Institut National Polytechnique de Grenoble

THESE
pour obtenir le grade de
DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

DISCIPLINE : INFORMATIQUE

présentée par
Kamel MEKHNACHA
le 16/07/1999

Méthodes Probabilistes Bayesiennes pour la prise en compte des Incertitudes Géométriques : Application à la CAO-Robotique.

Composition du jury :

M.	Claude PUECH	Président
	Etienne DOMBRE	Rapporteur
	Alain GIRAUD	Rapporteur
	Emmanuel MAZER	Directeur de thèse
	Alain RIVIÈRE	Examinateur
	Pierre BESSIÈRE	Co-directeur de thèse

Thèse préparée au sein du laboratoire Leibniz/IMAG

Table des matières

Présentation générale et plan de lecture

Introduction

Ce travail est le fruit d'une réflexion générale menée au sein de l'équipe Laplace du laboratoire Leibniz. Cette réflexion a comme objectif le développement d'un cadre formel permettant d'aborder le problème de l'incomplétude des modèles de représentation. La théorie bayesienne du raisonnement plausible a été choisie comme fondement théorique et la robotique comme un des domaines applicatifs.

La représentation et la gestion de l'incomplétude sont en effet des problèmes inévitables en robotique. La prise en compte de la dimension incertaine et incomplète du modèle du monde physique dans lequel les robots évoluent est une question fondamentale.

L'utilisation de modèles géométriques en robotique passe nécessairement par une modélisation plus ou moins réaliste du monde physique. La validité des calculs menés sur ces modèles dépend fortement de leur fidélité par rapport à la réalité, mais elle dépend aussi de leur capacité de représenter et d'exploiter les éventuels écarts entre le modèle et la réalité lors de la résolution d'un problème donné.

Le travail effectué dans le cadre de cette thèse porte sur cette problématique : comment représenter et exploiter des modèles géométriques incertains en robotique ?

Nous répondons à cette question en défendant la thèse suivante : le formalisme bayesien est une méthode de modélisation et d'inférence permettant de prendre en compte les incertitudes géométriques. Il permet de résoudre de nombreux problèmes géométriques trouvés en CAO robotique.

Nous proposons une méthode de spécification de problèmes géométriques prenant en compte les incertitudes. Nous établissons les mécanismes permettant de passer de ces problèmes à des problèmes d'inférence bayesienne et présentons les méthodes numériques permettant de conduire ces inférences. Nous montrons que les difficultés liées au calcul bayesien peuvent être surmontées en pratique en fournissant des solutions à de nombreux problèmes non triviaux.

Dans ce cadre, nous avons développé un système prototype de CAO robotique permettant de modéliser et de résoudre de nombreux problèmes géométriques en présence d'incertitude. Ce système a pour objectif d'illustrer les aspects les plus importants de notre méthode à savoir : la généralisation de la notion de contrainte géométrique pour la spécification de problèmes et le couplage de plusieurs méthodes numériques : intégration, optimisation et relaxation, pour leur résolution.

Plan de lecture

Ce document se compose de cinq chapitres et de trois annexes.

Le chapitre 1 présente en détail quelques travaux représentatifs concernant la gestion des incertitudes géométriques et la spécification de tâches par contraintes géométriques dans les systèmes de CAO robotique.

Dans le second chapitre, nous exposons notre méthode de spécification des problèmes géométriques. Nous commençons par présenter un ensemble de notions et de définitions que nous considérons utiles pour la modélisation géométrique 3D indépendamment du problème des incertitudes. Nous rappelons ensuite les principes du calcul bayesien, puis nous reprenons la notion géométrique de position relative en la décrivant non pas par un ensemble de scalaires, mais par une distribution de probabilité. Nous montrons enfin comment passer d'un problème géométrique à un double problème d'intégration et d'optimisation.

Le troisième chapitre décrit la méthode générique permettant de résoudre ces problèmes. Il présente la méthode de résolution numérique développée pour aborder les problèmes d'intégration et d'optimisation. Cette méthode permet de trouver la configuration du système satisfaisant au mieux les contraintes du problème. Elle est basée sur une méthode de Monte-Carlo pour l'intégration et sur un algorithme génétique pour l'optimisation. L'originalité de notre méthode est de contrôler le procédé d'intégration par le procédé d'optimisation. Le principe est de réduire le temps de calcul en commençant le procédé d'optimisation avec des estimations grossières des intégrales, puis d'affiner cette approximation à mesure que l'on se rapproche de l'optimum.

Le chapitre 4 expose l'application des concepts présentés à la résolution de problèmes robotiques concrets. Nous montrons la souplesse et la généralité de l'approche proposée en abordant des problèmes de diverses natures : l'inversion géométrique sous incertitudes pour des mécanismes éventuellement redondants, l'étalonnage de la position et de la forme d'objets à partir de mesures capteur et le positionnement de robots. Nous décrivons pour chaque application la modélisation correspondante, nous discutons des résultats obtenus et de l'apport de notre approche pour une telle application.

Enfin, le dernier chapitre présente une description du système de CAO dont les principes ont été précédemment décrits. L'objectif est de donner un bref aperçu de ses principaux modules, de ses fonctionnalités et de son implantation informatique.

Dans l'annexe A, nous présentons les primitives géométriques implémentées dans le système ainsi que les formules des contraintes égalités et inégalités décrivant les différents types de contact. L'annexe B regroupe des méthodes de tirage indirect de Monte-Carlo. Elle contient également des méthodes de tirage direct à partir des distributions gaussiennes. L'annexe C présente les détails de l'implantation de la méthode d'intégration numérique que nous avons développée.

Chapitre 1

Contexte

L'utilisation d'un modèle géométrique pour représenter un environnement réel (objets et mécanismes articulés) présente un intérêt indiscutable pour la simulation et la planification automatique de tâches robotiques. Toutefois, la validité de ces représentations ne peut jamais être considérée comme absolue et de nombreuses approches (dont la nôtre) adoptent une démarche visant à exprimer explicitement la dimension incertaine de ces modèles.

L'utilisation de modèles d'incertitude en robotique est envisagée de multiple manières. Citons par exemple la modélisation probabiliste de l'environnement avec des approches de type "uncertainty grid" [Moravec88] ou l'utilisation de modèles de déplacement incertain pour la planification de mouvements [LozanoPèrez87, DeLaRos96, Bouilly96, Alami94]. La notion d'incomplétude et d'incertitude des modèles peut aussi faire l'objet de travaux plus fondamentaux où l'incertitude est vue comme la conséquence fondamentale et irrémédiable de l'incomplétude des modèles formels [Bessière98a, Bessière98b, Dedieu95, Lebeltel97].

Nous nous limiterons à la description de travaux plus spécifiquement liés à la programmation automatique des robots. Notre objectif est d'illustrer notre problématique en la situant dans ce domaine d'application. Plus particulièrement, nous nous intéresserons aux modules de spécification géométrique utilisés dans les systèmes de programmation dits de "niveau objet".

Après une présentation succincte de la problématique de la programmation automatique des robots par contraintes géométriques, nous avons choisi de faire une présentation détaillée de trois approches, chacune représentative d'un type particulier de démarche.

Nous commencerons par les travaux de Taylor [Taylor76] qui présentent, en plus de leur intérêt scientifique, un intérêt historique comme étant l'une des premières tentatives de réalisation d'un système de programmation de niveau objet faisant intervenir les incertitudes dans le processus de spécification de but. Ce travail présente un bon exemple de prise en compte des incertitudes par propagation de contraintes.

Puis nous examinerons le travail de Puget [Puget89] portant sur une méthode de propagation des incertitudes dans un but de vérification/correction de programmes. Ce travail montre l'intérêt et les limites des hypothèses gaussiennes.

Enfin, nous nous intéresserons aux travaux de Piccin [Piccin95] portant sur un système de programmation par spécification de contraintes géométriques dans l'espace de travail. Notre intérêt est ici de montrer une méthode basée sur le calcul des variations afin de souligner l'alternative bayesienne que nous proposons.

1.1 Programmation de niveau objet

La programmation de niveau objet [Mazer87] est caractérisée par une description symbolique de la tâche en termes de relations géométriques à réaliser. Cette description symbolique est associée à un modèle géométrique de l'environnement. Ce modèle doit inclure une représentation morphologique (formes) et spatiale (positions relatives) des objets et des mécanismes articulés de la scène. Une représentation des incertitudes peut être également incluse [Troccaz86, Taylor76]. L'intérêt majeur de ce type de programmation est de faciliter la tâche du programmeur en prenant en charge les problèmes liés à la recherche de stratégie de déplacement permettant la réalisation des relations géométriques requises. Plusieurs modules sont nécessaires à l'interprétation d'une telle spécification, citons :

- L'interprétation des contraintes géométriques.
- La planification automatique de trajectoires [Latombe91, Boissonnat88, Faver88, Ahuactzin94].
- La synthèse de mouvements référencés capteurs [Samson91].
- La planification de mouvement de saisie [Mason85, Murray94]...

Nous nous limitons ici à certains aspects de l'interprétation des contraintes géométriques.

Le système de programmation RAPT [Wesley80] est le premier système proposant une méthode d'interprétation de la position d'un assemblage à partir d'un ensemble de contraintes géométriques. L'interpréteur de RAPT est basé principalement sur un système de réécriture inspiré du calcul formel. Dans certains cas très simples, le calcul de la solution peut être analytique [Celaya92]. Toutefois, dans le cas général où plusieurs contraintes doivent être satisfaites simultanément avec des mécanismes à plusieurs degrés de liberté, une résolution numérique des contraintes paraît plus appropriée. Nous citons dans ce contexte le système de résolution de AUTOPASS [Lieberman77], celui de CATIA de la société Dassault-Systèmes [Borrel85] permettant la spécification par contraintes de contact entre des primitives géométriques de base (point, droite, plan), ainsi que des travaux de A.Giraud [Giraud91].

1.2 Travaux de Taylor

Dans ses travaux de PhD, Taylor [Taylor76] présente un système de programmation de niveau objet basé sur la propagation des contraintes. À partir d'une spécification de niveau objet, le système procède à la génération automatique d'un programme de niveau effecteur (AL) correspondant. Il choisit la stratégie d'exécution la mieux adaptée compte tenu des incertitudes géométriques sur l'espace de travail. Ce système permet en particulier la représentation des incertitudes géométriques exprimées comme des contraintes, et dispose de méthodes de calcul capables de propager ces incertitudes pour une tâche donnée.

Nous présentons dans ce paragraphe les aspects les plus importants de ce système concernant la modélisation de l'environnement, la représentation et la propagation des incertitudes.

1.2.1 Modélisation de l'environnement

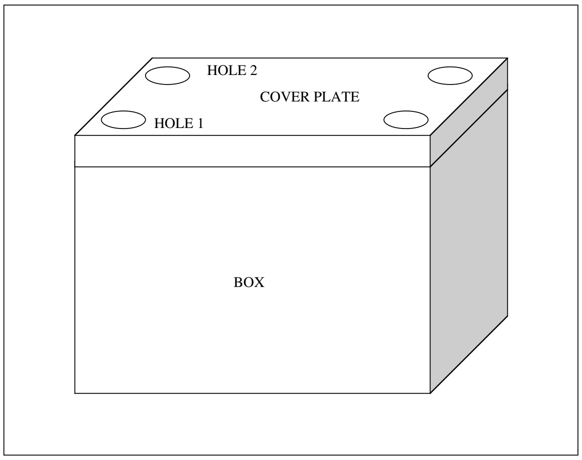

FIG. 1.1 - *Exemple d'objet composé (tiré de [Taylor76]).*

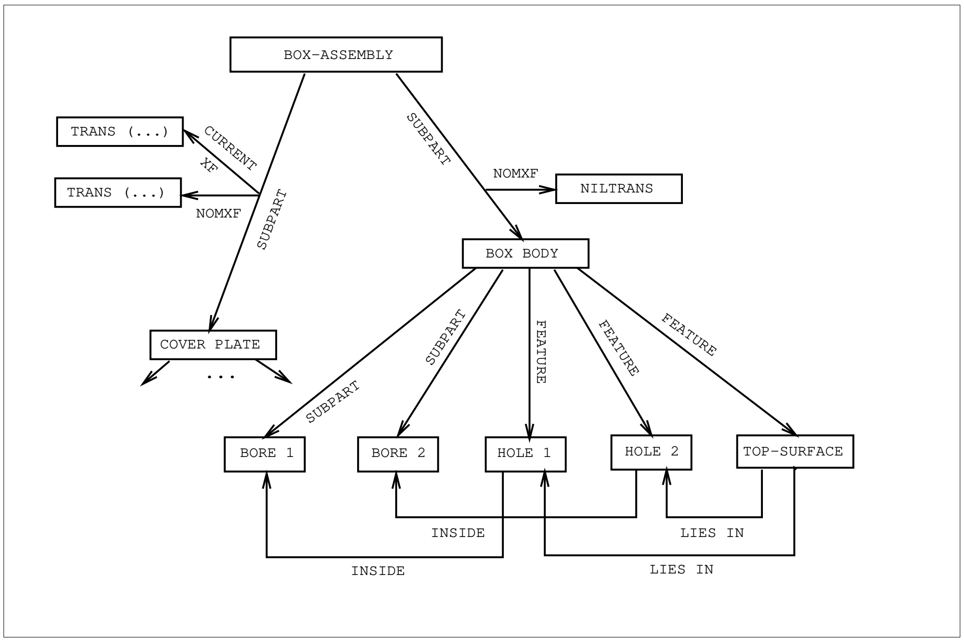

FIG. 1.2 - *Modèle de la boîte de la figure 1.1 (tiré de [Taylor76]).*

Taylor utilise les notions d'*objet* et *sous-objet* pour représenter la structure de la scène ainsi que les formes des objets. Cette modélisation est basée sur une représentation sous forme de graphe. Les nœuds de ce graphe représentent les objets et les sous-objets tandis que les arcs reliant ces nœuds décrivent les relations entre ces derniers.

Un exemple de cette modélisation tiré de [Taylor76] est donné par les figures 1.1 et 1.2. La figure 1.1 représente un exemple d'assemblage tandis que la figure 1.2 donne le graphe permettant de modéliser cet assemblage en spécifiant les types des relations géométriques qui existent entre ses constituants.

Lors de la spécification, les relations géométriques sont exprimées comme des expressions symboliques. Ces expressions seront traduites par le système pour générer des **contraintes** impliquant les paramètres fixes des positions entre les sous-objets, et des scalaires libres représentant les degrés de liberté de ces relations. Une expression $(rel, A, B, \cdots)$ représentant une relation géométrique entre deux constituants A et B, sera alors traduite en un ensemble de contraintes sur des paramètres $\lambda_1, \cdots, \lambda_n$ de la transformation $T_{AB}(\lambda_1, \cdots, \lambda_n)$. Ces contraintes peuvent être :

- des *contraintes d'égalité (de contact)*

$$F(\lambda_1, \cdots, \lambda_n) = 0$$

- des *contraintes d'inégalité*

$$C(\lambda_1, \cdots, \lambda_n) \geq 0$$

À noter que les paramètres $\lambda_1, \cdots, \lambda_n$ doivent être :

- facilement reliés aux paramètres de translation et de rotation exprimés dans un repère connu,
- liés par des contraintes dont la forme est suffisamment simple pour être utilisée par le système.

1.2.2 Utilisation des relations géométriques de contact

Supposons qu'une entité F_B d'un objet B est en relation d'appartenance avec une entité F_A d'un autre objet A. À partir de l'expression symbolique F_B *sur* F_A, le système procède comme suit :

- L'ensemble de points de l'entité F_B est exprimé comme une *expression génératrice*

$$\begin{cases} F_B = \{v \mid v = f_B(\mu_1, \cdots, \mu_n)\} \\ C_B(\mu_1, \cdots, \mu_n) \geq 0 \end{cases}$$

  où $\mu_1, \cdots, \mu_n$ sont des scalaires libres caractérisant l'entité F_B exprimée dans le système de coordonnées de l'objet B.

- L'entité F_A sera exprimée, quant à elle, en utilisant une *expression de contrainte* :

$$F_A = \{u \mid G_A(u) = 0 \ \ et \ \ C_A(u) \geq 0\}$$

– Comme tous les points de F_B sont sur F_A, on obtient le système :

$$\begin{cases} G_A(T_{AB} * f_B(\mu_1, \cdots, \mu_n)) = 0 \\ C_A(T_{AB} * f_B(\mu_1, \cdots, \mu_n)) \geq 0 \\ C_B(\mu_1^B, \cdots, \mu_n) \geq 0 \end{cases}$$

où T_{AB} est la transformation géométrique entre les objets A et B.

La subtitution des paramètres μ_i permet d'obtenir des contraintes sur les paramètres de la transformation T_{AB}.

Exemple

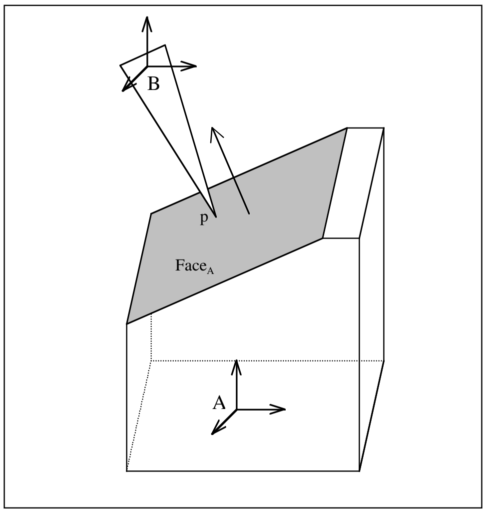

FIG. 1.3 - *Exemple de relation* point sur face.

Prenons l'exemple présenté par la figure 1.3, représentant une relation de point sur face :

Supposons que la face $Face_A$ appartenant à l'objet A est donnée par un ensemble de points :

$$Face_A = \{v \mid n \cdot v = d \;\; et \;\; G\, v \geq b\}$$

où n est le vecteur normal à $Face_A$ exprimé dans le repère de l'objet A, d et b sont des scalaires et G une matrice. L'opérateur (.) représente le produit scalaire.

L'équation $n \cdot v = d$ permet d'exprimer que le point v est sur le plan support de la face $Face_A$ tandis que l'inégalité $Gv \geq b$ représente le fait que le v est à l'intérieur de la région limitée par les bords de la face.

Supposons maintenant que le point p appartenant à un autre objet B est sur la face $Face_A$. Cela permet d'obtenir le système de contraintes suivant sur les paramètres de la transformation T_{AB} :

$$\begin{cases} n \cdot R\, p = d - n \cdot t \\ G * R\, p + G\, t \geq b \end{cases}$$

avec

$$T_{AB} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

1.2.3 Propagation des incertitudes

Le calcul de la propagation des incertitudes dans ce système est fait par la propagation de toutes les contraintes égalités $F_j(\lambda_1^j, \cdots \lambda_n^j)$ et les contraintes inégalités $C_i(\lambda_1^i, \cdots \lambda_n^i)$.

Supposons qu'on s'intéresse à calculer la position nominale T_{AB} et/ou la précision ΔT_{AB} d'un objet A par rapport à un autre objet B. ΔT_{AB} représente la transformation exprimant l'erreur de position telle que :

$$T_{AB}^{eff} = T_{AB} * \Delta T_{AB}$$

Une position T_{AB} connue sans incertitude aura une matrice d'erreur $\Delta T_{AB} = I_4$ où I_4 est la matrice identité 4×4.

L'objectif est de générer un ensemble de valeurs δ_k correspondant à la position (position nominal et/ou erreur de position) entre A et B. En supposant que les δ_k sont suffisamment petits, un système de contraintes linéaires $M\delta$? b (où ? représente des égalités et inégalités) est trouvé.

L'algorithme utilisé par Taylor consiste à :

1. Trouver toutes les chaînes de relations sans cycles (chemins) reliant A et B, et écrire pour chaque chaîne (chemin)

   $$(A, N_1, \cdots, N_k, B)$$

   l'équation

   $$T_{AB} = T_{AN_1} * \cdots * T_{N_k B}$$

   Pour chaque transformation $T_{N_i N_{i+1}}$ est associé un ensemble de paramètres $\lambda_{N_i N_{i+1}}$. Sur ces paramètres, on peut avoir des contraintes égalités et inégalités. transformation.

   Si on s'intéresse à l'erreur ΔT_{AB}, on aura en plus l'équation

   $$\Delta T_{AB} = T_{N_k B}^{-1} * \cdots T_{AN_1}^{-1} * T_{AN_1} * \Delta T_{AN_1} * \cdots * T_{N_k B} * \Delta T_{N_k B}$$

   Cette équation est obtenue en utilisant plusieurs fois la propriété suivante :

   Si deux transformations T_{ij} et T_{jk} représentent respectivement les positions entre les repères $\Re_i$, $\Re_j$ et entre les repères $\Re_j$, $\Re_k$, et si les erreurs qui leur sont associées sont respectivement ΔT_{ij} et ΔT_{jk}, alors la position nominale T_{ik} du repère $\Re_k$ par rapport à $\Re_i$ est donnée par :

   $$T_{ik} = T_{ij} * T_{jk}$$

   Pour calculer l'erreur ΔT_{ik} sur T_{ik} on a :

   $$\begin{aligned} T_{ik}^{eff} &= T_{ik} * \Delta T_{ik} && (1.1) \\ &= T_{ij} * T_{jk} * \Delta T_{ik} && (1.2) \end{aligned}$$

D'autre part on a :

$$T_{ik}^{eff} = T_{ij}^{eff} * T_{jk}^{eff} \quad (1.3)$$
$$= T_{ij} * \Delta T_{ij} * T_{jk} * \Delta T_{jk} \quad (1.4)$$

Ce qui donne finalement :

$$\Delta T_{ik} = T_{jk}^{-1} * \Delta T_{ij} * T_{jk} * \Delta T_{jk}$$

2. Une fois toutes les équations correspondant à tous les chemins ont été trouvées et simplifiées, Linéariser les contraintes obtenues à l'issue de l'étape précédente en supposant que les valeurs des paramètres des transformations $T_{N_iN_{i+1}}$ sont suffisamment proches des valeurs des paramètres des transformations nominales $T^0_{N_iN_{i+1}}$.

3. Utiliser les techniques de la programmation linéaire pour résoudre le système linéaire obtenu à la fin de l'étape précédente.

1.2.4 Discussion

Notre principale remarque concerne la représentation des incertitudes par des contraintes d'inégalités ou intervalles. Cela conduit à un effet ensembliste lors de la propagation de ces incertitudes. Autrement dit l'estimation la plus pessimiste va être considérée. C'est particulièrement le cas des références multiples (cas des cycles) où plusieurs chemins existent entre deux repères pour lesquels on veut estimer l'incertitude sur leur position relative.

Ce choix conduit à l'impossibilité de prendre en compte des a priori non uniformes sur un intervalle ou bien une région. En effet, la représentation des incertitudes par des contraintes inégalités sur les degrés de liberté d'une transformation est équivalente à avoir un a priori uniforme sur la région qui correspond à ces inégalités.

1.3 Travaux de Puget

Puget [Puget89] aborde le problème de la représentation et de la propagation des incertitudes géométriques dans le but de la vérification-correction de programmes. Étant donné un état initial I représentant les connaissances sur les positions des objets dans l'espace de travail et sur les incertitudes associées à ces positions, la vérification d'un programme P consiste à s'assurer de sa validité vis-à-vis des incertitudes dans le modèle (respect des contraintes de précision imposées par le problème). La correction d'un programme, quant à elle, consiste à modifier ce dernier pour le rendre valide.

L'objet de ce paragraphe est de présenter les aspects de ce travail les plus intéressants pour nous, à savoir : la modélisation de l'environnement, la représentation des incertitudes, et le calcul de la propagation de ces incertitudes.

1.3.1 Modélisation de l'environnement

La modélisation de l'environnement est basée sur la représentation des objets sous forme de graphe appelé *Graphe d'état*. Les nœuds de ce graphe représentent les repères attachés aux objets, et les arcs reliant ces nœuds représentent les valeurs nominales des positions relatives entre les différents objets et les incertitudes associées à ces positions. La position relative entre deux repères $\Re_i$ et $\Re_j$ est représentée par une matrice de transformation (4×4) T_{ij}.

Les figures 1.4 et 1.5 représentent respectivement un exemple d'un environnement de travail (un manipulateur et des objets), et le graphe d'état correspondant.

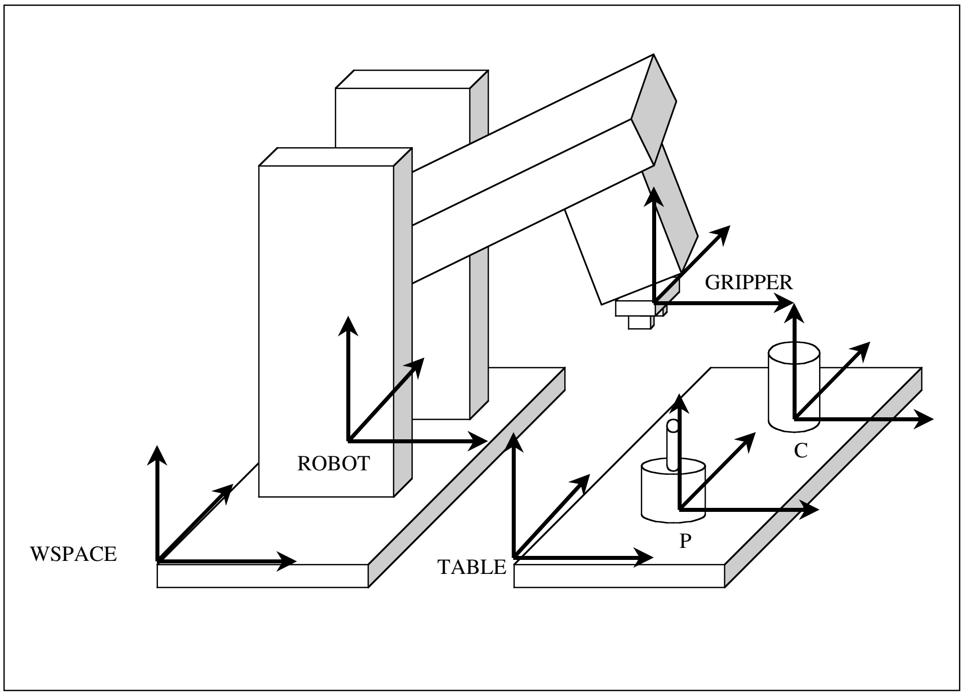

FIG. 1.4 - *Exemple d'environnement (tiré de [Puget89]).*

1.3.2 Représentation des incertitudes

Puget utilise une notion d'erreur de position. Pour une position théorique T_{12} d'un repère $\Re_2$ dans un autre repère $\Re_1$, la position réelle T'_{12} diffère de cette position théorique par une erreur de position ϵ (exprimée par définition dans le repère $\Re_2$) telle que :

$$T'_{12} = T_{12} * \epsilon$$

À cette erreur est associé un 6-vecteur aléatoire $\vec{\epsilon}$ dont les trois premières coordonnées τ représentent l'erreur de translation, et les trois dernières ω représentent l'erreur en rotation :

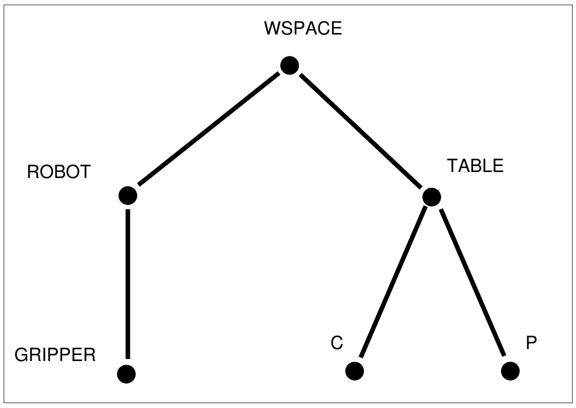

FIG. 1.5 - *Graphe d'état correspondant à l'exemple donné par la figure 1.4.*

$$\vec{\epsilon} = \begin{pmatrix} \tau \\ \omega \end{pmatrix}$$

La relation entre la matrice de transformation $\epsilon = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ et $\vec{\epsilon} = (\tau\ \omega)^T$ est :

$$\begin{aligned} t &= \tau && (1.5) \\ R &= \cos(a)I_3 + (1-\cos(a))uu^T - \sin(a)\{u\} && (1.6) \end{aligned}$$

avec $a = norm(\omega)$, $u = \frac{\omega}{a}$ et

$$\{u\} = \begin{pmatrix} 0 & u3 & -u2 \\ -u3 & 0 & u1 \\ u2 & -u1 & 0 \end{pmatrix}$$

Le choix qui a été fait est de considérer $\vec{\epsilon}$ comme un vecteur aléatoire distribué suivant une loi **gaussienne** (normale) de moyenne nulle et de matrice de covariance (matrice 6×6) Δ :

$$\forall i, j \in [1, 6], \Delta_{ij} = E(\epsilon_i \epsilon_j)$$

où E est l'espérance mathématique et ϵ_i est la i^{me} coordonnée de $\vec{\epsilon}$.

Cette matrice de covariance Δ_{AB} associée à une transformation T_{AB} représentant la position relative entre deux repères A et B, quantifie l'incertitude sur cette position. Une position connue sans incertitude aura une matrice de covariance $\Delta = 0$.

1.3.3 Propagation des incertitudes

Étant donné un graphe d'état G représentant la scène, le calcul de la position T_{AB} d'un repère $\Re_B$ par rapport à un autre repère $\Re_B$ et l'incertitude Δ_{AB} associée à cette position consiste à :

1. trouver un chemin $way(\Re_i, \Re_j)$ reliant $\Re_i$ à $\Re_j$ à partir du graphe G,
2. composer les positions et les incertitudes le long de ce chemin.

Combinaison de deux chemins

Soit deux chemins successifs $way(\Re_i, \Re_j)$ et $way(\Re_j, \Re_k)$, associés respectivement aux positions T_{ij}, T_{jk} et aux incertitudes Δ_{ij}, Δ_{jk}. La position nominale T_{ik} du repère $\Re_k$ par rapport à $\Re_i$ est donnée par :

$$T_{ik} = T_{ij} * T_{jk}$$

Pour calculer l'erreur ϵ_{ik} sur T_{ik} on a :

$$\begin{aligned} T'_{ik} &= T_{ik} * \epsilon_{ik} && (1.7) \\ &= T_{ij} * T_{jk} * \epsilon_{ik} && (1.8) \end{aligned}$$

D'autre part on a :

$$\begin{aligned} T'_{ik} &= T'_{ij} * T'_{jk} && (1.9) \\ &= T_{ij} * \epsilon_{ij} * T_{jk} * \epsilon_{jk} && (1.10) \end{aligned}$$

Ce qui donne finalement :

$$\epsilon_{ik} = T_{jk}^{-1} * \epsilon_{ij} * T_{jk} * \epsilon_{jk}$$

Pour calculer la matrice de covariance Δ_{ik} correspondant à cette erreur ϵ_{ik}, Puget utilise une approximation au premier ordre (linéarisation) en utilisant les propriétés suivantes :

1. Soit ϵ_1 et ϵ_2 deux matrices d'erreur, $\vec{\epsilon}_1$ et $\vec{\epsilon}_2$ les 6-vecteurs qui leur sont associés. Le 6-vecteur $\vec{\epsilon'}$ associé à la matrice $\epsilon' = \epsilon_1 * \epsilon_2$ est égal au premier ordre à $\vec{\epsilon}_1 + \vec{\epsilon}_2$.
2. Si Δ_1 et Δ_2 sont les matrices de covariance associées respectivement à $\vec{\epsilon}_1$ et $\vec{\epsilon}_2$, alors la matrice de covariance associée à $\vec{\epsilon'} = \vec{\epsilon}_1 + \vec{\epsilon}_2$ est $\Delta' = \Delta_1 + \Delta_2$.
3. Soit ϵ une matrice d'erreur, $\vec{\epsilon} = (\tau\omega)^T$ le 6-vecteur qui lui est associé et T une transformation de matrice de rotation r et de translation $t = (t_x t_y t_z)^T$. Au premier ordre, le 6-vecteur $\vec{\epsilon'} = (\tau'\omega')^T$ associé à la transformation $\epsilon' = T^{-1} * \epsilon * T$ est :

   $$\vec{\epsilon'} \simeq M\vec{\epsilon}$$

   avec

   $$M = \begin{pmatrix} r^{-1} & r^{-1}\mathcal{M}_t \\ 0 & r^{-1} \end{pmatrix}$$

   et

   $$\mathcal{M}_t = \begin{pmatrix} 0 & t_z & -t_y \\ -t_z & 0 & t_x \\ t_y & -t_x & 0 \end{pmatrix}$$

4. Si Δ est la matrice de covariance associée à un vecteur $\vec{v}$, la matrice associée à $\vec{v}' = M\vec{v}$ est $\Delta' = M\Delta M^T$

En utilisant successivement ces propriétés, la matrice de covariance Δ_{ik} associée à ϵ_{ik} sera donnée par :

$$\Delta_{ik} = M_{jk}\Delta_{ij}M_{jk}^t + \Delta_{jk}$$

Combinaison des arcs d'un chemin

Étant donné un chemin $way(\Re_i, \Re_j)$

$$way(\Re_i, \Re_j) = \Re_i Arc(\Re_i, \Re_1)\Re_1 Arc(\Re_1, \Re_2)\Re_2 \cdots \Re_n Arc(\Re_n, \Re_j)\Re_j$$

la combinaison des $(n+1)$ arcs constituant ce chemin se fait récursivement en n décompositions en utilisant le résultat de la section 1.3.3 :

– premier pas :

$$\begin{aligned} T_{ij} &= T_{i1} * T_{1j} \\ \Delta_{ij} &= M_{1j}\Delta_{i1}M_{1j}^t + \Delta_{1j} \end{aligned}$$

– second pas :

$$\begin{aligned} T_{1j} &= T_{12} * T_{2j} \\ \Delta_{1j} &= M_{2j}\Delta_{12}M_{2j}^t + \Delta_{2j} \end{aligned}$$

$\vdots$

– dernier pas :

$$\begin{aligned} T_{(n-1)j} &= T_{(n-1)n} * T_{nj} \\ \Delta_{(n-1)j} &= M_{nj}\Delta_{(n-1)n}M_{nj}^t + \Delta_{nj} \end{aligned}$$

1.3.4 Présence des cycles et problème de la cohérence du modèle

La présence de cycles dans le graphe d'état pose un problème de cohérence du modèle. La présence d'un cycle signifie que la position d'un repère (différent de WSPACE) est définie par rapport à plusieurs autres repères (existence de plusieurs chemins).

Le choix fait pour le maintien de la cohérence du modèle est d'interdire la création de cycles. Si l'ajout d'un nouvel arc (représentant une nouvelle mesure) entre deux repères $\Re_1$ et $\Re_2$ engendre la création d'un cycle, cet arc n'est pas retenu et l'information apportée par celui-ci est reportée sur tous les arcs constituant le chemin $way(\Re_1, \Re_2)$ (changement des positions nominales et des matrices de covariance associées).

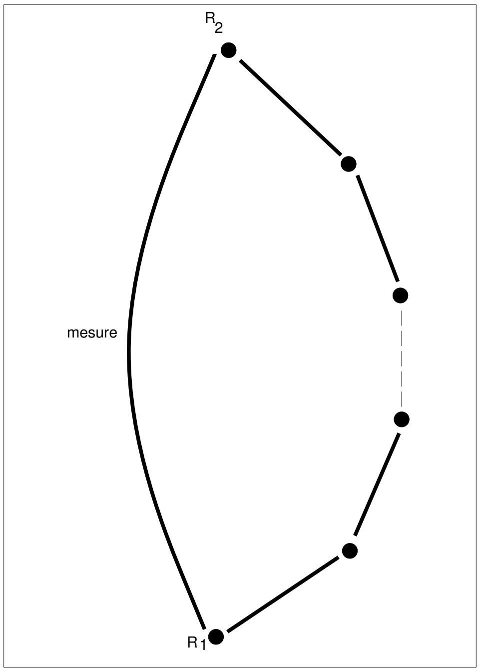

FIG. 1.6 - *Création d'un cycle lors de l'intégration d'une nouvelle mesure.*

Intégration d'une mesure

La mise à jour des n arcs de ce chemin utilise un filtre de Kalman [Lowe89, Zhang92] après linéarisation des formules.

Si l'on dispose d'une mesure de la position relative des repères $\Re_1$ et $\Re_2$ déjà reliés par un chemin $way(\Re_1, \Re_2)$ (Figure 1.6) constitué de n arcs et en notant :

- $\overline{T}_{12}$ la valeur nominale de la transformation mesurée, et $\overline{\Delta}_{12}$ la matrice de covariance associée,
- pour tout arc $Arc(\Re_k, \Re_l)$ du chemin, $\Delta'_{kl} = M_{l1} * \Delta_{kl} * M_{l1}^t$
- vec la fonction qui associe le 6-vecteur à une transformation

alors les n arcs du chemin seront mis à jour de la manière suivante :

- la nouvelle valeur T^*_{ij} d'une transformation T_{ij} est donnée par :

$$T^*_{ij} = T_{ij} * T_{1i}^{-1} * vec^{-1}\left(\Delta'_{ij}\left(\sum_{way(\Re_1,\Re_2)} \Delta'_{kl} + \overline{\Delta}_{12}\right)^{-1} vec\left(\overline{T}_{12} * T_{12}^{-1}\right)\right) * T_{1i}$$

– et la nouvelle matrice de covariance associée par :

$$\Delta_{ij}^{*} = M_{j1}^{-1}\left(I - \Delta'_{ij}\left(\sum_{way(\Re_1,\Re_2)} \Delta'_{kl} + \overline{\Delta}_{12}\right)^{-1}\right)\Delta'_{ij}\left(M_{j1}^{-1}\right)^{t}$$

Utilisation des contacts

Les relations géométriques de type contact entre deux objets permettent de réduire les variances sur un sous-ensemble de composantes du 6-vecteur d'erreur $\vec{\epsilon}$ de position entre ces objets.

Les contraintes égalités appliquées aux degrés de liberté et induites par ces contacts sont utilisées comme équations de mesure pour un filtre de Kalman (ou bien un filtre de Kalman étendu pour les contraintes non linéaires). Le vecteur d'état de ce filtre est le 6-vecteur d'erreur $\vec{\epsilon}$ sur la position relative des objets en contact. Cela permet d'obtenir une matrice de covariance Δ^* *plus petite* que la matrice de covariance initiale Δ.

1.3.5 Discussion

L'utilisation exclusive des distributions gaussiennes pour la représentation des incertitudes présente l'avantage majeur de la commodité des calculs nécessaires pour leurs manipulations. Toutefois on peut adresser les critiques suivantes :

– Le calcul de la propagation des incertitudes fait systématiquement appel à une linéarisation des formules. Cette linéarisation suppose que les erreurs sur les orientations sont suffisamment petites pour que cette approximation reste valide. Puget a montré que l'effet de ces linéarisations n'est pas important dans les approximations élémentaires qu'il utilise, cependant l'erreur engendrée par l'utilisation répétée de ces approximations le long d'un chemin n'a pas été quantifiée.

– L'information donnée par les contraintes inégalités induites par les contacts géométriques n'est pas exploitée. C'est le cas, par exemple, du contact *face sur face* qui stipule que tous les points appartenant à la première face ne dépassent pas les bords de la deuxième face. Cette information peut considérablement réduire les incertitudes sur les positions relatives.

– L'interdiction de la création et de l'utilisation des cycles dans le modèle de l'environnement engendre une importante perte d'information. Cette perte s'explique par la surestimation des incertitudes finales comme cela est expliqué au chapitre 6 du mémoire [Puget89], et surtout par l'irréversibilité du processus (impossibilité de la suppression d'une mesure qu'on a intégrée).

– L'impossibilité de formuler des a priori non gaussiens sur les paramètres des modèles.

1.4 Travaux de Piccin

Piccin [Piccin95] présente un système de spécification par contraintes de tâches de manipulation. Ces travaux s'intéressent en particulier aux problèmes de génération de trajectoires dans des espaces de travail imparfaitement connus, et proposent des méthodes permettant d'affiner le modèle de l'environnement en utilisant des mesures pour identifier des paramètres de positions et de tailles pour les objets présents dans cet environnement.

Nous présentons dans cette partie les principaux concepts et méthodes utilisés pour la spécification et la résolution de problèmes.

1.4.1 Modélisation des mécanismes

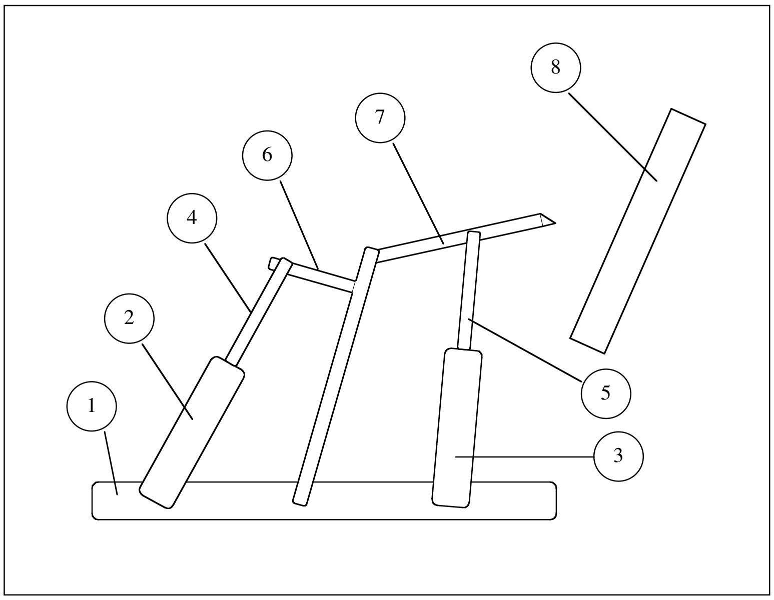

FIG. 1.7 - *Mécanisme à deux degrés de liberté (tiré de [Piccin95]).*

Un mécanisme est représenté par un graphe appelé *graphe des liaisons cinématiques*. Les sommets de ce graphe correspondent à des référentiels appartenant aux corps du mécanisme tandis que les arcs décrivent les situations (positions) et les mobilités relatives de ces référentiels.

Concept de lien et mobilité relative

Un lien représentant un arc du graphe des liaisons définit la position relative entre deux référentiels ainsi que la nature de la mobilité entre eux. La position relative entre deux référentiels est définie par un *vecteur de configuration* de $\Re^6$ $Q = [tx\ ty\ tz\ rx\ ry\ rz]^T$. Les trois premières composantes représentent le vecteur de translation entre les deux référentiels et les trois dernières représentent le vecteur de rotation (sa direction définit l'axe de rotation et son module donne l'angle de rotation autour de cet axe). L'information sur la mobilité relative entre les deux référentiels est donnée par un 6-vecteur d'*étiquettes*. Chaque composante de ce vecteur d'étiquettes est un caractère alphabétique. Un exemple des valeurs de ces étiquettes est donné par le tableau 1.1.

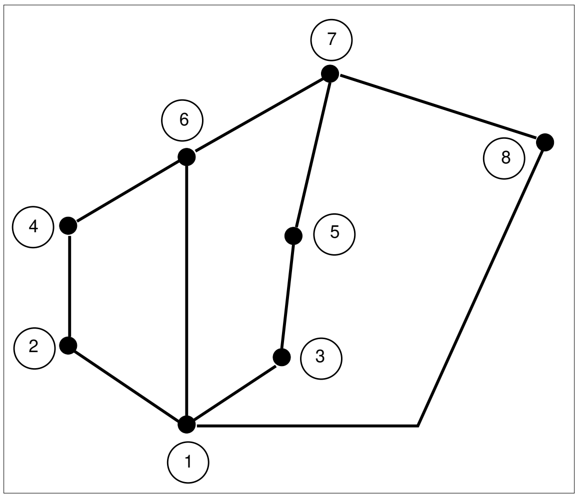

FIG. 1.8 - *Graphe du mécanisme (tiré de [Piccin95]).*

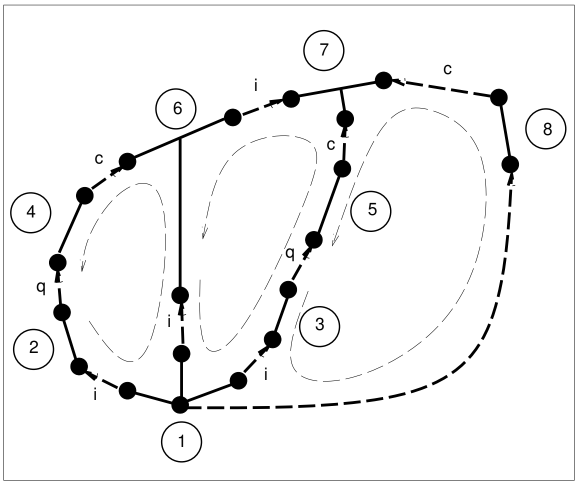

FIG. 1.9 - *Graphe étendu (tiré de [Piccin95]).*

Exemple

La figure 1.7 représente un exemple de modélisation d'un bras articulé à deux degrés de libertés animé par deux vérins. Ces derniers sont représentés par les deux couples de corps $(2, 4)$ et $(3, 5)$. Le graphe représentant ce mécanisme est donné dans la figure 1.8. La figure 1.9 représente le graphe étendu des liaisons cinématiques utilisé par le système pour modéliser ce mécanisme. Sur ce graphe, les arcs en trait continu représentent les liaisons fixes (liens fixes) et les arcs en trait pointillé décrivent les liaisons mobiles.

Étiquette	Interprétation
N, P	Paramètre nul et non nul
Q	Variable inconnue (articulaire)
C	Variable de contrainte (à annuler)
R	Variable intermédiaire (valeur indifférente)

TAB. 1.1 - *Exemple d'étiquettes de variables (tiré de [Piccin95]).*

1.4.2 Extraction des contraintes

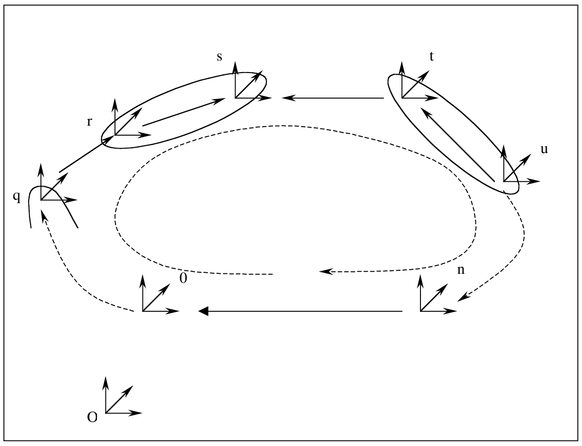

FIG. 1.10 - *Cycle cinématique (tiré de [Piccin95]).*

L'écriture des conditions de fermeture des cycles du graphe des liaisons permet d'obtenir les contraintes décrivant un problème donné. Ces conditions de fermeture peuvent être exprimées soit en suivant un raisonnement géométrique sur les positions relatives des référentiels, soit par un raisonnement cinématique sur les vitesses relatives (dérivées des variables géométriques) entre ces derniers. La première option (raisonnement géométrique) engendre, dans le cas général, des termes non linéaires. En revanche, la deuxième option (raisonnement cinématique) permet d'obtenir des équations linéaires reliant les dérivées des variables géométriques, ce qui explique le choix qui a été fait dans ces travaux.

