THÈSE[1]

Pour l'obtention du grade de Docteur de l'Université de Poitiers

(Faculté de Sciences Fondamentales et appliquées)

(Diplôme national - Arrêté du 25 Avril 2002)

ÉCOLE DOCTORALE SCIENCES POUR L'INGENIEUR

Secteur de Recherche :

GENIE MECANIQUE, PRODUCTIQUE, TRANSPORT

Présentée par :

Samir LAHOUAR

CONTRIBUTION A LA MODELISATION ET A LA PLANIFICATION DE TRAJECTOIRES DE ROBOTS PARALLELES EVITANT LES SINGULARITES ET LES COLLISIONS

Directeurs de thèse : Saïd ZEGHLOUL, Lotfi ROMDHANE

Soutenue le 30 Janvier 2008
devant la Commission d'Examen

JURY

M. P. WENGER	Directeur de Recherche – I.R.C.Cy. Nantes	**Rapporteur**
M. P. BIDAUD	Professeur des Universités – U. P M. C.	**Rapporteur**
M. H. BEL HADJ SALAH	Maître de Conférences – L.G.M. -E.N.I. Monastir	**Examinateur**
M. A. BEN AMARA	Maître de Conférences – L.G.M. -E.N.I. Monastir	**Examinateur**
M. L. ROMDHANE	Professeur des Universités – L.G.M. -E.N.I. Sousse	**Examinateur**
M. S. ZEGHLOUL	Professeur des Universités – L.M.S. -U. Poitiers	**Examinateur**

---

[1] Cette thèse s'est déroulée dans le cadre d'une cotutelle entre l'Université de Poitiers - UP (FRANCE) et l'École Nationale d'Ingénieurs de Monastir - ENIM (TUNISIE)

Contribution à la modélisation et à la planification de trajectoires de robots parallèles évitant les singularités et les collisions

L'objectif principal de ce travail est l'étude de la planification de trajectoires pour les robots parallèles en prenant en considération les collisions et les singularités. Les développements ont été intégrés dans le logiciel SMAR du LMS. Ce logiciel, à l'origine limité aux robots sériels, a été enrichi en changeant sa structure pour permettre la modélisation et l'analyse des robots parallèles. Une étude des robots parallèles et des notions qui s'y rapportent, ainsi que les différentes structures de robots et les types de maillons et de liaisons qui les composent sont décrites. Les chaînes fermées sont modélisées à l'aide de différents types de contraintes entre maillons. Une méthode pour générer les équations du modèle cinématique en séparant les variables actives, passives et opérationnelles est présentée. Une méthode de planification de trajectoires permettant d'éviter les singularités et les collisions internes et externes est proposée. Cette méthode combine deux modes: un mode de recherche en profondeur et un mode de recherche en largeur. Un exemple de robot 3-RPR est traité pour illustrer les développements de cette méthode.

Des résultats expérimentaux sont présentés pour valider le modèle. La première expérimentation est réalisée sur le robot translateur conçu au LGM Monastir et construit au L.M.S. La seconde expérience a été conduite à l'Université de Cassino sur un robot à câbles. Enfin, la dernière expérimentation est originale puisqu'elle a été menée sur un robot chien, qui a été traité comme un robot parallèle. Les résultats obtenus ont servi à valider les modèles développés dans la thèse.

Mots clés : Robots parallèles, planification de trajectoire, évitement de singularités, évitement de collisions

Contribution to parallel robots modeling and path planning avoiding singularities and collisions

The main objective of this work is the analysis and the modeling of parallel robots. The problem of path planning avoiding collisions and singularities is studied. The software SMAR, developed in LMS, was used to implement the different algorithms. This software, initially limited to serial robots, was modified in order to cope with parallel robots. First, an analysis of the parallel structures and their different links and joints, is described. Closed kinematic chains are modeled using different types of geometry constraints, which led us to propose a method to generate automatically the kinematic model and separate the passive, active and operational parameters. Second, a method of path planning avoiding both singularities and collisions is proposed. This method combines two modes: a depth search mode and a width search mode. A 3-RPR example is given to illustrate the obtained results.

The second part of this thesis presents some experiments to validate the proposed models. The first experiment used a translational robot designed in LGM Monastir and built in LMS. The second experiment was done at the University of Cassino on a 4-cable-driven-robot. The last experiment using a dog robot, showed how this type of structures can be treated as a parallel one. This approach turned out to be an effective way to control such complex structures in a simple and effective way.

Keywords: Parallel robots, path planning, singularity avoidance, collision avoidance

Avant-propos

Ce travail a été réalisé pendant trois années de thèse en cotutelle entre le Laboratoire de Mécanique des Solides de Poitiers dirigé par M. Olivier Bonneau et le Laboratoire de Génie Mécanique de Monastir dirigé par M. Abdelwahab Dogui. Je les remercie de m'avoir permis de conduire mes recherches dans de bonnes conditions.

Mes plus profonds remerciements vont naturellement à mes directeurs de thèse M. Saïd Zeghloul et M. Lotfi Romdhane. Leurs qualités scientifiques et humaines m'ont permis de mener à bien ce travail.

Je remercie très sincèrement les membres du jury et surtout les rapporteurs, M. Philippe Wenger et M. Philippe Bidaud d'avoir accepté cette tâche.

Je remercie également M. Hédi Bel Haj Salah et M. Abdelmajid Ben Amara pour avoir accepté de juger ce travail, malgré les tâches administratives que cela a nécessité.

Mes remerciements vont aussi à M. Marco Ceccarelli du LARM de Cassino en Italie pour m'avoir accueilli dans son laboratoire, au sein duquel une partie de ce travail a été effectuée. Cette coopération a été une expérience très enrichissante pour moi.

Je n'oublie pas mes collègues et surtout Jean-Pierre, Pascal, Antoine, Luc, Fred, Amine et Naël pour leurs lumières et la bonne ambiance au sein de l'équipe. J'adresse mes remerciements à l'ensemble des membres du laboratoire et surtout aux superbes secrétaires Laurence et Djamila pour leur gentillesse et leur disponibilité.

Je veux exprimer ma sincère amitié aux copains qui m'ont soutenu pendant mon séjour en France. J'adresse donc une « Spéciale dédicace » à Cyril, Julie, Estelle, Racole, J-B, Nathalie et Thierry sans oublier Hussein et Tarek.

Je veux exprimer mon amitié aux moniteurs du CIES Centre pour les bons moments que nous avons passés pendant les stages résidentiels à la Rochelle.

Ce travail est dédié naturellement à mes parents, ainsi qu'à toute ma famille sans laquelle ce travail n'aurait pas été possible.

Introduction

L'émergence de la robotique « intelligente » conduit de nos jours à rendre les robots de plus en plus autonomes et de plus en plus faciles à piloter. Avec une simple action définissant la tâche désirée, un opérateur peut gérer le mouvement de plusieurs robots coopérant dans le cadre de l'exécution d'une tâche unique. Nul besoin de se soucier aujourd'hui des trajectoires individuelles de chaque robot, alors qu'il y a encore quelques années, l'opérateur devait envisager chaque situation en spécifiant la trajectoire complète du robot afin d'éviter les collisions avec les différents objets paramétrés au préalable.

Sur le plan conceptuel, les robots sont également de plus en plus fiables et précis. Beaucoup de nouvelles architectures de robots sont proposées par les chercheurs afin de répondre à ces exigences de fiabilité et de précision. Ces architectures sont le plus souvent de type parallèle, appelées ainsi, par opposition aux robots sériels. Dans les robots sériels, les différentes articulations sont montées en série (voir l'exemple de la figure I.1). Les robots parallèles sont composés, quant à eux, de deux plateformes : une fixe et une mobile. Ces deux plateformes sont reliées par des jambes montées en parallèle. Les robots parallèles présentent plusieurs avantages ; par exemple une plus grande rigidité offrant une précision meilleure que celle relative aux robots sériels. Les pièces en mouvement sont aussi plus légères ; d'où une diminution des effets inertiels. En revanche cette structure correspond à des chaînes cinématiques fermées ce qui complique la modélisation, réduit le volume de l'espace de travail et fait apparaître des singularités à l'intérieur de l'espace de travail. Les singularités sont des postures où le robot devient incontrôlable ; elles sont dangereuses pour la structure et il est nécessaire de les éviter lors du mouvement du robot.

Ainsi ce mémoire de thèse s'inscrit dans ce contexte, où l'on souhaite accroître autonomie et fiabilité de ces structures parallèles, prometteuses en termes d'applications robotiques. La planification de trajectoire de ces robots parallèles en évitant les collisions et les singularités de l'espace de travail sera donc au centre de notre étude.

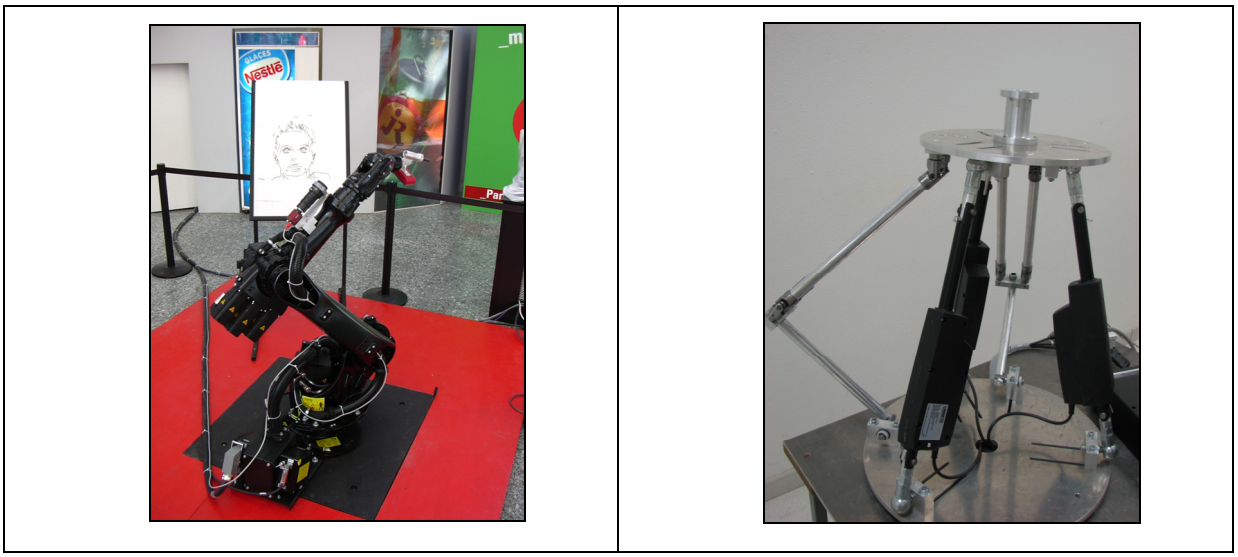


Figure I.1. Un robot sériel et un robot parallèle

La planification de trajectoire sans collision a été largement étudiée dans le cadre des robots mobiles et des robots sériels. Ce problème reste toutefois peu abordé dans le cas des robots parallèles. On peut expliquer cet état de fait, d'une part en raison du développement assez récent des robots parallèles en comparaison avec les robots sériels et les robots mobiles ; et d'autre part à cause de l'espace de travail réduit des robots parallèles, qui sont souvent placés dans des sites peu encombrés pour effectuer leurs tâches. Ainsi, le sujet le plus largement étudié dans ce cadre est le problème de faisabilité des trajectoires. Il s'agit alors de vérifier qu'une trajectoire ne passe pas par une configuration singulière et de donner, le cas échéant, une trajectoire qui se rapproche de la trajectoire initiale évitant les configurations singulières.

Les robots parallèles sont de plus en plus répandus, que ce soit dans l'industrie, dans le monde médical,… aussi ils doivent être intégrés fréquemment dans des milieux dynamiques. Les robots parallèles, tels que les robots à câbles et les treillis articulaires, qui disposent d'un grand espace de travail et qui sont utilisés pour la manutention dans les usines sont en augmentation constante. Ces raisons nous ont conduit à étudier la planification de trajectoire pour les robots parallèles tout en prenant en considération à la fois les singularités et les collisions. Ces collisions peuvent être internes, c'est-à-dire, entre les différentes pièces du robot. Elles peuvent aussi être externes, entre le robot et les différents objets qui se trouvent dans l'espace de travail. Ces objets seront appelés obstacles.

Pour faire les simulations et pour trouver les trajectoires, nous avons besoin d'un outil qui intègre la géométrie du robot ainsi que la géométrie des objets qui se trouvent dans l'espace de travail afin de pouvoir détecter les collisions. Cet outil doit aussi être capable de gérer les

chaînes cinématiques fermées des robots parallèles, ce qui est compliqué étant donné la diversité et le nombre d'architectures de robots parallèles qui sont proposées. Les logiciels de Conception Assistée par Ordinateur (CAO) classiques peuvent gérer des mécanismes à chaînes fermées, mais ne donnent pas d'information sur la façon dont sont régis les paramètres de ces chaînes. Nous avons donc choisi de nous appuyer sur un logiciel de CAO Robotique : le Système de Modélisation et d'Animation des Robots (SMAR). Ce logiciel, développé au Laboratoire de Mécanique des Solides de Poitiers (LMS), permet de modéliser et d'animer les robots sériels. Il dispose d'un modeleur géométrique simple et d'un module de calcul de distance, ce qui permet d'approximer les formes géométriques des robots et des objets. Le module de calcul de distance permet aussi de détecter les collisions. Afin d'intégrer un grand nombre d'architectures de robots parallèles, nous avons modifié l'architecture de modélisation et nous avons rajouté un module permettant de générer des équations de fermeture de chaînes en fonction de la modélisation du robot. Ces équations permettent de donner un mouvement compatible avec les chaînes du robot et permettent aussi de détecter les singularités.

Les aspects traités dans ce travail concernent les points suivants : la modélisation des robots parallèles, les configurations singulières du robot, la détection des collisions internes entre les composants du robot, ainsi que les collisions entre le robot et les objets de l'espace de travail. Une méthode de planification de trajectoire intégrant ces différents points devra ainsi être mise en oeuvre.

Dans le premier chapitre, une étude générale des robots parallèles est effectuée à travers une présentation non exhaustive des différentes architectures existantes. Nous rappelons les définitions des modèles géométriques direct et inverse, des modes d'assemblages et des singularités. Les singularités sont également étudiées au cours de cette présentation, avant d'aborder les méthodes de planification de trajectoires sans collision utilisées pour les robots sériels. A l'issue de cette étude, un état de l'art relatif à la planification sans collision des robots parallèles est fait.

Avec le second chapitre, nous abordons l'outil développé pour modéliser et simuler le mouvement des robots parallèles. L'utilisation de l'outil pour la génération des équations cinématiques régissant le mouvement des robots est illustrée. La méthode d'analyse de ces équations est détaillée, ainsi que leur utilisation pour générer le mouvement des robots et détecter les singularités.

Le troisième chapitre est ensuite consacré à l'adaptation des méthodes de planification de trajectoire existantes aux robots parallèles. Nous introduisons une nouvelle méthode qui prend en considération les collisions (internes des pièces des robots et externes du robot avec les objets de l'espace de travail), ainsi que l'évitement des configurations singulières.

Au sein du quatrième chapitre, nous illustrons les approches développées dans ce mémoire par des résultats sur plusieurs dispositifs expérimentaux. Le premier exemple concerne la mise en œuvre de notre approche avec un robot translateur à trois axes linéaires : le RAF (développement conjoint entre le Laboratoire de Génie Mécanique de Monastir et le LMS). Le second exemple aborde le pilotage d'un robot à quatre câbles du Laboratory of Robotics and Mechatronics (LARM) de Cassino en Italie. Le troisième et dernier exemple, montre comment cette approche a pu être étendue au problème de la génération de mouvements de marche d'un robot à quatre pattes de type chien en le considérant comme un robot parallèle.

Enfin nous terminons ce mémoire par une conclusion générale où un bilan des travaux effectués est dressé. Des perspectives possibles relatives aux approches développées sont dégagées pour faire suite à cette étude.

Chapitre I

Etat de l'art

Dans ce chapitre, nous commençons tout d'abord par présenter les robots parallèles en énonçant quelques définitions et nous donnons une liste non exhaustive des différentes architectures qu'on trouve dans la littérature. Nous introduisons des notions nécessaires pour l'analyse et la manipulation des robots parallèles. Puis nous présentons les méthodes de planification de trajectoire les plus courantes.

Introduction

Notre objectif est d'étudier la planification de trajectoire des robots parallèles tout en évitant les singularités et les collisions.

Nous nous focalisons sur certains types de robots parallèles. Soit à cause de leur grand espace de travail, soit à cause de leurs positions singulières facilement détectables, soit parce que nous disposons de ces robots au LMS et nous pouvons les utiliser pour faire des essais expérimentaux. Les robots ayant un grand espace de travail peuvent être utilisés afin de tester la planification de trajectoire sans collision. Alors qu'on peut utiliser les robots ayant des singularités facilement détectables, pour vérifier et valider l'évitement des singularités.

Après avoir présenté les robots parallèles, nous poserons quelques problèmes relatifs à ces robots et nous insisterons sur le problème de singularités tout en donnant un aperçu sur les travaux qui ont déjà été faits à ce sujet.

Par la suite nous présenterons un état de l'art sur la planification de trajectoire d'une manière générale et nous recenserons ce qui a été fait dans le cadre des robots parallèles en particuliers.

1 Définitions

Un mécanisme est un assemblage de pièces conçu afin de convertir des mouvements et/ou des efforts.

Un robot est un mécanisme piloté par une unité de commande qui réalise automatiquement des tâches telles que la manipulation ou l'usinage.

Une chaîne cinématique est un ensemble de pièces liées par des liaisons mécaniques. On appelle chaîne cinématique simple toute chaîne cinématique où chaque pièce a une ou deux

liaisons seulement. On appelle chaîne cinématique complexe toute chaîne cinématique contenant une pièce (ou plusieurs pièces) ayant trois liaisons ou plus.

Un robot sériel est un robot formé par une chaîne cinématique simple contenant une base fixe et un organe terminal, qui ont chacun une seule liaison. Un robot parallèle est un robot en chaîne cinématique complexe, constitué d'un organe terminal et d'une base fixe reliés entre eux par des chaînes cinématiques indépendantes [Merlet97].

La figure I.2 montre la modélisation CAO d'un robot sériel de type KUKA et un robot parallèle le RAF fabriqué au LMS.

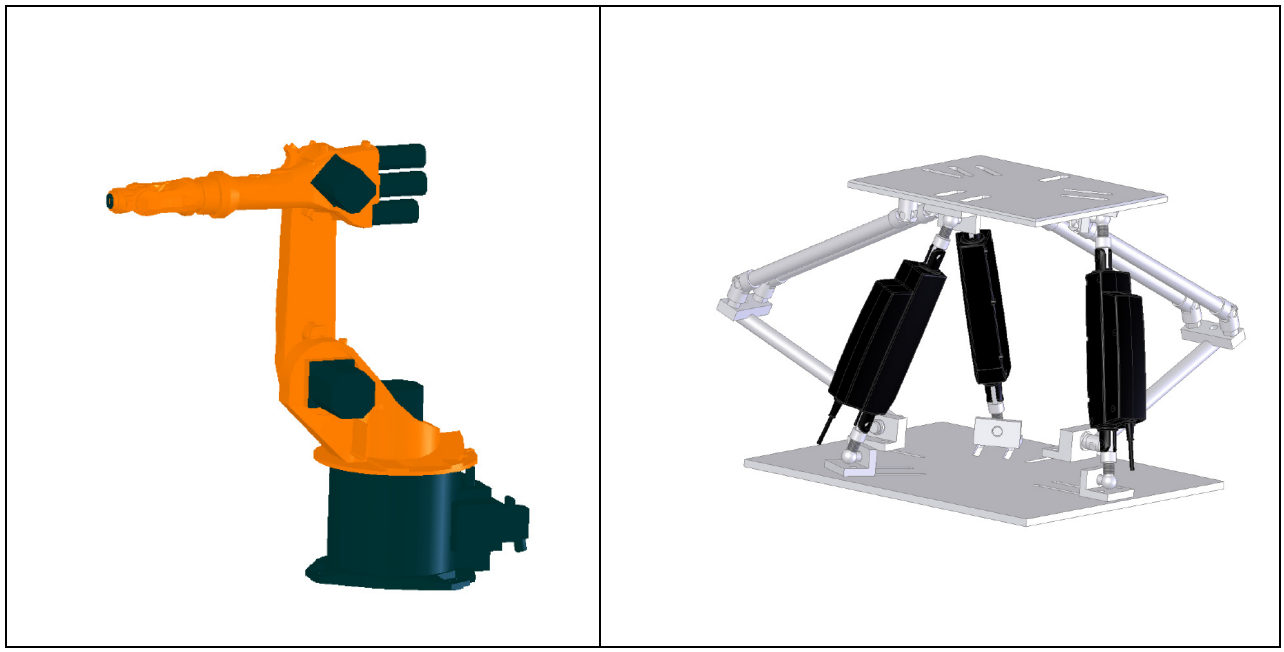


Figure I.2. Modèle CAO d'un robot sériel KUKA et un robot parallèle le RAF

Un manipulateur est un robot utilisé pour la manipulation. Dans la littérature, les robots sériels et les robots parallèles sont considérés comme étant des robots manipulateurs par opposition aux robots mobiles. On note aussi qu'il existe des robots manipulateurs mobiles. Dans ce mémoire on utilise les termes manipulateurs parallèles ou mécanismes parallèles ou encore robots parallèles pour décrire la même structure.

On appelle degrés de liberté d'un système mécanique (qu'on note ddl) les mouvements indépendants qui restent possibles. Le ddl d'une liaison mécanique entre deux pièces correspond au nombre de translations et de rotations indépendantes entre ces deux pièces. Ainsi une liaison pivot et une liaison glissière ont chacune un seul ddl, alors qu'une liaison pivot glissant a deux ddl et une liaison rotule en a trois.

Le ddl d'un mécanisme, appelé aussi degré de mobilité, correspond au nombre de mouvements indépendants possibles dans le mécanisme.

Dans un robot parallèle, on appelle jambe, une chaîne cinématique indépendante reliant l'organe terminal à la base fixe.

On dit qu'un robot est pleinement parallèle s'il a autant de jambes que l'organe terminal présente de ddl. Chaque jambe comporte au plus un actionneur. Aucune pièce d'une jambe ne doit avoir plus de deux liaisons [Pierrot 91].

En robotique, on utilise souvent l'appellation articulation pour une liaison mécanique entre deux pièces. Une liaison pivot est appelée articulation rotoïde (qu'on note R). Une liaison glissière est appelée articulation prismatique (P) et une liaison rotule est appelée articulation sphérique (S). On appelle aussi articulation universelle (U), la liaison correspondant à un joint de cardan, c'est-à-dire, la liaison composée de deux liaisons pivots d'axes perpendiculaires.

Dans le cas des robots parallèles, les articulations ne sont pas toutes motorisées, contrairement au cas des robots sériels. Les articulations motorisées sont appelées articulations actives, alors que les articulations non motorisées sont appelées articulations passives.

Les coordonnées opérationnelles sont utilisées pour repérer la position et l'orientation de l'organe terminal d'un manipulateur.

Les variables d'un mécanisme sont l'ensemble des articulations actives, l'ensemble des articulations passives et l'ensemble des coordonnées opérationnelles.

On appelle mécanisme plan, un mécanisme ayant des mouvements plans. Les mouvements plans sont des mouvements qui s'effectuent dans un plan, ils se décomposent en deux translations d'axes parallèles au plan et en une rotation d'axe perpendiculaire au plan.

Un mécanisme qui n'est pas plan est dit spatial.

2 Les robots parallèles : types et classification

Dans cette étude, nous présentons quelques types de robots parallèles intéressants pour étudier la planification de trajectoire, tout en évitant à la fois les collisions et les singularités. Une étude relativement complète des différentes architectures des robots parallèles peut être trouvée dans [Merlet 97].

Nous distinguons deux catégories de robots parallèles : les robots plans et les robots spatiaux. Puis, dans ces deux catégories nous classons les robots parallèles selon leur nombre de ddl.

Les robots à câbles sont des robots parallèles où les jambes sont remplacées par des câbles. Ces robots peuvent être plans ou spatiaux, nous avons choisi de les classer dans une catégorie particulière pour les distinguer des autres robots parallèles.

Nous remarquons que dans le cas des robots pleinement parallèles, le nombre de ddl ne peut excéder trois pour les robots plans et ne peut excéder six pour les robots spatiaux.

2.1 Les robots plans

Nous ne considérons dans ce mémoire que le cas des robots plans pleinement parallèles. Comme il y a autant de ddl de l'organe terminal que de jambes dans un robot pleinement parallèle et comme il n'y a qu'un seul actionneur par jambe, les robots pleinement parallèles plans sont soit à deux ddl soit à trois ddl.

2.1.1 Robots plans à 2ddl

Dans le cas où l'on ne considère que les robots constitués d'articulations de types rotoïdes et prismatiques, McCloy [McCloy 98] a démontré qu'il pouvait y avoir vingt architectures différentes. Ce nombre se réduit à six si l'on suppose que les actionneurs sont attachés au sol, qu'il n'y a pas d'articulation prismatique passive et qu'aucun actionneur ne supporte le poids d'un autre actionneur. Ces architectures sont montrées sur la figure I.3. Les articulations rotoïdes actives sont représentées par des cercles blancs à l'intérieur tandis que les articulations rotoïdes passives sont représentées par des cercles noirs pleins.

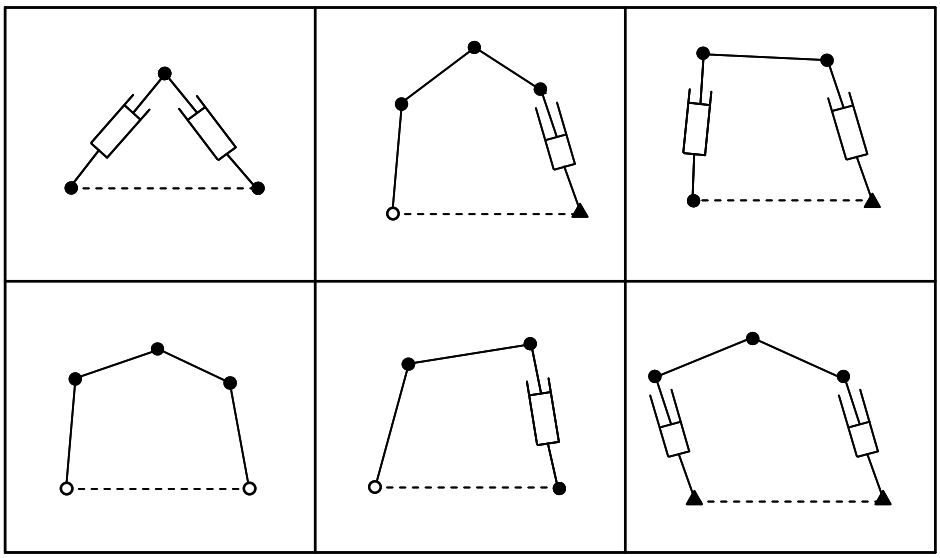


Figure I.3. Robots plans à 2 ddl

2.1.2 Robots plans à 3 ddl

Hayes et Husty [Hayes 03] ont étudié les plateformes généralisées planes à trois chaînes. Ces robots sont composés d'une plateforme mobile reliée par trois chaînes cinématiques. Chaque chaîne contient trois articulations qui peuvent être, soit de type rotoïde (R), soit de type prismatique (P). Ainsi, les types de chaînes possibles sont les séquences suivantes : RRR, RPR, RRP, RPP, PRR, PPR, PRP. Les chaînes de type PPP sont à exclure puisqu'elles imposent une orientation particulière à la plateforme mobile. En d'autres termes, aucune combinaison de translations dans le plan ne permet d'avoir un changement de l'orientation.

Il y a alors sept types de chaînes utilisés. Dans chaque chaîne il n'y a qu'une seule liaison actionnée. Le tableau I.1 montre les 18 chaînes possibles. Les liaisons actionnées sont soulignées. Les chaînes passives sont de type RR, PR, RP, ou bien PP. Les chaînes passives de type PP sont à rejeter, puisqu'une plateforme ayant une chaîne pareille soit n'est pas contrôlable, soit elle est difficilement assemblable quand les liaisons actives sont spécifiées [Hayes 04]. Parmi ces 18 chaînes, il y a 8 paires de chaînes symétriques. Ce qui amène certains auteurs à enlever les chaînes marquées ~ dans le tableau 1, lors de l'étude des différentes architectures possibles [Bonev 03]. Ceci rend le nombre de chaînes à étudier égal à 10. Si on considère que le sens de montage des chaînes entre la base et l'organe terminal est important nous considérons les 18 chaînes pour calculer le nombre total de plateformes possibles. Ce nombre correspond à la somme du cas où toutes les chaînes sont identiques plus le cas où il y a deux chaînes identiques plus le cas où il y a trois chaînes différentes :

$$C_{18}^{1} + 2 \times C_{18}^{2} + C_{18}^{3} = 1140 \qquad \text{(I.1)}$$

Table I.1. Structures possibles des jambes d'un robot plan à 3 ddl

Chaîne passive RR	**Chaîne passive PR**	**Chaîne passive RP**
$\underline{R}RR$	$\underline{R}PR$	$\underline{R}RP$
$R\underline{R}R$	$P\underline{R}R$	$R\underline{R}P^{\sim}$
$RR\underline{R}^{\sim}$	$PR\underline{R}^{\sim}$	$RP\underline{R}^{\sim}$
$\underline{P}RR$	$\underline{P}PR$	$\underline{P}RP$
$R\underline{P}R$	$P\underline{P}R^{\sim}$	$R\underline{P}P$
$RR\underline{P}^{\sim}$	$PR\underline{P}^{\sim}$	$RP\underline{P}^{\sim}$

On montre dans la figure I.4, deux exemples de robots plans à trois ddl ayant trois chaînes identiques.

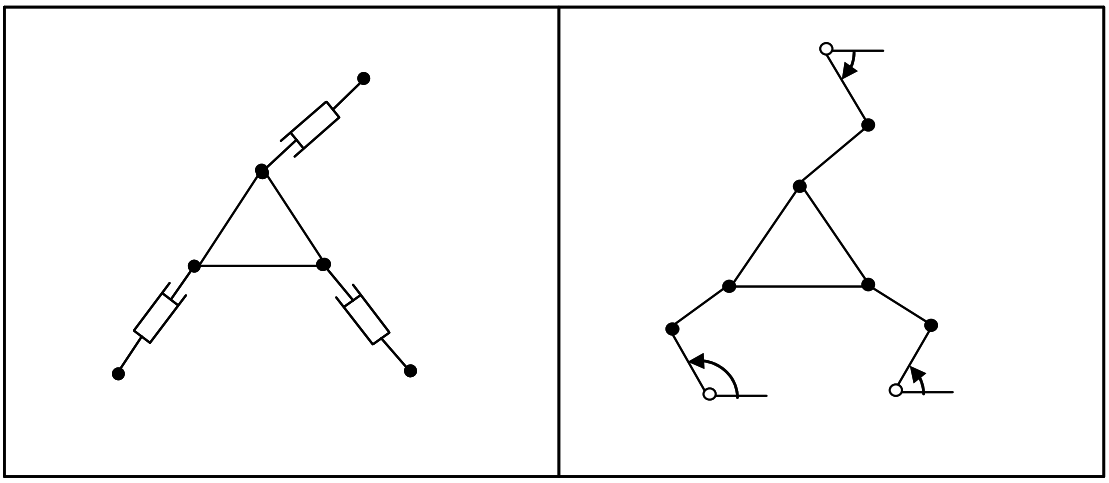


Figure I.4. Exemples de robots plans à 3 ddl le 3-RPR et le 3-RRR

2.2 Les robots à mouvements spatiaux

Les robots à mouvements spatiaux se décomposent principalement en robots à trois ddl et en robots à six ddl. Il existe évidemment des robots à quatre ddl et des robots à cinq ddl. Ici l'objectif étant de montrer les robots parallèles les plus pertinents, voir [Merlet 97] pour plus de détails.

Il existe aussi des robots à mouvements spatiaux qui ont plus de 6 ddl, ces robots ne sont évidemment pas des robots pleinement parallèles.

2.2.1 Robots à 3 ddl

Les robots parallèles à trois ddl se décomposent principalement en robots translateur et en robots pour la rotation. Il existe aussi des robots qui présentent à la fois des rotations et des translations comme c'est le cas du manipulateur de Lee par exemple [Lee 88] (voir figure I.5). Ce manipulateur permet une translation sur l'axe vertical et deux rotations : une de roulis et une de tangage.

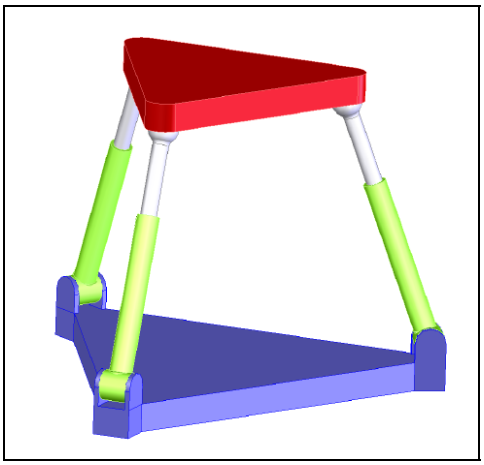


Figure I.5. Manipulateur de Lee

2.2.1.1 Robots translateurs

Ce sont des robots ayant trois ddl de translation. Ils présentent un intérêt important pour les opérations de transfert. Ces robots sont largement utilisés dans les applications de conditionnement dans la mesure où dans ces applications, les trois translations sont suffisantes. Un mouvement de rotation selon l'axe z est souvent rajouté à l'organe terminal.

Le robot delta (figure I.6) a été proposé par Clavel [Clavel 88]. Ce robot utilise des actionneurs rotatifs. Il est basé sur une propriété intéressante des parallélogrammes pour ne garder que des translations de la plateforme mobile.

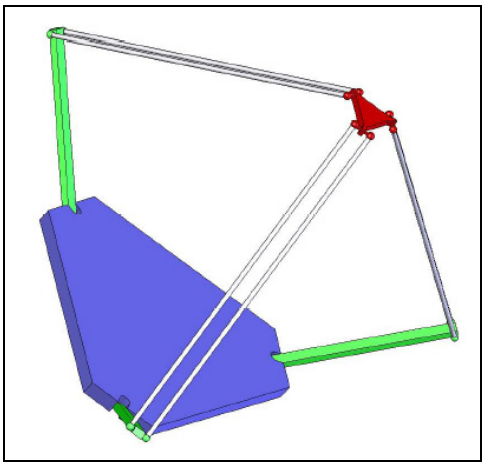


Figure I.6. Le robot delta

Le robot 3UPU (figure I.7) est constitué de trois chaînes composées de deux joints de cardans reliés par un actionneur prismatique. Un choix judicieux des axes des joints de cardan peut empêcher la rotation de la plateforme mobile [Tsai 96].

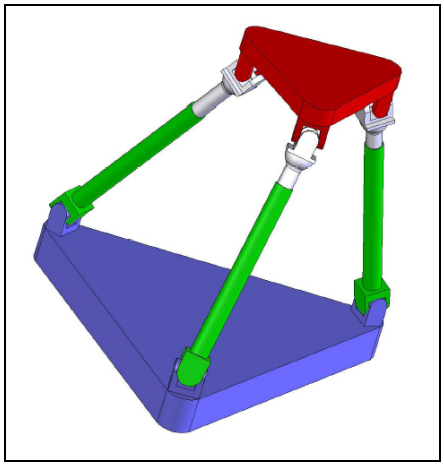


Figure I.7. Le 3-UPU

Romdhane [Romdhane 99] a proposé un translateur RAF (voir figure I.8). Ce translateur comporte trois jambes SPS et deux chaînes passives qui éliminent les rotations. Une jambe SPS est composée de deux liaisons sphériques reliées par un actionneur prismatique.

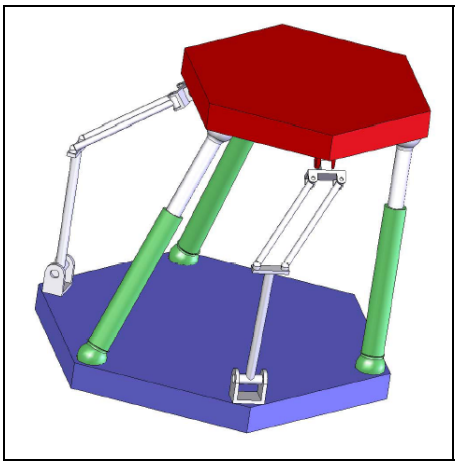


Figure I.8. Le RAF

2.2.1.2 Robots pour rotation

Ces manipulateurs sont une alternative intéressante aux poignets utilisés dans les robots sériels, ils permettent les trois rotations autour d'un point. La figure I.9 montre le poignet de Hayward [Hayward 89]. Ce robot se compose de 4 actionneurs prismatiques qui permettent d'orienter un plateau monté sur rotule.

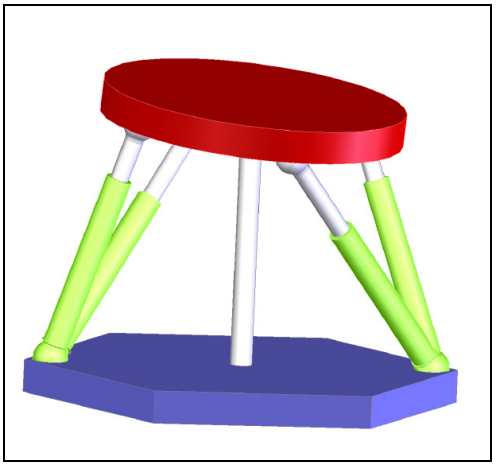


Figure I.9. Le poignet de Hayward

Le poignet sphérique de Gosselin [Gosselin 88] est montré dans la figure10. Ce poignet comporte trois chaînes à actionneurs rotatifs dont les axes sont concourants au centre du plateau mobile.

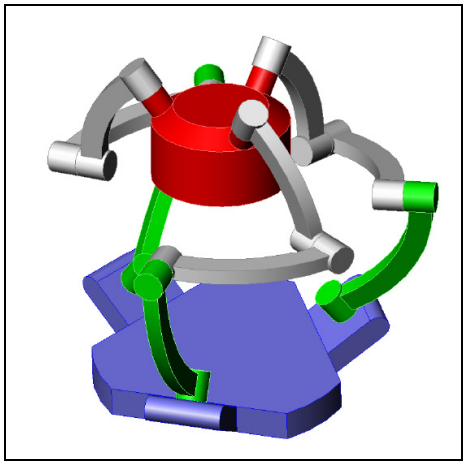


Figure I.10. Poignet de Gosselin

2.2.2 Robots à six ddl

La plateforme de Stewart (figure I.11) permet de déplacer et d'orienter la plateforme mobile à l'aide de six vérins. Cette structure est très utilisée dans les simulateurs de vols et dans les cinémas dynamiques. Les avantages de cette structure sont les bonnes propriétés dynamiques et les inconvénients sont les positions singulières fréquentes et l'espace de travail réduit.

Dans la littérature, ce qui est communément appelé plateforme de Stewart avait été proposé en réalité par Gough en 1956 [Gough 56] afin de tester les pneus. Stewart [Stewart 65] a proposé une autre architecture en 1965 pour un simulateur de vol et cela sans mentionner la plateforme de Gough. Un des correcteurs de l'article de Stewart a fait remarquer la ressemblance typologique avec la plateforme de Gough et depuis, paradoxalement la plateforme de Gough est devenue populaire sous le nom de plateforme de Stewart.

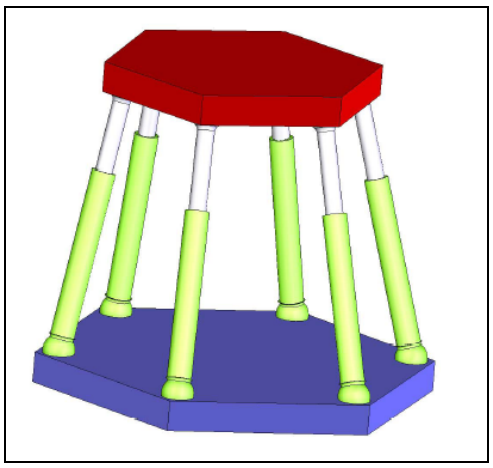


Figure I.11. Plateforme de Stewart

Il existe beaucoup de variantes possibles de la plateforme de Stewart. La figure I.12 montre un cas particulier où les points d'attache sur la plateforme mobile sont confondus deux à deux. Ce robot est appelé la plateforme 6-3 car il y a six points d'attache sur la base et trois sur la plateforme mobile.

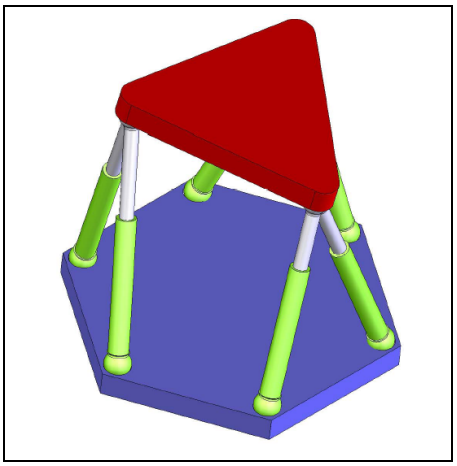


Figure I.12. Plateforme 6-3

Gao et al. [Gao 06] ont défini la plateforme de Stewart généralisée, comme étant deux pièces rigides liés par six contraintes de distances et/ou angulaire, entre des points, des liaisons et/ou des plans appartenant à la base et à la plateforme mobile respectivement. Ils ont montré qu'il y a 3850 formes possibles de plateforme de Stewart généralisée. Ils donnent une borne supérieure au nombre de solutions du modèle géométrique direct de toutes les plateformes de Stewart généralisés.

Lallemand [Lallemand 97] a proposé un robot à 6 ddl composé de deux robots delta (figure I.13). Un premier robot delta permet le positionnement, le deuxième permet l'orientation de l'organe terminal. Ce robot est intéressant dans la mesure où il permet de découpler la position et l'orientation.

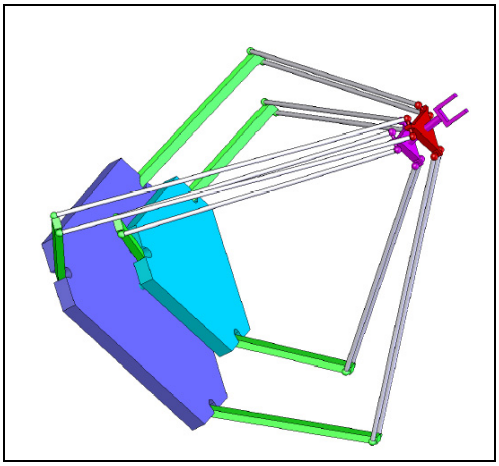


Figure I.13. Le robot 2 delta

2.2.3 Robots à plus de 6 ddl

Parmi les robots parallèles ayants plus de six ddl, nous nous intéressons au Logabex [Charentus 90]. Ce robot est composé de 4 plateformes de Stewart (voir figure I.14). Le Logabex dispose de 24 ddl, il est utilisé pour les interventions dans les sites nucléaires.

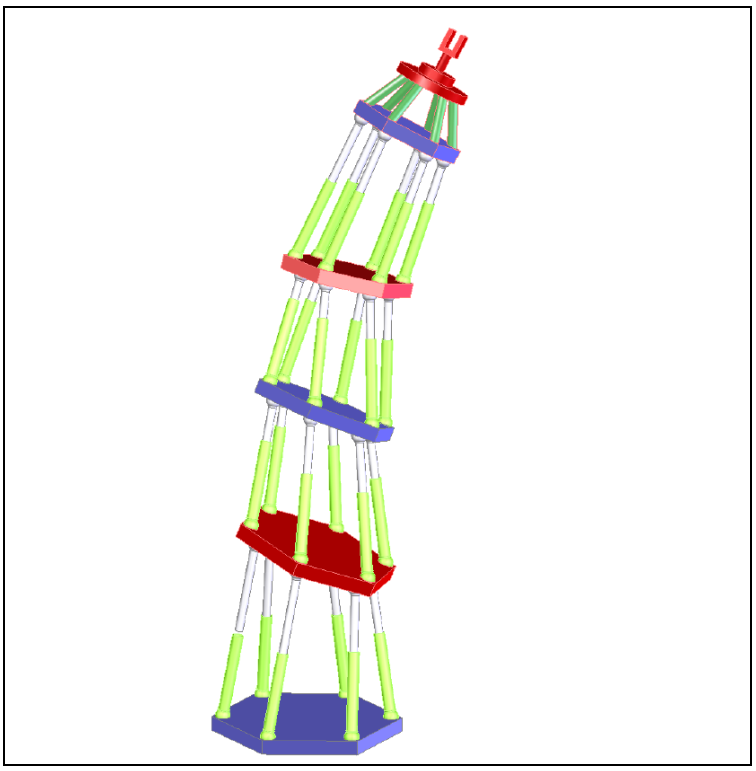


Figure I.14. Le Logabex

2.3 Les robots à câbles

Les robots à câbles ont une structure comparable à celle des robots parallèles. Ils possèdent des caractéristiques intéressantes, par exemple, un grand espace de travail en comparaison avec les manipulateurs parallèles classiques. En plus de cela les robots à câbles ont une structure légère qui présente de bonnes propriétés dynamiques, un rapport charge utile/masse élevé et un prix de construction réduit. Cependant les tâches réalisables par les robots à câbles sont limitées. Cela vient du fait que les câbles ne travaillent qu'en traction. En effet les câbles ne peuvent que tirer la plateforme mobile et ne peuvent pas la repousser.

La collision des câbles avec les objets qui peuvent se trouver dans l'espace de travail est très dangereux et donc à éviter, d'où la nécessité de la planification de trajectoire sans collision pour ce type de robots.

Les robots à câbles peuvent être soit totalement contraints soit sous contraints. Dans le cas où ils sont totalement contraints, la position et l'orientation de la plateforme mobile sont définies par les longueurs des câbles, alors que dans le cas sous contraint, la gravité intervient pour définir la position et l'orientation.

On appelle m le nombre de câbles et n le nombre de ddl de la plateforme mobile. On peut dire qu'un robot à câble est totalement contraint si l'équation (2) est vérifiée.

$$m \geq n+1 \tag{I.2}$$

Dans la littérature on trouve plusieurs travaux qui traitent des robots à câbles voir par exemple [Oh 03], [Bouchard 06] et [Stump 06].

Un robot à quatre câbles a été développé au Laboratorio di Robotica e Meccatronica (LARM) de Cassino en Italie [Ottaviano02]. Ce robot peut être utilisé dans le plan et dans ce cas il est totalement contraint (voir figure I.15).

La figure I.16 montre le cas spatial où le robot est sous contraint.

