_i, f_m) 1 \leq i < m\}$;</p> <p style="padding-left: 40px;">$G := G \cup \{f_m\}$;</p> <p style="padding-left: 20px;">finSi</p> <p>finTantque</p> <p>Retourner G;</p>
--

Exemple B.1.1.1. Avec l'ordre lexicographique $y \prec x$ et la liste de polynôme $F = [f_1 = x^2y - 1, f_2 = xy^2 - 3]$, on applique l'algorithme de Buchberger.

1. $P = \{[f_1, f_2]\}$: on calcule $f_3 := \text{Spol}(f_1, f_2) = -y + 3x$ puis $f_3 := \text{RÉDUCTION}(f_3, G, \prec) = 3x - y$. On ajoute donc f_3 dans G : $G = [f_1, f_2, f_3]$ et $P = \{[f_1, f_3], [f_2, f_3]\}$.
2. $P = \{[f_1, f_3], [f_2, f_3]\}$: on calcule $f_4 := \text{Spol}(f_1, f_3) = -3 + xy^2$ puis $f_4 := \text{RÉDUCTION}(f_4, G, \prec) = 0$.
3. $P = \{[f_2, f_3]\}$: on calcule $f_4 := \text{Spol}(f_2, f_3) = -9 + y^3$ puis $f_4 := \text{RÉDUCTION}(f_4, G, \prec) = y^3 - 9$. On ajoute donc un nouvel élément dans G : $G = [f_1, f_2, f_3, f_4]$ et $P = \{[f_1, f_4], [f_2, f_4], [f_3, f_4]\}$.
4. $P = \{[f_1, f_4], [f_2, f_4], [f_3, f_4]\}$: on calcule $f_5 := \text{Spol}(f_1, f_4) = -y^2 + 9x^2$ puis $f_5 := \text{RÉDUCTION}(f_5, G, \prec) = 0$, voir l'exemple B.1.0.1.
5. $P = \{[f_2, f_4], [f_3, f_4]\}$: on calcule $f_5 := \text{Spol}(f_2, f_4) = -3y + 9x$ puis $f_5 := \text{RÉDUCTION}(f_5, G, \prec) = 0$.
6. $P = \{[f_3, f_4]\}$: on calcule $f_5 := \text{Spol}(f_3, f_4) = -y^4 + 27x$ puis $f_5 := \text{RÉDUCTION}(f_5, G, \prec) = 0$.
7. $P = \emptyset$: l'algorithme se termine et retourne $G = [x^2y - 1, xy^2 - 3, 3x - y, y^3 - 9]$.

L'algorithme de Buchberger se termine parce que les appels à la fonction RÉDUCTION se terminent. Le corollaire suivant donne un moyen algorithmique permettant de vérifier qu'une liste de polynômes est une base de Gröbner.



B.1.2 Existence et unicité

Corollaire B.1.2.1. (Buchberger) Soit G un sous-ensemble fini de polynômes dans $\mathbb{Q}[X_1, \dots, X_n]$. G est une base de Gröbner si et seulement si

1. $\text{Spol}(f, g) \xrightarrow[G]{*} 0$ pour tout $(f, g) \in G^2$ tels que $f \neq g$;
2. $\text{RÉDUCTION}(\text{Spol}(f, g), G, \prec) = 0$ pour tout $(f, g) \in G^2$.

Une des applications des bases de Gröbner est de détecter si le système admet des solutions. Ainsi, le système admet des solutions dans \mathbb{C}^n si et seulement si G ne contient pas 1.

Définition B.1.2.2. On dit que $V \subset \mathbb{C}^n$ est une \mathbb{Q} -variété algébrique s'il existe $f_1, \dots, f_k \in \mathbb{Q}[X_1, \dots, X_n]$ tel que

$$V = \{x \in \mathbb{C}^n \mid f_1(x) = \dots = f_k(x) = 0\}$$

Soit $x \in \mathbb{C}^n$ tel que $f_1(x) = \dots = f_k(x) = 0$. Il est alors immédiat que pour tout polynôme $p \in \langle f_1, \dots, f_k \rangle$, $p(x) = 0$.

Définition B.1.2.3. Soit I un idéal de $\mathbb{Q}[X_1, \dots, X_n]$. On appelle variété algébrique associée à I (noté $V(I)$ par la suite) le sous-ensemble de points $x \in \mathbb{C}^n$ tels que pour tout $p \in I$, $p(x) = 0$.

En effet, si $1 \in G$, comme tout polynôme de $\mathbb{Q}[X_1, \dots, X_n]$ se réduit à 0 par 1 donc $I = \mathbb{Q}[X_1, \dots, X_n]$ et $V(I) = \emptyset$. Réciproquement, si $V = \emptyset$, alors l'idéal associé à V est égal à $\mathbb{Q}[X_1, \dots, X_n]$. En particulier $1 \in I$ et comme G est une base de Gröbner, il existe $g \in G$ tel que $\text{LM}(g)$ divise 1, ce qui implique $g = 1$.

Le théorème suivant donne une caractérisation des bases de Gröbner des idéaux de dimension zéro.

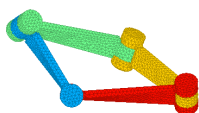
Définition B.1.2.4. Soit G une base de Gröbner dans $\mathbb{Q}[X_1, \dots, X_n]$. Si pour tout $i \in \{1, \dots, n\}$, il existe $p_i \in G$ tel que $\text{LM}(p_i) = X_i^{k_i}$ où $k_i \in \mathbb{N}$, alors l'ensemble des points annulant les polynômes de G est de dimension zéro.

Il est souligné dans la section 1.5.1.2 qu'une base de Gröbner selon la définition 1.5.1.7 n'est évidemment pas unique. Ainsi si g_1, g_2 est une base de Gröbner d'un idéal I , et si $f \in I$ alors $\{g_2 + g_1, g_2\}, \{g_1, g_2, f\}$ et $\{g_1, xg_2, g_2\}$ sont également des bases de Gröbner de I . Cette construction de bases de Gröbner possède une infinité de combinaison pour un idéal donné. En rendant tous les polynômes unitaires et en supprimant successivement les $g \in G$ tel que $\text{LT}(g) \in \langle \text{LT}(G \setminus \{g\}) \rangle$, on obtient une base de Gröbner *minimale*. L'une de ces bases est caractérisée par la définition suivante.

Définition B.1.2.5. (Base de Gröbner réduite) Soit I un idéal de $\mathbb{Q}[X_1, \dots, X_n]$. Un sous-ensemble fini $G \subset I$ est une base de Gröbner réduite de I pour l'ordre \prec si

- $\text{LC}(g) = 1$ pour tout $g \in G$;
- pour tout $g \in G$, aucun monôme de g n'appartient à $\langle \text{LT}(G \setminus \{g\}) \rangle$.

Pour la suite, lorsque nous parlerons de "la base de Gröbner d'un idéal", il s'agira de la base de Gröbner réduite de cet idéal pour un ordre donné. Par exemple dans le cas de polynômes univariés, $\mathbb{Q}[x]$ est principal donc tout idéal est engendré par un unique élément unitaire, qui forme l'unique élément de la base de Gröbner réduite.



Une des propriétés des bases de Gröbner est l'unicité de la réduction d'un polynôme par une base de Gröbner [Bar04].

B.1.3 Amélioration de l'algorithme de Buchberger

B.1.3.1 Algorithme F4

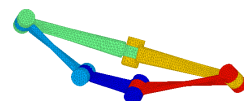
L'algorithme original de Buchberger est souvent impossible à exécuter en un temps raisonnable pour calculer les Bases de Gröbner de n'importe quel ensemble de polynômes. L'algorithme F4 développé par Faugère [Fau99, McK04] est une version améliorée de l'algorithme traditionnel de Buchberger. La version basique se résume à l'algorithme présenté ci-dessous. Les notations se trouvent dans [Fau99].

Algorithm 3: Algorithme F4

<p>Entrées: $\left\{ \begin{array}{l} F \text{ un sous-ensemble fini de } R[\underline{x}] \\ Sel \text{ une liste de fonction } List(Pairs) \rightarrow List(Pairs) \text{ tel que } Sel(l) \neq \emptyset \text{ si } l \neq \emptyset \end{array} \right.$</p> <p>Sorties: un sous-ensemble fini de $R[\underline{x}]$</p> <p>$G := F, \tilde{F}_0^+ = F$ et $d := 0$</p> <p>$P := \{(Pair(f, g) f, g \in G \text{ avec } f \neq g)\}$</p> <p>Tant que $P \neq \emptyset$ faire</p> <p style="padding-left: 2em;">$d := d + 1$</p> <p style="padding-left: 2em;">$P_d := Sel(P)$</p> <p style="padding-left: 2em;">$P := P \setminus P_d$</p> <p style="padding-left: 2em;">$F := Spol(P_d)$</p> <p style="padding-left: 2em;">$\tilde{F}_d^+ = Reduction(L_d, G)$</p> <p style="padding-left: 2em;">pour $h \in \tilde{F}_d^+$ faire</p> <p style="padding-left: 4em;">$P := P \cup \{Pair(h, g) g \in G\}$</p> <p style="padding-left: 4em;">$G := G \cup \{h\}$</p> <p style="padding-left: 2em;">fin</p> <p>finTantque</p> <p>Retourner G</p>
--

B.1.3.2 Algorithme F5

Faugère a publié un nouvel algorithme, nommé F5, pour calculer une Base de Gröbner sans réduire f à 0 [Fau02]. Cependant, d'un point de vue efficacité, il est recommandé d'utiliser F4. La seule différence structurale entre F5 et l'algorithme de Buchberger est la réduction d'un polynôme par rapport à une liste de polynômes peut retourner plusieurs polynômes. Cet algorithme utilise 3 fonctions auxiliaires : les définitions de "CritPair" (construction de pair critique si le nouveau critère ne peut pas être appliqué), "Spol" (construction du Spolynôme) et "Reduction" (réduction de polynômes par rapport à la liste courant) sont reportées jusqu'à



la fin de la section [Fau02].

Algorithm 4: Algorithmme *F5*

Entrées: $\begin{cases} i \text{ un entier et } f_i \text{ un polynôme} \\ G_{i+1} \text{ un sous-ensemble finit du } R, \text{ tel que } \text{poly}(G_{i+1}) \\ \text{est une base de Gröbner de } Id(f_{i+1}, \dots, f_m) \end{cases}$

$r_i := (\mathbf{F}_i, f_i) \in R$
 $\varphi_{i+1} = \text{NF}(\cdot, \text{poly}(G_{i+1}))$
 $G_i := G_{i+1} \cup \{r_i\}$
 $P := \text{Sort}([\text{CritPair}(r_i, r, i, \varphi_{i+1}) | r \in G_{i+1}])$ by degree

Tant que $P \neq \emptyset$ **faire**

$d := \text{deg}(\text{premier}(P))$
 $P_d := \{p \in P | \text{deg}(p) = d\}$
 $P := P \setminus P_d$
 $F := \text{Spol}(P_d)$
 $R_d := \text{Reduction}(F, G_i, i, \varphi_{i+1})$

pour $r \in R_d$ **faire**

$P := P \cup \{\text{CritPair}(r, p, i, \varphi_{i+1}) | p \in G_i\}$
 $G_i := G_i \cup \{r\}$

fin

$P := \text{Sort } P$ for the degree

finTantque

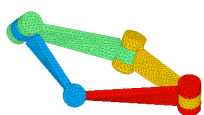
Retourner G_i

B.1.4 Autre algorithmme

Il existe un autre algorithmme de calcul d'une Base de Gröbner. Il n'utilise ni divisions ni S -polynômes. Il est basé sur le contrôle de la croissance du rang des matrices de Sylvester généralisées successives. Ces matrices sont définies de manière incrémentale en rajoutant à chaque étape de nouvelles colonnes et les lignes nécessaires. L'idée principale est de borner inférieurement et supérieurement les rangs des futures matrices en tant que fonctions du numéro de l'étape en cours, jusqu'à ce que ces fonctions coïncident [Lom].

B.2 Simplification d'une équation d'assemblage de la famille du système à 4 barres

Dans le 4^{ème} chapitre, nous obtenons 6 polynômes avec l'aide de l'algorithmme du calcul des bases de Gröbner. Les 6 polynômes peuvent s'écrire sous une forme plus courte par rapport aux



équations (B.1) ~ (B.6). Les 6 équations correspondantes aux polynômes sont :

$$\begin{aligned}
 & -2 c^2 L^3 L^1^2 - 2 c^3 L^3 L^1^2 + L^4 - 2 L^4 L^3 L^2 + 8 c_4 c_3 c_2 c_1 L^3 L^1^2 - 2 c^3 c^1 L^3 L^2 L^2 + \\
 & c^3 c^1 L^2 L^4 - 2 c^3 c^1 L^4 L^1^2 - 2 c^3 c^1 L^3 L^1^2 - 2 c^3 c^1 L^2 L^1^2 + c^3 c^1 L^1^4 - c^1 L^4 - \\
 & c^3 L^4 + 2 c^3 L^4 L^3 L^2 + 2 c^1 L^4 L^3 L^2 - c^3 L^3 L^4 + 2 c^3 L^3 L^2 L^2 + 2 c^1 L^3 L^2 L^2 - c^3 L^2 L^4 + \\
 & 2 c^3 L^2 L^1^2 + 2 c^1 L^2 L^1^2 + 2 c^3 L^2 L^1^2 + 2 c^1 L^2 L^1^2 - c^3 L^1^4 - 2 c^4 c^2 L^3 L^1^2 + \\
 & c^3 c^1 L^4 - 2 c^3 c^1 L^4 L^3 L^2 + c^3 c^1 L^3 L^4 - c^1 L^3 L^4 - c^1 L^2 L^4 - c^1 L^1^4 - 2 c^4 L^3 L^1^2 - 2 c^1 L^3 L^1^2 + \\
 & L^1^4 + 4 L^3 L^1^2 - 2 L^3 L^2 L^2 - 2 L^4 L^1^2 - 2 L^2 L^1^2 + L^2^4 + L^3^4 = 0
 \end{aligned} \tag{B.1}$$

$$s_3 s_1 L_4 L_2 - s_4 s_2 L_3 L_1 = 0 \tag{B.2}$$

$$s_1^2 + c_1^2 - 1 = 0 \tag{B.3}$$

$$s_2^2 + c_2^2 - 1 = 0 \tag{B.4}$$

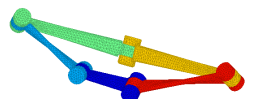
$$s_3^2 + c_3^2 - 1 = 0 \tag{B.5}$$

$$s_4^2 + c_4^2 - 1 = 0 \tag{B.6}$$

$$\tag{B.7}$$

L'équation (B.1) peut être simplifiée :

