HAL
open science

Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et singularités dans un espace de travail en translation

Jean-Pierre Merlet

To cite this version:

Jean-Pierre Merlet. Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et singularités dans un espace de travail en translation. RR-3243, INRIA. 1997. inria-00073446

HAL Id: inria-00073446

https://inria.hal.science/inria-00073446v1

Submitted on 24 May 2006

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

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

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et Singularités dans un espace de travail en translation

Jean-Pierre MERLET

N° 3243

Septembre 1997

THÈME 2

Rapport de recherche

ISSN 0249-6399

Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et Singularités dans un espace de travail en translation

Jean-Pierre MERLET

Thème 2 — Génie logiciel
et calcul symbolique
Projet Prisme

Rapport de recherche n° 3243 — Septembre 1997 — 45 pages

Résumé : Lors de la conception d'un robot parallèle il est nécessaire d'évaluer certaines de ses caractéristiques afin de vérifier qu'elles sont conformes au cahier des charges. Pour certaines d'entre elles cette évaluation doit se faire sur tout l'espace de travail du robot car la valeur de la caractéristique est dépendante de la position et de l'orientation de l'organe terminal. La méthode habituelle consiste à discrétiser l'espace de travail afin d'établir approximativement soit les extremums de la caractéristique soit l'existence d'une condition particulière sur celle-ci. L'espace de travail étant en général de dimension 6, cette méthode se relève lourde en temps de calcul.

Nous présentons ici des algorithmes permettant d'évaluer efficacement certaines de ces caractéristiques pour tout type d'espace de travail en translation, réduisant ici la nécessité de la discrétisation à l'espace de travail en orientation. On établit ainsi les valeurs extrémales:

- des coordonnées articulaires.
- des vitesses articulaires nécessaires à la réalisation d'un couple vitesse angulaire et de translation de l'organe terminal
- des raideurs
- des forces articulaires résultant de l'application d'une charge sur la plate-forme mobile

On montrera aussi que l'on peut déterminer la présence de singularités dans l'espace de travail ainsi que leur localisation.

Mots-clé : modèle géométrique, raideur, cinétique, singularités, statique, robot parallèle, théorie des mécanismes

(Abstract: pto)

Unité de recherche INRIA Sophia Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex (France)
Téléphone : 04 93 65 77 77 – Télécopie : 04 93 65 77 65

Efficient computation of some features of parallel robots: Extrema of the stiffness and of the articular coordinates, forces and velocities and Singularities in a Translation Workspace

Abstract: During the design of a parallel manipulator it is necessary to evaluate some of its features in order to verify their conformity with the requirements. For some of them the evaluation must be performed over the whole workspace of the manipulator as their values are dependent upon the position/orientation of the end-effector. The usual method rely on the discretization of the workspace in order to establish approximatively either the extrema of the feature or that some condition on the feature is fulfilled. The workspace being in general of dimension 6, this method is computer intensive.

We present here some algorithms enabling to compute efficiently some of these features in the translationnal workspace (consequently only the orientation workspace has to be discretized). We establish the extremal values of:

- the articular coordinates
- the articular velocities necessary to perform a given translation/orientastion velocity of the end-effector
- the stiffness
- the articular forces induced by a given load on the platform

We show also that it is possible to establish the presence of singularities in the workspace together with the location of the singularity.

Key-words: kinematics, stiffness, articular velocities, articular forces, singularities, parallel robot, mechanism theory

1 Introduction

La conception optimale d'un robot parallèle nécessite d'évaluer un certain nombre de caractéristiques de ces robots. La grande majorité des caractéristiques intervenant pour définir le concept d'optimalité sont des quantités qui sont dépendantes de la position/orientation de l'organe terminal et l'on doit rechercher si elles satisfont certaines critères lorsque le robot se déplace sur tout son espace de travail. Citons par exemple la raideur d'un robot: un robot optimal pourra être défini comme celui pour lequel le minimum de la raideur obtenue lorsque l'organe terminal décrit un espace de travail donné est maximal.

Un autre critère d'optimalité peut concerner les variables articulaires (les longueurs des segments pour un robot de type plate-forme de Gough, figure 1). L'optimalité d'un robot peut être, par exemple, définie comme la géométrie permettant de parcourir un espace de travail donné avec des actionneurs dont la course est minimale.

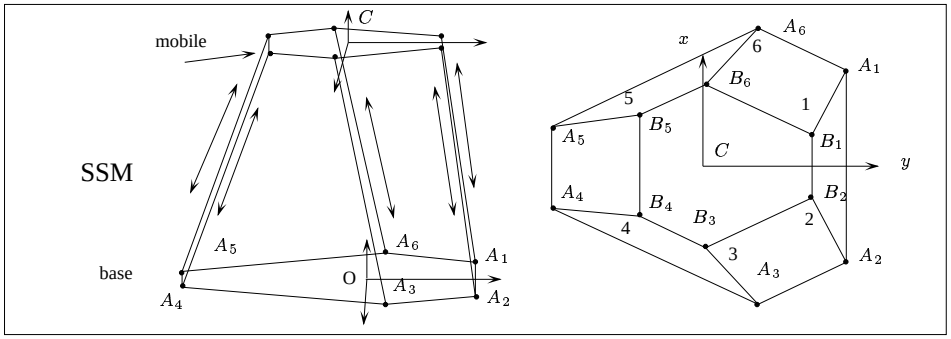

FIG. 1 – *Un robot parallèle à 6 degrés de liberté: la plate-forme de Gough. La position de la plate-forme mobile est définie par la position d'un point particulier, C, et par une matrice de rotation. La plate-forme mobile est reliée à la base par 6 segments possédant au point A_i un joint de Cardan et au point B_i une rotule. Dans chacun des segments un actionneur linéaire permet de modifier la longueur du segment et il est facile de calculer quelle doit être la longueur des 6 segments pour que la plate-forme occupe une position/orientation donnée.*

Les caractéristiques qui interviennent dans ces critères font donc en général intervenir la position/orientation de l'organe terminal. L'optimalité nécessite alors de calculer ces caractéristiques sur tout l'espace de travail. Or, s'il est en général facile d'établir la valeur de la caractéristique pour une configuration de l'organe terminal, son évaluation sur un ensemble de configuration est beaucoup plus difficile. Pour procéder à cette évaluation on recourt généralement à une discrétisation de l'espace de travail mais cette méthode a de nombreux inconvénients:

– même si le temps de calcul de la caractéristique dans une configuration de l'organe terminal peut être faible, l'espace de travail étant de dimension 6 une discrétisation conduit rapide-

ment à un temps de calcul important augmentant très rapidement avec le pas utilisé dans la discrétisation

– on n'a aucune garantie sur l'exactitude des valeurs obtenues (par exemple si la caractéristique subit des variations importantes entre deux configurations de la discrétisation)

Le temps de calcul est un élément important dans la conception optimale. Pour les robots parallèles (et les mécanismes en général) une approche classique de conception optimale consiste à définir une fonction de coût à partir des caractéristiques recherchées, fonction qui dépend de la géométrie du robot, et d'utiliser une procédure numérique d'optimisation pour minimiser la fonction et donc déterminer la géométrie optimale. Cette procédure nécessite de nombreuses évaluations de la fonction, et donc des caractéristiques qui y interviennent. Elle est délicate à mettre en œuvre pour des mécanismes complexes: difficulté de la définition de la fonction de coût, des poids à affecter aux divers critères (qui s'expriment souvent dans des unités différentes), de la sensibilité du résultat à ces poids [8] et problème de minima locaux. De plus les critères de type à seuil sont difficiles à intégrer dans une telle fonction.

Notre approche pour les robots parallèles de type plate-forme de Gough, utilise une autre méthode: la géométrie du robot dépend de paramètres ce qui permet de construire un espace des paramètres dans lequel chaque point correspond à une géométrie de robot unique. Certaines des caractéristiques présentes dans le cahier des charges sont utilisées pour déterminer des régions fermées de l'espace des paramètres qui contiennent l'ensemble des points représentatifs des robots satisfaisant le critère voulu pour ces caractéristiques. L'intersection de ces régions est ensuite discrétisée et en chacun des points de la discrétisation les caractéristiques intervenant dans le critère d'optimalité sont évaluées. Le robot satisfaisant au mieux l'ensemble des critères est considéré comme optimal.

A l'heure actuelle nous disposons de deux algorithmes permettant de calculer des régions fermées de l'espace des paramètres:

– contrainte d'espace de travail: étant donné des longueurs minimum et maximum pour les segments on peut déterminer l'ensemble des géométries de robot dont l'espace de travail inclut un ensemble de segments décrivant des trajectoires de C, l'orientation de la plate-forme étant constante (mais non nécessairement identique) pour tout point d'un segment donné [22].

– contrainte sur les vitesses articulaires: étant donnée une borne sur les vitesses articulaires on sait déterminer l'ensemble des robots dont la plate-forme peut se déplacer avec une vitesse donnée quelle que soit la position de C dans un espace de travail donné, ceci sans que les vitesses articulaires n'excédent à aucun moment la borne [20].

Cette approche présente de nombreux avantages: le robot optimal n'est recherché que dans une zone de l'espace des paramètres de taille réduite et la discrétisation permet de s'affranchir des problèmes de minima locaux sous réserve que la taille de la grille de discrétisation soit suffisamment fine (ce qui implique que l'on doit évaluer l'optimalité de nombreuses géométries). Si cette approche permet une meilleure maîtrise du concept d'optimalité, elle nécessite donc toutefois de nombreuses évaluation des caractéristiques, comme l'approche fonction de coût.

Notre méthodologie de conception est en cours d'intégration dans un outil logiciel, `DEMOCRAT`, qui permet à un utilisateur non expert de déterminer le robot optimal. En dehors de la détermination

des régions fermées de l'espace des paramètres le principal effort de l'utilisateur est de décrire son cahier des charges sous forme procédurale dans un langage dont la syntaxe est proche du langage C. Lors de la discrétisation la procédure est évaluée et retourne vers le superviseur un entier décrivant l'évaluation du robot courant: 0 si le robot ne respecte pas le cahier des charges, 1 s'il le respecte, 2 s'il le respecte et que ses performances sont meilleures que celles des robots déjà examinés. L'interprétation de la procédure repose sur un automate (écrit en Lex-Yacc) et dans le langage des instructions spéciales permettent d'obtenir l'évaluation de certaines performances du robot. Le temps d'évaluation de la procédure est donc étroitement dépendant de l'efficacité des programmes qui permettent le calcul des performances du robot.

Les algorithmes que nous allons décrire dans ce rapport permettent une évaluation efficace et rapide de certaines caractéristiques intervenant dans le domaine des robots parallèles pour tout type d'espace de travail *en translation*, c'est-à-dire pour l'ensemble des positions du point C dans un objet 3D, l'orientation de la plate-forme étant fixée. Il sera donc encore nécessaire de procéder à une discrétisation pour les orientations mais seulement en dimension 3 ce qui réduit considérablement les temps de calcul globaux.

Le type de robot que nous allons considérer est la plate-forme de Gough (figure 1) mais les algorithmes pourront être généralisés à d'autres types d'architectures.

Ce robot est constitué d'une base et d'un plateau mobile reliés par 6 chaînes cinématiques. Aux extrémités de chaque chaîne on trouve un joint de Cardan et une rotule et dans chacune des chaînes un vérin permet de modifier la longueur des segments. La modification des 6 longueurs permet de commander la position/orientation de la plate-forme mobile par rapport à la base. Les *coordonnées articulaires* de ce type de robot sont donc les longueurs des segments. Les *vitesses articulaires* sont les vitesses linéaires des vérins, qui conduisent à une vitesse de translation du centre du plateau mobile et à une vitesse angulaire de celui-ci. C sera défini comme le point sur le plateau mobile pour lequel on spécifie l'espace de travail, ces coordonnée dans le repère de référence étant notées (x, y, z). Les points d'attache des segments à la base sont les centres des rotules et sont notés A_i, alors que les points d'attache des segments au plateau mobile seront notés B_i. Les tests de temps de calcul ont été effectués sur un robot dérivé de la "main gauche" INRIA dont les dimensions sont données en annexe.

Les problèmes de conception de ce type de manipulateur ont été abordés dans les références suivantes: [1, 2, 3, 6, 7, 10, 11, 13, 14, 16, 18, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34]. Mais dans aucune de ces référence les auteurs ne se sont intéressés à l'évaluation rapide des performances des robots parallèles. Nous allons déterminer, lorsque le centre du plateau mobile se déplace dans un espace de travail donné, les caractéristiques suivantes:

- longueurs des segments minimales et maximales
- vitesse articulaires minimales et maximales lorsque le plateau mobile doit se déplacer à une vitesse de translation et à une vitesse angulaire données
- extremums des raideurs
- extremums des forces dans les segments lorsque le plateau mobile est soumis à une charge donnée

– présence de singularité dans l'espace de travail

Finalement nous examinerons le calcul des extremums des erreurs de positionnement de la plate-forme pour des erreurs bornées sur les erreurs de mesure des variables articulaires: nous présenterons un algorithme permettant de calculer ces erreurs avec une erreur garantie mais qui nécessite un temps de calcul important.

2 Extremums des longueurs des segments

Pour une configuration donnée du plateau mobile la longueur d'un segment est la norme du vecteur $\mathbf{AB}$. On a :

$$\mathbf{AB} = \mathbf{OA} + \mathbf{OC} + \mathbf{CB}$$

En introduisant la matrice de rotation R permettant d'exprimer les composantes d'un vecteur exprimé dans le repère fixe à partir des composantes de ce vecteur dans le repère mobile on obtient:

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + R\mathbf{CB_r}$$

où $\mathbf{CB_r}$ est le vecteur définissant la position de B dans le repère mobile. Dans cette équation la définition de la position/orientation du plateau mobile donne $\mathbf{OC}$ et R. La géométrie, connue, du robot est définie par les vecteurs $\mathbf{AO}$ et $\mathbf{CB_r}$. Ainsi les données permettent de calculer les composantes du vecteur $\mathbf{AB}$, donc sa norme qui constitue les longueurs des segments ρ.

2.1 Extremums pour un parallélépipède

Dans cette section on suppose que l'espace de travail est un parallélépipède rectangle. Lorsque le point de référence C du plateau mobile se déplace dans ce parallélépipède les points d'articulation B se déplacent dans un parallélépipède de même taille, obtenu en translatant le parallélépipède décrit par C par le vecteur $\mathbf{CB}$, constant car l'orientation est constante. Pour des raisons physiques on peut supposer que ce parallélépipède ne contient pas le point d'articulation A.

Dans ces conditions les résultats suivants sont immédiats:

– la longueur maximum des segments est obtenue lorsque le point B occupe l'un des sommets du parallélépipède

– la longueur minimum des segments est obtenue lorsque le point B occupe l'un des sommets du parallélépipède sauf si la projection A_p du point A dans le plan d'une face du parallélépipède (étant entendu que pour deux faces parallèles on ne retient que la plus proche du point A) est contenue dans la face, auquel cas la distance minimum est la distance de A à A_p.