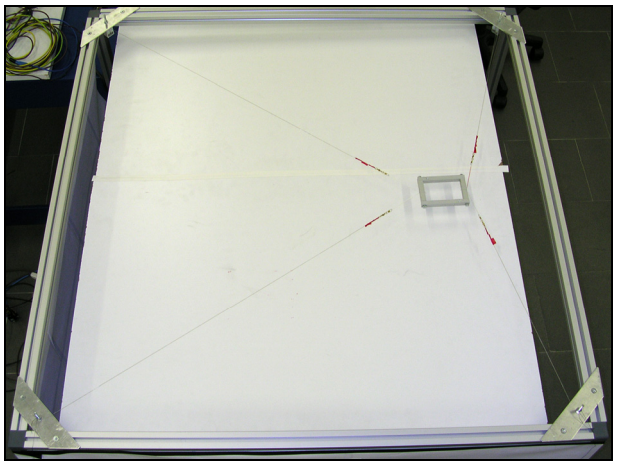


Figure I.15. Cas plan du robot à 4 câbles du LARM

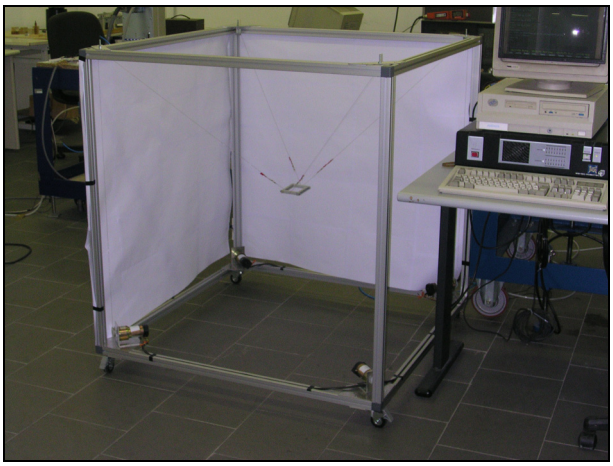


Figure I.16. Cas spatial du robot à 4 câbles du LARM

3 Modèle géométrique

Le modèle géométrique d'un robot, consiste à écrire les relations entre les variables articulaires actives et la configuration de l'organe terminal.

On appelle modèle géométrique direct, le modèle qui permet de donner la position et l'orientation de l'organe terminal en fonction des variables articulaires actives. Tandis que, le modèle géométrique inverse permet d'écrire les variables articulaires actives en fonction des coordonnées opérationnelles.

Le modèle géométrique direct est simple à obtenir dans le cas des robots sériels. Il existe pour le cas des robots sériels, des techniques qui permettent de donner directement le modèle géométrique direct, comme par exemple le paramétrage de Denavit-Hartenberg [Denavit 55]. Le modèle géométrique inverse, n'est pas simple dans le cas des robots sériels, il est trouvé soit analytiquement au cas par cas, soit numériquement. Alors que, dans le cas des robots parallèles, le modèle géométrique inverse est plus simple à trouver. On parle même de dualité entre les robots parallèles et les robots sériels.

3.1 Cas des robots sériels

Pour les robots sériels, le modèle géométrique direct est unique et simple à obtenir. On prend comme exemple un robot plan 2R montré dans la figure I.17.

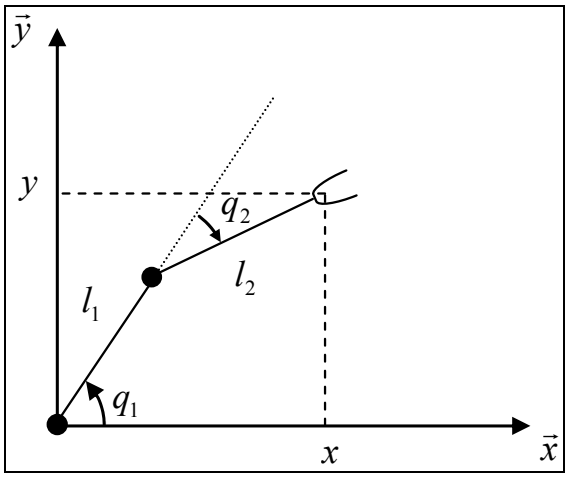


Figure I.17. Robot sériel plan 2R

On écrit le modèle géométrique direct du robot 2R de la manière suivante :

$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \tag{I.3}$$

$$y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \tag{I.4}$$

Le modèle géométrique inverse consiste à trouver q_1 et q_2 en fonction de x et y. Ce problème peut avoir jusqu'à deux solutions, on parle dans ce cas de coude haut et de coude bas (voir figure I.18).

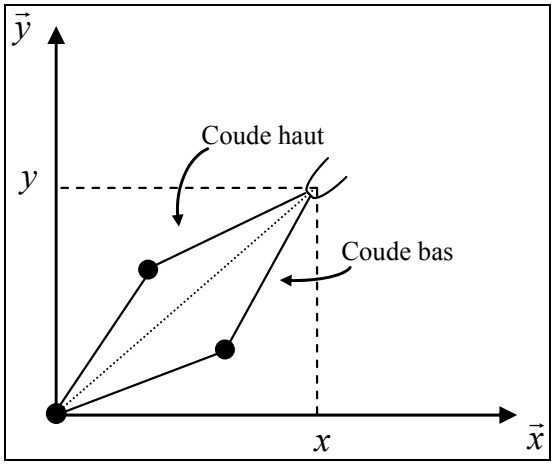


Figure I.18. Configuration coude haut et coude bas

La figure I.19, montre l'ensemble des positions pour lesquelles au moins une solution du modèle géométrique inverse existe. Cet ensemble est appelé espace de travail, il correspond pour le cas du robot 2R à l'équation (I.5).

$$x^2 + y^2 \leq (l_1 + l_2)^2 \text{ ou } x^2 + y^2 \geq (l_1 - l_2)^2 \tag{I.5}$$

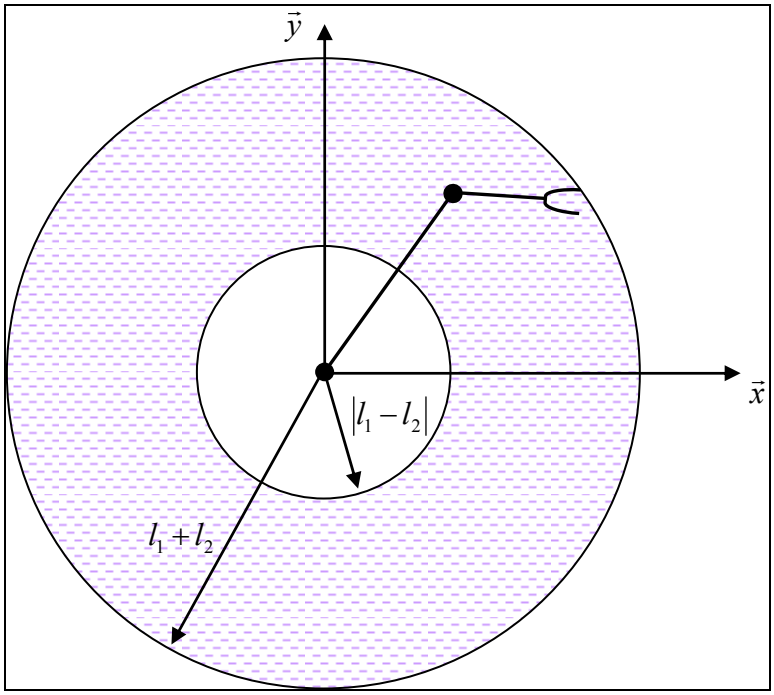


Figure I.19. Espace de travail du robot 2R

Dans ce cas, pour obtenir le modèle géométrique inverse, on commence par éliminer q_1 dans les équations (I.3) et (I.4), en écrivant $(3)^2 + (4)^2$, on obtient l'équation (I.6).

$$\cos q_2 = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \tag{I.6}$$

On calcule α selon l'équation (I.7).

$$\alpha = \arccos\left[\frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2}\right] \tag{I.7}$$

Si le terme entre crochets est compris entre -1 et +1, c'est-à-dire si l'organe terminal est dans l'espace de travail du robot, on obtient les solutions suivantes pour q_2

$$q_2 = \pm\alpha \tag{I.8}$$

En développant les équations (I.3) et (I.4), et en résolvant le système par rapport aux deux inconnues $\cos q_1$ et $\sin q_1$, on obtient :

$$q_1 = \mathrm{ATAN2}\left(y\left(l_1 + l_2\cos q_2\right) - xl_2\sin q_2, x\left(l_1 + l_2\cos q_2\right) + yl_2\sin q_2\right) \tag{I.9}$$

3.2 Cas des robots parallèles

Pour les robots parallèles, le modèle géométrique inverse est, dans la plupart des cas, unique et simple. Il est obtenu plus facilement que dans le cas des robots sériels.

On prend comme exemple, le robot 3-RPR de la figure I.20.

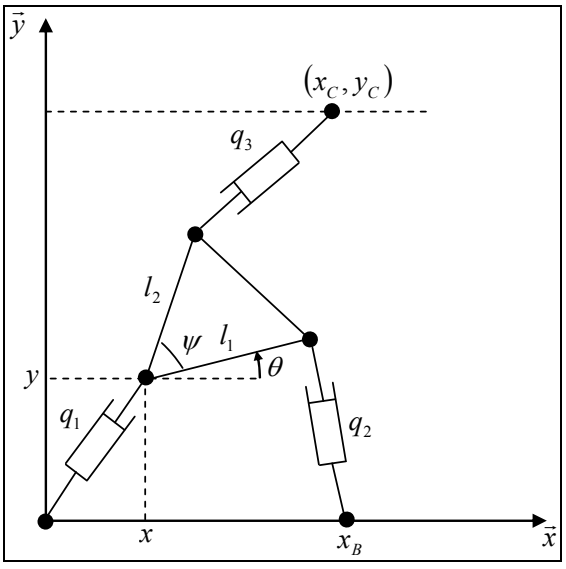


Figure I.20 Robot 3-RPR

On obtient les équations suivantes pour le robot 3-RPR :

$$q_1 = \sqrt{x^2 + y^2} \tag{I.10}$$

$$q_2 = \sqrt{\left(x + l_1 \cos\theta - x_B\right)^2 + \left(y + l_1 \sin\theta\right)^2} \tag{I.11}$$

$$q_3 = \sqrt{\left(x + l_2 \cos(\theta + \psi) - x_C\right)^2 + \left(y + l_2 \sin(\theta + \psi) - y_c\right)^2} \tag{I.12}$$

Il existe certains cas où le modèle géométrique inverse n'est pas unique comme c'est le cas du 3-RRR (voir figure I.21).

Pour trouver le modèle géométrique inverse, il suffit de considérer chaque jambe comme étant un robot sériel de type 2R, ce qui donne deux solutions pour chaque jambe. Nous avons en tout 8 solutions du modèle géométrique inverse du 3-RRR.

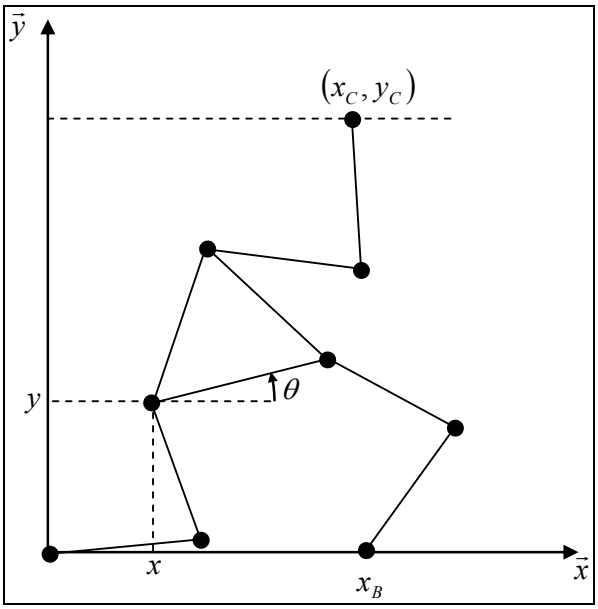


Figure I.21. Robot 3-RRR

Le modèle géométrique direct n'est pas unique dans la plupart des cas des robots parallèles, il dépend souvent de la manière dont il a été monté. Pour cette raison, on parle de modes d'assemblage. On donne un exemple simple d'un robot plan, le 2-RPR qu'on montre dans la figure I.22.

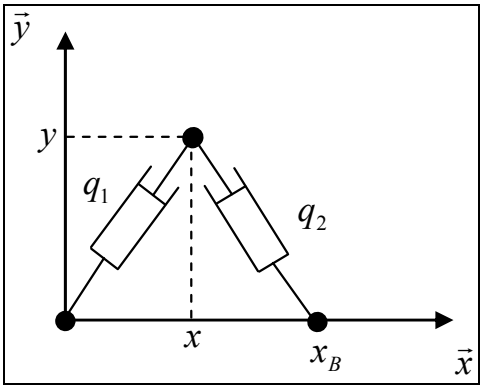


Figure I.22. Robot 2-RPR

Les solutions du modèle géométrique direct du 2-RPR, correspondent aux points d'intersection des cercles de centres O et B et de rayons respectifs q_1 et q_2. Ces points sont les solutions du système suivant :

$$x^2 + y^2 = q_1^2 \tag{I.13}$$

$$(x - x_B)^2 + y^2 = q_2^2 \tag{I.14}$$

Le modèle géométrique direct admet deux solutions dans ce cas. On dit qu'il y a deux modes d'assemblage.

Dans le cas du 3-RPR, il existe 4 modes d'assemblage d'après [Hayes 04].

Pour la plateforme de Stewart, Ronga [Ronga92] a démontré que le nombre de modes d'assemblage est borné par 40, alors que Dietmaier [Dietmaier 98] a utilisé un algorithme astucieux pour trouver une plateforme de Stewart ayant 40 modes d'assemblage.

4 Modèle cinématique

Le modèle cinématique complète le modèle géométrique, en écrivant les relations entre les vitesses des variables articulaires et les vitesses de l'organe terminal. La propriété intéressante du modèle cinématique est sa linéarité par rapport aux vitesses. Ainsi il est plus simple à manipuler que le modèle géométrique et permet d'utiliser des propriétés qui découlent de la résolution des systèmes linéaires.

4.1 Cas des robots sériels

Le modèle cinématique direct est simple à trouver dans le cas des robots sériels [Lallemand 94]. La matrice jacobienne est obtenue en écrivant les torseurs cinématiques correspondants aux liaisons entre les différentes pièces du robot.

On appelle $\vec{u}_i$ le vecteur unitaire de l'axe de la liaison entre les solides S_{i-1} et S_i (figure I.23).

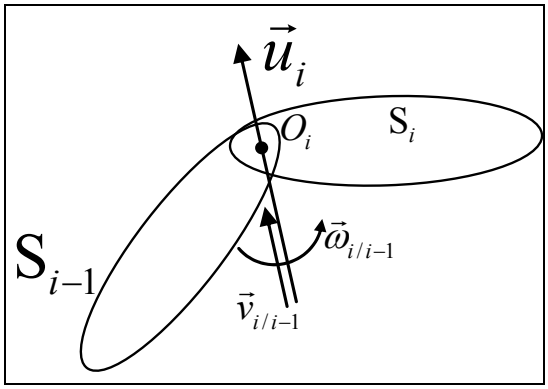


Figure I.23. Liaison entre deux solides.

On note σ_i une variable booléenne valant 1 si la liaison est une translation et 0 si c'est une rotation. On note $\overline{\sigma}_i = 1 - \sigma_i$ et on appelle q_i le paramètre articulaire de la liaison. Le mouvement de S_i/S_{i-1} est soit une rotation de vecteur vitesse $\vec{\omega}_{i/i-1} = \dot{q}_i \vec{u}_i$, soit une translation de vecteur vitesse $\vec{v}_{O_{i/i-1}} = \dot{q}_i \vec{u}_i$. Le torseur cinématique du mouvement de S_i/S_{i-1}, exprimé en un point P quelconque, s'écrit :

$$C\left(P \in S_i / S_{i-1}\right) = \begin{cases} \vec{\omega}_{i/i-1} = \overline{\sigma}_i \dot{q}_i \vec{u}_i \\ \vec{v}_{P \in S_i / S_{i-1}} = \vec{v}_{O_{i/i-1}} + \vec{\omega}_{i/i-1} \wedge \overrightarrow{O_i P} = \sigma_i \dot{q}_i \vec{u}_i + \overline{\sigma}_i \dot{q}_i \vec{u}_{iP} \end{cases} \quad \text{(I.15)}$$

où

$$\vec{u}_{iP} = \vec{u}_i \wedge \overrightarrow{O_i P} \quad \text{(I.16)}$$

Le modèle cinématique des robots sériels s'écrit sous la forme suivante :

$$\dot{\mathbf{X}} = \mathbf{J}\dot{\mathbf{q}} \tag{I.17}$$

On prend comme exemple un robot sériel plan 2RP montré dans la figure I.24.

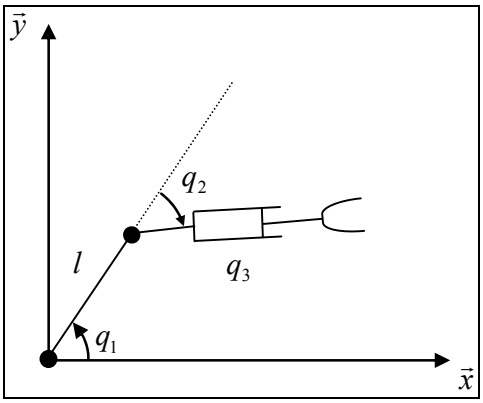


Figure I.24. Robot 2RP plan

Le modèle cinématique du robot plan 2RP s'écrit en écrivant les torseurs cinématiques des liaisons :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} -ls_1 - q_3 s_{12} & -q_3 s_{12} & c_{12} \\ lc_1 + q_3 c_{12} & q_3 c_{12} & s_{12} \\ 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} \tag{I.18}$$

Où

$$s_i = \sin q_i \; ; c_i = \cos q_i \; ; s_{ij} = \sin(q_i + q_j); c_{ij} = \cos(q_i + q_j) \tag{I.19}$$

4.2 Cas des robots parallèles

Dans la plupart des cas, des robots parallèles, le modèle cinématique inverse est facile à obtenir. Une ligne de la jacobienne inverse cinématique, dépend d'une jambe du robot parallèle. La figure I.25 montre une plateforme mobile liée à la plateforme fixe par l'intermédiaire d'une jambe A_iB_i.

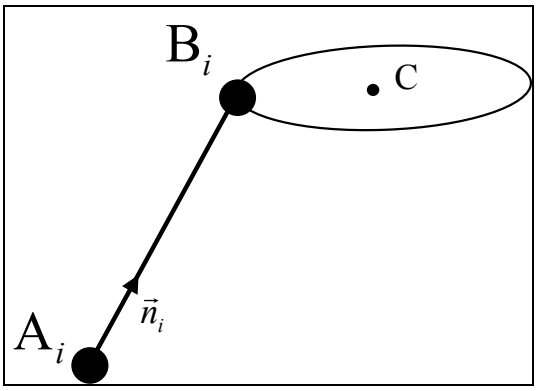


Figure I.25. Jambe A_iB_i *d'un robot parallèle*

On écrit :

$$\overrightarrow{A_iB_i} = q_i\vec{n}_i \tag{I.20}$$

Où $\vec{n}_i$ est un vecteur unitaire défini de la manière suivante :

$$\vec{n}_i = \frac{\overrightarrow{A_iB_i}}{\left\|\overrightarrow{A_iB_i}\right\|} \tag{I.21}$$

La vitesse du point B_i est donnée par :

$$\vec{V}_{B_i/\Re_0} = \left.\frac{d\overrightarrow{A_iB_i}}{dt}\right|_{\Re_0} = \dot{q}_i\vec{n}_i + q_i\vec{\Omega}(J_i/\Re_0)\wedge\vec{n}_i \tag{I.22}$$

Où $\vec{\Omega}(J_i/\Re_0)$ est le vecteur de vitesse de rotation de la jambe i par rapport au référentiel fixe $\Re_0$. A partir de l'équation (I.22) on peut écrire

$$\vec{V}_{B_i/\Re_0}.\vec{n}_i = \dot{q}_i \tag{I.23}$$

En utilisant la loi de transport on a :

$$\vec{V}_{B_i/\Re_0} = \vec{V}_{C/\Re_0} + \vec{\Omega}(m/\Re_0) \wedge \overrightarrow{CB_i} \tag{I.24}$$

Où $\vec{\Omega}(m/\Re_0)$ est le vecteur de vitesse de rotation de la plateforme mobile par rapport au référentiel fixe $\Re_0$.

On a alors

$$\dot{q}_i = \vec{V}_{C/\Re_0}.\vec{n}_i + \left(\vec{\Omega}(m/\Re_0) \wedge \overrightarrow{CB_i}\right)\vec{n}_i \tag{I.25}$$

Ce qui peut s'écrire de la manière suivante :

$$\dot{q}_i = \vec{n}_i.\vec{V}_{C/\Re_0} + \left(\overrightarrow{CB_i} \wedge \vec{n}_i\right)\vec{\Omega}(m/\Re_0) \tag{I.26}$$

Comme exemple, on peut prendre le robot plan 3-R<u>P</u>R monté dans la figure I.26

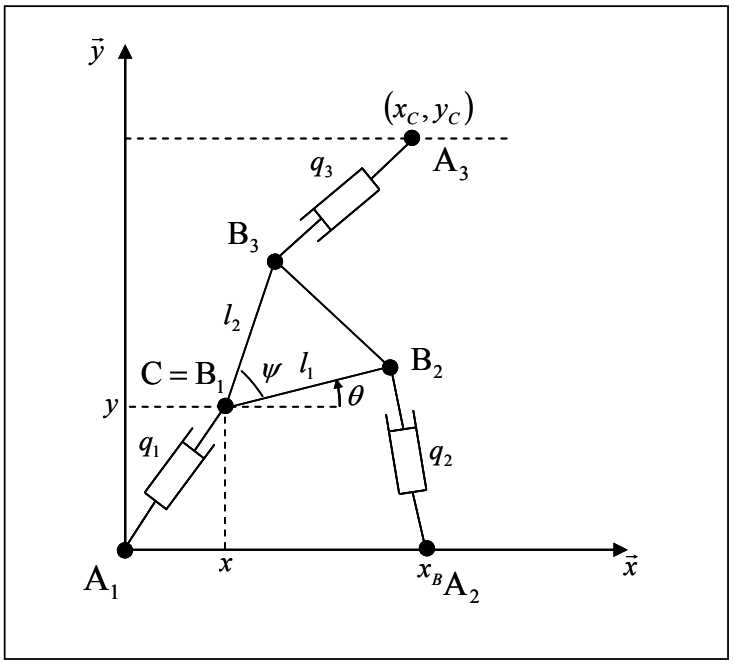


Figure I.26. Robot plan 3-R<u>P</u>R

Pour cet exemple on a

$$\vec{n}_1 = \begin{pmatrix} \dfrac{x}{q_1} \\ \dfrac{y}{q_1} \\ 0 \end{pmatrix}; \vec{n}_2 = \begin{pmatrix} \dfrac{x + l_1 c_\theta - x_B}{q_2} \\ \dfrac{y + l_1 s_\theta}{q_2} \\ 0 \end{pmatrix}; \vec{n}_3 = \begin{pmatrix} \dfrac{x + l_2 c_{\theta\psi} - x_C}{q_3} \\ \dfrac{y + l_2 s_{\theta\psi} - y_C}{q_3} \\ 0 \end{pmatrix} \tag{I.27}$$

Où

$$s_\gamma = \sin\gamma\,; c_\gamma = \cos\gamma\,; s_{\gamma\psi} = \sin(\gamma + \psi); c_{\gamma\psi} = \cos(\gamma + \psi) \tag{I.28}$$

On a :

$$\overrightarrow{\mathrm{CB}_1} = \vec{0} \tag{I.29}$$

$$\overrightarrow{\mathrm{CB}_2} = l_1 c_\theta \vec{x} + l_1 s_\theta \vec{y} \tag{I.30}$$

$$\overrightarrow{\mathrm{CB}_3} = l_2 c_{\theta\psi} \vec{x} + l_2 s_{\theta\psi} \vec{y} \tag{I.31}$$

Le modèle cinématique du 3-RPR s'écrit alors de la manière suivante :

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \begin{pmatrix} \dfrac{x}{q_1} & \dfrac{y}{q_1} & 0 \\ \dfrac{x + l_1 c_\theta - x_B}{q_2} & \dfrac{y + l_1 s_\theta}{q_2} & \dfrac{l_1 c_\theta y - l_1 s_\theta x + l_1 s_\theta x_B}{q_2} \\ \dfrac{x + l_2 c_{\theta\psi} - x_C}{q_3} & \dfrac{y + l_2 s_{\theta\psi} - y_C}{q_3} & \dfrac{l_2 c_{\theta\psi} y - l_2 c_{\theta\psi} y_C - l_2 s_{\theta\psi} x + l_2 s_{\theta\psi} x_C}{q_3} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{I.32}$$

L'équation (I.32) peut aussi être obtenue en dérivant les équations (I.10), (I.11) et (I.12).

4.3 Cas général

Dans le cas général, il suffit de dériver le modèle géométrique par rapport au temps, afin d'obtenir le modèle cinématique. Le modèle géométrique s'écrit sous la forme suivante :

$$F\left(\mathbf{X}_{op}, \mathbf{q}_a\right) = 0 \tag{I.33}$$

Où F est une fonction vectorielle. $\mathbf{X}_{op}$ les variables opérationnelles et $\mathbf{q}_a$ les variables actives.

En dérivant le modèle de l'équation (I.33), on aboutit au modèle cinématique suivant :

$$\mathbf{A}\dot{\mathbf{X}}_{op} + \mathbf{B}\dot{\mathbf{q}}_a = 0 \tag{I.34}$$

Où $\mathbf{A}$ et $\mathbf{B}$ sont deux matrices jacobiennes. $\mathbf{A}$ est appelée matrice jacobienne parallèle et $\mathbf{B}$ matrice jacobienne sérielle d'après [Chablat 98].

On prend pour exemple le robot 2-RPR de la figure I.27.

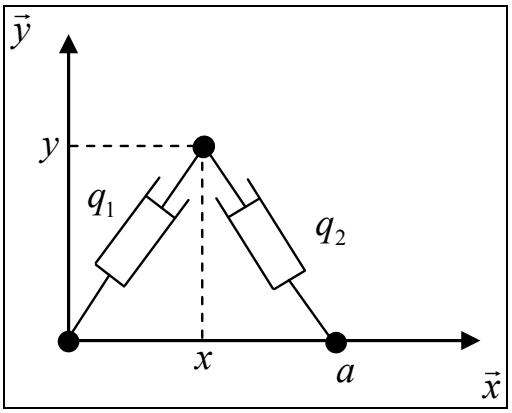


Figure I.27. Robot 2-RPR

Le modèle géométrique s'écrit de la manière suivante :

$$F\left(\mathbf{X}_{op},\mathbf{q}_a\right)=\begin{pmatrix} q_1^2-\left(x^2+y^2\right) \\ q_2^2-\left((x-a)^2+y^2\right)\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \tag{I.35}$$

avec,

$$\mathbf{X}_{op}=\begin{Bmatrix}x\\y\end{Bmatrix} \text{ et } \mathbf{q}_a=\begin{Bmatrix}q_1\\q_2\end{Bmatrix} \tag{I.36}$$

L'équation (I.35) permet de donner le modèle cinématique suivant :

$$\begin{bmatrix}-x & -y\\ a-x & -y\end{bmatrix}\begin{Bmatrix}\dot{x}\\\dot{y}\end{Bmatrix}+\begin{bmatrix}q_1 & 0\\0 & q_2\end{bmatrix}\begin{Bmatrix}\dot{q}_1\\\dot{q}_2\end{Bmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \tag{I.37}$$

5 Singularités

Tous les manipulateurs qu'ils soient parallèles ou sériels, présentent des configurations où le nombre de degrés de liberté du manipulateur est différent de celui des autres configurations. Ces configurations sont appelées configurations singulières. Les manipulateurs peuvent alors gagner ou perdre des degrés de liberté dans ces configurations singulières. De plus, ces configurations peuvent être à la frontière de l'espace de travail comme pour les robots sériels. Elles peuvent être aussi à l'intérieur de l'espace de travail comme c'est le cas de plusieurs robots parallèles. Ces dernières configurations sont gênantes lors de la génération des trajectoires, puisque les configurations singulières et même leur voisinage provoquent soit une augmentation importante des efforts appliqués dans les articulations ce qui peut endommager le manipulateur, soit une perte de rigidité qui se traduit par une instabilité de la plateforme et une perte de contrôle du manipulateur. Plusieurs travaux ont été effectués dans ce domaine. La plupart de ces travaux utilisent une méthode analytique fondée sur l'étude des matrices cinématiques du manipulateur. Il existe aussi d'autres méthodes géométriques pour détecter les singularités en utilisant la géométrie de Grassmann [Merlet 89].

5.1 Méthode analytique

Cette méthode repose sur l'étude des matrices cinématiques du manipulateur. La méthode retenue par [Gosselin 90] utilise le modèle cinématique général donné par l'équation (I.34).

Les singularités sont détectées en analysant les matrices $\mathbf{A}$ et $\mathbf{B}$. Il existe trois types de singularités, qui sont les singularités parallèles, les singularités sérielles et les singularités parallèles/sérielles

5.1.1 Les singularités parallèles

Ces singularités sont dues à la perte de rang de la matrice jacobienne parallèle $\mathbf{A}$. Dans ce cas, il est possible de déplacer la plateforme mobile tout en bloquant les actionneurs. Ces singularités, propres aux manipulateurs parallèles, existent à l'intérieur de l'espace de travail, ce qui est extrêmement gênant lors de la planification de trajectoire. Ces positions sont à éviter puisqu'elles correspondent à une perte de contrôlabilité du manipulateur.

5.1.2 Les singularités sérielles

Les singularités sérielles sont dues à la perte de rang de la matrice jacobienne sérielle $\mathbf{B}$. Dans ce cas, certaines vitesses de la plateforme ne peuvent pas être obtenues. Ces singularités représentent les limites de l'espace de travail.

5.1.3 Les singularités parallèles/sérielles

Ces singularités correspondent à une perte de rang simultanée des matrices jacobiennes $\mathbf{A}$ et $\mathbf{B}$. Dans ce cas il est possible de déplacer la plateforme mobile tout en bloquant les actionneurs et en plus de cela, il existe des vitesses de la plateforme qui ne peuvent pas êtres obtenues.

5.1.4 Exemple

Comme exemple, on peut prendre le robot 2-RPR montré dans la figure I.27. Le modèle cinématique est donné par l'équation (I.38).

$$\mathbf{A} = \begin{bmatrix} -x & -y \\ a-x & -y \end{bmatrix} \text{ et } \mathbf{B} = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \tag{I.38}$$

Les singularités parallèles sont données par l'équation suivante

$$\det(\mathbf{A}) = 0 \Rightarrow ay = 0 \Rightarrow y = 0 \tag{I.39}$$

Ces singularités séparent l'espace de travail du robot 2-RPR en deux domaines différents qu'on montre dans la figure I.28.

Les singularités sérielles délimitent l'espace de travail du robot, elles correspondent au cas où l'un des deux vérins est en butée.

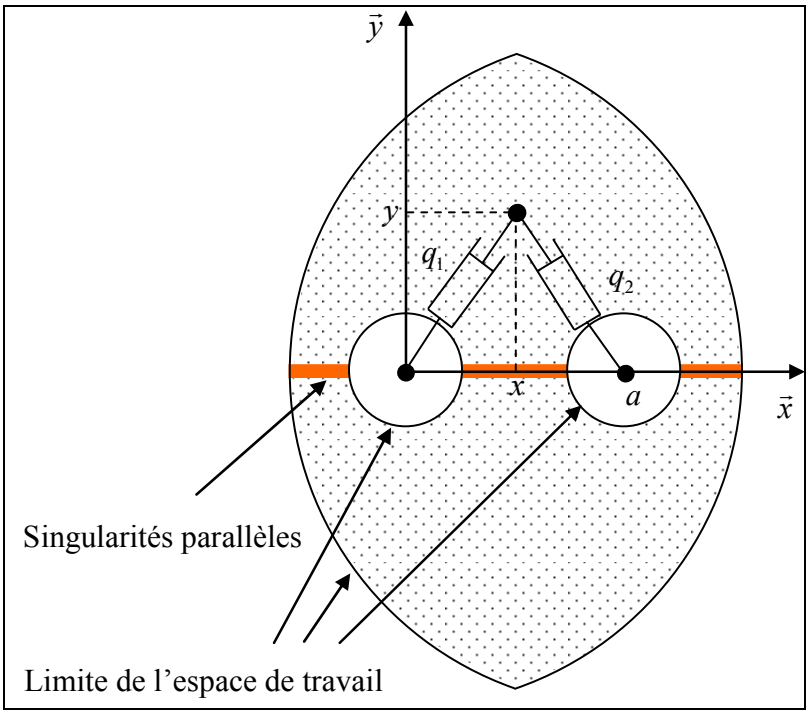


Figure I.28. Singularités du robot 2-RPR

5.2 Autres Méthodes

La méthode analytique que nous avons présentée à des limitations puisqu'elle ne prend pas en considérations les variables passives. Il a été démontré dans [Zlatanov 98], qu'il existe des singularités, dans le cas général des chaînes cinématiques fermées, qui ne peuvent pas être détectées par les équations ne faisant intervenir que les variables actionnées et les variables passives. Pour résoudre ce problème, Merlet [Merlet 89] a utilisé une méthode en se basant sur la géométrie de Grassmann qui utilise les vecteurs de Plücker.

L'étude de singularité est un sujet encore largement étudié, car les robots parallèles sont très diversifiés et il n'existe pas une méthode systématique permettant d'obtenir le modèle géométrique ou cinématique de ces robots et de les analyser. Dans [Guanfeng 03] des équations de fermeture sont générées, en considérant que le robot parallèle est un ensemble de robots sériels et en écrivant pour chacun de ces robots sériels le modèle géométrique, ce qui donne des équations avec toutes les variables opérationnelles et les variables articulaires

passives et actives. Ces équations sont ensuite analysées et permettent de détecter trois types de singularités que les auteurs appellent singularités actives (A), singularités de l'organe terminal (E) et singularité des paramètres (P). Dans [Bonev 03] la théorie des vis a été utilisée pour analyser les singularités des mécanismes plans à 3 ddl.

Dans ce mémoire, nous allons utiliser les équations de fermeture de chaînes, qui contiennent à la fois les variables actionnées, les variables passives et les variables opérationnelles afin de générer les mouvements des robots et afin de détecter les configurations singulières. Les équations de fermeture de chaînes sont générées automatiquement lors de la modélisation du robot, elles se basent sur les contraintes géométriques utilisées pour assembler la structure. Cette approche est détaillée dans les chapitres suivants.

6 Planification de trajectoires

Dans la suite, nous nous intéressons également à la planification de trajectoire des robots parallèles.

La planification de trajectoire est un sujet d'étude très important en robotique. Elle se base sur trois domaines scientifiques variés. Ces domaines sont la robotique, l'intelligence artificielle et le contrôle. Le problème étudié par la planification de trajectoire, consiste à trouver un chemin « sûr » qui emmène un robot d'une position initiale à une position finale désirée pour effectuer une tâche. Par « sûr » on entend que le robot ne doit pas entrer en collision ni avec les autres objets qui peuvent se trouver sur son chemin, ni avec lui-même. Il ne doit pas non plus passer par des singularités qui risquent de l'endommager ou de faire perdre sa contrôlabilité. Ce problème est très vaste et admet plusieurs variantes comme la planification de trajectoire pour les robots mobiles, la coopération de plusieurs robots, la planification sous contraintes différentielles et la planification des mécanismes à chaînes fermées. Il peut être appliqué dans d'autres domaines que la robotique, tel que la chimie par exemple pour la modélisation des molécules. Une liste plus complète est donnée dans [LaValle 06].

6.1 Classification des types de méthodes

Traditionnellement, les méthodes de planification de trajectoires ont été classées en trois types de méthodes qui sont, les méthodes globales, les méthodes locales et les méthodes mixtes.

6.1.1 Les méthodes globales

Les méthodes globales ([Paden 89], [Lengyel 90], [Kondo 91]) supposent une connaissance totale de l'environnement. Ces méthodes opèrent généralement en deux phases. La première phase consiste à modéliser l'espace libre accessible par le robot. Ce qui est fait généralement hors ligne une seule fois pour le robot dans son environnement. La deuxième phase se base sur le modèle généré par la première phase afin de trouver une trajectoire. Cette phase correspond à des requêtes effectuées en ligne qui sont rapides, vu que le travail le plus lourd en temps de calcul est effectué hors ligne. Les méthodes globales ne sont pas adaptées aux environnements dynamiques ou bien aux environnements qui changent lors du déplacement du robot. Tout changement dans l'environnement du robot, nécessite de recalculer la première phase qui est très gourmand en temps de calcul. De plus, le temps de calcul augmente exponentiellement avec le nombre de ddl du robot.

6.1.2 Les méthodes locales

Pour palier les problèmes de temps de calcul des méthodes globales et pour permettre la navigation des robots dans des environnements dynamiques, les méthodes locales ont été proposées. L'une des plus populaire de ces méthodes est la méthode des champs de potentiel proposée par Khatib [Khatib 86]. Cette méthode suppose que le robot évolue dans un champ potentiel qui attire le robot vers la position souhaitée, et qui repousse les différentes pièces du robot loin des obstacles présents dans l'environnement. Il existe une autre méthode locale aussi populaire, appelée la méthode des contraintes. Cette méthode, proposée par Faverjon et Tournassoud [Favejon 87], attire le robot vers son but tout en considérant les obstacles comme étant des contraintes sur le mouvement du robot. A cause du caractère local de ces méthodes qui ne connaissent pas tout leur environnement, le robot peut se bloquer dans un minimum local et ne peut plus avancer vers son but. Nous avons proposé deux types de méthodes de planification de trajectoires pour les robots sériels [Lahouar 06a]. La première méthode [Lahouar 06b] est adaptée aux environnements dynamiques et aux robots ayant un grand ddl qui coopèrent dans le même environnement. Elle se base sur la méthode des contraintes pour évoluer vers le but et elle permet d'éviter les minima locaux en choisissant une direction de contournement d'obstacle à chaque fois qu'un minimum local est détecté. Le robot juge qu'il est dans un minimum local, du moment où il n'arrive plus à avancer vers son but qu'il n'a pas atteint encore. La deuxième méthode que nous avons proposée est une méthode mixte.

6.1.3 Les méthodes mixtes

Les méthodes mixtes, sont une combinaison des deux types de méthodes globales et locales. La méthode mixte que nous avons proposée [Lahouar 05], consiste à utiliser une méthode locale pour avancer vers le but. Dans le cas où une configuration de blocage est rencontrée, une méthode globale prend le relais afin de trouver la meilleure façon de contourner l'obstacle qui bloque le robot. Une fois cet obstacle contourné, le robot continue à évoluer vers son but en utilisant la méthode locale. Cette méthode est adaptée aux robots sériels placés dans des environnements fortement encombrés.

D'autres méthodes mixtes qui se basent sur d'autres principes ont été proposés. Il y a la méthode des chemins élastiques qui consiste à créer une trajectoire en utilisant une méthode globale, puis une méthode locale est utilisée afin de faire la commande en temps réel du robot [Quinlan 93]. Un autre principe proposé dans [Helguera 01], consiste à utiliser une méthode locale jusqu'à ce qu'une configuration de blocage est rencontrée. Dans le cas d'un blocage, un graphe représentant l'environnement proche de la configuration de blocage est construit. Une méthode globale qui se base sur un algorithme de recherche en A* est utilisée pour faire sortir le robot de la position de blocage.

6.1.4 Autres méthodes

Cette dernière dizaine d'années, les recherches se sont orientées vers des méthodes de planification de trajectoires basées sur les cartes probabilistes (Probabilistic roadmaps). Ces méthodes ont été introduites par les travaux de Kavraki et al. [Kavraki 96] afin de remédier au problème de temps de calcul des méthodes globale. Cette méthode permet de modéliser l'espace de travail en choisissant des positions au hasard et en les plaçant dans un diagramme qui représente les chemins sans collisions entre les différentes positions. Pour chaque nouvelle position insérée, les chemins vers les autres positions sont vérifiées et rajoutées dans le diagramme. Plusieurs variantes de cette méthode ont été proposées pour la rendre soit plus rapide comme la méthode basée sur le graphe de visibilité [Simeon 00], soit pour résoudre le problème des couloirs étroits avec la méthode de l'axe médian [Wilmarth 99], soit pour essayer d'appliquer ces méthodes a des environnements dynamiques [Bohlin 00].

D'autres méthodes inspirées de la biologie ont également été proposées, comme par exemple la méthode du réseau de neurones proposée par Yang [Yang 03]. Cette approche présente de bons résultats dans les environnements dynamiques des robots à 2ddl, par contre, des complications apparaissent pour les robots avec un nombre de ddl supérieur à deux car le nombre de neurones augmente exponentiellement avec le nombre de ddl du robot.

6.2 Cas des robots parallèles

La planification de trajectoire pour les robots parallèles est largement étudiée pour le cas de l'évitement des singularités. Les méthodes proposées, soit se contentent de vérifier qu'une trajectoire ne passe pas par des configurations singulières comme c'est le cas dans [Merlet 01], soit donnent une trajectoire évitant les singularités à partir d'une trajectoire de référence comme dans [Su 03] et [Battacharya 98], soit une trajectoire est calculée directement entre la position initiale et la position finale comme c'est le cas dans [Sen 03], [Dash 05].

Chablat [Chablat 98] a étudié les domaines d'unicité et de parcourabilité des manipulateurs parallèles ce qui a permis une meilleure compréhension de la planification de trajectoire de ce type de robots.

Le problème de collision dans le cas des robots parallèles, est nettement moins étudié. On cite les travaux de Cortès [Cortès 03] qui s'est intéressé à la génération des trajectoires sans collision que ce soit les collisions internes ou avec les objets qui peuvent se trouver dans l'environnement des robots parallèles, par contre, il n'a pas abordé le problème des singularités. Le but de ce travail est d'étudier la planification de trajectoire des robots parallèles en prenant en considération à la fois les collisions et les singularités, les travaux dans ce sens étant rares [Alba 07].

Conclusion

Dans ce chapitre, nous avons présenté les robots parallèles et les notions qui s'y rapportent. Ces robots sont très diversifiés. Nous avons rappelé qu'il existe 1140 architectures possibles de robots plans et dans le seul cas où on n'utilise que les articulations de type rotoïde ou de type prismatique. Nous avons rappelé aussi que dans le cas des robots spatiaux, il existe 3850 formes possibles de plateformes de Stewart généralisées. Toutes ces architectures ne sont pas évidentes à modéliser comme il n'existe pas de méthode systématique permettant de le faire. En plus de ces difficultés de modélisation, il existe des configurations singulières à éviter. Ces configurations singulières rendent la simulation des robots parallèles très compliquée. Il existe plusieurs méthodes permettant de détecter ces singularités. Parmi ces méthodes, il y a la méthode analytique qui se base sur l'analyse des matrices cinématiques. Cette analyse permet de classer les singularités en singularités parallèles, singularités sérielles et singularités parallèles/sérielles. La méthode analytique ne prend pas en considération les variables passives, vu que ces variables ont été éliminés du modèle cinématique étudié. Il a été démontré qu'il existe des singularités particulières qu'on peut détecter en prenant en compte les articulations passives. Pour résoudre ce problème, des méthodes se basant sur la géométrie de Grassmann ont été utilisées. Nous proposons dans le

chapitre suivant d'utiliser toutes les variables du système pour pouvoir analyser l'état du robot et pour détecter les singularités.

Nous avons recensé aussi les méthodes de planification de trajectoire dans le cas général. Nous avons évoqué le cas des robots parallèles et nous avons remarqué que la plupart de ces méthodes, s'intéressent soit à l'évitement des singularités, soit à l'évitement des collisions conjointement. Les méthodes qui évitent les singularités et les collisions à la fois sont rares.

Chapitre II

Modélisation et simulation des robots

Dans le premier chapitre, nous avons passé en revue plusieurs types d'architectures de robots parallèles. La plupart de ces architectures ont été étudiées séparément. Nous avons également présenté quelque approches de la planification de trajectoire. Dans ce chapitre nous nous intéressons à la modélisation et à la simulation des mouvements des robots parallèles.

Introduction

Etant donné la dispersion des types des robots parallèles, il est difficile d'avoir un paramétrage systématique permettant de définir le modèle géométrique comme c'est le cas du paramétrage de Denavitt-Hartenberg pour les robots sériels. Les robots parallèles sont étudiés au cas par cas, ou bien sont regroupés en classes ayant le même procédé d'obtention du modèle géométrique. Pour animer et simuler le mouvement des robots parallèles, il est envisageable de concevoir un logiciel où le modèle géométrique du robot est intégré au cas par cas [Ben 05]. Cette méthode donne de bons résultats mais ne permet pas d'intégrer la géométrie des pièces constituant le robots. Or le fait d'intégrer la géométrie du robot est nécessaire pour s'assurer qu'il n'y a pas de collision lors de l'animation du robot. Les collisions peuvent êtres soit internes entre deux pièces du robot, soit entre le robot et un objet de l'environnement.

Un logiciel de CAO classique comme SolidWorks ou CATIA peut être utilisé pour générer et animer un modèle 3D du robot. Dans ce cas, le robot est un assemblage composé de pièces liées par des contraintes permettant de définir les liaisons.

Au LMS de Poitiers, nous disposons d'un logiciel de modélisation et d'animation des robots sériels. Ce logiciel comporte un module de planification de trajectoire. Dans le cadre de ce travail, nous avons étendu ce logiciel aux robots parallèles.

1 Modélisation

1.1 Architectures des robots

1.1.1 Définitions

Nous commençons par définir quelques notions nécessaires pour la modélisation des robots.