$$\begin{aligned}
 & -2 c^2 L^3 L^1^2 - 2 c^3 L^3 L^1^2 + L^4 - 2 L^4 L^3 L^2 + 8 c_4 c_3 c_2 c_1 L^3 L^1^2 - 2 c^3 c^1 L^3 L^2 L^2 + \\
 & c^3 c^1 L^2 L^4 - 2 c^3 c^1 L^4 L^1^2 - 2 c^3 c^1 L^3 L^1^2 - 2 c^3 c^1 L^2 L^1^2 + c^3 c^1 L^1^4 - c^1 L^4 - \\
 & c^3 L^4 + 2 c^3 L^4 L^3 L^2 + 2 c^1 L^4 L^3 L^2 - c^3 L^3 L^4 + 2 c^3 L^3 L^2 L^2 + 2 c^1 L^3 L^2 L^2 - c^3 L^2 L^4 + \\
 & 2 c^3 L^2 L^1^2 + 2 c^1 L^2 L^1^2 + 2 c^3 L^2 L^1^2 + 2 c^1 L^2 L^1^2 - c^3 L^1^4 - 2 c^4 c^2 L^3 L^1^2 + \\
 & c^3 c^1 L^4 - 2 c^3 c^1 L^4 L^3 L^2 + c^3 c^1 L^3 L^4 - c^1 L^3 L^4 - c^1 L^2 L^4 - c^1 L^1^4 - 2 c^4 L^3 L^1^2 - \\
 & 2 c^1 L^3 L^1^2 + L^1^4 + 4 L^3 L^1^2 - 2 L^3 L^2 L^2 - 2 L^4 L^1^2 - 2 L^2 L^1^2 + L^2^4 + L^3^4 \\
 & = 8 c_4 c_3 c_2 c_1 L^3 L^1^2 - 2 c^2 L^3 L^1^2 - 2 c^3 L^3 L^1^2 - 2 L^4 L^2 L^2 + 2 c^3 L^3 L^2 L^2 + 2 c^1 L^3 L^2 L^2 - \\
 & c^3 L^2 L^4 + 2 c^3 L^4 L^1^2 + 2 c^1 L^4 L^1^2 + 2 c^3 L^2 L^1^2 + 2 c^1 L^2 L^1^2 - c^3 L^1^4 - c^1 L^3 L^4 - c^1 L^2 L^4 - \\
 & c^1 L^1^4 + (L^2 - L^1^2 - L^3^2 + L^4^2)^2 + c^3 c^1 L^4 - 2 c^3 c^1 L^4 L^3 L^2 + c^3 c^1 L^3 L^4 - 2 c^3 c^1 L^3 L^2 L^2 + \\
 & c^3 c^1 L^2 L^4 - 2 c^3 c^1 L^4 L^1^2 - 2 c^3 c^1 L^3 L^1^2 - 2 c^3 c^1 L^2 L^1^2 + c^3 c^1 L^1^4 - c^1 L^4 - c^3 L^4 + \\
 & 2 c^3 L^4 L^3 L^2 + 2 c^1 L^4 L^3 L^2 - c^3 L^3 L^4 - 2 c^4 c^2 L^3 L^1^2 + 2 L^3 L^1^2 - 2 c^4 L^3 L^1^2 - 2 c^1 L^3 L^1^2 \\
 & = (L^2 - L^1^2 - L^3^2 + L^4^2)^2 + 2 L^3 L^1^2 - 2 L^4 L^2 L^2 + 2 L^1 L^3 L^2 (4 c_1 c_2 c_3 c_4 - c^2 - c^2 c^4 - c^4) - \\
 & c^3 (2 L^3 L^1^2 - 2 L^3 L^2 L^2 + L^2^4 - 2 L^4 L^1^2 - 2 L^2 L^1^2 + L^1^4 + L^4 - 2 L^4 L^3 L^2 + L^3^4) - \\
 & c^1 (2 L^3 L^1^2 - 2 L^3 L^2 L^2 + L^2^4 - 2 L^4 L^1^2 - 2 L^2 L^1^2 + L^1^4 + L^4 - 2 L^4 L^3 L^2 + L^3^4) + \\
 & c^1 c^3 (-2 L^3 L^1^2 - 2 L^3 L^2 L^2 + L^2^4 - 2 L^4 L^1^2 - 2 L^2 L^1^2 + L^1^4 + L^4 - 2 L^4 L^3 L^2 + L^3^4) \\
 & = (L^2 - L^1^2 - L^3^2 + L^4^2)^2 + 2 L^3 L^1^2 - 2 L^4 L^2 L^2 + 2 L^1 L^3 L^2 (4 c_1 c_2 c_3 c_4 - c^2 - c^2 c^4 - c^4) - \\
 & c^3 ((L^2 - L^1^2 - L^3^2 + L^4^2)^2 - 2 L^4 L^2 L^2) - c^1 ((L^2 - L^1^2 - L^3^2 + L^4^2)^2 - 2 L^4 L^2 L^2) + \\
 & c^1 c^3 (L^2 - L^1^2 - L^3^2 + L^4^2)^2 - 2 L^4 L^2 L^2 - 4 L^3 L^1^2 \\
 & = s_1^2 s_3^2 (L^2 - L^1^2 - L^3^2 + L^4^2)^2 - 2 s_3^2 s_1^2 L^4 L^2 L^2 + 2 L^1 L^3 L^2 (1 + 4 c_1 c_2 c_3 c_4 - c^2 - c^2 c^4 - \\
 & c^4 - 2 c^1 c^3) \\
 & = s_1^2 s_3^2 (L^2 - L^1^2 - L^3^2 + L^4^2)^2 - 2 s_3^2 s_1^2 L^4 L^2 L^2 + L^1 L^3 L^2 (2 s_2^2 s_4^2 - (2 c_1 c_3 - 2 c_2 c_4)^2) \\
 & = (L^1^2 - L^2^2 + L^3^2 - L^4^2)^2 - \left(\frac{2 L_1 L_3 c_1 c_3}{s_1 s_3} - \frac{2 L_1 L_3 c_2 c_4}{s_1 s_3} \right)^2 \\
 & = 0
 \end{aligned} \tag{B.8}$$



Supposons $s_4 s_2 \neq 0$, l'équation (B.2) peut se transformer en :

$$L_1 L_3 = \frac{s_3 s_1 L_4 L_2}{s_4 s_2} \quad (\text{B.9})$$

Substituons l'équation (B.9) dans l'équation (B.8) :

$$\begin{aligned} & (L_1^2 - L_2^2 + L_3^2 - L_4^2)^2 - \left(\frac{2 L_1 L_3 c_1 c_3}{s_1 s_3} - \frac{2 L_1 L_3 c_2 c_4}{s_1 s_3} \right)^2 \\ &= (L_1^2 - L_2^2 + L_3^2 - L_4^2)^2 - \left(\frac{2 L_1 L_3 c_1 c_3}{s_1 s_3} - \frac{2 L_2 L_4 c_2 c_4}{s_2 s_4} \right)^2 \\ &= 0 \end{aligned} \quad (\text{B.10})$$

Finalement, nous avons

$$\begin{aligned} & (L_1^2 - L_2^2 + L_3^2 - L_4^2)^2 - \left(\frac{2 L_1 L_3 c_1 c_3}{s_1 s_3} - \frac{2 L_1 L_3 c_2 c_4}{s_1 s_3} \right)^2 \\ &= (L_1^2 - L_2^2 + L_3^2 - L_4^2)^2 - \left(\frac{2 L_1 L_3}{\tan(a_1) \tan(a_3)} - \frac{2 L_4 L_2}{\tan(a_4) \tan(a_2)} \right)^2 \\ &= 0 \end{aligned} \quad (\text{B.11})$$

$$s_3 s_1 L_4 L_2 - s_4 s_2 L_3 L_1 = \sin(a_3) \sin(a_1) L_4 L_2 - \sin(a_2) \sin(a_4) L_3 L_1 = 0 \quad (\text{B.12})$$

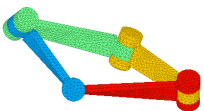
Les équations (B.11-B.12) sont les conditions d'assemblage pour les systèmes de solides 4R en 3D. A noter que les relations trigonométriques doivent également être respectées.

$$\left\{ \begin{array}{l} (L_1^2 - L_2^2 + L_3^2 - L_4^2)^2 - \left(\frac{2 L_1 L_3}{\tan(a_1) \tan(a_3)} - \frac{2 L_4 L_2}{\tan(a_4) \tan(a_2)} \right)^2 = 0 \\ \sin(a_3) \sin(a_1) L_4 L_2 - \sin(a_2) \sin(a_4) L_3 L_1 = 0 \end{array} \right. \quad (\text{B.13})$$

B.3 Application de l'algorithme de conditions de mobilité avec la méthode Jin et Yang

Il est listé dans chapitre 3 qu'avec la méthode Jin et Yang, les équations d'un système de 4R est :

$$\left\{ \begin{array}{l} \cos a_2 \cdot \cos a_3 - \sin a_3 \cdot \sin a_2 \cdot \cos tC - \cos a_4 \cdot \cos a_1 + \sin a_1 \cdot \sin a_4 \cdot \cos tA = 0 \\ L_2(\cos a_3 \cdot \sin a_2 + \cos a_2 \cdot \sin a_3 \cdot \cos tC) + L_3(\cos a_2 \cdot \sin a_3 + \cos a_3 \cdot \sin a_2 \cdot \cos tC) \\ - L_4(\cos a_1 \cdot \sin a_4 + \cos a_4 \cdot \sin a_1 \cdot \cos tA) - L_1(\cos a_4 \cdot \sin a_1 + \cos a_1 \cdot \sin a_4 \cdot \cos tA) = 0 \\ L_3 \cdot \sin a_2 \cdot \sin tC + L_4 \cdot \sin tA \cdot \sin a_1 = 0 \\ L_2 \cdot \sin tC \cdot \sin a_3 + L_1 \cdot \sin a_4 \cdot \sin tA = 0 \end{array} \right. \quad (\text{B.14})$$



Application de l'algorithme de conditions de mobilité avec la méthode Jin et Yang

Comme pour l'obtention des conditions d'assemblage, le paquet d'équations doit se transformer en système polynômial dans la phase d'initialisation. Avec la mise en équation de Jin&Yang, nous n'avons pas suivi exactement la méthode proposée dans le chapitre 4, ce qui a été proposé précédemment. Car avec un portable standard², le temps du calcul de l'élimination pour cet exemple est trop élevé. Après une première observation, les 4 équations peuvent être regroupées en deux en fonction de tA et tC : dans chaque groupe, il y a à la fois $\sin tA$, $\cos tA$, $\sin tC$, $\cos tC$.

$$syst1 : \begin{cases} L_2(\cos a3 \cdot \sin a2 + \cos a2 \cdot \sin a3 \cdot \cos tC) + L_3(\cos a2 \cdot \sin a3 + \cos a3 \cdot \sin a2 \cdot \cos tC) \\ -L_4(\cos a1 \cdot \sin a4 + \cos a4 \cdot \sin a1 \cdot \cos tA) - L_1(\cos a4 \cdot \sin a1 + \cos a1 \cdot \sin a4 \cdot \cos tA) = 0 \\ L_3 \cdot \sin a2 \cdot \sin tC + L_4 \cdot \sin tA \cdot \sin a1 = 0 \end{cases} \quad (B.15)$$

$$syst2 : \begin{cases} \cos a2 \cdot \cos a3 - \sin a3 \cdot \sin a2 \cdot \cos tC - \cos a4 \cdot \cos a1 + \sin a1 \cdot \sin a4 \cdot \cos tA = 0 \\ L_2 \cdot \sin tC \cdot \sin a3 + L_1 \cdot \sin a4 \cdot \sin tA = 0 \end{cases} \quad (B.16)$$

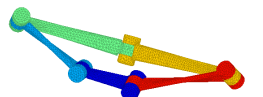
Les variables $\sin(ai)$, $\cos(ai)$, $\sin(ti)$ et $\cos(ti)$ sont ensuite remplacées par les symboles si et ci . Les identités trigonométriques sont des relations supplémentaires à ajouter dans chacun des groupes. Ainsi, il y a 4 inconnues dans les équations (B.15) et (B.16) : sA , cA , sC , cC .

$$syst1 : \begin{cases} L2(c3 s2 + c2 s3 cC) + L3(c2 s3 + c3 s2 cC) - L4(c1 s4 + c4 s1 cA) - L1(c4 s1 + c1 s4 cA) = 0 \\ L3 s2 sC + L4 sA s1 = 0 \\ cA^2 + sA^2 - 1 = 0 \\ cC^2 + sC^2 - 1 = 0 \end{cases} \quad (B.17)$$

$$syst2 : \begin{cases} c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA = 0 \\ L2 sC s3 + L1 s4 sA = 0 \\ cA^2 + sA^2 - 1 = 0 \\ cC^2 + sC^2 - 1 = 0 \end{cases} \quad (B.18)$$

D'abord, nous allons éliminer 3 inconnues parmi les 4. Le choix est diver. Pour cet exemple, sC , cC et sA sont éliminés. Les équations (B.19) et (B.20) sont les résultats de l'élimination pour les équations (B.15) et (B.16), respectivement.

2. Processeur Intel Core i5 2.50 GHz, Système d'exploitation 64 bits et 4,00 Go de RAM



$$\begin{aligned}
 & cA^2 c4^2 s2^2 s1^2 L4^2 L3^2 - cA^2 c3^2 s2^2 s1^2 L4^2 L3^2 - 2 cA^2 s3 c3 s2 c2 s1^2 L4^2 L3 L2 - \\
 & cA^2 s3^2 c2^2 s1^2 L4^2 L2^2 + 2 cA^2 s4 c4 s2^2 s1 c1 L4 L3^2 L1 + cA^2 s4^2 s2^2 c1^2 L3^2 L1^2 + \\
 & 2 cA s4 c4 s2^2 s1 c1 L4^2 L3^2 - 2 cA c4 s3 s2^2 c2 s1 L4 L3^3 - 2 cA c4 c3 s2^3 s1 L4 L3^2 L2 + \\
 & 2 cA c4^2 s2^2 s1^2 L4 L3^2 L1 + 2 cA s4^2 s2^2 c1^2 L4 L3^2 L1 - 2 cA s4 s3 s2^2 c2 c1 L3^3 L1 - \\
 & 2 cA s4 c3 s2^3 c1 L3^2 L2 L1 + 2 cA s4 c4 s2^2 s1 c1 L3^2 L1^2 + c3^2 s2^2 s1^2 L4^2 L3^2 + s4^2 s2^2 c1^2 L4^2 L3^2 - \\
 & 2 s4 s3 s2^2 c2 c1 L4 L3^3 - c3^2 s2^4 L3^4 + s3^2 s2^2 c2^2 L3^4 + 2 s3 c3 s2 c2 s1^2 L4^2 L3 L2 - \\
 & 2 s4 c3 s2^3 c1 L4 L3^2 L2 + s3^2 c2^2 s1^2 L4^2 L2^2 + c3^2 s2^4 L3^2 L2^2 - s3^2 s2^2 c2^2 L3^2 L2^2 + \\
 & 2 s4 c4 s2^2 s1 c1 L4 L3^2 L1 - 2 c4 s3 s2^2 c2 s1 L3^3 L1 - 2 c4 c3 s2^3 s1 L3^2 L2 L1 + c4^2 s2^2 s1^2 L3^2 L1^2 = 0 \quad (B.19)
 \end{aligned}$$

$$\begin{aligned}
 & cA^2 s4^2 s1^2 L2^2 - cA^2 s4^2 s2^2 L1^2 + 2 cA s4 c3 c2 s1 L2^2 - 2 cA s4 c4 s1 c1 L2^2 - s3^2 s2^2 L2^2 + \\
 & c3^2 c2^2 L2^2 - 2 c4 c3 c2 c1 L2^2 + c4^2 c1^2 L2^2 + s4^2 s2^2 L1^2 = 0 \quad (B.20)
 \end{aligned}$$

Le degré de cA le plus élevé est deux. Les équations (B.19) et (B.20) sont univariées en fonction de cA ($\cos(tA)$). Concernant le ddl du mécanisme, les équations (B.19) et (B.20) doivent toujours être vérifiées quel que soit la valeur de cA . En suivant cette idée, nous avons récupéré les coefficients de cA^2 , cA et les termes constants de chaque équation. Nous les avons stockés dans deux listes différentes. Ainsi, deux nouveaux systèmes sont construits.

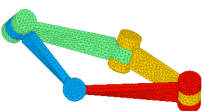
L'idée suivante est de calculer la base de Gröbner sur chaque liste de coefficients. Nous voulons que l'équation (B.19) et (B.20) soient respectées quelle que soit la valeur de cA . Cela veut dire que tous les coefficients de cA doivent être identiquement nuls.

$$\left\{ \begin{array}{l} \vdots \\ s2^2 L3^2 (c4^2 s1^2 L4^2 + L4^2 c1^2 s4^2 - 2 L3 s4 s3 c2 c1 L4 - L3^2 s2^2 c3^2 + c2^2 s3^2 L3^2 - \\ 2 s2 s4 c3 c1 L4 L2 + s2^2 c3^2 L2^2 - L2^2 c2^2 s3^2 + 4 L1 L4 c1 s1 c4 s4 - \\ 2 L3 c4 s3 c2 s1 L1 - 2 s2 c4 c3 s1 L2 L1 + L1^2 s1^2 c4^2 + c1^2 s4^2 L1^2) = 0 \\ -2 c4 c3 s2^3 s1 L3^2 L2 L1 + c3^2 s2^2 s1^2 L4^2 L3^2 + s4^2 s2^2 c1^2 L4^2 L3^2 - \\ 2 s4 s3 s2^2 c2 c1 L4 L3^3 - c3^2 s2^4 L3^4 + s3^2 s2^2 c2^2 L3^4 + 2 s3 c3 s2 c2 s1^2 L4^2 L3 L2 - \\ 2 s4 c3 s2^3 c1 L4 L3^2 L2 + s3^2 c2^2 s1^2 L4^2 L2^2 + c3^2 s2^4 L3^2 L2^2 - s3^2 s2^2 c2^2 L3^2 L2^2 + \\ 2 s4 c4 s2^2 s1 c1 L4 L3^2 L1 - 2 c4 s3 s2^2 c2 s1 L3^3 L1 + c4^2 s2^2 s1^2 L3^2 L1^2 = 0 \\ (L1 c1 s4 + c4 s1 L4) s2^2 L3^2 (-c3 L2 s2 - c2 s3 L3 + L4 c1 s4 + L1 c4 s1) = 0 \end{array} \right. \quad (B.21)$$

$$\left\{ \begin{array}{l} -(s1 s4 - s2 s3)(s1 s4 + s2 s3) = 0 \\ (-L2 s3 + L1 s4)(L1 s4 + L2 s3) = 0 \\ (-s1 L2 + L1 s2)(L1 s2 + s1 L2) = 0 \\ c2 c3 - c4 c1 = 0 \end{array} \right. \quad (B.22)$$

Ces résultats montrent qu'il existe des solutions pour que tous les coefficients de cA soient identiquement nuls. Et ces solutions sont portées par les équations (B.21) et (B.22).

Dans la liste (B.21), il y a 357 équations. On ne montre que 3 polynômes car les autres polynômes sont trop longs et seulement une partie des polynômes peuvent être factorisés. Néanmoins, il est difficile de résumer sans l'aide numérique. Par exemple, pour que l'équation $(L1 c1 s4 + c4 s1 L4) s2^2 L3^2 (-c3 L2 s2 - c2 s3 L3 + L4 c1 s4 + L1 c4 s1) = 0$ soit respectée, il faut au moins que l'un des deux facteurs vaut zéro (ou les deux). C'est avec une configuration numérique d'un Bennett mobile que l'on sait que $(-c3 L2 s2 - c2 s3 L3 + L4 c1 s4 + L1 c4 s1) = 0$ mais $(L1 c1 s4 + c4 s1 L4) \neq 0$.