Prenons le cas de la figure 1.10; la condition de fermeture du cycle s'écrit en utilisant la loi de composition des vitesses (torseurs) des référentiels par rapport à un point de référence O comme suit :

$$\sum_{i=1}^{n-1} V^0_{i,i+1} + V^0_{n,1} = 0$$

En utilisant les relations de changement de point ainsi que la relation linéaire instantanée qui existe entre le torseur et la dérivée $\dot{Q}_{ij}$ du vecteur de configuration, la contrainte s'écrit finalement sous la forme :

$$\sum_i M_i \ \Delta Q_i = 0$$

où M_i est une matrice 6×6 dont les termes sont donnés dans [Piccin95], et ΔQ_i sont les vecteurs incrémentaux correspondant aux vecteurs de configurations Q_i.

L'ensemble des m cycles présents dans le graphe donne un système d'équations :

$$\left\{ \sum_i M_i \ \Delta Q_i = 0 \right\}_{j=1}^{m}$$

1.4.3 Résolution des contraintes

Dans le paragraphe précédent nous avons montré comment extraire les équations de contraintes décrivant des conditions de fermeture d'un cycle. La fermeture d'un cycle correspond en fait à la position finale que le manipulateur doit rejoindre. À ces équations peuvent s'ajouter, dans le cas général, des contraintes supplémentaires à satisfaire lors du déplacement vers l'objectif. C'est particulièrement le cas de la présence des obstacles à éviter lors du mouvement.

Le troisième type de problème est celui de la ation (étalonnage). À partir d'un ensemble de contacts entre le robot et un objet de son environnement, le système doit exploiter ces mesures pour identifier les paramètres de la position de l'objet en question. Ce problème revient à la minimisation d'une fonction.

Ces trois types de problème se formulent de la manière suivante :

$$\begin{cases} (a) & g(x) = 0 & \text{Objectif de position.} \\ (b) & g(x^*) = 0 & \text{Génération de trajectoire sous contraintes.} \\ & \text{avec } \ h(x_n) \geq 0, n \in [0, N] & \\ (c) & \min_x f^T(x) f(x) & \text{Problème d'identification paramétrique.} \\ & \text{avec } \ g(x) = 0 & \end{cases}$$

où g, f et h sont des fonctions, et $\{x_n\}$ représentent les positions intermédiaires vers l'objectif x^*.

La méthode de résolution utilisée par ce système est basée sur un mécanisme d'élimination de variables avec raisonnement numérique/formel pour la résolution des systèmes linéaires obtenus par l'analyse cinématique. Le résultat de ce traitement est utilisé par un schéma itératif de Newton-Raphson. La méthode proposée utilise une correction locale des directions de recherche pour assurer la satisfaction des éventuelles contraintes inégalités. Une méthode du simplexe est utilisée pour cette correction.

1.4.4 Discussion

Le système proposé par Piccin est dédié aux problèmes de planification dans des environnements imparfaitement connus. Ainsi, dans le cas de la téléprogrammation des robots distants, le modèle de l'environnement de travail ne peut être qu'approximatif. Pour aborder le problème de la validité du modèle, le système utilise un étalonnage des positions des objets présents dans l'environnement à partir des mesures obtenues par la mise en contact répétée d'un point du robot sur un point de l'objet en question.

Nous adressons deux remarques principales sur ce type d'approche :

- Elle ne donne pas la possibilité d'utiliser le modèle initial de l'environnement (sans étalonnage) en prenant en compte explicitement les incertitudes sur ce modèle. Il est préférable, à notre avis, d'avoir une notion d'incertitude représentant l'état de connaissance sur le modèle, et de prendre en compte ces incertitudes dans le processus de planification. Cela permet en particulier de n'avoir recours à l' étalonnageque lorsque le critère de précision qu'on s'est fixé sur l'exécution de la tâche à effectuer n'est pas respecté.
- Pour l étalonnage des positions des objets, l'utilisation d'une estimation de type moindres carrés (résultat des algorithmes de type Newton-Raphson pour les systèmes surdéterminés) à partir des *mesures* issues des contacts, ne permet pas la prise en compte des informations disponibles a priori ni sur les modèles des objets à étalonner, ni sur le modèle du robot utilisé comme instrument de mesure (incertitudes sur les paramètres du robot lui-même).

Le style de modélisation par spécification de contraintes sur les paramètres de pose utilisé dans ce travail nous parait bien adapté pour la modélisation de tâches robotiques. Ce style de modélisation a été retenu dans notre travail et sera développé dans le prochain chapitre.

1.5 Conclusion

Nous avons présenté dans ce chapitre la problématique de la prise en compte explicite des incertitudes géométriques dans les systèmes de programmation de "niveau objet". Nous avons fait une présentation détaillée de trois approches que nous avons jugées représentatives dans ce domaine. Pour chaque approche, nous avons souligné les avantages et les limites en termes de la richesse des informations prise en compte par l'approche, ainsi qu'en termes de performance et commodité des calculs requis par la méthode. Le but était, bien entendu, de dégager les principaux problèmes pour pouvoir les aborder par la méthode que nous proposons dans cette thèse.

Chapitre 2

Spécification de problèmes géométriques

Le but de ce chapitre est de présenter les notions et les principes de base utilisés pour la modélisation géométrique probabiliste. Il comporte quatre parties :

Nous commençons par la présentation d'un ensemble de notions et de définitions que nous considérons utiles pour la modélisation géométrique indépendamment de toutes références au calcul probabiliste. Cette modélisation est basée sur une représentation classique sous forme de graphe. Les sommets de ce graphe représentent les repères, tandis que les arcs représentent les informations sur les poses relatives entre ces repères. Nous introduisons aussi dans cette partie la notion de relation de contact entre entités géométriques standard (faces, arêtes, cylindres...), et les contraintes d'égalité et d'inégalité induites par ces relations.

Dans la deuxième partie, nous présentons le calcul bayesien comme un formalisme permettant d'expliciter et de manipuler les incertitudes. Nous introduisons les principes généraux de ce formalisme et la méthodologie qu'il utilise.

Dans la troisième partie, nous montrons comment nous étendons les modèles géométriques présentés dans la première partie, pour prendre en compte les incertitudes géométriques (imprécisions). Cette extension consiste essentiellement à représenter les poses relatives entre repères, non pas par des scalaires, mais par des distributions de probabilité sur les paramètres décrivant ces poses.

La quatrième partie, quant à elle, est consacrée à l'utilisation du formalisme bayesien précédemment exposé pour représenter et propager d'une manière uniforme les incertitudes et les contraintes géométriques.Nous montrerons en particulier comment passer de la spécification d'un problème quelconque de géométrie à un double problème d'intégration et d'optimisation.

2.1 Définitions de base et notations pour la modélisation géométrique

L'objectif de cette partie est de définir un ensemble de notions de base et de notations qui seront utilisées pour la modélisation géométrique dans le reste de ce document.

2.1.1 Pose relative

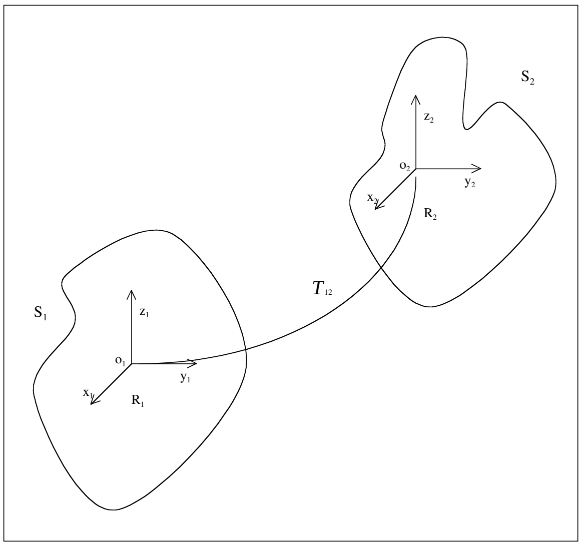

FIG. 2.1 - *Pose relative entre deux corps.*

Dans la pratique, une grande catégorie des problèmes géométriques peut être spécifiée en modélisant les objets et les systèmes articulés présents dans l'environnement comme des corps rigides, et en décrivant les situations (poses) et les mobilités relatives entre ces derniers.

Dans le cas général, la pose relative d'un objet (solide) S_1 (Figure 2.1) associé à un repère $\Re_1$ par rapport à un autre objet S_2 (associé au repère $\Re_2$) est donnée par 6 paramètres indépendants, représentant la transformation géométrique (3 paramètres de translation et 3 de rotation) transformant $\Re_1$ en $\Re_2$. Cette localisation relative dans l'espace peut s'exprimer de multiples manières. Étant donné l'utilisation numérique que nous allons faire de cette représentation, nous utilisons les deux représentations :

1. **Vecteur de pose de $I\!R^6$** :

$$Q_{12} = (t_x t_y t_z r_x r_y r_z)^T \tag{2.1}$$

Les trois premières coordonnées $\tau = (t_x t_y t_z)^T$ de ce 6-vecteur représentent la translation entre les deux repères tandis que les trois dernières $\omega = (r_x r_y r_z)^T$ représentent l'orientation. Son module $a = ||\omega||$ est égal à l'angle de rotation et sa direction $u = \frac{\omega}{||\omega||}$

est l'axe de rotation. Dans la suite nous nous intéresserons aux lois de probabilité sur ces coordonnées.

Nous utiliserons dans le reste de ce document la notation $Q_{ij}.t_x, Q_{ij}.t_y, \cdots, Q_{ij}.r_z$ pour dénoter les 6 coordonnées du 6-vecteur Q_{ij}.

2. **Matrice de transformation homogène** 4×4 :

$$T_{12} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \tag{2.2}$$

où R représente la matrice de rotation de S_2 par rapport à S_1 et t représente le 3-vecteur de translation entre les origines de $\Re_1$ et $\Re_2$.

Cette représentation en matrices homogènes présente l'avantage de permettre d'écrire la pose entre deux repère $\Re_b$ et $\Re_f$ reliés par une chaîne de transformations

$$(T_{b1}, T_{12}, \cdots, T_{n-1n}, T_{nf})$$

comme une multiplication de matrices :

$$T_{bf} = T_{b1} * T_{12} * \cdots * T_{n-1n} * T_{nf}$$

Ces deux représentations sont reliées par les équations suivantes :

$$\tau = t$$

et

$$R = \cos(a) I_3 + (1 - \cos(a)) u u^T - \sin(a)\{u\}$$

avec I_3 la matrice identité 3 et

$$\{u\} = \begin{pmatrix} 0 & u3 & -u2 \\ -u3 & 0 & u1 \\ u2 & -u1 & 0 \end{pmatrix}$$

Pour passer d'une représentation à une autre, nous disposons de deux fonctions :

- *vect* qui accepte une matrice de transformation 4×4 comme paramètre d'entrée, et rend le 6-vecteur de pose équivalent comme sortie :

$$vect(T_{ij}) = Q_{ij}$$

- *mat* qui accepte un 6-vecteur de pose comme entrée, et rend une matrice de transformation 4×4 comme sortie :

$$mat(Q_{ij}) = T_{ij}$$

Les définitions de ces deux fonctions sont données dans l'annexe A.

2.1.2 Objet - Forme - Entité géométrique

La représentation des objets (corps) par de simples repères ne permet pas de représenter les formes associées à ces derniers. La modélisation des formes nécessite l'utilisation de représentations plus ou moins complexes suivant la complexité des objets à modéliser. Une représentation utilisant des primitives géométriques simples (parallélépipède, cylindre, sphère) peut être suffisante dans certains cas élémentaires. Par contre, pour des environnements plus complexes la modélisation des formes des objets nécessite des représentations plus complexes comme les arbres CSG ou contenir des informations topologiques comme dans les modèles B-rep du type DCEL.

Pour des raisons de simplicité nous avons choisi un ensemble restreint d'entités géométriques comme éléments de modélisation. Nous nous intéresserons donc plus particulièrement aux primitives suivantes :

- Les sommets (points)
- Les arêtes (segments de droites)
- Les faces polygonales
- Les cylindres
- Les sphères

Malgé sa simplicité, cette représentation permet de définir la forme d'un objet comme étant un ensemble d'entités géométriques simples. Cela permet aussi de décrire les poses relatives entre les objets en explicitant les relations de contact entre différentes entités. Nous reviendrons dans la suite de ce chapitre sur cette notion de contact.

Un objet est alors défini par :

- un repère propre à l'objet appelé repère intrinsèque,
- la liste des entités géométriques le constituant. Chaque entité étant localisée par rapport au repère intrinsèque par un repère qui lui est attaché.

La représentation des différentes entités géométriques de base et la procédure de placement des repères attachés à ces entités sont détaillées dans l'annexe A.

2.1.3 Contact géométrique - contraintes de contact

Une relation géométrique de contact entre deux entités géométriques i et j appartenant à deux objets est une suppression ou bien limitation de la plage de variation d'un ou plusieurs degrés de liberté (ddl) du 6-vecteur Q_{ij} représentant la position/orientation relative sur les 6 ddl possibles. Cette notion de contact induit des contraintes d'égalité $O(Q_{ij}) = 0$ et d'inégalité $C(Q_{ij}) \leq 0$ sur les valeurs que peuvent prendre ces degrés de liberté. Nous nous intéressons dans ce travail aux relations de contact suivantes :

- Point sur Point
- Point sur Arête

- Point sur Face
- Point sur Cylindre
- Point sur Sphère
- Arête sur Arête
- Face sur Face

Prenons l'exemple du contact Face sur Face :

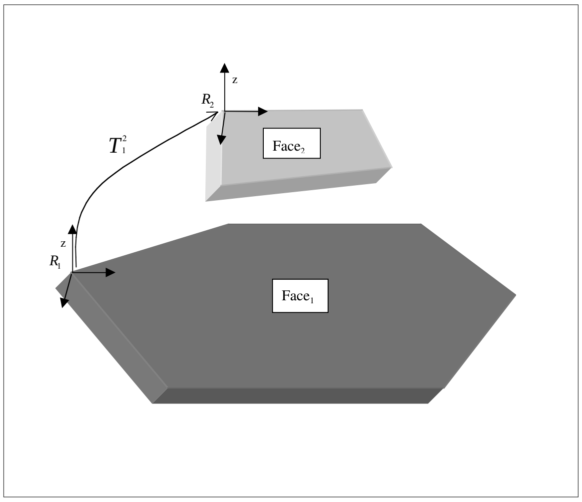

FIG. 2.2 - *Contact de type Face sur Face.*

Un contact de type *Face sur Face* (Figure 2.2) entre deux faces $Face_2$ et $Face_1$ ($Face_2$ sur $Face_1$) induit les contraintes suivantes :

Si l'on dénote $Q_{12} = (t_x t_y t_z r_x r_y r_z)^T$ le vecteur de pose reliant les repères R_1 et R_2 attachés à $Face_1$ et $Face_2$ respectivement, on aura les 3 contraintes égalités suivantes :

$$\begin{cases} t_z & = & 0 \\ r_x & = & 0 \\ r_y & = & 0 \end{cases}$$

Si l'on dénote P_i^1 $i = 1 \cdots n_1$ et P_j^2 $j = 1 \cdots n_2$ l'ensemble des sommets de $Face_1$ et de $Face_2$ respectivement exprimés dans R_1, et si l'on considère le cas où $Face_1$ est convexe, on obtient les $n_1 \times n_2$ contraintes inégalités (Figure 2.3) :

$$\forall i = 1 \cdots n_1, \forall j = 1 \cdots n_2, \quad v_{ij}.z < 0$$

où :

$$v_{ij} = (p_i^1 - p_j^2) \wedge (p_{i+1}^1 - p_j^2)$$

Les contraintes d'égalité représentent le fait que le plan support de $Face_1$ est sur le plan support de $Face_2$, tandis que les contraintes d'inégalité représentent le fait que tous les sommets de $Face_2$ doivent être à l'intérieur de la face $Face_1$ (Figure 2.3).

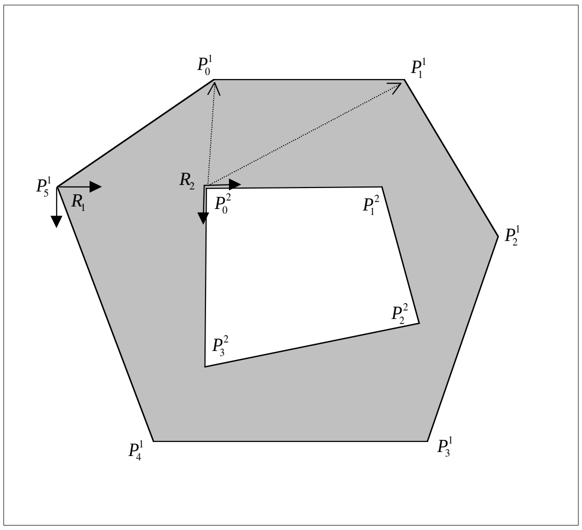

FIG. 2.3 - *Les contraintes inégalités induites par le contact Face sur Face.*

Les autres types de contacts géométriques ainsi que les contraintes qui en résultent sont détaillés dans l'annexe A.

2.1.4 Graphe cinématique

Nous définissons un *Graphe cinématique* comme étant un graphe orienté donné par un ensemble de n sommets (nœuds) $S = \{S_1, \cdots, S_n\}$ représentant un ensemble de repères, et un ensemble de m arcs (arêtes) $A = \{A_{i_1 j_1}, \cdots, A_{i_m j_m}\}$ où A_{ij} désigne un arc allant du sommet père S_i vers son fils S_j. Chaque arc A_{ij} définit un *lien cinématique*. Il est étiqueté par un vecteur de pose Q_{ij} de $I\!R^6$ représentant la pose relative du sommet fils S_j par rapport au sommet père S_i. Il est aussi étiqueté par un ensemble (éventuellement vide) de contraintes d'égalité $O(Q_{ij}) = 0$ et d'inégalité $C(Q_{ij}) < 0$ induites par une éventuelle relation de contact entre deux entités géométriques attachées à S_i et S_j. Un arc définit ce que nous appelons *lien cinématique*.

Ce graphe a les propriétés suivantes :

- Caractère orienté : Ce graphe est par définition orienté. Dans le cas général, les vecteurs de poses Q_{ij} et Q_{ji} sont différents.

- Connexité : Ce graphe est de nature connexe car seuls nous intéressent les repères reliés directement ou indirectement entre eux.

- Présence de cycles : Ce graphe peut contenir des cycles.

2.1.5 Présence des cycles et la cohérence du modèle

Nous venons de mentionner que le graphe cinématique peut contenir des cycles. Un cycle de ce graphe représente l'existence de plus d'un chemin reliant deux repères. L'utilisation de chacun de ces chemins permet de calculer une pose relative entre les deux repères en question. Le calcul de ces poses peut donner des résultats différents, ce qui rend le modèle *incohérent*. Pour que le modèle soit cohérent, toutes les poses calculées en utilisant les différents chemins doivent être identiques. Nous appelons cela la contrainte de *fermeture des cycles* et nous l'exprimons comme suit :

Pour tout cycle C constitué de k arcs :

$$\forall i = 1 \cdots n \;\; T_{i\ i} = T_{i\ i+1}^{s_{i\ i+1}} * T_{i+1\ i+2}^{s_{i+1\ i+2}} * \cdots * T_{k-1\ k}^{s_{k-1\ k}} * T_{k\ 1}^{s_{k\ 1}} * T_{1\ 2}^{s_{1\ 2}} * \cdots * T_{i-1\ i}^{s_{i-1\ i}} = I_4$$

où I_4 représente la matrice identité 4×4 et $s_{ij} \in \{-1, 1\}$ représente le sens du parcours de l'arc A_{ij}.

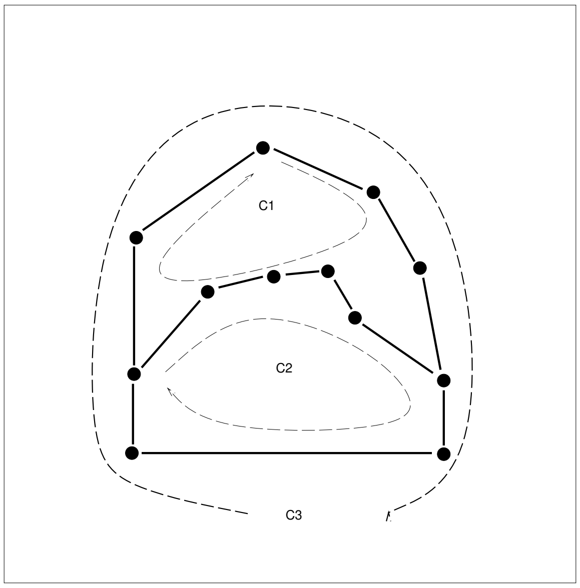

FIG. 2.4 - *Cas d'un graphe connexe contenant 12 sommets et 13 arcs. Le nombre minimal de cycles permettant de couvrir complètement ce graphe est de* $13 - 12 + 1 = 2$.

Le nombre minimal de cycles permettant de couvrir complètement un graphe connexe de n sommets et m arcs est de $m - n + 1$ [Gondran90]. Dans la figure 2.4 par exemple, le nombre de cycles que l'on peut extraire de ce graphe est de 3 (les cycles $C1$, $C2$ et $C3$ dans la figure). Le nombre de sommets et d'arcs de ce graphe étant de 12 et 13 respectivement, alors le nombre minimal de cycles permettant de couvrir complètement ce graphe est de 2.

L'extraction de ces cycles conduit, dans le cas général, à plusieurs solutions possibles. L'algorithme que nous avons implanté pour ce travail permet d'extraire les $m - n + 1$ cycles de longueurs minimales. Cela permet pour l'exemple 2.4 de trouver les deux cycles $C2$ et $C3$.

2.1.6 Problème géométrique

Étant donné un graphe cinématique défini par un ensemble de n sommets $S = \{S_1, \cdots, S_n\}$ et un ensemble de m arcs $A = \{A_{i_1 j_1}, \cdots, A_{i_m j_m}\}$, la spécification d'un problème géométrique à partir de ce graphe revient à définir pour chaque arc A_{ij} le rôle (nature) de chaque paramètre du vecteur de pose Q_{ij} vis-à-vis du problème. Pour cela, nous introduisons la notion de *statut*. Cette notion permet de caractériser le rôle de chaque paramètre, et d'associer à chaque vecteur de pose $Q_{i_k j_k}$ $k = 1 \cdots m$, un 6-vecteur de statuts. Nous définissons ainsi trois catégories de variables :

- *Inconnu* (X) pour les paramètres dont les valeurs sont les inconnues du problème.
- *Fixe* (F) pour les paramètres dont les valeurs sont connues et inchangeables.
- *Libre* (L) pour les paramètres qui peuvent prendre indifféremment n'importe quelle valeur permettant de satisfaire les éventuelles contraintes inégalités sur ces paramètres.

Si on définit le vecteur Q de $\mathbb{R}^{6m}$ représentant la concaténation des m 6-vecteurs Q_{ij}, la spécification d'un problème géométrique correspond à une partition de ce vecteur en trois sous-ensembles X, F et L correspondant aux paramètres de Q ayant les statuts *Inconnu*, *Fixe* et *Libre* respectivement.

Étant donné un vecteur de paramètres fixes (F), la résolution du problème revient à trouver les vecteurs X et L permettant de satisfaire :

- Les contraintes d'égalité représentant les conditions de fermeture des cycles (Contraintes globales).
- Les contraintes d'égalité et d'inégalité induites par les relations de contact (Contraintes locales).

2.1.7 Exemples de problèmes géométriques

Le but de ce paragraphe est d'illustrer la méthode de spécification utilisée. Nous donnerons deux exemples simples représentant deux types de problèmes :

- Le premier exemple représente un problème simple d'inversion géométrique.
- Le second décrit un problème de calibration (étalonnage) de la pose d'un objet en utilisant des mesures capteur.

Pour des raisons de simplicité, nous considérons dans ces exemples le cas 2D. Cette méthode de spécification reste valide pour le cas 3D.

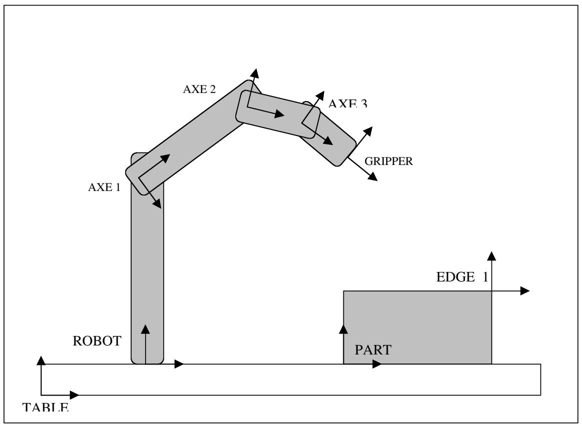

FIG. 2.5 - *Exemple simple d'inversion géométrique. Le problème à résoudre consiste à mettre le point final du bras sur une arête d'un objet.*

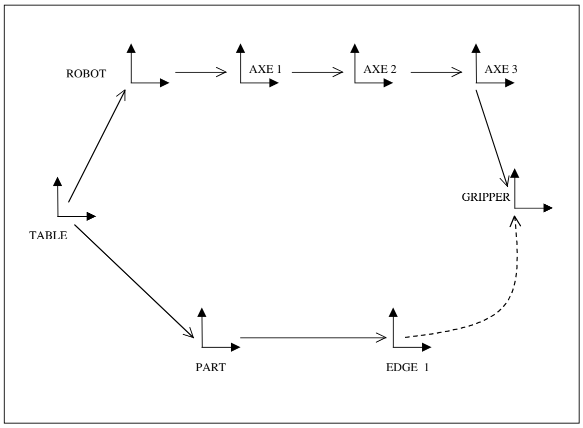

FIG. 2.6 - *Graphe cinématique correspondant au problème d'inversion géométrique.*

Élaboration d'un problème d'inversion géométrique

L'exemple présenté dans la figure 2.5 représente un problème d'inversion géométrique. Dans cet exemple, nous voulons que le point final (noté GRIPPER) du bras planaire à 3 degrés de liberté rejoigne l'arête notée *EDGE1* de l'objet *PART*. Plus précisément, nous voulons obtenir les valeurs $(\theta_1, \theta_2, \theta_3)$ des degrés de liberté du bras qui permettent de mettre le point final *GRIPPER* sur l'arête *EDGE1*. Nous présentons dans la figure 2.6 le graphe cinématique correspondant à ce problème. Les arcs de ce graphe en traits continus représentent les liens cinématiques décrivant les informations sur les poses relatives des repères, tandis que les arcs en traits pointillés expriment la relation que l'on souhaite obtenir.

L'ensemble des sommets de ce graphe est :

$$S = \{TABLE, ROBOT, PART, EDGE1, AXE1, AXE2, AXE3, GRIPPER\}$$

L'ensemble des arcs est donné, quant à lui, par le tableau 2.1.

L'ensemble X des inconnues du problème (correspondant aux articulations θ_1, θ_2 et θ_3 du bras) est :

$$X = \{Q_{ROBOT\ AXE1}.rz, Q_{AXE1\ AXE2}.rz, Q_{AXE2\ AXE3}.rz\}$$

Arcs	Valeurs de (t_x, t_y, r_z)	Statuts	Contraintes inégalités
$TABLE - ROBOT$	(100.0, 10.0, 0.0)	(F, F, F)	
$TABLE - PART$	(300.0, 10.0, 0.0)	(F, F, F)	
$PART - EDGE$	(150.0, 30.0, 0.0)	(F, F, F)	
$ROBOT - AXE1$	$(0.0,\ l_1,\ \theta_1)$	(F,F,X)	$a_1 \leq \theta_1 \leq b_1$
$AXE1 - AXE2$	$(0.0,\ l_2,\ \theta_2)$	(F,F,X)	$a_2 \leq \theta_2 \leq b_2$
$AXE2 - AXE3$	$(l_3,\ 0.0,\ \theta_3)$	(F,F,X)	$a_3 \leq \theta_3 \leq b_3$
$AXE3 - GRIPPER$	$(l_4,\ 0.0,\ 0.0)$	(F, F, F)	
$EDGE1 - GRIPPER$	(?, 0.0,?)	(L, F, L)	$-150.0 \leq x \leq 0.0$

TAB. 2.1 - *Statuts et contraintes inégalités pour le problème d'inversion géométrique.*

Pour ce problème, le graphe correspondant contient un seul cycle. La contrainte égalité de la fermeture de ce cycle s'écrit :

$$\begin{gathered} T_{TABLE\ ROBOT} * T_{ROBOT\ AXE1} * T_{AXE1\ AXE2} * T_{AXE2\ AXE3} * T_{AXE3\ GRIPPER} \\ = \\ T_{TABLE\ PART} * T_{PART\ EDGE1} * T_{EDGE1\ GRIPPER} \end{gathered}$$

Les contraintes inégalités $a_i \leq \theta_i \leq b_i,\ i \in \{1, 2, 3\}$ dans la quatrième, cinquième et sixième ligne du tableau 2.1 représentent les butées mécaniques associées aux trois degrés de liberté du bras. La contrainte $-150.0 \leq x \leq 0.0$ dans la dernière ligne du tableau exprime le fait que le point final *GRIPPER* doit être sur l'arête *EDGE1*.

Élaboration d'un problème de calibration (étalonnage) de pose

Le problème que nous cherchons à résoudre dans ce deuxième exemple est d'étalonner la pose de l'objet *PART* par rapport au repère *TABLE* en utilisant 3 mesures d'un capteur de distance monté à l'extrémité du bras (voir figure 2.7). Pour chaque mesure $M_i,\ \ i = 1 \cdots 3$, le bras est ramené à une configuration $(\theta_1^i, \theta_2^i, \theta_3^i)$ et la distance D_i entre l'extrémité du bras (laser) et une des arêtes *EDGE1* et *EDGE2* appartenant à l'objet *PART* est mesurée. Pour cet exemple, les inconnues du problème ne sont plus les paramètres de la configuration du robot comme pour l'exemple précédent mais les paramètres de la pose (x, y, α) de l'objet *PART* par rapport à la table *TABLE*. Le graphe cinématique associé à ce problème est donné dans la figure 2.8.

L'ensemble S des sommets de ce graphe est :

$$S = \{ \quad TABLE, ROBOT, PART, EDGE1, EDGE2, AXE1_1, AXE1_2, AXE1_3,$$

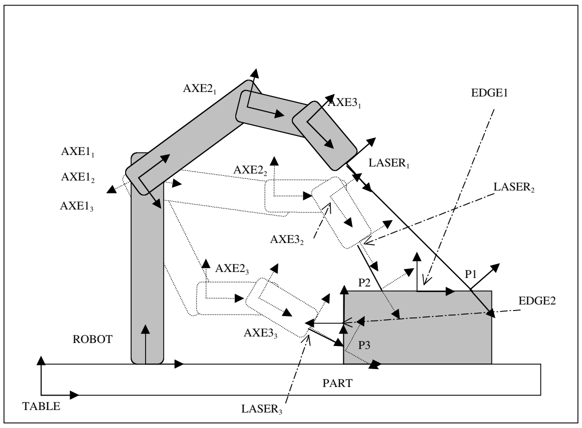

FIG. 2.7 - *Exemple de problème d'étalonnage. Le problème consiste à trouver la pose d'un objet à partir d'un ensemble de mesures capteur.*

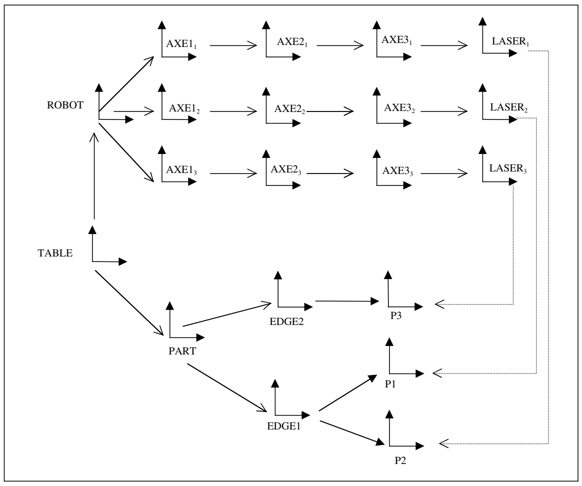

FIG. 2.8 - *Graphe cinématique correspondant au problèm d'étalonnage.*

$AXE2_1, AXE2_2, AXE2_3, AXE3_1, AXE3_2, AXE3_3, LASER_1, LASER_2,$
$LASER_3\}$

L'ensemble des arcs est donné, quant à lui, par le tableau 2.2.

L'ensemble X des inconnues du problème (correspondant à la pose (x, y, α) de $PART$) est :

$$X = \{Q_{TABLE\ PART}.tx, Q_{TABLE\ PART}.ty, Q_{TABLE\ PART}.rz\}$$

Arcs	Valeurs de (t_x, t_y, r_z)	Statuts	Contraintes inégalités
$TABLE - ROBOT$	(100.0, 10.0, 0.0)	(F, F, F)	
$TABLE - PART$	$(x,\ y,\ \alpha)$	(X, X, X)	
$PART - EDGE1$	(150.0, 30.0, 0.0)	(F, F, F)	
$PART - EDGE2$	$(0.0,\ 40.0,\ \pi/2)$	(F, F, F)	
$ROBOT - AXE1_1$	$(0.0,\ l_1,\ \theta_1^1)$	(F,F,F)	
$ROBOT - AXE1_2$	$(0.0,\ l_1,\ \theta_1^2)$	(F,F,F)	
$ROBOT - AXE1_3$	$(0.0,\ l_1,\ \theta_1^3)$	(F,F,F)	
$AXE1_1 - AXE2_1$	$(0.0,\ l_2,\ \theta_2^1)$	(F,F,F)	
$AXE1_2 - AXE2_2$	$(0.0,\ l_2,\ \theta_2^2)$	(F,F,F)	
$AXE1_3 - AXE2_3$	$(0.0,\ l_2,\ \theta_2^3)$	(F,F,F)	
$AXE2_1 - AXE3_1$	$(l_3,\ 0.0,\ \theta_3^1)$	(F,F,F)	
$AXE2_2 - AXE3_2$	$(l_3,\ 0.0,\ \theta_3^2)$	(F,F,F)	
$AXE2_3 - AXE3_3$	$(l_3,\ 0.0,\ \theta_3^3)$	(F,F,F)	
$AXE3_1 - LASER_1$	$(l_4,\ 0.0,\ 0.0)$	(F, F, F)	
$AXE3_2 - LASER_2$	$(l_4,\ 0.0,\ 0.0)$	(F, F, F)	
$AXE3_3 - LASER_3$	$(l_4,\ 0.0,\ 0.0)$	(F, F, F)	
$LASER_1 - P1$	$(0.0, D_1\ ,\ 0.0)$	(F, F, F)	
$LASER_2 - P2$	$(0.0,\ D_2,\ 0.0)$	(F, F, F)	
$LASER_3 - P3$	$(0.0,\ D_3,\ 0.0)$	(F, F, F)	
$EDGE1 - P1$	(?, 0.0,?)	(L, F, L)	$-40.0 \leq x \leq 40.0$
$EDGE1 - P2$	(?, 0.0,?)	(L, F, L)	$-40.0 \leq x \leq 40.0$
$EDGE2 - P3$	(?, 0.0,?)	(L, F, L)	$-20.0 \leq x \leq 20.0$

TAB. 2.2 - *Statuts et contraintes inégalités pour le problème d'étalonnage.*

Pour ce problème, le graphe correspondant contient 3 cycles minimaux. Les contraintes égalités de la fermeture de ces cycles s'écrivent :

$$\begin{gathered} T_{TABLE\ ROBOT} * T_{ROBOT\ AXE1_1} * T_{AXE1_1\ AXE2_1} * \\ T_{AXE2_1\ AXE3_1} * T_{AXE3_1\ LASER_1} * T_{LASER_1\ P1} \\ = \\ T_{TABLE\ PART} * T_{PART\ EDGE1} * T_{EDGE1\ P1} \end{gathered}$$

$$\begin{gathered} T_{TABLE\ ROBOT} * T_{ROBOT\ AXE1_2} * T_{AXE1_2\ AXE2_2} * \\ T_{AXE2_2\ AXE3_2} * T_{AXE3_2\ LASER_2} * T_{LASER_2\ P2} \\ = \\ T_{TABLE\ PART} * T_{PART\ EDGE1} * T_{EDGE2\ P2} \end{gathered}$$

$$\begin{gathered} T_{TABLE\ ROBOT} * T_{ROBOT\ AXE1_3} * T_{AXE1_3\ AXE2_3} * \\ T_{AXE2_3\ AXE3_3} * T_{AXE3_3\ LASER_3} * T_{LASER_3\ P3} \\ = \\ T_{TABLE\ PART} * T_{PART\ EDGE2} * T_{EDGE2\ P3} \end{gathered}$$

Dans les trois dernières lignes du tableau 2.2, les contraintes inégalités expriment le fait que les trois points de mesures $P1$, $P2$ et $P3$ se situent sur les arêtes *EDGE1* et *EDGE2*.

2.2 Principes du calcul bayesien

Cette partie décrit les notions théoriques avec lesquelles nous abordons la gestion des incertitudes géométriques. Notre objectif est de présenter brièvement les principes du calcul bayesien comme une méthode de modélisation et d'inférence permettant de prendre en compte les incertitudes. L'application de ce fromalisme connaît aujourd'hui un regain d'intérêt marquant dans différents domaines de recherches, allant de la physique théorique [Jaynes96, Neal93] jusqu'à la vision par ordinateur [Kato94, Weiss98] et la robotique mobile [Bessière98a, Bekerman92].

Nous nous plaçons plus précisément dans le cadre d'une théorie du raisonnement probabiliste appelée *Probability as Logic* (PaL) proposée par le physicien E.T.Jaynes [Jaynes96]. Cette théorie considère le calcul des probabilités comme une généralisation de la logique formelle (booléenne). Elle permet d'étendre la logique à des propositions dont la vérité n'est pas connue avec certitude et de formaliser la notion de raisonnement plausible. Dans cette théorie, la notion de probabilité est utilisée non pas dans son sens *fréquentiste* où elle caractérise la *fréquence* d'une mesure physique donnée, mais plutôt dans son sens *subjectiviste* où elle exprime un *état de connaissance* représentant les *informations* dont on dispose sur le phénomène en question. Cette utilisation des probabilités découle directement de la formalisation de la notion intuitive de *plausibilité* proposée par Cox [Cox46, Cox61]. En se fixant un ensemble de desiderata définissant la notion de plausibilité pour une proposition logique, Cox montre que l'unique façon de manipuler cette notion (en restant fidèle aux desiderata de départ) est donnée par la théorie des probabilités. La notion de plausibilité est ainsi formalisée par la notion mathématique bien définie de *probabilité*[1].

2.2.1 Définitions fondamentales

Probabilité d'une proposition logique

Une proposition logique est un énoncé qui peut être soit vrai soit faux. Nous définissons la plausibilité (probabilité) $p(A)$ d'une proposition A comme le degré de certitude accordé à sa véracité.

En réalité, toute probabilité d'une proposition A ne peut être donnée qu'au vu d'un ensemble de *connaissances préalables* que l'on dénotera I. Il est alors plus convenable d'expliciter dans les notations les connaissances I qui ont permis d'assigner une valeur p_A à une proposition A en écrivant $p(A|I) = p_A$. Toutefois, dans le but d'alléger les notations, nous nous dispensons d'écrire ces connaissances I dans les formules que nous manipulons dans le reste de ce document. En effet à un instant donné nous ne nous intéressons qu'à un seul ensemble de connaissances préalables I.

Nous utiliserons, dans ce qui suit, la notation AB pour la conjonction de A et B, $A + B$ pour leur disjonction et $\neg A$ et $\neg B$ pour leurs négations respectives.

1. Dans un récent article [Halpern99], la démonstration de Cox de l'unicité de la théorie de probabilité pour la manipulation de la plausibilié a été mise en cause. Toutefois, cette mise en cause ne concerne en aucun cas la capacité de la théorie des probabilités de manipuler la notion de plausibilité.

Règles de calcul

À partir des desiderata définissant le raisonnement plausible, Cox montre que ce raisonnement doit utiliser 2 règles fondamentales qui sont à la base de toute inférence probabiliste. Étant donnée les propositions logiques A, B et C, ces deux règles sont données par :

- **Règle du produit**
  qui donne la probabilité d'une conjonction :

$$p(AB|C) = p(A|C)p(B|AC) = p(B|C)p(A|BC) \tag{2.3}$$

- **Règle de normalisation**
  qui exprime le fait que la somme des probabilités d'une proposition et de sa négation est égale à 1 :

$$p(A|C) + p(\neg A|C) = 1 \tag{2.4}$$

2.2.2 Autres définitions et notations utiles

Probabilité d'une disjonction

L'utilisation des règles 2.3 et 2.4 permet d'écrire la probabilité d'une disjonction comme suit :

$$p(A + B|C) = p(A|C) + p(B|C) - p(AB|C)$$

Cette dernière règle s'écrit dans le cas où A et B sont indépendantes comme :

$$p(A + B|C) = p(A|C) + p(B|C)$$

Cette règle est obtenue comme suit :

$$\begin{aligned}
p(A + B|C) &= 1 - p(\neg A \neg B|C) = 1 - p(\neg A|C)p(\neg B|\neg AC) \\
&= 1 - p(\neg A|C)\left[1 - p(B|\neg AC)\right] = p(A|C) + p(\neg AB|C) \\
&= p(A|C)p(B|C)p(\neg A|BC) = p(A|C) + p(B|C)\left[1 - p(A|BC)\right] \\
&= p(A|C) + p(B|C) - p(AB|C)
\end{aligned}$$

Variable discrète - Distribution de probabilité discrète

Jusque ici, seule la probabilité d'une proposition logique a été définie. Supposons que l'on s'intéresse à assigner des probabilités aux différentes valeurs numériques d'une variable discrète x. Nous devons dans ce cas définir un ensemble de propositions logiques mutuellement exclusives $A_n \equiv$ "$x = n''$, où n prend toutes les valeurs possibles de x, et considérer les probabilités $p(A_n) = f(n)$. La fonction f représente une *distribution de probabilité discrète* sur la plage de variation de la variable x.

Pour éviter toute ambiguïté, nous utilisons la notation $p_x(x)$ pour désigner une distribution de probabilité sur une variable x. Dans le cas où aucune confusion ne serait possible, nous nous permettons d'alléger la notation en écrivant tout simplement $p(x)$.

Les deux règles 2.3 et 2.4 restent valides pour les cas des variables et s'écrivent respectivement pour deux variables x et y comme suit :

$$p(xy) = p(x)p(y|x) = p(y)p(x|y)$$
$$\sum_x p(x) = 1$$

Variable continue - Densité de probabilité

Pour parler de probabilité pour une variable continue x en utilisant les résultats obtenus pour les propositions logiques, nous reprenons le raisonnement donné par Jaynes :

Commençons par définir les deux propositions suivantes :

$$F_1 \equiv (x \leq q) \tag{2.5}$$

$$F_2 \equiv (x > q) \tag{2.6}$$

Ces deux propositions sont exhaustives et mutuellement exclusives. La probabilité de F_1 va dépendre de q en définissant une fonction G telle que :

$$p(F_1) = G(q)$$

Cette fonction G est bien évidemment croissante.

Intéressons nous maintenant à la proposition $W \equiv (a < x \leq b)$ (la valeur de x est dans l'intervalle $]a, b]$). Notons A et B les propositions suivantes :

$$A \equiv (x \leq a) \tag{2.7}$$

$$B \equiv (x \leq b) \tag{2.8}$$

Nous avons l'égalité $B = A + W$. De plus, comme A et W sont mutuellement exclusives, l'application de la règle de la somme permet d'écrire :

$$p(B) = p(A) + p(W)$$

Ce qui donne finalement :

$$p(a < x \leq b) = p(W) = G(b) - G(a)$$

Si G est différentiable, nous pourrons écrire :

$$p(a < x \leq b) = \int_a^b g(x)dx$$

avec $g(x) = \frac{dG(x)}{dx}$

La fonction g est appelée la *densité de probabilité* ou bien la *distribution de probabilité*. Nous l'appellerons dans le reste de ce document *distribution de probabilité* et nous utiliserons la notation $p_x(x)$ pour dénoter la distribution sur une variable continue. Nous nous permettrons d'écrire tout simplement $p(x)$ dans les contextes où aucune confusion n'est possible.

Les deux règles fondamentales 2.3 et 2.4 restent valides pour les cas des variables continues et s'écrivent respectivement pour deux variables x et y comme suit :

$$p(xy) = p(x)p(y|x) = p(y)p(x|y)$$
$$\int p(x)dx = 1$$

La distribution $p(xy)$ est appelée la *distribution conjointe* de x et y.

Marginalisation - Distribution marginale

Soit x et y deux variables numériques. On appelle *distribution marginale* de x par rapport à y la distribution de probabilité :

- $p(x) = \sum_y p(xy) = \sum_y p(y)p(x|y)$ dans le cas où y est discrète
- $p(x) = \int p(xy)dy = \int p(y)p(x|y)dy$ dans le cas où y est continue

Formule de Bayes

Une transformation mathématique directe de la règle du produit permet d'obtenir la fameuse formule dite de *Bayes*

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)} = \frac{p(x)p(y|x)}{\sum_x p(x)p(y|x)}$$

2.2.3 Méthodologie d'utilisation

Dans ce paragraphe, nous présentons les principes de la méthodologie que nous suivons pour modéliser un état de connaissances probabiliste, ainsi que pour utiliser cet état de connaissances pour répondre à un problème donné. Cette méthodologie passe par deux phases :

1. La spécification des connaissances préalables (construction d'un état de connaissances).
2. L'utilisation de cet état de connaissances pour une question (problème) donnée.

Nous allons, dans ce qui suit, décrire plus précisément chacune de ces deux phases.

Spécification

La spécification d'un *état de connaissance probabiliste* est la formalisation de l'ensemble des connaissances dont le modélisateur dispose et qu'il juge pertinentes pour modéliser un phénomène donné. Cette spécification se fait en 3 étapes :

1. La définition des variables pertinentes.

2. La construction de la distribution conjointe sur cet ensemble de variables en la décomposant en distributions élémentaires.

3. La définition des formes paramétriques de ces distributions élémentaires.

Nous décrirons dans ce qui suit chacune de ces étapes.

- **Définition des variables**
  La première étape est la définition de l'ensemble des variables utilisées pour la modélisation. Ces variables sont les paramètres jugés pertinents et suffisants pour modéliser le phénomène auquel on s'intéresse. On notera $V = \{v_i\}_{i=0}^{n}$ l'ensemble de ces n variables.