1.1.1.1 Matrice homogène

La matrice homogène est une matrice 4 × 4 permettant de passer d'un repère R_j à un repère R_i (figure II.1). Elle s'écrit de la manière suivante :

$$ {}^{i}\mathbf{T}_{j} = \left[\begin{array}{ccc|c} & {}^{i}\mathbf{A}_{j} & & \mathbf{O}_{i}\mathbf{O}_{j} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad \text{(II.1)} $$

Où O_i et O_j sont les origines des repères R_i et R_j respectivement. La matrice ${}^{i}\mathbf{A}_{j}$ est une matrice orthonormale définissent l'orientation du repère R_j par rapport au repère R_i.

$$ {}^{i}\mathbf{A}_{j} = \begin{array}{c} \\ e_1^i \\ e_2^i \\ e_3^i \end{array} \begin{array}{c} \begin{array}{ccc} e_1^j & e_2^j & e_3^j \end{array} \\ \begin{pmatrix} & & \\ & \cdots & \\ & & \end{pmatrix} \end{array} \qquad \text{(II.2)} $$

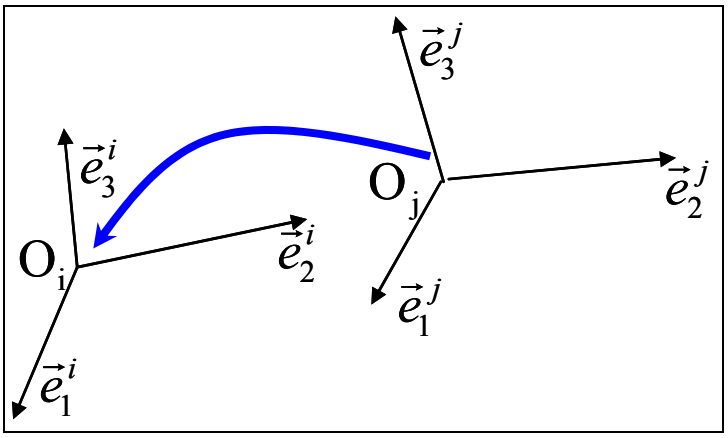


Figure II.1. Passage d'un repère R_j à un repère R_i

Dans SMAR, nous avons choisi d'écrire ${}^{i}\mathbf{A}_{j}$ comme étant le produit d'une matrice rotation d'angle α par rapport à l'axe e_1, d'une rotation d'angle β par rapport à l'axe e_2 et d'une rotation d'angle γ par rapport à l'axe e_3.

$$ {}^{i}\mathbf{A}_{j} = Rot(e_1,\alpha)\times Rot(e_2,\beta)\times Rot(e_3,\gamma) \tag{II.3} $$

Ce qui nous donne :

$$ {}^{i}\mathbf{A}_{j} = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos+\cos\alpha\sin & -\sin\alpha\sin\beta\sin+\cos\alpha\cos & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos+\sin\alpha\sin & \cos\alpha\sin\beta\sin+\sin\alpha\cos & \cos\alpha\cos\beta \end{bmatrix} \tag{II.4} $$

Les coordonnées homogènes d'un point P dans le repère R_i s'écrivent

$$ \mathbf{P}^{i} = \begin{bmatrix} P_1^i & P_2^i & P_3^i & 1 \end{bmatrix}^T \tag{II.5} $$

Un vecteur homogène **U** est écrit comme suit dans le repère R_i

$$\mathbf{U}^i = \begin{bmatrix} U_1^i & U_2^i & U_3^i & 0 \end{bmatrix}^T \tag{II.6}$$

Nous avons les relations suivantes

$$\mathbf{P}^i = {}^i\mathbf{T}_j \mathbf{P}^j \tag{II.7}$$

$$\mathbf{U}^i = {}^i\mathbf{T}_j \mathbf{U}^j \tag{II.8}$$

1.1.1.2 Maillon

Un maillon est un ensemble de pièces du robot. Il ne doit y avoir aucun mouvement possible entre les différentes pièces d'un même maillon.

1.1.1.3 Liaison

Une liaison décrit les mouvements possibles entre deux maillons. Un robot est composé alors de plusieurs maillons liés entre eux par des liaisons (figure II.2).

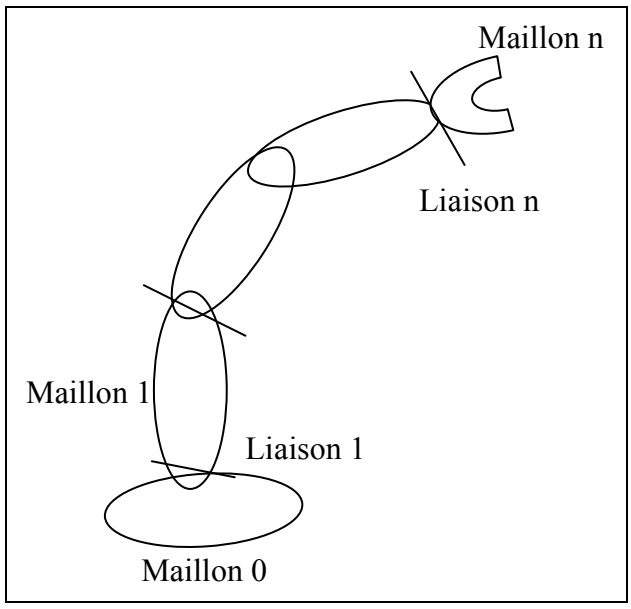


Figure II.2. Un robot se compose de maillons et de liaisons

1.1.1.4 Graphe de liaisons

Un graphe de liaisons est un graphe minimal permettant de visualiser les liaisons entre les différents maillons du robot (figure II.3).

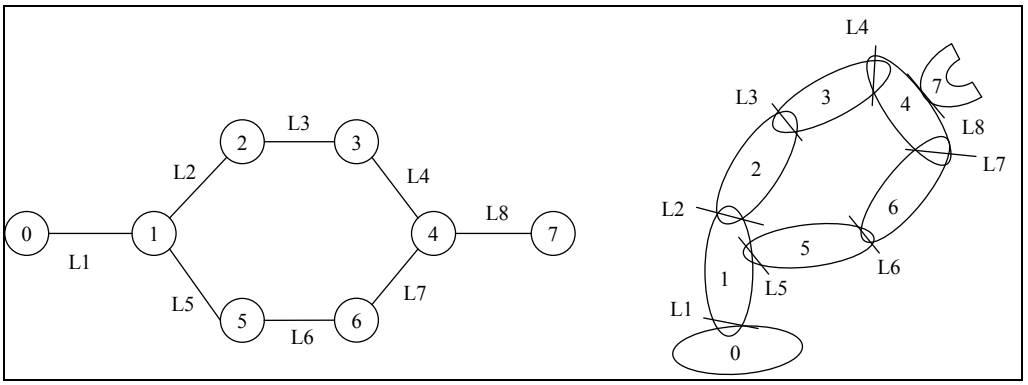


Figure II.3. Graphe de liaisons

1.1.2 Types de chaînes

Une chaîne est un ensemble de maillons où chaque maillon a au moins une liaison avec les autres maillons.

1.1.2.1 Chaînes ouvertes

Une chaîne ouverte est une chaîne où le premier maillon et le dernier maillon ont chacun une seule liaison, et où chacun des autres maillons a deux liaisons (figure II.4).

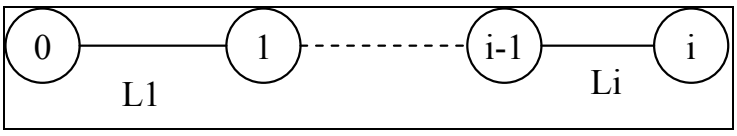


Figure II.4. Chaîne ouverte

1.1.2.2 Chaînes fermées

Une chaîne fermée est une chaîne ou chacun des maillons a au moins deux liaisons. Nous appelons chaîne fermée simple une chaîne ou chaque maillon a exactement deux liaisons avec les autres maillons (figure II.5). Une chaîne fermée complexe est une chaîne fermée qui n'est pas simple (figure II.6).

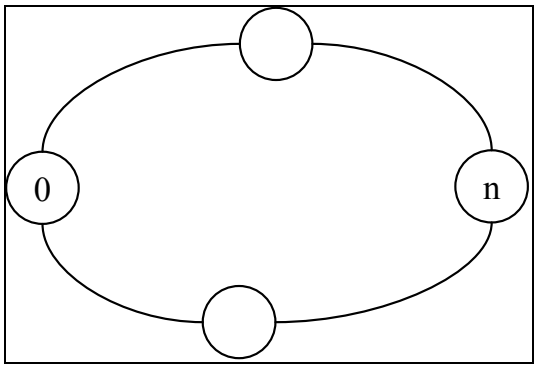


Figure II.5. Chaîne fermée simple

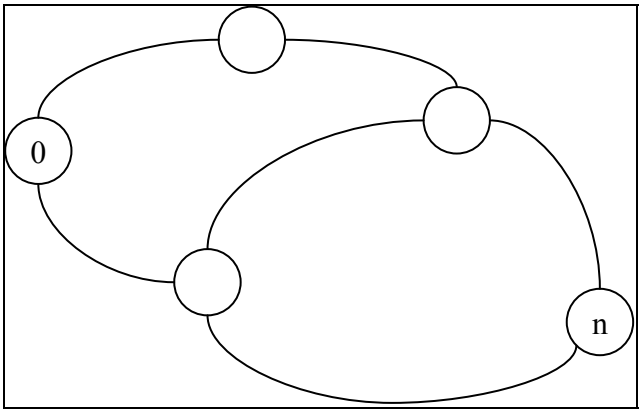


Figure II.6. Chaîne fermée complexe

1.1.2.3 Chaînes mixtes

Une chaîne est dite mixte si elle n'est ni ouverte ni fermée. Une chaîne mixte peut être considérée comme une combinaison de chaînes ouvertes et de chaînes fermées (figure II.7).

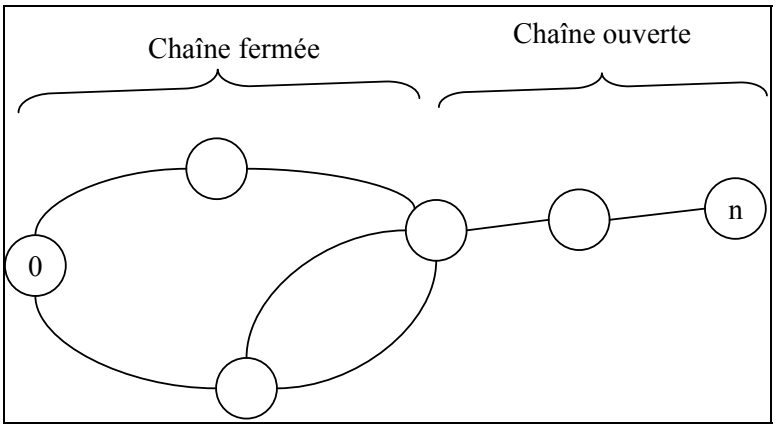


Figure II.7. Chaîne mixte

1.1.3 Types de robots

Nous distinguons trois types de robots selon le type de la chaîne de leur graphe de liaisons. SMAR permet de modéliser ces trois types de robots.

1.1.3.1 Robots sériels

Un robot sériel est composé d'une chaîne cinématique ouverte (Figure II.8).

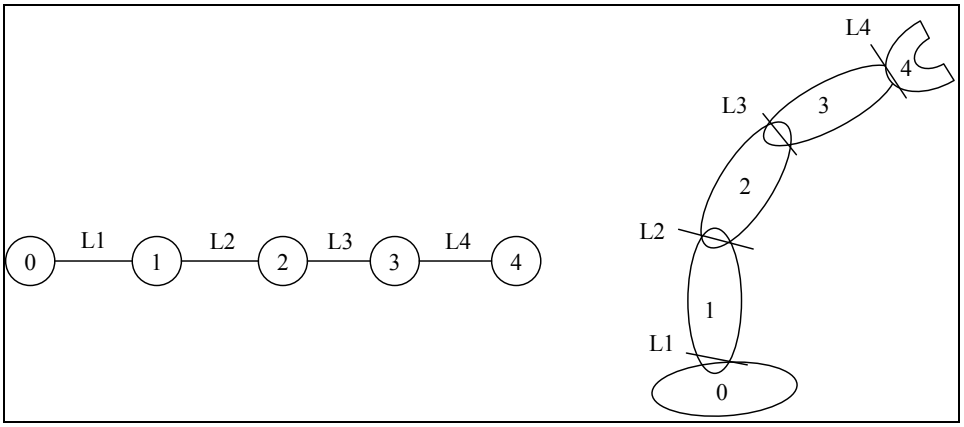


Figure II.8. Robot sériel

1.1.3.2 Robots parallèles

Les robots parallèles sont constitués de chaînes cinématiques fermées (figure II.9). Un robot parallèle est composé d'une base fixe et d'une plateforme mobile. La plateforme est reliée à la base par des jambes. Il y a deux types de liaisons dans un robot parallèle : les liaisons actives et les liaisons passives. Les liaisons actives sont des liaisons actionnées. Le mouvement relatif de ces liaisons est imposé par les actionneurs, tandis que les liaisons passives ne sont pas actionnées, leurs mouvements dépendent du mouvement global de la structure.

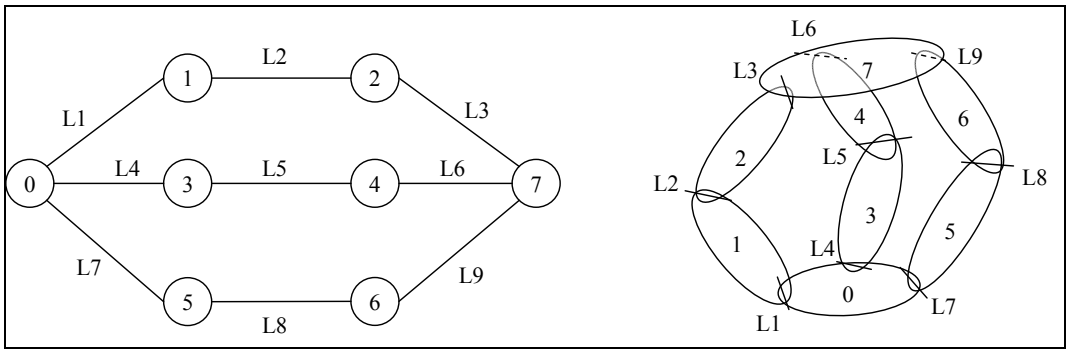


Figure II.9. Robot parallèle

1.1.3.3 Robots mixtes

Les robots mixtes sont constitués par une chaîne mixte (figure II.10).

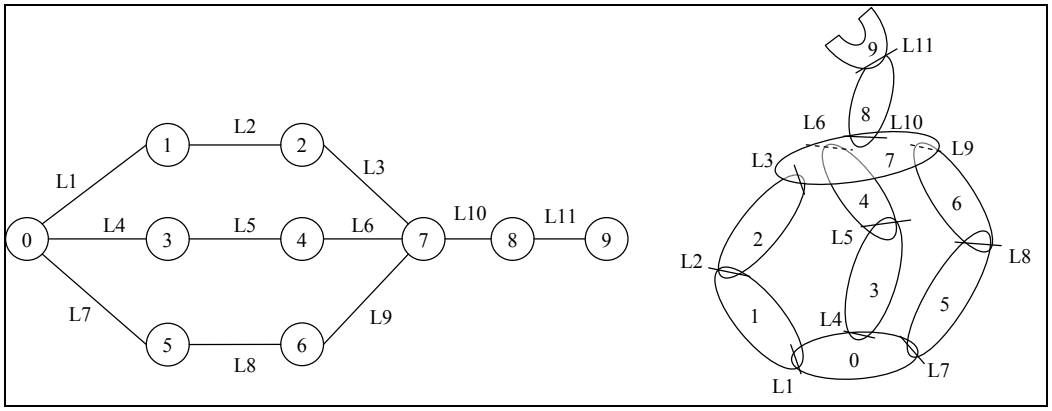


Figure II.10. Robot mixte

1.2 Modélisation dans SMAR

1.2.1 Types de maillons

Trois types de maillons sont pris en compte dans SMAR (figure II.11).

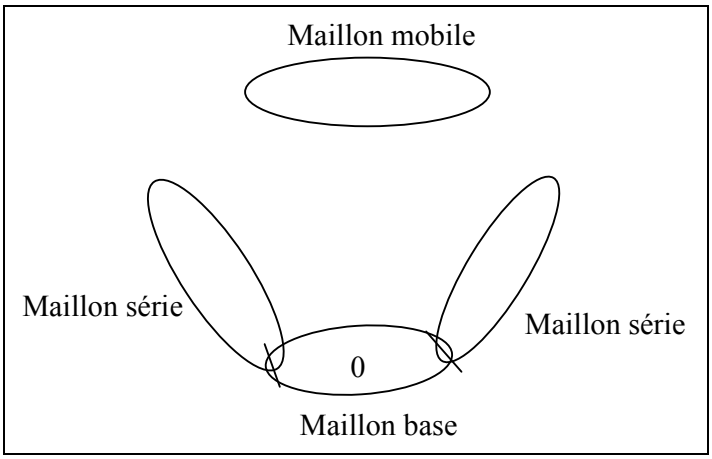


Figure II.11. Types de maillons dans SMAR

1.2.1.1 Maillon base

Le maillon base est le maillon initial du robot. Il est unique pour chaque robot. C'est le maillon de référence pour les autres maillons du robot. Ce maillon est lié à l'environnement du robot par une matrice homogène définissant le repère R_0 du robot. Pour déplacer le robot il suffit d'agir sur la matrice homogène du maillon base.

1.2.1.2 Maillon série

Le maillon série est un maillon défini par rapport à un autre maillon du robot qu'on appelle maillon père dudit maillon série. Il y a une matrice homogène qui donne la pose (position et orientation) du maillon série par rapport au maillon père. Cette matrice se décompose en deux matrices homogènes. La première permet l'orientation et le positionnement selon l'axe et la position de la liaison, on l'appelle matrice position. La deuxième matrice homogène définit la liaison on l'appelle matrice liaison (figure II.12).

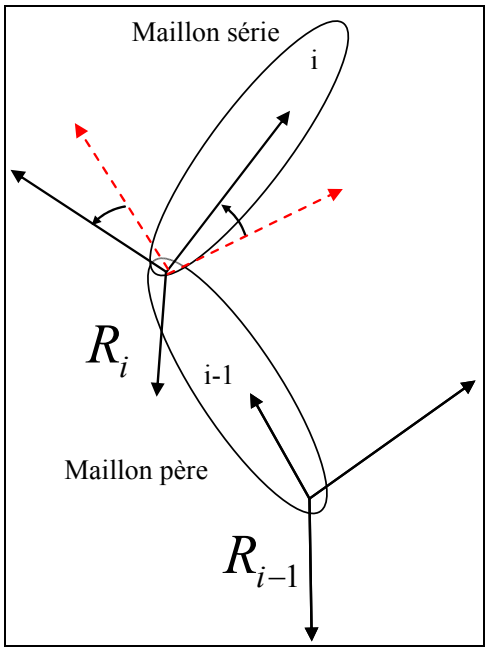


Figure II.12. Maillon série

1.2.1.3 Maillon mobile

Un maillon mobile est un maillon libre, il est repéré par une matrice homogène par rapport au maillon base. Dans le cas général, cette matrice homogène dépend de 3 paramètres (x,y,z) pour la position et de 3 paramètres (α,β,γ) pour l'orientation. Pour le cas plan cette matrice dépend uniquement de 3 paramètres (x,y,θ), x et y pour la position et θ pour l'orientation. Pour un maillon mobile plan (noté i), la matrice homogène s'écrit

$$ {}^{0}\mathbf{T}_i = \left[\begin{array}{ccc|c} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad \text{(II.9)} $$

1.2.2 Types de liaisons

Dans SMAR, les liaisons sont utilisées entre un maillon série et le maillon père correspondant. La matrice homogène liée au maillon série est la matrice de transformation du repère liée au maillon série par rapport au maillon père correspondant. Nous avons précisé que cette matrice se décompose en une matrice position et une matrice liaison. La matrice liaison permet de définir la liaison selon les liaisons de bases supportées par SMAR. Les

liaisons de base sont définies selon les axes principaux du repère. La matrice position permet d'orienter et de positionner la liaison. Nous définissons deux types de liaisons de base selon le nombre de ddl.

1.2.2.1 Liaisons simples

Une liaison simple est une liaison qui a un seul degré de liberté. Les liaisons supportées par SMAR sont les liaisons rotoïdes selon e_1, e_2 ou e_3 et les liaisons prismatiques selon e_1, e_2 ou e_3. Pour modéliser une liaison selon un axe quelconque il suffit de définir la matrice position pour que l'axe de la liaison coïncide avec l'un des axes e_1, e_2 ou e_3. Il existe alors six liaisons de base.

1.2.2.2 Liaisons composées

Une liaison composée est une liaison qui a plus d'un degré de liberté. Ces liaisons peuvent s'écrire comme étant une composition de liaisons simples. Les liaisons dans SMAR sont :

1. Liaison sphérique

La liaison sphérique se compose de trois liaisons rotoïdes selon les axes e_1, e_2 et e_3 respectivement.

2. Liaison cylindrique

La liaison cylindrique se compose d'une liaison rotoïde selon l'un des axes e_1, e_2 ou e_3, d'une part, et d'une liaison prismatique selon le même axe, d'autre part.

3. Liaison universelle

La liaison universelle se compose de deux liaisons rotoïdes selon deux axes différents parmi les axes e_1, e_2 et e_3.

1.2.3 Types de contraintes

Dans SMAR, nous avons implanté des contraintes entre les maillons, pour permettre la modélisation des chaînes fermées. Les contraintes permettent d'écrire des équations en fonction des paramètres de position des maillons mobiles et des paramètres des liaisons des maillons séries.

1.2.3.1 Contraintes simples

Les contraintes utilisées sont soient des contraintes de coïncidence entre deux points, soient des contraintes de parallélisme entre deux axes ou deux plans, soit des contraintes définissant une liaison linéaire annulaire des contraintes d'encastrement entre deux maillons.

1. Contrainte de coïncidence entre deux points

Une contrainte de coïncidence d'un point P appartenant à la fois au maillon i et au maillon j (figure II.13), s'écrit :

$$ {}^{0}\mathbf{T}_i P^i = {}^{0}\mathbf{T}_j P^j \qquad \text{(II.10)} $$

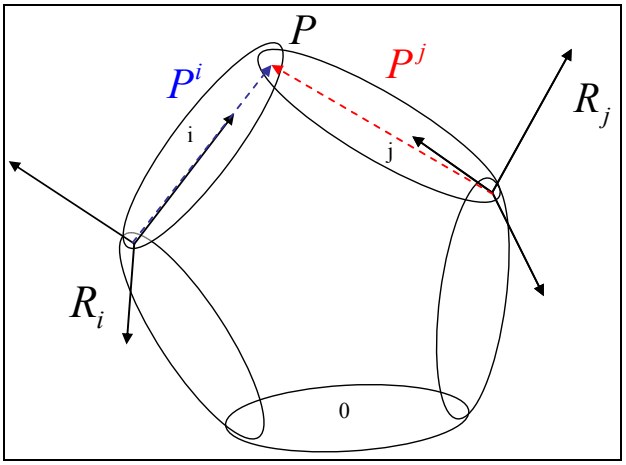


Figure II.13. Coïncidence entre deux points

2. Contrainte de parallélisme entre deux axes

Soit $\mathbf{U}^i$ les composantes du vecteur directeur d'un axe lié à un maillon i et $\mathbf{V}^j$ les coordonnées du vecteur directeur d'un axe lié à un maillon j. La contrainte de parallélisme entre les deux axes (figure II.14), s'écrit :

$$ {}^{0}\mathbf{T}_i \mathbf{U}^i \wedge {}^{0}\mathbf{T}_j \mathbf{V}^j = 0 \qquad \text{(II.11)} $$

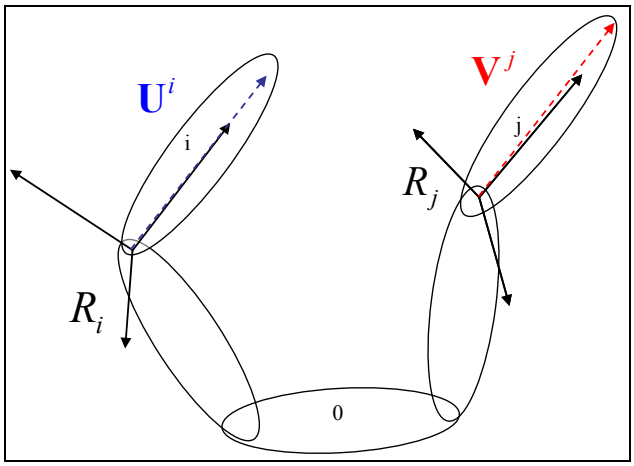


Figure II.14. Parallélisme entre deux axes

3. Contrainte de parallélisme entre deux plans

Soit $\mathbf{n}^i$ un vecteur normal d'un plan lié à un maillon i et $\mathbf{n}^j$ un vecteur normal d'un plan lié à un maillon j. La contrainte de parallélisme entre les deux plans (figure II.15), s'écrit :

$$ {}^0\mathbf{T}_i\mathbf{n}^i \wedge {}^0\mathbf{T}_j\mathbf{n}^j = 0 \qquad \text{(II.12)} $$

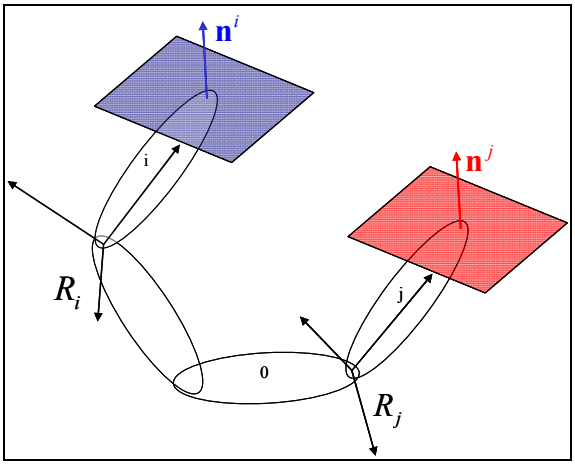


Figure II.15. Parallélismes entre deux plans

4. Contrainte de liaison linéaire annulaire

Ces contraintes sont nécessaires pour avoir des liaisons prismatiques et des liaisons cylindriques. Cette contrainte nécessite un point sur chacun des deux maillons et une direction définie dans l'un des deux maillons (figure II.16). En choisissant A_1 le point dans le maillon

i et A_2 le point dans le maillon j. Soit $\mathbf{U}^0$, le vecteur directeur de la liaison linéaire annulaire, écrit dans le repère R_0. L'équation correspondant à cette contrainte s'écrit :

$$\left({}^0\mathbf{T}_j \, A_2^j - {}^0\mathbf{T}_i \, A_1^i\right) \wedge \mathbf{U}^0 = 0 \tag{II.13}$$

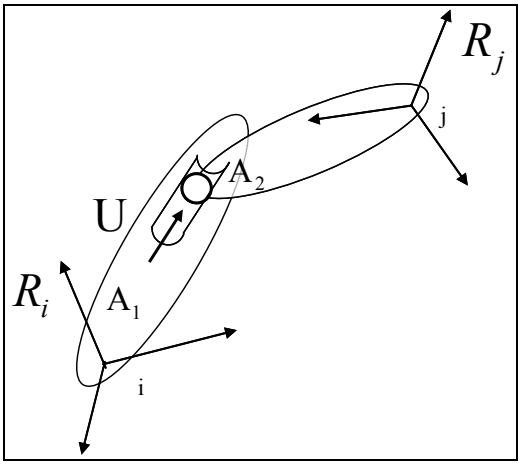


Figure II.16. Contrainte de linéaire annulaire

5. Contrainte d'appui ponctuel

Cette contrainte permet de définir une liaison d'appui ponctuel entre deux maillons (figure II.17). Le maillon i comporte un plan défini par un point A et une normale $\mathbf{n}$. Le maillon j comporte un point B qui doit rester en contact avec le plan maillon j. L'équation correspondant à cette contrainte s'écrit :

$$\left({}^0\mathbf{T}_j \, B^j - {}^0\mathbf{T}_i \, A^i\right)^T {}^0\mathbf{T}_i \mathbf{n}^i = 0 \tag{II.14}$$

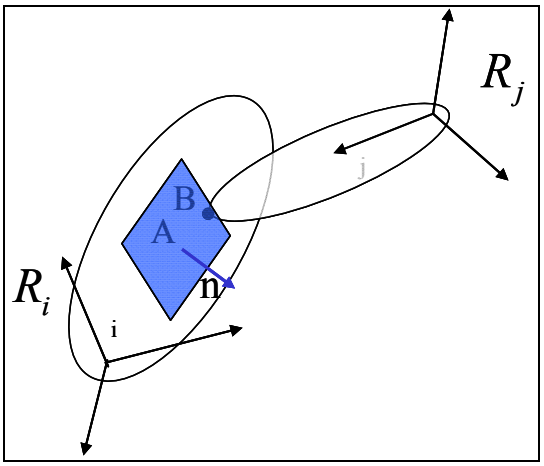


Figure II.17. Contrainte d'appui ponctuel

6. Contrainte d'encastrement entre deux maillons

SMAR donne la possibilité d'encastrer deux maillons i et j dans une configuration particulière (figure II.18). La contrainte d'encastrement s'écrit

$$ {}^{i}\mathbf{T}_{j} = \left({}^{0}\mathbf{T}_{i}\right)^{-1} {}^{0}\mathbf{T}_{j} = \mathbf{T}_{c} \qquad \text{(II.15)} $$

Où $\mathbf{T}_c$ est une matrice homogène constante.

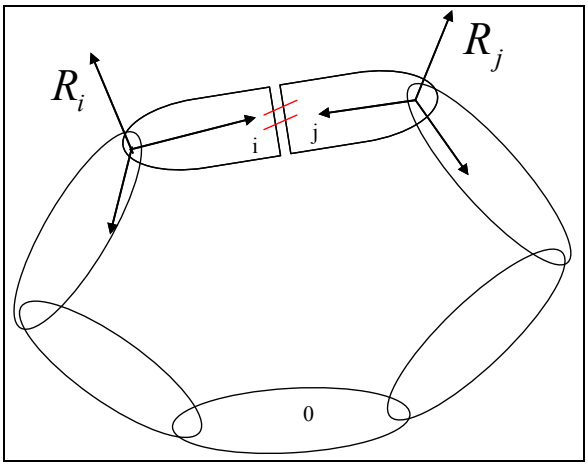


Figure II.18. Encastrement entre deux maillons

1.2.3.2 Liaisons composées de contraintes

En combinant des contraintes simples, nous pouvons définir un grand nombre de liaisons entre deux maillons. Ces liaisons sont l'appui plan, la liaison linéaire rectiligne, la liaison rotoïde, la liaison pivot glissant et la liaison glissière. Pour chacune de ces liaisons il existe plusieurs méthodes de les définir en utilisant les contraintes simples.

1. Liaison d'appui plan

La liaison d'appui plan peut être réalisée en utilisant trois contraintes d'appui ponctuel. Elle peut être réalisée plus simplement en utilisant une contrainte d'appui ponctuel et une contrainte de parallélisme entre deux plans (figure II.19).

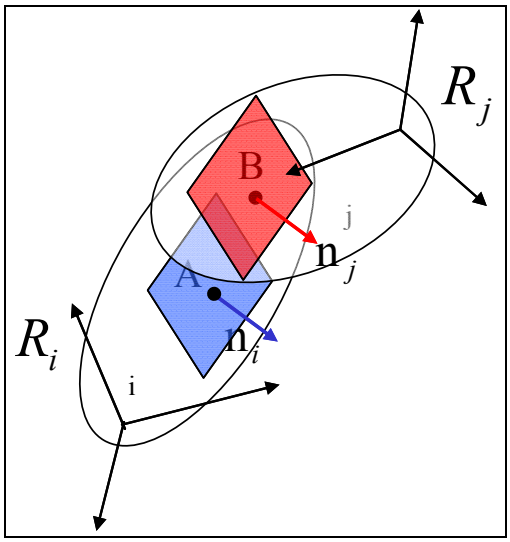


Figure II.19. Contrainte d'appui plan

2. Liaison linéaire rectiligne

La liaison linéaire rectiligne peut être définie comme deux contraintes d'appui ponctuel. Elle peut être définie aussi par une seule contrainte ponctuelle et un vecteur $\mathbf{U}$ définissant la direction de la liaison linéaire rectiligne (figure II.20). Le vecteur doit être perpendiculaire à la normale n définissant le plan de la liaison linéaire rectiligne. Ce qui donne, en plus de l'équation donnée par la liaison ponctuelle, l'équation suivante.

$$\left({}^{0}\mathbf{T}_i \, \mathrm{n}^i\right)^T {}^{0}\mathbf{T}_j \mathbf{U}^j = 0 \qquad \text{(II.16)}$$

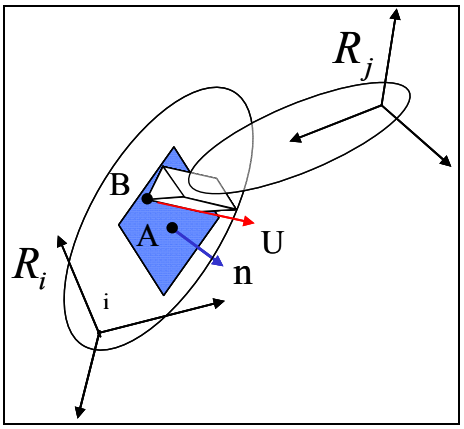


Figure II.20. Contrainte linéaire rectiligne

3. Liaison cylindrique

On peut obtenir une liaison cylindrique à l'aide de deux contraintes linéaires annulaires, ou bien à partir d'une contrainte linéaire annulaire et une contrainte de parallélisme entre deux vecteurs (figure II.21).

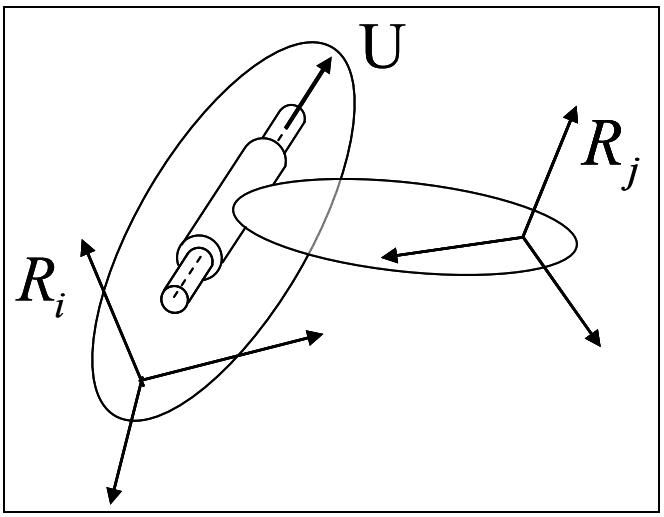


Figure II.21. Liaison pivot glissant

4. Liaison rotoïde

La liaison rotoïde peut être considérée comme une liaison pivot glissant, à laquelle on rajoute une contrainte d'appui ponctuel pour arrêter la translation. Elle peut aussi être définie comme étant deux contraintes de coïncidence entre deux points (figure II.22).

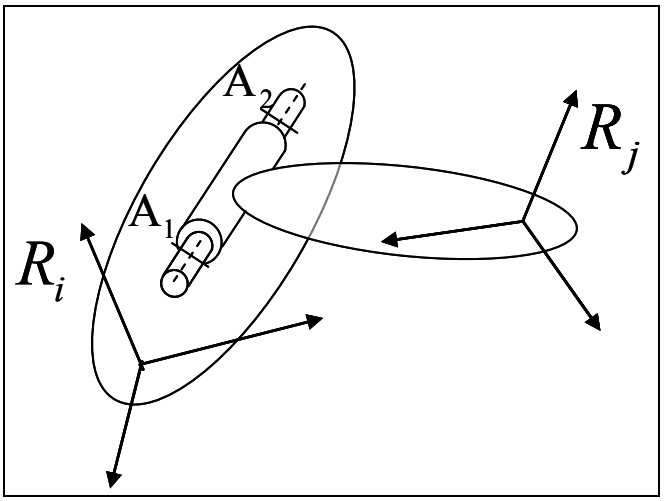


Figure II.22. Liaison rotoïde

5. Liaison prismatique

Une liaison prismatique est une liaison cylindrique à laquelle on rajoute, dans le but d'arrêter la rotation, soit une contrainte d'appui ponctuel, soit une contrainte de parallélisme entre deux vecteurs (figure II.23).

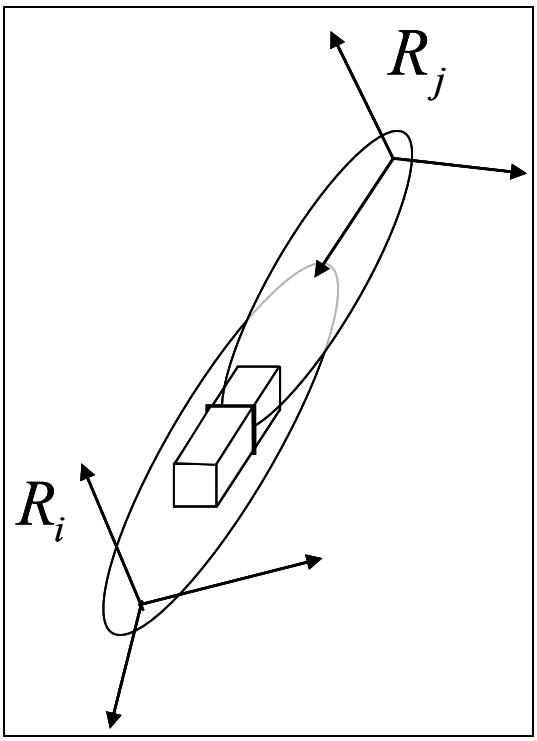


Figure II.23. Liaison glissière

1.2.4 Génération des équations de fermeture de chaînes

SMAR est un outil qui permet de modéliser un grand nombre d'architectures de robots. Il permet aussi de générer des équations selon les contraintes utilisées lors de la modélisation.

Nous proposons deux méthodes pour modéliser les robots parallèles en utilisant SMAR.

Nous montrons aussi comment aboutir aux équations du modèle géométrique direct pour les robots sériels.

1.2.4.1 Première méthode de modélisation des robots parallèles

Pour modéliser les robots parallèles, nous définissons la base du robot comme le maillon base et la plateforme comme étant un maillon mobile. Pour chacune des jambes, il faut choisir une liaison passive que nous transformons en contraintes, cette liaison est appelée liaison de fermeture. La jambe est modélisée par des maillons séries liés soit au maillon base soit au maillon mobile. Pour chaque liaison de fermeture, SMAR écrit les contraintes liant les deux maillons correspondants, ce qui donne les équations liant les paramètres de la plateforme mobile en fonction des paramètres des liaisons restantes (figure II.24).

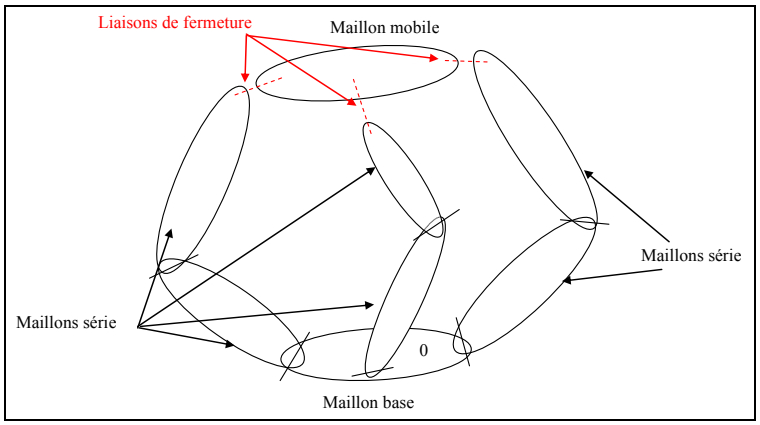


Figure II.24. Première méthode de modélisation des robots parallèles

1.2.4.2 Deuxième méthode de modélisation des robots parallèles

La plateforme mobile est modélisée par plusieurs maillons séries où chacun est lié à une jambe. Puis dans une configuration particulière du robot, des contraintes d'encastrement sont imposées entre les maillons constituant la plateforme mobile (figure II.25). Cette méthode donne des équations liant les paramètres des liaisons entre eux. Afin de rajouter les variables opérationnelles de la plateforme, un maillon mobile peut être encastré avec les maillons constituant la plateforme mobile.

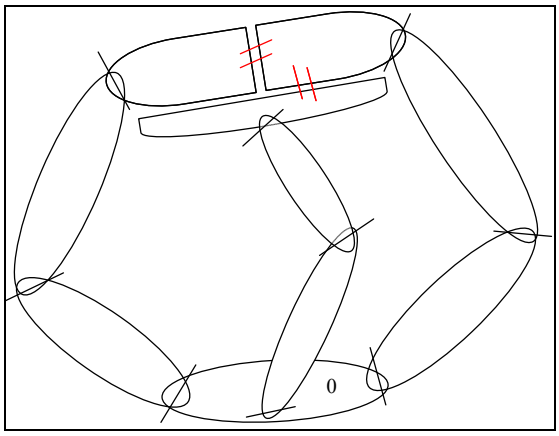


Figure II.25. Deuxième méthode de modélisation des robots parallèles

1.2.4.3 Modèle géométrique direct des robots sériels

Un robot sériel est modélisé par des maillons séries formant la chaîne ouverte. Le dernier maillon représente l'organe terminal. Pour générer le modèle géométrique direct du robot sériel, nous créons un maillon mobile que nous rattachons à l'organe terminal. Nous ajoutons une contrainte d'encastrement entre le maillon mobile et le dernier maillon série. Les paramètres du maillon mobile correspondent aux variables opérationnelles. Ainsi, la contrainte d'encastrement nous donne les équations du modèle géométrique direct (figure II.26). Nous donnons en annexe la modélisation d'un robot lynx (Annexe A)

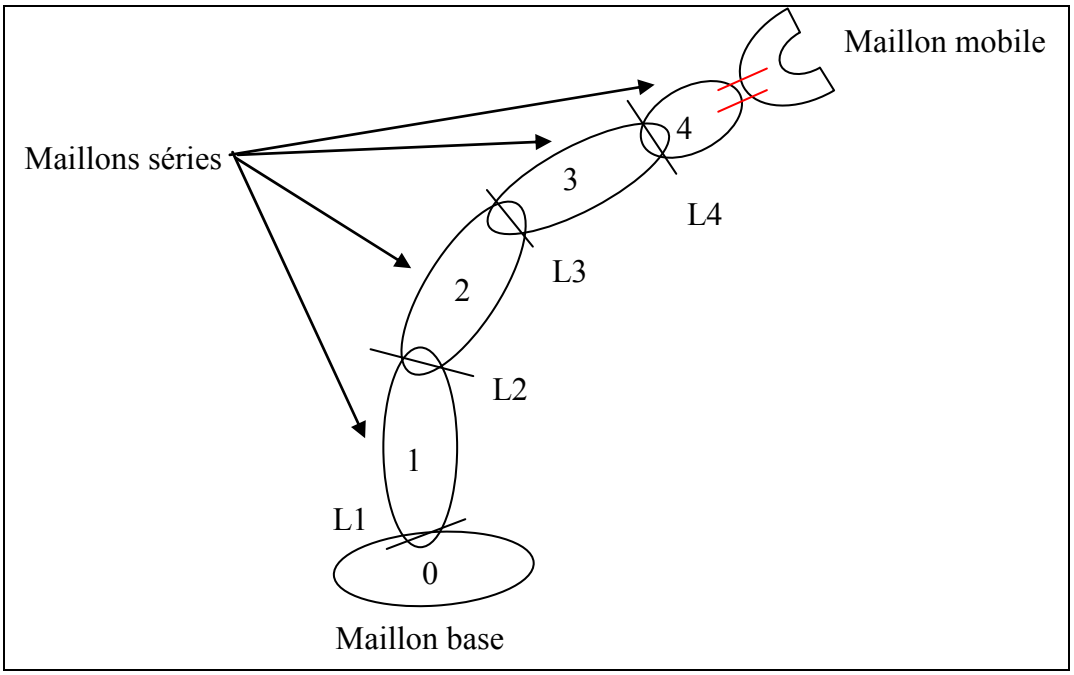


Figure II.26. Méthode de modélisation des robots sériels

1.2.4.4 Modélisation cinématique d'un mécanisme

L'idée principale derrière cette modélisation consiste à rajouter une liaison fictive entre le maillon de base et l'organe terminal (figure II.27). Cette liaison est une liaison libre, elle dépend de six paramètres cinématiques qui représentent le torseur cinématique de l'organe terminal par rapport au repère lié au maillon de base.

En se basant sur le graphe des liaisons, nous générons les équations cinématiques correspondants au mécanisme. Ces équations sont données en écrivant la fermeture cinématique des cycles indépendants. La différence dans notre cas, par rapport à la théorie des mécanismes classique, est que nous avons une liaison fictive. Cette liaison fictive rajoute six inconnues cinématiques au système d'équations, mais n'influe pas sur les résultats puisqu'il y a un cycle indépendant supplémentaire qui apparaît avec cette liaison. Le cycle supplémentaire rajoute six équations au système d'équations cinématiques.

Les équations cinématiques sont écrites sous la forme matricielle suivante :

$$\mathbf{E}_c\dot{\mathbf{X}} = 0 \tag{II.17}$$

Où $\dot{\mathbf{X}}$ est le vecteur contenant toutes les variables cinématiques et $\mathbf{E}_c = \dfrac{\partial F}{\partial \mathbf{X}}$.

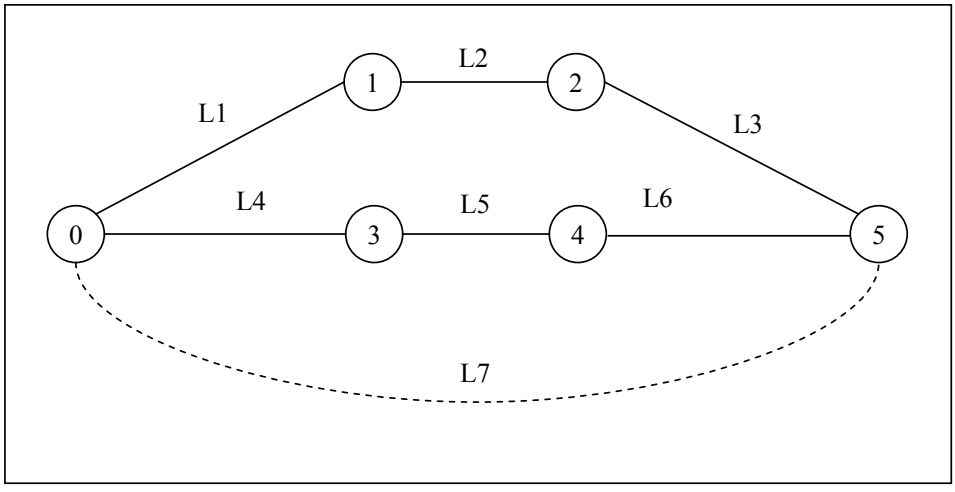


Figure II.27. Modélisation cinématique

Dans les autres méthodes que nous avons présentées, nous obtenons des équations géométriques. Ensuite, nous dérivons ces équations par rapport au temps pour obtenir les équations cinématiques. La génération des équations et la dérivation sont effectués de manière automatique dans SMAR. Dans cette méthode, les équations cinématiques sont obtenues directement. Des difficultés apparaissent avec cette méthode. La première difficulté consiste à retrouver une position initiale du mécanisme à partir du graphe des liaisons qui revient à retrouver les variables géométriques du système à partir des équations cinématiques. Nous n'avons pas retenu cette méthode à cause de ce problème.

2 Analyse des équations

La modélisation des manipulateurs sous SMAR en utilisant les contraintes, permet d'obtenir des équations à respecter pour garder la fermeture des chaînes cinématiques. Ces équations font intervenir les variables passives, les variables actives et les variables opérationnelles du manipulateur. On peut écrire les équations générées par SMAR sous la forme suivante :

$$F(\mathbf{X}) = 0 \tag{II.18}$$

Où F est une fonction vectorielle de dimension n où n est le nombre d'équations et $\mathbf{X}$ le vecteur de toutes les variables du système. On note N le nombre de variables. Ces variables sont ensuite classées en r variables opérationnelles $\mathbf{X}_{op}$, p variables passives $\mathbf{q}_p$ et a variables actives $\mathbf{q}_a$.

En dérivant l'équation (II.18) par rapport au temps, on obtient une équation cinématique de la manière suivante :

$$\mathbf{E}_c\dot{\mathbf{X}} = 0 \tag{II.19}$$

Avec $\mathbf{E}_c$ est la matrice Jacobienne de $F(\mathbf{X})$.