Il suffit donc de calculer les longueurs des segments pour les 8 sommets du parallélépipède pour trouver les longueurs maximales. Puis on vérifie si la projection de A est sur l'une des faces: si c'est le cas la longueur minimale est obtenu pour cette configuration, sinon c'est la longueur minimale obtenue lorsque C est en l'un des 8 sommets.

2.2 Extremums pour une sphère

Dans cette section on suppose que l'espace de travail est une sphère. Lorsque le point de référence C du plateau mobile se déplace dans sa sphère les points d'articulation B se déplacent dans une sphère S de même taille, obtenue en translatant la sphère décrite par C par le vecteur $\mathbf{CB}$, constant car l'orientation est constante. Pour des raisons physiques on peut supposer que cette sphère ne contient pas le point d'articulation A.

Soit le segment issu de A et passant par le centre de la sphère S. Il est clair que les distances de A à B extrémales sont obtenues pour les points d'intersection de ce segment avec la sphère S.

2.3 Extremums pour une région

Dans cette section on suppose que l'espace de travail est défini par un ensemble de coupes polygonales horizontales à différentes hauteurs, contenant le même nombre de sommets. Entre deux sections l'espace de travail est le polyèdre obtenu en joignant les sommets de même numéro des deux polygones. Un exemple d'un tel espace de travail est présenté en figure 2.

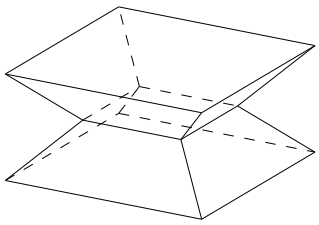

FIG. 2 – *Exemple d'espace de travail défini par 3 coupes horizontales. Cet exemple constituera notre volume test dans les sections suivantes.*

Sans perte de généralité on ne va considérer que l'espace de travail compris entre deux sections (il suffira de répéter le processus pour chaque couple successif de coupes) et l'on cherche à déterminer les extremums avec une précision ϵ fixée. Cet espace va être décomposé en autant de boîtes que nécessaire pour déterminer les longueurs extrémales. Une liste de boîte, initialisée avec la boîte englobante de l'espace de travail B_0, est maintenue à jour dans l'algorithme. De plus un tableau est créé pour sauvegarder les longueurs extrémales trouvées lors des itérations de l'algorithme.

A l'itération k l'algorithme effectue les étapes suivantes :

1. si la boîte B_k est complètement en dehors de l'espace de travail on passe à la boîte suivante dans la liste.

2. si la boîte B_k est complètement à l'intérieur de l'espace de travail on calcule les longueurs extrémales pour cette boîte. Le tableau des longueurs extrémales est mis à jour.

3. si la boîte B_k est partiellement à l'intérieur de l'espace de travail on calcule les longueurs extrémales pour cette boîte.
   (a) si toutes les longueurs sont incluses dans les limites définies dans le tableau des longueurs extrémales on passe à la boîte suivante.
   (b) sinon si pour chaque segment les écarts entre les longueurs extrémales pour la boîte B_k n'excédent pas ϵ le tableau des longueurs extrémales est mis à jour avec la valeur moyenne des longueurs extrémales et l'on passe à la boîte suivante.
   (c) sinon on crée 8 nouvelles boîtes à partir de B_k en divisant chacune de ses dimensions par 2. Ces boîtes sont placées à la fin de la liste et l'on passe à la boîte suivante dans la liste.

L'algorithme stoppe lorsque la liste est vide. Cet algorithme permet de déterminer de manière très efficace les extremums des longueurs des segments, avec une précision arbitraire. Il permet de gérer n'importe quel type de définition de l'espace de travail si l'on dispose d'un test permettant de statuer sur l'appartenance d'une boîte à l'espace de travail.

2.4 Temps de calcul

Sur une station de travail SUN Ultra1 les temps de calcul s'établissent de la manière suivante : boîte (0,26 ms), sphère (0.691 ms), région avec une précision sur les longueurs extrémales d'au moins 0.02% (767 ms).

3 Extremums des vitesses articulaires

Dans cette section on désire déterminer quelles sont les valeurs minimales et maximales des vitesses articulaires nécessaires à la réalisation de vitesse de translation et angulaire du plateau mobile, lorsque C évolue dans un espace de travail donné[1]. La relation entre les vitesses articulaires et les vitesses cartésiennes a été établi par de nombreux chercheurs [9, 15, 17, 31, 35]. On a:

$$\dot{\rho} = J^{-1}\mathcal{V}$$

où $\dot{\rho}$ représente le vecteur des 6 vitesses articulaires, $\mathcal{V}$ le vecteur constitué des trois vitesses en translation de C et des trois composantes de la vitesse angulaire. J^{-1} est la matrice jacobienne inverse du manipulateur, sur laquelle nous reviendrons dans la section consacrée aux singularités. Cette matrice est dépendante de la configuration X de l'organe terminal puisque qu'une ligne de cette matrice s'écrit:

$$\frac{\mathbf{AB}}{||\mathbf{AB}||} \qquad \frac{\mathbf{CB} \wedge \mathbf{AB}}{||\mathbf{AB}||}$$

1. Une implantation de ce programme est disponible par `ftp` anonyme sur `zenon.inria.fr`, répertoire `prisme/Speed`

3.1 Extremums pour un segment

Supposons que le point C se déplace sur un segment défini par ses deux extrémités M_1, M_2. Toute position de C peut alors être définie par:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1M_2} \tag{1}$$

où λ est un scalaire dans l'intervalle [0,1]. Une ligne de la matrice jacobienne inverse s'écrit:

$$\frac{\mathbf{A_iB_i}}{||\mathbf{A_iB_i}||} \ , \ \mathbf{CB_i} \wedge \frac{\mathbf{A_iB_i}}{||\mathbf{A_iB_i}||}$$

On remarque que pour un segment donné la vitesse articulaire ne dépend pas des vitesses articulaires des autres segments: on peut donc se passer de l'indice du segment. Si l'on décompose le vecteur $\mathcal{V}$ en deux vecteurs, $\mathbf{V}$ le vecteur vitesse de translation de C et Ω le vecteur des vitesses angulaires on a:

$$\dot{\rho} = \frac{\mathbf{AB}.\mathbf{V} + (\mathbf{CB} \wedge \mathbf{AB}).\mathbf{\Omega}}{||\mathbf{AB}||} \tag{2}$$

Le vecteur $\mathbf{AB}$ peut s'écrire:

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + \mathbf{CB} = \mathbf{AO} + \mathbf{OM_1} + \mathbf{CB} + \lambda \mathbf{M_1M_2}$$

Dans cette équation les trois premiers vecteurs sont constants. On peut donc écrire $\mathbf{AB}$ sous une forme simplifiée:

$$\mathbf{AB} = \mathbf{U} + \lambda \mathbf{M_1M_2}$$

La norme de ce vecteur s'écrit donc:

$$||\mathbf{AB}|| = \sqrt{\lambda^2 ||\mathbf{M_1M_2}||^2 + 2\lambda \mathbf{U}.\mathbf{M_1M_2} + ||\mathbf{U}||^2}$$

De la même manière on a:

$$\mathbf{CB} \wedge \mathbf{AB} = \mathbf{CB} \wedge \mathbf{U} + \lambda \mathbf{CB} \wedge \mathbf{M_1M_2}$$

En utilisant les équations précédentes l'équation (2) peut s'exprimer de manière simple par:

$$\dot{\rho} = \frac{a_1\lambda + a_2}{\sqrt{a_3\lambda^2 + a_4\lambda + a_5}} \tag{3}$$

avec:

$$\begin{aligned}
a_1 &= \mathbf{M_1M_2}.\mathbf{V} + (\mathbf{CB} \wedge \mathbf{M_1M_2}).\mathbf{\Omega} \\
a_2 &= \mathbf{U}.\mathbf{V} + (\mathbf{CB} \wedge \mathbf{U}).\mathbf{\Omega} \\
a_3 &= ||\mathbf{M_1M_2}||^2 \\
a_4 &= 2\,\mathbf{U}.\mathbf{M_1M_2} \\
a_5 &= ||\mathbf{U}||^2
\end{aligned}$$

Les coefficients a_i dépendent simplement de la géométrie du robot ainsi que des coordonnées des extrémités du segment trajectoire. En dérivant cette expression par rapport à λ on obtient une expression dont le numérateur N est linéaire en λ.

$$N = \lambda(\frac{a_1 a_4}{2} - a_2 a_3) + a_1 a_5 - \frac{a_2 a_4}{2} \tag{4}$$

En conséquence les extremums de la vitesse articulaire seront obtenu soit pour $\lambda = 0, 1$ où pour la valeur de λ qui annule N, à condition que cette valeur soit comprise dans l'intervalle [0,1]. Il est donc aisé de calculer ces extremums.

3.2 Extremums pour un rectangle horizontal

Supposons que l'espace de travail soit défini comme un rectangle horizontal tel que les coordonnées de C (x, y, z) vérifient:

$$x_1 \leq x \leq x_2 \quad y_1 \leq y \leq y_2$$

On cherche donc les extremums des vitesses articulaires sous ces contraintes, ce qui peut se résoudre à l'aide des techniques classiques d'optimisation. On définit deux nouvelles variables α, β telles que :

$$x = x_1 + \frac{(1 + \sin\alpha)(x_2 - x_1)}{2}$$
$$y = y_1 + \frac{(1 + \sin\beta)(y_2 - y_1)}{2}$$

Les extremums de $\dot{\rho}$ satisfont alors les conditions suivantes:

$$\frac{\partial\dot{\rho}}{\partial\alpha} = 0 \tag{5}$$
$$\frac{\partial\dot{\rho}}{\partial\beta} = 0 \tag{6}$$

Les équations (5),(6) s'écrivent:

$$\cos(\alpha)\ F_1(\alpha, \beta) = 0 \qquad \cos(\beta)\ F_2(\alpha, \beta) = 0 \tag{7}$$

En conséquence les extremums des vitesses articulaires sont obtenues soit sur les côtés du rectangle soit aux points où $F_1 = F_2 = 0$. Avec l'algorithme décrit dans la section précédente on peut calculer les vitesses extrémales sur les arêtes constituant les côtés du rectangle. Il reste ensuite à déterminer s'il existe une solution au problème $F_1 = F_2 = 0$ et à comparer les résultats avec les vitesses articulaires obtenues pour les arêtes. Dans les équations $F_1 = F_2 = 0$ les inconnues α, β apparaissent via leur sinus uniquement. F_1 est linéaire en $\sin\alpha$. Cette quantité est calculée et le résultat est reporté dans F_2. Cette équation devient alors un polynôme de degré 3 en $\sin\beta$. Ce polynôme est résolu et pour chacune des solutions réelles dans l'intervalle [-1,1] on détermine la valeur correspondante de $\sin\alpha$. On obtient ainsi une liste de positions de C susceptible de conduire à un extremum pour les vitesses articulaires. A cette liste on ajoute les positions de C obtenues pour les arêtes du rectangle et les extremums des vitesses articulaires seront obtenus en comparant les vitesses articulaires calculées pour chaque posture de la liste et en retenant les valeurs extrêmes.

3.3 Extremums dans un parallélépipède rectangle

Supposons maintenant que l'espace de travail est une parallélépipède rectangle. Les points de l'espace de travail vérifient:

$$x_1 \leq x \leq x_2 \quad y_1 \leq y \leq y_2 \quad z_1 \leq z \leq z_2 \tag{8}$$

et l'on veut calculer les extremums des vitesses articulaires sous ces contraintes.

3.3.1 L'approche par optimisation

On définit 3 nouvelles variables α, β, γ tel que:

$$x = x_1 + \frac{(1+\sin\alpha)(x_2 - x_1)}{2} \tag{9}$$

$$y = y_1 + \frac{(1+\sin\beta)(y_2 - y_1)}{2} \tag{10}$$

$$z = z_1 + \frac{(1+\sin\gamma)(z_2 - z_1)}{2} \tag{11}$$

$\dot{\rho}$ devient une fonction de α, β, γ dont les valeurs extrémales vérifient les équations suivantes:

$$\frac{\partial\dot{\rho}}{\partial\alpha} = 0 \tag{12}$$

$$\frac{\partial\dot{\rho}}{\partial\beta} = 0 \tag{13}$$

$$\frac{\partial\dot{\rho}}{\partial\gamma} = 0 \tag{14}$$

Ces trois équations sont des fonctions des sinus et cosinus des trois inconnues. La première de ces équations est linéaire dans le terme $\sin\alpha$ et aucune des trois équations ne contient de terme en $\cos\alpha$. On résout donc la première équation en $\sin\alpha$ et l'on reporte le résultat dans les deux autres équations. Celles-ci ne contiennent alors plus que des termes en sinus des inconnues β, γ. On obtient ainsi deux équations:

$$\sum_{i,j=0,4,i+j\leq 4} a_{ij} T_1^i T_2^j = 0 \qquad \sum_{i,j=0,4,i+j\leq 4} b_{ij} T_1^i T_2^j = 0$$

où $T_1 = \sin(\beta), T_2 = \sin(\mu)$. Il n'a cependant pas été possible de réduire ces deux équations à un polynôme monovariable. On peut alors penser à utiliser une méthode numérique comme l'homotopie ou l'arithmétique d'intervalles pour résoudre ce système d'équations. Nous avons cependant choisi une autre approche.

3.3.2 L'approche approximative

3.3.2.1 Le principe Dans cette approche on va déterminer les extremums des vitesses articulaires avec une erreur maximum $\epsilon > 0$ garantie. L'idée est de calculer les vitesses extrémales pour

des rectangles horizontaux à différentes hauteurs z en s'assurant que la différence entre les valeurs extrémales des vitesses articulaires entre deux rectangles successifs n'excède pas ϵ. Ainsi on part du rectangle horizontal correspondant à l'altitude minimale de la boîte puis on calcule la nouvelle altitude du rectangle suivant de façon à ce que la contrainte précédente soit vérifiée. On répète ensuite le calcul pour le nouveau rectangle jusqu'à ce que l'altitude du rectangle soit au moins égale à l'altitude maximale de la boîte. A ce point on aura calculé les vitesses articulaires extrémales avec une précision au moins égale à ϵ.

3.3.2.2 Détermination des altitudes A l'itération k de l'algorithme les vitesses articulaires maximales sont $\dot{\rho}_k$ et l'altitude du rectangle courant est z_k. On désire calculer un accroissement d'altitude dz^2 telle que:

$$\dot{\rho}(z_k + dz^2) = \dot{\rho}_k + \epsilon \tag{15}$$

En fait cette équation est délicate à utiliser car elle fait apparaître des racines carrés de part la nature de la jacobienne inverse. Rappelons que l'on a:

$$\dot{\rho} = \frac{F(z, \mathcal{V})}{\rho(z)}$$

où F et ρ^2 sont des fonctions algébriques de z. L'équation (15) peut être alors transformée en:

$$F^2(z_k + dz^2) - (\dot{\rho}_k + \epsilon)^2 \rho^2(z_k + dz^2) = 0 \tag{16}$$

qui est alors algébrique. On peut alors formuler le problème d'optimisation qui est de trouver le minimum de dz^2 tel que (16) est satisfaite sous les contraintes sur x, y définies par (8). On introduit la fonction d'optimisation H:

$$H = dz^2 + \mu(F^2(z_k + dz^2) - (\dot{\rho}_k + \epsilon)^2 \rho^2(z_k + dz^2))$$

dans laquelle x, y ont été remplacés par (9,10). Le minimum de dz sera solution du système d'équations:

$$\frac{\partial H}{\partial dz} = 0 \quad \frac{\partial H}{\partial \mu} = 0$$
$$\frac{\partial H}{\partial \alpha} = 0 \quad \frac{\partial H}{\partial \beta} = 0$$

Les équations précédentes conduisent à de nombreux cas particuliers, qui ne seront pas détaillés ici, mais qui se ramènent tous à la résolution d'équations de degré 2. Il sera cependant nécessaire de vérifier que les solutions en dz ne conduise pas à $\dot{\rho} = -\dot{\rho}_k - \epsilon$ qui est aussi solution de l'équation (15).