- **Décomposition de la distribution conjointe - Structure de dépendance**
  Étant donné un ensemble de variables $V = \{v_i\}_{i=0}^{n}$ du modèle, la deuxième étape dans la spécification d'un état de connaissance probabiliste est de donner une méthode de calcul de la distribution conjointe $p(v_0 \cdots v_n)$ décrivant la distribution de probabilité sur l'ensemble de ces variables. Cette étape permet d'expliciter les connaissances du modélisateur sur les dépendances entre variables. Cela revient à trouver une décomposition de la distribution conjointe (en utilisant les règles des probabilités) permettant de simplifier cette dernière en exploitant les éventuelles indépendances conditionnelles entre les variables, pour l'écrire comme un produit de termes élémentaires plus simple à calculer.

  Prenons en exemple une distribution conjointe $p(xyz)$ de 3 variables x, y et z. On peut toujours écrire :

  $$p(xyz) = p(z)p(y|z)p(x|yz)$$

  Si l'on suppose que x et y sont indépendantes sachant la valeur de z, on aura :

  $$p(xyz) = p(z)p(y|z)p(x|z)$$

  Pour que cette décomposition soit utilisable pour l'inférence, on doit être capable de calculer les 3 termes élémentaires $p(z)$, $p(y|z)$ et $p(x|z)$: pour toutes les valeurs que les variables x, y et z peuvent prendre.

- **Formes paramétriques des distributions élémentaires**
  La dernière étape dans la description d'un état de connaissance probabiliste est de donner, pour chaque distribution élémentaire résultant de la décomposition, une forme paramétrique permettant de calculer cette distribution pour chaque valeur des variables qui apparaissent dans cette distribution.

  Ces formes paramétriques peuvent être pour l'exemple précédent :

  - $p(z) = Normal(0, 10) = \frac{1}{10\sqrt{2\pi}} e^{-\frac{1}{2}\frac{z^2}{100}}$
  - $p(y|z) = Uniform(z - 100, z + 100) = \begin{cases} \frac{1}{200} & si \quad z - 100 \leq y \leq z + 100 \\ 0 & sinon \end{cases}$
  - $p(x|z) = Normal(z, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-z)^2}$

Utilisation

L'utilisation de l'état de connaissances (distribution conjointe) issu de la spécification passe par trois étapes :

1. La définition de la question (une distribution conditionelle) correspondant à un problème .
2. L'inférence de la formule de cette distribution à partir de la distribution conjointe.
3. Le choix (décision) d'une valeur (réponse au problème) à partir de cette distribution.

- **Question**
  Étant donné une décomposition calculable de la distribution conjointe $p(V) = p(v_0 \cdots v_n)$, on peut s'intéresser à un instant donné à la distribution $p(V_g|V_d)$ d'un sous ensemble de variables V_g de V conditionnée par un autre sous ensemble de variables V_d de V.

  Cela revient en fait à définir une partition $\{V_g V_d V_m\}$ de l'ensemble V ($V_g \cup V_d \cup V_m = V$ et $V_g \cap V_d = V_d \cap V_m = V_m \cap V_g = \phi$).

  L'ensemble V_d représente les variables dont les valeurs sont supposées connues pour la question (problème) à laquelle on va s'intéresser. V_g représente les inconnues du problème et V_m les variables *manquantes* dont les valeurs ne sont connues qu'avec des distributions de probabilité.

  Reprenons l'exemple précédent et supposons que l'on s'intéresse à la distribution $p(x|[y = 1])$. On a : $V_d = \{y\}$, $V_g = \{x\}$ et $V_m = \{z\}$.

- **Inférence**
  La première étape dans la résolution d'un problème donné par une distribution $p(V_g|V_d)$, est d'inférer symboliquement une formule calculable de cette distribution à partir de la distribution conjointe $p(V)$ en utilisant les règles de probabilité. Cette inférence consiste essentiellement à faire des marginalisations sur l'ensemble V_m des variables manquantes. Cela revient à écrire :

  $$\begin{aligned} p(V_g|V_d) &= \frac{p(V_g V_d)}{p(V_d)} \\ &= \frac{\sum_{V_m} p(V_g V_d V_m)}{p(V_d)} \end{aligned}$$

  Il est suffisant, dans le cas général, d'avoir la valeur de cette distribution à un facteur multiplicatif près (parce qu'on cherche un maximum). Ce qui permet d'écrire :

  $$p(V_g|V_d) \propto \sum_{V_m} p(V_g V_d V_m) \tag{2.9}$$

  Dans le cas où les variables sont de nature continue, l'équation 2.9 s'écrit :

  $$p(V_g|V_d) \propto \int p(V_g V_d V_m) dV_m \tag{2.10}$$

Pour notre exemple, cela revient à écrire la distribution $p(x|y)$ comme :

$$p(x|y) \propto \int p(z)p(x|z)p(y|z)dz$$

Notons la présence de l'intégrale multidimensionnelle sur l'espace défini par l'ensemble V_m. Le calcul de cette intégrale (qui peut porter sur des espaces de grande dimension) représente un des problèmes difficiles dans l'utilisation du calcul bayesien. Le problème d'intégration sur des espaces de grande dimension sera justement l'objet du chapitre suivant.

- **Décision**
Disposant d'une formule permettant de calculer la distribution représentant le problème à résoudre $p(V_g|V_d)$ pour toute valeur de V_g, la dernière étape de la résolution consiste à *choisir* une valeur de V_g à partir de la cette distribution.

Dans ce travail, nous n'aborderons pas le problème de la décision dans son cadre général [2]. Nous nous intéressons seulement à la distribution sur l'espace des solutions du problème. Cette distribution représente un état de connaissance purement descriptif, et le choix d'un point (configuration) dans l'espace des solutions passe par l'une des deux stratégies suivantes :

  - Effectuer un tirage à partir de la distribution $p(V_g|V_d)$. Cette stratégie peut être utilisée pour restituer l'incertitude induite par cette distribution. Un ensemble de tirages permet ainsi d'avoir une idée de la distribution pour simuler le comportement du système.
  - Prendre la valeur de V_g qui maximise cette distribution :

$$V_g^* = \max_{V_g} p(V_g|V_d) \tag{2.11}$$

C'est cette dernière stratégie, maximisant la distribution $p(V_g|V_d)$, que nous utilisons pour aborder les problèmes de géométrie. On parle plus précisément d'une estimation au Maximum A Posteriori (MAP). Cela représente un problème d'optimisation sur l'espace défini par l'ensemble de variables V_g, ce qui nécessite une méthode d'optimisation bien adaptée à ce type de fonctions. Ce problème d'optimisation est le deuxième point difficile dans l'utilisation du calcul bayesien et sera aussi traité dans le chapitre suivant.

2.3 Spécification probabiliste d'un problème

2.3.1 Graphe cinématique probabiliste

Dans ce travail, nous avons fait le choix de représenter l'état de connaissance sur la pose relative entre deux objets par une distribution de probabilité sur les paramètres de cette

2. Dans plusieurs travaux, ce problème est abordé en introduisant une *fonction de coût* de la décision prise (voir [Jaynes96, Pearl91])

pose. Ce choix est justifié par le fait que la *probabilité* représente une notion mathématique bien définie qui permet de représenter les incertitudes et de raisonner en présence de ces dernières.

À partir de ce choix, nous définissons un *Graphe cinématique probabiliste* comme un graphe orienté donné par un ensemble de n sommets $S = \{S_1, \cdots, S_n\}$ représentant un ensemble de repères, et un ensemble de m arcs (arêtes) $A = \{A_{i_1j_1}, \cdots, A_{i_mj_m}\}$ où A_{ij} désigne un arc allant du sommet père S_i vers son fils S_j. Chaque arc A_{ij}, appelé *lien cinématique probabiliste*, est étiqueté par une distribution de probabilité $p(Q_{ij})$ sur le 6-vecteur Q_{ij} de $\mathbb{R}^6$ représentant la pose du sommet fils S_j par rapport au sommet père S_i. Il est aussi étiqueté par un ensemble (éventuellement vide) de contraintes d'égalité $O(Q_{ij}) = 0$ et d'inégalité $C(Q_{ij}) < 0$ induites par une éventuelle relation de contact entre les deux entités géométriques attachées à S_i et S_j.

2.3.2 Problème géométrique probabiliste

Étant donné un graphe cinématique probabiliste défini par un ensemble de n sommets $S = \{S_1, \cdots, S_n\}$ et un ensemble de m arcs $A = \{A_{i_1j_1}, \cdots, A_{i_mj_m}\}$, la spécification d'un problème géométrique probabiliste à partir de ce graphe revient à définir pour chaque arc A_{ij} le rôle de chaque paramètre du vecteur de pose Q_{ij} vis-à-vis du problème en question. Nous gardons pour cette version probabiliste les deux statuts *Inconnu* et *Fixe* introduits auparavant. La signification du statut *Libre* va changer du fait qu'un paramètre ayant ce statut va prendre sa valeur suivant une distribution de probabilité.

À partir de cette définition, chaque paramètre aura comme statut l'une des trois valeurs suivantes :

- *Inconnu* (X) pour les paramètres représentant les inconnues du problème et dont les valeurs doivent être inférées pour trouver une solution au problème.

- *Fixe* (F) pour les paramètres dont les valeurs sont connues et ne peuvent être changées (données par des scalaires).

- *Libre* (L) pour les paramètres ne faisant pas partie des inconnues du problème et dont les valeurs ne sont connues que par des distributions de probabilité. Ces paramètres permettent d'exprimer les incertitudes géométriques liées au modèle.

Si le vecteur Q de $\mathbb{R}^{6*m}$ représente la concaténation des m 6-vecteurs Q_{ij}, la spécification d'un problème géométrique correspond à une partition de ce vecteur en trois sous-ensembles X, L et F correspondant aux paramètres de Q ayant respectivement les statuts $Inconnu$, $Fixe$ et $Libre$.

2.3.3 Exemples de problèmes géométriques probabilistes

Pour illustrer la méthode de modélisation probabiliste, nous reprenons les deux problèmes géométriques présentés auparavant et nous introduisons une dimension probabiliste en supposant des incertitudes sur les paramètres de poses.

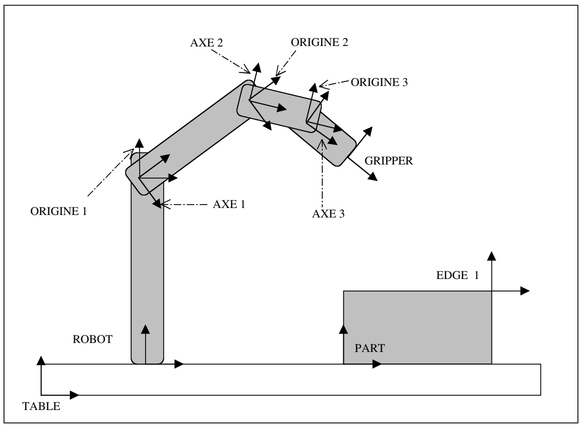

FIG. 2.9 - *Exemple d'un problème simple d'inversion géométrique sous incertitudes géométriques. Ce problème consiste à ramener le point final du bras sur une arête d'un objet. Les longueurs des corps du bras, les zéros des articulations ainsi que les positions de la base ROBOT et de l'objet PART sont entachés d'incertitude.*

Élaboration d'un problème d'inversion géométrique sous incertitudes

La figure 2.9 représente une version probabiliste de l'exemple d'inversion géométrique présenté auparavant. Considérons maintenant les incertitudes sur le modèle du bras (longueurs des corps ainsi que les zéros des articulations), ainsi que sur les poses de la base du bras (repères *ROBOT*) et de l'objet *PART* par rapport à la table (TABLE). Pour modéliser les incertitudes sur les zéros géométriques du bras, nous introduisons (par rapport à la version sans incertitudes de ce problème) trois repères intermédiaires *ORIGINE1*, *ORIGINE2* et *ORIGINE3*.

Notre objectif est de mettre alors le point final du bras *GRIPPER* sur l'arête *EDGE1* en prenant en considération les incertitudes sur le modèle.

L'objectif est spécifié également comme un lien cinématique probabiliste (arc $A_{EDGE1\ GRIPPER}$). Cet arc représente le niveau de contrainte que l'on souhaite avoir sur les paramètres de la pose de *GRIPPER* par rapport à EDGE1. Pour ce cas, le seul paramètre que nous voulons contraindre est la "distance" entre le point final et l'arête que l'on veut ramener à zéro. Cela se traduit par une distribution de probabilité non-uniforme avec un maximum à zéro, et des distributions uniformes sur les paramètres qu'on ne désire pas contraindre.

La figure 2.10 représente le graphe cinématique correspondant à ce problème. L'ensemble des sommets de ce graphe est :

$$\begin{aligned} S = \{ \quad & TABLE, ROBOT, PART, EDGE1, ORIGINE1, AXE1, ORIGINE2, AXE2, \\ & ORIGINE3, AXE3, GRIPPER\} \end{aligned}$$

et l'ensemble des arcs est donné dans le tableau 2.3.

Dans la troisième ligne du tableau 2.3, les égalités $t_x = 150.0, t_y = 30.0, r_z = 0.0$ sur les paramètres de la pose de *EDGE1* par rapport à *PART* représentent le fait qu'on a

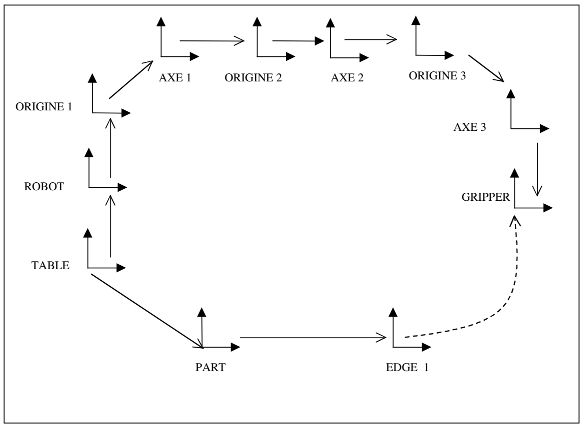

FIG. 2.10 - *Graphe cinématique correspondant au problème d'inversion. L'arc en pointillé correspond à l'objectif fixé.*

supposé que la forme de l'objet est connue sans incertitude. Une incertitude sur cette forme se traduirait par des distributions plus *larges* sur ces paramètres.

Dans la quatrième ligne, les distributions

$$p(t_y) = Normal(l_{1_0}, \sigma_{l_1})$$

$$p(r_z) = Normal(0.0, \sigma_{z_1})$$

sur la pose de *ORIGINE1* par rapport à la base *ROBOT* représentent respectivement les incertitudes gaussiennes sur la longueur du premier corps du robot, et sur le zéro de la première articulation de ce dernier.

Dans la dernière ligne, les ditributions

$$p(t_x) = Uniforme(-150.0, 0)$$

$$p(t_y) = Normal(0.0, \sigma_y)$$

$$p(r_z) = Uniforme(-\pi/2, \pi/2)$$

sur la pose du point final *GRIPPER* par rapport à l'arête *EDGE1*, traduisent respectivement que le point doit être entre les deux bords de l'arête avec une précision de σ_y et que l'orientation est quelconque.

Élaboration d'un problème d'étalonnage de pose en prenant en compte les incertitudes géométriques

Pour cet exemple, nous reprenons le problème d'étalonnage de pose présenté dans la section précédente (figure 2.7) et nous introduisons les informations sur les incertitudes liées au modèle géométrique correspondant. Le but de cet exemple est d'identifier les paramètres de la pose (x, y, α) de l'objet *PART* par rapport à la table *TABLE* en utilisant les mesures capteur.

Arcs	Distributions sur (t_x, t_y, r_z)	Statuts	C. inégalités
$TABLE - ROBOT$	$p(t_x, t_y, r_z) = Normal(\mu_R, \Sigma_R)$	(L, L, L)	
$TABLE - PART$	$p(t_x) = Uniforme(290.0, 310.0)$, $t_y = 10.0, r_z = 0.0$	(L, F, F)	
$PART - EDGE$	$t_x = 150.0, t_y = 30.0, r_z = 0.0$	(F, F, F)	
$ROBOT - ORIGINE1$	$t_x = 0.0, p(t_y) = Normal(l_{1_0}, \sigma_{l_1})$, $p(r_z) = Normal(0.0, \sigma_{z_1})$	(F, L, L)	
$ORIGINE1 - AXE1$	$t_x = 0.0, t_y = 0.0$, $p(r_z) = Uniforme(a_1, b_1)$	(F, F, X)	
$AXE1 - ORIGINE2$	$t_x = 0.0, p(t_y) = Normal(l_{2_0}, \sigma_{l_2})$, $p(r_z) = Normal(0.0, \sigma_{z_2})$	(F, L, L)	
$ORIGINE2 - AXE2$	$t_x = 0.0, t_y = 0.0$, $p(r_z) = Uniforme(a_2, b_2)$	(F, F, X)	
$AXE2 - ORIGINE3$	$p(t_x) = Normal(l_{3_0}, \sigma_{l_3})$, $t_x = 0.0, p(r_z) = Normal(0.0, \sigma_{z_3})$	(F, L, L)	
$ORIGINE3 - AXE3$	$t_x = 0.0, t_y = 0.0$, $p(r_z) = Uniforme(a_3, b_3)$	(F, F, X)	
$AXE3 - GRIPPER$	$p(t_x, t_y, r_z) = Normal(\mu_G, \Sigma_G)$	(L, L, L)	
$EDGE1 - GRIPPER$	$p(t_x) = Uniforme(-150.0, 0)$, $p(t_y) = Normal(0.0, \sigma_y)$, $p(r_z) = Uniforme(-\pi/2, \pi/2)$	(L, L, L)	

TAB. 2.3 - *Distributions, statuts et contraintes inégalités pour le problème d'inversion géométrique sous incertitudes.*

Nous considérons dans cet exemple les incertitudes géométriques liées au modèle du bras (longueurs des corps et zéros des articulations) et à la pose du capteur *LASER* sur le bras. Nous considérons également les incertitudes sur la mesure D_i rendue par le capteur (précision du capteur).

L'introduction d'une distribution sur la pose (x, y, α) permet d'exprimer une estimation *a priori* sur cette pose avant de la confronter aux mesures. L'influence de cet a priori par rapport à la solution dépend, bien entendu :

- du degré de *confiance* que le modélisateur accorde à son estimation *a priori*. Une distribution *étroite* autour d'une valeur nominale traduit un degré de confiance élevé,
- de la précision des mesures qui est liée à la précision de l'outil de mesure (la précision du modèle du bras, la précision de la pose du capteur sur le bras, et la précision de la réponse du capteur).

Le graphe cinématique correspondant à ce problème ainsi que la liste des sommets de ce graphe et les inconnues du problème sont les mêmes que ceux de la version non probabiliste (2.7). L'ensemble des arcs, quant à lui, est donné par le tableau 2.4.

Dans la deuxième ligne de la table 2.4, la distribution

$$p(t_x, t_y, r_z) = Normal(\mu_P, \Sigma_P)$$

sur la pose de l'objet *PART* par rapport à la table *TABLE* représente l'a priori sur les paramètres que l'on souhaite estimer.

Dans la ligne décrivant le lien $LASER_1 - P1$, la distribution

$$p(t_y) = Normal(D_1, \sigma_{LASER})$$

de la pose du point *P1* par rapport au repère $LASER_1$ représente l'a priori sur la réponse du capteur (précision du capteur).

2.4 Application du calcul bayesien pour la construction d'un état de connaissance

À ce stade de notre développement, nous savons construire un graphe cinématique probabiliste résumant le problème posé. L'objectif de cette partie est de montrer comment passer automatiquement de ce graphe à un état de connaissance probabiliste équivalent. Ce passage nous permettra d'utiliser les outils théoriques présentés au début de ce chapitre pour calculer une solution au problème.

Pour ce faire, nous allons reprendre la méthodologie présentée dans la section 2.2, à savoir :

1. La spécification du problème en donnant :
   - les variables du problème,
   - une décomposition de la distribution conjointe sur ces variables,
   - les formes paramétriques des distributions élémentaires issues de la décomposition.
2. L'utilisation de la distribution conjointe issue de la spécification consiste à :
   - poser une question en définissant les rôles des variables pour le problème,
   - inférer (en utilisant les règles des probabilités) la distribution marginale sur l'espace des inconnues,
   - choisir une solution à partir de cette distribution marginale.

2.4.1 Spécification

Variables du problème

La distribution conjointe sera définie sur les variables suivantes :

1. L'ensemble des paramètres (variables continues) des poses relatives entre les n repères (sommets) du graphe cinématique et suivant les m arcs $\{Q_{i_1j_1}, \cdots, Q_{i_mj_m}\}$.
2. Un ensemble de $p = m - n + 1$ variables booléennes (propositions) $\{K_i\}_{i=1}^p$ où p est le nombre minimal de cycles dans le graphe. Ces propositions expriment les contraintes globales de fermeture des cycles définies au début de ce chapitre. Pour chaque cycle c_i, la proposition K_i est définie comme : $K_i \equiv$ "c_i est fermé".

3. Un autre ensemble de m variables booléennes (propositions) $\{H_j\}_{j=1}^{m}$ repésentant les contraintes locales sur les paramètres $\{Q_{i_1j_1}, \cdots, Q_{i_mj_m}\}$. Pour chaque 6-vecteur $Q_{i_kj_k}$, la proposition H_k est définie comme : $H_k \equiv$ "la contrainte $C_k(Q_{i_kj_k}) < 0$ est respectée".

Décomposition de la distribution conjointe - Structure de dépendance

La distribution de probabilité conjointe sur l'ensemble des variables du problème est donnée par :

$$p(Q_{i_1j_1} \cdots Q_{i_mj_m} H_1 \cdots H_m K_1 \cdots K_p)$$

Pour trouver une décomposition de cette distribution conjointe, nous considérons les hypothèses suivantes :

1. Les vecteurs de poses $\{Q_{i_1j_1}, \cdots, Q_{i_mj_m}\}$ sont indépendants. Cela exprime le fait que ces vecteurs ne sont reliés que par les contraintes de fermeture des cycles exprimées par les propositions K_k $k = 1 \cdots p$ assurant la cohérence du modèle. Cela permet d'écrire :
$$p(Q_{i_1j_1} \cdots Q_{i_mj_m}) = p(Q_{i_1j_1}) \cdots p(Q_{i_mj_m}) \tag{2.12}$$
2. Sachant les paramètres de poses $\{Q_{i_1j_1}, \cdots, Q_{i_mj_m}\}$, les propositions K_k $k = 1 \cdots p$ sont mutuellement indépendantes. Cette hypothèse est justifiée car nous ne considérons qu'un ensemble minimal de cycles dans le graphe (pas de redondance dans les cycles considérés).
3. Chaque proposition H_k ne dépend que des paramètres $Q_{i_kj_k}$.
4. Les propositions H_k $k = 1 \cdots m$ sont mutuellement indépendantes car les contraintes locales sont indépendantes les unes des autres. Elles sont aussi indépendantes des K_k $k = 1 \cdots p$.

Nous pouvons alors écrire la distribution conjointe comme suit :

$$\begin{array}{l}
p(Q_{i_1j_1} \cdots Q_{i_mj_m} H_1 \cdots H_m K_1 \cdots K_p) \\
= \\
p(Q_{i_1j_1}) p(Q_{i_2j_2} | Q_{i_1j_1}) \cdots p(Q_{i_mj_m} | Q_{i_1j_1} \cdots Q_{i_{m-1}j_{m-1}}) \\
p(H_1 \cdots H_m | Q_{i_1j_1} \cdots Q_{i_mj_m} K_1 \cdots K_p) \\
p(K_1 \cdots K_p | Q_{i_1j_1} \cdots Q_{i_mj_m}) \\
= \\
p(Q_{i_1j_1}) \cdots p(Q_{i_mj_m}) \\
p(H_1 | Q_{i_1j_1}) \cdots p(H_m | Q_{i_mj_m}) \\
p(K_1 | Q_{i_1j_1} \cdots Q_{i_mj_m}) \cdots p(K_p | Q_{i_1j_1} \cdots Q_{i_mj_m})
\end{array}$$

Formes paramétriques des distributions élémentaires

- Par définition, les propositions $\{K_k\}_{k=1}^{p}$ représentent les contraintes de fermeture des cycles $\{c_k\}_{k=1}^{p}$ du graphe :

$$\forall k = 1 \cdots p, \quad p(K_k | Q_{i_1j_1} \cdots Q_{i_mj_m}) = \begin{cases} 1 & \text{si le cycle } c_k \textit{ est } \text{fermé} \\ 0 & \text{sinon} \end{cases} \tag{2.13}$$

- De la même manière, les propositions $\{H_k\}_{k=1}^m$ sont définies par :

$$\forall k = 1 \cdots p, \quad p(H_k | Q_{i_k j_k}) = \begin{cases} 1 & \text{si } C(Q_{i_k j_k}) \leq 0 \text{ et } O(Q_{i_k j_k}) = 0 \\ 0 & \text{sinon} \end{cases} \tag{2.14}$$

- Pour chaque pose $Q_{i_k j_k}$, $k = 1 \cdots m$, la distribution de probabilité $p(Q_{i_k j_k})$ représente l'a priori sur cette pose. Cette distribution peut exprimer des informations de diverses natures :
  - Spécification du fabricant sur la dimension d'une pièce par exemple (valeur nominale et intervalle de tolérance)
  - Une estimation préalable d'un ensemble de paramètres. C'est le cas, par exemplee, d'un étalonnage d'un bras manipulateur, permettant d'obtenir les valeurs nominales et une matrice de covariances des paramètres du modèle géométrique du bras.
  - Une mesure directe d'une pose relative avec une précision donnée. C'est la cas d'une mesure d'un capteur laser de distance ou bien d'un capteur de vision.
  - Une contrainte que l'on souhaite satisfaire sur un ensemble de paramètres.

Les formes paramétriques que nous avons utilisées dans ce travail sont :

- La distribution gaussienne (normale) monovariée notée $Normal$ ou bien G

$$Normal(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

  où μ et σ sont la moyenne et l'écart-type respectivement.
- La distribution gaussienne (normale) multivariée à n dimensions

$$Normal(X, X_0, \Sigma) = \frac{1}{(2\pi)^{n/2}\sqrt{Det(\Sigma)}} e^{-\frac{1}{2}(X-X_0)^T \Sigma^{-1} (X-X_0)}$$

  où X_0 et Σ sont le vecteur moyen et la matrice de covariances respectivement.
- La distribution uniforme notée $Uniforme$

$$Uniforme(x, a, b) = \frac{1}{b-a}$$

  avec a et b les bornes de la distribution.
- La distribution *Demi gaussienne* représentant une gaussienne associée à une contrainte inégalité

$$HG(x, \mu, \sigma) = \begin{cases} \frac{2}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} & \text{si } x \geq \mu \\ \\ 0 & \text{sinon} \end{cases}$$

L'utilisateur peut étendre le choix des formes paramétriques en définissant d'autres distributions. La définition d'une nouvelle forme (distribution) passe par la définition de sa formule, et éventuellement d'une méthode de tirage direct sur cette distribution (sinon un tirage indirect de Metropolis sera utilisé).

Dans le reste de ce document et dans le but d'alléger les notations, nous nous dispensons d'écrire dans les formules les variables sur lesquelles porte la distribution. Par exemple, on écrira $p(x) = Normal(\mu_x, \sigma_x)$ au lieu d'écrire $p(x) = Normal(x, \mu_x, \sigma_x)$

2.4.2 Utilisation

Question

L'objectif de cette partie est de construire à partir de la distribution conjointe une distribution de probabilité marginale sur les inconnues X du problème. Cette distribution représente une description probabiliste de l'espace des solutions. Elle est donnée par :

$$p(X|H_1 \cdots H_m K_1 \cdots K_p)$$

Inférence

Pour la simplification de la distribution conjointe en exploitant les contraintes $K_j, j = 1 \cdots p$, nous sommes amenés à sélectionner un sous-ensemble de paramètres $\{O_j\}_{j=1}^{p} \subset \{Q_{i_k j_k}\}_{k=1}^{m}$ de la manière suivante :

Reprenons le graphe cinématique et sélectionnons pour chaque cycle $c_j,\ j = 1 \cdots p$ correspondant à une contrainte de fermeture K_j, un de ses liens ne contenant aucun paramètre inconnu (étiqueté à X). Pour le choix de ce lien, deux cas sont possibles :

- Le cycle correspond à un cycle créé lors de la spécification de la tâche. Dans ce cas, le lien à sélectionner correspond à la contrainte ajoutée et que l'on souhaite satisfaire pour résoudre le problème. Ce lien est toujours étiqueté $(LLLLLL)$ avec des distribution plus ou moins étroite sur ses paramètres.

- Le cycle fait partie de la modélisation de la scène. C'est la cas par exemple d'un manipulateur dans la structure est représentée par par un parallélogramme, ou bien d'un objet dont la position par rapport à un autre est calculable suivant deux chemins différents. Dans ce cas, le lien à sélectionner correspond à une contrainte à maintenir. Pour choisir ce dernier, nous sommes amenés à prendre le lien contenant le plus grand nombre de paramètres étiquetés à L. L'idéal serait de trouver un lien étiqueté à $(LLLLLL)$. Si un tel lien n'existe pas, le lien contenant le maximum de paramètres étiquetés à L est choisi. Les paramètres de ce lien étiquetés à F seront étiquetés temporairement à L avec une distribution très étroite (qui tend vers un Dirac) autour des valeurs fixes de ces paramètres. dans la pratique, une distribution gaussienne est prise.

Si l'on renomme l'ensemble des 6-vecteurs de poses correspondant aux liens ainsi obtenus $\{O_j\}_{j=1}^{p}$, cela nous permet d'écrire :

$$
\begin{array}{l}
p(Q_{i_1j_1} \cdots Q_{i_mj_m} H_1 \cdots H_m K_1 \cdots K_p) \\
= \\
p(Q_{i_1j_1}) \cdots p(Q_{i_{m-p}j_{m-p}}) \\
p(O_1) \cdots p(O_p) \\
p(H_1|Q_{i_1j_1}) \cdots p(H_{m-p}|Q_{i_{m-p}j_{m-p}}) p(H_{m-p+1}|O_1) \cdots p(H_m|O_p) \\
p(K_1|Q_{i_1j_1} Q_{i_{m-p}j_{m-p}} O_1 \cdots O_p) \cdots p(K_m|Q_{i_1j_1} Q_{i_{m-p}j_{m-p}} O_1 \cdots O_p)
\end{array}
$$

Considérons maintenant pour chaque 6-vecteur $Q_{i_kj_k}, pour\ k = 1 \cdots m-p$, les statuts des paramètres. Nous définissons alors une partition de ces paramètres en trois sous-ensembles :

1. La liste X des inconnues qui est la concaténation de tous les paramètres dont le statut est *Inconnu*.

2. La liste L des variables dites libres qui est la concaténation de tous les autres paramètres dont le statut est *Libre*.

3. La liste F des variables dont le statut est *Fixe*.

Pour alléger les notations, nous nous permettons de ne pas écrire les variables fixes F dans la formule de la distribution conjointe. Cela est expliqué par le fait que les distributions sur ces paramètres sont des Diracs. Toutefois, les valeurs de ces variables sont utilisées pour écrire les contraintes de fermeture.

En utilisant ces notations, la distribution conjointe est donnée par :

$$
\begin{array}{l}
p(Q_{i_1j_1} \cdots Q_{i_mj_m} H_1 \cdots H_m K_1 \cdots K_p) \\
= \\
p(X)p(L) \\
p(O_1) \cdots p(O_p) \\
p(H_1|XL) \cdots p(H_{m-p}|XL) p(H_{m-p+1}|O_1) \cdots p(H_m|O_p) \\
p(K_1|XLO_1 \cdots O_p) \cdots p(K_m|XLO_1 \cdots O_p)
\end{array}
$$

En utilisant la règle du produit, nous pouvons écrire :

$$
\begin{array}{l}
p(XLO_1 \cdots O_p|H_1 \cdots H_m K_1 \cdots K_p) \\
= \\
\frac{p(Q_{i_1j_1} \cdots Q_{i_mj_m} H_1 \cdots H_m K_1 \cdots K_p)}{p(H_1 \cdots H_m K_1 \cdots K_p)} \\
\propto \\
p(X)p(L) \\
p(O_1) \cdots p(O_p) \\
p(H_1|XL) \cdots p(H_{m-p}|XL) p(H_{m-p+1}|O_1) \cdots p(H_m|O_p) \\
p(K_1|XLO_1 \cdots O_p) \cdots p(K_m|XLO_1 \cdots O_p)
\end{array}
$$

En utilisant la règle de marginalisation sur les variables L et $O_1 \cdots O_p$, on obtient :

$$
\begin{aligned}
& p(X|H_1 \cdots H_m K_1 \cdots K_p) \\
& \qquad \propto \\
& p(X) \int \cdots \int \int \quad dO_1 \cdots dO_p dL \\
& \qquad p(L) \\
& \qquad p(O_1) \cdots p(O_p) \\
& \qquad p(H_1|XL) \cdots p(H_{m-p}|XL) p(H_{m-p+1}|O_1) \cdots p(H_m|O_p) \\
& \qquad p(K_1|XLO_1 \cdots O_m) \cdots p(K_p|XLO_1 \cdots O_p)
\end{aligned}
$$

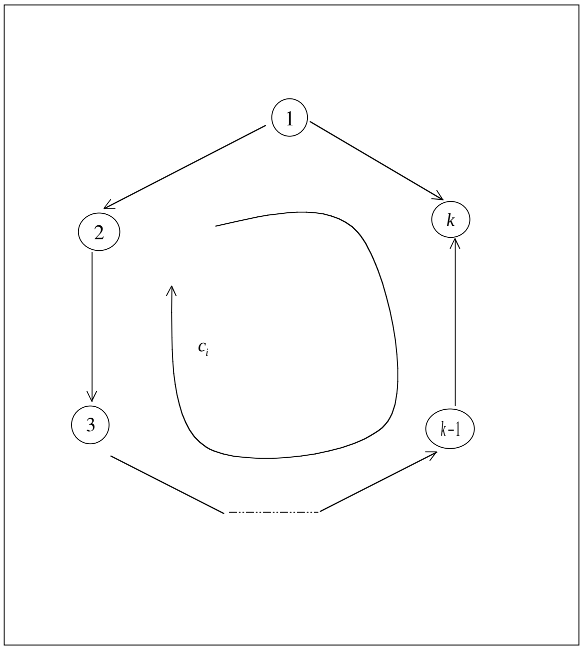

FIG. 2.11 - *Exemple d'un cycle dans un graphe cinématique.*

Considérons un cycle c_j et intéressons nous à la condition de sa fermeture K_j. Si ce cycle est composé de k arcs (figure 2.11), cette condition de fermeture s'exprime en considérant les deux repères successifs 1 et k, par :

$$T_{1k} = T_{12}^{s_{12}} * T_{23}^{s_{23}} * \cdots * T_{k-1\ k}^{s_{k-1k}} \tag{2.15}$$

Ou encore :

$$O_j = Q_{1k} = vect(T_{12}^{s_{12}} * T_{23}^{s_{23}} * \cdots * T_{k-1k}^{s_{k-1k}}) = F_j(Q_{12}, Q_{23}, \cdots, Q_{k-1k}) \tag{2.16}$$

avec $vect$ la fonction permettant d'obtenir le 6-vecteur Q de pose à partir de la matrice 4×4 équivalente et $s_{ij} \in \{1, -1\}$ est le sens du parcours.

Comme les conditions de fermeture $K_j,\ j = 1 \cdots p$ représentent des contraintes égalités (2.16), nous pouvons réécrire cette distribution conjointe en remplaçant chaque vecteur $O_j,\ j = 1 \cdots p$ par la partie droite F_j de l'équation 2.16 équivalente. Cela permet de supprimer les intégrales sur les paramètres $O_j,\ j = 1 \cdots p$:

$$
\begin{aligned}
p(X|H_1 \cdots H_m K_1 \cdots K_p) \propto p(X) \int \quad & dL \\
& p(L) p_{O_1}(F_1(X, L)) \cdots p_{O_p}(F_p(X, L)) \\
& p(H_1|XL) \cdots p(H_{m-p}|XL) \\
& p(H_{m-p+1}|O_1) \cdots p(H_m|O_p)
\end{aligned}
$$

où, rappelons-le, $p_{O_i}(O_i) = p(O_i)$.

Cette suppression des intégrales sur les paramètres $O_j,\ j = 1 \cdots p$ est justifiée par le fait que la valeur de la fonction à intégrer est nulle dans tous les points où les contraintes de fermeture ne sont pas respectées ($p(K_i|, i = 1 \cdots p)$ est nulle). C'est-à-dire que seuls les points respectant ces contraintes contribuent dans le calcul de l'intégrale.

Si l'on dénote pour chaque arc A_{ij} par L_{ij} l'ensemble des paramètres appartenant à L et par X_{ij} l'ensemble des paramètres appartenant à X, on obtient finalement :

$$
\begin{aligned}
p(X|H_1 \cdots H_m K_1 \cdots K_p) \quad \propto \quad & p(X) \\
& \int dL \\
& p(L_{i_1 j_1}) p(H_1|X_{i_1 j_1} L_{i_1 j_1}) \\
\vdots \quad & \\
& p(L_{i_{m-p} j_{m-p}}) p(H_{m-p}|X_{i_{m-p} j_{m-p}} L_{i_{m-p} j_{m-p}}) \\
& p_{O_1}(F_1(X, L)) p(H_{m-p+1}|F_1(X, L)) \\
\vdots \quad & \\
& p_{O_p}(F_p(X, L)) p(H_m|F_p(X, L))
\end{aligned}
$$

Pour chaque lien $A_{i_k j_k}, k = 1 \cdots m - p$, dans le produit $p(L_{i_k j_k}) p(H_k|X L_{i_k j_k})$ le terme $p(L_{i_k j_k})$ représente la distribution *a priori* sur les paramètres libres du lien, tandis que le terme $p(H_k|X L_{i_k j_k})$ représente une *correction* de cet a priori par l'exploitation des contraintes locales sur ces paramètres.

La distribution $p(X)$ est appelée la *distribution a priori* sur X (avant confrontation aux contraintes), tandis que la distribution $p(X|K_1 \cdots K_m H_1 \cdots H_m)$ est appelée la *distribution a posteriori* sur X (après confrontation aux contraintes).

Décision

À ce stade de notre développement, nous disposons de la distribution (fonction objectif) $p(X|H_1 \cdots H_m K_1 \cdots K_p)$ correspondant à un problème donné. La dernière étape de la résolution consiste à choisir une solution X^* à partir de cette distribution. Nous avons choisi de prendre pour solution la valeur de X qui maximise $p(X|H_1 \cdots H_m K_1 \cdots K_p)$:

$$X^* = \max_X p(X|HK) \tag{2.17}$$

avec $H = \{H_i\}_{i=1}^m$ et $K = \{K_i\}_{i=1}^p$.

Comme nous l'avons souligné au début de ce chapitre, ce calcul présente deux difficutés intrinsèquement liées à l'utilisation du calcul bayesien :

1. Le problème de l'intégration (sommation) sur l'espace décrit par les variables L. Cet espace peut être de très grande dimension dans la pratique.

2. Le problème de l'optimisation dans l'espace défini par les variables X. La fonction considérée peut avoir des formes très complexes.

Ces deux problèmes de calcul seront l'objet du chapitre suivant où seront développées les méthodes numériques permettant d'aborder ces problèmes.

2.4.3 Exemple illustratif

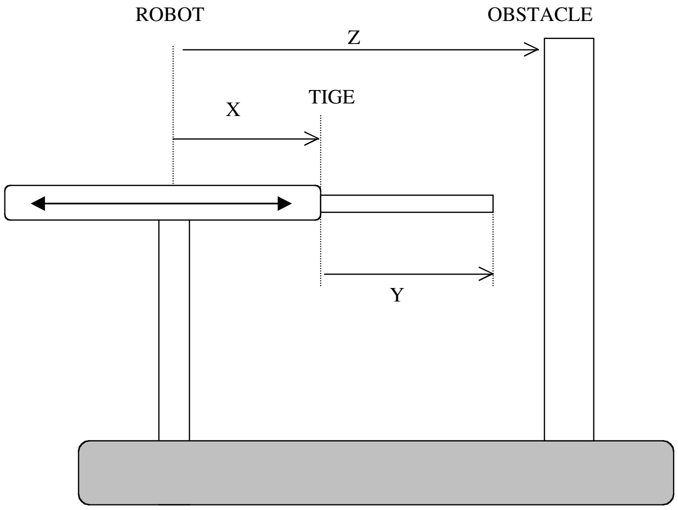

FIG. 2.12 - *Un simple exemple d'inversion géométrique consistant à atteindre un obstacle.*

Pour illustrer, présentons un exemple très simple (figure 2.12) sur lequel nous montrons l'utilisation de la méthode proposée. Il s'agit d'un bras à 1 degré de liberté prismatique x, prolongé par une tige de longueur y. L'objectif est d'atteindre un obstacle à une distance z de la base du bras.

Supposons que notre a priori sur la distance z est une distribution gaussienne d'écart-type (précision) σ_z autour d'une valeur nominale μ_z :

$$p(z) = Normale(\mu_z, \sigma_z) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{1}{2}\frac{(z-\mu_z)^2}{\sigma_z^2}}$$

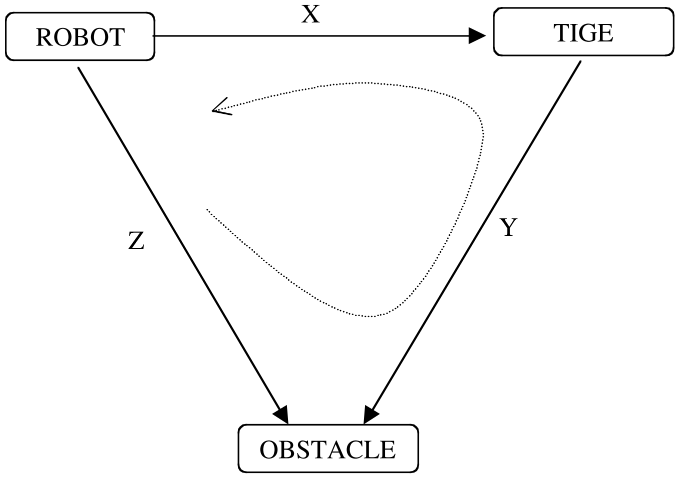

FIG. 2.13 - . *Le graphe cinématique correspondant.*

De même pour la longueur y de la tige. L'a priori sur cette longueur est donné également par une gaussienne :

$$p(y) = Normale(\mu_y, \sigma_y) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{1}{2}\frac{(y-\mu_y)^2}{\sigma_y^2}}$$

Pour la valeur articulaire x, notre a priori pour ce problème représente tout simplement les butées mécaniques. Cet a priori sera exprimé par une distribution uniforme entre ces butées a et b:

$$p(x) = Uniforme(a, b) = \begin{cases} \frac{1}{(b-a)} & \text{si } a \leq x \leq b \\ 0 & \text{sinon} \end{cases}$$

Les statuts des variables x, y et z sont respectivement X, L et L. La contrainte de fermeture du cycle du graphe cinématique 2.13 (modèle géométrique) est dans ce cas :

$$-z + x + y = 0$$

On définie alors la variable K correspondant à cette contrainte de fermeture :

$$p(K) = \begin{cases} 1 & \text{si } -z + x + y = 0 \\ 0 & \text{sinon} \end{cases}$$

Écrivons maintenant la distribution conjointe sur l'ensemble des variables :

$$p(xyzK) = p(x)p(y|x)p(z|xy)p(K|xyz)$$

Comme les variables x, y, z ne sont reliées que par la contrainte de fermeture, on peut écrire :

$$p(xyzK) = p(x)p(y)p(z)p(K|xyz)$$

Notre problème d'inversion consiste à trouver la distribution marginale sur notre degré de liberté x respectant la contrainte de fermeture :

$$p(x|K) = \frac{p(xK)}{p(K)} = \frac{1}{p(K)} \int\int p(xyzK)dzdy$$

Ce qui donne :

$$p(x|K) = \frac{1}{p(K)} \int\int p(x)p(y)p(z)p(K|xyz)dzdy$$

Si l'on choisit la variable z pour la remplacer en exploitant la contrainte de fermeture (z correspond à un lien O), on obtient :

$$\begin{aligned} p(x|K) &\propto p(x) \int p(y)p_z(x+y)dy \\ &= p(x) \int \frac{1}{2\pi\sigma_y\sigma_z} e^{-\frac{1}{2}\frac{(y-\mu_y)^2}{\sigma_y^2}} e^{-\frac{1}{2}\frac{(x+y-\mu_z)^2}{\sigma_z^2}} dy \end{aligned}$$

Si l'on dénote

$$I(x) = \int\int \frac{1}{2\pi\sigma_y\sigma_z} e^{-\frac{1}{2}\frac{(y-\mu_y)^2}{\sigma_y^2}} e^{-\frac{1}{2}\frac{(x+y-\mu_z)^2}{\sigma_z^2}} dy$$

le calcul de cette intégrale analytiquement est possible dans ce cas particulier en utilisant par exemple le changement de variable suivant :

$$y' = \frac{\sqrt{\sigma_y^2+\sigma_z^2}}{\sigma_y\sigma_z}\left(y + \frac{\sigma_y^2}{\sigma_y^2+\sigma_z^2}(x-\mu_z) - \frac{\sigma_z^2}{\sigma_y^2+\sigma_z^2}\mu_y\right)$$

et l'égalité :

$$\int_{-\infty}^{\infty} e^{-y'^2} dy' = \sqrt{\pi}$$

Le résultat est une distribution normale sur x de moyenne $\mu_z - \mu_y$ et d'écart-type $\sqrt{\sigma_y^2+\sigma_z^2}$:

$$I(x) \propto \frac{1}{\sqrt{2\pi}\sqrt{\sigma_y^2+\sigma_z^2}} e^{-\frac{1}{2}\frac{(x-(\mu_z-\mu_y))^2}{\sigma_y^2+\sigma_z^2}}$$

On obtient finalement :

$$p(x|K) \propto \begin{cases} e^{-\frac{1}{2}\frac{(x-(\mu_z-\mu_y))^2}{\sigma_y^2+\sigma_z^2}} & \text{si } a \le x \le b \\ 0 & \text{sinon} \end{cases}$$

Dans le cas où $\mu_z - \mu_y \in [a, b]$, l'optimum de la distribution $p(x|K)$ est obtenu pour $x = \mu_z - \mu_y$. Dans le cas contraire, l'optimum (permettant de s'approcher le mieux possible du but) sera obtenu à $x = a$ si $\mu_z - \mu_y \le a$ et à $x = b$ si $\mu_z - \mu_y \ge b$.