En réarrangeant l'équation (II.19), en séparant les variables en variables actives, passives et opérationnelles, nous obtenons une équation de la forme suivante :

$$\mathbf{M}\dot{\mathbf{q}}_a + \mathbf{N}\dot{\mathbf{q}}_p + \mathbf{K}\dot{\mathbf{X}}_{op} = 0 \tag{II.20}$$

Où $\mathbf{M}$ est une matrice $n \times a$, $\mathbf{N}$ est une matrice $n \times p$ et $\mathbf{K}$ est une matrice $n \times r$

Les matrices nous permettent de calculer la mobilité du système, de classer le système en système redondant ou système sous actionné et nous permettent de juger de la pertinence du choix des variables opérationnelles et des variables actionnées. Les singularités sont détectées à partir des matrices $\mathbf{M}$, $\mathbf{N}$ et $\mathbf{K}$. Nous montrons comment on obtient le modèle cinématique direct et le modèle cinématique inverse à partir de ces équations.

2.1 Calcul de la mobilité

La mobilité m du système est définie de la manière suivante :

$$m = N - rg(\mathbf{E}_c) \tag{II.21}$$

La mobilité m correspond à la mobilité utile définie par les variables actionnées $\dot{\mathbf{q}}_a$ et à la mobilité interne. Ainsi on a :

$$m \geq a \tag{II.22}$$

Cette contrainte doit être satisfaite pour pouvoir commander le robot. Le cas contraire indique qu'il y a un problème de conception ou bien une position singulière à éviter.

2.2 Etat du système

L'état du système est déterminé en examinant les matrices $\mathbf{M}$, $\mathbf{N}$ et $\mathbf{K}$.

Puisqu'on considère que les variables opérationnelles ne peuvent pas varier si on bloque les variables actives, la matrice $\mathbf{K}$ doit être de rang égal au nombre de variables opérationnelles.

$$rg(\mathbf{K}) = r \tag{II.23}$$

Le système est redondant si on arrive à modifier les actionneurs entre eux sans modifier les autres variables:

$$rg(\mathbf{M}) < a \tag{II.24}$$

Il est sous actionné si :

$$rg(\mathbf{M}) < r \tag{II.25}$$

L'équation (II.25) revient à dire que dans le cas où le système est sous actionné, il existe plus de variables opérationnelles que ce qu'il en faut pour décrire le mouvement généré par les actionneurs.

L'équation suivante doit être satisfaite :

$$rg([\mathbf{NK}]) = rg(\mathbf{N}) + r \tag{II.26}$$

Dans le cas où l'équation (II.26) n'est pas satisfaite, les variables opérationnelles ne sont pas indépendantes des variables passives, c'est dire que les variables opérationnelles peuvent bouger lorsque les variables actives sont bloquées. Ce cas est dangereux pour le mécanisme il correspond à un cas de singularité. Nous pouvons faire le même raisonnement entre les variables actives et les variables passives :

$$rg([\mathbf{MN}]) = rg(\mathbf{M}) + rg(\mathbf{N}) \tag{II.27}$$

La condition (II.27) donne le cas ou les variables actives et les variables passives sont indépendantes. Si cette condition n'est pas satisfaite, cela correspond à des singularités, mais ces singularités n'empêchent pas le mécanisme de fonctionner correctement comme c'est le cas des mécanismes redondants.

Si l'une des équations (II.23) ou (II.26) n'est pas satisfaite, cela indique qu'il y a soit un problème de singularité, soit un problème de classement des variables en variables actives, variables passive ou variables opérationnelles.

Une perte de rang de l'une des matrices $\mathbf{M}$, $\mathbf{N}$ ou $\mathbf{K}$ correspond à une singularité, cela signifie qu'au moins l'une des variables du système peut varier de manière non contrôlée.

2.3 Elimination des variables passives

L'élimination des variables passives permet de déterminer le modèle cinématique direct ou le modèle cinématique inverse sous sa forme classique. Cela permet de trouver les variables actionnées en fonction des variables opérationnelles pour le modèle cinématique inverse. Ainsi, on peut simuler le mouvement du robot. Pour éliminer les variables passives, nous cherchons une matrice $\mathbf{O}$ telle que :

$$\mathbf{O}^T\mathbf{N} = 0 \text{ et } rg\left(\mathbf{O}^T\mathbf{K}\right) = rg\left(\mathbf{K}\right) \tag{II.28}$$

La matrice $\mathbf{O}$ est construite en faisant une orthogonalisation de Gramm-Schmidt. En utilisant cette orthogonalisation on construit des vecteurs générés par les colonnes de la matrice $\mathbf{K}$ et orthogonaux aux colonnes de la matrice $\mathbf{N}$. La matrice $\mathbf{O}$ n'existe que si $\mathbf{K}$ et $\mathbf{N}$ sont indépendantes c'est-à-dire l'équation (II.26) est vérifiée. Dans le cas contraire, il y a une variation des variables opérationnelles et des variables passives sans avoir de variation des variables actives, ce qui correspond à une singularité ou à un problème de conception.

En multipliant l'équation (II.20) par la matrice $\mathbf{O}^T$ nous obtenons l'équation suivante :

$$\mathbf{B}\dot{\mathbf{q}}_a + \mathbf{A}\dot{\mathbf{X}}_{op} = 0 \tag{II.29}$$

avec

$$\mathbf{B} = \mathbf{O}^T\mathbf{M} \tag{II.30}$$

$$\mathbf{A} = \mathbf{O}^T\mathbf{K} \tag{II.31}$$

La matrice $\mathbf{B}$ est carrée et inversible dans le cas où le robot n'est pas redondant et qu'il n'est pas dans une configuration singulière. Pour les robots redondants on peut utiliser la pseudo inverse de la manière suivante :

$$\dot{\mathbf{q}}_a = \mathbf{G}^+\dot{\mathbf{X}}_{op} \tag{II.32}$$

avec

$$\mathbf{G}^{+} = \mathbf{B}^{T}\left(\mathbf{B} \quad \mathbf{B}^{T}\right)^{-1}\mathbf{A} \tag{II.33}$$

Pour trouver les variables passives, il suffit de résoudre l'équation (II.20) en injectant les variables actives et les variables opérationnelles.

3 Exemples

Pour illustrer de façon pratique les notions développées dans le paragraphe précédent, nous utilisons quelques exemples de robot modélisés sous SMAR et nous donnons les équations de fermeture générées.

3.1 Robot 2-RPR

Un robot 2-RPR est un robot parallèle plan simple composé de deux jambes contenant chacune une liaison rotoïde, une liaison prismatique et une deuxième liaison rotoïde (voir figure II.28).

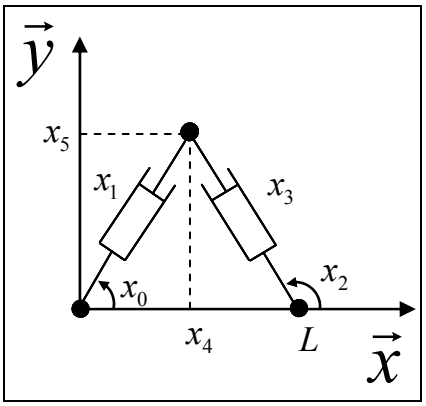


Figure II.28. Robot 2-RPR

Pour modéliser le robot 2-RPR on utilise la première méthode. Ce robot contient un maillon base auquel sont reliés deux maillons séries qui correspondent aux deux liaisons rotoïdes passives. Les liaisons passives correspondent aux variables x_0 et x_2. A chacun des deux maillons séries est relié un nouveau maillon série correspondant à l'actionneur. Les deux nouveaux maillons séries correspondent aux liaisons prismatiques de variables x_1 et x_3. Pour modéliser l'organe terminal nous rajoutons un maillon mobile. Dans ce cas, nous n'avons

besoin que de deux variables (x_4 et x_5) pour positionner le maillon mobile, car il a un mouvement dans le plan et on n'a pas besoin de la rotation. Les autres variables du maillon mobile ne vont pas intervenir dans les équations. Pour générer les équations de fermeture de chaînes, on utilise deux contraintes de coïncidence entre deux points. Ces contraintes font intervenir le maillon libre et les deux maillons séries des vérins. Nous obtenons alors les équations suivantes :

$$\begin{cases} x_1 \cos x_0 - x_4 = 0 \\ x_1 \sin x_0 - x_5 = 0 \\ x_3 \cos x_2 - x_4 + L = 0 \\ x_3 \sin x_2 - x_5 = 0 \end{cases} \tag{II.34}$$

En dérivant ces équations on obtient :

$$\begin{bmatrix} c_0 & 0 \\ s_0 & 0 \\ 0 & c_2 \\ 0 & s_2 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_3 \end{Bmatrix} + \begin{bmatrix} -x_1 s_0 & 0 \\ x_1 c_0 & 0 \\ 0 & -x_3 s_2 \\ 0 & x_3 c_2 \end{bmatrix} \begin{Bmatrix} \dot{x}_0 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{Bmatrix} \dot{x}_4 \\ \dot{x}_5 \end{Bmatrix} = 0 \tag{II.35}$$

Pour ce cas on trouve facilement une matrice $\mathbf{O}$ sans passer par l'orthogonalisation de Gramm-Schmidt. La matrice $\mathbf{O}^T$ s'écrit :

$$\mathbf{O}^T = \begin{bmatrix} c_0 & s_0 & 0 & 0 \\ 0 & 0 & c_2 & s_2 \end{bmatrix} \tag{II.36}$$

En multipliant l'équation (II.35) par la matrice $\mathbf{O}^T$, nous obtenons :

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{II.37}$$

$$\mathbf{A} = \begin{bmatrix} -c_0 & -s_0 \\ -c_2 & -s_2 \end{bmatrix} \tag{II.38}$$

La manière classique d'obtenir ces résultats consiste à réarranger l'équation (II.34) de telle sorte à éliminer les variables passives.

$$\begin{cases} {x_1}^2 = {x_4}^2 + {x_5}^2 \\ {x_3}^2 = (x_4 - L)^2 + {x_5}^2 \end{cases} \tag{II.39}$$

En dérivant l'équation (II.39), on obtient les mêmes résultats donnés par les équations (II.37) et (II.38). On remarque que :

$$c_0 = \frac{x_4}{x_1}; s_0 = \frac{x_5}{x_1}; c_2 = \frac{x_4 - L}{x_3}; s_2 = \frac{x_5}{x_3} \tag{II.40}$$

En passant par l'orthogonalisation de Gramm-Schmidt, on obtient la matrice $\mathbf{O}_g$ suivante :

$$\mathbf{O}_g = \begin{bmatrix} c_0^2 & -s_0^3 c_0 + s_2 c_2 c_0^2 \\ c_0 s_0 & -s_0^4 + s_2 c_2 s_0 c_0 \\ c_2^2 & -s_2^3 c_2 + s_0 c_0 c_2^2 \\ c_2 s_2 & -s_2^4 + s_0 c_0 s_2 c_2 \end{bmatrix} \tag{II.41}$$

En multipliant (II.35) par la matrice ${\mathbf{O}_g}^T$, nous obtenons :

$$\mathbf{B}_g = \begin{bmatrix} c_0 & c_2 \\ c_0 c_2 s_2 - s_0^3 & c_0 c_2 s_0 - s_2^3 \end{bmatrix} \tag{II.42}$$

$$\mathbf{A}_g = \begin{bmatrix} -c_0^2 - c_2^2 & -s_0 c_0 - s_2 c_2 \\ s_0^3 c_0 - s_2 c_2 c_0^2 + s_2^3 c_2 - s_0 c_0 c_2^2 & s_0^4 - 2 s_2 c_2 s_0 c_0 + s_2^4 \end{bmatrix} \tag{II.43}$$

Pour retrouver l'équation (II.38) il suffit de multiplier l'équation (II.43) par la matrice ${\mathbf{B}_g}^{-1}$.

$$\mathbf{B}_g^{-1} = \frac{1}{c_2 s_0 - s_2 c_0} \begin{bmatrix} c_0 c_2 s_0 - s_2^3 & -c_2 \\ -c_0 c_2 s_2 + s_0^3 & c_0 \end{bmatrix} \tag{II.44}$$

3.2 Robot RR-RRR

La figure II.29 montre le robot RR-RRR plan modélisé sous SMAR. Ce robot est en fait un mécanisme à 5 barres. Le graphe de liaisons de ce robot est donné par la figure II.30. Pour modéliser le robot RR-RRR nous avons utilisé la première méthode de modélisation des robots parallèles. Le maillon 0 est un maillon base et le maillon 5 est un maillon mobile, il dépend de 3 paramètres (x, y, θ) qui permettent de le positionner par rapport à la base. Les liaisons L3 et L6 sont choisies comme étant des liaisons de fermeture. L3 est remplacée par une première contrainte de coïncidence entre le point $(l,0)$ lié au repère du maillon 2 et le point $(0,0)$ origine du repère lié au maillon 5. L6 est remplacée par une deuxième contrainte de coïncidence entre le point $(l,0)$ lié au repère du maillon 4 et le point $(0,0)$ origine du repère lié au maillon 5. Les liaisons L1, L2, L4 et L5 sont des liaisons rotoïdes définies par le tableau II.1. Le point de la liaison est donné par rapport au repère lié au maillon père.

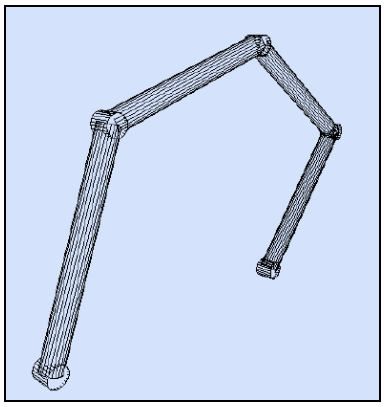


Figure II.29. RR-RRR modélisé sous SMAR

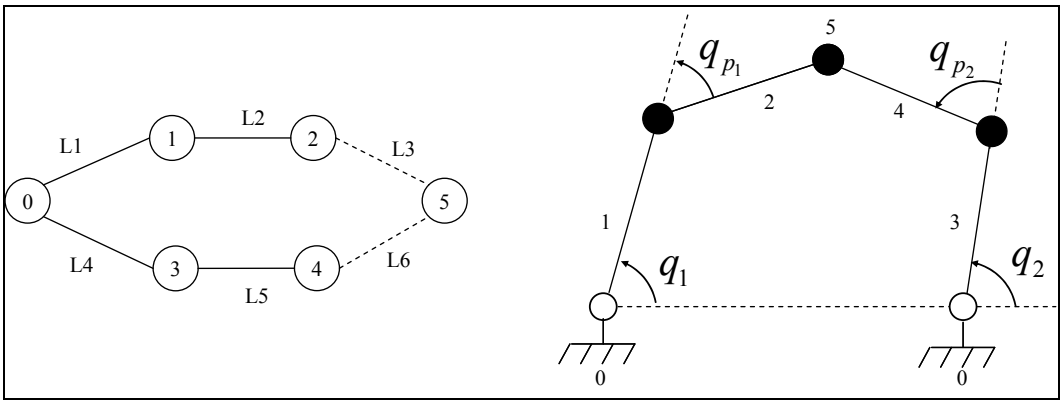


Figure II.30. Graphe de liaisons du robot RR-RRR

Table II.1 Définition des liaisons rotoïdes

Liaison	Maillon père	Maillon série	Point de liaison	Paramètre de la liaison
L1	0	1	$(0,0)$	q_1
L2	1	2	$(l,0)$	q_{p1}
L4	0	3	$(L,0)$	q_2
L5	3	4	$(l,0)$	q_{p2}

La contrainte donnée par la liaison L3 s'écrit

$$ {}^0\mathbf{T}_2 \begin{pmatrix} l \\ 0 \\ 0 \\ 1 \end{pmatrix} = {}^0\mathbf{T}_5 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{II.45} $$

Or

$$ {}^0\mathbf{T}_2 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \tag{II.46} $$

Ce qui donne

$$\begin{cases} x = l\cos q_1 + l\cos\left(q_1 + q_{p1}\right) \\ y = l\sin q_1 + l\sin\left(q_1 + q_{p1}\right) \end{cases} \tag{II.47}$$

La contrainte donnée par la liaison L6 s'écrit :

$${}^0\mathbf{T}_4 \begin{pmatrix} l \\ 0 \\ 0 \\ 1 \end{pmatrix} = {}^0\mathbf{T}_5 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{II.48}$$

Or

$${}^0\mathbf{T}_4 = {}^0\mathbf{T}_3\,{}^3\mathbf{T}_4 \tag{II.49}$$

Ce qui donne :

$$\begin{cases} x = l\cos q_2 + l\cos\left(q_2 + q_{p2}\right) + L \\ y = l\sin q_2 + l\sin\left(q_2 + q_{p2}\right) \end{cases} \tag{II.50}$$

Le modèle géométrique du robot est donné par le système d'équations suivant :

$$\begin{cases} l\cos q_1 + l\cos\left(q_1 + q_{p1}\right) - x = 0 \\ l\sin q_1 + l\sin\left(q_1 + q_{p1}\right) - y = 0 \\ l\cos q_2 + l\cos\left(q_2 + q_{p2}\right) + L - x = 0 \\ l\sin q_2 + l\sin\left(q_2 + q_{p2}\right) - y = 0 \end{cases} \tag{II.51}$$

En dérivant l'équation (II.47), nous obtenons les matrices suivantes :

$$\mathbf{M} = \begin{bmatrix} -l\left(\sin q_1 + \sin\left(q_1 + q_{p1}\right)\right) & 0 \\ l\left(\cos q_1 + \cos\left(q_1 + q_{p1}\right)\right) & 0 \\ 0 & -l\left(\sin q_2 + \sin\left(q_2 + q_{p2}\right)\right) \\ 0 & l\left(\cos q_2 + \cos\left(q_2 + q_{p2}\right)\right) \end{bmatrix} \tag{II.52}$$

$$\mathbf{N} = \begin{bmatrix} -l\sin\left(q_1 + q_{p1}\right) & 0 \\ l\cos\left(q_1 + q_{p1}\right) & 0 \\ 0 & -l\sin\left(q_2 + q_{p2}\right) \\ 0 & l\cos\left(q_2 + q_{p2}\right) \end{bmatrix} \tag{II.53}$$

$$\mathbf{K} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{II.54}$$

On peut trouver une matrice $\mathbf{O}$ qui satisfait l'équation (II.28) sans passer par la méthode d'orthogonalisation.

$$\mathbf{O}^T = \begin{bmatrix} \cos\left(q_1 + q_{p1}\right) & \sin\left(q_1 + q_{p1}\right) & 0 & 0 \\ 0 & 0 & \cos\left(q_2 + q_{p2}\right) & \sin\left(q_2 + q_{p2}\right) \end{bmatrix} \tag{II.55}$$

On obtient alors les résultats suivants :

$$\mathbf{B} = \begin{bmatrix} l\sin q_{p1} & 0 \\ 0 & l\sin q_{p2} \end{bmatrix} \tag{II.57}$$

$$\mathbf{A} = \begin{bmatrix} -\cos\left(q_1 + q_{p1}\right) & -\sin\left(q_1 + q_{p1}\right) \\ -\cos\left(q_2 + q_{p2}\right) & -\sin\left(q_2 + q_{p2}\right) \end{bmatrix} \tag{II.58}$$

Ces résultats sont plus simples à obtenir avec cette méthode que dans le cas où on procède à une élimination des variables passives avant de dériver par rapport au temps. Nous obtenons les mêmes résultats dans les deux cas.

3.3 Robot 3-RPR

La figure II.31 définit le robot 3-RPR plan que nous avons modélisé sous SMAR. La figure II.32 donne le graphe de liaisons du robot 3-RPR. Le tableau II.2 définit les liaisons. Nous avons pris comme liaisons de fermeture les liaisons L3, L6 et L9. Le maillon 0 est un maillon base alors que le maillon 7 est un maillon mobile. Le tableau II .3 définit les contraintes de coïncidences qui remplacent les liaisons de fermeture.

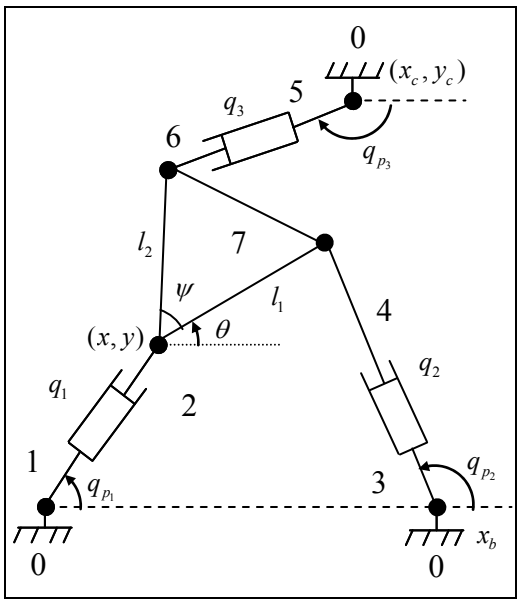


Figure II.31. Robot 3-RPR

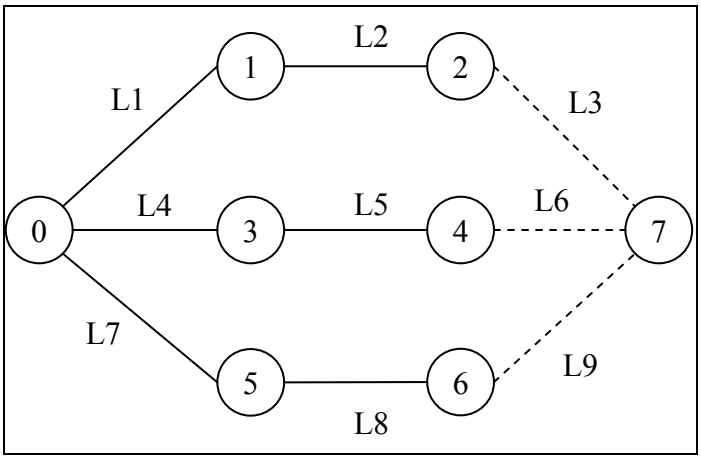


Figure II.32. Graphe de liaisons du robot 3-RPR

Table II.2 Définition des liaisons du robot 3-RPR

Liaison	Type	Maillon père	Maillon série	Point de liaison	Paramètre de la liaison
L1	Rotoïde	0	1	$(0,0)$	q_1
L2	Prismatique	1	2	$(0,0)$	q_{p1}
L4		0	3	$(x_b,0)$	q_2
L5	Prismatique	3	4	$(0,0)$	q_{p2}
L7		0	5	(x_c,y_c)	q_3
L8	Prismatique	5	6	$(0,0)$	q_{p3}

Table II.3 Contraintes de coïncidences entre deux points

Liaison	Maillon i	Maillon j	P^i	P^i
L3	2	7	$(0,0)$	$(0,0)$
L6	4	7	$(0,0)$	$(l_1,0)$
L9	6	7	$(0,0)$	$(l_2\cos\psi, l_2\sin\psi)$

La contrainte de coïncidence correspondante à la liaison L3, s'écrit de la manière suivante :

$$ {}^0\mathbf{T}_2\begin{pmatrix}0\\0\\0\\1\end{pmatrix}={}^0\mathbf{T}_7\begin{pmatrix}0\\0\\0\\1\end{pmatrix} \tag{II.59}$$

Ce qui nous donne les équations suivantes :

$$\begin{cases} x = q_1\cos q_{p1} \\ y = q_1\sin q_{p1} \end{cases} \tag{II.60}$$

La contrainte de coïncidence correspondante à la liaison L6, s'écrit :

$$ {}^{0}\mathbf{T}_{4}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}={}^{0}\mathbf{T}_{7}\begin{pmatrix}l_1\\0\\0\\1\end{pmatrix} \tag{II.61} $$

Nous obtenons ces équations :

$$ \begin{cases} x+l_1\cos\theta = q_2\cos q_{p2}+x_b \\ y+l_1\sin\theta = q_2\sin q_{p2} \end{cases} \tag{II.62} $$

La contrainte de coïncidence correspondante à la liaison L9, s'écrit :

$$ {}^{0}\mathbf{T}_{6}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}={}^{0}\mathbf{T}_{7}\begin{pmatrix}l_2\cos\psi\\l_2\sin\psi\\0\\1\end{pmatrix} \tag{II.63} $$

Ce qui donne :

$$ \begin{cases} x+l_2\cos(\theta+\psi) = q_3\cos q_{p3}+x_c \\ y+l_2\sin(\theta+\psi) = q_3\sin q_{p3}+y_c \end{cases} \tag{II.64} $$

Le modèle géométrique s'écrit alors :

$$ \begin{cases} q_1\cos q_{p1}-x=0 \\ q_1\sin q_{p1}-y=0 \\ q_2\cos q_{p2}+x_b-x-l_1\cos\theta=0 \\ q_2\sin q_{p2}-y-l_1\sin\theta=0 \\ q_3\cos q_{p3}+x_c-x-l_2\cos(\theta+\psi)=0 \\ q_3\sin q_{p3}+y_c-y-l_2\sin(\theta+\psi)=0 \end{cases} \tag{II.65} $$

En dérivant l'équation (II.65) nous obtenons les résultats suivants :

$$\mathbf{M} = \begin{bmatrix} \cos q_{p1} & 0 & 0 \\ \sin q_{p1} & 0 & 0 \\ 0 & \cos q_{p2} & 0 \\ 0 & \sin q_{p2} & 0 \\ 0 & 0 & \cos q_{p3} \\ 0 & 0 & \sin q_{p3} \end{bmatrix} \tag{II.66}$$

$$\mathbf{N} = \begin{bmatrix} -q_1 \sin q_{p1} & 0 & 0 \\ q_1 \cos q_{p1} & 0 & 0 \\ 0 & -q_2 \sin q_{p2} & 0 \\ 0 & q_2 \cos q_{p2} & 0 \\ 0 & 0 & -q_3 \sin q_{p3} \\ 0 & 0 & q_3 \cos q_{p3} \end{bmatrix} \tag{II.67}$$

$$\mathbf{K} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & l_1 \sin\theta \\ 0 & -1 & -l_1 \cos\theta \\ -1 & 0 & l_2 \sin(\theta + \psi) \\ 0 & -1 & -l_2 \cos(\theta + \psi) \end{bmatrix} \tag{II.68}$$

On trouve facilement une matrice $\mathbf{O}$ qui satisfait l'équation (II.28).

$$\mathbf{O}^T = \begin{bmatrix} \cos q_{p1} & \sin q_{p1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos q_{p2} & \sin q_{p2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos q_{p3} & \sin q_{p3} \end{bmatrix} \tag{II.69}$$

En multipliant l'équation (II.20) par $\mathbf{O}^T$ nous obtenons les résultats suivants :

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{II.70}$$

$$\mathbf{A} = \begin{bmatrix} -\cos q_{p1} & -\sin q_{p1} & 0 \\ -\cos q_{p2} & -\sin q_{p2} & l_1 \sin(\theta - q_{p2}) \\ -\cos q_{p3} & -\sin q_{p3} & l_2 \sin(\theta + \psi - q_{p3}) \end{bmatrix} \tag{II.71}$$

Ces résultats correspondent aux résultats trouvés dans l'équation (I.32).

3.4 Plateforme de Stewart

La plateforme de Stewart est montrée par la figure II.59. La figure II.60 donne le graphe de liaison. Le maillon 0 est un maillon base et le maillon 13 est un maillon mobile. Nous définissons, pour chaque jambe i allant de 1 à 6 du robot, les liaisons dans le tableau II.4.

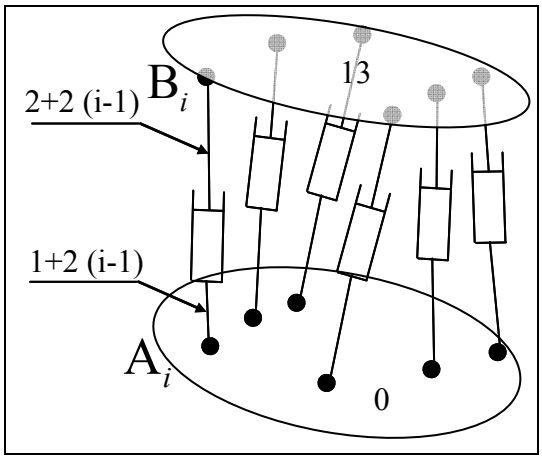


Figure II.33. Plateforme de Stewart

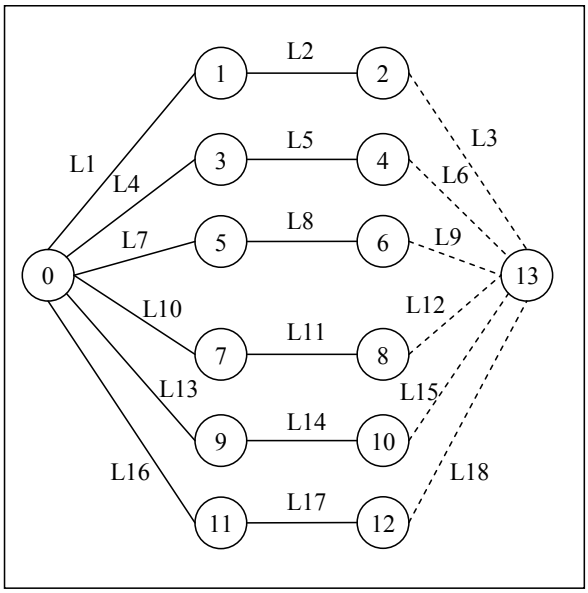


Figure II.34. Graphe de liaisons de la plateforme de Stewart

Table II.4 Liaisons de la plateforme de Stewart

Liaison	Type	Maillon père	Maillon série	Point de liaison	Paramètre de la liaison
L3i-2	Sphérique	0	2i-1	(a_{ix}, a_{iy}, a_{iz})	$(q_{pix}, q_{piy}, q_{piz})$
L3i-1	Prismatique	2i-1	2i	$(0,0,0)$	q_i

Pour chaque jambe i, la liaison L3i est une liaison de fermeture, elle est remplacée par une contrainte de coïncidence entre le point $(0,0,0)$ du maillon 2i et le point (b_{ix}, b_{iy}, b_{iz}) du maillon 13. Cette contrainte s'écrit :

$$ {}^{0}\mathbf{T}_{2i} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = {}^{0}\mathbf{T}_{13} \begin{pmatrix} b_{ix} \\ b_{iy} \\ b_{iz} \\ 1 \end{pmatrix} \tag{II.72} $$

Ce qui nous donne pour chaque jambe i l'équation suivante :

$$\begin{cases} a_{ix} + q_i \sin q_{piy} = x + b_{ix}\cos\beta\cos\gamma - b_{iy}\cos\beta\sin\gamma + b_{iz}\sin\beta \\ a_{iy} - q_i \sin q_{pix}\cos q_{piy} = y + b_{ix}(\sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma) + \\ \qquad b_{iy}(-\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) + \\ \qquad b_{iz}\sin\alpha\cos\beta \\ a_{iz} - q_i \cos q_{pix}\cos q_{piy} = z + b_{ix}(-\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + \\ \qquad b_{iy}(\cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma) + \\ \qquad b_{iz}\cos\alpha\cos\beta \end{cases} \quad \text{(II.73)}$$

En dérivant les équations données par toutes les jambes, nous obtenons les matrices $\mathbf{M}$, $\mathbf{N}$ et $\mathbf{K}$ de dimensions respectives 18×6, 18×12 et 18×6. Les résultats sont donnés dans l'annexe A.

Pour trouver les matrices $\mathbf{A}$ et $\mathbf{B}$, il suffit de trouver une matrice $\mathbf{O}$ qui vérifie l'équation (II.28). On peut prendre dans ce cas $\mathbf{O} = \mathbf{M}$. Les résultats sont donnés dans l'annexe A.

Conclusion

Dans ce chapitre nous avons présenté le logiciel de CAO robotique SMAR que nous avons développé pour la modélisation et la simulation des robots parallèles. Ce logiciel permet de modéliser une large gamme de mécanismes à chaînes cinématiques ouvertes ou fermées. Nous avons montré comment les équations sont calculées à partir des contraintes utilisées lors de la modélisation. En se basant sur la notion des contraintes, nous avons proposé deux méthodes de modélisation des robots parallèles, ainsi qu'une méthode de modélisation des robots sériels. Nous avons proposé aussi une méthode permettant de donner les équations cinématiques correspondant à un graphe de liaisons d'un mécanisme. Les équations générées lors de la modélisation sont ensuite analysées et utilisées pour la simulation. En utilisant ces équations, nous pouvons juger de la nature du mécanisme s'il est redondant ou sous actionné et nous pouvons détecter les configurations singulières. Les singularités détectées prennent en considération toutes les variables du système y compris les articulations passives. Nous avons donné aussi une méthode permettant d'éliminer systématiquement les variables opérationnelles afin de retrouver les équations cinématiques classiques données au cas par cas aux robots parallèles. Nous avons illustré à travers quelques exemples la similitude entre les résultats obtenus avec notre méthode et les résultats classiques des équations cinématiques.

Chapitre III

Planification de trajectoire des robots parallèles

Dans le chapitre II, nous avons montré comment modéliser les robots parallèles. Nous avons établi les équations à respecter pour garder une structure cohérente lors de la simulation de mouvement des robots parallèles. Nous avons donné les équations à satisfaire pour éviter les configurations singulières. Dans ce chapitre, nous attaquons le problème de planification de trajectoire.

Introduction

La plupart des études de planification de trajectoire des robots sériels se fait dans l'espace des configurations depuis les travaux d'Udupa [Udupa77]. L'utilisation de l'espace des configurations, dans le cas des robots sériels, est due à la simplicité d'obtention du modèle géométrique direct. Les obstacles de l'espace de travail, sont facilement transformés dans l'espace des configurations, mais demandent du temps. Un robot correspond à un point dans l'espace des configurations (figure III.1).

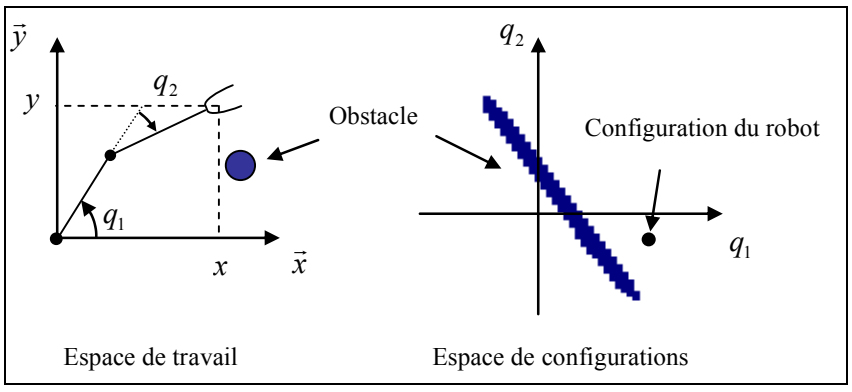


Figure III.1. Transformation des obstacles dans l'espace des configurations

L'utilisation de l'espace des configurations n'est pas justifiée dans le cas des robots parallèles vu que le modèle géométrique inverse est plus simple à obtenir. Nous proposons d'utiliser l'espace de travail dans le cas des robots parallèles en ramenant les obstacles à l'organe terminal (figure III.2). C'est-à-dire modéliser l'espace de travail de telle sorte que les parties libres de cet espace sont accessibles hors collision. Le fait de ramener les obstacles à

l'organe terminal permet de simplifier la planification de trajectoire et d'éviter que l'une des pièces du robot touche l'obstacle.

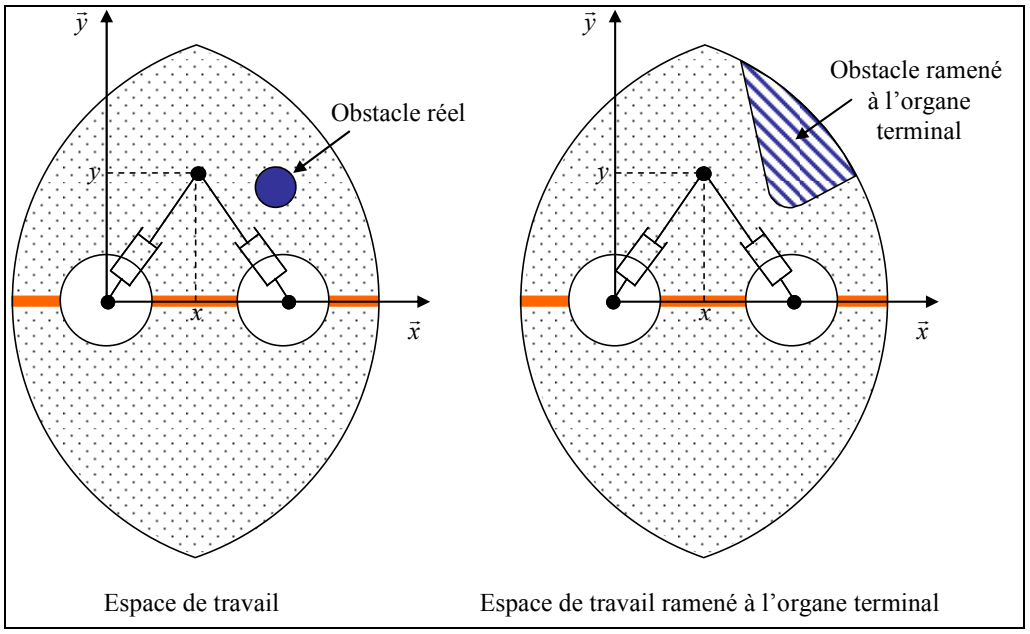


Figure III.2. Espace de travail

1 Méthode de planification

1.1 Adaptation des méthodes de planification au cas des robots parallèles

Il existe plusieurs méthodes de planification de trajectoire qui ont été utilisées pour les robots mobiles et pour les robots sériels. Ces méthodes peuvent êtres adaptées aux robots parallèles. Le plus simple c'est d'utiliser l'espace de travail ramené à l'organe terminal, puisque, dans le cas des robots parallèles, le modèle géométrique inverse est plus simple à obtenir que le modèle géométrique direct. Un point de l'espace de travail correspond à une configuration du robot et on peut en juger directement si le robot est en collision avec un obstacle de l'environnement ou non. Ainsi, plusieurs méthodes peuvent êtres utilisées avec les robots parallèles, comme par exemple, les méthodes basées sur les cartes probabilistes qu'on appelle PRM (de l'anglais Probabilistic RoadMaps). Il existe des travaux qui ont réussi à utiliser ce genre de méthodes avec les robots parallèles, tout en travaillant dans l'espace des configurations et en utilisant une méthode qui permet de générer des configurations compatibles aux équations de fermeture de chaînes [Cortès 03]. Le fait de générer des

configurations compatibles rend la planification de trajectoire plus complexe et nécessite plus de temps pour générer les cartes probabilistes.

Pour utiliser la méthode des cartes probabilistes avec les robots parallèles, il suffit juste de générer des positions aléatoires dans l'espace de travail (Figure III.3).

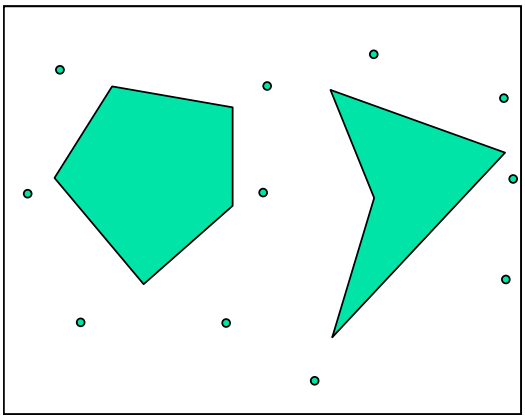


Figure III.3. Génération de positions aléatoires sans collision dans l'espace de travail

Pour chaque nouvelle position, les chemins pour rallier les positions les plus proches sont vérifiés à l'aide d'une méthode locale. Dans le cas où ces chemins ne croisent pas de collision ni de singularité, ils seront placés dans un graphe qui représente l'adjacence des positions (figure III.4).

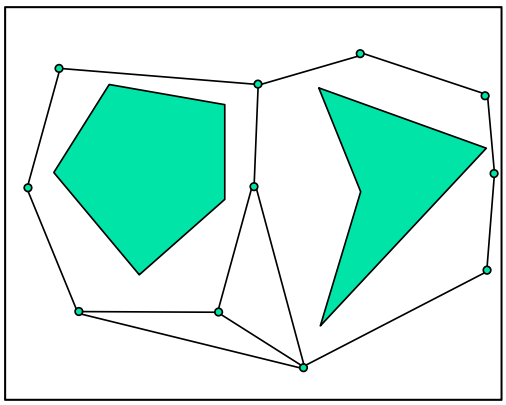


Figure III.4. Construction du PRM

1.2 Méthode choisie

La plupart des méthodes de planification de trajectoires peuvent être utilisées pour le cas des robots parallèles. Nous avons choisi d'adapter une méthode que nous avons utilisée pour le cas des robots sériels placés dans des environnements encombrés [Lahouar 05]. Cette méthode qu'on peut qualifier de méthode mixte, se base sur l'alternance de deux modes. Un

premier mode de recherche en profondeur permet d'évoluer vers le but lorsqu'il n'y a pas d'obstacle, ni de singularité au voisinage du robot. Le deuxième mode est un mode de recherche en largeur. Il est actif au voisinage d'une position singulière ou d'un obstacle. Il permet d'aboutir à la meilleure façon de contourner l'obstacle ou la singularité.

Pour illustrer cette méthode, nous utilisons la figure III.5. Dans cette figure on voit bien les deux modes. Puisqu'il n'y a pas d'obstacle au voisinage de la position A, le mode de recherche en profondeur est utilisé pour avancer vers le but (G) jusqu'à ce que la position B est atteinte. La position B est au voisinage d'un obstacle, on utilise alors le mode de recherche en largeur afin de trouver la meilleure stratégie pour éviter cet obstacle. Dans le cas de la figure III.5, il existe deux stratégies de contournement de l'obstacle. La première stratégie consiste à aller vers le haut et la deuxième consiste à aller vers le bas. Le mode de recherche en largeur explore les deux stratégies jusqu'à atteindre la position C. Dans cette position, l'obstacle ne gène plus le robot pour atteindre son but. La meilleure stratégie dans ce cas, consiste à contourner vers la droite. Dans la position C, le mode de recherche en profondeur est repris jusqu'à la position G.

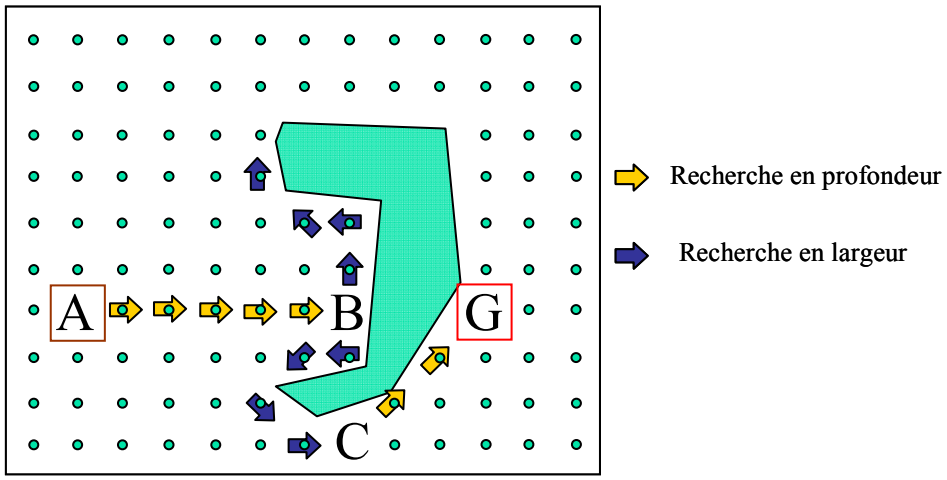


Figure III.5. Alternance entre les modes de recherche en profondeur et en largeur

1.2.1 Mode de recherche en profondeur

A chaque étape de cette méthode, on effectue un test sur la position la plus proche au but du voisinage de la position actuelle du robot. Si cette position ne correspond pas à une collision, ni à une singularité, elle est alors placée à la tête de la file d'attente (figure III.6). Par contre, si cette position correspond à une collision ou une singularité, le mode de recherche en largeur est activé.

Le mode de recherche en profondeur commence par la position initiale A. Il continue ensuite à chaque itération par prendre la position à la tête de la file d'attente. Le fait de rajouter les nouvelles positions à la tête de la file d'attente et d'extraire à chaque fois la position à la tête de la file d'attente aussi, correspond à une file d'attente de type FILO (First In Last Out). Ce type de files d'attentes donne le mode de recherche en profondeur.

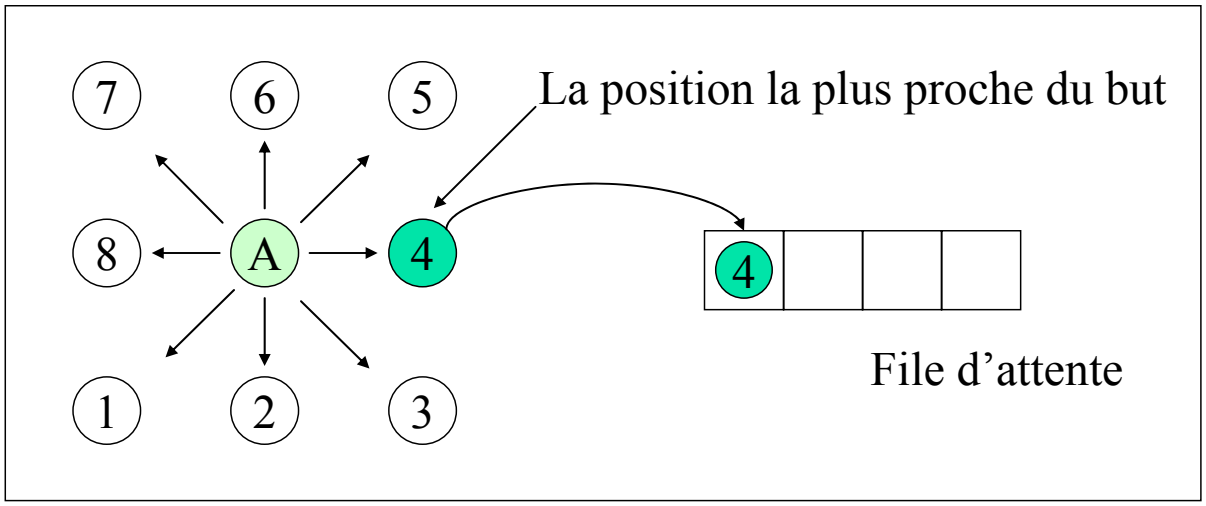


Figure III.6. Mode de recherche en profondeur

1.2.2 Mode de recherche en largeur

La position voisine de la position B et la plus proche du but correspond à une collision ou une singularité. Pour cette raison, le mode de recherche en largeur est actif dans cette position B. Dans ce mode, les positions non encore explorées qui ne correspondent pas à une collision ou à une singularité, qui sont à la fois au voisinage de la position actuelle et au voisinage de l'obstacle, sont placées à la queue de la file d'attente (figure III.7). En ne prenant en compte que les positions au voisinage de l'obstacle (collision ou singularité), on s'assure que le robot suit les limites de l'obstacle jusqu'à ce qu'il le contourne. Cela permet d'éviter de s'attarder sur des positions qui n'aident pas à trouver le meilleur moyen de contourner l'obstacle et qui rendent la recherche de trajectoire plus lente.