Pour ce qui concerne la détermination des vitesses minimales la méthode est similaire en prenant $\epsilon < 0$.

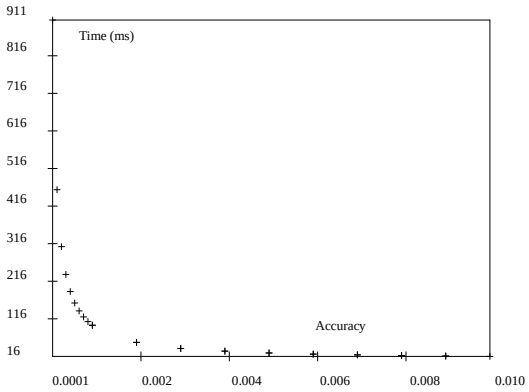

FIG. 3 – *Temps de calcul des valeurs extrémales des vitesses articulaires pour un espace de travail de type parallélépipède rectangle en fonction de la précision demandée sur les vitesses.*

3.3.2.3 Temps de calcul Le temps de calcul de l'algorithme précédent est clairement dépendant de la précision ϵ avec laquelle les valeurs extrémales des vitesses articulaires doivent être déterminées. La figure 3 illustre sur un exemple le temps de calcul en fonction de la précision, obtenu sur une station de travail SUN Ultra 1. On peut constater que même si l'on désire une très bonne précision sur les vitesses extrémales le temps de calcul reste raisonnable.

Les avantages de la méthode proposée par rapport à une méthode de discrétisation sont clairs:

- l'erreur avec laquelle les vitesses extrémales sont obtenues est garantie, ce qui serait difficile à obtenir avec une méthode de discrétisation.
- le temps de calcul des vitesses pour une configuration donnée de l'organe terminal est de l'ordre de 0.154 ms. L'examen de la figure 3 permet d'observer qu'avec une précision de 0.01 le temps de calcul de l'algorithme est équivalent à celui que l'on obtiendrait en calculant les vitesses en 105 points de la boîte, ce qui correspond à moins de 5 points de discrétisation sur chaque axe. Une aussi mauvaise discrétisation ne permettrait pas d'atteindre une bonne précision.

3.4 Vitesses extrémales pour tout type d'espace de travail

Le faible temps de calcul de l'algorithme décrit pour les boîtes permet d'étendre le calcul des vitesses extrémales à des espaces de travail plus complexes.

On suppose ici que l'espace de travail peut être décrit à l'aide de polygones horizontaux disposés successivement comme dans la section 2.3.

On considère l'espace de travail compris entre deux polygones successifs, qui va être décomposé en autant de boîtes que nécessaire pour déterminer les vitesses articulaires extrémales avec la précision ϵ souhaitée. Une liste de boîtes, initialisée avec la boîte englobante de l'espace de travail

B_0, est maintenue à jour dans l'algorithme. De plus un tableau est crée pour sauvegarder les vitesses extrémales.

A l'itération k l'algorithme effectue les étapes suivantes :

1. si la boîte B_k est complètement en dehors de l'espace de travail on passe à la boîte suivante dans la liste.
2. si la boîte B_k est complètement à l'intérieur de l'espace de travail on calcule les vitesses articulaires extrémales pour cette boîte. Le tableau des vitesses extrémales est mis à jour.
3. si la boîte B_k est partiellement à l'intérieur de l'espace de travail on calcule les vitesses articulaires extrémales pour cette boîte.
   (a) si toutes les vitesses sont incluses dans les limites définies dans le tableau des vitesses extrémales on passe à la boîte suivante.
   (b) sinon on crée 8 nouvelles boîte à partir de B_k en divisant chacune de ses dimensions par 2. Ces boîtes sont placées à la fin de la liste et l'on passe à la boîte suivante.

L'algorithme stoppe lorsque la liste est vide. Il permet de calculer les vitesses extrémales pour des espaces de travail complexes dans un temps raisonnable. Pour améliorer cette vitesse supposons que nous disposions d'un minorant et d'un majorant des extremums des vitesses articulaires pour chacune des boîtes apparaissant au cours de l'algorithme: si ces valeurs sont dans les limites actuelles des extremums on peut passer directement à la boîte suivante dans la liste. Pour le calcul du minorant et du majorant notons que chacune des vitesses articulaires s'exprime comme le quotient d'une fonction algébrique dans les inconnues x, y, z par une longueur de segment. On peut donc trouver facilement un minorant et un majorant du numérateur alors que l'algorithme décrit dans la section 2 permet de trouver les bornes exactes du dénominateur, ce qui permet de calculer un minorant et un majorant du quotient. Pour diminuer encore le temps de calcul il est aussi prudent de procéder à une initialisation des extremums des vitesses articulaires, ce qui peut être fait, par exemple, en calculant les extremums sur les arêtes des polygones définissant l'espace de travail.

Par exemple on a considéré l'espace de travail défini par les trois coupes suivantes: $z = 50, x \in [-10, 10], y \in [-10, 10]$, $z = 55, x \in [-5, 5], y \in [-5, 5]$, $z = 60, x \in [-10, 10], y \in [-10, 10]$ représenté en figure 2.

L'orientation est définie par les trois angles d'Euler égales à -60 degrés, la vitesse de translation de l'organe terminal est (-10,0,-10) et la vitesse angulaire (-10,0,10). La table 4 donne les temps de calcul en fonction de la précision demandée. On peut remarquer que le temps de calcul nécessaire

ϵ	1	0.5	0.1	0.05	0.01
Temps (ms)	330	380	3660	10350	73180

TAB. 1 – *Temps de calcul des vitesses extrémales pour le volume test, en fonction de la précision ϵ demandée*

pour déterminer les vitesses extrémales avec une précision égale à au moins 1% est d'environ 4s.

3.5 Vitesses extrémales sur l'espace complet

L'espace de travail peut être aussi défini comme un hypercube dans l'espace articulaire. L'algorithme est dans le principe identique, seul diffère le test sur l'appartenance des boîtes à l'espace de travail qui utilise l'algorithme décrit dans la section 2 permettant de calculer les extremums des longueurs des segments pour une boîte. Pour cela on part d'une boîte cartésienne pour C dont on est sûr qu'elle contient l'ensemble des positions de C satisfaisant les contraintes sur les variables articulaires. La détermination d'une telle boîte peut se faire d'une manière simple en considérant que le point C ne peut s'écarter d'un point A_i qu'au maximum d'une longueur $\rho^i_{max} + ||\mathbf{CB_i}|| = l$: pour une direction donnée $\mathbf{n}$ les coordonnées extrémales de C selon la direction sont $Max(A^n_i - l), Min(A^n_i + l)$ lorsque i varie de 1 à 6, où A^n_i représente les coordonnées selon la direction $\mathbf{n}$ du point A_i. Il est donc facile de déterminer un parallélépipède rectangle englobant l'espace de travail, qui est utilisé comme boîte initiale de l'algorithme précédent. Pour statuer sur l'appartenance d'une boîte vis-à-vis de l'hypercube articulaire il suffit de calculer les longueurs articulaires extrémales pour cette boîte en utilisant l'algorithme de la section 2: si les longueurs extrémales des 6 segments sont comprises dans l'hypercube articulaire alors la boîte est totalement interne, si les longueurs extrémales d'au moins un segment sont en dehors des limites de l'hypercube alors la boîte est externe, sinon elle est considéré comme partiellement interne (ce qui peut être faux: si individuellement chacune des longueurs des segment peut avoir une intersection avec les limites articulaires il est possible cependant qu'il n'existe pas de posture où cela soit vrai pour l'ensemble des segments, mais cela sera détecté au moment du découpage de la boîte).

Pour diminuer le temps de calcul nous utilisons le même type d'heuristique que dans la section précédente, l'initialisation des extremums des vitesses articulaires se faisant en prenant des points au hasard dans la boîte englobante de l'espace de travail et en calculant les vitesses articulaires en ces points s'ils appartiennent effectivement à l'espace de travail.

Le nombre de boîte que l'on crée avec cette méthode est important. La figure 4 montre l'ensemble des boîtes crées par l'algorithme lorsque la précision est 0.5.

Le temps de calcul de l'algorithme avec une précision de 0.1 (0.01) est de 28000 ms (62000 ms). Dans le cas particulier où la vitesse angulaire est nulle (voir section suivante) ce temps se réduit à 1270 ms (1380 ms).

3.6 Cas particulier

On va supposer dans cette section que la vitesse angulaire est nulle. En conséquence les vitesses articulaires s'écrivent:

$$\dot{\rho} = \mathbf{V}.\frac{\mathbf{AB}}{||\mathbf{AB}||} \tag{17}$$

Soit D_1 la ligne passant par A et de vecteur directeur $\mathbf{V}$ et D_2 la ligne associée au segment du robot. Si μ est l'angle entre ces deux lignes on a:

$$\dot{\rho} = ||\mathbf{V}|| \cos \mu \tag{18}$$

Si D_1 coupe le volume décrit par B en un point M alors la vitesse articulaire maximale sera $||\mathbf{V}||$ si $\mathbf{AM}.\mathbf{V}$ est positif ou la vitesse articulaire minimale sera $-||\mathbf{V}||$ si $\mathbf{AM}.\mathbf{V}$ est négatif. Dans la suite

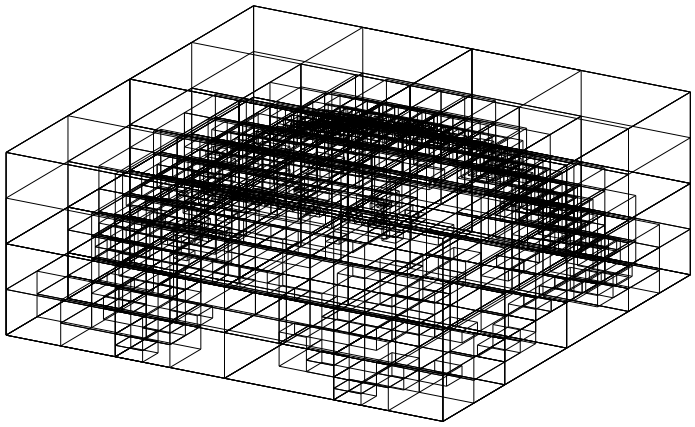

FIG. 4 – *Ensemble des boîtes crées lors du calcul des extremums des vitesses articulaires dans l'ensemble de l'espace de travail (précision=0.5)*

on va supposer que $\mathbf{AM}.\mathbf{V}$ est positif pour rechercher la vitesse maximale (dans le cas contraire on obtiendra la vitesse minimale).

Si D_1 ne coupe pas l'espace de travail la vitesse maximale sera obtenue pour une position de C sur une des arêtes des polygones définissant l'espace de travail. Il en sera de même pour la vitesse minimale. Dans ce cas l'algorithme permettant de calculer les vitesses extrémales sur un segment permet de déterminer les vitesses extrémales sur l'ensemble de l'espace de travail. Dans le cas où D_1 coupe l'espace de travail la vitesse maximale est simplement $||\mathbf{V}||$. On doit donc tester l'intersection de D_1 avec l'espace de travail. Cette ligne peut couper l'espace de travail soit en une de deux facettes horizontales (définies par les polygones des coupes), soit sur une facette obtenue en joignant deux arêtes des coupes.

Dans le premier cas on calcule l'intersection de D_1 avec le plan horizontal d'altitude la hauteur de la coupe. Si ce point est à l'intérieur du polygone, alors la droite coupe l'espace de travail et la vitesse maximale est $||\mathbf{V}||$.

Dans le second cas la facette est obtenue en joignant deux arêtes. Soit $(x_1, y_1, z_1), (x_2, y_2, z_1)$ les coordonnées des extrémités du premier segment et $(x_3, y_3, z_3), (x_4, y_4, z_3)$ les coordonnées des extrémités du second segment.

L'intersection de la facette avec le plan horizontal à l'altitude z est un segment défini par deux points X_k, X_{k+1} dont les coordonnées sont:

$$x_k = x_1 + (x_3 - x_1)(z - z_1)(z_3 - z_1) \quad (19)$$

$$y_k = y_1 + (y_3 - y_1)(z - z_1)(z_3 - z_1) \quad (20)$$

$$z_k = z \quad (21)$$

$$x_{k+1} = x_2 + (x_4 - x_2)(z - z_1)(z_3 - z_1) \quad (22)$$

$$y_{k+1} = y_2 + (y_4 - y_2)(z - z_1)(z_3 - z_1) \quad (23)$$
$$z_{k+1} = z \quad (24)$$

Un point $M(x, y, z)$ appartenant à ce segment a comme coordonnées:

$$x = x_k + \lambda(x_{k+1} - x_k) \quad (25)$$
$$y = y_k + \lambda(y_{k+1} - y_k) \quad (26)$$

où λ appartient à l'intervalle [0,1].Si M appartient à D_1 on a:

$$\mathbf{AM} \wedge \mathbf{V} = \mathbf{0} \quad (27)$$

Cette équations conduit à deux équations dans les inconnues z, λ. On peut alors distinguer deux cas:

1. si la composante verticale de la vitesse $\mathbf{V}$ est nulle alors z est la coordonnée en z de A et l'équation (27) est linéaire en λ. Après résolution si λ est dans l'intervalle [0,1] alors on a intersection entre D_1 et l'espace de travail.

2. si la composante verticale de la vitesse $\mathbf{V}$ est non nulle alors le résultant des deux équations (27) est une équation du deuxième degré en z. Cette équation est résolue et si une solution est dans l'intervalle $[z_1, z_2]$ la valeur de λ est calculée. Si cette valeur est dans l'intervalle [0,1] on a alors intersection entre D_1 et l'espace de travail

Cet algorithme est très rapide (le temps de calcul est de moins de 10 ms pour le volume test) et permet la détermination exacte des vitesses articulaires extrémales.

Remarquons que cet algorithme peut être étendu au cas où l'espace de travail est une sphère. On vérifie tout d'abord si D_1 coupe la sphère, auquel cas la vitesse maximale est $||\mathbf{V}||$. Sinon soit le plan défini par D_1 et le centre T de la sphère décrite par B lorsque C décrit l'espace de travail. Ce plan coupe la sphère décrite par B selon un cercle. Soit les deux points sur ce cercle tels que la tangente au cercle en ces points passent par A. Les vitesses articulaires extrémales sont obtenues en ces points puisqu'il optimise la valeur de $\cos \mu$.