Application de l'algorithme de conditions de mobilité avec la méthode Jin et Yang

Dans la liste (B.22), il y a 4 équations. La 3^{ème} équation est une relation de compatibilité publiée par Beggs [Beg66]. Jusqu'à maintenant, nous obtenons 361 équations de mobilité pour la famille du mécanisme du Bennett :

$$L1^2 \sin^2(\alpha_2) - L2^2 \sin^2(\alpha_1) = 0 \quad (\text{B.23})$$

$$\cos(a4) \cos(a1) - \cos(a2) \cos(a3) = 0 \quad (\text{B.24})$$

$$\sin^2(a3) \sin^2(a2) - \sin^2(a4) \sin^2(a1) = 0 \quad (\text{B.25})$$

$$(L1 \cos(a1) \sin(a4) + \cos(a4) \sin(a1) L4) \sin(a2)^2 L3^2 (-c3 L2 \sin(a2) - c2 s3 L3 + L4 \cos(a1) \sin(a4) + L1 \cos(a4) \sin(a1)) = 0 \quad (\text{B.26})$$

⋮

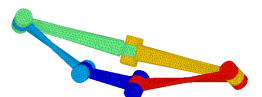
A l'exception de l'équation (B.23), les restes ne sont pas reconnues par les équations de Beggs. Mais les équations (B.24) et (B.25) donnent la symétrie des angles d'usinage.

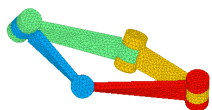
$$\begin{cases} a1 = a3 \\ a2 = a4 \end{cases} \quad (\text{B.27}) \qquad \begin{cases} a1 = -a3 \\ a2 = -a4 \end{cases} \quad (\text{B.28})$$

$$\begin{cases} a1 = -a3 \\ a2 = a4 \end{cases} \quad (\text{B.29}) \qquad \begin{cases} a1 = a3 \\ a2 = -a4 \end{cases} \quad (\text{B.30})$$

Ces 4 solutions ne sont pas toutes bonnes pour garantir la mobilité du mécanisme de Bennett. Juste avec les solutions (B.23) et (B.29), ou (B.23) et (B.30), les barres ne s'assemblent pas car il manque les relations en L_i . Dans cet exemple, les 361 équations sont des conditions nécessaires pour qu'un mécanisme de Bennet soit mobile. Mais avec la mise en équation de Jin et Yang, nous n'avons pas pu trouver toutes les expressions les plus simples pour un mécanisme de Bennett à cause de la puissance de l'ordinateur.

Malgré les 361 équations nécessaires trouvées, nous obtenons beaucoup trop d'équations par rapport aux équations de Beggs. La seule relation de mobilité que nous connaissons dans les listes (B.21) et (B.22) est l'équation (B.23). L'expression simple de la symétrie des longueurs des barres comme $L3 = L1$ ne saute pas aux yeux quand on regarde dans le détail les 361 équations. Ceci peut être lié au fait que le système d'équations soit séparé en deux groupes.





Annexe C

Annexes du codes Maple : conditions d'assemblage et de mobilité

Application des Bases de Gröbner sur la Variante de la méthode de Jin et Yang - Coupure nBnD

Les conditions d'assemblage du 4R

--- Ruixian RENAUD, 2013

```
> restart : Digits := 20 : with(FGb) :
Path set to C:\Program Files (x86)\Maple 13\bin.win/libfgbuni.
so
FGb/Maple interface package Version 1.34
JC Faugere (jcf@calfor.lip6.fr)
Type ?FGb for documentation
```

▼ Cas de Test pour le Bennett

▼ Cas mobile

```
> CasMobile := 'M02';
if CasMobile = 'M01' then
  TestMobile := [a1 =  $\frac{\text{Pi}}{6}$ , a2 = - $\frac{\text{Pi}}{3}$ , a3 =  $\frac{\text{Pi}}{6}$ , a4 = - $\frac{\text{Pi}}{3}$ , L1 = 1, L2 = sqrt(3), L3
    = 1, L4 = sqrt(3), tA = tA, tC = tA, tB = tB, tD = tB]
elif CasMobile = 'M02' then
  TestMobile := [a1 = 0.523598775598119, a2 = -1.047197551193130, a3
    = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
    = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
    = 1.732050807568877]
fi
CasMobile := M02
TestMobile := [a1 = 0.523598775598119, a2 = -1.047197551193130, a3
  = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
  = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
  = 1.732050807568877] (1.1.1)
```

▼ Cas assemblé

```
> CasAssemble := "A01";
if CasAssemble = "A01" then
```

```

TestAssemble := [a1 =
-0.29145679447786709199560462143289119350316759901206541927220608\
30872990149105089980715136487087625240, a2
= 0.4532304095150565547563982819282587930962961566922095547515565\
732596602087045671757713656471993516620, a3 =
-0.33544273508971189050892307516029058826751697067869036278142268\
76168793297678566597158301428260336517, a4
= 0.2590958683742541729479897614925188676084960577788412458828359\
042736751821491806922090312170285686631, L1 = 1., L2
= 0.8104936767180851979185298717434445363604235377236148429308060\
054859576461637322193712084302853844524, L3
= 0.9669539802906858090162169664147757243814887591700661219785286\
325045595333890430987488315400166010690, L4
= 1.4149558296992878215532608483382186245390610309097501920888844\
27248366634018641867738328050447509528, tA
= 1.8573981902894337555851156945750084595454432632809752560892493\
00386917873900707485566229692468595627, tB
= 1.4097044222573161198546043715930495697108993447621182114967030\
69673805266101878327714905667634996138, tC
= 1.1709225040373397738924075343350458504176651525135762280840532\
31161074767785344311068817246863908114, tD
= 1.9528494843106369422422510263265186150934620521543677204766656\
35486077710669202454992473679921614923 ]
elif CasAssemble = "A02" then
TestAssemble := [a1 = 0.927295217929077, a2 = -0.519146114299891, a3
= 0.625428863600991, a4 = -0.524290345559241, L1 = 250, L2 = 200, L3
= 406.20192023, L4 = 269.25824036, tA = 1.190289949618859 , tC
= 1.694200793582639 ]
elif CasAssemble = "A03" then
TestAssemble := [L1 = 1, L2 = 0.218668460463052, L3 = 0.129411879820720, L4
= 1.085378077837818, a1 = 0.69813170079773183077, a2 =
-1.2217304763960307039, a3 = 0.87266462599716478848, a4 =
-1.2906382533029200909 ]
elif CasAssemble = "A04" then
TestAssemble := [L1 = 1, L2 = 0.8104936767, L3 = 0.9669539802, L4
= 1.414955829, a1 = -0.2914567944, a2 = 0.4532304095 , a3 =
-0.3354427350 , a4 = 0.2590958683 ]
fi

```

CasAssemble := "A01"

```

TestAssemble := [a1 = (1.2.1)
-0.291456794477867091995604621432891193503167599012065419272206083\
0872990149105089980715136487087625240, a2
= 0.45323040951505655475639828192825879309629615669220955475155657\
32596602087045671757713656471993516620, a3 =
-0.335442735089711890508923075160290588267516970678690362781422687\
6168793297678566597158301428260336517, a4
= 0.25909586837425417294798976149251886760849605777884124588283590\
42736751821491806922090312170285686631, L1 = 1., L2
= 0.81049367671808519791852987174344453636042353772361484293080600\
54859576461637322193712084302853844524, L3
= 0.96695398029068580901621696641477572438148875917006612197852863\
25045595333890430987488315400166010690, L4
= 1.41495582969928782155326084833821862453906103090975019208888442\

```



```

7248366634018641867738328050447509528, tA
= 1.85739819028943375558511569457500845954544326328097525608924930\
0386917873900707485566229692468595627, tB
= 1.40970442225731611985460437159304956971089934476211821149670306\
9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\
5486077710669202454992473679921614923 ]

```

▼ Cas non-assemblé

```

> CasNA := "NA01";
if CasNA = "NA01" then
  TestNA := [a1 =
    -1.29145679447786709199560462143289119350316759901206541927220608\
    30872990149105089980715136487087625240, a2
    = 0.4532304095150565547563982819282587930962961566922095547515565\
    732596602087045671757713656471993516620, a3 =
    -0.33544273508971189050892307516029058826751697067869036278142268\
    76168793297678566597158301428260336517, a4
    = 0.2590958683742541729479897614925188676084960577788412458828359\
    042736751821491806922090312170285686631, L1 = 1.5, L2
    = 0.8104936767180851979185298717434445363604235377236148429308060\
    054859576461637322193712084302853844524, L3
    = 0.9669539802906858090162169664147757243814887591700661219785286\
    325045595333890430987488315400166010690, L4
    = 1.4149558296992878215532608483382186245390610309097501920888844\
    27248366634018641867738328050447509528, tA
    = 1.8573981902894337555851156945750084595454432632809752560892493\
    00386917873900707485566229692468595627, tB
    = 1.4097044222573161198546043715930495697108993447621182114967030\
    69673805266101878327714905667634996138, tC
    = 1.1709225040373397738924075343350458504176651525135762280840532\
    31161074767785344311068817246863908114, tD
    = 1.9528494843106369422422510263265186150934620521543677204766656\
    35486077710669202454992473679921614923 ]
fi

```

CasNA := "NA01"

```

TestNA := [a1 = (1.3.1)
-1.291456794477867091995604621432891193503167599012065419272206083\
0872990149105089980715136487087625240, a2
= 0.45323040951505655475639828192825879309629615669220955475155657\
32596602087045671757713656471993516620, a3 =
-0.335442735089711890508923075160290588267516970678690362781422687\
6168793297678566597158301428260336517, a4
= 0.25909586837425417294798976149251886760849605777884124588283590\
42736751821491806922090312170285686631, L1 = 1.5, L2
= 0.81049367671808519791852987174344453636042353772361484293080600\

```

```

54859576461637322193712084302853844524, L3
= 0.96695398029068580901621696641477572438148875917006612197852863\
25045595333890430987488315400166010690, L4
= 1.41495582969928782155326084833821862453906103090975019208888442\
7248366634018641867738328050447509528, tA
= 1.85739819028943375558511569457500845954544326328097525608924930\
0386917873900707485566229692468595627, tB
= 1.40970442225731611985460437159304956971089934476211821149670306\
9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\
5486077710669202454992473679921614923 ]

```

Procédure de Simplification: trouver le plus petit ensemble de polynômes générateurs d'un Idéal

```

> RRSimplification:=proc(polys,n,specs,inco,param)
#application pour trouver le plus petit eselbke de
polynômes générateurs
local i, supp_temp, pile, taille_supp_temp, taille_Gbtemp,
Gb_temp, jeton, taille_pile;
supp_temp:=[]:
pile:=[]:
Gb_temp:=[]:
for i from 1 to n do
supp_temp:= [op(polys[i+1..n])]:
taille_supp_temp:=nops(supp_temp):
Gb_temp:=fgb_gbasis_elim([op(pile),op(supp_temp),op(specs)
],0,[op(inco)], [op(param)]):
taille_Gbtemp:=nops(Gb_temp):
jeton:=evalb(polys=Gb_temp):
if jeton<>true then
pile:= [op(pile),polys[i]]:
taille_pile:=nops(pile):
end if:
end do:
print
(`#####`):
print("##### Fin de
la procédure
#####"):

```

```

print
(`#####`):
print(n-taille_pile, `polynômes sont supprimés.`):
print
(`#####`):
print(`Les`,taille_pile, `polynômes restants sont:`):
map(print, pile):
#return(pile):
end proc:

```

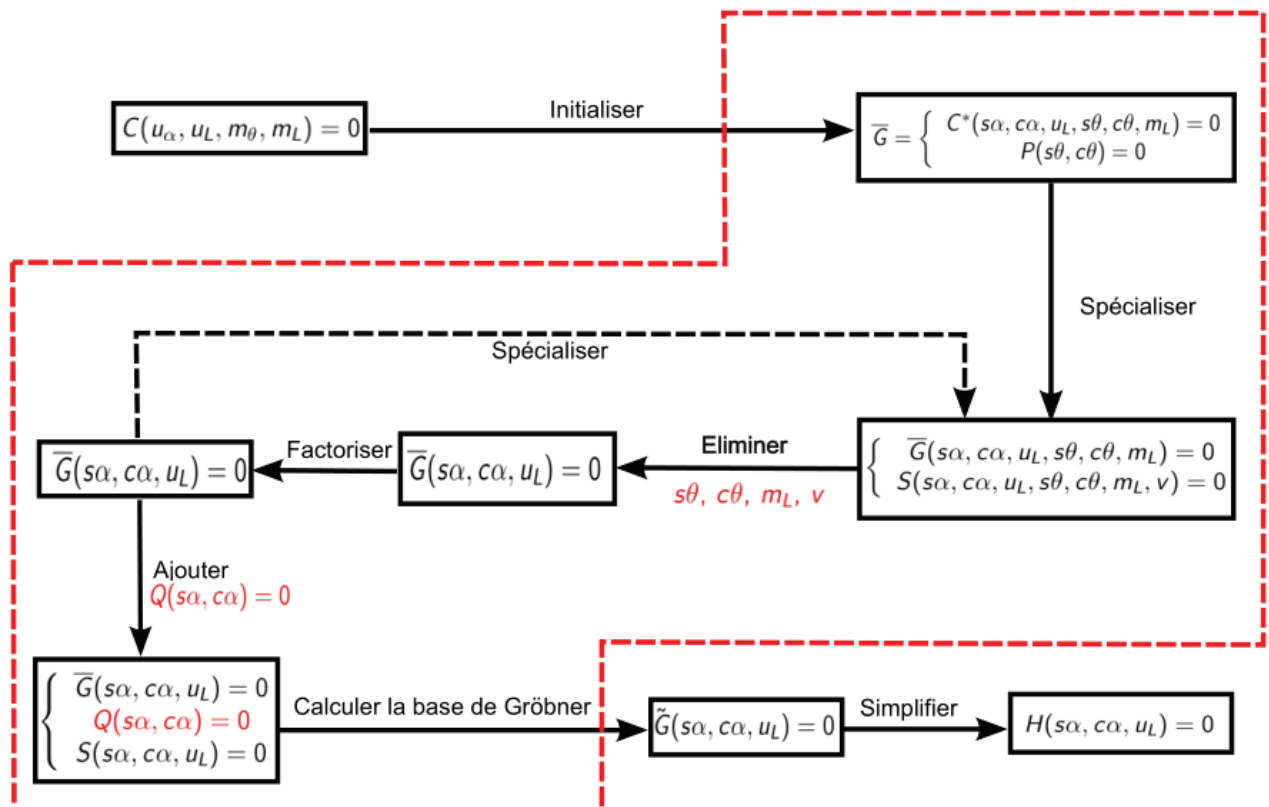


FIGURE 4.7 – L'obtention des conditions de mobilité dans un cas relativement complexe
La génération des conditions d'assemblage pour le 4R a suivi cet algorithme

Mise en équation

Système d'équations: avec la Variante de la méthode de Jin et Yang

$$\begin{aligned}
 > f11 := \cos(a2) \cdot \cos(a3) - \sin(a3) \cdot \sin(a2) \cdot \cos(tC) - \cos(a4) \cdot \cos(a1) + \sin(a1) \\
 &\quad \cdot \sin(a4) \cdot \cos(tA)
 \end{aligned}$$

(3.1.1)

$$f11 := \cos(a2) \cos(a3) - \sin(a3) \sin(a2) \cos(tC) - \cos(a4) \cos(a1) + \sin(a1) \sin(a4) \cos(tA) \quad (3.1.1)$$

$$\begin{aligned} > f12 := L4^2 + 2 \cdot L4 \cdot L1 \cdot \cos(tA) + L1^2 - L2^2 - L3^2 - 2 \cdot L2 \cdot L3 \cdot \cos(tC) \\ & \quad f12 := L4^2 + 2 L4 L1 \cos(tA) + L1^2 - L2^2 - L3^2 - 2 L2 L3 \cos(tC) \end{aligned} \quad (3.1.2)$$

$$\begin{aligned} > f13 := L3 \cdot \sin(a2) \cdot \sin(tC) + L4 \cdot \sin(tA) \cdot \sin(a1) \\ & \quad f13 := L3 \sin(a2) \sin(tC) + L4 \sin(tA) \sin(a1) \end{aligned} \quad (3.1.3)$$

$$\begin{aligned} > f14 := L2 \cdot \sin(tC) \cdot \sin(a3) + L1 \cdot \sin(a4) \cdot \sin(tA) \\ & \quad f14 := L2 \sin(tC) \sin(a3) + L1 \sin(a4) \sin(tA) \end{aligned} \quad (3.1.4)$$

$$> syst := [f11, f12, f13, f14, \cos(tC)^2 + \sin(tC)^2 - 1, \cos(tA)^2 + \sin(tA)^2 - 1]:$$

▼ **Vérification du système d'équations: avec un cas assemblé, le système d'équations est respecté**

$$\begin{aligned} > Verif := subs(TestAssemble, syst) : \\ > evalf(Verif) \\ & \quad [-1. 10^{-21}, 2. 10^{-20}, -2. 10^{-20}, -1. 10^{-20}, 0., -1. 10^{-20}] \end{aligned} \quad (3.2.1)$$