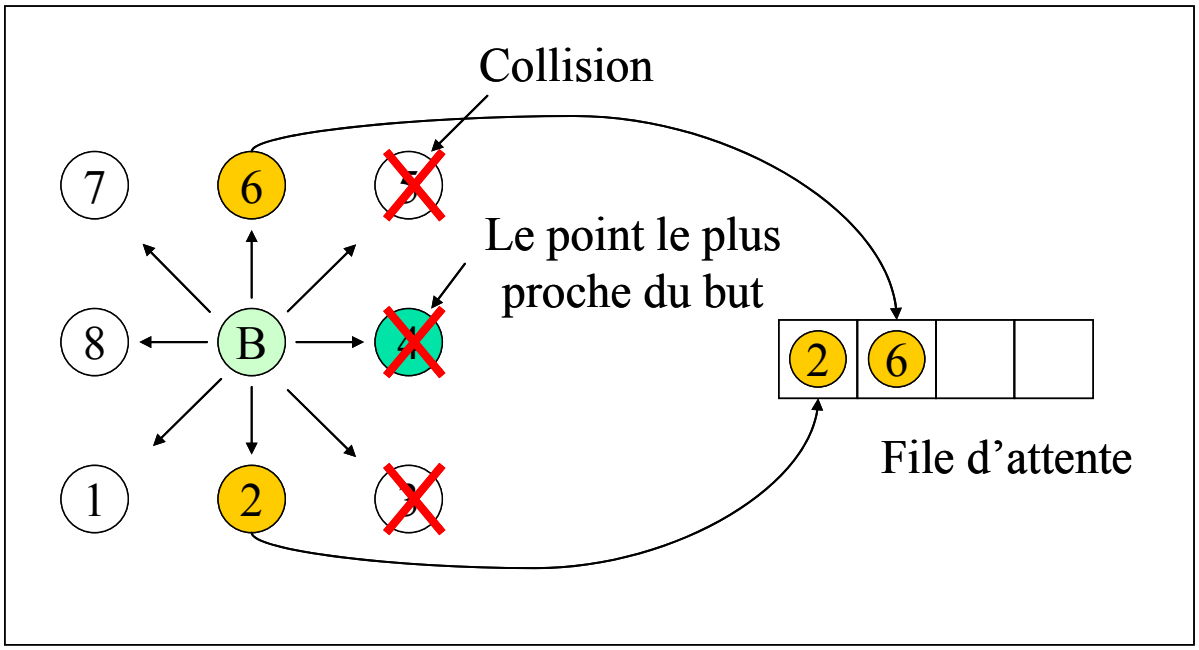


Figure III.7. Mode de recherche en largeur

Le fait de placer les nouvelles positions à la queue de la file d'attente et de retirer toujours la position à traiter à la tête de la file, correspond à une file d'attente de type FIFO (First In First Out). Ce qui donne le mode de recherche en largeur.

1.2.3 Algorithme de la méthode

L'algorithme de la méthode est illustré par le diagramme de la figure III.8. En entrée, cet algorithme prend une position initiale et une position finale. Il commence par placer la position initiale dans une file d'attente. L'algorithme est arrêté soit quand la file d'attente devient vide dans ce cas il n'y a pas de chemin possible, soit lorsqu'une position proche du but est atteinte dans ce cas une trajectoire est trouvée. A chaque fois l'algorithme place des positions dans la file d'attente en fonction du mode de recherche en profondeur ou en largeur. Chaque nouvelle position placée dans la file d'attente pointe vers la position qui la génére. Lorsqu'une position proche du but est atteinte, il suffit de remonter les positions générées jusqu'à atteindre la position initiale pour avoir les positions formant la trajectoire. Pour améliorer l'allure des trajectoires générées par l'algorithme, on refait la planification de trajectoire entre la position initiale, la position finale et les positions où l'algorithme change de mode de recherche : du mode de recherche en largeur au mode de recherche en profondeur.

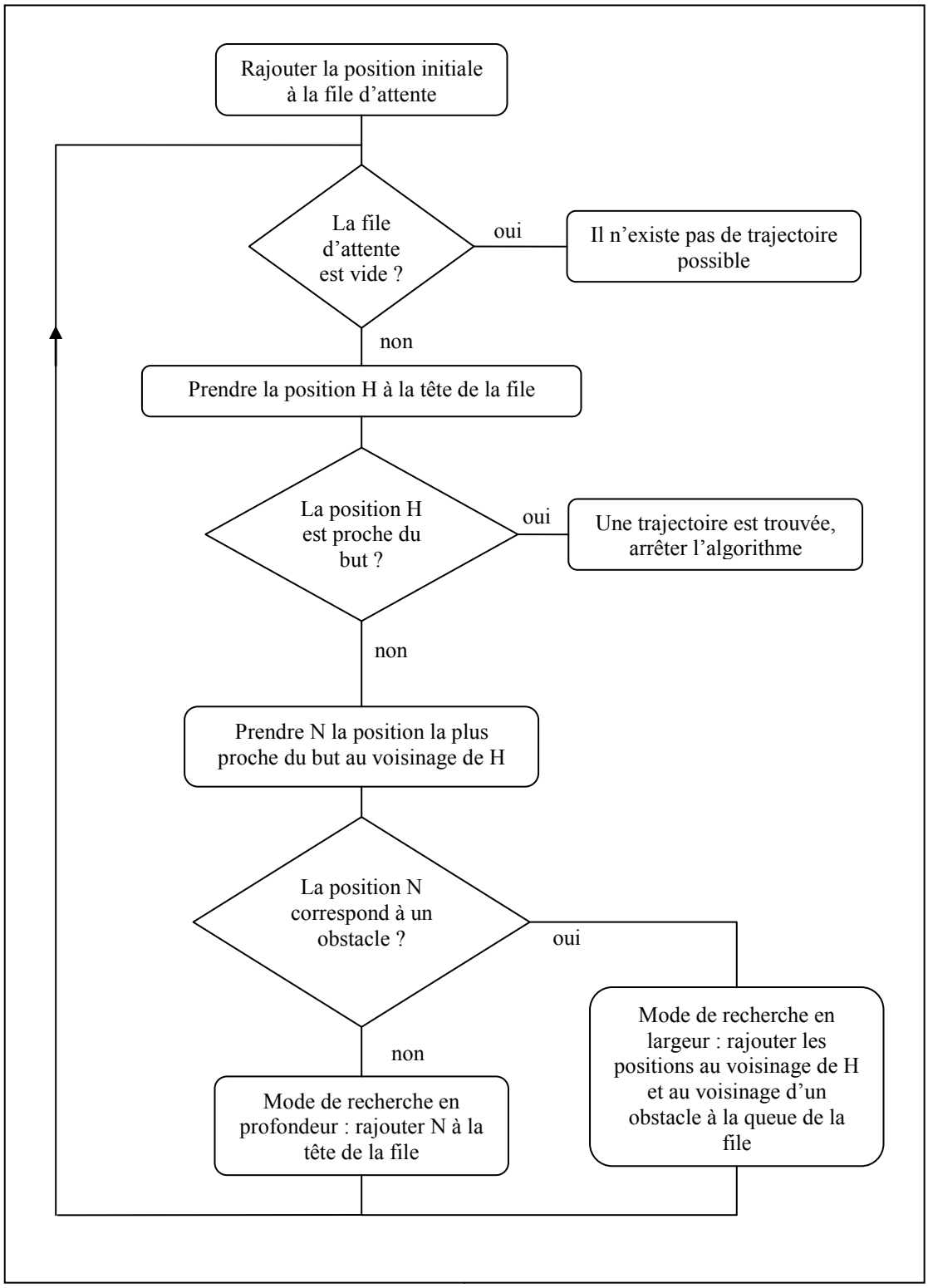


Figure III.8. Algorithme de la méthode

2 Détection de collision

Maintenant que nous avons expliqué notre approche de planification de trajectoire, nous allons présenter la méthode de détection de collision.

La détection de collision se base sur un algorithme de calcul de distance entre objets convexes. Cet algorithme converge rapidement en utilisant une méthode itérative (figure III.9).

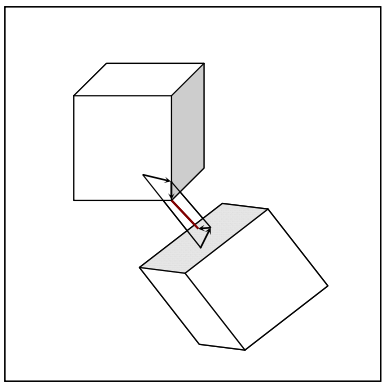


Figure III.9. Algorithme de calcul de distance basé sur une méthode itérative

2.1 Détection de collision entre deux objets

La méthode de calcul de distance que nous utilisons, a été proposée par [Zeghloul 96]. Nous la présentons brièvement ici, puisque nous l'avons utilisée comme base pour construire l'algorithme de détection de collision entre un objet et un segment.

2.1.1 Définition d'un objet

Nous ne considérons ici que le cas des polyèdres convexes. Les autres formes géométriques non convexes peuvent être approximées par une union de polyèdres convexes.

Un polyèdre **P** est un ensemble de points défini par l'équation suivante (figure III.10)

$$\mathbf{P} = \bigcap_{i=1}^{N} \left\{ \mathrm{X} \middle| \overrightarrow{\mathrm{OX}}.\vec{n}_i \le d_i \right\} \qquad \text{(III.1)}$$

Le polyèdre **P** correspond à l'intersection d'un ensemble de demi-espaces.

Une facette i du polyèdre est définie par un vecteur unitaire $\vec{n}_i$ normal à cette facette pointant vers l'extérieur de l'objet et par une distance d_i par rapport au point origine O d'un repère.

N est le nombre de facettes du polyèdre.

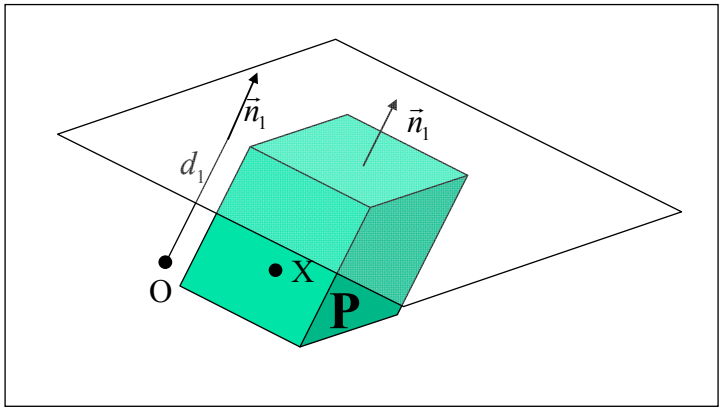


Figure III.10. Définition d'un polyèdre

2.1.2 Condition de Kuhn-Tucker

La figure III.11 montre deux points X_1 et X_2 appartenant à deux polyèdres $\mathbf{P}_1$ et $\mathbf{P}_2$. Nous appelons contraintes actives pour un point X, les contraintes correspondantes aux facettes auxquelles appartient le point X. C'est-à-dire si l'équation (III.2) est vérifiée :

$$\overrightarrow{OX}.\vec{n}_i = d_i \tag{III.2}$$

La condition de Kuhn-Tucker permet de juger si les deux points X_1 et X_2 réalisent la distance minimale entre $\mathbf{P}_1$ et $\mathbf{P}_2$. Cette condition est satisfaite lorsque les équations (III.3) sont vérifiées.

$$\begin{cases} \overrightarrow{X_1X_2} = \sum_{j=1}^{m_1} \alpha_{1j}\vec{n}_{1j} \text{ avec } \alpha_{1j} \geq 0 \\ \overrightarrow{X_2X_1} = \sum_{j=1}^{m_2} \alpha_{2j}\vec{n}_{2j} \text{ avec } \alpha_{2j} \geq 0 \end{cases} \tag{III.3}$$

m_i est le nombre de contraintes actives pour le point X_i. α_{ij} sont appelés les multiplicateurs de Kuhn-Tucker.

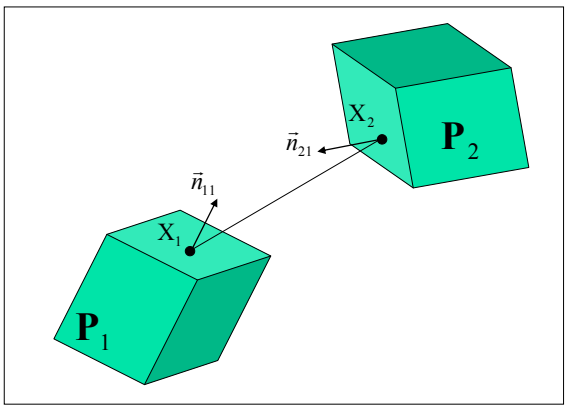


Figure III.11. Une étape de calcul de distance minimale

Lorsque les conditions de Kuhn-Tucker sont satisfaites pour X_1 et X_2, on peut calculer la distance entre les deux objets. Cette distance est donnée par la distance entre les deux points X_1 et X_2 qu'on peut calculer par l'équation (III.4).

$$d = \tfrac{1}{2}\overrightarrow{X_1X_2} \cdot \overrightarrow{X_1X_2} \qquad \text{(III.4)}$$

2.1.3 Direction de recherche

Pour trouver la direction de recherche, il existe deux cas en fonction de la condition de Kuhn-Tucker.

2.1.3.1 Condition de Kuhn-Tucker non vérifiée

Lorsque la condition de Kuhn-Tucker n'est pas satisfaite pour aucun des points X_1 et X_2, on utilise la méthode du gradient projeté de Rosen [Rosen 60]. Les directions de recherche sont données par l'équation (III.5).

$$\begin{cases} \vec{S}_1 = \dfrac{\mathbf{T}_1 \overrightarrow{X_1X_2}}{\left\| \mathbf{T}_1 \overrightarrow{X_1X_2} \right\|} \\ \vec{S}_2 = \dfrac{\mathbf{T}_2 \overrightarrow{X_2X_1}}{\left\| \mathbf{T}_2 \overrightarrow{X_2X_1} \right\|} \end{cases} \tag{III.5}$$

avec

$$\mathbf{T}_i = \mathbf{I} - \mathbf{N}_i \left(\mathbf{N}_i^{\,T} \mathbf{N}_i \right)^{-1} \mathbf{N}_i^{\,T} \tag{III.6}$$

Où $\mathbf{N}_i$ est la matrice dont les colonnes sont les composantes des vecteurs des contraintes actives au point X_i.

2.1.3.2 Condition de Kuhn-Tucker vérifiée pour un seul point

Lorsque la condition de Kuhn-Tucker est satisfaite pour un seul point, sans perte de généralité, on peut considérer que c'est X_2. La direction de recherche $\vec{S}_1$ est donnée par l'équation (III.5) et $\vec{S}_2$ est nulle ce qui conduit à un phénomène de zigzag. Pour éviter ce problème de zigzag qui apparaissait dans la méthode de Rosen [Rosen 60], Zeghloul [Zeghloul 96] a proposé une nouvelle formule de calcul de la direction $\vec{S}_2$ selon le nombre de contraintes actives de X_2.

Cas d'une seule contrainte active

Dans ce cas le point X_2 appartient à une facette de l'objet (figure III.12). La direction de recherche $\vec{S}_2$ est obtenue en projetant $\vec{S}_1$ sur le plan contenant X_2.

$$\vec{S}_2 = \frac{\vec{S}_1 - \left(\vec{n}_{21} . \vec{S}_1 \right) \vec{n}_{21}}{\left\| \vec{S}_1 - \left(\vec{n}_{21} . \vec{S}_1 \right) \vec{n}_{21} \right\|} \tag{III.7}$$

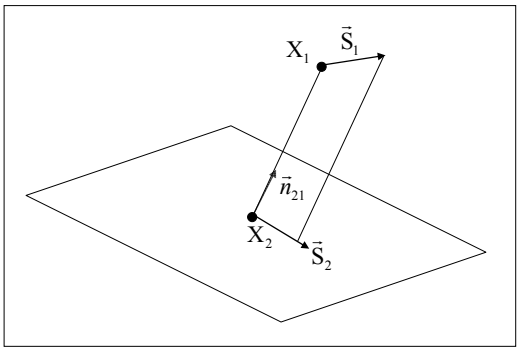


Figure III.12. Cas d'une seule contrainte active

Cas de deux contraintes actives

Dans ce cas, le point X_2 appartient à une arête de l'objet (figure III.13). On commence par écrire $\vec{S}_1$ sous la forme suivante :

$$\vec{S}_1 = \alpha_1 \vec{n}_{21} + \alpha_2 \vec{n}_{22} + \alpha_3 \left(\vec{n}_{21} \wedge \vec{n}_{22} \right) \tag{III.8}$$

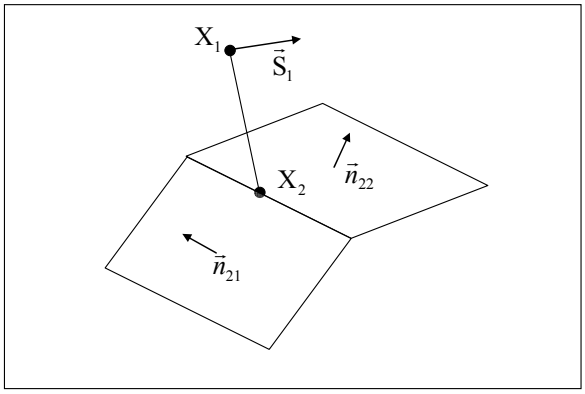


Figure III.13. Cas de deux contraintes actives

Si $\alpha_3 \neq 0$, $\vec{S}_2$ est un vecteur unitaire soit selon $\vec{n}_{21} \wedge \vec{n}_{22}$, soit selon $-\vec{n}_{21} \wedge \vec{n}_{22}$. Sinon $\vec{S}_1$ est projeté sur le plan correspondant au plus grand α_j.

$$\vec{S}_2 = \frac{\vec{S}_1 - \left(\vec{n}_{2j} . \vec{S}_1 \right) \vec{n}_{2j}}{\left\| \vec{S}_1 - \left(\vec{n}_{2j} . \vec{S}_1 \right) \vec{n}_{2j} \right\|} \tag{III.9}$$

Cas de trois contraintes actives ou plus

Dans ce cas, le point X_2 appartient à un sommet de l'objet (figure III.14). Pour chacune des arêtes actives nous calculons la quantité β_j donnée par l'équation (III.10).

$$\beta_j = \vec{S}_1 . \vec{e}_{2j} \tag{III.10}$$

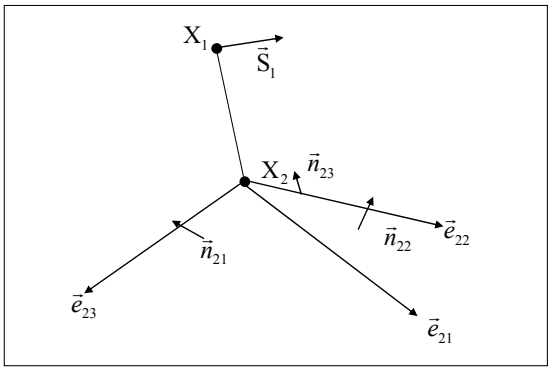


Figure III.14. Cas de trois contraintes actives

$\vec{S}_2$ est obtenue en projetant $\vec{S}_1$ sur l'arête qui correspond au β_j le plus grand.

$$\vec{S}_2 = \frac{\left(\vec{e}_{2j} . \vec{S}_1\right)\vec{e}_{2j}}{\left\|\left(\vec{e}_{2j} . \vec{S}_1\right)\vec{e}_{2j}\right\|} \tag{III.11}$$

2.1.3 Algorithme

Les étapes de l'algorithme de calcul de distance sont données dans la figure III.15.

1. Trouver un point X_1 intérieur à $\mathbf{P}_1$ et un point X_2 intérieur à $\mathbf{P}_2$
2. Trouver les contraintes actives des points X_1 et X_2
3. Vérifier la condition de Kuhn-Tucker. Si cette condition est satisfaite pour les deux points X_1 et X_2, la distance minimale est réalisée par ces deux points.
4. Calculer les directions de recherche $\vec{S}_1$ et $\vec{S}_2$ sur chaque objet
5. Trouver les points qui minimisent la distance selon les directions choisies
6. Affecter les points trouvés à X_1 et X_2, puis revenir à l'étape 2

Figure III.15. Algorithme de calcul de distance

2.2 Détection de collision entre un segment et un objet

Lors de la planification de trajectoire d'un robot à câbles, il est nécessaire d'éviter les collisions entre les câbles et les objets qui peuvent se trouver dans l'espace de travail. Un câble est considéré comme un segment défini par deux points A_1 et A_2 (figure III.16). Pour détecter la collision entre un objet et un segment, nous avons proposé une méthode rapide qui donne le résultat en un nombre d'itérations inférieur ou égal au nombre de facettes de l'objet.

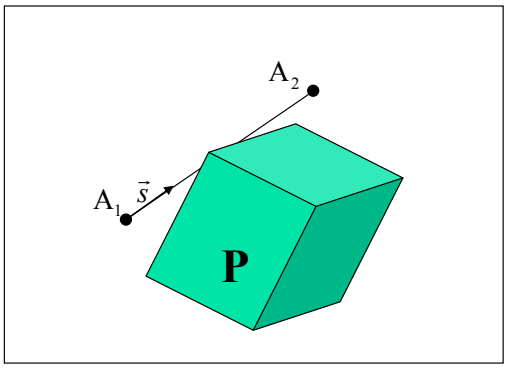


Figure III.16. Détection de collision entre un objet et un segment

Il était possible d'adapter l'algorithme de calcul de distance en remplaçant le segment par un cylindre et en utilisant l'algorithme de calcul de distance entre le cylindre et l'objet. L'une des difficultés de cette méthode est que le cylindre doit être recréé quand le segment change. Nous avons optés pour une nouvelle méthode qui parcoure les facettes de l'objet et qui s'arrête dès qu'elle trouve une facette permettant de juger que le segment est à l'extérieur de l'objet. Au fur et à mesure qu'on parcourt les facettes, on affine le calcul de deux paramètres

λ_{min} et λ_{max} qui donnent la portion du segment en collision avec l'objet. Initialement λ_{min} est nul et λ_{max} prend la longueur du segment. Si toutes les facettes sont parcourues, on compare λ_{min} et λ_{max}, dans le cas où λ_{min} est supérieur à λ_{max}, il n'y a pas de collision entre le segment et l'objet, dans le cas contraire la partie du segment en collision avec l'objet est donnée par λ_{min} et λ_{max}.

2.2.1 Différents cas de figures

Nous envisageons 3 cas de figure pour chaque facette i, en fonction du produit scalaire $\vec{s}.\vec{n}_i$ où $\vec{n}_i$ est la normale à la facette et $\vec{s}$ donnée par l'équation (III.12).

$$\vec{s} = \frac{\overrightarrow{A_1A_2}}{\left\|\overrightarrow{A_1A_2}\right\|} \tag{III.12}$$

On définit L la longueur du segment.

$$L = \left\|\overrightarrow{A_1A_2}\right\| \tag{III.13}$$

On pose

$$t_i = \vec{s}.\vec{n}_i \tag{III.14}$$

Cas 1 : $t_i > 0$

Ce cas est montré dans la figure III.17. Nous calculons λ_i définie par l'équation (III.15).

$$\lambda_i = \frac{d_i - \overrightarrow{OA_1}.\vec{n}_i}{t_i} \tag{III.15}$$

Dans le cas où $\lambda_i < 0$, on peut conclure qu'il n'y a pas de collision entre le segment et l'objet et on peut arrêter l'algorithme à ce stade.

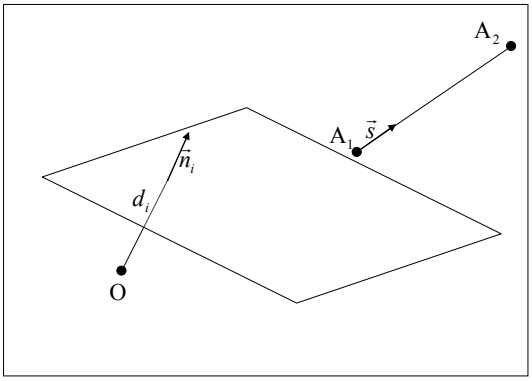


Figure III.17. Cas où $t_i > 0$

Cas 2 : $t_i < 0$

Ce cas est montré dans la figure III.18. Pour que le segment n'appartienne pas au demi-espace limité par la facette et contenant l'objet, il suffit de vérifier l'équation (III.16).

$$\overrightarrow{OA_2}.\vec{n}_i > d_i \qquad \text{(III.16)}$$

Ce qui revient à

$$\left(\overrightarrow{OA_1} + \overrightarrow{A_1A_2}\right)\vec{n}_i > d_i \qquad \text{(III.17)}$$

Or

$$\overrightarrow{A_1A_2} = L\vec{s} \qquad \text{(III.18)}$$

Puisque $t_i < 0$, l'équation (III.19) devient

$$L < \frac{d_i - \overrightarrow{OA_1}.\vec{n}_i}{t_i} \qquad \text{(III.19)}$$

On peut en conclure que lorsque $L < \lambda_i$, il n'y a pas de collision entre le segment et l'objet et on peut arrêter l'algorithme à ce stade.

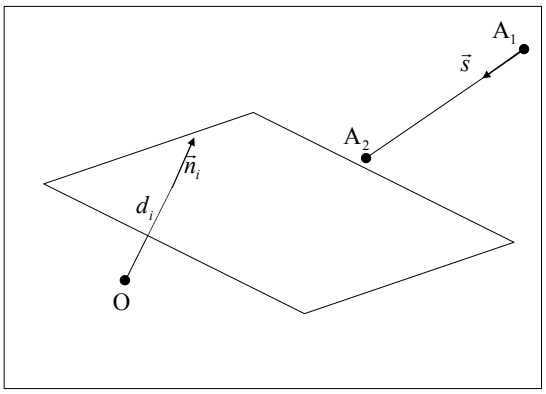


Figure III.18. Cas où $t_i < 0$

Cas 3 : $t_i = 0$

Dans ce cas, le segment est parallèle à la facette (figure III.19). Il suffit de regarder si le point A_1 par exemple est extérieur au demi-espace contenant l'objet pour en conclure qu'il n'y a pas de collision entre l'objet et le segment. Ceci correspond à l'équation (III.20).

$$d_i < \overrightarrow{OA_1}.\vec{n}_i \qquad \text{(III.20)}$$

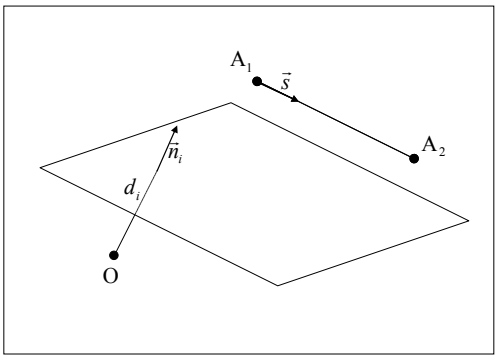


Figure III.19. Cas où $t_i = 0$

1. $\lambda_{max} = L$ et $\lambda_{min} = 0$
2. Pour $i = 1$ à N (le nombre de facettes)
3. Si $\mathbf{t}_i > 0$ alors
4. Si $\lambda_i < 0$, il n'y a pas de collision. On arrête l'algorithme
5. $\lambda_{max} \leftarrow \min(\lambda_{max}, \lambda_i)$
6. Fin Si
7. Si $\mathbf{t}_i < 0$ alors
8. Si $\lambda_i > L$, il n'y a pas de collision. On arrête l'algorithme
9. $\lambda_{min} \leftarrow \max(\lambda_{min}, \lambda_i)$
10. Fin Si
11. Si $\mathbf{t}_i = 0$ alors
12. Si $d_i < \overrightarrow{OA_1}.\vec{n}_i$, il n'y a pas de collision. On arrête l'algorithme
13. Fin Si
14. Fin Pour
15. Si $\lambda_{min} > \lambda_{max}$, il n'y a pas de collision
16. Sinon, il y a une collision entre les points B_1 et B_2

Figure III.20. Algorithme de détection de collision entre un objet et un segment

2.2.2 Algorithme

Les étapes de la procédure de détection de collision entre un objet et un segment sont données par la figure III.20. Pour chaque facette, on vérifie si le segment n'appartient pas au demi-espace contenant l'objet. Dans ce cas, il n'y a pas de collision et l'algorithme est arrêté. Sinon, on récupère la valeur de λ_i qu'on place selon les cas, soit dans λ_{min} soit dans λ_{max}. Si toutes les facettes sont parcourues et que l'algorithme n'a pas été arrêté, on regarde si $\lambda_{min} > \lambda_{max}$, dans ce cas il n'y a pas de collision. Si au contraire, $\lambda_{min} < \lambda_{max}$, on en conclut qu'il y a une collision du segment et de l'objet, entre les points B_1 et B_2 définis par les équations (III.21) et (III.22).

$$\overrightarrow{A_1B_1} = \lambda_{min}\vec{s} \qquad \text{(III.21)}$$

$$\overrightarrow{A_1B_2} = \lambda_{max}\vec{s} \qquad \text{(III.22)}$$

3 Etude du cas du 3-RPR

Pour illustrer la méthode de planification de trajectoire, nous avons choisi de prendre pour exemple le robot 3-RPR.

3.1 Description du robot

Le robot utilisé est montré dans la figure III.21.

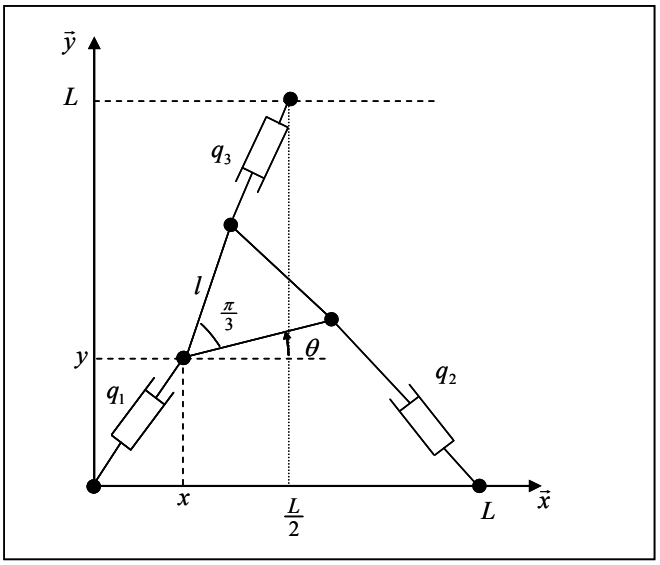


Figure III.21. Robot 3-RPR utilisé pour la planification

Nous utilisons les dimensions données par le tableau III.1. Nous définissons $q_{\min}$ comme étant l'allongement minimal de chacun des vérins et $q_{\max}$ comme étant l'allongement maximal.

Table III.1 Dimensions du robot choisi

l	L	$q_{\min}$	$q_{\max}$
15	75	10	200

Pour cet exemple, nous choisissons de considérer seulement la partie de l'espace de travail de θ allant de 0 à π. L'espace de travail de ce robot est donné par la figure III.22.

Les singularités de ce robot sont données par l'équation (III.23).

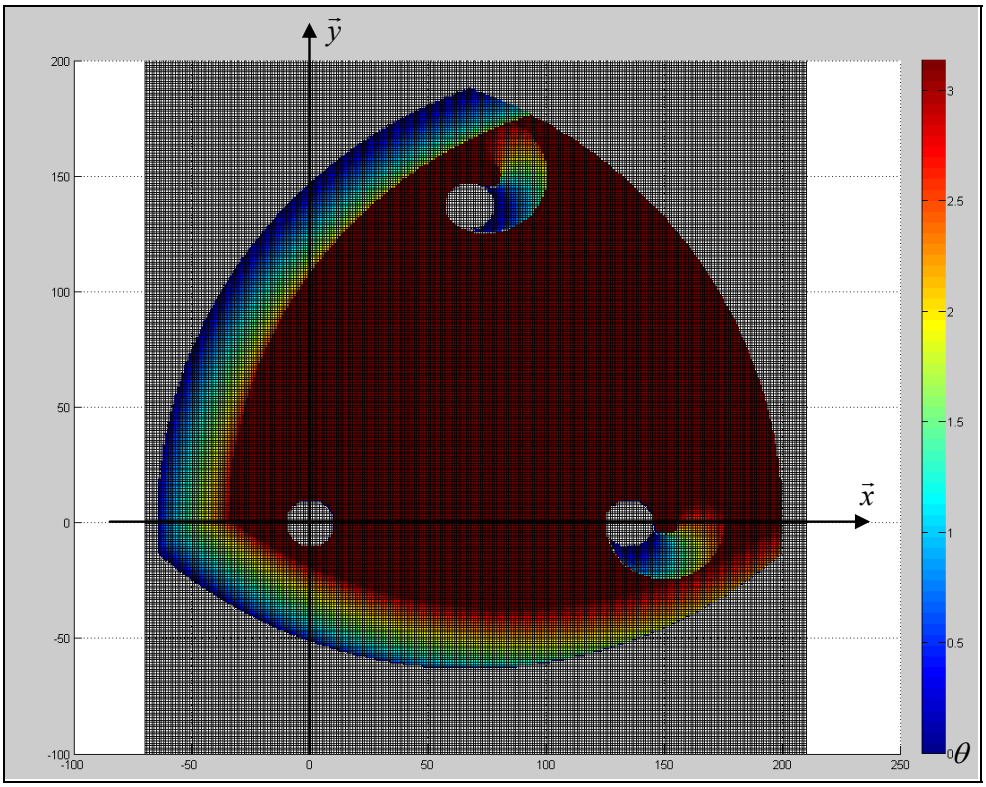


Figure III.22. Espace de travail du 3-RPR

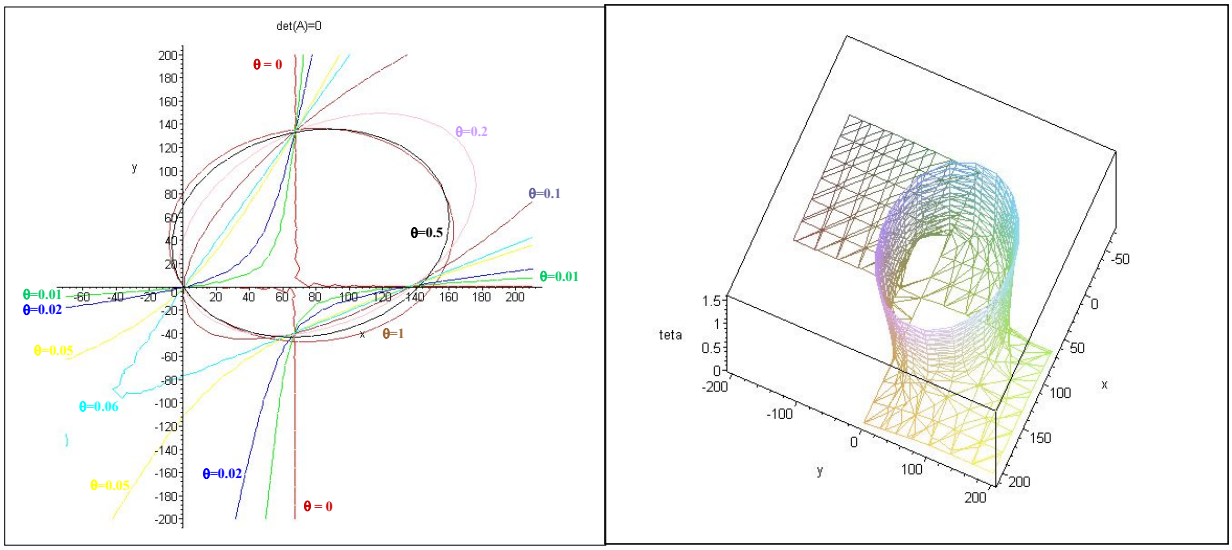


Figure III.23. Positions singulières du 3-RPR

$$\det(\mathbf{A}) = 0 \tag{III.23}$$

où

$$\mathbf{A} = \begin{pmatrix} x & y & 0 \\ x + l\cos\theta - L & y + l\sin\theta & l(L-x)\sin\theta + ly\cos\theta \\ x + l\cos\left(\theta + \frac{\pi}{3}\right) - \frac{L}{2} & y + l\sin\left(\theta + \frac{\pi}{3}\right) - L & -l\sin\left(\theta + \frac{\pi}{3}\right)\left(x - \frac{L}{2}\right) + l\cos\left(\theta + \frac{\pi}{3}\right)(y - L) \end{pmatrix} \tag{III.24}$$

Nous effectuons un balayage de l'espace de travail afin de chercher les positions de singularités. Pour les dimensions proposées de ce robot 3-RPR, les positions singulières sont montrées dans la figure III.23.

3.2 Evitement de singularités

Dans cette partie, nous testons la méthode sans placer d'obstacle, nous essayons de définir une tâche qui permet de montrer l'évitement des positions singulières.

3.2.1 Définition de la tâche

La tache de planification de trajectoire consiste à déplacer le robot de la position initiale P_i à la position finale P_f. Ces positions sont définies par le tableau III.2 et la figure III.24.

Table III.2 Positions initiale et finale

P_i	P_f
$(20,0,0.3)$	$(60,50,-0.6)$

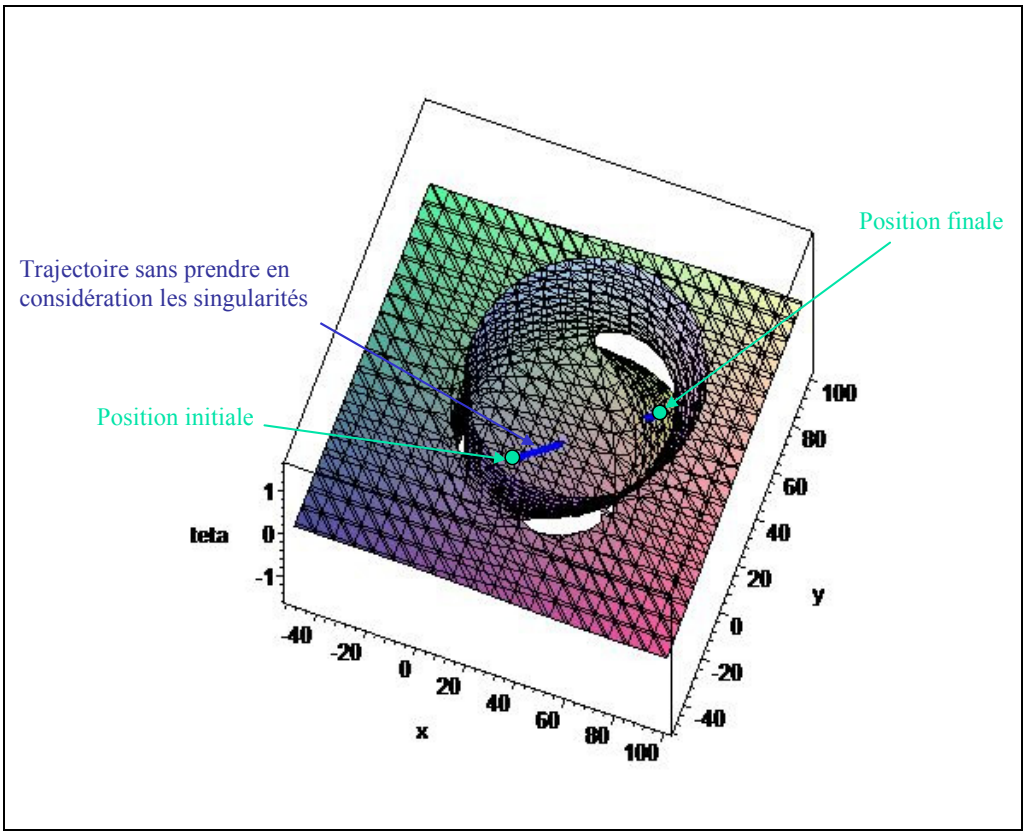


Figure III.24. Définition de la tâche de planification

Dans le cas où on ne considère pas les singularités, la trajectoire aurait été rectiligne entre la position initiale et la position finale, vu qu'il n'y a aucun obstacle à éviter.

3.2.2 Résultats

La figure III.25, montre les résultats de la planification de trajectoire donnés par la méthode proposée.

Les points en vert représentent les positions qui ont été investies afin de trouver la trajectoire sans collision. Les points en jaune, représentent les positions qui ont été placées dans la file d'attente et qui n'ont pas été investies lors de l'exécution de l'algorithme. On remarque qu'à proximité des positions singulières, la méthode a investi toutes les directions possibles qui longent les positions singulières. Pour générer la trajectoire finale, nous relions les points les plus significatifs des positions investies. Cela revient à relier la position initiale et la position finale aux points qui permettent de contourner les obstacles et les singularités.

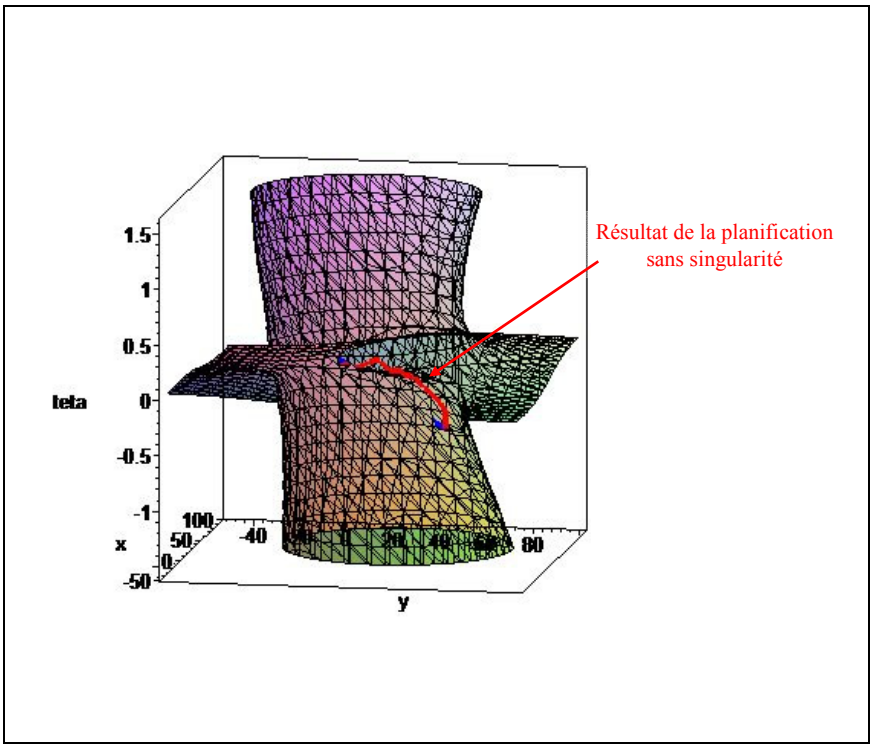


Figure III.25. Planification de trajectoire en évitant les singularités

Les séquences de cette trajectoire sont données par la figure III.26. La séquence A est la position initiale et D est la position finale. Les positions B et C permettent d'éviter les singularités. Concrètement, ces positions permettent d'éviter que deux vérins soient l'un en face de l'autre en train d'exercer des efforts opposés qui risquent d'endommager la structure.

3.3 Evitement d'obstacles et de singularité

Afin d'observer l'évolution des résultats en présence d'un obstacle. Nous avons compliqué la tâche en rajoutant une pince à la plateforme mobile du robot 3-RPR. Nous avons rajouté aussi un obstacle qui croise la trajectoire de la pince. Afin d'avoir une trajectoire sûre, il faut éviter à la fois l'obstacle et les positions singulières.

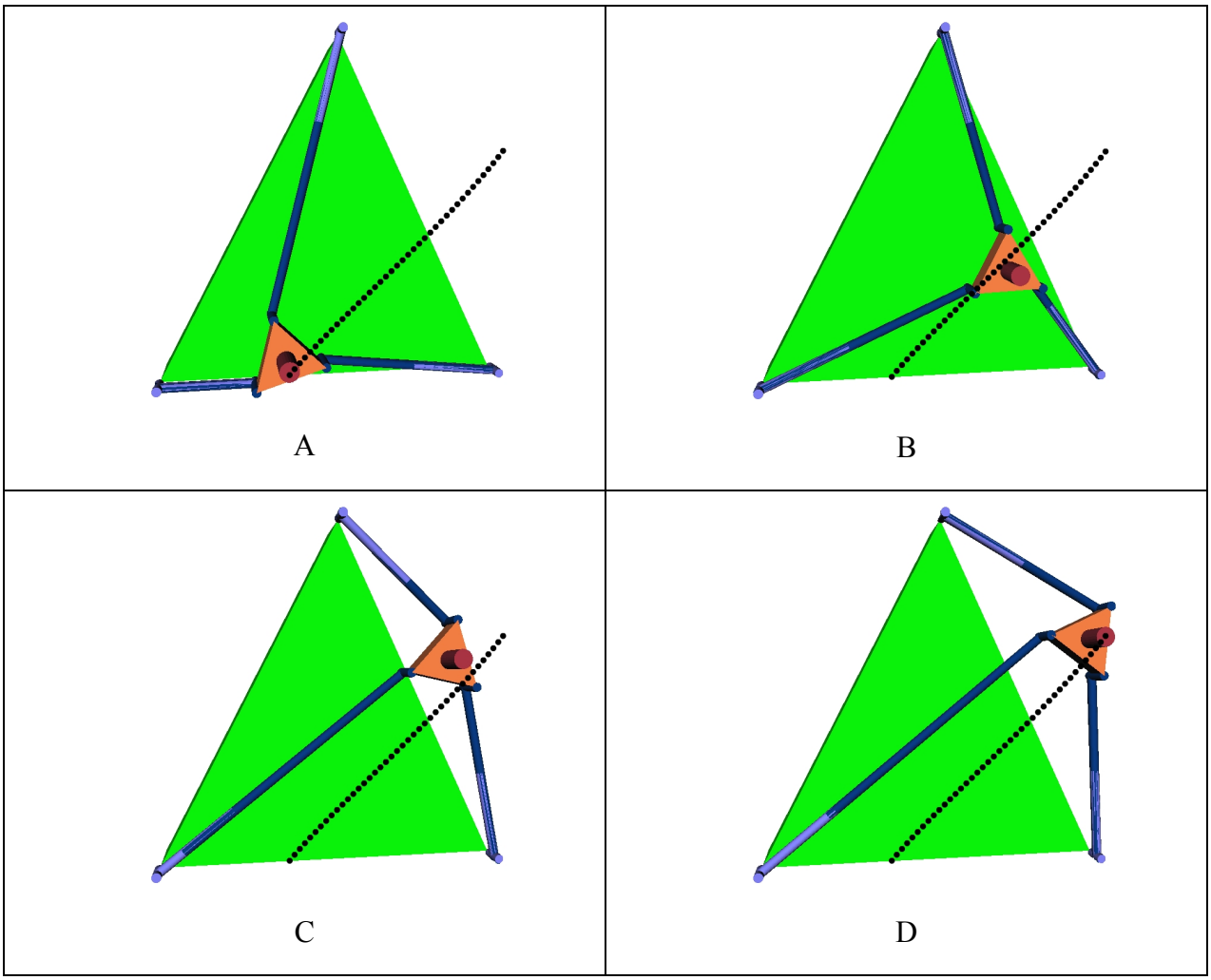


Figure III.26. Séquences du résultat de la planification

3.3.1 Définition de la tâche

Le robot est montré dans la figure III.27. La pince est approximée par un cylindre de rayon 1 et l'obstacle est un cylindre de rayon 3. Nous avons gardé la même position initiale et la même position finale.