3.7 Applications: détermination des angulations des segments

Nous avons vu dans la section précédente que si la vitesse angulaire est égale à 0 le calcul des vitesses extrémales revient à déterminer la valeur maximum de la quantité $\mathbf{AB}.\mathbf{V}/||\mathbf{AB}||$. Cette quantité est en fait le cosinus de l'angle entre le segment et le vecteur $\mathbf{V}$. On obtient ainsi les valeurs extrêmes de l'angle entre le segment et une direction fixe, donc l'angulation que doit pouvoir exécuter l'articulation passive du segment pour permettre au point C de décrire son espace de travail.

4 Extremums des raideurs

Dans cette section on suppose que chaque segment i est modélisable par un ressort de raideur k_i. La matrice de raideur K du robot est telle que :

$$\Delta\mathbf{F} = K\Delta\mathbf{X}$$

où $\mathbf{F}$ est le vecteur des forces généralisées (forces et moments par rapport à C) agissant sur la plate-forme et $\mathbf{X}$ les déplacements généralisés de la plate-forme.

Si l'on construit la matrice diagonale K_a de dimension 6×6 dont les éléments de la diagonale sont égales à k_i on sait [19] que la matrice de raideur K s'écrit:

$$K = J^{-T} K_a J^{-1} \tag{28}$$

On s'intéresse ici à la détermination des valeurs extrémales des raideurs principales, c'est-à-dire des éléments diagonaux de la matrice K. Pour une configuration donnée de l'organe terminal et pour un segment i notons $(x_{A_i B_i}, y_{A_i B_i}, z_{A_i B_i})$ les composantes du vecteur $\mathbf{A_i B_i}$ et (x_b^i, y_b^i, z_b^i) les composantes du vecteur du vecteur $\mathbf{CB_i}$. On appelle $k_x, k_y, k_z, k_{tx}, k_{ty}, k_{tz}$ les éléments diagonaux de K. D'après l'équation (28) on a alors:

$$k_x = \sum_{i=1}^{i=6} k_i \frac{x_{A_i B_i}^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{29}$$

$$k_y = \sum_{i=1}^{i=6} k_i \frac{z_{A_i B_i}^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{30}$$

$$k_z = \sum_{i=1}^{i=6} k_i \frac{z_{A_i B_i}^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{31}$$

$$k_{tx} = \sum_{i=1}^{i=6} k_i \frac{(y_{A_i B_i} z_b^i - z_{A_i B_i} y_b^i)^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{32}$$

$$k_{ty} = \sum_{i=1}^{i=6} k_i \frac{(z_{A_i B_i} x_b^i - x_{A_i B_i} z_b^i)^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{33}$$

$$k_{tz} = \sum_{i=1}^{i=6} k_i \frac{(x_{A_i B_i} y_b^i - y_{A_i B_i} z_b^i)^2}{x_{A_i B_i}^2 + y_{A_i B_i}^2 + z_{A_i B_i}^2} \tag{34}$$

$$\tag{35}$$

4.1 Extremums pour un segment

Supposons que le point C se déplace sur un segment défini par ses deux extrémités M_1, M_2. Toute position de C peut alors être définie par:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1 M_2} \tag{36}$$

où λ est un scalaire dans l'intervalle [0,1]. Le vecteur $\mathbf{AB}$ s'écrit:

$$\mathbf{AB} = \mathbf{AO} + \mathbf{CB} + \mathbf{OC}$$

et s'exprime donc en fonction de λ. Toutes les raideurs exprimés dans les équations (29-34) s'écrivent sous la forme générique suivante:

$$k_{x,y,z,tz,ty,tz} = \sum_{i=1}^{i=6} k_i \frac{a_i\lambda^2 + b_i\lambda + c_i}{A_i\lambda^2 + B_i\lambda + C_i} \tag{37}$$

où les coefficients a, b, c, A, B, C dépendent simplement de la raideur choisie, du segment trajectoire et des constantes géométriques du segment i.

La dérivation de l'équation (37) par rapport à λ conduit à un numérateur de degré 22 en λ (et de degré 21 si $a_i = b_i = 0$). Les raideurs extrêmes seront donc obtenues soit pour $\lambda = 0, 1$ où pour une racine réelle de ce polynôme, comprise dans l'intervalle [0,1].

4.2 Extremums pour des rectangles horizontaux

Supposons que l'espace de travail soit défini comme un rectangle horizontal avec les coordonnées de C (x, y, z) telles que:

$$x_1 \leq x \leq x_2 \quad y_1 \leq y \leq y_2$$

On cherche donc les extremums des vitesses articulaires sous ces contraintes, ce qui peut se résoudre à l'aide des techniques classiques d'optimisation. On définit deux nouvelles variables α, β tel que :

$$x = x_1 + \frac{(1 + \sin\alpha)(x_2 - x_1)}{2}$$
$$y = y_1 + \frac{(1 + \sin\beta)(y_2 - y_1)}{2}$$

On peut alors essayer de poser le problème de la découverte des extremums des raideurs en terme d'optimisation en fonction des inconnues α, β. Soit k un des éléments de la diagonale de K. La dérivation de k par rapport aux inconnues donne deux équations qui s'écrivent:

$$\frac{\partial k}{\partial \alpha} = \cos\alpha F_1(\alpha, \beta) = 0$$
$$\frac{\partial k}{\partial \beta} = \cos\beta F_2(\alpha, \beta) = 0$$

Les extremums seront donc obtenus soit sur une arête du rectangle (ce que l'on peut calculer avec l'algorithme de la section précédente) soit pour un point à l'intérieur du rectangle correspondant au cas $F_1 = F_2 = 0$. Malheureusement si l'on conserve la forme des équations (29-34) on obtient des expressions complexes pour F_1, F_2 qu'il ne nous a pas été possible d'exploiter. On a donc choisi de simplifier le problème en supposant que la manipulateur est élancé, c'est-à-dire que la hauteur de tous les points de l'organe terminal est largement supérieure aux coordonnées en x, y du point. On aura donc:

$$z_{A_iB_i} \gg x_{A_iB_i} \quad z_{A_iB_i} \gg y_{A_iB_i}$$

Sous les hypothèses précédentes on obtient:

$$k_x = \sum_{i=1}^{i=6} k_i(\frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{x^4_{A_iB_i}}{z^4_{A_iB_i}} - \frac{x^2_{A_iB_i}y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

F_2 est alors linéaire en $\sin\beta$ et après résolution F_1 devient un polynôme de degré 7 en $\sin\alpha$.

k_y s'écrit:

$$k_y = \sum_{i=1}^{i=6} k_i(\frac{y^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^4_{A_iB_i}}{z^4_{A_iB_i}} - \frac{x^2_{A_iB_i}y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

F_1 est alors linéaire en $\sin\alpha$ et après résolution F_2 devient un polynôme de degré 7 en $\sin\beta$.

k_z s'écrit:

$$k_z = \sum_{i=1}^{i=6} k_i(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

F_1 est linéaire en $\sin\alpha$ et ne contient pas l'inconnue β et F_2 est linéaire en $\sin\beta$ et ne contient pas l'inconnue α. On résout donc directement deux équations linéaires.

Pour la raideur k_{tx} on obtient:

$$k_{tx} = \sum_{i=1}^{i=6} k_i \frac{(y^i_b\ z_{A_iB_i} - y_{A_iB_i}\ z^i_b)^2}{z^2_{A_iB_i}}(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

que l'on peut réécrire sous la forme:

$$k_{tx} = \sum_{i=1}^{i=6} k_i(y^i_b)^2(1 - 2\frac{y_{A_iB_i}z^i_b}{y^i_b z_{A_iB_i}})(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

F_1 est alors linéaire en $\sin\alpha$ et après résolution F_2 devient un polynôme de degré 7 en $\sin\beta$.

Pour la raideur k_{ty} on obtient:

$$k_{ty} = \sum_{i=1}^{i=6} k_i \frac{(z^i_b\ x_{A_iB_i} - z_{A_iB_i}\ x^i_b)^2}{z^2_{A_iB_i}}(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

que l'on peut réécrire sous la forme:

$$k_{ty} = \sum_{i=1}^{i=6} k_i(y^i_b)^2(1 - 2\frac{x_{A_iB_i}z^i_b}{x^i_b\ z_{A_iB_i}})(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

F_2 est alors linéaire en $\sin\beta$ et après résolution F_1 devient un polynôme de degré 7 en $\sin\alpha$.

Enfin pour la raideur k_{tx} on obtient:

$$k_{tz} = \sum_{i=1}^{i=6} k_i \frac{(x^i_b\ y_{A_iB_i} - x_{A_iB_i}\ y^i_b)^2}{z^2_{A_iB_i}}(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

Le résultant de F_1, F_2 est un polynôme de degré 9 en $\sin\beta$ qui permet de résoudre le problème d'optimalité.

4.3 Extremums pour des rectangles verticaux

Supposons que l'espace de travail soit défini comme un rectangle vertical tel que les coordonnées de C (x, y, z) vérifient:

$$x_1 \leq x \leq x_2 \quad z_1 \leq z \leq z_2$$

Notons que le choix d'un rectangle dans un plan y fixé permet de calculer les extremums dans n'importe quel rectangle vertical par simple changement de repère. Par exemple une rotation de 90 degrés autour de l'axe z permettra d'utiliser le même algorithme pour calculer les extremums pour un rectangle dans un plan x fixé avec des bornes sur y, z.

Même avec les hypothèses simplificatrices adoptées ci-dessus le problème d'optimisation devient complexe. En effet le terme $z_{A_iB_i}$ qui intervient au dénominateur des expressions simplifiés devient ici une fonction d'une des inconnues alors que dans le cas des rectangles horizontaux il était constant. L'approche par optimisation n'a pu être utilisée que si les plateaux étaient parallèle. Dans ce cas pour la raideur k_x le résultant de F_1, F_2 est un polynôme de degré 10 en $\sin\beta$. Pour la raideur k_y, F_1 est linéaire en $\sin\alpha$ et F_2 devient un polynôme de degré 2 en $\sin\beta$. Les dérivées de la raideur k_z ne dépend que de $\sin\alpha$ et la raideur est donc extrémale soit pour $z = z_1$ ou $z = z_2$. Pour la raideur k_{tx} F_1 est linéaire en $\sin\alpha$ et après report F_2 devient un polynôme de degré 7 en $\sin\beta$. Pour la raideur k_{ty} F_1 est linéaire en $\sin\alpha$ et après report F_2 devient un polynôme de degré 2 en $\sin\beta$. Pour la raideur k_{tz} le résultant de F_1, F_2 est un polynôme de degré 10 en $\sin\alpha$.

Si les plateaux ne sont pas parallèles on pourra utiliser l'algorithme décrit dans la section suivante.

4.4 Extremums pour un parallélépipède rectangle

Reprenons les équations définissant les différentes raideurs principales. Pour k_x, k_y, k_{tz} nous avons:

$$k_x = \sum_{i=1}^{i=6} k_i \frac{x_{A_iB_i}^2}{x_{A_iB_i}^2 + y_{A_iB_i}^2 + z_{A_iB_i}^2} \tag{38}$$

$$k_y = \sum_{i=1}^{i=6} k_i \frac{z_{A_iB_i}^2}{x_{A_iB_i}^2 + y_{A_iB_i}^2 + z_{A_iB_i}^2} \tag{39}$$

$$k_{tz} = \sum_{i=1}^{i=6} k_i \frac{(x_{A_iB_i} y_b^i - y_{A_iB_i} z_b^i)^2}{x_{A_iB_i}^2 + y_{A_iB_i}^2 + z_{A_iB_i}^2} \tag{40}$$

Supposons que pour un rectangle horizontal obtenu à une certaine hauteur z comprise entre $z_1, z_2 > z_1$ pour C nous ayons obtenu un maxima k_m de la raideur en un point de coordonnées x, y. Étant données les formules précédentes l'altitude n'intervient qu'au dénominateur des expressions: il est donc clair qu'au point de coordonnées x, y, z_1 on obtiendra une raideur $k_M > k_m$. De même si k_m était un minima de la raideur on obtiendrait au point de coordonnés x, y, z_2 une raideur $k_M < k_m$. En conséquence les extremums des raideurs seront obtenus pour des points appartenant aux rectangle

horizontaux d'altitude z_1, z_2. Il est donc nécessaire de tester uniquement ces deux rectangles pour obtenir les extremums de ces raideurs.

Considérons maintenant les valeurs approchées des raideurs k_z, k_{tx}, k_{ty}:

$$k_z = \sum_{i=1}^{i=6} k_i(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$
$$k_{tx} = \sum_{i=1}^{i=6} k_i(y^i_b)^2(1 - 2\frac{y_{A_iB_i}z^i_b}{y^i_b\ z_{A_iB_i}})(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$
$$k_{ty} = \sum_{i=1}^{i=6} k_i(y^i_b)^2(1 - 2\frac{x_{A_iB_i}z^i_b}{x^i_b\ z_{A_iB_i}})(1 - \frac{x^2_{A_iB_i}}{z^2_{A_iB_i}} - \frac{y^2_{A_iB_i}}{z^2_{A_iB_i}})$$

On remarque que dans ces expressions approchées l'altitude z intervient seulement au dénominateur. Le même raisonnement que dans le paragraphe précédent peut être appliqué et il est donc nécessaire de tester uniquement les deux rectangles dans les plans $z = z_1, z_2$ pour obtenir les extremums de ces raideurs.

En conclusion la détermination des raideurs extrémales pour une boîte repose sur l'algorithme permettant le calcul des raideurs extrémales pour les rectangles horizontaux.

4.5 Raideurs extrémales pour tout type d'espace de travail

On suppose ici que l'espace de travail peut être décrit à l'aide de polygones horizontaux disposés successivement comme dans la section 2.3.

On ne considère que l'espace de travail compris entre deux sections, qui va être décomposé en autant de boîtes que nécessaire pour déterminer les raideurs extrémales avec la précision ϵ souhaitée. Une liste de boîtes, initialisée avec la boîte englobante de l'espace de travail B_0, est maintenue à jour dans l'algorithme. De plus un tableau T est créé pour sauvegarder les raideurs extrémales.

A l'itération k l'algorithme effectue les étapes suivantes :

1. si la boîte B_k est complètement en dehors de l'espace de travail on passe à la boîte suivante dans la liste

2. si la boîte B_k est complètement à l'intérieur de l'espace de travail on calcule les raideurs extrémales pour cette boîte. Le tableau des raideurs extrémales est mis à jour.

3. si la boîte B_k est partiellement à l'intérieur de l'espace de travail on coupe la boîte. L'algorithme de division prend en compte le fait que l'on a peu intérêt à découper les boîtes selon l'axe z. On procède de la manière suivante :

   (a) on découpe la boîte en deux selon l'axe x. Si l'une des boîtes est entièrement à l'intérieur de la région on substitue la boîte courante par celle-ci et l'on rajoute l'autre boîte issue de la bissection à la fin de la liste.

(b) si la bissection précédente n'a pas permis de trouver de boîte à l'intérieur on procède à une bissection selon l'axe y selon le même principe

(c) si on n'a toujours pas trouvé de boîte complètement à l'intérieur on procède à une bissection sur les axes x, y. Si l'une des quatre boîtes crées est à l'intérieur de la région on la substitue à la boîte courante et l'on met les trois autres boîtes en fin de liste.