Les figures 2.14, 2.15 et 2.16 donnent la forme de la distribution $p(x|K)$ pour différentes valeurs de a et b avec $\mu_z = 3$ et $\mu_y = 1$

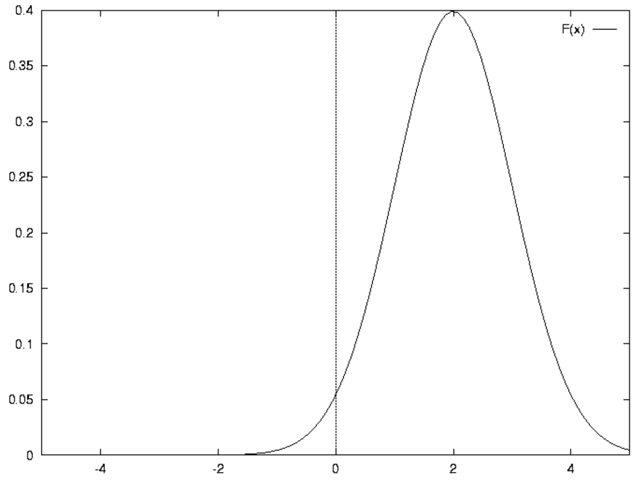

FIG. 2.14 - *Forme de la distribution $p(x|K)$ pour $a = -100.0$ et $b = 100.0$.*

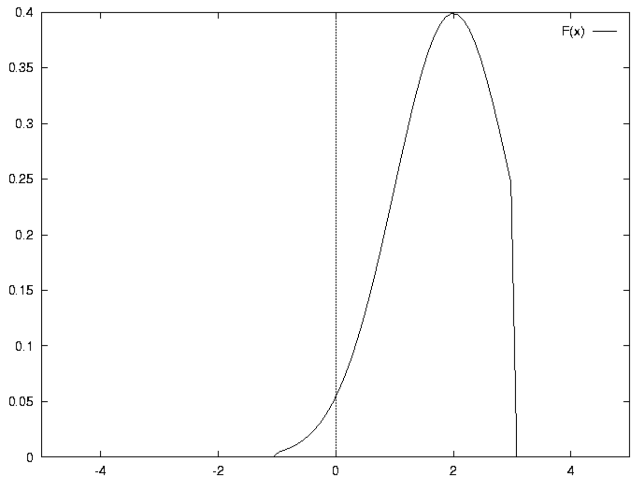

FIG. 2.15 - *Forme de la distribution $p(x|K)$ pour $a = -1.0$ et $b = 3.0$.*

2.5 Conclusion

La méthode de modélisation que nous venons de présenter dans ce chapitre constitue un outil pour la spécification de problèmes géométriques. Cette méthode repose sur la spécification par contraintes probabilistes, et permet de modéliser d'une manière homogène un grand nombre de problèmes géométriques. Citons par exemple : l'inversement géométrique, l'étalonnage de pose à partir de mesures capteur, le positionnement d'objets par contraintes géométriques et la conception d'une cellule robotique. Des exemples complets seront présentés dans le chapitre 4.

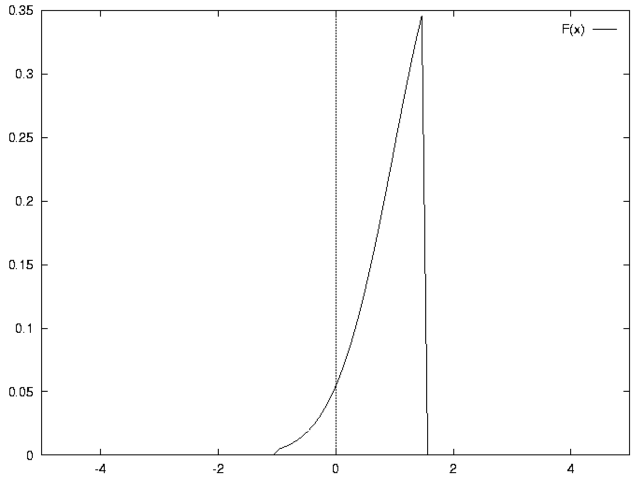

FIG. 2.16 - *Forme de la distribution $p(x|K)$ pour $a = -1.0$ et $b = 1.5$.*

L'utilisation d'un graphe cinématique probabiliste pour modéliser un problème permet de définir automatiquement une distribution de probabilité sur l'espace des solutions. La résolution du problème est donnée par l'optimisation de cette fonction (distribution) dont l'évaluation nécessite le calcul d'une intégrale. La méthode de résolution utilisée (intégration et optimisation) fait l'objet du chapitre suivant.

Arcs	Distributions sur (t_x, t_y, r_z)	Statuts	C. inégalités
$TABLE - ROBOT$	$p(t_x, t_y, r_z) = Normal(\mu_R, \Sigma_R)$	(L,L,L)	
$TABLE - PART$	$p(t_x, t_y, r_z) = Normal(\mu_P, \Sigma_P)$	(X,X,X)	
$PART - EDGE1$	$t_x = 150.0, t_y = 30.0, r_z = 0.0$	(F,F,F)	
$PART - EDGE2$	$t_x = 0.0, t_y = 4, r_z = \pi/2$	(F,F,F)	
$ROBOT - AXE1_1$	$t_x = 0.0, p(t_y) = Normal(l_{1_0}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_1^1, \sigma_{z_1})$	(F,L,L)	
$ROBOT - AXE1_2$	$t_x = 0.0, p(t_y) = Normal(l_{1_0}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_1^2, \sigma_{z_1})$	(F,L,L)	
$ROBOT - AXE1_3$	$t_x = 0.0, p(t_y) = Normal(l_{1_0}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_1^3, \sigma_{z_1})$	(F,L,L)	
$AXE1_1 - AXE2_1$	$t_x = 0.0, p(t_y) = Normal(l_{1_1}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_2^1, \sigma_{z_2})$	(F,L,L)	
$AXE1_2 - AXE2_2$	$t_x = 0.0, p(t_y) = Normal(l_{1_1}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_2^2, \sigma_{z_2})$	(F,L,L)	
$AXE1_3 - AXE2_3$	$t_x = 0.0, p(t_y) = Normal(l_{1_1}, \sigma_{l_1}),$ $p(r_z) = Normal(\theta_2^3, \sigma_{z_2})$	(F,L,L)	
$AXE2_1 - AXE3_1$	$t_x = 0.0, p(t_y) = Normal(l_{1_2}, \sigma_{l_2}),$ $p(r_z) = Normal(\theta_3^1, \sigma_{z_3})$	(F,L,L)	
$AXE2_2 - AXE3_2$	$t_x = 0.0, p(t_y) = Normal(l_{1_2}, \sigma_{l_2}),$ $p(r_z) = Normal(\theta_3^2, \sigma_{z_3})$	(F,L,L)	
$AXE2_3 - AXE3_3$	$t_x = 0.0, p(t_y) = Normal(l_{1_2}, \sigma_{l_2}),$ $p(r_z) = Normal(\theta_3^3, \sigma_{z_3})$	(F,L,L)	
$AXE3_1 - LASER_1$	$p(t_x, t_y, r_z) = Normal(\mu_L, \Sigma_L)$	(L,L,L)	
$AXE3_2 - LASER_2$	$p(t_x, t_y, r_z) = Normal(\mu_L, \Sigma_L)$	(L,L,L)	
$AXE3_3 - LASER_3$	$p(t_x, t_y, r_z) = Normal(\mu_L, \Sigma_L)$	(L,L,L)	
$LASER_1 - P1$	$t_x = 0.0, p(t_y) = Normale(D_1, \sigma_{LASER}),$ $r_z = 0.0$	(F,L,F)	
$LASER_2 - P2$	$t_x = 0.0, p(t_y) = Normale(D_2, \sigma_{LASER}),$ $r_z = 0.0$	(F, L,F)	
$LASER_3 - P3$	$t_x = 0.0, p(t_y) = Normale(D_3, \sigma_{LASER}),$ $r_z = 0.0$	(F,L,F)	
$EDGE1 - P1$	$p(t_x) = Uniforme(-150, 0.0), t_y = 0.0,$ $p(r_z) = Uniforme(-\pi/2, \pi/2)$	(L,F,L)	
$EDGE1 - P2$	$p(t_x) = Uniforme(-150, 0.0), t_y = 0.0,$ $p(r_z) = Uniforme(-\pi/2, \pi/2)$	(L,F,L)	
$EDGE2 - P3$	$p(t_x) = Uniforme(-20, 20.0), t_y = 0.0,$ $p(r_z) = Uniforme(-\pi/2, \pi/2)$	(L,F,L)	

TAB. 2.4 - *Statuts, distributions et contraintes inégalités pour le problème d'étalonnage.*

Chapitre 3

Méthode de résolution

Dans le chapitre précédent, nous avons montré comment construire un graphe cinématique probabiliste à partir de la spécification d'un problème de géométrie. Puis, nous avons montré comment utiliser ce graphe pour se ramener à un double problème d'intégration et d'optimisation :

$$X^* = \max_X [p(X|H_1 \cdots H_m K_1 \cdots K_p)] \tag{3.1}$$

avec :

$$\begin{aligned}
p(X|H_1 \cdots H_m K_1 \cdots K_p) \quad \propto \quad & p(X) \\
& \int dL \\
& p(L_{i_1 j_1}) p(H_1 | X_{i_1 j_1} L_{i_1 j_1}) \\
\vdots \quad & \\
& p(L_{i_{m-p} j_{m-p}}) p(H_{m-p} | X_{i_{m-p} j_{m-p}} L_{i_{m-p} j_{m-p}}) \\
& p_{O_1}(F_1(X,L)) p(H_{m-p+1} | F_1(X,L)) \\
\vdots \quad & \\
& p_{O_p}(F_p(X,L)) p(H_m | F_p(X,L))
\end{aligned}$$

où :

- X est le vecteur représentant les paramètres inconnus du problème.
- L est un vecteur représentant la concaténation de tous les paramètres *libres* $\{L_{i_k j_k}\}_{k=1}^{m-p}$

Si l'on dénote $G(X) = p(X|H_1 \cdots H_m K_1 \cdots K_p)$, la résolution d'un tel problème nécessite l'optimisation de la fonction $G(X)$ dont l'évaluation pour une valeur donnée de X nécessite le calcul d'une intégrale sur l'espace L. Cet espace peut être de très grande dimension (des dizaines) dans des problèmes réalistes de CAO.

L'objectif de ce chapitre est de présenter les méthodes nécessaires au calcul de cette intégrale, ainsi que les méthodes d'optimisation permettant de maximiser la fonction objectif ainsi obtenue. Nous verrons qu'une des particularités de notre méthode est de conduire l'optimisation sur des approximations de plus en plus fines de la fonction objectif.

Ce chapitre comporte deux parties :

Dans la première, nous commençons par aborder le problème du calcul des intégrales. Nous présentons en particulier les méthodes de Monte-Carlo comme des méthodes d'intégration numérique qui s'adaptent bien aux problèmes de grandes dimensions. Nous expliquerons comment nous utilisons ces méthodes en conjonction avec le graphe cinématique pour obtenir une approximation de la fonction objectif à maximiser.

La deuxième partie sera consacrée, quant à elle, au problème d'optimisation. Nous expliquerons en particulier la spécificité de ce type de problèmes où la valeur de la fonction objectif ne peut être qu'approximée (parce qu'on utilise une estimation des intégrales). Nous expliquerons également les problèmes relatifs à la forme de la fonction objectif. Nous décrirons par la suite l'algorithme génétique qui sera utilisé pour résoudre le problème d'optimisation. Nous montrerons les avantages d'un tel algorithme pour ce type de problème.

3.1 Problème du calcul des intégrales

L'utilisation du calcul intégral constitue la base des applications du calcul bayesien. Malheureusement, les méthodes analytiques d'évaluation des intégrales se révèlent souvent trop limitées pour les problèmes rencontrés dans la pratique où les fonctions impliquées peuvent avoir des formes complexes et les espaces d'intégration peuvent être de grande dimension.

Une solution à ce problème est de calculer numériquement des approximations de ces intégrales. Plusieurs méthodes de calcul numérique des intégrales sont envisageables. Leur fiabilité dépend essentiellement de la forme des fonctions à intégrer et de la dimension de l'espace d'intégration.

Nous présenterons les principales méthodes d'intégration numérique. Nous les comparerons en termes des erreurs d'approximation induites par chaque méthode. Cette comparaison dépend de la dimension de l'espace d'intégration et du nombre de points utilisés pour calculer l'approximation.

Les méthodes d'intégration numérique peuvent être classifiées en deux grandes catégories :

- Les méthodes déterministes basées sur la subdivision du domaine d'intégration.
- Les méthodes stochastiques dites de Monte-Carlo.

3.1.1 Méthodes basées sur la subdivision du domaine d'intégration

Principes

Soit l'intégrale :

$$I = \int_a^b f(x)dx \tag{3.2}$$

L'idée de ces méthodes est de subdiviser le domaine $[a, b]$ en n sous-intervalles $h = \frac{(b-a)}{n}$ et faire des hypothèses sur la forme de f à l'intérieur de ces intervalles pour donner une estimation de l'intégrale. Parmi ces méthodes, on peut citer par exemple :

- **Méthode du Trapèze**

  L'approximation pour cette méthode est donnée par :

$$I \simeq \frac{h}{2}\left[f(a) + 2\sum_{k=0}^{n-1} f(a+kh) + f(b)\right] \tag{3.3}$$

- **Méthode de Simpson**

  L'approximation est donnée par :

$$I \simeq \frac{h}{3}\left[f(a) + 4\sum_{k=1}^{n/2} f(a+(2k-1)h) + 2\sum_{k=1}^{\frac{n-2}{2}} f(a+2kh) + f(b)\right] \tag{3.4}$$

L'erreur d'estimation

Dans le cas d'un domaine d'intégration à une seule dimension, on peut démontrer que les méthodes d'intégration basées sur la subdivision de domaine donnent, dans le cas général, une estimation de l'intégrale avec une erreur qui varie comme h^2 (elle varie comme h^4 pour la méthode de Simpson). L'erreur d'estimation varie alors comme $\frac{1}{N^2}$ où N est le nombre de points utilisés pour cette estimation. Dans le cas d'un espace d'intégration à d dimensions, si l'on suppose que le nombre de pas de subdivision est N pour toutes les dimensions, l'erreur d'estimation variera alors comme $\frac{1}{N^{2/d}}$.

3.1.2 Méthodes de Monte-Carlo

Nous présentons dans cette partie les méthodes stochastiques dites de *Monte-Carlo*, pour l'estimation des intégrales. Ces méthodes ont été largement utilisées en physique et en économie (voir [Neal93] et [Geweke95] par exemple) et trouvent ces dernière années d'autres domaines d'applications comme l'image de synthèse [Keller96] et la robotique mobile [Dellaert99, Fox99], à titre d'exemples.

Nous nous intéressons dans cette partie aux fondements théoriques de ces méthodes et à leur application pour les problèmes d'intégration dans les espaces de grandes dimensions.

Principes

La solution la plus simple qu'on peut imaginer pour calculer une intégrale

$$I = \int p(x)g(x)\ d^d x \tag{3.5}$$

dans un espace à d dimensions, est de faire un tirage uniforme sur l'espace d'état x. Cela consiste à générer un ensemble de points $\{x^{(i)}\}_{i=1}^{N}$ uniformément distribués dans cet espace et estimer I par $\hat{I}$:

$$\hat{I} = \frac{1}{N} \sum_i p(x^{(i)}) g(x^{(i)}) \tag{3.6}$$

En général, les distributions de probabilité sur les espaces de grandes dimensions sont concentrées dans une petite région T de l'espace d'état. Par exemple, si l'espace d'état x contient un grand nombre de variables approximativement indépendantes, le volume $|T|$ de T sera donné par $|T| \simeq 2^{H(X)}$, où $H(X)$ est l'entropie de Shannon-Gibbs $H(X) = \sum_x p(x) \log_2 \frac{1}{p(x)}$ (voir [MacKay96]). Le nombre N de points tirés uniformement de l'espace d'état doit être suffisamment grand pour couvrir la région T où se trouve la plus grande partie de la masse de la distribution $p(x)$:

$$N \propto \frac{|X|}{|T|} \tag{3.7}$$

où $|X|$ est le volume de l'espace d'état.

À partir de là, l'exploration de l'espace d'intégration avec un tirage uniforme s'avère, dans le cas général, très coûteuse. Les méthodes moins coûteuses sont indispensables. Le principe de ces méthodes dites de Monte-Carlo, est d'approximer l'intégrale 3.5 en estimant l'espérance de la fonction $g(x)$ sous la distribution $p(x)$:

$$I = \int p(x) g(x) \; d^d x = \langle g(x) \rangle \tag{3.8}$$

$p(x)$ est la distribution de probabilité sur le d-vecteur x.

Supposons que l'on soit capable de générer un ensemble de tirages $\{x^{(i)}\}_{i=1}^N$ à partir de la distribution $p(x)$. Une manière de calculer cette intégrale est d'utiliser ces tirages (vecteurs) pour donner l'estimateur

$$\hat{I} = \frac{1}{N} \sum_i g(x^{(i)}) \tag{3.9}$$

L'erreur d'estimation

Si les vecteurs $\{x^{(i)}\}_{i=1}^N$ sont engendrés à partir de $p(x)$, la variance de l'estimateur $\hat{I} = \frac{1}{N} \sum_i g(x^{(i)})$ va diminuer comme $\frac{\sigma^2}{N}$ où σ^2 est la variance de g :

$$\sigma^2 = \int p(x) (g(x) - \hat{g})^2 \; d^d x \tag{3.10}$$

où $\hat{g}$ est la moyenne de g.

Ce résultat représente une des importantes propriétés des méthodes de Monte-Carlo :

La précision de l'estimation par les méthodes de Monte Carlo est indépendante de la dimension de l'espace d'intégration

Cette propriété explique l'avantage de l'utilisation de ces méthodes pour les espaces de grande dimension ($d > 4$).

Pour donner l'estimateur 3.9, ces méthodes supposent la capacité de générer, d'une manière ou d'une autre, des points à partir de la densité $p(x)$. Mais les questions suivantes se posent :

1. est-il toujours possible de tirer une valeur de x à partir de la définition de $p(x)$?

2. existe-t-il une manière simple de tirer selon cette distribution ?

La réponse à ces questions est non. Le tirage suivant une distribution $p(x)$ reste encore un problème très difficile, surtout pour les espaces de grande dimension. Il n'existe que quelques densités de probabilité de grande dimension à partir desquelles on peut directement générer des tirages. C'est le cas des distributions uniformes et Gaussiennes par exemple.

Une présentation des principales méthodes de tirage de Monte-Carlo est donnée dans l'annexe B à la fin de ce document.

3.2 Utilisation des méthodes de Monte-Carlo

La spécification d'un problème réaliste de robotique ou de CAO robotique nécessite en général l'utilisation d'un grand nombre de référentiels (repères). En conséquence, la dimension de l'espace défini par l'ensemble L des variables dites *libres* peut être très grande. Un problème réaliste peut contenir des dizaines voire des centaines de variables libres.

La méthode de résolution proposée nécessitant le calcul d'une intégrale sur cet espace, la méthode utilisée doit forcément répondre à deux préoccupations :

- La dimension de l'espace d'intégration.

- La limitation du domaine d'intégration aux régions de l'espace où les contraintes locales sont respectées (Les propositions $\{H_i\}_{i=1}^{m-p}$ ont la valeur *vrai*).

Ces deux préoccupation rendent le calcul analytique difficilement applicable en général. Nous avons donc choisi de calculer numériquement ces intégrales pour appliquer notre méthode à des domaines plus larges d'applications. L'utilisation d'une méthode d'intégration basée sur la subdivision du domaine d'intégration engendrerait un coût de calcul croissant exponentiellement avec la dimension de l'espace d'intégration. Les méthodes de Monte-Carlo, quant à elles, s'adaptent mieux aux problèmes posés du fait que le nombre de points utilisés ne dépend pas de la dimension du domaine d'intégration. Nous avons donc choisi ce type de méthodes et nous présentons leurs mises en œuvre dans notre système. Nous montrons en particulier comment l'utilisation de ces méthodes peut répondre au problème des contraintes locales sur l'espace d'intégration.

3.2.1 Principe

Rappelons la forme de l'intégrale à calculer :

$$I(X) \quad = \quad \int dL$$

$$
\begin{array}{l}
p(L_{i_1 j_1}) p(H_1 | X_{i_1 j_1} L_{i_1 j_1}) \\
\vdots \\
p(L_{i_{m-p} j_{m-p}}) p(H_{m-p} | X_{i_{m-p} j_{m-p}} L_{i_{m-p} j_{m-p}}) \\
p_{O_1}(F_1(X, L)) p(H_{m-p+1} | F_1(X, L)) \\
\vdots \\
p_{O_p}(F_p(X, L)) p(H_m | F_p(X, L))
\end{array}
$$

L'utilisation d'une méthode de Monte-Carlo pour estimer cette intégrale nécessite :

1. La capacité de générer (tirer) un ensemble de N points $\{L^{(i)}\}_{i=1}^{N}$ à partir de la distribution *a priori* $p(L)$ où chaque point conduit à un calcul respectant l'ensemble des contraintes inégalités $\{C_i\}_{i=1}^{m-p}$. Pour les contraintes locales d'égalité, pour la plupart des relations de contact (sauf *Point sur Cylindre* et *Point sur Sphere* voir Annexe A), une contrainte égalité porte sur un seul paramètre à la fois ce qui permet de prendre en compte cette contrainte en fixant tout simplement ce paramètre. Pour simplifier, nous allons considérer seulement ce cas.

2. Le calcul de l'estimation de l'intégrale $I(X)$ à partir de l'ensemble $\{L^{(i)}\}_{i=1}^{N}$. Ce calcul se fait comme suit :

$$
\begin{array}{rcl}
\hat{I}(X) & = & \frac{1}{N} \sum_{i}^{N} \\
& & p_{O_1}(F_1(X, L^{(i)})) p(H_{m-p+1} | F_1(X, L^{(i)})) \\
& \vdots & \\
& & p_{O_p}(F_p(X, L^{(i)})) p(H_m | F_p(X, L^{(i)}))
\end{array}
$$

Tirage de l'ensemble de points

La génération de l'ensemble des N points qui seront utilisés dans l'estimation peut être effectuée de diverses manières. Si l'on appelle S le *vecteur d'état* du système, la génération d'un nouvel état (point) $S^{(t+1)}$ à partir de l'état précédent $S^{(t)}$

$$
S^{(t)} = (S_1^{(t)}, S_2^{(t)}, \cdots, S_l^{(t)})
$$

où l est la dimension de S, nécessite :

– soit de mettre à jour toutes les composantes de S simultanément (*algorithme global*)

$$
S^{(t+1)} = (S_1^{(t+1)}, S_2^{(t+1)}, \cdots, S_l^{(t+1)})
$$

– soit de mettre à jour une seule composante S_i à la fois (*algorithme local*)

$$
S^{(t+1)} = (S_1^{(t)}, S_2^{(t)}, \cdots, S_i^{(t+1)}, \cdots, S_l^{(t)})
$$

Ces composantes $S_i, i = 1 \cdots l$ peuvent être elles mêmes multidimensionnelles.

Pour que l'utilisation d'une méthode de tirage locale soit fiable, il est préférable de trouver une décomposition pour laquelle les composantes $S_i, i = 1 \cdots l$ du vecteur d'état sont approximativement indépendantes. La dépendance entre ces composantes rend plus difficile l'exploration de l'espace d'intégration (voir [Neal93, Geweke95]). Dans les méthodes locales, l'état $S^{(t+1)}$ dépend forcément de l'état $S^{(t)}$. On parle alors dans ce cas de méthodes Markoviennes abrégées généralement MCMC pour *Markov Chain Monte Carlo* [MacKay96].

Dans notre cas, le vecteur L (concaténation des $L_{i_1 j_1}, \cdots L_{i_{m-p} j_{m-p}}$) est le vecteur d'état. Comme les paramètres appartenant à différents liens cinématiques sont indépendants, la décomposition du vecteur L en $m-p$ composantes $\{L_{i_k j_k}\}_{k=1}^{m-p}$ dans le but d'utiliser une méthode locale est simple. La mise à jour du vecteur d'état L

$$L^{(t)} = (L^{(t)}_{i_1 j_1}, L^{(t)}_{i_2 j_2}, \cdots, L^{(t)}_{i_k j_k}, \cdots, L^{(t)}_{i_{m-p} j_{m-p}})$$

nécessite la mise à jour d'une seule composante $L_{i_k j_k}$ de la manière suivante :

$$L^{(t+1)} = (L^{(t)}_{i_1 j_1}, L^{(t)}_{i_2 j_2}, \cdots, L^{(t+1)}_{i_k j_k}, \cdots, L^{(t)}_{i_{m-p} j_{m-p}})$$

N itérations de cette procédure permet de générer l'ensemble $\{L^{(i)}\}_{i=1}^{N}$ qui sera utilisé pour l'estimation de l'intégrale.

Intéressons nous maintenant à la mise à jour d'une composante $L_{i_k j_k}$ (ensemble de paramètres appartenant au même vecteur de position). Les paramètres appartenant à $L_{i_k j_k}$ ne sont pas nécessairement indépendants :

- Le modélisateur a pu vouloir exprimer explicitement des éventuelles corrélations entre les paramètres du vecteur $L_{i_k j_k}$. Dans le cas le plus général la distribution sur le vecteur de position est définie comme une nappe dans $\mathbb{R}^6$. C'est en particulier le cas d'une mesure directe (avec un instrument de mesure donnant des valeurs corrélées), ou bien d'une mesure indirecte (résultat d'une calibration préalable par exemple).
- Les paramètres du vecteur $L_{i_k j_k}$ sont corrélés implicitement par une contrainte inégalité $C(Q_{i_k j_k}) \leq 0$ induite par une relation de contact entre deux entités géométriques E_i et E_j. Ces contraintes locales (sur les paramètres d'un seul vecteur de position) représentent une *correction* de la distribution initiale $p(L_{i_k j_k})$ en réduisant le domaine de variation de ces paramètres.

Par conséquent, pour mettre à jour une composante $L_{i_k j_k}$, on doit résoudre deux problèmes :

- **Le tirage d'un point candidat :**
  Un candidat $L^c_{i_k j_k}$ est tiré à partir de $p(L_{i_k j_k})$. Pour effectuer ce tirage, deux cas de figure sont possibles :
  - Nous disposons d'une méthode de tirage directe à partir de $p(L_{i_k j_k})$ (cas des distributions uniformes et Gaussiennes).
  - Nous ne disposons pas de méthode de tirage directe à partir de $p(L_{i_k j_k})$. Dans ce cas, une méthode de tirage indirecte est utilisée. Pour un besoin de généralité, la méthode de tirage indirecte que nous avons choisie dans ce travail est l'algorithme de Metropolis (voir l'annexe B).

- **Test de la validité du candidat :**
  Un calcul géométrique qui dépend du type de contact entre E_i et E_j est fait en utilisant le point candidat. Ce calcul permet de vérifier la contrainte $C(Q_{i_k j_k}) \leq 0$ ainsi calculée avec $k = 1 \cdots m - p$. Le candidat $L^c_{i_k j_k}$ sera accepté si cette contrainte n'est pas violée ($p(H_k | X L_{i_k j_k}) = 1$). Dans le cas contraire, il sera rejeté. L'acceptation du point candidat engendre la modification du vecteur d'état L et le rejet du point laisse le vecteur d'état inchangé.

  Nous rappelons que ce test de la validité du candidat porte seulement sur les contraintes $H_k, k = 1 \cdots m - p$. Cela permet, pour chacune de ces contraintes H_k, un calcul local ne portant que sur les paramètres L_k et X_k.

Cette méthode de tirage peut être vue comme un cas particulier du tirage par rejet (présenté dans l'annexe B). Nous appelons cette méthode de tirage *un tirage par rejet catégorique* parce que le candidat est soit accepté avec une probabilité de 1, soit rejeté avec une probabilité de 1 également.

La figure 3.1 représente un exemple de relations *face sur face*. La contrainte locale induite par cette relation correspond à ce que tous les points de la face de dessus ne dépassent pas les bornes de la face de dessous.

Les formules des contraintes liées à différentes relations de contact sont données dans l'annexe A.

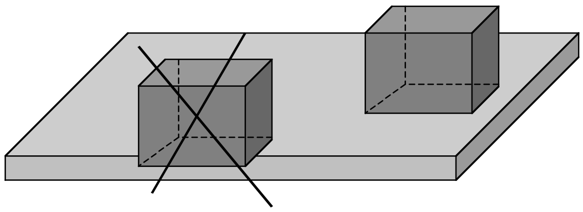

FIG. 3.1 - *Principe du tirage par rejet catégorique : le point est rejeté parce qu'il viole la contrainte* Face sur Face *entre les deux objets.*

3.2.2 Optimisation des calculs

L'utilisation d'une méthode de tirage locale pour la mise à jour du vecteur d'état L permet de réduire le coût du calcul de l'estimation. Si l'on dénote pour un point donné $L^{(t)}$ $F_i^{(t)}(X)$, $i = 1 \cdots p$ les valeurs des fonctions $F_i(X)$, alors les valeurs des $F_i(X)$ à l'étape suivante $F_i^{(t+1)}(X)$ peuvent être obtenues par une mise à jour des $F_i^{(t)}(X)$. Le coût de la mise à jour dépend, bien entendu, de la position de l'arc correspondant à la composante $L_{i_k j_k}$ qui vient d'être changée dans les cycles c_i, $i = 1 \cdots p$. En particulier, la mise à jour d'un arc ne modifie que les valeurs des fonctions correspondant aux cycles qui contiennent l'arc en question.

Les algorithmes de mise à jour sont présentés en annexe C.

3.3 Méthode d'optimisation

Nous avons présenté dans la première partie de ce chapitre la méthode utilisée pour l'estimation des intégrales présentes dans la fonction objectif représentant un problème donné. Cette deuxième partie sera consacrée à la méthode d'optimisation qui nous permettra de trouver l'optimum de cette fonction.

Supposons que l'on s'intéresse au problème d'optimisation suivant :

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ \max_x f(x) \end{aligned} \tag{3.11}$$

On peut classifier les méthodes de résolution du problème 3.11 en deux grandes catégories :

- Les méthodes locales.
- Les méthodes globales.

3.3.1 Les méthodes locales

Le principe de ces méthodes est de partir d'un point initial donné x_0 et d'essayer de converger à chaque itération vers un maximum local.

Classiquement le schéma itératif de recherche de cet optimum est de la forme :

$$x_{n+1} = x_n + \Delta x_n$$

où l'incrément Δx_n peut être évalué de multiples manières.

Pour la méthode du gradient par exemple, l'incrément Δx_n est calculé comme suit :

$$\Delta x_n = \alpha \frac{\partial f(x_n)}{\partial x}$$

où α est un réel positif.

L'utilisation de la plupart de ces méthodes suppose l'existence des dérivées (la dérivée de l'ordre 1 et peut être plus) de la fonction $f(x)$. Ces algorithmes ne permettent, par définition, que de trouver des optima locaux qui dépendent du point initial x_0.

3.3.2 Les méthodes globales

L'objectif d'un algorithme d'optimisation global est de trouver un optimum global au problème 3.11. Ces algorithmes sont souvent stochastiques. Leur principe général est que la probabilité de se trouver dans une configuration x soit proportionnelle à la valeur de $f(x)$ en ce point.

$$p(x) \propto f(x) \tag{3.12}$$

La solution finale ne dépend pas, en principe, du (ou des) point (s) de départ x_0.

On peut citer parmi ces méthodes :

- Les méthodes de Monte Carlo (Metropolis) [Neal93].

- Les méthodes du recuit simulé [Neal93, Laarhoven87, Kirkpatrick83].
- Les algorithmes génétiques [Holland75].

3.3.3 Critères de choix de la méthode d'optimisation

La méthode d'optimisation que nous allons utiliser pour ce type de problème, doit répondre à un ensemble de critères liés à la forme et à la nature des fonctions à optimiser. Cette méthode doit être :

1. Globale : la distribution objectif est souvent multimodale. Une méthode locale peut échouer en ne donnant qu'un maximum local de la fonction.
2. Ne nécessitant pas les dérivées : la fonction objectif peut être non dérivable.
3. Permettant un calcul multiprécision de la fonction objectif : le calcul de la fonction objectif avec une grande précision peut être très coûteux en temps de calcul. On voudrait pouvoir commencer l'optimisation avec des approximations plus ou moins *grossières* de la fonction à optimiser, et affiner l'approximation au fur et à mesure du processus d'optimisation.
4. Donnant la possibilité de parallélisation pour augmenter les performances de la méthode pour les problèmes où le coût de la fonction objectif peut être très élevé.

Nous avons opté dans notre application pour un algorithme de type *Génétique*, parce qu'il répond justement à ces critères. Dans ce qui suit, nous commencerons par présenter les principes généraux de ce type d'algorithmes. Nous présenterons ensuite les problèmes pratiques liés à l'utilisation de ces algorithmes pour notre application, nous donnerons des éléments de solution pour ces problèmes et nous finirons par présenter l'algorithme implanté dans notre système.

3.3.4 Principes des algorithmes génétiques

Les algorithmes génétiques (désormais abrégés AG) sont une technique d'optimisation stochastique inspirée de l'évolution biologique des espèces. Cette technique a été introduite par Holland [Holland75] au début des années 70 et elle est souvent utilisée dans l'optimisation globale d'une fonction F sur un espace de recherche S souvent très grand.

Vu la facilité relative de la mise en œuvre de ces algorithmes et leur indépendance vis-à-vis du domaine d'application, ces méthodes ont été largement utilisées pour les problèmes d'optimisation en intelligence artificielle [Grefenstette88, Kenneth88] et en robotique [Ahuactzin94, Falkenauer91].

L'utilisation d'un AG commence par une phase d'initialisation. Un ensemble de points que l'on appellera *population initiale d'individus* est généré aléatoirement dans l'espace de recherche S. Un individu représente une solution au problème, et son *adéquation* peut être mesurée par la valeur de la fonction F en ce point. Le principe fondamental des algorithmes génétiques est : "meilleure est l'adéquation d'un individu, plus sa probabilité de sélection est grande". Les individus sélectionnés se reproduisent, c'est-à-dire se voient appliquer des "opérateurs génétiques" afin de générer de nouveaux individus. La taille de la population étant donnée, les meilleurs de ces nouveaux individus viennent remplacer les plus mauvais. Cela induit donc un phénomène de compétition entre les individus.

Le processus génétique est réitéré sur la nouvelle population jusqu'à un certain critère d'arrêt. L'algorithme génétique séquentiel standard est le suivant : On génère aléatoirement une population initiale d'individus, puis on exécute le cycle suivant (Figure 3.2):

1. **Sélection :** Au moyen de la fonction F, des paires d'individus sont sélectionnées. La probabilité de sélection d'un individu I croit avec la valeur de F en ce point.

2. **Reproduction :** Des opérateurs génétiques sont appliqués aux paires d'individus sélectionnées, pour générer de nouveaux individus.

3. **Évaluation :** La fonction F est calculée pour les individus venant d'être créés.

4. **Remplacement :** Certains individus de la population sont remplacés par les nouveaux individus (plus adéquats). Une nouvelle population est alors ainsi créée.

Les solutions (configurations) sont souvent codées sous la forme de chaînes de bits appelées *génotypes*, qui constituent alors les individus en compétition.

Les opérateurs génétiques les plus utilisés durant la reproduction sont le "croisement" et la "mutation".

Étant donnés deux individus, l'opérateur de croisement consiste à engendrer un nouvel individu en combinant les génotypes de ces individus. Une des méthodes possible pour faire cette combinaison est de couper les deux génotypes en un point choisi aléatoirement, et produire deux nouveaux individus après avoir échangé les parties coupées.

Étant donné un individu, l'opérateur de mutation change aléatoirement l'état de certains bits de l'individu.

On réitère ce cycle sur plusieurs générations, jusqu'à l'obtention d'une génération de solutions suffisamment stables (un critère de convergence peut être défini). La solution retenue finalement est le meilleur individu. Toutefois, l'étude de la population finale entière est souvent riche en enseignements car elle peut contenir des solutions alternatives au problème posé.

3.4 Utilisation des algorithmes génétiques

Nous avons présenté dans la section précédente les principes des AG comme des méthodes d'optimisation globale. Nous expliquerons dans cette partie les problèmes liés à l'utilisation de tels algorithmes pour l'optimisation des fonctions représentant nos problèmes de géométrie, pour finir par présenter l'algorithme général que nous proposons.

Problèmes liés à la forme de la fonction objectif

Pour les problèmes rencontrés dans la pratique, la fonction $G(X)$ à optimiser peut avoir un support (domaine où la valeur est non nulle) très étroit pour les problèmes fortement contraints. Le fait d'initialiser la population avec des individus pouvant être très loin des *solutions*, peut engendrer des valeurs très faibles (souvent nulles) de la fonction objectif $G(X)$.

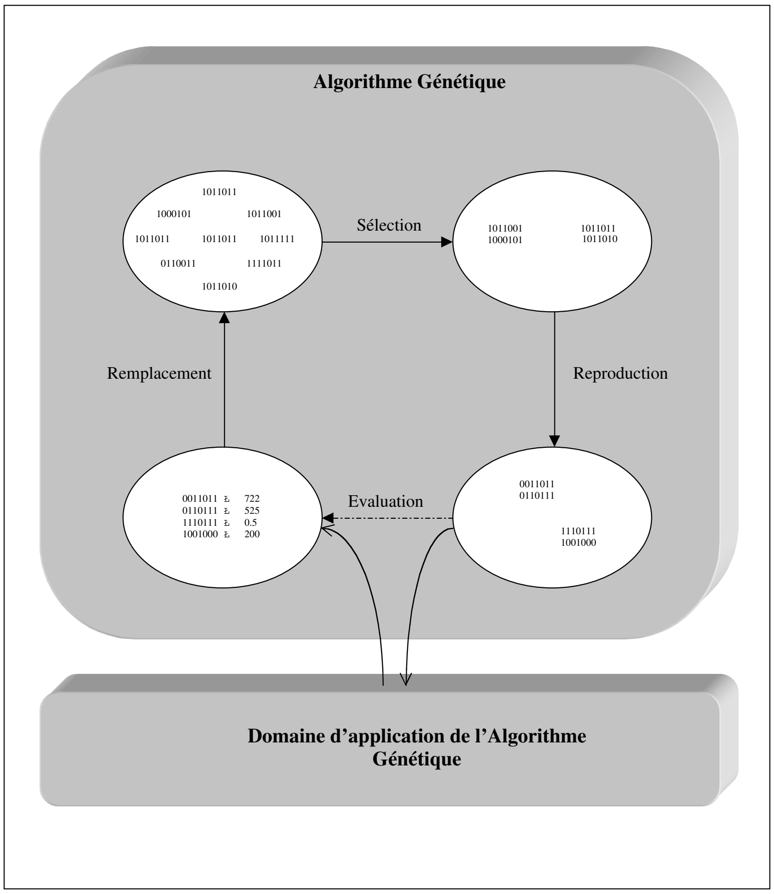

FIG. 3.2 - *Cycle d'exécution d'un algorithme génétique.*

Cela empêche l'algorithme d'évoluer vers l'optimum de la fonction, et son comportement devient proche d'une exploration aléatoire.

Le deuxième problème provient de la nature *bruitée* de la fonction objectif. La fonction $\hat{G}(X)$ à laquelle on a accès par le calcul numérique n'est qu'une approximation plus ou moins précise de la fonction exacte $G(X)$. Comme cette approximation est obtenue par une méthode de Monte-Carlo, sa variance varie linéairement en fonction du nombre N des tirages utilisés pour l'estimation. L'estimation avec une précision suffisante peut s'avérer très coûteuse en temps de calcul, ce qui rend l'utilisation d'un grand nombre de points durant tout le processus d'optimisation non envisageable.

Nous verrons dans ce qui suit les adaptations qu'on a jugées nécessaires pour remédier à ces deux problèmes.

Notion de Température - Relaxation des contraintes

Pour le problème de l'initialisation de l'algorithme dû à l'étroitesse du support de la fonction objectif, une idée inspirée des algorithmes classiques de recuit simulé consiste à introduire une notion de *température*. Le principe est de commencer par élargir le support de la fonction à maximiser. Pour cela, on va donner aux configurations éloignées des optima des valeurs suffisamment élevées de la fonction objectif.

L'idée est d'introduire un paramètre supplémentaire qu'on notera T (température) à la fonction objectif $G(X)$. On obtient une autre fonction que l'on notera $G_T(x)$, plus régulière, ayant un support plus large que celui de $G(x)$ et

$$\lim_{T\to 0} G_T(X) = G(X)$$

L'élargissement du support de la fonction objectif G passe par l'élargissement de tous les termes (distributions) élémentaires de cette fonction à savoir :

- Les distributions $p_{O_i}(F_i(X, L))$, avec $i = 1 \cdots p$.
- Les contraintes inégalités $p(H_j|F_j(X, L))$, avec $j = m - p + 1 \cdots m$.

Les figures 3.3 et 3.4 montrent respectivement un exemple de distribution élémentaire Gaussienne (Normale) et le cas d'une contrainte d'inégalité sur un intervalle. Les deux courbes représentent respectivement les deux fonctions $f(x)$ et $f_T(x)$:

- Pour la distribution Gaussienne :

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \\ f_T(x) &= \frac{1}{\sqrt{2\pi}\sigma(1+T)} e^{-\frac{1}{2}\frac{x^2}{[\sigma(1+T)]^2}} \end{aligned}$$

- Pour la contrainte d'inégalité sur l'intervalle $[a, b]$:

$$f(x) = \begin{cases} 1 & \text{si } a \leq x \leq b \\ 0 & \text{sinon} \end{cases}$$

$$f_T(x) = \begin{cases} 1 & \text{si } a \leq x \leq b \\ e^{-\frac{(x-a)^2}{(b-a)T}} & \text{si } x < a \\ e^{-\frac{(x-b)^2}{(b-a)T}} & \text{sinon} \end{cases}$$

Pour le cas général où les contraintes peuvent être plus complexes, la distribution de probabilité pour chaque contrainte $c_i(x) \leq 0$ aura la formule[1] :

1. Il ne s'agit pas, bien entendu, d'une distribution proprement dit mais seulement d'un *noyau* parce qu'elle ne satisfait pas la condition de normalisation. Comme l'utilisation faite de cette *distribution* permet de la calculer à un facteur multiplicatif près, la violation de cette condition de normalisation ne pose aucun problème.

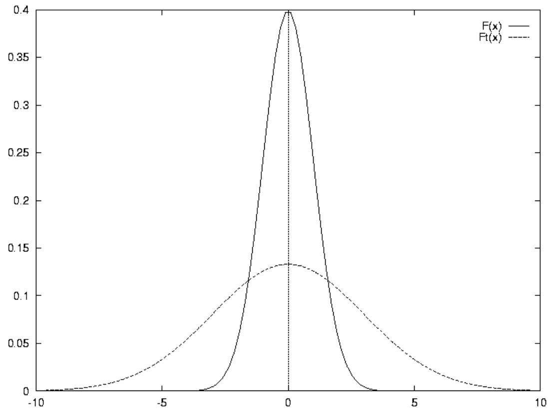

FIG. 3.3 - *Cas d'une distribution gaussienne.*

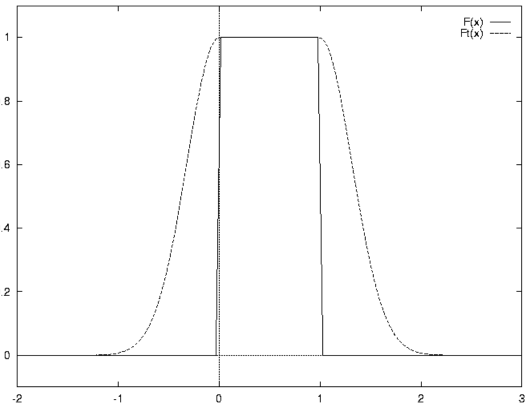

FIG. 3.4 - *Cas d'une contrainte.*

$$f_T^i(x) = \begin{cases} 1 & \text{si } c_i(x) \leq 0 \\ e^{-\frac{c_i^2(x)}{T}} & \text{sinon} \end{cases}$$

et la distribution $f_T(x)$ correspondant à la contrainte $C(x) \leq 0$ sera la multiplication $f_T(x) = \prod_i^n f_T^i(x)$.

Les figures 3.5 et 3.6 représentent la distribution correspondant aux contraintes inégalités pour une relation *Point sur Face* à différentes températures.

Les figures 3.7 et 3.8 représentent un exemple de la fonction objectif G_T pour deux valeurs différentes de T. Il s'agit d'une fonction à deux variables (x, y) représentées par le plan de la figure, et la valeur de la fonction est donnée par le niveau de gris correspondant.

L'utilisation de cette notion de température est basée sur la supposition qu'une population d'un AG initialisé et exécuté pendant quelques cycles avec G_{t_1} comme fonction d'évaluation représente une bonne initialisation pour un autre AG utilisant comme fonction

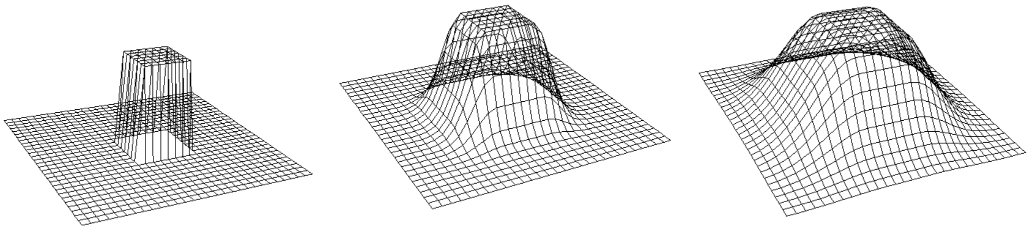

FIG. 3.5 - *La distribution correspondant à la contrainte inégalité pour une relation* Point sur Face *à différentes températures. À gauche la contrainte originelle ($T = 0.0$). Au milieu la contrainte relaxée avec une température $T = 50.0$. À droite la contrainte à une température $T = 100.0$.*

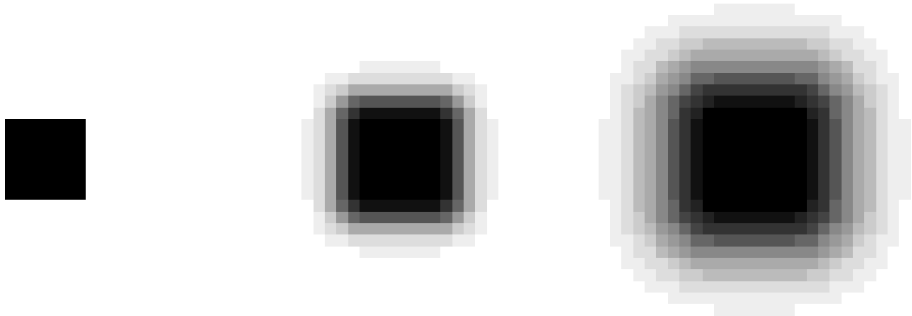

FIG. 3.6 - *La distribution correspondant à la contrainte inégalité en niveaux de gris à différentes températures ($T = 0.0$, $T = 50.0$ et $T = 100.0$ de gauche à droite).*

d'évaluation G_{t_2} où $t_1 > t_2$.

Cette température est initialisée à une valeur T_0 suffisamment grande pour avoir des points de départ avec des valeurs non-nulles. Elle est ensuite diminuée progressivement jusqu'à 0 (pour avoir la fonction originelle).

Notion de multiprécision

Comme la précision de l'estimation $\hat{G}(X)$ de la fonction objectif dépend du nombre N de points utilisés pour cette estimation, une idée est d'introduire N comme un paramètre supplémentaire pour définir une autre fonction $G_N(X)$.