▼ **Initialisation : substitution des sinus et cosinus par les symboles**

$$\begin{aligned} > subs_syst := subs(\sin(a1) = s1, \cos(a1) = c1, \sin(a2) = s2, \cos(a2) = c2, \sin(a3) = s3, \cos(a3) = c3, \sin(a4) = s4, \cos(a4) = c4, \sin(tA) = sA, \cos(tA) = cA, \sin(tC) = sC, \cos(tC) = cC, syst) \\ subs_syst := [c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA, L4^2 + 2 L4 L1 cA + L1^2 - L2^2 - L3^2 - 2 L2 L3 cC, L3 s2 sC + L4 sA s1, L2 sC s3 + L1 s4 sA, cC^2 + sC^2 - 1, cA^2 + sA^2 - 1] \end{aligned} \quad (3.3.1)$$

▼ **Calcul de la condition d'assemblage**

▼ **Premier élimination: le système d'équations est totalement pris pour le calcul, sans être coupé en deux groupes. Les paramètres de mobilité (sC, cC, sA, cA) sont éliminés. Dans le résultat, il y a 80 polynômes.**

Ces 80 polynômes sont la base de Gröbner de la sous-variété de l'idéal.

```

> with(FGb) :
> syst := [op(subs_syst)] :
> gb_subs_syst := factor(fgb_gbasis_elim(subs_syst, 0, [sC, cC, sA, cA], [s4, c4, s3,
    c3, s2, c2, s1, c1, L4, L3, L2, L1])) :
> taille := nops(gb_subs_syst)

```

taille := 80 (4.1.1)

▼ Vérification du premier calcul: Ces 80 équations sont vérifiées pour un cas assemblé. La précision est 10^{*-20}

```

> Injection1 := subs((c1 = cos(a1), s1 = sin(a1), c2 = cos(a2), s2 = sin(a2), c3
    = cos(a3), s3 = sin(a3), c4 = cos(a4), s4 = sin(a4), sC = sin(tC), cC
    = cos(tC), sA = sin(tA), cA = cos(tA)), gb_subs_syst) :
> Verif1 := subs(TestAssemble, Injection1) :
> evalf(Verif1)

```

(4.2.1)

```

[-7.00001 10-21, -2.1558768001555445010 10-24, -8.7802982406334903313 10-24,
-4.7580 10-19, -1.5330 10-20, -4.485 10-20, -2.16 10-19, 2.07 10-20, -1.154 10-19,
2.765 10-19, -7.38 10-20, 1.576 10-19, 2. 10-19, -3.16 10-19, 0., 0., -3.3 10-19,
-1.64834 10-19, 9.9 10-20, 1.10 10-19, -5.6 10-19, -1.0 10-18, -5. 10-19,
9.207 10-20, 2. 10-19, -1.14 10-19, 2. 10-20, 1.19 10-19, 7.3 10-19, -4.2 10-21,
1.20 10-20, 1. 10-19, 1.25 10-19, -3.75 10-19, 1. 10-19, 1. 10-19, -2.1 10-20, 4.4 10-20,
3.669 10-19, 1.82 10-20, -1.312 10-19, 7. 10-20, -3.09 10-19, -4.803 10-19,
1.4239 10-18, 1.50 10-19, -3.6 10-19, -7.4 10-19, 6. 10-20, 5. 10-20, 2. 10-19, 0., 0.,
0., 0., 0., -3. 10-20, 1. 10-20, 0., 0., 0., -0., 0., -1. 10-19, -6.0 10-20, -0., -0., 0.,
1. 10-20, 2.661 10-19, -5.64 10-20, 1.74 10-19, 1.483 10-19, 2.5 10-20, -7.6 10-21,
-2.4 10-19, 7. 10-20, -3.4 10-20, -0., -1. 10-19]

```

▼ Etude du premier calcul: les 80 polynômes sont factorisés et très longs.

```

> map(print, gb_subs_syst[79..80]) :

```

$$\begin{aligned}
& -(s4 s2 L3 L1 - s3 s1 L4 L2) (-2 s4 s1 L3^3 L2 L1^2 - 2 s3 s2 L4 L3^2 L1^3 \\
& - 2 L2^3 s4 L3 s1 L4^2 + L2 s4 L3 s1 L4^4 - 2 L2 L4^2 s1 s4 L3^3 + L1 s3 s2 L4^5 \\
& - 2 L1^3 s2 s3 L4^3 + L4 s2 s3 L1^5 - 2 L1^3 s2 L4 L2^2 s3 - 2 L1^2 L3 s4 s1 L2^3 \\
& + L1 L4 s3 s2 L3^4 - 2 L1 L4^3 s3 s2 L3^2 - 2 L1 L2^2 s2 s3 L4^3 + L1 L4 s2 s3 L2^4 \\
& + L1^4 s4 s1 L3 L2 - 4 L1^3 L2 L3 c3 c2 L4 + 4 L1^3 c4 c1 L4 L2 L3 \\
& + 6 L1^2 L3 s4 s1 L4^2 L2 - 4 L1 L3^3 c4 c1 L4 L2 + 4 L1 L3^3 c3 c2 L4 L2 \\
& + 6 L1 L2^2 s3 s2 L4 L3^2 + 4 L1 L3 c3 c2 L4 L2^3 - 4 L1 L3 c4 c1 L4 L2^3
\end{aligned}$$

$$\begin{aligned}
& + 4 L1 L3 c4 c1 L4^3 L2 - 4 L1 L3 c3 c2 L4^3 L2 - 2 L2^3 s1 s4 L3^3 + L2 s1 s4 L3^5 \\
& + s4 L3 s1 L2^5) \\
s4 s3 s2 s1 L4^4 - 2 s4 s3 s2 s1 L4^2 L3^2 + s4 s3 s2 s1 L3^4 - 2 s4 s3 s2 s1 L4^2 L2^2 & \quad (4.3.1) \\
- 2 s4 s3 s2 s1 L3^2 L2^2 + s4 s3 s2 s1 L2^4 + 4 s3^2 s2^2 L4 L3 L2 L1 \\
- 4 c3^2 c2^2 L4 L3 L2 L1 + 4 s4^2 s1^2 L4 L3 L2 L1 + 8 c4 c3 c2 c1 L4 L3 L2 L1 \\
- 4 c4^2 c1^2 L4 L3 L2 L1 - 2 s4 s3 s2 s1 L4^2 L1^2 - 2 s4 s3 s2 s1 L3^2 L1^2 \\
- 2 s4 s3 s2 s1 L2^2 L1^2 + s4 s3 s2 s1 L1^4
\end{aligned}$$

▼ **Spécialisations: le facteur le plus répété est $s4*s2*L3*L1-s3*s1*L4*L2=0$, 16 fois. Ce facteur est donc une contrainte (2ème astuce)**

.

Saturation:

1. Longueur non nulle.

2. Le système n'est pas plan: c'est-à-dire que $s1, s2, s3$ et $s4$ n'égalent pas à 0 en même temps.

Enfin, l'algorithme trouve 6 polynômes. Ces 6 polynômes sont considérés comme les conditions "brutes" d'assemblage présentés dans Chapitre 4

$$\begin{aligned}
> S1 := [s4 s2 L1 L3 - L4 L2 s1 s3, L4 \cdot L3 \cdot L2 \cdot L1 \cdot uu - 1, s1 \cdot vv - 1, s2 \cdot ww - 1, s3 \cdot xx \\
- 1, s4 \cdot yy - 1] \\
S1 := [s4 s2 L3 L1 - s3 s1 L4 L2, L4 L3 L2 L1 uu - 1, s1 vv - 1, s2 ww - 1, s3 xx \\
- 1, s4 yy - 1] \quad (4.4.1)
\end{aligned}$$

▼ **2ème élimination**

$$\begin{aligned}
> subs_syst_total := [op(gb_subs_syst), op(S1)] : \\
> gb_subs_syst_total := fgb_gbasis_elim(subs_syst_total, 0, [sC, cC, sA, cA, uu, vv, \\
ww, xx, yy], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1]) : \\
> nops(gb_subs_syst_total) \\
6 \quad (4.5.1) \\
> map(print, factor(gb_subs_syst_total)) : \\
-4 L1 L2^2 L3 L4 c4^2 c1^2 + 8 c3 c2 L4 L3 L2^2 L1 c4 c1 - 2 s4 s3 s2 s1 L3^2 L2^3 \\
- 2 s4^2 s2^2 L4 L3 L2^2 L1 + s4^2 s2^2 L4^3 L3 L1 - 2 s4 s3 s2 s1 L3^2 L2 L1^2 \\
- 2 s4 s3 s2 s1 L2^3 L1^2 + s4 s3 s2 s1 L2 L1^4 - 2 s4^2 s2^2 L4 L3 L1^3
\end{aligned}$$

$$\begin{aligned}
& + s4 s3 s2 s1 L3^4 L2 - 2 s4^2 s2^2 L4 L3^3 L1 + s4 s3 s2 s1 L2^5 \\
& + 4 s3^2 s2^2 L4 L3 L2^2 L1 - 4 c3^2 c2^2 L4 L3 L2^2 L1 + 4 s4^2 s1^2 L4 L3 L2^2 L1 \\
& - 4 L1^2 L2 L3^2 c4^2 c1^2 + 8 c3 c2 L3^2 L2 L1^2 c4 c1 - 2 s3^2 s1^2 L3^2 L2^3 \\
& - 4 c3^2 c2^2 L3^2 L2 L1^2 - 2 s4 s3 s2 s1 L4 L3^3 L1 - 2 s4 s3 s2 s1 L4 L3 L1^3 \\
& + s3^2 s1^2 L2^5 + s3^2 s1^2 L2 L1^4 + s3^2 s1^2 L3^4 L2 + 4 s3^2 s2^2 L3^2 L2 L1^2 \\
& - 2 s4^2 s2^2 L3^2 L2 L1^2 - 2 s3^2 s1^2 L2^3 L1^2 + s4 s3 s2 s1 L4^3 L3 L1 \\
& + 4 s4^2 s1^2 L3^2 L2 L1^2 - 2 s3^2 s1^2 L3^2 L2 L1^2 \\
& s4^2 s2^2 L4^4 - 2 s4^2 s2^2 L4^2 L3^2 + s4^2 s2^2 L3^4 - 2 s4^2 s2^2 L4^2 L2^2 + 4 s3^2 s2^2 L4^2 L2^2 \\
& - 4 c3^2 c2^2 L4^2 L2^2 + 4 s4^2 s1^2 L4^2 L2^2 + 8 c4 c3 c2 c1 L4^2 L2^2 \\
& - 4 c4^2 c1^2 L4^2 L2^2 - 2 s4^2 s2^2 L3^2 L2^2 + s4^2 s2^2 L2^4 - 2 s4^2 s2^2 L4^2 L1^2 \\
& - 2 s4^2 s2^2 L3^2 L1^2 - 2 s4^2 s2^2 L2^2 L1^2 + s4^2 s2^2 L1^4 \\
& s4 s3 s2 s1 L4^4 - 2 s4 s3 s2 s1 L4^2 L3^2 + s4 s3 s2 s1 L3^4 - 2 s4 s3 s2 s1 L3^2 L2^2 \\
& + s4 s3 s2 s1 L2^4 - 2 s4^2 s2^2 L4 L3 L2 L1 + 4 s3^2 s2^2 L4 L3 L2 L1 \\
& - 4 c3^2 c2^2 L4 L3 L2 L1 + 4 s4^2 s1^2 L4 L3 L2 L1 + 8 c4 c3 c2 c1 L4 L3 L2 L1 \\
& - 4 c4^2 c1^2 L4 L3 L2 L1 - 2 s4 s3 s2 s1 L4^2 L1^2 - 2 s4 s3 s2 s1 L3^2 L1^2 \\
& - 2 s4 s3 s2 s1 L2^2 L1^2 + s4 s3 s2 s1 L1^4 \\
& s3^2 s1^2 L4^4 - 2 s3^2 s1^2 L4^2 L3^2 + s3^2 s1^2 L3^4 - 2 s3^2 s1^2 L3^2 L2^2 + s3^2 s1^2 L2^4 \\
& - 2 s3^2 s1^2 L4^2 L1^2 - 2 s4^2 s2^2 L3^2 L1^2 + 4 s3^2 s2^2 L3^2 L1^2 - 4 c3^2 c2^2 L3^2 L1^2 \\
& + 4 s4^2 s1^2 L3^2 L1^2 - 2 s3^2 s1^2 L3^2 L1^2 + 8 c4 c3 c2 c1 L3^2 L1^2 - 4 c4^2 c1^2 L3^2 L1^2 \\
& - 2 s3^2 s1^2 L2^2 L1^2 + s3^2 s1^2 L1^4 \\
& \qquad \qquad \qquad s3 s1 L4 L2 - s4 s2 L3 L1
\end{aligned} \tag{4.5.2}$$

$$\begin{aligned}
& \text{with(Groebner) : HilbertDimension}(gb_subs_syst_total) \\
& \qquad \qquad \qquad 10
\end{aligned} \tag{4.5.3}$$

▼ Ajouter Q=(sa,ca)

$$\begin{aligned}
& \text{[> } Q := [s1^2 + c1^2 - 1, s2^2 + c2^2 - 1, s3^2 + c3^2 - 1, s4^2 + c4^2 - 1] :
\end{aligned}$$

▼ Calcul de la base de Gröbner

$$\begin{aligned}
& \text{[> } CS := fgb_gbasis([op}(gb_subs_syst_total), op}(Q)], 0, [], [s4, c4, s3, c3, s2, c2, \\
& \qquad \qquad \qquad s1, c1, L4, L3, L2, L1]) : nops}(CS) \\
& \qquad \qquad \qquad 32
\end{aligned} \tag{4.7.1}$$

$$\begin{aligned}
& \text{[> } with(Groebner) : HilbertDimension}(CS); \\
& \qquad \qquad \qquad 6
\end{aligned} \tag{4.7.2}$$

▼ Vérification des équations d'assemblage: ces 6 équations d'assemblage sont vérifiées par un cas assemblé

$$\begin{aligned}
& \text{[> } Injection3 := subs((c1 = cos(a1), s1 = sin(a1), c2 = cos(a2), s2 = sin(a2), c3
\end{aligned}$$

```

= cos(a3), s3 = sin(a3), c4 = cos(a4), s4 = sin(a4), sB = sin(tB), cB = cos(tB),
sD = sin(tD), cD = cos(tD), CS) :
> Verif3 := simplify(Injection3) : nops(Verif3)
                                     32
                                     (4.8.1)
> degree(Verif3[1])
                                     12
                                     (4.8.2)
> Verif4 := subs(TestAssemble, Injection3) :
> evalf(Verif4)
[1. 10-20, -1. 10-20, -1. 10-20, 0., -1. 10-20, -2. 10-20, 0., -3. 10-20, -4. 10-20, 1. 10-20,
5. 10-20, -5. 10-21, -1.7 10-19, -1.13 10-19, -1. 10-19, -2.2 10-19, 6. 10-20,
-6. 10-20, 4. 10-20, -1. 10-19, 1. 10-19, 0., 1. 10-19, 0., 0., 1. 10-20, -2. 10-20, 0., 0.,
-1. 10-20, -1. 10-20, 0.]
(4.8.3)
> noyau := fgb_gbasis_elim([op(gb_subs_syst_total[5..6]), op(CS[29..32]), L4·L3
·L2·L1·uu - 1], 0, [uu], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1]) :
taille_noyau := nops(noyau)
taille_noyau := 32
(4.8.4)
> evalb(CS = noyau)
true
(4.8.5)

```

▼ **Simplification automatique : trouver le plus petit ensemble de polynômes générateurs qui représente le même idéal que le système de départ.**

```

> RRSimplification(noyau, taille_noyau, [L4·L3·L2·L1·uu - 1], [uu], [s4, c4, s3, c3,
s2, c2, s1, c1, L4, L3, L2, L1]) :
#####\
#####
"##### Fin de la procédure
#####"
#####\
#####
26, polynômes sont supprimés.
#####\
#####
Les, 6, polynômes restants sont:
-c12 L34 - c12 L24 - c12 L14 - 2 L42 L32 + L34 - 2 c42 L32 L12 - 2 c12 L32 L12 + L24
+ 8 c4 c3 c2 c1 L32 L12 - 2 L32 L22 + 4 L32 L12 - 2 L42 L12 - 2 L22 L12
+ c32 c12 L44 - 2 c32 c12 L42 L32 + c32 c12 L34 - 2 c32 c12 L32 L22 + c32 c12 L24
- 2 c32 c12 L42 L12 - 2 c32 c12 L32 L12 - 2 c32 c12 L22 L12 + c32 c12 L14
- c12 L44 - c32 L44 + 2 c32 L42 L32 + 2 c12 L42 L32 - c32 L34 + 2 c32 L32 L22
+ 2 c12 L32 L22 - c32 L24 + 2 c32 L42 L12 + 2 c12 L42 L12 + 2 c32 L22 L12
+ 2 c12 L22 L12 - c32 L14 - 2 c42 c22 L32 L12 + L44 + L14 - 2 c22 L32 L12
- 2 c32 L32 L12

```




$$s_3 s_1 L_4 L_2 - s_4 s_2 L_3 L_1$$

$$s_4^2 + c_4^2 - 1$$

$$s_3^2 + c_3^2 - 1$$

$$s_2^2 + c_2^2 - 1$$

$$s_1^2 + c_1^2 - 1$$

(4.9.1)

Application des Bases de Gröbner sur la Variante de la méthode de Jin et Yang - Coupure nBnD