(d) sinon on crée 8 nouvelles boîte à partir de B_k en divisant chacune de ses dimensions par 2. Ces boîtes sont placées à la fin de la liste et l'on passe à la boîte suivante

L'algorithme stoppe lorsque la liste est vide. Il permet de calculer les raideurs extrémales pour des espaces de travail complexes dans un temps raisonnable à condition d'applique un certain nombre d'heuristiques:

– on initialise les valeurs des extremums des raideurs en calculant les extremums sur les arêtes des polygones

– pour limiter les calculs exacts d'extremums on calcule à l'aide d'une analyse par intervalle un minorant et un majorant des extremums pour les segments, les rectangles et les boîtes: si ces valeurs sont dans les intervalles définis par les valeurs courantes des extremums on n'effectue pas le calcul

– si une boîte est partiellement à l'intérieur de l'espace de travail et que les dimensions de la boîte sont petites on suppose que la boîte est à l'intérieur de l'espace de travail et l'on applique une procédure plus rapide que la procédure générale: les raideurs sont simplement calculées aux sommets et au centre de la boîte.

4.6 Raideurs extrémales sur l'espace complet

L'algorithme précédent peut être étendu pour déterminer les extremums des raideurs dans un espace défini par un hypercube dans les coordonnées articulaires.

On initialise l'algorithme précédent avec une boîte cartésienne englobant l'espace de travail complet du robot (par exemple en utilisant la méthode décrite dans la section 3.5). On détermine ensuite l'appartenance d'une boîte à l'espace défini en articulaire en utilisant l'algorithme de détermination des longueurs articulaires extrêmes pour une boîte cartésienne décrite dans la section 2. On utilise par ailleurs le même type d'heuristiques que pour l'espace de travail défini à partir de coupe, en remplaçant l'initialisation des extremums par une procédure qui prend au hasard des points dans la boîte englobante et qui met à jour la valeur initiale des extremums si le point est bien à l'intérieur de l'espace de travail. De la même façon la procédure de calcul rapide des extremums prend en compte l'appartenance des points à l'espace de travail articulaire.

4.7 Temps de calcul

Le calcul des raideurs principales pour une configuration de l'organe terminal est d'environ 0.02 ms. Dans cette section pour comparer le temps de calcul entre les algorithmes décrit et le temps

obtenu par la méthode de discrétisation on utilisera la notion de *pas* qui indiquera quel est le pas de discrétisation sur chaque axe de l'objet décrivant l'espace de travail pour que les temps de calcul entre méthode directe et discrétisation soient similaires. Plus le pas sera faible plus la méthode directe est intéressante.

Les temps de calcul sont établis sur une station de travail SUN Ultra1 pour la "main gauche" INRIA. Les résultats sont indiqués dans la table 2. A la lecture de ces tables il apparaît que plus

Espace de travail	temps	pas
segment	10	202
rectangle horizontal	80	58
rectangle vertical	90	63
boîte	150	21

TAB. 2 – *Temps de calcul (en ms) et pas pour la méthode directe et différents types d'espace de travail*

l'espace est complexe plus la méthode directe est avantageuse. Notons cependant que même pour un segment la méthode directe peut être largement avantageuse dès que la longueur du segment est suffisante.

Pour les espaces de type région avec un seuil sur les dimensions de la boîte de 0.1 le temps de calcul est de 33s pour un pas de 60.

Pour l'espace complet les résultats sont plus mitigés. La forme générale de l'espace de travail d'un robot parallèle conduit à la création d'un nombre important de boîtes, fortement dépendant du seuil sur la dimension des boîtes à partir duquel on fait appel à la procédure rapide. Ainsi pour un seuil de 1 le temps de calcul est de l'ordre de 56 secondes pour un pas de 47 (avec cependant une amélioration du résultat qui peut aller jusqu'à 5%), pour un seuil de 0.5 il est de l'ordre de 141 secondes et un pas de 61 (avec une amélioration qui peut aller jusqu'à 6 %) et pour un seuil de 0.1 il est de l'ordre de 5 mn et le pas de 84 (avec une amélioration qui va jusqu'à 10 %).

5 Extremums des forces articulaires

Un point important lors de la conception d'un robot parallèle est la détermination des forces dans les segments lorsque le plateau mobile est soumis à une un torseur de force donné (par exemple issu d'une charge posée sur le plateau mobile). La relation entre le vecteur des forces articulaires τ et le torseur d'efforts $\mathcal{F}$ s'écrit:

$$\tau = J^T \mathcal{F} \tag{41}$$

Malheureusement la matrice J^T est lourde à manipuler, on va donc donc plutôt s'intéresser au système linéaire:

$$\tau = J^{-T} \mathcal{F} \tag{42}$$

Si τ_j est la j^{eme} composante du vecteur τ et si H est la matrice obtenue en remplaçant la j^{eme} colonne de J^{-T} par le vecteur $\mathcal{F}$ on a:

$$\tau_j = \frac{|H|}{|J^{-T}|} \tag{43}$$

On note que tous les éléments des colonnes H_i de H lorsque i varie de 1 à 6, $i \neq j$, ont comme numérateur ρ_i. Soit la matrice H' obtenue à partir de la matrice H en supprimant le numérateur des colonnes 1 à 6. On a alors:

$$|H| = \frac{|H'|}{\prod_{i=1, i\neq j}^{i=6} \rho_i} \tag{44}$$

On introduit alors la semi-jacobienne J_s^{-1} obtenue en supprimant tous les dénominateurs des éléments de J^{-1}. On a alors:

$$|J^{-T}| = \frac{|J_s^{-T}|}{\prod_{i=1}^{i=6} \rho_i} \tag{45}$$

En conséquence:

$$\tau_j = \rho_j \frac{|H'|}{|J_s^{-T}|} \tag{46}$$

Supposons alors que les éléments des matrices H', J_s^{-T} dépendent d'un paramètre λ et intéressons nous à la dérivée de τ_j par rapport à λ. On a:

$$\frac{\partial \tau_j}{\partial \lambda} = \frac{((\frac{\partial \rho_j}{\partial \lambda}|H'| + \frac{\partial |H'|}{\partial \lambda}\rho_j)|J_s^{-T}| - \rho_j |H'| \frac{\partial |J_s^{-T}|}{\partial \lambda})}{|J_s^{-T}|^2} \tag{47}$$

Comme on a $\rho_j = \sqrt{\rho_j^2}$ on obtient:

$$\frac{\partial \rho_j}{\partial \lambda} = \frac{\frac{\partial \rho_j^2}{\partial \lambda}}{2\rho_j}$$

En réduisant l'équation (47) au même dénominateur on obtient donc:

$$\frac{\partial \tau_j}{\partial \lambda} = \frac{((\frac{\partial \rho_j^2}{\partial \lambda}|H'| + 2\frac{\partial |H'|}{\partial \lambda}\rho_j^2)|J_s^{-T}| - 2\rho_j^2 |H'| \frac{\partial |J_s^{-T}|}{\partial \lambda})}{2\rho_j |J_s^{-T}|^2} \tag{48}$$

Le numérateur de cette expression a l'avantage d'être algébrique en terme du paramètre λ.

5.1 Extremums pour un segment

Supposons que le point C se déplace sur un segment défini par ses deux extrémités M_1, M_2. Toute position de C peut alors être définie par:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1 M_2} \tag{49}$$

où λ est un scalaire dans l'intervalle [0,1]. En substituant cette relation dans l'équation (48) on voit que le numérateur de la dérivée de τ_j par rapport à λ s'écrit de manière algébrique en fonction de λ. Si l'on suppose que l'origine du repère de référence est placé en A_1 (ce qu'il est toujours possible de faire) et après de longs calculs on obtient que cette dérivée s'écrit comme un polynôme de degré 7 en λ. Les extremums des forces articulaires s'obtiendront donc soit pour une des racines réelles de ce polynôme comprise dans l'intervalle [0,1] soit pour λ égal 0 ou 1.

Notons que si la base est plane ($za_i = 0$) et que la ligne est orientée selon l'axe x (ou y puisque l'on peut tourner librement le repère de référence autour de l'axe z) alors le polynôme n'est que de degré 6. De plus si la plate-forme et la base sont parallèles alors le degré du polynôme n'est plus que de 3 (nous appellerons ce cas le *cas spécial*).

5.2 Extremums pour un rectangle

5.2.1 Principe

On considère la cas où l'espace de travail est un rectangle que l'on va supposer être dans le plan $x - y$ (un changement de repère permet de toujours de ramener à ce cas). Si l'on suppose que $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$ on peut alors être tenté de poser:

$$x = x_1 + (1 + \sin\alpha)(x_2 - x_1)/2 \quad y = x_1 + (1 + \sin\beta)(x_2 - x_1)/2$$

Les extremums de τ seront alors obtenus lorsque:

$$\frac{\partial\tau}{\partial\alpha} = 0 \qquad \frac{\partial\tau}{\partial\beta} = 0$$

Ce faisant on obtient des polynômes de degré respectivement (7,8), (8,7) en $\sin\alpha, \sin\beta$, système qu'il est difficile de résoudre. On adopte alors une autre démarche. On va chercher à déterminer les extremums de τ avec une précision ϵ fixée. On commence par calculer les extremums τ_{min}, τ_{max} de τ sur le segment défini par $y = y_1, x_1 \leq x \leq x_2$ à l'aide de la procédure décrite dans la section précédente. On cherche ensuite à déterminer quelle est la valeur d'un déplacement dy^2 tel que les extremums de τ sur le segment $y = y_1 + dy^2, x_1 \leq x \leq x_2$ soient égale à $\tau_{min} - \epsilon$, $\tau_{max} + \epsilon$. On écrit donc les équations:

$$\tau = \tau_{min} - \epsilon \quad \tau = \tau_{max} + \epsilon$$

qui s'exprime en fonction de dy^2, λ, avec $x = x_1 + \lambda(x_2 - x_1)$. On suppose que dy^2 est petit et l'on développe au premier ordre les équations autour de $dy = 0$. On obtient deux équations de type:

$$A_2 dy^2 + A_0 = 0$$

Soit

$$dy^2 = -\frac{A_0}{A_2}$$

qui est une fonction de λ. La dérivée de dy^2 par rapport à λ est un polynôme de degré 22 en λ. On calcule alors ces racines pour déterminer la plus faible valeur de dy^2. Deux cas peuvent alors se produire:

- la valeur de dy^2 ainsi calculée est effectivement petite: on procède alors au calcul de la force articulaire au point défini par les λ, dy^2 calculés
- la valeur de dy^2 ainsi calculé soit n'existe pas (dy^2 est négatif) ou n'est pas petite: on prend alors comme valeur de dy^2 une valeur faible (0.3 dans notre implantation) et l'on calcule l'extremum de la force articulaire pour le segment $y = y_1 + dy^2, x_1 \leq x \leq x_2$

Dans les deux cas on a calculé une force articulaire: on met alors à jour les extremums courants τ_{min}, τ_{max} de la force articulaire et l'on réitère le processus en remplaçant y_1 par $y_1 + dy^2$. On arrête le processus lorsque la valeur de y excède y_2. En procédant ainsi on balaye le rectangle par des segments parallèles à l'axe y

5.2.2 Réduction du temps de calcul

Pour réduire le temps de calcul on utilise l'heuristique suivante: à partir de la formulation de la force articulaire on calcule l'expression du numérateur de sa dérivée en fonction de y (car le dénominateur est forcément positif) qui s'exprime de manière algébrique; par une simple analyse par intervalles on estime les valeurs extrêmes du numérateur: si les valeurs extrêmes sont de même signe alors la force est monotone en y et seul les extremums pour les segments ayant les y extrêmes sont calculés pour établir les extremums pour le rectangle complet.

5.2.3 Cas spécial

Dans ce cas la simplification de l'expression de la force articulaire permet d'utiliser l'approche par optimisation. La résolution des équations de contrainte se ramène à résoudre indépendamment deux équations du second ordre.

5.3 Extremums dans une boîte

5.3.1 Principe

Au vu de la section précédent il paraît clair que l'approche par optimisation va rencontrer des problèmes puisque l'on aurait à résoudre un système de trois équations de degré élevé. On va alors adopter une démarche identique à celle employée pour les rectangles. On suppose ici la boîte est définie par $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, $z_1 \leq z \leq z_2$. On calcule tout d'abord les extremums de la force articulaire pour le rectangle défini par $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, $z = z_1$, ceci à l'aide de la procédure décrite dans la section précédente. On obtient alors une estimée courantes des extremums τ_{min}, τ_{max}. On va ensuite chercher quelle est la variation minimum d'altitude dz^2 telle que les extremums $\tau^n_{min}, \tau^n_{max}$ calculés pour le nouveau rectangle à l'altitude $z_1 + dz^2$ soient telle que $\tau^n_{min} = \tau_{min} - \epsilon$ ou $\tau^n_{max} = \tau_{max} + \epsilon$, où ϵ est la précision désirée.

Là aussi on suppose que dz^2 est petit et on développe les équations:

$$\tau = \tau_{min} - \epsilon \quad \tau = \tau_{max} + \epsilon$$

au premier ordre. On obtient deux équations de type:

$$A_2 dz^2 + A_0 = 0$$

Soit

$$dz^2 = -\frac{A_0}{A_2}$$

qui est une fonction de λ, μ, si l'on écrit $x = x_1 + \lambda(x_2 - x_1)$, $y = y_1 + \mu(y_2 - y_1)$. Il s'agit alors de déterminer la valeur minimum de dz^2. On se trouve encore confronté à un problème car la dérivée de dz^2 par rapport à λ, μ sont des équations de degré élevé. Toutefois comme nous ne sommes intéressé que par les solutions dans l'intervalle [0,1] on va développer au second ordre ces équations autour des intervalles construit à partir des valeurs [0,1/2,1]. Dans chaque cas le résultant des deux équations ainsi obtenues est un polynôme de degré 4 en λ. Après résolution on obtient les valeurs de λ, μ ce qui permet de calculer les valeurs possibles de dz^2. De la même manière que dans la section précédente il y a alors deux cas:

- la valeur de dz^2 ainsi calculée est effectivement petite: on procède alors au calcul de la force articulaire au point défini par les λ, μ, dz^2 calculés
- la valeur de dz^2 ainsi calculé soit n'existe pas (dz^2 est négatif) ou n'est pas petite: on prend alors comme valeur de dz^2 une valeur faible (0.3 dans notre implantation) et l'on calcule l'extremum de la force articulaire pour le rectangle $y_1 \leq y \leq y_2, x_1 \leq x \leq x_2, z = z_1 + dz^2$

Dans les deux cas on a calculé une force articulaire: on met alors à jour les extremums courants τ_{min}, τ_{max} de la force articulaire et l'on réitère le processus en remplaçant z_1 par $z_1 + dz^2$. On arrête le processus lorsque la valeur de z excède z_2. En procédant ainsi on balaye la boîte par des rectangles parallèles au plan $x - y$

5.3.2 Réduction du temps de calcul

Pour réduire le temps de calcul on utilise une heuristique similaire à celle utilisée pour les rectangles: à partir de la formulation de la force articulaire on calcule l'expression du numérateur de sa dérivée en fonction de z (car le dénominateur est forcément positif) qui s'exprime de manière algébrique; par une simple analyse par intervalles on estime les valeurs extrêmes du numérateur: si les valeurs extrêmes sont de même signe alors la force est monotone en z et seul les extremums pour les rectangles ayant les z extrêmes sont calculés pour établir les extremums pour la boîte complète.