Supposons qu'on ait initialisé et exécuté pendant quelques cycles un AG ayant $G_{n_1}(X)$ comme fonction d'évaluation. Alors la population de cet AG est une bonne initialisation pour un autre AG ayant $G_{n_2}(X)$ où $n_2 > n_1$ comme fonction d'évaluation. Cette notion de multiprécision permet de commencer le processus d'optimisation avec une approximation peu précise (peu coûteuse) de la fonction d'évaluation, et d'affiner cette approximation

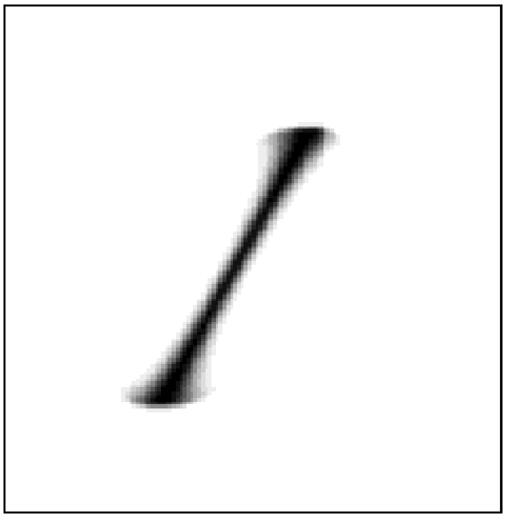

FIG. 3.7 - *Support de la fonction* $G_T(x,y)$ *avec* $T = 0.0$

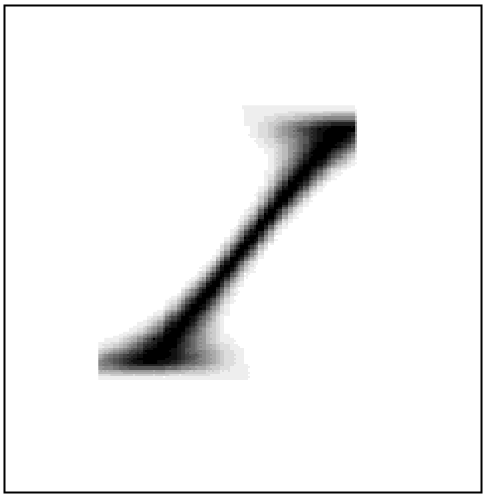

FIG. 3.8 - *Support de la fonction* $G_T(x,y)$ *avec* $T = 10.0$

au cours de l'optimisation. Pour cela, les AG présentent l'avantage majeur de permettre à plusieurs *solutions* de *cohabiter*. Une *meilleure* solution peut par la suite émerger après augmentation de la précision de l'estimation (augmentation du nombre de points utilisés pour l'estimation).

3.4.1 L'algorithme général d'optimisation

Nous avons expliqué dans la section précédente les principaux problèmes liés à l'utilisation des AG pour résoudre notre problème d'optimisation, et nous avons donné les éléments de solutions à ces problèmes. Nous allons expliquer comment ces éléments seront combinés pour proposer un algorithme génétique qui permette de résoudre en un temps raisonnable les problèmes très contraints que nous nous posons.

Dans ce qui suit, nous allons libeller la fonction d'évaluation de l'AG (la fonction objectif) par la température T et le nombre de points N utilisés pour l'estimation. Elle sera dénotée $G_N^T(X)$.

L'algorithme d'optimisation utilisé peut être décrit par les trois étapes suivantes :

1. Initialisation.
2. Diminution de la température.
3. Augmentation du nombre de points pour l'estimation des intégrales.

Initialisation

La population de l'algorithme génétique est initialisée d'une manière aléatoire en tirant au hasard un ensemble de configurations dans l'espace de recherche X (inconnues du problème). Pour minimiser le coût de l'évaluation de la fonction objectif, le nombre N de points utilisés pour l'estimation des intégrales est initialisé à un nombre N_0 assez faible (on se contente d'une estimation qui peut être grossière de la fonction objectif au début).

Le choix de la température initiale T_0 est crucial pour le bon fonctionnement de l'algorithme. Cette valeur dépend fortement du problème à résoudre; plus le problème est contraint, plus la valeur de T_0 doit être grande pour permettre une relaxation suffisante des contraintes. Il est alors nécessaire d'avoir une méthode d'initialisation automatique de T_0 pour s'adapter à la complexité du problème. Une manière de faire est d'initialiser T_0 à 0 puis d'augmenter la température jusqu'à ce que tous les éléments de la population aient des valeurs non nulles, ce qui est toujours possible vu la forme de la relaxation.

INITIALISATION(AG)

POUR tout individu population[i] $\in$ population de AG FAIRE
    REPETER
        population[i] = random(S)
        val = $G_{N_0}^T$(population[i])
        si (val = 0.0)
            T = T + ΔT
    JUSQU'A (val > 0.0)

FINPOUR
FIN

À la fin de l'initialisation, tous les individus de la population auront des valeurs non nulles. Il est à noter qu'il faut réévaluer toute la population après cette étape d'initialisation parce que les individus ont été évalués à différentes températures.

Diminution de la température

Une fois la population initialisée, nous procédons à une diminution progressive de la température T du problème pour se rapprocher de plus en plus de la fonction objectif à maximiser ($T = 0.0$). La diminution de la température peut être envisagée de diverses manières. Nous avons fait le choix de faire une diminution géométrique de T (en la multipliant par un facteur) :

Après un nombre fixe de cycles n_{cycles} d'exécution de l'AG, la température T est multipliée par un facteur α ($0 < \alpha < 1$). Une valeur trop petite de α risque d'engendrer une divergence de l'algorithme tandis qu'une valeur trop près de 1 risque d'augmenter considérablement le temps d'exécution. Cette valeur à été fixée expérimentalement à 0.8. Le changement de la température nécessite bien évidemment la réévaluation de toute la population de l'AG.

Cette partie de l'algorithme s'achève quand la température T est inférieure à une petite valeur T_ϵ. Le nombre de points N reste égal à N_0 pour toute cette étape de l'algorithme.

Pour retrouver la fonction objectif originelle, la température est fixée à la fin de cette étape à 0, et la population de l'AG est réévaluée à cette température.

TEMP_REDUCTION(AG)

TANTQUE (T > T_ϵ) FAIRE
        POUR i=1 JUSQU'A n_cycles FAIRE
                Executer(AG)
        FINPOUR
        T = T * α
FINTANTQUE
FIN

Augmentation du nombre de points pour l'estimation des intégrales

À la fin de l'étape précédente, la population de l'AG contient plusieurs solutions réalisables au problème. Notre but maintenant est d'avoir des estimations de plus en plus précises pour chaque individu de la population, pour pouvoir départager ces solutions.

Comme la précision de cette estimation dépend du nombre N de points utilisés, on va procéder à l'augmentation de N au fur et à mesure de l'évolution de l'algorithme. L'idée est de multiplier, après un nombre fixe de cycles d'exécution de l'AG, le nombre de points N utilisés pour l'estimation par un facteur β ($\beta > 1$). La multiplication de N par β va augmenter bien évidemment la précision de l'estimation de la fonction d'évaluation, parce que la variance de cette estimation va être divisée par β :

$$Var(G^0_{\beta N}) = \frac{1}{\beta} Var(G^0_N)$$

L'algorithme s'arrête quand le nombre maximal de points N_{max} est atteint.

N_POINTS_AUGMENTATION(AG)

```
DEBUT
    TANTQUE (N < N_max) FAIRE
        POUR i=1 JUSQU'A n_cycles FAIRE
            Executer(AG)
        FINPOUR
        N = N * β
    FINTANTQUE
FIN
```

3.4.2 Détails d'implantation de l'AG proposé

Organisation du génotype

Un génotype représenté par une chaîne de bits est la concaténation de l'ensemble des n chaînes correspondant aux n inconnues du problème.

Pour chaque inconnue, la chaîne de bits est générée en discrétisant avec un pas donné la plage de valeurs que cette dernière peut prendre. Le nombre de bits nb_i nécessaire pour coder une variable v_i prenant sa valeur dans un intervalle $[a_i, b_i]$ avec un précision p_i est de :

$$nb_i = \log_2(\frac{b_i - a_i}{p_i})$$

Par exemple, pour un paramètre de rotation prenant sa valeur entre 0.0 et 2π avec un pas de discrétisation de 10^{-4}, le nombre de bits nécessaires pour coder ce paramètre est de 16.

La valeur a_i sera codée par la chaîne de bits $\underbrace{00\cdots0}_{nb_i}$, tandis que la valeurs de b_i sera codée par la chaîne $\underbrace{11\cdots1}_{nb_i}$.

La dimension nb du génotype est donnée alors par :

$$nb = \sum_{i=1}^{n} nb_i$$

Pour un problème ayant, par exemple, 12 paramètres de rotation codés tous entre 0.0 et 2π avec un pas de discrétisation de 10^{-4}, le nombre de bits nécessaires pour le codage du génotype est de $12 \times 16 = 192$. C'est-à-dire que le vecteur $v = (v_1, \cdots v_12)$ peut prendre 2^{192} valeurs possibles allant de $\underbrace{(0.0, 0.0, \cdots, 0.0)}_{12}$ codé par $\underbrace{00\cdots0}_{192}$ jusqu'à $\underbrace{(2\pi, 2\pi, \cdots, 2\pi)}_{12}$ codé par $\underbrace{11\cdots1}_{192}$.

Le décodage d'un génotype pour passer de la chaîne de bits à un vecteur de réels est aussi nécessaire. Ce décodage est effectué par le procédé inverse.

Le processus de reproduction

Pour l'opérateur de croisement, nous avons choisi une variante qui consiste à échanger les bits entre deux individus selon un masque aléatoire spécifiant les éléments des chaînes à garder de chacun des parents :

Les bits à 1 du masque représentent les bits à garder du premier parent, et les bits à 0 ceux à garder du second. Par exemple :

Masque	Parents	Individu produit
0010011	0100101	1101001
	1111011	

La sélection de deux individus dans le but de les croiser est restreinte à un voisinage (figure 3.11). Cela empêche qu'un individu ayant une bonne évaluation remplace tous les autres prématurément et permet de garder, dans une population donnée, un nombre maximal de solutions. Pour cela, la population de l'algorithme est organisée en grille à 2 dimensions. Le nombre d'individus de la population a été fixé expérimentalement à 64 (grille de 8 × 8individus).

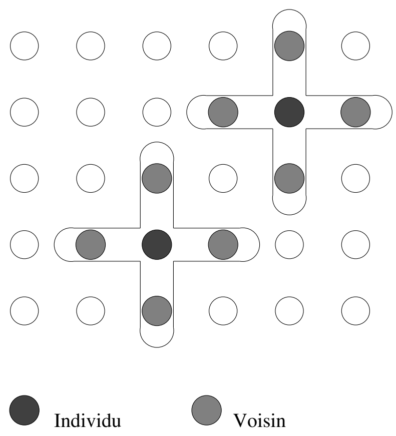

FIG. 3.9 - *Notion de voisinage pour un individu de la population*

L'opérateur de mutation utilisé consiste à changer la valeur (de 0 à 1 ou bien de 1 à 0) d'un bit du génotype pris au hasard.

Convergence de l'estimateur de l'intégrale

Dans la section précédente, nous avons indiqué que l'augmentation du nombre de points utilisés pour l'estimation des intégrales continue jusqu'à atteindre un nombre maximal N_{max}. Dans la pratique, ce nombre de point dépend du problème. Il est défini comme le nombre permettant d'atteindre la convergence des estimateurs de nos intégrales. Nous définissons la convergence d'un estimateur de la valeur $G(p[i])$ pour un individu $p[i]$ de l'AG comme étant le critère donné pour deux nombres N_t et N_{t+1} par :

$$Converge\_Integ(p[i]) = \begin{cases} vrai & si \ \frac{|G_{N_t}(p[i]) - G_{N_{t+1}}(p[i])|}{G_{N_t}(p[i])} < \epsilon_1 \\ faux & sinon \end{cases}$$

où ϵ_1 est un seuil à partir duquel l'estimation de la valeur d'un individu, en utilisant successivement N_t points et N_{t+1} points, est considérée inchangée.

Nous définissons par conséquent la convergence de l'estimation des intégrales pour un AG comme :

$$Converge\_Integ(AG) = \begin{cases} vrai & si \ \forall p[i] \in population, \ Converge\_Integ(p[i]) = vrai \\ faux & sinon \end{cases}$$

À chaque augmentation du nombre de points N , et à chaque réévaluation de la population, nous vérifions si les estimateurs des intégrales pour tous les individus ont convergé. Si c'est le cas, le nombre de points ne sera plus augmenté avant la réévaluation suivante.

Convergence de l'optimisation

Pour le critère d'arrêt de l'algorithme génétique, nous avons choisi le critère classique suivant :

$$Converge\_Optim(AG) = \begin{cases} vrai & si \ \frac{SFitness_{t+1}(AG) - SFitness_t(AG)}{SFitness_t(AG)} < \epsilon_2 \\ faux & sinon \end{cases}$$

où

$$SFitness = \sum_{p[i] \in population} G(p[i])$$

et ϵ_2 est un seuil à partir duquel $SFitness$ est supposée inchangée.

Le processus d'optimisation s'arrête quand la somme des valeurs de “fitness” des individus de la population n'augmente plus d'une manière significative entre deux générations successives.

Implémentation *parallèle*

En pratique et dans le but d'avoir un algorithme plus global, l'algorithme implanté comprend une partie *parallèle*. Par *parallèle*, nous entendons l'application d'une manière indépendante de n AG sans aucune communication entre ces derniers. Les populations obtenues à la fin de chacun de ces algorithme sonr rassemblées pour servir de population initiale à un AG global. Pour les n AG parallèles, nous appliquons les étapes d'initialisation et de diminution de température en laissant le nonmbre de points N fixé à N_0. Tandis que pour l'AG global, nous appliquons l'étape d'augmentation du nombre de points (à une température $T = 0.0$) dans le but de départager les différents solutions issues des AG parallèles. Ce schéma d'exécution est donné dans la figure 3.10.

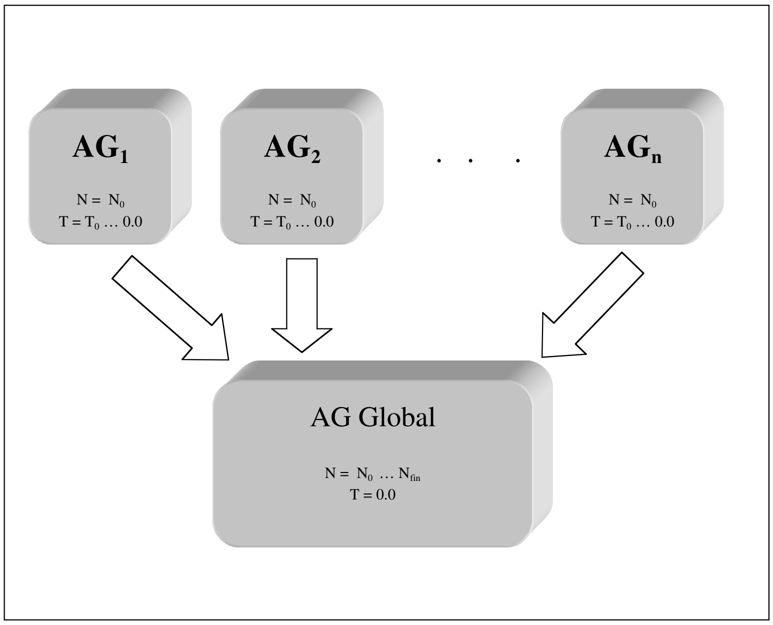

FIG. 3.10 - *Schéma de l'exécution de l'AG proposé.*

Méthode de sélection

En plus de cette implémentation parallèle, la sélection de deux individus dans le but de les croiser est restreinte à un voisinage (figure 3.11). Cela empêche qu'un individu ayant une bonne évaluation remplace tous les autres prématurément et permet de garder, dans une population donnée, un nombre maximal de solutions.

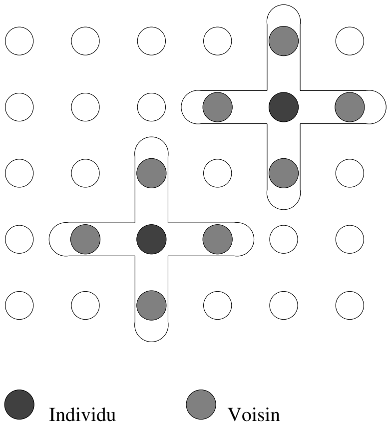

FIG. 3.11 - *Notion de voisinage pour un individu de la population*

3.4.3 Trace de l'exécution de l'algorithme génétique pour un problème simple

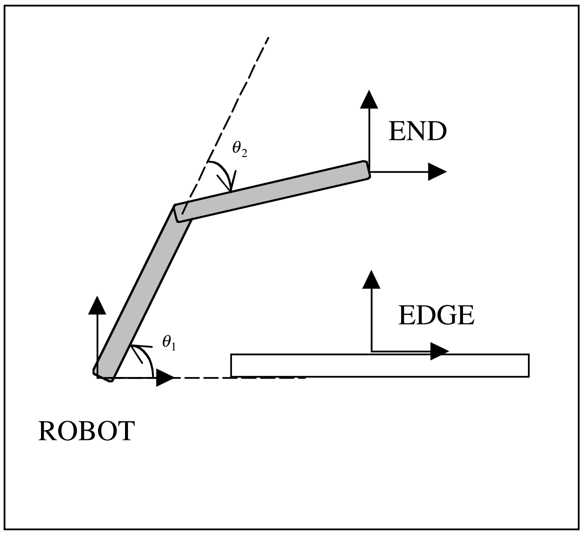

FIG. 3.12 - *Exemple simple d'inversion cinématique sous incertitudes géométriques.*

Considérons, par exemple, le problème d'inversion cinématique sous incertitudes présenté dans la figure 3.12. Les incertitudes les plus significatives sont supposées sur les zéros géométriques des deux articulations rotoïdes du bras planaire. Le but est de ramener le point final END du bras sur l'arête supérieure EDGE de l'objet.

Intuitivement, les meilleures solutions pour ce problème sous-déterminé sont celles permettant plus de précision sur la position obtenue entre le point END est l'arête EDGE. La structure du bras permet d'avoir une meilleure précision dans les configurations repliées qui minimisent la propagation des incertitudes sur les zéros géométriques.

Les figures 3.13, 3.14 et 3.15 représentent quelques solutions présentes dans la population de l'algorithme génétique pour des températures de 15.0, 5.0 et 0.0. La figure 3.16 représente, quant à elle, les deux solutions retenues après exécution de l'AG avec une température de 0.0 et un nombre de points $N = N_{max}$.

3.5 Conclusion

Nous avons présenté dans ce chapitre la méthode de résolution correspondant à un problème géométrique. Nous avons présenté en particulier comment aborder les deux problèmes liés à la modélisation bayesienne à savoir :

- Le problème de calcul des intégrales.

- Le problème d'optimisation de la fonction objectif.

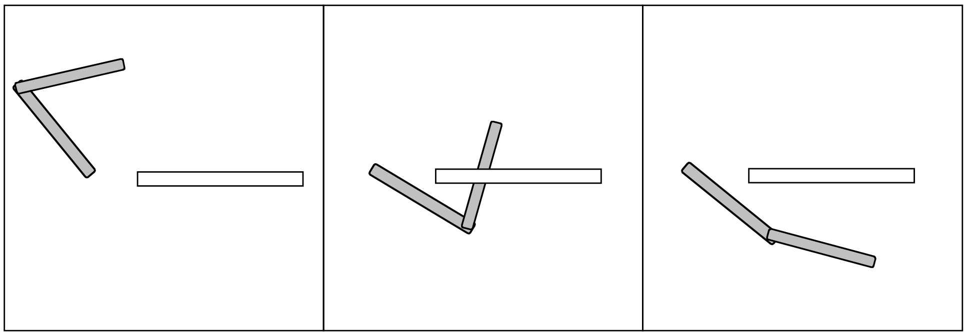

FIG. 3.13 - *Quelques configurations de la population de l'AG à une température $T = 15.0$ et avec un nombre de points $N = N_0 = 10$. Ces configurations violent généralement les contraintes originelles à cause de la relaxation.*

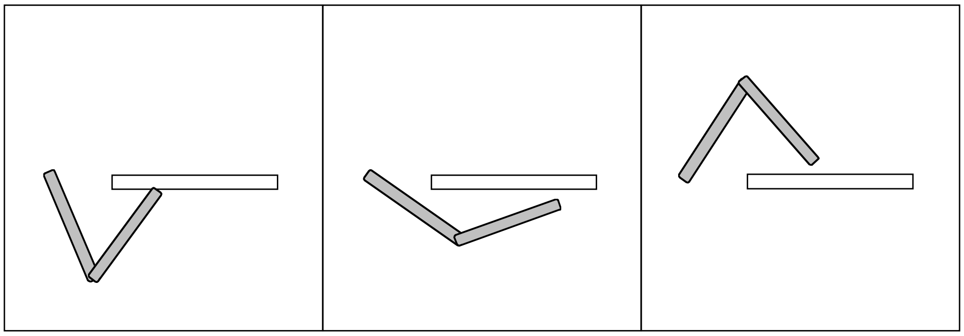

FIG. 3.14 - *Quelques configurations de la population de l'AG à une température $T = 5.0$ et avec un nombre de points $N = N_0 = 10$. La qualité des configurations s'améliore progressivement et les contraintes sont de plus en plus respectées.*

La méthode de résolution proposée est basée sur l'utilisation d'une méthode de Monte-Carlo pour le calcul numérique des intégrales et sur un algorithme génétique multiprécision pour l'optimisation. Le principe de la méthode consiste à contrôler le procédé d'intégration par le procédé d'optimisation dans le but de réduire le coût du calcul en commençant l'optimisation par des approximations qui peuvent être peu précises au début, et en affinant ces approximations au fur et à mesure. Pour remédier au problème de la complexité de la forme de la fonction objectif pour les problèmes très contraints, une notion de température inspirée des méthodes de recuit simulé a été introduite. Cette notion permet une relaxation provisoire des contraintes au début du procédé d'optimisation. Elle permet par la suite de retrouver progressivement les contraintes originelles en décroissant la température.

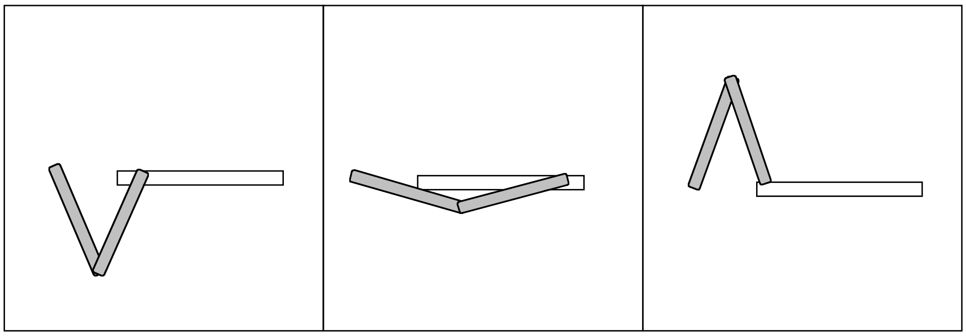

FIG. 3.15 - *Quelques configurations de la population de l'AG à une température $T = 0.0$ et avec un nombre de points $N = N_0 = 10$. Les valeurs de ces configurations correspondent à la fonction objectif originelle (sans relaxation), mais la précision de l'estimation de cette fonction en utilisant un petit nombre de points ne permet pas de départager ces configurations.*

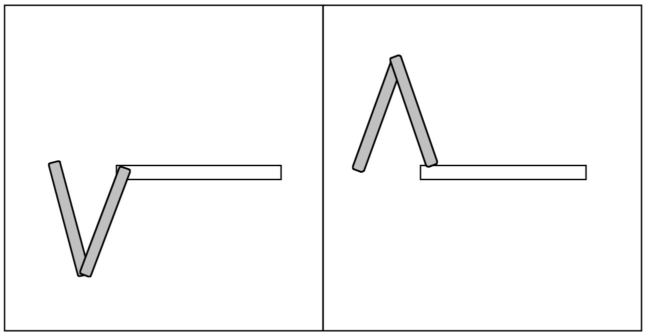

FIG. 3.16 - *Les deux configurations présentes dans la population après exécution de l'AG à une température $T = 0.0$ et avec un nombre de points $N = N_{\max} = 1000$. Ces configurations sont équivalentes et permettent effectivement d'avoir une précision maximale sur la position que l'on souhaite obtenir entre le point END et l'arête EDGE.*

Chapitre 4

Applications

L'objectif de ce chapitre est de présenter l'utilisation concrète de la méthodologie de spécification présentée dans les chapitres précédents, pour des problèmes de robotique et de CAO robotique.

Nous commençons dans la première partie par une présentation de la méthode utilisée pour la modélisation des systèmes articulés et des mesures capteurs.

La deuxième sera consacrée à la description de quelques applications concrètes de notre système pour modéliser et résoudre différents problèmes de géométrie à savoir :

- Les problèmes d'inversion géométrique sous incertitudes.
- Les problèmes d'inversion géométrique utilisant des mesures capteurs supplémentaires pour affiner le modèle.
- Les problèmes d'étalonnage de poses à partir de données capteurs.
- L'étalonnage de la pose et de la forme d'un objet.
- Les problèmes de conception d'une cellule robotique (par exemple le positionnement du robot pour pouvoir atteindre dans des conditions optimales des points de l'environnement de travail).

Bien que ces problèmes paraissent de natures différentes, nous montrerons qu'ils peuvent être modélisés et résolus en utilisant la même méthodologie et le même moteur de résolution. Nous décrirons en particulier pour chaque application la modélisation correspondante, et les apports de notre approche pour une telle application. Les résultats obtenus ainsi que quelques indices de complexité seront donnés pour chaque exemple.

4.1 Méthode de modélisation

Nous abordons dans ce paragraphe la méthode de modélisation utilisée. Cette dernière présente l'avantage de permettre une grande uniformité et homogénéité de la représentation des objets *statiques* de la scène, des corps d'une chaîne cinématique (mécanisme articulé), et des mesures obtenues par un capteur de *distance*. Dans ce qui suit, nous allons nous intéresser en particulier à la modélisation des mécanismes articulés et des mesures capteurs.

Dans notre système, il n'existe pas de différence entre la modélisation d'une chaîne articulée et celle d'un ensemble de primitives géométriques statiques. Toutes deux utilisent la notion de transformation géométrique entre repères pour spécifier la pose relative de deux primitives. Seuls changent les distributions de probabilité affectées aux paramètres nécessaires à la représentation de ces transformations, et le rôle de chacun de ces paramètres pour un problème donné.

Par exemple, pour modéliser un bras manipulateur à six degrés de liberté, nous procédons à la décomposition de ce dernier en un ensemble de corps rigides, chacun étant attaché à un repère $R_1 \ldots, R_6$.

L'affectation des distributions de probabilité sur les paramètres du vecteur

$$Q_{R_iR_{i+1}} = (t_x, t_y, t_z, r_x, r_y, r_z)^T$$

de pose relative entre les deux repères R_i et R_{i+1} définit nos connaissances préalables sur une articulation donnée et sa relation avec l'articulation suivante. Cela permet de modéliser les imprécisions liées au modèle de l'articulation, ainsi que nos connaissances sur le problème. Examinons plusieurs types de connaissances préalables envisageables dans le cas de l'inversion géométrique.

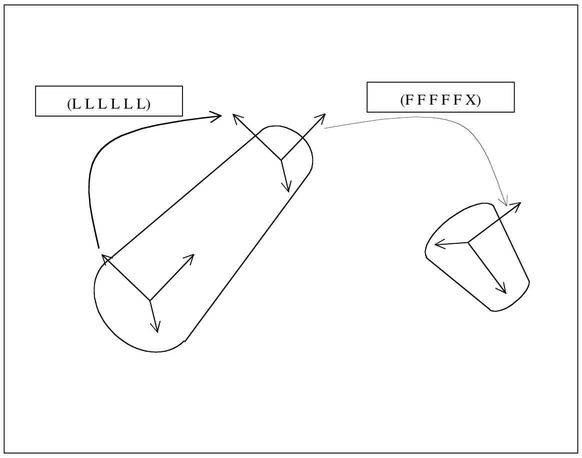

FIG. 4.1 - *Modélisation simple d'une articulation.*

Modélisation simple d'une articulation sans parallaxe de l'axe de rotation pour une inversion géométrique

Nous modélisons notre connaissance préalable en choisissant de ne considérer que les imprécisions de fabrication sur la longueur l du lien cinématique et sur la position y, z de l'axe de rotation. Les distributions choisies sont des gaussiennes. Les distributions de Dirac sur r_x et r_y expriment le choix de ne pas tenir compte des incertitudes sur ces dimensions et que l'on affecte implicitement la valeur "Fixe" aux statuts de ces paramètres. La distribution uniforme entre les butées min_{r_z} et max_{r_z} sur le paramètre r_z précise que nous n'avons pas de connaissance préalable sur la position de l'articulation dans le cas d'une inversion géométrique (aucune configuration n'est favorisée).

$$\begin{aligned}
p(t_x) &= Normal(l_0, \sigma_l) \\
p(t_y) &= Normal(0.0, \sigma_{t_y}) \\
p(t_z) &= Normal(0.0, \sigma_{t_z}) \\
p(r_x) &= \delta_0 \\
p(r_y) &= \delta_0 \\
p(r_z) &= Uniforme(min_{r_z}, max_{r_z})
\end{aligned}$$

où δ est la distribution de Dirac.

Modélisation simple d'une articulation avec parallaxe de l'axe de rotation pour une inversion géométrique

On peut tout aussi bien modéliser ce que l'on connaît des imprécisions sur les parallaxes en affectant d'autres distributions à r_x et r_y, comme par exemple :

$$\begin{aligned}
p(t_x) &= Normal(l_0, \sigma_l) \\
p(t_y) &= Normal(0.0, \sigma_{t_y}) \\
p(t_z) &= Normal(0.0, \sigma_{t_z}) \\
p(r_x) &= Normal(0.0, \sigma_{r_x}) \\
p(r_y) &= Normal(0.0, \sigma_{r_y}) \\
p(r_z) &= Uniforme(min_{r_z}, max_{r_z})
\end{aligned}$$

Modélisation de l'imprécision sur le zéro géométrique

On peut affiner la modélisation de la scène en ajoutant des repères intermédiaires dans la chaîne cinématique; cela permet par exemple de modéliser les connaissances préalables sur l'erreur de zéro géométrique. Ainsi on peut créer un repère intermédiaire R_i' entre R_i et R_{i+1} et utiliser les distribution sur les paramètres nouvellement crées pour modéliser plus finement notre connaissance sur l'articulation.

$$\begin{aligned}
p(t_x^{i'}) &= Normal(l_0, \sigma_l) \\
p(t_y^{i'}) &= Normal(0.0, \sigma_{t_y}) \\
p(t_z^{i'}) &= Normal(0.0, \sigma_{t_z}) \\
p(r_x^{i'}) &= Normal(0.0, \sigma_{r_x}) \\
p(r_y^{i'}) &= Normal(0.0, \sigma_{r_y}) \\
p(r_z^{i'}) &= Normal(0.0, \sigma_{r_z})
\end{aligned}$$

La distribution sur $r_z^{i'}$ modélise notre connaissance sur le zéro géométrique de l'articulation.

$$\begin{aligned}
p(t_x^{i+1}) &= \delta_0 \\
p(t_y^{i+1}) &= \delta_0 \\
p(t_z^{i+1}) &= \delta_0 \\
p(r_x^{i+1}) &= \delta_0 \\
p(r_y^{i+1}) &= \delta_0 \\
p(r_z^{i+1}) &= Uniforme(min_{r_z}, max_{r_z})
\end{aligned}$$

La distribution sur r_z^{i+1} modélise notre connaissance ou plutôt notre ignorance de la position d'un axe dans le cas d'une inversion géométrique.

Modélisation fine d'une articulation

En pratique, d'autres paramètres peuvent être ajoutés à des liens géométriques probabilistes. C'est ainsi que pour modéliser les erreurs de commande en position, on rajoute une variable θ à chaque articulation. Les distributions sur les variables $t_x^{i+1}, t_y^{i+1}, t_z^{i+1}, r_x^{i+1}, r_y^{i+1}, \theta, r_z^{i+1}$, prennent alors les formes suivantes :

$$\begin{aligned}
p(t_x^{i+1}) &= \delta_0 \\
p(t_y^{i+1}) &= \delta_0 \\
p(t_z^{i+1}) &= \delta_0 \\
p(r_x^{i+1}) &= \delta_0 \\
p(r_y^{i+1}) &= \delta_0 \\
p(\theta) &= Uniforme(min_\theta, max_\theta) \\
p(r_z^{i+1}|\theta) &= Normal(f(\theta), \sigma_{r_z})
\end{aligned}$$

où $f(\theta)$ est la fonction correspondant au rapport de réduction et σ_{r_z} l'erreur supposée sur la commande.

En fait de nombreuses variables non forcément géométriques peuvent être rajoutées au système. Par exemple, on peut modéliser la dilatation d'un axe en fonction de la température et faire de la température t une donnée du problème d'inversion. Ceci donnerait par exemple :

$$\begin{aligned}
P(t) &= Uniforme(-10, 30) \\
p(t_x^{i'}|t) &= Normal((1 + k * (t - t_0))l_0, \sigma_l) \\
p(t_y^{i'}) &= Normal(0.0, \sigma_{t_y}) \\
p(t_z^{i'}) &= Normal(0.0, \sigma_{t_z}) \\
p(r_x^{i'}) &= Normal(0.0, \sigma_{r_x}) \\
p(r_y^{i'}) &= Normal(0.0, \sigma_{r_y}) \\
p(r_z^{i'}) &= Normal(0.0, \sigma_{r_z})
\end{aligned}$$

Ces distributions peuvent être données par le constructeur, ou bien issues d'un étalonnage préalable du bras.

4.1.1 Modélisation d'un bras articulé

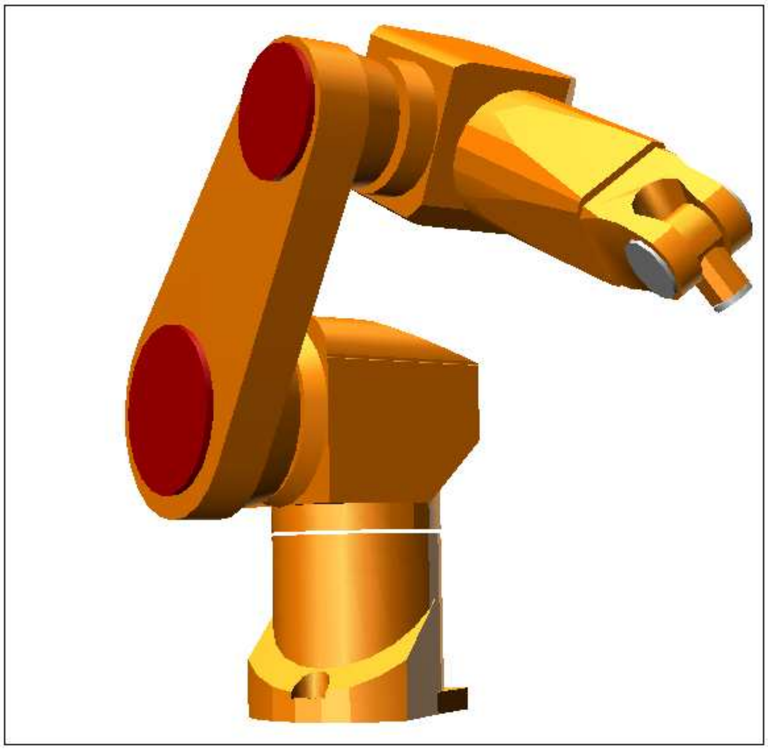

FIG. 4.2 - *Le bras articulé Stäubli Rx90 à 6 degrés de liberté.*

Dans les simulations que nous présenterons dans le reste de ce chapitre, nous avons utilisé un ou plusieurs modèles d'un bras Stäubli Rx90 à 6 liaisons rotoïdes (figure 4.2). Le modèle par décomposition du bras en un ensemble de corps suivant l'approche décrite ci-dessus est donné dans la figure 4.3. Les poses relatives entre ces différents corps sont données par des liens cinématiques probabilistes décrivant pour chaque articulation les incertitudes qui lui sont attachées.

4.1.2 Modélisation d'une mesure

Dans ce paragraphe, nous nous intéressons à la modélisation des mesures provenant de capteurs et la manière de les intégrer dans les modèles utilisés par le système.

Les capteurs fournissant directement des mesures de distance peuvent être connectés au système avec une interface simple n'exigeant aucun effort de prétraitement. C'est le

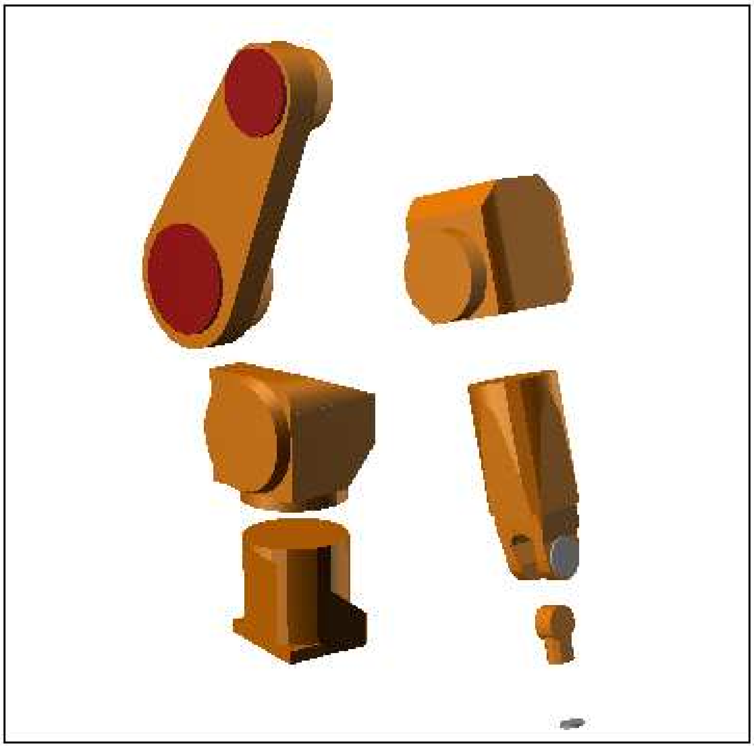

FIG. 4.3 - *Décomposition du bras en un ensemble de corps rigides.*

cas des télémètres laser et des proximètres infrarouges ou ultrasons. Dans le cas des capteurs fournissant des informations de haut niveau comme les caméras et les capteurs de vision 3D par exemple, un prétraitement est indispensable pour extraire des mesures de position/orientation utilisables par le système.

Dans les deux cas, des a priori sur les valeurs des mesures peuvent être disponibles. Dans le cas d'un capteur de distance, l'a priori sur la mesure peut être tout simplement la spécification du fabricant, ou bien issu d'un étalonnage préalable du capteur. Cet a priori peut être typiquement une distribution gaussienne autour de la valeur nominale $p(m) = G(m_0, \sigma_m)$ avec une précision donnée σ_m. En plus de cet a priori sur la réponse du capteur, la pose du capteur (du repère attaché au capteur) peut être elle-même entachée d'incertitude. C'est le cas d'un capteur non étalonné ou bien d'un capteur monté au bout d'un mécanisme articulé : la propagation des incertitudes le long de la chaîne cinématique correspondant à ce mécanisme engendre plus d'imprécision sur la pose du capteur (figure 4.4).

Dans le cas d'un capteur nécessitant un prétraitement, l'a priori sur la réponse du capteur peut être évalué en prenant en compte la précision de la mesure originelle (de haut niveau) et la nature du prétraitement effectué sur cette dernière.

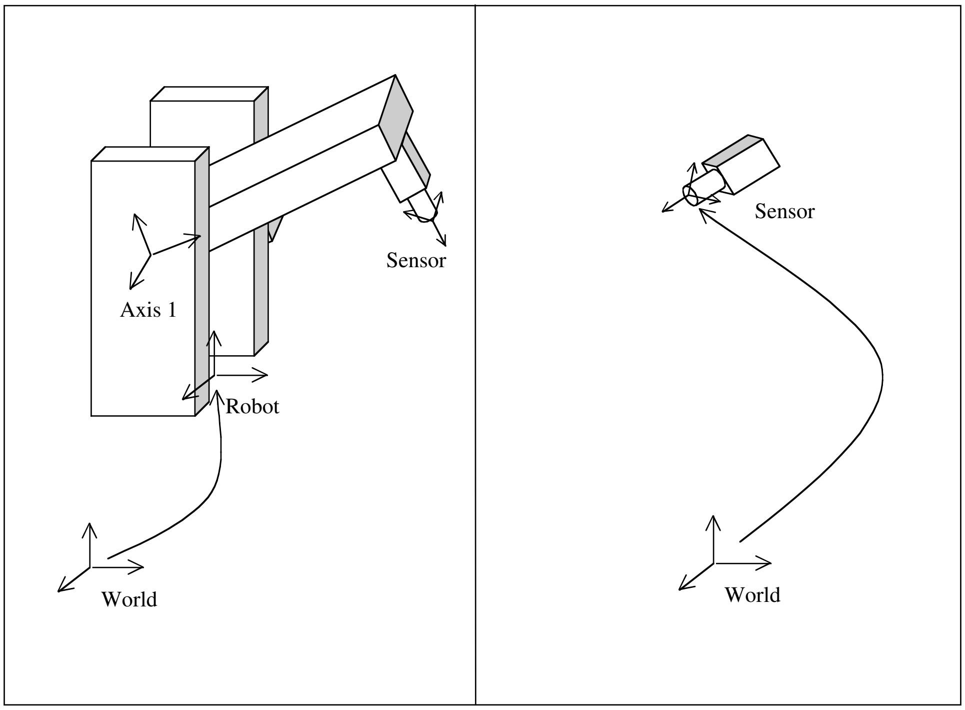

FIG. 4.4 - *Modélisation d'un capteur de distance. À gauche, un capteur monté à l'extrémité d'un bras articulé dont le modèle géométrique est entaché d'incertitude. À droite, un capteur dont la pose par rapport à un repère de référence n'est connue qu'avec incertitude.*

4.2 Application au problèmes d'inversion géométrique

Cette partie sera consacrée à l'utilisation de l'approche bayesienne proposée pour modéliser et résoudre les problèmes d'inversion géométrique sous incertitudes géométriques. Ces incertitudes (ou imprécisions) représentent, à un instant donné, notre état de connaissance (plus ou moins précis) sur le modèle géométrique de l'environnement dans lequel on voudrait effectuer une tâche donnée. Nous commencerons par définir la notion d'inversion géométrique en robotique et illustrer l'extension que nous proposons à la notion de *contrainte* géométrique. Nous montrerons comment le formalisme bayesien permet de calculer la propagation des incertitudes sur le modèle géométrique de l'environnement, et d'exploiter cette propagation lors de la résolution des problèmes d'inversion géométrique pour trouver des solutions permettant de satisfaire *au mieux* les contraintes décrivant le problème en question.

4.2.1 Notion d'inversion géométrique en robotique

Pour un système articulé à n degrés de liberté, l'inversion géométrique consiste à trouver la configuration (ou bien les configurations) permettant de mettre un objet (ou une entité) de ce système dans une pose donnée par rapport à un autre objet. Cela revient à calculer le (les) vecteur (s) Q de l'espace des configurations du système permettant d'obtenir une valeur objective O^* d'un vecteur O de dimension m de l'espace de travail.

Habituellement, dans les systèmes de robotique et de CAO, ces problèmes d'inversion sont décrits comme des systèmes de *contraintes* égalités et éventuellement inégalités en dehors de toute prise en compte de l'incertitude :

$$\begin{cases} O(Q) & = & 0 \\ C(Q) & \geq & 0 \end{cases}$$

Dans ce cadre, la solution d'un problème d'inversion revient à résoudre ce système de contraintes et à trouver *une solution* Q^* satisfaisant ces contraintes (sous réserve qu'elle existe).

Dans le cas où plusieurs solutions sont possibles (le système est redondant par rapport à la tâche), une de ces solutions est choisie par l'introduction d'un critère extérieur à la définition initiale du problème. Le critère du choix d'une solution particulière dépend généralement de la méthode de résolution utilisée. Par exemple, pour une méthode de résolution de type descente de gradient, le critère sera de minimiser la distance entre le point initial Q^0 et la solution Q^* dans l'espace des configurations.

Intéressons nous maintenant à la prise en compte des incertitudes sur le modèle géométrique. Pour une configuration donnée, on ne peut plus parler d'une valeur du vecteur objectif mais seulement d'une distribution de probabilité sur ce vecteur. Cela est dû à la propagation des incertitudes sur le modèle et dépend, bien entendu, de la configuration dans laquelle se trouve le système. Dans l'approche que nous proposons, nous introduisons une définition plus générale de la notion de *contrainte*. Nous définissons une *contrainte* non pas comme une équation (ou une inéquation) sur les paramètres des poses relatives, mais comme des distributions de probabilités sur ces paramètres. Ces distributions vont exprimer le niveau de contrainte que l'on souhaite obtenir sur chacune des composantes du vecteur de pose O. Cette formulation permet d'exprimer explicitement les *précisions* (les niveaux de contrainte) que l'on souhaite obtenir sur chaque composante, ce qui rend explicite le critère du choix de la solution pour les cas redondants.

Nous présentons à titre d'exemple un problème d'inversion géométrique pour un système articulé à 3 degrés de liberté. Nous montrons comment le critère de choix de la solution est donné explicitement lors de la description du problème à résoudre, et comment différentes distributions sur les composantes de la pose objective donnent dans le cas général des solutions différentes. Puis nous validerons les résultats expérimentaux en montrant que les résultats donnés par le système sont conformes à ceux prédits par un calcul analytique dans le cas où les hypothèses de linéarisation s'appliquent (incertitude gaussienne avec écart type faible).

4.2.2 Exemple simple d'inversion géométrique sous incertitudes géométriques pour un mécanisme à 3 degrés de liberté

Considérons le système articulé à 3 degrés de liberté présenté dans la figure 4.5. Le premier degré de liberté de ce mécanisme est prismatique tandis que le second et le troisième sont rotoïdes. Nous nous intéressons dans cet exemple à un simple problème d'inversion géométrique qui consiste à mettre le point final (le centre du repère END) du mécanisme

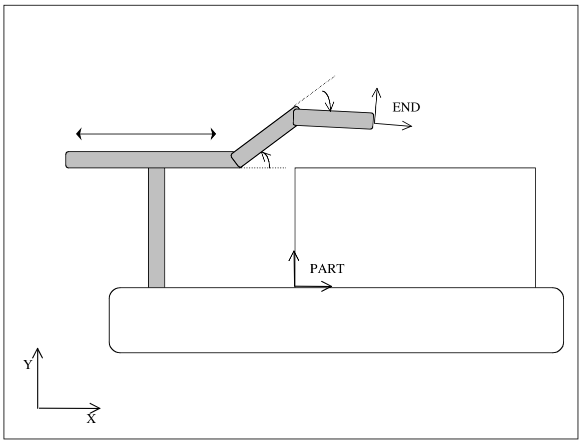

FIG. 4.5 - *Exemple simple d'un problème d'inversion géométrique sous incertitudes géométriques pour un mécanisme à 3 degrés de liberté.*

dans une pose $O = (x_o, y_o)$ par rapport aux repère $PART$. Nous cherchons plus précisément à trouver la configuration $Q = (q_1, q_2, q_3)$ du bras qui permet d'obtenir une pose relative $O = (x_o, y_o)$ entre les repères $PART$ et END. Nous supposons que les erreurs sur les zéros géométriques des deux articulations rotoïdes sont données par des distributions gaussiennes autour de 0.0.