Les conditions d'assemblage du 4R

--- Ruixian RENAUD, 2014, 20 février

L'objectif du code est de parcourir la branch où

$$s4 \cdot s2 \cdot L3 \cdot L1 - s3 \cdot s1 \cdot L4 \cdot L2 \neq 0 :$$

```
> restart: Digits := 20: with(FGb):
Path set to C:\Program Files (x86)\Maple 13\bin.win/libfgbuni.
so
FGb/Maple interface package Version 1.34
JC Faugere (jcf@calfor.lip6.fr)
Type ?FGb for documentation
```

▼ Cas de Test pour le Bennett

▼ Cas mobile

```
> CasMobile := 'M02';
if CasMobile = 'M01' then
  TestMobile := [a1 =  $\frac{\text{Pi}}{6}$ , a2 = - $\frac{\text{Pi}}{3}$ , a3 =  $\frac{\text{Pi}}{6}$ , a4 = - $\frac{\text{Pi}}{3}$ , L1 = 1, L2 = sqrt(3), L3
    = 1, L4 = sqrt(3), tA = tA, tC = tA, tB = tB, tD = tB]
elif CasMobile = 'M02' then
  TestMobile := [a1 = 0.523598775598119, a2 = -1.047197551193130, a3
    = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
    = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
    = 1.732050807568877]
fi
                                CasMobile := M02
TestMobile := [a1 = 0.523598775598119, a2 = -1.047197551193130, a3
  = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
  = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
  = 1.732050807568877] (1.1.1)
```

Cas assemblé

```
> CasAssemble := "A01";
if CasAssemble = "A01" then
  TestAssemble := [a1 =
    -0.29145679447786709199560462143289119350316759901206541927220608\
    30872990149105089980715136487087625240, a2
    = 0.4532304095150565547563982819282587930962961566922095547515565\
    732596602087045671757713656471993516620, a3 =
    -0.33544273508971189050892307516029058826751697067869036278142268\
    76168793297678566597158301428260336517, a4
    = 0.2590958683742541729479897614925188676084960577788412458828359\
    042736751821491806922090312170285686631, L1 = 1., L2
    = 0.8104936767180851979185298717434445363604235377236148429308060\
    054859576461637322193712084302853844524, L3
    = 0.9669539802906858090162169664147757243814887591700661219785286\
    325045595333890430987488315400166010690, L4
    = 1.4149558296992878215532608483382186245390610309097501920888844\
    27248366634018641867738328050447509528, tA
    = 1.8573981902894337555851156945750084595454432632809752560892493\
    00386917873900707485566229692468595627, tB
    = 1.4097044222573161198546043715930495697108993447621182114967030\
    69673805266101878327714905667634996138, tC
    = 1.1709225040373397738924075343350458504176651525135762280840532\
    31161074767785344311068817246863908114, tD
    = 1.9528494843106369422422510263265186150934620521543677204766656\
    35486077710669202454992473679921614923 ]
  elif CasAssemble = "A02" then
    TestAssemble := [a1 = 0.927295217929077, a2 = -0.519146114299891, a3
      = 0.625428863600991, a4 = -0.524290345559241, L1 = 250, L2 = 200, L3
      = 406.20192023, L4 = 269.25824036, tA = 1.190289949618859, tC
      = 1.694200793582639 ]
  elif CasAssemble = "A03" then
    TestAssemble := [L1 = 1, L2 = 0.218668460463052, L3 = 0.129411879820720, L4
      = 1.085378077837818, a1 = 0.69813170079773183077, a2 =
      -1.2217304763960307039, a3 = 0.87266462599716478848, a4 =
      -1.2906382533029200909 ]
  elif CasAssemble = "A04" then
    TestAssemble := [L1 = 1, L2 = 0.8104936767, L3 = 0.9669539802, L4
      = 1.414955829, a1 = -0.2914567944, a2 = 0.4532304095, a3 =
      -0.3354427350, a4 = 0.2590958683 ]
fi
```

CasAssemble := "A01"

```
TestAssemble := [a1 = (1.2.1)
-0.291456794477867091995604621432891193503167599012065419272206083\
0872990149105089980715136487087625240, a2
= 0.45323040951505655475639828192825879309629615669220955475155657\
32596602087045671757713656471993516620, a3 =
-0.335442735089711890508923075160290588267516970678690362781422687\
6168793297678566597158301428260336517, a4
= 0.25909586837425417294798976149251886760849605777884124588283590\
42736751821491806922090312170285686631, L1 = 1., L2
```

```

= 0.81049367671808519791852987174344453636042353772361484293080600\
54859576461637322193712084302853844524, L3
= 0.96695398029068580901621696641477572438148875917006612197852863\
25045595333890430987488315400166010690, L4
= 1.41495582969928782155326084833821862453906103090975019208888442\
7248366634018641867738328050447509528, tA
= 1.85739819028943375558511569457500845954544326328097525608924930\
0386917873900707485566229692468595627, tB
= 1.40970442225731611985460437159304956971089934476211821149670306\
9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\
5486077710669202454992473679921614923 ]

```

▼ Cas non-assemblé

```

> CasNA := "NA01";
  if CasNA = "NA01" then
    TestNA := [a1 =
      -1.29145679447786709199560462143289119350316759901206541927220608\
      30872990149105089980715136487087625240, a2
      = 0.4532304095150565547563982819282587930962961566922095547515565\
      732596602087045671757713656471993516620, a3 =
      -0.33544273508971189050892307516029058826751697067869036278142268\
      76168793297678566597158301428260336517, a4
      = 0.2590958683742541729479897614925188676084960577788412458828359\
      042736751821491806922090312170285686631, L1 = 1.5, L2
      = 0.8104936767180851979185298717434445363604235377236148429308060\
      054859576461637322193712084302853844524, L3
      = 0.9669539802906858090162169664147757243814887591700661219785286\
      325045595333890430987488315400166010690, L4
      = 1.4149558296992878215532608483382186245390610309097501920888844\
      27248366634018641867738328050447509528, tA
      = 1.8573981902894337555851156945750084595454432632809752560892493\
      00386917873900707485566229692468595627, tB
      = 1.4097044222573161198546043715930495697108993447621182114967030\
      69673805266101878327714905667634996138, tC
      = 1.1709225040373397738924075343350458504176651525135762280840532\
      31161074767785344311068817246863908114, tD
      = 1.9528494843106369422422510263265186150934620521543677204766656\
      35486077710669202454992473679921614923 ]
    fi

```

CasNA := "NA01"

```

TestNA := [a1 =
  -1.291456794477867091995604621432891193503167599012065419272206083\
  0872990149105089980715136487087625240, a2
  = 0.4532304095150565547563982819282587930962961566922095547515565\
  732596602087045671757713656471993516620, a3 =

```

(1.3.1)

```

-0.335442735089711890508923075160290588267516970678690362781422687\
6168793297678566597158301428260336517, a4
= 0.25909586837425417294798976149251886760849605777884124588283590\
42736751821491806922090312170285686631, L1 = 1.5, L2
= 0.81049367671808519791852987174344453636042353772361484293080600\
54859576461637322193712084302853844524, L3
= 0.96695398029068580901621696641477572438148875917006612197852863\
25045595333890430987488315400166010690, L4
= 1.41495582969928782155326084833821862453906103090975019208888442\
7248366634018641867738328050447509528, tA
= 1.85739819028943375558511569457500845954544326328097525608924930\
0386917873900707485566229692468595627, tB
= 1.40970442225731611985460437159304956971089934476211821149670306\
9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\
5486077710669202454992473679921614923 ]

```

Mise en équation

Système d'équations: avec la Variante de la méthode de Jin et Yang

```

> f11 := cos(a2)*cos(a3)-sin(a3)*sin(a2)*cos(tC)-cos(a4)*
cos(a1)+sin(a1)*sin(a4)*cos(tA);
f11 := cos(a2) cos(a3) - sin(a3) sin(a2) cos(tC) - cos(a4) cos(a1)
+ sin(a1) sin(a4) cos(tA)

```

(2.1.1)

```

> f12 := L4^2+2*L4*L1*cos(tA)+L1^2-L2^2-L3^2-2*L2*L3*cos
(tC);
f12 := L4^2 + 2 L4 L1 cos(tA) + L1^2 - L2^2 - L3^2 - 2 L2 L3 cos(tC)

```

(2.1.2)

```

> f13 := L3*sin(a2)*sin(tC)+L4*sin(tA)*sin(a1);
f13 := L3 sin(a2) sin(tC) + L4 sin(tA) sin(a1)

```

(2.1.3)

```

> f14 := L2*sin(tC)*sin(a3)+L1*sin(a4)*sin(tA);
f14 := L2 sin(tC) sin(a3) + L1 sin(a4) sin(tA)

```

(2.1.4)

```

> syst := [f11, f12, f13, f14, cos(tC)^2+sin(tC)^2-1, cos
(tA)^2+sin(tA)^2-1]:

```

Vérification du système d'équations: avec un cas assemblé, le système d'équations est respecté

```

> Verif := subs(TestAssemble, syst):
> evalf(Verif);
      [-1. 10-21, 2. 10-20, -2. 10-20, -1. 10-20, 0., -1. 10-20]

```

(2.2.1)

Initialisation : substitution des sinus et cosinus par les symboles

```

> subs_syst := subs(sin(a1) = s1, cos(a1) = c1, sin(a2) =
s2, cos(a2) = c2, sin(a3) = s3, cos(a3) = c3, sin(a4) =
s4, cos(a4) = c4, sin(tA) = sA, cos(tA) = cA, sin(tC) =
sC, cos(tC) = cC, syst);
subs_syst := [c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA, L42 + 2 L4 L1 cA + L12 - L22
- L32 - 2 L2 L3 cC, L3 s2 sC + L4 sA s1, L2 sC s3 + L1 s4 sA, cC2 + sC2 - 1,
cA2 + sA2 - 1]

```

(2.3.1)

Calcul de la condition d'assemblage

Premier élimination: le système d'équations est totalement pris pour le calcul, sans être coupé en deux groupes. Les paramètres de mobilité (sC, cC, sA, cA) sont éliminés. Dans le résultat, il y a 80 polynômes.

Ces 80 polynômes sont la base de Gröbner de la sous-variété de l'idéal.

```

> with(FGb):
> gb_subsys := factor(fgb_gbasis_elim([op(subs_syst)],
0, [sC, cC, sA, cA], [s4, c4, s3, c3, s2, c2, s1, c1, L4,
L3, L2, L1], {'index' = 30000000})):
> taille := nops(gb_subsys);
      taille := 80

```

(3.1.1)

Vérification du premier calcul: Ces 80 équations sont vérifiées pour un cas assemblé. La précision est 10^{^(-20)}

```

> Injection1 := subs(c1 = cos(a1), s1 = sin(a1), c2 = cos
(a2), s2 = sin(a2), c3 = cos(a3), s3 = sin(a3), c4 = cos

```

```

(a4), s4 = sin(a4), sC = sin(tC), cC = cos(tC), sA = sin
(tA), cA = cos(tA), gb_subs_syst):
> Verif1 := subs(TestAssemble, Injection1):
> evalf(Verif1);
[-3.798 10-21, -2.1558768001555445010 10-24, -6.4284326404638054211 10-24,      (3.2.1)
-3.1121 10-19, -3.608 10-21, -4.3 10-20, -2.49 10-19, 4.4 10-20, -7.989 10-20,
7.034 10-19, 3.100 10-19, -1.485 10-19, 2.372 10-19, -1.8 10-19, 0., -1.452 10-20,
-3. 10-19, -3.2 10-20, 1.52 10-19, 4.8759 10-20, -5. 10-19, -6.8354 10-19,
-3.39 10-19, 1.29 10-19, -3.395 10-19, -2.6 10-20, 5.40 10-20, 1.25 10-19, 6. 10-19,
-1. 10-20, 2. 10-20, 1.79 10-19, 2.003 10-19, -4.095 10-19, -5.169 10-19, 4.614 10-19,
-1.2 10-20, 9.5 10-20, 5. 10-20, 2. 10-19, 1.0 10-19, 7.7 10-19, -2.76 10-19, -6. 10-19,
5.2 10-19, 1. 10-19, 8.5 10-20, -9.9 10-19, 3.56 10-20, -7.042 10-20, 1.21 10-19, 0., 0.,
0., 0., 0., -4. 10-20, 9.0 10-21, 0., 0., 0., -0., 0., -1.3 10-20, -4. 10-20, -0., -0., 0.,
1. 10-20, -2.3 10-19, -3. 10-21, 2. 10-19, 2.0148 10-19, 2.17 10-20, -1.7 10-20,
-2. 10-19, 0., 0., -0., -1. 10-19]

```

▼ **Etude du premier calcul: les 80 polynômes sont factorisés et très longs.**

```
[> #map(print, gb_subs_syst):
```

▼ **Spécialisations: nous avons déjà parcouru le facteur**
 $s3 \cdot s1 \cdot L2 \cdot L4 - s4 \cdot s2 \cdot L1 \cdot L3 = 0$: **Maintenant, nous allons**
parcourir $s3 \cdot s1 \cdot L2 \cdot L4 - s2 \cdot s4 \cdot L1 \cdot L3 \neq 0$:

Saturation:

1. Longueur non nulle.

2. $s3 \cdot s1 \cdot L2 \cdot L4 - s2 \cdot s4 \cdot L1 \cdot L3 \neq 0$

Nous obtenons cette fois-ci 17 polynômes. Certains entre eux peuvent être factorisés: le 16ème par exemple.

```
[> S1 := [(s4*s2*L3*L1-s3*s1*L4*L2)*zz-1, L4*L3*L2*L1*uu-1];
      S1 := [(s4 s2 L3 L1 - s3 s1 L4 L2) zz - 1, L4 L3 L2 L1 uu - 1]      (3.4.1)
[> subs_syst_total := [op(gb_subs_syst), op(S1)]:
```

```
> gb_subs_syst_total := fgb_gbasis_elim(subs_syst_total, 0,
[uu,zz], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2,
L1], {'index'=3000000000}):nops(gb_subs_syst_total);
17 (3.4.2)
```

```
> factor(gb_subs_syst_total[16]);
(L2 + L1 - L3 + L4) (L2 + L1 - L3 - L4) (-L2 + L1 + L3 - L4) (-L2 + L1 + L3
+ L4) (L2 + L1 + L3 + L4) (L2 + L1 + L3 - L4) (-L2 + L1 - L3 - L4) (-L2
+ L1 - L3 + L4) (3.4.3)
```

▼ **Utilise 2ème polynômes pour parcourir les solutions des 17 polynômes dans le cas $s3 \cdot s1 \cdot L2 \cdot L4 - s2 \cdot s4 \cdot L1 \cdot L3 \neq 0$:**

```
> S2 := [(L2+L1+L3+L4)*vv-1]:syst_s2 := [op
(gb_subs_syst_total), op(S2)]:
> gb1 := fgb_gbasis_elim(syst_s2, 0, [uu,vv,zz], [s4, c4, s3,
c3, s2, c2, s1, c1, L4, L3, L2, L1], {'index'=3000000000}
):nops(gb1);
18 (3.5.1)
```

```
> factor(gb1[18]);
(L2 + L1 - L3 + L4) (L2 + L1 - L3 - L4) (-L2 + L1 + L3 - L4) (-L2 + L1 + L3
+ L4) (-L2 + L1 - L3 - L4) (L2 + L1 + L3 - L4) (-L2 + L1 - L3 + L4) (3.5.2)
```

▼ **Spécialisation:**

$s1 \cdot s2 \cdot s3 \cdot s4 \cdot uu - 1, L1 \cdot L2 \cdot L3 \cdot L4 \cdot vv - 1$: avec chaque facteur de gb1[18], respectivement.