La figure III.28 montre l'espace de travail ramené à l'organe terminal. On voit bien dans cette figure la transformation de l'obstacle dans cet espace.

La tâche de planification de trajectoire est montrée dans la figure III.29. Le robot doit éviter l'obstacle et les positions singulières.

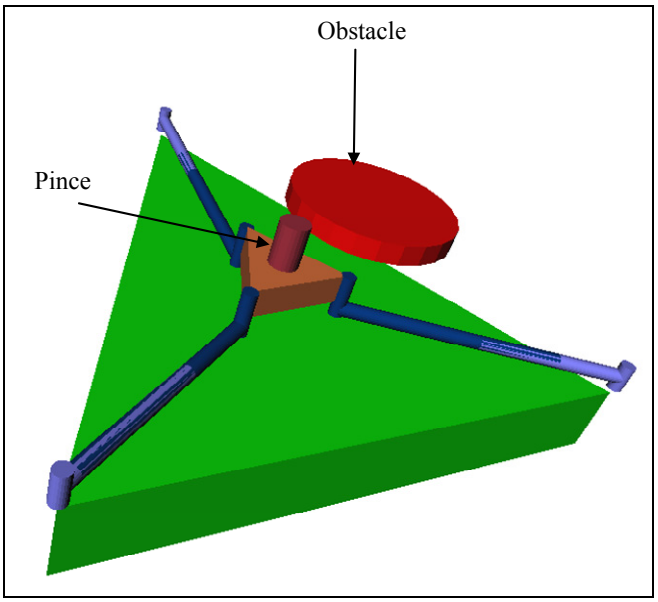


Figure III.27. Planification de trajectoire en présence d'un obstacle

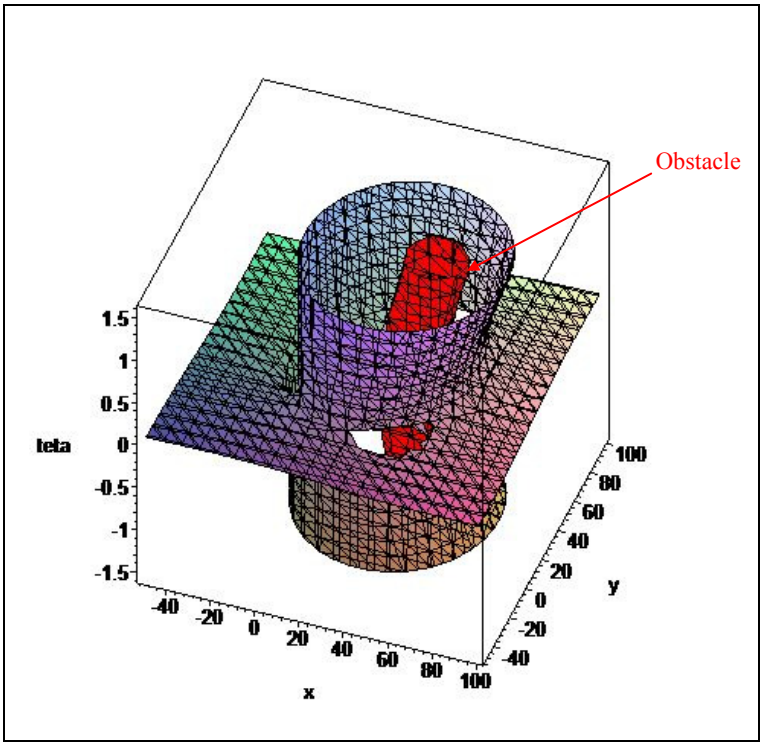


Figure III.28. Espace opérationnel ramené à l'organe terminal

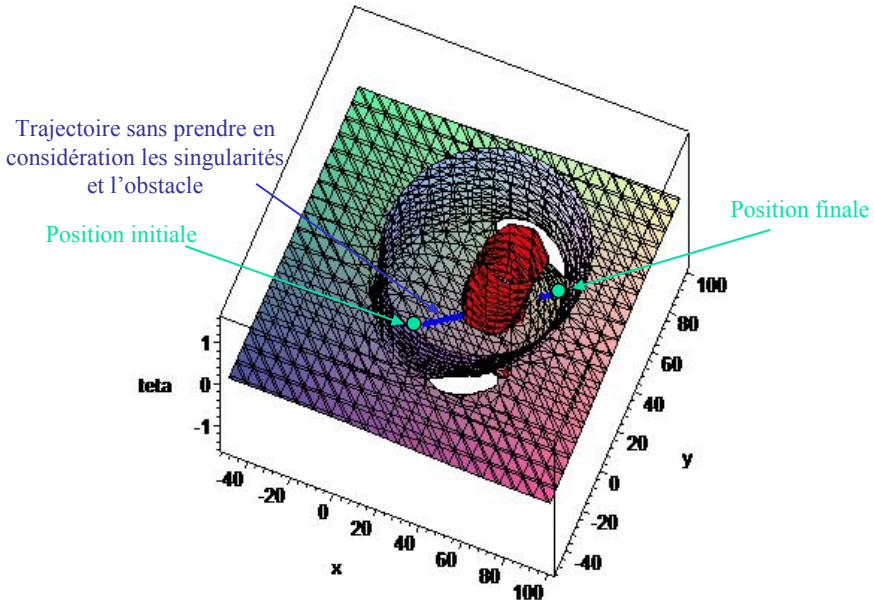


Figure III.29. Définition de la tâche de planification

3.3.2 Résultats

La figure III.30 donne les résultats de la planification de trajectoire dans l'espace de travail ramené à l'organe terminal. Les points en vert représentent les points explorés par l'algorithme. Ces points montrent que lors de la recherche de trajectoire, la méthode englobe l'obstacle et les singularités qui l'empêchent d'avancer vers le but. Lorsqu'une position permettant le retour au mode de recherche en profondeur est trouvée, elle est utilisée pour générer la trajectoire.

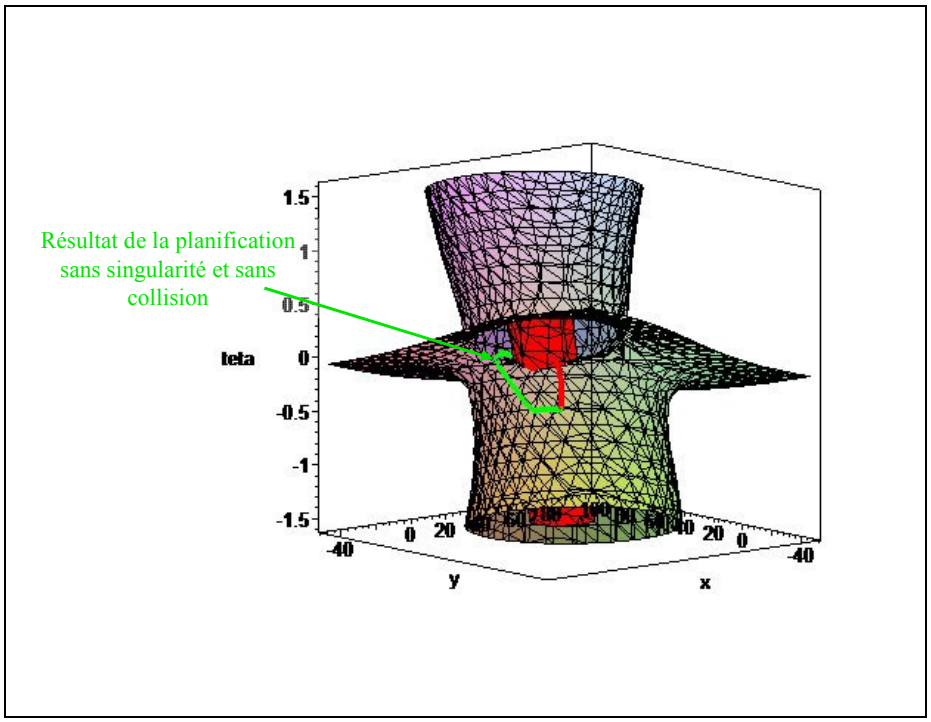
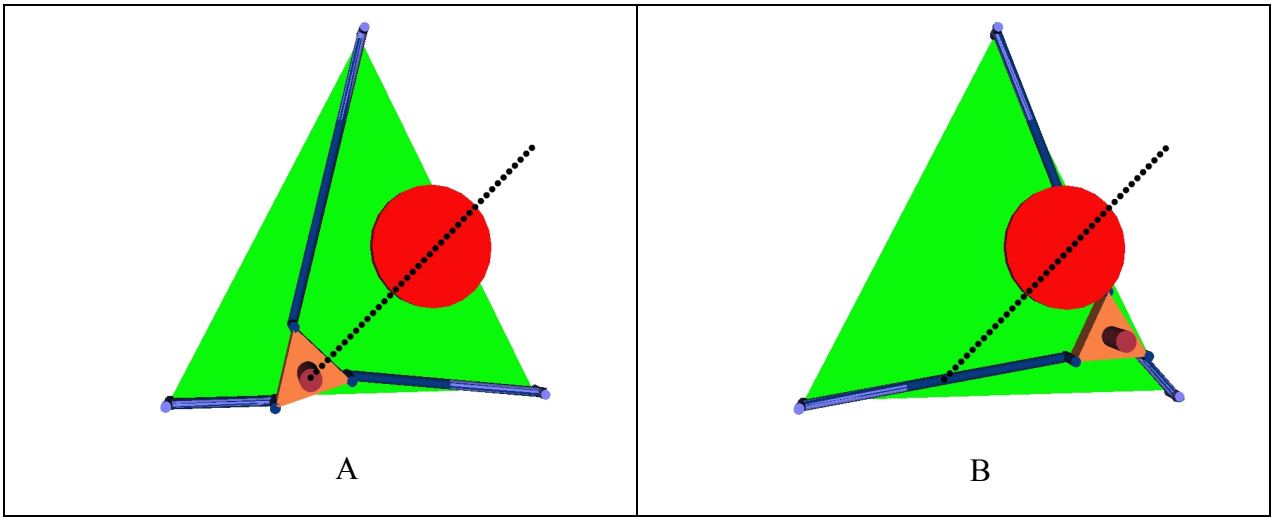


Figure III.30. Planification de trajectoire en évitant les singularités et les obstacles



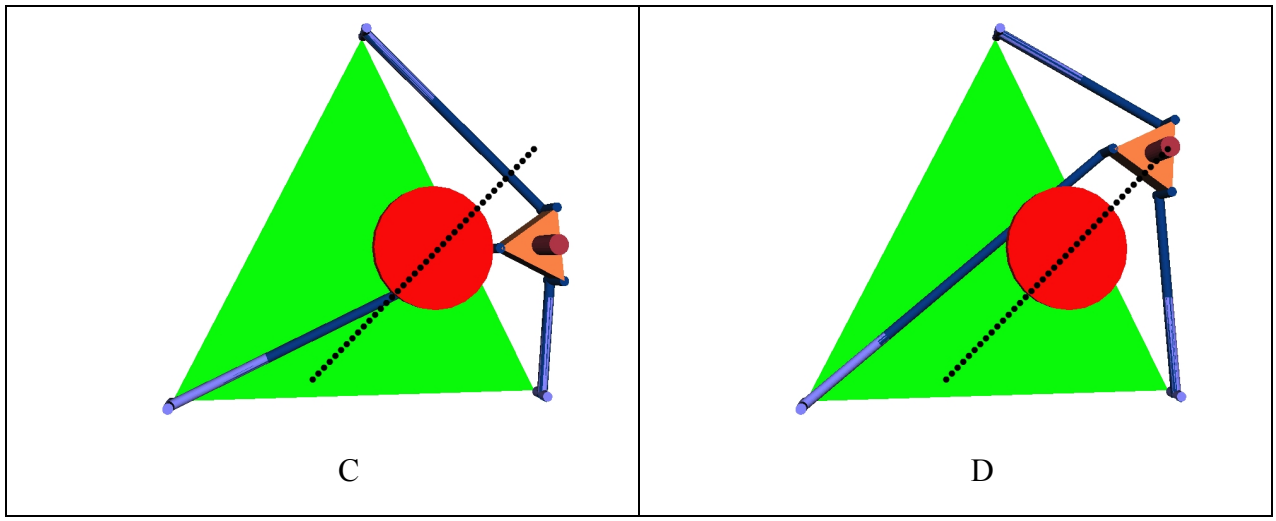


Figure III.31. Résultats de planification

Les séquences de cette trajectoire sont données par la figure III.31. A est la position de départ et D la position finale. Les positions B et C permettent d'éviter l'obstacle et les singularités.

Conclusion

Dans ce chapitre nous avons proposé une méthode de planification de trajectoire adaptée au cas des robots parallèles. Nous avons montré qu'il est possible d'adapter n'importe quelle méthode en utilisant comme espace de recherche l'espace de travail ramené à l'organe terminal. Nous avons proposé d'utiliser un algorithme de distance pour détecter les collisions entre un segment et un objet convexe. Cette méthode est très pratique dans le cas des robots à câbles pour garantir que lors de la simulation, les câbles n'entrent pas en collision avec les objets qui peuvent se trouver dans l'espace de travail du robot. Nous avons montré, à travers un exemple plan, l'évitement de singularités puis l'évitement d'un obstacle en plus des singularités.

Chapitre IV

Résultats et simulations

Dans le chapitre III, nous avons proposée une méthode de planification de trajectoire adaptée aux robots parallèles. Nous avons expliqué cette méthode à travers un exemple en simulation. Dans ce chapitre nous donnons les résultats expérimentaux effectués sur des robots réels.

Introduction

Les tests expérimentaux sur les robots parallèles se sont effectués sur un robot translateur et sur un robot à câbles. Le robot translateur utilisé est une plateforme expérimentale disponible au LMS. Pour les expérimentations sur le robot à câble, nous avons utilisé le robot à câbles du LARM, lors d'un stage d'un mois que j'ai effectué en Italie.

Les simulations ont été effectuées sur un PC AMD Athlon 64, en utilisant le logiciel SMAR. Les résultats ont été obtenus en une vingtaine de secondes pour chaque tâche de planification.

Nous avons modélisé aussi un robot mobile à quatre pattes de type chien en le considérant comme étant un robot parallèle. La méthode de génération de trajectoire de marche utilise le fait que le robot à pattes est considéré comme un robot parallèle. Nous montrons les résultats expérimentaux réalisés sur un robot chien développé au LMS.

1 Le translateur

Pour tester la méthode de planification de trajectoire proposée, nous avons utilisé un robot parallèle translateur appelé RAF. Ce robot translateur a été construit au LMS en se basant sur le travail de Affi et al. [Affi04], qui ont développé une synthèse dimensionnelle du 3-TPM proposé par [Romdhane02]. Ce robot est un translateur spatial (figure IV.1).

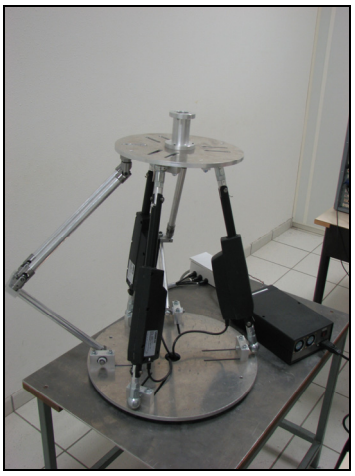


Figure IV.1. Robot translateur RAF

Il est composé de trois vérins et de deux chaînes cinématiques passives. Les chaînes passives permettent d'enlever les rotations de la plateforme mobile. Chaque chaîne passive est composée d'un bras lié à la base par une liaison rotoïde. Deux avant-bras viennent à la suite par l'intermédiaire de deux joints de Cardan (figure IV.2). Deux autres joints de Cardan viennent relier la chaîne passive à la plateforme mobile.

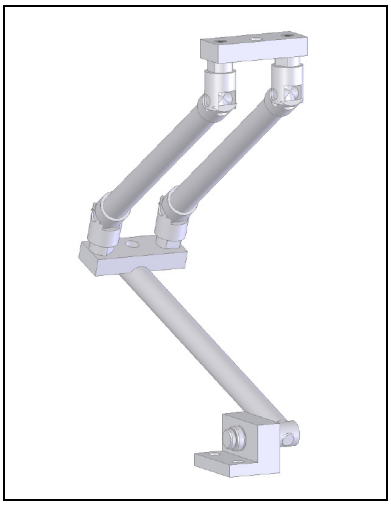


Figure IV.2. Chaîne passive

La plateforme mobile a donc trois translations par rapport au bâti.

1.1 Modélisation sous SMAR

La figure IV.3 montre le graphe de liaisons de ce robot. Ce robot est composé de 22 maillons dont un maillon base et un maillon mobile. Les autres maillons sont des maillons

séries. Chaque vérin est composé de deux maillons reliés par une liaison prismatique. Chaque chaîne passive est composée de 7 maillons, dont une représentant le bras et trois pour chaque avant-bras.

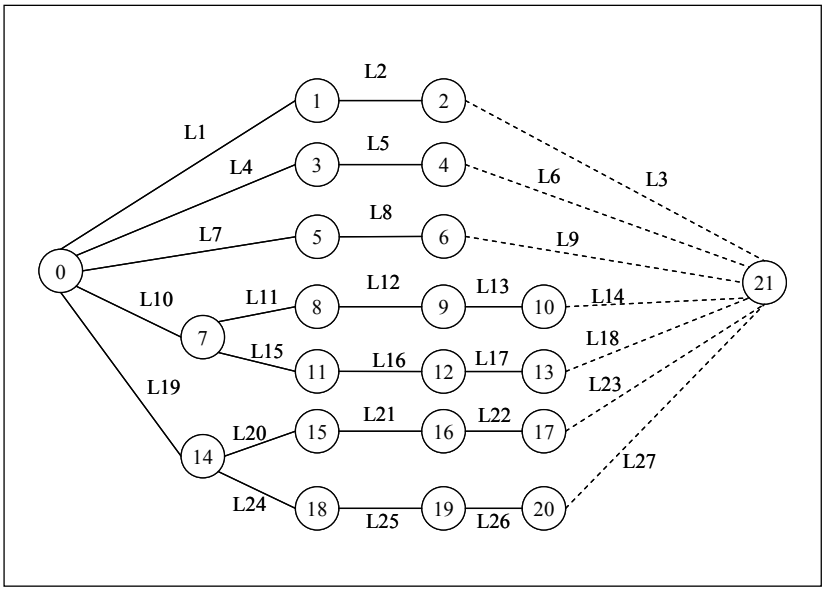


Figure IV.3. Graphe des liaisons

Table IV.1 Définition des liaisons

Liaison	Type	Point
L1	Sphérique	$(185,-106,20)$
L2	Prismatique d'axe $\vec{z}$	$(0,0,298)$
L4	Sphérique	$(-185,-106,20)$
L5	Prismatique d'axe $\vec{z}$	$(0,0,298)$
L7	Sphérique	$(0,214,20)$
L8	Prismatique d'axe $\vec{z}$	$(0,0,298)$
L10	Pivot d'axe $\vec{x}$	$(-50,-180,25)$
L11	Pivot d'axe $\vec{x}$	$(30,52,275)$
L12	Pivot d'axe $\vec{z}$	$(0,0,0)$
L13	Pivot d'axe $\vec{z}$	$(0,331,0)$
L15	Pivot d'axe $\vec{x}$	$(-30,52,275)$
L16	Pivot d'axe $\vec{z}$	$(0,0,0)$
L17	Pivot d'axe $\vec{z}$	$(0,331,0)$
L19	Pivot d'axe $\vec{x}$	$(-180,50,25)$
L20	Pivot d'axe $\vec{x}$	$(30,52,275)$

L21	Pivot d'axe $\vec{z}$	$(0,0,0)$
L22	Pivot d'axe $\vec{z}$	$(0,331,0)$
L24	Pivot d'axe $\vec{x}$	$(-30,52,275)$
L25	Pivot d'axe $\vec{z}$	$(0,0,0)$
L26	Pivot d'axe $\vec{z}$	$(0,331,0)$

Les liaisons sont définies dans le tableau IV.1. Les points sont donnés par rapport au repère lié au maillon père. Tous les maillons ont la même orientation par rapport à leur maillon père sauf le maillon 14 qui subit une rotation $R\left(\vec{z},-\frac{\pi}{2}\right)$. Les positions des repères sont données par les points des liaisons par rapport au maillon père. Les dimensions sont données en mm. Les axes sont pris en considérant un paramétrage normal par rapport au maillon père à chaque fois.

Les liaisons en pointillé montrées dans le graphe des liaisons correspondent à des liaisons de fermeture de chaînes. Il y a trois liaisons sphériques entre les vérins et le maillon mobile. Pour réaliser ces liaisons, on utilise des contraintes de coïncidence entre deux points. Les autres liaisons de fermeture de chaînes correspondent à des liaisons rotoïdes. Chacune de ces liaisons est définie par deux contraintes de coïncidence entre deux points. Les contraintes sont données dans le tableau IV.2. Dans ce tableau le point 1 correspond aux coordonnées du point dans le maillon 1, alors que le point 2 correspond aux coordonnées du point dans le maillon 2. Chacune des liaisons L14, L18, L23 et L27 correspond à deux contraintes de coïncidence entre deux points.

Table IV.2 Définition des contraintes de coïncidence entre deux points

Liaison	Maillon 1	Maillon 2	Point 1	Point 2
L3	2	21	$(0,0,92)$	$(85,-49,-20)$
L6	4	21	$(0,0,92)$	$(-85,-49,-20)$
L9	6	21	$(0,0,92)$	$(0,98,-20)$
L14	10	21	$(5,0,0)$	$(35,-134,-47)$
			$(-5,0,0)$	$(25,-134,-47)$
L18	13	21	$(5,0,0)$	$(-25,-134,-47)$
			$(-5,0,0)$	$(-35,-134,-47)$
L23	17	21	$(5,0,0)$	$(-126,-35,-47)$
			$(-5,0,0)$	$(-126,-25,-47)$
L27	20	21	$(5,0,0)$	$(-126,25,-47)$
			$(-5,0,0)$	$(-126,35,-47)$

1.2 Définition de la tâche

La tâche voulue consiste à déplacer le robot translateur d'une position initiale jusqu'à une position finale. Nous rajoutons des obstacles dans l'espace de travail afin de tester la planification de trajectoire. La figure IV.4 montre les positions initiale et finale ainsi qu'une position intermédiaire sans rajouter d'obstacle à l'espace de travail.

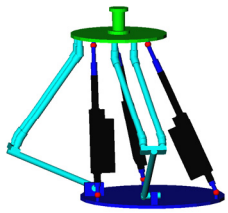
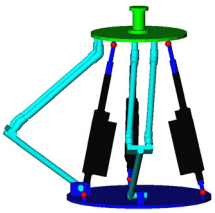
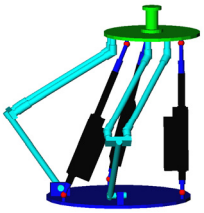
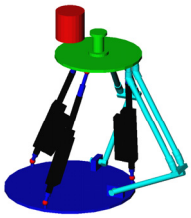
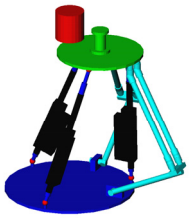
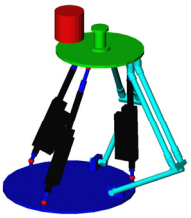
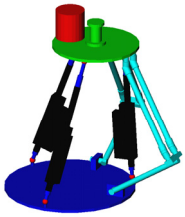
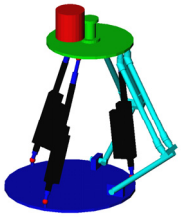
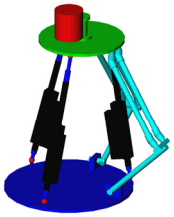
		
Position initiale	Position intermédiaire	Position finale

Figure IV.4. Positions du robot translateur en absence d'obstacles

1.3 Résultats

1.3.1 Simulation sous SMAR

La figure IV.5 montre les séquences de planification de trajectoire en présence d'un obstacle cylindrique. Le robot contourne l'obstacle par le coté en suivant sa géométrie.

		
A	B	C
		
D	E	F

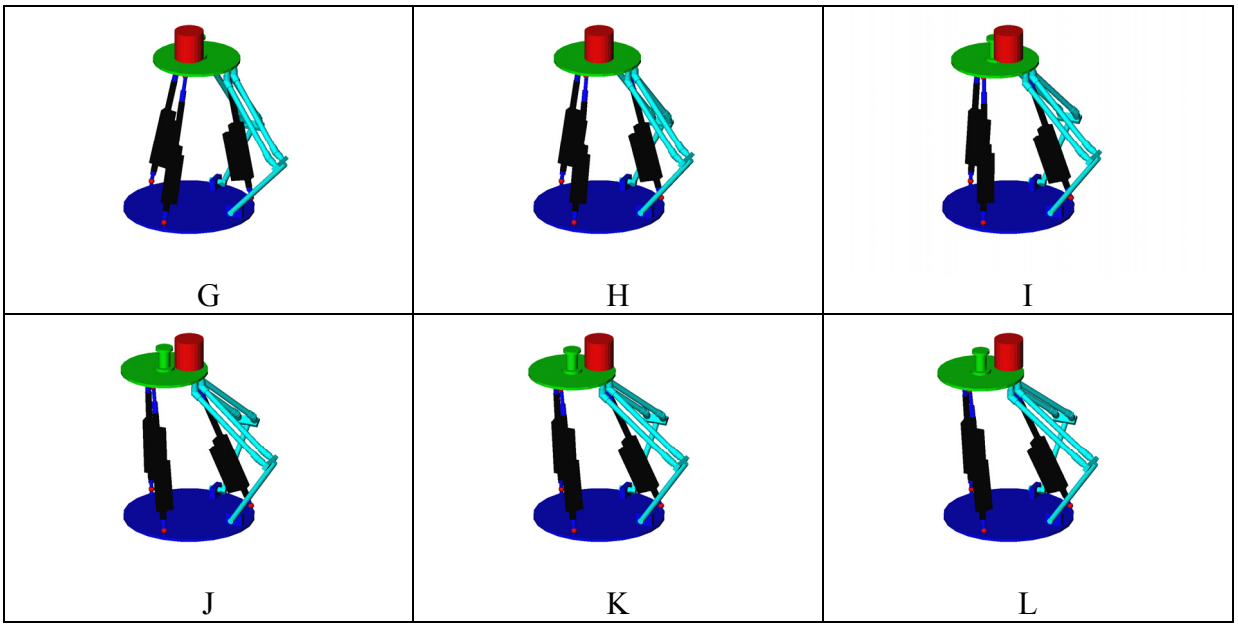
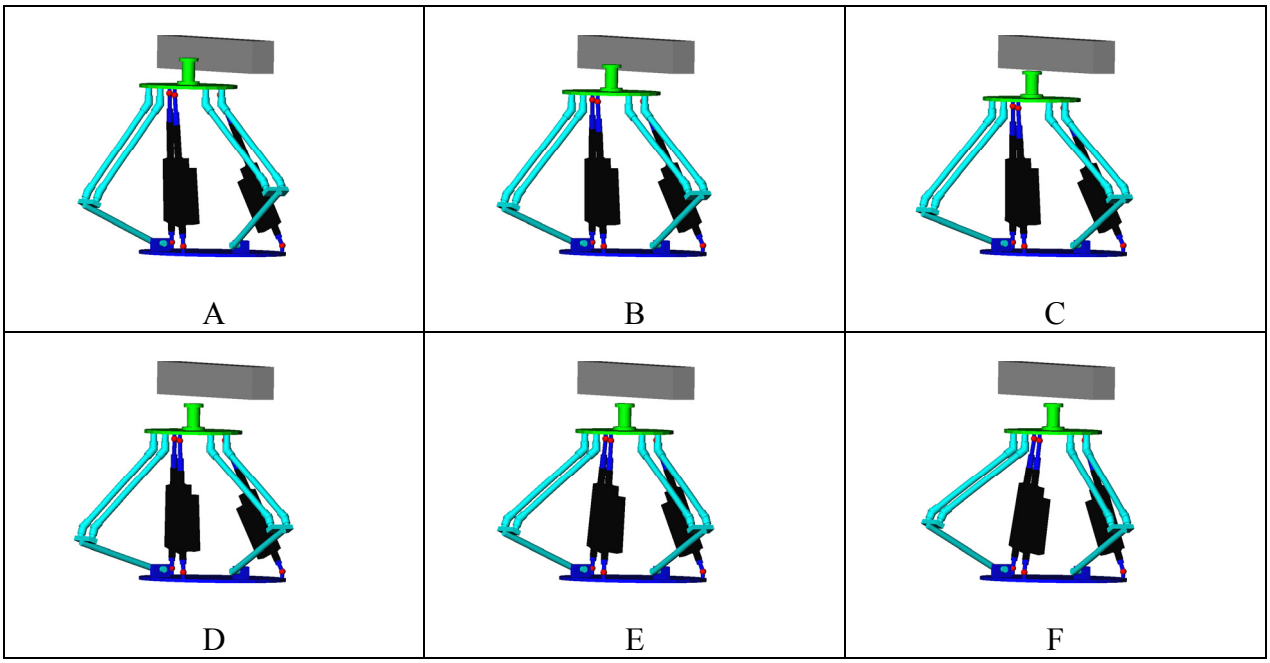


Figure IV.5. Planification de trajectoire en présence d'un obstacle cylindrique

Nous avons voulu compliquer la tache en changeant l'obstacle de telle manière qu'il ne soit plus possible de contourner l'obstacle dans le même plan, comme c'est le cas de la figure IV.5. Nous avons placé un obstacle de forme parallélépipédique. Nous montrons les résultats dans la figure IV.6.



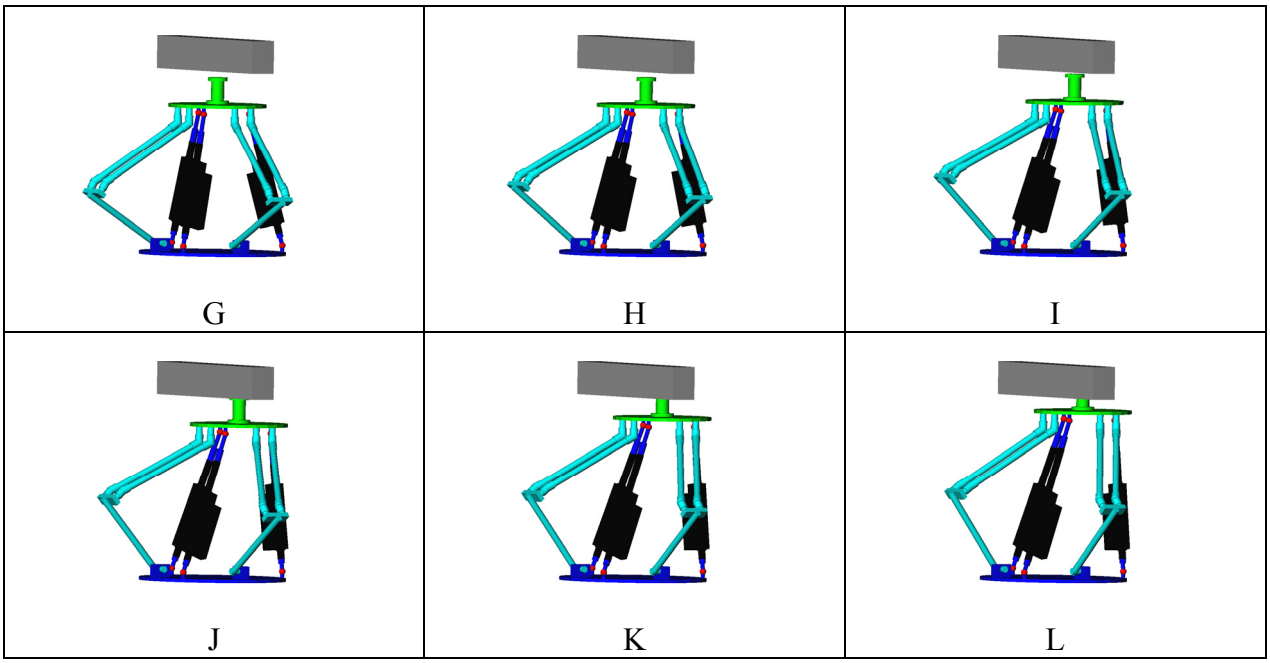


Figure IV.6. Planification de trajectoire en présence d'un parallélépipédique

Les résultats montrent que le robot contourne l'obstacle par le bas, car ce n'est plus possible de contourner par le côté. Le contournement par le côté a été plus avantageux pour l'obstacle cylindrique tel qu'il est placé. Le critère pris pour la comparaison, correspond à la longueur de la trajectoire résultante. Cette longueur est caractérisée par le nombre de points définissant cette trajectoire en fonction de la précision choisie.

1.3.2 Résultats expérimentaux

Nous avons testé les résultats de la simulation sur la structure réelle. Pour cela, le robot translateur dispose d'une unité de commande développée au LMS. Cette unité de commande comporte trois cartes d'axes qui permettent chacune de commander un vérin de manière synchrone. Les cartes d'axes ont été développées au LMS dans le but d'être intégrées dans un robot humanoïde. Nous les avons utilisées dans plusieurs autres dispositifs. Ces cartes ont été conçues pour un usage en temps réel. L'architecture de la carte (voir figure IV.7) s'articule autour d'un microcontrôleur 16 bits cadencé à 29.49MHz, fabriqué par Microchip et d'un circuit de puissance permettant de délivrer une intensité qui peut atteindre les 3A. Les cartes d'axes ont les fonctionnalités suivantes : entrées/sorties logiques, entrées d'interruption externe, timers internes programmables, convertisseurs rapides analogiques/numériques, sortie signal PWM, entrée codeur incrémental, communication CAN, communication RS232… Ces caractéristiques permettent d'intégrer les dispositifs de pilotage de l'axe, les

butées de fin de courses, le vérin (puissance, codeur, flag thermique, mesure de courant), le réseau CAN et la liaison sérielle pour communiquer avec un superviseur.

L'unité de commande communique avec SMAR par l'intermédiaire d'une liaison série. Une seule carte reçoit les informations puis les dispatche aux autres cartes par le réseau CAN. SMAR peut échanger des informations avec l'unité de commande, il peut donner des consignes de positions et peut récupérer l'allongement des vérins. Le lien entre le SMAR, l'unité de commande et la structure mécanique est synthétisé dans la figure IV.8.

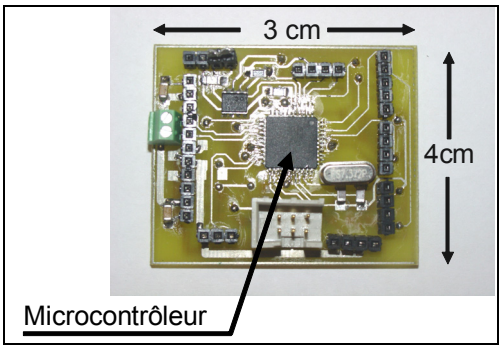


Figure IV.7. Une carte d'axe

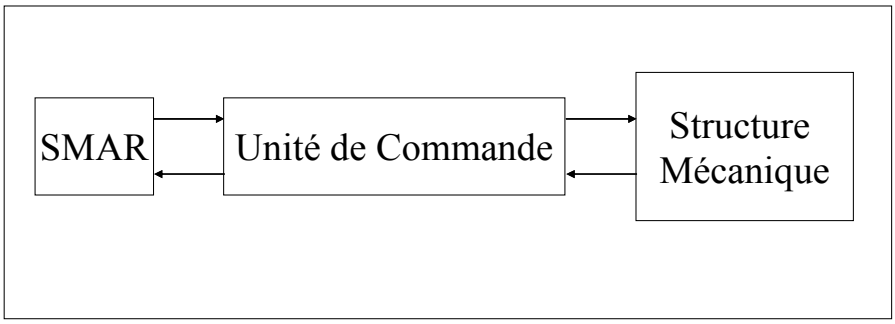
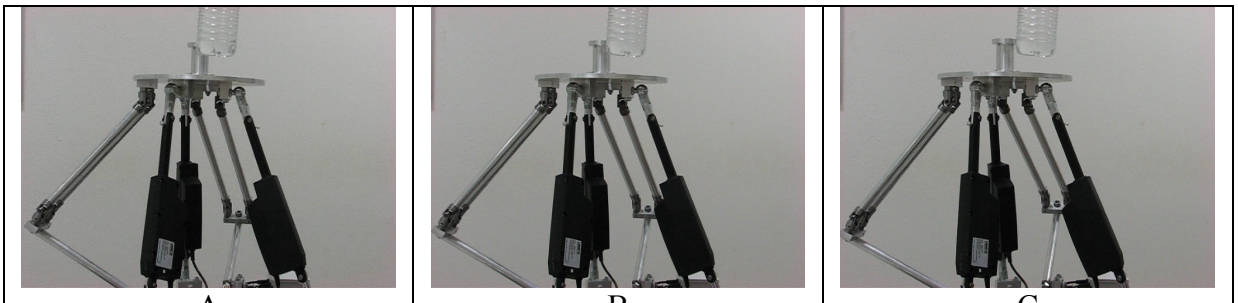


Figure IV.8. Utilisation de SMAR pour commander le RAF



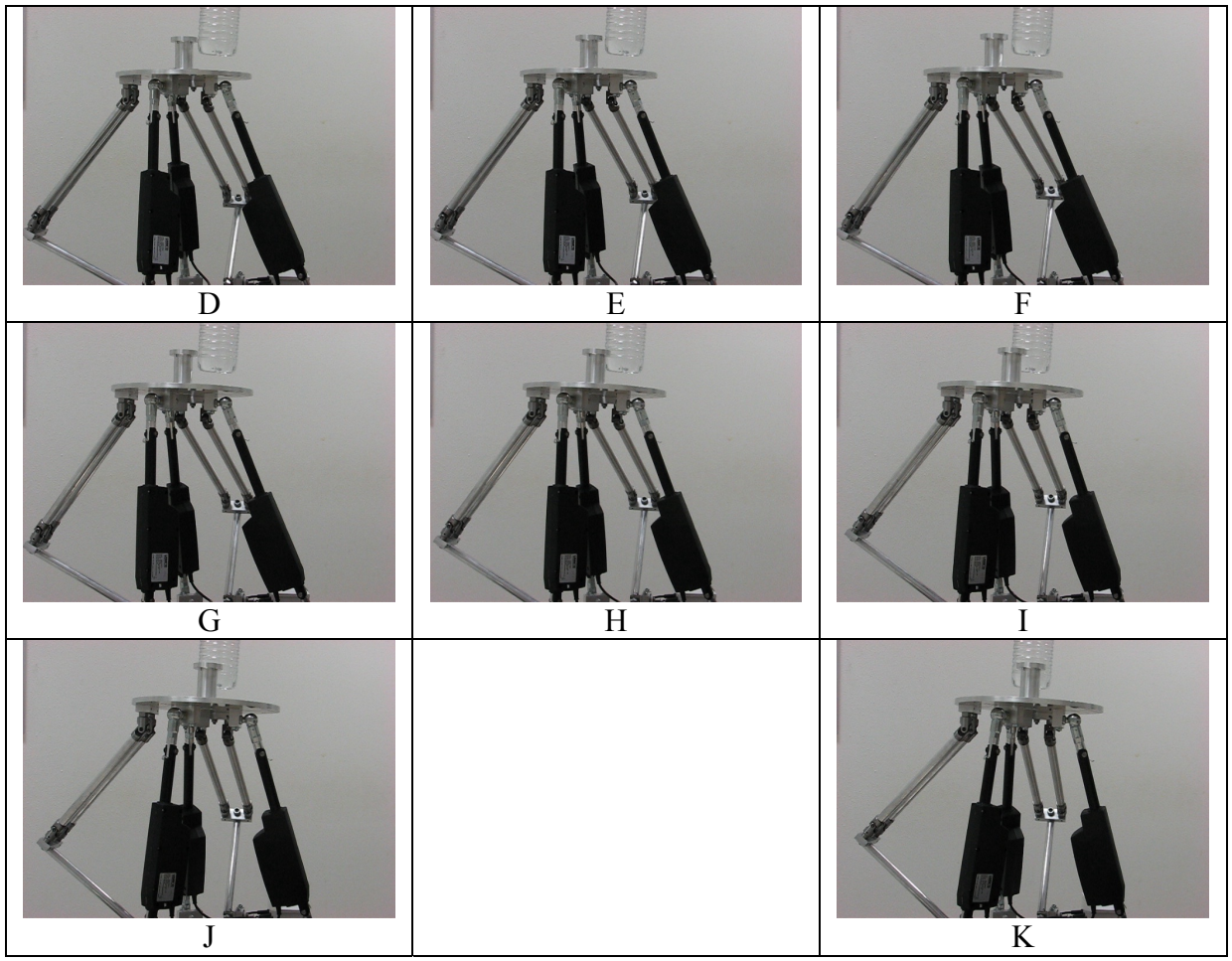


Figure IV.9. Résultats expérimentaux du RAF

La figure IV.9 montre les séquences de contournement de l'obstacle cylindrique. Cette planification est faite hors ligne sous SMAR en premier lieu, en prenant les mêmes dimensions et la même position que l'obstacle. Ensuite, nous établissons une communication entre SMAR et l'unité de commande, ce qui met la structure réelle dans la configuration correspondante à la simulation. La trajectoire résultante est ensuite exécutée sous SMAR déplaçant ainsi la structure réelle. Comme nous ne disposons pas de retour sur l'état de l'espace de travail, la planification de trajectoire est effectuée en hors-ligne. Nous avons commencé par construire notre espace de travail réel que nous avons modélisé sous SMAR et nous avons effectué les simulations que nous avons fait exécuter par la suite au robot. Il est envisageable de rajouter des capteurs ou une caméra pour détecter les obstacles et pour effectuer une planification de trajectoire en temps réel.

2 Le robot à câbles

Nous avons testé l'algorithme de planification de trajectoire sur le robot à câbles développé au LARM. Ce robot comporte quatre câbles attachés à une plateforme mobile par deux points d'attaches (figure IV.10).

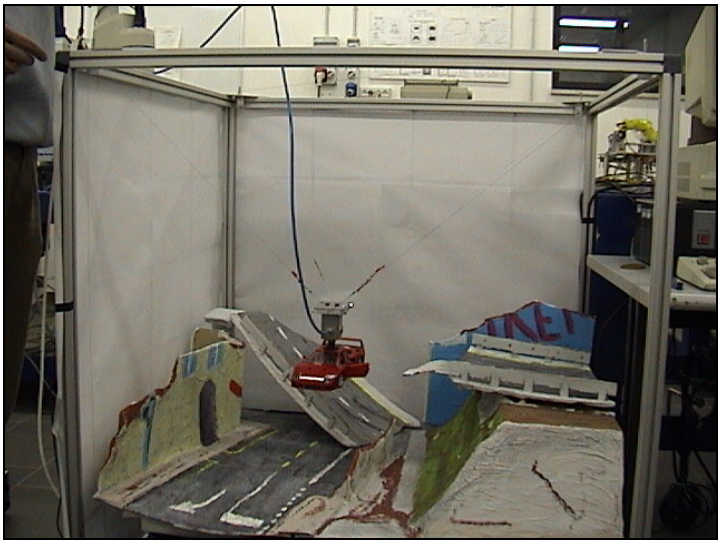


Figure IV.10. Robot à câbles du LARM

2.1 Modélisation

Pour modéliser le robot à câbles sous SMAR, nous utilisons un maillon base où sont placés les éléments de la structure qui maintiennent les câbles. Nous utilisons aussi un maillon mobile qui permet de modéliser la plateforme. Nous avons rajouté un module au SMAR qui permet de dessiner les câbles et de vérifier qu'ils ne soient pas en collision avec les objets qui peuvent se trouver dans l'espace de travail (voir chapitre III).

La détection de singularité pour les robots à câbles est plus compliquée que dans le cas des robots parallèles classiques. Ceci est dû au fait que la tension de chacun des câbles doit être tout le temps positive. Si ce n'est pas le cas, la plateforme mobile devient instable.

Les câbles peuvent seulement tirer la plateforme mais ne peuvent pas la pousser, la tension des câbles est assurée par le poids de la plateforme mobile. Pour garder la stabilité de la plateforme mobile, nous imposons aux positions et à l'orientation de cette plateforme de respecter les équations suivantes (voir figure IV.11) :

$$h \leq x_G \leq L - h \qquad \text{(IV.1)}$$

$$0 \le y_G \le L \tag{IV.2}$$

$$-L \le z_G < 0 \tag{IV.3}$$

$$-\psi_1 \le \theta \le \psi_2 \tag{IV.4}$$

Avec

$$\psi_1 = \min\left(\left|\arctan\frac{L-y_G}{x_G}\right|, \left|\arctan\frac{y_G}{L-x_G}\right|\right) \tag{IV.5}$$

et

$$\psi_2 = \min\left(\left|\arctan\frac{y_G}{x_G}\right|, \left|\arctan\frac{L-y_G}{L-x_G}\right|\right) \tag{IV.6}$$

Les dimensions h et L ainsi que l'angle θ et le point G sont définis dans la figure IV.11.

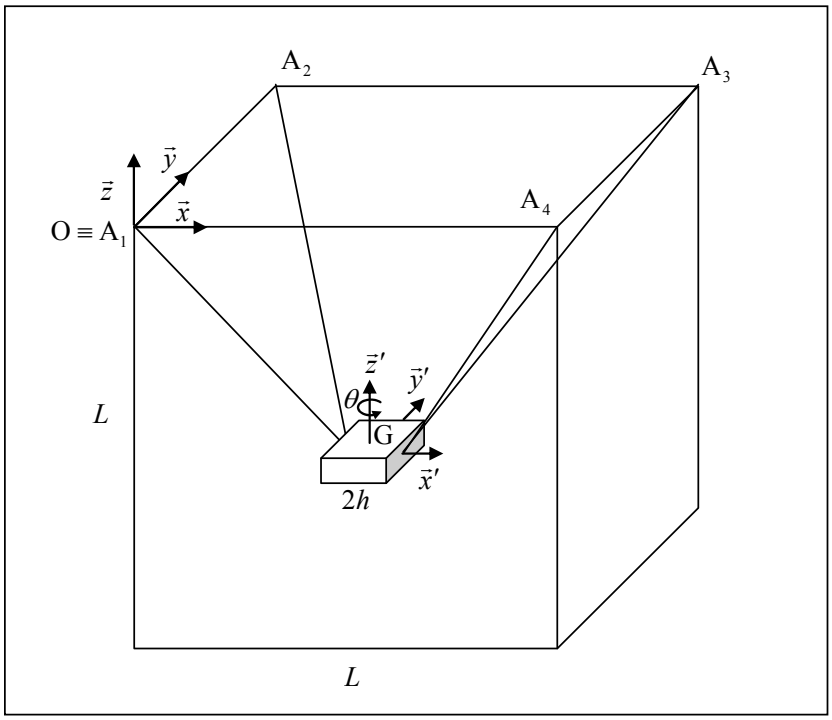


Figure IV.11. Schéma du robot à câbles

2.2 Résultats de simulation

La tâche consiste à déplacer la plateforme mobile du robot à câbles d'une position initiale à une position finale en évitant les obstacles qui se trouvent dans l'espace de travail. Les câbles du robot ne doivent pas entrer en collision avec les obstacles non plus. Les contraintes (IV.1) à (IV.4) doivent être respectées lors du mouvement de la plateforme.