5.3.3 Cas spécial

Dans ce cas la simplification de l'expression de la force articulaire permet d'utiliser l'approche par optimisation. La résolution des équations de contrainte se ramène à résoudre indépendamment quatre équations monovariable de degré 2,3,4,8.

5.4 Forces extrémales pour tout type d'espace de travail

Le temps de calcul raisonnable de l'algorithme décrit pour les boîtes permet d'étendre le calcul des forces extrémales à des espaces de travail plus complexes.

On suppose ici que l'espace de travail peut être décrit à l'aide de polygones horizontaux disposés successivement comme dans la section 2.3.

On considère l'espace de travail compris entre deux polygones successifs, qui va être décomposé en autant de boîtes que nécessaire pour déterminer les vitesses articulaires extrémales avec la précision ϵ souhaitée. Une liste de boîtes, initialisée avec la boîte englobante de l'espace de travail B_0, est maintenue à jour dans l'algorithme. De plus un tableau est crée pour sauvegarder les forces extrémales.

A l'itération k l'algorithme effectue les étapes suivantes :

1. si la boîte B_k est complètement en dehors de l'espace de travail on passe à la boîte suivante dans la liste.
2. si la boîte B_k est complètement à l'intérieur de l'espace de travail on calcule les forces articulaires extrémales pour cette boîte. Le tableau des forces extrémales est mis à jour.
3. si la boîte B_k est partiellement à l'intérieur de l'espace de travail on calcule les forces articulaires extrémales pour cette boîte.
   (a) si toutes les forces sont incluses dans les limites définies dans le tableau des forces extrémales on passe à la boîte suivante.
   (b) sinon on crée 8 nouvelles boîte à partir de B_k en divisant chacune de ses dimensions par 2. Ces boîtes sont placées à la fin de la liste et l'on passe à la boîte suivante.

L'algorithme stoppe lorsque la liste est vide. Tel qu'exposé l'algorithme peut être relativement lent en raison d'un nombre important de boîte crées. On utilise deux heuristiques pour l'accélérer:

– on initialise le tableau des forces extrémales en calculant les extremums des forces le long des arêtes des polygones définissant l'espace de travail.

– pour limiter le nombre de calcul des forces extrémales dans une boîte on va calculer un minorant et un majorant pour les extremums des forces articulaires de chaque boîte et on ne procédera au calcul effectif des forces extrémales que si le minorant est plus petit que la force minimum courante ou le majorant plus grand que la force maximale courante. Pour calculer le minorant et le majorant rappelons que la force articulaire s'écrit sous la forme $\rho\, U/V$ où U, V s'exprime de manière algébrique par rapport à x, y, z. En utilisant une analyse par intervalle simple il est aisé d'obtenir des minorants et des majorants pour U, V et donc un minorant et un majorant pour leur rapport (on traitera spécialement le cas où 0 appartient à l'intervalle sur V). Quant au terme ρ on peut utiliser directement les résultats de la section 2 pour obtenir son intervalle de variation exact, ce qui conduit avec l'intervalle de variation sur U/V à un intervalle sur la force articulaire.

Moyennant ces heuristiques on obtient un algorithme qui permet de calculer les vitesses extrémales pour des espaces de travail complexes dans un temps raisonnable.

5.5 Forces extrémales sur l'espace complet

L'espace de travail peut être aussi défini comme un hypercube dans l'espace articulaire. L'algorithme est dans le principe identique, seul diffère le test sur l'appartenance des boîtes à l'espace de travail qui utilise l'algorithme décrit dans la section 2 permettant de calculer les extremums des longueurs des segments pour une boîte. Pour cela on part d'une boîte cartésienne B_0 pour C tel que les longueurs extrémales des segments pour toute position de C à l'intérieur de la boîte soient incluse dans l'hypercube articulaire. La détermination d'une telle boîte a été présentée dans la section 3.5. On utilise alors B_0 comme boîte initiale de l'algorithme précédent. Pour statuer sur l'appartenance d'une boîte vis-à-vis de l'hypercube articulaire il suffit de calculer les longueurs articulaires extrémales pour cette boîte en utilisant l'algorithme de la section 2: si les longueurs extrémales des 6 segments sont comprises dans l'hypercube articulaire alors la boîte est considéré comme totalement interne, si les longueurs extrémales d'au moins un segment sont en dehors des limites de l'hypercube alors la boîte est externe, sinon elle est considéré comme partiellement interne.

Lors de l'implantation il est nécessaire de respecter certaines précautions:

- il est prudent d'initialiser le tableau des forces extrémales: pour cela nous prenons des points au hasard dans l'espace de travail et l'on calcule les forces pour ces positions
- si pour une boîte les longueurs extrêmes des segments sont largement supérieure aux limites articulaires il est préférable de couper la boîte plutôt que de calculer d'abord les extremums des forces
- il faut traiter spécialement les boîtes de dimensions très faibles (on peut créer des boîtes dont le centre est pratiquement sur la frontière de l'espace de travail, qui serait alors découpées très finement)

5.6 Temps de calcul

Pour un SUN Ultra 1 remarquons tout d'abord que le temps de calcul des forces articulaires pour une posture donnée de l'organe terminal est de l'ordre de 0.49 ms.

Pour le calcul des extremums sur un segment le temps de calcul s'établit à 55 ms dans le cas général et à 15 ms si la base est plane. Pour un rectangle horizontal et une précision de 1 (avec des forces externes produisant des forces articulaires de l'ordre de plusieurs centaines d'unité) le temps de calcul dans le cas d'un robot général s'établit à 2500 ms alors que dans le cas où la base est plane ce temps s'établit à environ 1600 ms (il faut cependant dire que ce cas a fait l'objet d'une implantation beaucoup plus soignée que le cas général). Dans le cas spécial ce temps se réduit à 40 ms. La figure 5 montre le temps de calcul lorsque l'on a une base plane en fonction de la précision souhaitée.

Pour la boîte on peut déjà estimer la complexité de l'algorithme et la comparer à celle de la méthode de discrétisation avec n point par axe (on procédera donc à n^3 calcul des forces articulaires). En remarquant que l'on effectue un balayage des rectangles selon l'axe y et de la boîte selon l'axe z la complexité est donc en n^2. Il conviendra cependant de choisir comme axe y, z les deux axes dont les variations sont les plus faibles.

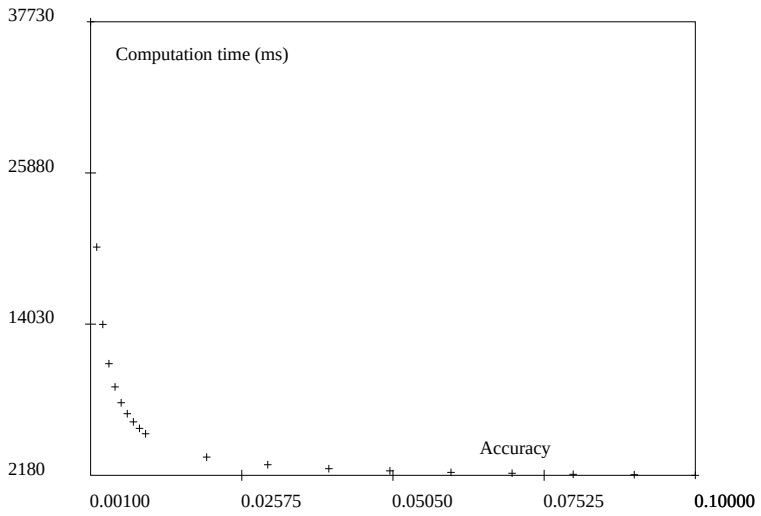

FIG. 5 – *Temps de calcul des forces extrémales en fonction de la précision souhaitée pour un rectangle horizontal*

Pour une précision de 1 (avec des forces externes produisant des forces articulaires de l'ordre de plusieurs centaines d'unité) le temps de calcul s'établit à 640 a 21000 ms pour un robot général et 3000 à 4000 ms dans le cas où la base est plane (avec là aussi une implantation plus soignée). Dans le cas spécial le temps de calcul s'établit entre 120 et 230 ms. La figure 6 montre le temps de calcul lorsque l'on a une base plane en fonction de la précision souhaitée.

Pour les calculs dans une région on a considéré l'espace de travail défini par les trois coupes suivantes: $z = 50, x \in [-10, 10], y \in [-10, 10]$, $z = 55, x \in [-5, 5], y \in [-5, 5]$, $z = 60, x \in [-10, 10], y \in [-10, 10]$ représenté en figure 2.

Pour un robot général le temps de calcul varie entre 65s et 165s, alors que si la base est plane ce temps s'établit entre 7 et 9 secondes (1.2s dans le cas spécial). La sensibilité du temps de calcul à la précision demandée est illustrée par la figure 7. On peut voir que même pour une précision de 0.1 N le temps de calcul est raisonnable à 6570 ms.

Pour l'espace complet pour une précision de 1 (5) le temps de calcul s'établit à 4.5 mn (100 secondes). Dans le cas spécial ce temps tombe à 2.5 mn (60 secondes).

6 Singularités

Déterminer si un espace de travail contient une singularité est très important pour la conception des robots parallèles. Rappelons que parmi l'ensemble des manières permettant d'introduire les singularités on peut utiliser celle reposant sur la relation entre les forces articulaires dans les segments

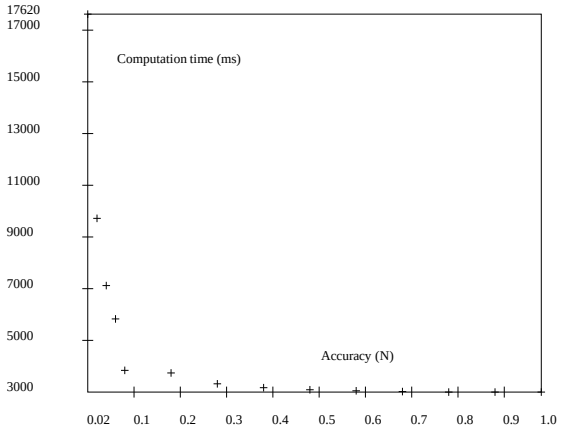

FIG. 6 – *Temps de calcul des forces extrémales en fonction de la précision souhaitée pour une boîte*

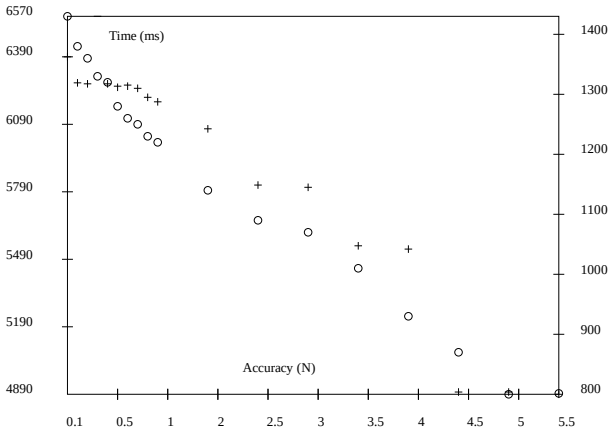

FIG. 7 – *Temps de calcul des forces extrémales en fonction de la précision pour un robot à base plane (les croix représentent le cas général alors que les cercles représentent le cas spécial), cas d'une région.*

τ et les forces/moments $\mathcal{F}$ s'exerçant sur la plate-forme. Cette relation s'écrit:

$$\tau = J^{-T}\mathcal{F} \tag{50}$$

Le vecteur τ est donc solution d'un système linéaire d'équations. Chacune des composantes du vecteur τ s'écrit donc avec comme dénominateur le déterminant de la matrice jacobienne inverse. Si ce déterminant est égal à 0 les forces articulaires peuvent aller à l'infini, provoquant la destruction du robot. On peut aussi montrer que dans ce cas on perd le contrôle du mécanisme puisqu'il peut y avoir mouvement de la plate-forme alors que les segments ne changent pas de longueur. Rappelons qu'une ligne de la matrice jacobienne inverse s'écrit:

$$\frac{\mathbf{AB}}{||\mathbf{AB}||} \qquad \frac{\mathbf{CB} \wedge \mathbf{AB}}{||\mathbf{AB}||}$$

En conséquence le déterminant de cette matrice est clairement dépendant de la posture de l'organe terminal. Il est éventuellement possible de calculer ce déterminant mais son expression est très complexe et ne permet pas à priori de déterminer si une configuration singulière existe dans un espace de travail donné. Nous avons proposé une méthode [21] qui permet de montrer que les singularités surviennent pour des dispositions géométriques particulières des lignes associées aux segments. En exprimant individuellement chacune de ces conditions on arrive alors à exprimer un ensemble de conditions de singularités, beaucoup plus simple que l'expression complète du déterminant. Il suffit de vérifier alors que les variétés décrites par ces conditions intersectent l'espace de travail. Toutefois cette approche est complexe dans le cas général et est plus approprié à une analyse graphique des lieux de singularité ou à une analyse très fine des différents types de singularité.

On peut alors penser à des méthodes numériques de minimisation. Cette approche est cependant peu sûre dans le sens où un résultat négatif (pas de posture conduisant à une annulation du déterminant) serait entaché d'un doute. Il se pose de plus un problème numérique: remarquons en effet que les termes de la matrice jacobienne inverse ne sont pas sans dimension. En effet les composantes du vecteur $\mathbf{CB} \wedge \mathbf{AB}/||\mathbf{AB}||$ dépendent de l'unité de longueur utilisée et qu'en conséquence il en sera de même pour la valeur du déterminant. Il se pose donc le problème de la définition numérique du zéro.

Nous allons proposer dans cette section un algorithme qui permet de détecter la présence d'une singularité dans un espace de travail.

Pour cela remarquons que l'expression du déterminant de J^{-1} est complexe: un développement complet conduit à une expression de plusieurs dizaines de milliers de termes. De plus cette expression fait intervenir les longueurs des segments qui s'exprime comme racine carré en fonction de la position de C. Définissons alors la *semi-jacobienne inverse* J_s^{-1} par:

$$J_s^{-1} = ((\mathbf{AB} \qquad \mathbf{CB} \wedge \mathbf{AB}))$$

On a alors:

$$|J^{-1}| = \frac{|J_s^{-1}|}{\prod_{i=1}^{i=6} \rho_i}$$

En conséquence on a:

$$|J^{-1}| = 0 \Leftrightarrow |J_s^{-1}| = 0$$

L'expression de $|J_s^{-1}|$ sera maintenant algébrique en fonction des coordonnées de C et son expression sera plus simple que celle de $|J^{-1}|$.

On s'intéresse maintenant à la présence des singularités dans un espace de travail. Le principe des algorithmes que nous allons décrire est de rechercher le minimum S_m et le maximum S_M du déterminant de la semi-jacobienne inverse lorsque C décrit son espace de travail. La fonction $|J_s^{-1}|$ étant continue une singularité sera présente seulement si:

$$S_M \; S_m \leq 0$$

En utilisant cette approche la robustesse numérique est améliorée sauf dans le cas où S_M ou S_m est proche de 0. Par contre on a renforcé la sensibilité vis-à-vis du choix des unités de longueurs.