Si l'on ne considère pas les incertitudes sur le modèle géométrique, il est clair qu'il s'agit d'un problème sous-déterminé qui accepte une infinité de solutions. Nous présentons dans ce qui suit deux variantes de ce problème :

- Dans le premier cas, la précision que l'on souhaite obtenir dans la direction de l'axe des X est considérée comme prioritaire pour la tâche.
- Dans le deuxième, la précision suivant l'axe des Y est considérée comme prioritaire.

Cas 1

Dans le premier cas, nous avons supposé que nous voulions plus de précision suivant la direction X que suivant la direction Y (nous avons un niveau plus élevé de contrainte sur la position en X que sur celle en Y). Cela se traduit par une distribution de probabilité plus *étroite* sur le paramètre t_x du vecteur de pose O que celle sur le paramètre t_y. La figure 4.6 montre la solution donnée par le système pour des distributions Gaussiennes sur t_x et t_y avec des moyennes de x_o et y_o, et des écarts-types de 1.0mm et 10.0mm respectivement.

Cas 2

Dans le deuxième cas, nous avons choisi de favoriser la précision sur la direction Y. Nous avons tout simplement inversé les écarts-types des Gaussiennes sur les paramètres t_x et t_y, c'est-à-dire qu'on a pris un écart-type de 10.0mm pour t_x et 1.0mm pour t_y. La figure 4.7 montre la solution trouvée par le système pour ce cas.

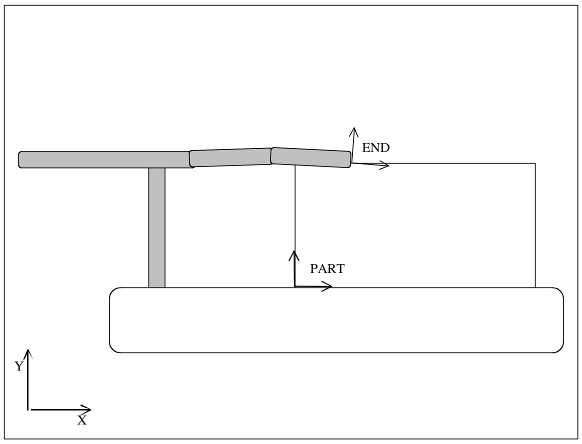

FIG. 4.6 - *Solution du problème d'inversion géométrique pour des écarts-types de 1.0mm et 10.0mm sur les directions X et Y respectivement. Dans cette configuration, la propagation des incertitudes sur les paramètres de rotation est moins importante dans la direction X.*

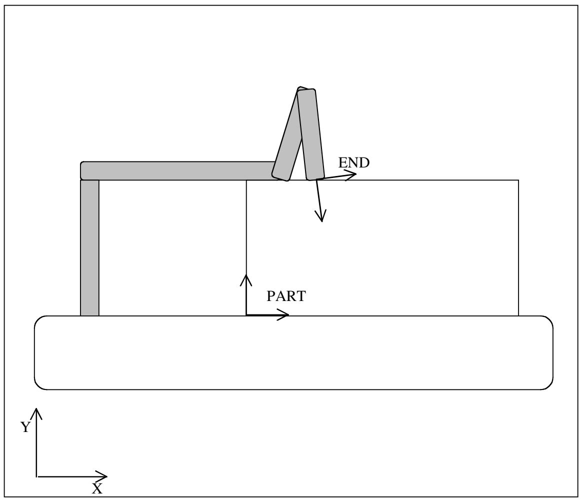

FIG. 4.7 - *Solution du problème d'inversion géométrique pour des écarts-types de 10.0mm et 1.0mm sur les directions X et Y respectivement. Dans cette configuration, la propagation des incertitudes sur les paramètres de rotation est moins importante dans la direction Y.*

Discussion

Pour valider analytiquement ces résultats en utilisant un calcul variationnel, nous allons supposer que les conditions nécessaires pour la linéariation du modèle et la forme des distributions sont respectées. Supposons alors que les seules erreurs de modélisation portent sur les zéros géométriques $\Delta\Theta = (\delta_{\theta_1}, \delta_{\theta_2})^T$ des deux articulations rotoïdes. De plus, supposons que ces erreurs sont assez petites pour permettre une linéarisation (approximation au premier ordre) et qu'elles sont distribuées suivant une gaussienne de matrice de variances-covariances $\Sigma_{\Delta\Theta}$. Il est possible dans ces conditions de trouver analytiquement la matrice des variances-covariances Σ_p sur la position $p = (x, y)^T$ du point END par rapport au repère $PART$ dans une configuration donnée, en utilisant la jacobienne $J = \frac{\partial p}{\partial \Delta\Theta}$. On obtient :

$$\Sigma_p = J \Sigma_{\Delta\Theta} J^T$$

où :

$$p = \begin{pmatrix} x_p + l_0 + \cos(\theta_1 + \delta_{\theta_1}) l_1 + \cos(\theta_1 + \delta_{\theta_1} + \theta_2 + \delta_{\theta_2}) l_2 \\ y_p + \sin(\theta_1 + \delta_{\theta_1}) l_1 + \sin(\theta_1 + \delta_{\theta_1} + \theta_2 + \delta_{\theta_2}) l_2 \end{pmatrix}$$

avec l_0, l_1 et l_2 les longueurs des trois corps du bras, et $(x_p, y_p)^T$ la position relative de $PART$ par rapport à la base du bras.

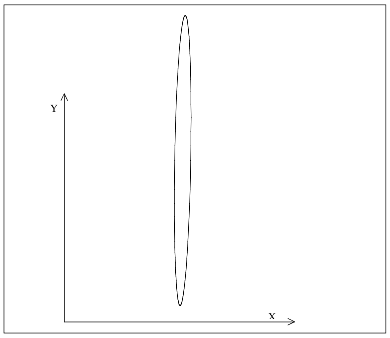

FIG. 4.8 - *Visualisation du volume d'incertitude correspondant à la matrice de variances-covariances Σ_p pour le premier cas. Les incertitudes sur la position de l'extrémité du bras sont moins importantes dans la direction X.*

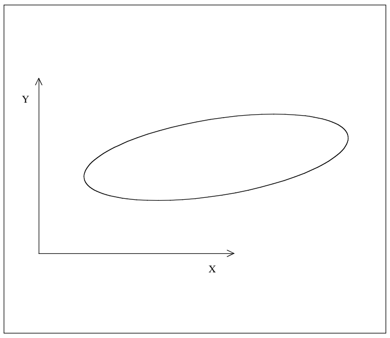

FIG. 4.9 - *Visualisation du volume d'incertitude correspondant à la matrice de variances-covariances Σ_p pour le deuxième cas. Les incertitudes sur la position de l'extrémité du bras sont moins importantes dans la direction Y.*

Les figures 4.8 et 4.9 représentent respectivement les volumes d'incertitude correspondants aux matrices de variances-covariances Σ_p obtenues par linéarisation pour les deux configurations données dans les figures 4.6 et 4.7. On note que pour ces deux configurations, les précisions en X et Y sont conformes à la spécification : le système est plus précis en X dans le cas de la figure 4.8 et plus précis en Y dans le cas de la figure 4.9.

4.2.3 Exemple avec un mécanisme à 12 degrés de liberté

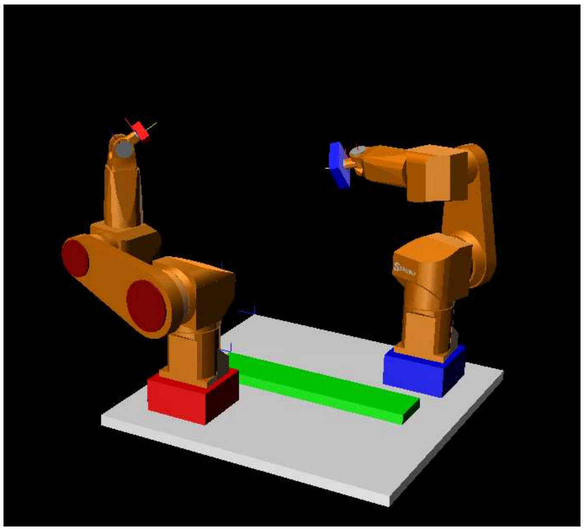

FIG. 4.10 - *Exemple d'inversion géométrique avec deux bras Stäubli. Le problème consiste à mettre les deux parallélépipèdes l'un contre l'autre (face sur face).*

Nous présentons dans cette partie un problème de CAO robotique avec un mécanisme à 12 degrés de liberté. La figure 4.10 représente deux bras Stäubli Rx90. À l'extrémité de chaque bras, nous avons attaché un objet ayant la forme d'un parallélépipède. Le problème d'inversion géométrique auquel nous nous intéressons est de positionner les deux parallélépipèdes l'un contre l'autre avec pour seule contrainte qu'une face du premier soit contenue dans une face du deuxième.

Les corps des deux bras sont modélisés comme des objets liés par des liens probabilistes. Ces liens représentent les incertitudes (imprécisions) sur le modèle géométrique des bras. Les deux parallélépipèdes sont eux-mêmes attachés aux derniers corps des bras par des liens probabilistes. Le graphe cinématique correspondant à ce problème est présenté dans la figure 4.11.

Ce problème est largement redondant parce que le système dispose de 12 degrés de liberté tandis que la tâche à réaliser s'exprime seulement par trois contraintes sur la pose relative entre les repères attachés aux faces qu'on veut mettre l'une contre l'autre, à savoir une sur un paramètre de translation (t_z) et deux sur deux paramètres de rotations (r_x et r_y).

Nous avons choisi dans cet exemple de considérer deux modèles d'incertitude différents pour chacun des deux bras. L'imprécision sur les zéros géométriques du bras à droite de la figure 4.10 que l'on appellera dans ce qui suit *bras1* est supposée 5 fois plus importante que celle du bras à gauche de la figure (*bras2*).

Notre but est de commenter qualitativement la solution trouvée par notre méthode et de montrer l'influence de la propagation des incertitudes géométriques dans le choix de cette solution.

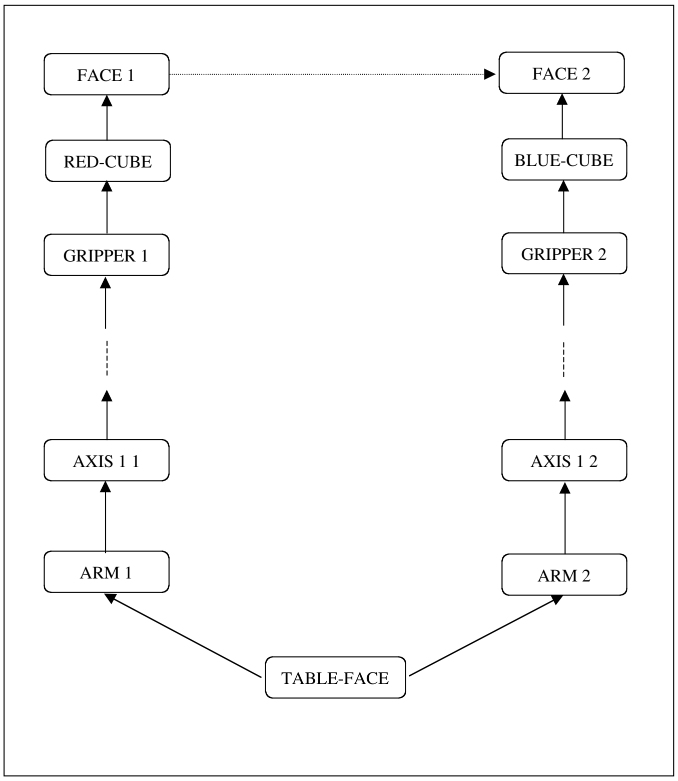

FIG. 4.11 - *Graphe cinématique correspondant au problème.*

La figure 4.13 montre la solution trouvée par le système pour cet exemple. Cette solution a deux particularités remarquables :

1. Le bras 1, moins précis, est replié sur lui-même pour minimiser la propagation des incertitudes sur les paramètres de rotation.
2. Les axes de rotations sont perpendiculaires à la direction commune de la normale des deux faces.

A posteriori, nous pouvons justifier la solution trouvée par quelques remarques simples de calcul varationnel en supposant que les incertitudes sont assez petites pour permettre la linéarisation et sont distribuées suivant une loi gaussienne.

Dans ces conditions, la propagation d'une erreur d'écart-type σ_i sur chacun des paramètres de rotation θ_i autour d'un axe $\vec{u}_i$ donne lieu, pour un point final f d'une chaîne

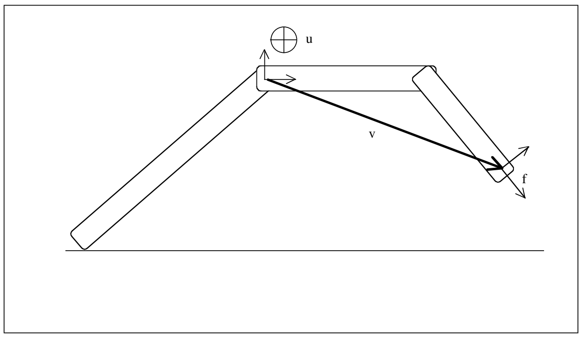

FIG. 4.12 - *Propagation de l'incertitude sur un paramètre de rotation.*

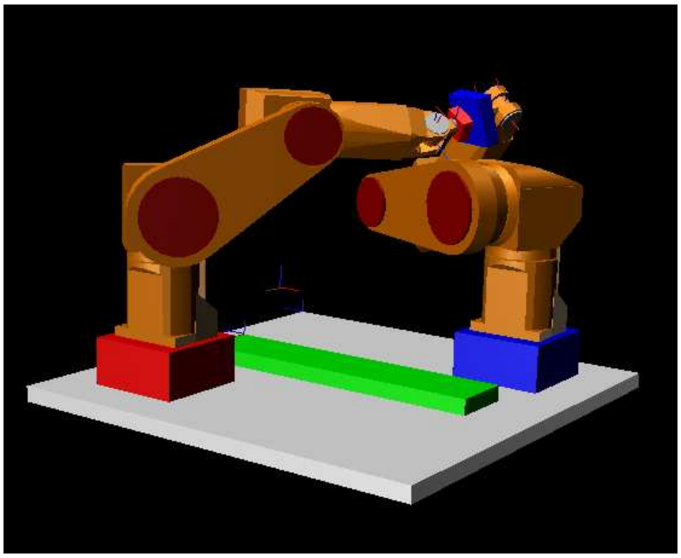

FIG. 4.13 - *Solution trouvée par le système. Le bras le moins précis est replié sur lui-même. De plus, les axes de rotations sont perpendiculaires à la direction commune de la normale des deux faces.*

cinématique (figure 4.12), à une erreur de matrice de variances-covariances $\Sigma_{\vec{\rho}_i}$ sur la position $\vec{\rho}$ de ce point avec :

$$\Sigma_{\vec{\rho}_i} = J_i \sigma_i J_i^T$$

où J_i est la Jacobienne de la position $\vec{\rho}$ par rapport au paramètre θ_i :

$$J_i = \frac{\partial \vec{\rho}}{\partial \theta_i} = \vec{u}_i \wedge \vec{v}_i \tag{4.1}$$

Si on applique le principe de l'accumulation des erreurs gaussiennes, la matrice de variances-covariances de l'erreur totale sur la position du point f sera donnée par :

$$\Sigma_{\vec{\rho}} = \sum_{i=1}^{n} J_i \sigma_i J_i^T$$

où n est le nombre d'articulations dans la chaîne cinématique.

L'idéal serait alors de trouver une configuration qui permet d'annuler les Jacobiennes J_i pour tout $i = 1 \cdots n$.

Pour notre problème, en repliant le bras le moins précis, le système cherche à diminuer le module de l'erreur sur $\vec{\rho}$ pour toutes les rotations du $bras1$. En plaçant tous les axes de rotation perpendiculaires aux deux faces, il diminue l'influence de l'erreur sur la position le long de la normale au deux faces. Notons par ailleurs que la propagation des incertitudes en rotation ne dépend pas du lieu d'assemblage. Le tableau 4.2 donne une description qualitative de la solution.

4.2.4 Tableau récapitulatif

Dans cette partie nous résumons dans le tableau 4.1 les principales caractéristiques et performances de notre algorithme pour cette expérimentation. Le nombre d'évaluation du cycle (nombre de tirage) du graphe correspondant à ce problème a été de l'ordre de 2×10^5. La machine utilisée est un powerPC G3 à 400 MHz.

Dim. Esp. Intégration	50
Dim. Esp. Optimisation	12
Nb cycles	1
Nb repères	28
Nb contraintes inégalités	16
Temps de calcul (secondes)	13

TAB. 4.1 - *Indices sur la complexité du problème.*

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	5.2070	-71.1645	**0.4340**	**-0.0032**	**-0.0024**	0.2522
	2.8104	5.9147	**1.9547**	**0.0104**	**0.0143**	0.0170

TAB. 4.2 - *Les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer (correspondant à la mise en relation des deux faces). Pour la configuration trouvée par le système, ces valeurs ont été calculées empiriquement à partir d'une simulation de Monte-Carlo sur les paramètres libres du modèle en utilisant* 10^5 *points. En gras, les valeurs pour les paramètres contraints par le problème à savoir* t_z, r_x *et* r_y *correspondant à la relation* face sur face.

4.2.5 Discussion

Nous avons illustré notre approche sur un problème d'inversion géométrique sous incertitudes géométriques. Nous avons montré comment la méthode proposée permet de prendre en compte les incertitudes d'une manière homogène et générale. Aucune supposition n'est faite sur les modèles d'incertitude (les formes des distributions) ni sur la linéarité (ou bien la possibilité de linéarisation) du modèle. Cette méthode permet en particulier d'exploiter les redondances possibles du système articulé par rapport à la tâche à effectuer pour essayer de trouver la *meilleure* solution en termes de précision.

4.3 Application à un problème d'inversion utilisant des mesures supplémentaires

4.3.1 Définition du problème

L'objectif de cette section est de décrire comment intégrer une mesure capteur dans un modèle probabiliste de l'environnement. L'intégration d'une mesure a comme objectif d'affiner le modèle initial en introduisant une contrainte supplémentaire sur une pose relative entre deux entités données de l'environnement.

Sur le graphe cinématique correspondant, l'ajout d'une mesure se traduit par l'ajout d'un nouveau lien (arc) entre deux sommets de ce graphe et engendre la création d'un nouveau cycle sur ce dernier.

Plus précisément, nous allons nous intéresser à un problème d'inversion géométrique sous incertitudes qui consiste à poser un cube sur une table (figure 4.14). La seule contrainte pour ce problème serait de mettre une face du cube sur la face attachée à la table. Nous supposons, bien entendu, que le modèle de l'environnement est entaché d'incertitude (incertitudes sur le modèle géométrique du bras ainsi que sur la pose de la table). Nous supposons de plus que les incertitudes attachées à la pose de la table sont beaucoup plus importantes que celles qui sont attachées au modèle du bras.

Nous allons considérer dans ce qui suit deux cas :

1. Dans le premier cas, seul le modèle a priori de la scène sera utilisé (modèle probabiliste du bras ainsi que celui de la table). Aucune mesure supplémentaire ne sera intégrée pour affiner ce modèle.

2. Dans le deuxième, une mesure supplémentaire sera utilisée pour affiner le modèle initial. Nous utiliserons un capteur de distance monté sur l'extrémité du bras et permettant de rendre, dans une configuration donnée du bras, la distance d'un objet par rapport à l'extrémité de ce dernier.

Notre objectif est de montrer comment notre système tient compte de cette nouvelle information pour produire une solution mieux adaptée.

4.3.2 Cas sans utilisation de mesures

Pour ce cas, le problème sera traduit par le graphe cinématique donné dans la figure 4.15.

Intéressons nous aux incertitudes sur l'orientation de la table.Plus un point attaché à la table est éloigné du repère intrinsèque de cette dernière, plus l'imprécision sur sa position devient importante (voir figure 4.16).

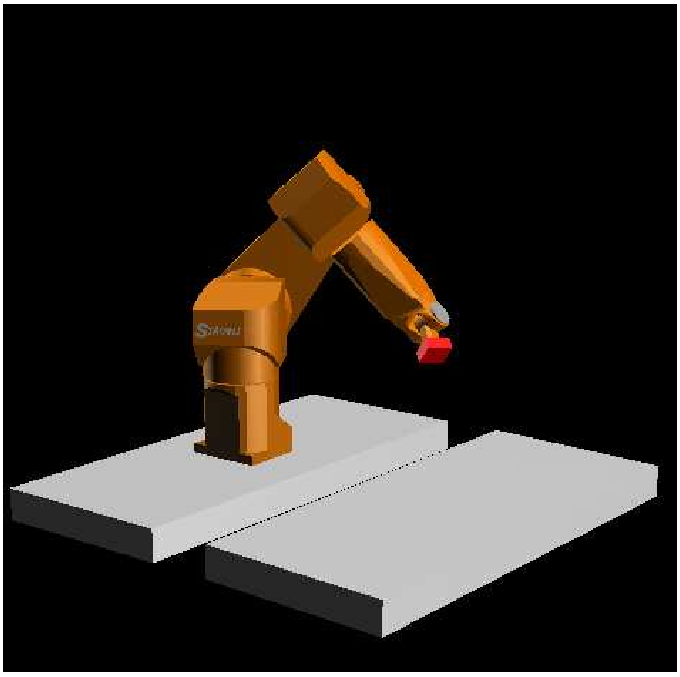

FIG. 4.14 - *Le problème d'inversion géométrique consitant à mettre le cube sur la table.*

Pour notre problème, la précision de la composante de translation de la pose relative entre la table et le cube dépend de la distance du point choisi sur la table par rapport au repère attaché à cette dernière. Les solutions permettant d'avoir une distance minimale entre le cube et le repère attaché à la table seront alors favorisées parce qu'elles engendrent une imprécision minimale sur la distance entre la face attachée au cube et celle attachée à la table.

La figure 4.17 représente la solution trouvée par le système pour ce problème et le tableau 4.3 donne une description qualitative de cette solution.

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	65.4269	-57.5253	**-0.0400**	**-0.0024**	**0.0025**	-0.0303
	7.0710	8.2460	**1.2929**	**0.0101**	**0.0101**	0.0112

TAB. 4.3 - *Les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer (correspondant à la mise en relation des deux faces) et sans utilisation de la mesure capteur. Pour la configuration donnée par le système, ces valeurs ont été calculées empiriquement à partir d'une simulation utilisant* 10^5 *points. En gras, les valeurs pour les paramètres contraints par le problème à savoir* t_z, r_x *et* r_y *correspondant à la relation* face sur face.

Pour montrer l'influence du choix de la solution dans cet exemple, prenons la configu-

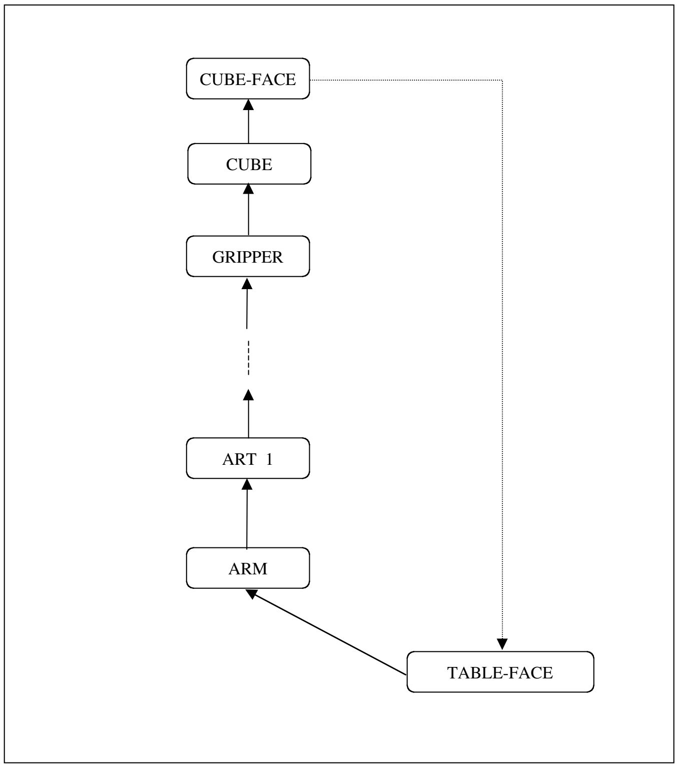

FIG. 4.15 - *Graphe cinématique correspondant au problème et sans utilisation d'une mesure supplémentaire.*

ration donnée dans la figure 4.18. Le tableau 4.4 nous permet de comparer qualitativement les deux solutions.

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	181.3308	-1217.8653	**-0.1736**	**0.0001**	**0.0039**	-1.5295
	15.2856	8.7215	**12.2675**	**0.0111**	**0.0112**	0.0112

TAB. 4.4 - *Les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer (correspondant à la mise en relation des deux faces) pour la deuxième configuration et sans l'utilisation de la mesure.*

4.3.3 Cas avec utilisation d'une mesure capteur supplémentaire.

Supposons maintenant que nous avons une mesure supplémentaire que nous intégrerons dans le modèle initial de l'environnement.

La mesure donnée dans la figure 4.19 représente la distance du point terminal du bras par rapport à la face de la table. Cette mesure est exprimée comme une distribution autour

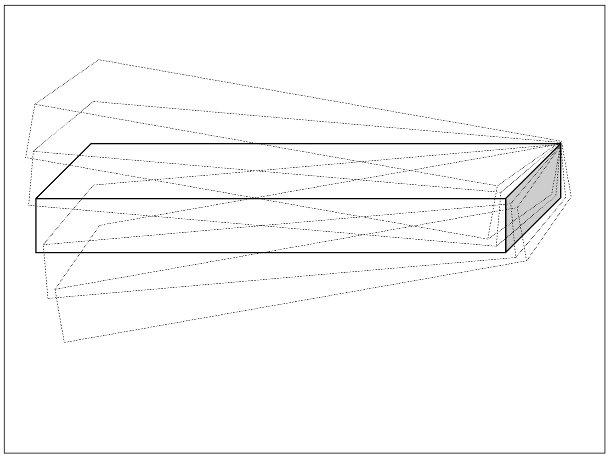

FIG. 4.16 - *L'influence de la propagation de l'incertitude sur l'orientation de la table dépend de la position du point à atteindre.*

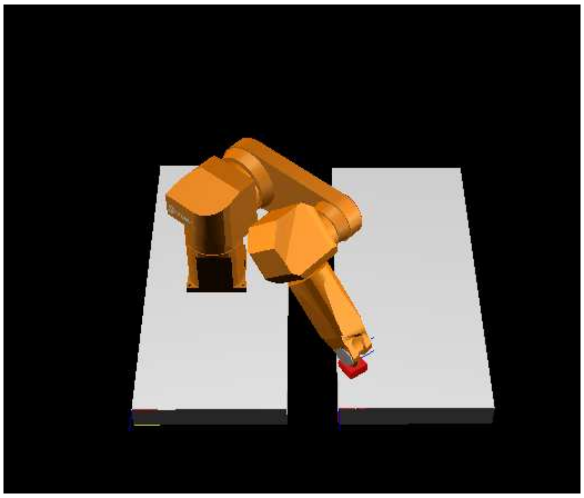

FIG. 4.17 - *Solution trouvée par le système avant l'utilisation de la mesure capteur. Cette solution permet d'avoir une distance minimale du cube par rapport au repère attaché à la table, ce qui minimise l'incertitude sur la distance entre la face attachée au cube et celle attachée à la table.*

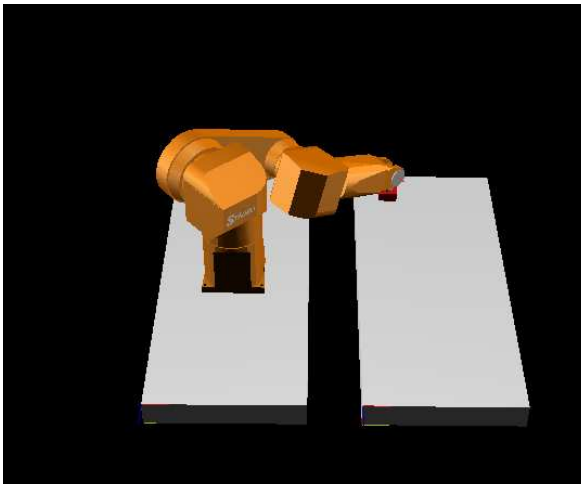

Fig. 4.18 - *Une autre solution possible. Cette dernière ne minimise pas la propagation des incertitudes sur l'orientation de la table. On peut voir en comparant les tableaux 4.4 et 4.3 la différence de qualité entre cette configuration et celle donné par le système (figure 4.17).*

de la valeur retournée par le laser. Cette distribution représente la connaissance dont on dispose sur la réponse de ce capteur (précision). Ce type d'informations est généralement donné par le constructeur. L'intégration de cette mesure permet d'avoir plus d'informations sur l'orientation relative entre le bras et la table, du fait qu'elle restreint la pose relative entre le point final du bras et la table.

Dans ce cas, le problème sera traduit par le graphe cinématique donné dans la figure 4.20. Nous soulignons, par rapport au cas précédent, l'ajout d'un arc supplémentaire correspondant à la mesure.

La figure 4.21 représente la solution trouvée par le système. On peut remarquer que cette solution permet de minimiser la propagation des imprécisions liées au bras car ces dernières deviennent plus significatives que celles liées à l'orientation de la table. Celles-ci ont été réduites en utilisant la mesure. Le tableau 4.5 donne une description qualitative de cette solution.

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	201.2305	-705.0413	**0.0228**	**0.0015**	**0.0018**	-0.0528
	10.0425	9.2750	**0.7257**	**0.0065**	**0.0074**	0.0142

Tab. 4.5 - *Les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer (correspondant à la mise en relation des deux faces) après utilisation de la mesure capteur.*

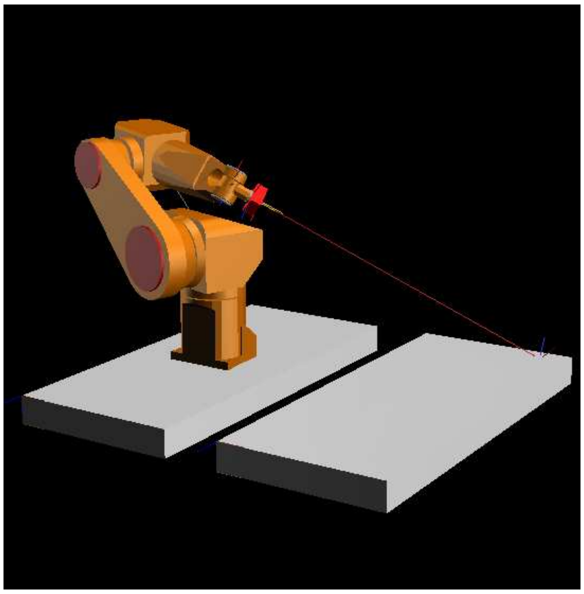

FIG. 4.19 - *Prise d'une mesure sur la distance du repère attaché au laser par rapport à la face de la table.*

4.3.4 Tableaux récapitulatifs

Nous résumons dans les tableaux 4.6 et 4.7 les principales caractéristiques et performances de notre algorithme pour cette expérimentation.

Dim. Esp. Intégration	21
Dim. Esp. Optimisation	6
Nb cycles	1
Nb repères	9
Nb contraintes inégalités	16
Temps de calcul (secondes)	9

TAB. 4.6 - *Complexité et performance pour le premier cas (sans utilisation d'une mesure).*

4.3.5 Discussion

Nous avons présenté un exemple d'intégration de mesures capteurs pour affiner le modèle géométrique de l'environnement. Notre objectif était de montrer la parfaite homogénéité de représentation et d'utilisation entre les informations fournies a priori au système par l'utilisateur, et celles fournies par des mesures capteurs.

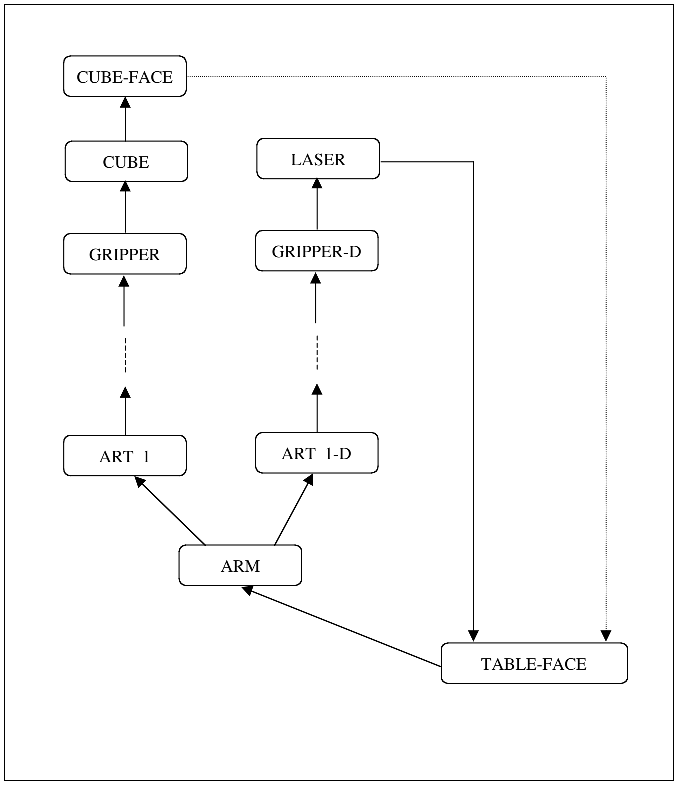

FIG. 4.20 - *Graphe cinématique correspondant au problème après utilisation d'une mesure supplémentaire.*

Dim. Esp. Intégration	21
Dim. Esp. Optimisation	6
Nb cycles	2
Nb repères	15
Nb contraintes inégalités	20
Temps de calcul (secondes)	11

TAB. 4.7 - *Complexité et performance pour le deuxième cas (après utilisation de la mesure).*

4.4 Application à un problème d'étalonnage de pose

4.4.1 Définition du problème

Nous abordons dans cette section le problème d'identification de paramètres à partir d'un ensemble de mesures. Nous nous intéressons plus précisément à un problème d'étalonnage de la pose d'un objet dans une scène à partir de mesures de distance sur des entités appartenant à cet objet.

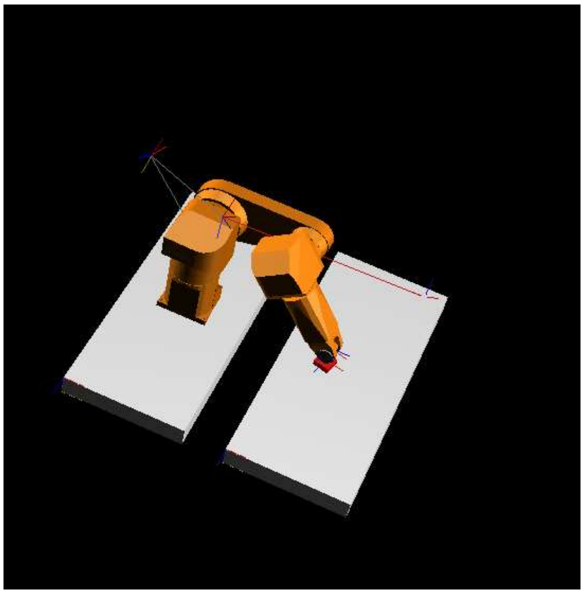

FIG. 4.21 - *Solution avec l'utilisation de la mesure capteur. Comme les incertitudes sur l'orientation de la table ne sont plus significatives, cette solution permet de réduire l'influence des incertitudes sur les articulations du bras.*

Supposons l'exemple présenté dans la figure 4.22. L'objectif est d'identifier la pose d'une voiture dans la scène à partir d'un ensemble de mesures.

Le protocole expérimental est le suivant : un capteur télémétrique laser est attaché à l'extrémité du bras. Dans une configuration donnée, ce capteur permet de mesurer la distance entre ce point et des éléments de surface situés sur la voiture (des faces correspondant aux vitres). On déplace le bras dans plusieurs configurations. Pour chaque configuration, une mesure de distance est prise. Cela est équivalent à l'utilisation d'un ensemble de capteurs placés aux points correspondant aux poses de l'extrémité du bras pour les différentes configurations (voir figure 4.23).

Le graphe cinématique correspondant à ce problème est donné dans la figure 4.24. On peut voir sur la figure 4.25 les points de mesures et les faces correspondantes pour l'ensemble de mesures utilisées.

Nous supposons dans cet exemple que la voiture est placée sur le sol. Les paramètres de pose à estimer seront les coordonnées en x et en y ainsi que l'orientation α sur le plan modélisant le sol. Nous utilisons un ensemble de 5 mesures. Cela se traduit par la duplication du modèle du bras dans les différentes configurations, et l'ajout de 5 liens supplémentaires entre les points définis par les mesures et les faces représentant les vitres de la voiture.

La précision de chaque mesure dépend non seulement de la précision du capteur mais aussi de la configuration dans laquelle cette dernière a été prise. Cela est dû à la propagation des imprécisions liées au modèle du bras (notre instrument de mesure).

FIG. 4.22 - *Problème de l'étalonnage de la pose de la voiture dans l'environnement.*

4.4.2 Tableau récapitulatif

Nous avons conduit une expérience simulée. En plus des imprécisions sur le modèle du bras, nous avons introduit une incertitude gaussienne de moyenne nulle et d'écart-type de $0.5mm$ sur la réponse du capteur. Le tableau 4.8 donne une comparaison des valeurs fixées à la simulation à celles obtenues comme résultat de l'étalonnage.

Le tableau 4.10 donne une description qualitative du résultat de l'étalonnage. Des indices sur la complexité du problème et les performances du système pour la résolution de ce dernier sont résumés dans le tableau 4.9.

	$x(mm)$	$y(mm)$	$\alpha(mm)$
Pose initiale de la voiture	300.0000	-800.0000	-0.7854
Résultat de l'étalonnage	300.5859	-800.1953	-0.7852

TAB. 4.8 - *Les valeurs initiales des paramètres à étalonner et le résultat de l'étalonnage.*

4.4.3 Discussion

L'exemple présenté dans cette section représente une application de la méthode bayesienne pour l'identification des paramètres à partir de mesures. L'utilisation d'une telle

FIG. 4.23 - *La prise de mesures en utilisant un capteur laser.*

Dim. Esp. Intégration	30
Dim. Esp. Optimisation	3
Nb cycles	5
Nb repères	41
Nb contraintes inégalités	20
Temps de calcul (secondes)	19

TAB. 4.9 - *Indices sur la complexité du problème et la performance du système pour cette application.*

méthode pour ce problème permet en particulier :

– de prendre en compte des *a priori* sur les paramètres à estimer et sur le capteur utilisé. Cela permet d'expliciter et d'utiliser les connaissances dont on dispose sur les paramètres de la pose de la voiture ainsi que sur la qualité des mesures données par le capteur.

– de prendre en compte, pour chaque mesure, de la précision de cette dernière en propageant les incertitudes sur les autres paramètres représentant le modèle du bras. Cela permet implicitement d'avoir une *pondération* des mesures en fonction de leurs précisions. Plus la mesure est précise, plus elle aura d'importance pour estimer les valeurs des paramètres qu'on veut étalonner.

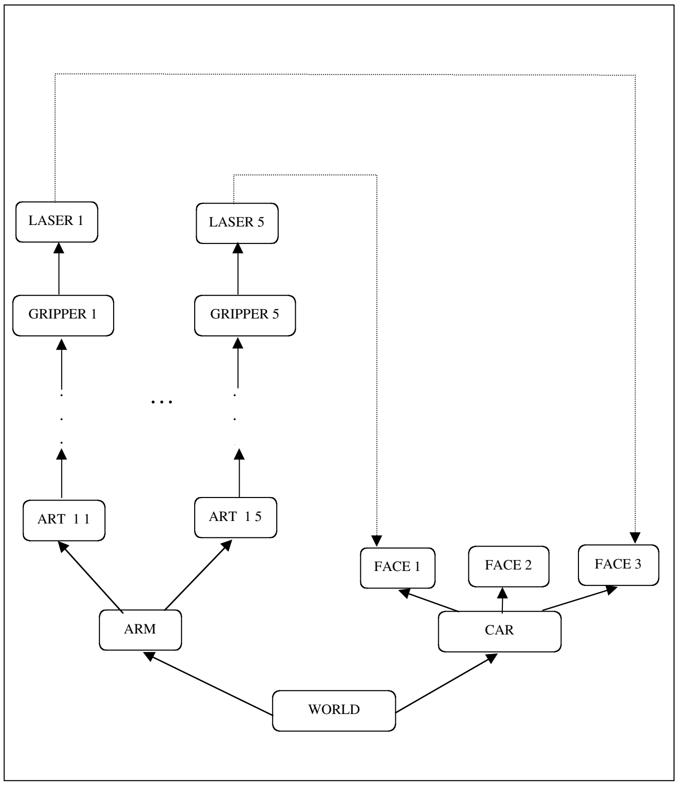

FIG. 4.24 - *Graphe correspondant au problème d'étalonnage.*

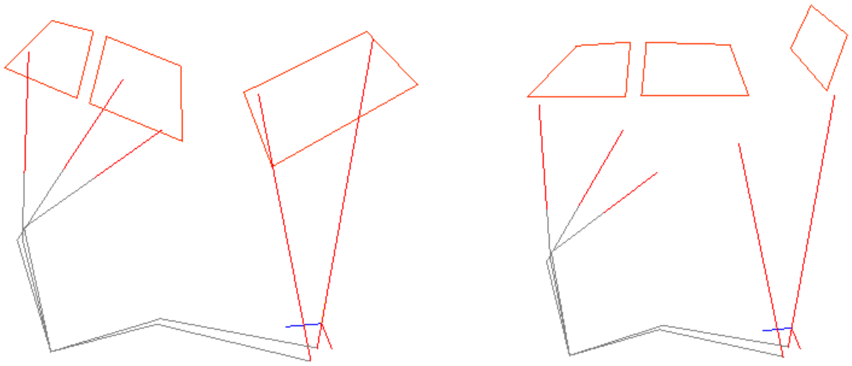

FIG. 4.25 - *Les points de mesure et les faces correspondantes de la voiture. À gauche, les points de mesures et les faces correspondantes lors de la prise des mesures. À droite, les points et les faces à remettre en relations pour le problème d'étalonnage.*

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	165.9974	-107.1010	**0.2141**	0.4401	-0.9388	-0.4409
	4.6612	4.3287	**5.4445**	0.0021	0.0025	0.0041
Cycle 2	171.3668	-133.9731	**-0.3217**	0.4670	-0.6254	-0.3768
	4.7652	2.1952	**4.1345**	0.0024	0.0024	0.0039
Cycle 3	457.2623	-8.0994	**-1.0607**	0.3265	-0.4476	-0.3585
	4.3215	1.4717	**3.2640**	0.0026	0.0021	0.0039
Cycle 4	404.6485	-600.4315	**0.0176**	-0.9623	-1.2690	-2.5372
	5.6778	1.3148	**2.6829**	0.0020	0.0027	0.0053
Cycle 5	248.2639	-1.7194	**0.2510**	-0.7522	-1.4978	-2.2861
	8.0590	0.9047	**3.4238**	0.0021	0.0020	0.0052

TAB. 4.10 - *Pour chaque mesure, les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer. Pour la configuration donnée par le système, ces valeurs ont été calculées empiriquement à partir d'une simulation utilisant* 10^5 *points. En gras, les valeurs pour le paramètre contraint par le problème à savoir* t_z *correspondant à la relation* point sur face.

4.5 Application à un problème d'étalonnage de la pose et des dimensions d'un parallélépipède

4.5.1 Définition du problème

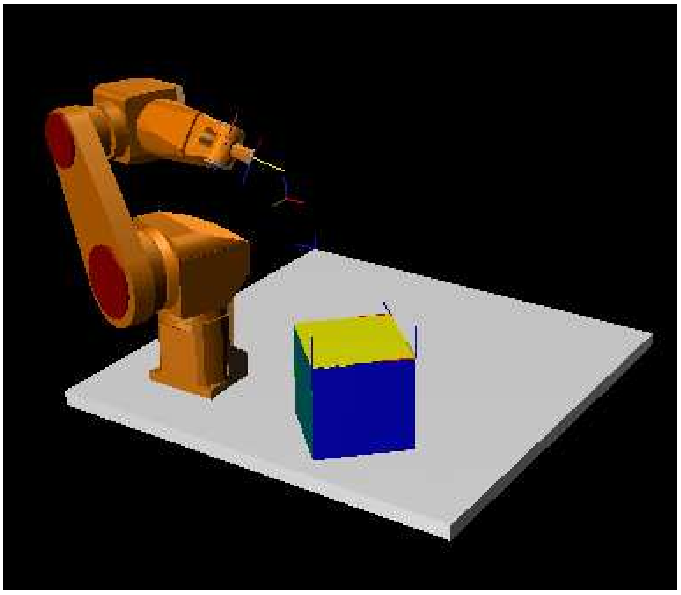

FIG. 4.26 - *Problème de l'étalonnage de la pose et des dimensions d'un parallélépipède.*

Nous nous intéressons dans cette application à l'étalonnage de la pose et de la forme d'un objet 3D. Plus précisément, le but est d'identifier les paramètres de pose d'un parallélépipède sur une table ainsi que les dimensions de ce dernier (figure 4.26).

Le protocole expérimental est le suivant : pour chaque mesure, le robot est amené à une configuration permettant de mettre en contact son point final avec une face du parallélépipède. La répétition de la mise en contact sur les faces du parallélépipède donne lieu à l'ensemble des mesures qui seront utilisées pour l'étalonnage (figure 4.27).