La figure IV.12 montre les résultats de la simulation sous SMAR. Ces résultats correspondent à trois tâches. La première est entre la séquence A et la séquence C. La deuxième est entre C et G. Alors que la dernière est entre G et I. Cela permet d'emmener le robot de manière sûre à la position initiale de la tâche puis de le ramener à sa position de repos.

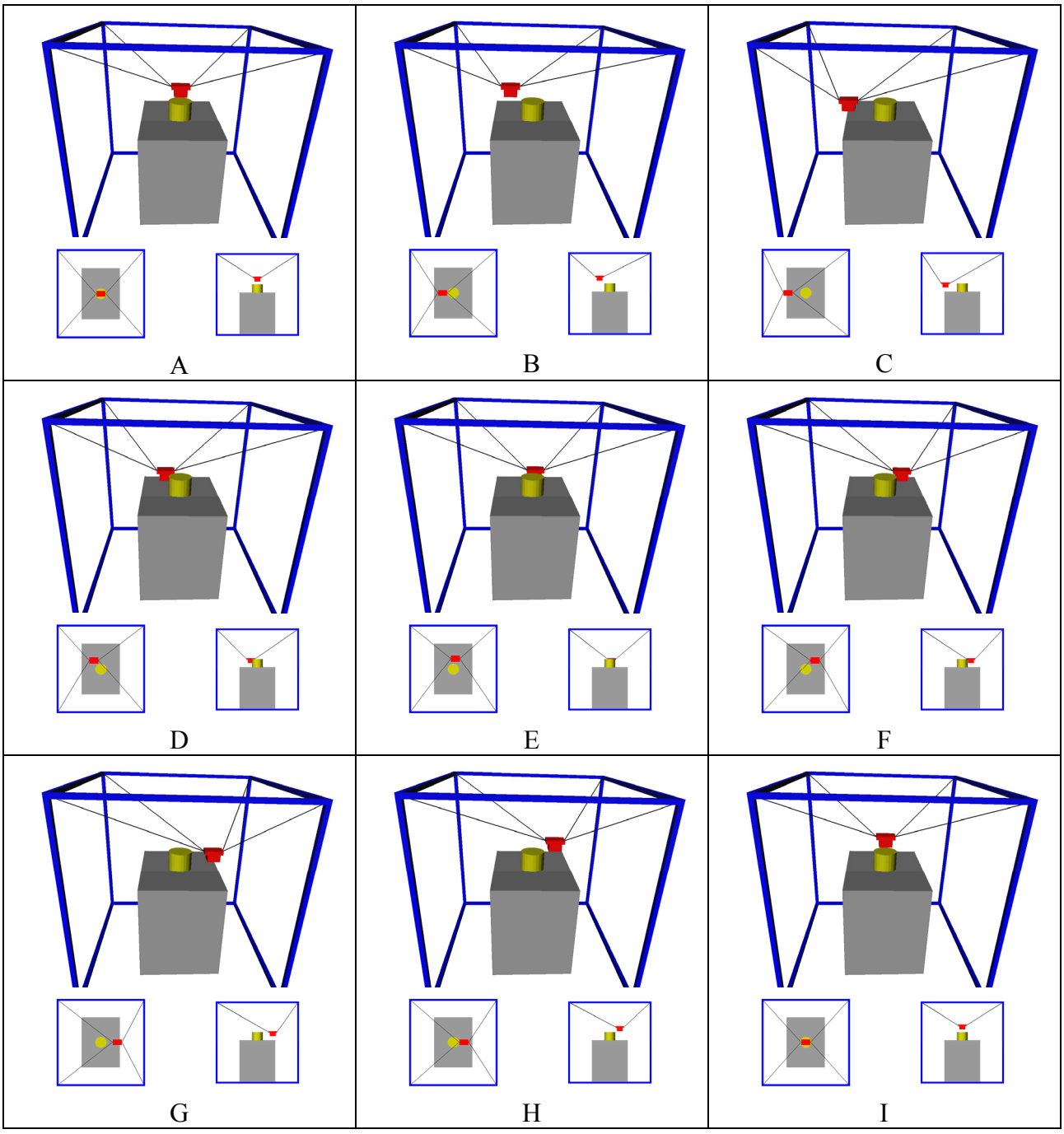


Figure IV.12. Résultats de simulation sous SMAR

La figure IV.13 montre les résultats de simulation sous SMAR d'une deuxième tâche de planification de trajectoire. Dans cette tâche, nous avons voulu montrer l'évitement de collision entre les obstacles et les câbles du robot. Nous avons choisi la position initiale et la position finale de telle sorte que la plateforme mobile soit obligée d'aller vers le haut avant de contourner les obstacles.

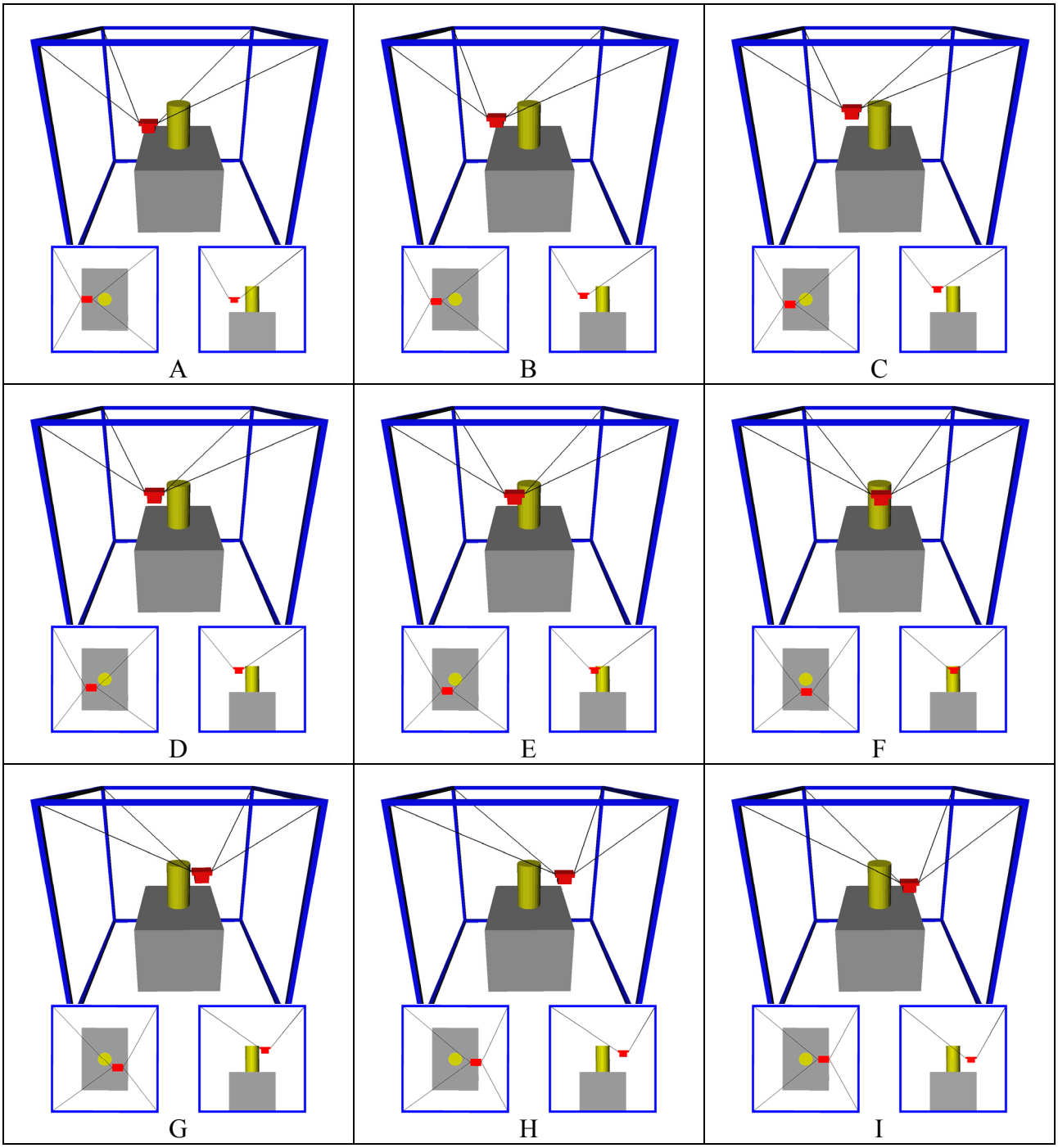


Figure IV.13.Deuxième tâche de planification

2.3 Résultats expérimentaux

Le robot à câbles est contrôlé par une unité de commande industrielle qui dispose de son propre langage de programmation. Cette unité peut être commandée par un utilisateur qui lui transmet les ordres de mouvement l'un après l'autre. Elle peut accepter aussi un fichier de commandes (.cbu) qui sera exécuté automatiquement. Pour commander le robot à câbles du LARM (figure IV.14), nous avons rajouté un module dans SMAR permettant de générer directement des fichiers (.cbu). Ce fichier est généré, soit à partir d'une séquence de positions, soit comme résultat de lancement du module de planification de trajectoires. Ce fichier est ensuite transféré à l'unité de commande qui l'exécute afin d'agir sur les câbles et de faire bouger la plateforme mobile.

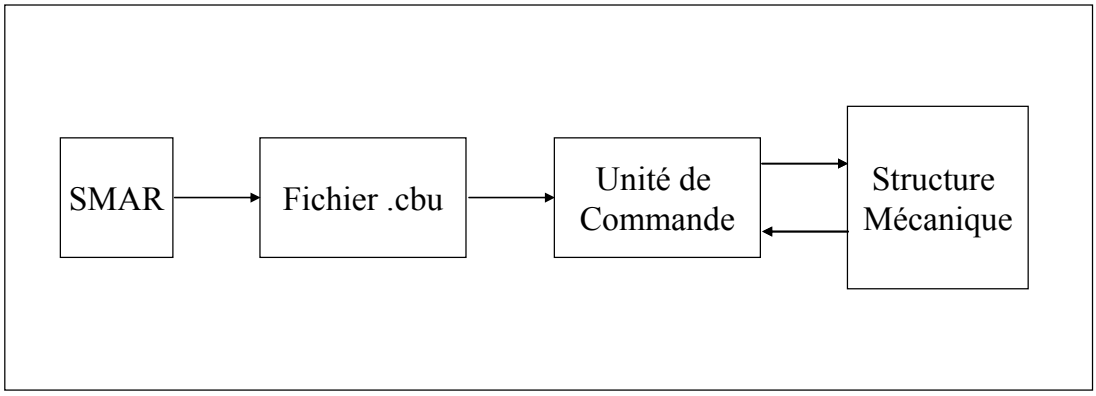


Figure IV.14. Utilisation de SMAR pour commander le robot à câbles

La figure IV.15 montre la variation des longueurs des câbles qui permet de générer le mouvement de la plateforme donné par la simulation sous SMAR.

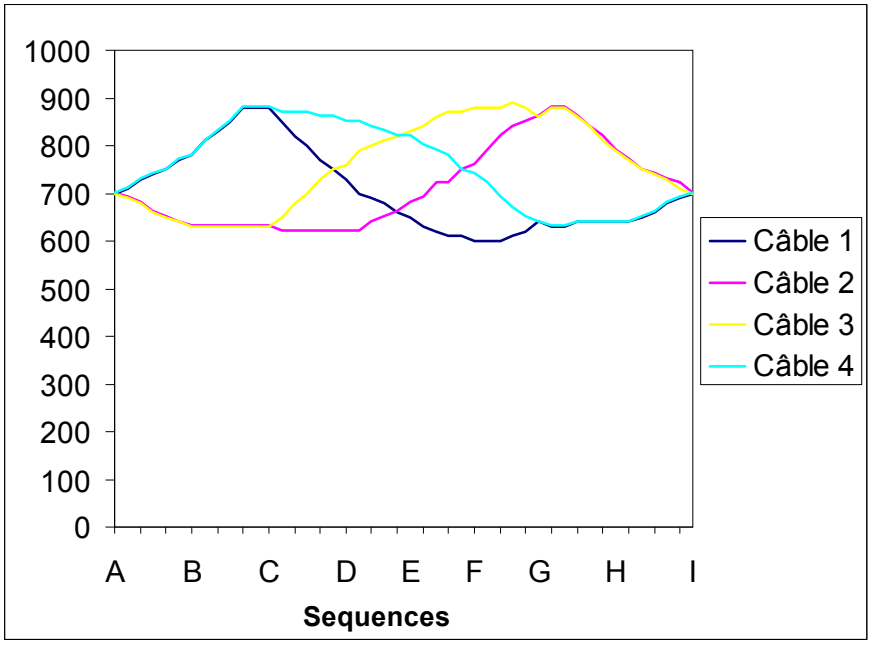
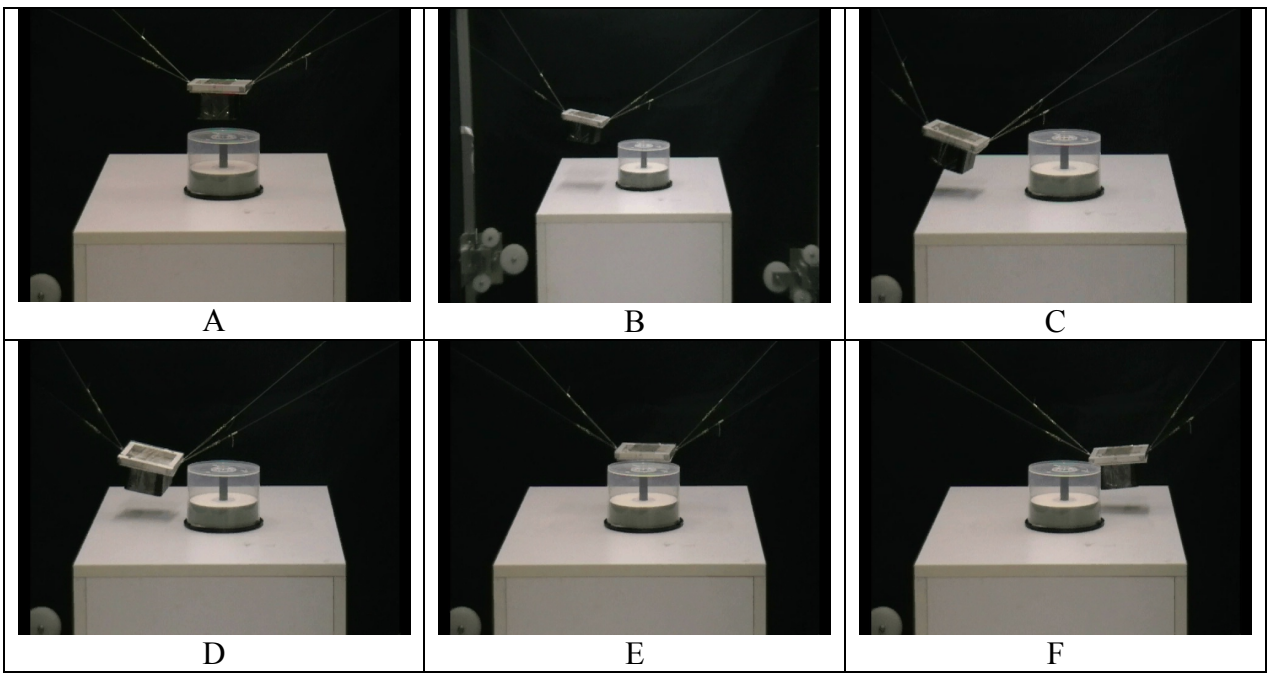
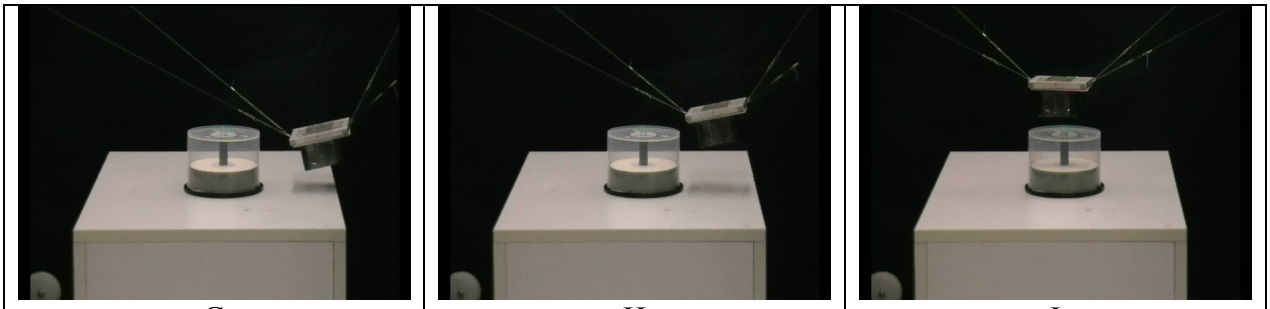


Figure IV.15. Longueur des câbles

La figure IV.16 montre les séquences de mouvement du robot réel. Pour générer ces séquences, nous avons exécuté le fichier cbu correspondant à la tâche de planification montré dans la figure IV.12.





G H I

Figure IV.16. Simulation expérimentale du robot à câbles

3 Le robot chien

Nous disposons d'un robot chien au LMS appelé GIPO (figure IV.17). Ce robot est une transformation du robot humanoïde Robonova de la firme Sud-Coréenne Hitec. GIPO est composé de 17 servomoteurs, dont trois pour chacune des jambes, quatre au niveau de la tête et un pour la queue. Nous avons gardé la carte de commande délivrée avec Robonova. Cette carte est basée sur un microprocesseur ATMEL ATMega128 et peut commander jusqu'à 24 servomoteurs. Elle contient 8 convertisseurs Analogiques/Numériques et une interface RS232.

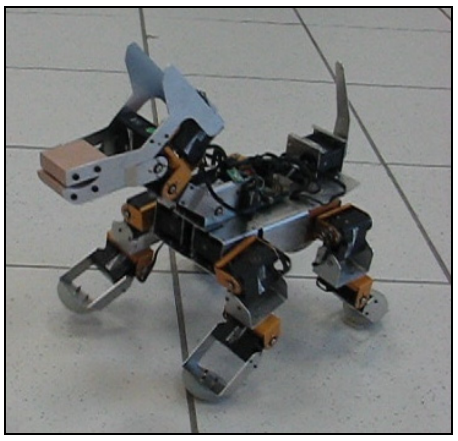


Figure IV.17. GIPO

Nous avons voulu profiter du travail que nous avons effectué pour les robots parallèles afin de générer des mouvements de marche de GIPO. Nous l'avons modélisé sous SMAR comme étant un robot parallèle quelque peu particulier.

3.1 Modélisation

La figure IV.18 montre le modèle de GIPO sous SMAR.

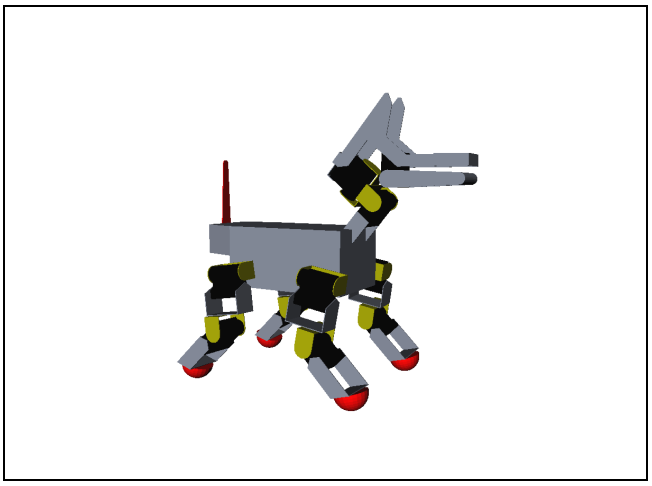


Figure IV.18. Modèle de GIPO sous SMAR

Nous donnons le graphe des liaisons dans la figure IV.19. Dans ce graphe, nous considérons le maillon base 0 lié au sol. Le corps du robot est modélisé par un maillon mobile. Les jambes la queue et la tête sont modélisées par des maillons séries qui viennent s'intercaler sur le maillon mobile. Nous rajoutons à la base, quatre porteurs fictifs de type trois translations (x,y,z). Ces porteurs permettent de déplacer la pointe de chacun des pieds dans l'espace. Il y a quatre liaisons de fermeture définies par des liaisons de coïncidence entre deux points qui relient les porteurs aux jambes.

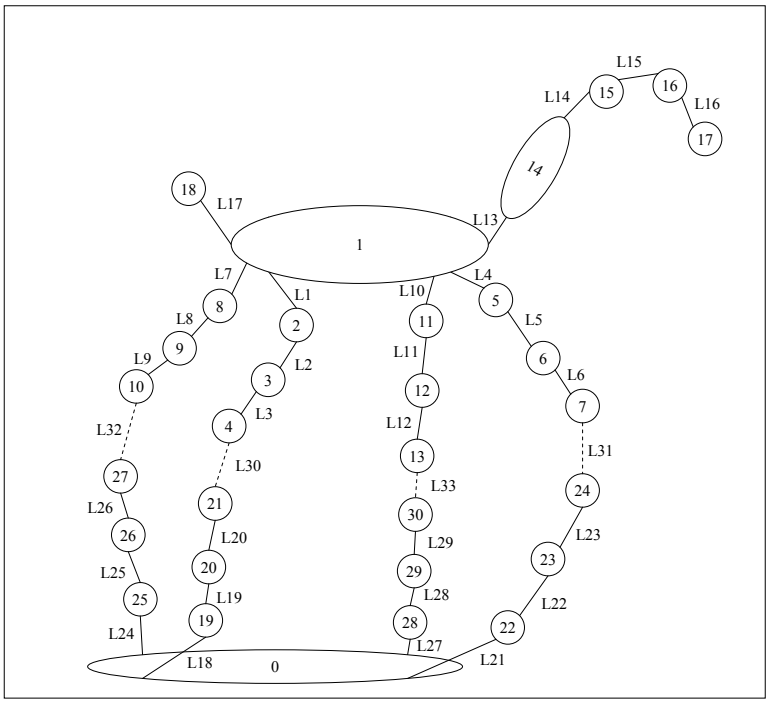


Figure IV.19. Graphe des liaisons

Le tableau IV.3 donne la définition des liaisons permettant de modéliser le robot chien comme étant un robot parallèle.

Table IV.3 Définition des liaisons

Liaison	Type	Point
L1	Pivot d'axe $\vec{y}$	$(58,-50,-38)$
L2	Pivot d'axe $\vec{x}$	$(0,-20,0)$
L3	Pivot d'axe $\vec{y}$	$(-9,-5,-70)$
L4	Pivot d'axe $\vec{y}$	$(118,-50,-38)$
L5	Pivot d'axe $\vec{x}$	$(0,-20,0)$
L6	Pivot d'axe $\vec{y}$	$(9,-5,-70)$
L7	Pivot d'axe $\vec{y}$ $\vec{y}$	$(58,50,-38)$
L8	Pivot d'axe $\vec{x}$	$(0,20,0)$
L9	Pivot d'axe $\vec{y}$	$(-9,5,-70)$
L10	Pivot d'axe $\vec{y}$	$(118,50,-38)$
L11	Pivot d'axe $\vec{x}$	$(0,20,0)$
L12	Pivot d'axe $\vec{y}$	$(9,5,-70)$
L13	Pivot d'axe $\vec{y}$	$(188,0,0)$
L14	Pivot d'axe $\vec{y}$	$(0,0,67)$
L15	Pivot d'axe $\vec{x}$	$(0,0,38)$

L16	Pivot d'axe $\vec{y}$	$(31,0,-35)$
L17	Pivot d'axe $\vec{x}$	$(0,0,0)$
L18	Pivot d'axe $\vec{x}$	$(0,0,0)$
L19	Pivot d'axe $\vec{y}$	$(0,0,0)$
L20	Pivot d'axe $\vec{z}$	$(0,0,0)$
L21	Pivot d'axe $\vec{x}$	$(0,0,0)$
L22	Pivot d'axe $\vec{y}$	$(0,0,0)$
L23	Pivot d'axe $\vec{z}$	$(0,0,0)$
L24	Pivot d'axe $\vec{x}$	$(0,0,0)$
L25	Pivot d'axe $\vec{y}$	$(0,0,0)$
L26	Pivot d'axe $\vec{z}$	$(0,0,0)$
L27	Pivot d'axe $\vec{x}$	$(0,0,0)$
L28	Pivot d'axe $\vec{y}$	$(0,0,0)$
L29	Pivot d'axe $\vec{z}$	$(0,0,0)$

Les liaisons de fermeture de chaînes correspondent à des contraintes de coïncidence entre deux points. Ces contraintes sont définies par le tableau IV.4.

Table IV.4. Définition des contraintes de coïncidence entre deux points

Liaison	Maillon 1	Maillon 2	Point 1	Point 2
L30	21	4	$(37,-79,-155)$	$(-4,1,-57)$
L31	24	7	$(250,-79,-155)$	$(4,1,-57)$
L32	27	10	$(37,79,-155)$	$(-4,-1,-57)$
L33	30	13	$(250,79,-155)$	$(4,-1,-57)$

3.2 Génération des mouvements de marche

Pour générer des mouvements de marche stables, nous partons de l'hypothèse que la projection du centre de gravité sur le sol doit être tout le temps à l'intérieur du polyèdre formé par les points de contact avec le sol (figure IV.20). Nous avons rajouté un module qui permet de trouver le centre de gravité à chaque instant et qui permet de vérifier que la position est stable.

Pour générer des configurations du robot chien, nous agissons sur la position du maillon mobile et nous fixons les positions des pointes des pieds en agissant sur les quatre porteurs. Nous laissons libres les liaisons des jambes. Ces liaisons sont calculées en fonction des

paramètres fixés, automatiquement par SMAR, en utilisant les équations de fermeture de chaînes.

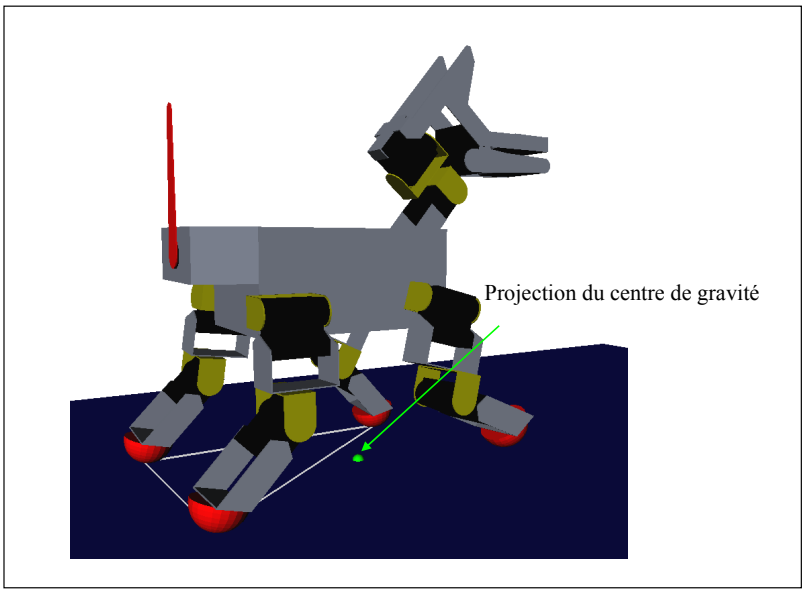


Figure IV.20. Position instable du robot chien

Les mouvements de marche sont générés en choisissant un cycle de levée de pied. Pour la stabilité nous ne levons qu'un seul pied à la fois. Lorsqu'un pied est levé, on vérifie que le robot reste dans une position stable. Si pour le pied levé, le robot devient instable, nous pouvons agir sur la position du maillon mobile pour ramener le robot à une position stable tout en gardant les pieds aux mêmes endroits au sol.

3.2.1 Choix d'une séquence de marche

Dans l'approche que nous proposons afin de générer la marche du robot chien nous choisissons une séquence de levée des pieds, puis un pas et une distance d'avance du tronc pour chaque étape.

Partant d'une configuration initiale où les quatre points de contact entre les pieds et le sol forment un rectangle (figure IV.21 séquence A) et finissant à cette même configuration, la marche se compose de trois phases. La phase de début de la marche consister à ramener le robot à une configuration intermédiaire en avançant seulement deux pieds. La deuxième phase constitue un cycle de marche : elle consiste à déplacer les quatre pieds et de revenir à la configuration intermédiaire. Pour avoir un long mouvement de marche continue, il suffit de répéter cette deuxième phase autant de fois que nécessaire. La dernière phase permet de revenir à la configuration initiale du robot, en déplaçant seulement deux pieds.

Dans le cas de GIPO, le centre de gravité est situé légèrement vers l'avant, ce qui nous impose de commencer par lever l'un des pieds arrières. Nous avons choisi la séquence de levée de pied suivante : pied arrière droit, pied avant droit, pied arrière gauche puis le pied avant gauche.

3.2.2 Planification de trajectoire de la marche

Pour générer la trajectoire de la marche, pour chaque séquence, on impose comme position finale la position de la pointe du pied à lever par l'intermédiaire du porteur, ainsi que la position du tronc. Les autres positions des pointes des pieds sont imposées le long de la planification de trajectoire. Toutes les autres variables sont considérées comme étant passives.

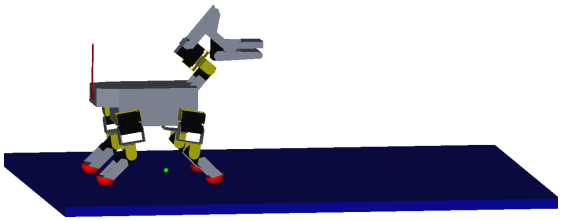
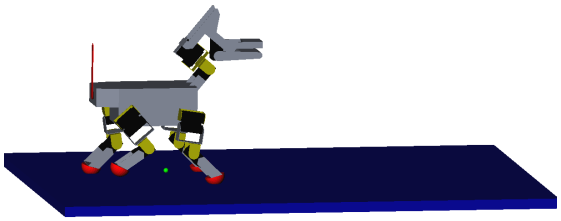
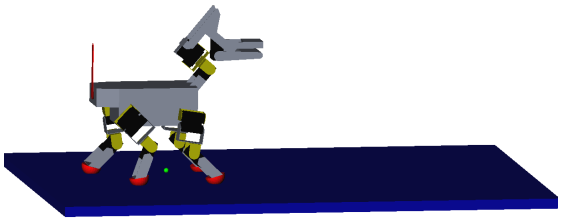
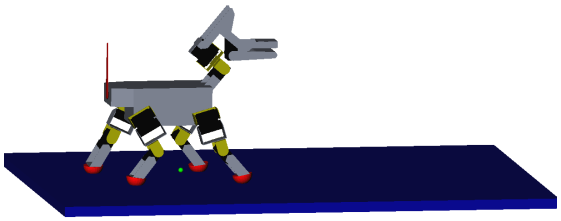
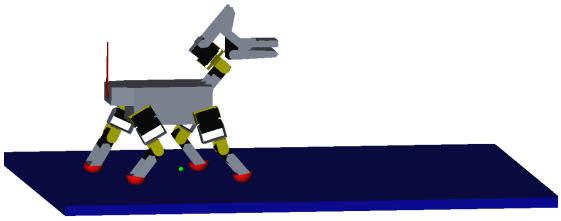
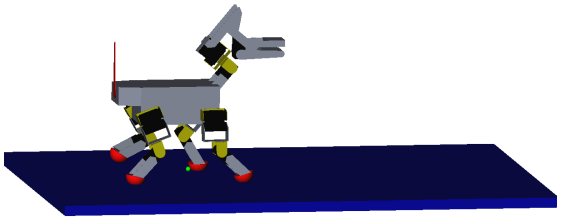
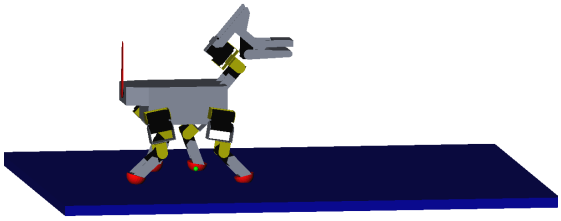
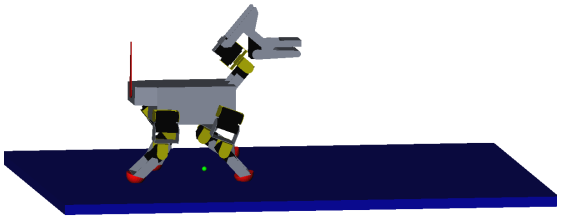
Vu les dimensions du robot nous avons choisi de lever le pied à chaque fois de 20 mm et d'avancer le pied d'un pas de 60 mm. Sachant que pour la première et la dernière phase on n'avance les pieds que d'un demi pas. Nous avons choisi aussi d'avancer le tronc de 15 mm à chaque fois ce qui permet d'avancer le robot d'un pas à chaque cycle et de revenir à la même configuration.

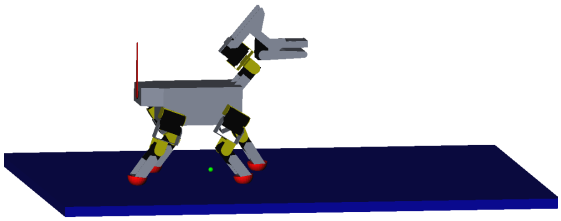
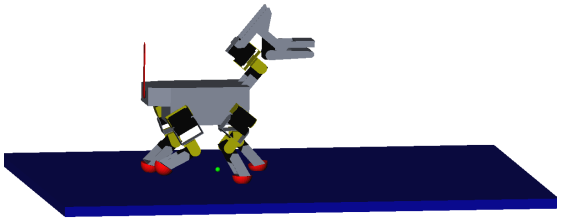
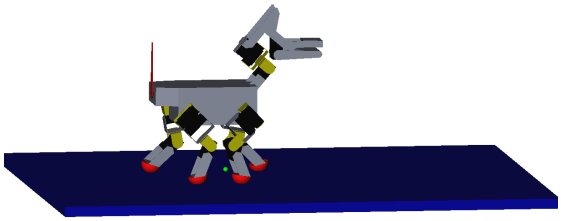
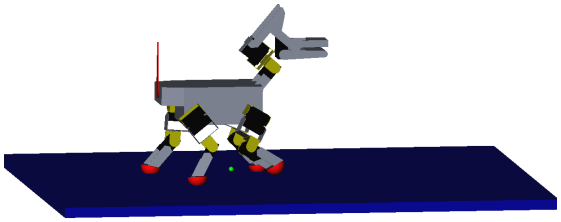
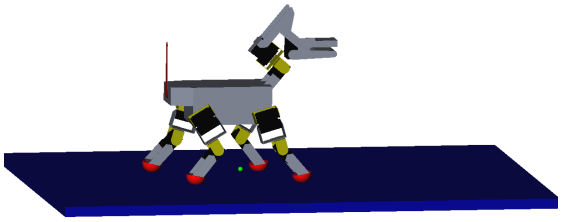
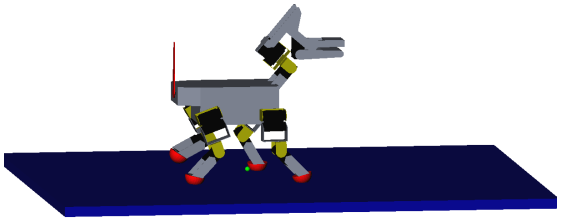
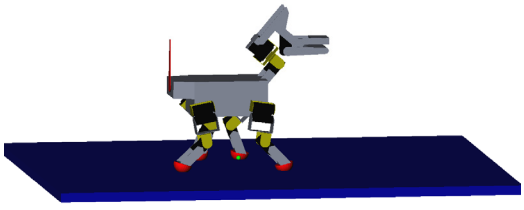
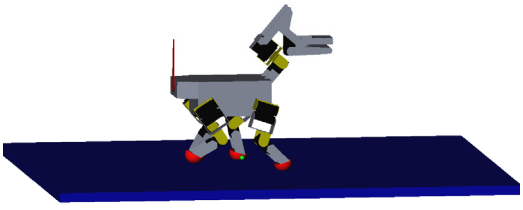
3.3 Mouvement de rotation

Pour générer un mouvement de rotation de 15°, nous avons commencé par chercher la position finale en libérant les porteurs (des pointes des pieds) et en imposant aux articulations du robot de rester fixes. Nous avons ensuite imposé une rotation au tronc du robot, ce qui nous permet de récupérer les positions des pointes des pieds dans la configuration finale du robot. Nous avons divisé se mouvement en quatre séquence de levée de pied, en commençant par lever les pieds les plus proches de leurs positions finales. Le mouvement est obtenu par une planification de trajectoire similaire à celle de la marche pour chacune des séquences.

3.4 Résultats

Les résultats de simulation d'un mouvement de marche sous SMAR, sont donnés dans la figure IV.21.

 A	 B
 C	 D
 E	 F
 G	 H

 I	 J
 K	 L
 M	 N
 O	 P

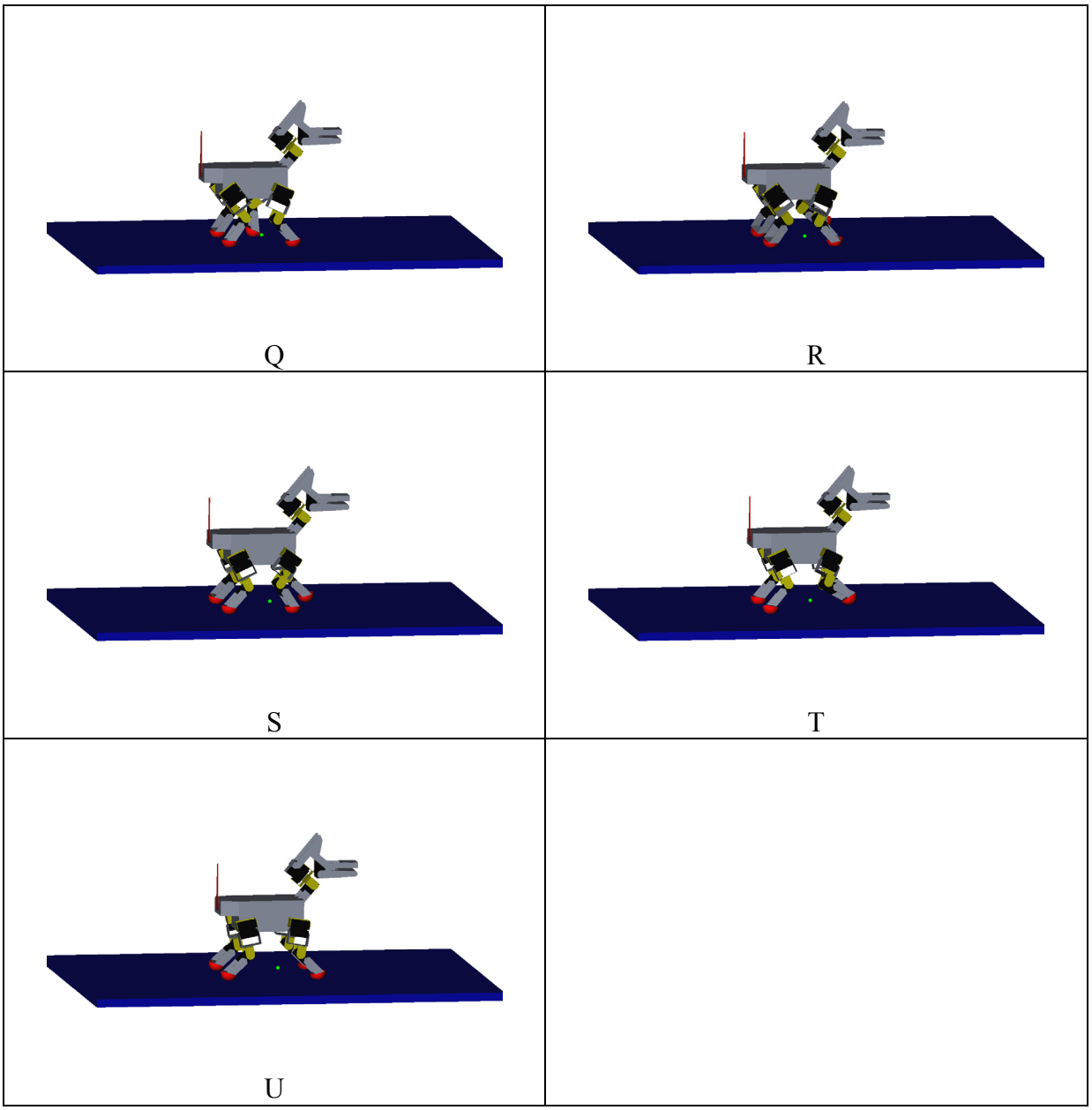


Figure IV.21. Séquences de marche

La marche montrée dans la figure IV.21 se décompose en trois phases. Une phase de départ entre la séquence A et la séquence G. Cette phase permet de commencer la marche à partir de la position initiale. La deuxième phase entre la séquence G et la séquence O, permet de faire un cycle. Ce cycle peut être répété autant de fois que l'on souhaite faire durer la marche. La dernière phase entre la séquence O et la séquence U, permet de remettre le robot dans sa position initiale lorsqu'on souhaite arrêter la marche.

Le robot chien comporte une carte de commande des servomoteurs. Cette carte livrée par Hitec avec un Robonova se programme par un langage spécifique : le robobasic. Nous avons

rajouté un module à SMAR, qui permet de générer un programme en robobasic à partir des mouvements de simulations donnés. Le programme en robobasic est ensuite compilé et transmis à la carte de commande (figure IV.22).

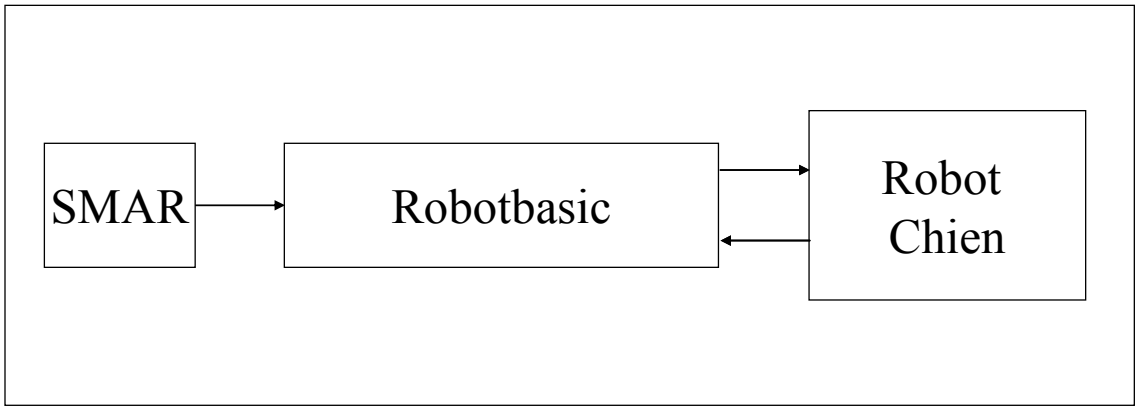
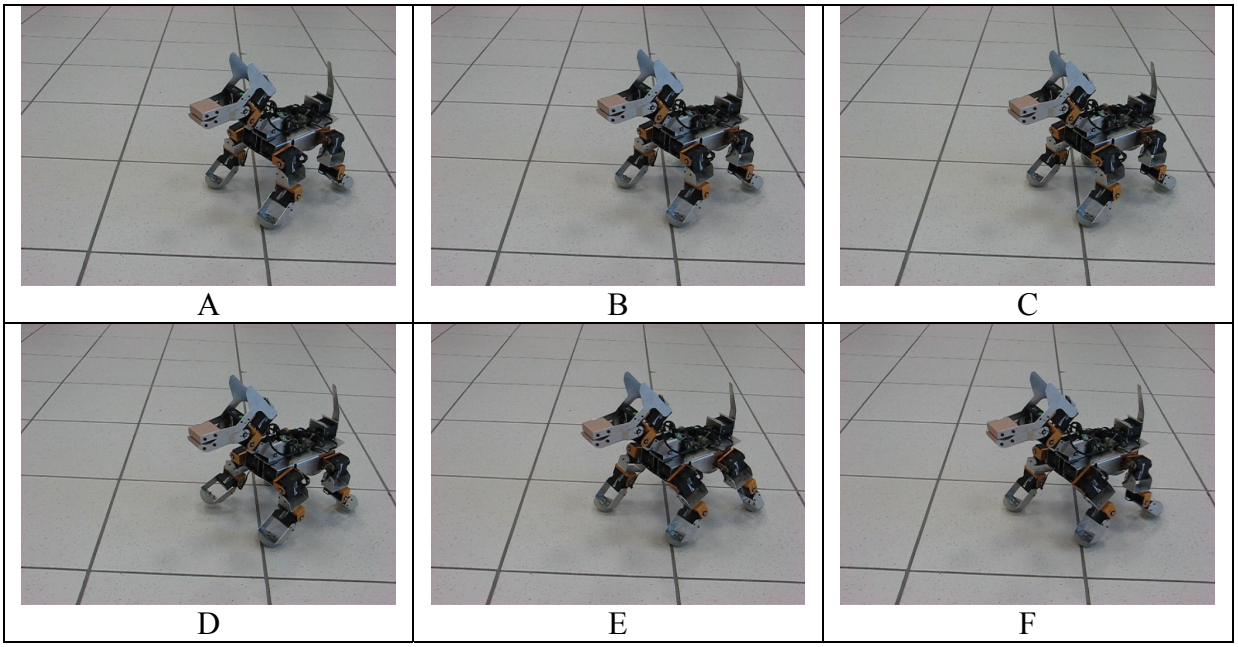


Figure IV.22.Utilisation de SMAR pour commander le GIPO

Les résultats expérimentaux correspondant à la figure IV.21 sont donnés par la figure IV.23.