On peut maintenant exposer les algorithmes pour la recherche de S_m, S_M selon les types d'espace de travail.

6.1 Singularité sur un segment

Sur un segment allant de M_1 à M_2 on utilise le paramétrage:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1 M_2}$$

$|J_s^{-1}|$ devient alors un polynôme P de degré 3 en λ. Les extremums sont alors obtenus soit pour $\lambda = 0, 1$ où pour les valeurs de λ qui annulent la dérivée de P. Notons que l'on pourra détecter ici la présence d'une singularité permanente (par exemple dans le cas de la "main gauche" INRIA lorsque le plateau mobile est parallèle à la base et tournée à 90 degrés par rapport à celle-ci) puisque tous les coefficients du polynôme seront nuls.

6.2 Singularité dans un rectangle

Le problème est la recherche d'optimum sous la contrainte

$$x_1 \leq x \leq x_2 \quad y_1 \leq y \leq y_2$$

pour les rectangles horizontaux et

$$x_1 \leq x \leq x_2 \quad z_1 \leq z \leq z_2$$

pour les rectangles verticaux. On a pu obtenir la valeur de $|J_s^{-1}|$ en fonction de $x, y(z)$ En posant

$$x = x_1 + \frac{(1 + \sin\alpha)(x_2 - x_1)}{2}$$
$$y(z) = y_1 + \frac{(1 + \sin\beta)(y_2 - y_1)}{2}$$

on obtient que les extremums sont obtenus soit sur les bords du rectangle (on applique alors l'algorithme de la section précédente) soit à l'intérieur. Dans ce dernier cas si l'on pose $T_1 = tan(\alpha/2), T_2 = tan(\beta/2)$ on obtient deux équations de contraintes en $T_1, T2$. Pour les rectangles horizontaux leur résultant est un polynôme de degré 4 en T_2. Dans le cas particulier où la base est plane les équations de contraintes sont linéaires en T_1, T_2. Pour les rectangles verticaux le résultant est toujours de degré 4.

6.3 Singularité dans un parallélépipède rectangle

6.3.1 Robot à base plane

Dans ce cas on pose:

$$x = x_1 + \frac{(1+\sin\alpha)(x_2 - x_1)}{2}$$
$$y = y_1 + \frac{(1+\sin\beta)(y_2 - y_1)}{2}$$
$$z = z_1 + \frac{(1+\sin\mu)(z_2 - z_1)}{2}$$

puis

$$T_1 = tan(\alpha/2) \quad T_2 = tan(\beta/2) \quad T_3 = tan(\mu/2)$$

En dérivant $|J_s^{-1}|$ par rapport aux inconnues T_1, T_2, T_3 on obtient trois équations de contrainte. Les dérivées par rapport à T_1, T_2 sont linéaires dans ces inconnues. Après report dans la dérivée par rapport à T_3 on obtient un polynôme de degré 8 en T_3.

6.3.2 Robot général

Dans ce cas l'approche par optimisation est plus délicate car chacune des équations de contrainte est de degré 2 dans chacune des variables: le calcul des solutions de ce système devient alors ardu. Nous utilisons alors une autre méthode: le déterminant s'exprimant comme une fonction algébrique dans les inconnues x, y, z (qui sont bornées) une simple analyse par intervalle permet d'obtenir un minorant et un majorant de la valeur de ce déterminant sur la boîte. Si ces deux quantités ont même signe il n'y a pas de singularité dans la boîte. Dans le cas contraire il faut affiner le résultat puisqu'il peut y avoir un pessimisme excessif sur la valeur du minorant et du majorant. Pour cela on crée une liste de 8 boîte en divisant les dimensions de la boîte initiale par 2. On recommence alors le calcul du minorant et du majorant sur chaque boîte de la liste. Si ces quantités ont même signe on passe à la boîte suivante dans la liste, sinon on divise à nouveau la boîte courante. Ce faisant on est susceptible de créer un nombre important de sous-boîtes. Pour diminuer le temps de calcul on a développé un algorithme spécifique aux boîtes de petite dimensions en développant la valeur du déterminant au troisième ordre en $x_2 - x_1, y_2 - y_1, z_2 - z_1$. Les équations de contraintes deviennent alors linéaires dans les trois inconnues T_1, T_2, T_3 et l'on peut calculer les extremums du déterminant pour la boîte.

6.4 Singularité dans tout type d'espace de travail

On suppose ici que l'espace de travail peut être décrit à l'aide de polygones horizontaux disposés successivement comme dans la section 2.3.

On ne considère que l'espace de travail compris entre deux sections, qui va être décomposé en autant de boîtes que nécessaire pour déterminer si le produit $S_M\ S_m$ peut être nul ou négatif pour une posture incluse dans l'espace de travail.

Une liste de boîte est maintenue à jour dans l'algorithme. Elle est initialisée avec la boîte englobante de l'espace de travail B_0. De plus un tableau est crée pour sauvegarder S_M^e, S_m^e.

L'algorithme renvoit 1 s'il a détecté une singularité dans l'espace de travail.

A l'itération k l'algorithme effectue les étapes suivantes :

1. si la boîte B_k est complètement en dehors de l'espace de travail on passe à la boîte suivante dans la liste.
2. si la boîte B_k est complètement à l'intérieur de l'espace de travail on calcule les extremums de $|J_s^{-1}|$ pour cette boîte. Si cette boîte contient une singularité on renvoit 1.
3. si la boîte B_k est partiellement à l'intérieur de l'espace de travail on calcule les extremums de $|J_s^{-1}|$ pour cette boîte. Si ces extremums sont de même signe on passe à la boîte suivante dans la liste. Sinon :
   (a) si les positions de C correspondant aux extremums sont dans l'espace de travail l'algorithme renvoit 1
   (b) sinon on crée 8 nouvelles boîtes à partir de B_k en divisant chacune de ses dimensions par 2. Ces boîtes sont placées à la fin de la liste et l'on passe à la boîte suivante.

L'algorithme renvoit 0 lorsque la liste est vide. Le temps de calcul de cet algorithme est évidemment très dépendant de l'espace de travail. La figure 8 montre ainsi un espace de travail polyédrique et le découpage en boîte effectué par l'algorithme pour déterminer la présence d'une singularité dans la partie de l'espace travail comprise entre les deux premières coupes.

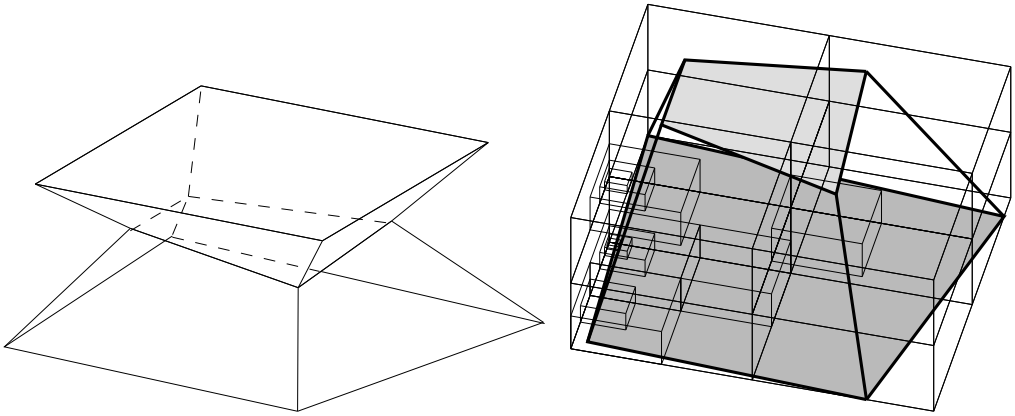

FIG. 8 – *Un volume de travail et son découpage en boîte qui permet de montrer la présence d'une singularité dans le volume*

6.5 Singularité dans l'espace complet

L'algorithme précédent peut être étendu pour déterminer la présence de singularité dans un espace défini par un hypercube dans les coordonnées articulaires.

On initialise l'algorithme précédent avec une boîte cartésienne englobant l'espace de travail complet du robot (par exemple en utilisant la méthode décrite dans la section 3.5). On détermine ensuite l'appartenance d'une boîte à l'espace défini en articulaire en utilisant l'algorithme de détermination des longueurs articulaires extrêmes pour une boîte cartésienne décrite dans la section 2.

6.6 Robustesse numérique

Comme mentionné précédemment on peut rencontrer des problèmes numériques lors de l'utilisation de l'algorithme, en particulier si le déterminant, calculé pour une posture où il est extrémal, est proche de 0 (rappelons qu'en raison du caractère non homogène de la jacobienne inverse on pourra toujours choisir une unité de longueur telle que pour une posture donnée la valeur du déterminant soit aussi petite que l'on veut).

Pour résoudre ce problème on peut commencer par modifier l'unité de longueur utilisée de façon à ce que les termes de la semi-jacobienne inverse aient un ordre de grandeur raisonnable. Ceci fait il reste à gérer les cas où le déterminant est effectivement proche de 0. Pour cela nous utilisons la méthode de Clarkson, telle qu'implantée par H. Bronnïmann, qui permet l'évaluation exacte du signe d'un déterminant dont les éléments sont codés sur des entiers, n'excédant pas 2^{53} [5]. Dans notre implantation si le déterminant calculé numériquement a une valeur inférieure à une valeur fixée nous normalisons les flottants a_{ij} de J_s^1 en prenant la partie entière de $2^{53}a_{ij}/Max(a_{ij})$, puis en utilisant la méthode de Clarkson sur la matrice d'entiers ainsi obtenue. Si cette méthode n'est pas parfaite (la normalisation pouvant entraîner un changement de signe du déterminant) elle améliore cependant sensiblement la robustesse numérique.

6.7 Temps de calcul

Les temps de calcul s'établissent de la manière suivante (entre parenthèses le cas où la base est plane):

segment	rectangle horizontal	rectangle vertical	boîte
2.875	0.24 (0.13) ms	0.6 (0.15) ms	20-131 (13.3) ms

A noter que la majorité du temps de calcul pour un segment est consacrée au calcul d'un certain nombre de constantes qui seront utilisées dans tous les cas. Une fois ces constantes calculées le temps se réduit à 0.03 ms. La recherche de singularité dans le volume test prend environ 40 ms si la base est plane et 1300 ms dans le cas général. Pour l'espace complet le temps est extrêmement dépendant de la géométrie et de l'orientation: pour le robot test avec des angles d'orientation tous nuls il va s'établir à 20 ms alors que pour un angle θ à 100 degrés il s'établira à 1620 ms (les temps sont sensiblement similaires que la base soit plane ou non).

6.8 Singularités pour les mouvements de rotation élémentaire

Dans cette section on va supposer que la position du point C est fixée et que seul des mouvements de rotation autour de C sont permis, en l'occurence qu'un seul des angles d'Euler qui définit l'orientation de la plate-forme mobile va varier. On pose alors:

$$\psi = \psi_1 + \lambda(\psi_2 - \psi_1) \quad \text{ou} \theta = \theta_1 + \lambda(\theta_2 - \theta_1) \quad \text{ou} \phi = \phi_1 + \lambda(\phi_2 - \phi_1)$$

Pour chacun des mouvements de rotation élémentaire on peut calculer le déterminant de J_s^{-1} comme fonction des sinus et cosinus de la quantité $\lambda(\alpha_2 - \alpha_1)$ où α représente l'angle d'Euler de la rotation élémentaire choisie. La dérivation de ce déterminant par rapport à λ donne une expression dans les sinus et cosinus de $\lambda(\alpha_2 - \alpha_1)$ qui est transformé en équation algébrique en utilisant le changement de variable classique. Quelle que soit la rotation élémentaire choisie on obtient un polynôme de degré 10 dans l'inconnue $\tan(\lambda(\alpha_2 - \alpha_1)/2)$. La résolution numérique de ce polynôme permet de déterminer les valeurs de λ dans l'intervalle [0,1] conduisant à un extremum du déterminant de J_s^{-1}.

Notons que cette analyse a permis de confirmer un fait peu connu. On sait que pour un ensemble de longueurs de segment fixé il existe plusieurs configurations possibles pour la plate-forme mobile. A une certaine époque on pensait (on espérait même..) que ces différentes configurations étaient séparées par des surfaces de singularité (si cela était vrai il aurait été alors possible de trier les solutions pour déterminer la configuration effective du robot puisqu'elle aurait été la seule à pouvoir être atteinte à partir de la position de montage initial sans passer par une singularité). Malheureusement cette hypothèse s'est avérée inexacte comme l'ont montré sur des exemples Innocenti [12] et Chablat [4]. Nous disposons actuellement d'un algorithme qui permet de trouver l'ensemble des configurations pour un robot à base plane. En utilisant les résultats des sections précédentes il est facile de tester si l'on peut passer d'une configuration $S_1 = (C_1, \psi_1, \theta_1, \phi_1)$ à une autre $S_2 = (C_2, \psi_2, \theta_2, \phi_2)$ sans passer par une singularité en suivant une trajectoire particulière: on procède d'abord à une translation pour amener le point C de C_1 à C_2 à l'autre puis on change individuellement chacun des angles d'Euler pour amener la plate-forme dans l'orientation de S_2. On a alors pu trouver de très nombreux exemples où cette trajectoire ne traversait pas de singularité.

6.9 Singularités sur une courbe paramétrée

On suppose ici que le point C décrit une courbe paramétrée polynomiale:

$$x = f_1(t) \quad y = f_2(t) \quad z = f_3(t)$$

où t est le paramètre et f_1, f_2, f_3 des polynômes. Le déterminant de la semi-jacobienne s'écrit comme fonction de x, y, z comme:

$$\begin{aligned} & b_1 + b_2\, y^2 + b_3\, z^2 + b_4\, x + b_5\, y + b_6\, z + b_7\, z^3 + \\ & b_8\, x^2 + b_9\, y^2 z + b_{10}\, z x^2 + b_{11}\, x z y + \\ & b_{12}\, z^2 y + b_{13}\, z^2 x + b_{14}\, z y + b_{15}\, x z + b_{16}\, x y \end{aligned}$$

où les coefficients b_i ne dépendent que de la géométrie et de l'orientation de la plate-forme mobile. A partir des expressions de x, y, z on déduit une formulation polynomiale du déterminant. Il est alors aisé de calculer sa dérivée pour déterminer les extremums du déterminant.