▼ *Spécialisation*

$S3 := L2 + L1 + L3 - L4$: $S3_bis := (L2 + L1 + L3 - L4) \cdot ww - 1$:

, respectivement appliquée sur gb1[18].

```
> S3 := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, L2+L1+L3-L4];
syst_s3 := [op(gb1), op(S3)]:
S3 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, L2 + L1 + L3 - L4] (3.6.1.1)
```



```
> CS3 := fgb_gbasis_elim(syst_s3, 0, [uu,vv,xx], [s4, c4,
s3, c3, s2, c2, s1, c1, L4, L3, L2, L1], {'index'=
30000000000}): nops(CS3);
3 (3.6.1.2)
```

```
> map(print, factor(CS3)) :
-(c4 c1 - c2 c3 + s3 s2 + s4 s1) (c4 c1 - c2 c3 - s3 s2 - s4 s1)
(c4 c1 - c2 c3 + s3 s2 + s4 s1) (L3 + L1) (L1 + L2)
L4 - L3 - L2 - L1 (3.6.1.3)
```

```
> S3_bis:=s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,(L2+L1+L3-L4)
*ww-1:syst_s3_bis:=[op(gb1),S3_bis]:
> CS3bis:=fgb_gbasis_elim(syst_s3_bis,0,[uu,vv,ww],[s4,
c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],{'index'=
30000000000}):nops(CS3bis);
17 (3.6.1.4)
```

```
> factor(CS3bis[17]);
-(-L2+L1-L3-L4)(L2+L1-L3+L4)(-L2+L1+L3-L4)(-L2
+L1-L3+L4)(L2+L1-L3-L4)(-L2+L1+L3+L4) (3.6.1.5)
```

```
>
```

▼ Spécialisation

$S4 := -L2 + L1 - L3 - L4$: $S4_bis := (-L2 + L1 - L3 - L4)$
 $\cdot ww - 1$:

respectivement appliquée sur CS3bis[17]

```
> S4:= [s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,-L2+L1-L3-L4];
syst_s4:=[op(CS3bis),op(S4)]:
S4 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, -L2 + L1 - L3 - L4] (3.6.2.1)
```

```
> CS4:=fgb_gbasis_elim(syst_s4, 0, [uu,vv], [s4, c4, s3,
c3, s2, c2, s1, c1, L4, L3, L2, L1],{'index'=
30000000000}): nops(CS4);
3 (3.6.2.2)
```

```
> map(print, factor(CS4)) :
-(c4 c1 - c2 c3 + s3 s2 + s4 s1) (c4 c1 - c2 c3 - s3 s2 - s4 s1)
(c4 c1 - c2 c3 + s3 s2 + s4 s1) (-L3 + L1) (L1 - L2)
L2 - L1 + L3 + L4 (3.6.2.3)
```

```
> S4_bis:= [s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,(-L2+L1-L3-
L4)*ww-1];syst_s4_bis:=[op(CS3bis),op(S4_bis)]:
S4_bis := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, (-L2 + L1 - L3 - L4) ww
- 1] (3.6.2.4)
```

```
> CS4bis:=fgb_gbasis_elim(syst_s4_bis, 0, [uu,vv,ww],
[s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],
{'index'=30000000000}): nops(CS4bis);
9 (3.6.2.5)
```

```
> factor(CS4bis[9]);
(L2 + L1 - L3 - L4) (-L2 + L1 + L3 + L4) (L2 + L1 - L3 + L4) (-L2 + L1
+ L3 - L4) (-L2 + L1 - L3 + L4) (3.6.2.6)
>
```

Spécialisation

$S5 := L2 + L1 - L3 + L4 : S5_bis := (L2 + L1 - L3 + L4) \cdot ww - 1 :$

,respectivement appliquée sur CS4bis[9]

```
> S5:=[s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,L2+L1-L3+L4];
syst_s5:=[op(CS4bis),op(S5)]:
S5 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, L2 + L1 - L3 + L4] (3.6.3.1)
```

```
> CS5:=fgb_gbasis_elim(syst_s5, 0, [uu,vv], [s4, c4, s3,
c3, s2, c2, s1, c1, L4, L3, L2, L1],{'index'=
3000000000}):nops(CS5);
2 (3.6.3.2)
```

```
> map(print,factor(CS5)):
-c4 c1 + c2 c3 + s3 s2 + s4 s1
L2 + L1 - L3 + L4 (3.6.3.3)
```

```
> S5_bis:=[s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,(L2+L1-L3+
L4)*ww-1];syst_s5_bis:=[op(CS4bis),op(S5_bis)]:
S5_bis := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, (L2 + L1 - L3 + L4) ww - 1] (3.6.3.4)
```

```
> CS5bis:=fgb_gbasis_elim(syst_s5_bis, 0, [uu,vv,ww],
[s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],
{'index'=3000000000}):nops(CS5bis);
8 (3.6.3.5)
```

```
> factor(CS5bis[8]);
(-L2 + L1 - L3 + L4) (L2 + L1 - L3 - L4) (-L2 + L1 + L3 - L4) (-L2
+ L1 + L3 + L4) (3.6.3.6)
```

Spécialisation

$S6 := -L2 + L4 - L3 + L1 : S6_bis := (-L2 + L4 - L3 + L1) \cdot ww - 1 :$

,respectivement appliquée sur CS5bis[8]

```
> S6:=[s1*s2*s3*s4*uu-1,L1*L2*L3*L4*vv-1,-L2+L4-L3+L1];
syst_s6:=[op(CS5bis),op(S6)]:
S6 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, -L2 + L4 - L3 + L4] (3.6.4.1)
```

```
> CS6:=fgb_gbasis_elim(syst_s6, 0, [uu,vv], [s4, c4, s3,
c3, s2, c2, s1, c1, L4, L3, L2, L1],{'index'=
```

```

30000000000}) :nops (CS6);
                                     3
(3.6.4.2)

> map(print, factor (CS6)) :
  -(c4 c1 - c2 c3 - s3 s2 + s4 s1) (c4 c1 - c2 c3 + s3 s2 - s4 s1)
    (c4 c1 - c2 c3 + s3 s2 - s4 s1) (-L3 + L1) (L1 - L2)
      -L2 + L1 - L3 + L4
(3.6.4.3)

> S6_bis := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, (-L2+L4-L3+
  L1)*ww-1]; syst_s6_bis := [op (CS6_bis), op (S6_bis)]:
S6_bis := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, (-L2 + L1 - L3 + L4) ww
  - 1]
(3.6.4.4)

> CS6bis := fgb_gbasis_elim (syst_s6_bis, 0, [uu, vv, ww],
  [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],
  {'index'=30000000000}) :nops (CS6bis);
                                     4
(3.6.4.5)

> factor (CS6bis [4]);
  (-L2 + L1 + L3 - L4) (-L2 + L1 + L3 + L4) (L2 + L1 - L3 - L4)
(3.6.4.6)

> #map (print, factor (CS6bis)) :

```

▼ Spécialisation

$S7 := L2 + L1 - L3 - L4 : S7_bis := (L2 + L1 - L3 - L4) \cdot ww - 1 :$

,respectivement appliquée sur CS6bis[4]

```

> S7 := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, L2+L1-L3-L4];
  syst_s7 := [op (CS6bis), op (S7)]:
  S7 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, L2 + L1 - L3 - L4]
(3.6.5.1)

> CS7 := fgb_gbasis_elim (syst_s7, 0, [uu, vv], [s4, c4, s3,
  c3, s2, c2, s1, c1, L4, L3, L2, L1], {'index'=
  30000000000}) :nops (CS7);
                                     2
(3.6.5.2)

> map (print, factor (CS7)) :
  -c4 c1 + c2 c3 + s3 s2 - s4 s1
    L4 + L3 - L2 - L1
(3.6.5.3)

> S7_bis := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, (L2+L1-L3-
  L4)*ww-1]; syst_s7_bis := [op (CS6bis), op (S7_bis)]:
S7_bis := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, (L2 + L1 - L3 - L4) ww - 1]
(3.6.5.4)

> CS7bis := fgb_gbasis_elim (syst_s7_bis, 0, [uu, vv, ww],
  [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],
  {'index'=30000000000}) :nops (CS7bis);
                                     4
(3.6.5.5)

> factor (CS7bis [4]);
  -(-L2 + L1 + L3 + L4) (-L2 + L1 + L3 - L4)
(3.6.5.6)

```

```
LL > #map(print, factor(CS7bis)) :
```

Spécialisation

$S8 := -L2 + L1 + L3 + L4 : S8_bis := (-L2 + L1 + L3 + L4) \cdot ww - 1 :$

,respectivement appliquée sur CS7bis[4]

```
> S8 := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, -L2+L1+L3+L4];
syst_s8 := [op(CS7bis), op(S8)]:
S8 := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, -L2 + L1 + L3 + L4] (3.6.6.1)
```

```
> CS8 := fgb_gbasis_elim(syst_s8, 0, [uu, vv], [s4, c4, s3,
c3, s2, c2, s1, c1, L4, L3, L2, L1], {'index' =
30000000000}):nops(CS8);
2 (3.6.6.2)
```

```
> map(print, factor(CS8)) :
-c4 c1 + c2 c3 + s3 s2 + s4 s1
-L2 + L1 + L3 + L4 (3.6.6.3)
```

```
> S8_bis := [s1*s2*s3*s4*uu-1, L1*L2*L3*L4*vv-1, (-L2+L1+L3+
L4)*ww-1]; syst_s8_bis := [op(CS7bis), op(S8_bis)]:
S8_bis := [s1 s2 s3 s4 uu - 1, L1 L2 L3 L4 vv - 1, (-L2 + L1 + L3 + L4) ww
- 1] (3.6.6.4)
```

```
> CS8bis := fgb_gbasis_elim(syst_s8_bis, 0, [uu, vv, ww],
[s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1],
{'index' = 30000000000}):nops(CS8bis);
2 (3.6.6.5)
```

```
> map(print, factor(CS8bis)) :
-c4 c1 + c2 c3 + s3 s2 - s4 s1
L2 - L1 - L3 + L4 (3.6.6.6)
```

Jusqu'à ICI nous avons parcouru l'arbre de solutions complètes.

Application des Bases de Gröbner sur la Variante de la méthode de Jin et Yang - Coupure nBnD

Les conditions de mobilité du 4R

--- Ruixian RENAUD, 2013

[> restart : Digits := 20 :

▼ Cas de Test pour les 4R

▼ Cas mobile

```
> CasMobile := 'M02';
   if CasMobile = 'M01' then
     TestMobile := [ a1 =  $\frac{\text{Pi}}{6}$ , a2 = - $\frac{\text{Pi}}{3}$ , a3 =  $\frac{\text{Pi}}{6}$ , a4 = - $\frac{\text{Pi}}{3}$ , L1 = 1, L2 = sqrt(3), L3
                   = 1, L4 = sqrt(3), tA = tA, tC = tA, tB = tB, tD = tB ]
   elif CasMobile = 'M02' then
     TestMobile := [ a1 = 0.523598775598119, a2 = -1.047197551193130, a3
                   = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
                   = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
                   = 1.732050807568877 ]
   fi
                                     CasMobile := M02
TestMobile := [ a1 = 0.523598775598119, a2 = -1.047197551193130, a3
               = 0.523598775601326, a4 = -1.047197551196589, tA = 2.884851671491811, tC
               = 2.884851671491812, L1 = 1, L2 = 1.732050807568877, L3 = 1, L4
               = 1.732050807568877 ] (1.1.1)
```

▼ Cas assemblé

```
> CasAssemble := "A01";
   if CasAssemble = "A01" then
     TestAssemble := [ a1 =
                   -0.29145679447786709199560462143289119350316759901206541927220608\
                   30872990149105089980715136487087625240, a2
                   = 0.4532304095150565547563982819282587930962961566922095547515565\
                   732596602087045671757713656471993516620, a3 =
                   -0.33544273508971189050892307516029058826751697067869036278142268\
```

```

76168793297678566597158301428260336517, a4
= 0.2590958683742541729479897614925188676084960577788412458828359\
042736751821491806922090312170285686631, L1 = 1., L2
= 0.8104936767180851979185298717434445363604235377236148429308060\
054859576461637322193712084302853844524, L3
= 0.9669539802906858090162169664147757243814887591700661219785286\
325045595333890430987488315400166010690, L4
= 1.4149558296992878215532608483382186245390610309097501920888844\
27248366634018641867738328050447509528, tA
= 1.8573981902894337555851156945750084595454432632809752560892493\
00386917873900707485566229692468595627, tB
= 1.4097044222573161198546043715930495697108993447621182114967030\
69673805266101878327714905667634996138, tC
= 1.1709225040373397738924075343350458504176651525135762280840532\
31161074767785344311068817246863908114, tD
= 1.9528494843106369422422510263265186150934620521543677204766656\
35486077710669202454992473679921614923 ]

```

elif *CasAssemble* = "A02" **then**

```

TestAssemble := [a1 = 0.927295217929077, a2 = -0.519146114299891, a3
= 0.625428863600991, a4 = -0.524290345559241, L1 = 250, L2 = 200, L3
= 406.20192023, L4 = 269.25824036, tA = 1.190289949618859 , tC
= 1.694200793582639 ]

```

elif *CasAssemble* = "A03" **then**

```

TestAssemble := [L1 = 1, L2 = 0.218668460463052, L3 = 0.129411879820720, L4
= 1.085378077837818, a1 = 0.69813170079773183077, a2 =
-1.2217304763960307039, a3 = 0.87266462599716478848, a4 =
-1.2906382533029200909 ]

```

fi

CasAssemble := "A01"

TestAssemble := [*a1* = (1.2.1)

```

-0.291456794477867091995604621432891193503167599012065419272206083\
0872990149105089980715136487087625240, a2
= 0.45323040951505655475639828192825879309629615669220955475155657\
32596602087045671757713656471993516620, a3 =
-0.335442735089711890508923075160290588267516970678690362781422687\
6168793297678566597158301428260336517, a4
= 0.25909586837425417294798976149251886760849605777884124588283590\
42736751821491806922090312170285686631, L1 = 1., L2
= 0.81049367671808519791852987174344453636042353772361484293080600\
54859576461637322193712084302853844524, L3
= 0.96695398029068580901621696641477572438148875917006612197852863\
25045595333890430987488315400166010690, L4
= 1.41495582969928782155326084833821862453906103090975019208888442\
7248366634018641867738328050447509528, tA
= 1.85739819028943375558511569457500845954544326328097525608924930\
0386917873900707485566229692468595627, tB
= 1.40970442225731611985460437159304956971089934476211821149670306\
9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\

```

Cas non-assemblé

```

> CasNA := "NA01";
if CasNA = "NA01" then
  TestNA := [a1 =
    -1.29145679447786709199560462143289119350316759901206541927220608\
    30872990149105089980715136487087625240, a2
    = 0.4532304095150565547563982819282587930962961566922095547515565\
    732596602087045671757713656471993516620, a3 =
    -0.33544273508971189050892307516029058826751697067869036278142268\
    76168793297678566597158301428260336517, a4
    = 0.2590958683742541729479897614925188676084960577788412458828359\
    042736751821491806922090312170285686631, L1 = 1.5, L2
    = 0.8104936767180851979185298717434445363604235377236148429308060\
    054859576461637322193712084302853844524, L3
    = 0.9669539802906858090162169664147757243814887591700661219785286\
    325045595333890430987488315400166010690, L4
    = 1.4149558296992878215532608483382186245390610309097501920888844\
    27248366634018641867738328050447509528, tA
    = 1.8573981902894337555851156945750084595454432632809752560892493\
    00386917873900707485566229692468595627, tB
    = 1.4097044222573161198546043715930495697108993447621182114967030\
    69673805266101878327714905667634996138, tC
    = 1.1709225040373397738924075343350458504176651525135762280840532\
    31161074767785344311068817246863908114, tD
    = 1.9528494843106369422422510263265186150934620521543677204766656\
    35486077710669202454992473679921614923 ]

```

fi

CasNA := "NA01"

```

TestNA := [a1 = (1.3.1)
  -1.291456794477867091995604621432891193503167599012065419272206083\
  0872990149105089980715136487087625240, a2
  = 0.45323040951505655475639828192825879309629615669220955475155657\
  32596602087045671757713656471993516620, a3 =
  -0.335442735089711890508923075160290588267516970678690362781422687\
  6168793297678566597158301428260336517, a4
  = 0.25909586837425417294798976149251886760849605777884124588283590\
  42736751821491806922090312170285686631, L1 = 1.5, L2
  = 0.81049367671808519791852987174344453636042353772361484293080600\
  54859576461637322193712084302853844524, L3
  = 0.96695398029068580901621696641477572438148875917006612197852863\
  25045595333890430987488315400166010690, L4
  = 1.41495582969928782155326084833821862453906103090975019208888442\
  7248366634018641867738328050447509528, tA
  = 1.85739819028943375558511569457500845954544326328097525608924930\
  0386917873900707485566229692468595627, tB
  = 1.40970442225731611985460437159304956971089934476211821149670306\

```

```

9673805266101878327714905667634996138, tC
= 1.17092250403733977389240753433504585041766515251357622808405323\
1161074767785344311068817246863908114, tD
= 1.95284948431063694224225102632651861509346205215436772047666563\
5486077710669202454992473679921614923 ]

```

▼ Procédure de Simplification automatique: trouver le plus petit ensemble de polynômes générateurs d'un Idéal

```

> RRSimplification:=proc (polys, n, specs, inco, param)
#####
#####

#####          Application pour trouver le plus
petit eselbke de polynômes générateurs
#####
#####
#####

local i, supp_temp, pile, taille_supp_temp, taille_Gbtemp,
Gb_temp, jeton, taille_pile;
supp_temp:=[]:
pile:=[]:
Gb_temp:=[]:
#polys:=expand(polys):
for i from 1 to n do
supp_temp:=[op(polys[i+1..n])]:
taille_supp_temp:=nops(supp_temp):
Gb_temp:=fgb_gbasis_elim([op(pile),op(supp_temp),op(specs)
],0,[op(inco)],[op(param)]):
taille_Gbtemp:=nops(Gb_temp):
jeton:=evalb(factor(polys) = factor(Gb_temp)):
if jeton<>true then
pile:=[op(pile),polys[i]]:
taille_pile:=nops(pile):
end if:
end do:
print
(`#####
#####`):
print("#####          Fin de
la procédure
#####"):

```



```

print
(`#####`):
#####`):
print(n-taille_pile, `polynômes sont supprimés.`):
print
(`#####`):
#####`):
print(`Les`,taille_pile, `polynômes restants sont:`):
#map(print, pile):
return(pile):
end proc:

```

Obtention des Conditions de Mobilité pour la famille du 4R en 3D

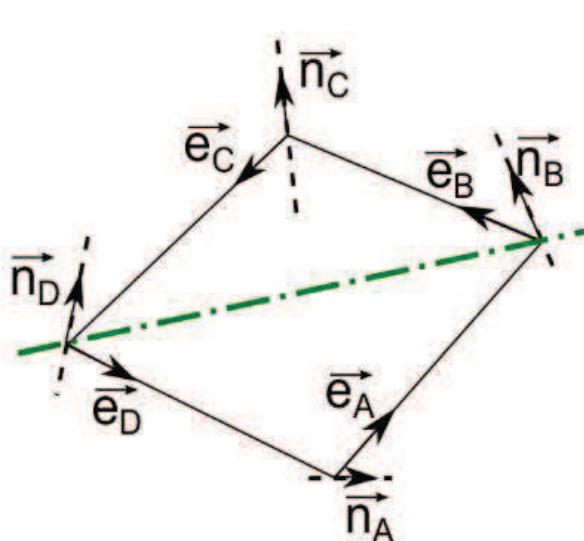


FIGURE 3.8 – La coupe effectuée sur les liaisons pivots \vec{n}_B et \vec{n}_D

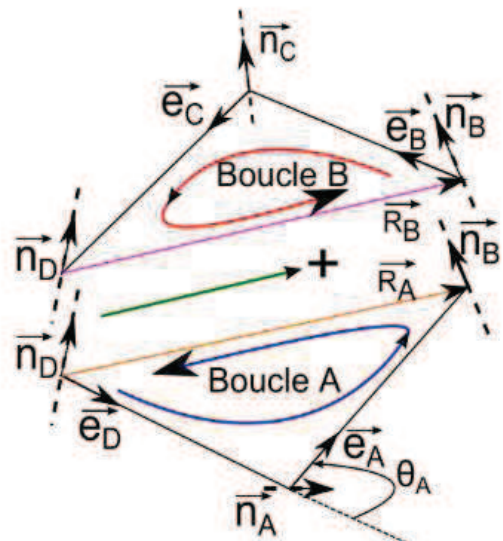


FIGURE 3.9 – La coupe sur les liaisons pivots \vec{n}_B et \vec{n}_D fait intervenir les angles θ_A et θ_C

Initialisation: transformer le système d'équations en système polynômial

$$\begin{aligned}
 & \left[\begin{aligned}
 > f11 := \cos(a2) \cdot \cos(a3) - \sin(a3) \cdot \sin(a2) \cdot \cos(tC) - \cos(a4) \cdot \cos(a1) + \sin(a1) \\
 & \quad \cdot \sin(a4) \cdot \cos(tA) \\
 f11 & := \cos(a2) \cos(a3) - \sin(a3) \sin(a2) \cos(tC) - \cos(a4) \cos(a1) \\
 & \quad + \sin(a1) \sin(a4) \cos(tA)
 \end{aligned} \right. \quad (3.1.1)
 \end{aligned}$$

$$\begin{aligned}
 & \left[\begin{aligned}
 > f12 := L4^2 + 2 \cdot L4 \cdot L1 \cdot \cos(tA) + L1^2 - L2^2 - L3^2 - 2 \cdot L2 \cdot L3 \cdot \cos(tC) \\
 f12 & := L4^2 + 2 L4 L1 \cos(tA) + L1^2 - L2^2 - L3^2 - 2 L2 L3 \cos(tC)
 \end{aligned} \right. \quad (3.1.2)
 \end{aligned}$$

$$\begin{aligned}
 & \left[\begin{aligned}
 > f13 := L3 \cdot \sin(a2) \cdot \sin(tC) + L4 \cdot \sin(tA) \cdot \sin(a1) \\
 f13 & := L3 \sin(a2) \sin(tC) + L4 \sin(tA) \sin(a1)
 \end{aligned} \right. \quad (3.1.3)
 \end{aligned}$$

$$\begin{aligned} &> f14 := L2 \cdot \sin(tC) \cdot \sin(a3) + L1 \cdot \sin(a4) \cdot \sin(tA) \\ &\quad f14 := L2 \sin(tC) \sin(a3) + L1 \sin(a4) \sin(tA) \quad (3.1.4) \\ &> syst := [f11, f12, f13, f14, \cos(tC)^2 + \sin(tC)^2 - 1, \cos(tA)^2 + \sin(tA)^2 - 1]: \end{aligned}$$

Vérification: vérifier avec une configuration d'un 4R assemblé pour voir si le système d'équations sont respectées.

$$\begin{aligned} &> Verif := subs(TestAssemble, syst) : \\ &> evalf(Verif) \\ &\quad [-1. 10^{-21}, 2. 10^{-20}, -2. 10^{-20}, -1. 10^{-20}, 0., -1. 10^{-20}] \quad (3.1.5) \end{aligned}$$

Transformation: substitution des sinus et des cosinus

$$\begin{aligned} &> subs_syst := subs(\sin(a1) = s1, \cos(a1) = c1, \sin(a2) = s2, \cos(a2) = c2, \\ &\quad \sin(a3) = s3, \cos(a3) = c3, \sin(a4) = s4, \cos(a4) = c4, \sin(tA) = sA, \cos(tA) \\ &\quad = cA, \sin(tC) = sC, \cos(tC) = cC, syst) \\ &\quad subs_syst := [c2 c3 - s3 s2 cC - c4 c1 + s1 s4 cA, L4^2 + 2 L4 L1 cA + L1^2 - L2^2 \\ &\quad - L3^2 - 2 L2 L3 cC, L3 s2 sC + L4 sA s1, L2 sC s3 + L1 s4 sA, cC^2 + sC^2 \\ &\quad - 1, cA^2 + sA^2 - 1] \quad (3.1.1.1) \end{aligned}$$

1ère élimination: éliminer 3 inconnues sC, cC, sA

$$\begin{aligned} &> with(FGb) : \\ &> syst := [op(subs_syst)] : \\ &> gb_subs_syst := factor(fgb_gbasis_elim(subs_syst, 0, [sC, cC, sA], [cA, s4, c4, s3, \\ &\quad c3, s2, c2, s1, c1, L4, L3, L2, L1])) : \\ &> taille := nops(gb_subs_syst) \quad \text{taille := 289} \quad (3.2.1) \end{aligned}$$

$$\begin{aligned} &> map(print, collect(gb_subs_syst[288..289], cA)) : \\ &\quad (s4 s1 L1^2 + 2 c3 c2 L4 L1 - 2 c4 c1 L4 L1 + s4 s1 L4^2 - L3^2 s4 s1 - s1 L2^2 s4) cA \\ &\quad + 2 s4 s1 L4 L1 + c3 c2 L1^2 - c4 c1 L1^2 - c4 c1 L4^2 - c3 c2 L3^2 + c4 c1 L3^2 \\ &\quad + c3 c2 L4^2 - c3 c2 L2^2 + c4 c1 L2^2 - 2 s3 s2 L3 L2 \\ &\quad (2 s4 s1 L3 L2 - 2 L1 s2 s3 L4) cA + 2 c3 c2 L3 L2 - 2 c4 c1 L3 L2 - s3 s2 L4^2 \\ &\quad + s3 s2 L3^2 + s3 s2 L2^2 - s3 s2 L1^2 \quad (3.2.2) \end{aligned}$$

Vérification du premier calcul: ces 289 équations polynômiales doivent être toujours vérifiées par un 4R assemblé

$$\begin{aligned} &> Injection1 := subs((c1 = \cos(a1), s1 = \sin(a1), c2 = \cos(a2), s2 = \sin(a2), c3 \\ &\quad = \cos(a3), s3 = \sin(a3), c4 = \cos(a4), s4 = \sin(a4), sC = \sin(tC), cC \\ &\quad = \cos(tC), sA = \sin(tA), cA = \cos(tA)), gb_subs_syst) : \end{aligned}$$

> Verif1 := subs(TestAssemble, Injection1) :

> evalf(Verif1)

[1.301093 10⁻²⁰, -2.1558768001555445010 10⁻²⁴,
-6.4284326404638054211 10⁻²⁴, -5. 10⁻¹⁹, -2.1 10⁻²⁰, -2.4 10⁻²⁰,
-3.01 10⁻¹⁹, 7.11 10⁻²⁰, 1.497 10⁻¹⁹, 8.100 10⁻¹⁹, -5.280711 10⁻¹⁸,
-6.64 10⁻²⁰, -4. 10⁻¹⁹, -1.1 10⁻¹⁹, -5.912 10⁻²⁰, 9.149 10⁻¹⁹, -1.01425 10⁻¹⁸,
-1.237 10⁻¹⁹, 1.87 10⁻¹⁸, 9.6 10⁻¹⁹, -1.4670 10⁻¹⁹, -3. 10⁻²⁰, -8.60 10⁻²¹,
1.4 10⁻¹⁹, -2.2 10⁻¹⁸, 5.0 10⁻²⁰, -3.1 10⁻¹⁹, -5.6 10⁻¹⁹, -1.133 10⁻¹⁹,
-2.73 10⁻¹⁸, -1.1 10⁻¹⁹, 0., -2.4 10⁻¹⁹, 0., 2.18 10⁻²¹, -4. 10⁻¹⁹,
-1.32827 10⁻¹⁹, 5.9 10⁻²⁰, 1.30 10⁻¹⁹, 3.0 10⁻²⁰, -6. 10⁻¹⁹, -5.675 10⁻¹⁹,
-9.0 10⁻²⁰, 3.00677 10⁻²⁰, 4. 10⁻¹⁹, 1.6 10⁻²⁰, 1.835 10⁻¹⁸, 1. 10⁻¹⁹, 9.4 10⁻²⁰,
-8. 10⁻²¹, 1.360 10⁻¹⁹, -6.5 10⁻²¹, 2. 10⁻²⁰, 1.88 10⁻¹⁸, -2.3 10⁻²⁰, 2.1 10⁻¹⁹,
-8.6 10⁻²⁰, -4.59 10⁻¹⁹, 5.3386 10⁻²⁰, 1. 10⁻¹⁹, -5. 10⁻²⁰, -1. 10⁻²⁰, 1.1 10⁻¹⁹,
-3.0 10⁻¹⁹, -1. 10⁻²¹, 1.70 10⁻²⁰, -1.8 10⁻²⁰, 3.2 10⁻²⁰, 1.62 10⁻¹⁹,
-3.08 10⁻¹⁹, 9.1 10⁻¹⁹, -1. 10⁻¹⁹, 1. 10⁻¹⁹, -1.4 10⁻¹⁹, 3.8 10⁻¹⁹, -1.90 10⁻¹⁹,
-1.42 10⁻¹⁸, 8.3 10⁻¹⁹, 4.496 10⁻²⁰, -2.3892 10⁻¹⁹, -6.190 10⁻¹⁹, 4.7 10⁻¹⁸,
-1. 10⁻²⁰, 1.50 10⁻¹⁹, 1.0 10⁻¹⁹, -4. 10⁻²⁰, -2.1 10⁻²⁰, 2.2 10⁻¹⁹, -7.29 10⁻²⁰,
-3.1 10⁻¹⁹, -1.6545 10⁻¹⁸, -7.4 10⁻²⁰, -3.8 10⁻²⁰, 1.6 10⁻²⁰, 2.89 10⁻¹⁹,
2.5 10⁻²⁰, 1.9 10⁻²⁰, -6.4 10⁻¹⁹, 1. 10⁻²¹, -1.5 10⁻²⁰, 4.57 10⁻¹⁹, -7. 10⁻²⁰,
-8.05 10⁻²⁰, 2. 10⁻²⁰, -1.5 10⁻¹⁸, -4.8 10⁻¹⁹, -4.79 10⁻¹⁹, 6. 10⁻²¹, -0.,
-4.2 10⁻¹⁹, 3.8 10⁻²⁰, -5.99 10⁻¹⁹, -1. 10⁻¹⁹, -4. 10⁻¹⁹, -3. 10⁻²⁰, 2. 10⁻²⁰,
6. 10⁻²⁰, -2.3 10⁻¹⁹, 1.72 10⁻¹⁸, 8.7 10⁻¹⁹, -2.5 10⁻¹⁹, -1.120 10⁻¹⁸, 5.1 10⁻¹⁹,
2.64 10⁻¹⁹, 2.64 10⁻¹⁸, -7.4 10⁻¹⁹, 1.1247 10⁻¹⁹, -2.0626 10⁻¹⁹, -7.0 10⁻²⁰, 0.,
0., 0., 0., -2.0 10⁻²⁰, 1. 10⁻²⁰, 0., 0., 0., -0., 0., -5.9 10⁻²⁰, 1.8 10⁻²¹, -0.,
-0., 0., 2. 10⁻²⁰, 2.4 10⁻¹⁹, -5. 10⁻²⁰, 2.09 10⁻¹⁹, -5.40 10⁻¹⁹, 9.99 10⁻²⁰,
1.5 10⁻²⁰, -6.5 10⁻²⁰, -1.7 10⁻¹⁹, -1. 10⁻²⁰, 4.06 10⁻²⁰, -0., 2.6 10⁻²¹,
-1.5 10⁻²⁰, -8. 10⁻²¹, 2. 10⁻²¹, -9. 10⁻²¹, -4.4 10⁻¹⁹, -3.5 10⁻²⁰, -1.8 10⁻¹⁹,
1.6 10⁻¹⁹, -1.0 10⁻¹⁹, 9. 10⁻¹⁹, -2.0 10⁻²⁰, 6.5 10⁻²⁰, -1.5 10⁻¹⁹, 0., 1. 10⁻²⁰,
1.80 10⁻¹⁸, 1.2 10⁻²⁰, -1.0343 10⁻¹⁸, 1.83 10⁻¹⁹, 1.9 10⁻¹⁹, -3.6 10⁻²⁰,
2.13 10⁻¹⁹, -1.9 10⁻²⁰, -1.968 10⁻¹⁹, 3. 10⁻²⁰, -8. 10⁻²⁰, 3.3 10⁻¹⁹, 9. 10⁻²¹,
-1. 10⁻²⁰, -7.1 10⁻²⁰, -3.28 10⁻²⁰, 2. 10⁻²¹, 2.0 10⁻²⁰, 4. 10⁻²⁰, 1. 10⁻¹⁹,
-8. 10⁻²¹, -8.11 10⁻²¹, -8.8 10⁻²⁰, -2. 10⁻¹⁹, -0., -4. 10⁻²⁰, 1.52 10⁻¹⁹,
1.966 10⁻¹⁹, 1.1 10⁻¹⁹, -9. 10⁻²⁰, 2.9 10⁻²⁰, 1.0 10⁻²⁰, -1.3 10⁻¹⁹, 7. 10⁻¹⁹,
1. 10⁻¹⁹, 1. 10⁻¹⁹, 5. 10⁻²⁰, 0., -2.0 10⁻²⁰, 1.2 10⁻¹⁹, 4.7 10⁻²⁰, -1. 10⁻¹⁹,
-2. 10⁻²¹, -1. 10⁻¹⁹, -4.0 10⁻²⁰, -1.18 10⁻¹⁹, -7.6 10⁻²⁰, -7.9 10⁻²¹,
-6. 10⁻²⁰, -6. 10⁻²⁰, 6. 10⁻²⁰, -3.3 10⁻²⁰, 6.5 10⁻²⁰, 3. 10⁻²⁰, 2.445 10⁻²⁰,
-2. 10⁻²¹, 1. 10⁻²⁰, 2. 10⁻²¹, 4. 10⁻²⁰, 1. 10⁻²⁰, 1.840 10⁻²⁰, 1.8 10⁻²⁰,
-1.2 10⁻²², -4. 10⁻¹⁹, -1.2 10⁻¹⁹, 2.5 10⁻²⁰, -1.407 10⁻²⁰, 1.4 10⁻¹⁹, -6. 10⁻¹⁹,
1. 10⁻¹⁹, 0., 1. 10⁻¹⁹, 0., -5. 10⁻²⁰, 4.63 10⁻²⁰, -1. 10⁻²⁰, -6.3 10⁻²⁰,

(3.2.1.1)

```

-1.9 10-19, -1.7 10-20, 8. 10-20, -5. 10-20, 5. 10-20, -2. 10-20, 0., 6. 10-20, 0.,
0., 0., 5. 10-21, -1. 10-20, -0., -1.2 10-20, -9. 10-21, 0., -8. 10-20, -7. 10-19,
-3. 10-21, -0., -2. 10-21, 0., -1. 10-20, 6. 10-20, -2. 10-20, -0., 3. 10-20,
-1. 10-19, -4.7 10-20, -9. 10-20, -1.3 10-20, -2. 10-21, 3. 10-20, -5. 10-20, 0.,
1. 10-20, 0.]