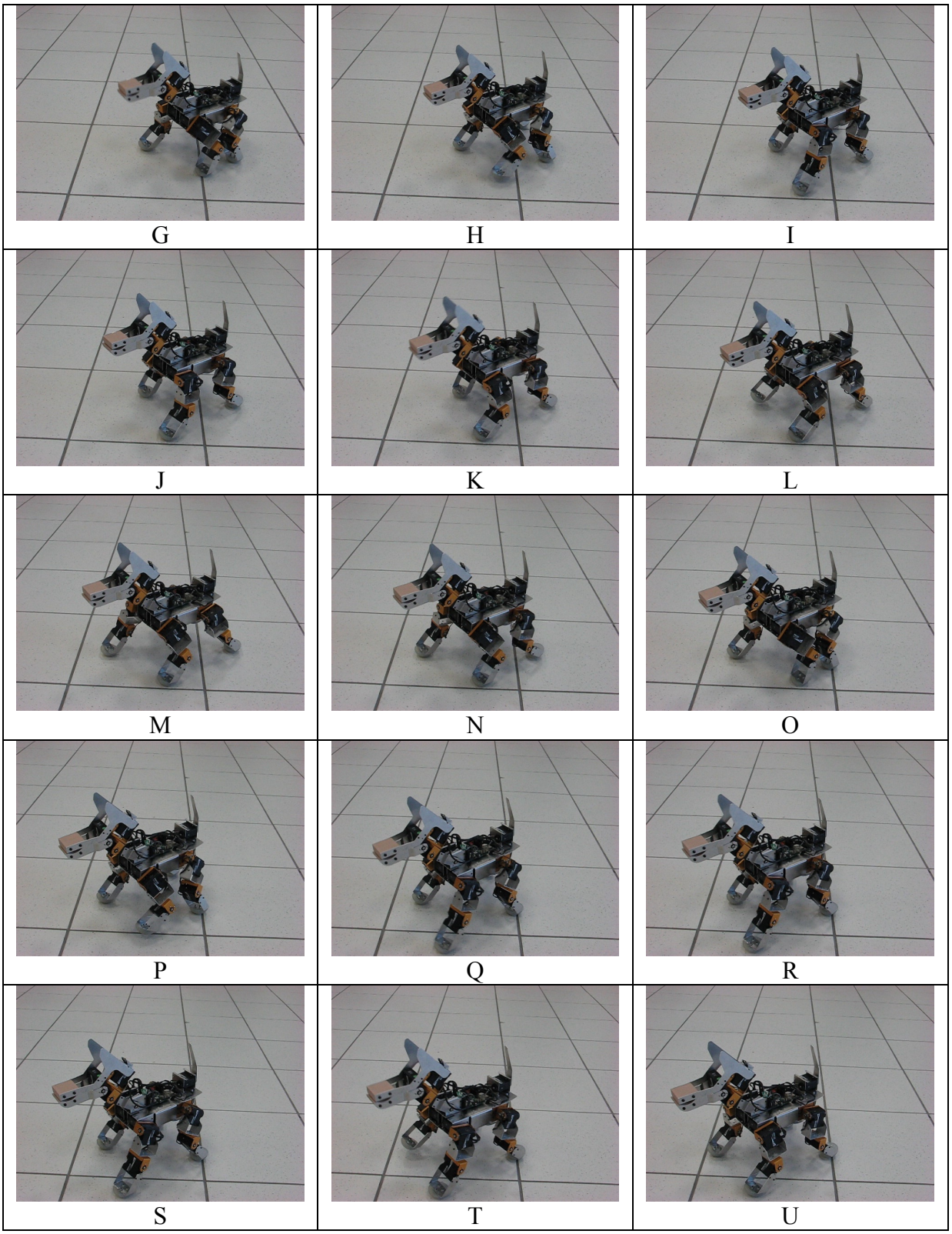


Figure IV.23. Résultats expérimentaux de la marche

La figure IV.24 montre les résultats de simulation sur SMAR d'un mouvement de rotation du robot chien d'un angle de 15°.

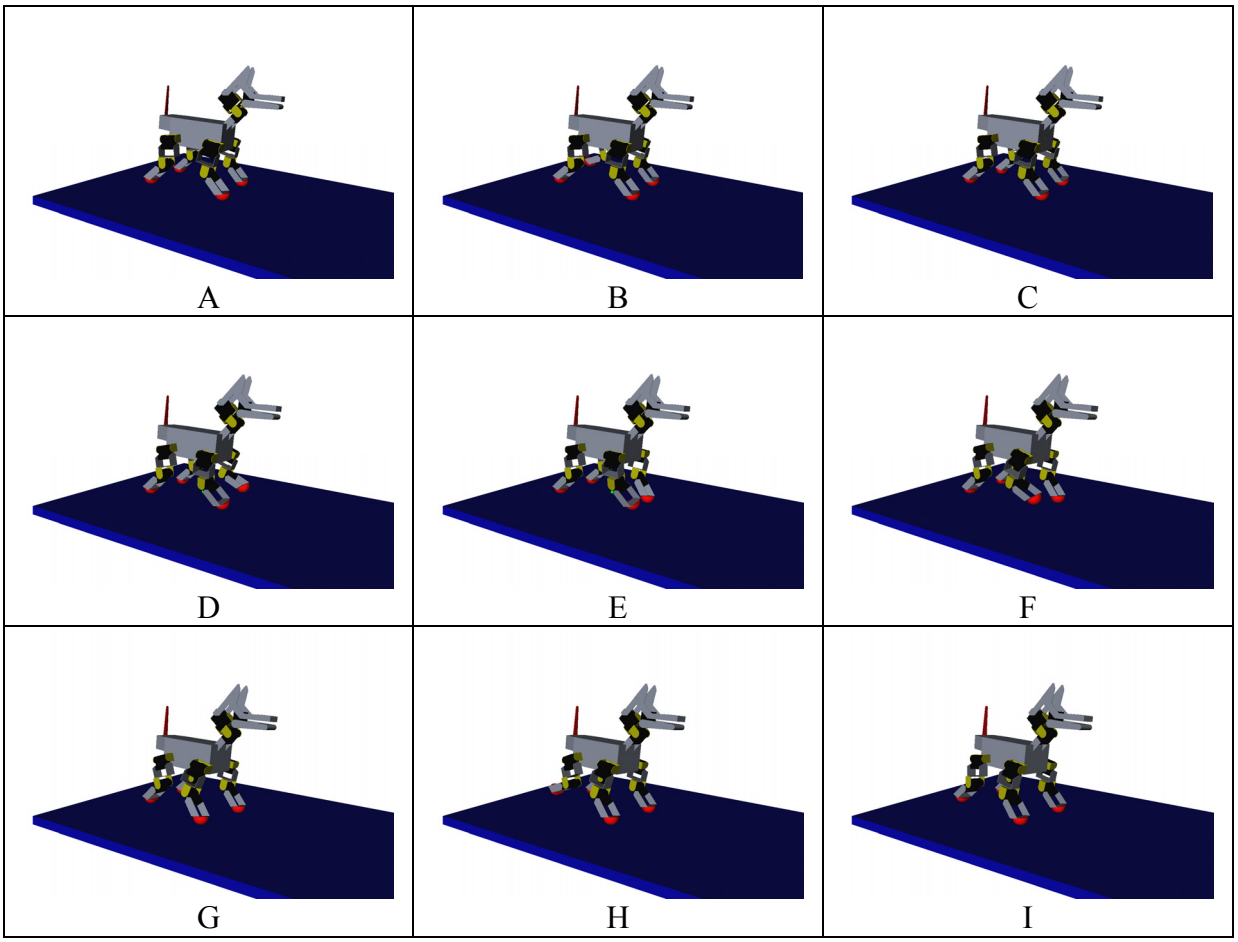
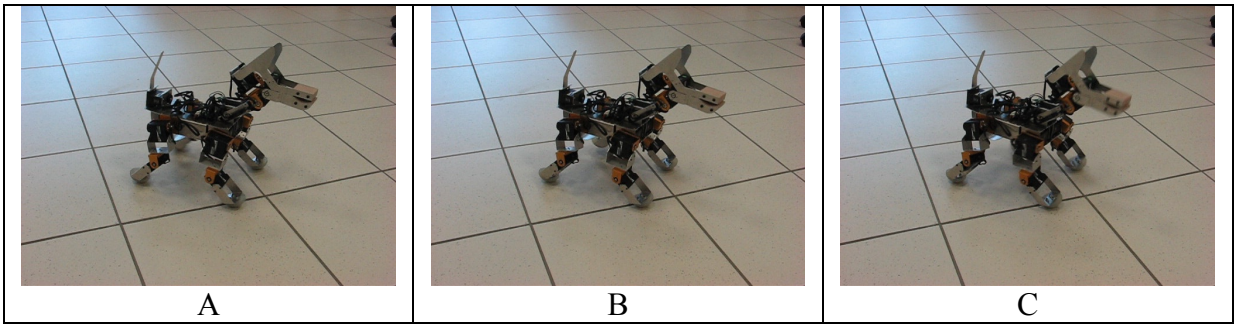


Figure IV.24. Simulation de la rotation de GIPO

Les résultats expérimentaux de la rotation de GIPO, sont montrés dans la figure IV.25.



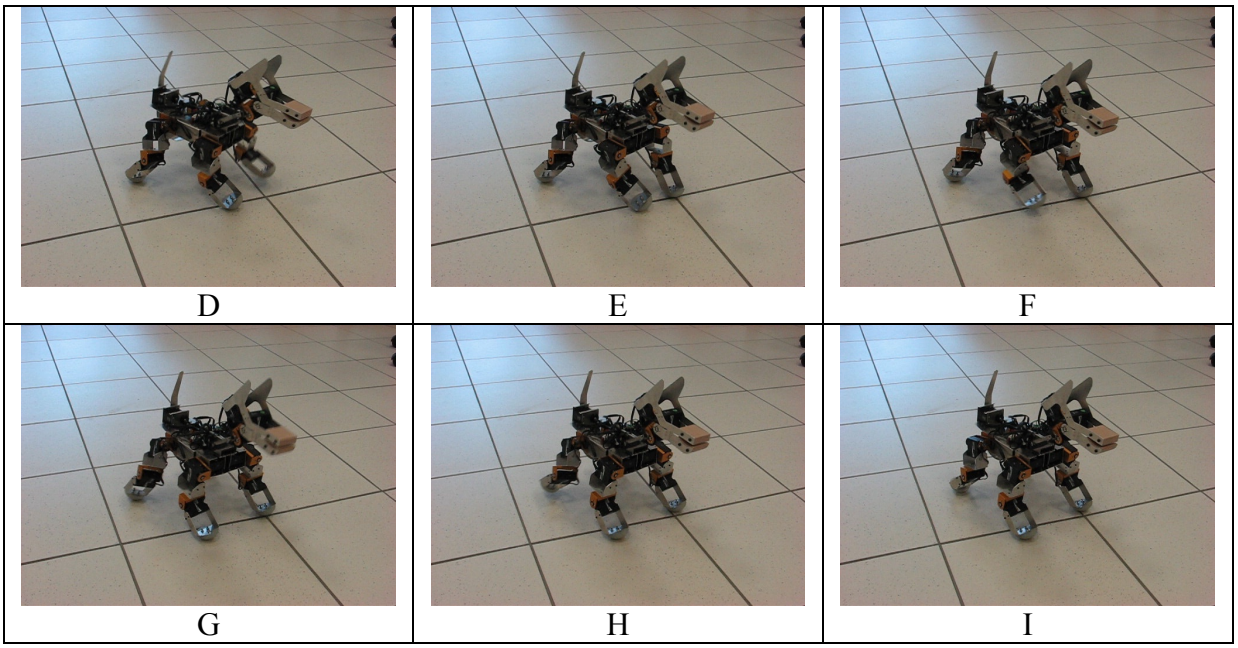


Figure IV.25. Résultats expérimentaux de la rotation de GIPO

Conclusion

Dans ce chapitre nous avons montré les résultats expérimentaux que nous avons obtenus. Pour le robot translateur, comme il a été réalisé au LMS et comme nous maîtrisons toute sa partie commande, nous avons pu le relier directement par une liaison série entre le PC et la partie commande du robot. Ainsi, nous avons pu contrôler le robot à partir de SMAR directement. Les trajectoires sont générées dans un premier temps en simulation, puis, elles sont exécutées sur la structure réelle. Dans ce cas, les résultats de la simulation sont directement exécutés, on arrive à voir à l'écran ce que fait réellement la structure. Pour le robot à câble ce n'était pas le même cas puisque l'unité de commande était indépendante. Les résultats sont passés à travers un fichier généré par SMAR dans le format donné par le constructeur de la partie commande. Pour le robot chien, SMAR donnait les résultats sous forme de programme en basic qu'on recompilait par l'intermédiaire du logiciel donné avec la carte de commande. Le programme est ensuite transmis à la carte de commande du chien à travers une liaison série et la marche est testée expérimentalement sur le robot. Les résultats expérimentaux sont satisfaisants et correspondent bien à la simulation.

Conclusions et perspectives

Dans ce travail, nous nous sommes fixés comme objectif, l'étude de la planification de trajectoires pour les robots parallèles en prenant en considération les collisions et les singularités. Nous avons commencé par étudier les robots parallèles et les notions qui s'y rapportent. Nous avons ainsi, rappelé l'existence de 1140 architectures de robots parallèles plans possibles, dans le seul cas de l'utilisation d'articulations prismatiques et rotoïdes. Il existe également 3850 architectures cinématiques de plateformes de Stewart généralisées. Les chercheurs ont fait des efforts considérables pour regrouper les robots parallèles en classes distinctes permettant de faciliter leur modélisation. Cette modélisation n'est pas évidente étant donnée la diversité des architectures des robots parallèles et le manque de méthodes permettant de générer le modèle géométrique de ces robots. Les modèles géométriques sont soit générés au cas par cas, soit générés numériquement. La difficulté est de pouvoir exprimer automatiquement les variables opérationnelles en fonction des seules variables actionnées en éliminant les variables passives du modèle. Le modèle cinématique de ces robots est également étudié ; il peut au choix, soit s'écrire comme la dérivée par rapport au temps du modèle géométrique, soit s'écrire directement en fonction des torseurs de vitesses associés aux liaisons. L'étude de ce modèle permet d'analyser l'état du robot et permet de détecter les positions singulières à éviter.

Avec l'analyse des configurations singulières, nous avons examiné quelques méthodes de détection de singularités ; ce problème est largement étudié dans la littérature. Les méthodes les plus complètes sont celles qui prennent en compte un modèle cinématique complet contenant toutes les variables du système (aussi bien les variables actives que les variables passives).

Nous nous sommes ensuite intéressés à la planification de trajectoires, la plupart des méthodes proposées vérifient qu'une trajectoire donnée ne passe pas par des configurations singulières ou par des positions mal conditionnées. Parallèlement des travaux ont abordé le problème d'évitement d'obstacles sans s'intéresser toutefois aux singularités. Ainsi les méthodes qui intègrent à la fois les singularités et les collisions sont peu nombreuses dans la littérature.

Afin de développer notre approche de la planification de trajectoires tout en évitant les collisions et les singularités, nous avons évalué quelques logiciels commerciaux aptes à supporter notre développement. Les logiciels de CAO classiques, tels que Solidworks, permettent de modéliser avec précision la géométrie des pièces avec un rendu réaliste. Les

modules permettant de calculer les distances entre les différentes pièces et de détecter les collisions sont intéressants ; en contrepartie le problème de ces logiciels est qu'ils sont peu ouverts au développement et que la gestion des déplacements des pièces et des contraintes associées reste opaque. Les logiciels adaptés aux simulations cinématiques et dynamiques, tels qu'ADAMS, ne sont pas quant à eux, adaptés à la planification de trajectoire. Le passage des singularités et la prise en compte des collisions ne sont pas gérés par ces logiciels. Nous avons également remarqué des problèmes lors de l'utilisation d'ADAMS avec des architectures de robot compliquées présentant quelques degrés d'hyperstatisme.

Nous avons ainsi opté pour le développement de notre propre outil permettant de modéliser les robots parallèles et de simuler leur mouvement tout en prenant en compte la cinématique et les collisions. Pour cela, nous nous sommes appuyés sur le logiciel SMAR développé depuis des années au LMS. Ce logiciel permet à la base, la modélisation et la simulation des robots sériels. Il contient un module permettant le calcul de distance et la détection des collisions. Dans ce travail, nous avons adapté la structure du logiciel pour lui permettre de gérer les architectures des robots parallèles. Les contraintes permettant la modélisation des chaînes fermées ont été ajoutées. Deux méthodes de modélisation des robots parallèles et de génération analytique des équations de fermeture de chaînes ont été développées. Un résultat intéressant a permis de proposer une méthode pour la génération automatique du modèle géométrique des robots sériels. Le modèle cinématique est alors directement obtenu en dérivant les équations de fermeture de chaînes données par les contraintes. Ce modèle cinématique complet contient les variables actives, passives et opérationnelles.

Un effort d'analyse et d'interprétation du modèle cinématique donné par les équations de fermeture de chaînes a été réalisé en s'appuyant sur la théorie classique des mécanismes. Notre contribution dans ce domaine a permis de proposer une méthode générale d'élimination des variables passives pour retrouver automatiquement le modèle cinématique classique, jusqu'alors établi au cas par cas, avec les approches proposées dans la littérature. Cette modélisation simple basée sur l'orthogonalisation de Gramm-Schmidt permet de retrouver le modèle cinématique classique qui relie les variations des variables actives (vitesses articulaires) aux vitesses opérationnelles. Pour obtenir l'évolution des variables passives, il suffit alors d'utiliser simplement le modèle cinématique complet.

La modélisation des robots parallèles (ou des mécanismes en général) en utilisant la notion de contraintes peut aboutir à des équations redondantes. Un travail d'analyse et de simplification de ces équations peut être à notre avis un sujet de recherche important qui complètera cette méthode et qui permettra d'améliorer considérablement les performances du logiciel. L'utilisation de méthodes de l'intelligence artificielle comme les moteurs

d'inférences est une voie intéressante à explorer. Une étude analytique des équations nous paraît également intéressante ; elle permettrait la construction des espaces de travail, la détection des singularités et la synthèse de nouvelles architectures.

Nous avons abordé ensuite le problème de planification de trajectoire en évitant les collisions et les singularités des robots parallèles. Nous avons constaté qu'il était plus facile d'utiliser l'espace de travail ramené à l'organe terminal. Cet espace consiste à considérer le robot comme étant un point (le point défini par l'organe terminal). Les obstacles dans cet espace correspondent à l'ensemble des points définis par la position de l'organe terminal, lorsque le robot est en collision. L'utilisation de cet espace permet à notre avis d'adapter aux robots parallèles la plupart des méthodes de planification de trajectoire sans collision utilisés avec les robots sériels ou les robots mobiles.

Nous avons utilisé une méthode globale à titre d'exemple, afin d'évaluer cette approche. Après avoir intégré cette méthode dans le logiciel SMAR, nous avons testé cette méthode en simulation, puis expérimentalement sur un robot translateur construit au LMS. Les résultats obtenus sont satisfaisants et mettent en valeur la bonne adéquation entre la simulation et l'expérience. Le dispositif expérimental offre des possibilités intéressantes, dans la mesure où le logiciel développé lors de ce travail permet une commande et une visualisation 3D en temps réel du robot. L'ajout de quelques capteurs pourra ainsi permettre la planification en ligne de trajectoire dans des environnements dynamiques.

La méthode de planification de trajectoire a parallèlement été validée sur un robot à câbles construit au LARM. A cette fin, nous avons proposé une méthode de détection de collision entre un objet et un segment permettant de détecter les collisions entre les câbles du robot et les objets se trouvant dans l'espace de travail. Cette méthode très rapide est simple à implémenter. Elle converge rapidement et ne pénalise pas l'algorithme de planification de trajectoire. Les simulations de la planification de trajectoire donnent de bons résultats ; les exemples choisis illustrent en effet l'évitement des collisions entre la plateforme mobile, les câbles et les objets se trouvant dans l'environnement. Toutefois les seules considérations géométriques et cinématiques pour les robots à câbles ne sont pas suffisantes. Il faut également considérer la stabilité statique et dynamique du robot dans la mesure où le poids de la plateforme mobile est une donnée importante à intégrer. Un travail dans ce sens permettra d'améliorer les résultats expérimentaux obtenus.

Finalement, nous avons proposé une méthode permettant de générer la marche d'un robot mobile à pattes. Nous avons ainsi modélisé dans SMAR un robot à quatre pattes de type chien en le considérant comme un robot parallèle. Cette méthode facilite énormément la génération des mouvements de marche. Un module spécifique au robot chien a été développé dans notre logiciel. Ce module permet de déterminer la position du centre de gravité du robot. On peut

ainsi déterminer si le robot est dans une position stable ou non, en fonction de la position du chien et de ses pieds au sol. Les simulations et les expérimentations réalisées ont donné des résultats concluants. Cette méthode peut cependant être améliorée en optimisant la séquence de levée des pieds et en optimisant certains paramètres comme la longueur du pas et la longueur d'avancement du tronc pour chaque pas. Un travail de recherche dans ce sens pourrait venir compléter notre travail.

Annexe A

Modélisation d'un robot sériel Lynx sous SMAR

Le robot Lynx est un robot sériel expérimental commercialisé par Lynxmotion dans un but éducatif. Il est du type 5R et contient une pince au bout. La pince peut être considérée comme deux mécanismes à 4 barres commandés par un seul moteur.

Les moteurs de ce robot sont des servomoteurs commercialisés par Hitec. Le couple délivré par ces moteurs ne dépasse pas les 1,3Nm.

Nous avons modélisé ce robot sous SMAR (voir Figure A.1) en nous basons sur le tableau A.1 définissant les liaisons. Le graphe des liaisons est donné par la figure A.2. Le maillon 6 est un maillon libre qu'on encastre au maillon 5. Ce maillon représente l'organe effecteur, il est modélisé par le repère lié à l'outil dans la figure A.1.

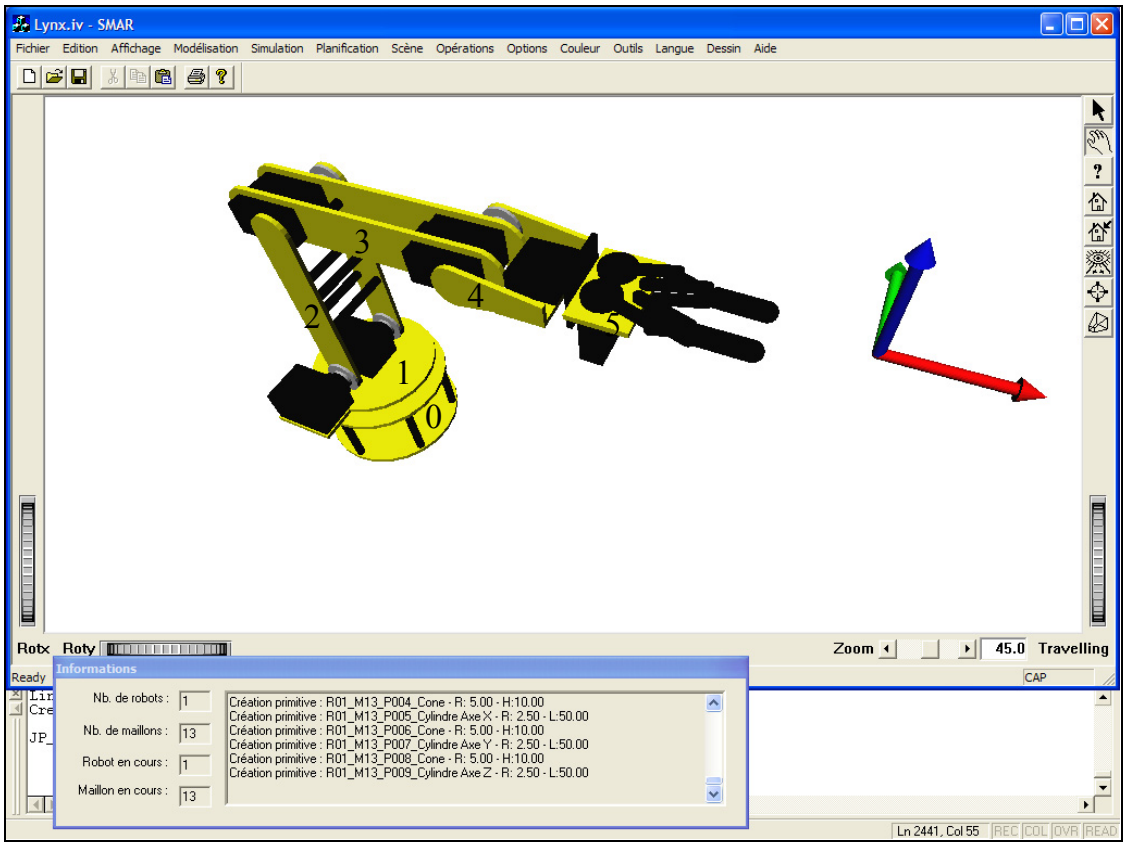


Figure A.1. Le robot Lynx modélisé sous SMAR.

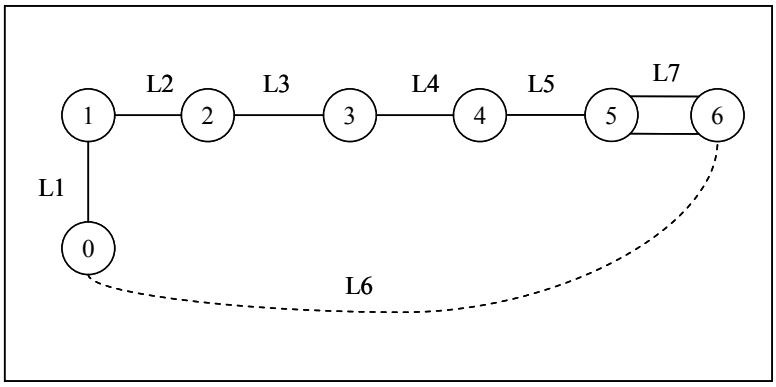


Figure A.2. Graphe des liaisons

Table A.1 Définition des liaisons

Liaison	Type	Point
L1	Pivot d'axe $\vec{z}$	$(0,0,60)$
L2	Pivot d'axe $\vec{y}$	$(0,0,13)$
L3	Pivot d'axe $\vec{y}$	$(0,0,121)$
L4	Pivot d'axe $\vec{y}$	$(116,0,0)$
L5	Pivot d'axe $\vec{x}$	$(55,0,0)$

La liaison L6 est une liaison libre, elle permet de donner les six variables opérationnelles qui donnent la position de l'organe effecteur. La liaison L7 correspond à la contrainte d'encastrement entre le maillon 5 et le maillon 6. Le repère du maillon 6 est placé au point de coordonnées $(135,0,0)$ par rapport au repère lié au maillon 5 et l'orientation du repère du maillon 6 est la même que celle du repère du maillon 5. Cette contrainte d'encastrement donne 12 équations correspondant aux termes de la matrice homogène.

$$(c_0c_1c_2 - c_0s_1s_2)c_3 - (c_0c_1s_2 + c_0s_1c_2)s_3 - c_6c_7 = 0 \quad (A.1)$$

$$-s_0c_4 + ((c_0c_1c_2 - c_0s_1s_2)s_3 + (c_0c_1s_2 + c_0s_1c_2)c_3)s_4 + c_6s_7 = 0 \quad (A.2)$$

$$s_0s_4 + ((c_0c_1c_2 - c_0s_1s_2)s_3 + (c_0c_1s_2 + c_0s_1c_2)c_3)c_4 - s_6 = 0 \quad (A.3)$$

$$190(c_0c_1c_2 - c_0s_1s_2)c_3 - 190(c_0c_1s_2 + c_0s_1c_2)s_3 + 116c_0c_1c_2 - 116c_0s_1s_2 + 121c_0s_1 - x_8 = 0 \quad (A.4)$$

$$(s_0c_1c_2 - s_0s_1s_2)c_3 - (s_0c_1s_2 + s_0s_1c_2)s_3 - s_5s_6c_7 - c_5s_7 = 0 \quad (A.5)$$

$$c_0c_4 + ((s_0c_1c_2 - s_0s_1s_2)s_3 + (s_0c_1s_2 + s_0s_1c_2)c_3)s_4 + s_5s_6s_7 - c_5c_7 = 0 \quad (A.6)$$

$$-c_0s_4 + ((s_0c_1c_2 - s_0s_1s_2)s_3 + (s_0c_1s_2 + s_0s_1c_2)c_3)c_4 + s_5c_6 = 0 \quad (A.7)$$

$$190(s_0c_1c_2 - s_0s_1s_2)c_3 - 190(s_0c_1s_2 + s_0s_1c_2)s_3 + 116s_0c_1c_2 - 116s_0s_1s_2 + 121s_0s_1 - x_9 = 0 \quad (A.8)$$

$$(-s_1c_2 - c_1s_2)c_3 - (-s_1s_2 + c_1c_2)s_3 + c_4s_6c_7 - s_4s_7 = 0 \quad (A.9)$$

$$((-s_1c_2 - c_1s_2)s_3 + (-s_1s_2 + c_1c_2)c_3)s_4 - c_4s_6s_7 - s_4c_7 = 0 \quad (A.10)$$

$$((-s_1c_2 - c_1s_2)s_3 + (-s_1s_2 + c_1c_2)c_3)c_4 - c_4c_6 = 0 \quad (A.11)$$

$$190(-s_1c_2 - c_1s_2)c_3 - 190(-s_1s_2 + c_1c_2)s_3 + 73 - 116s_1c_2 - 116c_1s_2 + 121c_1 - x_{10} = 0 \quad (A.12)$$

Ces équations sont liées, nous avons en réalité besoin que de 6 équations parmi ces 12.

Annexe B

Equations cinématiques d'une plateforme de Stewart

Le modèle géométrique direct est donné pour chaque jambe i par l'éqution (II.73). En dérivant cette équation, on obtient :

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & & & \\ & \ddots & 0 & \\ & & \mathbf{M}_i & \\ & 0 & & \ddots \\ & & & \mathbf{M}_6 \end{bmatrix} \tag{B.1}$$

Avec

$$\mathbf{M}_i \begin{bmatrix} \sin q_{piy} \\ -\sin q_{pix} \cos q_{piy} \\ -\cos q_{pix} \cos q_{piy} \end{bmatrix} \tag{B.2}$$

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & & & \\ & \ddots & 0 & \\ & & \mathbf{N}_i & \\ & 0 & & \ddots \\ & & & \mathbf{N}_6 \end{bmatrix} \tag{B.3}$$

$$\mathbf{N}_i = \begin{bmatrix} 0 & q_i \cos q_{piy} \\ -q_i \cos q_{pix} \cos q_{piy} & q_i \sin q_{pix} \sin q_{piy} \\ q_i \sin q_{pix} \cos q_{piy} & q_i \cos q_{pix} \sin q_{piy} \end{bmatrix} \tag{B.4}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & & & & \\ & \ddots & & 0 & \\ & & \mathbf{K}_i & & \\ & 0 & & \ddots & \\ & & & & \mathbf{K}_6 \end{bmatrix} \tag{B.5}$$

avec

$$\mathbf{K}_i = \begin{bmatrix} -1 & 0 & 0 & 0 & k_{i1} & k_{i2} \\ 0 & -1 & 0 & k_{i3} & k_{i4} & k_{i5} \\ 0 & 0 & -1 & k_{i6} & k_{i7} & k_{i8} \end{bmatrix} \tag{B.6}$$

$$k_{i1} = b_{ix} \sin\beta \cos\gamma - b_{iy} \sin\beta \sin\gamma - b_{iz} \cos\beta \tag{B.7}$$

$$k_{i2} = b_{ix} \cos\beta \sin\gamma + b_{iy} \cos\beta \cos\gamma \tag{B.8}$$

$$k_{i3} = b_{ix}\left(-\cos\alpha \sin\beta \cos\gamma + \sin\alpha \sin\gamma\right) + b_{iy}\left(\cos\alpha \sin\beta \sin\gamma + \sin\alpha \cos\gamma\right) - b_{iz} \cos\alpha \cos\beta \tag{B.9}$$

$$k_{i4} = -b_{ix} \sin\alpha \cos\beta \cos\gamma + b_{iy} \sin\alpha \cos\beta \sin\gamma + b_{iz} \sin\alpha \sin\beta \tag{B.10}$$

$$k_{i5} = b_{ix}\left(\sin\alpha \sin\beta \sin\gamma - \cos\alpha \cos\gamma\right) + b_{iy}\left(\sin\alpha \sin\beta \cos\gamma + \cos\alpha \sin\gamma\right) \tag{B.11}$$

$$k_{i6} = b_{ix}\left(-\sin\alpha \sin\beta \cos\gamma - \cos\alpha \sin\gamma\right) + b_{iy}\left(\sin\alpha \sin\beta \sin\gamma - \cos\alpha \cos\gamma\right) + b_{iz} \sin\alpha \cos\beta \tag{B.12}$$

$$k_{i7} = b_{ix} \cos\alpha \cos\beta \cos\gamma - b_{iy} \cos\alpha \cos\beta \sin\gamma + b_{iz} \cos\alpha \sin\beta \tag{B.13}$$

$$k_{i8} = b_{ix}\left(-\cos\alpha \sin\beta \sin\gamma - \sin\alpha \cos\gamma\right) + b_{iy}\left(-\cos\alpha \sin\beta \cos\gamma - \sin\alpha \sin\gamma\right) \tag{B.14}$$

Pour trouver les matrices $\mathbf{A}$ et $\mathbf{B}$, il suffit de multiplier les matrices $\mathbf{M}$ et $\mathbf{K}$ par $\mathbf{M}^T$. On obtient :

$$\mathbf{B} = \mathbf{I}_{6\times 6} \tag{B.15}$$

Où $\mathbf{I}_{6\times6}$ la matrice identité.

$$\mathbf{A} = \mathbf{M}^T\mathbf{K} \tag{B.16}$$

Bibliographie

[Affi 04] Z. Affi, L. Romdhane et A. Maalej, « Dimensional synthesis of a 3-translational-DOF in-parallel manipulator for a desired workspace », European Journal of Mechanics – A/Solids 23 (2), pp. 311–324. 2004.

[Alba 07] O. Alba-Gomez, « Optimisation de trajectoires de robots parallèles redondants », Thèse de doctorat de l'école centrale de Nantes et de l'institut technologique de la Laguna. Septembre 2007.

[Battacharya 98] S. Battacharya, H. Hatwal et A. Ghosh, « Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators », Mechanism and Machine Theory 33 (7) 965-974. 1998.

[Ben 05] A. Ben Sghaier et L. Romdhane, « A software package for parallel mechanisms modeling and simulation ». In Computational Kinematics, Cassino, 4-6 Mai 2005.

[Bohlin 00] R. Bohlin et L. Kavraki, « Path planning using lazy prm », Proceedings of IEEE International Conference on Robotics and Automation, pp. 521–528, San Francisco, CA, Avril 2000.

[Bonev 03] I. Bonev, D. Zlatanov, C. Gosselin, « Singularity Analysis of 3-DOF Planar Parallel Mechanisms via screw theory », ASME Journal of Mechanical Design, volume 125, pp. 573-581, 2003.

[Bouchard 06] S. Bouchard et C. Gosselin, « Kinematic sensitivity of a very large cable-driven parallel mechanism », Proceedings of IDETC/CIE ASME 2006 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Philadelphia, Pennsylvania, Septembre 2006.

[Chablat 98] D. Chablat, « Domaine d'unicité et parcourabilité pour les manipulateurs pleinement parallèles », Thèse de doctorat de l'école centrale de Nantes. Novembre 1998.

[Charentus 90] S. Charentus, « Modélisation et commande d'un robot manipulateur redondant composé de plusieurs plate-formes », Thèse de doctorat de l'Université Paul Sabatier, Toulouse, 13 Avril 1990.

[Clavel 88] R. Clavel, « DELTA, a fast robot with parallel geometry », 18th Int. Symp. on Industrial Robots (ISIR), pages 91-100, Lausanne, 26-28 Avril 1988.

[Cortès 03] J. Cortés, « Motion Planning Algorithms for General Closed-Chain Mechanisms », Thèse de doctorat de l'INP de Toulouse Décembre 2003.

[Dash 05] A. Dash, I-Meng Chen, S. Yeo et G. Yang, « Workspace generation and planning singularity-free path for parallel manipulators », Mechanism and Machine Theory 40, 776-805, 2005.

[Denavit 55] J. Denavit et R. S. Hartenberg, « A Kinematic Notation for Lower-Pair Mechanisms based on Matrices », ASME Journal of Applied Mechanics, Juin 1955.

[Dietmaier 98] P. Dietmaier, « The Stewart-Gough platform of general geometry can have 40 real postures », ARK, pages 7-16, Strobl, 29 Juin-4 Juillet, 1998.

[Favejon 87] B. Faverjon et P. Touranssoud, « A local based approach for path planning of manipulators with a high number of degrees of freedom », Proceedings of IEEE International Conference on Robotics and Automation, pp. 1152–1159, Raleigh, Mars 1987.

[Gao 06] Xiao-Shan Gao, D. Lei; Qizheng Liao, Gui-Fang Zhang, « Generalized Stewart-Gough platforms and their direct kinematics », IEEE Transactions on Robotics, pages 141-151, Vol 21, Issue 2, Avril 2005.

[Gosselin 88] C. Gosselin, « Kinematic analysis optimization and programming of parallel robotic manipulators ». Thèse de doctorat de McGill University, Montréal, 15 Juin 1988.

[Gosselin 90] C. Gosselin, J. Angeles, « Singularity analysis of closed loop kinematic chains », IEEE, transaction on Robotics and Automation. Vol. 6, pp. 281-290, Juin 1990.

[Gough 56] V. E. Gough, « Contribution to discussion of papers on research in automobile stability, control and tyre performance », Proc. Auto Div. Inst. Mech. Eng., 1956-1957.

[Guanfeng 03] Guanfeng Liu, Yunjiang Lou et Zexiang Li, « Singularities of Parallel Manipulators, A Geometric Treatment », IEEE Transactions On Robotics and Automation, Vol. 19, N. 4, Août 2003.

[Hayes 03] M.J.D. Hayes, M.L. Husty, « On the kinematic constraint surfaces of general three-legged planar robot platforms », Mechanism and Machine Theory 38, pp. 379–394, 2003.

[Hayes 04] M.J.D. Hayes, P.J. Zsombor-Murray, C. Chen, « Unified Kinematic Analysis of General Planar Parallel Manipulators » ASME Journal of Mechanical Design, vol 126, no 5, pp. 866-874. septembre 2004.

[Hayward 89] V. Hayward, et R. Kurtz, « Preliminary study of serial-parallel redundant manipulator », NASA Conference on Space Telerobotics, pages 39-48, Pasadena, 31 Janvier 1989.

[Helguera 01] C. Helguera, « Contribution à la résolution du problème de minima locaux dans une méthode de planification de trajectoires sans collision pour robots manipulateurs. Thèse de l'université de Poitiers. Avril 2001.

[Kavraki 96] L. Kavraki; P. Svestka, J-C. Latombe et M. Overmars, « Probabilistic roadmaps for path planning in high dimensional configuration spaces », IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566–580, Août 1996.

[Khatib 86] O. Khatib, « Real-time obstacle avoidance for manipulators and mobile robots », The International Journal of Robotics Research, Vol. 5, No. 1, pp. 90–99, 1986.

[Kondo 91] K. Kondo, « Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free-space enumeration », Proceedings of IEEE International Conference on Robotics and Automation, pp. 267–277, Sacramento, CA, Avril 1991.

[Lahouar 05] S. Lahouar, S. Zeghloul et L. Romdhane, « Path planning for manipulator robots in cluttered environments », WSEAS TRANSACTIONS On Systems Issue 5, Volume 4, pp. 555-560, Mai 2005.

[Lahouar 06a] S. Lahouar, S. Zeghloul et L. Romdhane, « Collision free path planning for Multi-DoF Manipulators », Dans le livre « Industrial

Robotics: Theory, Modeling and Control ». ISBN 3-86611-285-8. Pro Literatur Verlag, Décembre 2006.

[Lahouar 06b] S. Lahouar, S. Zeghloul et L. Romdhane, « Real time path planning for multi-DOF robot manipulators in dynamic environment », International Journal of Advanced Robotic Systems, Volume 3, Number 2, pp. 125-132, Juin 2006.

[Lallemand 94] J.-P. Lallemand et S. Zeghloul. « Robotique Aspects fondamentaux, Modelisation mécanique, CAO robotique, Commande », ISBN 2-225-84354-6. MASSON, Paris 1994.

[Lallemand 97] J-P. Lallemand, A. Goudali, et S. Zeghloul, « The 6-dof 2-Delta parallel robot », Robotica, 15(4):407-416, Juillet- Août, 1997.

[LaValle 06] S. LaValle, « Planning Algorithms », Cambridge University Press, ISBN 0521862051. 2006.

[Lee 88] K-M. Lee et D. Shah, « Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator », IEEE Journal of Robotics and Automation, 4(3):354-360, Juin 1988.

[Lengyel 90] J. Lengyel; M. Reichert; B. Donald et D. Greenberg, « Real-time robot motion planning using rasterizing computer graphics hardware », Computer Graphics, Vol. 24, No. 4, pp. 327–335. Août 1990.

[McCloy 90] D. McCloy, « Some comparisons of serial-driven and parallel-driven manipulators », Robotica, volume 8, pp. 355-362, 1990.

[Merlet 01] J-P. Merlet, « A Generic Trajectory Verifier for the Motion Planning of Parallel Robots », Transactions of the ASME 2001.

[Merlet 89] J-P. Merlet, « Singular configurations of parallel manipulators and Grassmann Geometry », The international journal of robotics research Vol. 8, N. 5, Octobre 1989.

[Merlet 97] J-P. Merlet, « Les robots parallèles », 2eme edition, HERMES, Paris, 1997.

[Oh 03] S. Oh et S. Agrawal, « Cable-suspended planar parallel robots with redundant cables: controller with positive cable tensions », In IEEE Int. Conf. on Robotics and Automation, pages 3023-3028, Taipei, 14-19, Septembre 2003.

[Ottaviano 02] E. Ottaviano, M. Ceccarelli, M. Toti, C. Avila Carrasco, « CaTraSys (Cassino Traking System): A Wire System for Experimental Evaluation of Robot Workspace », Journal of Robotics and Mechatronics, Vol. 14, n.1, pp.78-87, 2002.

[Paden 89] B. Paden, A. Mess et M. Fisher, « Path planning using a jacobian-based free space generation algorithm » Proceedings of IEEE International Conference on Robotics and Automation, pp. 1732–1737, Scottsdale, Arizona, 1989.

[Pierrot 91] F. Pierrot, « Robot pleinement parallèle légers : conception, modélisation et commande », Thèse de doctorat Montpellier, Avril 1991.

[Quinlan 93] S. Quinlan et O. Khatib, « Elastic bands : connecting path planning and control », IEEE International Conference on Robotics and Automation, 1993.

[Romdhane 99] L. Romdhane, « Design and analysis of a hybrid serial-parallel manipulator », Mechanism and Machine Theory 34 7, pp. 1037–1056.

[Ronga 92] F. Ronga et T. Vust « Stewart platforms without computer? », Conf. Real Analytic and Algebraic Geometry, pages 197-212, Trento, 1992.

[Rosen 60] J.B. Rosen, « The Gradient Projection Method for Nonlinear Programming », Part I, SIAM J on Applied Mathematics, 8, 181-217, 1960.

[Sen 03] S. Sen, B. Drasgupta et A. K. Mallik, « Variational approach for singularity-free path planning of parallel manipulators », Mechanism and Machine Theory 38 (2003) 1165-1183.

[Simeon 00] T. Siméon; J-P. Laumond et C. Nissoux, « Visibility-based probalistic roadmaps for motion planning », Advanced Robotics Journal, Volume 14, No. 6, pp. 477-494, 2000.

[Stewart 65] D. Stewart, « A platform with 6 degrees of freedom ». Proc. of the Institution of mechanical engineers, 180(Part 1, 15):371-386, 1965.

[Stump 06] E. Stump et V. Kumar, « Workspaces of cable-actuated parallel manipulators », ASME Journal of mechanical design, vol.128. pp.159-167, Janvier 2006.

[Su 03] Hai-Jun Su, P. Dietmaier et J. M. McCarthy, « Trajectory Planning for Spatial Platform Linkages », Proc. Of the 11th World Congress in Mechanism and Machine Science, Tianjin, China, Août 18-21, 2003.

[Tsai 96] Tsai L-W., « Kinematics of a three-dof platform with three extensible limbs ». ARK, pages 401-410, Portoroz-Bernadin, 22-26 Juin 1996.

[Udupa 77] S. Udupa, « Collision Detection and Avoidance in Computer Controlled Manipulators », Thèse de doctorat du Department of Electrical Engineering, California Institute of Technology, 1977.

[Wilmarth 99] S. Wilmarth, N. Amato, et P. Stiller, « Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space », Proceedings of IEEE International Conference on Robotics and Automation, pp. 1024–1031, Leuven, Belgium, Mai 1999.

[Yang 03] S. Yang, « Biologically inspired robot behavior engineering », Dans le livre « Biologically inspired neural network approaches to real-time collision free robot motion planning », pp. 143–172, Springer Verlag, ISBN 3790815136, 2003.

[Zeghloul 96] S. Zeghloul et P. Rambeaud, « A fast algorithm for distance calculation between convex objects using the optimization approach », International Journal of Robotica, Vol. 14, pp. 355-363, 1996.

[Zlatanov 98] D. Zlatanov, R. G. Fenton, B. Benhabib, « Identification and classification of the singular configurations of mechanisms », Mechanism and Machine Theory, Vol. 33, N°6, pp.743-760, 1998.

Table des matières

Introduction 5

Chapitre I 9

Etat de l'art 9

Introduction 9

1 Définitions 9

2 Les robots parallèles : types et classification 11

2.1 Les robots plans 12

2.2 Les robots à mouvements spatiaux 14

2.3 Les robots à câbles 20

3 Modèle géométrique 22

3.1 Cas des robots sériels 22

3.2 Cas des robots parallèles 25

4 Modèle cinématique 27

4.1 Cas des robots sériels 28

4.2 Cas des robots parallèles 29

4.3 Cas général 33

5 Singularités 34

5.1 Méthode analytique 34

5.2 Autres Méthodes 36

6 Planification de trajectoires 37

6.1 Classification des types de méthodes 37

6.2 Cas des robots parallèles 40

Conclusion 40

Chapitre II 43

Modélisation et simulation des robots 43

Introduction 43
1 Modélisation 44
1.1 Architectures des robots 44
1.2 Modélisation dans SMAR 50
2 Analyse des équations 64
2.1 Calcul de la mobilité 65
2.2 Etat du système 65
2.3 Elimination des variables passives 67
3 Exemples 68
3.1 Robot 2-RPR 68
3.2 Robot RR-RRR 71
3.3 Robot 3-RPR 75
3.4 Plateforme de Stewart 79
Conclusion 81
Chapitre III 83
Planification de trajectoire des robots parallèles 83
Introduction 83
1 Méthode de planification 84
1.1 Adaptation des méthodes de planification au cas des robots parallèles 84
1.2 Méthode choisie 85
2 Détection de collision 90
2.1 Détection de collision entre deux objets 90
2.2 Détection de collision entre un segment et un objet 96
3 Etude du cas du 3-RPR 101
3.1 Description du robot 101
3.2 Evitement de singularités 103
3.3 Evitement d'obstacles et de singularité 105
Conclusion 110

Chapitre IV 111

Résultats et simulations 111

Introduction 111

1 Le translateur 111

1.1 Modélisation sous SMAR 112

1.2 Définition de la tâche 115

1.3 Résultats 115

2 Le robot à câbles 120

2.1 Modélisation 120

2.2 Résultats de simulation 122

2.3 Résultats expérimentaux 125

3 Le robot chien 127

3.1 Modélisation 128

3.2 Génération des mouvements de marche 130

3.3 Mouvement de rotation 132

3.4 Résultats 132

Conclusion 139

Conclusions et perspectives 141

Annexe A 145

Modélisation d'un robot sériel Lynx sous SMAR 145

Annexe B 149

Equations cinématiques d'une plateforme de Stewart 149

Bibliographie 153

Table des matières 159