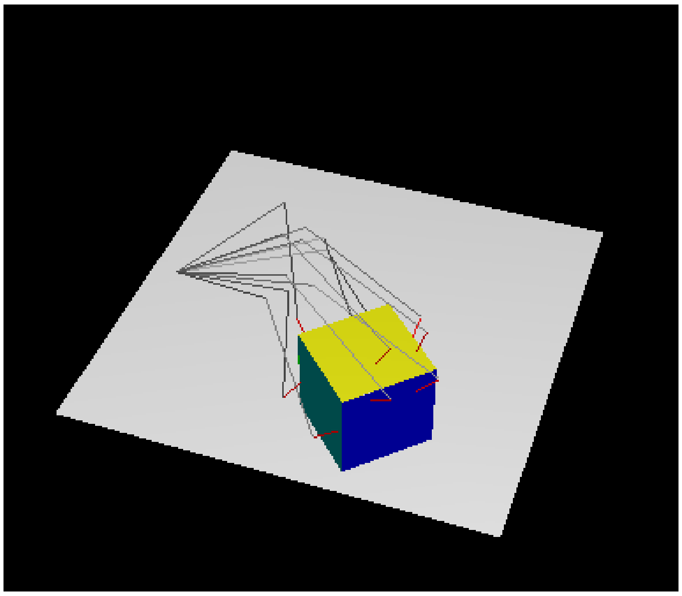

FIG. 4.27 - *Ensemble de mises en contact utilisé pour l'étalonnage.*

Le parallélépipède est supposé placé sur la table. Les paramètres de pose à identifier seront les deux paramètres de position en x et y et le paramètre d'orientation α. Les paramètres sx, sy et sz représentant les dimensions du parallélépipède sont les distances entre les faces parallèles de l'objet deux à deux. La figure 4.28 montre les points de contact et les faces correspondantes à remettre en relation pour résoudre ce problème d'étalonnage.

Pour cet exemple, nous avons utilisé un ensemble de 10 mises en contact. Pour chaque face (sauf pour la face inférieure qui est en contact avec la table), deux mesures ont été prises.

Le graphe cinématique construit pour ce problème est donné dans la figure 4.29. Ce graphe contient, bien entendu, 10 cycles correspondant aux dix mises en contact.

Nous avons conduit une simulation de ce problème. Pour les mesures, nous avons supposé la mise en contact à des distances non nulles du point final du bras par rapport aux faces du parallélépipède. Le tableau 4.11 donne les valeurs des erreurs (distances non-nulles) sur les mises en contact pour les 10 mesures.

Nous soulignons que toutes les erreurs de mise en contact ont des valeurs positives. Cela s'explique par le fait que le point final du bras est toujours du côté inverse de la matière par rapport aux faces du parallélépipède.

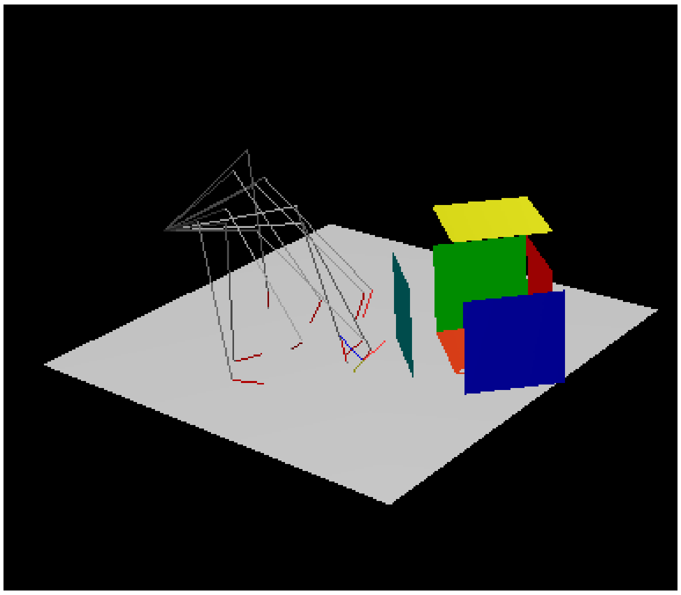

FIG. 4.28 - *Points de contact et faces du parallélépipède à remettre en relation.*

	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5
Err. simulées (mm)	0.677	0.567	0.303	0.792	0.724
	Contact 6	Contact 7	Contact 8	Contact 9	Contact 10
Err. simulées (mm)	0.791	0.883	0.858	0.383	0.111

TAB. 4.11 - *Les valeurs des erreurs lors de la simulation de la mise en contact.*

Le tableau 4.12 donne les valeurs fixées à la simulation et le résultat de l'étalonnage.

	$x(mm)$	$y(mm)$	$\alpha(mm)$	$sx(mm)$	$sy(mm)$	$sz(mm)$
Valeurs de la simulation	900.000	-900.000	0.7854	300.000	300.000	300.000
Résultat de l'étalonnage	900.195	-900.000	0.7853	299.238	299.238	299.238

TAB. 4.12 - *Les valeurs initiales (de la simulation) des paramètres à étalonner et le résultat de l'étalonnage.*

Comme le point final du robot ne peut en aucun cas être du côté de la matière par rapport aux faces, la distribution de probabilité sur les mesures doit forcément prendre en compte cette information. Nous avons utilisé pour cet exemple une distribution *demi Gaussienne* :

$$p(d) = \begin{cases} \frac{2}{\sqrt{2\pi}\sigma_c} e^{-\frac{1}{2}\frac{d^2}{\sigma_c^2}} & \text{si } d \geq 0 \\ 0 & \text{sinon} \end{cases} \tag{4.2}$$

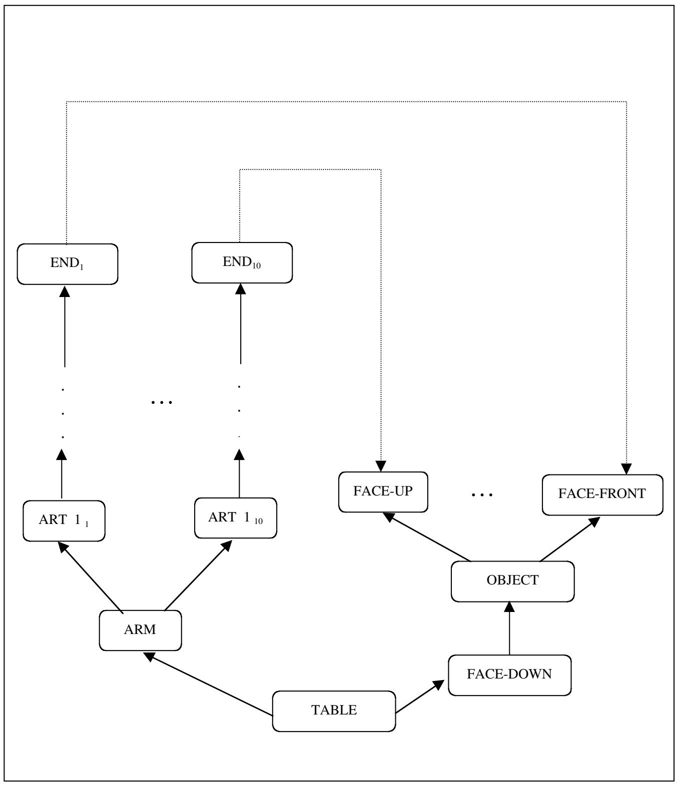

FIG. 4.29 - *Graphe cinématique correspondant au problème d'étalonnage de la pose et des dimensions du parallélépipède.*

où σ_c était de $0.5mm$.

Dim. Esp. Intégration	30
Dim. Esp. Optimisation	6
Nb cycles	10
Nb repères	77
Nb contraintes inégalités	40
Temps de calcul (secondes)	23

TAB. 4.13 - *Indices sur la complexité du problème et la performance du système pour cette application.*

4.5.2 Discussion

Nous avons présenté dans cette partie une application de l'approche proposée pour la spécification et la résolution d'un problème d'étalonnage de la pose et de la forme d'un objet.

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	212.8891	-28.3453	**0.9715**	0.0130	0.8783	-1.9170
	1.5942	0.7618	**1.1370**	0.0050	0.0038	0.0014
Cycle 2	64.5026	-61.4791	**0.8172**	-0.1242	1.3136	-1.8133
	1.5632	0.798	**0.6703**	0.00486	0.0037	0.0014
Cycle 3	152.1722	-205.9615	**1.0891**	-0.4643	0.2038	-0.8607
	1.1265	0.9492	**0.7254**	0.0017	0.0016	0.0049
Cycle 4	24.8071	-213.2750	**1.4797**	-0.3270	0.1361	-1.2425
	1.1664	0.6704	**0.8460**	0.0018	0.0013	0.0049
Cycle 5	24.1094	-142.4724	**1.1870**	0.2187	0.2093	-0.8322
	1.5822	0.6646	**1.40763**	0.0022	0.0046	0.0019
Cycle 6	277.6144	-177.7119	**1.0944**	-0.0529	0.4236	-0.3170
	1.5588	0.4604	**0.4431**	0.0013	0.0050	0.0017
Cycle 7	269.0878	-261.6988	**1.1943**	0.0798	0.1929	2.8028
	0.5140	0.4371	**1.3847**	0.0069	0.0020	0.0016
Cycle 8	46.5764	-59.8367	**0.9798**	-0.1556	0.4813	2.9427
	0.6459	0.6329	**0.6629**	0.0073	0.00177	0.0020
Cycle 9	166.0281	-266.4623	**0.1823**	-0.0776	-1.7300	-2.5276
	0.6201	0.8559	**0.8628**	0.0065	0.0025	0.0032
Cycle 10	294.6391	-63.2253	**0.2209**	1.4013	-0.4221	-1.9743
	0.6253	0.6364	**0.5626**	0.0054	0.0028	0.0035

TAB. 4.14 - *Pour chaque mise en contact, les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer. Pour la configuration donnée par le système, ces valeurs ont été calculées empiriquement à partir d'une simulation utilisant* 10^5 *points. En gras, les valeurs pour le paramètre contraint par le problème à savoir* t_z *correspondant à la relation* point sur face. *À noter que toutes les valeurs moyennes (en gras) issues de la simulation a posteriori sont positives. Cela est le résultat de l'utilisation de l'a priori (distribution) non-symétrique sur les mesures.*

Le but était de montrer l'homogénéité de la représentation des paramètres géométriques de pose et de formes, et surtout la capacité de prendre en compte, d'une manière simple est assez générale, les informations liées au problème auquel on s'intéresse. Le phénomène de la non-interpénétrabilité entre le robot et le parallélépipède a été pris en compte en utilisant une distribution non-symétrique sur la *distance* entre le point final et les faces de l'objet à la mise en contact.

4.6 Application à un problème de positionnement d'un robot

4.6.1 Définition du problème

L'application que nous présentons dans cet exemple correspond à un problème de conception d'une cellule robotique. Nous nous intéressons plus précisément au positionnement d'un robot dans un environnement de travail, dans le but de pouvoir atteindre, dans des conditions optimales, un ensemble de n points de cet environnement.

Nous entendons par *conditions optimales* une configuration de la cellule qui permet de satisfaire au mieux un ensemble de contraintes (liens probabilistes) correspondant aux relations que l'on souhaite obtenir entre le point final du robot et chacun des n points de contrôle correspondant à une tâche donnée. Le but serait alors d'obtenir une position du robot qui permet, en plus de l'accessibilité de ces points, une meilleure précision de la position du point final du robot par rapport à ces derniers.

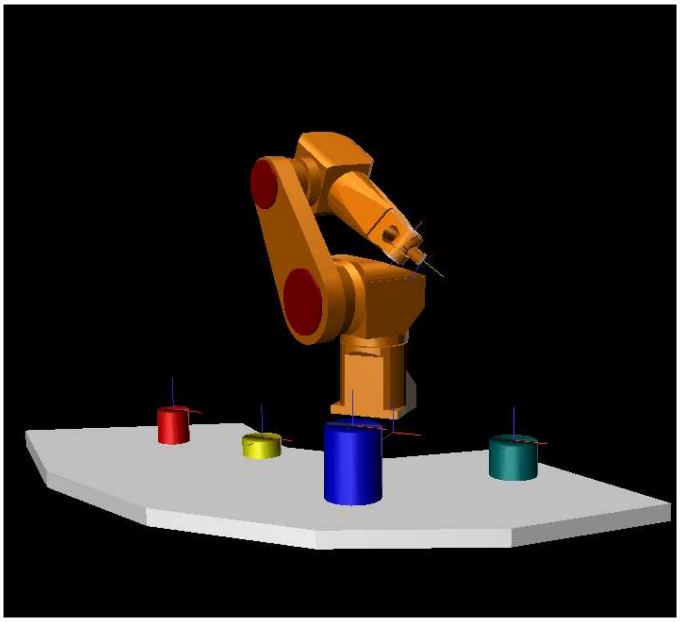

FIG. 4.30 - *Problème de positionnement du robot. Le but est de trouver sa position pour pouvoir atteindre avec une précision maximale les quatre points attachés aux quatre cylindres sur la table.*

Supposons l'exemple présenté dans la figure 4.30. Le robot est supposé reposé sur le sol, ce qui laisse deux degrés de liberté t_x, t_y de la position du repère attaché à la base de ce dernier par rapport au repère de référence de l'environnement. On s'intéresse pour cet exemple à positionner le robot pour pouvoir atteindre 4 points attachés aux cylindres sur la table.

La position du point final du bras par rapport à chaque point de contrôle dépend de la position de la base du bras et de la configuration de ce dernier. Il est nécessaire donc dans la pratique de trouver simultanément la position de la base dans la cellule et les différentes configurations permettant d'atteindre les différents points. Pour un bras à 6 degrés de liberté, cela donne lieu à un problème de $6 \times 4 + 2$ inconnues.

Le graphe cinématique correspondant à ce problème est construit en dupliquant 4 fois le modèle du bras selon le schéma donné par la figure 4.31.

La figure 4.32 présente la solution trouvée par le système pour ce problème, et le tableau

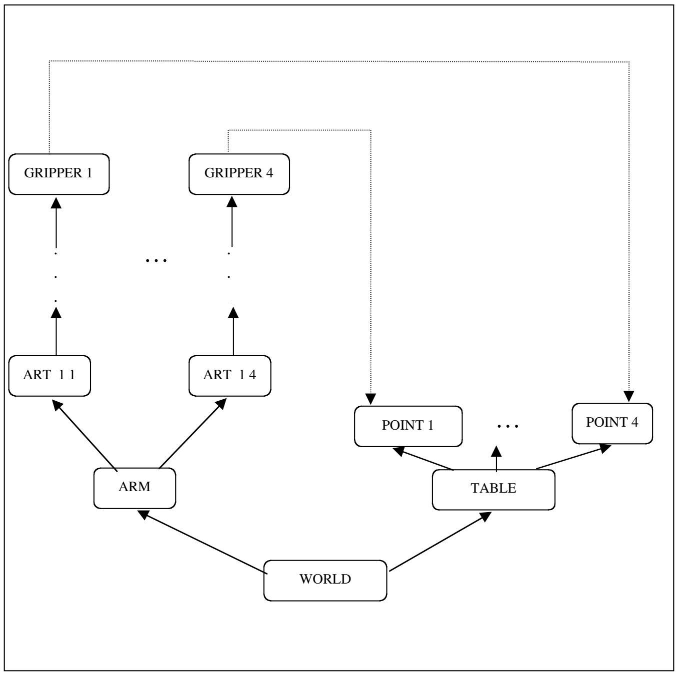

FIG. 4.31 - *Graphe cinématique correspondant au problème du positionnement du robot. Pour chaque point à atteindre, le modèle du bras est dupliqué.*

4.16 donne une analyse qualitative de cette solution.

4.6.2 Tableau récapitulatif

Dim. Esp. Intégration	12
Dim. Esp. Optimisation	26
Nb cycles	4
Nb repères	61
Nb contraintes inégalités	0
Temps de calcul (secondes)	49

TAB. 4.15 - *Indices sur la complexité du problème et la performance du système pour cette application.*

4.6.3 Discussion

Nous avons présenté dans cette partie une application de l'approche proposée pour un problème de positionnement de robot dans un environnement de travail pour pouvoir accéder à un ensemble de points de ce dernier dans les conditions optimales de précision. Le but

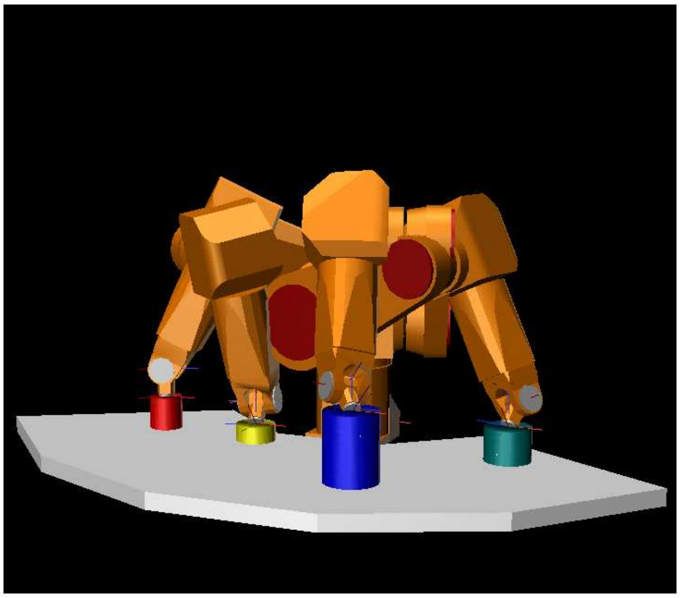

FIG. 4.32 - *Résultat de la résolution du problème de positionnemnt du robot. La figure présente la position du robot et les 4 configurations permettant d'atteindre les 4 points.*

Cycles (à fermer)	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rad)$	$r_y(rad)$	$r_z(rad)$
Cycle 1	**0.0289**	**-0.2989**	**0.0846**	0.8806	-1.5196	1.6783
	0.5899	**0.6487**	**0.6131**	0.0021	0.0010	0.0016
Cycle 2	**0.0464**	**0.0112**	**-0.0378**	0.7368	-0.4281	1.5022
	0.5390	**0.4063**	**0.5019**	0.0017	0.0014	0.0013
Cycle 3	**0.7865**	**-0.5886**	**0.5152**	0.4221	0.7244	-0.5008
	0.6504	**0.7385**	**0.526**	0.0018	0.0013	0.0012
Cycle 4	**0.7071**	**-0.0585**	**0.5098**	0.4730	-0.7235	1.4669
	0.5954	**0.6889**	**0.6099**	0.0014	0.0014	0.0018

TAB. 4.16 - *Pour chaque point de contrôle, les moyennes (première ligne) et les écarts-types (deuxième ligne) a posteriori sur les 6 paramètres du lien à fermer. Pour la solution trouvée par le système, ces valeurs ont été calculées empiriquement à partir d'une simulation utilisant* 10^5 *points. En gras, les valeurs pour les paramètres contraints par le problème à savoir* t_x, t_y *et* t_z *correspondant à la relation* point sur point.

était de montrer la souplesse et la généralité de la méthode proposée, ainsi que la capacité du système à aborder des problèmes dont le nombre de variables peut être important.

4.7 Conclusion

Nous avons présenté dans ce chapitre l'utilisation concrète de l'approche proposée pour un ensemble de problèmes géométriques. Nous avons montré en particulier que la méthodologie de spécification des problèmes associée à la méthode de résolution constitue un outil assez souple et général pour résoudre de nombreux problèmes de CAO robotique.

Chapitre 5

Implémentation

Nous présentons dans ce chapitre une description informatique du système de CAO dont les principes ont été décrits dans les chapitres précédents. L'objectif est de donner un bref aperçu sur ses principaux modules, sur ses fonctionnalités et les principes de son implémentation.

5.1 Présentation générale

L'utilisation d'une représentation graphique présente un intérêt indiscutable pour modéliser des scènes géométriques 3D. La facilité d'utilisation qui découle de cette représentation rend la modélisation des problèmes géométriques plus facile et permet surtout à l'utilisateur de mieux percevoir les solutions calculées par le système pour un problème donné. Dans le cas où l'on s'intéresse aux incertitudes géométriques et à leur propagation, le support graphique présente l'intérêt supplémentaire de permettre une *visualisation* de ces incertitudes en simulant leur propagation sur l'écran. Cela permet en particulier une perception plus concrète de cette propagation dans une configuration donnée, et rend surtout possible de comparer *graphiquement* deux solutions pour un problème donné. Nous revenons dans la suite de ce chapitre sur cette notion de *visualisation des incertitudes*.

Le système est constitué de différents modules organisés autour d'un modeleur d'objets 3D. Le schéma 5.1 donne une description générale des principaux modules du système ainsi que les connexions entre ces derniers.

Les principaux modules du système sont :

- Un interpréteur de scripts pour la construction d'une scène.
- Un système multifenêtre de visualisation graphique 3D. Ce système est basé sur la bibliothèque graphique *Quickdraw3D* proposée par *Apple*.
- Une interface utilisateur pour la spécification probabiliste des problèmes géométriques.
- Un module de résolution dont les principes ont été décrits dans le chapitre 3.

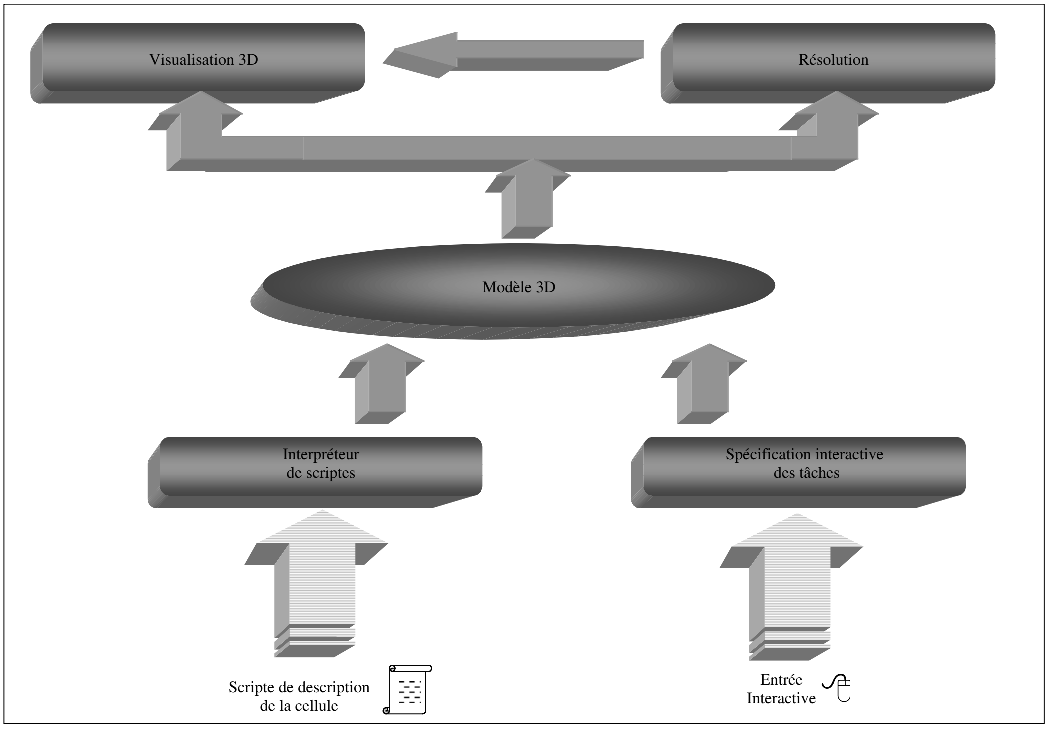

FIG. 5.1 - *Architecture générale du système.*

5.1.1 Construction d'une cellule

La construction d'un modèle 3D d'une cellule nécessite la description des formes de l'ensemble des objets de cette cellule ainsi que les poses et les mobilités relatives entre ces derniers. Nous utilisons pour cela un langage déclaratif permettant de définir les formes des objets et la structure de la scène (positions et mobilités relatives).

La forme d'un objet est décrite comme un ensemble de primitives géométriques de base à savoir : les faces, les arêtes, les points, les sphères et les cylindres. Chaque primitive dispose d'un repère qui lui est associé et qui est attaché au repère intrinsèque de l'objet. Cette description engendre, en plus du modèle 3D interne de ces primitives géométriques, un objet graphique correspondant à la forme de l'objet. Toutefois, et dans le but d'avoir une représentation graphique plus réaliste, nous nous donnons la possibilité d'ajouter à la forme d'un objet une partie purement graphique (n'ayant aucune représentation interne). Cela permet en particulier d'importer des objets graphiques complexes de systèmes commerciaux et de ne modéliser que les parties qui nous intéressent pour un problème donné.

La structure de la scène est définie en décrivant les liens cinématiques probabilistes entre les différents objets. Cela revient à définir les distributions de probabilité sur les paramètres de la pose relative, et le statut de chaque paramètre pour définir les mobilités relatives [1].

1. Les paramètres qui auront le statut X (inconnue) représentent les degrés de liberté des mécanismes

Dans ce qui suit, nous présentons les primitives de base du langage de spécification que nous avons défini.

Création des entités géométriques

La création des primitives géométriques manipulées par le système fait appel à une méthode de création spécialisée pour chaque primitive. La création d'une entité engendre automatiquement la création d'un repère qui lui sera associé :

– *New_Vertex*(x, y)

– *New_Edge*(vertex1, vertex2)

– *New_Face*(list_of_vertexes)

– *New_Sphere*(center, ray)

– *New_Cylinder*(center, ray, direction)

La création d'une primitive géométrique construit, en plus du modèle géométrique interne, l'entité graphique qui lui correspond.

Création des objets

Par définition, un objet est constitué d'un ensemble (qui peut être vide) de primitives géométriques (entités géométriques). La liste de ces entités est donnée comme paramètre à la primitive de création des objets. Par défaut, nous ne considérons pas les incertitudes sur les formes des objets, et la création d'un objet engendre la création du repère intrinsèque de l'objet et d'un lien *fixe* entre ce repère et chacun des repères attachés aux entités géométriques. Dans le cas où on voudrait modéliser des incertitudes sur la forme, des liens probabilistes entre le repère intrinsèque de l'objet et les repères attachés aux entités géométriques doivent être spécifiés :

New_Object(list_of_geometric_entities, additional_graphical_object)

L'ajout du paramètre supplémentaire *additional_graphical_object* permet de créer des objets *graphiques* plus réalistes en ajoutant des formes plus complexes que nous n'avons pas besoin de modéliser. Les objets graphiques supplémentaires sont construits en utilisant des instructions Quickdraw3D ou bien tout simplement chargés à partir de fichiers graphiques.

Description des liens cinématiques probabilistes

La définition d'un lien probabiliste entre deux entités (repères ou bien primitives géométriques) *father_item* et *son_item* nécessite la définition de la distribution de probabilité sur les paramètres de la position Q entre les deux entités, et le 6-vecteur des statuts de ces paramètres :

– *New_Link*(statut_vector, distribution)

---

articulés. Les statuts de ces paramètres peuvent être changés en fonction du problème que l'on voudrait résoudre à un instant donné (cas de l'étalonnagepar exemple)

– *Attach*(father_item, son_item, link)

Dans le cas où *father_item* et *son_item* sont des entités géométriques, la création d'un lien entre ces deux entités génère automatiquement les contraintes supplémentaires sur les paramètres de Q.

La construction d'une cellule se fait à partir d'un script constitué d'un ensemble d'instructions appartenant au langage précédemment décrit. Cette solution permet de définir simplement des cellules complexes et facilite leur modification et leur réutilisation.

Une cellule (scène) est représentée comme un fichier Common Lisp qui contient les instructions nécessaires pour sa construction. Le chargement d'une cellule est fait par l'évaluation par l'interpréteur du script la décrivant. À la fin de cette évaluation, un *viewer* Quickdraw3D (fenêtre graphique) est créé avec le modèle graphique correspondant à la cellule comme modèle.

Le système proprement dit (sans l'interpréteur et le système de visualisation) est écrit en C++ dans l'esprit d'une conception orientée objet. Il est interfacé à l'interpréteur Common Lisp pour les fonctions de haut niveau. La conception objet permet en particulier de rendre les éventuelles extensions du système plus faciles et plus intuitives. L'exploitation des notions d'héritage et de méthodes virtuelles permet d'introduire facilement des nouvelles fonctionnalités en définissant des nouvelles sous-classes. Par exemple, pour ajouter une nouvelle primitive géométrique et un nouveau type de contact, il suffit d'implémenter les méthodes virtuelles de leurs super-classes *Geometric_Entity* et *Contact* respectivement.

Pour pouvoir exploiter le système sur d'autres plates-formes pour lesquelles les bibliothèques graphiques Quickdraw3D ne sont pas disponibles, il est possible de compiler le système en utilisant une directive de compilation pour construire une représentation de l'environnement sans le modèle graphique correspondant. Cela nous a permis de porter nos pogrammes, qui ont été développés sur une plate-forme MacOS, sur une plate-forme Unix sans aucun changement. Le même principe peut être utilisé pour remplacer la bibliothèque graphique Quickdraw3D par une autre, en remplaçant le contenu des méthodes construisant les modèles graphiques pour chaque primitive.

5.1.2 Système Graphique

Une cellule géométrique construite à partir d'un script est associée à une fenêtre graphique (figure 5.2). Le système de visualisation 3D implanté est basé sur la bibliothèque graphique *Quickdraw3D* qui fait partie du produit *Quicktime* développé et proposé par la société *Apple* pour les systèmes MacOS et Windows95/98/NT. Cette bibliothèque propose, en plus des fonctionnalités graphiques permettant la création, le positionnement et le déplacement des différentes primitives graphiques, un *viewer* intégré qui peut être utilisé d'une manière très simple par n'importe quelle application. L'application a la charge de construire un *groupe* graphique représentant le modèle à visualiser, et de passer ce groupe comme paramètre au viewer. Ce dernier prend en charge le calcul des faces cachées et de l'éclairage, et l'affichage sur écran en utilisant différents types de rendus à savoir : un rendu réaliste, un rendu en fils de fer ou bien un rendu en nuage de points (figure 5.3). Il permet aussi de changer les points de vue par rapport à la scène graphique d'une manière interactive sur écran, ou bien en définissant des points de vue particuliers représentés par des caméras virtuelles définies par l'utilisateur.

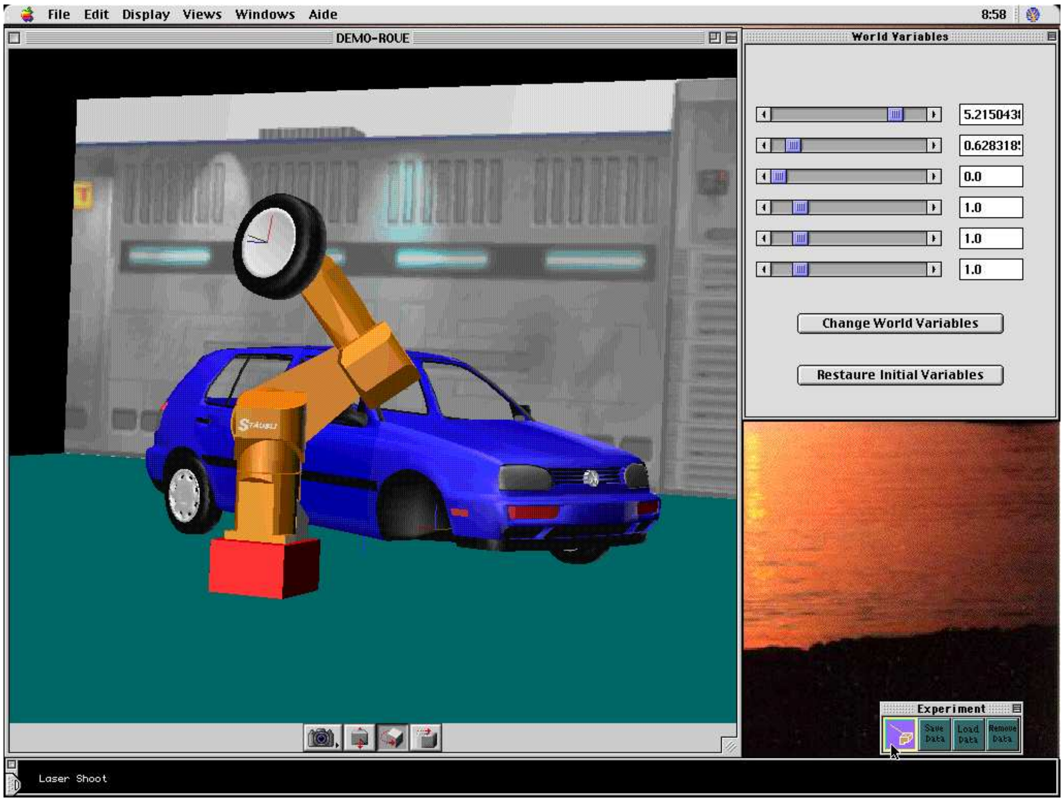

FIG. 5.2 - *Exemple d'écran du système.*

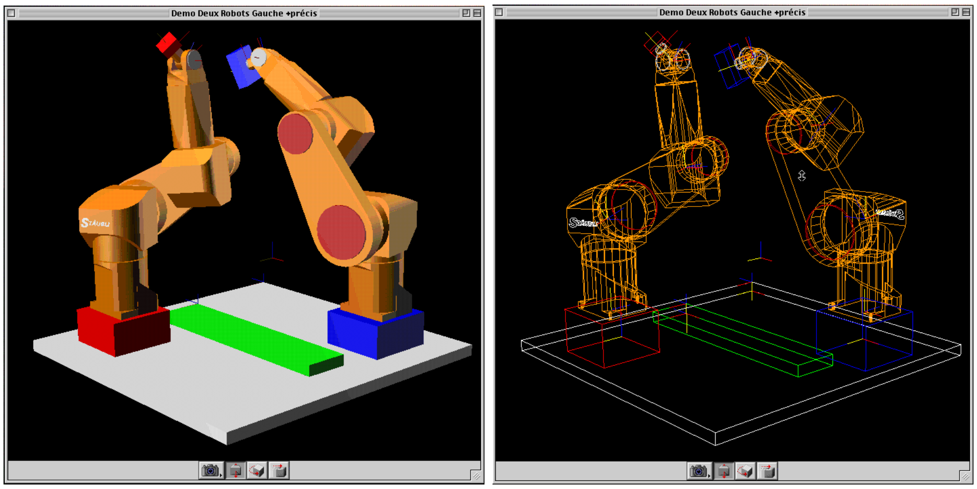

FIG. 5.3 - *Utilisation du viewer et les différents modes de rendu. À gauche le viewer avec un rendu réaliste. À droite le viwer avec un rendu en fils de fer.*

À la construction d'une cellule, le système génère automatiquement, en plus de la fenêtre graphique, une boîte de dialogue constituée d'un ensemble de curseurs représentant les valeurs articulaires des mécanismes articulés simulés (figure 5.4). Ces curseurs permettent à l'utilisateur de faire bouger directement ces mécanismes dans l'espace des configurations. D'une manière analogue, la résolution d'un problème géométrique provoque la mise à jour des valeurs articulaires qui étaient inconnues.

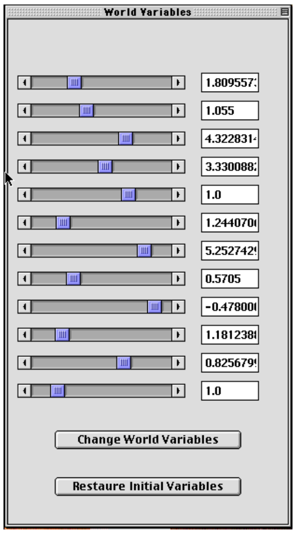

FIG. 5.4 - *Curseurs correspondant aux inconnues d'une cellule.*

Les degrés de liberté (articulations) de la cellule sont définis dans le script décrivant cette dernière. Il s'agit tout simplement des paramètres ayant la valeur X (Inconnu) comme statut. Toutefois, il est possible de modifier les inconnues du problème intéractivement en utilisant une boîte de dialogue (figure 5.5).

Visualisation des incertitudes

Comme la représentation interne des positions relatives est décrite par des distributions de probabilité et non par des valeurs scalaires, il est souhaitable d'utiliser une représentation graphique capable de rendre compte de cet aspect probabiliste des positions des objets dans la scène. Cela revient à trouver une manière de *visualiser* l'état de connaissance probabiliste représentant le modèle de la scène. Cette visualisation permet en particulier à l'utilisateur d'avoir une idée plus concrète de la notion d'imprécision du modèle, et permet surtout de se rendre compte des différences d'incertitude liées à différentes configurations.

Le principe de la méthode proposée est de simuler les incertitudes sur les positions des différents objets graphiques. Cela est réalisé par une simulation de Monte-Carlo qui consiste à tirer les paramètres des poses des objets dans la scène graphique à partir des distributions de probabilité sur ces paramètres. Au lieu d'afficher, à un instant donné, un objet dans une position fixe dans la scène, le principe de cette visualisation consiste à effectuer, avec une fréquence donnée, des tirages représentant l'état de connaissance sur les poses des objets

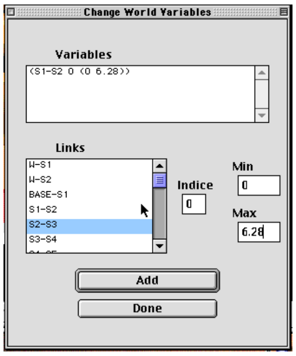

FIG. 5.5 - *Changement des variables (inconnues) de la cellule. Pour chaque nouvelle variable inconnue, le lien correspondant, l'indice du paramètre (de 0 pour t_x à 5 pour r_z) et le domaine de variation doivent être spécifiés.*

dans la scène, et d'afficher chaque fois la réalisation (configuration de la scène) obtenue par le tirage.

Plusieurs possibilités se présentent pour réaliser cette visualisation. La plus intuitive serait de garder plusieurs réalisations (plusieurs tirages) de la scène avec un degré de transparence et d'afficher simultanément ces réalisations. Cela aurait comme effet d'avoir des régions floues correspondant aux objets dont les positions sont entachées de plus d'imprécision. Cette solution nécessite une grande quantité de mémoire pour dupliquer et afficher plusieurs fois le modèle de la scène, et nécessite surtout l'utilisation d'un accélérateur graphique pour afficher le modèle et calculer les transparences avec une fréquence acceptable.

La solution retenue finalement pour la visualisation des incertitudes consiste à afficher à chaque cycle une seule réalisation du modèle (configuration). Cela permet d'avoir une fréquence d'affichage assez élevée, ce qui donne une bonne perception des incertitudes sur le modèle géométrique de la scène.

5.1.3 Spécification d'un problème

Étant donné une cellule, la spécification d'un problème consiste à ajouter au graphe cinématique représentant le modèle initial de cette cellule un ensemble de liens cinématiques supplémentaires (contraintes supplémentaires), créant ainsi de nouveaux cycles. La résolution d'un problème est basée sur ce nouveau graphe cinématique.

Nous avons réalisé une interface de spécification simple et intuitive pour spécifier les problèmes.

Deux modes de spécification sont possibles :

- Spécification par relation Repère-Repère.

- Spécification par relation Entité-Entité.

Spécification par relation *Repère-Repère*

La spécification par une relation de type *Repère-Repère* consiste à choisir, à partir de la liste de tous les repères de la scène, une paire ordonnée de repères, puis à définir pour chaque paramètre définissant la pose relative entre ces deux repères, une distribution de probabilité définissant le niveau de contrainte que l'on souhaite obtenir sur ce paramètre. La sélection de la distribution est faite à partir d'un menu permettant de définir la forme paramétrique de la distribution choisie, ainsi que les valeurs des paramètres de cette distribution (voir figure 5.6).

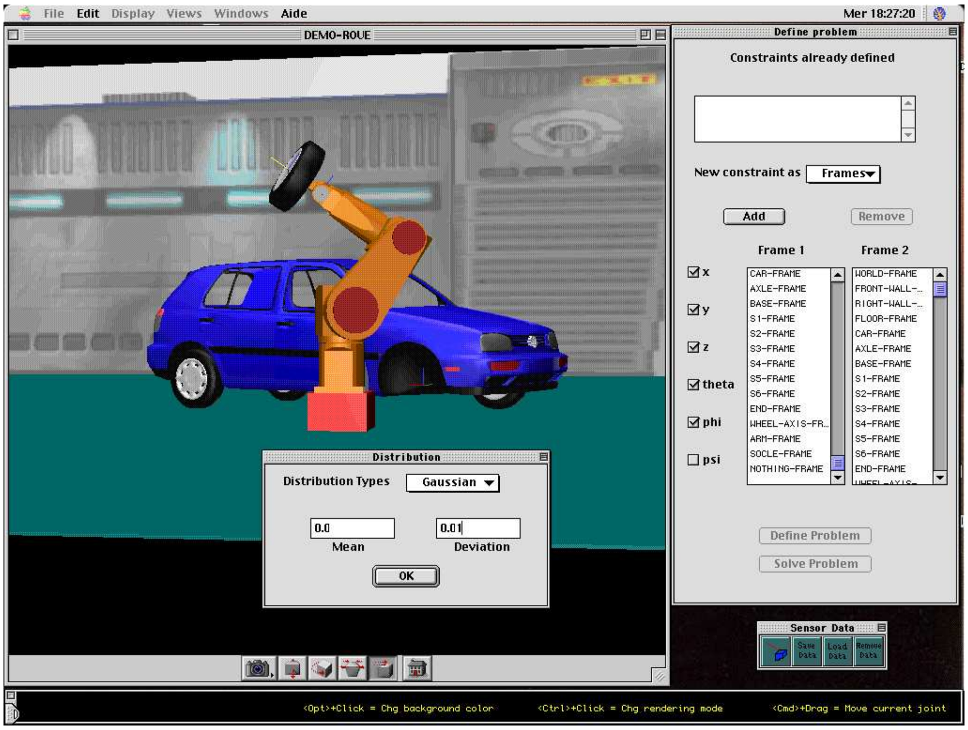

FIG. 5.6 - *Spécification des contraintes en mode* Repère-Repère. *Les deux repères sélectionnés dans la boîte de dialogue seront mis en relation (celui de gauche par rapport à celui de droite). Le choix des paramètres à restreindre parmi les six possibles (x, y, z, theta, phi, psi) sur la pose relative de ces deux repère se fait en spécifiant la distribution sur chacun de ces paramètres.*

Spécification par relation *Entité-Entité*

La spécification par une relation de type *Entité-Entité* consiste à choisir à partir de la liste de toutes les entités (primitives) géométriques présentes dans la cellule (faces, arêtes, points, sphères, cylindres), une paire ordonnée d'entités, et de définir les distributions de probabilité pour fixer les niveaux de contraintes imposés par la relation entre ces deux entités. Le système génère automatiquement les contraintes inégalités supplémentaires correspondant au type de la relation (voir figure 5.7).

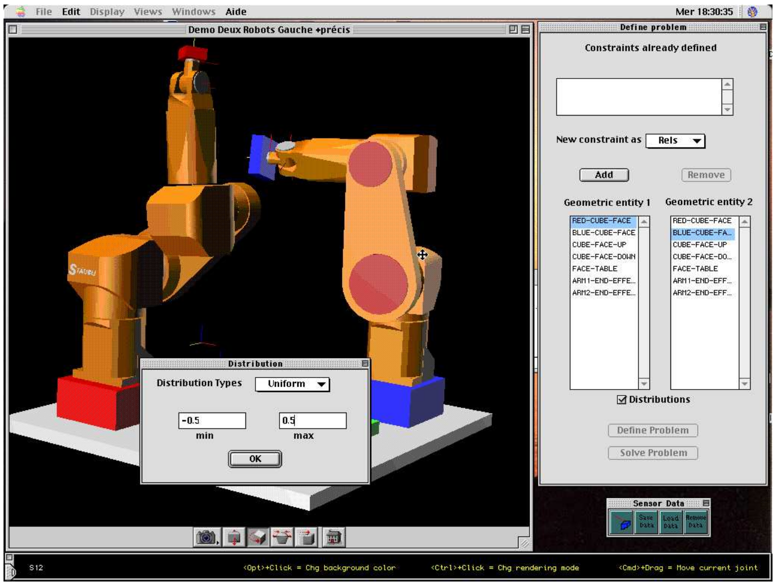

FIG. 5.7 - *Spécification des contraintes en mode* Entité-Entité. *Les deux entités sélectionnées dans la boîte de dialogue seront mises en relation (celle de gauche par rapport à celle de droite). Les paramètres à restreindre sont identifiés automatiquement à partir des types des entités sélectionnées et l'utilisateur est invité à donner les distributions souhaitées sur ces paramètres.*

5.1.4 Utilisation des mesures

Dans ce travail, nous avons utilisé des mesures de type *distance* pour conduire des expérimentations virtuelles. Le système proposé contient un ensemble de fonctionnalités permettant de simuler et d'utiliser ce type de mesures (figure 5.8).

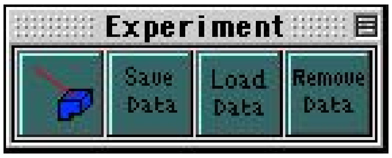

FIG. 5.8 - *Outils pour l'acquisition et l'intégration des mesures capteur.*

Acquisition des mesures

- ***Cas d'une mesure capteur***
  Pour simuler un capteur de distance, le système permet de définir un repère de la scène sur lequel le capteur va être monté ainsi que la dimension du capteur (1 à 6) (figure 5.9). Pour simuler une éventuelle erreur sur la réponse du capteur, l'utilisateur est invité à spécifier une distribution de probabilité représentant ses a priori sur la précision du capteur simulé.

  Dans le cas général, le capteur (le repère qui lui est attaché) peut être fixé sur l'extrémité d'un système articulé permettant de le positionner. Le procédé d'obtention de mesures consiste donc à faire bouger le mécanisme articulé à une configuration donnée dans laquelle on souhaite prendre une mesure, et de calculer la pose (un vecteur de 1 à 6 valeurs) entre le repère attaché au capteur et une entité géométrique de l'environnement. Une erreur représentant la précision du capteur et tirée suivant la distribution donnée par l'utilisateur est ajoutée par la suite à la mesure initiale.

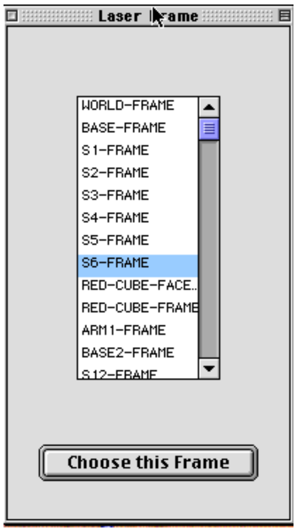

FIG. 5.9 - *Définition du repère sur lequel le laser va être monté.*

- ***Cas d'une relation de contact***
  Pour ce cas, la prise de mesures consiste à essayer d'établir une relation de contact entre deux entités de l'environnement. Cette mise en contact peut être effectuée de deux manières :

  - Soit en modifiant manuellement les valeurs articulaires du système pour ramener la distance entre les deux entités à une valeur proche de zéro. Toutefois, la contrainte de non interpénétrabilité entre les deux entités doit être assurée soit par le système de CAO qui empêche les collisions, soit par l'utilisateur lui-même.
  - Soit en utilisant le système de résolution en spécifiant un problème de mise en relation entre les deux entités. Les distributions utilisées pour spécifier ce problème

doivent assurer la contrainte de non interpénétrabilité entre les deux entités. Par exemple

$$p(d) = \begin{cases} \frac{2}{\sqrt{2\pi}\sigma_c} e^{-\frac{1}{2}\frac{d^2}{\sigma_c^2}} & \text{si } d \geq 0 \\ 0 & \text{sinon} \end{cases} \tag{5.1}$$

Pour les deux cas (mesure capteur et relation de contact), l'itération de ce procédé conduit à un ensemble de mesures. Chaque mesure est représentée par un 4-uplet contenant :

1. La configuration (valeurs articulaires) dans laquelle la mesure a été prise. Ce sont les valeurs des codeurs dans le cas d'un robot.
2. L'identité du repère correspondant au capteur. Cela permet en particulier d'utiliser plusieurs capteurs à la fois.
3. L'identité de l'entité représentant la cible de la mesure.
4. La valeur de la mesure (avec une erreur additionnelle) à cette configuration. Cette valeur est de 0 pour le cas des relations des contacts.