```

▼ Ajouter Q(sa,ca)=0

```
[> Q := [s12 + c12 - 1, s22 + c22 - 1, s32 + c32 - 1, s42 + c42 - 1]:
```

▼ Spécialisation:

```
[> S1 := [L4·L3·L2·L1·uu - 1, s1·s2·s3·s4·vv - 1]:
```

▼ Extraction des coefficients: Récupérer les coefficients de cA, les mettre dans une liste, puis recalculer la base de Gröbner

```
[> new_liste := [ ]:
> for i from 1 to taille do
  terme[i] := collect(gb_subs_syst[i], cA) :
  Coefficients[i] := [coeffs(terme[i], [cA], 'k')]:
  new_liste := [op(new_liste), op(Coefficients[i])]:
end do:
> nops(new_liste)
537
(3.5.1)
```

▼ 2ème élimination

```
[> nBnD_mobilite := fgb_gbasis_elim([op(new_liste), op(Q), op(S1)], 0, [uu,
vv], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1]) :#map(print,
factor(nBnD_mobilite)):
> taille_mobilite := nops(nBnD_mobilite)
taille_mobilite := 107
(3.5.1.1)
```

▼ Cas L1-L3

▼ Spécialisation:

```
[> S2 := [L1·L2·L3·L4·uu - 1, s1·s2·s3·s4·vv - 1, L1 - L3]
```

$S2 := [L4 L3 L2 L1 uu - 1, s1 s2 s3 s4 vv - 1, L1 - L3]$ (3.5.2.1.1)

3ème élimination:

```
> temp1 := factor(fgb_gbasis_elim([op(nBnD_mobilite), op(S2)], 0, [uu,
  vv], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1])) : taille_temp1
  := nops(temp1);#map(print, factor(temp1)) :
  taille_temp1 := 43 (3.5.2.2.1)
```

Spécialisation:

```
> S3 := [L1·L2·L3·L4·uu - 1, s1·s2·s3·s4·vv - 1, L1 - L3, L2 - L4]
  S3 := [L4 L3 L2 L1 uu - 1, s1 s2 s3 s4 vv - 1, L1 - L3, L2 - L4] (3.5.2.3.1)
```

4ème élimination:

```
> temp2 := fgb_gbasis_elim([op(temp1), op(S3)], 0, [uu, vv], [s4, c4, s3, c3,
  s2, c2, s1, c1, L4, L3, L2, L1]) : taille_temp2 := nops(temp2);#map(print,
  temp2) :
  taille_temp2 := 22 (3.5.2.4.1)
```

Simplification automatique: le plus petit ensemble de polynômes générateurs qui représentent le même idéal

```
> S4 := [L1·L2·L3·L4·uu - 1]
  S4 := [L4 L3 L2 L1 uu - 1] (3.5.2.5.1)
> Simp1 := RRSimplification(temp2, taille_temp2, S4, [uu, vv], [s4, c4, s3, c3,
  s2, c2, s1, c1, L4, L3, L2, L1]) : taille_Simp1 := nops(Simp1); map(print,
  Simp1) :
#####
#####
"##### Fin de la procédure
#####"
#####
#####
  9, polynômes sont supprimés.
#####
#####
  Les, 13, polynômes restants sont:
  taille_Simp1 := 13
```

$$\begin{aligned}
& s4 s3 L2^2 - s2 s1 L2^2 - s4 s3 L1^2 + s2 s1 L1^2 \\
& c3^2 L2^2 - c3^2 L1^2 - c2^2 L1^2 + c1^2 L1^2 - L2^2 + L1^2 \\
& s3 s1 L2^2 - s4 s2 L1^2 \\
& c1^2 L2^2 - c2^2 L1^2 - L2^2 + L1^2 \\
& s4^2 - 1 + c3^2 + c2^2 - c1^2 \\
& c4^2 - c3^2 - c2^2 + c1^2 \\
& s3^2 + c3^2 - 1 \\
& s3 s2 - s4 s1 \\
& s2^2 + c2^2 - 1 \\
& c2 c3 - c4 c1 \\
& s1^2 + c1^2 - 1 \\
& L4 - L2 \\
& L3 - L1
\end{aligned}$$

(3.5.2.5.2)

> with(Groebner) : HilbertDimension(Simpl)
3

(3.5.2.5.3)

▼ Résolution et simplification manuelle

> Injection5 := subs((c1 = cos(a1), s1 = sin(a1), c2 = cos(a2), s2 = sin(a2), c3 = cos(a3), s3 = sin(a3), c4 = cos(a4), s4 = sin(a4), sB = sin(tB), cB = cos(tB), sD = sin(tD), cD = cos(tD)), Simpl) :

> Eq_CM_nBnD := [] :

> for i from 1 to taille_Simpl do
temp[i] := {Injection5[i]=0} :
Eq_CM_nBnD := [op(Eq_CM_nBnD), op(temp[i])] :
end do:

> map(print, Eq_CM_nBnD) :

$$\begin{aligned}
& \sin(a4) \sin(a3) L2^2 - \sin(a2) \sin(a1) L2^2 - \sin(a4) \sin(a3) L1^2 \\
& + \sin(a2) \sin(a1) L1^2 = 0
\end{aligned}$$

$$\begin{aligned}
& \cos(a3)^2 L2^2 - \cos(a3)^2 L1^2 - \cos(a2)^2 L1^2 + \cos(a1)^2 L1^2 - L2^2 + L1^2 \\
& = 0
\end{aligned}$$

$$\sin(a3) \sin(a1) L2^2 - \sin(a4) \sin(a2) L1^2 = 0$$

$$\cos(a1)^2 L2^2 - \cos(a2)^2 L1^2 - L2^2 + L1^2 = 0$$

$$\sin(a4)^2 - 1 + \cos(a3)^2 + \cos(a2)^2 - \cos(a1)^2 = 0$$

$$\cos(a4)^2 - \cos(a3)^2 - \cos(a2)^2 + \cos(a1)^2 = 0$$

$$\sin(a3)^2 + \cos(a3)^2 - 1 = 0$$

$$\sin(a3) \sin(a2) - \sin(a4) \sin(a1) = 0$$

$$\sin(a2)^2 + \cos(a2)^2 - 1 = 0$$

$$\cos(a2) \cos(a3) - \cos(a4) \cos(a1) = 0$$

$$\sin(a1)^2 + \cos(a1)^2 - 1 = 0$$

$$L4 - L2 = 0$$

$$L3 - L1 = 0$$

(3.5.2.6.1)

> CM_nBnD_mobile := solve({Eq_CM_nBnD[8], Eq_CM_nBnD[10]}, {a1,

$a2\}$)
 $CM_nBnD_mobile := \{a1 = a3, a2 = a4\}, \{a1 = -a3, a2 = -a4\}$ (3.5.2.6.2)

▼ Substitution des 2 solutions et simplification

▼ Substitution de $a1=a3, a2=a4$

> $Res1_inject := subs((c1 = \cos(a1), s1 = \sin(a1), c2 = \cos(a2), s2 = \sin(a2), c3 = \cos(a3), s3 = \sin(a3), c4 = \cos(a4), s4 = \sin(a4), sB = \sin(tB), cB = \cos(tB), sD = \sin(tD), cD = \cos(tD)), Simp1) :$

> $res1 := simplify(subs(a3 = a1, a4 = a2, Res1_inject), \{\cos(a1)^2 + \sin(a1)^2 = 1, \cos(a2)^2 + \sin(a2)^2 = 1\}) : map(print, res1) :$

$$0$$

$$\cos(a1)^2 L2^2 - \cos(a2)^2 LI^2 - L2^2 + LI^2$$

$$L2^2 - \cos(a1)^2 L2^2 - LI^2 + \cos(a2)^2 LI^2$$

$$\cos(a1)^2 L2^2 - \cos(a2)^2 LI^2 - L2^2 + LI^2$$

$$0$$

$$0$$

$$0$$

$$0$$

$$0$$

$$0$$

$$0$$

$$L4 - L2$$

$$L3 - L1$$

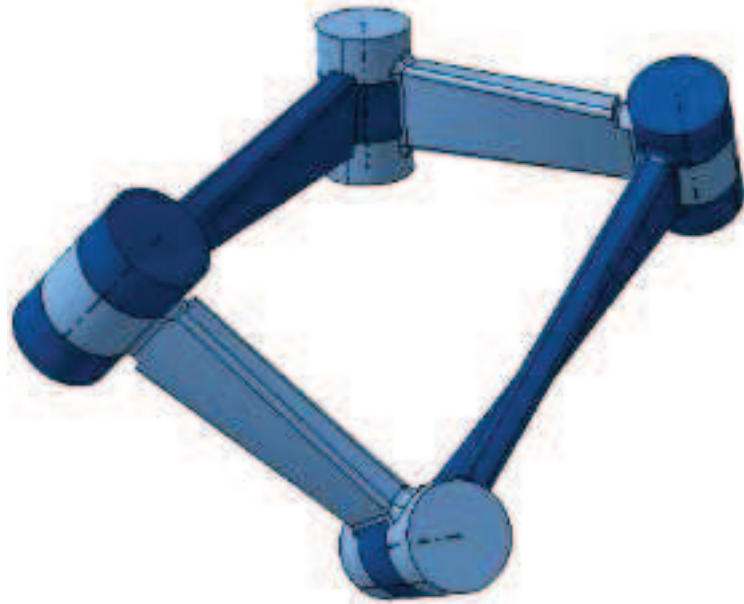
(3.5.2.6.1.1.1)

▼ Etude exhaustive: les 2 solutions symboliques des angles d'usinage

D'après les résultats CM_angle , seulement 4 solutions sont en symbolique. Elles sont

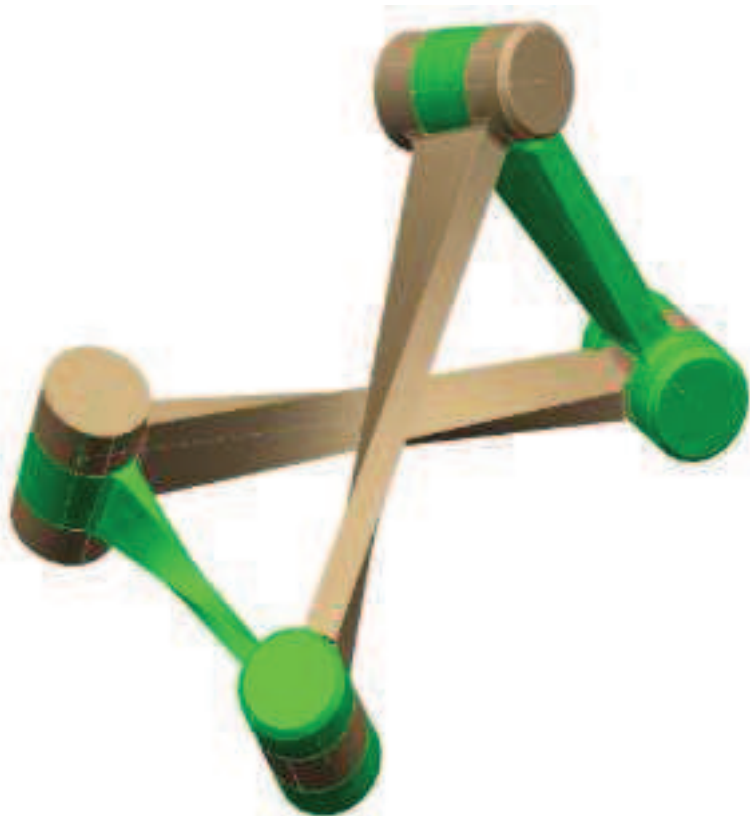
1. $\{a1 = a3, a4 = a2\} :$

Avec $L1=L3, L2=L4$ et $s1^2 L2^2 - s2^2 LI^2 = 0$, cet ensemble d'équations est la condition de mobilité pour la famille du mécanisme de Bennett publié par Beggs 1966.



2. $\{a_1 = a_3 - \pi, a_4 = a_2 - \pi\}$:

Avec $L_1=L_3, L_2=L_4$ et $s_1^2 L_2^2 - s_2^2 L_1^2 = 0$, cet ensemble d'équations est aussi la condition de mobilité du mécanisme de Bennett. Les mécanismes respectant cette condition possède la mobilité globale.



▼ Cas L2-L4

▼ Spécialisation:

```
[ > S5 := [L1·L2·L3·L4·uu - 1, s1·s2·s3·s4·vv - 1, L1 - L2]
      S5 := [L4 L3 L2 L1 uu - 1, s1 s2 s3 s4 vv - 1, L1 - L2] (3.5.3.1.1)
```

▼ 3ème élimination:

```
[ > temp3 := factor(fgb_gbasis_elim([op(nBnD_mobilite), op(S5)], 0, [uu,
      vv], [s4, c4, s3, c3, s2, c2, s1, c1, L4, L3, L2, L1])) : taille_temp3
      := nops(temp3);#map(print, factor(temp3)) :
      taille_temp3 := 28 (3.5.3.2.1)
```

▼ Spécialisation:

```
[ > S6 := [L1·L2·L3·L4·uu - 1, s1·s2·s3·s4·vv - 1, L1 - L2, L3 - L4]
      S6 := [L4 L3 L2 L1 uu - 1, s1 s2 s3 s4 vv - 1, L1 - L2, L3 - L4] (3.5.3.3.1)
```

▼ 4ème élimination:

```
[ > temp4 := fgb_gbasis_elim([op(temp3), op(S6)], 0, [uu, vv], [s4, c4, s3, c3,
      s2, c2, s1, c1, L4, L3, L2, L1]) : taille_temp4 := nops(temp4);#map(print,
      temp4) :
      taille_temp4 := 16 (3.5.3.4.1)
```

▼ Simplification automatique: le plus petit ensemble de polynômes générateurs qui représentent le même idéal

```
[ > S7 := [L1·L2·L3·L4·uu - 1, s1·s2·s3·s4·vv - 1]
      S7 := [L4 L3 L2 L1 uu - 1, s1 s2 s3 s4 vv - 1] (3.5.3.5.1)
```

```
[ > Simp2 := RRSimplification(temp4, taille_temp4, S7, [uu, vv], [s4, c4, s3, c3,
      s2, c2, s1, c1, L4, L3, L2, L1]) : taille_Simp2 := nops(Simp2); map(print,
      Simp2) :
```

```
#####\
#####
"##### Fin de la procédure
#####"
#####\
#####
```

7, polynômes sont supprimés.

```
#####
```

```
#####
```

Les, 9, polynômes restants sont:

$taille_Simp2 := 9$

$$c4^2 - c3^2$$

$$s3^2 + c3^2 - 1$$

$$s3 s2 - s4 s1$$

$$s2^2 - 1 + c1^2$$

$$c2 c3 - c4 c1$$

$$c2^2 - c1^2$$

$$s1^2 + c1^2 - 1$$

$$L4 - L3$$

$$L2 - L1$$

(3.5.3.5.2)

```
> with(Groebner) : HilbertDimension(Simp2)
```

4

(3.5.3.5.3)

▼ Résolution et simplification manuelle

```
> Injection6 := subs( ( c1 = cos(a1), s1 = sin(a1), c2 = cos(a2), s2 = sin(a2), c3  
= cos(a3), s3 = sin(a3), c4 = cos(a4), s4 = sin(a4), sB = sin(tB), cB  
= cos(tB), sD = sin(tD), cD = cos(tD) ), Simp2 ) :
```

```
> Eq_CM_nBnD := [ ] :
```

```
> for i from 1 to taille_Simp2 do
```

```
temp[i] := {Injection6[i]=0} :
```

```
Eq_CM_nBnD := [op(Eq_CM_nBnD), op(temp[i])] :
```

```
end do:
```

```
> map(print, Eq_CM_nBnD) :
```

$$\cos(a4)^2 - \cos(a3)^2 = 0$$

$$\sin(a3)^2 + \cos(a3)^2 - 1 = 0$$

$$\sin(a3) \sin(a2) - \sin(a4) \sin(a1) = 0$$

$$\sin(a2)^2 - 1 + \cos(a1)^2 = 0$$

$$\cos(a2) \cos(a3) - \cos(a4) \cos(a1) = 0$$

$$\cos(a2)^2 - \cos(a1)^2 = 0$$

$$\sin(a1)^2 + \cos(a1)^2 - 1 = 0$$

$$L4 - L3 = 0$$

$$L2 - L1 = 0$$

(3.5.3.6.1)

```
> CM_nBnD_mobile := solve( {Eq_CM_nBnD[3], Eq_CM_nBnD[5]}, {a1,  
a3} )
```

```
CM_nBnD_mobile := {a1 = a2, a3 = a4}, {a1 = -a2, a3 = -a4}
```

(3.5.3.6.2)

▼ Substitution des 2 solutions et simplification

▼ Substitution de $a1=a2, a3=a4$

```
> Res2_inject := subs( ( c1 = cos(a1), s1 = sin(a1), c2 = cos(a2), s2  
= sin(a2), c3 = cos(a3), s3 = sin(a3), c4 = cos(a4), s4 = sin(a4),
```

```

sB = sin(tB), cB = cos(tB), sD = sin(tD), cD = cos(tD)), Simp2) :
> res2 := simplify(subs(a2 = a1, a4 = a3, Res1_inject), {cos(a1)^2
+ sin(a1)^2 = 1, cos(a2)^2 + sin(a2)^2 = 1, cos(a3)^2 + sin(a3)^2
= 1}) : map(print, res2) :
0
0
0
0
0
0
0
0
L4 - L3
L2 - L1

```

(3.5.3.6.1.1.1)

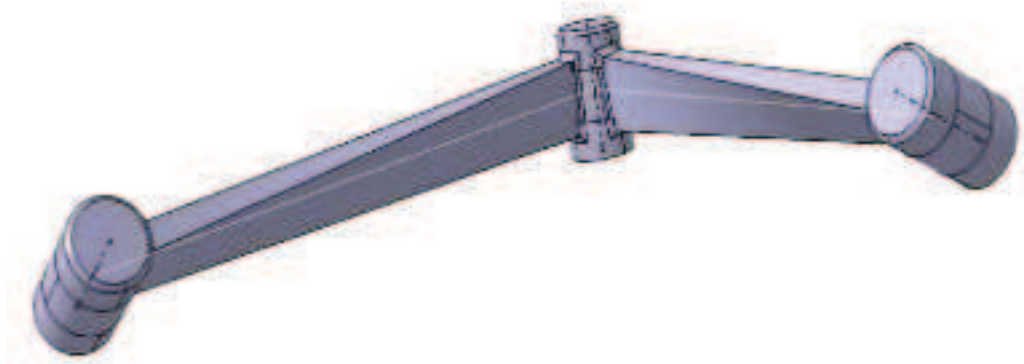


FIGURE 3.11 – Un mécanisme 4R mobile avec deux solides superposés aux deux autres.