7 Extremum des erreurs de positionnement

Les erreurs de positionnement de la plate-forme $\Delta\mathbf{X}$ sont liées aux erreurs de mesure $\Delta\rho$ sur les variables articulaires par la relation:

$$\Delta\mathbf{X} = J\Delta\rho \tag{51}$$

Le but de cette section est de déterminer les extremums de $\Delta\mathbf{X}$ lorsque les $\Delta\rho$ sont bornées. Notons immédiatement que ce calcul permet de déterminer les extremums des vitesses de la plate-forme pour des vitesses articulaires bornées. Si l'on note J_s^{ij} le mineur de la matrice semi-jacobienne inverse obtenu en supprimant la i^{eme} ligne et la j^{eme} colonne on obtient:

$$\Delta X_i = \sum_{j=1}^{j=6} \frac{\Delta\rho_j\ \rho_j |J_s^{ji}|}{|J_s^{-1}|} \tag{52}$$

Contrairement aux autres sections une analyse par optimisation des extremums de ΔX_i est difficile même si l'espace de travail est un simple segment car les dérivées de l'expression sont complexes et non algébriques. Nous avons donc décidé d'utiliser une analyse par intervalles pour déterminer les extremums lorsque l'espace de travail est un parallélépipède rectangle B_0. En effet nous avons vu dans la section 2 comment calculer les extremums de ρ_j et de plus les expressions de $|J_s^{ji}|$ et $|J_s^{-1}|$ sont relativement simples et algébriques dans les inconnues x, y, z. A partir des majorants des extremums de ces trois expressions il est aisé de calculer un majorant des extremums de $\rho_j|J_s^{ji}|/|J_s^{-1}|$, puis de choisir la borne sur $\Delta\rho_j$ qui amène à un extremum de chaque terme de la somme (52). Nous pouvons alors calculer l'erreur sur ΔX_i avec une erreur ϵ_i maximale garantie de la manière suivante. Pour cela on crée une liste $\mathcal{L}$ de boîte initialisée avec la boîte B_0.

1. on calcule les erreurs $\Delta\mathbf{X}$ aux sommets de B_0. Le tableau $\mathcal{T}$ des valeurs extremums des $\Delta\mathbf{X}$ est initialisé à l'aide de ces valeurs. Un index k est initialisé à 0.

2. on calcule les majorants des erreurs avec l'analyse par intervalles pour la boîte B_k de la liste $\mathcal{L}$:

   (a) si tous les majorants sont inclus dans les limites de $\mathcal{T} \pm \epsilon$ on pose $k = k + 1$ et l'on revient en 2)

   (b) sinon on découpe la boîte B_k en 8 boîtes en divisant ces dimensions par 2. Chacune de ces boîtes est ajoutée à $\mathcal{L}$ si au moins un des majorants des erreurs pour la boîte est en dehors des limites de $\mathcal{T} \pm \epsilon$. On pose $k = k + 1$ et l'on revient en 2)

L'algorithme stoppe lorsque l'ensemble des boîtes ont été examinées. Si l'algorithme est simple dans son principe il est relativement lent. En effet l'analyse par intervalles produit une version très pessimiste des majorants, ce qui conduit à considérer un nombre de boîtes important. Nous avons pu observer que très fréquemment les extremums sont obtenus pour les sommets de B_0, sans avoir en pu expliquer la raison. Prenons un exemple: pour le robot test nous avons défini B_0 par les intervalles [-5,5],[-5,5],[50,55] avec des erreurs articulaires comprises en [-1,1]. Les extremums des erreurs sont alors $\pm 15.48, \pm 14.47, \pm 2.43, \pm 0.308, \pm 0.33, \pm 1.68$ et sont obtenus au sommets de B_0. Si l'on

fixe les erreurs ϵ à $\pm 0.1, \pm 0.1, \pm 0.1, \pm 0.01, \pm 0.01, \pm 0.01$ l'algorithme crée 1073 boîtes (figure 9) et nécessite 41s de temps de calcul, alors que dès le départ on a initialisé $\mathcal{T}$ avec les valeurs des extremums.

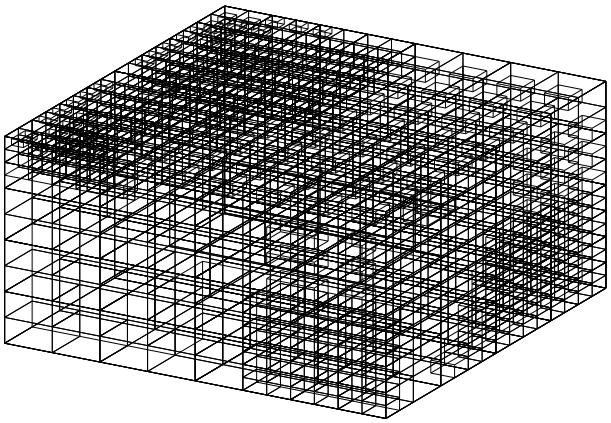

FIG. 9 – *Découpage d'une boîte en sous-boîte pour le calcul des extremums des erreurs de positionnement*

8 Conclusion

Les algorithmes présentés dans les sections précédentes permettent de déterminer dans un temps raisonnable la valeur exacte (ou approchée avec une précision arbitraire) quelques une des caractéristiques des robots parallèles qui interviennent dans la phase de conception. Ils sont sans doute encore susceptibles d'être améliorés pour une meilleure vitesse d'exécution. Nous n'avons cependant ici traité que les espaces de travail en translation et il est donc encore nécessaire d'effectuer une discrétisation pour la partie orientation.

A noter aussi une application intéressante des algorithmes reposant sur le découpage d'un espace de travail en boîte. Il arrive fréquemment que l'on désire trouver une trajectoire reliant deux points dans l'espace de travail qui présente certaines caractéristiques (par exemple une contrainte sur la raideur minimale du robot sur la trajectoire). Pour cela il serait tout à fait concevable de procéder préalablement à un découpage de l'espace de travail en cellule cubique où les extremums de la caractéristique sont connus. Une fois ce découpage effectué la détermination de la trajectoire optimale se ramène à un parcours de graphe entre les centres des cellules, problème pour lesquels on dispose de nombreuses solutions.

Enfin mentionnons que si les algorithmes ont été présentés pour une architecture particulière de robot parallèle il serait toutefois envisageable d'appliquer ces techniques à d'autres architectures.

9 Annexe

Les coordonnées des points A, B du robot test sont:

-9.7	9.1	0
9.7	9.1	0
12.76	3.9	0
3	-13	0
-3	-13	0
-12.76	3.9	0

-3	7.3	0
3	7.3	0
7.822	-1.052	0
4.822	-6.248	0
-4.822	-6.248	0
-7.822	-1.052	0

Les longueurs minimum et maximum des segments sont respectivement de 50 et 55. Dans le cas de test nécessitant un robot sans base plane les coordonnées en z de certains points A ont été arbitrairement mis à 1.

Références

[1] Beck A.S. et McCloy D. A comparative study of the power requirement of several basic robot designs. In *16th Int. Symp. on Industrial Robot designs*, pages 77–86, Bruxelles, 30 Août-2 Septembre 1986.

[2] Bernelli-Zazzera F. et Gallieni D. Analysis and design of an hexapod mechanism for autonomous payload pointing. In *46th IAF Congress*, Oslo, 2-6 Octobre 1995.

[3] Bhattacharya S., Hatwal H., et Ghosh A. On the optimum design of a Stewart platform type parallel manipulators. *Robotica*, 13(2):133–140, Mars- Avril 1995.

[4] Chablat D. et Wenger P. Domaine d'unicité pour les robots parallèles. Rapport de Recherche 96-13, Ecole Centrale, Nantes, Décembre 1996.

[5] Clarkson K. L. Safe and effective determinant evaluation. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, 1992.

[6] Claudinon B. et Lievre J. Test facility for rendez-vous and docking. In *36th Congress of the IAF*, pages 1–6, Stockholm, 7-12 Octobre 1985.

[7] Douady D. *Contribution à la modélisation des robots parallèles: conception d'un nouveau robot à 3 liaisons et six degrés de liberté*. Thèse de doctorat, Université Paris VI, Paris, 9 Décembre 1991.

[8] Erdman A.G. *Modern Kinematics*. Wiley, New-York, 1993.

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

[10] Gosselin C., Perreault T., et Vaillancourt C. Smaps: a computer-aided design package for the analysis and optimization of a spherical parallel manipulators. In *ISRAM*, pages 115–120, Hawaï, 14-18 Août 1994.

[11] Han C-S., Hudgens J.C., Tesar D., et Traver A.E. Modeling, synthesis, analysis and design of high resolution micromanipulator to enhance robot accuracy. In *IEEE Int. Workshop on Intelligent Robot and Systems (IROS)*, pages 1153–1162, Osaka, 3-5 Novembre 1991.

[12] Innocenti C. et Parenti-Castelli V. Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators. In *22nd Biennial Mechanisms Conf.*, pages 553–560, Scottsdale, 13-16 Septembre 1992.

[13] Kang H.J. et Freeman R.A. An interactive software package (MAP) for the dynamic modeling and simulation of parallel robotic systems including redundancy. In *ASME Int. Computer in Engineering Conf.*, pages 117–123, Boston, 5-9 Septembre 1990.

[14] Larochelle P.M. Design of 3-dof spherical robotic mechanisms. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1826–1830, Milan, 30 Août-2 Septembre 1995.

[15] Ling S-H. et Huang M.Z. Kinestatic analysis of general parallel manipulators. In *ASME Mechanisms Design Conf.*, Minneapolis, 14-16 Septembre 1994.

[16] Ma O. et Angeles J. Optimum architecture design of platform manipulator. In *ICAR*, pages 1131–1135, Pise, 19-22 Juin 1991.

[17] Martinez J.M.R. et Duffy J. A simple method for the velocity and acceleration analysis of in-parallel platforms. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 842–846, Milan, 30 Août-2 Septembre 1995.

[18] Masory O. et Wang J. Workspace evaluation of Stewart platforms. In *22nd Biennial Mechanisms Conf.*, pages 337–346, Scottsdale, 13-16 Septembre 1992.

[19] Merlet J-P. *Les Robots parallèles*. Hermès, Paris, 1990.

[20] Merlet J-P. Articular velocities of parallel manipulators, Part II: Finding all the robots with fixed extremal articular velocity for performing a fixed cartesian velocity over a whole workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 3262–3267, Albuquerque, 21-28 Avril 1997.

[21] Merlet J-P. Singular configurations of parallel manipulators and Grassmann geometry. *Int. J. of Robotics Research*, 8(5):45–56, Octobre 1989.

[22] Merlet J-P. Designing a parallel robot for a specific workspace. Rapport de Recherche 2527, INRIA, Avril 1995.

[23] Murray A.P., Pierrot F., Dauchez P., et J.M. McCarthy. On the design of parallel manipulators for a prescribed workspace: a planar quaternion approach. In J. Lenarčič V. Parenti-Castelli, editor, *Recent Advances in Robot Kinematics*, pages 349–357. Kluwer, 1996.

[24] Ouerfelli M. et Kumar V. Optimization of a spherical five bar parallel drive linkage. In *ASME Design Automation Conf.*, Miami, 22-25 Septembre 1991.

[25] Pittens K.H. et Podhorodeski R.P. A family of Stewart platforms with optimal dexterity. *J. of Robotic Systems*, 10(4):463–479, Juin 1993.

[26] Ramachandran S. et al. A finite element approach to the design and dynamic analysis of platform type manipulators. *Finite elements in Analysis and Design*, 10:335–350, 1992.

[27] Reboulet C. et Leguay S. The interest of redundancy for the design of a spherical parallel manipulator. In J. Lenarčič V. Parenti-Castelli, editor, *Recent Advances in Robot Kinematics*, pages 369–378. Kluwer, 1996.

[28] Ropponen T. et Arai T. Accuracy analysis of a modified Stewart platform manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 521–525, Nagoya, 25-27 Mai 1995.

[29] Salcudean S.E. et al. A six degree-of-freedom, hydraulic, one person motion simulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 2437–2443, San Diego, 8-13 Mai 1994.

[30] Smith III W.F. et Nguyen C.C. Mechanical analysis and design of a six-degree-of-freedom robotic wrist for Space assembly. In *Proc. 23th South Eastern Symp. on System*, pages 177–181, Columbia, 10-12 Mars 1991.

[31] Sorli M. et al. Mechanics of Turin parallel robot. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1880–1885, Milan, 30 Août-2 Septembre 1995.

[32] Stoughton R. et Arai T. Kinematic optimization of a chopsticks-type micromanipulator, 1991.

[33] Warnaar D.B. et Chew M. Kinematic synthesis of deployable-foldable truss structures using graph theory, Part 2: generation of deployable truss module using design concepts. *ASME J. of Mechanical Design*, 117(1):117–122, Mars 1995.

[34] Zanganeh K.E. et Angeles J. On the isotropic design of general six-degree-of-freedom parallel manipulators. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 213–220. Kluwer, 1995.

[35] Zanganeh K.E. et Angeles J. Instantaneous kinematics and design of a novel redundant parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 3043–3048, San Diego, 8-13 Mai 1994.

Table des matières

1 Introduction **3**

2 Extremums des longueurs des segments **6**
2.1 Extremums pour un parallélépipède . . . 6
2.2 Extremums pour une sphère . . . 7
2.3 Extremums pour une région . . . 7
2.4 Temps de calcul . . . 8

3 Extremums des vitesses articulaires **8**
3.1 Extremums pour un segment . . . 9
3.2 Extremums pour un rectangle horizontal . . . 10
3.3 Extremums dans un parallélépipède rectangle . . . 11
3.3.1 L'approche par optimisation . . . 11
3.3.2 L'approche approximative . . . 11
3.3.2.1 Le principe . . . 11
3.3.2.2 Détermination des altitudes . . . 12
3.3.2.3 Temps de calcul . . . 13
3.4 Vitesses extrémales pour tout type d'espace de travail . . . 13
3.5 Vitesses extrémales sur l'espace complet . . . 15
3.6 Cas particulier . . . 15
3.7 Applications: détermination des angulations des segments . . . 17

4 Extremums des raideurs **17**
4.1 Extremums pour un segment . . . 18
4.2 Extremums pour des rectangles horizontaux . . . 19
4.3 Extremums pour des rectangles verticaux . . . 21
4.4 Extremums pour un parallélépipède rectangle . . . 21
4.5 Raideurs extrémales pour tout type d'espace de travail . . . 22
4.6 Raideurs extrémales sur l'espace complet . . . 23
4.7 Temps de calcul . . . 23

5 Extremums des forces articulaires **24**
5.1 Extremums pour un segment . . . 25
5.2 Extremums pour un rectangle . . . 26
5.2.1 Principe . . . 26
5.2.2 Réduction du temps de calcul . . . 27
5.2.3 Cas spécial . . . 27
5.3 Extremums dans une boîte . . . 27
5.3.1 Principe . . . 27
5.3.2 Réduction du temps de calcul . . . 28
5.3.3 Cas spécial . . . 28

5.4 Forces extrémales pour tout type d'espace de travail 29
5.5 Forces extrémales sur l'espace complet 30
5.6 Temps de calcul 30

6 Singularités **31**
6.1 Singularité sur un segment 34
6.2 Singularité dans un rectangle 34
6.3 Singularité dans un parallélépipède rectangle 35
6.3.1 Robot à base plane 35
6.3.2 Robot général 35
6.4 Singularité dans tout type d'espace de travail 35
6.5 Singularité dans l'espace complet 37
6.6 Robustesse numérique 37
6.7 Temps de calcul 37
6.8 Singularités pour les mouvements de rotation élémentaire 38
6.9 Singularités sur une courbe paramétrée 38

7 Extremum des erreurs de positionnement **39**

8 Conclusion **40**

9 Annexe **41**

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifi que,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399