Intégration des mesures

Les mesures générées suivant le protocole décrit ci-dessus sont enregistrées sous la forme d'un fichier. En plus des mesures, le fichier peut contenir une instruction de changement des statuts des paramètres. C'est en particulier le cas d'un étalonnage où les variables inconnues qui ont servi à la prise des mesures ne seront plus les inconnues du problème.

Le chargement (intégration) d'un fichier de n mesures engendre :

- La duplication n fois du modèle (partie du graphe) correspondant au système articulé. Chaque duplication correspond à une mesure (une configuration).
- La création et l'ajout de n liens correspondant aux mesures. La distribution de probabilité associée à ces liens représente l'a priori de l'utilisateur sur la réponse du capteur. Cette distribution doit être spécifiée par l'utilisateur.
- Dans le cas d'étalonnage, le changement des statuts des paramètres pour mettre les statuts des paramètres à étalonner à X (Inconnu).

5.2 Conclusion

Le developpement d'un système prototype de CAO nous a permis de conduire une expérimentation assez poussée des méthodes que nous proposons. Cela nous a facilité considérablement la spécification des problèmes et l'appréciation des résultats de leur résolution grâce à l'utilisation d'un support graphique.

Nous nous sommes efforcés, lors du développement de ce prototype, d'avoir un système facilement extensible et réutilisable en adoptant une conception par objets. Le but était d'obtenir un programme assez ouvert et utilisable sur différentes plates-formes.

Chapitre 6

Perspectives et conclusion

Dans ce chapitre, nous présentons certaines directions de recherche et nos conclusions sur le travail décrit dans cette thèse. Tout d'abord, un ensemble d'extensions et d'améliorations possibles de l'approche proposée sera présenté. Puis, nous aborderons nos conclusions sur ce travail en rappelant les résultats théoriques et expérimentaux obtenus.

6.1 Discussion et perspectives

6.1.1 Extension de la méthode de modélisation

La méthode de modélisation utilisée étant basée sur la spécification des fonctions directes, l'intégration des paramètres non géométriques ne pose aucun problème particulier. Dès lors qu'on est capable de donner un modèle direct pour un phénomène donné en utilisant un ensemble de paramètres, la nature de la méthode utilisée (formalisme bayesien) permet d'exploiter implicitement les éventuelles erreurs de modélisation lors de la résolution des problèmes inverses.

Par exemple, nous pensons que la prise en compte des efforts statiques dans les mécanismes articulés ainsi que les aspects dynamiques est tout à fait envisageable. Cela ne pourrait qu'accroître les capacités du système concernant les phénomènes non forcément géométriques. L'utilisation d'un paramètre de température, abordée au chapitre 4, est un autre exemple concret de cette possibilité.

6.1.2 Amélioration de la méthode d'optimisation

L'un des principaux reproches fait à l'utilisation des algorithmes génétiques comme méthode d'optimisation globale est l'absence de résultats significatifs sur la convergence de ces algorithmes [Rudolph94]. Nous pouvons justifier l'utilisation de ce type d'algorithme dans l'approche que nous avons proposée par les arguments suivants :

1. L'utilisation d'un algorithme de type "recuit simulé" pose dans la pratique le même problème. La convergence de ce type d'algorithme n'est démontrée que pour des conditions très contraignantes quant à la décroissance de la température de l'algorithme

[Laarhoven87]. L'application de ces conditions en pratique rend l'algorithme inutilisable et peu différent d'une recherche systématique dans l'espace des solutions.

2. La présence pour un algorithme génétique d'une population de solutions (et non d'une seule solution à un instant donné) permet plus de robustesse dans les calculs. Cet avantage est plus significatif dans notre cas où la fonction objectif n'est connue que par une approximation.

3. La présence de plusieurs solutions permet un calcul multiprécision de la fonction objectif. L'augmentation de la précision de l'approximation au fur et à mesure du procédé d'optimisation permet de commencer par des évaluations peu coûteuses de la fonction objectif, puis de départager les différentes solutions par la suite.

4. La facilité de la mise en œuvre et de la parallélisation de ces algorithmes.

Toutefois, l'utilisation d'une méthode locale en fin du processus ne peut qu'améliorer la qualité des solutions trouvées. Nous pensons qu'il est possible dans la plupart des cas de calculer la dérivée de la fonction objectif. Pour ce faire, nous pouvons par exemple exploiter la propriété suivante :

$$g(x) = \frac{\partial \left[\int F(x,y)dy\right]}{\partial x} = \int \left[\frac{\partial F(x,y)}{\partial x}\right] dy$$

pour toute fonction F dérivable. Cela permet de calculer une approximation de cette dérivée par une méthode de Monte-Carlo en écrivant :

$$g(x) \simeq \sum_i \left[\frac{\partial F(x,y^{(i)})}{\partial x}\right]$$

L'utilisation de cette dérivée permet d'appliquer une méthode locale (descente du gradient par exemple) au voisinage de la solution trouvée par l'algorithme génétique.

Une autre direction de recherche, pour l'amélioration de la méthode de calcul, est d'étudier la possibilité d'appliquer des méthodes probabilistes variationnelles [Jaakkola99]. Ces méthodes deterministes permettent une approximation plus performante de l'inférence sur les modèles probabilistes.

6.1.3 Extension pour aborder le problème de planification de trajectoire

Dans ce travail, nous nous sommes intéressés à trouver, pour un problème de géométrie, une configuration finale satisfaisant au mieux les contraintes de la spécification sans se préoccuper de la manière de passer d'une configuration initiale vers la solution retenue. Pour les problèmes d'inversion géométrique pour un mécanisme articulé, ce problème de planification de trajectoire sous incertitudes n'est pas des plus triviaux. L'extension du système pour aborder ce problème de planification de trajectoire nous paraît envisageable. En particulier, nous souhaitons pouvoir exprimer explicitement les incertitudes sur les paramètres de poses des objets et les prendre en compte dans la planification de la trajectoire entre la configuration initiale et la solution retenue.

6.1.4 Nouvelles applications potentielles

En plus des applications en robotique et en CAO robotique que nous avons abordées dans cette thèse, les potentialités d'exploitation de l'approche proposée nous paraîssent prometteuses. La généralité et l'homogénéité de nos méthodes permettent, a priori, l'utilisation de cette approche sans un important effort supplémentaire pour les adapter à une application particulière.

Par exemple, l'utilisation de ces méthodes pour l'analyse et la synthèse des tolérancements pour la fabrication et l'assemblage des pièces mécaniques nous paraît très prometteuse. Pour cette application, la méthode que nous avons proposée présente un intérêt du fait qu'elle permet une exploitation maximale de toutes les informations disponibles sur les tolérances intermédiaires, sans se préoccuper des conditions nécessaires à la linéarisation du modèle comme le proposent la plupart des travaux qui ont été fait dans ce domaine [Gaunet94, Whitney94, Sanderson97].

Cette application sera certainement développée dans le futur proche. Elle fait l'objet d'une discussion avec la société Dassault-Systèmes pour une éventuelle application de nos méthodes pour l'analyse des tolérancements dans le logiciel de CAO CATIA.

6.2 Conclusion

Nous avons présenté dans ce mémoire nos travaux sur la spécification bayesienne des problèmes de géométrie en CAO robotique et leur résolution.

Nous nous sommes efforcés de répondre d'une part, à la représentation et la manipulation des incertitudes géométriques, d'autre part, à l'utilisation de ces représentations lors de la résolution de problèmes géométriques sous incertitudes.

L'approche proposée est basée sur le formalisme bayesien pour la spécification et la résolution. Elle peut être vue comme une extension de la notion de contrainte géométrique. Cette notion est utilisée comme étant, dans le cas général, une distribution de probabilité sur les paramètres et non pas une égalité ou bien une inégalité. Dans un souci de généralité, aucune supposition n'a été faite sur les formes des distributions et sur les amplitudes des incertitudes liées aux modèles utilisés.

Pour la modélisation d'un problème, les mécanismes articulés et leur environnement de travail sont assimilés à des corps rigides. La modélisation des poses relatives entre ces derniers est donnée par des distributions de probabilité sur les paramètres de ces poses. De la même manière, les objectifs sont donnés en termes de contraintes (distributions) à établir ou bien à maintenir sur les poses relatives entre les corps rigides.

La structuration d'un problème ainsi spécifié sous la forme d'un graphe cinématique permet d'engendrer automatiquement une fonction objectif à maximiser. Cette fonction correspond à la distribution de probabilité sur l'espace des solutions et resulte de l'inférence bayesienne à partir de la distribution conjointe sur l'ensemble des paramètres du problème.

Dans le cas général, la fonction objectif ainsi obtenue contient une intégrale multidimensionnelle. La dimension de l'espace d'intégration dépend, bien entendu, du problème et peut être très importante. De plus, cette fonction est multimodale dans la plupart des cas, ce qui rend l'utilisation d'une méthode d'optimisation locale inadaptée.

Le problème de l'intégration sur des espaces de grandes dimensions est abordé par une méthode numérique de type Monte-Carlo. Ces méthodes présentent l'intérêt de bien s'adapter aux espaces de grandes dimensions. De plus, elles permettent dans notre cas de prendre en compte d'une manière simple les contraintes géométriques d'inégalités, en utilisant un tirage par rejet.

Ce procédé d'intégration est contrôlé pour l'optimisation de la fonction objectif par un algorithme génétique adaptatif permettant un calcul multiprécision de la fonction objectif. Cela permet en particulier de commencer l'optimisation avec des estimations de la fonction objectif qui peuvent être peu précises, et d'affiner ces estimations au fur et à mesure. Pour remédier au problème de la complexité de la forme de la fonction objectif pour les problèmes très contraints, une notion de température inspirée des méthodes de recuit simulé a été introduite. Cette notion permet une relaxation provisoire des contraintes au début du procédé d'optimisation. Elle permet par la suite de retrouver progressivement les contraintes originelles en décroissant la température.

L'approche proposée a fait l'objet d'une expérimentation poussée grâce au développement d'un environnement de programmation dédié. L'homogénéité qui découle de la méthode de spécification utilisée et la robustesse de la méthode de résolution proposée nous ont permis la mise en œuvre d'un grand nombre de problèmes géométriques qui peuvent paraître à première vue de natures très différentes.

Annexe A

Primitives géométriques et relations de contact

A.1 Relations de contact et contraintes de contact

A.1.1 Point sur Point

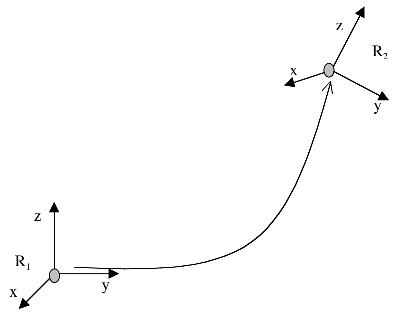

FIG. A.1 - *Contact Point sur Point.*

Une relation de type *Point sur Point* (figure A.1) induit les contraintes égalités suivantes :

$$\begin{cases} t_x & = & 0 \\ t_y & = & 0 \\ t_z & = & 0 \end{cases}$$

A.1.2 Point sur Arête

Une relation de type *Point sur Arête* (figure A.2) induit les contraintes suivantes :

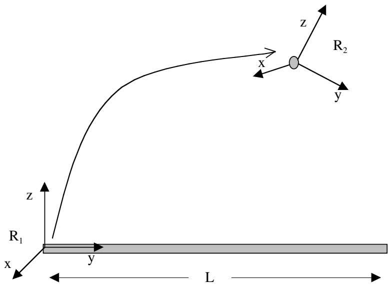

FIG. A.2 - *Contact Point sur Arête.*

$$\begin{cases} t_x & = & 0 \\ t_z & = & 0 \\ 0 & \leq & t_y & \leq & L \end{cases}$$

A.1.3 Point sur Face

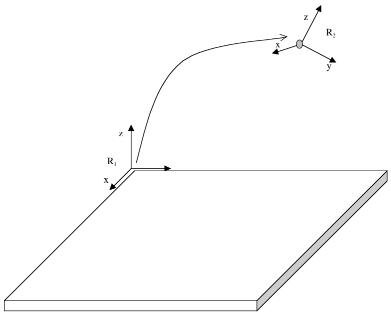

FIG. A.3 - *Contact Point sur Face.*

Un contact de type *Point sur Face* (figure A.3) d'un point pt sur une face F dont la liste des sommets est ${p_i}^{i=0..n}$ exprimés dans le repère attaché à F induit les contraintes :

$$\begin{cases} t_x & = & 0 \\ t_y & = & 0 \\ t_z & = & 0 \\ \forall i = 0 \cdots n, & v_i.z \leq 0 \end{cases}$$

avec :

$$v_i = (p_i - pt) \wedge (p_{i+1} - pt)$$

A.1.4 Point sur Cylindre

Une relation de type *Point sur Cylindre* (figure A.4) induit les contraintes suivantes :

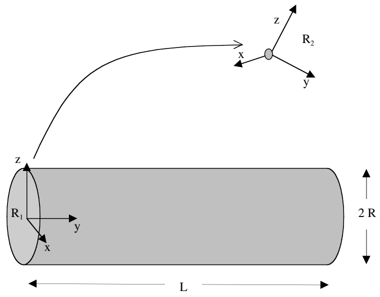

FIG. A.4 - *Contact Point sur Cylindre.*

$$\begin{cases} t_x^2 + t_z^2 & = & R^2 \\ 0 & \leq & t_y & \leq & L \end{cases}$$

A.1.5 Point sur Sphère

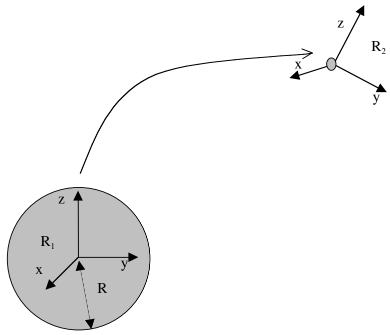

FIG. A.5 - *Contact Point sur Sphère.*

Une relation de type *Point sur Sphère* (figure A.5) induit la contrainte suivante égalité :

$$t_x^2 + t_y^2 + t_z^2 = R^2$$

A.1.6 Arête sur Arête

Une relation de type *Arête sur Arête* (figure A.6) induit les contraintes suivantes :

$$\begin{cases} t_x = 0 \\ t_z = 0 \\ r_x = 0 \\ r_y = 0 \\ 0 \leq t_y \leq L_1 - L_2 \end{cases}$$

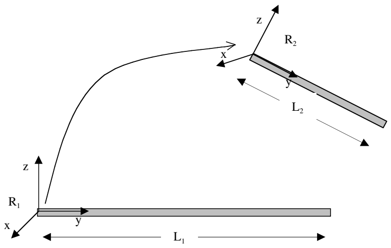

FIG. A.6 - *Contact Arête sur Arête.*

A.2 Passage d'une représentation en matrice homogène 4×4 à celle en 6-vecteur de pose et vice versa

Soit la matrice 4×4 de transformation T et le 6-vecteur de pose équivalent Q avec :

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

et

$$Q = (\rho \omega)^T$$

A.2.1 Passage d'un 6-vecteur de pose à une matrice de transformation 4×4

Le passage du 6-vecteur de pose Q à la matrice de transformation T (T $= mat(Q)$) est donné par les formules suivantes :

$$\begin{cases} t & = & \rho \\ R & = & \cos(a) I_3 + (1 - \cos(a)) u u^T - \sin(a) \{u\} \end{cases}$$

où :

$$a = ||\omega||$$

et

$$u = (u_1 u_2 u_3)^T = \frac{\omega}{a}$$

et

$$\{u\} = \begin{pmatrix} 0 & u_3 & -u_2 \\ -u_3 & 0 & u_1 \\ u_2 & -u_1 & 0 \end{pmatrix}$$

A.2.2 Passage d'une matrice de transformation 4×4 à un 6-vecteur de pose

Le passage de la matrice de transformation 4×4 T au 6-vecteur de pose Q ($Q = vect(T)$) est donné par les formules suivantes :

$$\begin{cases} \rho & = & t \\ \omega & = & a\frac{v}{\|v\|} \end{cases}$$

avec

$$a = \cos^{-1}(R_{11} + R_{22} + R_{33})$$

et

$$v = (R_{32} - R_{23} \quad R_{13} - R_{31} \quad R_{21} - R_{12})^T$$

et où :

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$$

Annexe B

Méthodes de tirage

Nous présentons dans ce qui suit quelques méthodes de tirage de Monte-Carlo. Pour plus de détails sur ces méthodes, le lecteur est encouragé à lire les articles de Neal [Neal93], MacKay [MacKay96] et de Tanner [Tanner98]. Nous présentons également une méthode de tirage direct à partir d'une distribution gaussienne monovaluée.

B.1 Tirage indirect de Monte-Carlo

B.1.1 Importance sampling

Importance sampling n'est pas une méthode pour générer des tirage à partir de la distribution $p(x)$; c'est juste une méthode pour estimer l'espérance de la fonction $g(x)$. Elle peut être vue comme une généralisation du tirage uniforme.

Pour illustrer cette méthode, prenons le cas d'une densité de dimension 1. On suppose qu'on soit capable d'évaluer cette densité pour tout x de l'espace d'état. Supposons aussi qu'on ne dispose pas de méthode directe pour effectuer des tirages à partir de cette distribution.

Supposons maintenant qu'on ait une autre densité de probabilité plus simple $q(x)$ qu'on peut évaluer pour toutes les valeurs de x, et à partir de laquelle on est capable de générer directement des tirages. On appellera cette densité *q la distribution de tirage*.

Pour cette méthode, on génère N tirages $\{x_q^{(i)}\}_{i=1}^N$ à partir de $q(x)$. Si ces points avaient été tirés à partir de $p(x)$, nous pourrions estimer I en utilisant l'équation 3.9. Mais comme ces tirages ont été générés à partir de $q(x)$ et non à partir de $p(x)$, les valeurs de x pour lesquelles $q(x)$ est supérieure à $p(x)$ vont être sur-représentées, et les valeurs pour lesquelles $q(x)$ est inférieure à $p(x)$ vont être sous-représentées. Pour prendre en compte ce fait, une notion de *poids* est introduite

$$w_i = \frac{p(x_q^{(i)})}{q(x_q^{(i)})} \tag{B.1}$$

Ces poids sont utilisés pour ajuster l'*importance* de chaque point dans l'estimateur

$$\hat{I} = \frac{\sum_i w_i g(x_q^{(i)})}{\sum_i w_i} \tag{B.2}$$

La figure B.1 montre un exemple de fonctions $p(x)$ et $q(x)$:

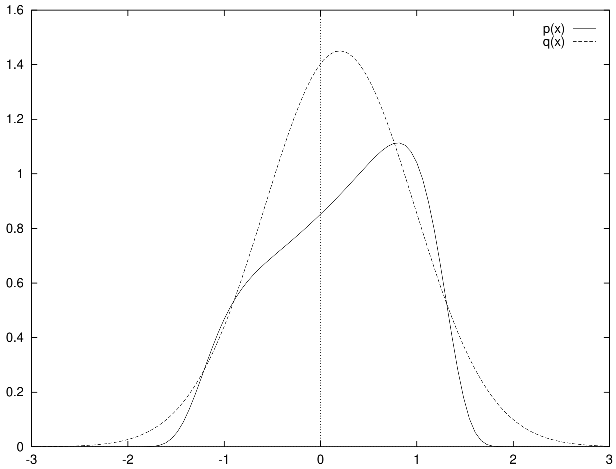

FIG. B.1 - *Exemple de fonctions $p(x)$ et $q(x)$: $p(x) = e^{0.4(x-0.4)-0.2x^6}$ et $q(x) = 4Normal(0.2, 1.21)$*

B.1.2 Le tirage avec rejet (Rejection sampling)

Prenons encore le cas d'une densité $p(x)$ de dimension 1 pour laquelle aucune méthode de tirage direct n'existe. Supposons qu'on dispose d'une densité plus simple $q(x)$ de laquelle on peut générer directement des tirages avec

$$\exists c,\ \forall x, cq(x) > p(x) \tag{B.3}$$

On génère un point x_q à partir de la densité de proposition $q(x)$ et on l'accepte avec une probabilité de $\frac{cq(x_q)}{p(x_q)}$:

1. évaluer $cq(x_q)$
2. générer un nombre aléatoire u uniformément distribué dans l'intervalle $[0, cq(x_q)]$
3. si $p(x) > u$ alors le point x_q est accepté, sinon il est rejeté.

La figure B.2 montre un exemple de fonctions $p(x)$ et $q(x)$:

Un autre exemple de cette méthode est donné par la figure B.3. On cherche à calculer empiriquement la surface d'un cercle de rayon R. Le calcul de cette surface consiste à estimer l'intégrale $\int\int dxdy$ sous la contrainte $x^2 + y^2 < R^2$. Un ensemble de points $\{(x_i, y_i)\}_{i=1}^N$ est tiré. Le tirage pour x et y est fait séparément à partir de la distribution uniforme sur $[0, R]$.

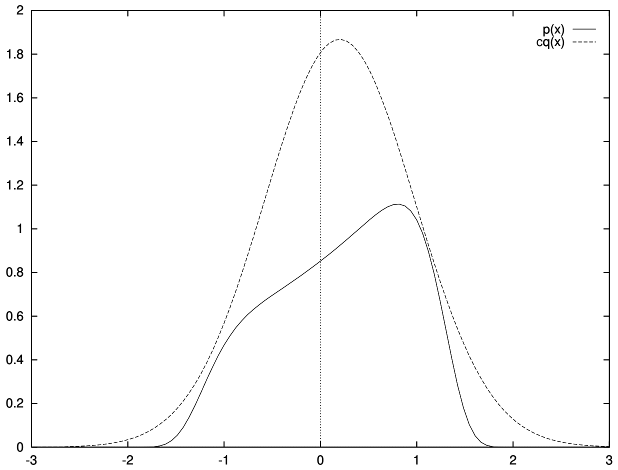

FIG. B.2 - *Exemple de fonctions* $p(x)$ *et* $q(x)$ *:* $p(x) = e^{0.4(x-0.4)-0.2x^6}$, $c = 5.15$ *et* $q(x) = Normal(0.2, 1.21)$.

La fonction de proposition $q(x, y)$ est la distribution uniforme sur $[0, R] \times [0, R]$. La fonction de tirage $p(x, y)$ quant à elle est donnée par :

$$\begin{cases} 1 & \text{si } x^2 + y^2 < R^2 \\ 0 & \text{sinon} \end{cases}$$

La surface de notre cercle va être : $R^2 \frac{N_v}{N}$ où N_v est le nombre de points valides et N est le nombre total des points tirés.

Il est clair que cette méthode marche bien quand la densité q est une bonne approximation de p. Dans le cas contraire, le taux de rejet va devenir très grand.

B.1.3 L'algorithme de Metropolis

Les méthodes qu'on vient de voir ne marchent bien que si la densité de proposition $q(x)$ est une bonne approximation de $p(x)$. Pour les problèmes complexes (surtout pour les espaces à grandes dimensions), il est souvent difficile de trouver de telles densités de probabilité.

L'algorithme de Metropolis utilise un processus Markovien dans lequel une séquence d'états $x^{(t)}$ est générée. Chaque nouvel état dépend de l'état précédent. Cet algorithme est un exemple des méthodes de "Markov Chain Monte Carlo" (MCMC).

Au lieu d'utiliser une seule "proposal" densité $q(x)$, l'algorithme de Metropolis utilise une densité qui dépend de l'état actuel $x^{(t)}$. La densité $q(x_c, x^{(t)})$ n'est pas forcément une bonne approximation de $p(x)$. Elle peut être une distribution très simple comme une gaussienne centrée en $x^{(t)}$ par exemple.

Etant dans un état $x^{(t)}$, un candidat x_c est généré à partir de $q(x_c, x^{(t)})$. Pour accepter ou pas le nouveau candidat x_c, on calcule

$$a = \frac{p(x_c)q(x_c, x^{(t)})}{p(x^{(t)})q(x^{(t)}, x_c)} \tag{B.4}$$

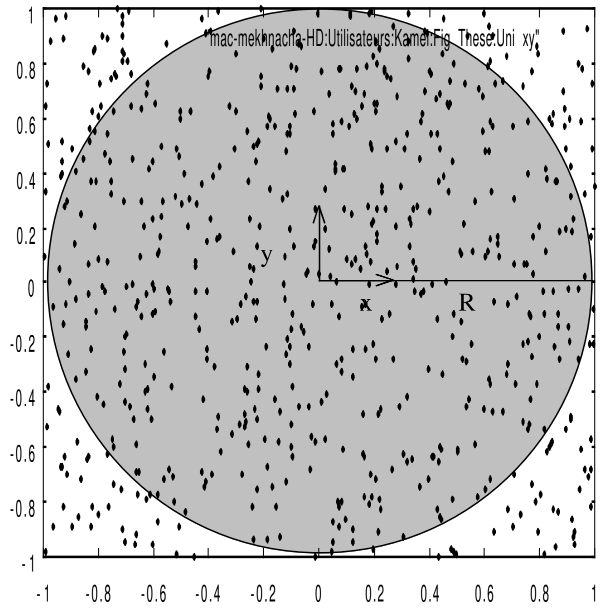

FIG. B.3 - *Les points qui sont à l'extérieur du cercle ($x^2 + y^2 > R^2$) ne sont pas pris en compte pour le calcul de la surface.*

Si $a > 1$ alors x_c est accepté, sinon x_c est accepté avec une probabilité de a.

Si x_c est accepté, le nouvel état sera $x^{(t+1)} = x_c$. Si x_c n'est pas accepté, le nouvel état sera $x^{(t+1)} = x^{(t)}$. Le rejet du candidat x_c engendre la réutilisation du point courant $x^{(t)}$.

Si la densité $q(x_c, x^{(t)})$ est symétrique par rapport à x_c et $x^{(t)}$ (une gaussienne centrée en $x^{(t)}$ par exemple), alors $\frac{q(x_c,x^{(t)})}{q(x^{(t)},x_c)} = 1$ et a devient

$$a = \frac{p(x_c)}{p(x^{(t)})} \tag{B.5}$$

Il existe deux variantes de l'algorithme de Metropolis :

- L'algorithme local, dans lequel les composantes du vecteur d'état x sont mises à jour une à la fois. Chaque composante x_i est mise à jour en utilisant une "proposal" q_i qui lui est propre, centrée sur sa valeur courante.

- L'algorithme global, dans lequel toutes les composantes sont mise à jour simultanément en utilisant une "proposal" centrée sur l'état courant.

B.1.4 Le tirage de Gibbs

Le tirage de Gibbs est une autre méthode MCMC pour effectuer des tirages à partir d'une distribution de probabilité $p(x)$ de dimension supérieure ou égale à 2. Cette méthode peut être vue comme une méthode de Metropolis avec une "proposal" distribution q qui est définie en terme de distributions conditionnelles de la distribution conjointe $p(x)$. On suppose que la distribution $p(x)$ est trop complexe pour pouvoir générer directement des tirages à partir d'elle, et qu'on est capable, par contre, de générer directement des tirages à partir des distributions conditionnelles $p(x_i|\{x_j\}_{j\neq i})$.

Pour un système avec N variables, une itération met à jour un seul paramètre x_i à la fois, en effectuant un tirage à partir de $p(x_i|\{x_j\}_{j\neq i})$

$$\begin{aligned} x_1^{(t+1)} &\sim p(x_1|x_2^{(t)}x_3^{(t)}\cdots x_N^{(t)}) \\ x_2^{(t+1)} &\sim p(x_2|x_1^{(t)}x_3^{(t)}\cdots x_N^{(t)}) \\ &\vdots \\ x_N^{(t+1)} &\sim p(x_N|x_1^{(t)}x_2^{(t)}x_3^{(t)}\cdots x_{N-1}^{(t)}) \end{aligned}$$

On peut voir le tirage de Gibbs comme un tirage de Metropolis dans lequel le condidat est toujours accepté.

B.2 Tirage direct à partir d'une distribution gaussienne monovariée

Disposant d'un générateur de nombres aléatoires uniformément distribués, il existe plusieurs méthodes pour générer des nombres aléatoires distribués suivant une distribution Gaussienne (Normale) (voir par exemple [Rubinstein81] pour plus de détails). Nous présentons dans ce paragraphe une de ces méthodes de tirage.

Plus généralement, dès lors que l'on dispose d'une équation décrivant la distribution à partir de laquelle on voudrait générer des tirages, il est possible d'utiliser une fonction de transformation. Cette fonction de transformation prend des nombres distribués suivant une distribution donnée comme entrée et donne des nombres distribués suivant une nouvelle distribution comme sortie. La plus importante de ces fonctions de transformation est celle connue sous le nom de "transformation de Box-Muller" [Box58]. Elle permet de transformer un ensemble de nombres aléatoires uniformément distribués, en un autre ensemble de nombres aléatoires distribués suivant une Gaussienne (loi normale).

La forme polaire de cette transformation est donnée par l'algorithme suivant permettant de générer un couple (p_1, p_2) de nombres aléatoires normalement distribués :

Tirage_Gauss(μ, σ)
DEBUT
    REPETER
        $v_1 = Random(-1.0, 1.0)$
        $v_2 = Random(-1.0, 1.0)$
        $w_1 = v1^2 + v2^2$
    JUSQU'A ($w_1 < 1.0$)
    $w_2 = (-2.0 \log(w_1))/w_1)^{1/2}$
    $z_1 = v_1 w_2$

$z_2 = v_2 w_2$
$p_1 = \mu + z_1 \sigma$
$p_2 = \mu + z_2 \sigma$
RETOURNER (p_1, p_2)
FIN

où la fonction $Random(a, b)$ permet de générer un nombre aléatoire uniformément distribué entre a et$b$, et μ et σ sont respectivement la moyenne et l'écart-type de la Gaussienne.

Annexe C

Optimisation des calculs pour la méthode d'intégration

L'utilisation d'une méthode de tirage locale pour la mise à jour du vecteur d'état L permet de réduire le coût du calcul de l'estimation. Si l'on dénote $F_i^{(t)}(X), \quad i = 1 \cdots p$ les valeurs des fonctions $F_i(X)$ pour un point donné L^t, le calcul des valeurs $F_i^{(t+1)}(X), \quad i = 1 \cdots p$ est une mise à jour des valeurs $F_i^{(t)}(X), \quad i = 1 \cdots p$. Le coût de mise à jour dépend de la position de l'arc correspondant à la composante $L_{i_k j_k}$ qui vient d'être changée dans les cycles $c_i, \quad i = 1 \cdots p$. En particulier, la mise à jour d'un arc ne modifie que les valeurs des fonctions correspondant aux cycles qui contiennent l'arc en question.

Nous présentons dans cette annexe l'algorithme permettant d'optimiser les calculs des intégrales pour un problème donné.

Nous mettons à jour pour chaque cycle $c_i, \ i = 1 \cdots p$ du graphe et pour chaque arc (S_j, S_{j+1}) de ce cycle les structures de données suivantes (figure C.1) :

- La matrice 4×4 de transformation $T_{j\ j+1}$ (de passage) donnant la position de S_{j+1} par rapport à S_j.

- Une matrice 4×4 de transformation $TG^i_{j\ 0}$ donnant la position du repère S_j par rapport au premier repère S_0 du chemin reliant S_0 au repère S_{k_i}, avec k_i le nombre de sommets appartenant au cycle c_i.

Pour un cycle c_i, la matrice $TG^i_{k_i\ 0}$ donnant la position du repère S_{k_i} par rapport à S_0 permet de calculer directement la fonction $F_i(X, L)$. Pour des valeurs données de L et de X, on a tout simplement :

$$F_i(X, L) = f_i = vect(TG^i_{k_i\ 0})$$

Il est clair qu'en utilisant cette méthode, la mise à jour d'une matrice $TG^i_{j\ 0}$ donnée n'est nécessaire que si l'arc qu'on vient de mettre à jour vient avant l'arc A_{S_{j-1}, S_j} dans le chemin reliant le repère S_o au repère S_{k_i}.

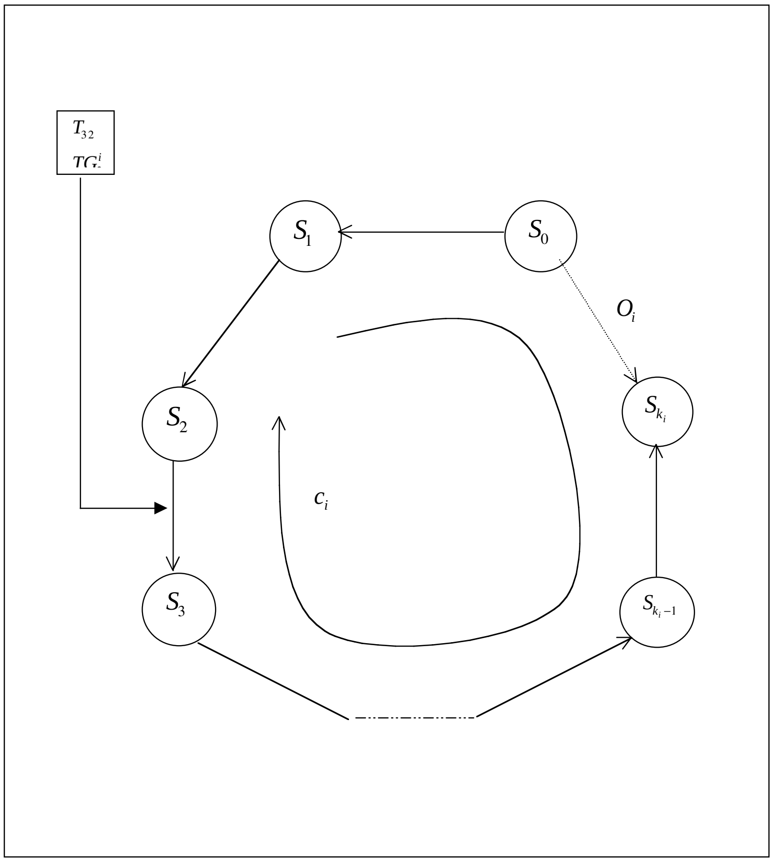

FIG. C.1 - *Structures de données pour la mise à jour locale de la fonction objective.*

Par conséquent, la mise à jour d'un paramètre appartenant à un lien représenté par l'arc $A_{S_j,S_{j+1}}$ nécessite :

1. la modification de la matrice de passage $T_{j\ j+1}$
2. la mise à jour de toutes les matrices $TG^i_{d\ 0},\ d > j$

L'approximation de l'intégrale $I(X)$ en utilisant un nombre de point N_points consiste alors à boucler sur l'ensemble des composantes $\{L_{i_k j_k}\}_{k=1}^m$ et les mettre à jour une à une. La mise à jour d'un vecteur appartenant à un arc $A_{S_j,S_{j+1}}$ nécessite la mise à jour des matrices $TG^i_{d\ 0},\ d > j,\ i = 1 \cdots p$ pour tous les cycles c_i contenant l'arc à lequel le paramètre mis à jour appartient.

On peut résumer cette méthode calcul par l'algorithme suivant :

Estimer_Integrale (N_points)
DEBUT
$np \leftarrow 0$
$somme \leftarrow 0.0$
$Arc\_a\_mettre\_a\_jour \leftarrow 0$
TANTQUE $np < N\_points$ FAIRE
$res \leftarrow Maj\_Arc(Arc\_a\_mettre\_a\_jour)$
$somme \leftarrow somme + res$
$Arc\_a\_mettre\_a\_jour \leftarrow (Arc\_a\_mettre\_a\_jour + 1) mod(m)$
$np \leftarrow np + 1$
FINTANTQUE
RETOURNER $somme/N\_points$
FIN

Maj_Arc (a)
DEBUT
Tirer une valeur de $L_{i_a j_a}$ suivant la procédure décrite dans la section 3.2.1 du chapitre 3
Mettre à jour la matrice de passage $T_{a\ a+1}$: $T_{a\ a+1} \leftarrow mat(L_{i_a j_a})$
POUR tous les cycles c_i i = 1 JUSU'A p contenant l'arc a FAIRE
POUR $d = a + 1$ JUSQU'A k_i FAIRE
$TG^i_{d\ 0} \leftarrow TG^i_{d-1\ 0} * T^{sens_{d-1\ d}}_{d-1\ d}$
FINPOUR
$f_i \leftarrow vect(TG^i_{k_i\ 0})$
FINPOUR
RETOURNER $p_{O_1}(f_1)p(H_{m-p+1}|f_1)$
$p_{O_2}(f_2)p(H_{m-p+2}|f_2)$
$\vdots$
$p_{O_p}(f_p)p(H_m|f_p)$
FIN

avec $sens_{d-1\ d} \in \{-1, 1\}$ est le sens du parcours de l'arc (S_{d-1}, S_d).

Bibliographie

[Ahuactzin94] J. M. Ahuactzin Larios. *Le Fil d'Ariane : Une méthode de planification générale. Application à la planification automatique de trajectoires.* Thèse de l'INP de Grenoble, septembre 1994.

[Alami94] R. Alami and T. Simeon. *Planning Robust Motion strategies for mobile robots.* International Conference on Robotics and Automation, Volume 2 page 1312-1318, San Diego, Californie, 1994.

[Bekerman92] M. Bekerman. *A Bayes-Maximum Entropy method for multi-sensor data fusion.* Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992.

[Bessière98a] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer et K. Mekhnacha. *Interprétation vs. Description I : Proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs.* Intellectica, 1998/1-2, 26-27, pp. 257-311.

[Bessière98b] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer et K. Mekhnacha. *Interprétation vs. Description II : Fondements mathématiques.* Intellectica, 1998/1-2, 26-27, pp. 313-336.

[Boissonnat88] J. D. Boissonnat. *Complexité géométrique et robotique. Technique de la Robotique - Perception et planification.* Hermès, 1988.

[Borrel85] P. Borrel, E. Dombre, A. Liégois, F. Bernard et D. Bourcier. *La fonction robotique dans le système CATIA. État de la Robotique en France.* Hermès, 1985.

[Bouilly96] B. Bouilly , T. Simeon , R. Alami. *A Numerical Technique for Planning Motion Strategies for a Mobile Robot in Presence of Uncertainties.* IEEE International Conference on Robotics and Automation (ICRA'95), Nagoya (Japon), pp.1327-1332, Mai 1995.

[Box58] G. E. P. Box and M. E. Muller 1958. *A note on the generation of random normal deviates.* Annals Math. Stat, V. 29, pp. 610-611, 1958.

[Celaya92] E. Celaya. *LMF : a program for positioning objects using geometric relationships.* 92' International Conference on Applications of Artificial Intelligence in Engineering, 1992.

[Cox46] R. T. Cox. *Probability, Frequency, and Reasonable Expectation.* American Journal of Physics, No17, 1946.

[Cox61] R. T. Cox. *The algebra of probable inference.* The John Hopkins Press, Baltimore, 1961.

[Dedieu95] E. Dedieu. *La représentation contingente, vers une réconciliation des approches fonctionnelles et structurelles de la robotique autonome.* Thèse de l'INP de Grenoble, septembre 1995.

[DeLaRos96] F. De la Rosa. *Planification de mouvements avec prise en compte explicite des incertitudes géométriques.* Thèse de l'INP de Grenoble, 1996.

[Dellaert99] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. *Monte Carlo Localization for Mobile Robots.* IEEE International Conference on Robotics and Automation (ICRA99), 1999.

[Falkenauer91] E. Falkenauer an S. Bouffouix. *A Genetic Algorithm for Job Shop.* Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, April 1991.

[Faver88] B. Faverjon et P. Tournassoud. *Techniques de la Robotique - Perception et planification, chapitre 7.* Hermès, 1988.

[Fox99] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. *Monte Carlo Localization: Efficient Position Estimation for Mobile Robots.* Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99), 1999.

[Gaunet94] D. GAUNET. *Modèle formel de tolérancement de position : Contribution à l'aide au tolérancement des mécanismes en CFAO.* Thèse de Doctorat de l'école normale supérieure de Cachan, février 1994.

[Geweke95] J. Geweke. *Monte Carlo Simulation and Numerical Integration.* Federal Reserve Bank of Minneapolis, Research Department Staff Report 192, May 1995.

[Giraud91] A. Giraud and T. Tran. *Kinematic modeling of tasks under constraints. Application to fine motion and calibration.* Chapter 2, pages 89-98, Springer-Verglas, 1991.

[Gondran90] M. Gondran et M. Minoux. *Graphes et algorithmes.* Eyrolle, 1990.

[Grefenstette88] J. J. Grefenstette. *Credit assignment in rule discovery systems based on genetic algorithms.* Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, D.C. in Machine learning, Kluwer Academic Publishers, 1988.

[Halpern99] J. Y. Halpern. *A conterexample to Theorems of Cox and Fine.* Journal of Artificial Intelligence Research 10, pp. 67-85, 1999.

[Holland75] J. H. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor : University of Michigan Pres, 1975.

[Jaakkola99] T. S. Jaakkoda and M. I. Jordan. *Variational Probabilistic Inference and the QMR-DT Network.* Journal of Artificial Intelligence Research 10, pages 291-322, May 1999.

[Jaynes96] E. Jaynes. *Probability Theory: the Logic of Science.* Partial version available at "bayes.wusl.edu"(ftp).

[Kato94] Z. Kato. *Modélisations Markoviennes multirésolutions en vision par ordinateur. Application à la segmentation d'images Spot.* Thèse dÉtat de l'université de Nice-Sofia Antipolis, décembre 1994.

[Keller96] A. Keller. *The fast Calculation of Form Factors using Low Discrepancy Point Sequences.* Proceedings of the 12th Spring Conference on Computer Graphics, pp. 195 - 204,Bratislava, 1996.

[Kenneth88] Kenneth de Jong. *Learning with genetic algorithms.* Computer Science Department, George Mason University, in Machine Learning, Kluwer Academic Publishers, 1988.

[Kirkpatrick83] S. Kirkpatrick, C. Gellat and M. Vecchi. *Optimization by simulated annealing.* Science 220, pp 671-680, 1983.

[Laarhoven87] P. V. Laarhoven an E. Aarts. *Simulated annealing : Theory and applications.* Reidel Pub., Dordrecht, Holland, 1987.

[Latombe91] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publisher, 1991.

[Lebeltel97] O. Lebeltel, P. Bessière, E. Dedieu and E. Mazer. *Modélisation Bayésienne d'Interaction Robot/Environnement Téléopérées.* Proceedings of SISOM'97 (Annual Symposium of the Institute of Solid Mechanics), ISSN 1223-7140, Bucharest, 1997, pp. 1-8.

[Lieberman77] L. I. Lieberman and M. A. Wesley. *AUTOPASS : an automatic programming system for computer controlled mechanical assembly.* IBM J.Res.Dev. 21,4, 1977.

[Lowe89] D. G. Lowe. *Review of "TINA: The Sheffield AIVRU vision system". by J. Porill et al.* In O.Khatib, J. Craig, and T. Lozano-Pèrez, editors, The Robotics Review I, pages 195-198. MIT Press, Cambridge, MA, 1989.

[LozanoPèrez87] T. Lozano-Pèrez. *A Simple Motion-Planning Algorithm for General Robot Manipulators.* IEEE Transactions on Robotics and Automation, *3*(3), 224–238.

[MacKay96] D. G. C. MacKay. *Introduction to Monte Carlo Methods.* Proceedings of an Erice summer school, ed. M.Jordan, 1996.

[Mantripragada98] R. Mantripragada and D. E. Whitney. *Modeling and Controlling Variation in Mechanical Assemblies using State Transition Models.* Proceedings of the 1998 IEEE International Conference on Robotics and Automation Leuven, Belgium, May 1998.

[Mason85] M.T. Mason and J.K Salisbury. *Robot Hands and the mechanics of manipulation.* MIT press 1985.

[Mazer87] E. Mazer. *Handey : un modèle de planificateur pour la programmation automatique des robots.* Thèse d'État, INP de Grenoble, décembre 1987.

[Moravec88] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. AI Magazine, 9(2):61-74, 1988.

[Murray94] R.Murray, X.Li and Sastry. *A mathematical introduction to robotic manipulation.* CRC press 1994.

[Neal93] R. M. Neal. *Probabilistic inference using Marcov Chain Monte Carlo Methods.* Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

[Pearl91] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann Publisher, San Mateo, California, USA, 1991.

[Piccin95] O. Piccin. *Spécification et résolution de tâches de manipulation complexes. Application à la téléprogrammation de robots distants.* Thèse de l'Université Paul Sabatier de Toulouse, novembre 1995.

[Puget89] P. Puget. *Vérification-Correction de programme pour la prise en compte des incertitudes en programmation automatique des robots.* Thèse INP de Grenoble, février 1989.

[Rubinstein81] R. Y. Rubinstein. *Simulation and the Monte Carlo method.* John Wiley and Sons, ISBN 0-471-08917-6, 1981.

[Rudolph94] G. Rudolph. *Convergence Analysis of Canonical Genetic Algorithms.* IEEE Transactions on Neural Networks, Vol.5 No. 1, January 1994.

[Samson91] C. Samson, M. Le Borgne and B. Espiau. *The Task Function Approach.* Clarendon Press, Oxford 1991.

[Sanderson97] A. C. Sanderson. *Assemblability based on Maximum Likelihood Configuration of Tolerances.* Presented at IEEE Symposium on Assembly and Task Planning, Marina del Rey, CA., August, 1997.

[Tanner98] M. A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* Springer Series in Statistics, 3rd edition, Springer Verglas, 1998.

[Taylor76] R. H. Taylor. *A synthesis of manipulator control programs from task-level specifications.* memo AIM 282, Stanford University, Computer Science Department, July 1976.

[Troccaz86] J. Pertin-Troccaz. *Modélisation du raisonnement géométrique pour la programmation des robots.* PhD thesis, INPG, mars 1986.

[Weiss98] Y. Weiss and E. H. Adelson. *Slow and Smooth: a Bayesian theory for the combination of local motion signals in human vision.* MIT AI Memo 1624, February, 1998.

[Wesley80] M. A. Wesley, T. Lozano-Pèrez, L. I. Lieberman, M. A. Lavin and D. D. Grossman. *A geometric Modeling System for Automated Mechanical Assembly.* IBM J. Res. Dev, Vo. 24, No 1, January 1980.

[Whitney94] D. E. Whitney, O. L. Gilbert, and M. Jastrzebski. *Representation of Geometric Variations using Matrix Transforms for Statistical Tolerance Analysis in Assemblies.* Research in Engineering Design, vol. 6, pp. 191-210, 1994.

[Zhang92] Z. Zhang and O. augeras. *3D Dynamic Scene Analysis: A sterio Based Approach.* Springers, Berlin, Heideberg, 1992.